

三菱電機 FA 統合コントローラ

MELSEC MX Controller

MELSEC MXコントローラ(MX-Rモデル)
プログラミングマニュアル

安全上のご注意

(ご使用前に必ずお読みください)

三菱電機シーケンサのご使用に際しては、各製品のマニュアルおよび各製品のマニュアルで紹介している関連マニュアルをよくお読みいただくと共に、安全に対して十分に注意を払って、正しい取扱いをしていただくようお願いいたします。

本マニュアルは必要なときに読めるよう大切に保管すると共に、必ず最終ユーザまでお届けいただくようお願いいたします。

はじめに

このたびは三菱電機シーケンサをお買い上げいただき、まことにありがとうございました。

本マニュアルは、プログラミングを行う場合に必要な命令、ファンクションおよびファンクションブロックについて、ご理解いただくためのマニュアルです。

ご使用前に本マニュアルや関連マニュアルをよくお読みいただき、コントローラの機能・性能を十分ご理解の上、正しくご使用くださるようお願いいたします。

また、本マニュアルで紹介するプログラム例を実際のシステムへ流用する場合は、対象システムにおける制御に問題がないことを十分検証ください。

本マニュアルにつきましては最終ユーザまでお届けいたしますよう、宜しく願い申し上げます。

目次

安全上のご注意	1
はじめに	2
関連マニュアル	20
用語	21
総称/略称	24

第1部 プログラミング概要

第1章 命令概要	26
1.1 命令の構成	26
1.2 データの指定方法	28
ビットデータ	31
16ビットデータ(ワードデータ)	32
32ビットデータ(ダブルワードデータ)	35
実数データ(浮動小数点データ)	38
文字列データ	41
1.3 実行条件	43
1.4 プログラミング時の注意事項	44
命令共通のエラー	44
命令実行時のデバイス、ラベルの範囲チェック	44
ロングタイム、ロング積算タイムデバイス使用時の動作	50
同期局に割り当てたリンクデバイスを命令引数に使用する場合の注意事項	52
同一デバイスのOUT命令、SET/RST命令、PLS/PLF命令使用時の動作	52
ファイルレジスタ使用時の制約事項	58
第2章 プログラム部品	59
2.1 プログラムブロック	60
2.2 ファンクション(FUN)	61
2.3 ファンクションブロック(FB)	65
2.4 注意事項	76
ファンクションを使用する場合	76
ファンクションブロックを使用する場合	76
ENの制御にMC/MCR命令を使用する場合の動作	80
第3章 ラダー言語	85
3.1 構成	85
回路記号	86
プログラム実行順序	87
ラダー言語でファンクションブロックを使用する場合の注意事項	87
3.2 インラインST	88
3.3 ステートメント/ノート	90
第4章 ST言語	91
4.1 構成	92
区切り記号	93
演算子	93

命令	94
構文	95
定数	104
ラベルとデバイス	105
コメント	107

第5章 FBD/LD言語 108

5.1 構成.....	108
部品	109
定数	116
ラベルとデバイス	116
5.2 インラインST	118
5.3 プログラム実行順序.....	119
部品の実行順序	119

第2部 命令/FUN/FB一覧

第6章 命令一覧 122

6.1 シーケンス命令	122
6.2 基本命令.....	126
6.3 応用命令.....	145
プログラム制御	145
データ処理.....	147
デバッグ, 故障診断.....	154
文字列処理.....	154
実数処理	157
乱数	164
デバイス操作.....	165
タイマ, カウンタ	166
時計	167
ユニットアクセス	170
PID演算命令.....	172
PID制御命令.....	172
アドオン用命令	173

第7章 ネットワーク命令一覧 174

7.1 Ethernet用命令	174

第8章 汎用FUN/汎用FB一覧 176

8.1 汎用ファンクション.....	176
8.2 汎用ファンクションブロック	187

第9章 モーション制御ファンクションブロック一覧 189

9.1 管理系ファンクションブロック	189
9.2 動作系ファンクションブロック	192
9.3 一般ファンクションブロック	196

第10章 ネットワークファンクションブロック一覧 197

10.1 Ethernet用ファンクションブロック	197
10.2 CC-Link IE TSN用ファンクションブロック	198

第3部 シーケンス命令/基本命令

第11章 シーケンス命令 200

11.1 接点命令	200
演算開始, 直列接続, 並列接続	200
パルス演算開始, パルス直列接続, パルス並列接続	202
パルス否定演算開始, パルス否定直列接続, パルス否定並列接続	205
11.2 結合命令	208
回路ブロック直列接続, 並列接続	208
演算結果反転	209
演算結果パルス化	210
エッジリレー演算結果パルス化	211
11.3 出力命令	213
アウト(タイマ, カウンタ, アナシエータを除く)	213
タイマ	215
ロングタイマ	218
カウンタ	221
ロングカウンタ	223
アナシエータ	225
デバイスのセット(アナシエータを除く)	226
デバイスのリセット(アナシエータを除く)	228
アナシエータのセット	230
アナシエータのリセット	232
立上り出力	234
立下り出力	236
ビットデバイス出力反転	238
ダイレクト出力のパルス化	239
11.4 シフト命令	241
ビットデバイスシフト	241
11.5 マスタコントロール命令	243
マスタコントロールのセット, リセット	243
11.6 終了命令	247
メインルーチンプログラム終了	247
シーケンスプログラム終了	248
11.7 停止命令	250
シーケンスプログラム停止	250
11.8 無処理命令	251
無処理(NOPLF)	251

第12章 基本命令 252

12.1 比較演算命令	252
BIN16ビットデータ比較	252
BIN32ビットデータ比較	254
BIN16ビットブロックデータ比較	256
BIN32ビットブロックデータ比較	259

12.2	算術演算命令	262
	BIN16ビット加算	262
	BIN16ビット減算	266
	BIN32ビット加算	270
	BIN32ビット減算	274
	BIN16ビット乗算	278
	BIN16ビット除算	280
	BIN32ビット乗算	282
	BIN32ビット除算	284
	BCD4桁加算	286
	BCD4桁減算	289
	BCD8桁加算	292
	BCD8桁減算	295
	BCD4桁乗算	298
	BCD4桁除算	300
	BCD8桁乗算	302
	BCD8桁除算	304
	BIN16ビットブロックデータ加算	306
	BIN16ビットブロックデータ減算	308
	BIN32ビットブロックデータ加算	310
	BIN32ビットブロックデータ減算	313
	16ビットBINデータインクリメント	316
	16ビットBINデータデクリメント	317
	32ビットBINデータインクリメント	318
	32ビットBINデータデクリメント	319
12.3	論理演算命令	321
	16ビットデータ論理積	321
	32ビットデータ論理積	324
	16ビットブロックデータ論理積	328
	16ビットデータ論理和	330
	32ビットデータ論理和	333
	16ビットブロックデータ論理和	337
	16ビットデータ排他的論理和	339
	32ビットデータ排他的論理和	342
	16ビットブロックデータ排他的論理和	346
	16ビットデータ否定排他的論理和	348
	32ビットデータ否定排他的論理和	351
	16ビットブロックデータ否定排他的論理和	355
12.4	ビット処理命令	357
	ワードデバイスのビットセット	357
	ワードデバイスのビットリセット	359
	16ビットテスト	361
	32ビットテスト	363
	ビットデバイスの一括リセット	365
12.5	データシフト命令	367
	BIN16ビットデータのnビット右シフト	367
	BIN16ビットデータのnビット左シフト	369
	nビットデータの1ビット右シフト	371
	nビットデータの1ビット左シフト	373
	nワードデータの1ワード右シフト	375
	nワードデータの1ワード左シフト	377

nビットデータのnビット右シフト	379
nビットデータのnビット左シフト	381
nワードデータのnワード右シフト	383
nワードデータのnワード左シフト	385
12.6 データ変換命令	387
BINデータ→BCD4桁変換	387
BINデータ→BCD8桁変換	389
BCD4桁→BIN16ビットデータ変換	391
BCD8桁→BIN32ビットデータ変換	393
単精度実数→符号付きBIN16ビットデータ	395
単精度実数→符号なしBIN16ビットデータ	397
単精度実数→符号付きBIN32ビットデータ	399
単精度実数→符号なしBIN32ビットデータ	401
倍精度実数→符号付きBIN16ビットデータ	403
倍精度実数→符号なしBIN16ビットデータ	405
倍精度実数→符号付きBIN32ビットデータ	407
倍精度実数→符号なしBIN32ビットデータ	409
符号付きBIN16ビットデータ→符号なしBIN16ビットデータ変換	411
符号付きBIN16ビットデータ→符号付きBIN32ビットデータ変換	413
符号付きBIN16ビットデータ→符号なしBIN32ビットデータ変換	415
符号なしBIN16ビットデータ→符号付きBIN16ビットデータ変換	417
符号なしBIN16ビットデータ→符号付きBIN32ビットデータ変換	419
符号なしBIN16ビットデータ→符号なしBIN32ビットデータ変換	421
符号付きBIN32ビットデータ→符号付きBIN16ビットデータ変換	423
符号付きBIN32ビットデータ→符号なしBIN16ビットデータ変換	425
符号付きBIN32ビットデータ→符号なしBIN32ビットデータ変換	427
符号なしBIN32ビットデータ→符号付きBIN16ビットデータ変換	429
符号なしBIN32ビットデータ→符号なしBIN16ビットデータ変換	431
符号なしBIN32ビットデータ→符号付きBIN32ビットデータ変換	433
BIN16ビットデータ→グレイコードデータ変換	435
BIN32ビットデータ→グレイコードデータ変換	437
BIN16ビットグレイコードデータ→BIN16ビットデータ変換	439
BIN32ビットグレイコードデータ→BIN32ビットデータ変換	441
BIN16ビットデータブロック→BCD4桁データ変換ブロック	443
ブロックBCD4桁データ→ブロックBIN16ビット変換データ	445
10進アスキーデータ→BIN16ビットデータ変換	447
10進アスキーデータ→BIN32ビットデータ変換	450
16進アスキーデータ→BIN16ビットデータ変換	454
16進アスキーデータ→BIN32ビットデータ変換	456
10進アスキーデータ→BCD4桁データ変換	459
10進アスキーデータ→BCD8桁データ変換	462
10進文字列→BIN16ビットデータ変換	465
10進文字列→BIN32ビットデータ変換	468
16進アスキー→16進データBIN変換	471
単精度実数→BCD分解	473
BIN16ビットデータ2の補数(符号反転)	475
BIN32ビットデータ2の補数(符号反転)	476
8→256ビットデコード	477
256→8ビットエンコード	479
7セグメントデコード	481
BIN16ビットデータの4ビット分離	484

	BIN16ビットデータの4ビット結合	486
	任意データのビット分離.....	488
	任意データのビット結合.....	490
	バイト単位データ分離.....	492
	バイト単位データ結合.....	494
12.7	データ転送命令	496
	BIN16ビットデータ転送	496
	BIN32ビットデータ転送	498
	BIN16ビットデータ否定転送.....	500
	BIN32ビットデータ否定転送.....	502
	1ビットデータ反転転送.....	504
	BIN16ビットデータブロック転送(16ビット).....	506
	BIN16ビットデータブロック転送(32ビット).....	508
	同一BIN16ビットデータブロック転送(16ビット).....	510
	同一BIN16ビットデータブロック転送(32ビット).....	512
	同一BIN32ビットデータブロック転送(16ビット).....	514
	同一BIN32ビットデータブロック転送(32ビット).....	516
	BIN16ビットデータ交換	518
	BIN32ビットデータ交換	520
	BIN16ビットデータブロック交換	522
	BIN16ビットデータ上下バイト交換	524
	BIN32ビットデータ上下バイト交換	525
	1ビットデータ転送.....	526
	nビットデータ転送.....	527

第4部 応用命令

第13章	プログラム制御	530
13.1	プログラム分岐命令	530
	ポインタ分岐.....	530
	ENDへジャンプ.....	533
13.2	プログラム実行制御命令.....	534
	割込禁止, 割込許可	534
	指定優先度以下の割込禁止	536
	割込プログラムマスク.....	541
	指定割込ポインタの禁止/許可	543
	割込プログラムからの復帰	545
	WDTリセット	546
13.3	構造化命令	547
	FOR~NEXT	547
	FOR~NEXT強制終了	550
	サブルーチンプログラムコール.....	552
	サブルーチンプログラムからのリターン	558
	サブルーチンプログラムの出力OFFコール	559
	プログラムファイル間サブルーチンコール.....	565
	プログラムファイル間サブルーチンプログラム出力OFFコール	572
	サブルーチンプログラムコール.....	577

第14章 データ処理	582
14.1 ローテーション命令	582
BIN16ビットデータの右ローテーション	582
BIN16ビットデータの左ローテーション	585
BIN32ビットデータの右ローテーション	588
BIN32ビットデータの左ローテーション	590
14.2 データテーブル操作命令	592
データテーブルからの先入データリード	592
データテーブルからの後入データリード	594
データテーブルへのデータライト	596
データテーブルのデータ挿入	598
データテーブルのデータ削除	600
14.3 データリード/ライト命令	602
データメモリからのデータリード	603
データメモリへのデータライト	605
14.4 ファイル操作命令	607
指定ファイルからのデータリード	607
指定ファイルへのデータライト	621
指定ファイルの削除	635
指定ファイルのコピー	638
指定ファイルの移動	643
指定ファイル名の変更	648
指定ファイルの状態取得	652
ファイル操作命令で発生するエラーコード	657
14.5 データ制御命令	658
BIN16ビットデータ上下限リミット制御	658
BIN32ビットデータ上下限リミット制御	660
BIN16ビットデータ不感帯制御	662
BIN32ビットデータ不感帯制御	664
BIN16ビットデータゾーン制御	666
BIN32ビットデータゾーン制御	668
BIN16ビット単位スケーリング(ポイント別座標データ)	670
BIN32ビット単位スケーリング(ポイント別座標データ)	673
BIN16ビット単位スケーリング(X/Y別座標データ)	676
BIN32ビット単位スケーリング(X/Y別座標データ)	678
14.6 データ処理命令	680
BIN16ビットデータサーチ	680
BIN32ビットデータサーチ	682
BIN16ビットデータサーチ(最小・同一・最大)	684
BIN32ビットデータサーチ(最小・同一・最大)	686
BIN16ビットデータビットチェック	688
BIN32ビットデータビットチェック	689
BIN16ビットデータ最大値検索	690
BIN32ビットデータ最大値検索	692
BIN16ビットデータ最小値検索	694
BIN32ビットデータ最小値検索	696
BIN16ビットデータソート	698
BIN32ビットデータソート	700
BIN16ビットデータ合計値算出	702
BIN32ビットデータ合計値算出	704

BIN16ビットデータ平均値算出.....	706
BIN32ビットデータ平均値算出.....	708
BIN16ビット平方根算出.....	710
BIN32ビット平方根算出.....	711
CRC演算.....	712

第15章 デバッグ, 故障診断 714

15.1 デバッグ, 故障診断命令.....	714
エラー表示または, アナウンシェータリセット.....	714

第16章 文字列処理 716

16.1 文字列処理命令.....	716
文字列比較.....	716
文字列の結合.....	719
文字列転送.....	723
Unicode対応文字列転送.....	725
BIN16ビットデータ→10進アスキー変換.....	727
BIN32ビットデータ→10進アスキー変換.....	732
BIN16ビットデータ→16進アスキー変換.....	737
BIN32ビットデータ→16進アスキー変換.....	741
BIN16ビットデータ→文字列変換.....	745
BIN32ビットデータ→文字列変換.....	748
BCD4桁データ→10進アスキーコード変換.....	751
BCD8桁データ→10進アスキーコード変換.....	755
単精度実数→文字列変換.....	760
16進BINデータ→16進アスキーコード変換.....	766
複数データの文字列変換結合.....	768
Unicode文字列→シフトJIS文字列変換.....	784
シフトJIS文字列→Unicode文字列変換(バイトオーダーマークなし).....	786
シフトJIS文字列→Unicode変換(バイトオーダーマークあり).....	788
文字列の長さ検出.....	790
文字列の右側からの取出し.....	792
文字列の左側からの取出し.....	794
文字列中の任意取出し.....	796
文字列中の任意置換え.....	798
文字列サーチ.....	801
文字列挿入.....	803
文字列削除.....	805

第17章 実数処理 807

17.1 浮動小数点命令.....	807
単精度実数比較.....	807
倍精度実数比較.....	809
単精度実数加算.....	812
単精度実数減算.....	816
倍精度実数加算.....	820
倍精度実数減算.....	824
単精度実数乗算.....	828
単精度実数除算.....	830
倍精度実数乗算.....	832

倍精度実数除算	834
符号付きBIN16ビットデータ→単精度実数変換	836
符号なしBIN16ビットデータ→単精度実数変換	837
符号付きBIN32ビットデータ→単精度実数変換	838
符号なしBIN32ビットデータ→単精度実数変換	840
倍精度実数→単精度実数変換	842
符号付きBIN16ビットデータ→倍精度実数変換	844
符号なしBIN16ビットデータ→倍精度実数変換	845
符号付きBIN32ビットデータ→倍精度実数変換	846
符号なしBIN32ビットデータ→倍精度実数変換	847
単精度実数→倍精度実数変換	848
文字列→単精度実数変換	849
BCDフォーマットデータ→単精度実数データ変換	853
単精度実数符号反転	855
倍精度実数符号反転	856
単精度実数データ転送	857
倍精度実数データ転送	858
単精度実数SIN演算	859
単精度実数COS演算	861
単精度実数TAN演算	863
単精度実数 SIN^{-1} 演算	865
単精度実数 COS^{-1} 演算	867
単精度実数 TAN^{-1} 演算	869
倍精度実数SIN演算	871
倍精度実数COS演算	873
倍精度実数TAN演算	875
倍精度実数 SIN^{-1} 演算	877
倍精度実数 COS^{-1} 演算	879
倍精度実数 TAN^{-1} 演算	881
BCD型SIN演算	883
BCD型COS演算	885
BCD型TAN演算	887
BCD型 SIN^{-1} 演算	889
BCD型 COS^{-1} 演算	891
BCD型 TAN^{-1} 演算	893
単精度実数角度→ラジアン変換	895
単精度実数ラジアン→角度変換	897
倍精度実数角度→ラジアン変換	899
倍精度実数ラジアン→角度変換	901
単精度実数平方根	903
倍精度実数平方根	905
単精度実数指数演算	907
倍精度実数指数演算	909
単精度実数自然対数演算	911
倍精度実数自然対数演算	913
BCD4桁平方根	915
BCD8桁平方根	917
単精度実数べき乗演算	919
倍精度実数べき乗演算	921
単精度実数常用対数演算	923
倍精度実数常用対数演算	925

単精度実数最大値検索.....	927
倍精度実数最大値検索.....	929
単精度実数最小値検索.....	931
倍精度実数最小値検索.....	933
第18章 乱数	935
18.1 乱数命令.....	935
乱数発生.....	935
系列変更.....	936
第19章 デバイス操作	937
19.1 インデックスレジスタ命令.....	937
インデックスレジスタの一括退避.....	937
インデックスレジスタの一括復帰.....	940
インデックスレジスタ/ロングインデックスレジスタ選択退避.....	941
インデックスレジスタ/ロングインデックスレジスタ選択復帰.....	945
19.2 ファイルレジスタ操作命令.....	947
ファイルレジスタのブロックNo.切換え.....	947
19.3 ファイルレジスタの1バイト単位のリード/ライト命令.....	949
ファイルレジスタの1バイトデータリード.....	949
ファイルレジスタの1バイトデータライト.....	951
19.4 間接アドレスリード命令.....	953
間接アドレス読出し.....	953
第20章 タイマ, カウンタ	955
20.1 特殊カウンタ命令.....	955
1相入力アップ/ダウンカウンタ.....	955
2相入力アップ/ダウンカウンタ.....	957
20.2 特殊タイマ命令.....	959
ティーチングタイマ.....	959
特殊機能タイマ.....	961
第21章 時計	963
21.1 時計用命令.....	963
時計データの読出し.....	963
時計データの書込み.....	965
時計データの加算.....	967
時計データの減算.....	969
時間データの変換(時分秒→秒).....	971
時間データの変換(秒→時分秒).....	973
日時データの変換(日時→秒).....	975
日時データの変換(秒→日時).....	977
日付比較.....	979
時刻比較.....	983
時刻比較出力.....	986
拡張時計データの読出し.....	988
拡張時計データの加算.....	990
拡張時計データの減算.....	992

21.2	タイミング計測命令.....	994
	タイミングパルス発生.....	994
	指定データの時間計測.....	996

第22章 ユニットアクセス 997

22.1	ユニットアクセス命令.....	997
	I/Oリフレッシュ.....	997
	選択リフレッシュ.....	999
	ユニットリフレッシュ.....	1001
	ユニットからの1ワード/2ワードデータリード(16ビット指定).....	1003
	ユニットへの1ワード/2ワードデータライト(16ビット指定).....	1006
	ユニットからの1ワード/2ワードデータリード(32ビット指定).....	1011
	ユニットへの1ワード/2ワードデータライト(32ビット指定).....	1015
	ユニット形名読出し.....	1020
	ユニット固有情報読出し.....	1024

第23章 PID演算命令 1031

23.1	概要.....	1031
	制御用データ.....	1033
	オートチューニング.....	1041
23.2	PID演算命令.....	1043

第24章 PID制御命令 1047

24.1	概要.....	1047
	演算方式.....	1047
	PID制御手順.....	1048
	その他の機能.....	1054
24.2	PID制御命令(不完全微分).....	1056
	PID制御用データの設定.....	1058
	PID演算.....	1061
	指定ループNo.の演算停止.....	1064
	指定ループNo.の演算開始.....	1065
	指定ループNo.のパラメータ変更.....	1066
24.3	PID制御命令(完全微分).....	1068
	PID制御用データの設定.....	1070
	PID演算.....	1072
	指定ループNo.の演算停止.....	1075
	指定ループNo.の演算開始.....	1076
	指定ループNo.のパラメータ変更.....	1077

第25章 アドオン 1079

25.1	アドオン用命令.....	1079
	アドオン実行命令.....	1079

第5部 ネットワーク命令

第26章 Ethernet用命令 1084

26.1	オープン/クローズ処理命令.....	1084
	コネクションの確立.....	1084

	コネクションの切断	1088
26.2	ソケット通信用命令	1090
	受信データのEND処理時読出し	1090
	受信データの命令実行時読出し	1094
	データ送信	1097
	コネクション情報の読出し	1101
	コネクションの送信先変更(UDP/IP)	1104
	コネクションの受信モード変更	1106
	ソケット通信受信データリード	1110
26.3	通信プロトコル支援機能命令	1112
	登録プロトコル実行	1112
26.4	SLMPフレーム送信命令	1120
	SLMPフレーム送信	1120
	暗号化したSLMPフレーム送信	1129
26.5	ファイル転送機能用命令	1138
	FTPクライアントファイル送付	1138
	FTPクライアントファイル取得	1144

第6部 汎用FUN

第27章	型変換ファンクション	1152
27.1	BOOL型→WORD型変換	1152
27.2	BOOL型→DWORD型変換	1154
27.3	BOOL型→INT型変換	1155
27.4	BOOL型→DINT型変換	1156
27.5	BOOL型→TIME型変換	1157
27.6	BOOL型→STRING型変換	1158
27.7	WORD型→BOOL型変換	1159
27.8	WORD型→DWORD型変換	1160
27.9	WORD型→INT型変換	1161
27.10	WORD型→DINT型変換	1162
27.11	WORD型→TIME型変換	1164
27.12	WORD型→STRING型変換	1165
27.13	DWORD型→BOOL型変換	1166
27.14	DWORD型→WORD型変換	1167
27.15	DWORD型→INT型変換	1169
27.16	DWORD型→DINT型変換	1171
27.17	DWORD型→TIME型変換	1172
27.18	DWORD型→STRING型変換	1173
27.19	INT型→BOOL型変換	1174
27.20	INT型→WORD型変換	1175
27.21	INT型→DWORD型変換	1176
27.22	INT型→DINT型変換	1178
27.23	INT型→BCD型変換	1179
27.24	INT型→REAL型変換	1181
27.25	INT型→LREAL型変換	1182
27.26	INT型→TIME型変換	1183
27.27	INT型→STRING型変換	1184
27.28	DINT型→BOOL型変換	1186
27.29	DINT型→WORD型変換	1187

27.30	DINT型→DWORD型変換	1189
27.31	DINT型→INT型変換	1190
27.32	DINT型→BCD型変換	1191
27.33	DINT型→REAL型変換	1193
27.34	DINT型→LREAL型変換	1194
27.35	DINT型→TIME型変換	1195
27.36	DINT型→STRING型変換	1196
27.37	BCD型→INT型変換	1198
27.38	BCD型→DINT型変換	1200
27.39	BCD型→STRING型変換	1203
27.40	REAL型→INT型変換	1205
27.41	REAL型→DINT型変換	1207
27.42	REAL型→LREAL型変換	1209
27.43	REAL型→STRING型変換	1211
27.44	LREAL型→INT型変換	1214
27.45	LREAL型→DINT型変換	1216
27.46	LREAL型→REAL型変換	1218
27.47	TIME型→BOOL型変換	1220
27.48	TIME型→WORD型変換	1221
27.49	TIME型→DWORD型変換	1222
27.50	TIME型→INT型変換	1223
27.51	TIME型→DINT型変換	1224
27.52	TIME型→STRING型変換	1225
27.53	STRING型→BOOL型変換	1227
27.54	STRING型→WORD型変換	1228
27.55	STRING型→DWORD型変換	1229
27.56	STRING型→INT型変換	1230
27.57	STRING型→DINT型変換	1232
27.58	STRING型→BCD型変換	1234
27.59	STRING型→REAL型変換	1236
27.60	STRING型→TIME型変換	1239
27.61	ビット配列→INT型変換	1241
27.62	ビット配列→DINT型変換	1242
27.63	INT型→ビット配列変換	1243
27.64	DINT型→ビット配列変換	1244
27.65	ビット配列のコピー	1245
27.66	ワードラベルの指定ビット読出し	1246
27.67	ワードラベルの指定ビット書込み	1248
27.68	ワードラベルの指定ビットコピー	1250
27.69	先頭データの取得	1252
第28章 単数値変数ファンクション		1253
28.1	絶対値	1253
28.2	平方根	1255
28.3	自然対数演算	1256
28.4	常用対数演算	1257
28.5	指数演算	1259
28.6	SIN演算	1260
28.7	COS演算	1261
28.8	TAN演算	1262

28.9	SIN^{-1} 演算	1263
28.10	COS^{-1} 演算	1264
28.11	TAN^{-1} 演算	1265
第29章 算術演算ファンクション		1266
29.1	加算	1266
29.2	乗算	1269
29.3	減算	1271
29.4	除算	1274
29.5	剰余	1276
29.6	べき乗	1278
29.7	代入	1279
第30章 ビットシフトファンクション		1281
30.1	nビット左シフト	1281
30.2	nビット右シフト	1283
30.3	nビット左ローテーション	1285
30.4	nビット右ローテーション	1287
第31章 ビット型ブールファンクション		1289
31.1	論理積, 論理和, 排他的論理和	1289
31.2	論理否定	1292
第32章 選択ファンクション		1293
32.1	選択値	1293
32.2	最大値, 最小値選択	1295
32.3	上下限リミット制御	1297
32.4	マルチプレクサ	1300
第33章 比較ファンクション		1302
33.1	比較	1302
33.2	比較	1304
第34章 文字列ファンクション		1306
34.1	文字列の長さ検出	1306
34.2	文字列の左側, 右側からの抽出	1308
34.3	文字列の抽出	1311
34.4	文字列の結合	1313
34.5	文字列の挿入	1315
34.6	文字列の削除	1317
34.7	文字列の置換	1319
34.8	文字列の検索	1322
第35章 時刻データ型ファンクション		1324
35.1	加算	1324
35.2	減算	1326
35.3	乗算	1328
35.4	除算	1330

第7部 汎用FB

第36章	二安定ファンクションブロック	1334
36.1	二安定ファンクションブロック(セット優先)	1334
36.2	二安定ファンクションブロック(リセット優先)	1336
第37章	エッジ検出ファンクションブロック	1338
37.1	立ち上りエッジ検出	1338
37.2	立ち下りエッジ検出	1340
第38章	カウンタ/タイマファンクションブロック	1342
38.1	アップカウンタ	1342
38.2	ダウンカウンタ	1344
38.3	アップダウンカウンタ	1346
38.4	カウンタファンクションブロック	1349
38.5	パルスタイマ	1351
38.6	オンディレイタイマ	1354
38.7	オフディレイタイマ	1356
38.8	タイマファンクションブロック	1358

第8部 モーション制御ファンクションブロック

第39章	モーション制御FBの概要	1364
39.1	モーション制御FB	1364
	モーション制御FBの使用方法	1364
	システム状態ごとのモーション制御FBの動作	1364
	モーション制御FBの種類	1365
	モーション制御の種類	1371
	エラー処理	1371
	制御に使用する単位	1371
	モーション制御FBの入出力変数	1372
	モーション制御FBの注意事項	1373
39.2	モーション制御FBの構成	1374
	ラダー	1374
	FBD/LD(ファンクションブロックダイアグラム/ラダー)	1375
	ST(ストラクチャードテキスト)	1375
第40章	モーション制御ファンクションブロックの変数一覧	1376
40.1	変数一覧	1376
	軸変数	1378
	軸グループ変数	1402
	システム変数	1408
	入力軸変数	1412
	出力軸変数	1414
	システム変数(アドバンス同期制御)	1419
	その他構造体の変数	1419
	その他構造体の変数(アドバンス同期制御)	1437
40.2	列挙子一覧	1448
	ENUM列挙子	1448

第41章 管理系のファンクションブロック	1456
41.1 軸グループ有効	1456
41.2 軸グループ無効	1460
41.3 運転可	1463
41.4 現在位置変更	1467
41.5 トルク制限値	1475
41.6 オーバライド値設定	1483
41.7 パラメータ読出	1487
41.8 パラメータ書込	1494
41.9 軸エラーリセット	1500
41.10 軸グループエラーリセット	1503
41.11 タッチプローブ有効	1507
41.12 タッチプローブ無効	1522
41.13 カムテーブル選択	1527
41.14 1サイクル現在値変更	1534
41.15 全軸運転可	1542
41.16 軸グループオーバライド値設定	1545
41.17 モーションエラーリセット	1549
41.18 制御演算周期同期	1553
41.19 アドバンスト同期制御1サイクル現在位置計算	1556
41.20 アドバンスト同期制御カム指令現在位置計算	1559
41.21 デジタルカムスイッチ出力	1562
第42章 動作系のファンクションブロック	1569
42.1 原点復帰	1569
42.2 強制停止	1579
42.3 グループ強制停止	1584
42.4 絶対値位置決め	1590
42.5 相対値位置決め	1602
42.6 JOG運転	1610
42.7 速度制御	1618
42.8 トルク制御	1625
42.9 速度制御(位置ループを含む)	1635
42.10 絶対値直線補間制御	1643
42.11 相対値直線補間制御	1658
42.12 絶対値円弧補間制御	1671
42.13 相対値円弧補間制御	1688
42.14 カム動作開始	1705
42.15 ギア動作開始	1732
42.16 加減算位置決め	1742
42.17 バックラッシュ補正フィルタ	1752
42.18 スムージングフィルタ	1765
42.19 移動方向制限フィルタ	1774
42.20 速度制限フィルタ	1782
42.21 軸制御開始ウェイト	1790
42.22 軸グループ制御開始ウェイト	1797
42.23 アドバンスト同期制御	1804
42.24 多軸位置決めデータ運転	1810
42.25 絶対位置追従制御	1819
42.26 モーションサイクリック位置制御	1830

42.27 モーションサイクリック速度制御	1838
42.28 モーションサイクリックトルク制御	1842

第43章 一般ファンクションブロック 1848

43.1 プロファイル読出	1848
43.2 プロファイル書込	1862

第9部 ネットワークファンクションブロック

第44章 Ethernet用ファンクションブロック 1876

44.1 コネクションの確立/切断	1876
44.2 ソケット通信	1881
44.3 SLMP	1887

第45章 CC-Link IE TSN用ファンクションブロック 1915

45.1 リモートリセット	1915

付録 1919

付1 命令処理時間	1919
命令処理時間の加算時間	1922
付2 ユニット専用命令使用可否	1923
付3 PIDの3定数の求め方	1934

索引 1936

命令索引 1938

改訂履歴	1947
製品の適用について	1948
保証について	1949
サービスのお問い合わせ	1950
商標	1950

関連マニュアル

本製品に関連するマニュアルには、下記のものがあります。
最新のマニュアルは、三菱電機FAサイトからダウンロードできます。
www.MitsubishiElectric.co.jp/fa

マニュアル名称[マニュアル番号]	内容
MELSEC MXコントローラ(MX-Rモデル) プログラミングマニュアル [SH-082643](本マニュアル)	プログラム言語の仕様、コントローラの命令、汎用ファンクション/ファンクションブロック、モーション制御ファンクションブロックの仕様について記載しています。
MELSEC MXコントローラ(MX-Rモデル) ユーザーズマニュアル [SH-082640]	コントローラの運転までの手順、仕様、デバイス、メモリ、機能、パラメータ、トラブルシューティングなどについて記載しています。
MELSEC iQ-R プログラミングマニュアル(ユニット専用命令編) [SH-081975]	インテリジェント機能ユニットの専用命令について記載しています。
GX Works3 オペレーティングマニュアル [SH-081214]	GX Works3のシステム構成、パラメータ設定、オンライン機能の操作方法などについて説明しています。

用語

本マニュアルでは、特に明記する場合を除き、下記の用語を使用して説明します。

用語	内容
インテリジェント機能ユニット	A/D、D/A変換ユニットなど、入出力以外の機能を持つユニットです。
オブジェクト	CANopen対応のデバイス機器が保有するデータです。
カムテーブル	カム制御に使用する演算プロファイルです。 (例)カムデータ形式の演算プロファイル、ロータリーカッター用カムデータ形式の演算プロファイル
カムテーブルの1点目	カムデータの補間方法指定によって意味が異なります。 ・区間ごとに指定、スプライン補間: テーブルNo.1の点 ・直線補間: ポイントNo.1の点
グランドマスタ	PTP(Precision Time Protocol)を使用した時刻同期における、時刻同期元となる機器です。
グローバルラベル	プロジェクト内で複数のプログラムデータを作成したときに、すべてのプログラムデータに対して有効となるラベルです。 グローバルラベルには、GX Works3が自動で生成するユニット固有のラベル(ユニットラベル)と任意で指定のデバイスに対して作成できるラベルがあります。
サイクリックデータ転送処理	1つのネットワークで、すべての局がサイクリック伝送を開始してから完了するまでの動作です。CPU機能部のシーケンススキャンと非同期に行われます。 サイクリックデータ転送処理時間は、データ量やトランジェント伝送要求などにより変動します。
CPU機能部	シーケンス制御およびEthernet用ポート(P3)でのEthernet通信を行う部分です。
CC-Link IE TSN機能部	CC-Link IE TSN用ポート(P1、P2)でのCC-Link IE TSN通信とEthernet通信を行う部分です。
モーション機能部	コントローラでモーション制御を行う部分です。
コア	コントローラに搭載されている演算回路です。コントローラには複数のコアが搭載されています。
基本コア	プログラムを実行するコアです。
拡張コア	実行する機能を割り当て可能なコアです。
シグナルフロー	プログラムやFBの各ステップにおける演算の前の実行/非実行状態です。
システムエラー	軸に関連しないエラーや警告モーション制御FBの軸情報(Axis)/軸グループ情報(AxesGroup)が不正の場合はシステムエラーとなります。
ジャーク	加速度または減速度の時間的な変化比率です。
ソフトLED	ハードウェアとしては搭載されず、ソフトウェアでのみ制御するLEDです。 ユニット診断で確認できます。
デバイス	データを格納するためのコントローラのメモリです。用途に応じてX/Y/M/Dなどがあります。
デバイス局	マスタ局以外の局(ローカル局、リモート局)です。
トランジェント伝送グループNo.	任意の局にトランジェント伝送するための番号です。トランジェント伝送の対象局をグループ指定すると、同じグループNo.の局にデータを送信できます。
バッファメモリ	設定値、モニタ値などのデータを格納するためのメモリです。
バッファリング	多重起動において、モーション制御FBが待機状態に入ることです。
バッファリングFB	多重起動が行われ、実行待ち状態(BusyがTRUE)となっているモーション制御FBです。
フィードバック位置	デバイス機器が返す位置アドレスをリングカウンタで丸めたものです。現在位置変更の影響を受けます。
フィードバック速度	実軸におけるデバイス機器からのフィードバック値を軸の単位系に変換したものです。
プログラムブロック	プログラムを構成するプログラム部品の集まりです。
プログラム部品	機能ごとに分けて定義されたプログラムの単位です。プログラムの部品化をすることで、プログラムを階層化したときの下位の処理を、処理内容や機能ごとにいくつかの単位に分け、単位ごとのプログラムを作成できます。
マスタ局	ネットワーク全体を制御する局です。すべての局とサイクリック伝送およびトランジェント伝送ができます。 1ネットワークに1台のみ存在します。
マルチキャストフィルタ	自局が受信したマルチキャストモードのサイクリックデータを、後続の局に流すか流さないかを選択できるフィルタ機能です。 マスタ局がシステム構成に応じて自動で設定するため、パラメータの設定は不要です。
マルチキャストモード	複数の局に対してサイクリックデータを送信する通信モードです。
モーション制御FB	モーション制御に関連するFBです。名称がMC_またはMCv_から始まります。
ユニキャストモード	1つの局に対してサイクリックデータを送信する通信モードです。
ユニットラベル	各ユニットが固有で定義しているメモリ(入出力信号やバッファメモリ)を、任意の文字列で表したものです。 使用するユニットからGX Works3が自動で生成し、グローバルラベルとして使用できます。
ラベル	入出力データや内部処理に任意の文字列を指定した変数です。
リモート局	ビット単位の入出力信号とワード単位の入出力データをサイクリック伝送する局です。トランジェント伝送もできます。

用語	内容
リンクデバイス	CC-Link IE TSN機能部またはネットワークユニットが内部に持っているデバイス(RX, RY, RWr, RWw, LB, LW)です。
ルーチング	他ネットワークと交信する場合の、通信経路の制御です。通信経路が自動的に選択される動的ルーチングと、任意の通信経路を設定する静的ルーチングがあります。
ローカル局	マスタ局および他ローカル局と、サイクリック伝送およびトランジェント伝送を行う局です。
仮想エンコーダ軸	変数から指令位置を生成する軸です。単軸同期制御の主軸に使用します。
仮想ドライブ軸	仮想的に指令を生成できる仮想軸です。
仮想軸	ネットワーク上のデバイス機器に紐付かない軸です。
仮想連結軸	単軸同期制御においてFB同士を接続して指令を伝達するための軸です。
解列	データリンク異常時に該当する局のデータリンクを停止する処理です。
外部信号高精度入力	デバイス局の信号検出時刻が紐づいた入力信号です。
指定位置	動作系FBで指定する指令位置アドレスです。
指定速度	動作系FBで指定する指令速度です。
指令現在位置	リングカウンタで丸めた指令位置アドレスです。現在位置変更の影響を受けます。
指令現在速度	モーション演算により生成される現在の制御値です。
軸	モーション制御を行う対象です。
軸グループ変数	軸グループに関するパラメータやデータを含む、AXES_GROUP型変数のインスタンスです。
軸ワーニング	軸関連のエラーや警告です。
軸変数	軸に関するパラメータやデータを含む、AXIS_*型変数のインスタンスです。
実エンコーダ軸	デバイス機器に接続したエンコーダの現在位置から指令位置を生成する軸です。単軸同期制御の主軸に使用します。
実ドライブ軸	CiA402ドライブプロファイルのcsp/csv/cstモード(逐次指令)に対応するCC-Link IE TSN対応デバイス機器に紐付いた軸です。
実軸	ネットワーク上のデバイス機器に紐付いた軸です。
主軸(入力)絶対座標	MC_CamTableSelect(カムテーブル選択)の入力変数主軸絶対座標(MasterAbsolute)の別称です。
従軸(出力)絶対座標	MC_CamTableSelect(カムテーブル選択)の入力変数従軸絶対座標(SlaveAbsolute)の別称です。
出力変数	FBの出力引数です。
切換え速度	・Aborting指定時に実行中のモーション制御FBの指令現在速度です。 ・Blending指定時に実行中のモーション制御FBが目標位置に到達した際の指令速度です。
専用命令	インテリジェント機能ユニットの機能を使用するためのプログラミングを容易にするための命令です。
送り機械位置	リングカウンタで丸めない指令位置アドレスです。原点を基準とし、現在位置変更の影響を受けません。
多軸補間制御	直線補間や円弧補間など複数軸が関連して動作する制御です。軸グループにより協調する軸を指定します。
多重起動	モーション制御FBを実行中に、同じ軸に対して別のモーション制御FBを実行することを示します。
単軸同期	Master(主軸)に同期したSlave(従軸)1軸の位置情報(指令)を出力する制御です。
中継局	1台のシーケンサに複数のネットワークユニットを装着し、他ネットワークへのトランジェント伝送を中継する局です。
通信プロトコル支援機能	GX Works3(通信プロトコル支援機能)で使用できる、下記の機能です。 ・相手機器に合わせたプロトコルの設定 ・プロトコル設定データの読出しおよび書込み
動作系FB	軸または軸グループを引数にとり、実行によって軸状態または軸グループ状態が変化するモーション制御FBです。
入力変数	FBの入力引数です。
汎用HUB	CC-Link協会が認定した、CC-Link IE TSN Class AのスイッチングHUBです。
復列	異常局が正常になったときに、該当する局のデータリンクを再開する処理です。
目標位置	動作系FBで指定した指令位置アドレスです。
予約アドレス	RFC6890により定義され、特殊な用途のために予約されているIPアドレスです。グローバルIPネットワークと接続しているときは使用できません。
累積現在位置	リングカウンタで丸めない指令位置アドレスです。現在位置変更の影響を受けます。
連続更新	モーション制御FBのContinuousUpdate入力がTRUEの間、入力値を連続的に制御に反映することを示します。
CC-Link IE TSN Class	CC-Link協会による、CC-Link IE TSNに対応した機器およびスイッチングHUBの、機能・性能によるランク分けです。 CC-Link IE TSN Classについては、CC-Link協会から発行されているCC-Link IE TSN敷設マニュアル(BAP-C3007-001)を参照してください。
CC-Link IE TSNプロトコルバージョン1.0	時刻同期にIEEE 802.1ASまたはIEEE 1588を使用した時分割方式で通信するプロトコルです。

用語	内容
CC-Link IE TSNプロトコルバージョン2.0	時刻同期にIEEE 802.1ASを使用した時分割方式に加え、タイムマネージド・ポーリング方式で通信するプロトコルです。 これに対応していると、システム構成が“CC-Link IE TSN Class B/A混在、または、CC-Link IE TSN Class Aのみ”の場合の制約が緩和され、接続できる機器が増えます。
CPUユニットロギング設定ツール	データロギングの設定および保守を行うためのソフトウェアです。
FBインスタンス	シーケンスプログラムに貼り付けられたファンクションブロックです。
GX LogViewer	データロギングで収集したデータを表示するためのソフトウェアです。
GX Works3	MELSECシーケンサソフトウェアパッケージの製品名です。
iQSS	iQ Sensor Solutionを示します。センサの立ち上げやメンテナンスなどを容易にする、センサ、シーケンサ、表示器、エンジニアリングツールの連携です。(□iQ Sensor Solution リファレンスマニュアル)
LB	リンクデバイスのリンクリレーです。ネットワークの各局からビット単位で送信されます。
LW	リンクデバイスのリンクレジスタです。ネットワークの各局から16ビット(1ワード)単位で送信されます。
MR Configurator2	サーボセットアップソフトウェアの製品名です。
RWr	リンクデバイスのリモートレジスタです。デバイス局からマスタ局に16ビット(1ワード)単位で入力されます。(ローカル局では一部異なる。)
RWw	リンクデバイスのリモートレジスタです。マスタ局からデバイス局に16ビット(1ワード)単位で出力されます。(ローカル局では一部異なる。)
RX	リンクデバイスのリモート入力です。デバイス局からマスタ局にビット単位で入力されます。(ローカル局では一部異なる。)
RY	リンクデバイスのリモート出力です。マスタ局からデバイス局にビット単位で出力されます。(ローカル局では一部異なる。)
SB	リンク特殊リレーの略称です。CC-Link IEのユニット動作状態、データリンク状態を示すビット単位の情報です。
SLMP	SeamLess Message Protocolを示します。 外部機器からSLMP対応機器、およびSLMP対応機器に接続されたシーケンサにアクセスするためのプロトコルです。
SW	リンク特殊レジスタの略称です。CC-Link IEのユニット動作状態、データリンク状態を示す16ビット(1ワード)単位の情報です。
TSN HUB	CC-Link協会が認定した、CC-Link IE TSN Class BのスイッチングHUBです。

総称/略称

本マニュアルでは、特に明記する場合を除き、下記の総称/略称を使用して説明します。

総称/略称	説明
エンジニアリングツール	コントローラやサーボアンプの設定から、プログラミング、デバッグ、保守までを行うためのツールです。
コントローラ	MXR300-16, MXR300-32, MXR300-64, MXR500-128, MXR500-256の総称です。
データリンク	サイクリック伝送およびトランジェント伝送の総称です。
ドライブユニット	サーボアンプなどのモータ駆動機器の総称です。
ベースユニット	基本ベースユニット、増設ベースユニットの総称です。
演算プロファイル	各種制御で使用する波形データの総称です。
局サブID番号	CC-Link対応通信ユニットに連結されているセンサのID番号を示します。
電源ユニット	MELSEC iQ-Rシリーズ電源ユニットを示します。
入出力ユニット	入力ユニット、出力ユニット、入出力混合ユニット、割込みユニットを示します。
CC-Link IE	下記の総称です。 • CC-Link IE TSN • CC-Link IEコントローラネットワーク(📖MELSEC iQ-R CC-Link IEコントローラネットワークユーザーズマニュアル(応用編)) • CC-Link IEフィールドネットワーク(📖MELSEC iQ-R CC-Link IEフィールドネットワークユーザーズマニュアル(応用編))
CC-Link IE TSNマスタ・ローカルユニット	RJ71GN11-T2, RJ71GN11-SX, RJ71GN11-EIP(CC-Link IE TSN部)を示します。
CC-Link IE TSN搭載ユニット	RJ71GN11-T2, RJ71GN11-SX, RJ71GN11-EIPの総称です。
CC-Link IEコントローラネットワーク搭載ユニット	RJ71GP21-SX形CC-Link IEコントローラネットワークユニット, RJ71GP21S-SX形CC-Link IEコントローラネットワークユニットおよびCC-Link IEコントローラネットワーク機能使用時の下記ユニットを示します。 • RJ71EN71
csp	サイクリック位置モード(Cyclic synchronous position mode)の略称(ドライバ側の制御モードの一種)です。
cst	サイクリックトルクモード(Cyclic synchronous torque mode)の略称(ドライバ側の制御モードの一種)です。
csv	サイクリック速度モード(Cyclic synchronous velocity mode)の略称(ドライバ側の制御モードの一種)です。
ct	押当て制御モード(Continuous operation to torque control mode)の略称(ドライバ側の制御モードの一種)です。
Ethernet機器	IP通信に対応した機器の総称です。(パソコン, ビジョンセンサ, バーコードリーダなど)
Ethernet搭載ユニット	下記ユニットの総称です。 • コントローラ(Ethernet機能使用時) • RJ71EN71(Ethernet機能使用時)
GOT	三菱電機グラフィックオペレーションターミナルGOT1000, GOT2000シリーズの総称です。
hm	原点復帰モード(Homing mode)の略称(ドライバ側の制御モードの一種)です。
ICMP	Internet Control Message Protocolの略称です。IPネットワーク上で発生したエラーやネットワークに関する情報をやりとりするためのプロトコルです。
iQSS対応機器	iQ Sensor Solutionに対応している機器を示します。 📖iQ Sensor Solution リファレンスマニュアル
MCFB	モーション制御FBの略称です。
MR-J5(W)-G	MR-J5-_G_(-RJ)/MR-J5W_-_G_/MR-J5D_-_G_サーボアンプの総称です。
PDO	Process Data Objectの略称です。複数のCANopenノード間で周期的に転送されるアプリケーションオブジェクトの集合体です。
PTP	Precision Time Protocolの略称です。ネットワーク内の機器の間で時刻を同期させるために使用する通信プロトコルです。
RJ71GN11	RJ71GN11-T2, RJ71GN11-SXの総称です。
SDO	Service Data Objectの略称です。任意のCANopenノードのオブジェクトディクショナリ内のオブジェクトエントリにアクセスするためのメッセージです。局間で非周期的に送受信します。
SLMP	Seamless Message Protocolの略称。パーソナルコンピュータや表示器などの外部機器およびSLMPに対応したユニット(Ethernet搭載ユニットやCC-Link IE TSNのユニットなど)からSLMP対応機器にアクセスするためのプロトコルです。

第1部

プログラミング概要

この部は下記の章構成となります。

1 命令概要

2 プログラム部品

3 ラダー言語

4 ST言語

5 FBD/LD言語

1 命令概要

1.1 命令の構成

コントローラで使える命令の多くは、命令部とオペランド部に分けることができます。

命令部とオペランド部の用途は、下記になります。

- 命令部: その命令の機能を示します。
- オペランド部: 命令で使用するデータを示します。

オペランド部は、ソースデータ、ディスティネーションデータ、数値データに分類されます。

ソース(s)

ソースは演算で使用するデータです。

各命令で指定するラベルやデバイスにより、下記のようになります。

種類	内容
定数	演算で使用する数値を指定します。 プログラム作成時に設定するため、プログラム実行中に変更できません。 定数を可変データで使用する場合は、インデックス修飾をしてください。 ^{*1}
デバイス ラベル	演算で使用するデータの格納されているデバイスまたはラベルを指定します。 演算を実行するまでに指定のデバイスまたはラベルへデータを格納しておく必要があります。 プログラム実行中、指定されたデバイスまたはラベルに格納するデータを変更することにより、その命令で使用するデータを変更できます。

*1 インデックス修飾については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

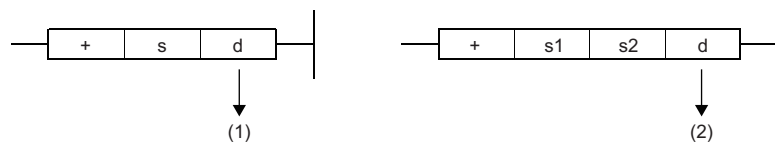
ディスティネーション(d)

ディスティネーションには、演算後のデータが格納されます。

ただし、命令によっては、演算前にディスティネーションに演算で使用するデータの格納が必要な場合があります。

例

BIN16ビットデータの加算命令の場合



(1): 演算実行前に、使用するデータを格納します。

(2): 演算結果のみ格納されます。

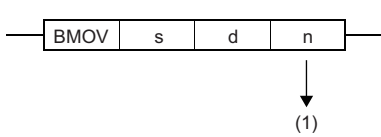
ディスティネーションには、必ずデータを格納するためのラベルやデバイスを設定します。

デバイス数/転送数/データ数/文字列数などの数値(n)

デバイス数，転送数，データ数，文字列数などの数値は，複数デバイスを使用する命令や繰り返す回数，処理するデータ数・文字列数などを指定する命令で使用するデバイス数，転送数，データ数，文字列数などを指定します。

例

ブロック転送命令の場合



(1): BMOV命令で転送する転送数を指定します。

デバイス数，転送数，文字数などのサイズ指定は，0~65535，0~4294967295が設定可能です。^{*1}

ただし，デバイス数，転送数，文字数などのサイズ指定が0の場合，その命令は無処理となります。

また，デバイスメモリ容量やファイル格納容量によっては，上記サイズ指定の上限値未満となる場合があります。^{*2}

^{*1} 設定範囲は命令によって異なります。詳細は各命令を参照してください。

^{*2} デバイスメモリ容量や，ファイル格納容量は機種によって異なります。詳細については，下記を参照してください。

使用するコントローラのユーザーズマニュアル

Point

転送数などの数値で大きい値を使用する場合，スキャンタイムが遅くなるので注意してください。

1.2 データの指定方法

コントローラの各命令で使用できるデータには、下記の種類があります。

データ	分類
ビットデータ	ビットデータ
16ビットデータ(ワードデータ)	符号付きBIN16ビットデータ 符号なしBIN16ビットデータ
32ビットデータ(ダブルワードデータ)	符号付きBIN32ビットデータ 符号なしBIN32ビットデータ
実数データ(浮動小数データ)	単精度実数データ 倍精度実数データ
BCDデータ	BCD4桁データ BCD8桁データ BCD16桁データ
文字列データ	文字列 Unicode文字列

デバイスで扱うデータ

設定データをデバイスで指定する場合について、指定可能なデバイス/定数の種類を示します。

データ型	内容	指定可能デバイス/定数 ^{*1}
ビット	ビットデータを取り扱うことができます。 ☞ 31ページ ビットデータ	• ビットデバイス • ワードデバイスのビット指定
ワード	ワードデータを取り扱うことができます。 ☞ 32ページ 16ビットデータ(ワードデータ)	• ワードデバイス • ビットデバイスの桁指定(K1~K4) ^{*2} • 10進定数 • 16進定数
符号付きBIN16ビット	16ビットデータを取り扱うことができます。	
符号なしBIN16ビット	符号付きと符号なしで、値の範囲が異なります。 ☞ 32ページ 16ビットデータ(ワードデータ)	
ダブルワード	ダブルワードデータを取り扱うことができます。 ☞ 35ページ 32ビットデータ(ダブルワードデータ)	• ワードデバイス • ダブルワードデバイス
符号付きBIN32ビット	32ビットデータまたは、16ビットデータが2個連続しているデータを取り扱うことができます。	• ビットデバイスの桁指定(K1~K8) ^{*2} • 10進定数 • 16進定数
符号なしBIN32ビット	符号付きと符号なしで、値の範囲が異なります。 ☞ 35ページ 32ビットデータ(ダブルワードデータ)	
BCD4桁	BCD4桁データを取り扱うことができます。 16ビットデータを4桁で区切り、各桁を0~9で指定します。	• ワードデバイス • ビットデバイスの桁指定(K1~K4) ^{*2} • 10進定数 • 16進定数
BCD8桁	BCD8桁データを取り扱うことができます。 32ビットデータを8桁で区切り、各桁を0~9で指定します。	• ワードデバイス • ダブルワードデバイス • ビットデバイスの桁指定(K1~K8) ^{*2} • 10進定数 • 16進定数
単精度実数	単精度実数データ(単精度浮動小数点データ)を取り扱うことができます。 ☞ 38ページ 単精度実数データの構成	• ワードデバイス • ダブルワードデバイス • 実数定数
倍精度実数	倍精度実数データ(倍精度浮動小数点データ)を取り扱うことができます。 ☞ 39ページ 倍精度実数データの構成	• ワードデバイス • ダブルワードデバイス • 実数定数
文字列	ASCIIコード、シフトJISコードの文字列データを取り扱うことができます。 ☞ 41ページ 文字列データ	• ワードデバイス • 文字列定数
Unicode文字列	Unicodeの文字列データを取り扱うことができます。 ☞ 41ページ 文字列データ	• ワードデバイス • 文字列定数
デバイス名	デバイスを直接指定できます。	• 使用可能デバイスで対応のデバイス名

^{*1} 命令によってソース(s)や数値データ(n)の指定するデータに、定数が使用できます。

^{*2} 指定方法については、各データ型の詳細ページを参照してください。

ラベルで扱うデータ

設定データをラベルで指定する場合について、指定可能なラベルの種類を示します。

■基本データ型

データ型(ラベル)	指定可能ラベル
ビット (BOOL)	・ビット型ラベル ・ワード[符号なし]/ビット列[16ビット]型ラベルのビット指定 ・ワード[符号付き]型ラベルのビット指定 ・タイマ/積算タイマ/ロングタイマ/ロング積算タイマ型ラベルの接点・コイル ・カウンタ/ロングカウンタ型ラベルの接点・コイル
ワード[符号なし]/ビット列[16ビット] (WORD)	・ワード[符号なし]/ビット列[16ビット]型ラベル ・ビット型ラベルの桁指定(K1~K4) ・タイマ/積算タイマ型ラベルの現在値 ・カウンタ型ラベルの現在値
ダブルワード[符号なし]/ビット列[32ビット] (DWORD)	・ダブルワード[符号なし]/ビット列[32ビット]型ラベル ・ビット型ラベルの桁指定(K1~K8) ・ロングタイマ/ロング積算タイマ型ラベルの現在値 ・ロングカウンタ型ラベルの現在値
ワード[符号付き] (INT)	・ワード[符号付き]型ラベル ・ビット型ラベルの桁指定(K1~K4) ・タイマ/積算タイマ型ラベルの現在値 ・カウンタ型ラベルの現在値
ダブルワード[符号付き] (DINT)	・ダブルワード[符号付き]型ラベル ・ビット型ラベルの桁指定(K1~K8) ・ロングタイマ/ロング積算タイマ型ラベルの現在値 ・ロングカウンタ型ラベルの現在値
単精度実数 (REAL)	・単精度実数型ラベル
倍精度実数 (LREAL)	・倍精度実数型ラベル
時間 (TIME)	・時間型ラベル
文字列 (STRING)	・文字列型ラベル
文字列[Unicode] (WSTRING)	・文字列[Unicode]型ラベル
ポインタ (POINTER)	・ポインタ型ラベル

Point

各ラベルの内容については、下記を参照してください。

 使用するコントローラのユーザーズマニュアル

■総称データ型

いくつかの基本データ型をまとめたラベルのデータ型です。

ファンクションやファンクションブロックの引数、戻り値などで複数のデータ型が許される場合に、総称データ型が使用されています。

総称データ型で定義されたラベルは、下位のデータ型のどの型でも使用できます。

データ型(ラベル)							指定可能なデータ型			
ANY* ¹	ANY_ELEMENTARY	ANY_BIT				ANY_BOOL	ビット			
						ANY_BITADDR* ¹	ビット			
						ANY16_U	ワード[符号なし]/ビット列[16ビット]			
						ANY32_U	ダブルワード[符号なし]/ビット列[32ビット]			
		ANY_WORDADDR				ANY_NUM	ANY_INT	ANY16	ANY16_S	ワード[符号付き]
								ANY16_U	ワード[符号なし]/ビット列[16ビット]	
							ANY32	ANY32_S	ダブルワード[符号付き], 時間	
								ANY32_U	ダブルワード[符号なし]/ビット列[32ビット]	
						ANY_REAL		ANYREAL_32	単精度実数	
								ANYREAL_64	倍精度実数	
						ANY_STRING		ANYSTRING_SINGLE	文字列	
								ANYSTRING_DOUBLE	文字列[Unicode]	
		ANY16_OR_STRING_SINGLE		ANY16_S	ワード[符号付き]					
				ANY16_U	ワード[符号なし]/ビット列[16ビット]					
				ANYSTRING_SINGLE	文字列					
		ANY_DT				ワード[符号付き], ワード[符号なし]/ビット列[16ビット]				
		ANY_TM				ワード[符号付き], ワード[符号なし]/ビット列[16ビット]				
		ANY_STRUCT* ¹							構造体	
		STRUCT							構造体	

*1 配列としても使用できます。

■総称データ型(配列)

下記の総称データ型については、配列要素数を定義してください。

データ型(ラベル)			指定可能なデータ型
ANYBIT_ARRAY			ビットの配列
ANYWORD_ARRAY	ANY16_ARRAY	ANY16_S_ARRAY	ワード[符号付き]の配列
		ANY16_U_ARRAY	ワード[符号なし]/ビット列[16ビット]の配列
	ANY32_ARRAY	ANY32_S_ARRAY	ダブルワード[符号付き]の配列, 時間の配列
		ANY32_U_ARRAY	ダブルワード[符号なし]/ビット列[32ビット]の配列
	ANY_REAL_ARRAY	ANY_REAL_32_ARRAY	単精度実数の配列
		ANY_REAL_64_ARRAY	倍精度実数の配列
	ANY_STRING_ARRAY	ANY_STRING_SINGLE_ARRAY	文字列の配列
		ANY_STRING_DOUBLE_ARRAY	文字列[Unicode]の配列
STRUCT_ARRAY			構造体の配列

ビットデータ

データサイズとデータの範囲

ビットデータは、接点やコイルなどを1ビット単位で扱うデータです。

データ名称	データサイズ	値の範囲
ビットデータ	1ビット	0, 1

ビットデバイス/ラベルにおけるビットデータの扱い

ビットデバイス/ラベル1点に対して、1ビットのデータを扱うことができます。

ワードデバイスにおけるビットデータの扱い

ワードデバイスは、ビットNo.指定を行うことにより、指定ビットNo.のビットデータを扱うことができます。

ビット指定の表記方法は、「ワードデバイス番号.ビットNo.」になります。

ビットNo.の指定は、16進数で0~Fの範囲で指定します。

たとえば、D0のビット5(b5)は「D0.5」、D0のビット10(b10)は「D0.A」で指定します。

ワードデバイスのビット指定が可能なデバイスは下記になります。

項目	デバイス
ワードデバイスのビット指定が可能なデバイス	- データレジスタ(D) - リンクレジスタ(W, J□¥W) - リンク特殊レジスタ(SW, J□¥SW) - ファンクションレジスタ(FD) - 特殊レジスタ(SD) - ユニットアクセスデバイス(U□¥G) - CPUバッファメモリアクセスデバイス(U3E□¥G) - ファイルレジスタ(R, ZR) - ユニットリフレッシュ用レジスタ(RD)

ワード型ラベルにおけるビットデータの扱い

ワード型ラベルでは、ビットNo.指定を行うことにより、指定ビットNo.のビットデータを扱うことができます。

ビット指定の表記方法は「ラベル名.ビットNo.」になります。

例



ビット指定が可能なラベルのデータ型は、下記になります。

項目	データ型
ビット指定が可能なラベルのデータ型	- ワード[符号付き](INT型) - ワード[符号なし]/ビット列[16ビット](WORD型) - タイマ(TIMER型)の現在値(N)^{*1} - 積算タイマ(RETENTIVETIMER型)の現在値(N)^{*1} - カウンタ(COUNTER型)の現在値(N)^{*1}

^{*1} ラダーの場合は指定できません。

16ビットデータ(ワードデータ)

データサイズとデータの範囲

16ビットデータには、符号付き16ビットデータ、符号なし16ビットデータがあります。
符号付き16ビットデータについて、負数は2の補数で表現します。

データ名称	データサイズ	値の範囲	
		10進表記	16進表記
符号付き16ビットデータ	16ビット(1ワード)	-32768~32767	0000H~FFFFH
符号なし16ビットデータ		0~65535	

ビットデバイスにおける16ビットデータの扱い

ビットデバイスは桁指定することにより、16ビットデータとして扱うことができます。

項目		表記	例
ビットデバイス	リンクダイレクトデバイス以外	K□ビットデバイス先頭番号 □: 桁数(1~4の範囲で指定)	K4X40 K2M113
	リンクダイレクトデバイス	J□¥K□ビットデバイス先頭番号 左側□: ネットワークNo. 右側□: 桁数(1~4の範囲で指定)	J1 ¥ K3B10 J10 ¥ K2Y40

ビット型配列のラベルにおける16ビットデータの扱い

ビット型配列のラベルは桁指定することにより、16ビットデータとして扱うことができます。
ビット型配列のラベルを桁指定して16ビットデータとして扱うときの表記方法を示します。

項目	表記	例
ビット型配列のラベル	K□ラベル名 □: 桁数(1~4の範囲で指定) ラベルの桁指定は、要素指定なしで指定します。	K1L_BOOL

桁指定の範囲

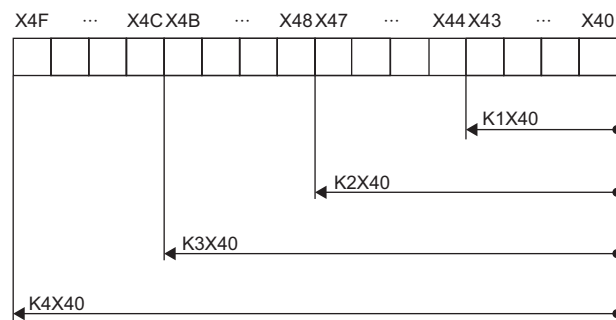
桁指定ごとの16ビットデータの範囲を示します。

桁指定	10進表記	16進表記
K1	0~15	0H~FH
K2	0~255	00H~FFH
K3	0~4095	000H~FFFH
K4	符号付き16ビットデータ: -32768~32767 符号なし16ビットデータ: 0~65535	0000H~FFFFH

例

X40を桁指定した場合の対象点数は、下記になります。

- K1X40→X40~X43の4点が対象
- K2X40→X40~X47の8点が対象
- K3X40→X40~X4Bの12点が対象
- K4X40→X40~X4Fの16点が対象



■ソース(s)側に桁指定したビットデバイスを指定した場合

命令のソース側に桁指定したビットデバイスを指定した場合、ディスティネーション側のワードデバイスには、ソース側で桁指定したビットより上位のビットに0が格納されます。

回路例	処理
• 16ビットデータの命令の場合	

■ディスティネーション(d)側に桁指定したビットデバイスを指定した場合

命令のディスティネーション側に桁指定がある場合、桁指定による点数がディスティネーション側の対象となります。桁指定による点数より上位のビットデバイスは、変化しません。

回路例	処理
• ソースデータがワードデバイスの場合	(1): 変化しません。

ワードデバイス/ラベルにおける16ビットデータの扱い

■ワードデバイス

ワードデバイス1点で16ビットデータを扱うことができます。

■ワード型ラベル

ワード型ラベル1点で16ビットデータを扱うことができます。

32ビットデータ(ダブルワードデータ)

データサイズとデータの範囲

32ビットデータには、符号付き32ビットデータ、符号なし32ビットデータがあります。

符号付き32ビットデータについて、負数は2の補数で表現します。

データ名称	データサイズ	値の範囲	
		10進表記	16進表記
符号付き32ビットデータ	32ビット(2ワード)	-2147483648~2147483647	00000000H~FFFFFFFFH
符号なし32ビットデータ		0~4294967295	

ビットデバイスにおける32ビットデータの扱い

ビットデバイスは桁指定することにより、32ビットデータとして扱うことができます。

項目	表記	例
ビットデバイス	リンクダイレクトデバイス以外	K8X80 K6B018
	リンクダイレクトデバイス	J1 ¥ K7B30 J10 ¥ K5X128

ビット型配列のラベルにおける32ビットデータの扱い

ビット型配列のラベルは桁指定することにより、32ビットデータとして扱うことができます。

ビット型配列のラベルを桁指定して32ビットデータとして扱うときの表記方法を示します。

項目	表記	例
ビット型配列のラベル	ラベルの桁指定は、要素指定なしで指定します。 ラダー以外の言語では、K5~K8の範囲で指定してください。(ラダー以外の言語の場合、K1~K4の範囲の桁指定はANY16として扱われます。)	K8L_BOOL

桁指定の範囲

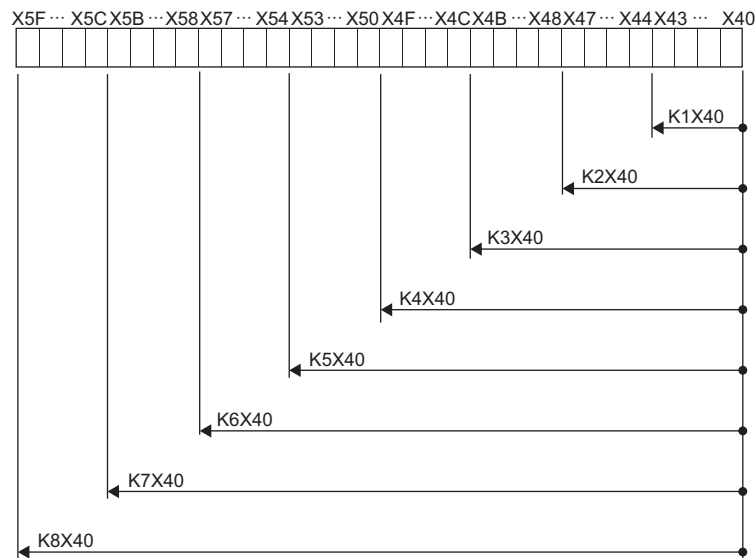
桁指定ごとの32ビットデータの範囲を示します。

桁指定	10進表記	16進表記
K1	0~15	0H~FH
K2	0~255	00H~FFH
K3	0~4095	000H~FFFH
K4	0~65535	0000H~FFFFH
K5	0~1048575	00000H~FFFFFH
K6	0~16777215	000000H~FFFFFFH
K7	0~268435455	0000000H~FFFFFFFH
K8	符号付き32ビットデータ: -2147483648~2147483647 符号なし32ビットデータ: 0~4294967295	00000000H~FFFFFFFFH

例

X40を桁指定した場合の対象点数は、下記になります。

- K1X40→X40~X43の4点が対象
- K2X40→X40~X47の8点が対象
- K3X40→X40~X4Bの12点が対象
- K4X40→X40~X4Fの16点が対象
- K5X40→X40~X53の20点が対象
- K6X40→X40~X57の24点が対象
- K7X40→X40~X5Bの28点が対象
- K8X40→X40~X5Fの32点が対象



実数データ(浮動小数点データ)

データサイズとデータの範囲

実数データには、32ビットで表現する単精度実数データと、64ビットで表現する倍精度実数データがあります。
実数データはビットデバイス以外のデバイスまたは単精度実数型ラベル、倍精度実数型ラベルにのみ格納可能です。

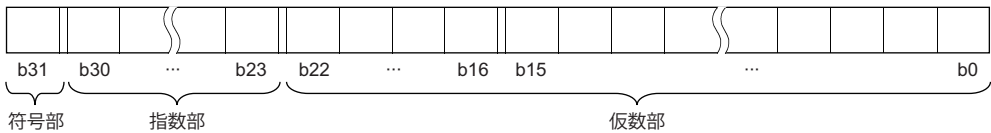
データ名称		データサイズ	値の範囲
単精度実数データ(単精度浮動小数点データ)	正数	32ビット(2ワード)	$2^{-126} \leq \text{実数} < 2^{128}$
	ゼロ		0
	負数		$-2^{128} < \text{実数} \leq -2^{-126}$
倍精度実数データ(倍精度浮動小数点データ)	正数	64ビット(4ワード)	$2^{-1022} \leq \text{実数} < 2^{1024}$
	ゼロ		0
	負数		$-2^{1024} < \text{実数} \leq -2^{-1022}$

単精度実数データの構成

単精度実数データは、符号部、仮数部、指数部により構成され、下記のように表現します。



単精度実数データを内部表現するときのビット構成と意味は、下記のとおりです。



■符号部(1ビット)

数値の正負を1ビットで表します。0は正数または0を示します。1は負数を示します。

■仮数部(23ビット)

単精度実数を2進数で1.XXXXX...×2^Nと表現したときのXXXXX...を示します。

■指数部(8ビット)

単精度実数を2進数で1.XXXXX...×2^Nと表現したときのNを示します。単精度実数における指数部の値とNの関係は下記のとおりです。

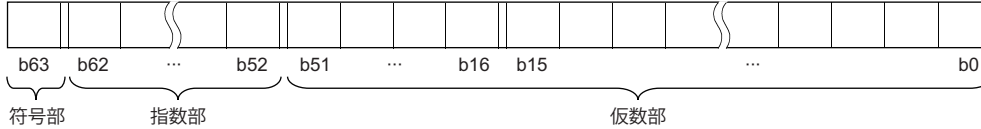
指数部(b24~b30)	FFH	FEH	FDH	...	81H	80H	7FH	7EH	...	02H	01H	00H
N	未使用	127	126	...	2	1	0	-1	...	-125	-126	未使用

倍精度実数データの構成

倍精度実数データは、符号部、仮数部、指数部により構成され、下記のように表現します。



倍精度実数データを内部表現するときのビット構成と意味は、下記のとおりです。



■符号部(1ビット)

数値の正負を1ビットで表します。0は正数または0を示します。1は負数を示します。

■仮数部(52ビット)

単精度実数を2進数で $1.XXXXX \dots \times 2^N$ と表現したときのXXXXX...を示します。

■指数部(11ビット)

単精度実数を2進数で $1.XXXXX \dots \times 2^N$ と表現したときのNを示します。単精度実数における指数部の値とNの関係は下記のとおりです。

指数部(b52~b62)	7FFH	7FEH	7FDH	...	401H	400H	3FFH	3FEH	...	02H	01H	00H
N	未使用	1023	1022	...	2	1	0	-1	...	-1021	-1022	未使用

注意事項

■エンジニアリングツールから単精度実数の入力値を設定する場合

エンジニアリングツールでは、単精度実数データは32ビットの単精度で処理するため、有効桁数は約7桁となります。単精度実数データの入力値が7桁を超えた場合、8桁目を四捨五入した値に丸めます。そのため、四捨五入後の値が-2147483648~2147483647の範囲を超えると、意図した値になりません。

例

入力値に「2147483647」を設定した場合、8桁目の「6」を四捨五入するため、「2147484000」として扱います。

例

入力値に「E1.1754943562」を設定した場合、8桁目の「3」を四捨五入するため、「E1.175494」として扱います。

入力値は下記の範囲としてください。下記の範囲を超えると変換エラーとなります。

小数表現: $0.0000000001 \leq \text{実数データの絶対値} \leq 999999900000.0$

指数表現: $1.175494351\text{E}-38 \leq \text{実数データの絶対値} \leq 3.402823466\text{E}+38$

■エンジニアリングツールから倍精度実数の入力値を設定する場合

エンジニアリングツールでは、倍精度実数データは64ビットの倍精度で処理するため、有効桁数は約15桁となります。倍精度実数データの入力値が15桁を超えた場合、16桁目を四捨五入した値に丸めます。そのため、四捨五入後の値が-2147483648~2147483647の範囲を超えると、意図した値になりません。

例

入力値に「2147483646.12345678」を設定した場合、16桁目の「6」を四捨五入するため、「2147483646.12346」として扱います。

例

入力値に「E1.7976931348623157+307」を設定した場合、16桁目の「5」を四捨五入するため、「E1.79769313486232+307」として扱います。

入力値は下記の範囲としてください。下記の範囲を超えると変換エラーとなります。

小数表現: $0.00000000000000000001 \leq \text{実数データの絶対値} \leq 9999999999999999000000.0$

指数表現: $2.22507385850721\text{E}-308 \leq \text{実数データの絶対値} \leq 1.79769313486231\text{E}+308$

エンジニアリングツールのモニタ機能では、コントローラの実数データをモニタできます。
実数データで「0」を表すときは、下記の範囲をすべて0にします。

- 単精度実数データの場合: b0~b31
- 倍精度実数データの場合: b0~b63

実数データの設定範囲は下記になります。^{*1}

- 単精度実数データの場合: $-2^{128} < [\text{単精度実数データ}] \leq -2^{126}$, 0 , $2^{-126} \leq [\text{単精度実数データ}] < 2^{128}$
- 倍精度実数データの場合: $-2^{1024} < [\text{倍精度実数データ}] \leq -2^{1022}$, 0 , $2^{-1022} \leq [\text{倍精度実数データ}] < 2^{1024}$

実数データで「-0」（最上位ビットのみ1）は指定しないでください。-0で実数演算を行うと演算エラーになります。

*1 オーバーフローおよびアンダーフロー時の動作や、特殊な値を入力したときの動作については、下記を参照してください。

 使用するコントローラのユーザーズマニュアル

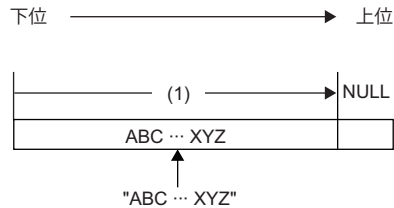
文字列データ

文字列データの形式

文字列データには下記の種類があり、NULLコードで終わる文字コード列を文字列として扱うことができます。

種類	文字コード	最終文字
文字列	ASCIIコード、シフトJISコード	NULL(00H)
Unicode文字列	Unicode(UTF-16(リトルエンディアン))	NULL(0000H)

文字列データをデバイスまたは配列に格納するときには、デバイス番号または配列の要素番号の下位から上位に向かって文字列データを格納します。



(1): 文字コード列

文字列の表記方法

ラダープログラムでの、文字列の表記方法を示します。

データ型		表記方法	表記例
文字列	STRING	文字列(ASCIIコード、シフトJISコード)、Unicode文字列をダブルクォーテーションマーク(")で囲む。	"ABC"
文字列[Unicode]	WSTRING		

STプログラムでの、文字列の表記方法を示します。

データ型		表記方法	表記例
文字列	STRING	文字列(ASCIIコード、シフトJISコード)をシングルクォーテーションマーク(')で囲む。	'ABC'
文字列[Unicode]	WSTRING	Unicode文字列をダブルクォーテーションマーク(")で囲む。	"ABC"

FBD/LDプログラムでの、文字列の表記方法を示します。

データ型		表記方法	表記例
文字列	STRING	文字列(ASCIIコード, シフトJISコード)をシングルクォーテーションマーク(')で囲む。	
文字列[Unicode]	WSTRING	Unicode文字列をダブルクォーテーションマーク(")で囲む。	

データの範囲

文字列データの範囲は下記となります。

種類	ラベルに設定可能な最大文字数	文字列定数に使用可能な最大文字数
文字列	半角255文字(終端のNULLは含みません。)	半角255文字(終端のNULLは含みません。)
Unicode文字列 ^{*1}	255文字(終端のNULLは含みません。)	255文字(終端のNULLは含みません。)

*1 Unicode文字列は基本多言語面までが使用できます。

データ格納に必要なワード数

文字列データはワードデバイスに格納できます。

文字列データの格納に必要なワード数は下記となります。

文字列のバイト数	文字列を格納するために必要なワード数	Unicode文字列を格納するために必要なワード数
0バイト	1[ワード]	1[ワード]
奇数バイト	(文字列のバイト数+1)÷2[ワード]	—(1文字が偶数バイトのため。)
偶数バイト	(文字列のバイト数÷2)+1[ワード]	文字数+1[ワード]

文字列データの格納先

文字列データの格納先のイメージを示します。

■文字列の場合

文字列の格納イメージにおいて、NULLはNULLコード(00H)を示します。

格納する文字列	D0から文字列データを格納したときのイメージ	ワード型ラベルの配列arrayA[0]から文字列データを格納したときのイメージ												
空文字列 (「”」または「”」)	D0 <table><tr><td>NULL</td><td>NULL</td></tr></table>	NULL	NULL	arrayA[0] <table><tr><td>NULL</td><td>NULL</td></tr></table>	NULL	NULL								
NULL	NULL													
NULL	NULL													
ABC	D0 <table><tr><td>B</td><td>A</td></tr></table> D1 <table><tr><td>NULL</td><td>C</td></tr></table>	B	A	NULL	C	arrayA[0] <table><tr><td>B</td><td>A</td></tr></table> arrayA[1] <table><tr><td>NULL</td><td>C</td></tr></table>	B	A	NULL	C				
B	A													
NULL	C													
B	A													
NULL	C													
ABCD	D0 <table><tr><td>B</td><td>A</td></tr></table> D1 <table><tr><td>D</td><td>C</td></tr></table> D2 <table><tr><td>NULL</td><td>NULL</td></tr></table>	B	A	D	C	NULL	NULL	arrayA[0] <table><tr><td>B</td><td>A</td></tr></table> arrayA[1] <table><tr><td>D</td><td>C</td></tr></table> arrayA[2] <table><tr><td>NULL</td><td>NULL</td></tr></table>	B	A	D	C	NULL	NULL
B	A													
D	C													
NULL	NULL													
B	A													
D	C													
NULL	NULL													

■Unicode文字列の場合

Unicode文字列の格納イメージにおいて、NULLはNULLコード(0000H)を示します。

格納する文字列	D0から文字列データを格納したときのイメージ	ワード型ラベルの配列arrayA[0]から文字列データを格納したときのイメージ																				
空文字列(「”」)	D0 <table><tr><td>NULL</td></tr></table>	NULL	arrayA[0] <table><tr><td>NULL</td></tr></table>	NULL																		
NULL																						
NULL																						
ABCD	<table><tr><td>D0</td><td>A</td></tr><tr><td>D1</td><td>B</td></tr><tr><td>D2</td><td>C</td></tr><tr><td>D3</td><td>D</td></tr><tr><td>D4</td><td>NULL</td></tr></table>	D0	A	D1	B	D2	C	D3	D	D4	NULL	<table><tr><td>arrayA[0]</td><td>A</td></tr><tr><td>arrayA[1]</td><td>B</td></tr><tr><td>arrayA[2]</td><td>C</td></tr><tr><td>arrayA[3]</td><td>D</td></tr><tr><td>arrayA[4]</td><td>NULL</td></tr></table>	arrayA[0]	A	arrayA[1]	B	arrayA[2]	C	arrayA[3]	D	arrayA[4]	NULL
D0	A																					
D1	B																					
D2	C																					
D3	D																					
D4	NULL																					
arrayA[0]	A																					
arrayA[1]	B																					
arrayA[2]	C																					
arrayA[3]	D																					
arrayA[4]	NULL																					

1.3 実行条件

実行条件の種類

命令の実行条件には、下記の種類があります。

実行条件		説明*1
ON時実行		ON中実行形の命令で、命令の前条件がONの間だけその命令を実行します。前条件がOFFの場合、その命令は実行せず、処理しません。
立上り実行		ON時1回実行形の命令で、命令の前条件の立上り時(OFF→ON)だけ命令を実行し、以降条件がONでもその命令を実行せず、処理しません。
OFF時実行		OFF中実行形の命令で、命令の前条件がOFFの間だけその命令を実行します。前条件がONの場合、その命令は実行せず、処理しません。
立下り実行		OFF時1回実行形の命令で、命令の前条件の立下り時(ON→OFF)だけ命令を実行し、以後条件がOFFでもその命令を実行せず、処理しません。
常時実行	—	常時実行の命令で、命令の前条件のON/OFFに関係なく常に実行します。前条件がOFFの場合、その命令はOFFという処理を行います。

*1 ST言語、FBD/LD言語では、命令中のENが命令の前条件になります。

各命令の実行条件

命令ごとに実行条件が異なります。各命令の実行条件は、各命令のページを参照ください。

ST言語、FBD/LD言語では、ENが実行条件となり、ENがTRUEの場合のみ実行します。ENOはENと同じ状態となります。

汎用ファンクション/ファンクションブロックでは、ENの有無で実行条件が異なります。EN無しの場合は常時実行です。EN付きの場合は、汎用ファンクション/ファンクションブロックのページを参照ください。

モーション制御ファンクションブロックの実行条件については、下記を参照してください。

📖 1367ページ 実行指令(Execute)型/有効(Enable)型

1.4 プログラミング時の注意事項

命令共通のエラー

下記に命令実行時にエラーとなる条件を示します。

エラー内容 ^{*1}	エラーコード
範囲外の入出力番号(000H~FFFH, 3E0H~3E3H以外)を指定した。	2800H
対象ユニットが存在しない入出力番号を指定した。	2801H
命令で指定できないユニットの入出力番号を指定した。	2803H
範囲外のネットワーク番号(1~239以外)を指定した。	2804H
存在しないネットワーク番号を指定した。	2805H
- 命令で指定したデバイスまたはラベルが使用可能な範囲を超えた。 - CPUパラメータのファイル設定でファイルレジスタを設定していない、またはプログラムで使用するファイルレジスタを設定していない状態で、ファイルレジスタにアクセスした。	2820H
- 命令で指定したユニットが持つバッファメモリの範囲を超えた。 - 命令で指定したユニットがバッファメモリを持たないユニットである。	2823H

*1 接点命令ではエラーは検出せず、演算結果が非導通状態となります。

命令実行時のデバイス、ラベルの範囲チェック

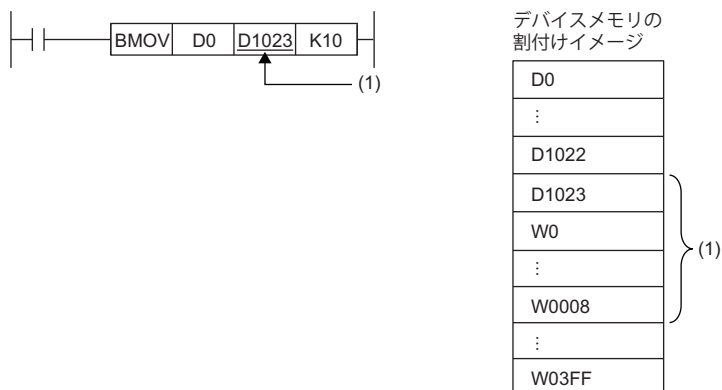
デバイスの範囲チェック

デバイスを命令で指定する場合、範囲のチェックを行うので、演算結果が該当デバイスの範囲内で収まるようなプログラムを作成してください。

該当デバイス範囲を超える指定をした場合、エラー (エラーコード: 2820H)が発生します。
デバイスに割り付けたラベルをプログラム中の命令で指定した場合も同様となります。

例

デバイスメモリの割付け上で、グローバルデバイスのD1023の次がW0になっている場合

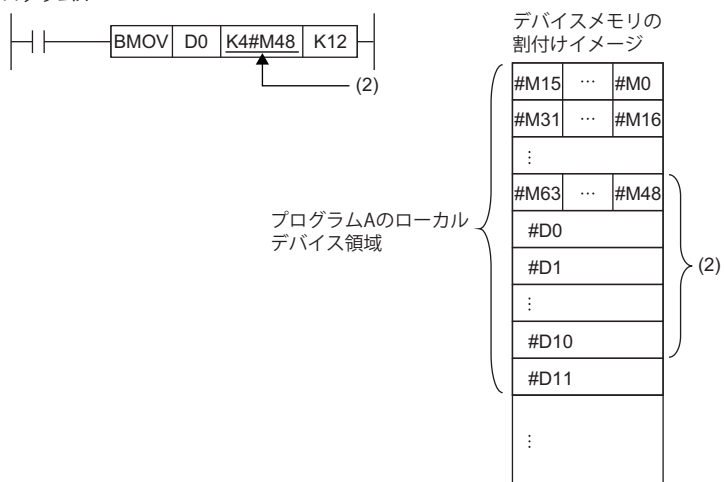


(1) 転送先はD1023~D1032に該当する範囲です。D1024~D1032が存在しないためエラーとなります。

例

ローカルデバイスをM0~M63, D0~D11の範囲で設定している場合

プログラムA



(2) 転送先はプログラムAのローカルデバイス#M48~#M239に該当する範囲です。#M64~#M239が存在しないためエラーになります。

ファイルレジスタの範囲チェック

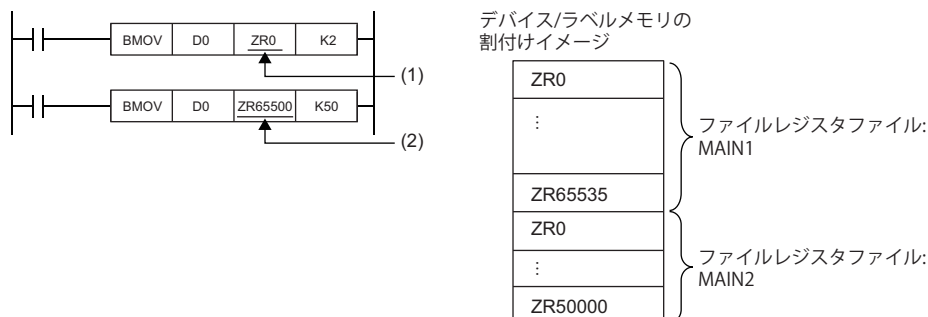
ファイルレジスタを命令で指定する場合、範囲のチェックを行うので、演算結果が該当ファイルレジスタの範囲内で収まるようなプログラムを作成してください。

ファイルレジスタ(ZR)で範囲を超える指定をした場合、エラー (エラーコード: 2820H)が発生します。

ファイルレジスタ(R)で使用するブロックNo.のファイルレジスタ範囲を超える指定をした場合、エラー (エラーコード: 2820H)が発生します。

例

ファイルレジスタ(ZR)を指定した場合

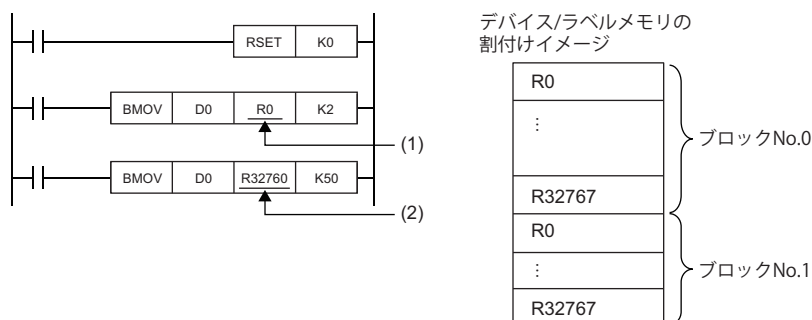


(1) 転送先はファイルレジスタMAIN1の範囲内です。ZR0~ZR1へデータが書き込まれます。

(2) 転送先はファイルレジスタMAIN1の範囲外です。ファイルレジスタMAIN1の領域範囲を超えるためエラーとなります。

例

ファイルレジスタ(R)を指定した場合



(1) 転送先はブロックNo.0のRデバイスの範囲内です。R0~R1へデータが書き込まれます。

(2) 転送先はブロックNo.0のRデバイスの範囲外です。ブロックNo.0のRデバイスの領域範囲を超えるためエラーとなります。

グローバル(ラッチ)ラベル, ローカル(ラッチ)ラベルの範囲チェック

グローバル(ラッチ)ラベル, ローカル(ラッチ)ラベルを命令で指定する場合, 範囲のチェックを行うので, 演算結果が該当ラベルの範囲内で収まるようなプログラムを作成してください。また, 桁指定, またはビット指定を行う場合も該当ラベルの範囲内で指定してください。

該当デバイス範囲を超える指定をした場合, エラー (エラーコード: 2820H)が発生します。

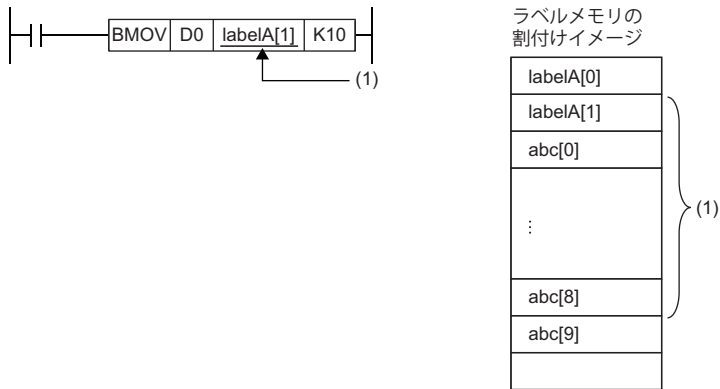
■配列型ラベル

引数に配列型ラベルを指定した場合は, 配列型ラベル全体をチェックします。また, 配列型ラベルの要素を指定した場合も, 配列型ラベルの範囲内をチェックします。

例

ラベルを下記の配列で設定している場合

ラベル名	データ型	配列要素数
labelA	ビット列[16ビット]	2
abc	ビット列[16ビット]	10



(1) 転送先はlabelA[1]から10点分に該当する範囲です。labelA[2]~labelA[9]が存在しないためエラーになります。

■構造体型ラベル

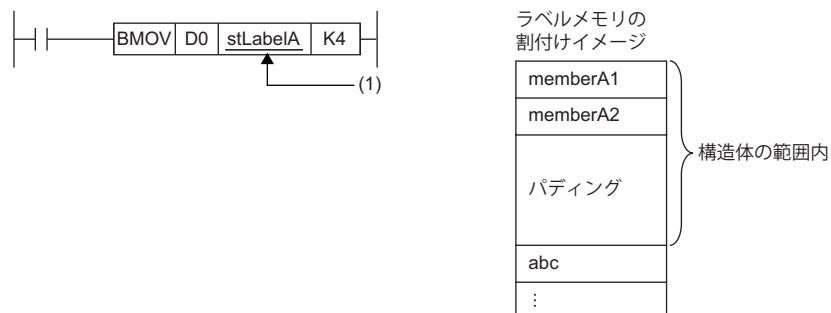
引数に構造体型ラベルを指定した場合は、構造体型ラベル全体をチェックします。また、構造体メンバを指定した場合は、該当メンバのデータ型に応じた範囲内をチェックします。

例

ラベルを下記の配列で設定している場合

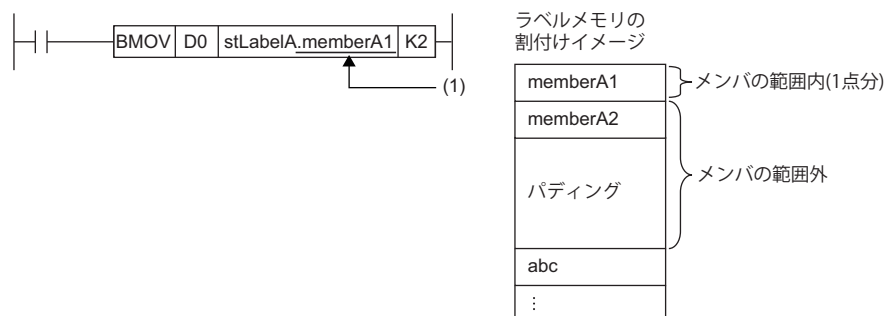
ラベル名	データ型		メンバ
stLabelA	構造体A	ワード/ビット列[16ビット]	memberA1
		ワード/ビット列[16ビット]	memberA2
abc	ワード/ビット列[16ビット]		—

・構造体型ラベルを指定した場合



(1) 転送先は構造体型ラベル"stLabelA"の先頭から4点分に該当する範囲です。構造体のパディング2点分を含めた範囲でアクセスするため、データが書き込まれます。

・メンバ"memberA1"を指定した場合



(1) 転送先はmemberA1の先頭から2点分に該当する範囲です。memberA1の範囲は1点分のためエラーになります。

ユニットラベル，ネットワークラベルの範囲チェック

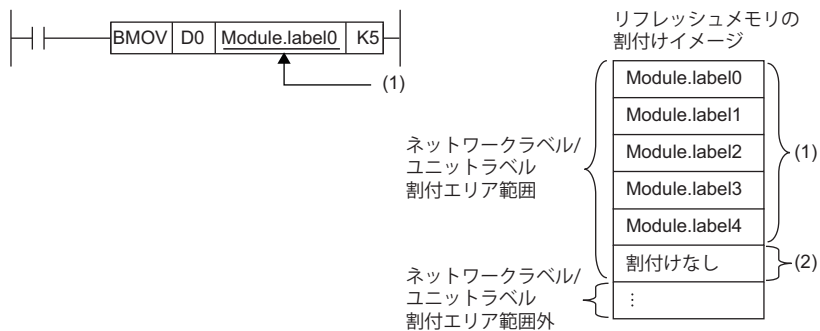
ユニットラベル，ネットワークラベルを命令で指定する場合，ネットワークラベル/ユニットラベル割付エリアで範囲のチェックを行うので，演算結果がネットワークラベル/ユニットラベル割付エリアの範囲内で収まるようなプログラムを作成してください。

該当ネットワークラベル/ユニットラベル割付エリアを超える指定をした場合，エラー（エラーコード: 2820H）が発生します。ネットワークラベル/ユニットラベル割付エリアの設定方法は使用するコントローラのユーザーズマニュアルを確認してください。

例

ユニットラベルを下記で設定している場合

インスタンス名	ラベル名	データ型	割付けデバイス
Module	label0	ワード[符号なし]	UV:0
	label1	ワード[符号なし]	UV:1
	label2	ワード[符号なし]	UV:2
	label3	ワード[符号なし]	UV:3
	label4	ワード[符号なし]	UV:4



- (1) 転送先はネットワークラベル/ユニットラベル割付エリアの範囲内です。Module.label0~Module.label4に書き込まれます。なお，データ型を跨いでいても書き込まれます。
- (2) 転送先の範囲外ですが，ネットワークラベル/ユニットラベル割付エリアの範囲内なので，このエリアへアクセスできます。

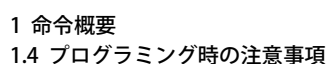
ロングタイマ、ロング積算タイマは、扱うデータが現在値の幅(32ビット)を超える場合、現在値の領域だけでなく前回値、接点・コイルの領域も使用して動作します。

現在値の一括転送にBMOV命令を使用する場合、現在値のみの一括転送はできないため、現在値、接点、コイルを一括転送し、転送後のデータにて現在値のみを使用してください。

現在値の一括転送にDMOV命令を使用する場合、現在値のみの転送をFOR~NEXT命令で繰返し、現在値を転送してください。

タイムデバイスの現在値を一括転送する場合

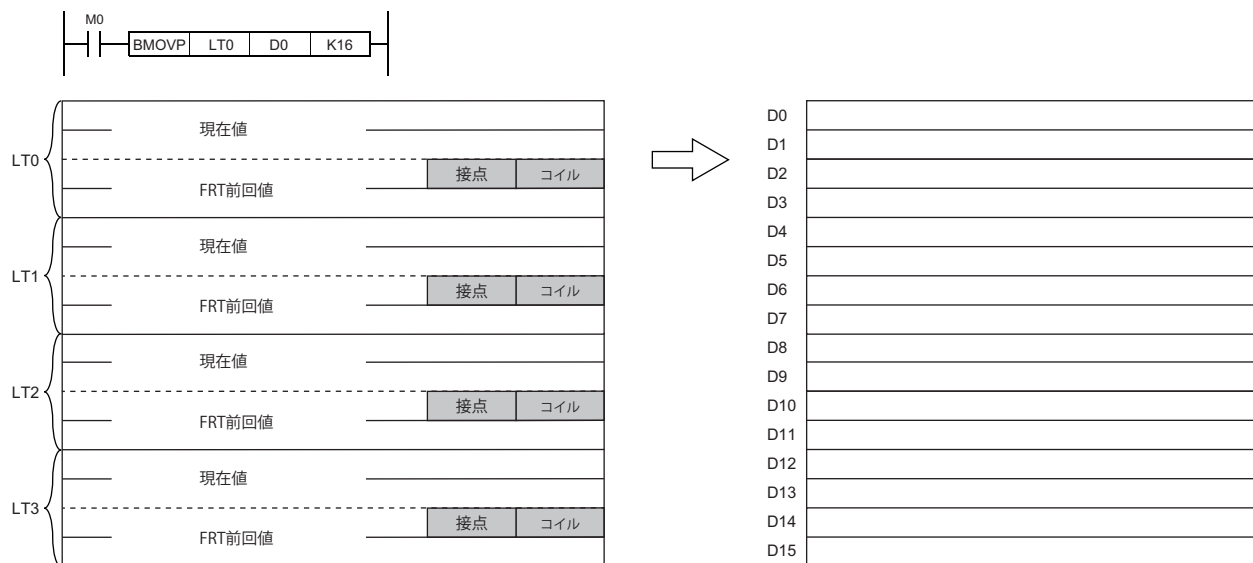
現在値の一括転送にBMOV命令を使用する場合は現在値のみを一括転送します。



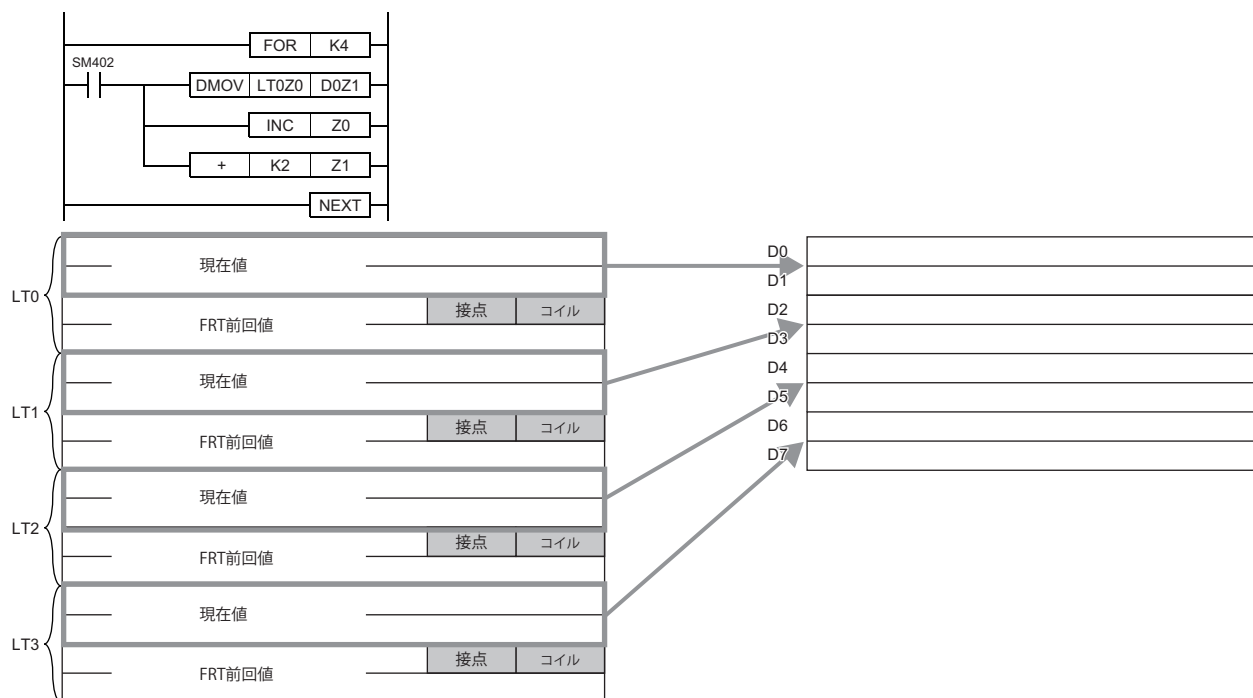
例

ロングタイマデバイスの現在値を一括転送する場合

現在値の一括転送にBMOV命令を使用する場合は現在値、接点、コイルを一括転送します。



現在値の一括転送にDMOV命令を使用する場合は現在値のみを一括転送します。



同期局に割り当てたリンクデバイスを命令引数に使用する場合の注意事項

CC-Link IE TSNの同期局に割り当てたリンクデバイスを命令引数に使用する場合、リンクリフレッシュを設定した上で、リフレッシュ先のデバイスまたはラベルを指定してください。命令引数にダイレクトアクセスデバイス(U□¥□□, J□¥□□)を指定した場合、リンクデバイスは正しく動作しません。

CC-Link-IE TSNの詳細については使用するコントローラのユーザズマニュアルを確認してください。

同一デバイスのOUT命令, SET/RST命令, PLS/PLF命令使用時の動作

同一デバイスを使用したOUT命令, SET/RST命令, PLS/PLF命令を1スキャンに複数実行した場合の動作について説明します。

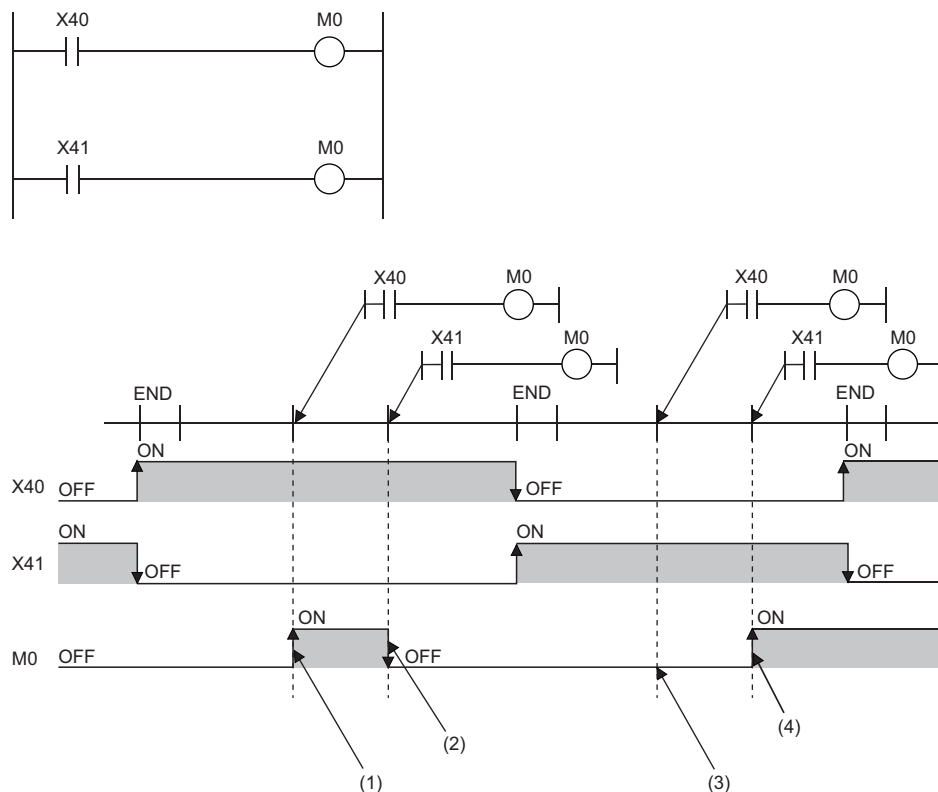
同一デバイスのOUT命令の場合

1スキャン中に同一デバイスのOUT命令を複数回実行しないでください。

1スキャン中に同一デバイスのOUT命令を複数回実行した場合は、各々のOUT命令実行時に、OUT命令までの演算結果により、指定デバイスがON/OFFします。

各々のOUT命令の実行時に、指定デバイスのON/OFFが決まるため、1スキャン中にON/OFFを繰り返すことがあります。

入力のX40とX41で、同一の内部リレー (M0)をON/OFFする回路を作成した場合の動作を示します。



- (1) X40がONのため、M0はONします。
- (2) X41がOFFのため、M0はOFFします。
- (3) X40がOFFのため、M0はOFFのままになります。
- (4) X41がONのため、M0はONします。

OUT命令で出力(Y)を指定すると、1スキャンの最後に実行したOUT命令のON/OFF状態が出力されます。

同一デバイスのSET/RST命令を使用した場合

■SET命令の場合

SET命令は実行指令がONのとき指定デバイスをONし、実行指令がOFFのときは無処理です。

このため同一デバイスのSET命令を1スキャンに複数回実行した場合は、1つでも実行指令がONしていると、指定デバイスはONになります。

■RST命令の場合

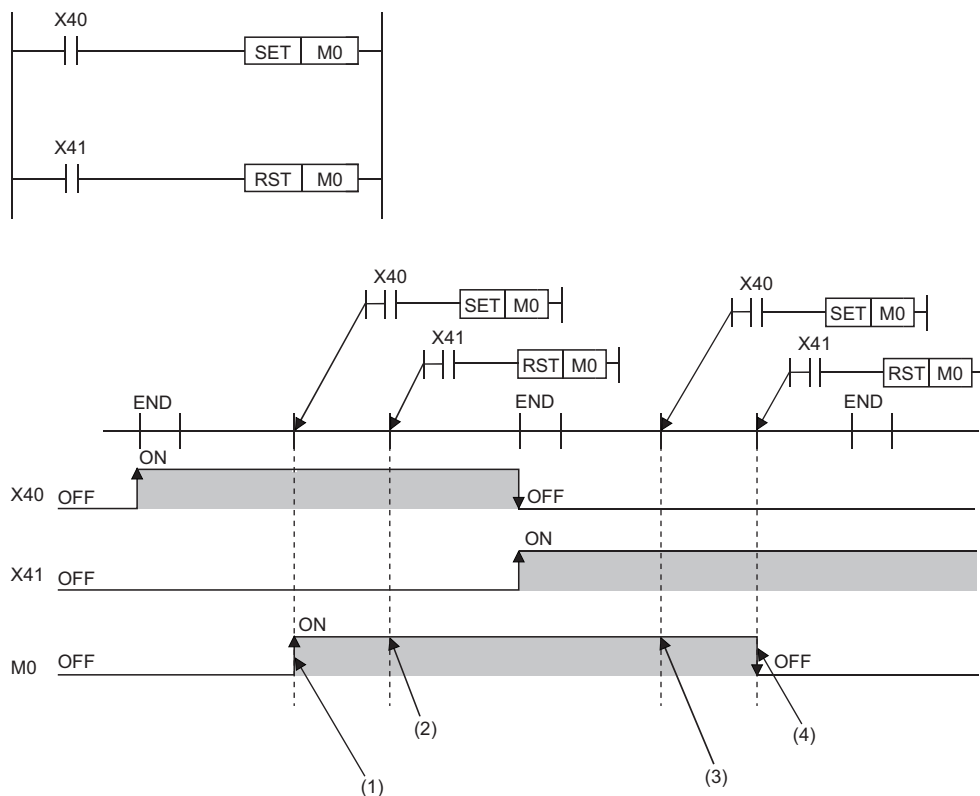
RST命令は実行指令がONのとき指定デバイスをOFFし、実行指令がOFFのときは無処理です。

このため同一デバイスのRST命令を1スキャンに複数回実行した場合は、1つでも実行指令がONしていると、指定デバイスはOFFになります。

■1スキャンに同一デバイスのSET命令とRST命令がある場合

1スキャンに同一デバイスのSET命令とRST命令がある場合、SET命令は実行指令がONのとき指定デバイスをONし、RST命令は実行指令がONのとき指定デバイスをOFFします。

SET命令とRST命令の実行指令がOFFの場合は、指定デバイスのON/OFF状態は変化しません。



- (1) X40がONのため、M0はONします。
- (2) X41がOFFのため、M0はONのままになります。(RST命令は無処理になります。)
- (3) X40がOFFのため、M0はONのままになります。(SET命令は無処理になります。)
- (4) X41がONのため、M0はOFFします。

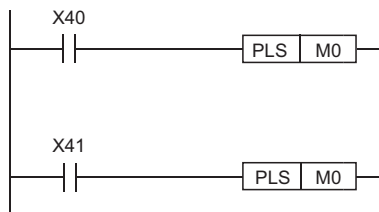
SET/RST命令で出力(Y)を指定すると、1スキャンの最後に実行したSET/RST命令のON/OFF状態が出力されます。

同一デバイスのPLS命令を使用した場合

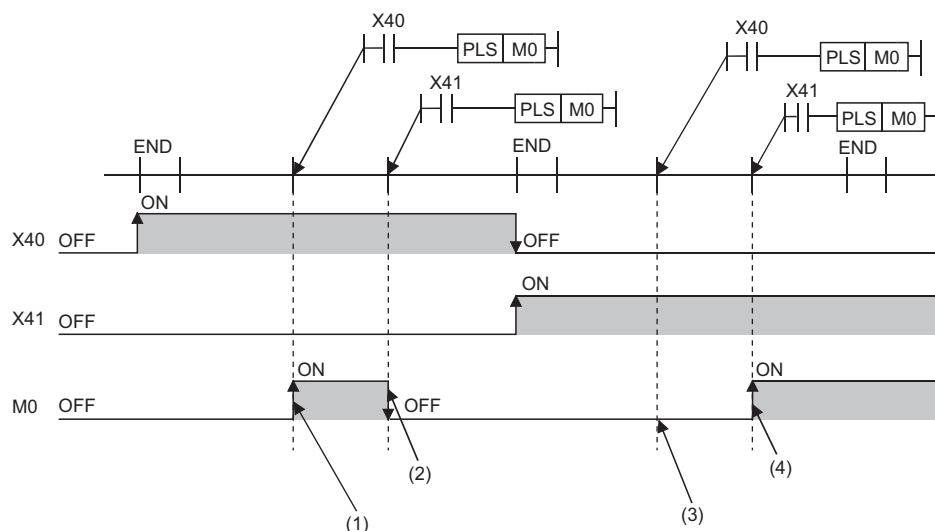
PLS命令は、実行指令のOFF→ON時に指定デバイスをONします。OFF→ON以外(OFF→OFF, ON→ON, ON→OFF)のとき、指定デバイスをOFFにします。

同一デバイスのPLS命令を1スキャンに複数実行した場合、各PLS命令の実行指令がOFF→ONのときに、指定デバイスをONします。OFF→ON以外のとき指定デバイスをOFFにします。

このため同一デバイスのPLS命令を1スキャンに複数回実行した場合は、PLS命令でONしたデバイスが、1スキャンONしないことがあります。

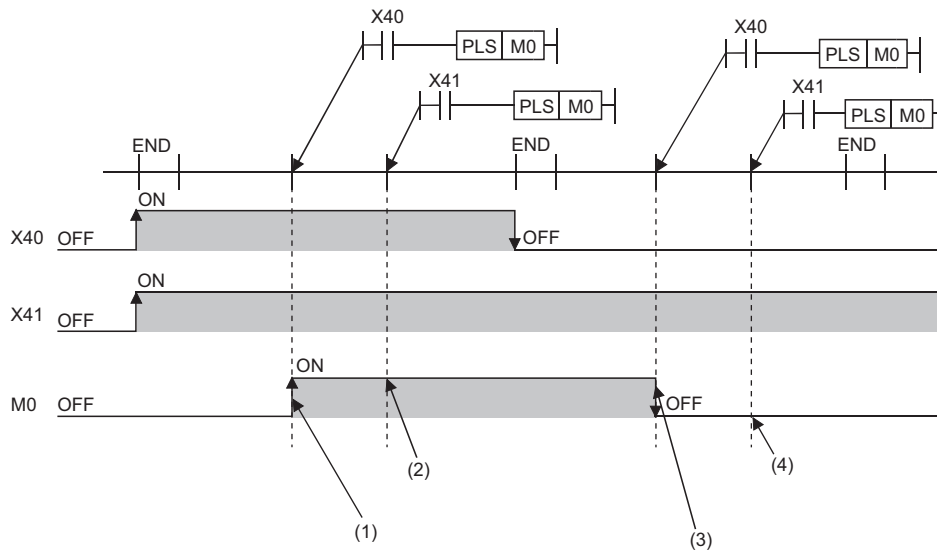


- X40とX41のON/OFFタイミングが異なるとき(指定デバイスが1スキャンONしない。)



- (1) X40がOFF→ONのため、M0はONします。
- (2) X41がOFF→ON以外のため、M0はOFFします。
- (3) X40がOFF→ON以外のため、M0はOFFのままになります。
- (4) X41がOFF→ONのため、M0はONします。

- X40とX41のOFF→ONが同一タイミングのとき



- (1) X40がOFF→ONのため、M0はONします。
- (2) X41がOFF→ONのため、M0はOFFのままになります。
- (3) X40がOFF→ON以外のため、M0はOFFします。
- (4) X41がOFF→ON以外のため、M0はOFFのままになります。

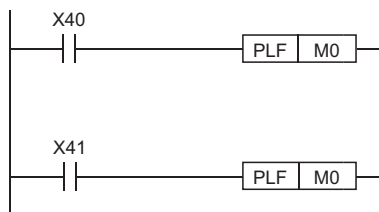
PLS命令で出力(Y)を指定すると、1スキャンの最後に実行したPLS命令のON/OFF状態が出力されます。

同一デバイスのPLF命令を使用した場合

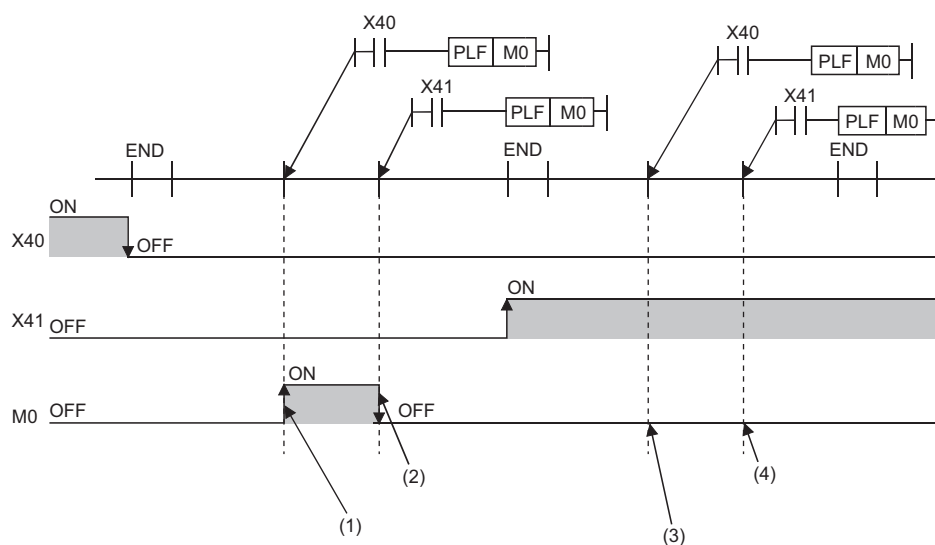
PLF命令は、実行指令のON→OFF時に指定デバイスをONします。ON→OFF以外(OFF→OFF, OFF→ON, ON→ON)のとき、指定デバイスをOFFにします。

同一デバイスのPLF命令を1スキャンに複数実行した場合、各PLF命令の実行指令がON→OFFのときに、指定デバイスをONします。ON→OFF以外のとき指定デバイスをOFFにします。

このため同一デバイスのPLF命令を1スキャンに複数回実行した場合は、PLF命令でONしたデバイスが、1スキャンONしないことがあります。

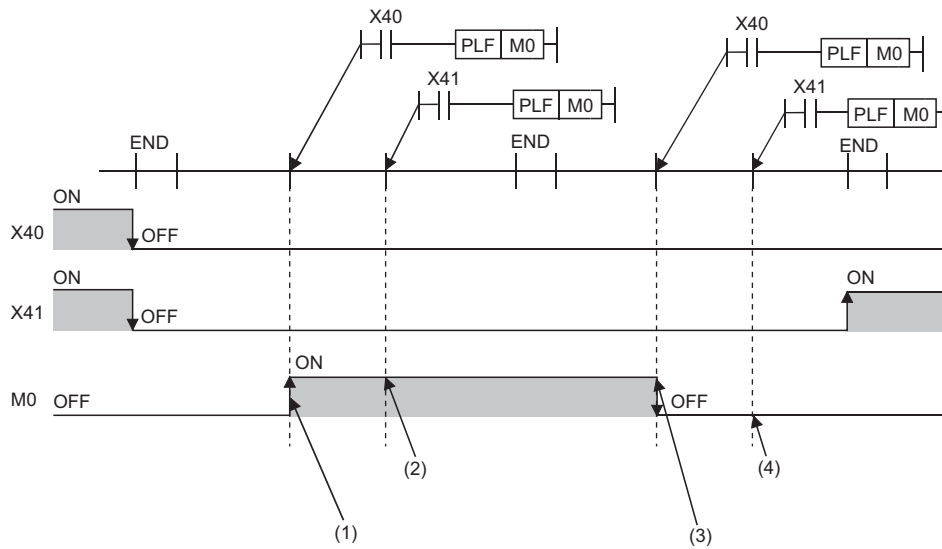


- X40とX41のON/OFFタイミングが異なるとき(指定デバイスが1スキャンONしない。)



- (1) X40がON→OFFのため、M0はONします。
- (2) X41がON→OFF以外のため、M0はOFFします。
- (3) X40がON→OFF以外のため、M0はOFFのままになります。
- (4) X41がON→OFF以外のため、M0はOFFのままになります。

- X40とX41のON→OFFが同一タイミングのとき



- (1) X40がON→OFFのため、M0はONします。
- (2) X41がON→OFFのため、M0はONのままになります。
- (3) X40がON→OFF以外のため、M0はOFFします。
- (4) X41がON→OFF以外のため、M0はOFFのままになります。

PLF命令で出力(Y)を指定すると、1スキンの最後に実行したPLF命令のON/OFF状態が出力されます。

ファイルレジスタ使用時の制約事項

リフレッシュデバイスにファイルレジスタを指定した場合、下記の制約事項について注意してください。

RSET命令でブロックNo.を切り換えた場合

RSET命令でブロックNo.を切り換えた場合、下記に注意してください。

- 切り換えたブロックNo.のファイルレジスタ(R)にリフレッシュされます。
- リフレッシュ直前でのブロックNo.のファイルレジスタ(R)にリフレッシュされます。

リフレッシュデータを読み書きしたい場合は、リフレッシュ直前でのブロックNo.を指定してください。

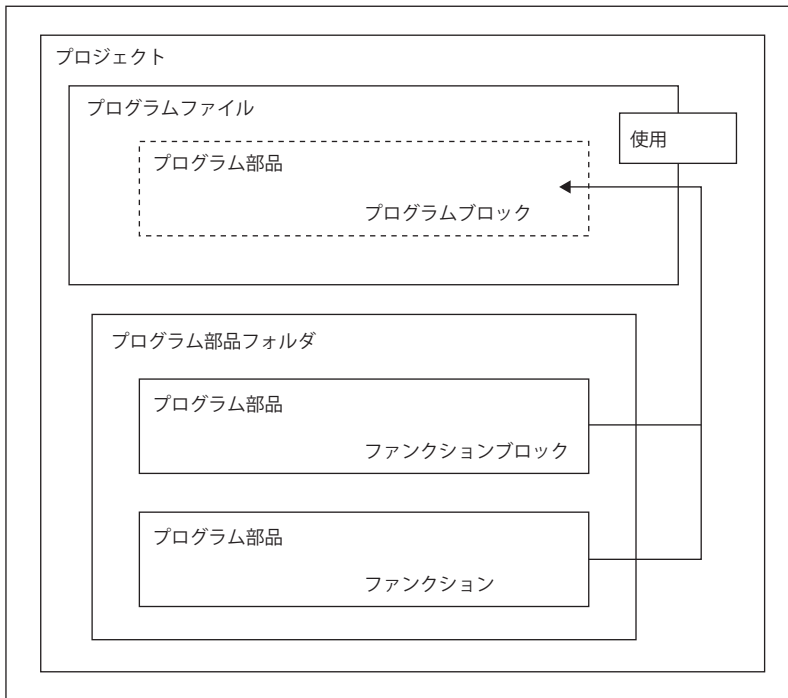
2 プログラム部品

プログラム部品には、下記の種類があります。

- プログラムブロック
- ファンクション
- ファンクションブロック

各プログラム部品は制御に合わせたプログラム言語で処理を記述できます。ファンクションとファンクションブロックでは、ラダー言語、ST言語またはFBD/LD言語で処理を記述できます。

ファンクションとファンクションブロックは、プログラムブロックから呼び出されて、実行されます。



Point

プログラムの部品化とは、プログラムを階層化したときの下位の処理を、処理内容や機能ごとにいくつかの単位に分け、単位ごとのプログラムを作成することです。

プログラムを部品化することで、独立性が高くなり、追加や交換が容易にできるように設計できます。

部品化するとよい処理には、下記のようなものがあります。

- プログラム中で繰り返し記述される処理
- 1つの機能として切り分けられる処理

本項では、ラベルを使用して各プログラム部品の説明をしています。

各プログラム部品のプログラム本体(ワークシート)では、デバイスも使用可能です。デバイスの詳細については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

Point

ST言語とFBD/LD言語では、一つのプログラム部品内に最大32のワークシートを作成できます。

複数ワークシートの実行順序は、エンジニアリングツールの“ワークシート実行順設定”画面から設定します。(📖GX Works3 オペレーティングマニュアル)

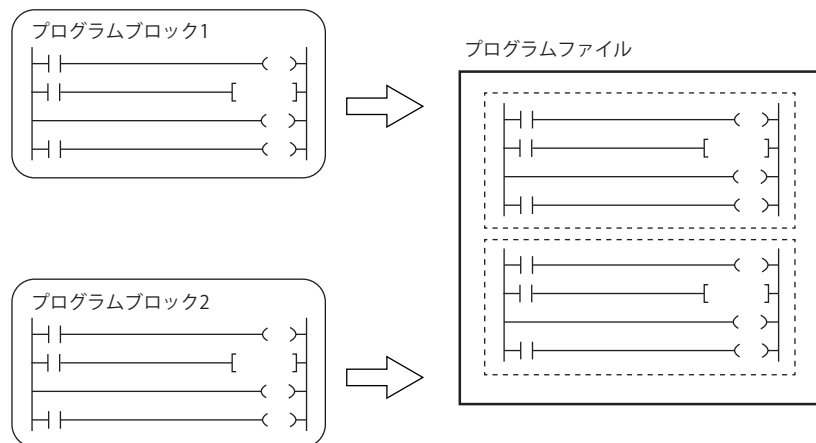
2.1 プログラムブロック

プログラムブロックは、プログラムを構成する単位になります。

プログラムブロックはプログラムファイル内に複数作成でき、プログラムファイル設定で指定した順番で実行されます。プログラムファイル設定で順番を指定していない場合は、プログラムブロック名の順番(昇順)に実行されます。

機能や処理ごとにプログラムブロックを切り分けると、プログラムの順番変更や交換が容易にできるように設計できます。

プログラムブロックのプログラム本体は、登録先のプログラムごとにプログラムファイルに格納します。



プログラムブロックの分割

プログラムブロックごとに、メインルーチンプログラム、サブルーチンプログラム、割込みプログラムを分けて作成できます。

種類	内容
メインルーチンプログラム	プログラムのステップ0からFEND命令までのプログラムです。
サブルーチンプログラム	ポインタ(P)からRET命令までのプログラムです。 サブルーチンコール命令(CALL命令、ECALL命令など)でコールされた場合のみ実行します。
割込みプログラム	割込みポインタ(I)からIRET命令までのプログラムです。 割込み要因が発生すると、その割込みポインタ番号に対応する割込みプログラムを実行します。

Point

- サブルーチンプログラムおよび割込みプログラムはFEND命令以降に作成します。FEND命令以降のプログラムはメインルーチンプログラムとして実行されません。例えば、2つめのプログラムブロックの最後でFEND命令を使用した場合、3つめのプログラムブロック以降はサブルーチンプログラムまたは割込みプログラムとなります。
- 分かりやすいプログラムを作成するために、対になるFOR命令とNEXT命令や、MC命令とMCR命令は一つのプログラムブロック内で使用してください。
- シンプルなプログラムの場合、1つのプログラムブロック内にメインルーチンを記述するだけで、コントローラで実行させることができます。

サブルーチンプログラム、割込みプログラムの詳細については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

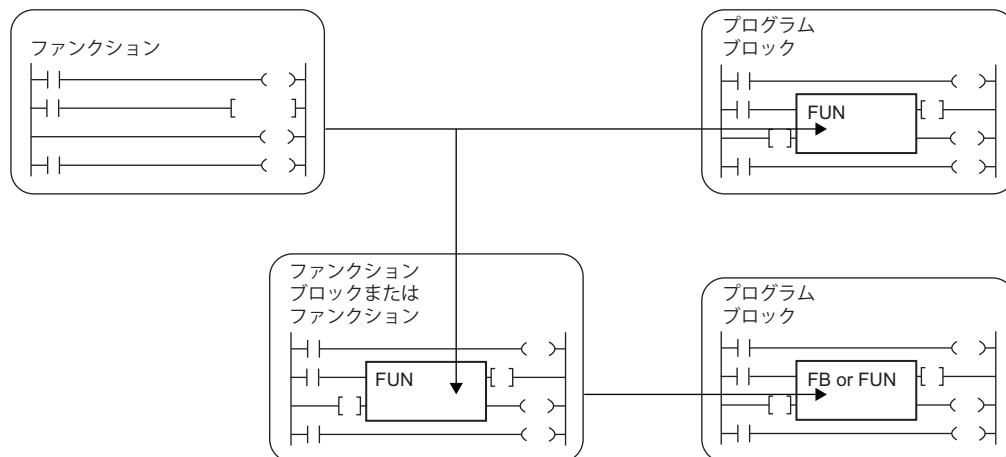
2.2 ファンクション(FUN)

ファンクションは、プログラムブロック、ファンクションブロック、および他のファンクションから使用されるプログラム部品です。

ファンクションは実行完了後、呼び出し元に値を渡します。その値を戻り値と呼びます。

ファンクションは同じ入力に対して、処理の結果として常に同じ戻り値を出力します。

単純で独立した、よく使うアルゴリズムを定義しておく、有効に再利用できます。



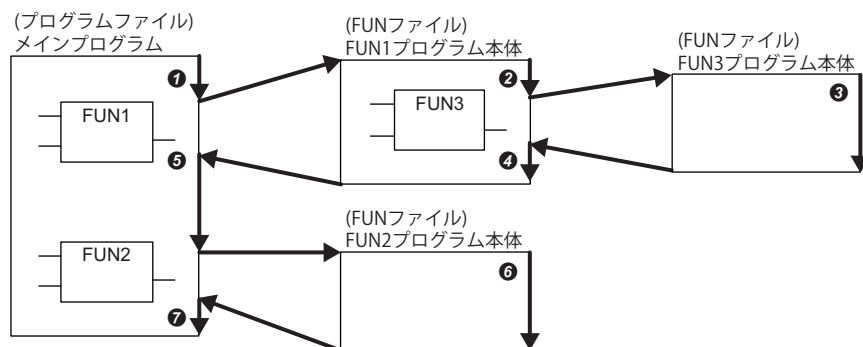
動作概要

ファンクションは、プログラム本体をFUNファイル内に格納し、実行時に呼び出し元プログラムからFUNファイル内のプログラム本体を呼び出して実行します。

例

メインプログラムからFUN1およびFUN2を呼び出し、FUN1はさらにFUN3を呼び出す場合(ネスト数: 3回)

①～⑦は、実行の流れ(順番)を示します。

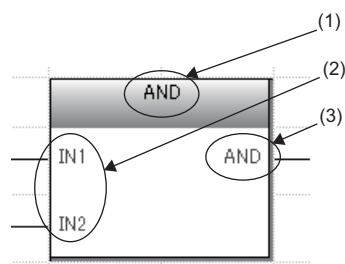


サブルーチン型ファンクションブロック、ファンクションのすべてを合わせて、32回までネストできます。

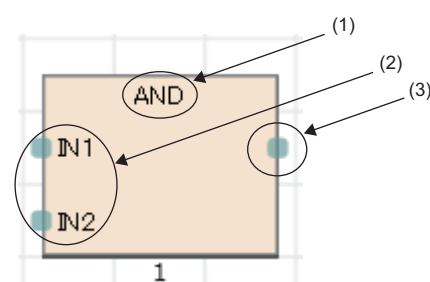
入力変数と出力変数について

ファンクションには、入力変数、出力変数を定義することができます。出力変数は、戻り値とは別の出力データを割り当てることができます。

ラダー言語の場合



FBD/LD言語の場合



- (1) ファンクション名
- (2) 入力変数
- (3) 出力変数

ファンクションの戻り値の名称は表示されません。

入力変数はVAR_INPUTのクラス、出力変数はVAR_OUTPUTのクラスで設定します。

Point

ファンクションで定義した変数は、ファンクションの呼び出しごとに上書きされます。呼び出しごとに変数の値を保持したい場合は、ファンクションブロックを使用する、または出力変数を異なる変数に退避するなどしてプログラミングしてください。

EN/ENOについて

ファンクションにEN(イネーブル入力)、ENO(イネーブル出力)を付けることにより、実行処理の制御ができます。

- ENには、ファンクションの実行条件とするブール型の変数を設定します。
- EN付きファンクションは、ENの実行条件がTRUEの場合のみ実行します。
- ENOには、ファンクションの実行結果を出力するブール型の変数を設定します。

ENの状態によるENOと演算結果の内容は下記になります。

EN	ENO	演算結果
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE	不定値

Point

- ラダー言語、FBD/LD言語のプログラムでは、ENOへの出力ラベルの設定は、必須ではありません。
- 汎用ファンクションでEN/ENOを使用する場合、EN付きファンクションは「ファンクション名_E」となります。

プログラムの作成

ファンクションのプログラムを作成する場合、下記の操作から実施します。

🔗 [ナビゲーションウィンドウ]⇒[FB/FUN]⇒右クリック⇒[データ新規作成]
 “基本設定”の“データ型”において“ファンクション”を選択

作成したプログラムは、FUNファイルに格納します。

🔗 [CPUパラメータ]⇒[プログラム設定]⇒[FB/FUNファイル設定]

1つのFUNファイルには、作成したプログラムを最大64個まで格納できます。

ファンクション内では、立上り実行/立下り実行命令は使用できません。

プログラムの作成に関連する内容については、下記を参照してください。

項目	参照先
ファンクションの作成方法	📖 GX Works3 オペレーティングマニュアル
コントローラに書き込めるFB/FUNファイルの数	📖 使用するコントローラのユーザーズマニュアル

■使用可能なデバイス/ラベル

ファンクションのプログラムで使用可能なデバイスおよびラベルの一覧は下記になります。

○: 使用可, △: 命令でのみ使用可(プログラムのステップを示すラベルとしては使用不可), ×: 使用不可

デバイス/ラベルの種類		使用可否
ラベル(ポインタ型以外)	グローバルラベル	×
	ローカルラベル	○*1
ラベル(ポインタ型)	ポインタ型グローバルラベル	×
	ポインタ型ローカルラベル	○
デバイス	グローバルデバイス	○
	ローカルデバイス	×
ポインタ	グローバルポインタ	△
	ローカルポインタ	×

*1 下記のデータ型は使用できません。

タイマ、積算タイマ、カウンタ、ロングタイマ、ロング積算タイマ、ロングカウンタ

Point

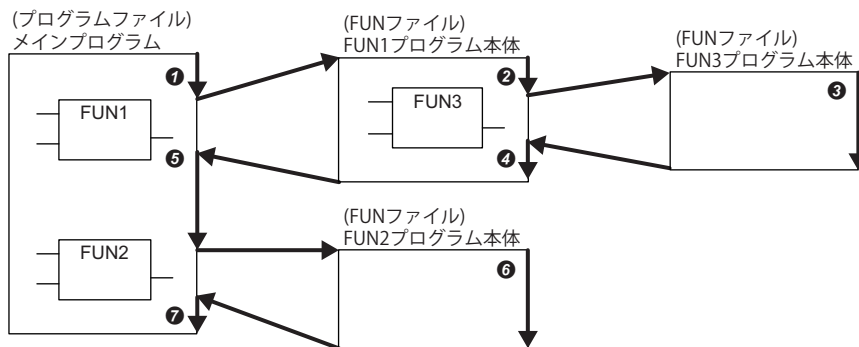
- ファンクションの戻り値は、ファンクション内でファンクション名をラベルとしてプログラムすることで設定できます。ファンクション名は、ラベルとして設定する必要はありません。ファンクションのプロパティにて、“戻り値の型”で設定したデータ型で使用できます。

ファンクションで定義するラベル

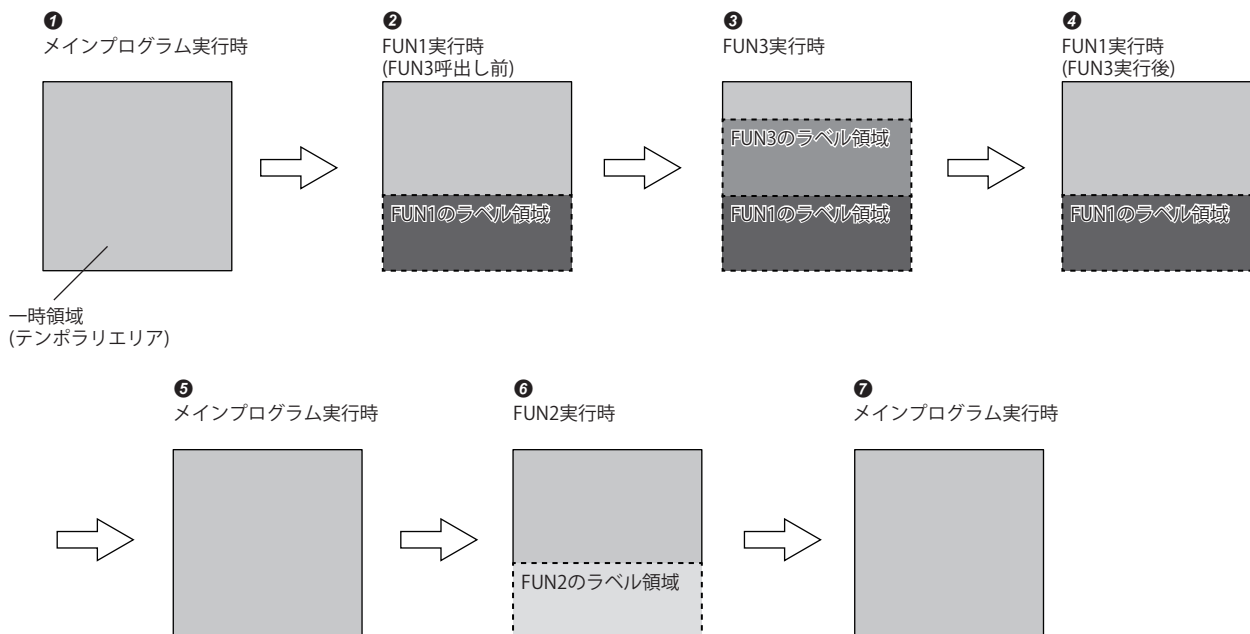
ファンクションで定義するラベルの割付け先は、ファンクション実行時にメモリ内の一時領域(テンポラリエリア)に確保し、実行完了時に解放します。

例

メインプログラムからFUN1およびFUN2を呼び出し、FUN1はさらにFUN3を呼び出す場合
(①~⑦は実行の流れ(順番)を示します)



上記のファンクションの実行動作に対するラベル割付けの状態は、下記になります。



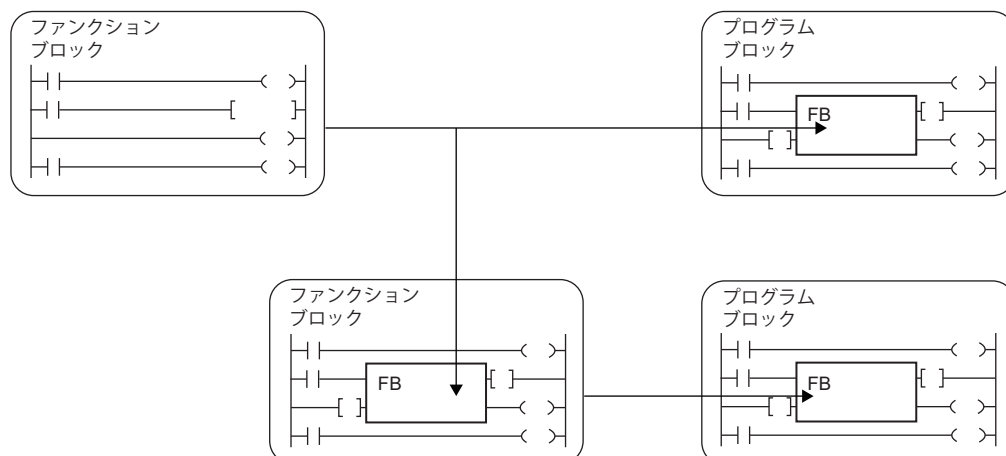
ファンクションで定義可能なラベルのクラスは、VAR, VAR_CONSTANT, VAR_INPUT, VAR_OUTPUTです。

Point

ファンクションで定義するラベルは不定値となるため、最初にアクセスする際にプログラムで初期化する必要があります。

2.3 ファンクションブロック(FB)

ファンクションブロックは、プログラムブロックや他のファンクションブロックから使用されるプログラム部品です。



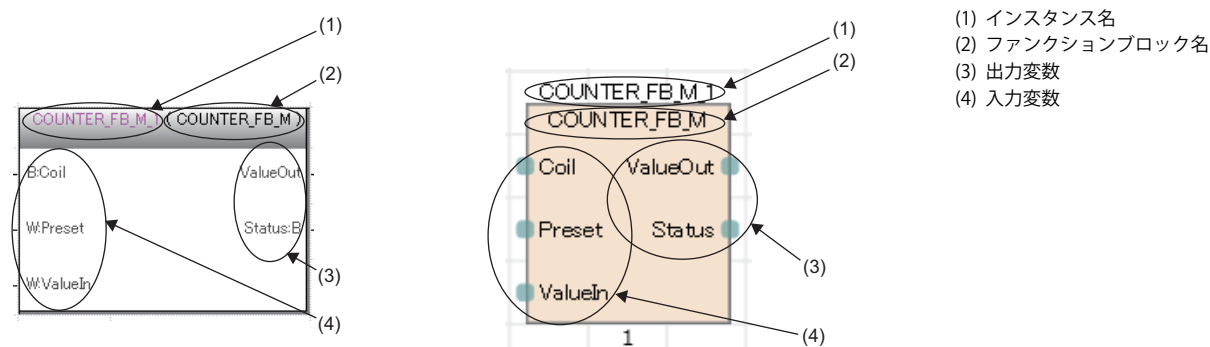
ファンクションブロックはファンクションと異なり、戻り値を持ちません。

ファンクションブロックは変数に値を保存できるので、入力された状態や処理結果が保持されます。

保持した値を次の処理に用いるため、同じ入力値でも毎回同じ結果を出力するとは限りません。

ラダー言語の場合

FBD/LD言語の場合



また、ファンクションブロックをプログラムで使うためには、インスタンスを定義する必要があります。

📖 68ページ インスタンス

Point

• ユニットFBの詳細は、下記を参照してください。

📖 使用するユニットのFBリファレンス

動作概要

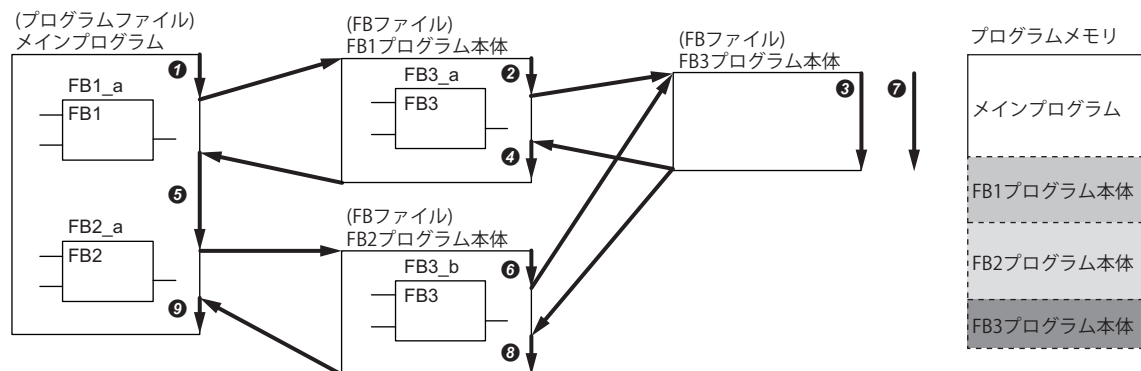
■サブルーチン型ファンクションブロック

サブルーチン型ファンクションブロックは、プログラム本体をFBファイル内に格納し、実行時に呼出し元プログラムからFBファイル内のプログラム本体を呼び出して実行します。

例

メインプログラムからFB1_aおよびFB2_aを呼び出し、FB1_aからはさらにFB3_aを、FB2_aからはさらにFB3_bを呼び出す場合(ネスト数: 3回)

①～⑨は実行の流れ(順番)を示します。

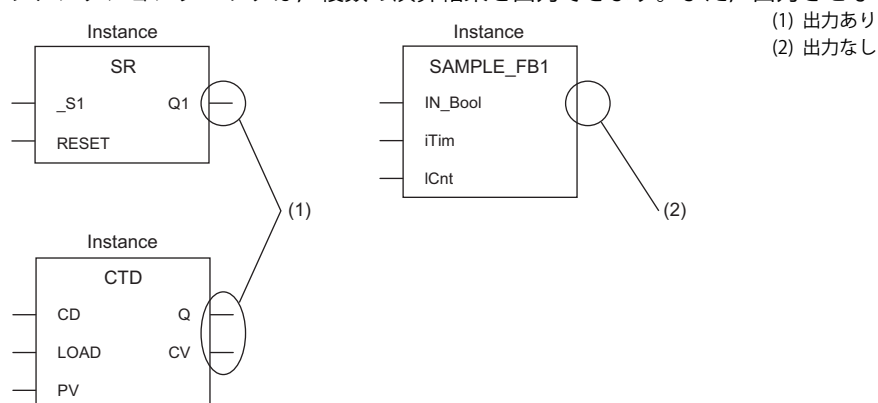


サブルーチン型ファンクションブロック、ファンクションのすべてを合わせて、32回までネストできます。

入力変数、出力変数、入出力変数について

ファンクションブロックには、入力変数、出力変数、入出力変数を定義する必要があります。

ファンクションブロックは、複数の演算結果を出力できます。また、出力させないこともできます。

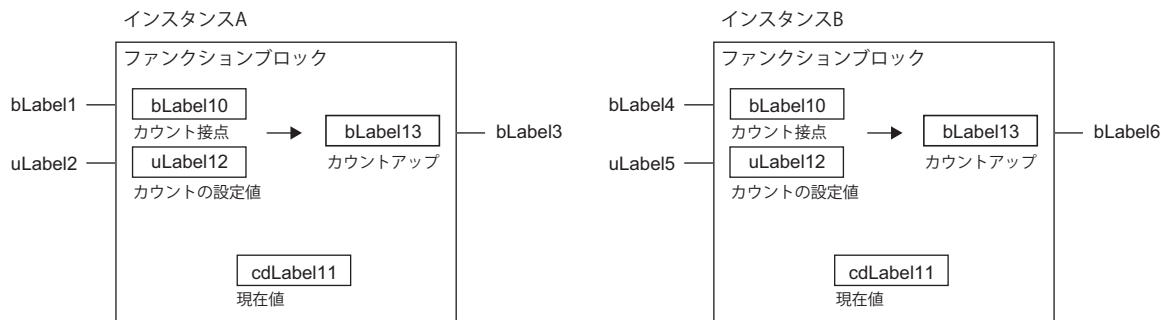


入力変数はVAR_INPUTのクラス、出力変数はVAR_OUTPUTとVAR_OUTPUT_RETAINのクラス、入出力変数はVAR_IN_OUTのクラスで設定します。

内部変数について

ファンクションブロックは内部変数を使います。内部変数には、ファンクションブロックのインスタンスごとに異なる領域にラベルが割り付けられます。同じラベル名であっても、インスタンスごとに異なる状態を保持できます。

例



入力変数がONになるとカウントを開始し、内部変数に保持した現在値が設定値に到達すると、出力変数をONするファンクションブロックです。同じファンクションブロックでも、インスタンスAとインスタンスBはそれぞれ独自の状態を保持するため、出力のタイミングは異なります。

内部変数はVAR, VAR_CONSTANT, VAR_RETAINのクラスで設定します。

外部変数と公開変数について

ファンクションブロックは外部変数(グローバルラベル)と公開変数が使用できます。

公開変数はVAR_PUBLIC, VAR_PUBLIC_RETAINのクラスで設定します。

インスタンス

■インスタンスとは

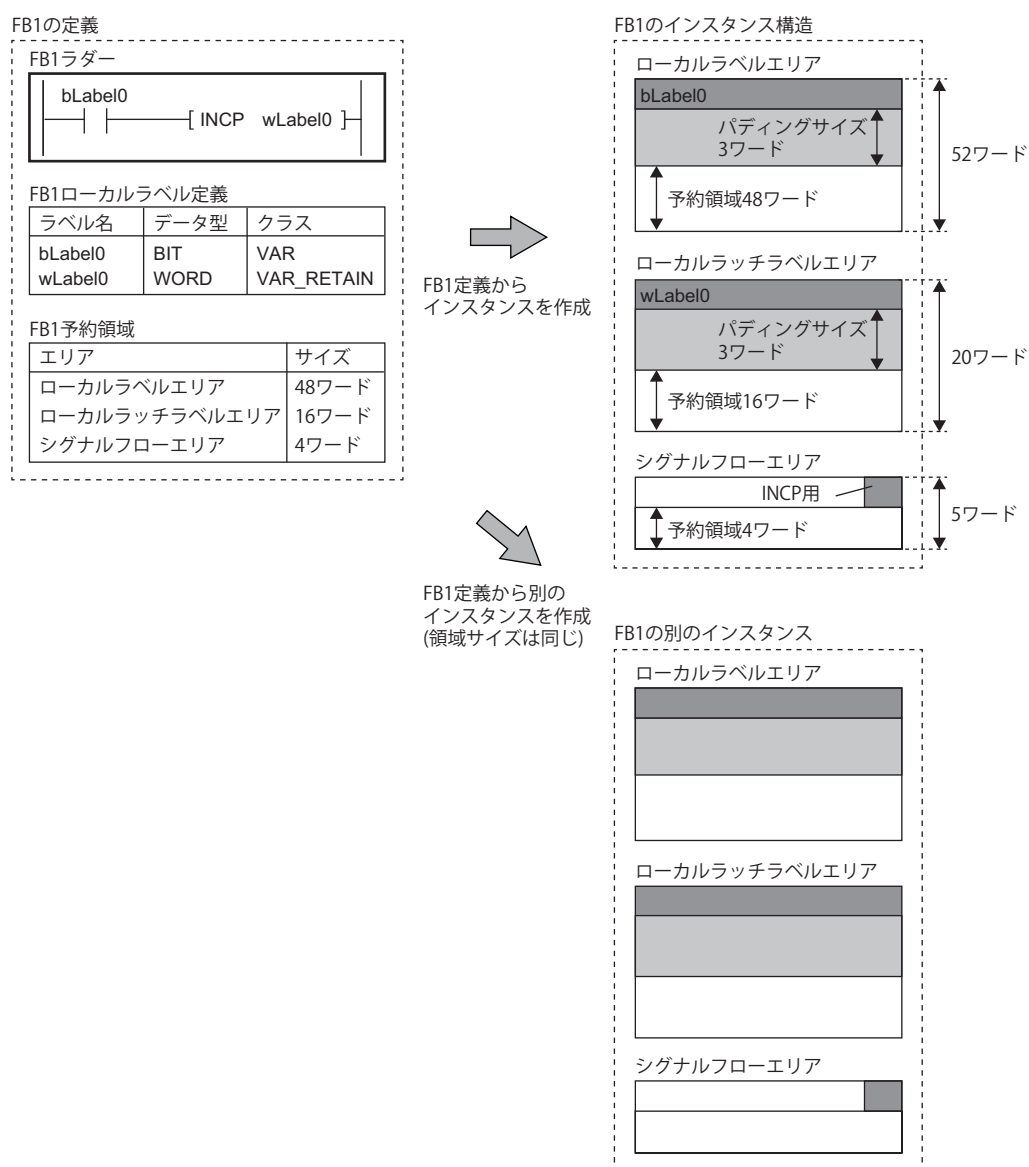
ファンクションブロックのインスタンスとは、ファンクションブロックの定義を元に割り付けたラベルです。一つのファンクションブロック定義から複数のインスタンスを作成することができます。

インスタンスは下記で構成されています。

項目	内容
ローカルラベルエリア	ファンクションブロックのローカルラベルを割り付けるエリアです。
ローカルラッチラベルエリア	ファンクションブロックのラッチ属性のローカルラベルを割り付けるエリアです。
シグナルフローエリア	ファンクションブロック定義内の命令が使用するシグナルフローを割り付けるエリアです。

例

インスタンスの構成



ローカルラベルエリアおよびローカルラッチラベルエリアは、4ワード単位でラベルの使用エリアを確保するため、上記の例では3ワード分(パディングサイズ)を確保しています。

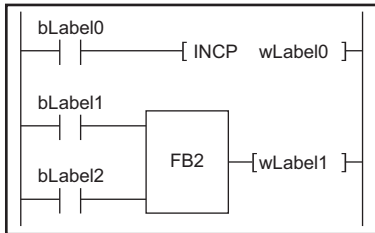
各エリアには予約領域が確保されます。予約領域は、変換やRUN中書込みなどでラベルの割付を維持したまま、ローカルラベルやシグナルフローメモリを参照する命令、ファンクションブロックのインスタンスを追加/変更するための領域です。追加/変更対象のデータ型に応じた領域を確保できない場合、全変換または変換+RUN中書込みが必要になります。

例

ファンクションブロックをネストしたインスタンスの構成(FB2の予約領域を変更してある場合)

FB1の定義

FB1ラダー



FB1ローカルラベル定義

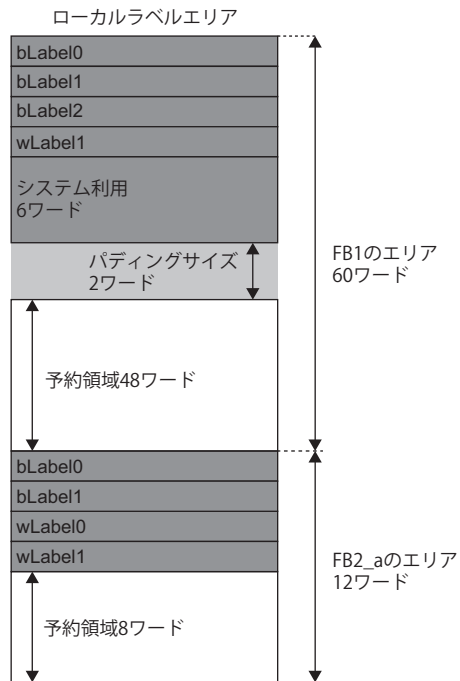
ラベル名	データ型	クラス
bLabel0	BIT	VAR
bLabel1	BIT	VAR
bLabel2	BIT	VAR
wLabel0	WORD	VAR_RETAIN
wLabel1	WORD	VAR
FB2_a	FB2	VAR

FB1予約領域

エリア	サイズ
ローカルラベルエリア	48ワード
ローカルラッチラベルエリア	16ワード
シグナルフローエリア	4ワード

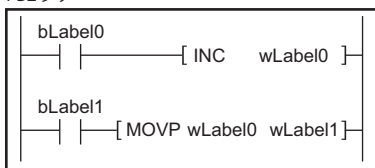
FB1定義から
インスタンスを作成

FB1のインスタンス構造



FB2の定義

FB2ラダー

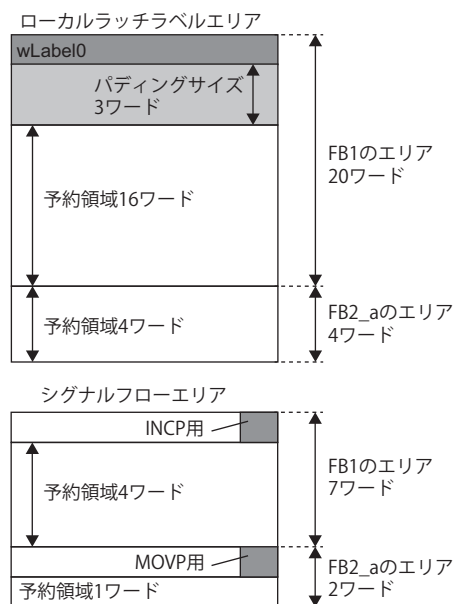


FB2ローカルラベル定義

ラベル名	データ型	クラス
bLabel0	BIT	VAR_INPUT
bLabel1	BIT	VAR_INPUT
wLabel0	WORD	VAR
wLabel1	WORD	VAR_OUTPUT

FB2予約領域

エリア	サイズ
ローカルラベルエリア	8ワード
ローカルラッチラベルエリア	4ワード
シグナルフローエリア	1ワード



上記のとおりFBがネストする場合、ローカルラベルとして宣言されたFBのインスタンスは、宣言元のインスタンスエリアの外に確保されます。また、宣言元のローカルラベルエリアにはシステムで使用する領域(6ワード分)が確保されます。上図のFB1にFB型のローカルラベルを追加または変更する場合は、FB1のローカルラベルエリアにシステム利用分の容量(6ワード)、FB1の宣言元となるプログラムのインスタンスエリアにFB2のインスタンスエリア分の容量(ローカルラベルエリア、ローカルラッチエリア、シグナルフローエリアのそれぞれに対しての容量)が必要となります。

全変換しないで割付を維持したままローカルラベルやFBインスタンスを追加/変更したい場合、予約領域に将来追加/変更する容量分をあらかじめ確保するか、変換+RUN中書込みを実施してください。予約領域の設定方法は、下記を参照してください。

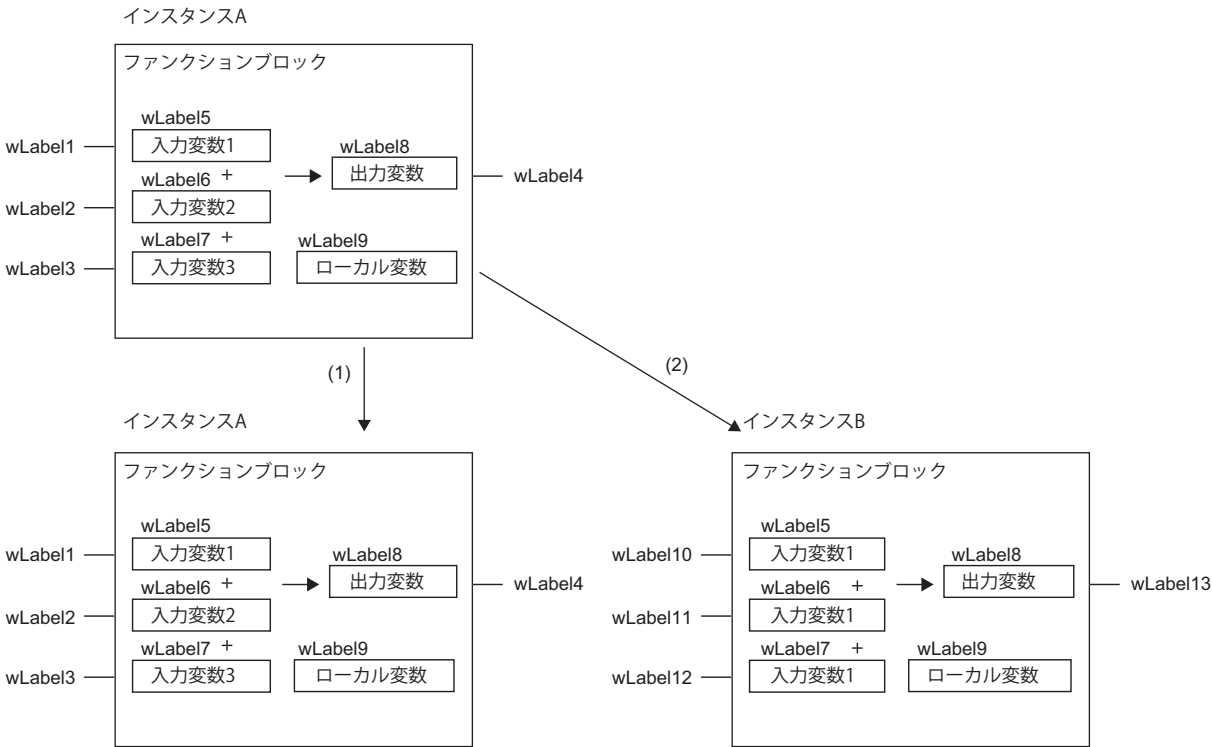
📖 GX Works3 オペレーティングマニュアル

■インスタンスの作成

ファンクションブロックを使用するためには、インスタンスを作成する必要があります。
ファンクションブロックのインスタンスを作成することによって、プログラムブロックや他のファンクションブロックから呼び出して使用できます。
インスタンスはグローバルラベルまたはローカルラベルで宣言します。

ラベルの種類	インスタンスの種類	クラス
グローバルラベル	グローバルFB	VAR_GLOBAL
ローカルラベル ^{*1}	ローカルFB	VAR

^{*1} プログラムブロックまたはファンクションブロックのローカルラベルとして宣言できます。ファンクションでは宣言できません。
1つのプログラム部品の中に、同じファンクションブロックを異なるインスタンスで使用できます。



- (1) 同じインスタンスの場合、同じ内部変数を使用します。
(2) 異なるインスタンスの場合、異なる領域の内部変数を使用します。

■インスタンスの容量

インスタンスの各データエリアの容量について、計算方法を示します。

・ローカルラベルエリアの容量

インスタンスのローカルラベルエリアの容量 = ラッチ属性以外のローカルラベルのデータ容量(総和) + 予約領域容量

項目	内容
ローカルラベル容量(ラッチ属性のローカルラベルを除く)	ローカルラベルとして使用するデータの総和です。 実際に使用するエリアの容量は、ラベルのメモリ割付けによって異なります。ラベルのメモリ割付けの詳細は、下記を参照してください。 📖GX Works3 オペレーティングマニュアル
予約領域容量	デフォルトで48ワードです。4ワード単位で設定できます。

・ローカルラッチラベルエリアの容量

インスタンスのローカルラッチラベルエリアの容量 = ラッチ属性のローカルラベルのデータ容量(総和) + 予約領域容量

項目	内容
ラッチ属性ローカルラベル容量	ラッチ属性のローカルラベルとして使用するデータの総和です。 実際に使用するエリアの容量は、ラベルのメモリ割付けによって異なります。ラベルのメモリ割付けの詳細は、下記を参照してください。 📖GX Works3 オペレーティングマニュアル
予約領域容量	デフォルトで16ワードです。4ワード単位で設定できます。

・シグナルフローエリアの容量

インスタンスのシグナルフローエリアの容量 = FB定義内の命令に使用するシグナルフローエリアの総和 + 予約領域容量

項目	内容
シグナルフローエリア容量	ファンクションブロック定義内の命令が使用するシグナルフローエリアの総和です。
予約領域容量	デフォルトで4ワードです。1ワード単位で設定できます。

Point

予約領域容量には、RUN中書込みで追加/変更が見込まれるローカルラベルやシグナルフローメモリの容量を設定してください。設定方法は、下記を参照してください。

📖GX Works3 オペレーティングマニュアル

ファンクションブロック定義のローカルラベルに一部の構造体ラベルを含む場合は、変換+RUN中書込みで予想領域容量を超える追加/変更はできず全変換が必要になります。詳細は使用するコントローラのユーザーズマニュアルを確認してください。

下記のようなファンクションブロックについては、予約領域をデフォルトより小さい値に設定することでコントローラのメモリを効率良く使用することができます。

- ・ローカルラベルの追加/変更やプログラムの変更を行わないデバッグ済みのサブルーチン型ファンクションブロック
- ・大量にインスタンスを宣言するサブルーチン型ファンクションブロック

初期値の設定

■ファンクションブロックのローカルラベルの初期値

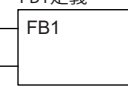
ファンクションブロックのローカルラベルは、ファンクションブロックの定義および、インスタンスごとに初期値が設定できます。

初期値が設定可能なローカルラベルのクラスは、VAR、VAR_RETAIN、VAR_INPUT、VAR_OUTPUT、VAR_OUTPUT_RETAIN、VAR_PUBLIC、VAR_PUBLIC_RETAINです。

■インスタンスの初期値

インスタンスの初期値の種類は下記になります。

種類	内容																												
デフォルト初期値	データ型ごとに決められた初期値のことです。ファンクションブロックのローカルラベルに初期値を設定しなかった場合、デフォルト初期値が適用されます。 FB1定義 FB1 FB1のローカルラベル定義 <table><thead><tr><th>ラベル名</th><th>初期値</th></tr></thead><tbody><tr><td>AAA</td><td></td></tr><tr><td>BBB</td><td></td></tr><tr><td>CCC</td><td></td></tr></tbody></table> (1) グローバルラベル定義 <table><thead><tr><th>ラベル名</th><th>初期値</th></tr></thead><tbody><tr><td>FB1_a</td><td>-</td></tr><tr><td>AAA</td><td>0</td></tr><tr><td>BBB</td><td>0</td></tr><tr><td>CCC</td><td>0</td></tr></tbody></table> プログラムブロック1のローカルラベル定義 <table><thead><tr><th>ラベル名</th><th>初期値</th></tr></thead><tbody><tr><td>FB1_b</td><td>-</td></tr><tr><td>AAA</td><td>0</td></tr><tr><td>BBB</td><td>0</td></tr><tr><td>CCC</td><td>0</td></tr></tbody></table> (2) (1) FB1定義でローカルラベルに初期値を設定していません。 (2) デフォルト初期値が適用されます。	ラベル名	初期値	AAA		BBB		CCC		ラベル名	初期値	FB1_a	-	AAA	0	BBB	0	CCC	0	ラベル名	初期値	FB1_b	-	AAA	0	BBB	0	CCC	0
ラベル名	初期値																												
AAA																													
BBB																													
CCC																													
ラベル名	初期値																												
FB1_a	-																												
AAA	0																												
BBB	0																												
CCC	0																												
ラベル名	初期値																												
FB1_b	-																												
AAA	0																												
BBB	0																												
CCC	0																												
FB定義初期値	ファンクションブロックのローカルラベルを定義する際に設定する初期値のことです。FB定義初期値が設定されている場合は、すべてのインスタンスに対して、同じ定義初期値が適用されます。 FB1定義 FB1 FB1のローカルラベル定義 <table><thead><tr><th>ラベル名</th><th>初期値</th></tr></thead><tbody><tr><td>AAA</td><td>1111</td></tr><tr><td>BBB</td><td>2222</td></tr><tr><td>CCC</td><td>3333</td></tr></tbody></table> (1) グローバルラベル定義 <table><thead><tr><th>ラベル名</th><th>初期値</th></tr></thead><tbody><tr><td>FB1_a</td><td>-</td></tr><tr><td>AAA</td><td>1111</td></tr><tr><td>BBB</td><td>2222</td></tr><tr><td>CCC</td><td>3333</td></tr></tbody></table> プログラムブロック1のローカルラベル定義 <table><thead><tr><th>ラベル名</th><th>初期値</th></tr></thead><tbody><tr><td>FB1_b</td><td>-</td></tr><tr><td>AAA</td><td>1111</td></tr><tr><td>BBB</td><td>2222</td></tr><tr><td>CCC</td><td>3333</td></tr></tbody></table> (2) (1) FB1定義でローカルラベルに初期値を設定しています。 (2) FB1の全インスタンスは同じ定義初期値で初期化します。	ラベル名	初期値	AAA	1111	BBB	2222	CCC	3333	ラベル名	初期値	FB1_a	-	AAA	1111	BBB	2222	CCC	3333	ラベル名	初期値	FB1_b	-	AAA	1111	BBB	2222	CCC	3333
ラベル名	初期値																												
AAA	1111																												
BBB	2222																												
CCC	3333																												
ラベル名	初期値																												
FB1_a	-																												
AAA	1111																												
BBB	2222																												
CCC	3333																												
ラベル名	初期値																												
FB1_b	-																												
AAA	1111																												
BBB	2222																												
CCC	3333																												

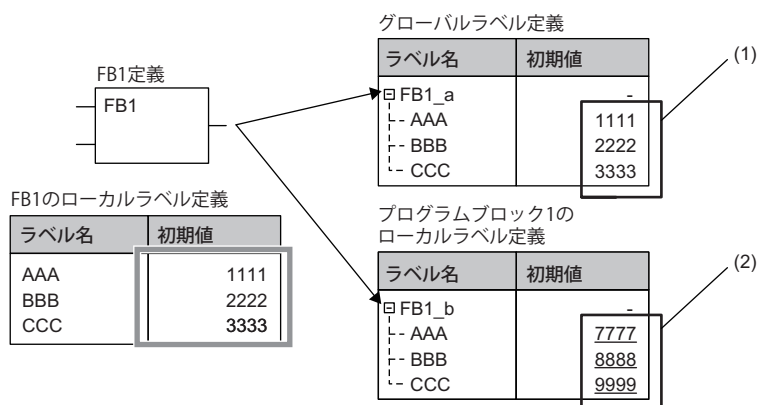
種類	内容																												
インスタンス初期値	グローバルラベルおよびプログラムブロックのローカルラベル定義に含まれる、インスタンスに設定する初期値のことです。 FB1定義  FB1のローカルラベル定義 <table border="1"> <thead> <tr> <th>ラベル名</th><th>初期値</th></tr> </thead> <tbody> <tr> <td>AAA</td><td>1111</td></tr> <tr> <td>BBB</td><td>2222</td></tr> <tr> <td>CCC</td><td>3333</td></tr> </tbody> </table> グローバルラベル定義 <table border="1"> <thead> <tr> <th>ラベル名</th><th>初期値</th></tr> </thead> <tbody> <tr> <td>FB1_a</td><td>-</td></tr> <tr> <td>-- AAA</td><td>3333</td></tr> <tr> <td>-- BBB</td><td>4444</td></tr> <tr> <td>-- CCC</td><td>5555</td></tr> </tbody> </table> プログラムブロック1のローカルラベル定義 <table border="1"> <thead> <tr> <th>ラベル名</th><th>初期値</th></tr> </thead> <tbody> <tr> <td>FB1_b</td><td>-</td></tr> <tr> <td>-- AAA</td><td>7777</td></tr> <tr> <td>-- BBB</td><td>8888</td></tr> <tr> <td>-- CCC</td><td>9999</td></tr> </tbody> </table> (1) FB1定義のインスタンスごとに初期値を設定可能です。	ラベル名	初期値	AAA	1111	BBB	2222	CCC	3333	ラベル名	初期値	FB1_a	-	-- AAA	3333	-- BBB	4444	-- CCC	5555	ラベル名	初期値	FB1_b	-	-- AAA	7777	-- BBB	8888	-- CCC	9999
ラベル名	初期値																												
AAA	1111																												
BBB	2222																												
CCC	3333																												
ラベル名	初期値																												
FB1_a	-																												
-- AAA	3333																												
-- BBB	4444																												
-- CCC	5555																												
ラベル名	初期値																												
FB1_b	-																												
-- AAA	7777																												
-- BBB	8888																												
-- CCC	9999																												

ファンクションブロックの初期値は、FB定義初期値とインスタンス初期値の両方を設定することが可能です。両方の初期値を設定した場合に適用される初期値の優先順位は下記になります。

優先順位	種類
高	インスタンス初期値
↑	
↓	FB定義初期値
低	デフォルト初期値

Point

FB定義初期値が設定されているファンクションブロックのインスタンスを2つ作成し、片方のみインスタンス初期値を設定した場合、インスタンス初期値を設定しなかったインスタンスはFB定義初期値が適用され、設定したインスタンスは、インスタンス初期値が適用されます。



- (1) インスタンス初期値を設定しなかった場合は、定義初期値で初期化されます。
 (2) インスタンス初期値を設定した場合は、インスタンス初期値で初期化されます。

■使用例

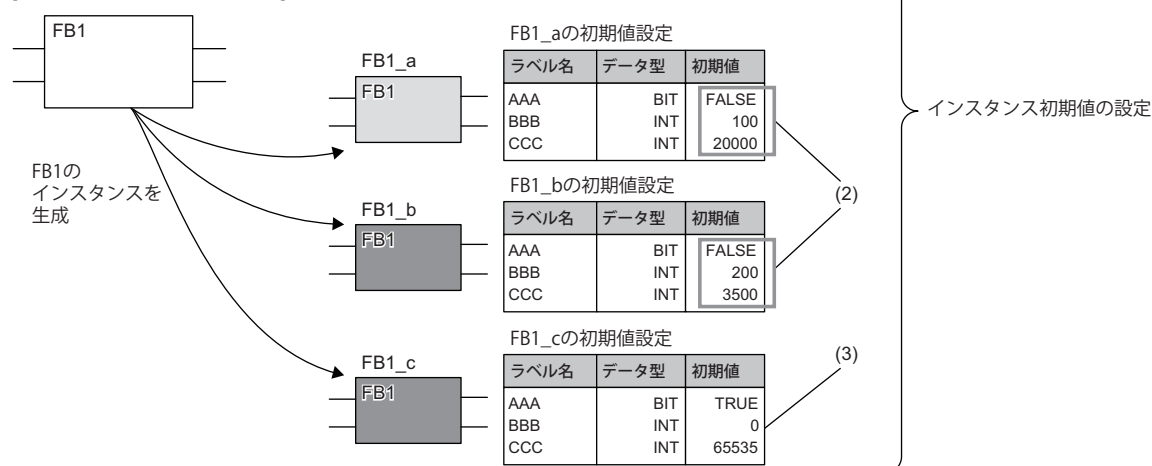
ファンクションブロックの初期値の使用例を示します。

[FB定義の作成]



FB定義初期値の設定

[FB定義からインスタンスを生成]



- (1) 全インスタンスで共通の初期値を設定する。
- (2) インスタンスごとに個別の初期値を設定可能です。
- (3) 個別の初期値を設定しない場合は、共通の初期値が適用されます。

EN/ENOについて

ファンクションブロックは、ファンクションと同様、EN(イネーブル入力)、ENO(イネーブル出力)を付けることにより、実行処理の制御ができます。(62ページ EN/ENOについて)

EN/ENOを付けたファンクションブロックのインスタンスが呼び出されるときに、ENへの実引数の割り付けは必須となります。

Point

モーション制御ファンクションブロックの実行条件については、下記を参照してください。

1367ページ 実行指令(Execute)型/有効(Enable)型

プログラムの作成

ファンクションブロックのプログラムを作成する場合、下記の操作から実施します。

① [ナビゲーションウィンドウ]⇒[FB/FUN]⇒右クリック⇒[データ新規作成]

“基本設定”の“データ型”において“ファンクションブロック”を選択

作成したプログラムはFBファイルに格納します。

② [CPUパラメータ]⇒[プログラム設定]⇒[FB/FUNファイル設定]

1つのFBファイルには、作成したプログラムを最大64個まで格納できます。

プログラムの作成に関連する内容については、下記を参照してください。

項目	参照先
ファンクションブロックの作成方法	① GX Works3 オペレーティングマニュアル
コントローラに書き込めるFB/FUNファイルの数	② 使用するコントローラのユーザーズマニュアル

■プログラムの種類

ファンクションブロックは下記の種類があり、ファンクションブロックのプログラム本体の格納の仕方が異なります。

- ・ サブルーチン型ファンクションブロック

詳細については、下記を参照してください。

③ 66ページ 動作概要

ユニットFB、汎用ファンクション、汎用ファンクションブロック、モーション制御ファンクションブロックは上記の選択はできません。

■固有プロパティ設定

ファンクションブロックのプログラムを作成する場合は、下記を設定することができます。(④ GX Works3 オペレーティングマニュアル)

項目	内容
ENの制御にMC/MCRを使用する	“はい”を選択した場合は、ENの制御にMC/MCR命令を使用します。“いいえ”を選択した場合は、ENの制御にCJ命令を使用します。FB内で立上り/立下り命令を使用している場合は、“はい”を選択してください。また、選択により、FB内で使用しているタイマ/カウンタやOUT命令の動作が異なります。詳細については、下記を参照してください。 ⑤ 84ページ ENの制御にMC/MCR命令を使用する場合の動作
EN/ENOを使用する	“はい”を選択した場合は、EN/ENOを持つファンクションブロックとなり、EN/ENOラベルはローカルラベルに登録しなくてもプログラムで使用可能となります。“いいえ”を選択した場合は、EN/ENOを持たないファンクションブロックとなります。 EN/ENOの詳細については、下記を参照してください。 ⑥ 74ページ EN/ENOについて

■使用可能なデバイス/ラベル

ファンクションブロックのプログラムで使用可能なデバイスおよびラベルの一覧は下記になります。

○:使用可, △:命令でのみ使用可(プログラムのステップを示すラベルとしては使用不可), ×:使用不可

デバイス/ラベルの種類		使用可否
ラベル(ポインタ型以外)	グローバルラベル	○
	ローカルラベル	○
ラベル(ポインタ型)	ポインタ型グローバルラベル	△
	ポインタ型ローカルラベル	○
デバイス	グローバルデバイス	○
	ローカルデバイス	×
ポインタ	グローバルポインタ	△
	ローカルポインタ	×

2.4 注意事項

ファンクションを使用する場合

ポインタについて

ローカルポインタ、ポインタ型グローバルラベルは、使用できません。

ファンクションブロックを使用する場合

ポインタについて

ローカルポインタは、使用できません。

インデックスレジスタを使用する場合

ファンクションブロックのプログラムで、インデックスレジスタを使用する場合は、インデックスレジスタの値保護のため、退避回路と復帰回路が必要です。

インデックスレジスタの退避時にインデックスレジスタの値を0にすることで、インデックス修飾の整合性チェック(デバイス番号がデバイス範囲を超えていないか)によるエラーを防止できます。

例

プログラム実行前にインデックスレジスタZ1, Z2を退避し、プログラム実行後に退避したインデックスレジスタを復帰する場合



インテリジェント機能ユニットの先頭I/O No.の指定

インテリジェント機能ユニットのバッファメモリや入出力信号にアクセスする場合は、インデックスレジスタを使用して先頭入出力番号を指定してください。

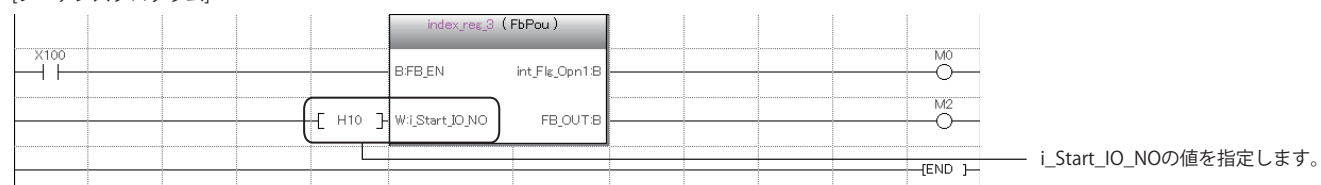
先頭入出力番号を入力変数として受け取ることで、装着位置が異なる複数のインテリジェント機能ユニットで、先頭入出力番号を変更することなく共通のファンクションブロックを使用できます。

例

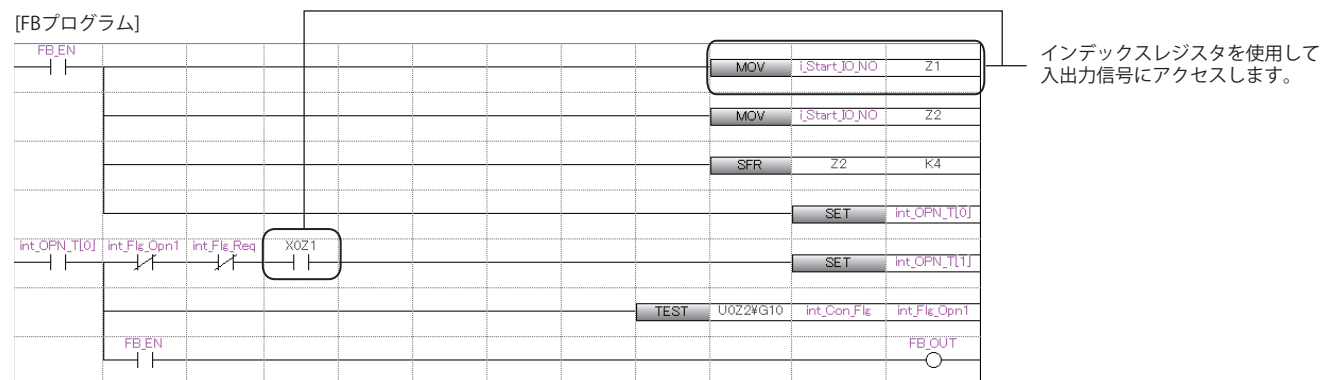
インテリジェント機能ユニットの入出力信号にアクセスする場合

インデックスレジスタを使用することで、対象のインテリジェント機能ユニットの入出力信号にアクセスできます。

[シーケンスプログラム]



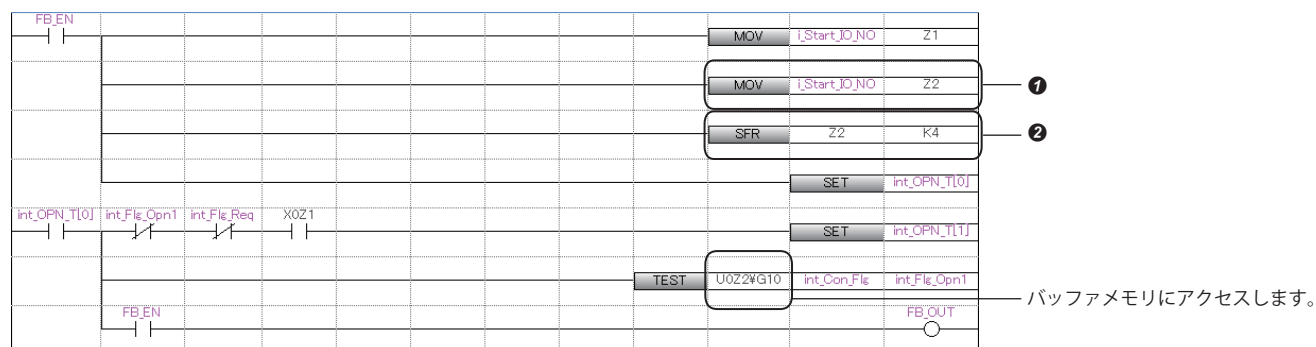
[FBプログラム]



例

インテリジェント機能ユニットのバッファメモリにアクセスする場合

- ① インデックスレジスタに対象インテリジェント機能ユニットの先頭入出力番号を入力します。
- ② SFR命令を使用し、値を右に4ビットシフト、または値を16で割った商を使用します。



マクロ型ファンクションブロックの複製(コピー)について

マクロ型ファンクションブロックを複数使用する場合は、部品選択ウィンドウからドラッグ&ドロップしてください。
FBコンパイル方式がマクロ型ファンクションブロックをプログラムエディタ上で複製した場合、複製したファンクションブロックはサブルーチン型ファンクションブロックに変更されます。
ユニットFBやFBライブラリを複製する場合は使用するファンクションブロックのリファレンスマニュアルを確認し、FBコンパイル方式にサブルーチン型と記載されているか確認してから使用してください。

Point

マクロ型ファンクションブロックとは、ユニットFB、FBライブラリで使用しているファンクションブロックの種類です。ユーザでは定義できません。

動作は、プログラミング時に呼出し元プログラムに対して、呼出し対象のプログラム本体を展開します。実行時は通常のプログラムと同様に展開されたプログラムを実行します。

マクロ型ファンクションブロックの呼出しについて

ポインタ指定先としてポインタ型ラベルを使用するマクロ型ファンクションブロックの各インスタンスは、プロジェクト内で一回のみ呼び出すようにしてください。ポインタ指定先としてポインタ型ラベルを使用するマクロ型ファンクションブロックの同一インスタンスが複数回呼び出されていると、変換エラーとなります。

ユニットFBの制約事項

ユニットFBを使用する場合、下記の制約事項があります。

- MC命令からMCR命令間でユニットFBを呼び出す場合、MC命令の接点はOFFしないでください。
- CJ命令、SCJ命令、JMP命令でユニットFBが呼び出されなくなるようなジャンプはしないでください。
- サブルーチンプログラム内でユニットFBを呼び出す場合、毎スキャン1回サブルーチンプログラムを実行してください。またFCALL(P)命令、EFCALL(P)命令、XCALL命令でサブルーチンプログラムの非実行処理を行わないでください。
- 割込みプログラム、初期実行タイププログラム、定周期実行タイププログラム、イベント実行タイププログラム内ではユニットFBを呼び出さないでください。
- FOR~NEXT命令間やインラインST、またはST言語の制御構文内(IF文、FOR文、CASE文など)ではユニットFBを呼び出さないでください。
- 一部のユニットFBの入力引数、出力引数に修飾したデバイスを指定するとエラーになる場合があります。修飾されたデバイスをユニットFBの入力引数、出力引数に指定したい場合は、ユニットFBを呼び出した後に値を転送する処理を記述してください。

ユニットFBの動作パラメータの変更について

ユニットFBの入カラベルや出カラベル以外の動作パラメータ(外部変数)の変更は、ラベル設定で変更できます。

- ユニットFBのインスタンスをローカルラベルにした場合は、“ローカルラベル設定”で変更できます。

🔍 [ナビゲーションウィンドウ]⇒[プログラム]⇒実行タイプ⇒プログラムファイル⇒プログラムブロック⇒[ローカルラベル]

- ユニットFBのインスタンスをグローバルラベルにした場合は、“グローバルラベル設定”で変更できます。

🔍 [ナビゲーションウィンドウ]⇒[ラベル]⇒[グローバルラベル]

例

ローカルラベル設定の場合

＜フィルタ＞ 詳細表示(Y) 表示設定(S) チェック(K)

	ラベル名	データ型	コメント
1	M+RCPU_Msynchronization_Delay1 OUT_00A	M+RCPU_Msynchronization_Delay1 OUT_00A	
2	M+RJ71GF11_DeviceRead_00B_1	M+RJ71GF11_DeviceRead_00B	
3			

拡張表示: 自動

M+RJ71GF11_DeviceRead_00B_1 (M+RJ71GF11_DeviceRead_00B)

クラス	ラベル名	データ型	初期値	コメント
VAR_INPUT	i_BEN	ビット		
VAR_INPUT	i_ExtModule	M+RJ71GF11		
VAR_INPUT	i_u2TargetAddress	ワ-16符号なし/ビット列(16ビット)(0.1)		
VAR_INPUT	i_uDataLength	ワ-16符号なし/ビット列(16ビット)		
VAR_INPUT	i_u2TargetDevice	文字列(0.3)		
VAR_INPUT	i_uChannel	ワ-16符号なし/ビット列(16ビット)		
VAR_OUTPUT	o_BEND	ビット		
VAR_OUTPUT	o_BOK	ビット		
VAR_OUTPUT	o_BErr	ビット		
VAR_OUTPUT	o_uErrId	ワ-16符号なし/ビット列(16ビット)		
VAR_OUTPUT	o_uReadData	ワ-16符号なし/ビット列(16ビット)		
VAR_PUBLIC	pbi_uCPU_Type	ワ-16符号なし/ビット列(16ビット)		
VAR_PUBLIC	pbi_uResendCountMax	ワ-16符号なし/ビット列(16ビット)		
VAR_PUBLIC	pbi_uTimeUnit	ワ-16符号なし/ビット列(16ビット)		
VAR_PUBLIC	pbi_uMonitorTime	ワ-16符号なし/ビット列(16ビット)		
VAR_PUBLIC	pbi_uStationSpecifcId	ビット		
VAR_PUBLIC	pbo_uResendCount	ワ-16符号なし/ビット列(16ビット)		
VAR_PUBLIC	pbo_u4ErrTime	ワ-16符号なし/ビット列(16ビット)(0.3)		
VAR_PUBLIC	pbo_uErrNetworkNo	ワ-16符号なし/ビット列(16ビット)		
VAR_PUBLIC	pbo_uErrStationNo	ワ-16符号なし/ビット列(16ビット)(0.17)		
VAR	u18ControlData	ワ-16符号なし/ビット列(16ビット)(0.17)		
VAR	b5_start	ビット		
VAR	b2_Comp	ビット(0.1)		

“初期値”欄に動作パラメータを設定します。

ENの制御にMC/MCR命令を使用する場合の動作

ファンクションブロックの固有プロパティ設定で“ENの制御にMC/MCRを使用する”を有効にした場合のファンクションブロック内で使用する命令、デバイス/ラベルの動作を示します。

ファンクションブロック内で使用する命令、デバイス/ラベル	ファンクションブロック内で使用する命令、デバイス/ラベルの状態	
	“ENの制御にMC/MCRを使用する”で“はい”選択時	“ENの制御にMC/MCRを使用する”で“いいえ”選択時
立上り/立下り命令(PLS命令、パルス化命令(OP)*1)	次にENがONしたとき、条件接点が成立していれば命令を実行します。	次にENがONしたとき、条件接点が成立していても命令を実行しない場合があります。
タイマ(低速/高速)、ロングタイマ	カウント値は0となり、コイル、接点ともOFFします。	現在の状態を保持します。
積算タイマ(低速/高速)、ロング積算タイマ、カウンタ、ロングカウンタ	コイルはOFFになるが、カウント値、接点とも現在の状態を保持します。	現在の状態を保持します。
OUT命令のデバイス部に指定したデバイス	強制的にOFFになります。	現在の状態を保持します。

*1 コイル側に指定した命令が対象です。

制約事項

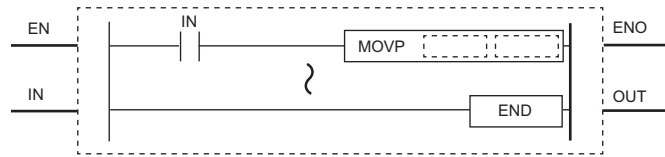
“ENの制御にMC/MCRを使用する”を“はい”にした場合、そのファンクションブロックの実行中はMC/MCR命令を使用しないでください。MC/MCR命令を使用した場合、ENの制御が正しく動作しない場合があります。

立上り/立下り命令の動作

立上り/立下り命令の動作を示します。

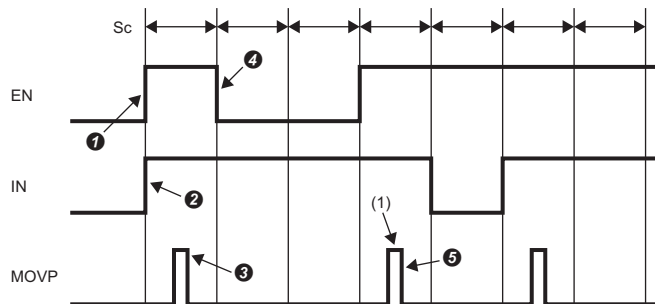
例

立上り命令を使用したサブルーチン型FB



■“ENの制御にMC/MCRを使用する”で“はい”選択時

ENがON時、条件接点が成立していれば命令を実行します。(図中(1))

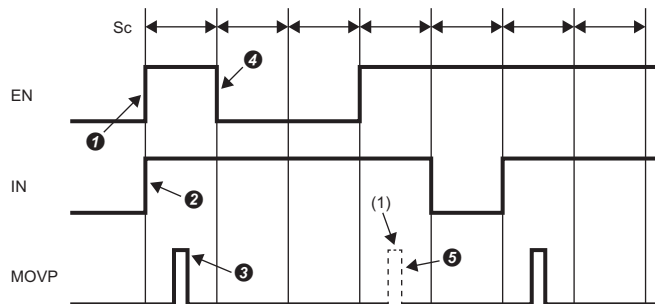


Sc: スキャン

- ① ENをONします。(お客様操作)
- ② INをONします。(お客様操作)
- ③ MOV_P命令を実行します。(コントローラ動作)
- ④ ENをOFFします。(お客様操作)
- ⑤ MOV_P命令を実行します。(コントローラ動作)

■“ENの制御にMC/MCRを使用する”で“いいえ”選択時

ENがOFF時、条件接点状態により命令の動作が異なります。(図中(1))



Sc: スキャン

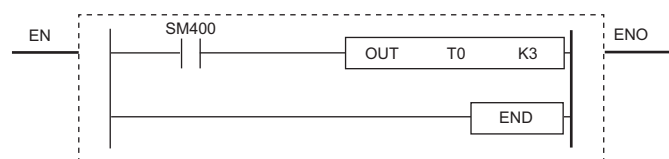
- ① ENをONします。(お客様操作)
- ② INをONします。(お客様操作)
- ③ MOV_P命令を実行します。(コントローラ動作)
- ④ ENをOFFします。(お客様操作)
- ⑤ ④でENがOFFされる直前の条件接点がOFFの場合は、MOV_P命令を実行します。(コントローラ動作)(④でENがOFFされる直前の条件接点がONの場合は、MOV_P命令を実行しません。)

タイマ(低速/高速), ロングタイマの動作

タイマ(低速/高速), ロングタイマの動作を示します。

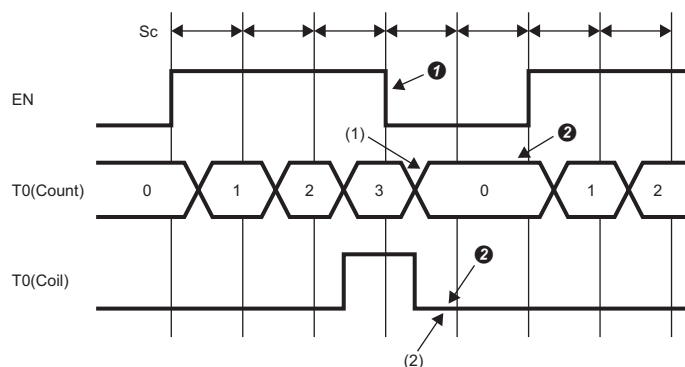
例

低速タイマを使用したサブルーチン型FB



■“ENの制御にMC/MCRを使用する”で“はい”選択時

カウント値は0となります。(図中(1))また, コイルはOFFになります。(図中(2))



Sc: スキャン

T0(Count): T0(カウント値)

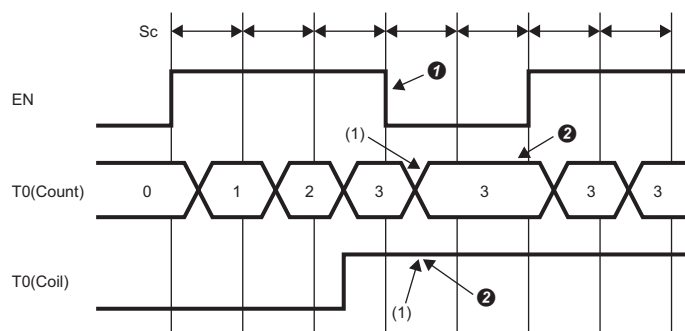
T0(Coil): T0(コイル)

① ENをOFFします。(お客様操作)

② コイルがOFFになり, タイマ値, カウント値をクリアします。(コントローラ動作)

■“ENの制御にMC/MCRを使用する”で“いいえ”選択時

カウント値, コイルとも現在の状態を保持します。(図中(1))



Sc: スキャン

T0(Count): T0(カウント値)

T0(Coil): T0(コイル)

① ENをOFFします。(お客様操作)

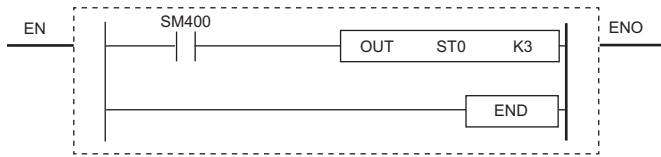
② 値を保持します。(コントローラ動作)

積算タイマ(低速/高速), ロング積算タイマ, カウンタ, ロングカウンタの動作

積算タイマ(低速/高速), ロング積算タイマ, カウンタ, ロングカウンタの動作を示します。

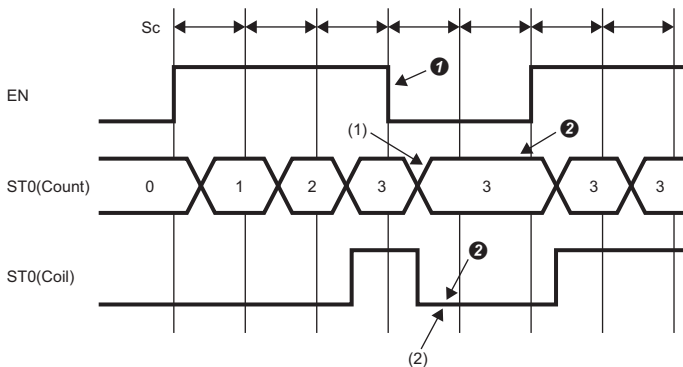
例

低速積算タイマを使用したサブルーチン型FB



■“ENの制御にMC/MCRを使用する”で“はい”選択時

カウント値は現在の状態を保持します。(図中(1))また、コイルはOFFになります。(図中(2))



Sc: スキャン

ST0(Count): T0(カウント値)

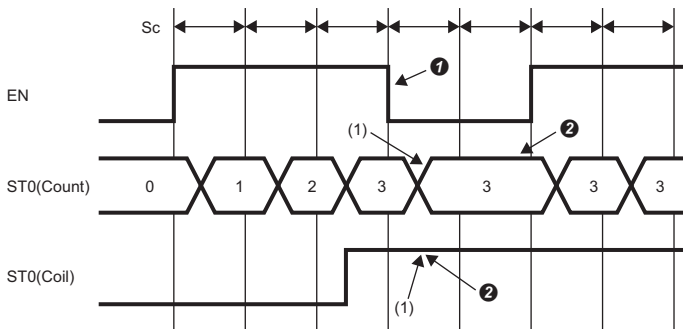
ST0(Coil): T0(コイル)

① ENをOFFします。(お客様操作)

② コイルはOFFになるが、カウント値、接点とも現在の状態を保持します。(コントローラ動作)

■“ENの制御にMC/MCRを使用する”で“いいえ”選択時

カウント値、コイルとも現在の状態を保持します。(図中(1))



Sc: スキャン

ST0(Count): T0(カウント値)

ST0(Coil): T0(コイル)

① ENをOFFします。(お客様操作)

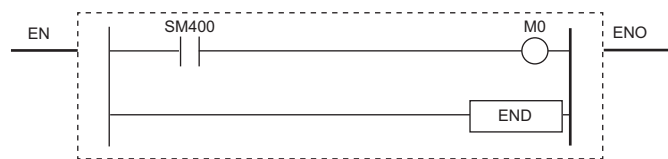
② 値を保持します。(コントローラ動作)

OUT命令のデバイス部に指定したデバイスの動作

OUT命令のデバイス部に指定したデバイスの動作を示します。

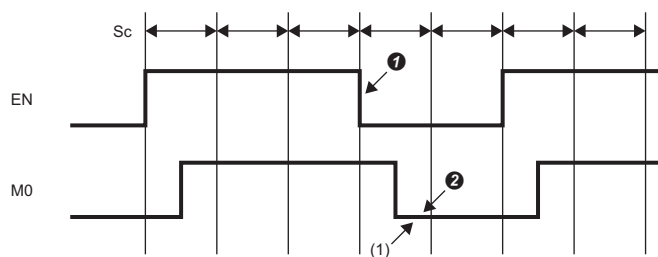
例

OUT命令のデバイス部にM0を使用したサブルーチン型FB



■“ENの制御にMC/MCRを使用する”で“はい”選択時

M0は強制的にOFFになります。(図中(1))



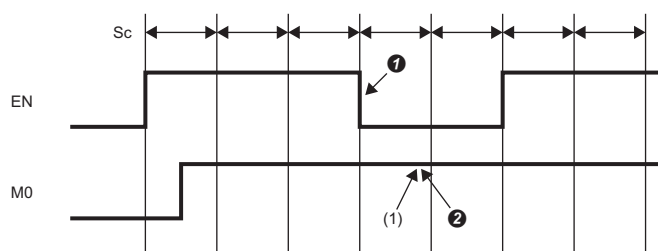
Sc: スキャン

① ENをOFFします。(お客様操作)

② コイルをOFFします。(コントローラ動作)

■“ENの制御にMC/MCRを使用する”で“いいえ”選択時

M0は現在の状態を保持します。(図中(1))



Sc: スキャン

① ENをOFFします。(お客様操作)

② コイルを保持します。(コントローラ動作)

接点やコイルからなる回路で、直列や並列の組合せでAND/ORからなる論理演算を表し、シーケンス制御を記述する言語です。

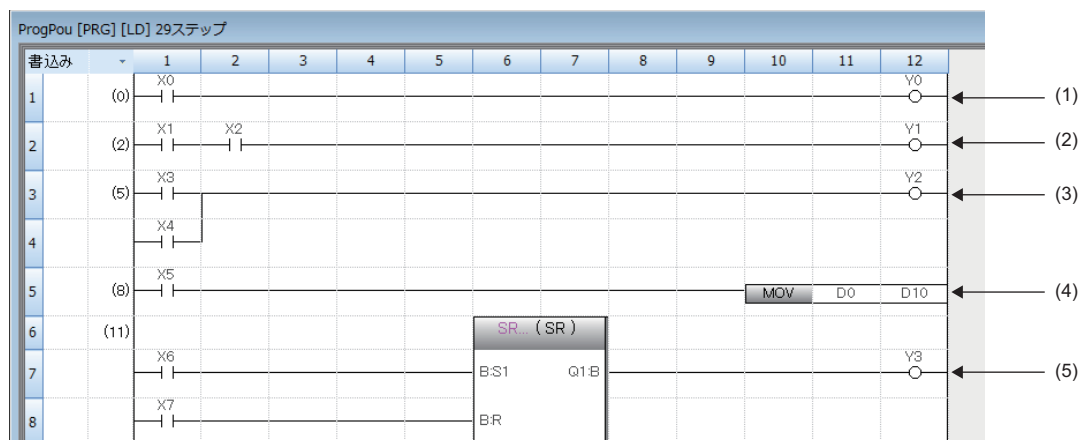
Point

本章では、ラダー言語の動作や仕様について説明しています。ラダープログラム作成時の操作方法については、下記を参照してください。

 GX Works3 オペレーティングマニュアル

3.1 構成

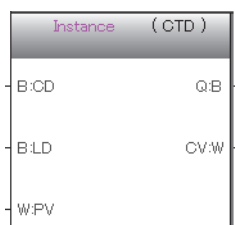
ラダー言語では、下記のような回路が作成できます。



- (1) 接点とコイルによる回路
- (2) 直列で構成された回路
- (3) 並列で構成された回路
- (4) 命令を使用した回路
- (5) 汎用ファンクション/ファンクションブロックを使用した回路

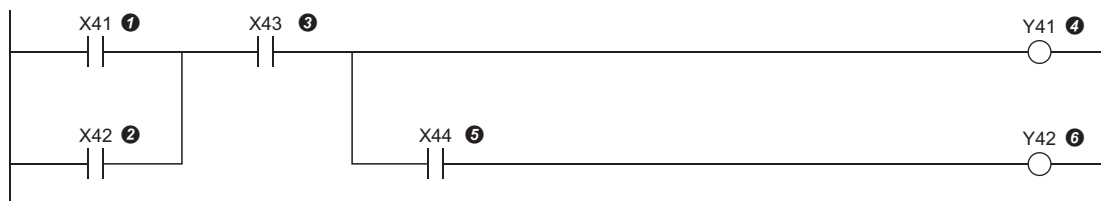
回路記号

ラダー言語でのプログラミングに使用できる回路記号を示します。

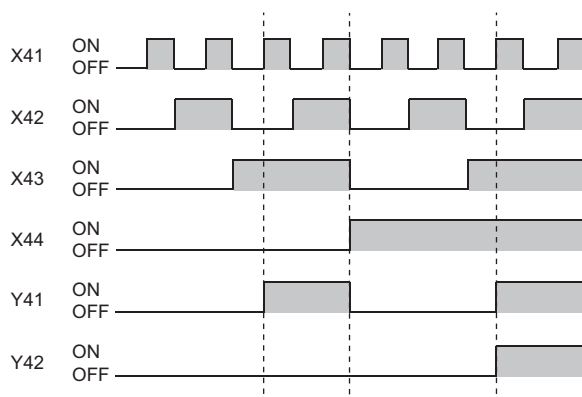
項目		内容
a接点		指定デバイスまたはラベルがONのとき導通します。
b接点		指定デバイスまたはラベルがOFFのとき導通します。
立上りパルス		指定デバイスまたはラベルの立上り時(OFF→ON)に導通します。
立下りパルス		指定デバイスまたはラベルの立下り時(ON→OFF)に導通します。
立上りパルス否定		指定デバイスまたはラベルのOFF時、ON時、および立下り時(ON→OFF)に導通します。
立下りパルス否定		指定デバイスまたはラベルのOFF時、ON時、および立上り時(OFF→ON)に導通します。
演算結果立上りパルス化		演算結果の立上り時(OFF→ON)に導通します。演算結果が立上り時以外の場合は、非導通になります。
演算結果立下りパルス化		演算結果の立下り時(ON→OFF)に導通します。演算結果が立下り時以外の場合は、非導通になります。
演算結果反転		直前までの演算結果を反転します。
コイル		演算結果を指定デバイスまたはラベルへ出力します。
命令		[]内で指定した命令を実行します。
折り返し		1つの回路の行に作成可能な接点数を超えた場合、折り返し元の記号と折り返し先の記号が作成され、回路の折り返しを行います。
ファンクション		ファンクションを実行します。 - ファンクションの作成方法(GX Works3 オペレーティングマニュアル) - 汎用ファンクション(176ページ 汎用ファンクション)
ファンクションブロック		ファンクションブロックを実行します。 - ファンクションブロックの作成方法(GX Works3 オペレーティングマニュアル) - 汎用ファンクションブロック(187ページ 汎用ファンクションブロック) - モーション制御ファンクションブロック(189ページ モーション制御ファンクションブロック一覧) - ユニットFB(使用するユニットのFBリファレンス)

プログラム実行順序

下記の番号順に実行されます。



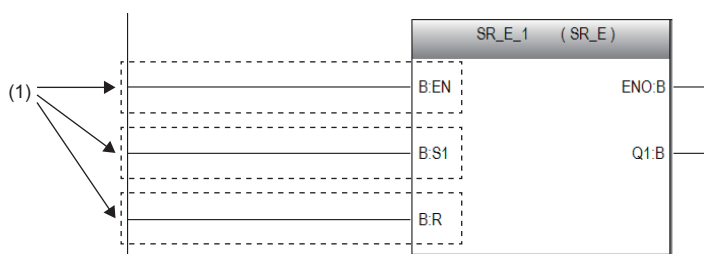
上記のプログラムを実行した場合、X41～X44のON/OFFによるY41、Y42のONのタイミングは下記になります。



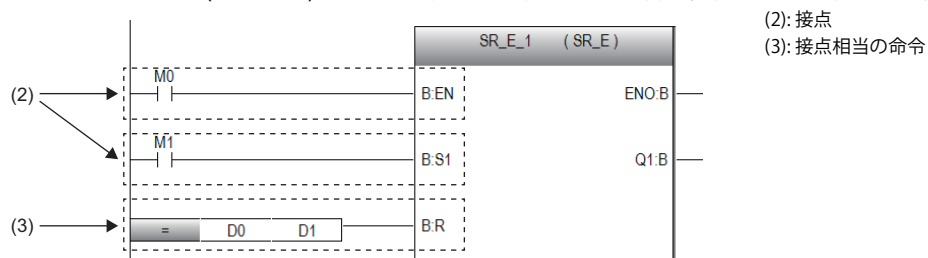
ラダー言語でファンクションブロックを使用する場合の注意事項

左母線から直接FBインスタンスに接続している場合の注意事項

FBインスタンスの入力回路部で、ENおよび入力変数(ビット型)が左母線と直接接続している場合、ON/OFF状態は変化しません。



ENおよび入力変数(ビット型)のON/OFF状態を変化させる場合は、接点または接点相当の命令を使用してください。



3.2 インラインST

インラインSTとは、ラダーエディタ内でコイル相当の命令のセルにSTプログラムを表示するインラインSTボックスを作成し、編集/モニタする機能です。

これにより、ラダープログラム内で数値演算や文字列処理が簡単に作成できます。

- ・インラインSTを使用しない場合のプログラム

[illegible]

- ・インラインSTを使用した場合のプログラム

[illegible]

仕様

インラインSTに記述するSTプログラムの仕様については、ST言語の仕様を参照してください。

👉 91ページ ST言語

注意事項

- ラダープログラムの1行に作成可能なインラインSTは1つです。
- ラダープログラムの1行にファンクション/ファンクションブロックと、インラインSTボックスの両方を使用することはできません。
- 接点相当の命令の位置でインラインSTボックスを作成すると、コイル相当の命令の位置にインラインSTボックスが作成されます。
- インラインST内で入力できる文字数は、最大20000文字です。(改行は2文字分としてカウントします。)
- インラインST内で「RETURN文」を使用すると、プログラムブロックの処理が終了するのではなく、インラインSTボックス内の処理が終了になります。
- ファンクションのプログラムにてインラインSTを使用する場合、インラインST内からファンクションブロックの呼び出しはできません。
- インラインSTでは、変換時にCJ命令を使用してプログラムの動作を制御します。インラインSTの接点がOFFの場合、インラインST内の処理はCJ命令により実行されません。そのため、たとえばインラインST内の代入文によりONしたデバイスは、インラインSTが実行されない場合でも、出力状態を保持します。CJ命令の詳細については、下記を参照してください。

📖 530ページ CJ, SCJ, JMP

- インラインST内では、OUT, OUTHの他、シグナルフローメモリの前回実行状態を参照する命令は使用できません。

📖 使用するコントローラのユーザーズマニュアル

3.3 ステートメント/ノート

ラダー回路では、ステートメントとノートが表示できます。

ステートメント

ステートメントを使用することにより、回路ブロックに対してコメントを付加ができます。付加することにより、処理などの流れを分かりやすくします。

ステートメントには、行間ステートメント/Pステートメント/Iステートメントがあります。

行間ステートメントは、ナビゲーションウィンドウのツリー上に表示できます。

■行間ステートメント

回路ブロック全体に対してコメントを付加します。

■Pステートメント

ポインタデバイスに対してコメントを付加します。

■Iステートメント

割込みポインタデバイスに対してコメントを付加します。

ノート

ノートを使用することにより、プログラム中のコイルや命令に対して、コメントを付加できます。

付加することにより、コイルや命令の内容などを分かりやすくします。

ステートメント/ノートの種別

ステートメントとノートの種別として、「一体」と「周辺」があります。

種別	種類	内容
一体	- 行間ステートメント - Pステートメント - Iステートメント - ノート	説明文の文字列は、変換時にプログラム内に埋め込まれず、プログラムの付属情報として保存されます。 1行分で1ステップ分を使用します。
周辺	- 行間ステートメント - Pステートメント - Iステートメント - ノート	説明文の文字列はプログラム内に埋め込まれず、プログラムの付属情報として保存されます。 1行分で1ステップ分を使用します。 入力したテキストの前に、自動的に*印が付きます。

ST言語は、ロジックの記述方式について規定した国際規格IEC61131-3で定義されている言語です。ST言語はC言語などに似た文法構造を持つ、テキスト形式のプログラム言語です。ラダー言語では表現しにくい複雑な処理をプログラミングする場合に適しています。

Point

本章では、ST言語の動作や仕様について説明しています。STプログラム作成時の操作方法については、下記を参照してください。

 GX Works3 オペレーティングマニュアル

ST言語では制御構文、演算式、ファンクションブロック(FB)、ファンクション(FUN)をサポートし、下記のような記述ができます。

例

条件文による選択分岐、繰り返し文などの制御構文

(*ラインA~Cで制御を行う*)

CASE ライン OF

1:

開始スイッチ := TRUE; (*コンベア稼動*)

2:

開始スイッチ := FALSE; (*コンベア停止*)

3:

開始スイッチ := TRUE; (*コンベア停止 警告*)

ELSE

警告ランプ := TRUE;

END_CASE;

IF 開始スイッチ = TRUE THEN (*コンベア稼動 処理100回*)

FOR 処理回数 := 0 TO 100 BY 1 DO

処理数 := 処理数 + 1;

END_FOR;

END_IF;

例

演算子(*, /, +, -, <, >, =など)を使用した式

D0 := D1 * D2 + D3 / D4 - D5;

IF D0 > D10 THEN

D0 := D10;

END_IF;

例

定義したファンクションブロックの呼出し

//FBデータ名: LINE1_FB

//入力変数: I_Test

//出力変数: O_Test

//入出力変数: IO_Test

//FBラベル名: FB1

FB1(I_Test:= D0, O_Test => D1, IO_Test:= D100);

例

汎用ファンクションの呼出し

(* BOOL型データをINT型/DINT型データへ変換 *)

wLabel2 := BOOL_TO_INT(bLabel1);

例

漢字などの全角文字の使用

//タンクのリミッタがONした時にバルブを閉じ、OFFした時にバルブを開く

IF タンクリミッタ = TRUE THEN

バルブ := FALSE; (* リミッタがONしたためバルブを閉じる *)

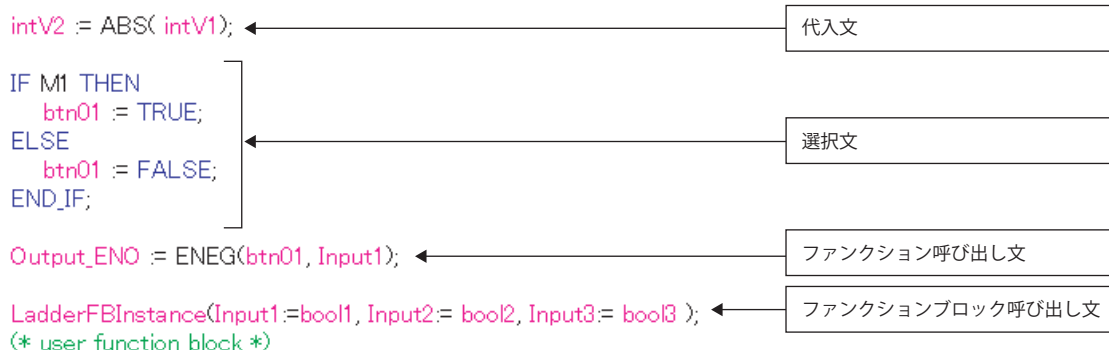
ELSE

バルブ := TRUE; /* リミッタがOFFしたためバルブを開く */

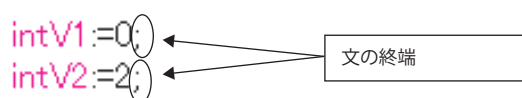
END_IF;

4.1 構成

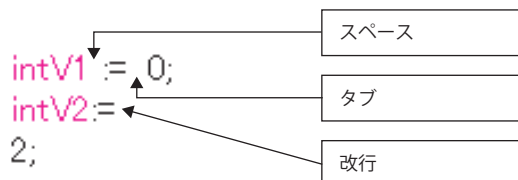
ST言語でのプログラミングは、演算子と構文で構成されます。



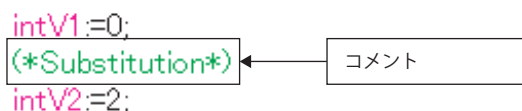
文の終端は、必ず「;」(セミコロン)を付けます。



スペース、タブ、改行は、演算子やデータの間に挿入できます。



プログラム中にコメントを挿入することができます。コメント文の前後に「(*コメント文*)」を記述します。



プログラムの構成要素

STプログラムは下記の要素から構成されます。

項目		例	参照先
区切り記号		;; (,)	☞ 93ページ 区切り記号
演算子		+, -, <, >, =	☞ 93ページ 演算子
予約語	構文	IF, CASE, WHILE, RETURN	☞ 95ページ 構文
	デバイス	X40, Y50, M100, ZR0	📖 使用するコントローラのユーザーズマニュアル
	データ型	BOOL, DWORD	📖 使用するコントローラのユーザーズマニュアル
	汎用ファンクション	ADD, REAL_TO_STRING_E	☞ 176ページ 汎用ファンクション
定数		123, "abc"	☞ 104ページ 定数
ラベル		Switch_A	☞ 105ページ ラベルとデバイス
コメント		(*ONする*)	☞ 107ページ コメント
その他の記号		半角スペース, 改行コード, TABコード	—

- 区切り記号、演算子、予約語は半角で記述してください。
- 予約語の詳細については、下記を参照してください。

📖 GX Works3 オペレーティングマニュアル

区切り記号

ST言語では、プログラムの構造を明確にするために、下記の区切り記号があります。

記号	内容
()	丸括弧式
[]	配列要素番号の指定
.(ピリオド)	構造体、ファンクションブロックのメンバの指定
,(カンマ)	引数の区切り
:(コロロン)	デバイス型指定子、CASE文の区切り
;(セミコロン)	文の終端
"(ダブルクォーテーションマーク)	Unicode文字列の表記
'(シングルクォーテーションマーク)	文字列(ASCII, シフトJIS)の表記
..(ピリオド2つ)	整数範囲指定

演算子

STプログラムで使用する演算子、対象データ型と演算結果のデータ型を示します。

演算子	対象データ型	演算結果型
*, /, +, -	ANY_NUM ^{*1}	ANY_NUM
<, >, <=, >=, =, <>	ANY_ELEMENTARY ^{*2}	ビット
MOD	ANY_INT	ANY_INT
AND, &, XOR, OR, NOT	ANY_BIT	ANY_BIT
**	ANY_REAL(底) ANY_NUM(指数) ^{*1}	ANY_REAL

^{*1} 「WORD#」または「DWORD#」による定数の表記は指定できません。(104ページ 定数の表記方法)

^{*2} WSTRING型のUnicode文字列は指定できません。

演算子の優先順位を示します。

演算子	内容	例	優先順位
()	丸括弧式	(2+3)*(4+5)	1
ファンクション()	ファンクション呼び出し式	CONCAT('AB','CD')	2
-	符号の反転	-10	3
NOT	ビット型補数	NOT TRUE	
**	べき乗	3.0**4	4
*	乗算	10*20	5
/	除算	20/10	
MOD	剰余算	17 MOD 10	
+	加算	1.4+2.5	6
-	減算	3-2	
<, >, <=, >=	比較	10>20	7
=	一致	T#26h=T#1d2h	8
<>	不一致	8#15<>13	
&, AND	論理積	TRUE AND FALSE	9
XOR	排他的論理和	TRUE XOR FALSE	10
OR	論理和	TRUE OR FALSE	11

- 1つの式の中に、優先順位が同じ演算子が複数ある場合は、左側の演算子から演算されます。
- 1つの式に記述できる演算子の使用個数は、最大1024個です。

命令

STプログラムで使用する命令について示します。

(1) (2) (3)

↓ ↓ ↓

`bResult := BMOV( TRUE, wValue0, 1, wValue1 );`

- (1) 戻り値を指定します。
(上記例では、ENOをbResultに代入しています。命令の場合、戻り値の記述は省略できます。)
- (2) 命令名を指定します。
- (3) 命令の引数を指定します。

命令のENへの引数指定省略

命令の場合、ENへの引数指定を省略できます。

その場合、ENにTRUEを割り付けているとみなして動作するため、下記のようなプログラム記述ができます。

```
//省略前
MOV(TRUE, SrcLabel, DstLabel);
//省略後
MOV(SrcLabel, DstLabel);
```

汎用ファンクション等、命令以外ではENは省略できません。

構文

STプログラムで使用できる構文を示します。

項目	内容	参照先
代入文	代入文	95ページ 代入文
サブプログラム制御文	ファンクションブロック呼び出し文, ファンクション呼び出し文	97ページ サブプログラム制御文
	RETURN文	
選択文	IF文(IF THEN, IF ELSE, IF ELSIF)	98ページ 選択文
	CASE文	
繰り返し文	FOR文	98ページ 繰り返し文
	WHILE文	
	REPEAT文	
	EXIT文	

構文は半角で記述してください。

代入文

書式	内容	記述例
<左辺>:=<右辺>;	右辺の式の結果を左辺のラベルやデバイスに代入する機能を持ちます。 右辺の式の結果と左辺のデータ型が同じである必要があります。	intV1:=0; intV2:=2;

配列型ラベルや構造体ラベルを使用する場合は、代入文の左辺と右辺のデータ型に注意してください。

配列型ラベルの場合、データ型と要素数が左辺と右辺で同じである必要があります。また要素は指定しないでください。

例

```
intAry1:=intAry2;
```

構造体ラベルの場合、データ型(構造体のデータ型)が左辺と右辺で同じである必要があります。

例

```
dutVar1:=dutVar2;
```

右辺がファンクション呼び出し式の場合、ファンクションの戻り値が左辺に代入されます。戻り値をラベルに代入する例を示します。

ファンクション	記述例
入力変数が1つのファンクションの場合(例: ABS)	Output1 := ABS(Input1);
入力変数が3つのファンクションの場合(例: MAX)	Output1 := MAX(Input1 , Input2 , Input3);
EN/ENOのあるファンクション(汎用ファンクション)の場合(例: MAX_E)	Output1 := MAX_E(boolEN , boolENO , Input1 , Input2 , Input3);
汎用ファンクション以外の場合(例: MOV)	boolENO := MOV(boolEN , Input1 , Output1); (ファンクションを実行した結果はENO, 第一引数(変数1)はENになります。)

■データ型の自動変換

ST言語では、異なるデータ型の代入や算術演算式を記述したときに、自動でデータ型を変換する場合があります。

例

自動変換の例

```
dintLabel1 := intLabel1 ;  
//代入文: INT型(intLabel1)の値をDINT型に自動変換して、左辺のDINT型(dintLabel1)に代入  
dintLabel1 := dintLabel2 + intLabel1 ;  
//算術演算式: INT型(intLabel1)の値をDINT型に自動変換して、DINT型の加算を実行  
DMOV(TRUE, wordLabel1, dwordLabel1);  
//命令、ファンクション、ファンクションブロック呼び出し文: WORD型入力引数(wordLabel1)の値を、DWORD型に自動変換して、転送を実行
```

型変換は、代入文、ファンクションブロックとファンクション(命令、汎用ファンクション、汎用ファンクションブロックを含む)への入力引数渡し(VAR_INPUT部)、算術演算式で行われます。

型変換時にデータが失われないようにするため、サイズが小さいデータ型からサイズが大きいデータ型へのみ変換されます。型変換は基本データ型のうち下記のデータ型が対象となります。

データ型	内容
ワード[符号付き]	変換後がダブルワード[符号付き]の場合、符号拡張した値に自動変換します。 単精度実数または倍精度実数の場合、変換前の整数と同じ値に自動変換します。 ^{*1}
ワード[符号なし]/ビット列[16ビット]	変換後がダブルワード[符号なし]/ビット列[32ビット]、またはダブルワード[符号付き]の場合、ゼロ拡張した値に自動変換します。 ^{*2} 単精度実数または倍精度実数の場合、変換前の整数と同じ値に自動変換します。 ^{*1}
ダブルワード[符号付き]	変換後が倍精度実数の場合、変換前の整数と同じ値に自動変換します。
ダブルワード[符号なし]/ビット列[32ビット]	
単精度実数	変換後が倍精度実数の場合、同じ値に自動変換します。

*1 16ビットのデータ(ワード[符号付き]またはワード[符号なし]/ビット列[16ビット])を、データ型がANY_REALの入力引数に引き渡した場合、単精度実数に自動変換します。

*2 ワード[符号なし]/ビット列[16ビット]のデータを、データ型がANY32の入力引数に引き渡した場合、ダブルワード[符号なし]/ビット列[32ビット]に自動変換します。

上記以外のデータ型では、型変換ファンクションを使用してください。

また下記の場合も、型変換が行われなため、型変換ファンクションを使用してください。

- 符号の異なる整数型同士の型変換
- データが失われる型同士の型変換

算術演算の結果を代入する場合の注意事項は下記を参照してください。

☞ 99ページ 算術演算式の結果を代入する場合

デバイスを使用する場合の注意事項は下記を参照してください。

☞ 106ページ デバイスを使用する場合のデータ型自動変換

サブプログラム制御文

■ファンクションブロック呼び出し文

書式	内容
インスタンス名(入力変数1:=変数1,...出力変数1=>変数2,...);	インスタンス名の後に、「()」で入力変数、出力変数への代入文を囲みます。 変数が複数ある場合は、各代入文の間を「,」(カンマ)で区切ります。
インスタンス名.入力変数1:=変数1; : インスタンス名(); 変数2:=インスタンス名.出力変数1;	ファンクションブロック呼び出しの前後で、入力引数、出力引数への代入文を列挙します。

ファンクションブロック呼び出し文の引数で使用する記号と、割付可能な式を示します。

種類	内容	使用する記号	割付可能な式
EN, VAR_INPUT	入力変数	:=	全ての式
ENO, VAR_OUTPUT, VAR_OUTPUT_RETAIN	出力変数	=>	変数のみ
VAR_IN_OUT	入出力変数	:=	変数のみ

ファンクションブロックの実行結果は、インスタンス名の後に「.」(ピリオド)を付けて出力変数を指定して、変数へ代入することにより格納されます。

ファンクションブロック	FB定義	記述例
入力変数が1つ、出力変数が1つのファンクションブロックの場合	FB名: FBADD FBインスタンス名: FBADD1 入力変数1: IN1 出力変数1: OUT1	FBADD1(IN1:= Input1); Output1 := FBADD1.OUT1;
入力変数が3つ、出力変数が2つのファンクションブロックの場合	FB名: FBADD FBインスタンス名: FBADD1 入力変数1: IN1 入力変数2: IN2 入力変数3: IN3 出力変数1: OUT1 出力変数2: OUT2	FBADD1(IN1:= Input1 ,IN2:= Input2 ,IN3:= Input3); Output1 := FBADD1.OUT1; Output2 := FBADD1.OUT2;

■ファンクション呼び出し文

戻り値のないファンクションや、パラメータにVAR_OUTPUT変数を含むファンクションは、ファンクション呼び出し式の後に「;」(セミコロン)をつけることで、文として実行できます。

書式	内容
ファンクション名(変数1,変数2,...);	ファンクション名に続いて、「()」で引数を囲みます。 引数が複数ある場合は「,」で区切ります。

■RETURN文

構文	書式	内容	記述例
■RETURN	RETURN;	プログラム、ファンクションブロック、ファンクションを途中で終了するために使用します。 RETURN文をプログラムで使用すると、プログラムの最後の文の次のステップにジャンプします。 RETURN文をファンクションブロックで使用すると、ファンクションブロックからリターンします。 RETURN文をファンクションで使用すると、ファンクションからリターンします。 RETURN文1つに対し、システムでポインタ型ラベルを1点使用します。	IF bool1 THEN RETURN; END_IF;

選択文

構文	書式	内容	記述例
■IF THEN	IF<ブール式>THEN <文...>; END_IF;	ブール式(条件式)が真(TRUE)のときに、文を実行します。ブール式が偽(FALSE)の場合、文は実行されません。ブール式は、単一のビット型変数の状態、あるいは数多くの変数を含んだ複雑な式のブール演算の結果として、真(TRUE)か偽(FALSE)を返す式であればどのような式でも使用できます。	IF bool1 THEN intV1 := intV1 + 1; END_IF;
■IF...ELSE	IF<ブール式>THEN <文1...>; ELSE <文2...>; END_IF;	ブール式(条件式)が真(TRUE)のときに、文1を実行します。ブール式の値が偽(FALSE)の場合は、文2を実行します。	IF bool1 THEN intV3 := intV3 + 1; ELSE intV4 := intV4 + 1; END_IF;
■IF...ELSIF	IF<ブール式1>THEN <文1...>; ELSIF<ブール式2>THEN <文2...>; ELSIF<ブール式3>THEN <文3...>; END_IF;	ブール式(条件式)1が真(TRUE)のときに、文1を実行します。ブール式1の値が偽(FALSE)でブール式2の値が真(TRUE)の場合は文2を実行します。ブール式1、2の値が偽(FALSE)でブール式3の値が真(TRUE)の場合は、文3を実行します。	IF bool1 THEN intV1 := intV1 + 1; ELSIF bool2 THEN intV2 := intV2 + 2; ELSIF bool3 THEN intV3 := intV3 + 3; END_IF;
■CASE	CASE<整数式>OF <整数選択値1>: <文1...>; <整数選択値2>: <文2...>; : <整数選択値n>: <文n...>; ELSE <文n+1...>; END_CASE;	整数式(条件式)の値と一致する整数の選択値を持った文が実行され、一致するものがない場合は、ELSE文に続く文を実行します。CASE文は、例えば単一の整数値や複雑な式の結果の整数値によって、選択文を実行する場合に使用できます。	CASE intV1 OF 1 : bool1 := TRUE; 2 : bool2 := TRUE; ELSE intV1 := intV1 + 1; END_CASE;

繰り返し文

構文	書式	内容	記述例
■FOR...DO	FOR<反復変数初期化> TO<最終値> BY<増加式>DO <文...>; END_FOR;	反復変数として使用するデータの初期化を行います。初期化された反復変数を増加式に従って加算または減算し、最終値を超えるまで、DOからEND_FOR内の1つ以上の文を繰り返し実行します。FOR...DO文終了後の反復変数は、終了時点での値を保持しています。	FOR intV1 := 0 TO 30 BY 1 DO intV3 := intV1 + 1; END_FOR;
■WHILE...DO	WHILE<ブール式>DO <文...>; END_WHILE;	ブール式(条件式)が真(TRUE)の間、1つ以上の文を実行します。ブール式は文の実行に先立って判定され、ブール式が偽(FALSE)の場合は、DO...END_WHILE内の文が実行されません。WHILE文中の<ブール式>は、結果が真か偽かを返すものであれば良いため、IF文中の<ブール式>で指定可能な式はすべて使用することができます。	WHILE intV1 = 30 DO intV1 := intV1 + 1; END_WHILE;
■REPEAT...UNTIL	REPEAT <文...>; UNTIL<ブール式> END_REPEAT;	ブール式(条件式)が偽(FALSE)の間、1つ以上の文を実行します。ブール式は文の実行後に判定され、値が真(TRUE)の場合は、REPEAT...UNTIL内の文が実行されません。REPEAT文中の<ブール式>は、結果が真か偽かを返すものであれば良いため、IF文中の<ブール式>で指定可能な式はすべて使用することができます。	REPEAT intV1 := intV1 + 1; UNTIL intV1 = 30 END_REPEAT;
■EXIT	EXIT;	繰り返し文の中でのみ使用可能な構文で、繰り返し文を途中で終了させます。反復ループを実行中にEXIT文に達すると、EXIT文以降の反復ループ処理は実行されません。繰り返し文を終えた次の行からプログラムの実行を継続します。	FOR intV1 := 0 TO 10 BY 1 DO IF intV1 > 10 THEN EXIT; END_IF; END_FOR;

注意事項

■代入文使用時

- 文字列の代入は、最大文字列長255文字です。最大文字列長を超える文字列を代入した場合は、変換エラーになります。
- タイマ型、カウンタ型の接点とコイルは、代入文の左辺には使用できません。
- ファンクションブロックのインスタンスは、代入文の左辺には使用できません。インスタンスの入力変数、出力変数、外部変数を代入式の左辺に使用してください。

■算術演算式の結果を代入する場合

算術演算式の結果を、データサイズの大きなデータ型の変数に代入する場合は、あらかじめ算術演算式の変数を左辺のデータ型に変換してから、演算を行ってください。

例

データサイズ16ビット(INT型)の算術演算結果を、32ビットのデータ型(DINT型)へ代入する場合

```
varDint1 := varInt1 * 10; //varInt1はINT型, varDint1はDINT型
```

算術演算式の演算結果は、入力オペランドのデータ型と同じデータ型となります。そのため上記のプログラムでは、varInt1*10の演算結果がINT型の範囲(-32768~32767)を超えた場合、オーバーフローまたはアンダーフローした演算結果がvarDint1に代入されます。

この場合、あらかじめ演算式のオペランドを左辺のデータ型に変換してから演算を行ってください。

```
varDint2 := INT_TO_DINT( varInt1 ); //INT型変数をDINT型変数に変換
varDint1 := varDint2 * 10; //DINT型で乗算して、演算結果を代入
```

■算術演算式で符号反転の演算子を使う場合

データ型の最小値に対して、符号反転の演算子(-)を使用すると、同じ値になります。

例えばINT 型の最小値の場合、-(-32768) = -32768となります。

そのためデータ型の自動変換の対象となる変数に、符号反転の演算子を使用すると、意図した結果にならない場合があります。

例

varInt1(INT型)の値が-32768、varDint1(DINT型)の値が0の場合

```
varDint2 := -varInt1 + varDint1;
```

この場合、(-varInt1)の値は-32768のままとなり、varDint2には-32768が代入されます。

算術演算式で符号反転の演算子を使う場合、あらかじめ算術演算の前にデータ型の自動変換を行うか、または符号反転の演算子を使わないプログラムを作成してください。

例

算術演算の前にデータ型の自動変換を行う場合

```
varDint3 := varInt1;
varDint2 := -varDint3 + varDint1;
```

例

符号反転の演算子を使わない場合

```
varDint2 := varDint1 - varInt1;
```

■単精度実数から倍精度実数へデータ型を変換する場合

単精度実数から倍精度実数への型変換(REAL_TO_LREAL)では、変換結果に誤差が生じる場合があります。
そのためデータ型の自動変換が行われる場合や、戻り値が実数型のファンクション(SINなど)を代入文の右辺や算術演算式のオペランドに使用した場合、意図した結果にならない場合があります。

例

誤差が生じる場合

```
varReal1 := -1234.567;  
varLReal1 := ABS(varReal1);
```

上記の場合、ABS(varReal1)の戻り値は単精度実数となり、その値を倍精度実数に型変換してvarLReal1に代入するため、誤差が生じます。

このような場合、代入先と同じデータ型(倍精度実数)でファンクションを実行するプログラムを作成してください。

例

誤差が生じない場合

```
varLReal2 := -1234.567;  
varLReal1 := ABS(varLReal2);
```

■ビット型ラベル使用時

選択文または繰り返し文でブール式(条件式)が一度成り立つと、<文>内でビット型ラベルをONしている場合、そのビット型ラベルは常時ONとなります。

例

常時ONとなるプログラム

STプログラム	STプログラム同等の処理となるラダープログラム
IF bLabel1 THEN bLabel2 := TRUE; END_IF;	

常時ONとなるのを避けるには、下記のようにビット型ラベルをOFFするプログラムを追加してください。

例

常時ONとなるのを避けるプログラム

STプログラム*1	STプログラム同等の処理となるラダープログラム
IF bLabel1 THEN bLabel2 := TRUE; ELSE bLabel2 := FALSE; END_IF;	

*1 上記のプログラムは、下記のように記述することもできます。
bLabel2:=bLabel1;
または
OUT(bLabel1,bLabel2);
ただし、OUT命令を<文>内で使用した場合は、常時ONとなるプログラムと同様の状態となります。

■タイマファンクションブロック、カウンタファンクションブロック使用時

選択文でブール式(条件式)は、タイマファンクションブロック、カウンタファンクションブロックの実行条件とは異なります。

例

タイマファンクションブロックの場合

変更前プログラム例

```
IF bLabel1 THEN
    TIMER_100_FB_M_1(Coil:= bLabel2 , Preset:= wLabel3 , ValueIn:= wLabel4 , ValueOut=> wLabel5 , Status=> bLabel6 );
END_IF;
(* bLabel1=ONかつbLabel2=ON時, 計数を開始します。*)
(* bLabel1=ONかつbLabel2=OFF時, 計数をクリアします。*)
(* bLabel1=OFFかつbLabel2=ON時, 計数を停止します。計数値はクリアしません。*)
(* bLabel1=OFFかつbLabel2=OFF時, 計数を停止します。計数値はクリアしません。*)
```

変更後プログラム例

```
TIMER_100_FB_M_1(Coil:= (bLabel1 & bLabel2) ,Preset:= wLabel3 ,ValueIn:= wLabel4 ,ValueOut=> wLabel5 ,Status=> bLabel6 );
```

例

カウンタファンクションブロックの場合

変更前プログラム例

```
IF bLabel1 THEN
    COUNTER_FB_M_1(Coil:= bLabel2 , Preset:= wLabel3 , ValueIn:= wLabel4 , ValueOut=> wLabel5 , Status=> bLabel6 );
END_IF;
(* bLabel1=ONかつbLabel2=ON/OFF時, 計数を+1します。*)
(* bLabel1=OFFかつbLabel2=ON/OFF時, 計数しません。*)
(* bLabel1=ON/OFFと計数+1とは連動していません。*)
```

変更後プログラム例

```
COUNTER_FB_M_1(Coil:= (bLabel1 & bLabel2) ,Preset:= wLabel3 ,ValueIn:= wLabel4 ,ValueOut=> wLabel5 ,Status=> bLabel6 );
```

上記の変更前プログラム例は、選択文が成立しない場合にタイマ、カウンタに関連する文が実行されないために起こります。

bLabel1の条件とbLabel1のAND条件によりタイマ、カウンタを動作させる場合は、制御文ではなく、ファンクションブロックのみを使用してください。

変更後のプログラムを使用することにより、タイマ、カウンタを動作させることができます。

■FOR...DO文使用時

- 反復変数には、構造体メンバおよび配列要素は使用できません。
- 反復変数で使用した型と<最終値の式>、<増加式>の型を一致させてください。
- <増加式>は省略可能です。省略した場合<増加式>は1として実行します。
- <増加式>に0を代入すると、FOR構文以下が実行されなくなることや、無限ループとなることがあります。
- FOR...DO構文ではFOR構文中の<文...>の実行後に反復変数のカウント処理を行います。反復変数のデータ型の最大値を上回るまたは最小値を下回るカウント処理が実行された場合、無限ループが発生します。

■立上り命令、立下り命令使用時

- IF文およびCASE文の実行条件がFALSEまたはCASE不一致になった場合、前回スキャン時のIF文およびCASE文中の立上り命令、立下り命令の実行結果は初期化されず、次の実行時に引き継がれます。
- 繰り返し文(FOR文、WHILE文またはREPEAT文)が終了後、前回スキャン時の繰り返し文の中の立上り命令、立下り命令の実行結果は初期化されず、次の実行時に引き継がれます。
- IF文およびCASE文の中で、立上り命令、立下り命令を使用する場合の動作を示します。

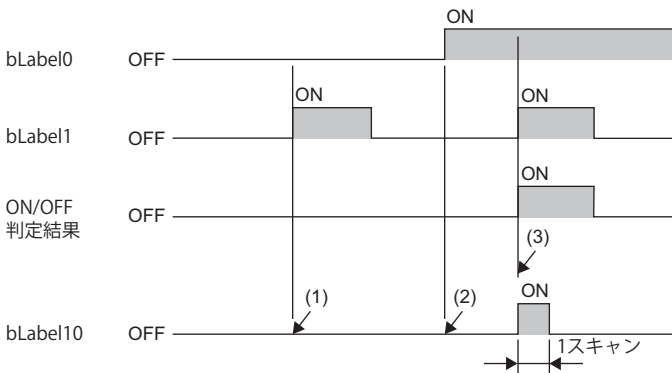
条件			動作結果		
IF文、CASE文の条件式	命令実行条件(EN)	前回スキャン時の命令のON/OFF判定結果	命令のON/OFF判定結果	立上り命令	立下り命令
TRUEまたはCASE一致	TRUE	ON	ON	実行しない	実行しない
		OFF	ON	実行する	実行しない
	FALSE	ON	OFF	実行しない	実行する
		OFF	OFF	実行しない	実行しない
FALSEまたはCASE不一致	TRUE	ON	OFF	実行しない	実行しない ^{*1}
		OFF	OFF	実行しない	実行しない
	FALSE	ON	OFF	実行しない	実行しない ^{*1}
		OFF	OFF	実行しない	実行しない

*1 立下り(ON→OFF)ですが、IF文またはCASE文の条件が不成立のため、命令は実行しません。

例

IF文の中でPLS命令(立上り実行)を使用した場合

```
IF bLabel0 THEN
    PLS(bLabel1, bLabel10);
END_IF;
```



- (1) bLabel0 = OFFの場合(IF文の条件式はFALSE)、ON/OFF判定結果はOFFとなり、PLS命令は実行しません。(bLabel10 = OFFのまま)
- (2) bLabel0 = ON(IF文の条件式はTRUE)かつ、bLabel1 = OFF(命令実行条件はOFF)の場合、ON/OFF判定結果はOFFとなり、PLS命令は実行しません。(bLabel10 = OFFのまま)
- (3) bLabel0 = ON(IF文の条件式はTRUE)かつ、bLabel1 = ON(命令実行条件はON)の場合、ON/OFF判定結果はOFF→ONとなり(立上り条件が成立)、PLS命令が実行されます。(bLabel10が1スキャンだけON)

- ・繰り返し文(FOR文, WHILE文またはREPEAT文)で, 立上り命令または立下り命令を実行させる場合は, エッジリレー (V) およびインデックス修飾を使用します。この場合, エッジリレー (V)を使用している命令ごとに, システムで1点エッジリレー (V)を使用します。そのため繰り返し文で使用する点数に加えて, 使用している命令数分のエッジリレー (V)を確保してください。

例

FOR文内で立上り命令と立下り命令を使用した場合

エッジリレー (V)を1箇所で使用した例

(エッジリレー (V)を合計11点(INC命令でV0~V10)まで使用します。)

```
FOR Z0 := 0 TO 9 BY 1 DO
  INC(EGP(M100Z0, V0Z0), D100Z0);
END_FOR;
```

エッジリレー (V)を2箇所で使用した例

(エッジリレー (V)を合計22点(INC命令でV0~V10, DEC命令でV11~V21)まで使用します。)

```
FOR Z0 := 0 TO 9 BY 1 DO
  INC(EGP(M100Z0, V0Z0), D100Z0);
  DEC(EGF(M200Z0, V11Z0), D200Z0);
END_FOR;
```

■マスタコントロール命令使用時

マスタコントロールOFF時の動作を示します。

- ・選択文(IF文またはCASE文)内, または繰り返し文(FOR文, WHILE文またはREPEAT文)内の文は, 無処理になります。
- ・選択文または繰り返し文の外では, 代入文の場合は無処理, 代入文以外の文は非実行処理になります。

例

選択文(IF文)内の文

```
MC(M0, N1, M1); //マスタコントロールOFF
IF M2 THEN
  M3 := M4; //マスタコントロールOFF時は無処理のため, M3は前スキャン時の値を保持
END_IF;
M20 := MCR(M0, N1);
```

例

選択文または繰り返し文の外の文(ビット代入文の場合)

```
MC(M0, N1, M1); //マスタコントロールOFF
M3 := M4; //マスタコントロールOFF時は無処理のため, M3は前スキャン時の値を保持
M20 := MCR(M0, N1);
```

例

選択文/繰り返し文の外の文(OUT命令の場合)

```
MC(M0, N1, M1); //マスタコントロールOFF
OUT(M2, M3); //マスタコントロールOFF時は非実行処理のため, M3をOFF
M20 := MCR(M0, N1);
```

定数

定数の表記方法

STプログラムでは通常の定数表記の他に、下記の定数表記ができます。

対応可能なデータ型	種類	表記方法 ^{*1}	表記例
ビット	ブール	使用するブール値の前に「BOOL#」をつけます。	BOOL#1, BOOL#0 BOOL#TRUE, BOOL#FALSE
ワード[符号なし]/ビット列 [16ビット]	2進数	2進数の前に「UINT#2#」または「WORD#2#」をつけます。 ^{*2}	UINT#2#10101010 WORD#2#10101010
	8進数	8進数の前に「UINT#8#」または「WORD#8#」をつけます。 ^{*2}	UINT#8#3370 WORD#8#3370
	10進数	10進数の前に「UINT#」または「WORD#」をつけます。 ^{*2}	UINT#123 WORD#123
	16進数	16進数の前に「UINT#16#」または「WORD#16#」をつけます。 ^{*2}	UINT#16#FF WORD#16#FF
ダブルワード[符号なし]/ビット列[32ビット]	2進数	2進数の前に「UDINT#2#」または「DWORD#2#」をつけます。 ^{*2}	UDINT#2#1100110011001100 DWORD#2#1100110011001100
	8進数	8進数の前に「UDINT#8#」または「DWORD#8#」をつけます。 ^{*2}	UDINT#8#33703370 DWORD#8#33703370
	10進数	10進数の前に「UDINT#」または「DWORD#」をつけます。 ^{*2}	UDINT#456789 DWORD#456789
	16進数	16進数の前に「UDINT#16#」または「DWORD#16#」をつけます。 ^{*2}	UDINT#16#FFFF DWORD#16#FFFF
ワード[符号付き]	2進数	2進数の前に「INT#2#」をつけます。	INT#2#01010101
	8進数	8進数の前に「INT#8#」をつけます。	INT#8#3370
	10進数	10進数の前に「INT#」をつけます。	INT#-123
	16進数	16進数の前に「INT#16#」をつけます。	INT#16#1F
ダブルワード[符号付き]	2進数	2進数の前に「DINT#2#」をつけます。	DINT#2#0011001100110011
	8進数	8進数の前に「DINT#8#」をつけます。	DINT#8#33703370
	10進数	10進数の前に「DINT#」をつけます。	DINT#-456789
	16進数	16進数の前に「DINT#16#」をつけます。	DINT#16#1F1F
単精度実数	実数	実数の前に「REAL#」をつけます。	REAL#2.34
	実数(指数表現)		REAL#1.0E6
倍精度実数	実数	実数の前に「LREAL#」をつけます。	LREAL#-2.34
	実数(指数表現)		LREAL#1.001E16
文字列	STRING	文字列(ASCII, シフトJIS)をシングルクォーテーションマーク(')で囲みます。	'ABC'
文字列[Unicode]	WSTRING	Unicode文字列をダブルクォーテーションマーク(")で囲みます。	"ABC"

*1 大文字/小文字は区別しません。またK, H, Eを使用した定数の表記とは同時に使用できません。

*2 ANY_NUMである算術演算子のオペランド、ファンクション呼び出し文、ファンクションブロック呼び出し文、ファンクション呼び出し式の引数の場合、「UINT#」または「UDINT#」による定数表記としてください。「WORD#」または「DWORD#」による定数表記の場合、ビット列と解釈され変換時にエラーとなります。

上記以外の定数の表記方法は、下記を参照してください。

📖 使用するコントローラのユーザーズマニュアル

Point

2進数, 8進数, 10進数, 16進数, 実数の表記では、アンダースコア(_)を使って数値を区切り、プログラムを見やすくさせることができます。例えばダブルワード[符号なし]の2進数表記の場合、下記のとおり記載できます。

UDINT#2#1100_1100_1100_1100

プログラムの処理上では、アンダースコア(_)による数値の区切りは無視されます。

ラベルとデバイス

指定方法

STプログラム上で、ラベルとデバイスを直接記述して使用できます。ラベルとデバイスは式の左辺、右辺、汎用ファンクション/ファンクションブロックの引数、戻り値などに使用できます。

ラベルとデバイスの詳細は、下記を参照してください。

📖 使用するコントローラのユーザーズマニュアル

■型指定付きデバイス表記

ワードデバイスは、デバイス名にデバイス型指定子をつけることで、任意のデータ型として使用できます。

デバイス型指定子	データ型	例	例の説明
なし	総称データ型のANY16 算術演算式などでデバイスのみ使用している場合は、ワード[符号付き]となります。 ただしFUN/FBの引数部分で、型指定なしのデバイスとして指定された場合は、引数定義のデータ型になります。	D0	D0に型指定子を付けない場合
:U	ワード[符号なし]/ビット列[16ビット]	D0:U	D0をワード[符号なし]/ビット列[16ビット]とした値
:D	ダブルワード[符号付き]	D0:D	D0, D1をダブルワード[符号付き]とした値
:UD	ダブルワード[符号なし]/ビット列[32ビット]	D0:UD	D0, D1をダブルワード[符号なし]/ビット列[32ビット]とした値
:E	単精度実数	D0:E	D0, D1を単精度実数とした値
:ED	倍精度実数	D0:ED	D0, D1, D2, D3を倍精度実数とした値

デバイス型指定子が使用可能なデバイスを示します。

- データレジスタ(D)
- リンクレジスタ(W)
- リンクダイレクトデバイス(J□¥W□)
- ユニットアクセスデバイス(U□¥G□)
- CPUバッファメモリアクセスデバイス(U3E□¥G□)
- ファイルレジスタ(R/ZR)
- リフレッシュデータレジスタ(RD)

桁指定または間接指定したデバイスには、デバイス型指定子は付与できません。

■デバイスの指定方法

デバイスの指定には、下記の方法が使用できます。

- インデックス修飾
- 間接指定
- ビット指定(📌 31ページ ビットデータ)
- 桁指定(📌 32ページ 16ビットデータ(ワードデータ), 📌 35ページ 32ビットデータ(ダブルワードデータ))

インデックス修飾と間接指定の詳細については、下記を参照してください。

📖 使用するコントローラのユーザーズマニュアル

注意事項

- STプログラムではポインタ型は使用できません。
- タイマ, カウンタ, 積算タイマのデバイスを現在値で使ったときのデータ型は, ワード[符号なし]/ビット列[16ビット]となります。ロングタイマ, ロングカウンタ, ロング積算タイマのデバイスを現在値で使ったときのデータ型は, ダブルワード[符号なし]/ビット列[32ビット]となります。
- 桁指定を使用して代入をする場合, 右辺と左辺のデータ型は合わせてください。

例

D0:=K5X40;

上記の場合, K5X40がダブルワード型, D0がワード型となるためプログラムはエラーとなります。

- 桁指定を使用して代入をする場合, 右辺>左辺の時は, 左辺の対象点数範囲内にデータ転送します。

例

K5X40:=2#1011_1101_1111_0111_0011_0001;

上記の場合, K5X40は対象点数20点のため, K5X40には1101_1111_0111_0011_0001(20桁)を代入します。

- カウンタ(C), タイマ(T), 積算タイマ(ST)の現在値(TNnなど)をワード[符号なし]/ビット列[16ビット]以外の型で使う場合, またはロングカウンタ(LC), ロングタイマ(LT), ロング積算タイマ(LST)の現在値(LTNnなど)をダブルワード[符号なし]/ビット列[32ビット]以外の型で使う場合は, 型変換ファンクションを使用してください。

例

varInt:=WORD_TO_INT(T0);(*型変換ファンクションを使用*)

■デバイスを使用する場合のデータ型自動変換

ワードデバイスをワード[符号付き]以外のデータ型で使用する場合, デバイス型指定子を付与してください。(105ページ 型指定付きデバイス表記)

例

D2, D3の値をダブルワード[符号なし]のラベルdwordLabel1に転送する場合

//デバイス型指定子を付与して, 正しいデータ型で転送する例

dwordLabel1 := D2:UD;

//デバイス型指定子を付与したD2:UDはダブルワード[符号なし]のため, dwordLabel1にD2, D3の値が転送されます。

//意図しない転送結果となる例

dwordLabel1 := D2;

//デバイス型指定子のないD2はワード[符号付き]のため, データ型をダブルワード[符号なし]に自動変換してから, dwordLabel1に転送します。

//そのためD2の値だけ転送され, D3の値は転送しません。

コメント

STプログラムで使えるコメントを示します。

コメント形式	コメント記号	内容	記述例
単一行コメント	//	開始記号「//」から行末までがコメントとして扱われます。	//コメント内容
複数行コメント	(**)	開始記号「(*)」から終了記号「(*)」までがコメントとして扱われます。 コメント内に改行が入力できます。	■改行なし (*コメント内容*) ■改行あり (*1行目コメント内容 2行目コメント内容*)
	/**/	開始記号「/*」から終了記号「*/」までがコメントとして扱われます。 コメント内に改行が入力できます。	■改行なし /*コメント内容*/ ■改行あり /*1行目コメント内容 2行目コメント内容*/

複数行コメントで、終了記号を含んだコメントは記述しないでください。

5

FBD/LD言語

特定処理を行うブロック、変数部品、定数部品を、データや信号の流れに沿って接続することで、プログラムを記述するグラフィック言語です。

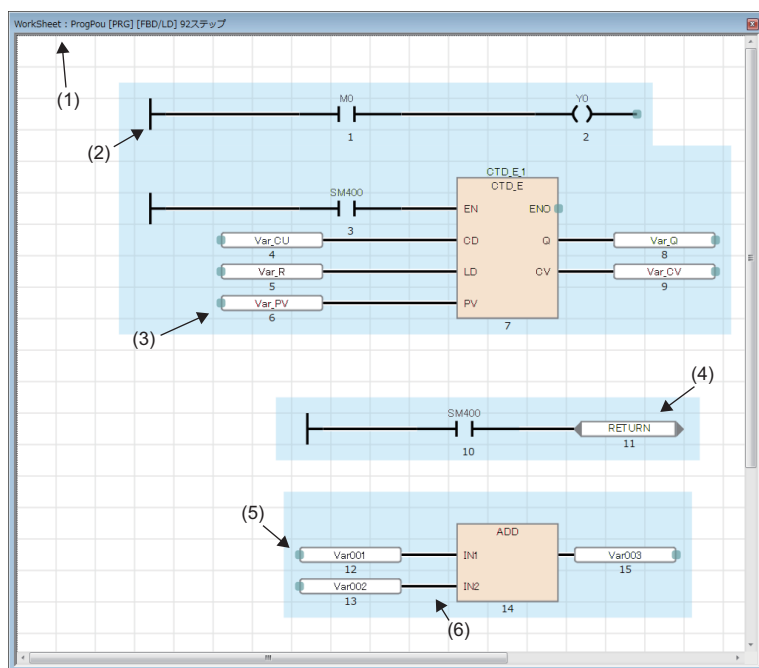
Point

- 本章では、FBD/LD言語の動作や仕様について説明しています。FBD/LDプログラム作成時の操作方法については、下記を参照してください。

 GX Works3 オペレーティングマニュアル

5.1 構成

FBD/LD言語では、下記のようなプログラムが作成できます。

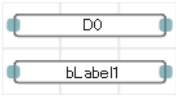
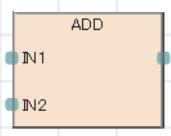
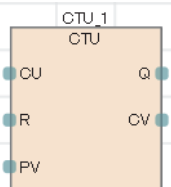


- (1) ワークシート
- (2) LD部品
- (3) FBD部品
- (4) 共通部品
- (5) 接続ポイント
- (6) 接続線

FBD/LD言語のプログラムでは、ファンクションブロック(FB)、ファンクション(FUN)、変数部品(ラベルとデバイス)、定数部品の出力ポイントから、他のファンクションブロックや変数部品などの入力ポイントへデータが流れます。

FBD部品

FBD/LDプログラムを構成するFBD部品を示します。

項目		内容
変数		変数は個々の値(データ)を格納するために使用します。変数にはデータ型が定められており、そのデータ型の値(データ)のみ格納します。 変数には、ラベルまたはデバイスを指定することができます。
定数		指定した定数値を出力します。
ファンクション(FUN)		ファンクションを実行します。 • ファンクションの作成方法(GX Works3 オペレーティングマニュアル) • 汎用ファンクション(176ページ 汎用ファンクション)
ファンクションブロック(FB)		ファンクションブロックを実行します。 • ファンクションブロックの作成方法(GX Works3 オペレーティングマニュアル) • 汎用ファンクションブロック(187ページ 汎用ファンクションブロック) • モーション制御ファンクションブロック(189ページ モーション制御ファンクションブロック一覧) • ユニットFB(使用するユニットのFBリファレンス)

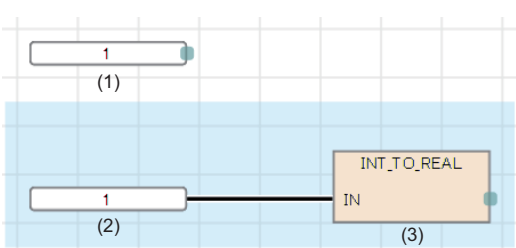
■定数部品のデータ型

定数部品の場合、定数値を入力した時点では定数値のデータ型は決定されません。データ型は定数部品とFBD部品を接続線でつないだときに決定されます。定数値のデータ型は、接続線でつないだ先のFBD部品と同じデータ型になります。

例

定数値に1を入力した場合

データ型候補はBOOL型、WORD型、DWORD型、INT型、DINT型、REAL型、LREAL型が存在するので、データ型を決定できません。定数部品とFBD部品を接続線でつなぐと、つないだ先にある部品の入力ポイントのデータ型になります。



- (1) データ型は未決定
- (2) INT型
- (3) INT型

■データ型の自動変換

接続される部品のデータ型が異なっているとき、自動でデータ型を変換する場合があります。

型変換時にデータが失われないようにするため、サイズが小さいデータ型からサイズが大きいデータ型へのみ変換されます。FBD/LD言語でのデータ型の自動変換は、ST言語と同じ動作です。詳細は、下記を参照してください。

96ページ データ型の自動変換

■ファンクションの入出力ポイント

- ファンクションは、すべての入力ポイントをほかの部品と接続しておく必要があります。
- ファンクションの入力変数と出力変数は、データ型が決まっており、入力ポイントと出力ポイントへ接続する部品もそれらのデータ型に合わせる必要があります。

■EN付きファンクション、EN付きファンクションブロックの出力変数を別の部品と接続する場合の注意事項

EN付きファンクションおよびEN付きファンクションブロックのENOがFALSE(演算停止)となる場合、接続する部品により、プログラムの動作が異なります。

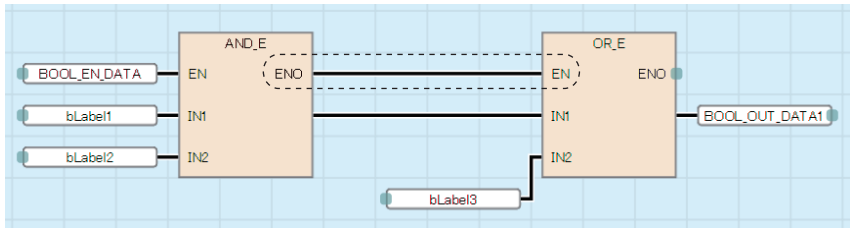
接続する部品	EN付きファンクションおよびEN付きファンクションブロックのENOがFALSE(演算停止)となる場合のプログラムの動作
・ファンクション ・ファンクションブロック	EN付きファンクションおよびEN付きファンクションブロックの出力変数に接続した入力変数の値は不定値となります。
コイル	EN付きファンクションおよびEN付きファンクションブロックの出力変数に接続したコイルの値はFALSEとなります。また、反転形コイルの場合はTRUEとなります。
上記以外	EN付きファンクションおよびEN付きファンクションブロックの出力変数に接続した部品の値は変更されません。(前回値となります。)

EN付きファンクションブロックの出力変数を別のファンクションまたはファンクションブロックの入力変数に直接接続すると、予期しない動作となる場合があります。

不定値が使用されないようにするために、下記に示す例のいずれかの方法で接続してください。

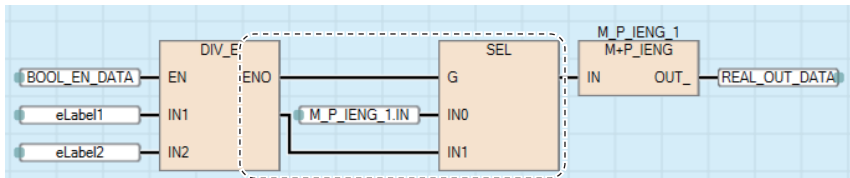
例

接続先をEN付きファンクションまたはEN付きファンクションブロックとし、接続元のENOと接続先のENも接続します。



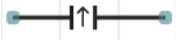
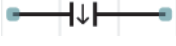
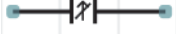
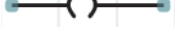
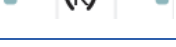
例

選択ファンクションSELを使用し、ENOがFALSEとなる場合は前回値が入力されるようにします。



LD部品

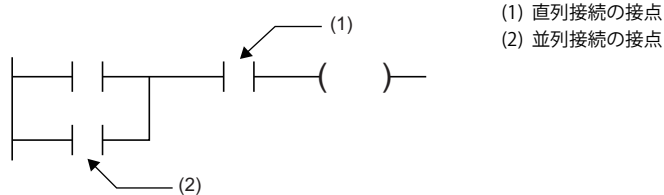
FBD/LDプログラムで使えるラダー部品を示します。

項目		内容
左母線		母線を示す部品です。ラダー回路を作成する時の起点になります。 左母線の出力は常時ONとなります。
a接点		指定デバイスまたはラベルがONのとき導通します。
b接点		指定デバイスまたはラベルがOFFのとき導通します。
立上りパルス		指定デバイスまたはラベルの立上り時(OFF→ON)に導通します。
立下りパルス		指定デバイスまたはラベルの立下り時(ON→OFF)に導通します。
立上りパルス否定		指定デバイスまたはラベルのOFF時, ON時, および立下り時(ON→OFF)に導通します。
立下りパルス否定		指定デバイスまたはラベルのOFF時, ON時, および立上り時(OFF→ON)に導通します。
コイル		演算結果を指定デバイスまたはラベルへ出力します。
反転形コイル		演算結果がOFFすると, 指定デバイスまたはラベルがONします。
セットコイル		演算結果がONすると, 指定デバイスまたはラベルがONします。 ONしたデバイスまたはラベルは, 演算結果がOFFになってもONのまま保持されます。
リセットコイル		演算結果がONすると指定デバイスまたはラベルがOFFします。 演算結果がOFFの場合, デバイスまたはラベルの状態は変化しません。

■接点記号のAND演算とOR演算

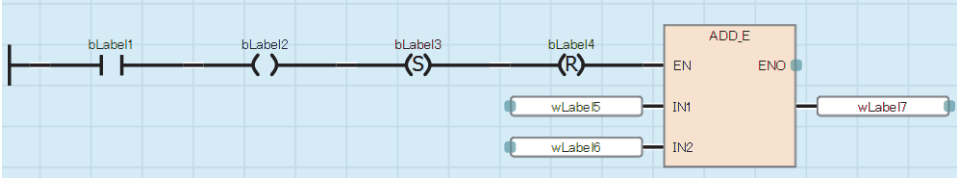
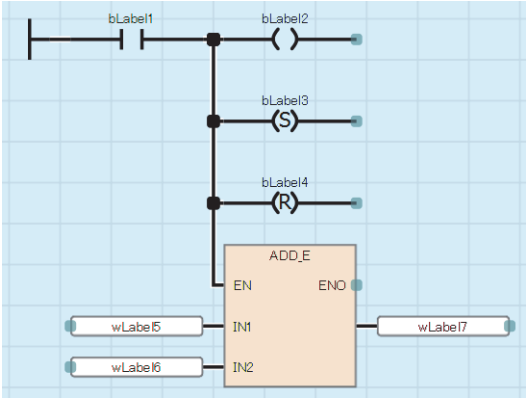
接点記号は, 回路図の接続状態に応じて, AND演算, OR演算を行い, 演算結果に反映します。

- 直列接続(1)のときは, それまでの演算結果とAND演算を行い, 演算結果とします。
- 並列接続(2)のときは, それまでの演算結果とOR演算を行い, 演算結果とします。



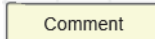
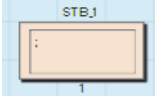
■コイルの出力接続ポイントに他の部品を接続する場合

コイルの出力接続ポイントに他の部品を直列接続すると、並列接続した場合と同じ動作になります。

項目	内容
プログラム例	
上記例の動作	

共通部品

FBD/LDプログラム上に配置する共通部品を示します。

項目		内容
ジャンプ		ジャンプ部品からジャンプラベルまで実行処理をジャンプします。ジャンプした部分については、実行されません。 ジャンプ部品へのON/OFF情報により、ジャンプを行うかどうかを制御します。 ON:ジャンプラベルに実行処理をジャンプする。 OFF:ジャンプせず、通常の実行処理を行う。
ジャンプラベル		同一プログラム内のジャンプ部品からの、ジャンプ先になります。ジャンプを行った場合は、ジャンプラベル以降の実行順序のプログラムから処理が実行されます。
コネクタ		接続線の代わりとして使用します。 対となるコネクタ部品に処理が移動します。 一つの出力コネクタに対して、一つまたは複数の入力コネクタを使用できます。
リターン		プログラム上のリターン部品以降の処理を中断します。リターン部品以降のプログラムやファンクション、ファンクションブロックの処理を実行させない場合に使用します。 リターン部品へのON/OFF情報により、リターン処理を行うかどうかを制御します。 ON:リターン処理を実行する。 OFF:リターン処理を行わず、通常の実行処理を行う。
コメント		コメントを記載する場合に使用します。
インラインST		FBD/LDエディタ内でSTプログラムを表示します。 挿入したインラインST部品をダブルクリックすることで、STエディタを表示しSTプログラムの編集/モニタを行います。 詳細は下記を参照してください。 ☞ 118ページ インラインST

■ジャンプ部品について

- ジャンプ部品でコイルのONしているタイマをジャンプさせた場合、正常な計測ができなくなります。
- ジャンプ部品の上側(実行順序が前)にジャンプラベルを配置できます。その場合、ウォッチドッグタイマの設定値を超えないように、ループから抜け出す方法を含めてプログラムを作成してください。
- ジャンプ部品とジャンプラベルには、ポインタ型のローカルラベルのみ指定できます。なお、構造体のメンバは使用できません。
- ポインタ分岐命令(CJ, SCJ, JMP)は使用できません。ジャンプする場合は、ジャンプ部品を使用してください。
- プログラムブロックの外側へのジャンプや外側からのジャンプはできません。

ジャンプ関連の動作	実行可否
プログラムブロックの外側へのジャンプ*1	実行不可
プログラムブロックの外側からのジャンプ*1	実行不可
サブルーチンプログラムの呼出し	実行可
サブルーチンプログラムとして呼び出される	実行不可

*1 BREAK命令による分岐を含みます。

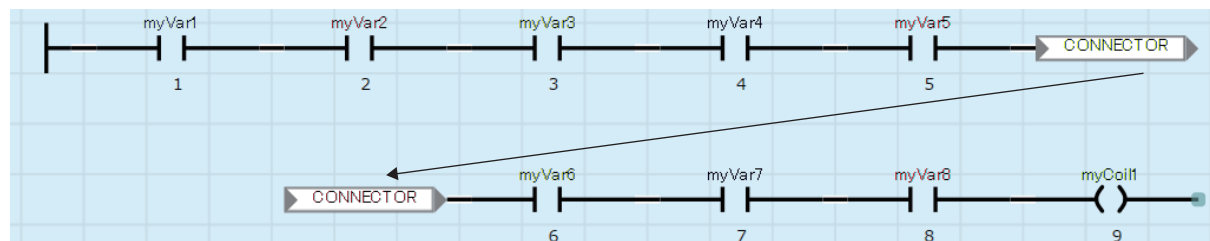
■リターン部品について

- リターン部品は、使用するプログラムやファンクション、ファンクションブロックによって動作が異なります。

使用するプログラム部品	内容
プログラム	プログラム部品の実行を終了します。
ファンクション	ファンクションを終了し、ファンクションを呼び出した命令の次のステップに戻ります。
ファンクションブロック	ファンクションブロックを終了し、ファンクションブロックを呼び出した命令の次のステップに戻ります。

■コネクタ部品について

コネクタ部品は、FBD/LDエディタの表示範囲内または印刷範囲内に、プログラムを配置したい場合に使用します。



接続線

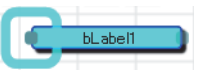
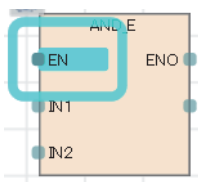
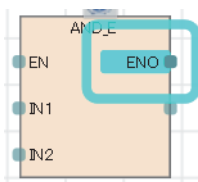
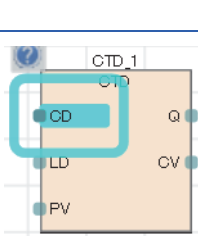
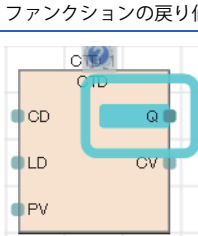
FBD部品、LD部品、共通部品の接続ポイント間をつなぐ線です。

部品をつなげて左端から右端に向かってデータを渡します。接続している部品のデータ型は一致しているか、データ型の自動変換が可能な型である必要があります。

接続ポイント

接続線によりFBD部品、LD部品、共通部品をつなげる際の端点です。

各部品の左側のポイントが入力側、右側のポイントが出力側を示します。

項目	入力接続ポイント	出力接続ポイント
接点		
コイル		
変数		
定数	—	
ファンクション		
ファンクションブロック		

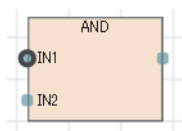
ファンクションの戻り値の名称は表示されません。

接続ポイントは接続されると非表示になります。

■入出力ポイントの反転

部品への入力または部品からの出力を、接続ポイントで反転できます。

反転状態の接続ポイントは黒丸で囲まれ、入力または出力するデータを反転(FALSE→TRUEまたはTRUE→FALSE)します。



反転可能なデータ型は、BOOL、WORD、DWORD、ANY_BIT、ANY_BOOLです。

定数

定数の表記方法

FBD/LDプログラムでの、文字列の表記方法を示します。

データ型		表記方法	表記例
文字列	STRING	文字列(ASCII, シフトJIS)をシングルクォーテーションマーク(')で囲む。	'ABC'
文字列[Unicode]	WSTRING	Unicode文字列をダブルクォーテーションマーク(")で囲む。	"ABC"

上記以外の定数の表記方法は、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

ラベルとデバイス

指定方法

FBD/LDプログラム上で、ラベルとデバイスを直接記述して使用できます。ラベルとデバイスは部品の入力ポイント、出力ポイント、汎用ファンクション/ファンクションブロックの引数、戻り値などに使用できます。

ラベルとデバイスの詳細は、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

■型指定付きデバイス表記

ワードデバイスは、デバイス名にデバイス型指定子をつけることで、任意のデータ型として使用できます。

FBD/LD言語でのデバイス型指定子と使用可能なデバイスは、ST言語と同じです。詳細は、下記を参照してください。

📖105ページ 型指定付きデバイス表記

ワードデバイスのデータ型を指定しない場合、デバイスの種類によってデータ型が決まります。

ワードデバイス	データ型
タイマデバイスの現在値(TN), 積算タイマデバイスの現在値(STN), カウンタデバイスの現在値(CN)	WORD
ロングタイマデバイスの現在値(LTN), ロング積算タイマデバイスの現在値(LSTN), ロングカウンタデバイスの現在値(LCN)	DWORD
ロングインデックスレジスタ(LZ)	DINT
上記以外	ANY16

注意事項

■ラベルを使用する場合

- 配列の添字には、ローカルデバイスは使用できません。ただし、添字に使用するデバイスを別のデバイスに代入し、代入先のデバイスを添字に指定すると使用できるようになります。

■デバイスを使用する場合のデータ型自動変換

ワードデバイスをワード[符号付き]以外のデータ型で使用する場合、デバイス型指定子を付与してください。(116ページ 型指定付きデバイス表記)

例

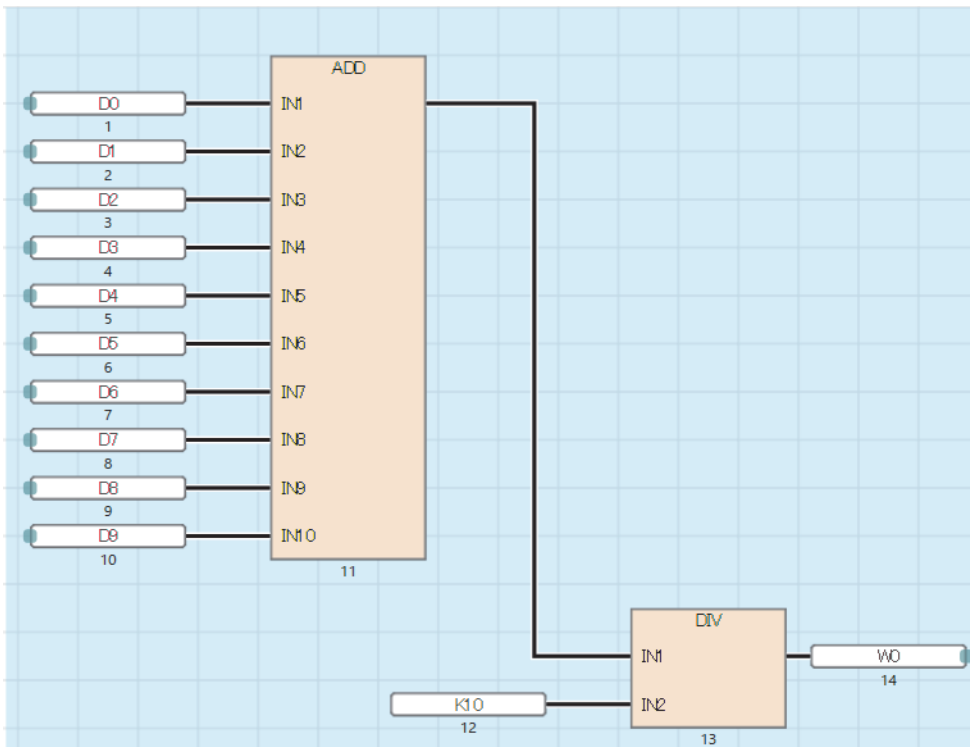
D2、D3の値をダブルワード[符号なし]のラベルdwordLabel1に転送する場合

デバイス型指定子を付与して、正しいデータ型で転送する例	意図しない転送結果となる例
D2:UD dwordLabel1 デバイス型指定子を付与したD2:UDはダブルワード[符号なし]のため、dwordLabel1にD2、D3の値が転送されます。	D2 dwordLabel1 デバイス型指定子のないD2はワード[符号付き]のため、データ型をダブルワード[符号なし]に自動変換してから、dwordLabel1に転送します。そのためD2の値だけ転送され、D3の値は転送しません。

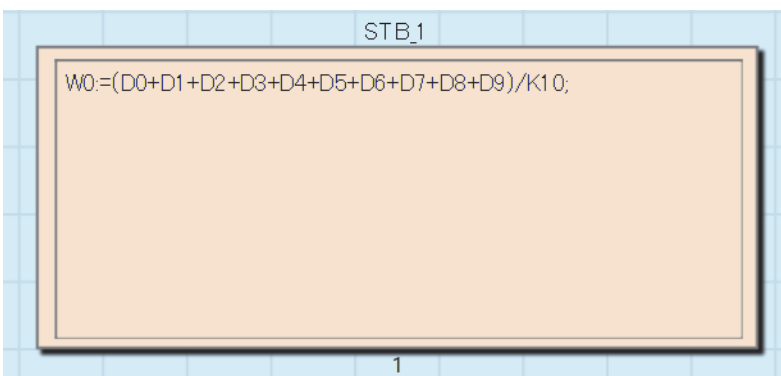
5.2 インラインST

インラインSTとは、FBD/LDエディタ内にSTプログラムを表示するインラインST部品を作成し、編集/モニタする機能です。これにより、FBD/LDプログラム内で、複雑な数値演算や文字列処理が簡単に作成できます。

- ・インラインSTを使用しない場合のプログラム



- ・インラインSTを使用した場合のプログラム



仕様

挿入したインラインST部品をダブルクリックすることで、STエディタを表示しSTプログラムを編集/モニタします。インラインSTに記述するSTプログラムの仕様については、ST言語の仕様を参照してください。

📖 91ページ ST言語

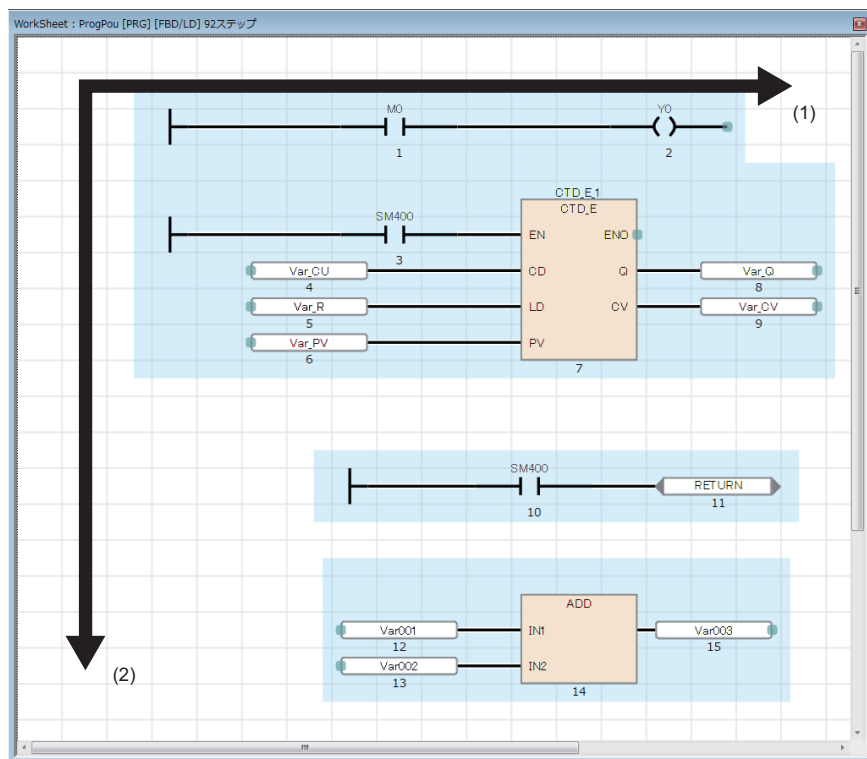
注意事項

- ・FBD/LDプログラムの一つのプログラム部品内に挿入できるインラインST部品の個数は最大64個です。
- ・インラインST内で「RETURN文」を使用すると、プログラムブロックの処理が終了するのではなく、インラインST部品内の処理が終了になります。
- ・FBD/LDプログラム中に挿入されたインラインST部品は、接続ポイントを持たないため、常に実行されます。

5.3 プログラム実行順序

部品の実行順序

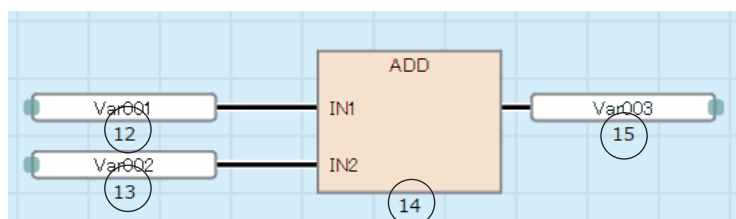
FBD/LDエディタ上の部品の実行順序は、部品の位置関係と接続状況によって決まります。



(1) 左から右に実行する。

(2) 上から下に実行する。

FBD/LDエディタ上に配置された各部品には、実行順序の番号が表示されます。プログラムを変換すると、確定した実行順序が表示されます。

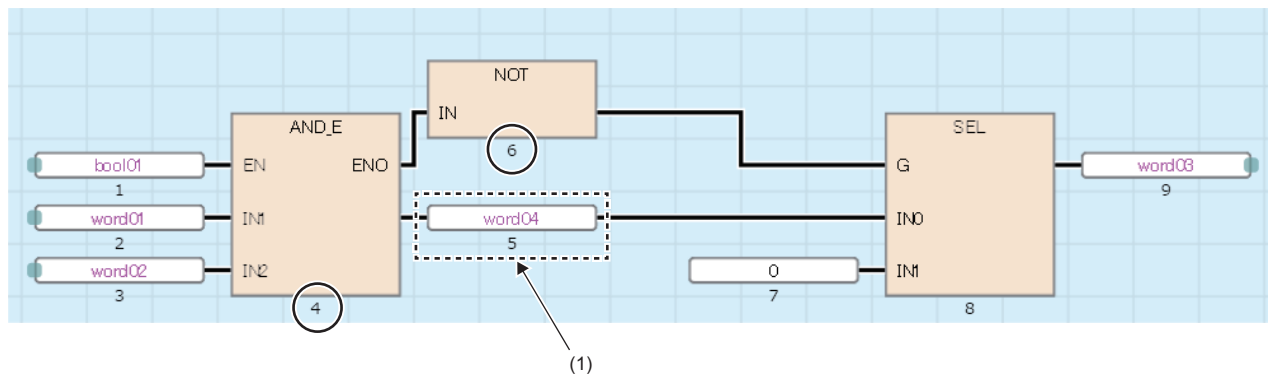


注意事項

ファンクションを使用したプログラムの場合、ファンクションの戻り値と別のファンクションの入力変数は直接接続せずに、間に変数部品を接続してください。

例

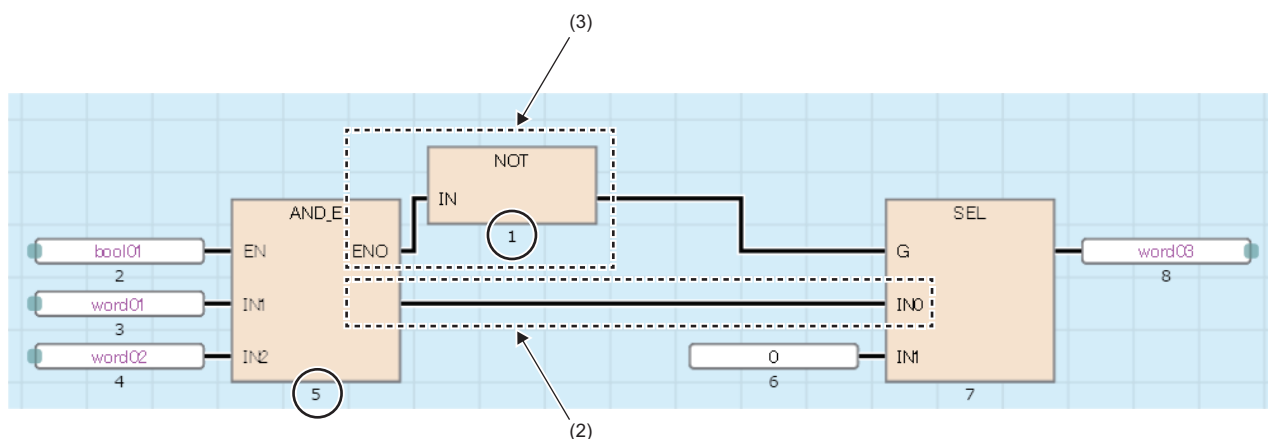
戻り値と入力変数の間に変数部品(1)を接続した場合



ファンクションの戻り値と入力変数を直接接続すると、意図した実行順序にならない場合があります。

例

意図した実行順序にならない場合



後方に配置した部品の入力に、前方に配置した部品の戻り値(2)と他の部品を経由した出力変数(3)を接続したため、実行順序が異なります。

第2部 命令/FUN/FB一覧

この部は下記の章構成となります。

6 命令一覧

7 ネットワーク命令一覧

8 汎用FUN/汎用FB一覧

9 モーション制御ファンクションブロック一覧

10 ネットワークファンクションブロック一覧

6 命令一覧

一覧表の見方を示します。

項目	内容
命令記号	命令名を示します。
処理内容	命令の概要を示します。
参照先	詳細説明の参照先を示します。

6.1 シーケンス命令

接点命令

■演算開始，直列接続，並列接続

命令記号	処理内容	参照先
LD	指定デバイスのON/OFF情報を演算結果とします。(a接点演算開始命令)	200ページ LD, LDI, AND, ANI, OR, ORI
LDI	指定デバイスのON/OFF情報を演算結果とします。(b接点演算開始命令)	
AND	指定デバイスのON/OFF情報と，それまでの演算結果のAND演算を演算結果とします。(a接点直列接続命令)	
ANI	指定デバイスのON/OFF情報と，それまでの演算結果のAND演算を演算結果とします。(b接点直列接続命令)	
OR	指定デバイスのON/OFF情報と，それまでの演算結果のOR演算を演算結果とします。(a接点1個の並列接続命令)	
ORI	指定デバイスのON/OFF情報と，それまでの演算結果のOR演算を演算結果とします。(b接点1個の並列接続命令)	

■パルス演算開始，パルス直列接続，パルス並列接続

命令記号	処理内容	参照先
LDP	指定ビットデバイスの立上り時(OFF→ON)にのみ導通します。(立上りパルス演算開始命令)	202ページ LDP, LDF, ANDP, ANDF, ORP, ORF
LDF	指定ビットデバイスの立下り時(ON→OFF)にのみ導通します。(立下りパルス演算開始命令)	
ANDP	それまでの演算結果とAND演算を行います。(立上りパルス直列接続命令)	
ANDF	それまでの演算結果とAND演算を行います。(立下りパルス直列接続命令)	
ORP	それまでの演算結果とOR演算を行います。(立上りパルス並列接続命令)	
ORF	それまでの演算結果とOR演算を行います。(立下りパルス並列接続命令)	

■パルス否定演算開始，パルス否定直列接続，パルス否定並列接続

命令記号	処理内容	参照先
LDPI	指定デバイスのOFF時，ON時，立下り時(ON→OFF)の場合に導通します。(立上りパルス否定演算開始命令)	205ページ LDPI, LDFI, ANDPI, ANDFI, ORPI, ORFI
LDFI	指定ビットデバイスの立上り時(OFF→ON)，OFF時，ON時に導通します。(立下りパルス否定演算開始命令)	
ANDPI	それまでの演算結果とAND演算を行います。(立上りパルス否定直列接続命令)	
ANDFI	それまでの演算結果とAND演算を行います。(立下りパルス否定直列接続命令)	
ORPI	それまでの演算結果とOR演算を行います。(立上りパルス否定並列接続命令)	
ORFI	それまでの演算結果とOR演算を行います。(立下りパルス否定並列接続命令)	

結合命令

■回路ブロック直列接続，並列接続

命令記号	処理内容	参照先
ANB	論理ブロック間のAND演算(論理ブロック間の直列接続)を行います。	208ページ ANB, ORB
ORB	論理ブロック間のOR演算(論理ブロック間の直列接続)を行います。	

■演算結果反転

命令記号	処理内容	参照先
INV	INV命令の直前までの演算結果を反転します。	209ページ INV

■演算結果パルス化

命令記号	処理内容	参照先
MEP	MEP命令までの演算結果の立上り時(OFF→ON)に，導通します。	210ページ MEP, MEF
MEF	MEF命令までの演算結果の立下り時(ON→OFF)に，導通します。	

■エッジリレー演算結果パルス化

命令記号	処理内容	参照先
EGP	EGP命令までの演算結果をエッジリレー (V)で記憶します。演算結果の立上り時(OFF→ON)に，導通します。	211ページ EGP, EGF
EGF	EGF命令までの演算結果をエッジリレー (V)で記憶します。演算結果の立下り時(ON→OFF)に，導通します。	

出力命令

■アウト(タイマ，カウンタ，アナンシェータを除く)

命令記号	処理内容	参照先
OUT	演算結果を指定されたデバイスへ出力します。	213ページ OUT

■タイマ，ロングタイマ

命令記号	処理内容	参照先
OUT T	OUT命令までの演算結果がONのとき，コイルがONしてタイマの計測を行います。タイムアップすると，a接点は導通し，b接点は非導通になります。 ・OUT T: 低速タイマ命令 ・OUTH T: 高速タイマ命令 ・OUT ST: 低速積算タイマ命令 ・OUTH ST: 高速積算タイマ命令 ・OUT LT: 低速ロングタイマ命令 ・OUT LST: 低速ロング積算タイマ命令	215ページ OUT T, OUTH T, OUT ST, OUTH ST
OUTH T		
OUT ST		
OUTH ST		
OUT LT		218ページ OUT LT, OUT LST
OUT LST		

■カウンタ，ロングカウンタ

命令記号	処理内容	参照先
OUT C	OUT命令までの演算結果がOFF→ONに変化したとき，カウンタの現在値(カウント値)を+1します。カウントアップすると，カウンタのa接点は導通し，b接点は非導通になります。 ・OUT C: カウンタ ・OUT LC: ロングカウンタ	221ページ OUT C
OUT LC		223ページ OUT LC

■アナンシェータ

命令記号	処理内容	参照先
OUT F	OUT F命令までの演算結果を指定のアナンシェータへ出力します。	225ページ OUT F

■デバイスのセット(アナンシェータを除く)

命令記号	処理内容	参照先
SET	指定ビットをONします。	226ページ SET

■デバイスのリセット(アナンシェータを除く)

命令記号	処理内容	参照先
RST	指定デバイスを0(OFF)にします。タイマ、カウンタについては、現在値を0、接点/コイルをOFFにします。	228ページ RST

■アナンシェータのセット, リセット

命令記号	処理内容	参照先
SET F	指定されたアナンシェータをONします。	230ページ SET F
RST F	指定されたアナンシェータをOFFします。	232ページ RST F

■立上り, 立下り出力

命令記号	処理内容	参照先
PLS	PLS指令のOFF→ON時に指定デバイスを1スキャンONさせます。	234ページ PLS
PLF	PLF指令のON→OFF時に指定デバイスを1スキャンONさせます。	236ページ PLF

■ビットデバイス出力反転

命令記号	処理内容	参照先
FF	指定されたデバイス状態を反転します。	238ページ FF

■ダイレクト出力のパルス化

命令記号	処理内容	参照先
DELTA	指定されたダイレクトアクセス出力(DY)をパルス出力します。	239ページ DELTA(P)
DELTAP		

シフト命令

■ビットデバイスシフト

命令記号	処理内容	参照先
SFT	1つ若いデバイスのON/OFF状態を指定したデバイスにシフトして、1つ若いデバイスをOFFします。	241ページ SFT(P)
SFTP		

マスタコントロール命令

■マスタコントロールのセット, リセット

命令記号	処理内容	参照先
MC	マスタコントロールを開始します。	243ページ MC, MCR
MCR	マスタコントロールを終了します。	

終了命令

■メインルーチンプログラム終了

命令記号	処理内容	参照先
FEND	メインルーチンプログラムと、サブルーチンプログラム、割込みプログラムを分割するとき使用します。	247ページ FEND

■シーケンスプログラム終了

命令記号	処理内容	参照先
END	プログラムの最終を示します。	248ページ END

停止命令

■シーケンスプログラム停止

命令記号	処理内容	参照先
STOP	コントローラの演算を停止します。(スイッチをSTOP側にした場合と同一です。)	250ページ STOP

無処理命令

■無処理

命令記号	処理内容	参照先
NOPLF	無処理の命令で、それまでの演算に何も影響を与えません。	251ページ NOPLF

6.2 基本命令

比較演算命令

■BIN16ビットデータ比較

命令記号	処理内容	参照先
LD=, AND=, OR=	指定した2つのBIN16ビットデータをa接点扱いで比較します。	252ページ LD□(_U), AND□(_U), OR□(_U)
LD=_U, AND=_U, OR=_U		
LD<>, AND<>, OR<>		
LD<>_U, AND<>_U, OR<>_U		
LD>, AND>, OR>		
LD>_U, AND>_U, OR>_U		
LD<=, AND<=, OR<=		
LD<=_U, AND<=_U, OR<=_U		
LD<, AND<, OR<		
LD<_U, AND<_U, OR<_U		
LD>=, AND>=, OR>=		
LD>=_U, AND>=_U, OR>=_U		

■BIN32ビットデータ比較

命令記号	処理内容	参照先
LDD=, ANDD=, ORD=	指定した2つのBIN32ビットデータをa接点扱いで比較します。	254ページ LDD□(_U), ANDD□(_U), ORD□(_U)
LDD=_U, ANDD=_U, ORD=_U		
LDD<>, ANDD<>, ORD<>		
LDD<>_U, ANDD<>_U, ORD<>_U		
LDD>, ANDD>, ORD>		
LDD>_U, ANDD>_U, ORD>_U		
LDD<=, ANDD<=, ORD<=		
LDD<=_U, ANDD<=_U, ORD<=_U		
LDD<, ANDD<, ORD<		
LDD<_U, ANDD<_U, ORD<_U		
LDD>=, ANDD>=, ORD>=		
LDD>=_U, ANDD>=_U, ORD>=_U		

■BIN16ビットブロックデータ比較

命令記号	処理内容	参照先
BKCMPE, BKCMPE<>, BKCMPE>, BKCMPE<=, BKCMPE<, BKCMPE>=	指定した2つのBIN16ビットブロックデータを比較します。	256ページ BKCMPE□(P)(_U)
BKCMPE=P, BKCMPE<>P, BKCMPE>P, BKCMPE<=P, BKCMPE<P, BKCMPE>=P		
BKCMPE=_U, BKCMPE<>_U, BKCMPE>_U, BKCMPE<=_U, BKCMPE<_U, BKCMPE>=_U		
BKCMPE=P_U, BKCMPE<>P_U, BKCMPE>P_U, BKCMPE<=P_U, BKCMPE<P_U, BKCMPE>=P_U		

■BIN32ビットブロックデータ比較

命令記号	処理内容	参照先
DBKCMP=, DBKCMP<>, DBKCMP>, DBKCMP<=, DBKCMP<, DBKCMP>=	指定した2つのBIN32ビットブロックデータを比較します。	259ページ DBKCMP□(P)(_U)
DBKCMP=P, DBKCMP<>P, DBKCMP>P, DBKCMP<=P, DBKCMP<P, DBKCMP>=P		
DBKCMP=_U, DBKCMP<>_U, DBKCMP>_U, DBKCMP<=_U, DBKCMP<_U, DBKCMP>=_U		
DBKCMP=P_U, DBKCMP<>P_U, DBKCMP>P_U, DBKCMP<=P_U, DBKCMP<P_U, DBKCMP>=P_U		

算術演算命令

■BIN16ビット加減算

命令記号	処理内容	参照先
+	指定した2つのBIN16ビットデータを加算します。(オペランドが2個の場合)	262ページ +(P)(_U) [オペランドが2個の場合]
+P		
+_U		
+P_U		
+	指定した2つのBIN16ビットデータを加算します。(オペランドが3個の場合)	264ページ +(P)(_U) [オペランドが3個の場合]
+P		
+_U		
+P_U		
-	指定した2つのBIN16ビットデータを減算します。(オペランドが2個の場合)	266ページ -(P)(_U) [オペランドが2個の場合]
-P		
-_U		
-P_U		
-	指定した2つのBIN16ビットデータを減算します。(オペランドが3個の場合)	268ページ -(P)(_U) [オペランドが3個の場合]
-P		
-_U		
-P_U		

■BIN32ビット加減算

命令記号	処理内容	参照先
D+	指定した2つのBIN32ビットデータを加算します。(オペランドが2個の場合)	270ページ D+(P)(<u>L</u> U) [オペランドが2個の場合]
D+P		
D+_ <u>L</u> U		
D+P_ <u>L</u> U		
D+	指定した2つのBIN32ビットデータを加算します。(オペランドが3個の場合)	272ページ D+(P)(<u>L</u> U) [オペランドが3個の場合]
D+P		
D+_ <u>L</u> U		
D+P_ <u>L</u> U		
D-	指定した2つのBIN32ビットデータを減算します。(オペランドが2個の場合)	274ページ D-(P)(<u>L</u> U) [オペランドが2個の場合]
D-P		
D-_ <u>L</u> U		
D-P_ <u>L</u> U		
D-	指定した2つのBIN32ビットデータを減算します。(オペランドが3個の場合)	276ページ D-(P)(<u>L</u> U) [オペランドが3個の場合]
D-P		
D-_ <u>L</u> U		
D-P_ <u>L</u> U		

■BIN16ビット乗除算

命令記号	処理内容	参照先
*	指定した2つのBIN16ビットデータを乗算します。	278ページ *(P)(<u>L</u> U)
*P		
*_ <u>L</u> U		
*P_ <u>L</u> U		
/	指定した2つのBIN16ビットデータを除算します。	280ページ /(P)(<u>L</u> U)
/P		
/ <u>L</u> U		
/P_ <u>L</u> U		

■BIN32ビット乗除算

命令記号	処理内容	参照先
D*	指定した2つのBIN32ビットデータを乗算します。	282ページ D*(P)(<u>L</u> U)
D*P		
D*_ <u>L</u> U		
D*P_ <u>L</u> U		
D/	指定した2つのBIN32ビットデータを除算します。	284ページ D/(P)(<u>L</u> U)
D/P		
D/ <u>L</u> U		
D/P_ <u>L</u> U		

■BCD4桁加減算

命令記号	処理内容	参照先
B+	指定した2つのBCD4桁データを加算します。(オペランドが2個の場合)	286ページ B+(P) [オペランドが2個の場合]
B+P		
B+	指定した2つのBCD4桁データを加算します。(オペランドが3個の場合)	287ページ B+(P) [オペランドが3個の場合]
B+P		
B-	指定した2つのBCD4桁データを減算します。(オペランドが2個の場合)	289ページ B-(P) [オペランドが2個の場合]
B-P		
B-	指定した2つのBCD4桁データを減算します。(オペランドが3個の場合)	290ページ B-(P) [オペランドが3個の場合]
B-P		

■BCD8桁加減算

命令記号	処理内容	参照先
DB+	指定した2つのBCD8桁データを加算します。(オペランドが2個の場合)	292ページ DB+(P) [オペランドが2個の場合]
DB+P		
DB+	指定した2つのBCD8桁データを加算します。(オペランドが3個の場合)	293ページ DB+(P) [オペランドが3個の場合]
DB+P		
DB-	指定した2つのBCD8桁データを減算します。(オペランドが2個の場合)	295ページ DB-(P) [オペランドが2個の場合]
DB-P		
DB-	指定した2つのBCD8桁データを減算します。(オペランドが3個の場合)	296ページ DB-(P) [オペランドが3個の場合]
DB-P		

■BCD4桁乗除算

命令記号	処理内容	参照先
B*	指定した2つのBCD4桁データを乗算します。	298ページ B*(P)
B*P		
B/	指定した2つのBCD4桁データを除算します。	300ページ B/(P)
B/P		

■BCD8桁乗除算

命令記号	処理内容	参照先
DB*	指定した2つのBCD8桁データを乗算します。	302ページ DB*(P)
DB*P		
DB/	指定した2つのBCD8桁データを除算します。	304ページ DB/(P)
DB/P		

■BIN16ビットブロックデータ加減算

命令記号	処理内容	参照先
BK+	指定した2つのBIN16ビットデータブロックを加算します。	306ページ BK+(P)(U)
BK+P		
BK+_U		
BK+P_U		
BK-	指定した2つのBIN16ビットデータブロックを減算します。	308ページ BK-(P)(U)
BK-P		
BK-_U		
BK-P_U		

■BIN32ビットブロックデータ加減算

命令記号	処理内容	参照先
DBK+	指定した2つのBIN32ビットデータブロックを加算します。	310ページ DBK+(P)(U)
DBK+P		
DBK+_U		
DBK+P_U		
DBK-	指定した2つのBIN32ビットデータブロックを減算します。	313ページ DBK-(P)(U)
DBK-P		
DBK-_U		
DBK-P_U		

■16ビットBINデータインクリメント、デクリメント

命令記号	処理内容	参照先
INC	指定したBIN16ビットデータを+1します。	316ページ INC(P)(U)
INCP		
INC_U		
INCP_U		
DEC	指定したBIN16ビットデータを-1します。	317ページ DEC(P)(U)
DECP		
DEC_U		
DECP_U		

■32ビットBINデータインクリメント、デクリメント

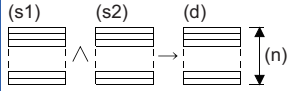
命令記号	処理内容	参照先
DINC	指定したBIN32ビットデータを+1します。	318ページ DINC(P)(U)
DINCP		
DINC_U		
DINCP_U		
DDEC	指定したBIN32ビットデータを-1します。	319ページ DDEC(P)(U)
DDECP		
DDEC_U		
DDECP_U		

論理演算命令

■16ビット/32ビットデータ論理積

命令記号	処理内容	参照先
WAND	指定した2つのBIN16ビットデータの論理積を行います。(オペランドが2個の場合)	321ページ WAND(P) [オペランドが2個の場合]
WANDP		
WAND	指定した2つのBIN16ビットデータの論理積を行います。(オペランドが3個の場合)	322ページ WAND(P) [オペランドが3個の場合]
WANDP		
DAND	指定した2つのBIN32ビットデータの論理積を行います。(オペランドが2個の場合)	324ページ DAND(P) [オペランドが2個の場合]
DANDP		
DAND	指定した2つのBIN32ビットデータの論理積を行います。(オペランドが3個の場合)	326ページ DAND(P) [オペランドが3個の場合]
DANDP		

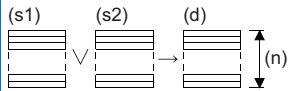
■16ビットブロックデータ論理積

命令記号	処理内容	参照先
BKAND	指定した2つのBIN16ビットデータブロックの論理積を行います。 	328ページ BKAND(P)
BKANDP		

■16ビット/32ビットデータ論理和

命令記号	処理内容	参照先
WOR	指定した2つのBIN16ビットデータの論理和を行います。(オペランドが2個の場合)	330ページ WOR(P) [オペランドが2個の場合]
WORP		
WOR	指定した2つのBIN16ビットデータの論理和を行います。(オペランドが3個の場合)	331ページ WOR(P) [オペランドが3個の場合]
WORP		
DOR	指定した2つのBIN32ビットデータの論理和を行います。(オペランドが2個の場合)	333ページ DOR(P) [オペランドが2個の場合]
DORP		
DOR	指定した2つのBIN32ビットデータの論理和を行います。(オペランドが3個の場合)	335ページ DOR(P) [オペランドが3個の場合]
DORP		

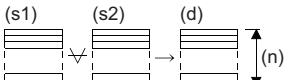
■16ビットブロックデータ論理和

命令記号	処理内容	参照先
BKOR	指定した2つのBIN16ビットデータブロックの論理和を行います。 	337ページ BKOR(P)
BKORP		

■16ビット/32ビットデータ排他的論理和

命令記号	処理内容	参照先
WXOR	指定した2つのBIN16ビットデータの排他的論理和を行います。(オペランドが2個の場合)	339ページ WXOR(P) [オペランドが2個の場合]
WXORP		
WXOR	指定した2つのBIN16ビットデータの排他的論理和を行います。(オペランドが3個の場合)	340ページ WXOR(P) [オペランドが3個の場合]
WXORP		
DXOR	指定した2つのBIN32ビットデータの排他的論理和を行います。(オペランドが2個の場合)	342ページ DXOR(P) [オペランドが2個の場合]
DXORP		
DXOR	指定した2つのBIN32ビットデータの排他的論理和を行います。(オペランドが3個の場合)	344ページ DXOR(P) [オペランドが3個の場合]
DXORP		

■16ビットブロックデータ排他的論理和

命令記号	処理内容	参照先
BKXOR	指定した2つのBIN16ビットデータブロックの排他的論理和を行います。	346ページ BKXOR(P)
BKXORP		

■16ビット/32ビットデータ否定排他的論理和

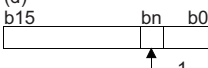
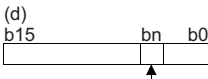
命令記号	処理内容	参照先
WXNR	指定した2つのBIN16ビットデータの否定排他的論理和を行います。(オペランドが2個の場合)	348ページ WXNR(P) [オペランドが2個の場合]
WXNRP		
WXNR	指定した2つのBIN16ビットデータの否定排他的論理和を行います。(オペランドが3個の場合)	349ページ WXNR(P) [オペランドが3個の場合]
WXNRP		
DXNR	指定した2つのBIN32ビットデータの否定排他的論理和を行います。(オペランドが2個の場合)	351ページ DXNR(P) [オペランドが2個の場合]
DXNRP		
DXNR	指定した2つのBIN32ビットデータの否定排他的論理和を行います。(オペランドが3個の場合)	353ページ DXNR(P) [オペランドが3個の場合]
DXNRP		

■16ビットブロックデータ否定排他的論理和

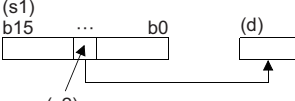
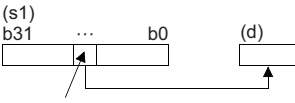
命令記号	処理内容	参照先
BKXNR	指定した2つのBIN16ビットデータブロックの否定排他的論理和を行います。	355ページ BKXNR(P)
BKXNRP		

ビット処理命令

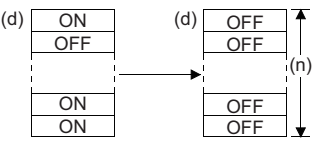
■ワードデバイスのビットセット/リセット

命令記号	処理内容	参照先
BSET	指定したワードデバイス内のnビット目をセット(1)します。	357ページ BSET(P)
BSETP		
BRST	指定したワードデバイス内のnビット目をリセット(0)します。	359ページ BRST(P)
BRSTP		

■ビットテスト

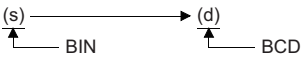
命令記号	処理内容	参照先
TEST	指定したワードデバイス内のnビット目を取り出します。	361ページ TEST(P)
TESTP		
DTEST	指定した2ワードデバイス内のnビット目を取り出します。	363ページ DTEST(P)
DTESTP		

■ビットデバイスの一括リセット

命令記号	処理内容	参照先
BKRST	指定したビットデバイスからn点分のビットデバイスをリセットします。	365ページ BKRST(P)
BKRSTP		

データ変換命令

■BINデータ→BCD4桁/8桁変換

命令記号	処理内容	参照先
BCD	指定したBIN16ビットデータ(0~9999)をBCD4桁データに変換します。	387ページ BCD(P)
BCDP		
DBCD	指定したBIN32ビットデータ(0~99999999)をBCD8桁データに変換します。	389ページ DBCD(P)
DBCDP		

■BCD4桁/8桁→BIN16ビット/32ビットデータ変換

命令記号	処理内容	参照先
BIN	指定したBCD4桁データ(0~9999)をBIN16ビットデータに変換します。	391ページ BIN(P)
BINP		
DBIN	指定したBCD8桁データ(0~99999999)をBIN32ビットデータに変換します。	393ページ DBIN(P)
DBINP		

■単精度実数→符号付きBIN16ビット/32ビットデータ

命令記号	処理内容	参照先
FLT2INT	指定した単精度実数(-32768~32767)を符号付きBIN16ビットデータに変換します。	395ページ FLT2INT(P)
FLT2INTP	 (1): 実数	
FLT2DINT	指定した単精度実数(-2147483648~2147483647)を符号付きBIN32ビットデータに変換します。	399ページ FLT2DINT(P)
FLT2DINTP	 (1): 実数	

■単精度実数→符号なしBIN16ビット/32ビットデータ

命令記号	処理内容	参照先
FLT2UINT FLT2UINTP	指定した単精度実数(0~65535)を符号なしBIN16ビットデータに変換します。 $\begin{array}{ccc} (s)+1, (s) & \longrightarrow & (d) \\ \uparrow & & \uparrow \\ & (1) & \text{BIN} \end{array}$ (1): 実数	397ページ FLT2UINT(P)
FLT2UDINT FLT2UDINTP	指定した単精度実数(0~4294967295)を符号なしBIN32ビットデータに変換する。 $\begin{array}{ccc} (s)+1, (s) & \longrightarrow & (d)+1, (d) \\ \uparrow & & \uparrow \\ & (1) & \text{BIN} \end{array}$ (1): 実数	401ページ FLT2UDINT(P)

■倍精度実数→符号付きBIN16ビット/32ビットデータ

命令記号	処理内容	参照先
DBL2INT DBL2INTP	指定した倍精度実数(-32768~32767)を符号付きBIN16ビットデータに変換します。 $\begin{array}{ccc} (s)+3, (s)+2, (s)+1, (s) & \longrightarrow & (d) \\ \uparrow & & \uparrow \\ & (1) & \text{BIN} \end{array}$ (1): 実数	403ページ DBL2INT(P)
DBL2DINT DBL2DINTP	指定した倍精度実数(-2147483648~2147483647)を符号付きBIN32ビットデータに変換します。 $\begin{array}{ccc} (s)+3, (s)+2, (s)+1, (s) & \longrightarrow & (d)+1, (d) \\ \uparrow & & \uparrow \\ & (1) & \text{BIN} \end{array}$ (1): 実数	407ページ DBL2DINT(P)

■倍精度実数→符号なしBIN16ビット/32ビットデータ

命令記号	処理内容	参照先
DBL2UINT DBL2UINTP	指定した倍精度実数(0~65535)を符号なしBIN16ビットデータに変換します。 $\begin{array}{ccc} (s)+3, (s)+2, (s)+1, (s) & \longrightarrow & (d) \\ \uparrow & & \uparrow \\ & (1) & \text{BIN} \end{array}$ (1): 実数	405ページ DBL2UINT(P)
DBL2UDINT DBL2UDINTP	指定した倍精度実数(0~4294967295)を符号なしBIN32ビットデータに変換します。 $\begin{array}{ccc} (s)+3, (s)+2, (s)+1, (s) & \longrightarrow & (d)+1, (d) \\ \uparrow & & \uparrow \\ & (1) & \text{BIN} \end{array}$ (1): 実数	409ページ DBL2UDINT(P)

■符号付きBIN16ビットデータ→符号なしBIN16ビット/32ビットデータ変換

命令記号	処理内容	参照先
INT2UINT INT2UINTP	(s)で指定した符号付きBIN16ビットデータを、符号なしBIN16ビットデータに変換し、(d)へ格納します。	411ページ INT2UINT(P)
INT2UDINT INT2UDINTP	(s)で指定した符号付きBIN16ビットデータを、符号なしBIN32ビットデータに変換し、(d)へ格納します。	415ページ INT2UDINT(P)

■符号付きBIN16ビットデータ→符号付きBIN32ビットデータ変換

命令記号	処理内容	参照先
INT2DINT INT2DINTP	(s)で指定した符号付きBIN16ビットデータを、符号付きBIN32ビットデータに変換し、(d)へ格納します。	413ページ INT2DINT(P)

■符号なしBIN16ビットデータ→符号付きBIN16ビット/32ビットデータ変換

命令記号	処理内容	参照先
UINT2INT	(s)で指定した符号なしBIN16ビットデータを、符号付きBIN16ビットデータに変換し、(d)へ格納します。	417ページ UINT2INT(P)
UINT2INTP		
UINT2DINT	(s)で指定した符号なしBIN16ビットデータを、符号付きBIN32ビットデータに変換し、(d)へ格納します。	419ページ UINT2DINT(P)
UINT2DINTP		

■符号なしBIN16ビットデータ→符号なしBIN32ビットデータ変換

命令記号	処理内容	参照先
UINT2UDINT	(s)で指定した符号なしBIN16ビットデータを、符号なしBIN32ビットデータに変換し、(d)へ格納します。	421ページ UINT2UDINT(P)
UINT2UDINTP		

■符号付きBIN32ビットデータ→符号付きBIN16ビットデータ変換

命令記号	処理内容	参照先
DINT2INT	(s)で指定した符号付きBIN32ビットデータを、符号付きBIN16ビットデータに変換し、(d)へ格納します。	423ページ DINT2INT(P)
DINT2INTP		

■符号付きBIN32ビットデータ→符号なしBIN16ビット/32ビットデータ変換

命令記号	処理内容	参照先
DINT2UINT	(s)で指定した符号付きBIN32ビットデータを、符号なしBIN16ビットデータに変換し、(d)へ格納します。	425ページ DINT2UINT(P)
DINT2UINTP		
DINT2UDINT	(s)で指定した符号付きBIN32ビットデータを、符号なしBIN32ビットデータに変換し、(d)へ格納します。	427ページ DINT2UDINT(P)
DINT2UDINTP		

■符号なしBIN32ビットデータ→符号付きBIN16ビット/32ビットデータ変換

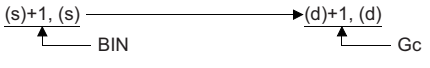
命令記号	処理内容	参照先
UDINT2INT	(s)で指定した符号なしBIN32ビットデータを、符号付きBIN16ビットデータに変換し、(d)へ格納します。	429ページ UDINT2INT(P)
UDINT2INTP		
UDINT2DINT	(s)で指定した符号なしBIN32ビットデータを、符号付きBIN32ビットデータに変換し、(d)へ格納します。	433ページ UDINT2DINT(P)
UDINT2DINTP		

■符号なしBIN32ビットデータ→符号なしBIN16ビットデータ変換

命令記号	処理内容	参照先
UDINT2UINT	(s)で指定した符号なしBIN32ビットデータを、符号なしBIN16ビットデータに変換し、(d)へ格納します。	431ページ UDINT2UINT(P)
UDINT2UINTP		

■BIN16ビット/32ビットデータ→BIN16ビット/32ビットグレイコードデータ変換

命令記号	処理内容	参照先
GRY	指定したBIN16ビットデータ(-32768~32767)をBIN16ビットグレイコードデータに変換します。	435ページ GRY(P)(_U)
GRYP	(s) → (d) ↑ ↑ BIN Gc Gc: グレイコード	
GRY_U	指定したBIN16ビットデータ(0~65535)をBIN16ビットグレイコードデータに変換します。	
GRYP_U	(s) → (d) ↑ ↑ BIN Gc Gc: グレイコード	

命令記号	処理内容	参照先
DGRY	指定したBIN32ビットデータ(-2147483648~2147483647)をBIN32ビットグレイコードデータに変換します。	437ページ DGRY(P)(_U)
DGRYP		
	 Gc: グレイコード	
DGRY_U	指定したBIN32ビットデータ(0~4294967295)をBIN32ビットグレイコードデータに変換します。	
DGRYP_U		
	 Gc: グレイコード	

■BIN16ビット/32ビットグレイコードデータ→BIN16ビット/32ビットデータ変換

命令記号	処理内容	参照先
GBIN GBINP	指定したBIN16ビットグレイコードデータ(-32768~32767)をBIN16ビットデータに変換します。 Gc: グレイコード	439ページ GBIN(P)(_U)
GBIN_U GBINP_U	指定したBIN16ビットグレイコードデータ(0~65535)をBIN16ビットデータに変換します。 Gc: グレイコード	
DGBIN DGBINP	指定したBIN32ビットグレイコードデータ(-2147483648~2147483647)をBIN32ビットデータに変換します。 Gc: グレイコード	441ページ DGBIN(P)(_U)
DGBIN_U DGBINP_U	指定したBIN32ビットグレイコードデータ(0~4294967295)をBIN32ビットデータに変換します。 Gc: グレイコード	

■BIN16ビットデータブロック→BCD4桁データ変換ブロック

命令記号	処理内容	参照先
BKBCD BKBCDP	(s)から(n)点分のBINデータを一括でBCDデータに変換し、(d)以降に格納します。	443ページ BKBCD(P)

■ブロックBCD4桁データ→ブロックBIN16ビット変換データ

命令記号	処理内容	参照先
BKBIN BKBINP	(s)から(n)点分のBCDデータを一括でBINデータに変換し、(d)以降に格納します。	445ページ BKBIN(P)

■10進アスキーデータ→BIN16ビット/32ビットデータ変換

命令記号	処理内容	参照先
DABIN DABINP DABIN_U DABINP_U	(s)で指定した10進アスキー値を1ワードBIN値に変換し、(d)で指定したワードデバイス番号に格納します。	447ページ DABIN(P)(_U)
DDABIN DDABINP DDABIN_U DDABINP_U	(s)で指定した10進アスキー値を2ワードBIN値に変換し、(d)で指定したワードデバイス番号に格納します。	450ページ DDABIN(P)(_U)

■16進アスキーデータ→BIN16ビット/32ビットデータ変換

命令記号	処理内容	参照先
HABIN HABINP	(s)で指定した16進アスキー値を1ワードBIN値に変換し、(d)で指定したワードデバイス番号に格納します。	454ページ HABIN(P)
DHABIN DHABINP	(s)で指定した16進アスキー値を2ワードBIN値に変換し、(d)で指定したワードデバイス番号に格納します。	456ページ DHABIN(P)

■10進アスキーデータ→BCD4桁/8桁変換

命令記号	処理内容	参照先
DABCD	(s)で指定した10進アスキー値を1ワードBCD値に変換し、(d)で指定したワードデバイス番号に格納します。	459ページ DABCD(P)
DABCDP		
DDABCD	(s)で指定した10進アスキー値を2ワードBCD値に変換し、(d)で指定したワードデバイス番号に格納します。	462ページ DDABCD(P)
DDABCDP		

■10進文字列→BIN16ビット/32ビットデータ変換

命令記号	処理内容	参照先
VAL	(s)で指定した小数点を含んだ文字列を1ワードBIN値と小数部桁数に変換し、(d1), (d2)で指定したデバイスに格納します。	465ページ VAL(P)(_U)
VALP		
VAL_U		
VALP_U		
DVAL	(s)で指定した小数点を含んだ文字列を2ワードBIN値と小数部桁数に変換し、(d1), (d2)で指定したデバイスに格納します。	468ページ DVAL(P)(_U)
DVALP		
DVAL_U		
DVALP_U		

■16進アスキー →16進データBIN変換

命令記号	処理内容	参照先
ASC2INT	(s)で指定したワードデバイス以降の16進アスキーデータを、(n)で指定した文字数分だけBIN値に変換して、(d)に指定したデバイス番号以降に格納します。	471ページ ASC2INT(P)
ASC2INTP		

■単精度実数→BCD分解

命令記号	処理内容	参照先
EMOD	(s1)の32ビット浮動小数点データを(s2)で指定した小数部桁数のBCDに変換して、(d)で指定したデバイスに格納します。	473ページ EMOD(P)
EMODP		

■BIN16ビット/32ビットデータ2の補数(符号反転)

命令記号	処理内容	参照先
NEG	BIN16ビットデバイスの符号を反転します。	475ページ NEG(P)
NEGP		
DNEG	BIN32ビットデバイスの符号を反転します。	476ページ DNEG(P)
DNEGP		

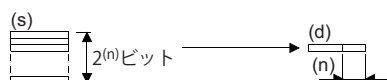
■8→256ビットデコード

命令記号	処理内容	参照先
DECO	指定されたデバイスの下位(n)ビットをデコードします。	477ページ DECO(P)
DECOP		



■256→8ビットエンコード

命令記号	処理内容	参照先
ENCO	2の(n)乗ビットのデータをエンコードします。	479ページ ENCO(P)
ENCOP		



■7セグメントデコード

命令記号	処理内容	参照先
SEG	デバイスの下位4ビットで指定された0~Fのデータを7セグメント表示データにデコードします。 	481ページ SEG(P)
SEGP		

■BIN16ビットデータの4ビット分離

命令記号	処理内容	参照先
DIS	(s)で指定の16ビットデータを4ビット単位で分離し、(d)から(n)点分の下位4ビットに格納します。 (n<4)	484ページ DIS(P)
DISP		

■BIN16ビットデータの4ビット結合

命令記号	処理内容	参照先
UNI	(s)で指定のデバイスから(n)点分の下位4ビットデータを結合し、(d)で指定のデバイスに格納します。 (n<4)	486ページ UNI(P)
UNIP		

■任意データのビット分離，結合

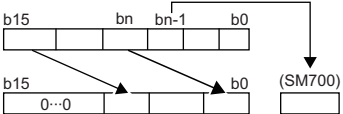
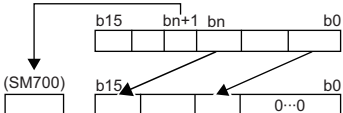
命令記号	処理内容	参照先
NDIS	(s1)で指定のデバイス以降のデータを(s2)以降で指定のビットに分離し、(d)で指定のデバイスから順に格納します。	488ページ NDIS(P)
NDISP		
NUNI	(s1)で指定のデバイス以降のデータを(s2)以降で指定のビットごとに結合し、(d)で指定のデバイスから順に格納します。	490ページ NUNI(P)
NUNIP		

■バイト単位データ分離，結合

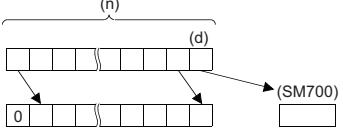
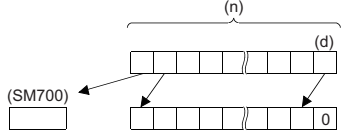
命令記号	処理内容	参照先
WTOB	(s)で指定のデバイスから、(n)点分の16ビットデータを8ビット単位に分解し、(d)で指定のデバイスから順に格納します。	492ページ WTOB(P)
WTOBP		
BTOW	(s)で指定のデバイスから、(n)点分の16ビットデータの下位8ビットを16ビットに結合し、(d)で指定のデバイスから順に格納します。	494ページ BTOW(P)
BTOWP		

データシフト命令

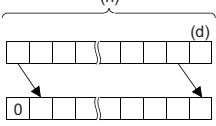
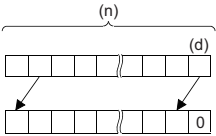
■BIN16ビットデータのnビット右シフト, 左シフト

命令記号	処理内容	参照先
SFR SFRP	指定したデバイスのBIN16ビットデータを右シフトさせます。シフト後の空きデータには0が格納されます。 	367ページ SFR(P)
SFL SFLP	指定したデバイスのBIN16ビットデータを左シフトさせます。シフト後の空きデータには0が格納されます。 	369ページ SFL(P)

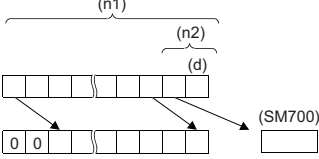
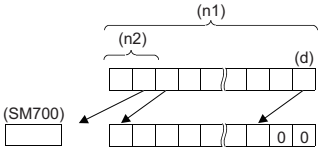
■nビットデータの1ビット右シフト, 左シフト

命令記号	処理内容	参照先
BSFR BSFRP	指定したデバイスからn点分を右へ1ビットシフトさせます。シフト後の空きデータには0が格納されます。 	371ページ BSFR(P)
BSFL BSFLP	指定したデバイスからn点分を左へ1ビットシフトさせます。シフト後の空きデータには0が格納されます。 	373ページ BSFL(P)

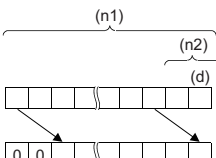
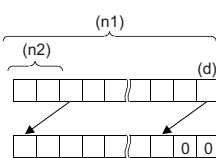
■nワードデータの1ワード右シフト, 左シフト

命令記号	処理内容	参照先
DSFR DSFRP	指定したデバイスからn点分を右へ1ワードシフトさせます。シフト後の空きデータには0が格納されます。 	375ページ DSFR(P)
DSFL DSFLP	指定したデバイスからn点分を左へ1ワードシフトさせます。シフト後の空きデータには0が格納されます。 	377ページ DSFL(P)

■nビットデータのnビット右シフト, 左シフト

命令記号	処理内容	参照先
SFTBR SFTBRP	指定したデバイスからnビット分のデータを右へnビットシフトさせます。シフト後の空きデータには0が格納されます。 	379ページ SFTBR(P)
SFTBL SFTBLP	指定したデバイスからnビット分のデータを左へnビットシフトさせます。シフト後の空きデータには0が格納されます。 	381ページ SFTBL(P)

■nワードデータのnワード右シフト, 左シフト

命令記号	処理内容	参照先
SFTWR SFTWRP	指定したデバイスからnワード分のデータを右へnワードシフトさせます。シフト後の空きデータには0が格納されます。 	383ページ SFTWR(P)
SFTWL SFTWLP	指定したデバイスからnワード分のデータを左へnワードシフトさせます。シフト後の空きデータには0が格納されます。 	385ページ SFTWL(P)

データ転送命令

■BIN16ビット/32ビットデータ転送

命令記号	処理内容	参照先
MOV	指定されたデバイスのBIN16ビットデータを転送します。	496ページ MOV(P)
MOVP	(s) → (d)	
DMOV	指定されたデバイスのBIN32ビットデータを転送します。	498ページ DMOV(P)
DMOVP	(s)+1, (s) → (d)+1, (d)	

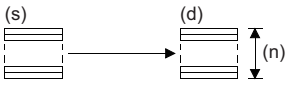
■BIN16ビット/32ビットデータ否定転送

命令記号	処理内容	参照先
CML	指定されたBIN16ビットデータをビットごとに反転し、転送します。	500ページ CML(P)
CMLP	(s) → (d)	
DCML	指定されたBIN32ビットデータをビットごとに反転し、転送します。	502ページ DCML(P)
DCMLP	(s)+1, (s) → (d)+1, (d)	

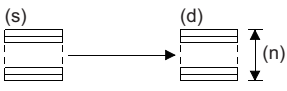
■1ビットデータ反転転送

命令記号	処理内容	参照先
CMLB	(s)で指定したビットデータを反転し、(d)へ格納します。	504ページ CMLB(P)
CMLBP		

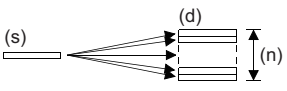
■BIN16ビットデータブロック転送(16ビット)

命令記号	処理内容	参照先
BMOV	指定されたデバイスから(n)点のBIN16ビットデータを一括転送します。	506ページ BMOV(P)
BMOVP	 (n)=1~65535	

■BIN16ビットデータブロック転送(32ビット)

命令記号	処理内容	参照先
BMOVL	指定されたデバイスから(n)点のBIN16ビットデータを一括転送します。	508ページ BMOVL(P)
BMOVLP	 (n)=1~4294967295	

■同一BIN16ビットデータブロック転送(16ビット)

命令記号	処理内容	参照先
FMOV	BIN16ビットデータを指定されたデバイスから(n)点に転送します。	510ページ FMOV(P)
FMOVP	 (n)=1~65535	

■同一BIN16ビットデータブロック転送(32ビット)

命令記号	処理内容	参照先
FMOVL	BIN16ビットデータを指定されたデバイスから(n)点に転送します。 (n)=1~4294967295	512ページ FMOVL(P)
FMOVLP		

■同一BIN32ビットデータブロック転送(16ビット)

命令記号	処理内容	参照先
DFMOV	BIN32ビットデータを、指定したデバイスから(n)点に転送します。 (n)=1~65535	514ページ DFMOV(P)
DFMOVLP		

■同一BIN32ビットデータブロック転送(32ビット)

命令記号	処理内容	参照先
DFMOVL	BIN32ビットデータを、指定したデバイスから(n)点に転送します。 (n)=1~4294967295	516ページ DFMOVL(P)
DFMOVLP		

■BIN16ビット/32ビットデータ交換

命令記号	処理内容	参照先
XCH	指定されたBIN16ビットデータを交換します。 (d1) ↔ (d2)	518ページ XCH(P)
XCHP		
DXCH	指定されたBIN32ビットデータを交換します。 (d1)+1, (d1) ↔ (d2)+1, (d2)	520ページ DXCH(P)
DXCHP		

■BIN16ビットデータブロック交換

命令記号	処理内容	参照先
BXCH	指定されたデバイスから(n)点のBIN16ビットデータを交換します。 	522ページ BXCH(P)
BXCHP		

■BIN16ビットデータ上下バイト交換

命令記号	処理内容	参照先
SWAP	指定したデバイスの上下各8ビットの値を交換します。 (d)	524ページ SWAP(P)
SWAPP		

■BIN32ビットデータ上下バイト交換

命令記号	処理内容	参照先
DSWAP	指定したデバイスの上下各8ビットの値を交換します。 (d)+1 b15 ... b8 b7 ... b0 8ビット 8ビット (d)+1 b15 ... b8 b7 ... b0 8ビット 8ビット (d) b15 ... b8 b7 ... b0 8ビット 8ビット (d) b15 ... b8 b7 ... b0 8ビット 8ビット	525ページ DSWAP(P)
DSWAPP		

■1ビットデータ転送

命令記号	処理内容	参照先
MOVB	(s)で指定したビットデータを、(d)へ格納します。	526ページ MOVB(P)
MOVBP		

■nビットデータ転送

命令記号	処理内容	参照先
BLKMOVB	(s)から(n)点分のビットデータを、(d)から(n)点分のビットデータへ一括転送します。	527ページ BLKMOVB(P)
BLKMOVBP		

6.3 応用命令

プログラム制御

プログラム分岐命令

■ポインタ分岐

命令記号	処理内容	参照先
CJ	同一プログラムファイル内の指定されたポインタ番号のプログラムを実行します。	530ページ CJ, SCJ, JMP
SCJ	次のスキャンから同一プログラムファイル内の指定されたポインタ番号のプログラムを実行します。	
JMP	無条件に同一プログラムファイル内の指令されたポインタ番号のプログラムを実行します。	

■ENDへジャンプ

命令記号	処理内容	参照先
GOEND	同一プログラムファイル内のFENDまたはEND命令へジャンプします。	533ページ GOEND

プログラム実行制御命令

■割込禁止, 割込許可

命令記号	処理内容	参照先
DI	割込みプログラムの実行を禁止します。	534ページ DI, EI
EI	割込み禁止状態を解除します。	

■指定優先度以下の割込禁止

命令記号	処理内容	参照先
DI	指定した優先度以下の割込みプログラムの実行を禁止します。	536ページ DI

■割込プログラムマスク

命令記号	処理内容	参照先
IMASK	指定された割込みポインタ番号の割込みプログラムを実行許可状態/実行禁止状態にします。	541ページ IMASK

■指定割込ポインタの禁止/許可

命令記号	処理内容	参照先
SIMASK	指定した割込みポインタ番号の割込みプログラムを, 実行許可状態/実行禁止状態にします。	543ページ SIMASK

■割込プログラムからの復帰

命令記号	処理内容	参照先
IRET	割込みプログラムの処理の終了を示します。	545ページ IRET

■WDTリセット

命令記号	処理内容	参照先
WDT	ウォッチドッグタイマをリセットします。	546ページ WDT(P)
WDTP		

構造化命令

■FOR~NEXT

命令記号	処理内容	参照先
FOR	FOR~NEXT命令間の処理を(n)回実行します。	547ページ FOR, NEXT
NEXT		

■FOR~NEXT強制終了

命令記号	処理内容	参照先
BREAK	FOR~NEXT命令による繰り返し処理を強制的に終了し、FOR~NEXT命令以降の処理に実行を移します。	550ページ BREAK(P)
BREAKP		

■サブルーチンプログラムコール

命令記号	処理内容	参照先
CALL	入力条件成立で(P)のサブルーチンプログラムを実行します。 ((s1)~(s5)はサブルーチンプログラムへの引数。)	552ページ CALL(P)
CALLP		

■サブルーチンプログラムからのリターン

命令記号	処理内容	参照先
RET	サブルーチンプログラムの終了を示します。	558ページ RET

■サブルーチンプログラムの出力OFFコール

命令記号	処理内容	参照先
FCALL	入力条件不成立時に(P)のサブルーチンプログラムの非実行処理を行います。 ((s1)~(s5)はサブルーチンプログラムへの引数。)	559ページ FCALL(P)
FCALLP		

■プログラムファイル間サブルーチンコール

命令記号	処理内容	参照先
ECALL	入力条件成立で指定プログラムの(P)の、サブルーチンプログラムを実行します。 ((s1)~(s5)はサブルーチンプログラムへの引数。)	565ページ ECALL(P)
ECALLP		

■プログラムファイル間サブルーチンプログラム出力OFFコール

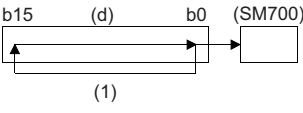
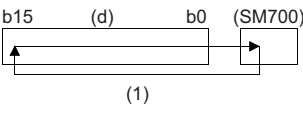
命令記号	処理内容	参照先
EFCALL	入力条件不成立時に指定プログラムの(P)の、サブルーチンプログラムの非実行処理を行います。 ((s1)~(s5)はサブルーチンプログラムへの引数。)	572ページ EFCALL(P)
EFCALLP		

■サブルーチンプログラムコール

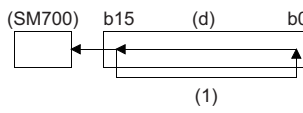
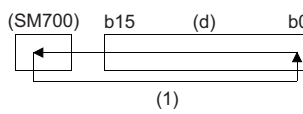
命令記号	処理内容	参照先
XCALL	入力条件成立で(P)のサブルーチンプログラムを実行します。 入力条件不成立時に(P)のサブルーチンプログラムの非実行処理を行います。 ((s1)~(s5)はサブルーチンプログラムへの引数。)	577ページ XCALL

ローテーション命令

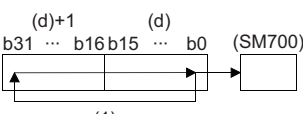
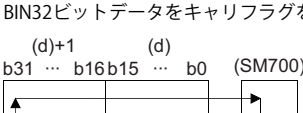
■BIN16ビットデータの右ローテーション

命令記号	処理内容	参照先
ROR RORP	BIN16ビットデータをキャリフラグを含めずに(n)ビット右へ回転させます。  (1): (n)ビット右ローテーション	582ページ ROR(P), RCR(P)
RCR RCRP	BIN16ビットデータをキャリフラグを含めて(n)ビット右へ回転させます。  (1): (n)ビット右ローテーション	

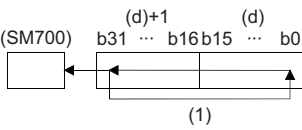
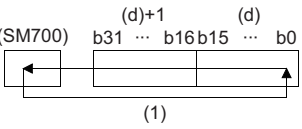
■BIN16ビットデータの左ローテーション

命令記号	処理内容	参照先
ROL ROLP	BIN16ビットデータをキャリフラグを含めずに(n)ビット左へ回転させます。  (1): (n)ビット左ローテーション	585ページ ROL(P), RCL(P)
RCL RCLP	BIN16ビットデータをキャリフラグを含めて(n)ビット左へ回転させます。  (1): (n)ビット左ローテーション	

■BIN32ビットデータの右ローテーション

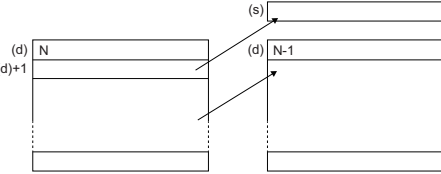
命令記号	処理内容	参照先
DROR DRORP	BIN32ビットデータをキャリフラグを含めずに(n)ビット右へ回転させます。  (1): (n)ビット右ローテーション	588ページ DROR(P), DRCR(P)
DRCR DRCRP	BIN32ビットデータをキャリフラグを含めて(n)ビット右へ回転させます。  (1): (n)ビット右ローテーション	

■BIN32ビットデータの左ローテーション

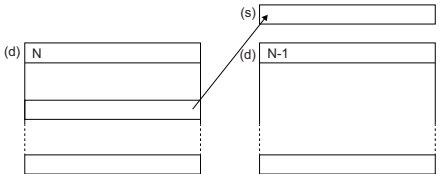
命令記号	処理内容	参照先
DROL DROL P	BIN32ビットデータをキャリフラグを含めずに(n)ビット左へ回転させます。  (1): (n)ビット左ローテーション	590ページ DROL(P), DRCL(P)
DRCL DRCL P	BIN32ビットデータをキャリフラグを含めて(n)ビット左へ回転させます。  (1): (n)ビット左ローテーション	

データテーブル操作命令

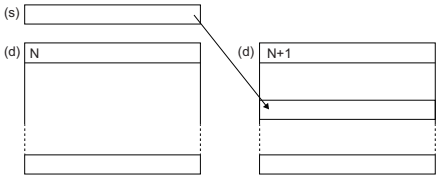
■データテーブルからの先入データリード

命令記号	処理内容	参照先
FIFR FIFR P	テーブルに最初に格納したデータを指定されたデバイスに格納します。  N: データ数	592ページ FIFR(P)

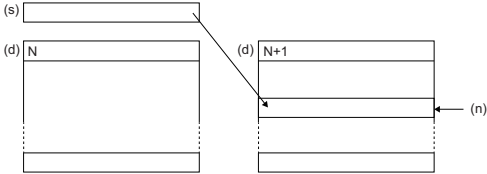
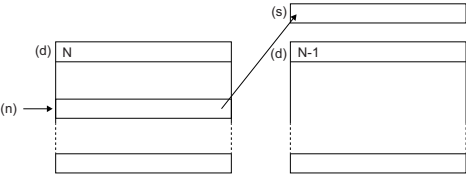
■データテーブルからの後入データリード

命令記号	処理内容	参照先
FPOP FPOP P	テーブルの最後に格納したデータを指定されたデバイスに格納します。  N: データ数	594ページ FPOP(P)

■データテーブルへのデータライト

命令記号	処理内容	参照先
FIFW FIFW P	BIN16ビットデータを、指定されたデータテーブルに格納します。  N: データ数	596ページ FIFW(P)

■データテーブルのデータ挿入，削除

命令記号	処理内容	参照先
FINS FINSP	BIN16ビットデータを指定されたデータテーブルの(n)番目に挿入します。  N: データ数	598ページ FINS(P)
FDEL FDELP	データテーブルの(n)番目のデータを削除します。  N: データ数	600ページ FDEL(P)

データリード/ライト命令

■データメモリからのデータリード

命令記号	処理内容	参照先
S.DEVLD SP.DEVLD	データメモリ上のデバイスデータ格納用ファイルからデータを読み出します。	603ページ S(P).DEVLD

■データメモリへのデータライト

命令記号	処理内容	参照先
SP.DEVST	データメモリ上のデバイスデータ格納用ファイルに書き込みます。	605ページ SP.DEVST

ファイル操作命令

■指定ファイルからのデータリード

命令記号	処理内容	参照先
SP.FREAD	指定したファイルからデータを読み出します。	607ページ SP.FREAD

■指定ファイルへのデータライト

命令記号	処理内容	参照先
SP.FWRITE	指定したファイルにデータを書き込みます。	621ページ SP.FWRITE

■指定ファイルの削除

命令記号	処理内容	参照先
SP.FDELETE	指定したファイル, またはフォルダを削除します。	635ページ SP.FDELETE

■指定ファイルのコピー

命令記号	処理内容	参照先
SP.FCOPY	指定したファイル, またはフォルダをコピーします。	638ページ SP.FCOPY

■指定ファイルの移動

命令記号	処理内容	参照先
SP.FMOVE	指定したファイル, またはフォルダを移動します。	643ページ SP.FMOVE

■指定ファイル名の変更

命令記号	処理内容	参照先
SP.FRENAME	指定したファイル名, またはフォルダ名を変更します。	648ページ SP.FRENAME

■指定ファイルの状態取得

命令記号	処理内容	参照先
SP.FSTATUS	指定したファイル, またはフォルダの状態を取得します。	652ページ SP.FSTATUS

データ制御命令

■BIN16ビット/32ビットデータ上下限リミット制御

命令記号	処理内容	参照先
LIMIT	指定した入力値(BIN16ビット値)が上下限リミット値の範囲内か否かにより, 出力値を制御します。	658ページ LIMIT(P)(_U)
LIMITP		
LIMIT_U		
LIMITP_U		
DLIMIT	指定した入力値(BIN32ビット値)が上下限リミット値の範囲内か否かにより, 出力値を制御します。	660ページ DLIMIT(P)(_U)
DLIMITP		
DLIMIT_U		
DLIMITP_U		

■BIN16ビット/32ビットデータ不感帯制御

命令記号	処理内容	参照先
BAND	指定した入力値(BIN16ビット値)が不感帯の上下限範囲内か否かにより, 出力値を制御します。	662ページ BAND(P)(_U)
BANDP		
BAND_U		
BANDP_U		
DBAND	指定した入力値(BIN32ビット値)が不感帯の上下限範囲内か否かにより, 出力値を制御します。	664ページ DBAND(P)(_U)
DBANDP		
DBAND_U		
DBANDP_U		

■BIN16ビット/32ビットデータゾーン制御

命令記号	処理内容	参照先
ZONE	指定した入力値(BIN16ビット値)にバイアス値を付加します。	666ページ ZONE(P)(_U)
ZONEP		
ZONE_U		
ZONEP_U		
DZONE	指定した入力値(BIN32ビット値)にバイアス値を付加します。	668ページ DZONE(P)(_U)
DZONEP		
DZONE_U		
DZONEP_U		

■BIN16ビット/32ビット単位スケーリング(ポイント別座標データ)

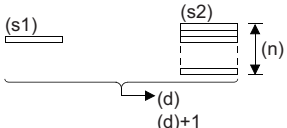
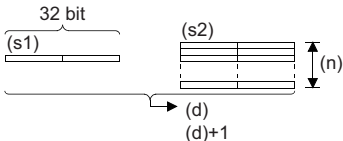
命令記号	処理内容	参照先
SCL	(s2)で指定したスケーリング用変換データ(16ビットデータ単位)に対し、(s1)で指定した入力値でスケーリングを行い、演算結果を(d)で指定したデバイスに格納します。 スケーリング変換は、(s2)で指定したデバイス以降に格納したスケーリング用変換データに基づいて行います。	670ページ SCL(P)(_U)
SCLP		
SCL_U		
SCLP_U		
DSCL	(s2)で指定したスケーリング用変換データ(32ビットデータ単位)に対し、(s1)で指定した入力値でスケーリングを行い、演算結果を(d)で指定したデバイスに格納します。 スケーリング変換は、(s2)で指定したデバイス以降に格納したスケーリング用変換データに基づいて行います。	673ページ DSCL(P)(_U)
DSCLP		
DSCL_U		
DSCLP_U		

■BIN16ビット/32ビット単位スケーリング(X/Y別座標データ)

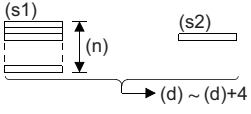
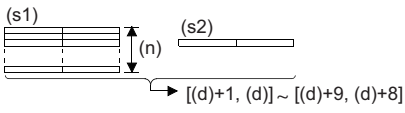
命令記号	処理内容	参照先
SCL2	(s2)で指定したスケーリング用変換データ(16ビットデータ単位)に対し、(s1)で指定した入力値でスケーリングを行い、演算結果を(d)で指定したデバイスに格納します。 スケーリング変換は、(s2)で指定したデバイス以降に格納したスケーリング用変換データに基づいて行います。	676ページ SCL2(P)(_U)
SCL2P		
SCL2_U		
SCL2P_U		
DSCL2	(s2)で指定したスケーリング用変換データ(32ビットデータ単位)に対し、(s1)で指定した入力値でスケーリングを行い、演算結果を(d)で指定したデバイスに格納します。 スケーリング変換は、(s2)で指定したデバイス以降に格納したスケーリング用変換データに基づいて行います。	678ページ DSCL2(P)(_U)
DSCL2P		
DSCL2_U		
DSCL2P_U		

データ処理命令

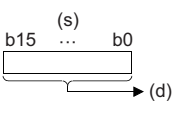
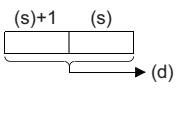
■BIN16ビット/32ビットデータサーチ

命令記号	処理内容	参照先
SERDATA SERDATAP	(s1)で指定したBIN16ビットデータを、(s2)で指定したBIN16ビットデータから(n)点分をサーチします。  (d): 一致した位置 (d)+1: 一致個数	680ページ SERDATA(P)
DSERDATA DSERDATAP	(s1)で指定したBIN32ビットデータを、(s2)で指定したBIN32ビットデータから(n)点分をサーチします。  (d): 一致した位置 (d)+1: 一致個数	682ページ DSERDATA(P)

■BIN16ビット/32ビットデータサーチ(最小・同一・最大)

命令記号	処理内容	参照先
SERMM SERMMP	(s1)で指定したBIN16ビットデータから(n)点分、(s2)で指定したBIN16ビットデータと同一データおよび最大値、最小値を検索します。 	684ページ SERMM(P)
DSERMM DSERMMP	(s1)で指定したBIN32ビットデータから(n)点分、(s2)で指定したBIN32ビットデータと同一データおよび最大値、最小値を検索します。 	686ページ DSERMM(P)

■BIN16ビット/32ビットデータビットチェック

命令記号	処理内容	参照先
SUM SUMP	指定したデバイスのBIN16ビットデータのうち、1になっているビットの総数を格納します。  (d): 1の総数	688ページ SUM(P)
DSUM DSUMP	指定したデバイスのBIN32ビットデータのうち、1になっているビットの総数を格納します。  (d): 1の総数	689ページ DSUM(P)

■BIN16ビット/32ビットデータ最大値検索

命令記号	処理内容	参照先
MAX MAXP MAX_U MAXP_U	(s)で指定のデバイスから(n)点分のデータを16ビット単位で検索し、最大値を(d)で指定のデバイスに格納します。	690ページ MAX(P)(_U)
DMAX DMAXP DMAX_U DMAXP_U	(s)で指定のデバイスから(n)点分のデータを32ビット単位で検索し、最大値を(d)で指定のデバイスに格納します。	692ページ DMAX(P)(_U)

■BIN16ビット/32ビットデータ最小値検索

命令記号	処理内容	参照先
MIN MINP MIN_U MINP_U	(s)で指定のデバイスから(n)点分のデータを16ビット単位で検索し、最小値を(d)で指定のデバイスに格納します。	694ページ MIN(P)(_U)
DMIN DMINP DMIN_U DMINP_U	(s)で指定のデバイスから(n)点分のデータを32ビット単位で検索し、最小値を(d)で指定のデバイスに格納します。	696ページ DMIN(P)(_U)

■BIN16ビット/32ビットデータソート

命令記号	処理内容	参照先
SORTD	(s1)で指定されたデバイスから(n)点分のデータを、16ビット単位でソートします。 (n)×((n)-1)÷2スキャン必要	698ページ SORTD(_U)
SORTD_U		
DSORTD	(s1)で指定されたデバイスから(n)点分のデータを、32ビット単位でソートします。 (n)×((n)-1)÷2スキャン必要	700ページ DSORTD(_U)
DSORTD_U		

■BIN16ビットデータ合計値算出

命令記号	処理内容	参照先
WSUM	(s)で指定されたデバイスから(n)点分の16ビットBINデータをすべて加算し、(d)で指定のデバイスに格納します。	702ページ WSUM(P)(_U)
WSUM_U		
WSUMP		
WSUMP_U		

■BIN32ビットデータ合計値算出

命令記号	処理内容	参照先
DWSUM	(s)で指定されたデバイスから(n)点分の32ビットBINデータをすべて加算し、(d)で指定のデバイスに格納します。	704ページ DWSUM(P)(_U)
DWSUM_U		
DWSUMP		
DWSUMP_U		

■BIN16ビット/32ビットデータ平均値算出

命令記号	処理内容	参照先
MEAN	(s)で指定されたデバイスから(n)点分(16ビット単位)の平均値を算出し、(d)で指定のデバイスに格納します。	706ページ MEAN(P)(_U)
MEANP		
MEAN_U		
MEANP_U		
DMEAN	(s)で指定されたデバイスから(n)点分(32ビット単位)の平均値を算出し、(d)で指定のデバイスに格納します。	708ページ DMEAN(P)(_U)
DMEANP		
DMEAN_U		
DMEANP_U		

■BIN16ビット/32ビット平方根算出

命令記号	処理内容	参照先
SQRT	指定したBIN16ビットデータの平方根を演算します。 $\sqrt{(s)} \rightarrow (d)$	710ページ SQRT(P)
SQ RTP		
DSQRT	指定したBIN32ビットデータの平方根を演算します。 $\sqrt{(s)+1}, (s) \rightarrow (d)$	711ページ DSQRT(P)
DSQ RTP		

■CRC演算

命令記号	処理内容	参照先
CRC	(s)で指定したデバイスを先頭に(n)点の8ビットデータ(バイト単位)について、CRC値を生成し(d)へ格納します。	712ページ CRC(P)
CRCP		

デバッグ，故障診断

デバッグ，故障診断命令

■エラー表示または，アナンシェータリセット

命令記号	処理内容	参照先
LEDR	コントローラのアナンシェータ表示や運転を続行できる自己診断エラーの表示をリセットします。	714ページ LEDR

文字列処理

文字列処理命令

■文字列比較

命令記号	処理内容	参照先
LD\$=, AND\$=, OR\$=	(s1)の文字列と(s2)の文字列を1文字ずつ比較します。	716ページ LD\$□, AND\$□, OR\$□
LD\$<>, AND\$<>, OR\$<>		
LD\$>, AND\$>, OR\$>		
LD\$<=, AND\$<=, OR\$<=		
LD\$<, AND\$<, OR\$<		
LD\$>=, AND\$>=, OR\$>=		

■文字列の結合

命令記号	処理内容	参照先
\$+	(d)で指定の文字列に(s)で指定の文字列を結合し，(d)以降に格納します。	719ページ \$(P) [オペランドが2個の場合]
\$+P		
\$+	(s1)で指定の文字列に(s2)で指定の文字列を結合し，(d)以降に格納します。	721ページ \$(P) [オペランドが3個の場合]
\$+P		

■文字列転送

命令記号	処理内容	参照先
\$MOV	(s)で指定の文字列を(d)で指定のデバイス以降に転送します。	723ページ \$MOV(P)
\$MOVP		
\$MOV_WS	(s)で指定のUnicode文字列を(d)で指定のデバイス以降に転送します。	725ページ \$MOV(P)_WS
\$MOVP_WS		

■BIN16ビット/32ビットデータ→10進アスキー変換

命令記号	処理内容	参照先
BINDA	(s)で指定した1ワードBIN値を10進アスキー値に変換し，(d)で指定したワードデバイス番号以降に格納します。	727ページ BINDA(P)(_U)
BINDAP		
BINDA_U		
BINDAP_U		
DBINDA	(s)で指定した2ワードBIN値を10進アスキー値に変換し，(d)で指定したワードデバイス番号以降に格納します。	732ページ DBINDA(P)(_U)
DBINDAP		
DBINDA_U		
DBINDAP_U		

■BIN16ビット/32ビットデータ→16進アスキー変換

命令記号	処理内容	参照先
BINHA	(s)で指定した1ワードBIN値を16進アスキー値に変換し、(d)で指定したワードデバイス番号以降に格納します。	737ページ BINHA(P)
BINHAP		
DBINHA	(s)で指定した2ワードBIN値を16進アスキー値に変換し、(d)で指定したワードデバイス番号以降に格納します。	741ページ DBINHA(P)
DBINHAP		

■BIN16ビット/32ビットデータ→文字列変換

命令記号	処理内容	参照先
STR	(s2)で指定した1ワードBIN値を、(s1)で指定した全桁数と小数部桁数の10進文字列に変換し、(d)で指定したデバイスに格納します。	745ページ STR(P)(_U)
STRP		
STR_U		
STRP_U		
DSTR	(s2)で指定した2ワードBIN値を、(s1)で指定した全桁数と小数部桁数の10進文字列に変換し、(d)で指定したデバイスに格納します。	748ページ DSTR(P)(_U)
DSTRP		
DSTR_U		
DSTRP_U		

■BCD4桁/8桁データ→10進アスキーコード変換

命令記号	処理内容	参照先
BCDDA	(s)で指定した1ワードBCD値を10進アスキー値に変換し、(d)で指定したワードデバイス番号以降に格納します。	751ページ BCDDA(P)
BCDDAP		
DBCDDA	(s)で指定した2ワードBCD値を10進アスキー値に変換し、(d)で指定したワードデバイス番号以降に格納します。	755ページ DBCDDA(P)
DBCDDAP		

■単精度実数→文字列変換

命令記号	処理内容	参照先
ESTR	(s1)で指定した単精度実数データを文字列に変換し、(d)で指定したデバイスに格納します。	760ページ ESTR(P)
ESTRP		

■16進BINデータ→16進アスキーコード変換

命令記号	処理内容	参照先
INT2ASC	(s)で指定したデバイス番号以降の1ワードBIN値を16進アスキーに変換し、(d)で指定したワードデバイス番号以降に(n)で指定した文字数分を格納します。	766ページ INT2ASC(P)
INT2ASCP		

■複数データの文字列変換結合

命令記号	処理内容	参照先
SPF	複数のデータを指定した表示形式に従い文字列に変換し、任意の文字列と連結して1つの文字列として出力します。	768ページ SPF(P)
SPFP		

■Unicode文字列→シフトJIS文字列変換

命令記号	処理内容	参照先
WS2SJIS	(s)で指定したUnicode文字列をシフトJIS文字列に変換し、(d)へ格納します。	784ページ WS2SJIS(P)
WS2SJISP		

■シフトJIS文字列→Unicode文字列変換(バイトオーダーマークなし)

命令記号	処理内容	参照先
SJIS2WS	(s)で指定したシフトJIS文字列をUnicode文字列に変換し、(d)へ格納します。	786ページ SJIS2WS(P)
SJIS2WSP		

■シフトJIS→Unicode変換(バイトオーダーマークあり)

命令記号	処理内容	参照先
SJIS2WSB	(s)で指定したシフトJIS文字列をUnicode文字列に変換し、先頭にバイトオーダーマークを付加して(d)へ格納します。	788ページ SJIS2WSB(P)
SJIS2WSBP		

■文字列の長さ検出

命令記号	処理内容	参照先
LEN	(s)で指定したデバイスに格納されている文字列データの長さ(文字数)を、(d)で指定したデバイスに格納します。	790ページ LEN(P)
LENP		

■文字列の右側/左側からの取出し

命令記号	処理内容	参照先
RIGHT	(s)で指定された文字列の最終文字から(n)文字分を、(d)の指定デバイスに格納します。	792ページ RIGHT(P)
RIGHTP		
LEFT	(s)で指定された文字列の先頭文字から(n)文字分を、(d)の指定デバイスに格納します。	794ページ LEFT(P)
LEFTP		

■文字列中の任意取出し、置換え

命令記号	処理内容	参照先
MIDR	(s1)で指定された文字列の(s2)で指定された位置から、指定された文字数分を(d)で指定されたデバイスに格納します。	796ページ MIDR(P)
MIDRP		
MIDW	(s1)の文字列から指定された文字数を、(d)の文字列の(s2)で指定した位置に格納します。	798ページ MIDW(P)
MIDWP		

■文字列サーチ

命令記号	処理内容	参照先
INSTR	(s1)の文字列を(s2)の文字列の(s3)文字目から検索して、一致した位置を(d)に格納します。	801ページ INSTR(P)
INSTRP		

■文字列挿入

命令記号	処理内容	参照先
STRINS	(s1)で指定された文字列データを、(d)で指定された文字列データの先頭から(s2)文字目(挿入位置)に挿入します。	803ページ STRINS(P)
STRINSP		

■文字列削除

命令記号	処理内容	参照先
STRDEL	(d)で指定された文字列データの先頭から、(s)文字目で指定された位置(削除開始位置)より、(n)文字分を削除します。	805ページ STRDEL(P)
STRDELP		

浮動小数点命令

■単精度実数比較

命令記号	処理内容	参照先
LDE=, ANDE=, ORE=	単精度実数をa接点扱いで比較演算します。	807ページ LDE□, ANDE□, ORE□
LDE<>, ANDE<>, ORE<>		
LDE>, ANDE>, ORE>		
LDE<=, ANDE<=, ORE<=		
LDE<, ANDE<, ORE<		
LDE>=, ANDE>=, ORE>=		

■倍精度実数比較

命令記号	処理内容	参照先
LDED=, ANDED=, ORED=	倍精度実数をa接点扱いで比較演算します。	809ページ LDED□, ANDED□, ORED□
LDED<>, ANDED<>, ORED<>		
LDED>, ANDED>, ORED>		
LDED<=, ANDED<=, ORED<=		
LDED<, ANDED<, ORED<		
LDED>=, ANDED>=, ORED>=		

■単精度実数加減算

命令記号	処理内容	参照先
E+	単精度実数を加算します。(オペランドが2個の場合)	812ページ E+(P) [オペランドが2個の場合]
E+P		
E+	単精度実数を加算します。(オペランドが3個の場合)	814ページ E+(P) [オペランドが3個の場合]
E+P		
E-	単精度実数を減算します。(オペランドが2個の場合)	816ページ E-(P) [オペランドが2個の場合]
E-P		
E-	単精度実数を減算します。(オペランドが3個の場合)	818ページ E-(P) [オペランドが3個の場合]
E-P		

■倍精度実数加減算

命令記号	処理内容	参照先
ED+	倍精度実数を加算します。(オペランドが2個の場合)	820ページ ED+(P) [オペランドが2個の場合]
ED+P		
ED+	倍精度実数を加算します。(オペランドが3個の場合)	822ページ ED+(P) [オペランドが3個の場合]
ED+P		
ED-	倍精度実数を減算します。(オペランドが2個の場合)	824ページ ED-(P) [オペランドが2個の場合]
ED-P		
ED-	倍精度実数を減算します。(オペランドが3個の場合)	826ページ ED-(P) [オペランドが3個の場合]
ED-P		

■単精度実数乗除算

命令記号	処理内容	参照先
E*	単精度実数を乗算します。	828ページ E*(P)
E*P		
E/	単精度実数を除算します。	830ページ E/(P)
E/P		

■倍精度実数乗除算

命令記号	処理内容	参照先
ED*	倍精度実数を乗算します。	832ページ ED*(P)
ED*P		
ED/	倍精度実数を除算します。	834ページ ED/(P)
ED/P		

■符号付きBIN16ビット/32ビットデータ→単精度実数変換

命令記号	処理内容	参照先
INT2FLT	(s)で指定した符号付きBIN16ビットデータを単精度実数に変換し、(d)へ格納します。	836ページ INT2FLT(P)
INT2FLTP		
DINT2FLT	(s)で指定した符号付きBIN32ビットデータを単精度実数に変換し、(d)へ格納します。	838ページ DINT2FLT(P)
DINT2FLTP		

■符号なしBIN16ビット/32ビットデータ→単精度実数変換

命令記号	処理内容	参照先
UINT2FLT	(s)で指定した符号なしBIN16ビットデータを単精度実数に変換し、(d)へ格納します。	837ページ UINT2FLT(P)
UINT2FLTP		
UDINT2FLT	(s)で指定した符号なしBIN32ビットデータを単精度実数に変換し、(d)へ格納します。	840ページ UDINT2FLT(P)
UDINT2FLTP		

■倍精度→単精度変換

命令記号	処理内容	参照先
DBL2FLT	(s)で指定した倍精度実数を単精度実数に変換し、(d)へ格納します。	842ページ DBL2FLT(P)
DBL2FLTP		

■符号付きBIN16ビット/32ビットデータ→倍精度実数変換

命令記号	処理内容	参照先
INT2DBL	(s)で指定した符号付きBIN16ビットデータを倍精度実数に変換し、(d)へ格納します。	844ページ INT2DBL(P)
INT2DBLP		
DINT2DBL	(s)で指定した符号付きBIN32ビットデータを倍精度実数に変換し、(d)へ格納します。	846ページ DINT2DBL(P)
DINT2DBLP		

■符号なしBIN16ビット/32ビットデータ→倍精度実数変換

命令記号	処理内容	参照先
UINT2DBL	(s)で指定した符号なしBIN16ビットデータを倍精度実数に変換し、(d)へ格納します。	845ページ UINT2DBL(P)
UINT2DBLP		
UDINT2DBL	(s)で指定した符号なしBIN32ビットデータを倍精度実数に変換し、(d)へ格納します。	847ページ UDINT2DBL(P)
UDINT2DBLP		

■単精度→倍精度変換

命令記号	処理内容	参照先
FLT2DBL	(s)で指定した単精度実数を倍精度実数に変換し、(d)へ格納します。	848ページ FLT2DBL(P)
FLT2DBLP		

■文字列→単精度実数変換

命令記号	処理内容	参照先
EVAL	(s)で指定した文字列を単精度実数に変換し、(d)で指定したデバイスに格納します。	849ページ EVAL(P)
EVALP		

■BCDフォーマットデータ→単精度実数データ変換

命令記号	処理内容	参照先
EREXP	(s1)のBCDデータを(s2)で指定した小数部桁数で単精度実数に変換して、(d)で指定したデバイスに格納します。	853ページ EREXP(P)
EREXPP		

■単精度実数符号反転

命令記号	処理内容	参照先
ENEG	単精度実数データの符号を反転します。	855ページ ENEG(P)
ENEGP		

$$\begin{array}{c} \overline{(d)+1, (d)} \longrightarrow (d)+1, (d) \\ \uparrow \\ (1) \end{array}$$

(1): 実数

■倍精度実数符号反転

命令記号	処理内容	参照先
EDNEG	倍精度実数データの符号を反転します。	856ページ EDNEG(P)
EDNEGP		

$$\begin{array}{c} \overline{(d)+3, (d)+2, (d)+1, (d)} \longrightarrow (d)+3, (d)+2, (d)+1, (d) \\ \uparrow \\ (1) \end{array}$$

(1): 実数

■単精度実数データ転送

命令記号	処理内容	参照先
EMOV	単精度実数データを指定されたデバイスへ転送します。	857ページ EMOV(P)
EMOVP		

$$\begin{array}{c} \overline{(s)+1, (s)} \longrightarrow (d)+1, (d) \\ \uparrow \\ (1) \end{array}$$

(1): 実数

■倍精度実数データ転送

命令記号	処理内容	参照先
EDMOV	倍精度実数データを指定されたデバイスへ転送します。	858ページ EDMOV(P)
EDMOVP		

$$\begin{array}{c} \overline{(s)+3, (s)+2, (s)+1, (s)} \longrightarrow (d)+3, (d)+2, (d)+1, (d) \\ \uparrow \\ (1) \end{array}$$

(1): 実数

■単精度実数SIN演算

命令記号	処理内容	参照先
SIN	単精度実数で指定した角度のSIN(正弦)値を演算します。	859ページ SIN(P)
SINP		

■単精度実数COS演算

命令記号	処理内容	参照先
COS	単精度実数で指定した角度のCOS(余弦)値を演算します。	861ページ COS(P)
COSP		

■単精度実数TAN演算

命令記号	処理内容	参照先
TAN	単精度実数で指定した角度のTAN(正接)値を演算します。	863ページ TAN(P)
TANP		

■単精度実数 SIN^{-1} 演算

命令記号	処理内容	参照先
ASIN	単精度実数で指定したSIN値から角度を演算します。	865ページ ASIN(P)
ASINP		

■単精度実数 COS^{-1} 演算

命令記号	処理内容	参照先
ACOS	単精度実数で指定したCOS値から角度を演算します。	867ページ ACOS(P)
ACOSP		

■単精度実数 TAN^{-1} 演算

命令記号	処理内容	参照先
ATAN	単精度実数で指定したTAN値から角度を演算します。	869ページ ATAN(P)
ATANP		

■倍精度実数SIN演算

命令記号	処理内容	参照先
SIND	倍精度実数で指定した角度のSIN(正弦)値を演算します。	871ページ SIND(P)
SINDP		

■倍精度実数COS演算

命令記号	処理内容	参照先
COSD	倍精度実数で指定した角度のCOS(余弦)値を演算します。	873ページ COSD(P)
COSDP		

■倍精度実数TAN演算

命令記号	処理内容	参照先
TAND	倍精度実数で指定した角度のTAN(正接)値を演算します。	875ページ TAND(P)
TANDP		

■倍精度実数 SIN^{-1} 演算

命令記号	処理内容	参照先
ASIND	倍精度実数で指定したSIN値から角度を演算します。	877ページ ASIND(P)
ASINDP		

■倍精度実数 COS^{-1} 演算

命令記号	処理内容	参照先
ACOSD	倍精度実数で指定したCOS値から角度を演算します。	879ページ ACOSD(P)
ACOSDP		

■倍精度実数 TAN^{-1} 演算

命令記号	処理内容	参照先
ATAND	倍精度実数で指定したTAN値から角度を演算します。	881ページ ATAND(P)
ATANDP		

■BCD型SIN演算

命令記号	処理内容	参照先
BSIN	BCD値で指定した角度のSIN(正弦)値を演算します。	883ページ BSIN(P)
BSINP		

$\text{SIN}(s) \longrightarrow \begin{array}{|c|} \hline (d) \\ \hline (d)+1 \\ \hline (d)+2 \\ \hline \end{array}$

(d): 符号
 (d)+1: 整数部
 (d)+2: 小数部

■BCD型COS演算

命令記号	処理内容	参照先
BCOS	BCD値で指定した角度のCOS(余弦)値を演算します。	885ページ BCOS(P)
BCOSP	$\text{COS}(s) \longrightarrow \begin{array}{ c } \hline (d) \\ \hline (d)+1 \\ \hline (d)+2 \\ \hline \end{array}$ (d): 符号 (d)+1: 整数部 (d)+2: 小数部	

■BCD型TAN演算

命令記号	処理内容	参照先
BTAN	BCD値で指定した角度のTAN(正接)値を演算します。	887ページ BTAN(P)
BTANP	$\text{TAN}(s) \longrightarrow \begin{array}{ c } \hline (d) \\ \hline (d)+1 \\ \hline (d)+2 \\ \hline \end{array}$ (d): 符号 (d)+1: 整数部 (d)+2: 小数部	

■BCD型SIN⁻¹演算

命令記号	処理内容	参照先
BASIN	BCD値で指定した値のSIN ⁻¹ (逆正弦)値を演算します。	889ページ BASIN(P)
BASINP	$\text{SIN}^{-1}(s) \longrightarrow \begin{array}{ c } \hline (d) \\ \hline (d)+1 \\ \hline (d)+2 \\ \hline \end{array}$ (d): 符号 (d)+1: 整数部 (d)+2: 小数部	

■BCD型COS⁻¹演算

命令記号	処理内容	参照先
BACOS	BCD値で指定した値のCOS ⁻¹ (逆余弦)値を演算します。	891ページ BACOS(P)
BACOSP	$\text{COS}^{-1}(s) \longrightarrow \begin{array}{ c } \hline (d) \\ \hline (d)+1 \\ \hline (d)+2 \\ \hline \end{array}$ (d): 符号 (d)+1: 整数部 (d)+2: 小数部	

■BCD型TAN⁻¹演算

命令記号	処理内容	参照先
BATAN	BCD値で指定した値のTAN ⁻¹ (逆正接)値を演算します。	893ページ BATAN(P)
BATANP	$\text{TAN}^{-1}(s) \longrightarrow \begin{array}{ c } \hline (d) \\ \hline (d)+1 \\ \hline (d)+2 \\ \hline \end{array}$ (d): 符号 (d)+1: 整数部 (d)+2: 小数部	

■単精度実数角度→ラジアン変換

命令記号	処理内容	参照先
RAD	角度の大きさの単位を、単精度実数で指定した度単位からラジアン単位に変換する。 度→ラジアン変換 $(s)+1, (s) \longrightarrow (d)+1, (d)$	895ページ RAD(P)
RADP		

■単精度実数ラジアン→角度変換

命令記号	処理内容	参照先
DEG	角度の大きさの単位を、単精度実数で指定したラジアン単位から度単位に変換します。 ラジアン→度変換 $(s)+1, (s) \longrightarrow (d)+1, (d)$	897ページ DEG(P)
DEGP		

■倍精度実数角度→ラジアン変換

命令記号	処理内容	参照先
RADD	角度の大きさの単位を、倍精度実数で指定した度単位からラジアン単位に変換します。 度→ラジアン変換 $(s)+3, (s)+2, (s)+1, (s) \longrightarrow (d)+3, (d)+2, (d)+1, (d)$	899ページ RADD(P)
RADDP		

■倍精度実数ラジアン→角度変換

命令記号	処理内容	参照先
DEGD	角度の大きさの単位を、倍精度実数で指定したラジアン単位から度単位に変換します。 ラジアン→度変換 $(s)+3, (s)+2, (s)+1, (s) \longrightarrow (d)+3, (d)+2, (d)+1, (d)$	901ページ DEGD(P)
DEGDP		

■単精度実数平方根

命令記号	処理内容	参照先
ESQRT	単精度実数で指定した値の平方根を演算します。	903ページ ESQRT(P)
ESQRTP	$\sqrt{(s)+1, (s)} \longrightarrow (d)+1, (d)$	

■倍精度実数平方根

命令記号	処理内容	参照先
EDSQRT	倍精度実数で指定した値の平方根を演算します。	905ページ EDSQRT(P)
EDSQRTP	$\sqrt{(s)+3, (s)+2, (s)+1, (s)} \longrightarrow (d)+3, (d)+2, (d)+1, (d)$	

■単精度実数指数演算

命令記号	処理内容	参照先
EXP	単精度実数で指定した値の指数を演算します。	907ページ EXP(P)
EXPP		

■倍精度実数指数演算

命令記号	処理内容	参照先
EXPD	倍精度実数で指定した値の指数を演算します。	909ページ EXPD(P)
EXPDP		

■単精度実数自然対数演算

命令記号	処理内容	参照先
LOG	単精度実数で指定した値の自然対数(e)を底としたときの対数を演算します。	911ページ LOG(P)
LOGP		

■倍精度実数自然対数演算

命令記号	処理内容	参照先
LOGD	倍精度実数で指定した値の自然対数(e)を底としたときの対数を演算します。	913ページ LOGD(P)
LOGDP		

■BCD4桁/8桁平方根

命令記号	処理内容	参照先
BSQRT	BCD値(4桁)で指定した値の平方根を演算します。 $\sqrt{(s)} \longrightarrow \begin{array}{ c } \hline (d) \\ \hline (d)+1 \\ \hline \end{array}$ (d): 整数部 (d)+1: 小数部	915ページ BSQRT(P)
BSQRTP		
BDSQRT	BCD値(8桁)で指定した値の平方根を演算します。 $\sqrt{(s)+1, (s)} \longrightarrow \begin{array}{ c } \hline (d) \\ \hline (d)+1 \\ \hline \end{array}$ (d): 整数部 (d)+1: 小数部	917ページ BDSQRT(P)
BDSQRTP		

■単精度実数べき乗演算

命令記号	処理内容	参照先
POW	単精度実数のべき乗を演算します。	919ページ POW(P)
POWP		

■倍精度実数べき乗演算

命令記号	処理内容	参照先
POWD	倍精度実数のべき乗を演算します。	921ページ POWD(P)
POWDP		

■単精度実数常用対数演算

命令記号	処理内容	参照先
LOG10	単精度実数で指定した値の常用対数(10を底とした対数)を演算します。	923ページ LOG10(P)
LOG10P		

■倍精度実数常用対数演算

命令記号	処理内容	参照先
LOG10D	倍精度実数で指定した値の常用対数(10を底とした対数)を演算します。	925ページ LOG10D(P)
LOG10DP		

■単精度実数最大値検索

命令記号	処理内容	参照先
EMAX	(s)から(n)点分で指定した単精度実数のブロックデータの最大値を、(d)で指定した検索結果(最大値)へ格納します。	927ページ EMAX(P)
EMAXP		

■倍精度実数最大値検索

命令記号	処理内容	参照先
EDMAX	(s)から(n)点分で指定した倍精度実数のブロックデータの最大値を、(d)で指定した検索結果(最大値)へ格納します。	929ページ EDMAX(P)
EDMAXP		

■単精度実数最小値検索

命令記号	処理内容	参照先
EMIN	(s)から(n)点分で指定した単精度実数のブロックデータの最小値を、(d)で指定した検索結果(最小値)へ格納します。	931ページ EMIN(P)
EMINP		

■倍精度実数最小値検索

命令記号	処理内容	参照先
EDMIN	(s)から(n)点分で指定した倍精度実数のブロックデータの最小値を、(d)で指定した検索結果(最小値)へ格納します。	933ページ EDMIN(P)
EDMINP		

乱数

乱数命令

■乱数発生，系列変更

命令記号	処理内容	参照先
RND	0~32767未満の乱数を発生し、(d)で指定したデバイスに格納します。	935ページ RND(P)
RNDP		
SRND	(s)で指定されたデバイスに格納されている16ビットBINデータの内容に従って、乱数系列を変更します。	936ページ SRND(P)
SRNDP		

インデックスレジスタ命令

■インデックスレジスタの一括退避、復帰

命令記号	処理内容	参照先
ZPUSH	インデックスレジスタの内容を、(d)で指定した領域に退避します。	937ページ ZPUSH(P)
ZPUSHP		
ZPOP	(d)で指定した領域に退避されているデータを、インデックスレジスタに読み出します。	940ページ ZPOP(P)
ZPOPP		

■インデックスレジスタ/ロングインデックスレジスタの選択退避、復帰

命令記号	処理内容	参照先
ZPUSH	(s)で指定したインデックスレジスタ、ロングインデックスレジスタの内容を、(d)で指定した領域に退避します。	941ページ ZPUSH(P)
ZPUSHP		
ZPOP	(d)で指定した領域に退避されているデータを、インデックスレジスタ、ロングインデックスレジスタに読み出します。	945ページ ZPOP(P)
ZPOPP		

ファイルレジスタ操作命令

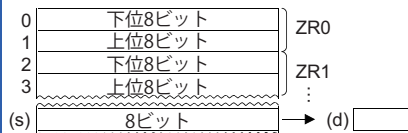
■ファイルレジスタのブロックNo.切換え

命令記号	処理内容	参照先
RSET	プログラム中で使用するファイルレジスタのブロックNo.を(s)で指定されたデバイスに格納されているブロックNo.に変更します。	947ページ RSET(P)
RSETP		

ファイルレジスタの1バイト単位のリード/ライト命令

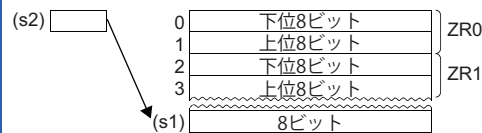
■ファイルレジスタの1バイトデータリード

命令記号	処理内容	参照先
ZRRDB	指定した通しバイトNo.のファイルレジスタの内容を読み出します。	949ページ ZRRDB(P)
ZRRDBP		



■ファイルレジスタの1バイトデータライト

命令記号	処理内容	参照先
ZRWRB	指定したデバイスの下位ビットの内容を通しバイトNo.のファイルレジスタに書き込みます。	951ページ ZRWRB(P)
ZRWRBP		



間接アドレスリード命令

■間接アドレス読出し

命令記号	処理内容	参照先
ADRSET	指定したデバイスの間接アドレスを読み出します。	953ページ ADRSET(P)
ADRSETP	(1): 指定デバイスの間接アドレス (2): デバイス名	

タイマ，カウンタ

特殊カウンタ命令

■1相入力アップ/ダウンカウンタ

命令記号	処理内容	参照先
UDCNT1	指定したカウンタの現在値を更新します。 (1): Cn現在値 (2): Cn接点	955ページ UDCNT1

■2相入力アップ/ダウンカウンタ

命令記号	処理内容	参照先
UDCNT2	指定したA相パルスとB相パルスの状態により，カウンタの現在値を更新します。 (1): Cn現在値 (2): Cn接点	957ページ UDCNT2

特殊タイマ命令

■ティーチングタイマ

命令記号	処理内容	参照先
TTMR	計測指令のON中の時間を秒単位で測定して，乗数を掛けた値を格納します。 (s)=0:1, (s)=1:10, (s)=2:100 T_{ON}: TTMRのON時間	959ページ TTMR

■特殊機能タイマ

命令記号	処理内容	参照先
STMR	STMR命令の入力条件のON/OFFにより，(d)で指定のビットデバイスから4点が下記の動作を行います。 • (d)+0: オフディレイタイマ出力 • (d)+1: オフ後ワンショットタイマ出力 • (d)+2: オン後ワンショットタイマ出力 • (d)+3: オンディレイ+オフディレイタイマ	961ページ STMR

時計用命令

■時計データの読出し

命令記号	処理内容	参照先							
DATERD	コントローラの時計素子より「年，月，日，時，分，秒，曜日」を読み出します。	963ページ DATERD(P)							
DATERDP	<table><tr><td>(d)</td></tr><tr><td>(d)+1</td></tr><tr><td>(d)+2</td></tr><tr><td>(d)+3</td></tr><tr><td>(d)+4</td></tr><tr><td>(d)+5</td></tr><tr><td>(d)+6</td></tr></table> (d): 年 (d)+1: 月 (d)+2: 日 (d)+3: 時 (d)+4: 分 (d)+5: 秒 (d)+6: 曜日	(d)	(d)+1	(d)+2	(d)+3	(d)+4	(d)+5	(d)+6	
(d)									
(d)+1									
(d)+2									
(d)+3									
(d)+4									
(d)+5									
(d)+6									

■時計データの書込み

命令記号	処理内容	参照先						
DATEWR	指定したデバイス以降に格納している時計データを，コントローラの時計素子に書き込みます。	965ページ DATEWR(P)						
DATEWRP	<table><tr><td>(s)</td></tr><tr><td>(s)+1</td></tr><tr><td>(s)+2</td></tr><tr><td>(s)+3</td></tr><tr><td>(s)+4</td></tr><tr><td>(s)+5</td></tr><tr><td>(s)+6</td></tr></table> (s): 年 (s)+1: 月 (s)+2: 日 (s)+3: 時 (s)+4: 分 (s)+5: 秒 (s)+6: 曜日		(s)	(s)+1	(s)+2	(s)+3	(s)+4	(s)+5
(s)								
(s)+1								
(s)+2								
(s)+3								
(s)+4								
(s)+5								
(s)+6								

■時計データの加算

命令記号	処理内容	参照先									
DATE+	時間データを加算します。	967ページ DATE+(P)									
DATE+P	(s1)<table><tr><td>hour</td></tr><tr><td>minute</td></tr><tr><td>second</td></tr></table> + (s2)<table><tr><td>hour</td></tr><tr><td>minute</td></tr><tr><td>second</td></tr></table> → (d)<table><tr><td>hour</td></tr><tr><td>minute</td></tr><tr><td>second</td></tr></table>	hour	minute	second	hour	minute	second	hour	minute	second	
hour											
minute											
second											
hour											
minute											
second											
hour											
minute											
second											

■時計データの減算

命令記号	処理内容	参照先								
DATE-	時間データを減算します。	969ページ DATE-(P)								
DATE-P	(s1)<table><tr><td>hour</td></tr><tr><td>minute</td></tr><tr><td>second</td></tr></table> - (s2)<table><tr><td>hour</td></tr><tr><td>minute</td></tr><tr><td>second</td></tr></table> → (d)<table><tr><td>hour</td></tr><tr><td>minute</td></tr><tr><td>second</td></tr></table>		hour	minute	second	hour	minute	second	hour	minute
hour										
minute										
second										
hour										
minute										
second										
hour										
minute										
second										

■時間データの変換(時分秒→秒)

命令記号	処理内容	参照先
TIME2SEC	時間データ(時, 分, 秒)を秒のデータに換算します。	971ページ TIME2SEC(P)
TIME2SECP		

■時間データの変換(秒→時分秒)

命令記号	処理内容	参照先
SEC2TIME	秒のデータを時間データ(時, 分, 秒)に換算します。	973ページ SEC2TIME(P)
SEC2TIMEP		

■日時データの変換

命令記号	処理内容	参照先
DATE2SEC	日時のデータ(年, 月, 日, 時, 分, 秒)を秒のデータに換算します。	975ページ DATE2SEC(P)(_U)
DATE2SECP	(s)	
DATE2SEC_U		
DATE2SEC_U		
SEC2DATE	秒のデータを日時のデータ(年, 月, 日, 時, 分, 秒, 曜日)に換算します。	977ページ SEC2DATE(P)(_U)
SEC2DATEP	(d)	
SEC2DATE_U		
SEC2DATEP_U		

■日付比較

命令記号	処理内容	参照先
LDDT=, ANDDT=, ORDT=	指定した日付データの比較, または日付データと現在日付を比較します。	979ページ LDDT□, ANDDT□, ORDT□
LDDT<>, ANDDT<>, ORDT<>		
LDDT>, ANDDT>, ORDT>		
LDDT<=, ANDDT<=, ORDT<=		
LDDT<, ANDDT<, ORDT<		
LDDT>=, ANDDT>=, ORDT>=		

■時刻比較

命令記号	処理内容	参照先
LDTM=, ANDTM=, ORTM=	指定した時刻データの比較、または指定した時刻データと現在時刻を比較します。	983ページ LDTM□, ANDTM□, ORTM□
LDTM<>, ANDTM<>, ORTM<>		
LDTM>, ANDTM>, ORTM>		
LDTM<=, ANDTM<=, ORTM<=		
LDTM<, ANDTM<, ORTM<		
LDTM>=, ANDTM>=, ORTM>=		

■時刻比較出力

命令記号	処理内容	参照先
TCMP	(s1), (s2), (s3)で指定した比較する時刻データと(s4)で指定した時刻データを比較し、その結果(小, 一致, 大)に応じて(d), (d)+1, (d)+2のいずれかがONします。	986ページ TCMP(P)
TCMPP		

■拡張時計データの読出し

命令記号	処理内容	参照先
S.DATERD	コントローラ内部の時計素子から1/1000秒を含む時計データを読み出します。	988ページ S(P).DATERD
SP.DATERD		

(d)
(d)+1
(d)+2
(d)+3
(d)+4
(d)+5
(d)+6
(d)+7

(d): 年
(d)+1: 月
(d)+2: 日
(d)+3: 時
(d)+4: 分
(d)+5: 秒
(d)+6: 曜日
(d)+7: 1/1000秒

■拡張時計データの加算

命令記号	処理内容	参照先
S.DATE+	時間データを加算します。	990ページ S(P).DATE+
SP.DATE+		

(s1) (s2) (d)

hour
minute
second
—
1/1000 second

+

hour
minute
second
—
1/1000 second

→

hour
minute
second
—
1/1000 second

■拡張時計データの減算

命令記号	処理内容	参照先
S.DATE-	時間データを減算します。	992ページ S(P).DATE-
SP.DATE-		

(s1) (s2) (d)

hour
minute
second
—
1/1000 second

-

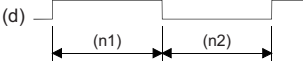
hour
minute
second
—
1/1000 second

→

hour
minute
second
—
1/1000 second

タイミング計測命令

■タイミングパルス発生

命令記号	処理内容	参照先
DUTY	ユーザ用タイミングクロックを指定したスキャン分ONさせ、指定したスキャン分OFFします。  (n1): (n1)スキャン (n2): (n2)スキャン (d): SM420~SM424	994ページ DUTY

■指定データの時間計測

命令記号	処理内容	参照先
TIMCHK	入力条件のON時間を計測し、設定した時間以上連続でONであれば、(d)で指定したデバイスをONにします。	996ページ TIMCHK

ユニットアクセス

ユニットアクセス命令

■I/Oリフレッシュ

命令記号	処理内容	参照先
RFS	1スキャンの途中で該当する入出力の部分リフレッシュを行います。	997ページ RFS(P)
RFSP		

■選択リフレッシュ

命令記号	処理内容	参照先
COM	入力条件成立で各種ユニットのリフレッシュ処理やサービス処理などを行います。	999ページ COM(P)
COMP		

■ユニットリフレッシュ

命令記号	処理内容	参照先
S.ZCOM	指定された各種ユニットのリフレッシュ処理を行います。	1001ページ S(P).ZCOM
SP.ZCOM		

■ユニットからの1ワード/2ワードデータリード(16ビット指定)

命令記号	処理内容	参照先
FROM	インテリジェント機能ユニット内のバッファメモリから、(n)ワード分のデータを16ビット単位で読み出します。	1003ページ FROM(P), DFROM(P)
FROMP		
DFROM	インテリジェント機能ユニット内のバッファメモリから、(n)×2ワード分のデータを16ビット単位で読み出します。	
DFROMP		

■ユニットへの1ワード/2ワードデータライト(16ビット指定)

命令記号	処理内容	参照先
TO	インテリジェント機能ユニットおよび自号機のコントローラ内のバッファメモリに、(n)ワード分のデータを16ビット単位で書き込みます。	1006ページ TO(P), DTO(P)
TOP		
DTO	インテリジェント機能ユニットおよび自号機のコントローラ内のバッファメモリに、(n)×2ワード分のデータを16ビット単位で書き込みます。	
DTOP		

■ユニットからの1ワード/2ワードデータリード(32ビット指定)

命令記号	処理内容	参照先
FROMD	インテリジェント機能ユニット内のバッファメモリから、(n)ワード分のデータを32ビット単位で読み出します。	1011ページ FROMD(P), DFROMD(P)
FROMDP		
DFROMD	インテリジェント機能ユニット内のバッファメモリから、(n)×2ワード分のデータを32ビット単位で読み出します。	
DFROMDP		

■ユニットへの1ワード/2ワードデータライト(32ビット指定)

命令記号	処理内容	参照先
TOD	インテリジェント機能ユニットおよび自号機のコントローラ内のバッファメモリに、(n)ワード分のデータを32ビット単位で書き込みます。	1015ページ TOD(P), DTOD(P)
TODP		
DTOD	インテリジェント機能ユニットおよび自号機のコントローラ内のバッファメモリに、(n)×2ワード分のデータを32ビット単位で書き込みます。	
DTODP		

■ユニット形名読出し

命令記号	処理内容	参照先
TYPERD	(H)で指定した先頭入出力番号のユニット形名を読み出し、(d)で指定したデバイス以降に格納します。	1020ページ TYPERD(P)
TYPERDP		

■ユニット固有情報読出し

命令記号	処理内容	参照先
UNIINFRD	(H)で指定されたユニットから、(n)で指定された点数分のユニット情報を、(d)で指定されたデバイス以降に格納します。	1024ページ UNIINFRD(P)
UNIINFRDP		

PID演算命令

PID演算

命令記号	処理内容	参照先
PID	(s1), (s2), (s3)で設定した値からPID演算を行い、サンプリングタイムごとに演算結果を(d)に格納します。	1043ページ PID

PID制御命令

PID制御命令(不完全微分)

■PID制御用データの設定

命令記号	処理内容	参照先
S.PIDINIT	(s)で指定したデバイス番号以降に設定されている使用ループ数分のPID制御用データを、一括でコントローラ内部に登録し、PID制御を可能にします。	1058ページ S(P).PIDINIT
SP.PIDINIT		

■PID演算

命令記号	処理内容	参照先
S.PIDCONT	サンプリング周期の計測とPID演算を行います。	1061ページ S(P).PIDCONT
SP.PIDCONT		

■指定ループNo.の演算停止/開始

命令記号	処理内容	参照先
S.PIDSTOP	(s)で指定されたループNo.のPID演算を停止します。	1064ページ S(P).PIDSTOP
SP.PIDSTOP		
S.PIDRUN	(s)で指定したループNo.のPID演算を開始します。	1065ページ S(P).PIDRUN
SP.PIDRUN		

■指定ループNo.のパラメータ変更

命令記号	処理内容	参照先
S.PIDPRMW	(s1)で指定したループNo.の演算パラメータを、(s2)で指定したデバイス番号以降に格納されているPID制御用データに変更します。	1066ページ S(P).PIDPRMW
SP.PIDPRMW		

PID制御命令(完全微分)

■PID制御用データの設定

命令記号	処理内容	参照先
PIDINIT	(s)で指定したデバイス番号以降に設定されている使用ループ数分のPID制御用データを、一括でコントローラ内部に登録し、PID制御を可能にします。	1070ページ PIDINIT(P)
PIDINITP		

■PID演算

命令記号	処理内容	参照先
PIDCONT	サンプリング周期の計測とPID演算を行います。	1072ページ PIDCONT(P)
PIDCONT P		

■指定ループNo.の演算停止/開始

命令記号	処理内容	参照先
PIDSTOP	(s)で指定されたループNo.のPID演算を停止します。	1075ページ PIDSTOP(P)
PIDSTOPP		
PIDRUN	(s)で指定したループNo.のPID演算を開始します。	1076ページ PIDRUN(P)
PIDRUNP		

■指定ループNo.のパラメータ変更

命令記号	処理内容	参照先
PIDPRMW	(s1)で指定したループNo.の演算パラメータを、(s2)で指定したデバイス番号以降に格納されているPID制御用データに変更します。	1077ページ PIDPRMW(P)
PIDPRMWP		

アドオン用命令

アドオン実行命令

命令記号	処理内容	参照先
ADDONEXE	インストールされたアドオンの実行を指示します。	1079ページ ADDONEXE(P)
ADDONEXEP		

7 ネットワーク命令一覧

7.1 Ethernet用命令

オープン/クローズ処理命令

■コネクションの確立

命令記号	処理内容	参照先
GP.SOCOPEN	(s1)で指定したコネクションのオープン処理を行います。	1084ページ GP.SOCOPEN, SP.SOCOPEN
SP.SOCOPEN		

■コネクションの切断

命令記号	処理内容	参照先
GP.SOCCLOSE	(s1)で指定したコネクションに対するクローズ処理を行います。(コネクションの切断)	1088ページ GP.SOCCLOSE, SP.SOCCLOSE
SP.SOCCLOSE		

ソケット通信用命令

■受信データのEND処理時読出し

命令記号	処理内容	参照先
GP.SOCCRCV	END処理時に(s1)で指定したコネクションの受信データをソケット通信受信データエリアから読み出します。	1090ページ GP.SOCCRCV, SP.SOCCRCV
SP.SOCCRCV		

■受信データの命令実行時読出し

命令記号	処理内容	参照先
G.SOCCRCVS	命令実行時に(s)で指定したコネクションの受信データをソケット通信受信データエリアから読み出します。	1094ページ G.SOCCRCVS, S.SOCCRCVS
S.SOCCRCVS		

■データ送信

命令記号	処理内容	参照先
GP.SOCSND	(s1)で指定したコネクションの相手機器へ、(s3)で設定したデータを送信します。	1097ページ GP.SOCSND, SP.SOCSND
SP.SOCSND		

■コネクション情報の読出し

命令記号	処理内容	参照先
GP.SOCCINF	(s1)で指定したコネクションのコネクション情報を読み出します。	1101ページ GP.SOCCINF, SP.SOCCINF
SP.SOCCINF		

■コネクションの交信先変更(UDP/IP)

命令記号	処理内容	参照先
GP.SOCCSET	(s1)で指定したコネクションの交信相手IPアドレス、交信相手ポート番号を変更します。 (UDP/IP通信時のみ)	1104ページ GP.SOCCSET, SP.SOCCSET
SP.SOCCSET		

■コネクションの受信モード変更

命令記号	処理内容	参照先
GP.SOCCRMODE	(s1)で指定したコネクションに対して、TCP/TLS受信モードおよび受信データサイズを変更します。	1106ページ GP.SOCCRMODE, SP.SOCCRMODE
SP.SOCCRMODE		

■ソケット通信受信データリード

命令記号	処理内容	参照先
G.SOCRDATA	(s1)で指定したコネクションのソケット通信受信データエリアから、(n)で指定したワード分のデータを読み出し、(d)で指定されたデバイス以降に格納します。	1110ページ G(P).SOCDATA, S(P).SOCDATA
GP.SOCRDATA		
S.SOCRDATA		
SP.SOCRDATA		

通信プロトコル支援機能命令

■登録プロトコル実行

命令記号	処理内容	参照先
GP.ECPRTCL	エンジニアリングツールの通信プロトコル支援機能で設定したプロトコルを実行します。	1112ページ GP.ECPRTCL, SP.ECPRTCL
SP.ECPRTCL		

SLMPフレーム送信命令

■SLMPフレーム送信

命令記号	処理内容	参照先
G.SLMPSND	SLMP対応機器に対してSLMPの伝文を送信します。	1120ページ G(P).SLMPSND, J(P).SLMPSND, SP.SLMPSND
GP.SLMPSND		
J.SLMPSND		
JP.SLMPSND		
SP.SLMPSND		

■暗号化したSLMPフレーム送信

命令記号	処理内容	参照先
G.SLMPSNDC	SLMP対応機器に対して暗号化したSLMPの伝文を送信します。	1129ページ G(P).SLMPSNDC
GP.SLMPSNDC		

ファイル転送機能用命令

■FTPクライアントファイル送付

命令記号	処理内容	参照先
GP.FTPCPUT	(s2)で指定したコントローラのファイルを、(s3)で指定したFTPサーバのフォルダパスへ送付します。	1138ページ GP.FTPCPUT, SP.FTPPUT
SP.FTPPUT		

■FTPクライアントファイル取得

命令記号	処理内容	参照先
GP.FTPCGET	(s2)で指定したFTPサーバのファイルを、(s3)で指定したコントローラのフォルダパスに取得します。	1144ページ GP.FTPCGET, SP.FTPGET
SP.FTPGET		

8 汎用FUN/汎用FB一覧

一覧表の見方を示します。

項目	内容
ファンクション記号, ファンクションブロック記号	ファンクション, ファンクションブロック名を示します。
処理内容	ファンクション, ファンクションブロックの概要を示します。
参照先	詳細説明の参照先を示します。

8.1 汎用ファンクション

型変換ファンクション

■BOOL型→WORD型/DWORD型変換

ファンクション記号	処理内容	参照先
BOOL_TO_WORD	BOOL型データをWORD型データへ変換します。	1152ページ BOOL_TO_WORD(_E)
BOOL_TO_WORD_E		
BOOL_TO_DWORD	BOOL型データをDWORD型データへ変換します。	1154ページ BOOL_TO_DWORD(_E)
BOOL_TO_DWORD_E		

■BOOL型→INT型/DINT型変換

ファンクション記号	処理内容	参照先
BOOL_TO_INT	BOOL型データをINT型データへ変換します。	1155ページ BOOL_TO_INT(_E)
BOOL_TO_INT_E		
BOOL_TO_DINT	BOOL型データをDINT型データへ変換します。	1156ページ BOOL_TO_DINT(_E)
BOOL_TO_DINT_E		

■BOOL型→TIME型変換

ファンクション記号	処理内容	参照先
BOOL_TO_TIME	BOOL型データをTIME型データへ変換します。	1157ページ BOOL_TO_TIME(_E)
BOOL_TO_TIME_E		

■BOOL型→STRING型変換

ファンクション記号	処理内容	参照先
BOOL_TO_STRING	BOOL型データをSTRING型データへ変換します。	1158ページ BOOL_TO_STRING(_E)
BOOL_TO_STRING_E		

■WORD型→BOOL型変換

ファンクション記号	処理内容	参照先
WORD_TO_BOOL	WORD型データをBOOL型データへ変換します。	1159ページ WORD_TO_BOOL(_E)
WORD_TO_BOOL_E		

■WORD型→DWORD型変換

ファンクション記号	処理内容	参照先
WORD_TO_DWORD	WORD型データをDWORD型データへ変換します。	1160ページ WORD_TO_DWORD(_E)
WORD_TO_DWORD_E		

■WORD型→INT型/DINT型変換

ファンクション記号	処理内容	参照先
WORD_TO_INT	WORD型データをINT型データへ変換します。	1161ページ WORD_TO_INT(_E)
WORD_TO_INT_E		
WORD_TO_DINT	WORD型データをDINT型データへ変換します。	1162ページ WORD_TO_DINT(_E)
WORD_TO_DINT_E		

■WORD型→TIME型変換

ファンクション記号	処理内容	参照先
WORD_TO_TIME	WORD型データをTIME型データへ変換します。	1164ページ WORD_TO_TIME(_E)
WORD_TO_TIME_E		

■WORD型→STRING型変換

ファンクション記号	処理内容	参照先
WORD_TO_STRING	WORD型データをSTRING型データへ変換します。	1165ページ WORD_TO_STRING(_E)
WORD_TO_STRING_E		

■DWORD型→BOOL型変換

ファンクション記号	処理内容	参照先
DWORD_TO_BOOL	DWORD型データをBOOL型データへ変換します。	1166ページ DWORD_TO_BOOL(_E)
DWORD_TO_BOOL_E		

■DWORD型→WORD型変換

ファンクション記号	処理内容	参照先
DWORD_TO_WORD	DWORD型データをWORD型データへ変換します。	1167ページ DWORD_TO_WORD(_E)
DWORD_TO_WORD_E		

■DWORD型→INT型/DINT型変換

ファンクション記号	処理内容	参照先
DWORD_TO_INT	DWORD型データをINT型データへ変換します。	1169ページ DWORD_TO_INT(_E)
DWORD_TO_INT_E		
DWORD_TO_DINT	DWORD型データをDINT型データへ変換します。	1171ページ DWORD_TO_DINT(_E)
DWORD_TO_DINT_E		

■DWORD型→TIME型変換

ファンクション記号	処理内容	参照先
DWORD_TO_TIME	DWORD型データをTIME型データへ変換します。	1172ページ DWORD_TO_TIME(_E)
DWORD_TO_TIME_E		

■DWORD型→STRING型変換

ファンクション記号	処理内容	参照先
DWORD_TO_STRING	DWORD型データをSTRING型データへ変換します。	1173ページ DWORD_TO_STRING(_E)
DWORD_TO_STRING_E		

■INT型→BOOL型変換

ファンクション記号	処理内容	参照先
INT_TO_BOOL	INT型データをBOOL型データへ変換します。	1174ページ INT_TO_BOOL(_E)
INT_TO_BOOL_E		

■INT型→WORD型/DWORD型変換

ファンクション記号	処理内容	参照先
INT_TO_WORD	INT型データをWORD型データへ変換します。	1175ページ INT_TO_WORD(_E)
INT_TO_WORD_E		
INT_TO_DWORD	INT型データをDWORD型データへ変換します。	1176ページ INT_TO_DWORD(_E)
INT_TO_DWORD_E		

■INT型→DINT型変換

ファンクション記号	処理内容	参照先
INT_TO_DINT	INT型データをDINT型データへ変換します。	1178ページ INT_TO_DINT(_E)
INT_TO_DINT_E		

■INT型→BCD型変換

ファンクション記号	処理内容	参照先
INT_TO_BCD	INT型データをBCD型データへ変換します。	1179ページ INT_TO_BCD(_E)
INT_TO_BCD_E		

■INT型→REAL型/LREAL型変換

ファンクション記号	処理内容	参照先
INT_TO_REAL	INT型データをREAL型データへ変換します。	1181ページ INT_TO_REAL(_E)
INT_TO_REAL_E		
INT_TO_LREAL	INT型データをLREAL型データへ変換します。	1182ページ INT_TO_LREAL(_E)
INT_TO_LREAL_E		

■INT型→TIME型変換

ファンクション記号	処理内容	参照先
INT_TO_TIME	INT型データをTIME型データへ変換します。	1183ページ INT_TO_TIME(_E)
INT_TO_TIME_E		

■INT型→STRING型変換

ファンクション記号	処理内容	参照先
INT_TO_STRING	INT型データをSTRING型データへ変換します。	1184ページ INT_TO_STRING(_E)
INT_TO_STRING_E		

■DINT型→BOOL型変換

ファンクション記号	処理内容	参照先
DINT_TO_BOOL	DINT型データをBOOL型データへ変換します。	1186ページ DINT_TO_BOOL(_E)
DINT_TO_BOOL_E		

■DINT型→WORD型/DWORD型変換

ファンクション記号	処理内容	参照先
DINT_TO_WORD	DINT型データをWORD型データへ変換します。	1187ページ DINT_TO_WORD(_E)
DINT_TO_WORD_E		
DINT_TO_DWORD	DINT型データをDWORD型データへ変換します。	1189ページ DINT_TO_DWORD(_E)
DINT_TO_DWORD_E		

■DINT型→INT型変換

ファンクション記号	処理内容	参照先
DINT_TO_INT	DINT型データをINT型データへ変換します。	1190ページ DINT_TO_INT(_E)
DINT_TO_INT_E		

■DINT型→BCD型変換

ファンクション記号	処理内容	参照先
DINT_TO_BCD	DINT型データをBCD型データへ変換します。	1191ページ DINT_TO_BCD(_E)
DINT_TO_BCD_E		

■DINT型→REAL型/LREAL型変換

ファンクション記号	処理内容	参照先
DINT_TO_REAL	DINT型データをREAL型データへ変換します。	1193ページ DINT_TO_REAL(_E)
DINT_TO_REAL_E		
DINT_TO_LREAL	DINT型データをLREAL型データへ変換します。	1194ページ DINT_TO_LREAL(_E)
DINT_TO_LREAL_E		

■DINT型→TIME型変換

ファンクション記号	処理内容	参照先
DINT_TO_TIME	DINT型データをTIME型データへ変換します。	1195ページ DINT_TO_TIME(_E)
DINT_TO_TIME_E		

■DINT型→STRING型変換

ファンクション記号	処理内容	参照先
DINT_TO_STRING	DINT型データをSTRING型データへ変換します。	1196ページ DINT_TO_STRING(_E)
DINT_TO_STRING_E		

■BCD型→INT型/DINT型変換

ファンクション記号	処理内容	参照先
BCD_TO_INT	BCD型データをINT型データへ変換します。	1198ページ BCD_TO_INT(_E)
BCD_TO_INT_E		
BCD_TO_DINT	BCD型データをDINT型データへ変換します。	1200ページ BCD_TO_DINT(_E)
BCD_TO_DINT_E		

■BCD型→STRING型変換

ファンクション記号	処理内容	参照先
BCD_TO_STRING	BCD型データをSTRING型データへ変換します。	1203ページ BCD_TO_STRING(_E)
BCD_TO_STRING_E		

■REAL型→INT型/DINT型変換

ファンクション記号	処理内容	参照先
REAL_TO_INT	REAL型データをINT型データへ変換します。	1205ページ REAL_TO_INT(_E)
REAL_TO_INT_E		
REAL_TO_DINT	REAL型データをDINT型データへ変換します。	1207ページ REAL_TO_DINT(_E)
REAL_TO_DINT_E		

■REAL型→LREAL型変換

ファンクション記号	処理内容	参照先
REAL_TO_LREAL	REAL型データをLREAL型データへ変換します。	1209ページ REAL_TO_LREAL(_E)
REAL_TO_LREAL_E		

■REAL型→STRING型変換

ファンクション記号	処理内容	参照先
REAL_TO_STRING	REAL型データをSTRING型(指数型式)データへ変換します。	1211ページ REAL_TO_STRING(_E)
REAL_TO_STRING_E		

■LREAL型→INT型/DINT型変換

ファンクション記号	処理内容	参照先
LREAL_TO_INT	LREAL型データをINT型データへ変換します。	1214ページ LREAL_TO_INT(_E)
LREAL_TO_INT_E		
LREAL_TO_DINT	LREAL型データをDINT型データへ変換します。	1216ページ LREAL_TO_DINT(_E)
LREAL_TO_DINT_E		

■LREAL型→REAL型変換

ファンクション記号	処理内容	参照先
LREAL_TO_REAL	LREAL型データをREAL型データへ変換します。	1218ページ LREAL_TO_REAL(_E)
LREAL_TO_REAL_E		

■TIME型→BOOL型変換

ファンクション記号	処理内容	参照先
TIME_TO_BOOL	TIME型データをBOOL型データへ変換します。	1220ページ TIME_TO_BOOL(_E)
TIME_TO_BOOL_E		

■TIME型→WORD型/DWORD型変換

ファンクション記号	処理内容	参照先
TIME_TO_WORD	TIME型データをWORD型データへ変換します。	1221ページ TIME_TO_WORD(_E)
TIME_TO_WORD_E		
TIME_TO_DWORD	TIME型データをDWORD型データへ変換します。	1222ページ TIME_TO_DWORD(_E)
TIME_TO_DWORD_E		

■TIME型→INT型/DINT型変換

ファンクション記号	処理内容	参照先
TIME_TO_INT	TIME型データをINT型データへ変換します。	1223ページ TIME_TO_INT(_E)
TIME_TO_INT_E		
TIME_TO_DINT	TIME型データをDINT型データへ変換します。	1224ページ TIME_TO_DINT(_E)
TIME_TO_DINT_E		

■TIME型→STRING型変換

ファンクション記号	処理内容	参照先
TIME_TO_STRING	TIME型データをSTRING型データへ変換します。	1225ページ TIME_TO_STRING(_E)
TIME_TO_STRING_E		

■STRING型→BOOL型変換

ファンクション記号	処理内容	参照先
STRING_TO_BOOL	STRING型データをBOOL型データへ変換します。	1227ページ STRING_TO_BOOL(_E)
STRING_TO_BOOL_E		

■STRING型→WORD型/DWORD型変換

ファンクション記号	処理内容	参照先
STRING_TO_WORD	STRING型データをWORD型データへ変換します。	1228ページ STRING_TO_WORD(_E)
STRING_TO_WORD_E		
STRING_TO_DWORD	STRING型データをDWORD型データへ変換します。	1229ページ STRING_TO_DWORD(_E)
STRING_TO_DWORD_E		

■STRING型→INT型/DINT型変換

ファンクション記号	処理内容	参照先
STRING_TO_INT	STRING型データをINT型データへ変換します。	1230ページ STRING_TO_INT(_E)
STRING_TO_INT_E		
STRING_TO_DINT	STRING型データをDINT型データへ変換します。	1232ページ STRING_TO_DINT(_E)
STRING_TO_DINT_E		

■STRING型→BCD型変換

ファンクション記号	処理内容	参照先
STRING_TO_BCD	STRING型データをBCD型データへ変換します。	1234ページ STRING_TO_BCD(_E)
STRING_TO_BCD_E		

■STRING型→REAL型変換

ファンクション記号	処理内容	参照先
STRING_TO_REAL	STRING型データをREAL型データへ変換します。	1236ページ STRING_TO_REAL(_E)
STRING_TO_REAL_E		

■STRING型→TIME型変換

ファンクション記号	処理内容	参照先
STRING_TO_TIME	STRING型データをTIME型データへ変換します。	1239ページ STRING_TO_TIME(_E)
STRING_TO_TIME_E		

■ビット配列→INT型/DINT型変換

ファンクション記号	処理内容	参照先
BITARR_TO_INT	ビット配列から指定のビット数をINT型データへ変換します。	1241ページ BITARR_TO_INT(_E)
BITARR_TO_INT_E		
BITARR_TO_DINT	ビット配列から指定のビット数をDINT型データへ変換します。	1242ページ BITARR_TO_DINT(_E)
BITARR_TO_DINT_E		

■INT型/DINT型→ビット配列変換

ファンクション記号	処理内容	参照先
INT_TO_BITARR	INT型データの低位nビットをビット配列に出力します。	1243ページ INT_TO_BITARR(_E)
INT_TO_BITARR_E		
DINT_TO_BITARR	DINT型データの低位nビットをビット配列に出力します。	1244ページ DINT_TO_BITARR(_E)
DINT_TO_BITARR_E		

■ビット配列のコピー

ファンクション記号	処理内容	参照先
CPY_BITARR	ビット配列を指定ビットサイズ分コピーします。	1245ページ CPY_BITARR(_E)
CPY_BITARR_E		

■ワードラベルの指定ビット読出し/書込み/コピー

ファンクション記号	処理内容	参照先
GET_BIT_OF_INT	ワードラベルの指定ビットを読み出します。	1246ページ GET_BIT_OF_INT(_E)
GET_BIT_OF_INT_E		
SET_BIT_OF_INT	ワードラベルの指定ビットに書き込みます。	1248ページ SET_BIT_OF_INT(_E)
SET_BIT_OF_INT_E		
CPY_BIT_OF_INT	ワードラベルの指定ビットを別のワードラベルの指定ビットにコピーします。	1250ページ CPY_BIT_OF_INT(_E)
CPY_BIT_OF_INT_E		

■先頭データの取得

ファンクション記号	処理内容	参照先
GET_BOOL_ADDR	指定したデータの先頭データをBOOL型、INT型、またはWORD型データとして出力します。	1252ページ GET_BOOL_ADDR, GET_INT_ADDR, GET_WORD_ADDR
GET_INT_ADDR		
GET_WORD_ADDR		

単数値変数ファンクション

■絶対値

ファンクション記号	処理内容	参照先
ABS	入力値の絶対値を出力します。	1253ページ ABS(_E)
ABS_E		

■平方根

ファンクション記号	処理内容	参照先
SQRT	入力値の平方根を演算します。	1255ページ SQRT(_E)
SQRT_E		

■自然対数演算

ファンクション記号	処理内容	参照先
LN	入力値の自然対数演算結果を出力します。	1256ページ LN(_E)
LN_E		

■常用対数演算

ファンクション記号	処理内容	参照先
LOG	入力値の常用対数(10を底とした対数)の演算結果を出力します。	1257ページ LOG(_E)
LOG_E		

■指数演算

ファンクション記号	処理内容	参照先
EXP	入力値の指数演算結果を出力します。	1259ページ EXP(_E)
EXP_E		

■SIN演算/COS演算/TAN演算

ファンクション記号	処理内容	参照先
SIN	入力値のSIN(正弦)値を出力します。	1260ページ SIN(_E)
SIN_E		
COS	入力値のCOS(余弦)値を出力します。	1261ページ COS(_E)
COS_E		
TAN	入力値のTAN(正接)値を出力します。	1262ページ TAN(_E)
TAN_E		

■SIN⁻¹演算/COS⁻¹演算/TAN⁻¹演算

ファンクション記号	処理内容	参照先
ASIN	入力値のSIN ⁻¹ (逆正弦)値を出力します。	1263ページ ASIN(_E)
ASIN_E		
ACOS	入力値のCOS ⁻¹ (逆余弦)値を出力します。	1264ページ ACOS(_E)
ACOS_E		
ATAN	入力値のTAN ⁻¹ (逆正接)値を出力します。	1265ページ ATAN(_E)
ATAN_E		

算術演算ファンクション

■加算

ファンクション記号	処理内容	参照先
ADD	入力値の和($(s1)+(s2)+\dots+(s28)$)を出力します。	1266ページ ADD(_E)
ADD_E		

■乗算

ファンクション記号	処理内容	参照先
MUL	入力値の積($(s1)\times(s2)\times\dots\times(s28)$)を出力します。	1269ページ MUL(_E)
MUL_E		

■減算

ファンクション記号	処理内容	参照先
SUB	入力値の差($(s1)-(s2)$)を出力します。	1271ページ SUB(_E)
SUB_E		

■除算

ファンクション記号	処理内容	参照先
DIV	入力値の商($(s1)\div(s2)$)を出力します。	1274ページ DIV(_E)
DIV_E		

■剰余

ファンクション記号	処理内容	参照先
MOD	入力値の余り($(s1)\div(s2)$)を出力します。	1276ページ MOD(_E)
MOD_E		

■べき乗

ファンクション記号	処理内容	参照先
EXPT	入力値のべき乗を出力します。	1278ページ EXPT(_E)
EXPT_E		

■代入

ファンクション記号	処理内容	参照先
MOVE	入力値の代入を出力します。	1279ページ MOVE(_E)
MOVE_E		

ビットシフトファンクション

■nビット左シフト, 右シフト

ファンクション記号	処理内容	参照先
SHL	入力値を(n)ビット数分左へシフトして出力します。	1281ページ SHL(_E)
SHL_E		
SHR	入力値を(n)ビット数分右へシフトして出力します。	1283ページ SHR(_E)
SHR_E		

■nビット左ローテーション, 右ローテーション

ファンクション記号	処理内容	参照先
ROL	入力値を(n)ビット数分左へ回転して出力します。	1285ページ ROL(_E)
ROL_E		
ROR	入力値を(n)ビット数分右へ回転して出力します。	1287ページ ROR(_E)
ROR_E		

ビット型ブールファンクション

■論理積，論理和，排他的論理和

ファンクション記号	処理内容	参照先
AND	入力値の論理積を出力します。	1289ページ AND(_E), OR(_E), XOR(_E)
AND_E		
OR	入力値の論理和を出力します。	
OR_E		
XOR	入力値の排他的論理和を出力します。	
XOR_E		

■論理否定

ファンクション記号	処理内容	参照先
NOT	入力値の論理否定を出力します。	1292ページ NOT(_E)
NOT_E		

選択ファンクション

■選択値

ファンクション記号	処理内容	参照先
SEL	選択した入力値を出力します。	1293ページ SEL(_E)
SEL_E		

■最大値，最小値選択

ファンクション記号	処理内容	参照先
MAX	入力値の最大値を出力します。	1295ページ MAX(_E), MIN(_E)
MAX_E		
MIN	入力値の最小値を出力します。	
MIN_E		

■上下限リミット制御

ファンクション記号	処理内容	参照先
LIMIT	上下限リミット制御された入力値を出力します。	1297ページ LIMIT(_E)
LIMIT_E		

■マルチプレクサ

ファンクション記号	処理内容	参照先
MUX	複数入力値の一つを出力します。	1300ページ MUX(_E)
MUX_E		

比較ファンクション

■比較

ファンクション記号	処理内容	参照先
GT	入力値のデータ比較結果を出力します。	1302ページ GT(_E), GE(_E), EQ(_E), LE(_E), LT(_E)
GT_E		
GE		
GE_E		
EQ		
EQ_E		
LE		
LE_E		
LT		
LT_E		
NE		1304ページ NE(_E)
NE_E		

文字列ファンクション

■文字列の長さ検出

ファンクション記号	処理内容	参照先
LEN	入力された文字列の長さを検出して出力します。	1306ページ LEN(_E)
LEN_E		

■文字列の左側, 右側からの抽出

ファンクション記号	処理内容	参照先
LEFT	入力された文字列データの左側から指定文字分を出力します。	1308ページ LEFT(_E), RIGHT(_E)
LEFT_E		
RIGHT	入力された文字列データの右側から指定文字分を出力します。	
RIGHT_E		

■文字列の抽出

ファンクション記号	処理内容	参照先
MID	入力された文字列の任意位置から指定文字分を出力します。	1311ページ MID(_E)
MID_E		

■文字列の結合

ファンクション記号	処理内容	参照先
CONCAT	文字列を結合して出力します。	1313ページ CONCAT(_E)
CONCAT_E		

■文字列の挿入

ファンクション記号	処理内容	参照先
INSERT	文字列の間へ文字列を挿入して出力します。	1315ページ INSERT(_E)
INSERT_E		

■文字列の削除

ファンクション記号	処理内容	参照先
DELETE	文字列の任意の範囲を削除して出力します。	1317ページ DELETE(_E)
DELETE_E		

■文字列の置換

ファンクション記号	処理内容	参照先
REPLACE	文字列の任意の範囲を置換して出力します。	1319ページ REPLACE(_E)
REPLACE_E		

■文字列の検索

ファンクション記号	処理内容	参照先
FIND	文字列を検索して検索結果を出力します。	1322ページ FIND(_E)
FIND_E		

時刻データ型ファンクション

■加算

ファンクション記号	処理内容	参照先
ADD_TIME	入力値(TIME型)の和((s1)+(s2))を出力します。	1324ページ ADD_TIME(_E)
ADD_TIME_E		

■減算

ファンクション記号	処理内容	参照先
SUB_TIME	入力値(TIME型)の差((s1)-(s2))を出力します。	1326ページ SUB_TIME(_E)
SUB_TIME_E		

■乗算

ファンクション記号	処理内容	参照先
MUL_TIME	入力値(TIME型)の積((s1)×(s2))を出力します。	1328ページ MUL_TIME(_E)
MUL_TIME_E		

■除算

ファンクション記号	処理内容	参照先
DIV_TIME	入力値(TIME型)の商((s1)÷(s2))を出力します。	1330ページ DIV_TIME(_E)
DIV_TIME_E		

8.2 汎用ファンクションブロック

■二安定ファンクションブロック

■二安定ファンクションブロック(セット優先)

ファンクションブロック 記号	処理内容	参照先
SR	2つの入力値を判別し、1(TRUE)または0(FALSE)を出力します。	1334ページ SR(_E)
SR_E		

■二安定ファンクションブロック(リセット優先)

ファンクションブロック 記号	処理内容	参照先
RS	2つの入力値を判別し、1(TRUE)または0(FALSE)を出力します。	1336ページ RS(_E)
RS_E		

■エッジ検出ファンクションブロック

■立ち上りエッジ検出

ファンクションブロック 記号	処理内容	参照先
R_TRIG	信号の立ち上がりを検出してパルス信号を出力します。	1338ページ R_TRIG(_E)
R_TRIG_E		

■立ち下りエッジ検出

ファンクションブロック 記号	処理内容	参照先
F_TRIG	信号の立ち下りを検出してパルス信号を出力します。	1340ページ F_TRIG(_E)
F_TRIG_E		

■カウンタファンクションブロック

■アップカウンタ

ファンクションブロック 記号	処理内容	参照先
CTU	信号の立ち上がり回数をカウントアップします。	1342ページ CTU(_E)
CTU_E		

■ダウンカウンタ

ファンクションブロック 記号	処理内容	参照先
CTD	信号の立ち上がり回数をカウントダウンします。	1344ページ CTD(_E)
CTD_E		

■アップダウンカウンタ

ファンクションブロック 記号	処理内容	参照先
CTUD	信号の立ち上がり回数をカウントアップ/カウントダウンします。	1346ページ CTUD(_E)
CTUD_E		

■カウンタファンクションブロック

ファンクションブロック 記号	処理内容	参照先
COUNTER_FB_M	実行条件が成立すると、カウントアップを実行します。	1349ページ COUNTER_FB_M

■タイマファンクションブロック

■パルスタイマ

ファンクションブロック 記号	処理内容	参照先
TP	指定された時間の間、信号をONします。	1351ページ TP(_E)
TP_E		

■オンディレイタイマ

ファンクションブロック 記号	処理内容	参照先
TON	指定された時間後に信号をONします。	1354ページ TON(_E)
TON_E		

■オフディレイタイマ

ファンクションブロック 記号	処理内容	参照先
TOF	指定された時間後に信号をOFFします。	1356ページ TOF(_E)
TOF_E		

■タイマファンクションブロック

ファンクションブロック 記号	処理内容	参照先
TIMER_10_FB_M	実行条件が成立すると、設定された時間までタイマカウントを実行します。	1358ページ TIMER_□_M
TIMER_100_FB_M		
TIMER_HIGH_FB_M		
TIMER_LOW_FB_M		
TIMER_CONT_FB_M		
TIMER_CONTHFB_M		

一覧の見方を説明します。

項目	内容
ファンクションブロック記号	ファンクションブロック名を示します。
処理内容	ファンクションブロックの概要を示します。
参照先	詳細説明の参照先を示します。

モーション制御ファンクションブロックの概要は下記を参照してください。

📖 1364ページ モーション制御FBの概要

9.1 管理系ファンクションブロック

管理系のモーション制御ファンクションブロック一覧を下記に示します。

軸グループ有効

ファンクションブロック記号	処理内容	参照
MC_GroupEnable	指定した軸グループの状態を「0: 軸グループ無効(GroupDisabled)」から「4: 待機中(GroupStandby)」に遷移させます。	📖 1456ページ 軸グループ有効

軸グループ無効

ファンクションブロック記号	処理内容	参照
MC_GroupDisable	指定した軸グループの状態を「0: 軸グループ無効(GroupDisabled)」に遷移させます。	📖 1460ページ 軸グループ無効

運転可

ファンクションブロック記号	処理内容	参照
MC_Power	指定した軸を運転可能状態に切替えます。	📖 1463ページ 運転可

現在位置変更

ファンクションブロック記号	処理内容	参照
MC_SetPosition	指定した軸の現在位置(指令位置, フィードバック位置)を変更します。	📖 1467ページ 現在位置変更

トルク制限値

ファンクションブロック記号	処理内容	参照
MCv_SetTorqueLimit	トルク制限値の変更を実行します。	📖 1475ページ トルク制限値

オーバーライド値設定

ファンクションブロック記号	処理内容	参照
MC_SetOverride	指定した軸の目標速度, 目標加速度, 目標減速度の変更を実行します。	📖 1483ページ オーバライド値設定

パラメータ読出

ファンクションブロック 記号	処理内容	参照
MC_ReadParameter	デバイス機器のオブジェクトの読出しを行います。	📖 1487ページ パラメータ 読出

パラメータ書込

ファンクションブロック 記号	処理内容	参照
MC_WriteParameter	デバイス機器のオブジェクトの書込みを行います。	📖 1494ページ パラメータ 書込

軸エラーリセット

ファンクションブロック 記号	処理内容	参照
MC_Reset	軸のエラー，警告をリセットします。	📖 1500ページ 軸エラーリ セット

軸グループエラーリセット

ファンクションブロック 記号	処理内容	参照
MC_GroupReset	軸グループおよび軸グループに属する各軸のエラー，警告をリセットします。	📖 1503ページ 軸グループ エラーリセット

タッチプローブ有効

ファンクションブロック 記号	処理内容	参照
MC_TouchProbe	トリガイメント発生により任意のデータを記録します。	📖 1507ページ タッチプ ローブ有効

タッチプローブ無効

ファンクションブロック 記号	処理内容	参照
MC_AbortTrigger	実行中のラッチを無効にします。	📖 1522ページ タッチプ ローブ無効

カムテーブル選択

ファンクションブロック 記号	処理内容	参照
MC_CamTableSelect	指定した演算プロファイル(カムデータ)を，展開エリアに格納します。	📖 1527ページ カムテーブ ル選択

1サイクル現在値変更

ファンクションブロック 記号	処理内容	参照
MCv_ChangeCycle	指定した演算プロファイル制御FBの1サイクル現在値を変更します。	📖 1534ページ 1サイクル現 在値変更

全軸運転可

ファンクションブロック 記号	処理内容	参照
MCv_AllPower	すべての軸を運転可能状態に切替えます。	1542ページ 全軸運転可

軸グループオーバーライド値設定

ファンクションブロック 記号	処理内容	参照
MC_GroupSetOverride	指定した軸グループの目標速度，目標加速度，目標減速度の変更を実行します。	1545ページ 軸グループ オーバーライド値設定

モーションエラーリセット

ファンクションブロック 記号	処理内容	参照
MCv_MotionErrorReset	モーションシステムのすべてのエラー，警告をリセットします。	1549ページ モーション エラーリセット

制御演算周期同期

ファンクションブロック 記号	処理内容	参照
MCv_SyncOperationCycles	軸の制御演算周期を設定した周期と同期します。	1553ページ 制御演算周 期同期

アドバンスト同期制御1サイクル現在位置計算

ファンクションブロック 記号	処理内容	参照
MCv_AdvPositionPerCycleCalc	指定したカムデータをもとに1サイクル現在位置を算出します。	1556ページ アドバンス ト同期制御1サイクル現在位 置計算

アドバンスト同期制御カム指令現在位置計算

ファンクションブロック 記号	処理内容	参照
MCv_AdvCamSetPositionCalc	指定したカムデータをもとにカム指令現在位置を算出します。	1559ページ アドバンス ト同期制御カム指令現在位置 計算

デジタルカムスイッチ出力

ファンクションブロック 記号	処理内容	参照
MC_DigitalCamSwitch	任意データの値に応じてON/OFF信号を出力します。	1562ページ デジタルカ ムスイッチ出力

9.2 動作系ファンクションブロック

動作系のモーション制御ファンクションブロック一覧を下記に示します。

原点復帰

ファンクションブロック 記号	処理内容	参照
MC_Home	指定した軸の原点復帰を行います。	 1569ページ 原点復帰

強制停止

ファンクションブロック 記号	処理内容	参照
MC_Stop	指定した軸を減速停止します。	 1579ページ 強制停止

グループ強制停止

ファンクションブロック 記号	処理内容	参照
MC_GroupStop	指定した軸グループを減速停止します。	 1584ページ グループ強制停止

絶対値位置決め

ファンクションブロック 記号	処理内容	参照
MC_MoveAbsolute	絶対位置の目標位置を指定し、位置決めを実行します。	 1590ページ 絶対値位置決め

相対値位置決め

ファンクションブロック 記号	処理内容	参照
MC_MoveRelative	相対位置の移動量を指定し、位置決めを実行します。	 1602ページ 相対値位置決め

JOG運転

ファンクションブロック 記号	処理内容	参照
MCv_Jog	指令速度に従いJOG運転を実行します。	 1610ページ JOG運転

速度制御

ファンクションブロック 記号	処理内容	参照
MC_MoveVelocity	ドライバをcsvに切換え、指定した速度に従い速度制御を行います。	 1618ページ 速度制御

トルク制御

ファンクションブロック 記号	処理内容	参照
MC_TorqueControl	ドライバをcstに切換え、指定した目標トルクに従いトルク制御を行います。	📖 1625ページ トルク制御

速度制御(位置ループを含む)

ファンクションブロック 記号	処理内容	参照
MCV_SpeedControl	位置ループを含む速度制御を実行します。	📖 1635ページ 速度制御(位置ループを含む)

絶対値直線補間制御

ファンクションブロック 記号	処理内容	参照
MCV_MoveLinearInterpolateAbsolute	指定した軸グループの絶対位置による目標位置を指定し、直線補間制御による位置決めを実行します。	📖 1643ページ 絶対値直線補間制御

相対値直線補間制御

ファンクションブロック 記号	処理内容	参照
MCV_MoveLinearInterpolateRelative	指定した軸グループの相対位置による移動量を指定し、直線補間制御による位置決めを実行します。	📖 1658ページ 相対値直線補間制御

絶対値円弧補間制御

ファンクションブロック 記号	処理内容	参照
MCV_MoveCircularInterpolateAbsolute	設定した軸グループの構成軸を用いて、絶対位置の終点、および補助点を設定し、2軸の円弧補間による位置決めを実行します。	📖 1671ページ 絶対値円弧補間制御

相対値円弧補間制御

ファンクションブロック 記号	処理内容	参照
MCV_MoveCircularInterpolateRelative	設定した軸グループの構成軸を用いて、始動時の現在位置から終点および補助点へ相対位置を設定し、2軸の円弧補間による位置決めを実行します。	📖 1688ページ 相対値円弧補間制御

カム動作開始

ファンクションブロック 記号	処理内容	参照
MC_CamIn	指定したカムデータに従ってカム動作を開始します。	📖 1705ページ カム動作開始

ギア動作開始

ファンクションブロック 記号	処理内容	参照
MC_GearIn	指定したギア比に従いギア動作を開始します。	📖 1732ページ ギア動作開始

加減算位置決め

ファンクションブロック 記号	処理内容	参照
MC_CombineAxes	指定した主軸2軸の移動量を加減算した値を指令位置とし、位置決めを行います。	☞ 1742ページ 加減算位置 決め

バックラッシュ補正フィルタ

ファンクションブロック 記号	処理内容	参照
MCv_BacklashCompensationFilter	移動方向に従い、機械系のバックラッシュ量を補正するフィルタ処理を行います。	☞ 1752ページ バックラッ シュ補正フィルタ

スムージングフィルタ

ファンクションブロック 記号	処理内容	参照
MCv_SmoothingFilter	指定した周波数に従い、フィルタ処理を行います。	☞ 1765ページ スムージン グフィルタ

移動方向制限フィルタ

ファンクションブロック 記号	処理内容	参照
MCv_DirectionFilter	設定した移動方向に対する移動を制限するフィルタ処理を行います。	☞ 1774ページ 移動方向制 限フィルタ

速度制限フィルタ

ファンクションブロック 記号	処理内容	参照
MCv_SpeedLimitFilter	設定した制限値の速度に制限するフィルタ処理を行います。	☞ 1782ページ 速度制限 フィルタ

軸制御開始ウェイト

ファンクションブロック 記号	処理内容	参照
MCv_MoveWait	指定した軸の多重起動した動作系FBの動作を、実行許可信号の入力により即時実行します。	☞ 1790ページ 軸制御開始 ウェイト

軸グループ制御開始ウェイト

ファンクションブロック 記号	処理内容	参照
MCv_GroupMoveWait	指定した軸グループの多重起動した動作系FBの動作を、実行許可信号の入力により即時実行します。	☞ 1797ページ 軸グループ 制御開始ウェイト

アドバンスト同期制御

ファンクションブロック 記号	処理内容	参照
MCv_AdvancedSync	指定したアドバンスト同期制御設定に従って同期制御を開始します。	☞ 1804ページ アドバンス ト同期制御

多軸位置決めデータ運転

ファンクションブロック 記号	処理内容	参照
MCv_MovePositioningData	指定した位置決めデータに従い、軸グループを動作します。	1810ページ 多軸位置決めデータ運転

絶対位置追従制御

ファンクションブロック 記号	処理内容	参照
MCv_PositionControl	指定された軸を制限速度、加速度、減速度、ジャークに従い目標位置まで移動させる機能です。また、指令が目標位置へ到達しても、再起動、連続更新で目標位置を変更し軸を移動できます。	1819ページ 絶対位置追従制御

モーションサイクリック位置制御

ファンクションブロック 記号	処理内容	参照
MCv_CyclicPosition	指定した目標位置に従いドライブユニットに目標位置を送信します。	1830ページ モーションサイクリック位置制御

モーションサイクリック速度制御

ファンクションブロック 記号	処理内容	参照
MCv_CyclicVelocity	指定した目標速度に従いドライブユニットに目標速度を送信します。	1838ページ モーションサイクリック速度制御

モーションサイクリックトルク制御

ファンクションブロック 記号	処理内容	参照
MCv_CyclicTorque	指定した目標位置に従いドライブユニットに目標トルクを送信します。	1842ページ モーションサイクリックトルク制御

9.3 一般ファンクションブロック

軸制御を行うためのモーション制御ファンクションブロック一覧を下記に示します。

プロフィール読出

ファンクションブロック 記号	処理内容	参照
MCv_ReadProfileData	指定した演算プロフィールデータを、展開エリアまたはファイルから読み出します。	📖 1848ページ プロファイル読出

プロフィール書込

ファンクションブロック 記号	処理内容	参照
MCv_WriteProfileData	指定した演算プロフィールデータを、展開エリアまたはファイルに書き込みます。	📖 1862ページ プロファイル書込

10 ネットワークファンクションブロック一覧

10

一覧の見方を説明します。

項目	内容
ファンクションブロック記号	ファンクションブロック名を示します。
処理内容	ファンクションブロックの概要を示します。
参照先	詳細説明の参照先を示します。

10.1 Ethernet用ファンクションブロック

コネクションの確立/切断

ファンクションブロック記号	処理内容	参照
M+MX-SQ_ConnectionOpen	コネクションをオープン(確立)します。	☞ 1876ページ M+MX-SQ_ConnectionOpen
M+MX-SQ_ConnectionClose	コネクションをクローズ(切断)します。	☞ 1879ページ M+MX-SQ_ConnectionClose

ソケット通信用ファンクションブロック

ファンクションブロック記号	処理内容	参照
M+MX-SQ_Recv_Socket	指定したコネクションにソケット通信で受信しているデータを読み出します。	☞ 1881ページ M+MX-SQ_Recv_Socket
M+MX-SQ_Send_Socket	指定したコネクションの相手機器へソケット通信でデータを送信します。	☞ 1884ページ M+MX-SQ_Send_Socket

SLMPフレーム用ファンクションブロック

ファンクションブロック記号	処理内容	参照
M+MX-SQ_SLMP_DeviceRead_IP	IPアドレス指定でSLMP対応機器のデバイスデータを読み出します。	☞ 1887ページ M+MX-SQ_SLMP_DeviceRead_IP
M+MX-SQ_SLMP_DeviceWrite_IP	IPアドレス指定でSLMP対応機器にデバイスデータを書き込みます。	☞ 1894ページ M+MX-SQ_SLMP_DeviceWrite_IP
M+MX-TSN_SLMP_DeviceRead_IP	IPアドレス指定でSLMP対応機器のデバイスデータを読み出します。	☞ 1901ページ M+MX-TSN_SLMP_DeviceRead_IP
M+MX-TSN_SLMP_DeviceWrite_IP	IPアドレス指定でSLMP対応機器にデバイスデータを書き込みます。	☞ 1908ページ M+MX-TSN_SLMP_DeviceWrite_IP

10.2 CC-Link IE TSN用ファンクションブロック

SLMPフレームのリモートリセット

ファンクションブロック記号	処理内容	参照
M+MX-TSN_RemoteReset2_IP	IPアドレス指定により対象局に対して、SLMP要求のリモートリセットを送信します。	☞ 1915ページ M+MX-TSN_RemoteReset2_IP

第3部

シーケンス命令/基本命令

この部は下記の章構成となります。

11 シーケンス命令

12 基本命令

11 シーケンス命令

11.1 接点命令

演算開始, 直列接続, 並列接続

LD, LDI, AND, ANI, OR, ORI

- LD: a接点演算開始命令/LDI: b接点演算開始命令
指定デバイスのON/OFF情報を演算結果とします。
- AND: a接点直列接続命令/ANI: b接点直列接続命令
指定デバイスのON/OFF情報と, それまでの演算結果のAND演算を演算結果とします。
- OR: a接点1個の並列接続命令/ORI: b接点1個の並列接続命令
指定デバイスのON/OFF情報とそれまでの演算結果のOR演算を演算結果とします。

ラダー	ST
LD LDI AND ANI OR ORI	対応していません。

FBD/LD

対応していません。

■実行条件

命令	実行条件
LD LDI AND ANI OR ORI	常時実行

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	接点として使用するデバイス	—	ビット	ANY_BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (DX)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	—	○	—	○	—	—	—	○

機能

■LD, LDI

- LD命令はa接点演算開始、LDI命令はb接点演算開始命令で、指定デバイスのON/OFF情報^{*1}を取り込み、演算結果とします。

^{*1} ワードデバイスのビット指定時は、指定ビットの1/0によりON/OFFします。

■AND, ANI

- AND命令はa接点直列接続、ANI命令はb接点直列接続命令で、指定ビットデバイスのON/OFF情報^{*1}を取り込み、それまでの演算結果とAND演算を行い、この値を演算結果とします。

^{*1} ワードデバイスのビット指定時は、指定ビットの1/0によりON/OFFします。

- エンジニアリングツールの回路編集モードでは下記ようになります。

・書込み: AND命令、ANI命令が直列に接続される場合は、24段までの回路を作成できます。

・読出し: AND命令、ANI命令が直列に接続される場合は、24段までの回路を表示できます。24段を超える場合は24段までが表示されます。

■OR, ORI

- OR命令はa接点1個の並列接続、ORI命令はb接点1個の並列接続命令で、指定デバイスのON/OFF情報^{*1}を取り込み、それまでの演算結果とOR演算を行い、この値を演算結果とします。

^{*1} ワードデバイスのビット指定時は、指定ビットの1/0によりON/OFFします。

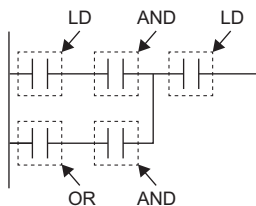
- エンジニアリングツールの回路編集モードでは下記ようになります。

・書込み: OR命令、ORI命令が23個連続で接続される回路まで作成できます。

・読出し: OR命令、ORI命令が23個連続で接続される回路まで表示できます。23個を超える回路は正常に表示できません。

■LD, LDI, AND, ANI, OR, ORIの組み合わせによる動作

下記に、LD, AND, ORの組み合わせによる動作例を示します。LDI, ANI, ORIも同様です。



Point

- ワードデバイスのビット指定時、ビットの指定は16進数で行います。(たとえば、D0のb11は「D0.0B」となります。)

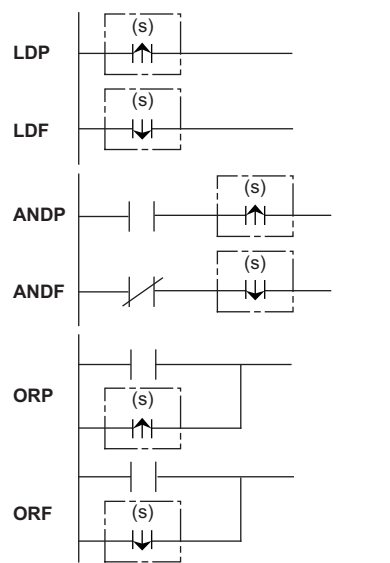
エラー

演算エラーはありません。

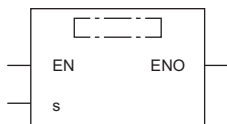
パルス演算開始, パルス直列接続, パルス並列接続

LDP, LDF, ANDP, ANDF, ORP, ORF

- LDP: 立上りパルス演算開始命令
指定ビットデバイスの立上り時(OFF→ON)にのみ導通します。
- LDF: 立下りパルス演算開始命令
指定ビットデバイスの立下り時(ON→OFF)にのみ導通します。
- ANDP: 立上りパルス直列接続命令/ANDF: 立下りパルス直列接続命令
それまでの演算結果とAND演算を行います。
- ORP: 立上りパルス並列接続命令/ORF: 立下りパルス並列接続命令
それまでの演算結果とOR演算を行います。

ラダー	ST
	<pre>ENO:=LDP(EN,s); ENO:=LDF(EN,s); ENO:=ANDP(EN,s); ENO:=ANDF(EN,s); ENO:=ORP(EN,s); ENO:=ORF(EN,s);</pre>

FBD/LD



■実行条件

命令	実行条件
LDP LDF ANDP ANDF ORP ORF	常時実行

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	接点として使用するデバイス	—	ビット	ANY_BOOL
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

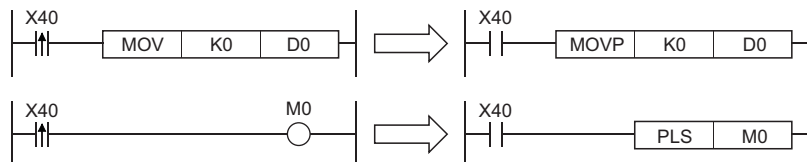
オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他 (DX)
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 口	Z	LT, LST, LC		K, H	E	\$	
(s)	○	○	○	○	—	○	—	○	—	—	○

機能

■LDP, LDF

- LDP命令は立上りパルス演算開始命令で、指定ビットデバイスの立上り時(OFF→ON)のみ導通します。ワードデバイスのビット指定時は、指定ビットが0→1に変化したときのみ導通します。LDP命令のみの場合は、ON中実行命令のパルス化命令(OP)と同一になります。

LDP命令を使用した回路を、LDP命令を使用しない回路に置き換えた場合は下記ようになります。



- LDF命令は立下りパルス演算開始命令で、指定ビットデバイスの立下り時(ON→OFF)に導通します。ワードデバイスのビット指定時は、指定ビットが1→0に変化したとき導通します。
- LDP命令をSTおよびFBD/LDで使用した場合、指定ビットデバイス(s)の立上り時(OFF→ON)に、ENOはONします。
- LDF命令をSTおよびFBD/LDで使用した場合、指定ビットデバイス(s)の立下り時(ON→OFF)に、ENOはONします。
- LDP, LDF命令をSTで使用した場合、ENに常時ONを指定してください。
- LDP, LDF命令をFBD/LDで使用した場合、ENに左母線、または常時ONの変数部品/定数部品を指定してください。

■ANDP, ANDF

- ANDP命令は立上りパルス直列接続命令、ANDF命令は立下りパルス直列接続命令で、それまでの演算結果とAND演算を行い、演算結果とします。ANDP命令、ANDF命令で使用するON/OFF情報は下記になります。

ANDP, ANDFで指定のデバイス		ANDPの状態	ANDFの状態
ビットデバイス	ワードデバイスのビット指定		
OFF→ON	0→1	ON	OFF
OFF	0	OFF	OFF
ON	1	OFF	OFF
ON→OFF	1→0	OFF	ON

- ANDP命令をSTおよびFBD/LDで使用した場合、ENと指定ビットデバイス(s)立上りのAND演算結果がONの時に、ENOはONします。ENは実行条件にはなりません。
- ANDF命令をSTおよびFBD/LDで使用した場合、ENと指定ビットデバイス(s)立下りのAND演算結果がONの時に、ENOはONします。ENは実行条件にはなりません。

■ORP, ORF

- ORP命令は立上りパルス並列接続命令，ORF命令は立下りパルス並列接続命令で，それまでの演算結果とOR演算を行い演算結果とします。ORP命令，ORF命令で使用するON/OFF情報は下記になります。

ORP, ORFで指定のデバイス		ORPの状態	ORFの状態
ビットデバイス	ワードデバイスのビット指定		
OFF→ON	0→1	ON	OFF
OFF	0	OFF	OFF
ON	1	OFF	OFF
ON→OFF	1→0	OFF	ON

- ORP命令をSTおよびFBD/LDで使用した場合，ENと指定ビットデバイス(s)立上りのOR演算結果がONの時，ENOはONします。ENは実行条件にはなりません。
- ORF命令をSTおよびFBD/LDで使用した場合，ENと指定ビットデバイス(s)立下りのOR演算結果がONの時，ENOはONします。ENは実行条件にはなりません。

■LDP, LDF, ANDP, ANDF, ORP, ORFの組み合わせによる動作

LDP, LDF, ANDP, ANDF, ORP, ORFの組み合わせによる動作例は，LD, AND, ORの組み合わせによる動作例と同様です。(200ページ LD, LDI, AND, ANI, OR, ORI)

エラー

演算エラーはありません。

パルス否定演算開始，パルス否定直列接続，パルス否定並列接続

LDPI, LDFI, ANDPI, ANDFI, ORPI, ORFI

- LDPI: 立上りパルス否定演算開始命令
指定デバイスのOFF時，ON時，立下り時(ON→OFF)の場合に導通します。
- LDFI: 立下りパルス否定演算開始命令
指定デバイスの立上り時(OFF→ON)，OFF時，ON時に導通します。
- ANDPI: 立上りパルス否定直列接続命令/ANDFI: 立下りパルス否定直列接続命令
それまでの演算結果とAND演算を行います。
- ORPI: 立上りパルス否定並列接続命令/ORFI: 立下りパルス否定並列接続命令
それまでの演算結果とOR演算を行います。

ラダー	ST
	<pre> ENO:=LDPI(EN,s); ENO:=LDFI(EN,s); ENO:=ANDPI(EN,s); ENO:=ANDFI(EN,s); ENO:=ORPI(EN,s); ENO:=ORFI(EN,s); </pre>

FBD/LD

■実行条件

命令	実行条件
LDPI LDFI ANDPI ANDFI ORPI ORFI	常時実行

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	接点として使用するデバイス	—	ビット	ANY_BOOL
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (DX)
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	—	○	—	○	—	—	—	○

機能

■LDPI, LDFI

- LDPI命令は立上りパルス否定演算開始命令で、指定ビットデバイスのOFF時、ON時、立下り時(ON→OFF)の場合に導通します。ワードデバイスのビット指定時は、指定ビットが0の場合、1の場合、1→0に変化した場合に導通します。
- LDFI命令は立下りパルス否定演算開始命令で、指定ビットデバイスの立上り時(OFF→ON)、OFF時、ON時の場合に導通します。ワードデバイスのビット指定時は、指定ビットが0の場合、1の場合、0→1に変化した場合に導通します。LDPI命令、LDFI命令で使用するON/OFF情報は下記になります。

LDPI, LDFIで指定のデバイス		LDPIの状態	LDFIの状態
ビットデバイス	ワードデバイスのビット指定		
OFF→ON	0→1	OFF	ON
OFF	0	ON	ON
ON	1	ON	ON
ON→OFF	1→0	ON	OFF

- LDPI命令をSTおよびFBD/LDで使用した場合、指定ビットデバイス(s)の立上り以外の時に、ENOはONします。
- LDFI命令をSTおよびFBD/LDで使用した場合、指定ビットデバイス(s)の立下り以外の時に、ENOはONします。
- LDPI, LDFI命令をSTで使用した場合、ENに常時ONを指定してください。
- LDPI, LDFI命令をFBD/LDで使用した場合、ENに左母線、または常時ONの変数部品/定数部品を指定してください。

■ANDPI, ANDFI

- ANDPI命令は立上りパルス否定直列接続命令、ANDFI命令は立下りパルス否定直列接続命令で、それまでの演算結果とAND演算を行い、演算結果とします。ANDPI命令、ANDFI命令で使用するON/OFF情報は下記になります。

ANDPI, ANDFIで指定のデバイス		ANDPIの状態	ANDFIの状態
ビットデバイス	ワードデバイスのビット指定		
OFF→ON	0→1	OFF	ON
OFF	0	ON	ON
ON	1	ON	ON
ON→OFF	1→0	ON	OFF

- ANDPI命令をSTおよびFBD/LDで使用した場合、ENと指定ビットデバイス(s)立上りのAND演算結果がON以外の時に、ENOはONします。ENは実行条件にはなりません。
- ANDFI命令をSTおよびFBD/LDで使用した場合、ENと指定ビットデバイス(s)立下りのAND演算結果がON以外の時に、ENOはONします。ENは実行条件にはなりません。

■ORPI, ORFI

- ORPI命令は立上りパルス否定並列接続命令、ORFI命令は立下りパルス否定並列接続命令で、それまでの演算結果とOR演算を行い、演算結果とします。ORPI命令、ORFI命令で使用するON/OFF情報は下記になります。

ORPI, ORFIで指定のデバイス		ORPIの状態	ORFIの状態
ビットデバイス	ワードデバイスのビット指定		
OFF→ON	0→1	OFF	ON
OFF	0	ON	ON
ON	1	ON	ON
ON→OFF	1→0	ON	OFF

- ORPI命令をSTおよびFBD/LDで使用した場合、ENと指定ビットデバイス(s)立上りのOR演算結果がON以外の時、ENOはONします。ENは実行条件にはなりません。
- ORFI命令をSTおよびFBD/LDで使用した場合、ENと指定ビットデバイス(s)立下りのOR演算結果がON以外の時、ENOはONします。ENは実行条件にはなりません。

■LDPI, LDFI, ANDPI, ANDFI, ORPI, ORFIの組み合わせによる動作

LDPI, LDFI, ANDPI, ANDFI, ORPI, ORFIの組み合わせによる動作例は、LD, AND, ORの組み合わせによる動作例と同様です。(☞ 200ページ LD, LDI, AND, ANI, OR, ORI)

エラー

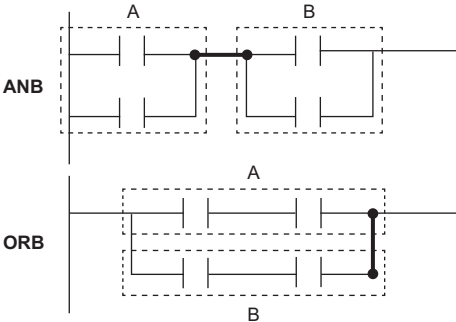
演算エラーはありません。

11.2 結合命令

回路ブロック直列接続，並列接続

ANB, ORB

- ANB: 回路ブロック直列接続命令
AブロックとBブロックのAND演算を行います。
- ORB: 回路ブロック並列接続命令
AブロックとBブロックのOR演算を行います。

ラダー	ST
 A: Aブロック B: Bブロック	対応していません。

FBD/LD
対応していません。

■実行条件

命令	実行条件
ANB ORB	常時実行

機能

■ANB

- AブロックとBブロックのAND演算を行い，演算結果とします。
- ANB命令のシンボルは接点シンボルではなく，接続シンボルです。

■ORB

- AブロックとBブロックのOR演算を行い，演算結果とします。
- ORB命令は，2接点以上の回路ブロックを並列接続します。1接点のみの並列接続は，OR命令，ORI命令を使用し，ORB命令は必要ありません。
- ORB命令のシンボルは接点シンボルではなく，接続シンボルです。

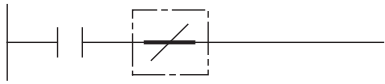
エラー

演算エラーはありません。

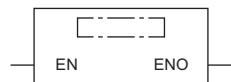
演算結果反転

INV

INV命令の直前までの演算結果を反転します。

ラダー	ST
	ENO:=INV(EN);

FBD/LD



■実行条件

命令	実行条件
INV	常時実行

機能

- INV命令の直前までの演算結果を反転します。

INV命令の直前までの演算結果	INV命令実行後の演算結果
OFF	ON
ON	OFF

エラー

演算エラーはありません。

Point

- INV命令は、INV命令の直前までの演算結果で動作しますので、AND(200ページ 演算開始, 直列接続, 並列接続)と同一位置で使用してください。INV命令は、LD, OR(200ページ 演算開始, 直列接続, 並列接続)の位置では使用できません。
- 回路ブロックを使用している場合は、回路ブロックの範囲で演算結果を反転します。INV命令とANB命令を併用して回路を動作させる場合、反転する範囲に注意してください。



破線部: 反転する範囲

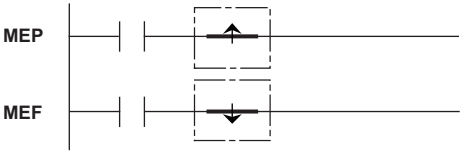
ANB命令の詳細については、下記を参照してください。

208ページ ANB, ORB

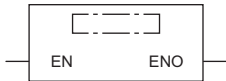
演算結果パルス化

MEP, MEF

- MEP: 演算結果パルス化命令(立上り)
MEP命令までの演算結果の立上り時(OFF→ON)に、導通します。
- MEF: 演算結果パルス化命令(立下り)
MEF命令までの演算結果の立下り時(ON→OFF)に、導通します。

ラダー	ST
	ENO:=MEP(EN); ENO:=MEF(EN);

FBD/LD



■実行条件

命令	実行条件
MEP MEF	常時実行

機能

■MEP

- MEP命令までの演算結果が立上り時(OFF→ON)に、ON(導通状態)になります。MEP命令までの演算結果が立上り時以外の場合は、OFF(非導通状態)になります。
- MEP命令を使用すると、複数の接点を直列に接続した場合のパルス化処理が容易になります。

■MEF

- MEF命令までの演算結果が立下り時(ON→OFF)に、ON(導通状態)になります。MEF命令までの演算結果が立下り時以外の場合は、OFF(非導通状態)になります。
- MEF命令を使用すると、複数の接点を直列に接続した場合のパルス化処理が容易になります。

エラー

演算エラーはありません。

Point

- MEP命令、MEF命令は、サブルーチンプログラムおよびFOR~NEXT命令などで、インデックス修飾した接点のパルス化を行うと、正常に動作しないことがあります。サブルーチンプログラム、FOR~NEXT命令でインデックス修飾した接点のパルス化を行う場合は、下記を参照してください。
☞ 211ページ EGP、EGF
- MEP命令、MEF命令は、MEP命令、MEF命令の直前にあるLD命令からMEP命令、MEF命令の直前までの演算結果で動作しますので、AND(☞ 200ページ 演算開始、直列接続、並列接続)と同一位置で使用してください。MEP命令、MEF命令は、LD、OR(☞ 200ページ 演算開始、直列接続、並列接続)の位置では使用できません。

エッジリレー演算結果パルス化

EGP, EGF

- EGP: エッジリレー演算結果パルス化命令(立上り)

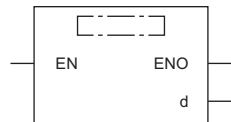
EGP命令までの演算結果をエッジリレー (V)で記憶します。演算結果の立上り時(OFF→ON)に、導通します。

- EGF: エッジリレー演算結果パルス化命令(立下り)

EGF命令までの演算結果をエッジリレー (V)で記憶します。演算結果の立下り時(ON→OFF)に、導通します。

ラダー	ST
	ENO:=EGP(EN,d); ENO:=EGF(EN,d);

FBD/LD



■実行条件

命令	実行条件
EGP	
EGF	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	演算結果を記憶するエッジリレー番号	—	ビット	ANY_BOOL*1
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 デバイス(V)に割り付けたビット型ラベルのみ使用できます。

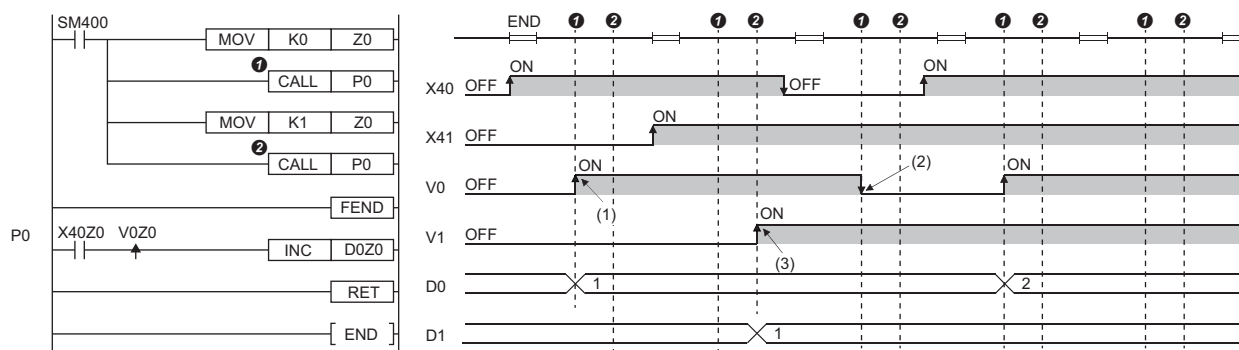
■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他 (V)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□		LT, LST, LC	LZ		K, H	E	\$	
(d)	—	—	—	—	—	—	—	—	—	—	—	○

機能

■EGP

- EGP命令までの演算結果をエッジリレー (V)で記憶します。
- EGP命令までの演算結果が立上り時(OFF→ON)に、ON(導通状態)になります。EGP命令までの演算結果が立上り時以外(ON→ON, ON→OFF, OFF→OFF)の場合は、OFF(非導通状態)になります。
- EGP命令は、サブルーチンプログラムやFOR~NEXT命令でインデックス修飾されたプログラムのパルス演算を行う場合に使用します。
- EGP命令は、AND命令扱いで使用できます。
- サブルーチンプログラムでEGP命令を使用した場合、動作は下記になります。



- (1): X40の立上りでONします。
(2): X40の立下りでOFFします。
(3): X41の立上りでONします。

■EGF

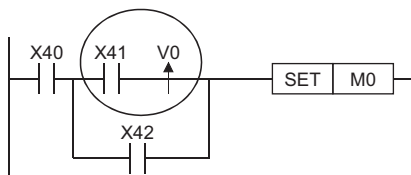
- EGF命令までの演算結果をエッジリレー (V)で記憶します。
- EGF命令までの演算結果が立下り時(ON→OFF)に、ON(導通状態)になります。EGF命令までの演算結果が立下り時以外(OFF→ON, ON→ON, OFF→OFF)の場合は、OFF(非導通状態)になります。
- EGF命令は、サブルーチンプログラムやFOR~NEXT命令で、インデックス修飾されたプログラムのパルス演算を行う場合に使用します。
- EGF命令は、AND命令扱いで使用できます。

エラー

演算エラーはありません。

Point

- EGP命令、EGF命令は、EGP命令、EGF命令の直前にあるLD命令からEGP命令、EGF命令の直前までの演算結果で動作しますので、AND(200ページ 演算開始, 直列接続, 並列接続)と同一位置で使用してください。EGP命令、EGF命令は、LD, OR(200ページ 演算開始, 直列接続, 並列接続)の位置では使用できません。
- EGP命令、EGF命令は、下記の回路ブロックの位置では使用できません。

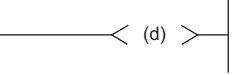


11.3 出力命令

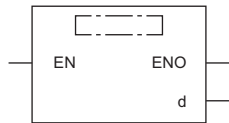
アウト(タイマ, カウンタ, アナンシェータを除く)

OUT

演算結果を指定されたデバイスへ出力します。

ラダー	ST
	ENO:=OUT(EN,d);

FBD/LD



■実行条件

命令	実行条件
OUT	常時実行

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	ON/OFFするデバイス番号	—	ビット	ANY_BOOL
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他 (DY)
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口		LT, LST, LC	LZ		K, H	E	\$	
(d)	○*1	○	○*2	○	—	—	—	○	—	—	—	○

*1 Fを使用する場合は, 下記の命令を参照してください。

☞ 225ページ OUT F

*2 T, STを使用する場合は, 下記の命令を参照してください。

☞ 215ページ OUT T, OUTH T, OUT ST, OUTH ST

Cを使用する場合は, 下記の命令を参照してください。

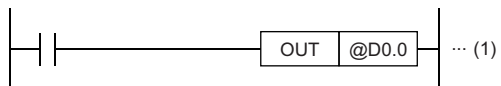
☞ 221ページ OUT C

機能

- OUT命令までの演算結果を指定されたデバイスへ出力します。

条件	演算結果	コイル/指定ビット
ビットデバイス使用時	OFF	OFF
	ON	ON
ワードデバイスのビット指定時	OFF	0
	ON	1

- 間接指定を使用する場合は、下記のようにビット指定します。



(1) D0に格納された間接アドレスの0ビット目に出力します。

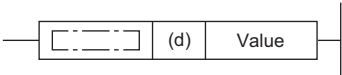
エラー

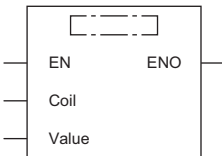
演算エラーはありません。

OUT T, OUTH T, OUT ST, OUTH ST

- OUT T: 低速タイマ命令
- OUTH T: 高速タイマ命令
- OUT ST: 低速積算タイマ命令
- OUTH ST: 高速積算タイマ命令

OUT命令までの演算結果がONのとき、コイルがONしてタイマの計測を行います。タイムアップすると、a接点は導通し、b接点は非導通になります。

ラダー	ST
	<pre>ENO:=OUT_T(EN,Coil,Value); ENO:=OUTH(EN,Coil,Value);</pre>
Value: 設定値	

FBD/LD

Value: 設定値 (□には、OUT_T, OUTHが入ります。)

■実行条件

命令	実行条件
OUT T OUTH T OUT ST OUTH ST	常時実行

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	タイマデバイスまたはタイマ型ラベル	—	ビット	ANY_BOOL
Coil				
Value	タイマの設定値	0~65535	符号なしBIN16ビット*1	ANY16*1
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ST, FBD/LDの場合、データ型はANY_INTになります。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他 (DY)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC		K, H	E	\$	
(d)	—	—	○*1	—	—	—	—	—	—	—	—
Coil	—	—	○*1	—	—	—	—	—	—	—	—
Value	—	—	○*2	○	—	—	—	○*3	—	—	—

*1 T, STのみ使用できます。

*2 T, ST, Cは使用できません。

*3 10進定数(K)のみ使用できます。

機能

- OUT命令までの演算結果がONのとき、(d)(ST, FBD/LDの場合はCoil)で指定したタイマのコイルはONして設定値まで計測します。タイムアップ(現在値 $\geq$ 設定値)すると、a接点は導通し、b接点是非導通になります。
- OUT命令までの演算結果がON $\rightarrow$ OFFに変化すると下記ようになります。

タイマの種類	タイマコイル	タイマの現在値	タイムアップ前		タイムアップ後	
			a接点	b接点	a接点	b接点
低速タイマ	OFF	0	非導通	導通	非導通	導通
高速タイマ						
低速積算タイマ	OFF	現在値を保持	非導通	導通	導通	非導通
高速積算タイマ						

- タイムアップ後、積算タイマの現在値のクリアと接点のOFFは、RST命令で行います。
- 設定値が0のときは、OUT命令実行時にタイムアップします。
- OUT命令実行時に、下記の処理を行います。
 - OUT T命令、OUTH T命令、OUT ST命令、OUTH ST命令のコイルのON/OFF
 - OUT T命令、OUTH T命令、OUT ST命令、OUTH ST命令の接点のON/OFF
 - OUT T命令、OUTH T命令、OUT ST命令、OUTH ST命令の現在値の変更
- OUT T命令、OUTH T命令、OUT ST命令、OUTH ST命令がON中に、JMP命令などでOUT T命令を飛ばした場合、現在値の更新および接点のON/OFFを行いません。
- 同一のOUT T命令、OUTH T命令、OUT ST命令、OUTH ST命令を同スキャン内で2回以上実行した場合は、実行した回数分の現在値を更新します。

Point

タイマの時限設定は、エンジニアリングツールのパラメータ設定で行います。

- 低速タイマ/低速積算タイマ: 1~10000ms(デフォルト: 100ms, 設定単位: 1ms)
- 高速タイマ/高速積算タイマ: 0.01~100.0ms(デフォルト: 10.0ms, 設定単位: 0.01ms)

タイマの計数方法については、下記を参照してください。

 使用するコントローラのユーザーズマニュアル

注意事項

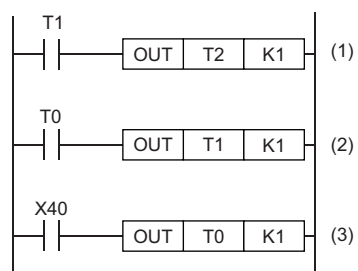
タイマの接点で別のタイマの計測を行うプログラムを作成する場合は、後で計測するタイマから順にプログラミングしてください。

タイマの設定値がスキャンタイムより短い設定値の場合、計測順にプログラミングを行うと、すべてのタイマが同ースキャンでONします。

- スキャンタイムより短い設定値のとき
- 設定値が1のとき

例

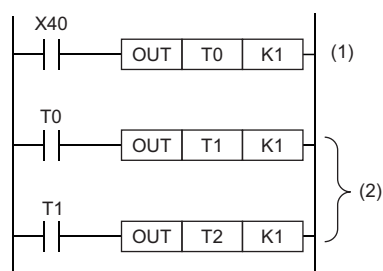
T0~T2のタイマを後から計測するタイマから順にプログラミングした場合



- (3) X40がONすると、T0のタイマが計測を開始します。
- (2) T0の接点がONした次のスキャンから、T1のタイマが計測を開始します。
- (1) T1の接点がONした次のスキャンから、T2のタイマが計測を開始します。

例

T0~T2のタイマを計測順にプログラミングした場合



- (1) X40がONすると、T0のタイマが計測を開始します。
- (2) T0の接点がONすると、T1、T2のタイマの接点もONします。

エラー

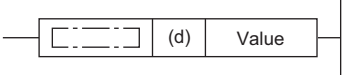
演算エラーはありません。

ロングタイマ

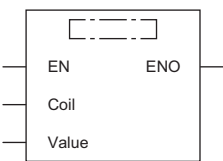
OUT LT, OUT LST

- OUT LT: 低速ロングタイマ命令
- OUT LST: 低速ロング積算タイマ命令

OUT命令までの演算結果がONのとき、コイルがONしてタイマの計測を行います。タイムアップすると、a接点は導通し、b接点は非導通になります。

ラダー	ST
	ENO:=OUT_T(EN,Coil,Value);
Value: 設定値	

FBD/LD



Value: 設定値
(口には、OUT_Tが入ります。)

■実行条件

命令	実行条件
OUT LT OUT LST	常時実行

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	ロングタイマデバイスまたはロングタイマ型ラベル	—	ビット	ANY_BOOL
Coil				
Value	ロングタイマの設定値	0~4294967295	符号なしBIN32ビット *1	ANY32*1
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ST, FBD/LDの場合、データ型はANY_INTになります。

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口		LT, LST, LC	LZ		K, H	E	\$	
(d)	—	—	—	—	—	○*1	—	—	—	—	—	—
Coil	—	—	—	—	—	—	—	—	—	—	—	—
Value	—	—	○*2	○	—	—	—	—	○*3	—	—	—

*1 LT, LSTのみ使用できます。

*2 T, ST, Cは使用できません。

*3 10進定数(K)のみ使用できます。

機能

- OUT命令までの演算結果がONのとき、(d)(ST, FBD/LDの場合はCoil)で指定したロングタイマのコイルはONして設定値まで計測します。タイムアップ(現在値>設定値)すると、a接点は導通し、b接点是非導通になります。
- OUT命令までの演算結果がON→OFFに変化すると下記ようになります。

タイマの種類	タイマコイル	タイマの現在値	タイムアップ前		タイムアップ後	
			a接点	b接点	a接点	b接点
ロングタイマ	OFF	0	非導通	導通	非導通	導通
ロング積算タイマ	OFF	現在値を保持	非導通	導通	導通	非導通

- タイムアップ後、ロング積算タイマの現在値のクリアと接点のOFFは、RST命令で行います。
- 設定値が0のときは、OUT命令実行時にタイムアップします。
- OUT命令実行時に、下記の処理を行います。
 - OUT LT命令、OUT LST命令のコイルのON/OFF
 - OUT LT命令、OUT LST命令の接点のON/OFF
 - OUT LT命令、OUT LST命令の現在値の変更
- OUT LT命令、OUT LST命令がON中に、JMP命令などでOUT LT命令を飛ばした場合、現在値の更新および接点のON/OFFを行いません。
- 同一のOUT LT命令、OUT LST命令を同スキャン内で2回以上実行した場合は、実行した回数分の現在値を更新します。

Point

タイマの時限設定は、エンジニアリングツールのパラメータ設定で行います。

- ロングタイマ/ロング積算タイマ: 0.001~1000ms(デフォルト: 0.001ms, 設定単位: 0.001ms)

タイマの計数方法については、下記を参照してください。

 使用するコントローラのユーザーズマニュアル

注意事項

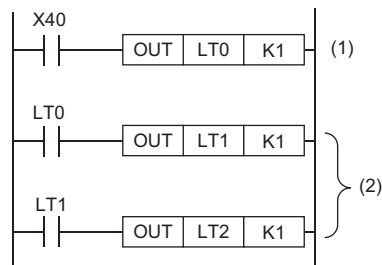
ロングタイマの接点で別のロングタイマの計測を行うプログラムを作成する場合は、計測するロングタイマから順にプログラミングしてください。

後から計測するロングタイマから順にプログラミングすると、タイマの計測誤差が大きくなる場合があります。

ロングタイマの設定値がスキャンタイムより短い設定値の場合、下記ようになります。

例

LT0~LT2のタイマを計測順にプログラミングした場合

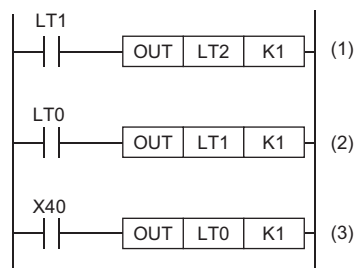


(1) X40がONすると、LT0のタイマが計測を開始します。

(2) LT0の接点がONすると、LT1、LT2の順にタイマが計測を開始します。

例

LT0~LT2のタイマを後から計測するロングタイマから順にプログラミングした場合



(3) X40がONすると、LT0のタイマが計測を開始します。

(2) LT0の接点がONした次のスキャンから、LT1のタイマが計測を開始します。

(1) LT1の接点がONした次のスキャンから、LT2のタイマが計測を開始します。

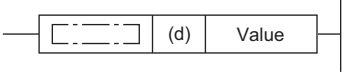
エラー

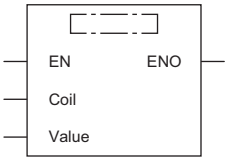
演算エラーはありません。

カウンタ

OUT C

OUT命令までの演算結果がOFF→ONに変化したとき、カウンタの現在値(カウント値)を+1します。カウントアップすると、カウンタのa接点は導通し、b接点は非導通になります。

ラダー	ST
 Value: 設定値	<pre>ENO:=OUT_C(EN,Coil,Value);</pre>

FBD/LD
 Value: 設定値

■実行条件

命令	実行条件
OUT C	常時実行

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	カウンタ番号	—	ビット	ANY_BOOL ^{*1}
Coil				
Value	カウンタの設定値	0~65535	符号なしBIN16ビット ^{*2}	ANY16 ^{*2}
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 カウンタ型ラベルのみ使用できます。

*2 ST, FBD/LDの場合、データ型はANY_INTになります。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(d)	—	—	○ ^{*1}	—	—	—	—	—	—	—	—
Coil	—	—	○ ^{*1}	—	—	—	—	—	—	—	—
Value	—	—	○ ^{*2}	○	—	—	—	○ ^{*3}	—	—	—

*1 Cのみ使用できます。

*2 T, ST, Cは使用できません。

*3 10進定数(K)のみ使用できます。

機能

- OUT命令までの演算結果がOFF→ONに変化したとき、(d)(ST, FBD/LDの場合はCoil)で指定したカウンタの現在値(カウント値)を+1します。カウントアップ(現在値 $\geq$ 設定値)すると、a接点は導通し、b接点是非導通になります。
- 設定値が0の場合は、上記と同一処理となります。
- 演算結果がONのままではカウントされません。(カウント入力パルス化する必要はありません。)
- カウントアップ後は、RST命令を実行するまでカウント値および接点の状態は変化しません。

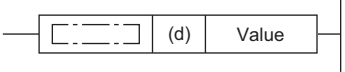
エラー

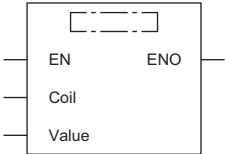
演算エラーはありません。

ロングカウンタ

OUT LC

OUT命令までの演算結果がOFF→ONに変化したとき、ロングカウンタの現在値(カウント値)を+1します。カウントアップすると、ロングカウンタのa接点は導通し、b接点は非導通になります。

ラダー	ST
	ENO:=OUT_C(EN,Coil,Value);
Value: 設定値	

FBD/LD

Value: 設定値 (口には、OUT_Cが入ります。)

■実行条件

命令	実行条件
OUT LC	常時実行

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	ロングカウンタ番号	—	ビット	ANY_BOOL ^{*1}
Coil				
Value	ロングカウンタの設定値	0~4294967295	符号なしBIN32ビット ^{*2}	ANY32 ^{*2}
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ロングカウンタ型ラベルのみ使用できます。

*2 ST, FBD/LDの場合、データ型はANY_INTになります。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	—	—	—	—	—	○ ^{*1}	—	—	—	—	—	—
Coil												
Value	—	—	○ ^{*2}	○	—	—	—	—	○ ^{*3}	—	—	—

*1 LCのみ使用できます。

*2 T, ST, Cは使用できません。

*3 10進定数(K)のみ使用できます。

機能

- OUT命令までの演算結果がOFF→ONに変化したとき、(d)(ST, FBD/LDの場合はCoil)で指定したロングカウンタの現在値(カウント値)を+1します。カウントアップ(現在値 $\geq$ 設定値)すると、a接点は導通し、b接点是非導通になります。
- 設定値が0の場合は、上記と同一処理となります。
- 演算結果がONのままではカウントされません。(カウント入力パルス化する必要はありません。)
- カウントアップ後は、RST命令を実行するまでカウント値および接点の状態は変化しません。

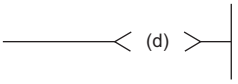
エラー

演算エラーはありません。

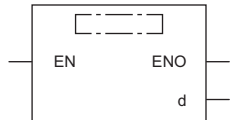
アナンシェータ

OUT F

OUT F命令までの演算結果を指定のアナンシェータへ出力します。

ラダー	ST
	ENO:=OUT(EN,d);

FBD/LD



(□には、OUTが入ります。)

■実行条件

命令	実行条件
OUT F	常時実行

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	ONするアナンシェータ番号	—	ビット	ANY_BOOL ^{*1}
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 アナンシェータに割り付けたラベルのみ使用できます。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(d)	○ ^{*1}	—	—	—	—	—	—	—	—	—	—

*1 Fのみ使用できます。

機能

- OUT F命令までの演算結果を指定のアナンシェータへ出力します。
- OUT F命令でアナンシェータ(F)をONする場合は下記になります。
 - コントローラ前面のUSER LEDが点灯します。
 - ONするアナンシェータ番号(F番号)が、特殊レジスタ(SD64~SD79)に格納されます。
 - SD63の内容が+1されます。
- SD63の内容が16の場合(アナンシェータがすでに16個ONしている場合)は、新たにアナンシェータがONしても、ONするアナンシェータ番号が、SD64~SD79には格納されません。
- OUT F命令でアナンシェータをOFFする場合は下記になります。
 - コイルはOFFしますが、USER LEDの状態、SD63~SD79の内容は変化しません。
 - USER LEDの消灯、SD63~SD79からOUT F命令でOFFしたアナンシェータを削除する場合は、RST F命令で行います。

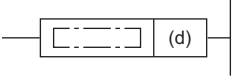
エラー

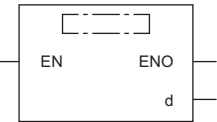
演算エラーはありません。

デバイスのセット(アナンシェータを除く)

SET

指定ビットをONします。

ラダー	ST
	ENO:=SET(EN,d);

FBD/LD


■実行条件

命令	実行条件
SET	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	セット(ON)するビットデバイス番号/ワードデバイスのビット指定	—	ビット	ANY_BOOL
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他(DY)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	○*1	○	—	—	—	—	—	○	—	—	—	○

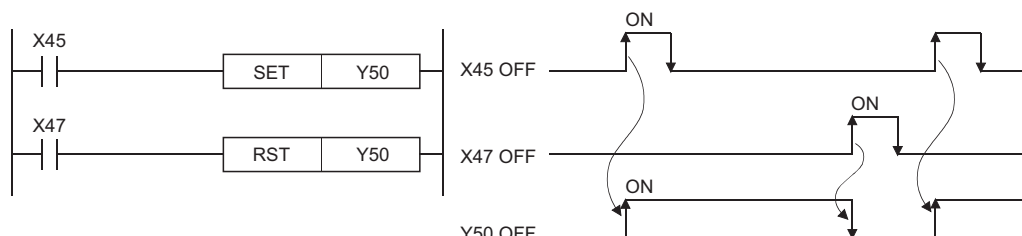
*1 Fを使用する場合は、下記の命令を参照してください。
230ページ SET F

機能

- 実行指令がONすると指定デバイスは下記の状態になります。

デバイス	デバイスの状態
ビットデバイス	コイル、接点をONします。
ワードデバイスのビット指定	指定ビットを1にします。

- ONさせたデバイスは、実行指令がOFFになってもONのまま保持されます。SET命令でONしたデバイスは、RST命令でOFFできます。



- 実行指令がOFFの場合、デバイスの状態は変化しません。

エラー

演算エラーはありません。

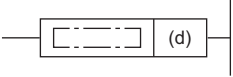
Point

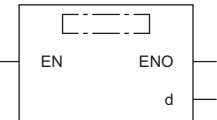
デバイスにXを使用する場合は、実入力で使用していないデバイス番号を使用してください。実入力デバイスと同一番号を使用すると、実入力のデータがSET命令で指定した入力Xに上書きされます。

デバイスのリセット(アナンシェータを除く)

RST

指定デバイスを0(OFF)にします。タイマ，カウンタについては，現在値を0，接点/コイルをOFFにします。

ラダー	ST
	ENO:=RST(EN,d);

FBD/LD


■実行条件

命令	実行条件
RST	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	リセットするビットデバイス番号/ワードデバイスのビット指定，またはリセットするワードデバイス番号	—	ビット/ワード/ダブルワード	ANY_ELEMENTARY
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他 (DY)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ	K, H	E	\$	
(d)	○*1	○	○	○	○	○	○	—	—	—	○

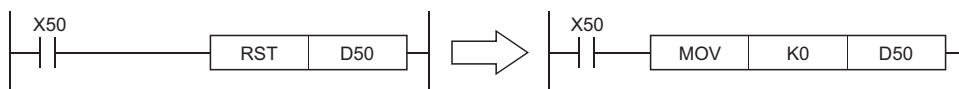
*1 Fを使用する場合は，下記の命令を参照してください。
232ページ RST F

機能

- 実行指令がONすると指定デバイスは下記の状態になります。

デバイス	デバイスの状態
ビットデバイス	コイル，接点をOFFします。
タイマ，カウンタ	現在値を0にし，コイル，接点をOFFします。
ワードデバイスのビット指定	指定ビットを0にします。
タイマ，カウンタ以外のワードデバイス	内容を0にします。

- 実行指令がOFFの場合，デバイスの状態は変化しません。
- RST命令でワードデバイス指定時の機能は，下記の回路と同一となります。



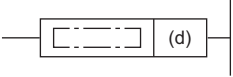
エラー

演算エラーはありません。

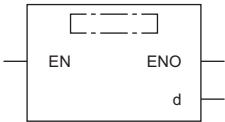
アナンシェータのセット

SET F

指定されたアナンシェータをONします。

ラダー	ST
	ENO:=SET(EN,d);

FBD/LD



(□には、SETが入ります。)

■実行条件

命令	実行条件
SET F	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	セットするアナンシェータ番号(F番号)	—	ビット	ANY_BOOL
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

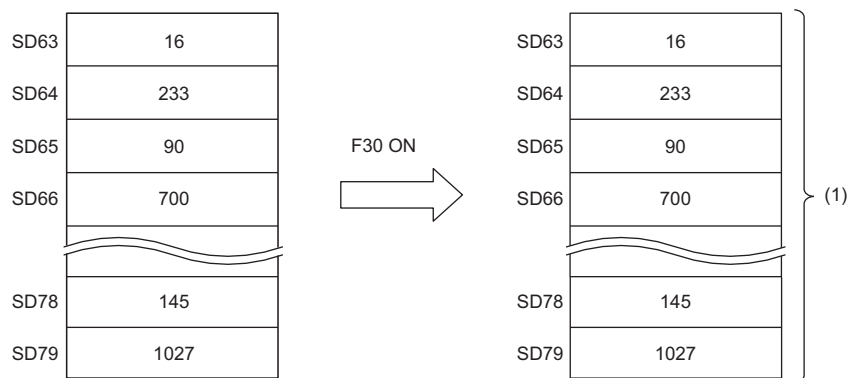
■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(d)	○*1	—	—	—	—	—	—	—	—	—	—	—	—

*1 Fのみ使用できます。

機能

- 実行指令がONしたとき、(d)で指定されたアナンシエータをONします。
- アナンシエータ(F)をONした場合は下記になります。
 - USER LEDが点灯されます。
 - ONしたアナンシエータ番号(F番号)が、特殊レジスタ(SD64~SD79)に格納されます。
 - SD63の内容が+1されます。
- SD63の内容が16の場合(アナンシエータがすでに16個ONしている場合)は、新たにアナンシエータがONしても、ONするアナンシエータ番号が、SD64~SD79には格納されません。



(1): 変化しません。

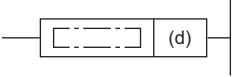
エラー

演算エラーはありません。

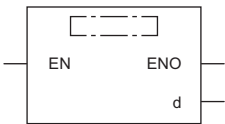
アナンシェータのリセット

RST F

指定されたアナンシェータをOFFします。

ラダー	ST
	ENO:=RST(EN,d);

FBD/LD



(口には、RSTが入ります。)

■実行条件

命令	実行条件
RST F	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	リセットするアナンシェータ番号(F番号)	—	ビット ^{*1}	ANY_BOOL ^{*1}
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ST、FBD/LDの場合、データ型はANY_ELEMENTARYになります。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	○ ^{*1}	—	—	—	—	—	—	—	—	—	—	—

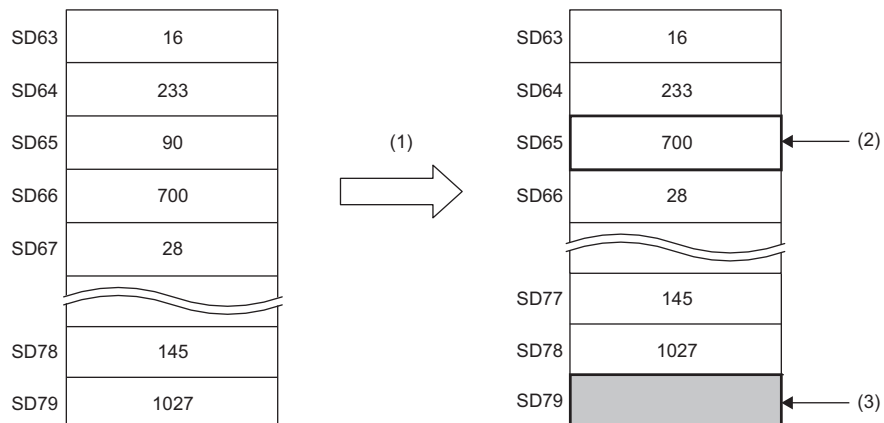
*1 Fのみ使用できます。

機能

- 実行指令がONしたとき、(d)で指定されるアナンシェータをOFFします。
- OFFしたアナンシェータ番号(F番号)は、特殊レジスタ(SD64~SD79)から削除され、SD63の内容が-1されます。
- SD63の内容が16のとき、RST命令によりSD64~SD79からアナンシェータ番号が削除されます。また、SD64~SD79に登録されていない番号のアナンシェータがONしていれば、その番号を新たに登録します。SD64~SD79のアナンシェータ番号がすべてOFFすると、コントローラ前面のUSER LEDが消灯します。

例

SD63が16で、登録されていないアナンシェータ番号がある場合



(1): F90をリセット

(2): SD66に格納されていたF番号が格納されます。

(3): 登録されていないF番号が格納されます。

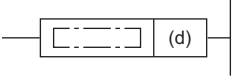
エラー

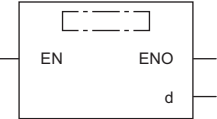
演算エラーはありません。

立上り出力

PLS

PLS指令のOFF→ON時に指定デバイスを1スキャンONさせます。

ラダー	ST
	ENO:=PLS(EN,d);

FBD/LD


■実行条件

命令	実行条件
PLS	

■設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	パルス化するデバイス番号	—	ビット	ANY_BOOL
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

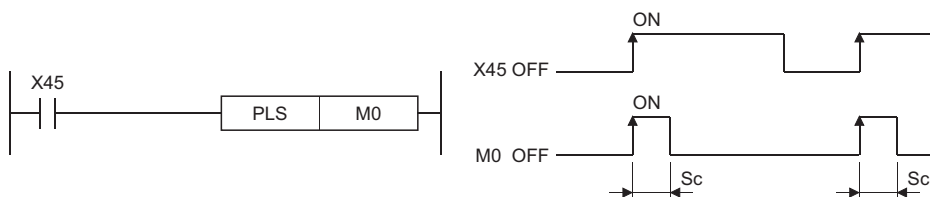
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (DY)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	○	○	○	○	—	—	—	○	—	—	—	○

機能

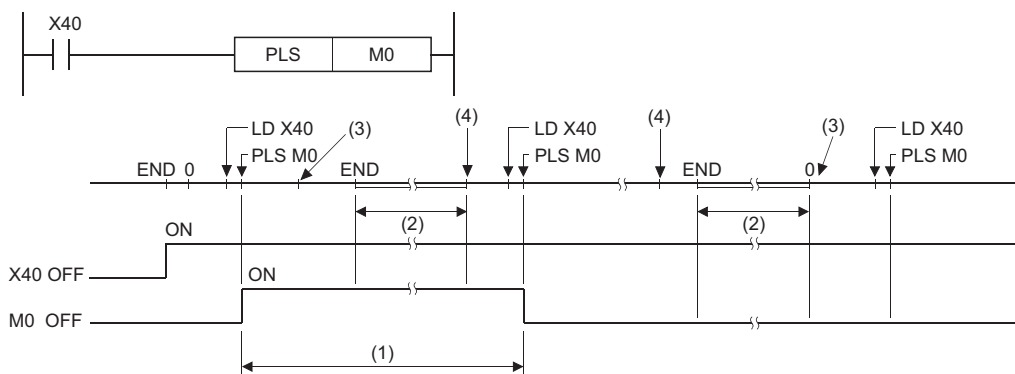
- 実行指令のOFF→ON時に指定デバイスをONし、実行指令のOFF→ON時以外(ON→ON, ON→OFF, OFF→OFF)のときはOFFさせます。1スキャン中に(d)で指定されたデバイスのPLS命令が1つの場合は、指定デバイスが1スキャンONします。同一デバイスに対して、PLS命令を1スキャンに複数回実行した場合の動作は、下記を参照してください。

☞ 52ページ 同一デバイスのOUT命令, SET/RST命令, PLS/PLF命令使用時の動作



Sc: 1スキャン

- PLS命令実行後にRUN→STOPにし、再度RUNにしてもPLS命令は実行されません。



- (1) PLS M0の1スキャン
- (2) コントローラの演算停止時間
- (3) コントローラのRUN/STOP/RESETスイッチをRUN→STOPにします。
- (4) コントローラのRUN/STOP/RESETスイッチをSTOP→RUNにします。

- 実行指令にラッチリレー (L)を指定し、ラッチリレーがON状態で電源をOFF→ONにすると、実行指令は1スキャン目にOFF→ONになり、PLS命令を実行して指定デバイスがONします。電源ON後1スキャン目でONしたデバイスは、次のPLS命令(立上り出力)でOFFします。
- PLS命令は命令実行後の次の命令実行時にOFF処理を行います。

Point

- PLS命令をCJ命令でジャンプさせたり、実行したサブルーチンプログラムをCALL(P)命令でコールしなかった場合、(d)で指定のデバイスが1スキャン以上ONになることがあるので、注意してください。

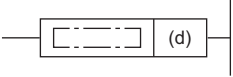
エラー

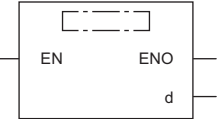
演算エラーはありません。

立下り出力

PLF

PLF指令のON→OFF時に指定デバイスを1スキャンONさせます。

ラダー	ST
	ENO:=PLF(EN,d);

FBD/LD


■実行条件

命令	実行条件
PLF	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	パルス化するデバイス番号	—	ビット	ANY_BOOL
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

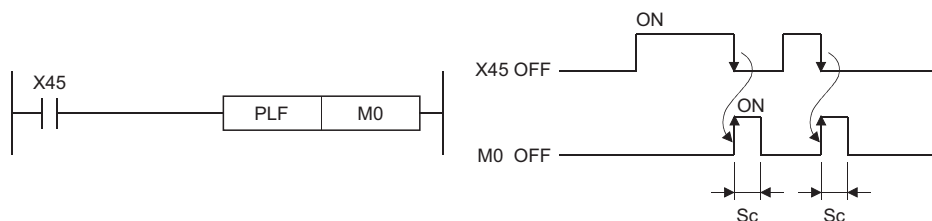
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (DY)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	○	○	○	○	—	—	—	○	—	—	—	○

機能

- 実行指令のON→OFF時に指定デバイスをONし、実行指令がON→OFF以外(OFF→OFF, OFF→ON, ON→ON)のときはOFFにします。1スキャン中に(d)で指定されたデバイスのPLF命令が1つの場合は、指定デバイスが1スキャンONします。同一デバイスに対して、PLF命令を1スキャンに複数回実行した場合の動作は、下記を参照してください。

☞ 52ページ 同一デバイスのOUT命令, SET/RST命令, PLS/PLF命令使用時の動作



Sc: 1スキャン

- PLF命令実行後にRUN→STOPにし、再度RUNにしてもPLF命令は実行されません。
- PLF命令は命令実行後の次の命令実行時にOFF処理を行います。

Point

- PLF命令をCJ命令でジャンプさせたり、実行したサブルーチンプログラムをCALL(P)命令でコールしなかった場合、(d)で指定のデバイスが1スキャン以上ONになることがあるので、注意してください。

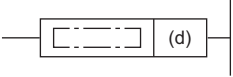
エラー

演算エラーはありません。

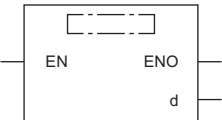
ビットデバイス出力反転

FF

指定されたデバイス状態を反転します。

ラダー	ST
	ENO:=FF(EN,d);

FBD/LD



■実行条件

命令	実行条件
FF	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	反転するデバイス番号	—	ビット	ANY_BOOL
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (DY)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□	G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(d)	○	○	○	○	—	—	—	○	—	—	—	○

機能

- 実行指令のOFF→ON時に、(d)で指定されたデバイス状態を反転します。

デバイス	デバイスの状態	
	FF命令実行前	FF命令実行後
ビットデバイス	OFF	ON
	ON	OFF
ワードデバイスのビット指定	0	1
	1	0

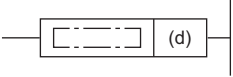
エラー

演算エラーはありません。

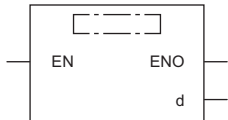
ダイレクト出力のパルス化

DELTA(P)

指定されたダイレクトアクセス出力(DY)をパルス出力します。

ラダー	ST
	<pre>ENO:=DELTA(EN,d); ENO:=DELTAP(EN,d);</pre>

FBD/LD



■実行条件

命令	実行条件
DELTA	
DELTAP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	パルス化するデバイス番号	—	ビット	ANY_BOOL ^{*1}
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 デバイス(DY)に割り付けたラベルのみ使用できます。

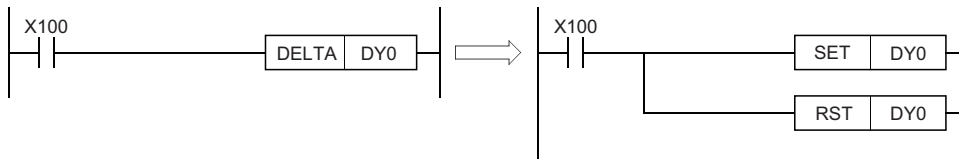
■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他 (DY)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(d)	—	—	—	—	—	—	—	—	—	—	○

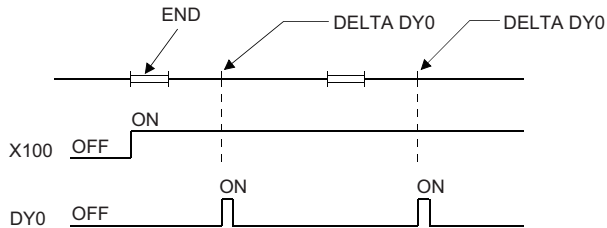
機能

- ・ (d)で指定したダイレクトアクセス出力(DY)をパルス出力します。(d)にDY0を指定した場合は、SET命令、RST命令を使用した回路と同一動作になります。

DELTA命令を使用した回路を、SET/RST命令を使用した回路に置き換えた場合は下記のようになります。



動作は下記ようになります。



注意事項

DELTA(P)命令は、インテリジェント機能ユニットへの立上り実行の指令に使用します。
出力ユニットを使用した実出力指令には使用できません。

エラー

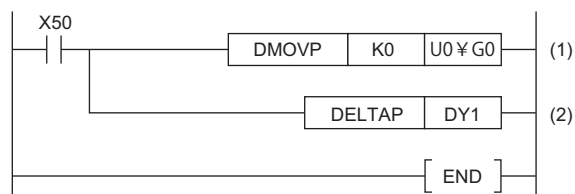
演算エラーはありません。

Point

DELTA(P)命令は、高速カウンタユニットのプリセット値の設定などに使用します。

(例)

X50がONしたとき、ベースユニットの-slot0に装着されている高速カウンタユニット(RD62P2)のCH1のプリセットを行うプログラム。



(1) RD62P2のバッファメモリ0, 1のアドレスにプリセット値(0)を格納

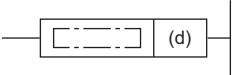
(2) プリセット指令の出力

11.4 シフト命令

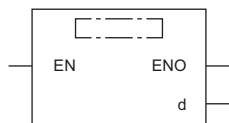
ビットデバイスシフト

SFT(P)

1つ若いデバイスのON/OFF状態を指定したデバイスにシフトして、1つ若いデバイスをOFFします。

ラダー	ST
	ENO:=SFT(EN,d); ENO:=SFTP(EN,d);

FBD/LD



■実行条件

命令	実行条件
SFT	
SFTP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	シフトするデバイス	—	ビット	ANY_BOOL
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (DY)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	○	○	○	○	—	—	—	○	—	—	—	○

機能

■ビットデバイスの場合

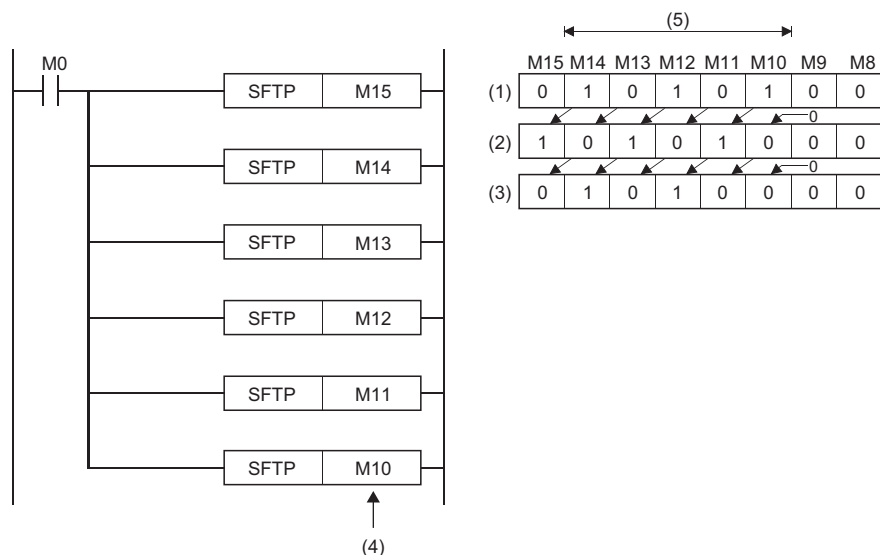
- ・(d)で指定したデバイスの1つ前のデバイスのON/OFF状態を、(d)で指定のデバイスにシフトします。(d)で指定したデバイスの1つ前のデバイスは、OFFになります。

Point

連続でSFT(P)命令を用いる場合は、デバイス番号の大きいものからプログラムを作成します。

例

SFTP命令でM10からM11までのデバイスをシフトした場合は下記ようになります。



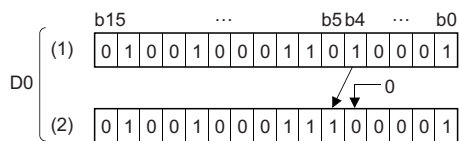
- (1): 初期状態
(2): 1回目のシフト入力後
(3): 2回目のシフト入力後
(5): シフト範囲

■ワードデバイスのビット指定の場合

- ・(d)で指定したビットの1つ前のビットの1/0状態を、(d)で指定のビットにシフトします。(d)で指定したビットの1つ前のビットは、0になります。

例

SFT(P)命令でD0.5(D0のビット5(b5))が指定された場合は、SFT(P)命令実行時にD0のb4の1/0をb5にシフトし、b4を0にします。



- (1): シフト実行前
(2): シフト実行後

エラー

演算エラーはありません。

11.5 マスタコントロール命令

マスタコントロールのセット, リセット

MC, MCR

- MC: マスタコントロールを開始します。
- MCR: マスタコントロールを終了します。

ラダー	ST
(1): マスタコントロール回路	<pre>ENO:=MC(EN,N,d); ENO:=MCR(EN,N);</pre>

FBD/LD	
MC	MCR

■実行条件

命令	実行条件
MC MCR	常時実行

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(N)	ネスティング	N0~N14	デバイス名	ANY16_S ^{*1}
(d)	ONさせるデバイス番号	—	ビット	ANY_BOOL
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 デバイス(N)に割り付けたラベル, または定数を割り付けたラベルのみ使用できます。

■使用可能デバイス

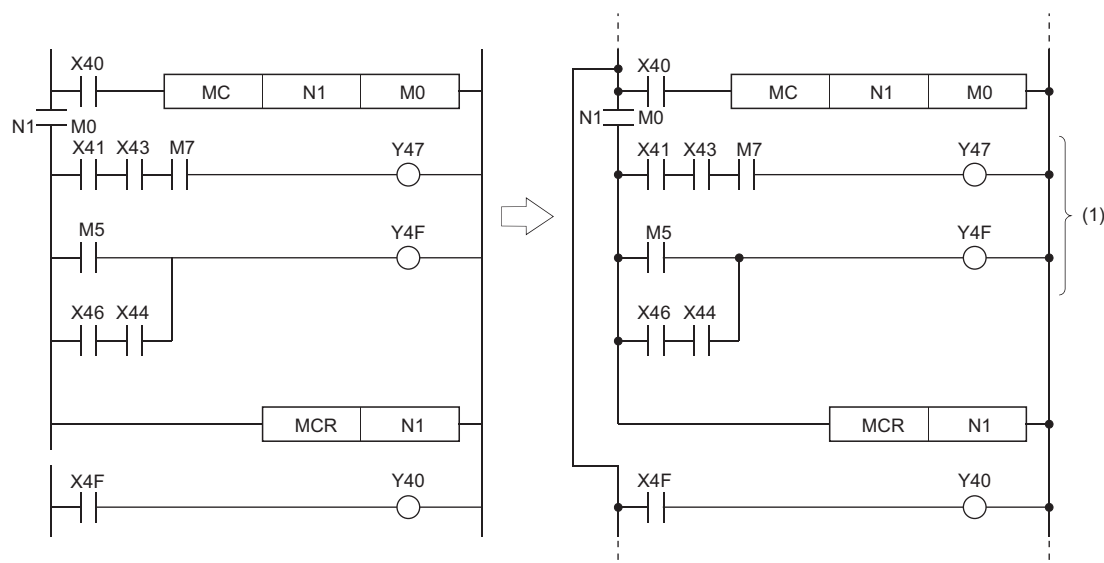
オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他	
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	N	DY
(N)	—	—	—	—	—	—	—	—	—	—	○	—
(d)	○	○	○	—	—	—	○	—	—	—	—	○

機能

マスタコントロール命令は回路の共通母線を開閉することによって、効率のよい回路切換えのプログラムを作成するための命令です。

マスタコントロールを使用した回路は下記のようになります。

(左: エンジニアリングツールの表示, 右: 実際の動作回路)



(1) X40がONのときのみ実行

■MC

- マスタコントロールの開始で、MC命令の実行指令がONの場合、MC命令からMCR命令間の演算結果は命令(回路)どおりとなります。MCの実行指令がOFFの場合、MC命令からMCR命令間の演算結果は下記のようになります。

デバイス	デバイスの状態
高速タイマ 低速タイマ	カウント値は0となり、コイル、接点ともOFFします。
高速積算タイマ 低速積算タイマ カウンタ	コイルはOFFになるがカウント値、接点とも現在の状態を保持します。
OUT命令中のデバイス	強制的にOFFします。
SET命令、RST命令中のデバイス SFT(P)命令中のデバイス 基本命令、応用命令中のデバイス	現在の状態を保持します。

- MC命令がOFFの場合もMC命令からMCR命令間の命令は実行されるので、スキャンタイムは短くなりません。

Point

- マスタコントロールを使用した回路中に、接点命令が不要な命令(FOR~NEXT命令など)がある場合、コントローラはMC命令の実行指令に関係なくその命令を実行します。
- 分かりやすいプログラムを作成するために、対になるMC命令とMCR命令は一つのプログラムブロック内で使用してください。

- MC命令は、(d)のデバイスを変えることにより、同一ネスティング(N)番号を何回でも使用できます。
- MC命令がON時は、(d)で指定されたデバイスのコイルがONします。またOUT命令などで同一のデバイスを使用するとダブルコイルとなりますので、(d)で指定されたデバイスはほかの命令中に使用しないでください。

■MCR

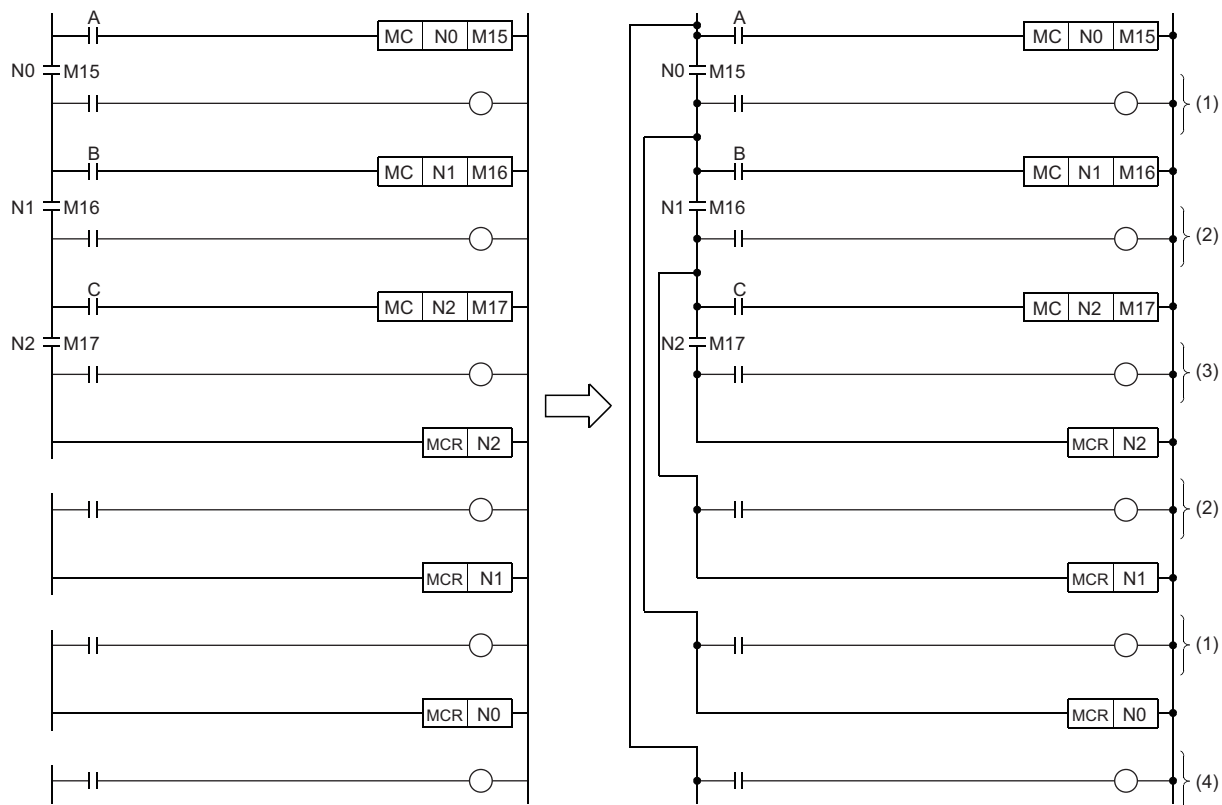
- マスタコントロールの解除命令で、マスタコントロールの範囲の終了を示します。
- MCR命令の前には接点命令はつけないでください。
- 同一ネスティング番号のMC命令とMCR命令をセットで使用します。ただし、MCR命令が1箇所集まっている入れ子構造のときは、最も若いネスティング(N)の番号1つで、すべてのマスタコントロールを終了させることができます。(注意事項参照)

Point

マスタコントロール命令は、入れ子構造にして使用できます。それぞれのマスタコントロール区間はネスティング(N)によって区別します。ネスティングはN0~N14まで使用できます。

入れ子構造を使用することによって、プログラムの実行条件を順々に制約していく回路を作成できます。入れ子構造を使った回路は下記ようになります。

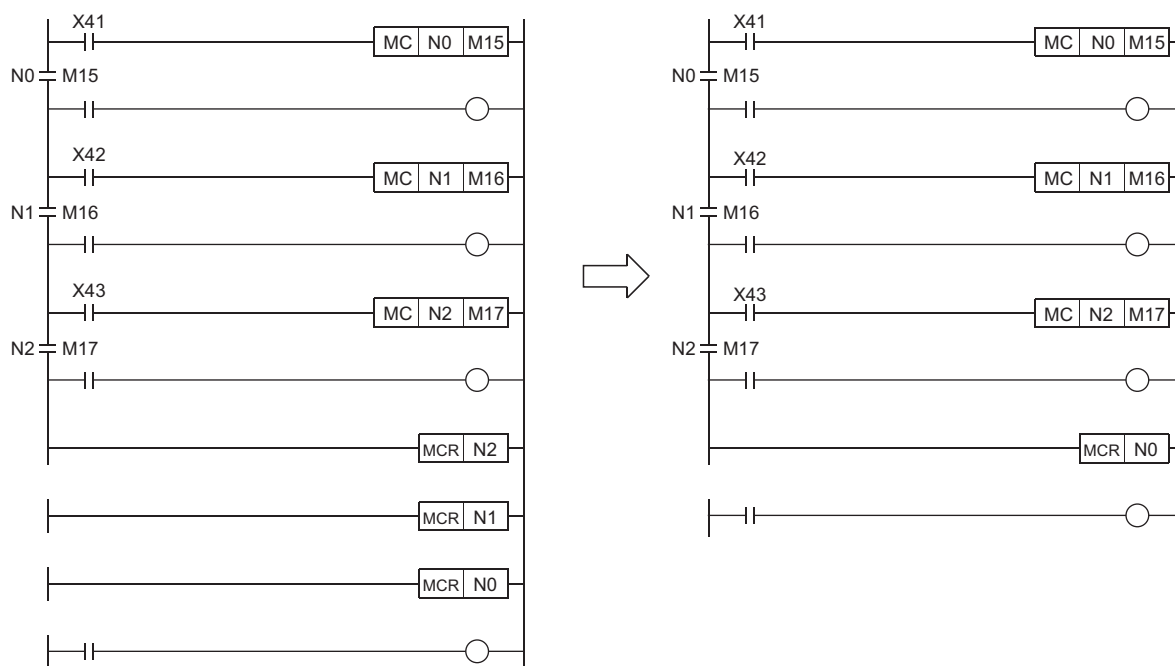
(左: エンジニアリングツールの表示, 右: 実際の動作回路)



- (1) AがONのとき実行
- (2) A, BがONのとき実行
- (3) A, B, CがONのとき実行
- (4) A, B, Cに関係なし

注意事項

- 入れ子は15個(N0~N14)までが可能です。入れ子にする場合、MC命令ではネスティング(N)の若番から用いていき、MCR命令は老番から用います。
- MCR命令が1箇所集まっている入れ子構造のときは、最も若いネスティング(N)の番号1つで、すべてのマスタコントロールを終了させることができます。



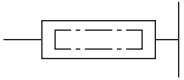
- MCR命令をST言語およびFBD/LD言語で使用了場合、EN(実行条件)に常時ON(TRUE/1/SM400)以外の変数や定数を指定すると変換エラーになります。

11.6 終了命令

メインルーチンプログラム終了

FEND

プログラムファイル内で、メインルーチンプログラムと、サブルーチンプログラム、割込みプログラムを分割するとき使用します。

ラダー	ST
	対応していません。

FBD/LD

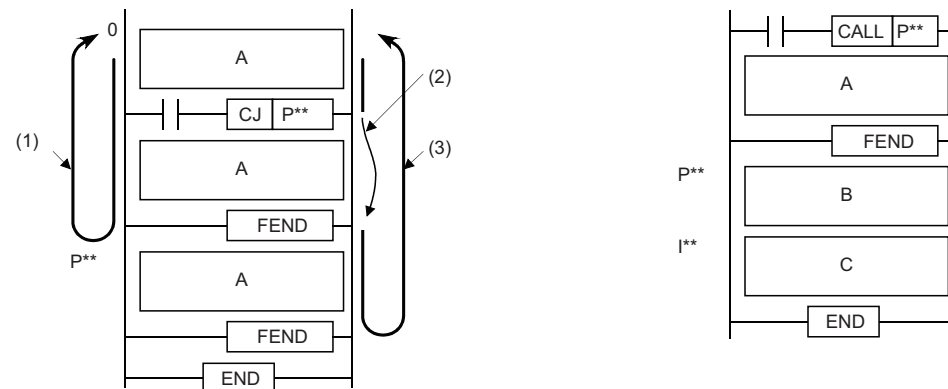
対応していません。

■実行条件

命令	実行条件
FEND	常時実行

機能

- FEND命令は、CJ命令などによりシーケンスプログラムの演算を分岐させる場合や、メインルーチンプログラムをサブルーチンプログラム、割込みポイント(I)による割込みプログラムに分割するとき使用します。
- FEND命令を実行すると、コントローラは実行していたプログラムを終了します。
- FEND命令以降のシーケンスプログラムもエンジニアリングツールで回路表示できます。
CJ命令を用いる場合 サブルーチンプログラム、割込みプログラムのある場合



A: メインルーチンプログラム

B: サブルーチンプログラム

C: 割込みプログラム

(1) CJ命令を実行しないときの演算

(2) CJ命令によるジャンプ

(3) CJ命令を実行したときの演算

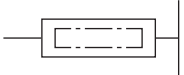
エラー

エラーコード (SD0)	内容
3341H	CALL(P)命令, FCALL(P)命令, ECALL(P)命令, EFCALL(P)命令を実行後、RET命令を実行する前にFEND命令を実行したとき。
3351H	割込みポイント(I)による割込みプログラム中でIRET命令を実行する前に、FEND命令を実行したとき。

シーケンスプログラム終了

END

プログラムの最終を示します。

ラダー	ST
	対応していません。

FBD/LD

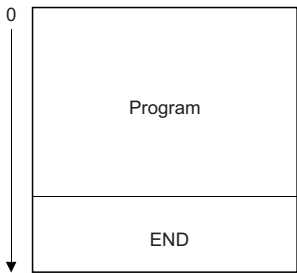
対応していません。

■実行条件

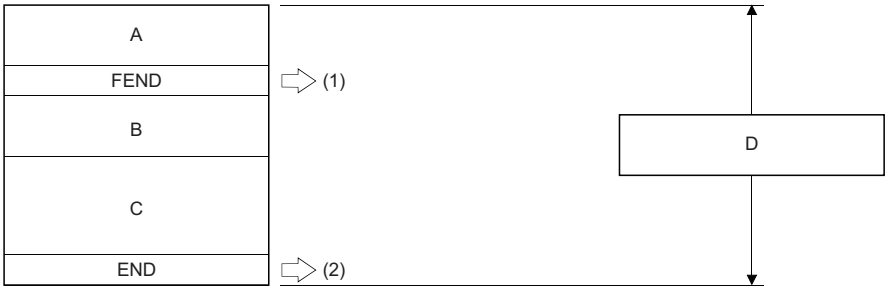
命令	実行条件
END	常時実行

機能

- メインルーチンプログラム，サブルーチンプログラム，割込みプログラムを含めたプログラムの最終を示します。
- END命令を実行すると，コントローラは実行していたプログラムを終了します。



- プログラムの途中でEND処理を必要とする場合は，FEND命令を使用してください。
- エンジニアリングツールの回路編集モードでプログラミングする場合，END命令は自動で入力され，編集はできません。
- メインルーチンプログラム，サブルーチンプログラム，割込みプログラムのある場合の終了命令の使い分けは，下記のようになります。



- A: メインルーチンプログラム
B: サブルーチンプログラム
C: 割込みプログラム
D: メインシーケンスプログラムエリア
(1): FEND命令が必要です。
(2): END命令が必要です。

Point

プログラムを複数のプログラムブロックに分けている場合のEND命令は、プログラムブロックの終了を示します。

END処理として実行するEND命令は、プログラム設定に登録したプログラムの最後に実行するプログラムになります。

エラー

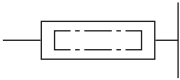
エラーコード (SD0)	内容
3341H	CALL(P)命令, FCALL(P)命令, ECALL(P)命令, EFCALL(P)命令を実行後, RET命令を実行する前にEND命令を実行したとき。
3351H	割込みポインタ(I)による割込みプログラム中でIRET命令を実行する前に, END命令を実行したとき。

11.7 停止命令

シーケンスプログラム停止

STOP

コントローラの演算を停止します。(スイッチをSTOP側にした場合と同一です。)

ラダー	ST
	ENO:=STOP(EN);

FBD/LD



■実行条件

命令	実行条件
STOP	

機能

- 実行指令がONすると、出力(Y)をリセットし、コントローラの演算を停止します。(スイッチをSTOP側にした場合と同一です。)
- STOP命令実行後、コントローラの演算を再開させるには、スイッチをRUN→STOPに戻した後、再度RUNの位置にします。

エラー

エラーコード (SD0)	内容
3341H	CALL(P)命令、FCALL(P)命令、ECALL(P)命令、EFCALL(P)命令、XCALL命令を実行後、RET命令を実行する前にSTOP命令を実行したとき。
3351H	割込みポイント(I)による割込みプログラム中でIRET命令を実行する前に、STOP命令を実行したとき。
3353H	定周期実行タイププログラム中でSTOP命令を実行したとき。
3354H	イベント実行タイププログラム中でSTOP命令を実行したとき。

11.8 無処理命令

11

無処理(NOPLF)

NOPLF

無処理の命令で、それまでの演算に何も影響を与えません。

ラダー	ST
	対応していません。

FBD/LD

対応していません。

■実行条件

命令	実行条件
NOPLF	常時実行

機能

- 無処理の命令で、それまでの演算に何も影響を与えません。

エラー

演算エラーはありません。

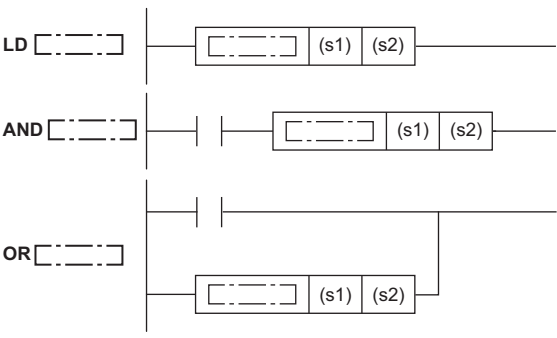
12 基本命令

12.1 比較演算命令

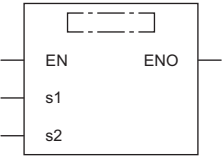
BIN16ビットデータ比較

LD□(_U), AND□(_U), OR□(_U)

指定した2つのBIN16ビットデータをa接点扱いで比較します。

ラダー	ST
	ENO:=LD_□(EN,s1,s2); ENO:=AND_□(EN,s1,s2); ENO:=OR_□(EN,s1,s2); ENO:=LD_□_U(EN,s1,s2); ENO:=AND_□_U(EN,s1,s2); ENO:=OR_□_U(EN,s1,s2); (□には, EQ, NE, GT, LE, LT, GEが入ります。)*¹
(□には, =(U), <>(U), >(U), <=(U), <(U), >=(U)が入ります。)	

FBD/LD



(□には, LD_, AND_, OR_と, EQ(U), NE(U), GT(U), LE(U), LT(U), GE(U)の組合せが入ります。)*¹

*¹ EQは=, NEは<>, GTは>, LEは<=, LTは<, GEは>=です。

■実行条件

命令	実行条件
LD□(_U), AND□(_U), OR□(_U)	常時実行

設定データ

■内容, 範囲, データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(s1)	LD□, AND□, OR□	比較データまたは、比較データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16_S
	LD□_U, AND□_U, OR□_U		0~65535	符号なしBIN16ビット	ANY16_U
(s2)	LD□, AND□, OR□	比較データまたは、比較データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16_S
	LD□_U, AND□_U, OR□_U		0~65535	符号なしBIN16ビット	ANY16_U
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- ・ (s1)で指定したデバイスのBIN16ビットデータと、(s2)で指定したデバイスのBIN16ビットデータを、a接点扱いで比較演算を行います。
- ・ 各命令の比較演算結果は下記になります。

命令記号(ラダー, FBD/LD)	条件	比較演算結果
=(_U), EQ(_U)	(s1)=(s2)	導通状態(ENOはON)
<>(_U), NE(_U)	(s1)≠(s2)	
>(_U), GT(_U)	(s1)>(s2)	
<=(_U), LE(_U)	(s1)≤(s2)	
<(_U), LT(_U)	(s1)<(s2)	
>=(_U), GE(_U)	(s1)≥(s2)	
=(_U), EQ(_U)	(s1)≠(s2)	非導通状態(ENOはOFF)
<>(_U), NE(_U)	(s1)=(s2)	
>(_U), GT(_U)	(s1)≤(s2)	
<=(_U), LE(_U)	(s1)>(s2)	
<(_U), LT(_U)	(s1)≥(s2)	
>=(_U), GE(_U)	(s1)<(s2)	

- ・ LD□, AND□, OR□命令で、(s1), (s2)に16進数の定数を指定する場合、最上位ビット(b15)が1になる数値(8~F)を指定すると、BIN値の負の数とみなし、比較演算を行います。
- ・ LD□命令をFBD/LDで使用した場合、ENに左母線、または常時ONの変数部品/定数部品を指定してください。
- ・ OR□命令をFBD/LDで使用した場合、ENにTRUEを指定するとENOはONします。ENOは実行条件にはなりません。

エラー

演算エラーはありません。

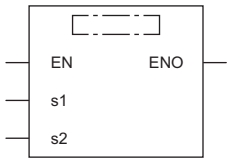
BIN32ビットデータ比較

LDD□(_U), ANDD□(_U), ORD□(_U)

指定した2つのBIN32ビットデータをa接点扱いで比較します。

ラダー	ST
	ENO:=LDD_□(EN,s1,s2); ENO:=ANDD_□(EN,s1,s2); ENO:=ORD_□(EN,s1,s2); ENO:=LDD_□_U(EN,s1,s2); ENO:=ANDD_□_U(EN,s1,s2); ENO:=ORD_□_U(EN,s1,s2); (□には, EQ, NE, GT, LE, LT, GEが入ります。)*¹

FBD/LD



(□には, LDD_, ANDD_, ORD_とEQ(_U), NE(_U), GT(_U), LE(_U), LT(_U), GE(_U)の組合せが入ります。)*¹

*¹ EQは=, NEは<>, GTは>, LEは<=, LTは<, GEは>=です。

■実行条件

命令	実行条件
LDD□(_U), ANDD□(_U), ORD□(_U)	常時実行

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	LDD□, ANDD□, ORD□	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	LDD□_U, ANDD□_U, ORD□_U	0~4294967295	符号なしBIN32ビット	ANY32_U
(s2)	LDD□, ANDD□, ORD□	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	LDD□_U, ANDD□_U, ORD□_U	0~4294967295	符号なしBIN32ビット	ANY32_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	—	—	—

機能

- ・(s1)で指定したデバイスのBIN32ビットデータと、(s2)で指定したデバイスのBIN32ビットデータを、a接点扱いで比較演算を行います。
- ・各命令の比較演算結果は下記になります。

命令記号(ラダー, FBD/LD)	条件	比較演算結果
D=(<u> </u> U), EQ(<u> </u> U)	(s1)=(s2)	導通状態(ENOはON)
D<>(<u> </u> U), NE(<u> </u> U)	(s1)≠(s2)	
D>(<u> </u> U), GT(<u> </u> U)	(s1)>(s2)	
D<=(<u> </u> U), LE(<u> </u> U)	(s1)≤(s2)	
D<(<u> </u> U), LT(<u> </u> U)	(s1)<(s2)	
D>=(<u> </u> U), GE(<u> </u> U)	(s1)≥(s2)	
D=(<u> </u> U), EQ(<u> </u> U)	(s1)≠(s2)	非導通状態(ENOはOFF)
D<>(<u> </u> U), NE(<u> </u> U)	(s1)=(s2)	
D>(<u> </u> U), GT(<u> </u> U)	(s1)≤(s2)	
D<=(<u> </u> U), LE(<u> </u> U)	(s1)>(s2)	
D<(<u> </u> U), LT(<u> </u> U)	(s1)≥(s2)	
D>=(<u> </u> U), GE(<u> </u> U)	(s1)<(s2)	

- ・LDD□, ANDD□, ORD□命令で、(s1), (s2)に16進数の定数を指定する場合、最上位ビット(b31)が1になる数値(8~F)を指定すると、BIN値の負の数とみなし、比較演算を行います。
- ・比較に使用するデータは、32ビットデータを扱う命令(DMOV(P)命令など)で指定してください。16ビットデータを扱う命令(MOV(P)命令など)で指定された場合、大小比較が正常に行えません。
- ・LDD□命令をFBD/LDで使用した場合、ENに左母線、または常時ONの変数部品/定数部品を指定してください。
- ・ORD□命令をFBD/LDで使用した場合、ENにTRUEを指定するとENOはONします。ENは実行条件にはなりません。

エラー

演算エラーはありません。

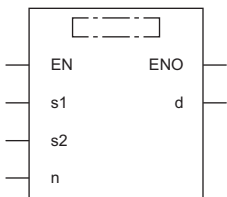
BIN16ビットブロックデータ比較

BKCMPP(P)(_U)

指定した2つのBIN16ビットブロックデータを比較します。

ラダー	ST
	ENO:=BKCMPP_□(EN,s1,s2,n,d); ENO:=BKCMPP_□P(EN,s1,s2,n,d); ENO:=BKCMPP_□_U(EN,s1,s2,n,d); (□には, EQ, NE, GT, LE, LT, GEが入ります。)*1
(□には, BKCMP=(P)(_U), BKCMP<>(P)(_U), BKCMP>(P)(_U), BKCMP<=(P)(_U), BKCMP<(P)(_U), BKCMP>=(P)(_U)が入ります。)	

FBD/LD



(□には, BKCMP_とEQ(P)(_U), NE(P)(_U), GT(P)(_U), LE(P)(_U), LT(P)(_U), GE(P)(_U)の組合せが入ります。)*1

*1 EQは=, NEは<>, GTは>, LEは<=, LTは<, GEは>=です。

■実行条件

命令	実行条件
BKCMPP(P)(_U)	
BKCMPP(P)(_U)	

設定データ

■内容, 範囲, データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(s1)	BKCMPP(P)	比較データまたは、比較データが格納されている先頭デバイス	-32768~32767	符号付きBIN16ビット	ANY16_S ^{*1}
	BKCMPP(P)_U		0~65535	符号なしBIN16ビット	ANY16_U ^{*1}
(s2)	BKCMPP(P)	比較データが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16_S ^{*1}
	BKCMPP(P)_U			符号なしBIN16ビット	ANY16_U ^{*1}
(d)		比較演算結果を格納する先頭デバイス	—	ビット	ANY_BOOL ^{*1}
(n)		比較するデータ数	0~65535	符号なしBIN16ビット	ANY16
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

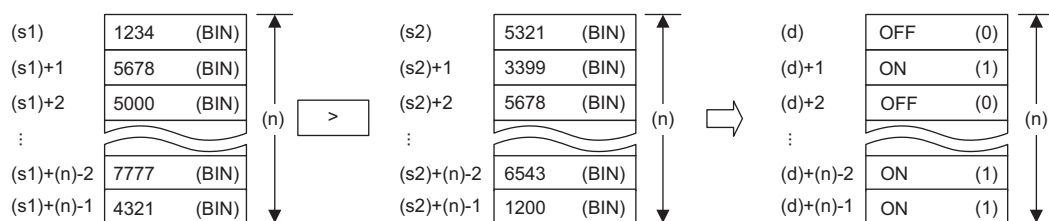
*1 ラベルで設定する場合は, 動作に必要な領域が確保できるよう配列を定義し, その配列型ラベルの要素を指定してください。

■使用可能デバイス

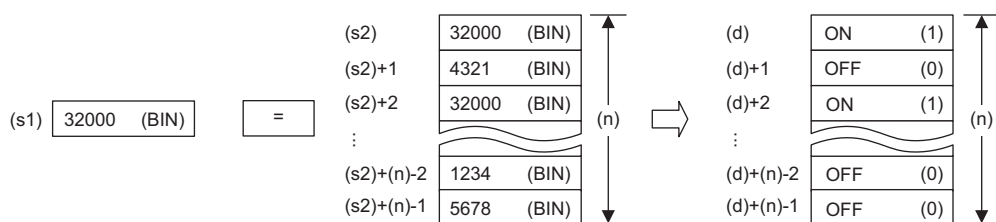
オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	—	—	○	—	—	—	○	○	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—
(d)	○	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

機能

- (s1)で指定したデバイスから(n)点分のBIN16ビットデータと、(s2)で指定したデバイスから(n)点分のBIN16ビットデータを比較し、(d)で指定したデバイス以降に演算結果を格納します。
- 比較条件が成立の場合は(d)の該当デバイスがONし、比較条件が不成立の場合は(d)の該当デバイスがOFFします。



- 比較演算は16ビット単位で行います。
- (s1)には定数が指定できます。



- 各命令の比較演算結果は下記になります。

命令記号(ラダー, FBD/LD)	条件	比較演算結果
BKCMP=(P)(_U), BKCMP_EQ(P)(_U)	(s1)=(s2)	ON(1)
BKCMP<>(P)(_U), BKCMP_NE(P)(_U)	(s1)≠(s2)	
BKCMP>(P)(_U), BKCMP_GT(P)(_U)	(s1)>(s2)	
BKCMP<=(P)(_U), BKCMP_LE(P)(_U)	(s1)≤(s2)	
BKCMP<(P)(_U), BKCMP_LT(P)(_U)	(s1)<(s2)	
BKCMP>=(P)(_U), BKCMP_GE(P)(_U)	(s1)≥(s2)	
BKCMP=(P)(_U), BKCMP_EQ(P)(_U)	(s1)≠(s2)	OFF(0)
BKCMP<>(P)(_U), BKCMP_NE(P)(_U)	(s1)=(s2)	
BKCMP>(P)(_U), BKCMP_GT(P)(_U)	(s1)≤(s2)	
BKCMP<=(P)(_U), BKCMP_LE(P)(_U)	(s1)>(s2)	
BKCMP<(P)(_U), BKCMP_LT(P)(_U)	(s1)≥(s2)	
BKCMP>=(P)(_U), BKCMP_GE(P)(_U)	(s1)<(s2)	

- (d)から(n)点に格納される比較演算結果がすべてON(1)のとき、SM704(ブロック比較信号)がONします。

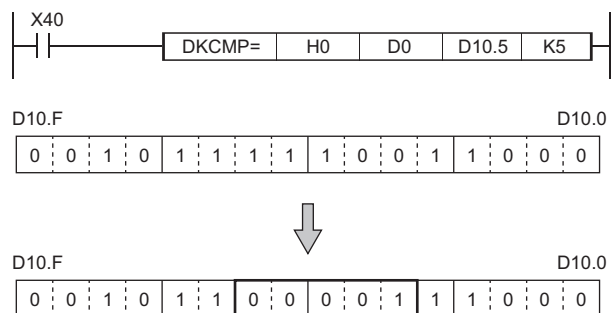
エラー

エラーコード (SD0)	内容
2821H	(s1)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲が重複しているとき。
	(s2)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲が重複しているとき。

Point

ワードデバイスのビット指定をした場合、下記のように演算結果を格納するビット指定デバイス以外は変化しません。

- 比較するデータ数(n): K5
- 比較演算結果を格納するデバイスの先頭番号(d): D10.5



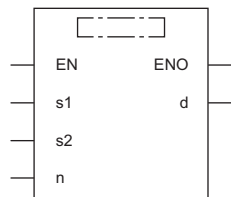
BIN32ビットブロックデータ比較

DBKCMP□(P)(_U)

指定した2つのBIN32ビットブロックデータを比較します。

ラダー	ST
(□には、DBKCMP=(P)(_U), DBKCMP<>(P)(_U), DBKCMP>(P)(_U), DBKCMP<=(P)(_U), DBKCMP<(P)(_U), DBKCMP>=(P)(_U)が入ります。)	ENO:=DBKCMP_□(EN,s1,s2,n,d); ENO:=DBKCMP_□P(EN,s1,s2,n,d); ENO:=DBKCMP_□_U(EN,s1,s2,n,d); ENO:=DBKCMP_□P_U(EN,s1,s2,n,d); (□には、EQ, NE, GT, LE, LT, GEが入ります。)*¹

FBD/LD



(□には、DBKCMP_とEQ(P)_U, NE(P)_U, GT(P)_U, LE(P)_U, LT(P)_U, GE(P)_Uの組合せが入ります。)*¹

*¹ EQは=, NEは<>, GTは>, LEは<=, LTは<, GEは>=です。

■実行条件

命令	実行条件
DBKCMP□(_U)	
DBKCMP□P(_U)	

設定データ

■内容、範囲、データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(s1)	DBKCMP□(P)	比較データまたは、比較データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S* ¹
	DBKCMP□(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U* ¹
(s2)	DBKCMP□(P)	比較データが格納されている先頭デバイス	—	符号付きBIN32ビット	ANY32_S* ¹
	DBKCMP□(P)_U			符号なしBIN32ビット	ANY32_U* ¹
(d)		比較演算結果を格納する先頭デバイス	—	ビット	ANY_BOOL* ¹
(n)		比較するデータ数	0~65535	符号なしBIN16ビット	ANY16
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

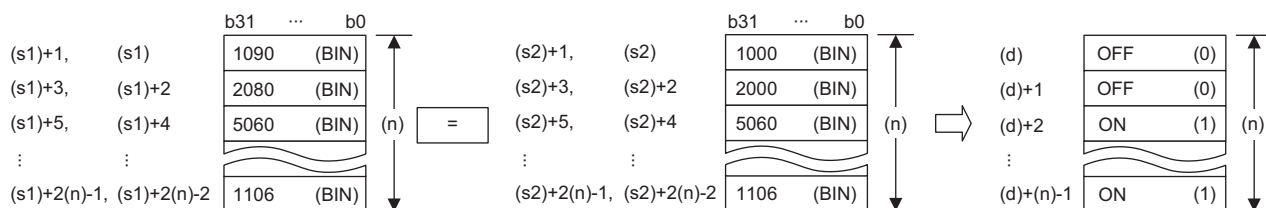
*¹ ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

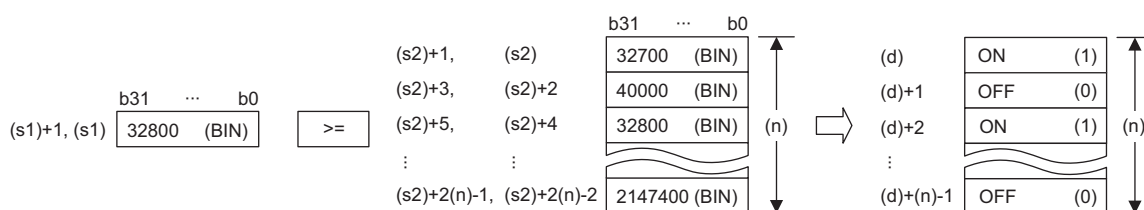
オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□G□, J□¥□	Z	LT, LST, LC	LZ	K, H		E	\$		
(s1)	—	—	○	—	—	—	—	○	○	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	○	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

機能

- (s1)で指定したデバイスから(n)点分のBIN32ビットデータと、(s2)で指定したデバイスから(n)点分のBIN32ビットデータを比較し、(d)で指定したデバイス以降に演算結果を格納します。
- 比較条件が成立の場合は(d)の該当デバイスがONし、比較条件が不成立の場合は(d)の該当デバイスがOFFします。



- 比較演算は、32ビット単位で行います。
- (s1)には定数が指定できます。



- (d)は(s1)から(n)点分のデバイス範囲と(s2)から(n)点分のデバイス範囲以外で指定します。
- 各命令の比較演算結果は下記になります。

命令記号(ラダー, FBD/LD)	条件	比較演算結果
DBKCMP=(P)(_U), DBKCMP_EQ(P)(_U)	(s1)=(s2)	ON(1)
DBKCMP<>(P)(_U), DBKCMP_NE(P)(_U)	(s1)≠(s2)	
DBKCMP>(P)(_U), DBKCMP_GT(P)(_U)	(s1)>(s2)	
DBKCMP<=(P)(_U), DBKCMP_LE(P)(_U)	(s1)≤(s2)	
DBKCMP<(P)(_U), DBKCMP_LT(P)(_U)	(s1)<(s2)	
DBKCMP>=(P)(_U), DBKCMP_GE(P)(_U)	(s1)≥(s2)	
DBKCMP=(P)(_U), DBKCMP_EQ(P)(_U)	(s1)≠(s2)	OFF(0)
DBKCMP<>(P)(_U), DBKCMP_NE(P)(_U)	(s1)=(s2)	
DBKCMP>(P)(_U), DBKCMP_GT(P)(_U)	(s1)≤(s2)	
DBKCMP<=(P)(_U), DBKCMP_LE(P)(_U)	(s1)>(s2)	
DBKCMP<(P)(_U), DBKCMP_LT(P)(_U)	(s1)≥(s2)	
DBKCMP>=(P)(_U), DBKCMP_GE(P)(_U)	(s1)<(s2)	

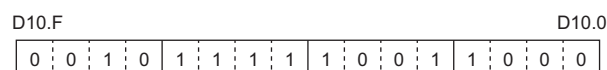
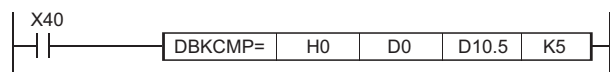
- (d)から(n)点に格納される比較演算結果がすべてON(1)のとき、SM704(ブロック比較信号)がONします。
- (n)で指定した値が0の場合は、無処理となります。

エラーコード (SD0)	内容
2821H	(s1)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲が重複しているとき。
	(s2)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲が重複しているとき。

Point

ワードデバイスのビット指定をした場合、下記のように演算結果を格納するビット指定デバイス以外は変化しません。

- 比較するデータ数(n): K5
- 比較演算結果を格納するデバイスの先頭番号(d): D10.5

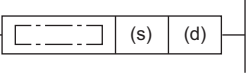


12.2 算術演算命令

BIN16ビット加算

+(P)(_U) [オペランドが2個の場合]

指定した2つのBIN16ビットデータを加算します。

ラダー	ST
	対応していません。 ( 264ページ +(P)(_U) [オペランドが3個の場合])

FBD/LD
対応していません。 ( 264ページ +(P)(_U) [オペランドが3個の場合])

■実行条件

命令	実行条件
+ +_U	
+P +P_U	

設定データ

■内容, 範囲, データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(s)	+(P)	加算データまたは、加算データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16_S
	+(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
(d)	+(P)	加算されるデータが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16_S
	+(P)_U		0~65535	符号なしBIN16ビット	ANY16_U

■使用可能デバイス

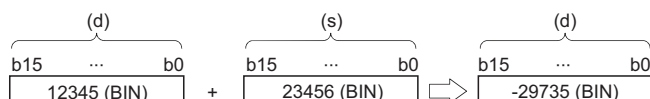
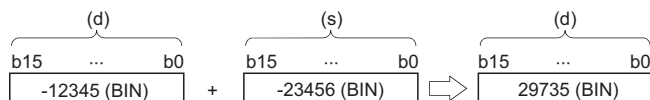
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

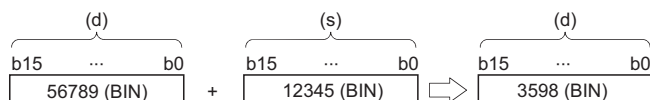
- (d)で指定したBIN16ビットデータと、(s)で指定したBIN16ビットデータの加算を行い、加算結果を(d)で指定したデバイスに格納します。



- 加算結果にオーバーフローが生じた場合、桁上げは無視されます。この場合、SM700(キャリフラグ)はONしません。
[+(P)の場合]



[+(P)_Uの場合]



エラー

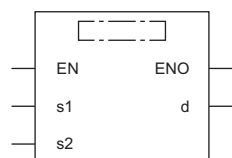
演算エラーはありません。

+ (P) (_ U) [オペランドが3個の場合]

指定した2つのBIN16ビットデータを加算します。

ラダー	ST
	<pre> ENO:=PLUS(EN,s1,s2,d); ENO:=PLUSP(EN,s1,s2,d); ENO:=PLUS_U(EN,s1,s2,d); ENO:=PLUSP_U(EN,s1,s2,d); </pre>

FBD/LD



(□には、PLUS、PLUSP、PLUS_U、PLUSP_Uが入ります。)

■実行条件

命令	実行条件
+ +_U	
+P +P_U	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	+(P)	-32768~32767	符号付きBIN16ビット	ANY16_S
	+(P)_U	0~65535	符号なしBIN16ビット	ANY16_U
(s2)	+(P)	-32768~32767	符号付きBIN16ビット	ANY16_S
	+(P)_U	0~65535	符号なしBIN16ビット	ANY16_U
(d)	+(P)	—	符号付きBIN16ビット	ANY16_S
	+(P)_U	—	符号なしBIN16ビット	ANY16_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

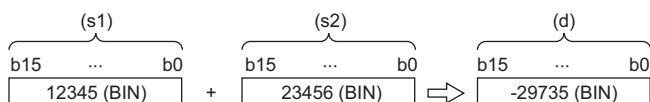
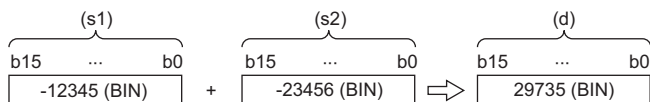
オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

機能

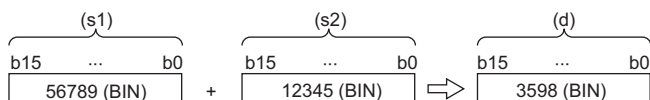
- (s1)で指定したBIN16ビットデータと、(s2)で指定したBIN16ビットデータの加算を行い、加算結果を(d)で指定したデバイスに格納します。



- 加算結果にオーバーフローが生じた場合、桁上げは無視されます。この場合、SM700(キャリフラグ)はONしません。
[+(P)の場合]



[+(P)_Uの場合]



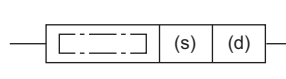
エラー

演算エラーはありません。

BIN16ビット減算

-(P)(_U) [オペランドが2個の場合]

指定した2つのBIN16ビットデータを減算します。

ラダー	ST
	対応していません。 ( 268ページ -(P)(_U) [オペランドが3個の場合])

FBD/LD
対応していません。 ( 268ページ -(P)(_U) [オペランドが3個の場合])

■実行条件

命令	実行条件
- -_U	
-P -P_U	

設定データ

■内容, 範囲, データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(s)	-(P)	減算データまたは、減算データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16_S
	-(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
(d)	-(P)	減算されるデータが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16_S
	-(P)_U		0~65535	符号なしBIN16ビット	ANY16_U

■使用可能デバイス

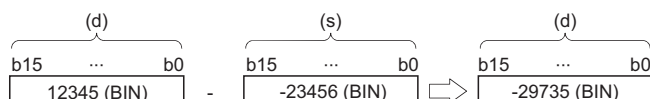
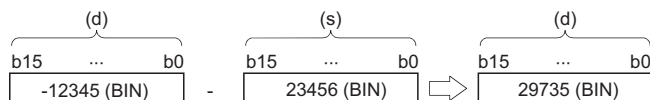
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

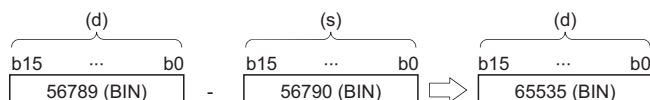
- (d)で指定したBIN16ビットデータと、(s)で指定したBIN16ビットデータの減算を行い、減算結果を(d)で指定したデバイスに格納します。



- 減算結果にアンダーフローが生じた場合、桁下げは無視されます。この場合、SM700(キャリフラグ)はONしません。
[-(P)の場合]



[-(P)_Uの場合]



エラー

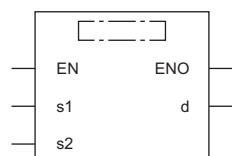
演算エラーはありません。

-(P)(_U) [オペランドが3個の場合]

指定した2つのBIN16ビットデータを減算します。

ラダー	ST
	<pre> ENO:=MINUS(EN,s1,s2,d); ENO:=MINUSP(EN,s1,s2,d); ENO:=MINUS_U(EN,s1,s2,d); ENO:=MINUSP_U(EN,s1,s2,d); </pre>

FBD/LD



(□には、MINUS、MINUSP、MINUS_U、MINUSP_Uが入ります。)

■実行条件

命令	実行条件
- -_U	
-P -P_U	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	-(P)	-32768~32767	符号付きBIN16ビット	ANY16_S
	-(P)_U	0~65535	符号なしBIN16ビット	ANY16_U
(s2)	-(P)	-32768~32767	符号付きBIN16ビット	ANY16_S
	-(P)_U	0~65535	符号なしBIN16ビット	ANY16_U
(d)	-(P)	—	符号付きBIN16ビット	ANY16_S
	-(P)_U	—	符号なしBIN16ビット	ANY16_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

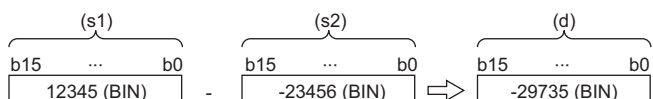
オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

機能

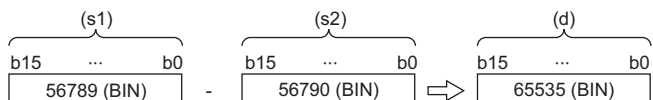
- (s1)で指定したBIN16ビットデータと、(s2)で指定したBIN16ビットデータの減算を行い、減算結果を(d)で指定したデバイスに格納します。



- 減算結果にアンダーフローが生じた場合、桁下げは無視されます。この場合、SM700(キャリフラグ)はONしません。
[-(P)の場合]



[-(P)_Uの場合]



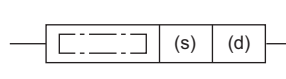
エラー

演算エラーはありません。

BIN32ビット加算

D+(P)(_U) [オペランドが2個の場合]

指定した2つのBIN32ビットデータを加算します。

ラダー	ST
	対応していません。 (参照 272ページ D+(P)(_U) [オペランドが3個の場合])
FBD/LD	
対応していません。 (参照 272ページ D+(P)(_U) [オペランドが3個の場合])	

■実行条件

命令	実行条件
D+ D+_U	
D+P D+P_U	

設定データ

■内容, 範囲, データ型

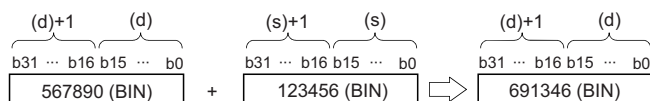
オペランド		内容	範囲	データ型	データ型(ラベル)
(s)	D+(P)	加算データまたは、加算データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	D+(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U
(d)	D+(P)	加算されるデータが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	D+(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U

■使用可能デバイス

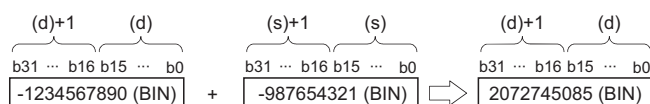
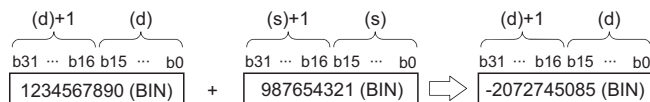
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

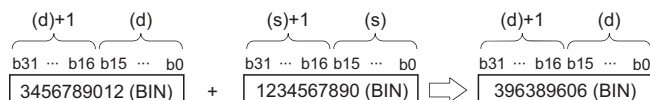
- (d)で指定したBIN32ビットデータと、(s)で指定したBIN32ビットデータの加算を行い、加算結果を(d)で指定したデバイスに格納します。



- 加算結果にオーバーフローが生じた場合、桁上げは無視されます。この場合、SM700(キャリフラグ)はONしません。
[D+(P)の場合]



[D+(P)_Uの場合]

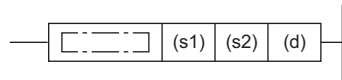


エラー

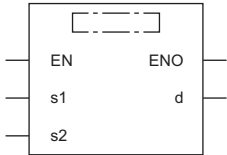
演算エラーはありません。

D+(P)(_U) [オペランドが3個の場合]

指定した2つのBIN32ビットデータを加算します。

ラダー	ST
	ENO:=DPLUS(EN,s1,s2,d); ENO:=DPLUSP(EN,s1,s2,d); ENO:=DPLUS_U(EN,s1,s2,d); ENO:=DPLUSP_U(EN,s1,s2,d);

FBD/LD



(□には、DPLUS, DPLUSP, DPLUS_U, DPLUSP_Uが入ります。)

■実行条件

命令	実行条件
D+ D+_U	
D+P D+P_U	

設定データ

■内容、範囲、データ型

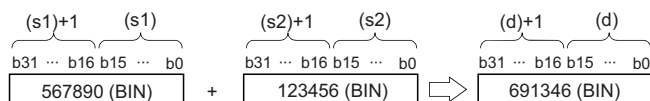
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	D+(P)	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	D+(P)_U	0~4294967295	符号なしBIN32ビット	ANY32_U
(s2)	D+(P)	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	D+(P)_U	0~4294967295	符号なしBIN32ビット	ANY32_U
(d)	D+(P)	—	符号付きBIN32ビット	ANY32_S
	D+(P)_U	—	符号なしBIN32ビット	ANY32_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

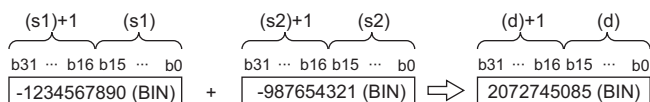
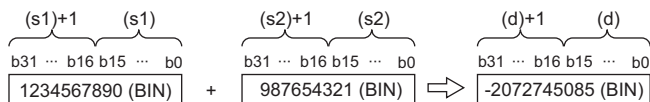
オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□	G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—	—

機能

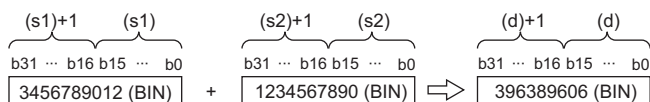
- (s1)で指定したBIN32ビットデータと、(s2)で指定したBIN32ビットデータの加算を行い、加算結果を(d)で指定したデバイスに格納します。



- 加算結果にオーバーフローが生じた場合、桁上げは無視されます。この場合、SM700(キャリフラグ)はONしません。
[D+(P)の場合]



[D+(P)_Uの場合]



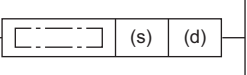
エラー

演算エラーはありません。

BIN32ビット減算

D-(P)(_U) [オペランドが2個の場合]

指定した2つのBIN32ビットデータを減算します。

ラダー	ST
	対応していません。 (参照 276ページ D-(P)(_U) [オペランドが3個の場合])

FBD/LD
対応していません。 (参照 276ページ D-(P)(_U) [オペランドが3個の場合])

■実行条件

命令	実行条件
D- D_U	
D-P D-P_U	

設定データ

■内容, 範囲, データ型

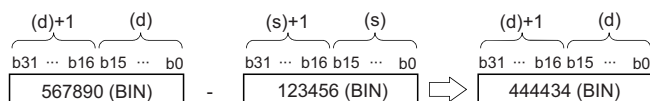
オペランド	内容	範囲	データ型	データ型(ラベル)
(s) D-(P)	減算データまたは, 減算データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
D-(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U
(d) D-(P)	減算されるデータが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
D-(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U

■使用可能デバイス

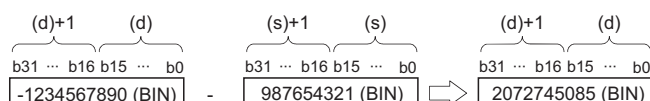
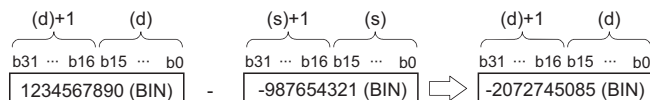
オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—	—

機能

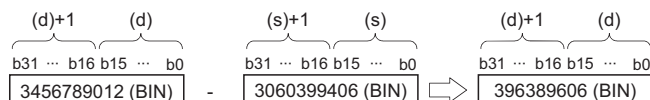
- (d)で指定されたBIN32ビットデータと、(s)で指定されたBIN32ビットデータの減算を行い、減算結果を(d)で指定したデバイスに格納します。



- 減算結果にアンダーフローが生じた場合、桁上げは無視されます。この場合、SM700(キャリフラグ)はONしません。
[D-(P)の場合]



[D-(P)_Uの場合]



エラー

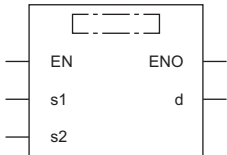
演算エラーはありません。

D-(P)(_U) [オペランドが3個の場合]

指定した2つのBIN32ビットデータを減算します。

ラダー	ST
	ENO:=DMINUS(EN,s1,s2,d); ENO:=DMINUSP(EN,s1,s2,d); ENO:=DMINUS_U(EN,s1,s2,d); ENO:=DMINUSP_U(EN,s1,s2,d);

FBD/LD



(□には、DMINUS、DMINUSP、DMINUS_U、DMINUSP_Uが入ります。)

■実行条件

命令	実行条件
D- D-_U	
D-P D-P_U	

設定データ

■内容、範囲、データ型

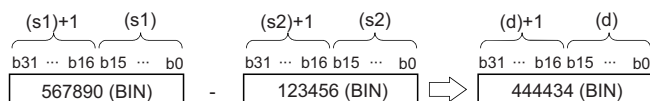
オペランド		内容	範囲	データ型	データ型(ラベル)
(s1)	D-(P)	減算されるデータまたは、減算されるデータが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	D-(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U
(s2)	D-(P)	減算データまたは、減算データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	D-(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U
(d)	D-(P)	演算結果を格納する先頭デバイス	—	符号付きBIN32ビット	ANY32_S
	D-(P)_U			符号なしBIN32ビット	ANY32_U
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

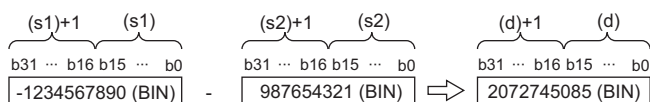
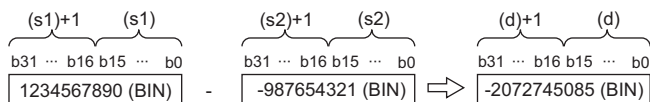
オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—	—

機能

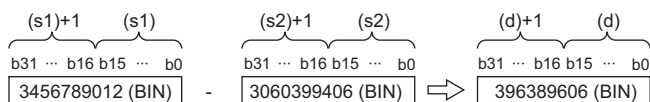
- (s1)で指定したBIN32ビットデータと、(s2)で指定したBIN32ビットデータの減算を行い、減算結果を(d)で指定したデバイスに格納します。



- 減算結果にアンダーフローが生じた場合、桁上げは無視されます。この場合、SM700(キャリフラグ)はONしません。
[D-(P)の場合]



[D-(P)_Uの場合]



エラー

演算エラーはありません。

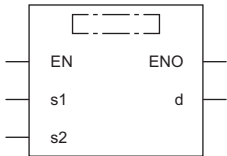
BIN16ビット乗算

*(P)(_U)

指定した2つのBIN16ビットデータを乗算します。

ラダー	ST	
	ENO:=MULTI(EN,s1,s2,d); ENO:=MULTIP(EN,s1,s2,d);	ENO:=MULTI_U(EN,s1,s2,d); ENO:=MULTIP_U(EN,s1,s2,d);

FBD/LD



(□には、MULTI、MULTIP、MULTI_U、MULTIP_Uが入ります。)

■実行条件

命令	実行条件
* *_U	
*P *P_U	

設定データ

■内容、範囲、データ型

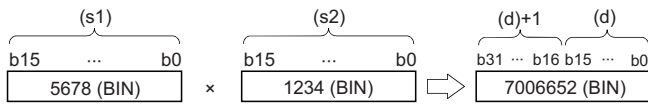
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	*(P)	-32768~32767	符号付きBIN16ビット	ANY16_S
	*(P)_U	0~65535	符号なしBIN16ビット	ANY16_U
(s2)	*(P)	-32768~32767	符号付きBIN16ビット	ANY16_S
	*(P)_U	0~65535	符号なしBIN16ビット	ANY16_U
(d)	*(P)	—	符号付きBIN32ビット	ANY32_S
	*(P)_U	—	符号なしBIN32ビット	ANY32_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ □	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

機能

- (s1)で指定したBIN16ビットデータと、(s2)で指定したBIN16ビットデータの乗算を行い、乗算結果を(d)で指定したデバイスに格納します。



- (d)がビットデバイスの場合には、下位のビットからの指定になります。

例

(d)がビットデバイスの場合の乗算結果

- K1...下位の4ビット(b0~b3)
- K4...下位の16ビット(b0~b15)
- K8...下位の32ビット(b0~b31)

エラー

演算エラーはありません。

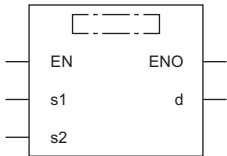
BIN16ビット除算

/ (P) (_ U)

指定した2つのBIN16ビットデータを除算します。

ラダー	ST	
	ENO:=DIVISION(EN,s1,s2,d); ENO:=DIVISIONP(EN,s1,s2,d);	ENO:=DIVISION_U(EN,s1,s2,d); ENO:=DIVISIONP_U(EN,s1,s2,d);

FBD/LD



(□には、DIVISION、DIVISIONP、DIVISION_U、DIVISIONP_Uが入ります。)

■実行条件

命令	実行条件
/	
/_U	
/P	
/P_U	

設定データ

■内容、範囲、データ型

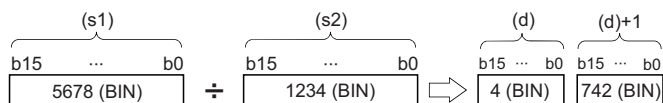
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	/ (P)	-32768~32767	符号付きBIN16ビット	ANY16_S
	/ (P)_U	0~65535	符号なしBIN16ビット	ANY16_U
(s2)	/ (P)	-32768~32767	符号付きBIN16ビット	ANY16_S
	/ (P)_U	0~65535	符号なしBIN16ビット	ANY16_U
(d)	/ (P)	—	符号付きBIN32ビット	ANY16_S_ARRAY (要素数: 2)
	/ (P)_U	—	符号なしBIN32ビット	ANY16_U_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—

機能

- (s1)で指定したBIN16ビットデータと、(s2)で指定したBIN16ビットデータの除算を行い、除算結果を(d)で指定したデバイスに格納します。



(d): 商

(d)+1: 余り

- 除算結果は、32ビットを使用して商と余りを格納します。ビットデバイスの場合は桁指定で指定したビット数を使用し、商および余りを格納します。
- 商...下位16ビットに格納されます。
- 余り...上位16ビットに格納されます。

エラー

エラーコード (SD0)	内容
3280H	(s2)で指定した値(除数)が0のとき。

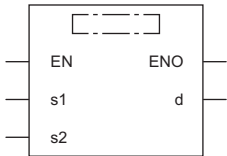
BIN32ビット乗算

D*(P)(_U)

指定した2つのBIN32ビットデータを乗算します。

ラダー	ST	
	ENO:=DMULTI(EN,s1,s2,d); ENO:=DMULTIP(EN,s1,s2,d);	ENO:=DMULTI_U(EN,s1,s2,d); ENO:=DMULTIP_U(EN,s1,s2,d);

FBD/LD



(□には、DMULTI、DMULTIP、DMULTI_U、DMULTIP_Uが入ります。)

■実行条件

命令	実行条件
D* D*_U	
D*P D*P_U	

設定データ

■内容、範囲、データ型

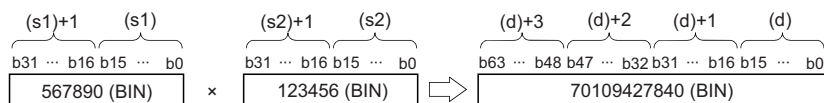
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	D*(P)	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	D*(P)_U	0~4294967295	符号なしBIN32ビット	ANY32_U
(s2)	D*(P)	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	D*(P)_U	0~4294967295	符号なしBIN32ビット	ANY32_U
(d)	D*(P)	—	符号付きBIN64ビット	ANY32_S_ARRAY (要素数: 2)
	D*(P)_U	—	符号なしBIN64ビット	ANY32_U_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	—	○	—	—	○	—	○	—	—	—	—

機能

- (s1)で指定したBIN32ビットデータと、(s2)で指定したBIN32ビットデータの乗算を行い、乗算結果を(d)で指定したデバイスに格納します。



- (d)がビットデバイスの場合には、乗算結果の下位32ビットまでが対象となり、上位32ビットを指定することはできません。ビットデバイスで乗算結果の上位32ビットデータが必要な場合には、一度ワードデバイスにデータを格納しておき、ワードデバイスの(d)+2, (d)+3のデータを指定ビットデバイスに転送してください。

例

(d)がビットデバイスの場合の乗算結果

- K1...下位の4ビット(b0~b3)
- K4...下位の16ビット(b0~b15)
- K8...下位の32ビット(b0~b31)

エラー

演算エラーはありません。

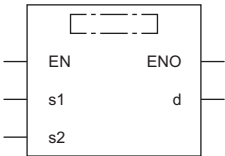
BIN32ビット除算

D/(P)(_U)

指定した2つのBIN32ビットデータを除算します。

ラダー	ST	
	ENO:=DDIVISION(EN,s1,s2,d); ENO:=DDIVISIONP(EN,s1,s2,d);	ENO:=DDIVISION_U(EN,s1,s2,d); ENO:=DDIVISIONP_U(EN,s1,s2,d);

FBD/LD



(□には、DDIVISION、DDIVISIONP、DDIVISION_U、DDIVISIONP_Uが入ります。)

■実行条件

命令	実行条件
D/ D/_U	
D/P D/P_U	

設定データ

■内容、範囲、データ型

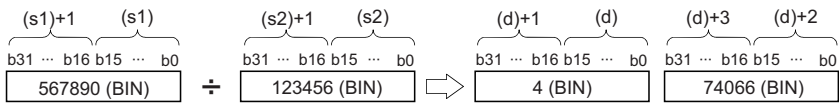
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	D/(P)	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	D/(P)_U	0~4294967295	符号なしBIN32ビット	ANY32_U
(s2)	D/(P)	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	D/(P)_U	0~4294967295	符号なしBIN32ビット	ANY32_U
(d)	D/(P)	—	符号付きBIN64ビット	ANY32_S_ARRAY (要素数: 2)
	D/(P)_U	—	符号なしBIN64ビット	ANY32_U_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	—	○	—	—	○	—	○	—	—	—	—

機能

- (s1)で指定したBIN32ビットデータと，(s2)で指定したBIN32ビットデータの除算を行い，除算結果を(d)で指定したデバイスに格納します。



- ワードデバイスの場合の除算結果は，64ビットを使用して商と余りが格納されます。商は下位32ビットに格納され，余りは上位32ビットに格納されます。ビットデバイスの場合は，32ビットを使用して商のみが格納されます。

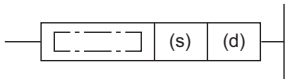
エラー

エラーコード (SD0)	内容
3280H	(s2)で指定した値(除数)が0のとき。

BCD4桁加算

B+(P) [オペランドが2個の場合]

指定した2つのBCD4桁データを加算します。

ラダー	ST
	対応していません。 (287ページ B+(P) [オペランドが3個の場合])
FBD/LD	
対応していません。 (287ページ B+(P) [オペランドが3個の場合])	

■実行条件

命令	実行条件
B+	
B+P	

設定データ

■内容、範囲、データ型

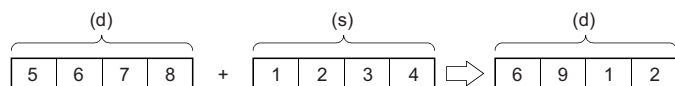
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	加算データまたは、加算データが格納されているデバイス	0~9999	BCD4桁	ANY16
(d)	加算されるデータが格納されているデバイス	0~9999	BCD4桁	ANY16

■使用可能デバイス

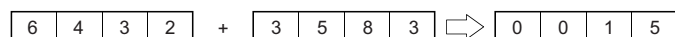
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 0	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 0	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- (d)で指定したBCD4桁データと、(s)で指定したBCD4桁データの加算を行い、加算結果を(d)で指定したデバイスに格納します。



- 加算結果が9999を超えた場合は、桁上げは無視されます。この場合、SM700(キャリフラグ)はONしません。



エラー

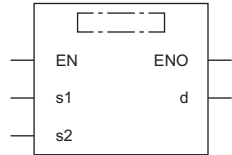
エラーコード (SD0)	内容
3285H	(s)のBCDデータが0~9999以外のとき。
	(d)のBCDデータが0~9999以外のとき。

B+(P) [オペランドが3個の場合]

指定した2つのBCD4桁データを加算します。

ラダー	ST
	ENO:=BPLUS(EN,s1,s2,d); ENO:=BPLUSP(EN,s1,s2,d);

FBD/LD



(□には、BPLUS、BPLUSPが入ります。)

■実行条件

命令	実行条件
B+	
B+P	

設定データ

■内容、範囲、データ型

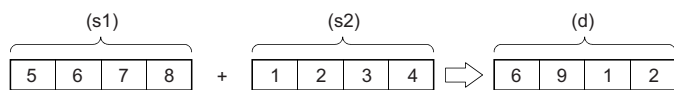
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	加算されるデータまたは、加算されるデータが格納されているデバイス	0~9999	BCD4桁	ANY16
(s2)	加算データまたは、加算データが格納されているデバイス	0~9999	BCD4桁	ANY16
(d)	演算結果を格納するデバイス	—	BCD4桁	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

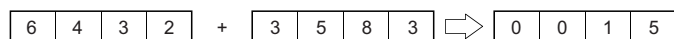
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- (s1)で指定したBCD4桁データと、(s2)で指定したBCD4桁データの加算を行い、加算結果を(d)で指定したデバイスに格納します。



- 加算結果が9999を超えた場合は、桁上げは無視されます。この場合、SM700(キャリフラグ)はONしません。



エラー

エラーコード (SD0)	内容
3285H	(s1)のBCDデータが0~9999以外るとき。
	(s2)のBCDデータが0~9999以外るとき。

BCD4桁減算

12

B-(P) [オペランドが2個の場合]

指定した2つのBCD4桁データを減算します。

ラダー	ST
	対応していません。 (参照 290ページ B-(P) [オペランドが3個の場合])

FBD/LD
対応していません。 (参照 290ページ B-(P) [オペランドが3個の場合])

■実行条件

命令	実行条件
B-	
B-P	

設定データ

■内容、範囲、データ型

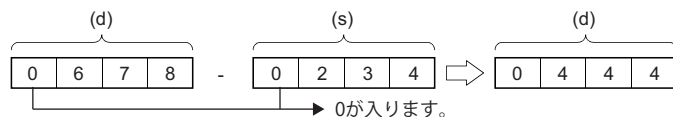
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	減算データまたは、減算データが格納されているデバイス	0~9999	BCD4桁	ANY16
(d)	減算されるデータが格納されているデバイス	0~9999	BCD4桁	ANY16

■使用可能デバイス

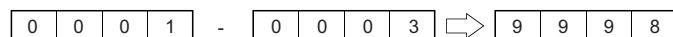
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- ・ (s)で指定したBCD4桁データと、(d)で指定したBCD4桁データの減算を行い、減算結果を(d)で指定したデバイスに格納します。



- ・ 減算結果にアンダーフローが生じた場合は、下記のようになります。この場合、SM700(キャリフラグ)はONしません。



エラー

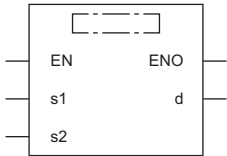
エラーコード (SD0)	内容
3285H	(s)のBCDデータが0~9999以外のとき。 (d)のBCDデータが0~9999以外のとき。

B-(P) [オペランドが3個の場合]

指定した2つのBCD4桁データを減算します。

ラダー	ST
	ENO:=BMINUS(EN,s1,s2,d); ENO:=BMINUSP(EN,s1,s2,d);

FBD/LD



(□には、BMINUS、BMINUSPが入ります。)

■実行条件

命令	実行条件
B-	
B-P	

設定データ

■内容、範囲、データ型

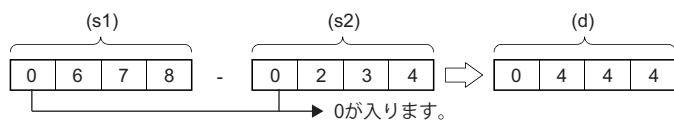
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	減算されるデータまたは、減算されるデータが格納されているデバイス	0~9999	BCD4桁	ANY16
(s2)	減算データまたは、減算データが格納されているデバイス	0~9999	BCD4桁	ANY16
(d)	演算結果を格納するデバイス	—	BCD4桁	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

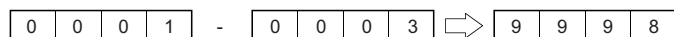
オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

機能

- (s1)で指定されたBCD4桁データと、(s2)で指定されたBCD4桁データの減算を行い、減算結果を(d)で指定したデバイスに格納します。



- 減算結果にアンダーフローが生じた場合は、下記のようになります。この場合、SM700(キャリフラグ)はONしません。



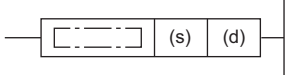
エラー

エラーコード (SD0)	内容
3285H	(s1)のBCDデータが0~9999以外るとき。
	(s2)のBCDデータが0~9999以外るとき。

BCD8桁加算

DB+(P) [オペランドが2個の場合]

指定した2つのBCD8桁データを加算します。

ラダー	ST
	対応していません。 (293ページ DB+(P) [オペランドが3個の場合])

FBD/LD
対応していません。 (293ページ DB+(P) [オペランドが3個の場合])

■実行条件

命令	実行条件
DB+	
DB+P	

設定データ

■内容、範囲、データ型

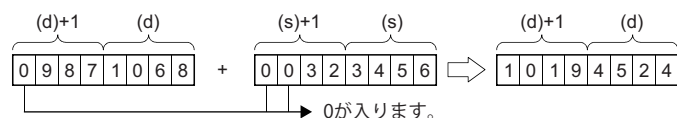
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	加算データまたは、加算データが格納されている先頭デバイス	0~99999999	BCD8桁	ANY32
(d)	加算されるデータが格納されている先頭デバイス	0~99999999	BCD8桁	ANY32

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 0	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 0	Z	LT, LST, LC		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

機能

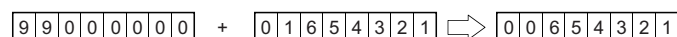
- (d)で指定したBCD8桁データと、(s)で指定したBCD8桁データの加算を行い、加算結果を(d)で指定したデバイスに格納します。



(d)+1, (s)+1: 上位4桁

(d), (s): 下位4桁

- 加算結果が99999999を超えた場合は、桁上げは無視されます。この場合、SM700(キャリフラグ)はONしません。



エラー

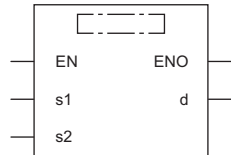
エラーコード (SD0)	内容
3285H	(s)のBCDデータが0~99999999以外するとき。 (d)のBCDデータが0~99999999以外するとき。

DB+(P) [オペランドが3個の場合]

指定した2つのBCD8桁データを加算します。

ラダー	ST
	ENO:=DBPLUS(EN,s1,s2,d); ENO:=DBPLUSP(EN,s1,s2,d);

FBD/LD



(□には、DBPLUS、DBPLUSPが入ります。)

■実行条件

命令	実行条件
DB+	
DB+P	

設定データ

■内容、範囲、データ型

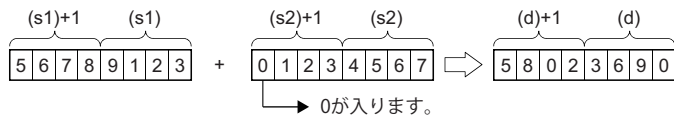
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	加算されるデータまたは、加算されるデータが格納されている先頭デバイス	0~99999999	BCD8桁	ANY32
(s2)	加算データまたは、加算データが格納されている先頭デバイス	0~99999999	BCD8桁	ANY32
(d)	演算結果を格納する先頭デバイス	—	BCD8桁	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—	—

機能

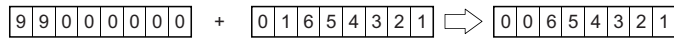
- (s1)で指定したBCD8桁データと、(s2)で指定したBCD8桁データの加算を行い、加算結果を(d)で指定したデバイスに格納します。



(d)+1, (s1)+1, (s2)+1: 上位4桁

(d), (s1), (s2): 下位4桁

- 加算結果が99999999を超えた場合は、桁上げは無視されます。この場合、SM700(キャリフラグ)はONしません。



エラー

エラーコード (SD0)	内容
3285H	(s1)のBCDデータが0~99999999以外するとき。
	(s2)のBCDデータが0~99999999以外するとき。

BCD8桁減算

DB-(P) [オペランドが2個の場合]

指定した2つのBCD8桁データを減算します。

ラダー	ST
	対応していません。 (296ページ DB-(P) [オペランドが3個の場合])

FBD/LD
対応していません。 (296ページ DB-(P) [オペランドが3個の場合])

■実行条件

命令	実行条件
DB-	
DB-P	

設定データ

■内容、範囲、データ型

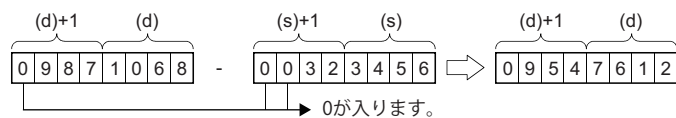
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	減算データまたは、減算データが格納されている先頭デバイス	0~99999999	BCD8桁	ANY32
(d)	減算されるデータまたは、減算されるデータが格納されている先頭デバイス	0~99999999	BCD8桁	ANY32

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 0	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 0	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

機能

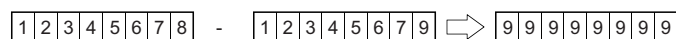
- ・ (d)で指定したBCD8桁データと、(s)で指定したBCD8桁データの減算を行い、減算結果を(d)で指定したデバイスに格納します。



(d)+1, (s)+1: 上位4桁

(d), (s): 下位4桁

- ・ 減算結果にアンダーフローが生じた場合は、下記ようになります。この場合、SM700(キャリフラグ)はONしません。

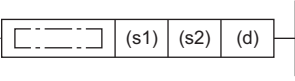


エラー

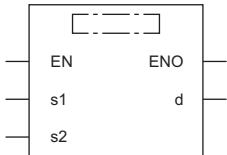
エラーコード (SD0)	内容
3285H	(s)のBCDデータが0~99999999以外するとき。
	(d)のBCDデータが0~99999999以外するとき。

DB-(P) [オペランドが3個の場合]

指定した2つのBCD8桁データを減算します。

ラダー	ST
	ENO:=DBMINUS(EN,s1,s2,d); ENO:=DBMINUSP(EN,s1,s2,d);

FBD/LD



(□には、DBMINUS、DBMINUSPが入ります。)

■実行条件

命令	実行条件
DB-	
DB-P	

設定データ

■内容、範囲、データ型

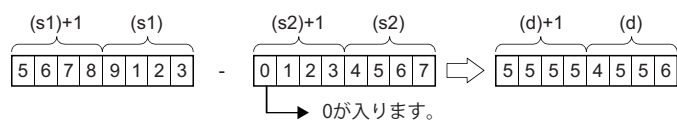
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	減算されるデータまたは、減算されるデータが格納されている先頭デバイス	0~99999999	BCD8桁	ANY32
(s2)	減算データまたは、減算データが格納されている先頭デバイス	0~99999999	BCD8桁	ANY32
(d)	演算結果を格納する先頭デバイス	—	BCD8桁	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—	—

機能

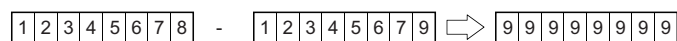
- (s1)で指定したBCD8桁データと、(s2)で指定したBCD8桁データの減算を行い、減算結果を(d)で指定したデバイスに格納します。



(d)+1, (s1)+1, (s2)+1: 上位4桁

(d), (s1), (s2): 下位4桁

- 減算結果にアンダーフローが生じた場合は、下記ようになります。この場合、SM700(キャリフラグ)はONしません。



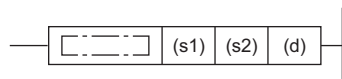
エラー

エラーコード (SD0)	内容
3285H	(s1)のBCDデータが0~99999999以外のとき。
	(s2)のBCDデータが0~99999999以外のとき。

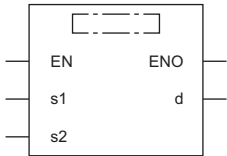
BCD4桁乗算

B*(P)

指定した2つのBCD4桁データを乗算します。

ラダー	ST
	対応していません。

FBD/LD



(□には、BMULTI, BMULTIPが入ります。)

■実行条件

命令	実行条件
B*	
B*P	

設定データ

■内容、範囲、データ型

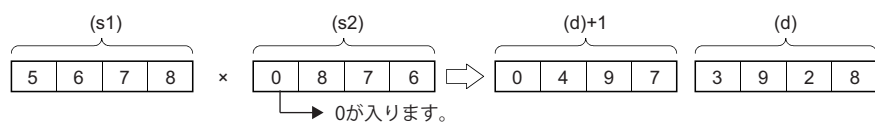
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	乗算されるデータまたは、乗算されるデータが格納されているデバイス	0~9999	BCD4桁	ANY16
(s2)	乗算データまたは、乗算データが格納されているデバイス	0~9999	BCD4桁	ANY16
(d)	演算結果を格納する先頭デバイス	—	BCD8桁	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

機能

- (s1)で指定したBCD4桁データと、(s2)で指定したBCD4桁データの乗算を行い、乗算結果を(d)で指定したデバイスに格納します。((d)+1が上位4桁、(d)が下位4桁)



エラー

エラーコード (SD0)	内容
3285H	(s1)のBCDデータが0~9999以外るとき。
	(s2)のBCDデータが0~9999以外るとき。

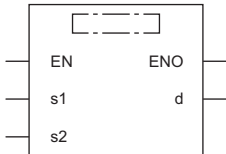
BCD4桁除算

B/(P)

指定した2つのBCD4桁データを除算します。

ラダー	ST
	対応していません。

FBD/LD



(□には、BDIVISION、BDIVISIONPが入ります。)

■実行条件

命令	実行条件
B/	
B/P	

設定データ

■内容、範囲、データ型

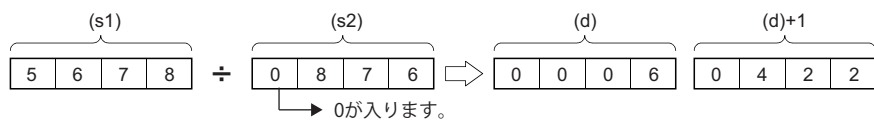
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	除算されるデータまたは、除算されるデータが格納されているデバイス	0~9999	BCD4桁	ANY16
(s2)	除算データまたは、除算データが格納されているデバイス	0~9999	BCD4桁	ANY16
(d)	演算結果を格納する先頭デバイス	—	BCD8桁	ANY16_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (s1)で指定したBCD4桁データと、(s2)で指定したBCD4桁データの除算を行い、除算結果を(d)で指定したデバイスに格納します。



(d): 商

(d)+1: 余り

- 除算結果は、32ビットを使用して商と余りを格納します。
- 商(BCD4桁)…下位16ビットに格納されます。
- 余り(BCD4桁)…上位16ビットに格納されます。

エラー

エラーコード (SD0)	内容
3280H	(s2)で指定した値(除数)が0のとき。
3285H	(s1)のBCDデータが0~9999以外のとき。
	(s2)のBCDデータが0~9999以外のとき。

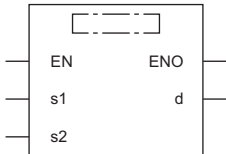
BCD8桁乗算

DB*(P)

指定した2つのBCD8桁データを乗算します。

ラダー	ST
	対応していません。

FBD/LD



(□には、DBMULTI、DBMULTIPが入ります。)

■実行条件

命令	実行条件
DB*	
DB*P	

設定データ

■内容、範囲、データ型

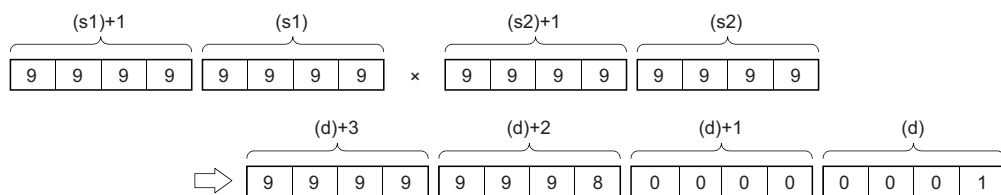
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	乗算されるデータまたは、乗算されるデータが格納されている先頭デバイス	0~99999999	BCD8桁	ANY32
(s2)	乗算データまたは、乗算データが格納されている先頭デバイス	0~99999999	BCD8桁	ANY32
(d)	演算結果を格納する先頭デバイス	—	BCD16桁	ANY32_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	—	○	—	—	○	—	○	○	—	—	—	—

機能

- (s1)で指定したBCD8桁データと、(s2)で指定したBCD8桁データの乗算を行い、乗算結果を(d)で指定したデバイスに格納します。



- (d)がビットデバイスの場合には、乗算結果は下位8桁(下位32ビット)までが対象となり、上位8桁(上位32ビット)を指定することはできません。

例

(d)がビットデバイスの場合の乗算結果

- K1...下位1桁(b0~b3)
- K4...下位4桁(b0~b15)
- K8...下位8桁(b0~b31)

エラー

エラーコード (SD0)	内容
3285H	(s1)のBCDデータが0~99999999以外るとき。
	(s2)のBCDデータが0~99999999以外るとき。

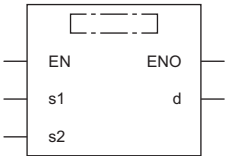
BCD8桁除算

DB/(P)

指定した2つのBCD8桁データを除算します。

ラダー	ST
	対応していません。

FBD/LD



(□には、DBDIVISION、DBDIVISIONPが入ります。)

■実行条件

命令	実行条件
DB/	
DB/P	

設定データ

■内容、範囲、データ型

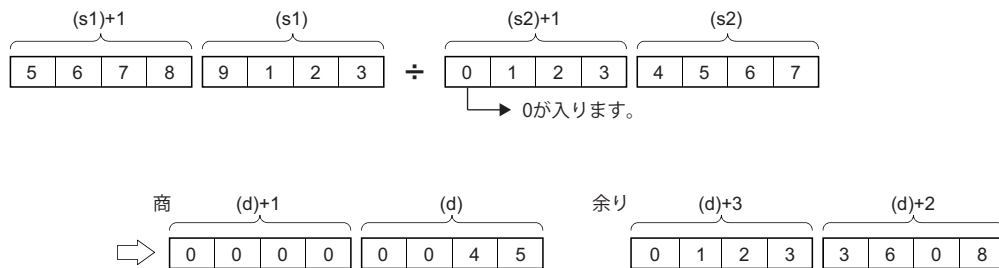
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	除算されるデータまたは、除算されるデータが格納されている先頭デバイス	0~99999999	BCD8桁	ANY32
(s2)	除算データまたは、除算データが格納されている先頭デバイス	0~99999999	BCD8桁	ANY32
(d)	演算結果を格納する先頭デバイス	—	BCD16桁	ANY32_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	—	○	—	—	○	—	○	—	—	—

機能

- (s1)で指定したBCD8桁データと、(s2)で指定したBCD8桁データの除算を行い、除算結果を(d)で指定したデバイスに格納します。



(d)+1, (d)+3: 上位4桁

(d), (d)+2: 下位4桁

- 除算結果は、64ビットを使用して商と余りを格納します。
- 商(BCD8桁)…下位32ビットに格納されます。
- 余り(BCD8桁)…上位32ビットに格納されます。
- (d)をビットデバイスで指定した場合、除算結果の余りは格納されません。

エラー

エラーコード (SD0)	内容
3280H	(s2)で指定した値(除数)が0のとき。
3285H	(s1)のBCDデータが0~99999999以外のとき。
	(s2)のBCDデータが0~99999999以外のとき。

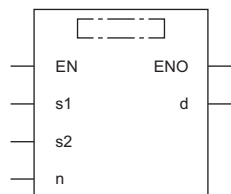
BIN16ビットブロックデータ加算

BK+(P)(_U)

指定した2つのBIN16ビットデータブロックを加算します。

ラダー	ST	
	ENO:=BKPLUS(EN,s1,s2,n,d); ENO:=BKPLUSP(EN,s1,s2,n,d);	ENO:=BKPLUS_U(EN,s1,s2,n,d); ENO:=BKPLUSP_U(EN,s1,s2,n,d);

FBD/LD



(□には、BKPLUS、BKPLUSP、BKPLUS_U、BKPLUSP_Uが入ります。)

■実行条件

命令	実行条件
BK+ BK+_U	
BK+P BK+P_U	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	BK+(P) BK+(P)_U	-32768~32767 0~65535	符号付きBIN16ビット 符号なしBIN16ビット	ANY16_S* ¹ ANY16_U* ¹
(s2)	BK+(P) BK+(P)_U	-32768~32767 0~65535	符号付きBIN16ビット 符号なしBIN16ビット	ANY16_S* ¹ ANY16_U* ¹
(d)	BK+(P) BK+(P)_U	—	符号付きBIN16ビット 符号なしBIN16ビット	ANY16_S* ¹ ANY16_U* ¹
(n)	加算データ個数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

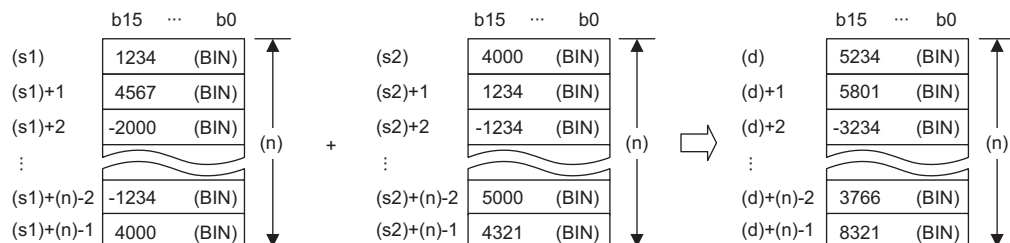
オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

機能

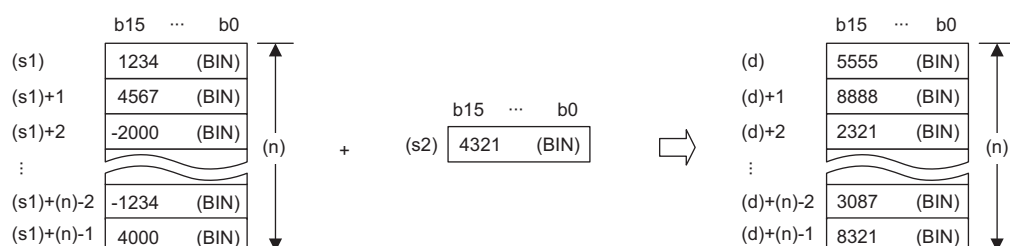
- (s1)で指定したデバイスから(n)点分のBIN16ビットデータと、(s2)で指定したデバイスから(n)点分のBIN16ビットデータ、または定数の加算を行い、加算結果を(d)で指定したデバイス以降に格納します。
- ブロック加算は16ビット単位で行います。

例

(s2)にデバイスを指定した場合(符号付き指定時)



(s2)に定数を指定した場合(符号付き指定時)



- 演算結果にオーバーフローが生じた場合は、下記のようになります。この場合、SM700(キャリフラグ)はONしません。

符号付きを指定した場合	符号なしを指定した場合
K32767 (H7FFF) + K2 (H0002) → K-32767 (H8001) K-32767 (H8001) + K-2 (HFFFE) → K32767 (H7FFF)	K65535 (HFFFF) + K1 (H0001) → K0 (H0000)

エラー

エラーコード (SD0)	内容
2821H	(s1)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲が重複しているとき。 ((s1)と(d)に同一デバイスを指定した場合を除く。)
	(s2)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲が重複しているとき。 ((s2)と(d)に同一デバイスを指定した場合を除く。)

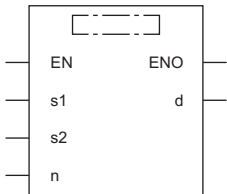
BIN16ビットブロックデータ減算

BK-(P)(_U)

指定した2つのBIN16ビットデータブロックを減算します。

ラダー	ST	
	ENO:=BKMINUS(EN,s1,s2,n,d); ENO:=BKMINUSP(EN,s1,s2,n,d);	ENO:=BKMINUS_U(EN,s1,s2,n,d); ENO:=BKMINUSP_U(EN,s1,s2,n,d);

FBD/LD



(□には、BKMINUS、BKMINUSP、BKMINUS_U、BKMINUSP_Uが入ります。)

■実行条件

命令	実行条件
BK- BK-_U	
BK-P BK-P_U	

設定データ

■内容、範囲、データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(s1)	BK-(P)	減算されるデータまたは、減算されるデータが格納されている先頭デバイス	-32768~32767	符号付きBIN16ビット	ANY16_S* ¹
	BK-(P)_U		0~65535	符号なしBIN16ビット	ANY16_U* ¹
(s2)	BK-(P)	減算データまたは、減算データが格納されている先頭デバイス	-32768~32767	符号付きBIN16ビット	ANY16_S* ¹
	BK-(P)_U		0~65535	符号なしBIN16ビット	ANY16_U* ¹
(d)	BK-(P)	演算結果を格納する先頭デバイス	—	符号付きBIN16ビット	ANY16_S* ¹
	BK-(P)_U			符号なしBIN16ビット	ANY16_U* ¹
(n)		減算データ個数	0~65535	符号なしBIN16ビット	ANY16
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

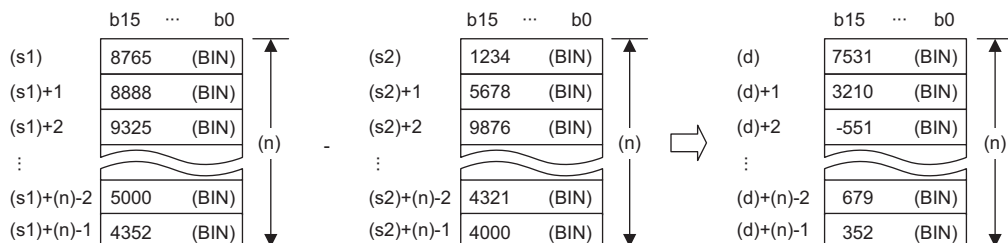
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

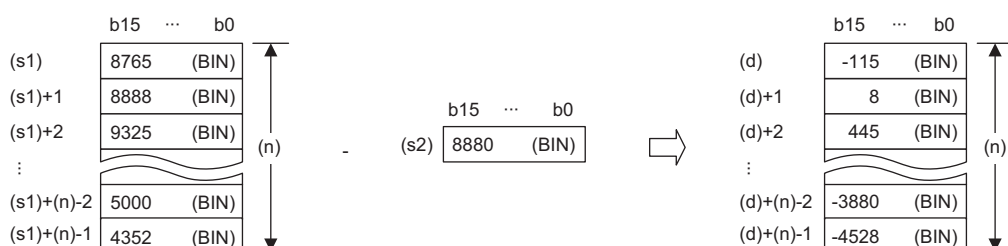
- (s1)で指定したデバイスから(n)点分のBIN16ビットデータと、(s2)で指定したデバイスから(n)点分のBIN16ビットデータ、または定数の減算を行い、減算結果を(d)で指定したデバイス以降に格納します。
- ブロック減算は16ビット単位で行います。

例

(s2)にデバイスを指定した場合



(s2)に定数を指定した場合



- 演算結果にオーバーフローが生じた場合は、下記のようになります。この場合、SM700(キャリフラグ)はONしません。

符号付きを指定した場合			符号なしを指定した場合		
K-32768 (H8000)	-	K2 (H0002)	⇒	K32766 (H7FFE)	
K32767 (H7FFF)	-	K-2 (HFFFE)	⇒	K-32767 (H8001)	
				K0 (H0000)	-
				K1 (H0001)	⇒
				K65535 (HFFFF)	

エラー

エラーコード (SD0)	内容
2821H	(s1)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲が重複しているとき。 ((s1)と(d)に同一デバイスを指定した場合を除く。)
	(s2)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲が重複しているとき。 ((s2)と(d)に同一デバイスを指定した場合を除く。)

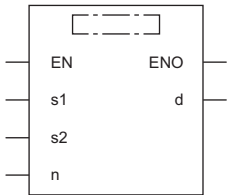
BIN32ビットブロックデータ加算

DBK+(P)(_U)

指定した2つのBIN32ビットデータブロックを加算します。

ラダー	ST	
	ENO:=DBKPLUS(EN,s1,s2,n,d); ENO:=DBKPLUSP(EN,s1,s2,n,d);	ENO:=DBKPLUS_U(EN,s1,s2,n,d); ENO:=DBKPLUSP_U(EN,s1,s2,n,d);

FBD/LD



(□には、DBKPLUS、DBKPLUSP、DBKPLUS_U、DBKPLUSP_Uが入ります。)

■実行条件

命令	実行条件
DBK+ DBK+_U	
DBK+P DBK+P_U	

設定データ

■内容、範囲、データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(s1)	DBK+(P)	加算されるデータまたは、加算されるデータが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S*1
	DBK+(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U*1
(s2)	DBK+(P)	加算データまたは、加算データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S*1
	DBK+(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U*1
(d)	DBK+(P)	演算結果を格納する先頭デバイス	—	符号付きBIN32ビット	ANY32_S*1
	DBK+(P)_U			符号なしBIN32ビット	ANY32_U*1
(n)		加算データ個数	0~65535	符号なしBIN16ビット	ANY16
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

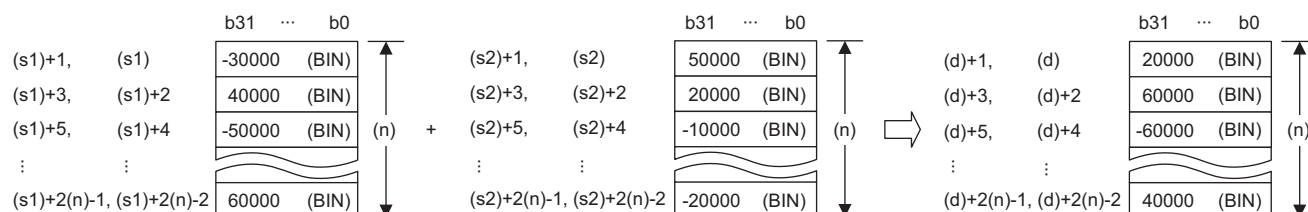
オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

機能

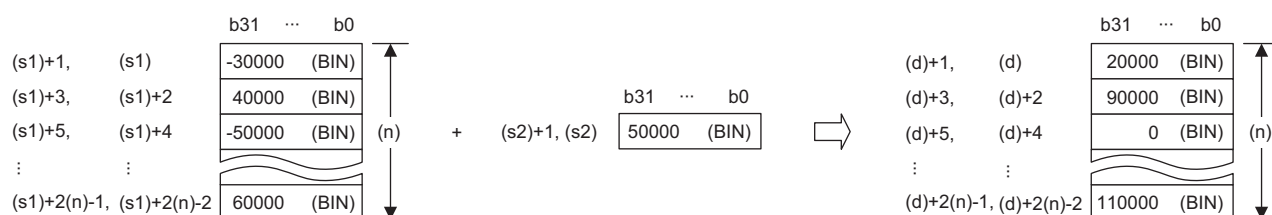
- (s1)で指定したデバイスから(n)点分のBIN32ビットデータと、(s2)で指定したデバイスから(n)点分のBIN32ビットデータ、または定数の加算を行い、加算結果を(d)で指定したデバイス以降に格納します。
- ブロック加算は32ビット単位で行います。

例

(s2)にデバイスを指定した場合(符号付き指定時)



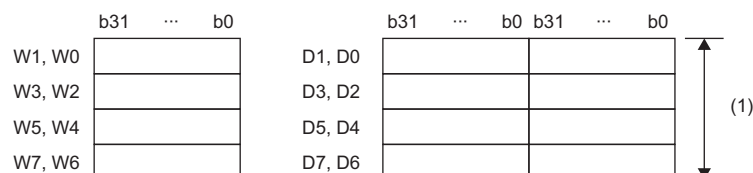
(s2)に定数を指定した場合(符号付き指定時)



- (s1)または(s2)が(d)と同一デバイス(完全一致)で指定された場合は、演算可能です。ただし、(s1)または(s2)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲を部分的に一致(重複)させた場合は、エラーとなります。

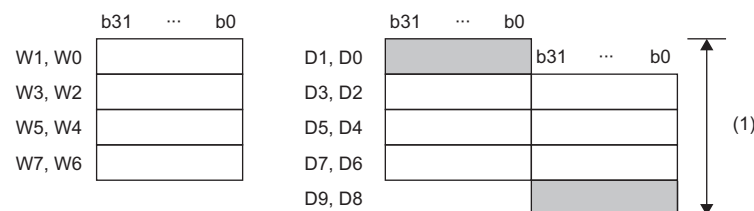
例

(s2)と(d)から4点分のデバイスが完全一致の場合



(1) 完全一致しているので演算可能です。

(s2), (d)から4点分のデバイスが部分的に一致の場合



(1) 部分的に一致しているので、演算エラーとなります。

- (n)で指定した値が0の場合は無処理となります。
- 演算結果にオーバーフローが生じた場合は、下記のようになります。この場合、SM700(キャリフラグ)はONしません。

符号付きを指定した場合	符号なしを指定した場合
K2147483647 (H7FFFFFFF) + K2 (H00000002) → K-2147483647 (H80000001) K-2147483647 (H80000001) + K-2 (HFFFFFFFE) → K2147483647 (H7FFFFFFF)	K4294967295 (HFFFFFFF) + K1 (H00000001) → K0 (H00000000)

エラー

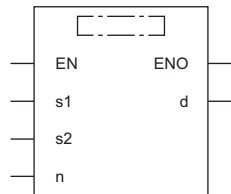
エラーコード (SD0)	内容
2821H	(s1)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲が重複しているとき。 ((s1)と(d)に同一デバイスを指定した場合を除く。)
	(s2)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲が重複しているとき。 ((s2)と(d)に同一デバイスを指定した場合を除く。)

BIN32ビットブロックデータ減算

DBK-(P)(_U)

指定した2つのBIN32ビットデータブロックを減算します。

ラダー	ST
	ENO:=DBKMINUS(EN,s1,s2,n,d); ENO:=DBKMINUSP(EN,s1,s2,n,d); ENO:=DBKMINUS_U(EN,s1,s2,n,d); ENO:=DBKMINUSP_U(EN,s1,s2,n,d);



(□には、DBKMINUS、DBKMINUSP、DBKMINUS_U、DBKMINUSP_Uが入ります。)

■実行条件

命令	実行条件
DBK- DBK-_U	
DBK-P DBK-P_U	

設定データ

■内容、範囲、データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(s1)	DBK-(P)	減算されるデータまたは、減算されるデータが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S*1
	DBK-(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U*1
(s2)	DBK-(P)	減算データまたは、減算データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S*1
	DBK-(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U*1
(d)	DBK-(P)	演算結果を格納する先頭デバイス	—	符号付きBIN32ビット	ANY32_S*1
	DBK-(P)_U			符号なしBIN32ビット	ANY32_U*1
(n)		減算データ個数	0~65535	符号なしBIN16ビット	ANY16
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

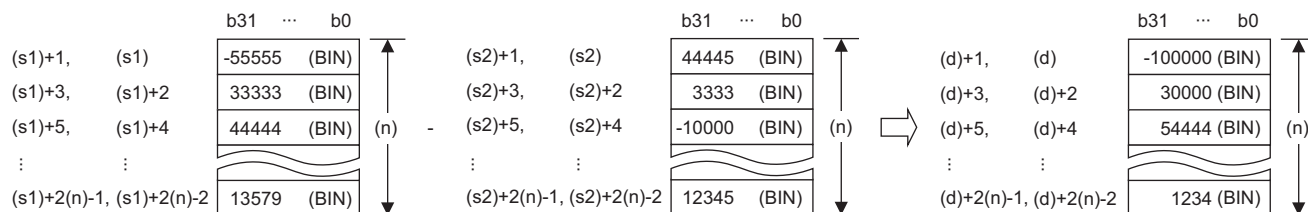
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

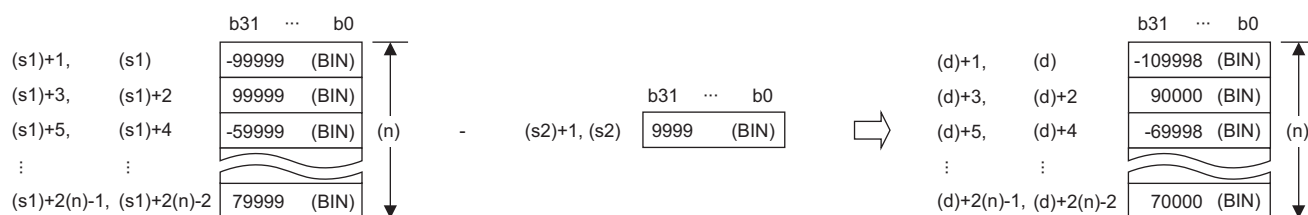
- (s1)で指定したデバイスから(n)点分のBIN32ビットデータと、(s2)で指定したデバイスから(n)点分のBIN32ビットデータ、または定数の減算を行い、減算結果を(d)で指定したデバイス以降に格納します。
- ブロック減算は32ビット単位で行います。

例

(s2)にデバイスを指定した場合(符号付き指定時)



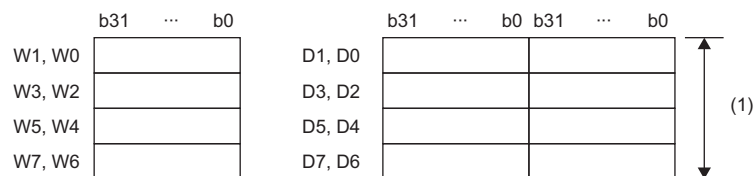
(s2)に定数を指定した場合(符号付き指定時)



- (s1)または(s2)が(d)と同一デバイス(完全一致)で指定された場合は、演算可能です。ただし、(s1)または(s2)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲を部分的に一致(重複)させた場合は、エラーとなります。

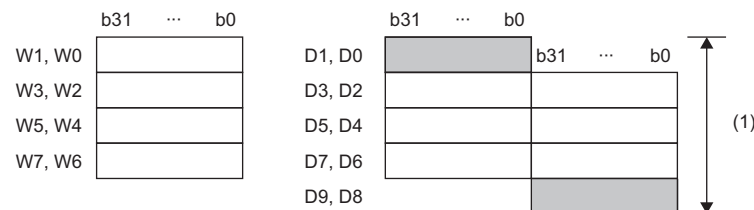
例

(s2)と(d)から4点分のデバイスが完全一致の場合



(1) 完全一致しているので演算可能です。

(s2), (d)から4点分のデバイスが部分的に一致の場合



(1) 部分的に一致しているので、演算エラーとなります。

- (n)で指定した値が0の場合は無処理となります。
- 演算結果にオーバーフローが生じた場合は、下記ようになります。この場合、SM700(キャリフラグ)はONしません。

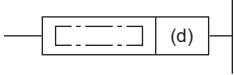
符号付きを指定した場合		符号なしを指定した場合	
K2147483647 (H7FFFFFFF)	- K-2 (HFFFFFFFE)	K-2147483647 (H80000001)	K0 (H00000000)
K-2147483647 (H80000001)	- K2 (H00000002)	K2147483647 (H7FFFFFFF)	K1 (H00000001)
			K4294967295 (HFFFFFFF)

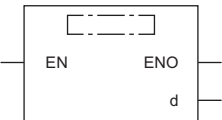
エラーコード (SD0)	内容
2821H	(s1)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲が重複しているとき。 ((s1)と(d)に同一デバイスを指定した場合を除く。)
	(s2)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲が重複しているとき。 ((s2)と(d)に同一デバイスを指定した場合を除く。)

16ビットBINデータインクリメント

INC(P)(_U)

指定したBIN16ビットデータを+1します。

ラダー	ST	
	ENO:=INC(EN,d); ENO:=INCP(EN,d);	ENO:=INC_U(EN,d); ENO:=INCP_U(EN,d);

FBD/LD


■実行条件

命令	実行条件
INC INC_U	
INCP INCP_U	

設定データ

■内容、範囲、データ型

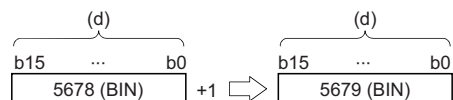
オペランド		内容	範囲	データ型	データ型(ラベル)
(d)	INC(P)	+1を行うデバイス	-32768~32767	符号付きBIN16ビット	ANY16_S
	INC(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□		LT, LST, LC	LZ		K, H	E	\$	
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- (d)で指定されたデバイス(BIN16ビットデータ)を+1します。



- (d)で指定されたデバイスの内容が32767のときにINC(P)命令を実行した場合は、-32768が(d)で指定されたデバイスに格納されます。(符号付きを指定した場合)
- (d)で指定されたデバイスの内容が65535のときにINC(P)_U命令を実行した場合は、0が(d)で指定されたデバイスに格納されます。(符号なしを指定した場合)

エラー

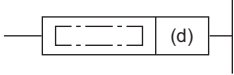
演算エラーはありません。

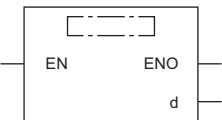
16ビットBINデータデクリメント

12

DEC(P)(_U)

指定したBIN16ビットデータを-1します。

ラダー	ST
	ENO:=DEC(EN,d); ENO:=DECP(EN,d);
	ENO:=DEC_U(EN,d); ENO:=DECP_U(EN,d);

FBD/LD


■実行条件

命令	実行条件
DEC DEC_U	
DECP DECP_U	

設定データ

■内容、範囲、データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(d)	DEC(P)	-1を行うデバイス	-32768~32767	符号付きBIN16ビット	ANY16_S
	DEC(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(d)	○	○	○	○	○	—	—	○	—	—	—

機能

- (d)で指定されたデバイス(BIN16ビットデータ)を-1します。



- (d)で指定されたデバイスの内容が-32768のときにDEC(P)命令を実行した場合は、32767が(d)で指定されたデバイスに格納されます。(符号付きを指定した場合)
- (d)で指定されたデバイスの内容が0のときにDEC(P)_U命令を実行した場合は、65535が(d)で指定されたデバイスに格納されます。(符号なしを指定した場合)

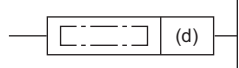
エラー

演算エラーはありません。

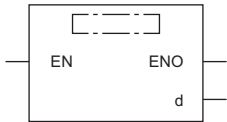
32ビットBINデータインクリメント

DINC(P)(_U)

指定したBIN32ビットデータを+1します。

ラダー	ST	
	ENO:=DINC(EN,d); ENO:=DINCP(EN,d);	ENO:=DINC_U(EN,d); ENO:=DINCP_U(EN,d);

FBD/LD



■実行条件

命令	実行条件
DINC DINC_U	
DINCP DINCP_U	

設定データ

■内容、範囲、データ型

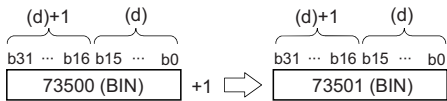
オペランド	内容	範囲	データ型	データ型(ラベル)
(d) DINC(P) DINC(P)_U	+1を行う先頭デバイス	-2147483648~2147483647 0~4294967295	符号付きBIN32ビット 符号なしBIN32ビット	ANY32_S ANY32_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口		LT, LST, LC	LZ		K, H	E	\$	
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (d)で指定されたデバイスの内容(BIN32ビットデータ)を+1します。



- (d)で指定されたデバイスの内容が2147483647のときに、DINC(P)(_U)命令を実行した場合は、-2147483648が(d)で指定されたデバイスに格納されます。(符号付きを指定した場合)
- (d)で指定されたデバイスの内容が4294967295のときに、DINC(P)(_U)命令を実行した場合は、0が(d)で指定されたデバイスに格納されます。(符号なしを指定した場合)

エラー

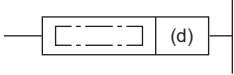
演算エラーはありません。

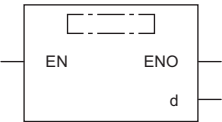
32ビットBINデータデクリメント

12

DDEC(P)(_U)

指定したBIN32ビットデータを-1します。

ラダー	ST
	ENO:=DDEC(EN,d); ENO:=DDECP(EN,d);
	ENO:=DDEC_U(EN,d); ENO:=DDECP_U(EN,d);

FBD/LD


■実行条件

命令	実行条件
DDEC DDEC_U	
DDECP DDECP_U	

設定データ

■内容, 範囲, データ型

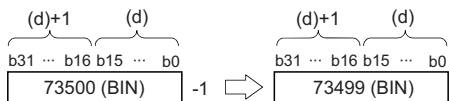
オペランド		内容	範囲	データ型	データ型(ラベル)
(d)	DDEC(P)	-1を行う先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	DDEC(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (d)で指定されたデバイスの内容(BIN32ビットデータ)を-1します。



- (d)で指定されたデバイスの内容が-2147483648のときに、DDEC(P)(U)命令を実行した場合は、2147483647が(d)で指定されたデバイスに格納されます。(符号付きを指定した場合)
- (d)で指定されたデバイスの内容が0のときDDEC(P)(U)命令を実行すると、-1が(d)で指定されたデバイスに格納されます。(符号付きを指定した場合)
- (d)で指定されたデバイスの内容が0のときに、DDEC(P)(U)命令を実行した場合は、4294967295が(d)で指定されたデバイスに格納されます。(符号なしを指定した場合)

エラー

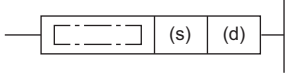
演算エラーはありません。

12.3 論理演算命令

16ビットデータ論理積

WAND(P) [オペランドが2個の場合]

指定した2つのBIN16ビットデータの論理積を行います。

ラダー	ST
	対応していません。 (322ページ WAND(P) [オペランドが3個の場合])

FBD/LD
対応していません。 (322ページ WAND(P) [オペランドが3個の場合])

■実行条件

命令	実行条件
WAND	
WANDP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	論理積を行うデータまたは、データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(d)	論理積の結果を格納するデバイス	-32768~32767	符号付きBIN16ビット	ANY16

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- ・(d)で指定されたデバイスのBIN16ビットデータと、(s)で指定されたデバイスのBIN16ビットデータの、ビットごとの論理積を求め、結果を(d)で指定されたデバイスに格納します。

	b15	...	b8	b7	...	b0										
(d)	1	1	1	1	1	0	0	0	0	1	1	1	1			
	AND															
(s)	b15	...	b8	b7	...	b0										
	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
	↓															
(d)	b15	...	b8	b7	...	b0										
	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0

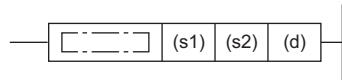
- ・ビットデバイスの場合、桁指定による点数以降のビットデバイスは0として演算します。

エラー

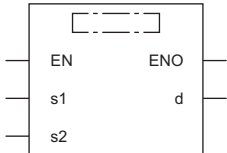
演算エラーはありません。

WAND(P) [オペランドが3個の場合]

指定した2つのBIN16ビットデータの論理積を行います。

ラダー	ST
	ENO:=WAND(EN,s1,s2,d); ENO:=WANDP(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
WAND	
WANDP	

設定データ

■内容、範囲、データ型

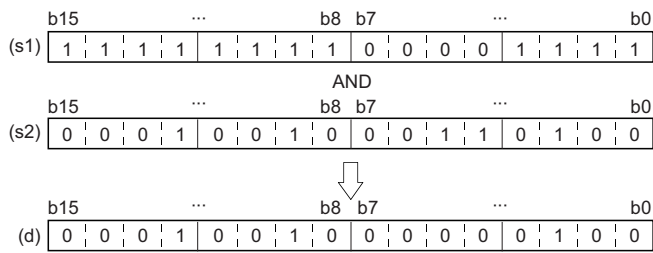
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	論理積を行うデータまたは、データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(s2)	論理積を行うデータまたは、データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(d)	論理積の結果を格納するデバイス	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

機能

- (s1)で指定されたデバイスのBIN16ビットデータと、(s2)で指定されたデバイスのBIN16ビットデータの、ビットごとの論理積を求め、結果を(d)で指定されたデバイスに格納します。



- ビットデバイスの場合、桁指定による点数以降のビットデバイスは0として演算します。

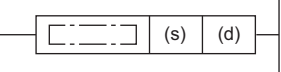
エラー

演算エラーはありません。

32ビットデータ論理積

DAND(P) [オペランドが2個の場合]

指定した2つのBIN32ビットデータの論理積を行います。

ラダー	ST
	対応していません。 (参照 326ページ DAND(P) [オペランドが3個の場合])

FBD/LD
対応していません。 (参照 326ページ DAND(P) [オペランドが3個の場合])

■実行条件

命令	実行条件
DAND	
DANDP	

設定データ

■内容，範囲，データ型

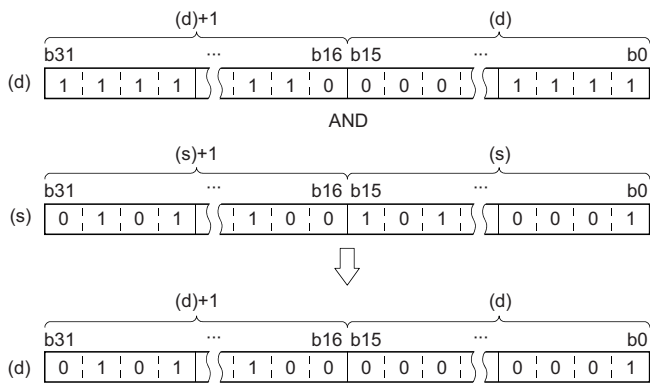
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	論理積を行うデータまたは，データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(d)	論理積の結果を格納する先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (d)で指定されたデバイスのBIN32ビットデータと、(s)で指定されたデバイスのBIN32ビットデータの、ビットごとの論理積を求め、結果を(d)で指定されたデバイスに格納します。



- ビットデバイスの場合、桁指定による点数以降のビットデバイスは0として演算します。

エラー

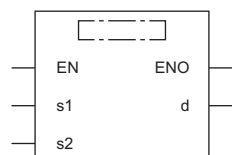
演算エラーはありません。

DAND(P) [オペランドが3個の場合]

指定した2つのBIN32ビットデータの論理積を行います。

ラダー	ST
	<pre>ENO:=DAND(EN,s1,s2,d); ENO:=DANDP(EN,s1,s2,d);</pre>

FBD/LD



■実行条件

命令	実行条件
DAND	
DANDP	

設定データ

■内容、範囲、データ型

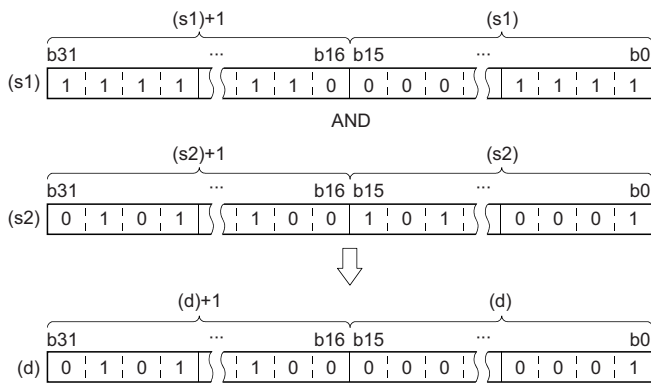
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	論理積を行うデータまたは、データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(s2)	論理積を行うデータまたは、データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(d)	論理積の結果を格納する先頭デバイス	—	符号付きBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (s1)で指定されたデバイスのBIN32ビットデータと、(s2)で指定されたデバイスのBIN32ビットデータの、ビットごとの論理積を求め、結果を(d)で指定されたデバイスに格納します。



- ビットデバイスの場合、桁指定による点数以降のビットデバイスは0として演算します。

エラー

演算エラーはありません。

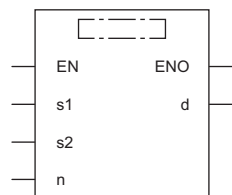
16ビットブロックデータ論理積

BKAND(P)

指定した2つのBIN16ビットデータブロックの論理積を行います。

ラダー	ST
	ENO:=BKAND(EN,s1,s2,n,d); ENO:=BKANDP(EN,s1,s2,n,d);

FBD/LD



■実行条件

命令	実行条件
BKAND	
BKANDP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	論理積を行うデータまたは、論理積を行うデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(s2)	論理積を行うデータまたは、データが格納されている先頭デバイス	-32768~32767	符号付きBIN16ビット	ANY16 ^{*1}
(d)	論理積の結果を格納する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(n)	演算データ個数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

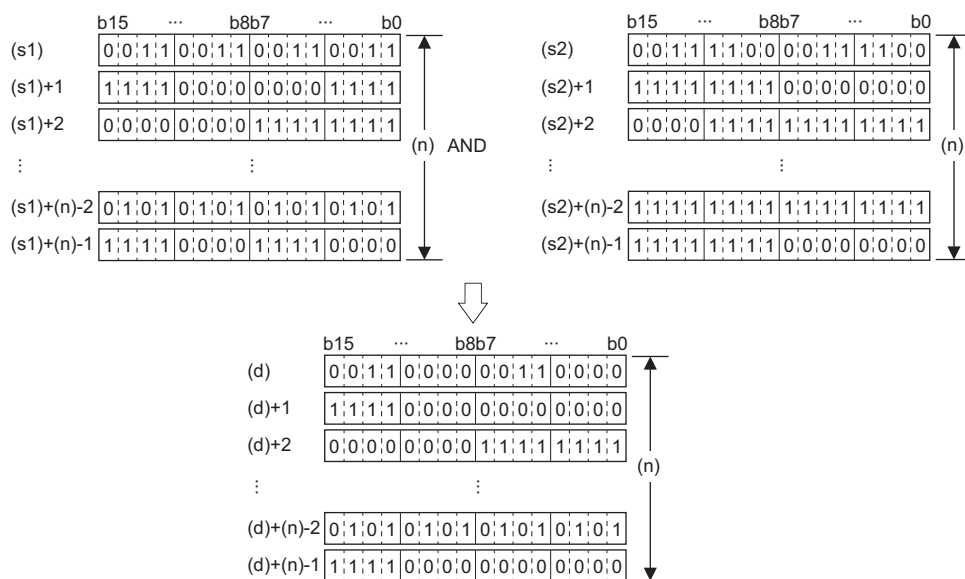
■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1) ^{*1}	—	—	○	—	—	—	○	—	—	—	—
(s2) ^{*1}	—	—	○	—	—	—	○	○	—	—	—
(d) ^{*1}	—	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

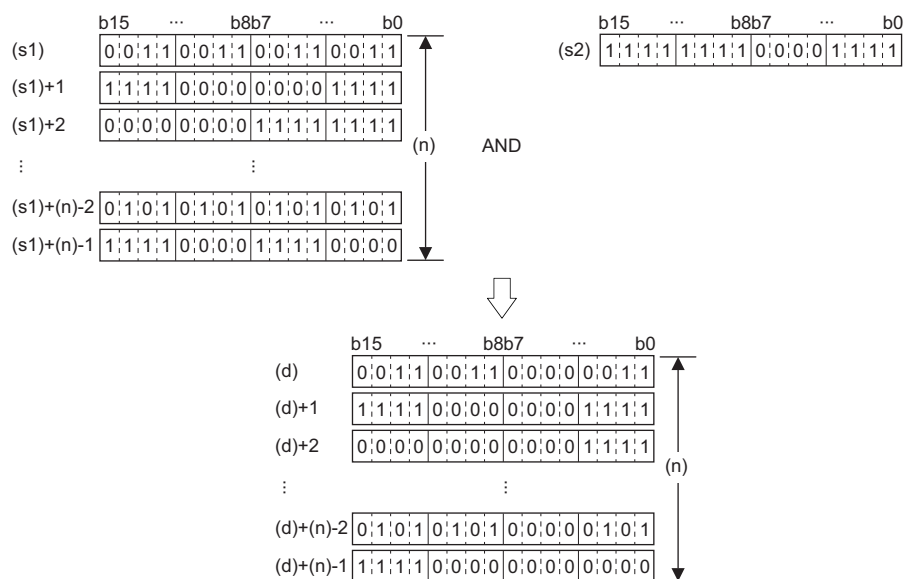
*1 (s1)と(d)または(s2)と(d)に同一デバイス番号を指定できます。

機能

- (s1)で指定されたデバイスから(n)点分の内容と、(s2)で指定されたデバイスから(n)点分の内容の論理積を求め、結果を(d)で指定されたデバイス以降に格納します。



- (s2)には-32768~32767(符号付きBIN16ビット)の定数が指定できます。



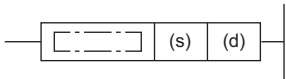
エラー

エラーコード (SD0)	内容
2821H	(s1)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲の一部分が重複しているとき。 ((s1)と(d)に同一デバイスを指定した場合を除く。)
	(s2)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲の一部分が重複しているとき。 ((s2)と(d)に同一デバイスを指定した場合を除く。)

16ビットデータ論理和

WOR(P) [オペランドが2個の場合]

指定した2つのBIN16ビットデータの論理和を行います。

ラダー	ST
	対応していません。 ( 331ページ WOR(P) [オペランドが3個の場合])

FBD/LD
対応していません。 ( 331ページ WOR(P) [オペランドが3個の場合])

■実行条件

命令	実行条件
WOR	
WOP	

設定データ

■内容、範囲、データ型

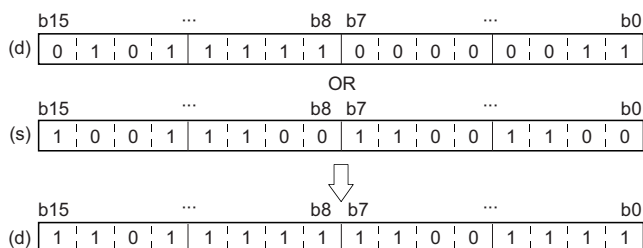
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	論理和を行うデータまたは、データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(d)	論理和の結果を格納するデバイス	-32768~32767	符号付きBIN16ビット	ANY16

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—

機能

- ・ (d)で指定されたデバイスのBIN16ビットデータと、(s)で指定されたデバイスのBIN16ビットデータの、ビットごとの論理和を求め、結果を(d)で指定されたデバイスに格納します。



- ・ ビットデバイスの場合、桁指定による点数以降のビットデバイスは0として演算します。

エラー

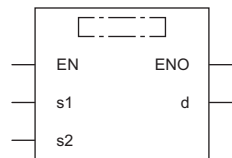
演算エラーはありません。

WOR(P) [オペランドが3個の場合]

指定した2つのBIN16ビットデータの論理和を行います。

ラダー	ST
	<pre>ENO:=WOR(EN,s1,s2,d); ENO:=WORP(EN,s1,s2,d);</pre>

FBD/LD



■実行条件

命令	実行条件
WOR	
WORP	

設定データ

■内容、範囲、データ型

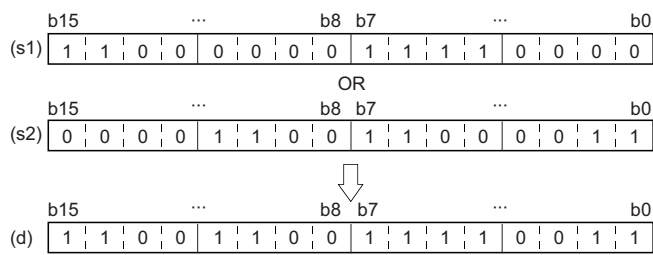
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	論理和を行うデータまたは、データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(s2)	論理和を行うデータまたは、データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(d)	論理和の結果を格納するデバイス	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

機能

- (s1)で指定されたデバイスのBIN16ビットデータと、(s2)で指定されたデバイスのBIN16ビットデータの、ビットごとの論理和を求め、結果を(d)で指定されたデバイスに格納します。



- ビットデバイスの場合、桁指定による点数以降のビットデバイスは0として演算します。

エラー

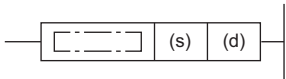
演算エラーはありません。

32ビットデータ論理和

12

DOR(P) [オペランドが2個の場合]

指定した2つのBIN32ビットデータの論理和を行います。

ラダー	ST
	対応していません。 (☞ 335ページ DOR(P) [オペランドが3個の場合])

FBD/LD
対応していません。 (☞ 335ページ DOR(P) [オペランドが3個の場合])

■実行条件

命令	実行条件
DOR	
DORP	

設定データ

■内容, 範囲, データ型

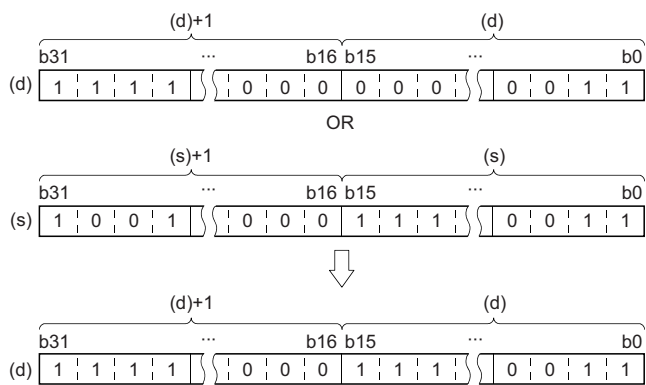
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	論理和を行うデータまたは、データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(d)	論理和の結果を格納する先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (d)で指定されたデバイスのBIN32ビットデータと、(s)で指定されたデバイスのBIN32ビットデータの、ビットごとの論理和を求め、結果を(d)で指定されたデバイスに格納します。



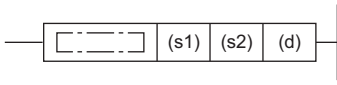
- ビットデバイスの場合、桁指定による点数以降のビットデバイスは0として演算します。

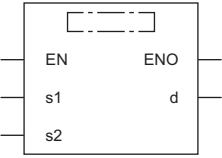
エラー

演算エラーはありません。

DOR(P) [オペランドが3個の場合]

指定した2つのBIN32ビットデータの論理和を行います。

ラダー	ST
	<pre>ENO:=DOR(EN,s1,s2,d); ENO:=DORP(EN,s1,s2,d);</pre>

FBD/LD


■実行条件

命令	実行条件
DOR	
DORP	

設定データ

■内容、範囲、データ型

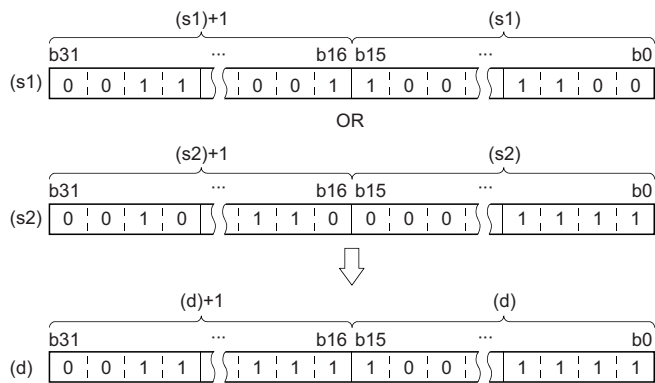
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	論理和を行うデータまたは、データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(s2)	論理和を行うデータまたは、データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(d)	論理和の結果を格納する先頭デバイス	—	符号付きBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□	G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (s1)で指定されたデバイスのBIN32ビットデータと、(s2)で指定されたデバイスのBIN32ビットデータの、ビットごとの論理和を求め、結果を(d)で指定されたデバイスに格納します。



- ビットデバイスの場合、桁指定による点数以降のビットデバイスは0として演算します。

エラー

演算エラーはありません。

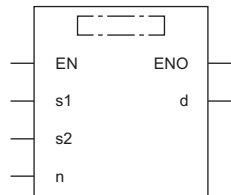
16ビットブロックデータ論理和

BKOR(P)

指定した2つのBIN16ビットデータブロックの論理和を行います。

ラダー	ST
	ENO:=BKOR(EN,s1,s2,n,d); ENO:=BKORP(EN,s1,s2,n,d);

FBD/LD



■実行条件

命令	実行条件
BKOR	
BKORP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	論理和を行うデータまたは、論理和を行うデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(s2)	論理和を行うデータまたは、データが格納されている先頭デバイス	-32768~32767	符号付きBIN16ビット	ANY16 ^{*1}
(d)	論理和の結果を格納する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(n)	演算データ個数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

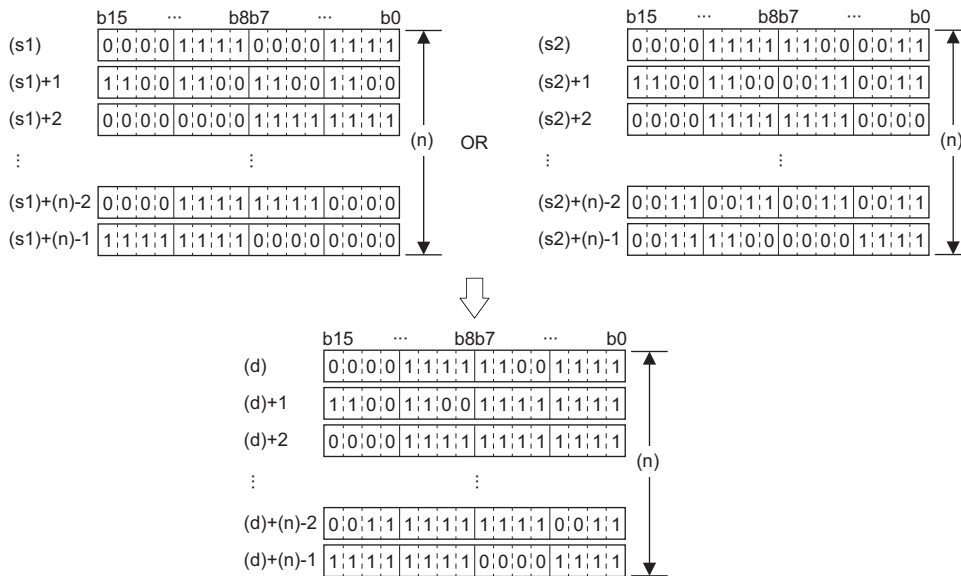
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1) ^{*1}	—	—	○	—	—	—	—	○	—	—	—	—
(s2) ^{*1}	—	—	○	—	—	—	—	○	○	—	—	—
(d) ^{*1}	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

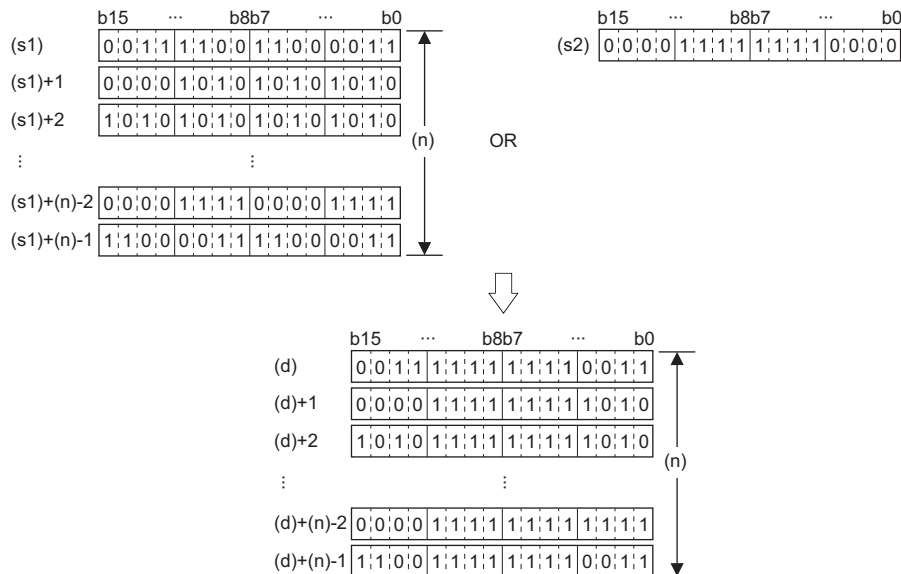
*1 (s1)と(d)または(s2)と(d)に同一デバイス番号を指定できます。

機能

- (s1)で指定されたデバイスから(n)点分の内容と、(s2)で指定されたデバイスから(n)点分の内容の論理和を求め、結果を(d)で指定されたデバイス以降に格納します。



- (s2)には-32768~32767(符号付きBIN16ビット)の定数が指定できます。



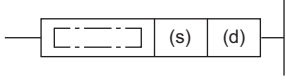
エラー

エラーコード (SD0)	内容
2821H	(s1)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲の一部分が重複しているとき。 ((s1)と(d)に同一デバイスを指定した場合を除く。)
	(s2)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲の一部分が重複しているとき。 ((s2)と(d)に同一デバイスを指定した場合を除く。)

16ビットデータ排他的論理和

WXOR(P) [オペランドが2個の場合]

指定した2つのBIN16ビットデータの排他的論理和を行います。

ラダー	ST
	対応していません。 (☞ 340ページ WXOR(P) [オペランドが3個の場合])
FBD/LD	
対応していません。 (☞ 340ページ WXOR(P) [オペランドが3個の場合])	

■実行条件

命令	実行条件
WXOR	
WXORP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	排他的論理和を行うデータまたは、データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(d)	排他的論理和の結果を格納するデバイス	-32768~32767	符号付きBIN16ビット	ANY16

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—

機能

- ・(d)で指定されたデバイスのBIN16ビットデータと、(s)で指定されたデバイスのBIN16ビットデータの、ビットごとの排他的論理和を求め、結果を(d)で指定されたデバイスに格納します。

	b15	...				b8	b7	...				b0				
(d)	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
	XOR															
(s)	b15	...				b8	b7	...				b0				
	0	0	0	1	1	0	1	1	1	1	1	1	0	0	0	0
																
(d)	b15	...				b8	b7	...				b0				
	1	0	1	1	0	0	0	1	0	1	0	1	1	0	1	0

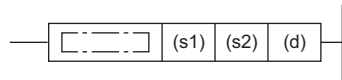
- ・ビットデバイスの場合、桁指定による点数以降のビットデバイスは0として演算します。

エラー

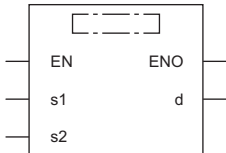
演算エラーはありません。

WXOR(P) [オペランドが3個の場合]

指定した2つのBIN16ビットデータの排他的論理和を行います。

ラダー	ST
	ENO:=WXOR(EN,s1,s2,d); ENO:=WXORP(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
WXOR	
WXORP	

設定データ

■内容、範囲、データ型

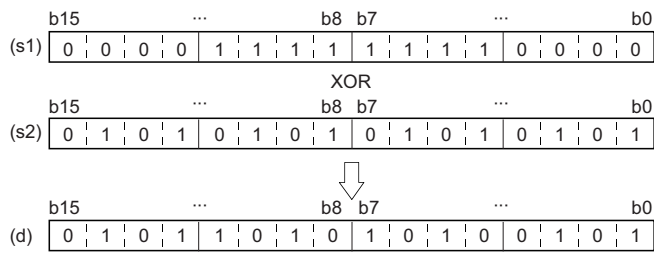
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	排他的論理和を行うデータまたは、データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(s2)	排他的論理和を行うデータまたは、データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(d)	排他的論理和の結果を格納するデバイス	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

機能

- (s1)で指定されたデバイスのBIN16ビットデータと、(s2)で指定されたデバイスのBIN16ビットデータの、ビットごとの排他的論理和を求め、結果を(d)で指定されたデバイスに格納します。



- ビットデバイスの場合、桁指定による点数以降のビットデバイスは0として演算します。

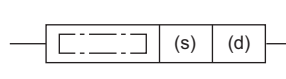
エラー

演算エラーはありません。

32ビットデータ排他的論理和

DXOR(P) [オペランドが2個の場合]

指定した2つのBIN32ビットデータの排他的論理和を行います。

ラダー	ST
	対応していません。 (344ページ DXOR(P) [オペランドが3個の場合])

FBD/LD
対応していません。 (344ページ DXOR(P) [オペランドが3個の場合])

■実行条件

命令	実行条件
DXOR	
DXORP	

設定データ

■内容, 範囲, データ型

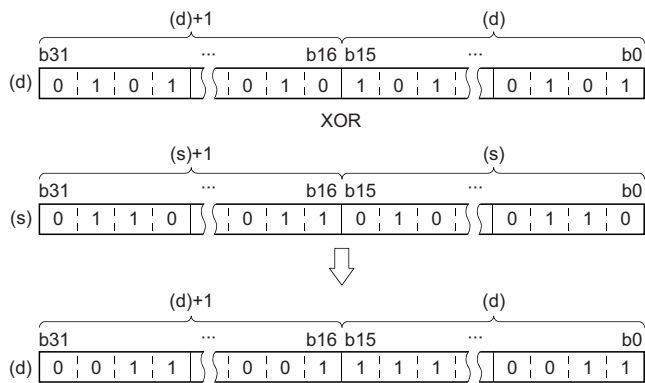
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	排他的論理和を行うデータまたは、データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(d)	排他的論理和の結果を格納する先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (d)で指定されたデバイスのBIN32ビットデータと、(s)で指定されたデバイスのBIN32ビットデータの、ビットごとの排他的論理和を求め、結果を(d)で指定されたデバイスに格納します。



- ビットデバイスの場合、桁指定による点数以降のビットデバイスは0として演算します。

エラー

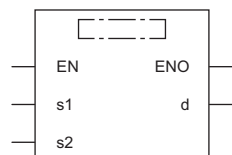
演算エラーはありません。

DXOR(P) [オペランドが3個の場合]

指定した2つのBIN32ビットデータの排他的論理和を行います。

ラダー	ST
	ENO:=DXOR(EN,s1,s2,d); ENO:=DXORP(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
DXOR	
DXORP	

設定データ

■内容、範囲、データ型

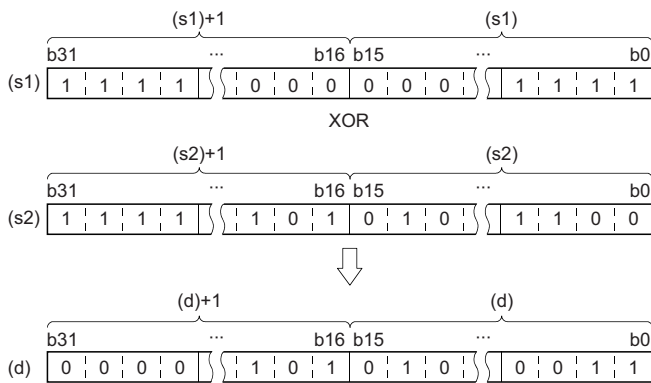
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	排他的論理和を行うデータまたは、データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(s2)	排他的論理和を行うデータまたは、データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(d)	排他的論理和の結果を格納する先頭デバイス	—	符号付きBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (s1)で指定されたデバイスのBIN32ビットデータと、(s2)で指定されたデバイスのBIN32ビットデータの、ビットごとの排他的論理和を求め、結果を(d)で指定されたデバイスに格納します。



- ビットデバイスの場合、桁指定による点数以降のビットデバイスは0として演算します。

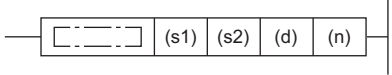
エラー

演算エラーはありません。

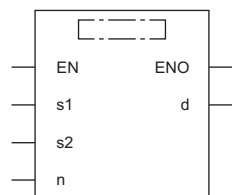
16ビットブロックデータ排他的論理和

BKXOR(P)

指定した2つのBIN16ビットデータブロックの排他的論理和を行います。

ラダー	ST
	ENO:=BKXOR(EN,s1,s2,n,d); ENO:=BKXORP(EN,s1,s2,n,d);

FBD/LD



■実行条件

命令	実行条件
BKXOR	
BKXORP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	論理演算データが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(s2)	論理演算データまたは、論理演算データが格納されている先頭デバイス	-32768~32767	符号付きBIN16ビット	ANY16 ^{*1}
(d)	演算結果を格納する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(n)	演算データ個数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

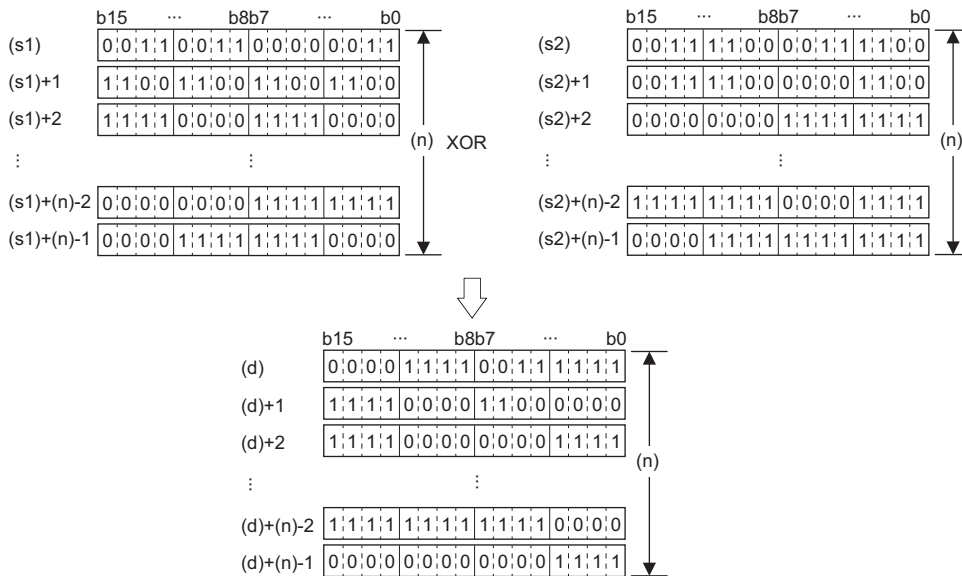
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

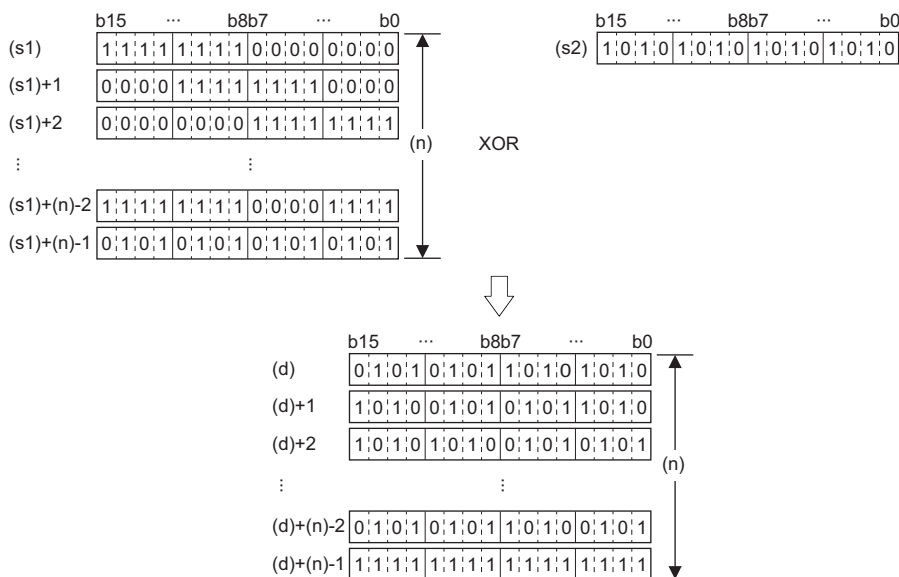
オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1) ^{*1}	—	—	○	—	—	—	○	—	—	—	—
(s2) ^{*1}	—	—	○	—	—	—	○	○	—	—	—
(d) ^{*1}	—	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

*1 (s1)と(d)または(s2)と(d)に同一デバイス番号を指定できます。

- (s1)で指定されたデバイスから(n)点分の内容と、(s2)で指定されたデバイスから(n)点分の内容の排他的論理和を求め、結果を(d)で指定されたデバイス以降に格納します。



- (s2)には-32768~32767(符号付きBIN16ビット)の定数が指定できます。



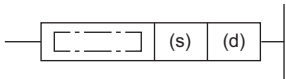
エラー

エラーコード (SD0)	内容
2821H	(s1)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲の一部分が重複しているとき。 ((s1)と(d)に同一デバイスを指定した場合を除く。)
	(s2)から(n)点分のデバイス範囲と、(d)から(n)点分のデバイス範囲の一部分が重複しているとき。 ((s2)と(d)に同一デバイスを指定した場合を除く。)

16ビットデータ否定排他的論理和

WXNR(P) [オペランドが2個の場合]

指定した2つのBIN16ビットデータの否定排他的論理和を行います。

ラダー	ST
	対応していません。 (349ページ WXNR(P) [オペランドが3個の場合])

FBD/LD
対応していません。 (349ページ WXNR(P) [オペランドが3個の場合])

■実行条件

命令	実行条件
WXNR	
WXNRP	

設定データ

■内容、範囲、データ型

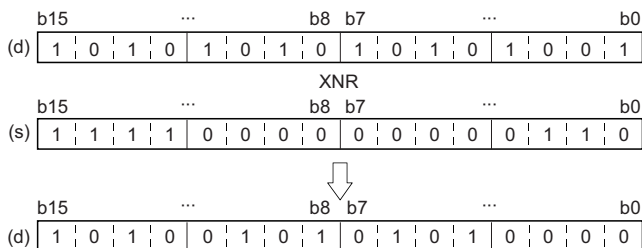
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	否定排他的論理和を行うデータまたは、データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(d)	否定排他的論理和の結果を格納するデバイス	-32768~32767	符号付きBIN16ビット	ANY16

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- ・ (d)で指定されたデバイスのBIN16ビットデータと、(s)で指定されたデバイスのBIN16ビットデータの否定排他的論理和を求め、結果を(d)で指定されたデバイスに格納します。



- ・ ビットデバイスの場合、桁指定による点数以降のビットデバイスは0として演算します。

エラー

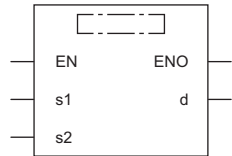
演算エラーはありません。

WXNR(P) [オペランドが3個の場合]

指定した2つのBIN16ビットデータの否定排他的論理和を行います。

ラダー	ST
	ENO:=WXNR(EN,s1,s2,d); ENO:=WXNRP(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
WXNR	
WXNRP	

設定データ

■内容、範囲、データ型

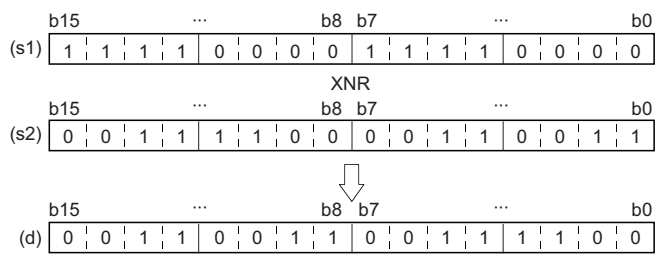
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1), (s2)	否定排他的論理和を求めるデータまたは、データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(d)	否定排他的論理和の結果を格納するデバイス	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- (s1)で指定されたデバイスのBIN16ビットデータと、(s2)で指定されたデバイスのBIN16ビットデータの否定排他的論理和を求め、結果を(d)で指定されたデバイスに格納します。



- ビットデバイスの場合、桁指定による点数以降のビットデバイスは0として演算します。

エラー

演算エラーはありません。

32ビットデータ否定排他的論理和

DXNR(P) [オペランドが2個の場合]

指定した2つのBIN32ビットデータの否定排他的論理和を行います。

ラダー	ST
	対応していません。 (☞ 353ページ DXNR(P) [オペランドが3個の場合])
FBD/LD 対応していません。 (☞ 353ページ DXNR(P) [オペランドが3個の場合])	

■実行条件

命令	実行条件
DXNR	
DXNRP	

設定データ

■内容, 範囲, データ型

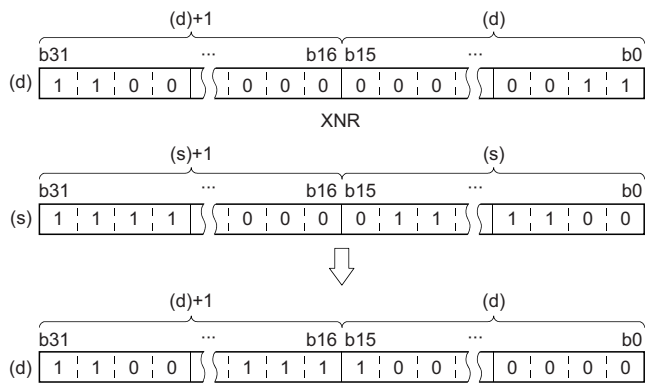
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	否定排他的論理和を行うデータまたは、データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(d)	否定排他的論理和の結果を格納する先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (d)で指定されたデバイスのBIN32ビットデータと、(s)で指定されたデバイスのBIN32ビットデータの否定排他的論理和を求め、結果を(d)で指定されたデバイスに格納します。



- ビットデバイスの場合、桁指定による点数以降のビットデバイスは0として演算します。

エラー

演算エラーはありません。

DXNR(P) [オペランドが3個の場合]

指定した2つのBIN32ビットデータの否定排他的論理和を行います。

ラダー	ST
	ENO:=DXNR(EN,s1,s2,d); ENO:=DXNRP(EN,s1,s2,d);

FBD/LD

■実行条件

命令	実行条件
DXNR	
DXNRP	

設定データ

■内容、範囲、データ型

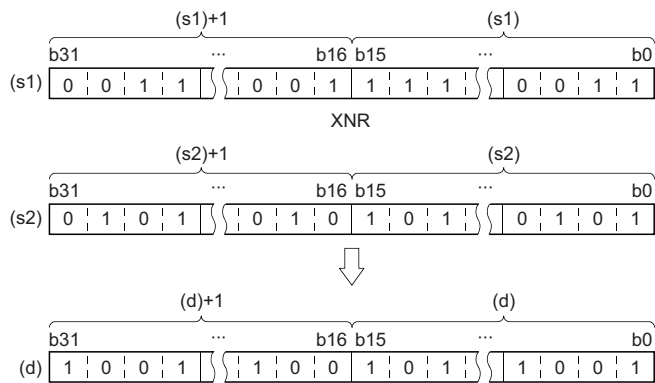
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1), (s2)	否定排他的論理和を求めるデータまたは、データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(d)	否定排他的論理和の結果を格納する先頭デバイス	—	符号付きBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (s1)で指定されたデバイスのBIN32ビットデータと、(s2)で指定されたデバイスのBIN32ビットデータの否定排他的論理和を求め、結果を(d)で指定されたデバイスに格納します。



- ビットデバイスの場合、桁指定による点数以降のビットデバイスは0として演算します。

エラー

演算エラーはありません。

16ビットブロックデータ否定排他的論理和

BKXNR(P)

指定した2つのBIN16ビットデータブロックの否定排他的論理和を行います。

ラダー	ST
	$ENO := BKXNR(EN, s1, s2, n, d);$ $ENO := BKXNRP(EN, s1, s2, n, d);$

FBD/LD

■実行条件

命令	実行条件
BKXNR	
BKXNRP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	論理演算データが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(s2)	論理演算データまたは、論理演算データが格納されている先頭デバイス	-32768~32767	符号付きBIN16ビット	ANY16 ^{*1}
(d)	演算結果を格納する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(n)	演算データ個数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

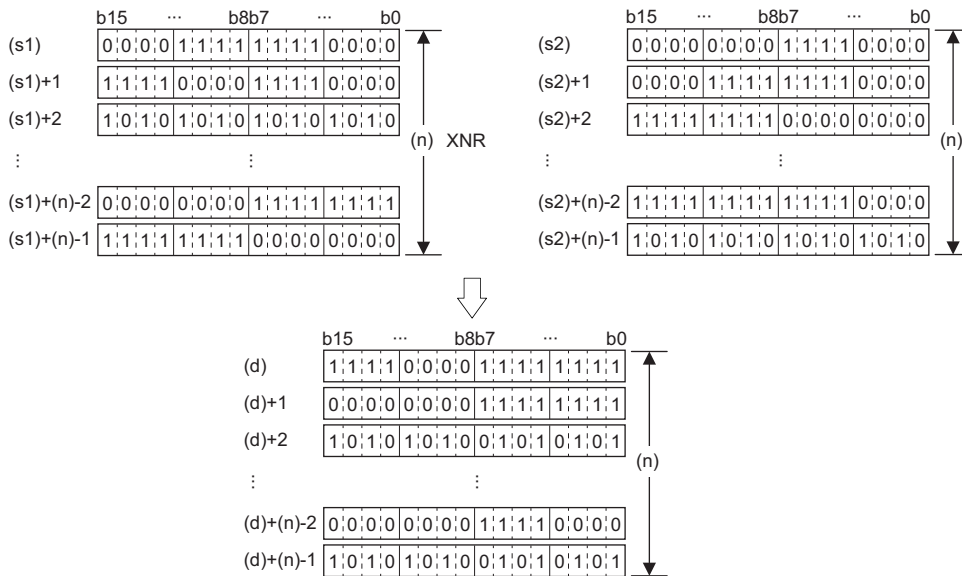
■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1) ^{*1}	—	—	○	—	—	—	○	—	—	—	—
(s2) ^{*1}	—	—	○	—	—	—	○	○	—	—	—
(d) ^{*1}	—	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

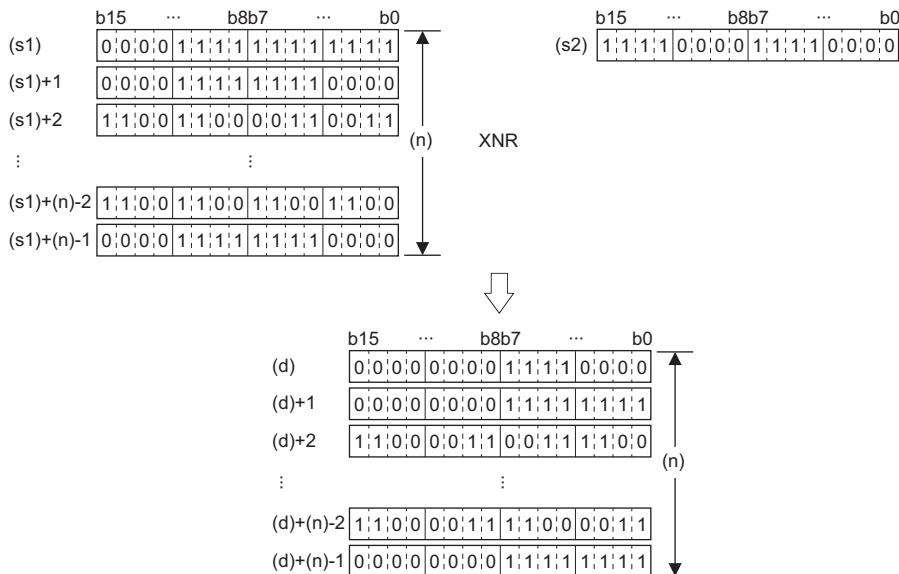
*1 (s1)と(d)または(s2)と(d)に同一デバイス番号を指定できます。

機能

- (s1)で指定されたデバイスから(n)点分の内容と、(s2)で指定されたデバイスから(n)点分の内容の否定排他的論理和を求め、結果を(d)で指定されたデバイス以降に格納します。



- (s2)には-32768~32767(符号付きBIN16ビット)の定数が指定できます。



エラー

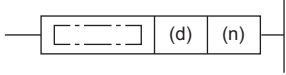
エラーコード (SD0)	内容
2821H	(s1)から(n)点分のデバイス範囲と(d)から(n)点分のデバイス範囲の一部分が重複しているとき。 ((s1)と(d)に同一デバイスを指定した場合を除く。)
	(s2)から(n)点分のデバイス範囲と(d)から(n)点分のデバイス範囲の一部分が重複しているとき。 ((s2)と(d)に同一デバイスを指定した場合を除く。)

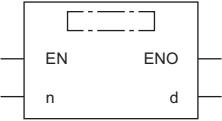
12.4 ビット処理命令

ワードデバイスのビットセット

BSET(P)

指定したワードデバイス内のnビット目をセット(1)します。

ラダー	ST
	ENO:=BSET(EN,n,d); ENO:=BSETP(EN,n,d);

FBD/LD


■実行条件

命令	実行条件
BSET	
BSETP	

設定データ

■内容，範囲，データ型

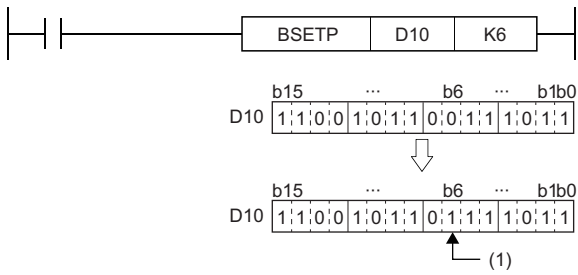
オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	ビットセットするデバイス	—	符号付きBIN16ビット	ANY16
(n)	ビットセットするビットの位置	0~15	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	○	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- (d)で指定のワードデバイスの(n)ビット目をセット(1)します。



(1) D10の6ビット目をセット(1)します。

- (n)で15を超えた場合は、下位4ビットのデータで実行します。

エラー

演算エラーはありません。

ワードデバイスのビットリセット

BRST(P)

指定したワードデバイス内のnビット目をリセット(0)します。

ラダー	ST
	ENO:=BRST(EN,n,d); ENO:=BRSTP(EN,n,d);

FBD/LD

■実行条件

命令	実行条件
BRST	
BRSTP	

設定データ

■内容，範囲，データ型

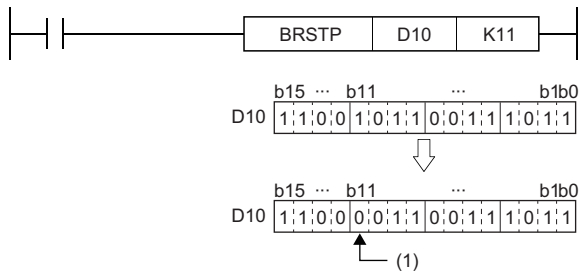
オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	ビットリセットするデバイス	—	符号付きBIN16ビット	ANY16
(n)	ビットリセットするビットの位置	0~15	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□		LT, LST, LC	LZ		K, H	E	\$	
(d)	○	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- (d)で指定のワードデバイスの(n)ビット目をリセット(0)します。



(1) D10の11ビット目をリセット(0)します。

- (n)で15を超えた場合は、下位4ビットのデータで実行します。

エラー

演算エラーはありません。

16ビットテスト

12

TEST(P)

指定したワードデバイス内のnビット目を取り出します。

ラダー	ST
	ENO:=TEST(EN,s1,s2,d); ENO:=TESTP(EN,s1,s2,d);

FBD/LD

■実行条件

命令	実行条件
TEST	
TESTP	

設定データ

■内容、範囲、データ型

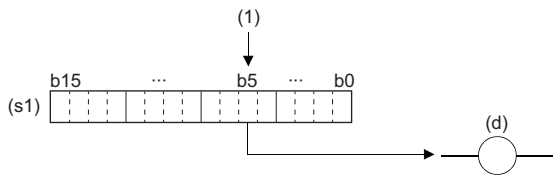
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	抽出するビットデータが格納されているデバイス番号	—	符号付きBIN16ビット	ANY16
(s2)	抽出するビットデータの位置	0~15	符号なしBIN16ビット	ANY16
(d)	抽出したビットデータを格納するビットデバイス番号	—	ビット	ANY_BOOL
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	—	—	—
(d)	○	○	○	○	—	—	—	○	—	—	—

機能

- (s1)で指定したワードデバイスから、(s2)で指定した位置のビットデータを取り出し、(d)で指定したビットデバイスに書き込みます。



(1): (s2)ビット((s2)=5の場合)

- (d)で指定したビットデバイスは、該当ビットが「0」のときOFFし、「1」のときONします。
- (s2)では、1ワードデータの各ビット位置(0~15)を指定します。(s2)に16以上を指定した場合は、 $(s2) \div 16$ の余りの値がビット位置になります。

例

(s2)=18のときは、 $18 \div 16 = 1$ 余り2のため、b2のデータになります。

エラー

演算エラーはありません。

32ビットテスト

12

DTEST(P)

指定した2ワードデバイス内のnビット目を取り出します。

ラダー	ST
	<pre>ENO:=DTEST(EN,s1,s2,d); ENO:=DTESTP(EN,s1,s2,d);</pre>

FBD/LD

■実行条件

命令	実行条件
DTEST	
DTESTP	

設定データ

■内容、範囲、データ型

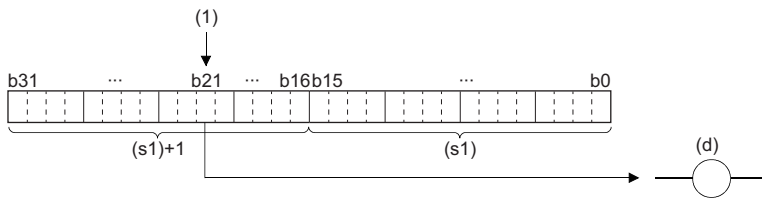
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	抽出するビットデータが格納されているデバイス番号	—	符号付きBIN32ビット	ANY32
(s2)	抽出するビットデータの位置	0~31	符号なしBIN16ビット	ANY16
(d)	抽出したビットデータを格納するビットデバイス番号	—	ビット	ANY_BOOL
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	—	○	○	○	○	○	○	—	—	—	—
(s2)	—	○	○	○	○	—	—	○	—	—	—
(d)	○	○	○	○	—	—	—	—	—	—	—

機能

- (s1)で指定されたダブルワードデバイス内から、(s2)で指定した位置のビットデータを取り出し、(d)で指定したビットデバイスに書き込みます。



(1): (s2)ビット((s2)=21の場合)

- (d)で指定したビットデバイスは、該当ビットが「0」のときOFFし、「1」のときONします。
- (s2)では、2ワードデータの各ビット位置(0~31)を指定します。(s2)に32以上を指定した場合は、 $(s2) \div 32$ の余りの値がビット位置になります。

例

(s2)=34のときは、 $34 \div 32 = 1$ 余り2のため、b2のデータになります。

エラー

演算エラーはありません。

ビットデバイスの一括リセット

BKRST(P)

指定したビットデバイスからn点分のビットデバイスをリセットします。

ラダー	ST
	ENO:=BKRST(EN,n,d); ENO:=BKRSTP(EN,n,d);

FBD/LD

■実行条件

命令	実行条件
BKRST	
BKRSTP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	リセットする先頭デバイス	—	ビット	ANY_BOOL
(n)	リセットするデバイス数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口		LT, LST, LC	LZ		K, H	E	\$	
(d)	○	○	○	—	—	○	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- (d)で指定されたビットデバイスから、(n)点分のビットデバイスをリセットします。
- ビットデバイスのリセット状態は下記ようになります。

デバイス	状態
アナンシェータ(F)	• (d)で指定されたアナンシェータ(F)番号から(n)点分をOFFします。 • OFFしたアナンシェータ番号をSD64~SD79から削除し、前詰めします。 • SD64~SD79に格納されているアナンシェータ数をSD63に格納します。
タイマ(T), 積算タイマ(ST), カウンタ(C), ロングタイマ(LT), ロング積算タイマ(LST), ロングカウンタ(LC)	• (d)で指定されたタイマ(T), 積算タイマ(ST), カウンタ(C), ロングタイマ(LT), ロング積算タイマ(LST), ロングカウンタ(LC)番号から、(n)点分の現在値を0にして、コイル接点をOFFします。
上記以外のビットデバイス	• (d)で指定されたデバイスから(n)点分のコイル、接点をOFFします。

- 指定されたデバイスがOFFの場合、デバイスの状態は変化しません。

エラー

演算エラーはありません。

12.5 データシフト命令

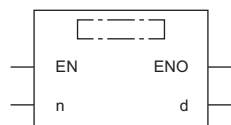
BIN16ビットデータのnビット右シフト

SFR(P)

指定したデバイスのBIN16ビットデータを右へ(n)ビット分シフトさせます。シフト後の空きデータには0が格納されます。

ラダー	ST
	ENO:=SFR(EN,n,d); ENO:=SFRP(EN,n,d);

FBD/LD



■実行条件

命令	実行条件
SFR	
SFRP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	シフトするデバイス	—	符号付きBIN16ビット	ANY16
(n)	シフト数	0~15	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ	K, H	E	\$	
(d)	○*1	○	○	○	○	—	—	○	—	—	—
(n)	○*1	○	○	○	○	—	—	○	—	—	—

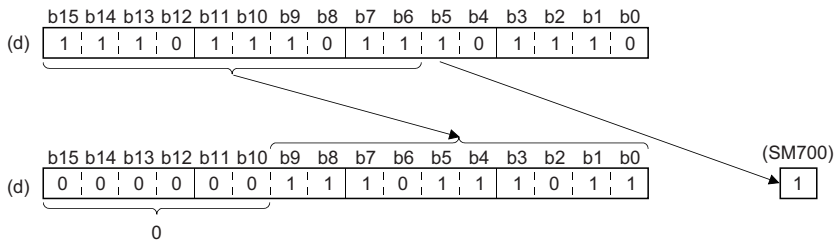
*1 FX, FYは使用できません。

機能

- (d)で指定したデバイスのBIN16ビットデータを、右へ(n)ビット分シフトさせます。
- 最上位から(n)ビット分は0になります。
- SM700(キャリフラグ)には、シフト範囲から一つ右の1ビットが格納されます。

例

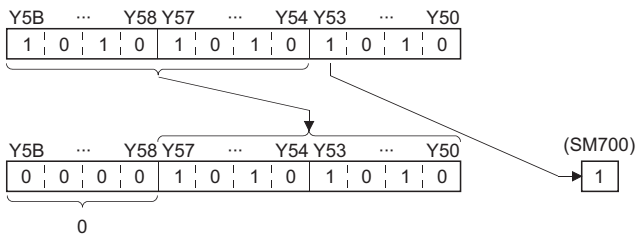
(n)=6の場合



- (d)にビットデバイスを指定した場合は、桁指定で指定したデバイス範囲で右シフトします。

例

(n)=4の場合



- 実際にシフトするビット数は、 $(n) \div (\text{桁指定で指定された点数})$ の余りになります。たとえば、 $(n)=15$ 、(桁指定で指定された点数)=8ビットのときは、 $15 \div 8 = 1$ 余り7のため、7ビットのシフトになります。
- (n) は0~15を指定します。 (n) に16以上の値が指定された場合は、 $(n) \div 16$ の余りの値で右シフトします。たとえば、 $(n)=18$ のときは、 $18 \div 16 = 1$ 余り2のため、右へ2ビット分シフトします。

エラー

演算エラーはありません。

BIN16ビットデータのnビット左シフト

SFL(P)

指定したデバイスのBIN16ビットデータを左へ(n)ビット分シフトさせます。シフト後の空きデータには0が格納されます。

ラダー	ST
	ENO:=SFL(EN,n,d); ENO:=SFLP(EN,n,d);

FBD/LD

■実行条件

命令	実行条件
SFL	
SFLP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	シフトするデバイス	—	符号付きBIN16ビット	ANY16
(n)	シフト数	0~15	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(d)	○*1	○	○	○	○	—	—	○	—	—	—	—	—
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—	—

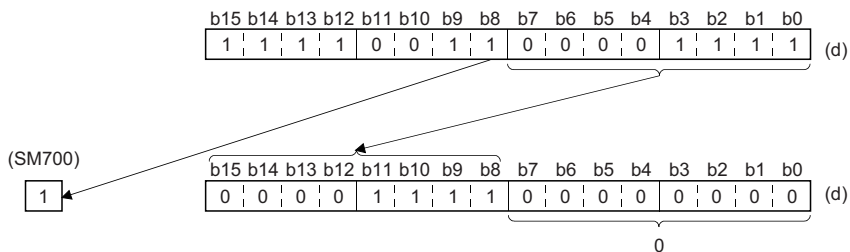
*1 FX, FYは使用できません。

機能

- (d)で指定したデバイスのBIN16ビットデータを、左へ(n)ビット分シフトさせます。
- 最下位から(n)ビット分は0になります。
- SM700(キャリフラグ)には、シフト範囲から一つ左の1ビットが格納されます。

例

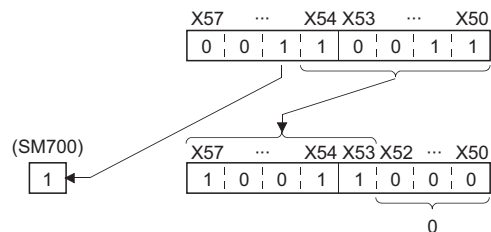
(n)=8の場合



- (d)にビットデバイスを指定した場合は、桁指定で指定したデバイス範囲で左シフトします。

例

(n)=5の場合



- 実際にシフトするビット数は、 $(n) \div (\text{桁指定で指定された点数})$ の余りになります。たとえば、 $(n)=15$ 、(桁指定で指定された点数)=8ビットのときは、 $15 \div 8 = 1$ 余り7のため、7ビットのシフトになります。
- (n)は0~15を指定します。(n)に16以上の値が指定された場合は、 $(n) \div 16$ の余りの値で左シフトします。たとえば、 $(n)=18$ のときは、 $18 \div 16 = 1$ 余り2のため、左へ2ビット分シフトします。

エラー

演算エラーはありません。

nビットデータの1ビット右シフト

BSFR(P)

指定したデバイスから(n)点分を右へ1ビットシフトさせます。シフト後の空きデータには0が格納されます。

ラダー	ST
	ENO:=BSFR(EN,n,d); ENO:=BSFRP(EN,n,d);

FBD/LD

■実行条件

命令	実行条件
BSFR	
BSFRP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	シフトするデバイス	—	ビット	ANY_BOOL
(n)	シフトするデータの数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

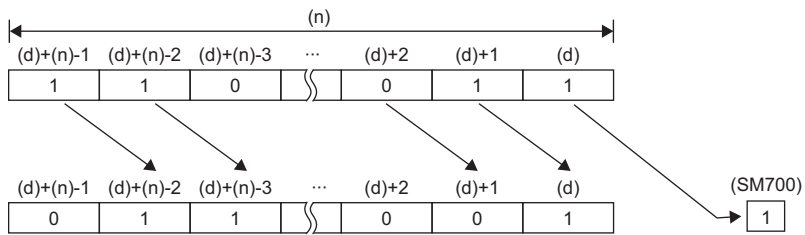
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	○*1	—	○	—	—	—	—	○	—	—	—	—
(n)	○*2	○	○	○	○	—	—	○	○	—	—	—

*1 T, C, STは使用できません。

*2 FX, FYは使用できません。

機能

- (d)で指定したデバイスから(n)点分を、右へ1ビットシフトさせます。
- 最上位から1ビット分は0になります。
- SM700(キャリフラグ)には、シフト範囲から一つ右の1ビットが格納されます。



- (n)で指定した値が0のときは無処理になります。

エラー

演算エラーはありません。

nビットデータの1ビット左シフト

BSFL(P)

指定したデバイスから(n)点分を左へ1ビットシフトさせます。シフト後の空きデータには0が格納されます。

ラダー	ST
	ENO:=BSFL(EN,n,d); ENO:=BSFLP(EN,n,d);

FBD/LD

■実行条件

命令	実行条件
BSFL	
BSFLP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	シフトするデバイス	—	ビット	ANY_BOOL
(n)	シフトするデータの数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

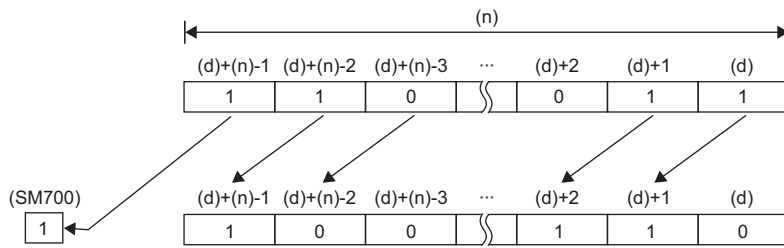
オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(d)	○*1	—	○	—	—	—	○	—	—	—	—
(n)	○*2	○	○	○	○	—	—	○	—	—	—

*1 T, C, STは使用できません。

*2 FX, FYは使用できません。

機能

- (d)で指定したデバイスから(n)点分を、左へ1ビットシフトさせます。
- 最下位から1ビット分は0になります。
- SM700(キャリフラグ)には、シフト範囲から一つ左の1ビットが格納されます。



- (n)で指定した値が0のときは無処理になります。

エラー

演算エラーはありません。

nワードデータの1ワード右シフト

DSFR(P)

指定したデバイスから(n)点分を右へ1ワードシフトさせます。シフト後の空きデータには0が格納されます。

ラダー	ST
	ENO:=DSFR(EN,n,d); ENO:=DSFRP(EN,n,d);

FBD/LD

■実行条件

命令	実行条件
DSFR	
DSFRP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	シフトするデバイス	—	ワード	ANY16
(n)	シフトするデータの数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

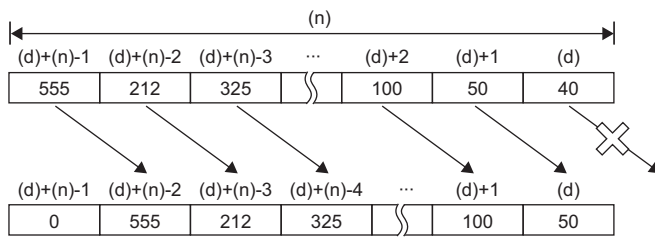
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—	—

*1 FX, FYは使用できません。

機能

- (d)で指定したデバイスから(n)点分のデータを、右へ1ワードシフトさせます。
- 最上位から1ワード分は0になります。



- (n)で指定した値が0のときは無処理になります。

エラー

演算エラーはありません。

nワードデータの1ワード左シフト

DSFL(P)

指定したデバイスから(n)点分を左へ1ワードシフトさせます。シフト後の空きデータには0が格納されます。

ラダー	ST
	ENO:=DSFL(EN,n,d); ENO:=DSFLP(EN,n,d);

FBD/LD

■実行条件

命令	実行条件
DSFL	
DSFLP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	シフトするデバイス	—	ワード	ANY16
(n)	シフトするデータの数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

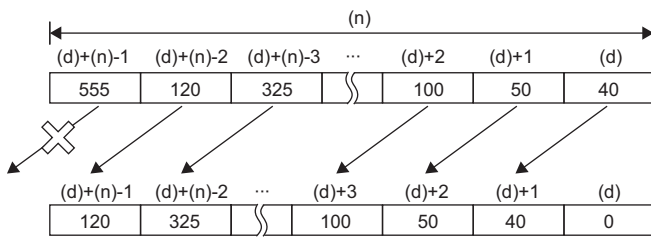
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—

*1 FX, FYは使用できません。

機能

- (d)で指定したデバイスから(n)点分のデータを、左へ1ワードシフトさせます。
- 最下位から1ワード分は0になります。



- (n)で指定した値が0の場合は無処理になります。

エラー

演算エラーはありません。

nビットデータのnビット右シフト

SFTBR(P)

指定したデバイスから(n1)ビット分のデータ範囲内で、右へ(n2)ビット分シフトさせます。シフト後の空きデータには0が格納されます。

ラダー	ST
	ENO:=SFTBR(EN,n1,n2,d); ENO:=SFTBRP(EN,n1,n2,d);

FBD/LD

■実行条件

命令	実行条件
SFTBR	
SFTBRP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	シフトするデバイス	—	ビット	ANY_BOOL
(n1)	シフトするデータの数	0~64	符号なしBIN16ビット	ANY16
(n2)	シフト数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(d)	○*1	—	○	—	—	—	○	—	—	—	—
(n1)	○*2	○	○	○	○	—	○	○	—	—	—
(n2)	○*2	○	○	○	○	—	○	○	—	—	—

*1 T, C, STは使用できません。

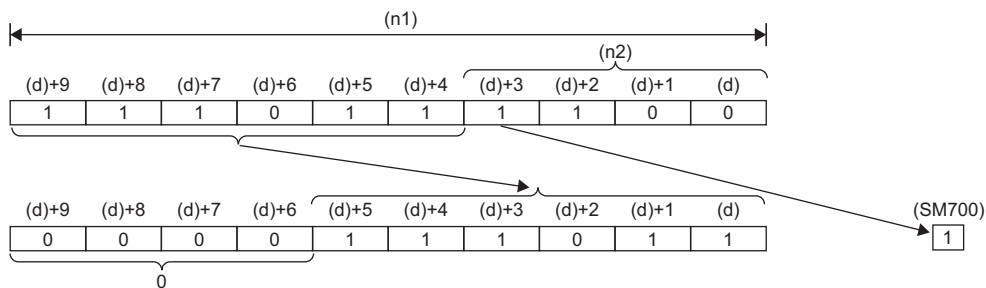
*2 FX, FYは使用できません。

機能

- (d)で指定したデバイスから(n1)ビット分のデータ範囲内で、右へ(n2)ビット分シフトさせます。
- SM700(キャリフラグ)には、シフト範囲から一つ右の1ビットが格納されます。

例

(n1)=10, (n2)=4の場合



- (n1), (n2)は(n1)>(n2)で指定します。(n1)≤(n2)の場合は、(n2)÷(n1)の余りの値でシフトされます。ただし、(n2)÷(n1)の余りの値が0のときは、無処理となります。
- (n1)は1~64で設定します。
- 最上位から(n2)ビット分は0となります。(n1)<(n2)の場合は、(n2)÷(n1)の余りの値分0となります。
- (n1)または(n2)で指定した値が0のとき、無処理となります。

エラー

エラーコード (SD0)	内容
3285H	(n1)が0~64以外のとき。

nビットデータのnビット左シフト

SFTBL(P)

指定したデバイスから(n1)ビット分のデータ範囲内で、左へ(n2)ビット分シフトさせます。シフト後の空きデータには0が格納されます。

ラダー	ST
	ENO:=SFTBL(EN,n1,n2,d); ENO:=SFTBLP(EN,n1,n2,d);

FBD/LD

■実行条件

命令	実行条件
SFTBL	
SFTBLP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	シフトするデバイス	—	ビット	ANY_BOOL
(n1)	シフトするデータの数	0~64	符号なしBIN16ビット	ANY16
(n2)	シフト数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(d)	○*1	—	○	—	—	—	○	—	—	—	—
(n1)	○*2	○	○	○	○	—	○	○	—	—	—
(n2)	○*2	○	○	○	○	—	○	○	—	—	—

*1 T, C, STは使用できません。

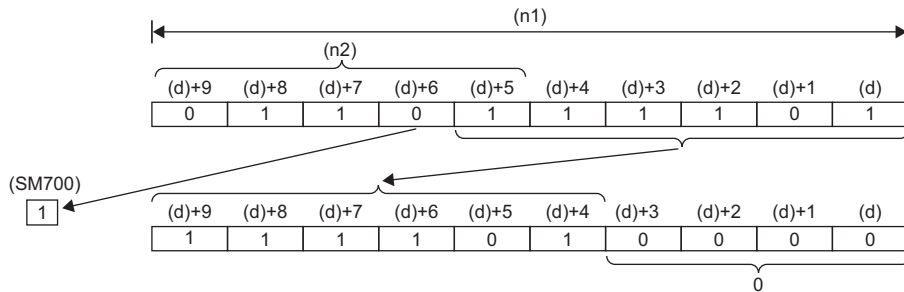
*2 FX, FYは使用できません。

機能

- (d)で指定したデバイスから(n1)ビット分のデータ範囲内で、左へ(n2)ビット分シフトさせます。
- SM700(キャリフラグ)には、シフト範囲から一つ左の1ビットが格納されます。

例

(n1)=10, (n2)=4の場合



- (n1), (n2)は(n1)>(n2)で指定します。(n1)≤(n2)の場合は、(n2)÷(n1)の余りの値でシフトされます。ただし、(n2)÷(n1)の余りの値が0のときは、無処理となります。
- (n1)は1~64で設定します。
- 最下位から(n2)ビット分は0となります。(n1)<(n2)の場合は、(n2)÷(n1)の余りの値分0となります。
- (n1)または(n2)で指定した値が0のとき、無処理となります。

エラー

エラーコード (SD0)	内容
3285H	(n1)が0~64以外のとき。

nワードデータのnワード右シフト

SFTWR(P)

指定したデバイスから(n1)ワード分のデータ範囲内で、右へ(n2)ワード分シフトさせます。シフト後の空きデータには0が格納されます。

ラダー	ST
	ENO:=SFTWR(EN,n1,n2,d); ENO:=SFTWRP(EN,n1,n2,d);

FBD/LD

■実行条件

命令	実行条件
SFTWR	
SFTWRP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	シフトするデバイス	—	ワード	ANY16
(n1)	シフトするデータの数	0~65535	符号なしBIN16ビット	ANY16
(n2)	シフト数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(d)	—	—	○	—	—	—	○	—	—	—	—
(n1)	○*1	○	○	○	○	—	○	○	—	—	—
(n2)	○*1	○	○	○	○	—	○	○	—	—	—

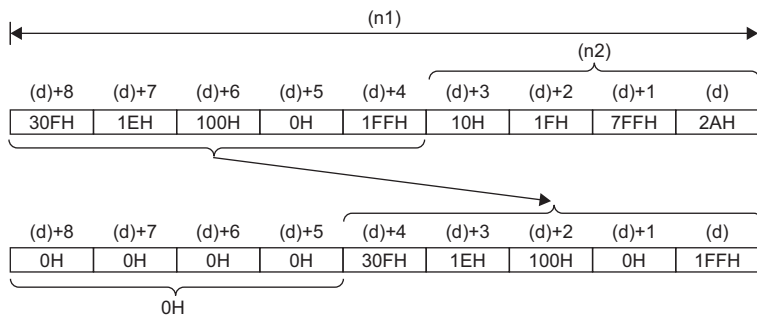
*1 FX, FYは使用できません。

機能

- (d)で指定したデバイスから(n1)ワード分のデータ範囲内で、右へ(n2)ワード分シフトさせます。
- 最上位から(n2)ワード分は0Hになります。
- (n1)または(n2)で指定した値が0Hのときは、無処理となります。
- $(n1) \leq (n2)$ の場合は、(d)で指定したデバイスから(n1)ワード分のデータは、すべて0Hとなります。

例

(n1)=9, (n2)=4の場合



エラー

演算エラーはありません。

nワードデータのnワード左シフト

SFTWL(P)

指定したデバイスから(n1)ワード分のデータ範囲内で、左へ(n2)ワードシフトさせます。シフト後の空きデータには0が格納されます。

ラダー	ST
	ENO:=SFTWL(EN,n1,n2,d); ENO:=SFTWL(P)(EN,n1,n2,d);

FBD/LD

■実行条件

命令	実行条件
SFTWL	
SFTWLP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	シフトするデバイス	—	ワード	ANY16
(n1)	シフトするデータの数	0~65535	符号なしBIN16ビット	ANY16
(n2)	シフト数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(d)	—	—	○	—	—	—	○	—	—	—	—
(n1)	○*1	○	○	○	○	—	○	○	—	—	—
(n2)	○*1	○	○	○	○	—	○	○	—	—	—

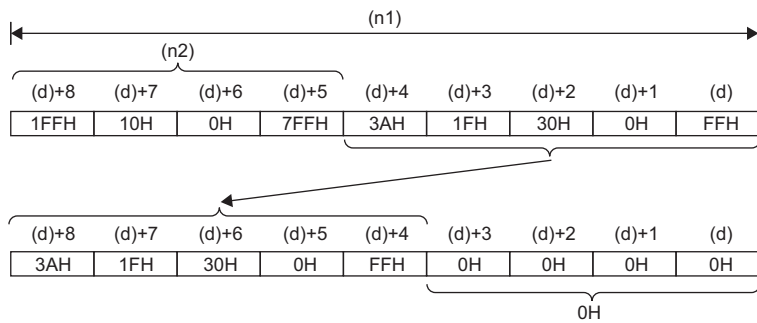
*1 FX, FYは使用できません。

機能

- (d)で指定したデバイスから(n1)ワード分のデータ範囲内で、左へ(n2)ワード分シフトさせます。
- 最下位から(n2)ワード分は0Hになります。
- (n1)または(n2)で指定した値が0Hのときは、無処理となります。
- $(n1) \leq (n2)$ の場合は、(d)で指定したデバイスから(n1)ワード分のデータは、すべて0Hとなります。

例

(n1)=9, (n2)=4の場合



エラー

演算エラーはありません。

12.6 データ変換命令

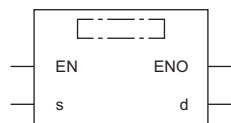
BINデータ→BCD4桁変換

BCD(P)

指定したBIN16ビットデータをBCD4桁データに変換します。

ラダー	ST
	ENO:=BCD(EN,s,d); ENO:=BCDP(EN,s,d);

FBD/LD



■実行条件

命令	実行条件
BCD	
BCDP	

設定データ

■内容, 範囲, データ型

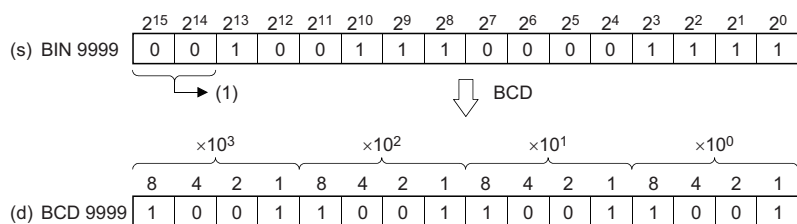
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータまたは, BINデータが格納されているデバイス	0~9999	符号付きBIN16ビット	ANY16
(d)	BCDデータを格納するデバイス	—	BCD4桁	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	○	○	○	○	—	—	○	○	—	—	—	—
(d)	—	○	○	○	○	—	—	○	—	—	—	—	—

機能

- (s)で指定されたデバイスのBIN16ビットデータ(0~9999)をBCD4桁データに変換して、(d)で指定されたデバイスに格納します。



(1): 必ず0にしてください。

エラー

エラーコード (SD0)	内容
3281H	(s)のデータが0~9999以外るとき。

BINデータ→BCD8桁変換

DBCD(P)

指定したBIN32ビットデータをBCD8桁データに変換します。

ラダー	ST
	ENO:=DBCD(EN,s,d); ENO:=DBCDP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
DBCD	
DBCDP	

設定データ

■内容，範囲，データ型

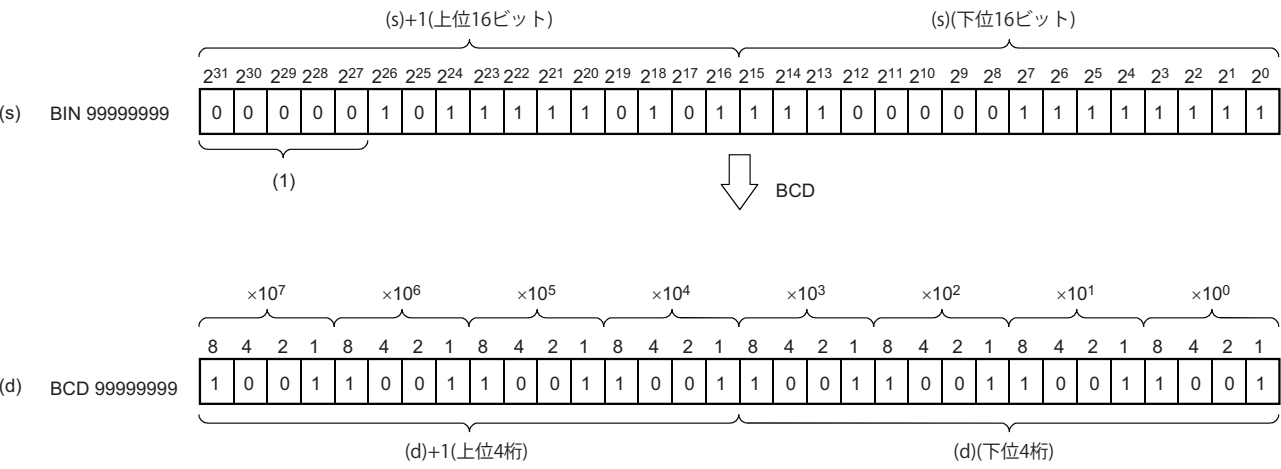
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータまたは，BINデータが格納されている先頭デバイス	0~99999999	符号付きBIN32ビット	ANY32
(d)	BCDデータを格納する先頭デバイス	—	BCD8桁	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定されたデバイスのBIN32ビットデータ(0~99999999)をBCD8桁データに変換して、(d)で指定されたデバイスに格納します。



(1): 必ず0にしてください。(上位5ビット)

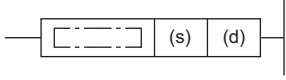
エラー

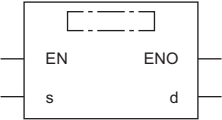
エラーコード (SD0)	内容
3281H	(s)のデータが0~99999999以外るとき。

BCD4桁→BIN16ビットデータ変換

BIN(P)

指定したBCD4桁データをBIN16ビットデータに変換します。

ラダー	ST
	ENO:=BIN(EN,s,d); ENO:=BINP(EN,s,d);

FBD/LD


■実行条件

命令	実行条件
BIN	
BINP	

設定データ

■内容, 範囲, データ型

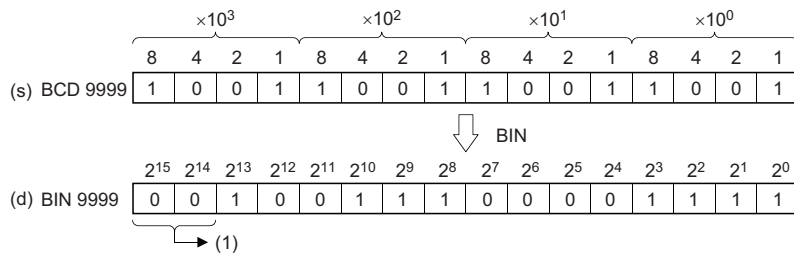
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BCDデータまたは、BCDデータが格納されているデバイス	0~9999	BCD4桁	ANY16
(d)	BINデータを格納するデバイス	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- (s)で指定されたデバイスのBCD4桁データ(0~9999)をBIN16ビットデータに変換して、(d)で指定されたデバイスに格納します。



(1): 必ず0になります。

エラー

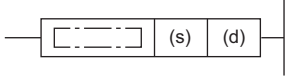
エラーコード (SD0)	内容
3281H	(s)の各桁に0~9以外の値があるとき。 ^{*1}

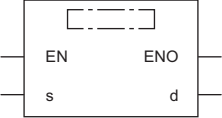
- ^{*1} SM754(BIN, DBIN命令エラー不可フラグ)をONしておくことにより、本エラーを検出しないようにすることができます。
 なお、範囲外の数値を設定した場合、SM754のON/OFFに関わらず命令は実行されません。
 また、BIN(P)命令の場合、エラーの有無に関わらず指令(実行条件)をOFF→ONさせるまで、次の演算は実行されません。

BCD8桁→BIN32ビットデータ変換

DBIN(P)

指定したBCD8桁データをBIN32ビットデータに変換します。

ラダー	ST
	ENO:=DBIN(EN,s,d); ENO:=DBINP(EN,s,d);

FBD/LD


■実行条件

命令	実行条件
DBIN	
DBINP	

設定データ

■内容、範囲、データ型

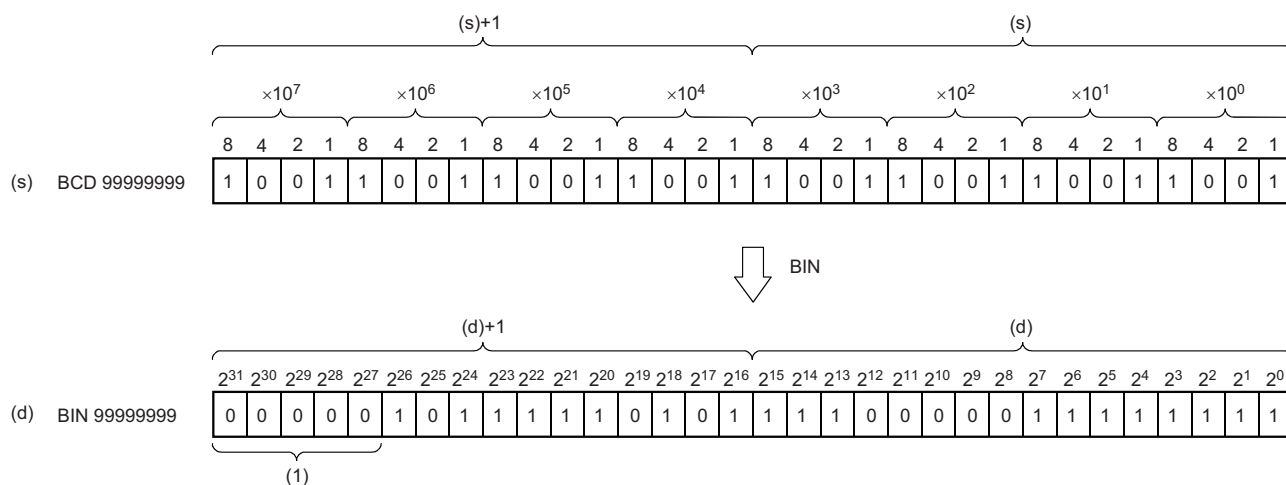
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BCDデータまたは、BCDデータが格納されている先頭デバイス	0~99999999	BCD8桁	ANY32
(d)	BINデータを格納する先頭デバイス	—	符号付きBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定されたデバイスのBCD8桁データ(0~99999999)をBIN32ビットデータに変換して、(d)で指定されたデバイスに格納します。



(1): 必ず0になります。

エラー

エラーコード (SD0)	内容
3281H	(s)の各桁に0~9以外の値があるとき。 ^{*1}

- ^{*1} SM754(BIN, DBIN命令エラー不可フラグ)をONしておくことにより、本エラーを検出しないようにすることができます。
 なお、範囲外の数値を設定した場合、SM754のON/OFFに関わらず命令は実行されません。
 また、DBIN(P)命令の場合、エラーの有無に関わらず指令(実行条件)をOFF→ONさせるまで、次の演算は実行されません。

単精度実数→符号付きBIN16ビットデータ

FLT2INT(P)

指定した単精度実数を符号付きBIN16ビットデータに変換します。

ラダー	ST
	ENO:=FLT2INT(EN,s,d); ENO:=FLT2INTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
FLT2INT	
FLT2INTP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	単精度実数または、単精度実数が格納されている先頭デバイス	-32768~32767	単精度実数	ANYREAL_32
(d)	BINデータを格納するデバイス	—	符号付きBIN16ビット	ANY16_S
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	—	—	○	○	○	○	○	—	○	—	—
(d)	○	○	○	○	○	—	—	—	—	—	—

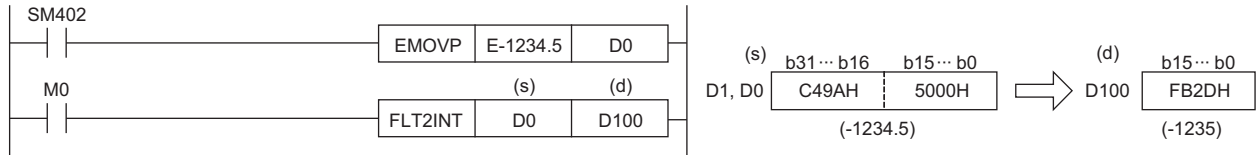
機能

- (s)で指定された単精度実数を，符号付きBIN16ビットデータに変換して，(d)で指定されたデバイスに格納します。
- 変換後のデータは，(s)で指定された単精度実数の小数点以下1桁目を四捨五入した値となります。
- エンジニアリングツールから入力値を設定する場合，丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は，下記を参照してください。

☞ 39ページ 注意事項

例

M0がONしたとき，D0～D1に格納された単精度実数を，符号付きBIN16ビットデータに変換して，D100へ格納します。



エラー

エラーコード (SD0)	内容
3281H	(s)で設定する単精度実数が-32768～32767の範囲外の時。
3282H	(s)に特殊な数を設定したとき。 • 設定した単精度実数が下記範囲内でない。 $0, 2^{-126} \leq (s) < 2^{128}$ • 設定したデバイス，ラベルの内容が-0，非正規化数，非数，$\pm\infty$の時。

単精度実数→符号なしBIN16ビットデータ

FLT2UINT(P)

指定した単精度実数を符号なしBIN16ビットデータに変換します。

ラダー	ST
	ENO:=FLT2UINT(EN,s,d); ENO:=FLT2UINTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
FLT2UINT	
FLT2UINTP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	単精度実数または, 単精度実数が格納されている先頭デバイス	0~65535	単精度実数	ANYREAL_32
(d)	BINデータを格納するデバイス	—	符号なしBIN16ビット	ANY16_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	○	○	○	○	○	○	—	○	—	—
(d)	○	○	○	○	○	—	—	○	○	—	—	—	—

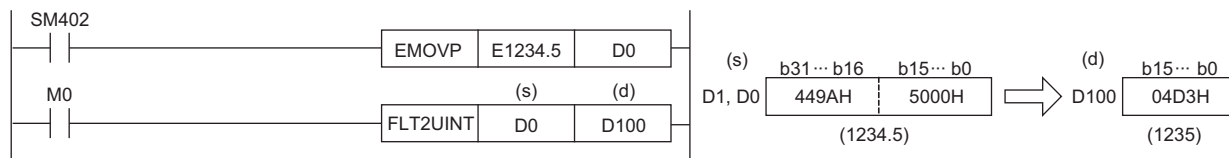
機能

- (s)で指定された単精度実数を，符号なしBIN16ビットデータに変換して，(d)で指定されたデバイスに格納します。
- 変換後のデータは，(s)で指定された単精度実数の小数点以下1桁目を四捨五入した値となります。
- エンジニアリングツールから入力値を設定する場合，丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は，下記を参照してください。

☞ 39ページ 注意事項

例

M0がONしたとき，D0～D1に格納された単精度実数を，符号なしBIN16ビットデータに変換して，D100へ格納します。



エラー

エラーコード (SD0)	内容
3281H	(s)で設定する単精度実数が0～65535の範囲外するとき。
3282H	(s)に特殊な数を設定したとき。 • 設定した単精度実数が下記範囲内でない。 $0, 2^{-126} \leq (s) < 2^{128}$ • 設定したデバイス，ラベルの内容が-0，非正規化数，非数，$\pm\infty$のとき。

単精度実数→符号付きBIN32ビットデータ

FLT2DINT(P)

指定した単精度実数を符号付きBIN32ビットデータに変換します。

ラダー	ST
	ENO:=FLT2DINT(EN,s,d); ENO:=FLT2DINTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
FLT2DINT	
FLT2DINTP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	単精度実数または、単精度実数が格納されている先頭デバイス	-2147483648~2147483647	単精度実数	ANYREAL_32
(d)	BINデータを格納する先頭デバイス	—	符号付きBIN32ビット	ANY32_S
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

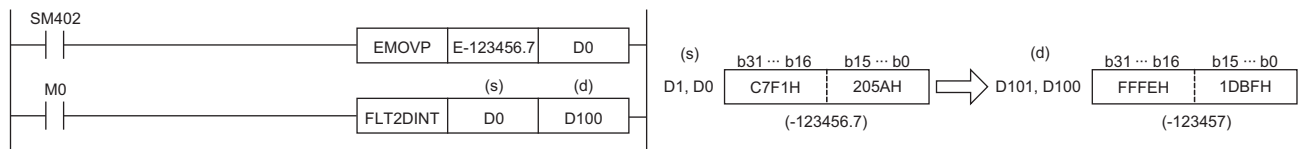
機能

- (s)で指定された単精度実数を，符号付きBIN32ビットデータに変換して，(d)で指定されたデバイスに格納します。
- 変換後のデータは，(s)で指定された単精度実数の小数点以下1桁目を四捨五入した値となります。
- エンジニアリングツールから入力値を設定する場合，丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は，下記を参照してください。

📖 39ページ 注意事項

例

M0がONしたとき，D0~D1に格納された単精度実数を，符号付きBIN32ビットデータに変換して，D100~D101へ格納します。



エラー

エラーコード (SD0)	内容
3281H	(s)で設定する単精度実数が-2147483648~2147483647の範囲外のとき。
3282H	(s)に特殊な数を設定したとき。 • 設定した単精度実数が下記範囲内でない。 $0, 2^{-126} \leq (s) < 2^{128}$ • 設定したデバイス，ラベルの内容が-0，非正規化数，非数，$\pm\infty$のとき。

単精度実数→符号なしBIN32ビットデータ

FLT2UDINT(P)

指定した単精度実数を符号なしBIN32ビットデータに変換する。

ラダー	ST
	ENO:=FLT2UDINT(EN,s,d); ENO:=FLT2UDINTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
FLT2UDINT	
FLT2UDINTP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	単精度実数または、単精度実数が格納されている先頭デバイス	0~4294967295	単精度実数	ANYREAL_32
(d)	BINデータを格納する先頭デバイス	—	符号なしBIN32ビット	ANY32_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	—	—	○	○	○	○	○	—	○	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

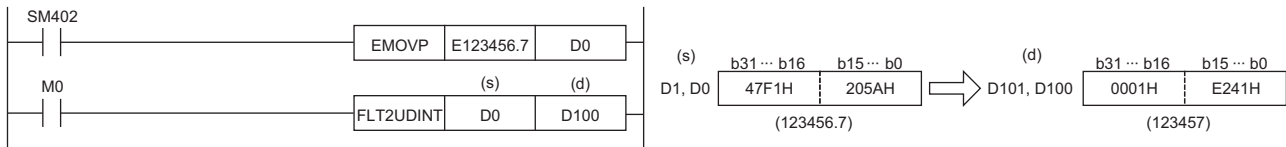
機能

- (s)で指定された単精度実数を，符号なしBIN32ビットデータに変換して，(d)で指定されたデバイスに格納します。
- 変換後のデータは，(s)で指定された単精度実数の小数点以下1桁目を四捨五入した値となります。
- エンジニアリングツールから入力値を設定する場合，丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は，下記を参照してください。

39ページ 注意事項

例

M0がONしたとき，D0～D1に格納された単精度実数を，符号なしBIN32ビットデータに変換して，D100～D101へ格納します。



エラー

エラーコード (SD0)	内容
3281H	(s)で設定する単精度実数が0～4294967295の範囲外るとき。
3282H	(s)に特殊な数を設定したとき。 • 設定した単精度実数が下記範囲内でない。 $0, 2^{-126} \leq (s) < 2^{128}$ • 設定したデバイス，ラベルの内容が-0，非正規化数，非数，$\pm\infty$のとき。

倍精度実数→符号付きBIN16ビットデータ

DBL2INT(P)

指定した倍精度実数を符号付きBIN16ビットデータに変換します。

ラダー	ST
	ENO:=DBL2INT(EN,s,d); ENO:=DBL2INTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
DBL2INT	
DBL2INTP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	倍精度実数または、倍精度実数が格納されている先頭デバイス	-32768~32767	倍精度実数	ANYREAL_64
(d)	BINデータを格納するデバイス	—	符号付きBIN16ビット	ANY16_S
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	—	—
(d)	○	—	○	—	○	—	—	○	—	—	—

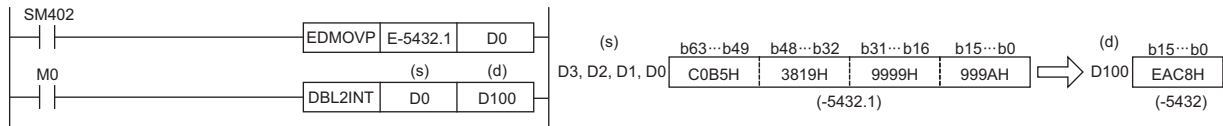
機能

- (s)で指定された倍精度実数を，符号付きBIN16ビットデータに変換して，(d)で指定されたデバイスに格納します。
- 変換後のデータは，(s)で指定された倍精度実数の小数点以下1桁目を四捨五入した値となります。
- エンジニアリングツールから入力値を設定する場合，丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は，下記を参照してください。

☞ 39ページ 注意事項

例

M0がONしたとき，D0～D3に格納された倍精度実数を，符号付きBIN16ビットデータに変換して，D100へ格納する。



エラー

エラーコード (SD0)	内容
3281H	(s)で設定する倍精度実数が-32768～32767の範囲外の時。
3282H	(s)に特殊な数を設定したとき。 • 設定した倍精度実数が下記範囲内でない。 $0, 2^{-1022} \leq (s) < 2^{1024}$ • 設定したデバイス，ラベルの内容が-0，非正規化数，非数，$\pm\infty$の時。

倍精度実数→符号なしBIN16ビットデータ

DBL2UINT(P)

指定した倍精度実数を符号なしBIN16ビットデータに変換します。

ラダー	ST
	ENO:=DBL2UINT(EN,s,d); ENO:=DBL2UINTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
DBL2UINT	
DBL2UINTP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	倍精度実数または、倍精度実数が格納されている先頭デバイス	0~65535	倍精度実数	ANYREAL_64
(d)	BINデータを格納するデバイス	—	符号なしBIN16ビット	ANY16_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□		LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	○	—	○	—	○	—	—	○	—	—	—	—

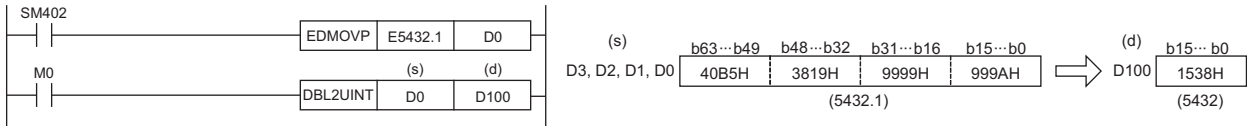
機能

- (s)で指定された倍精度実数を，符号なしBIN16ビットデータに変換して，(d)で指定されたデバイスに格納します。
- 変換後のデータは，(s)で指定された倍精度実数の小数点以下1桁目を四捨五入した値となります。
- エンジニアリングツールから入力値を設定する場合，丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は，下記を参照してください。

📖 39ページ 注意事項

例

M0がONしたとき，D0~D3に格納された倍精度実数を，符号なしBIN16ビットデータに変換して，D100へ格納します。



エラー

エラーコード (SD0)	内容
3281H	(s)で設定する倍精度実数が0~65535の範囲外するとき。
3282H	(s)に特殊な数を設定したとき。 • 設定した倍精度実数が下記範囲内でない。 $0, 2^{-1022} \leq (s) < 2^{1024}$ • 設定したデバイス，ラベルの内容が-0，非正規化数，非数，±∞のとき。

倍精度実数→符号付きBIN32ビットデータ

DBL2DINT(P)

指定した倍精度実数を符号付きBIN32ビットデータに変換します。

ラダー	ST
	ENO:=DBL2DINT(EN,s,d); ENO:=DBL2DINTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
DBL2DINT	
DBL2DINTP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	倍精度実数または、倍精度実数が格納されている先頭デバイス	-2147483648~2147483647	倍精度実数	ANYREAL_64
(d)	BINデータを格納する先頭デバイス	—	符号付きBIN32ビット	ANY32_S
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—
(d)	○	—	○	—	○	○	○	○	—	—	—

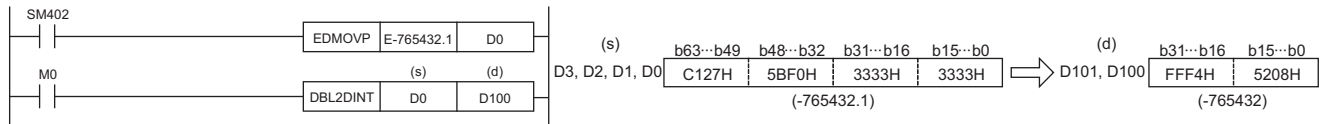
機能

- (s)で指定された倍精度実数を，符号付きBIN32ビットデータに変換して，(d)で指定されたデバイスに格納します。
- 変換後のデータは，(s)で指定された倍精度実数の小数点以下1桁目を四捨五入した値となります。
- エンジニアリングツールから入力値を設定する場合，丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は，下記を参照してください。

☞ 39ページ 注意事項

例

M0がONしたとき，D0~D3に格納された倍精度実数を，符号付きBIN32ビットデータに変換して，D100~D101へ格納する。



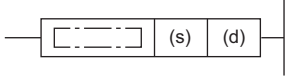
エラー

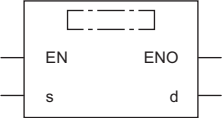
エラーコード (SD0)	内容
3281H	(s)で設定する倍精度実数が-2147483648~2147483647の範囲外のとき。
3282H	(s)に特殊な数を設定したとき。 • 設定した倍精度実数が下記範囲内でない。 $0, 2^{-1022} \leq (s) < 2^{1024}$ • 設定したデバイス，ラベルの内容が-0，非正規化数，非数，$\pm\infty$のとき。

倍精度実数→符号なしBIN32ビットデータ

DBL2UDINT(P)

指定した倍精度実数を符号なしBIN32ビットデータに変換します。

ラダー	ST
	ENO:=DBL2UDINT(EN,s,d); ENO:=DBL2UDINTP(EN,s,d);

FBD/LD


■実行条件

命令	実行条件
DBL2UDINT	
DBL2UDINTP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	倍精度実数または、倍精度実数が格納されている先頭デバイス	0~4294967295	倍精度実数	ANYREAL_64
(d)	BINデータを格納する先頭デバイス	—	符号なしBIN32ビット	ANY32_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	—	—
(d)	○	—	○	—	○	○	○	○	—	—	—

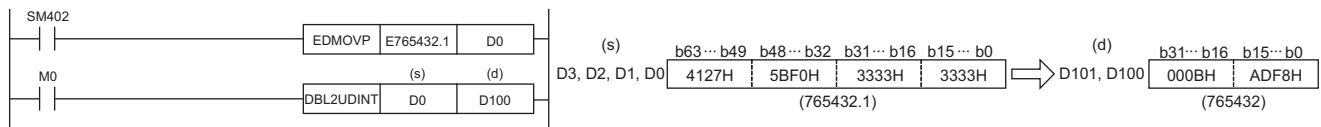
機能

- (s)で指定された倍精度実数を，符号なしBIN32ビットデータに変換して，(d)で指定されたデバイスに格納します。
- 変換後のデータは，(s)で指定された倍精度実数の小数点以下1桁目を四捨五入した値となります。
- エンジニアリングツールから入力値を設定する場合，丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は，下記を参照してください。

📖 39ページ 注意事項

例

M0がONしたとき，D0~D3に格納された倍精度実数を，符号なしBIN32ビットデータに変換して，D100~D101へ格納します。



エラー

エラーコード (SD0)	内容
3281H	(s)で設定する倍精度実数が0~4294967295の範囲外の時。
3282H	(s)に特殊な数を設定したとき。 • 設定した倍精度実数が下記範囲内でない。 $0, 2^{-1022} \leq (s) < 2^{1024}$ • 設定したデバイス，ラベルの内容が-0，非正規化数，非数，$\pm\infty$のとき。

符号付きBIN16ビットデータ→符号なしBIN16ビットデータ変換

INT2UINT(P)

指定した符号付きBIN16ビットデータを符号なしBIN16ビットデータに変換します。

ラダー	ST
	ENO:=INT2UINT(EN,s,d); ENO:=INT2UINTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
INT2UINT	
INT2UINTP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータまたは、BINデータが格納されているラベル	-32768~32767	符号付きBIN16ビット	ANY16_S
(d)	BINデータを格納するラベル	—	符号なしBIN16ビット	ANY16_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数				その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$		
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

Point

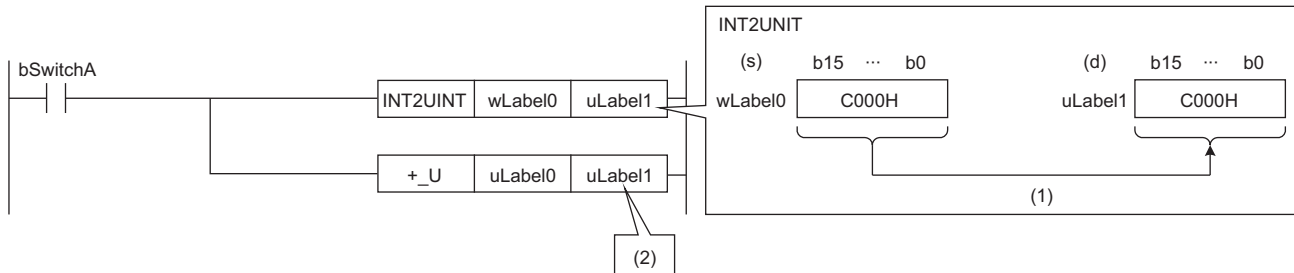
INT2UINT(P)命令はラベルでプログラミングする場合に、命令のオペランドに指定可能なデータ型と、指定されたラベルのデータ型を一致させるために使用します。
 デバイスでプログラミングする場合は、INT2UINT(P)命令を使用する必要はありません。

機能

- (s)で指定した符号付きBIN16ビットデータ(ANY16_S)を, 符号なしBIN16ビットデータ(ANY16_U)に変換して, (d)で指定したラベルに格納します。
- 下記にINT2UINT(P)命令の使用例を示します。

例

+_U命令はオペランドにANY16_Uを指定する必要があるため, +_U命令を実行する前にINT2UINT命令を使用して, ANY16_SのwLabel0をANY16_UのuLabel1へ変換します。
値としてはwLabel0をそのままuLabel1へ格納します。



bSwitchA: ビット

wLabel0: ワード[符号付き]

uLabel0, uLabel1: ワード[符号なし]/ビット列[16ビット]

(1): そのまま格納します。

(2): +_U命令のオペランドのデータ型に合わせてから演算します。

エラー

演算エラーはありません。

符号付きBIN16ビットデータ→符号付きBIN32ビットデータ変換

INT2DINT(P)

指定した符号付きBIN16ビットデータを符号付きBIN32ビットデータに変換します。

ラダー	ST
	ENO:=INT2DINT(EN,s,d); ENO:=INT2DINTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
INT2DINT	
INT2DINTP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータまたは，BINデータが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16_S
(d)	BINデータを格納する先頭デバイス	—	符号付きBIN32ビット	ANY32_S
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

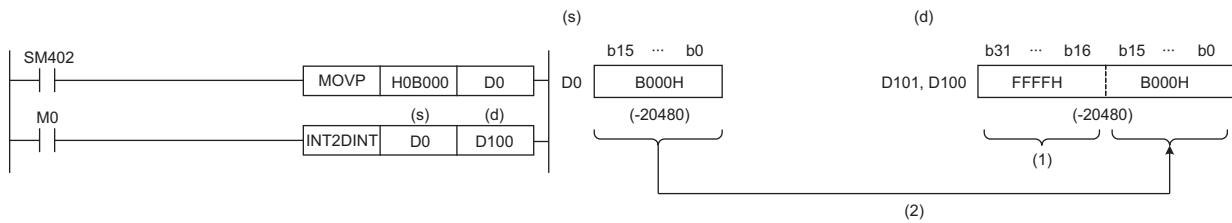
オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定された符号付きBIN16ビットデータを，符号付きBIN32ビットデータに変換して，(d)で指定されたデバイスに格納します。

例

M0がONしたとき，D0に格納された符号付きBIN16ビットデータを，符号付きBIN32ビットデータに変換して，D100~D101へ格納します。



(1): 変換前の最上位ビットの値で埋めます。

(2): 下位16ビットへ格納

エラー

演算エラーはありません。

符号付きBIN16ビットデータ→符号なしBIN32ビットデータ変換

INT2UDINT(P)

指定した符号付きBIN16ビットデータを符号なしBIN32ビットデータに変換します。

ラダー	ST
	ENO:=INT2UDINT(EN,s,d); ENO:=INT2UDINTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
INT2UDINT	
INT2UDINTP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータまたは, BINデータが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16_S
(d)	BINデータを格納する先頭デバイス	—	符号なしBIN32ビット	ANY32_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

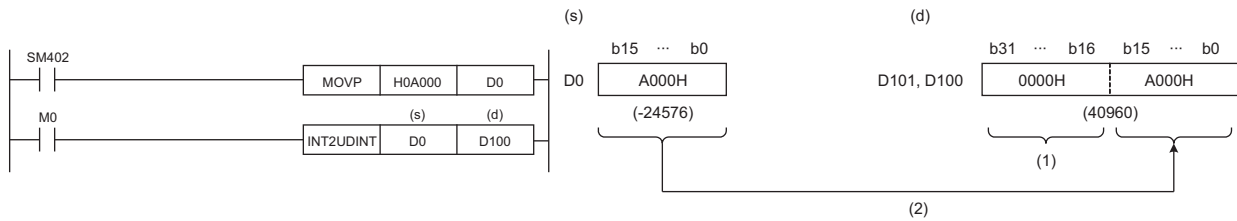
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定された符号付きBIN16ビットデータを，符号なしBIN32ビットデータに変換して，(d)で指定されたデバイスに格納します。

例

M0がONしたとき，D0に格納された符号付きBIN16ビットデータを，符号なしBIN32ビットデータに変換して，D100~D101へ格納します。



(1): 0を格納

(2): 下位16ビットへ格納

エラー

演算エラーはありません。

符号なしBIN16ビットデータ→符号付きBIN16ビットデータ変換

UINT2INT(P)

指定した符号なしBIN16ビットデータを符号付きBIN16ビットデータに変換します。

ラダー	ST
	ENO:=UINT2INT(EN,s,d); ENO:=UINT2INTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
UINT2INT	
UINT2INTP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータまたは、BINデータが格納されているラベル	0~65535	符号なしBIN16ビット	ANY16_U
(d)	BINデータを格納するラベル	—	符号付きBIN16ビット	ANY16_S
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口		LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

Point

UINT2INT(P)命令はラベルでプログラミングする場合に、命令のオペランドに指定可能なデータ型と、指定されたラベルのデータ型を一致させるために使用します。

デバイスでプログラミングする場合は、UINT2INT(P)命令を使用する必要はありません。

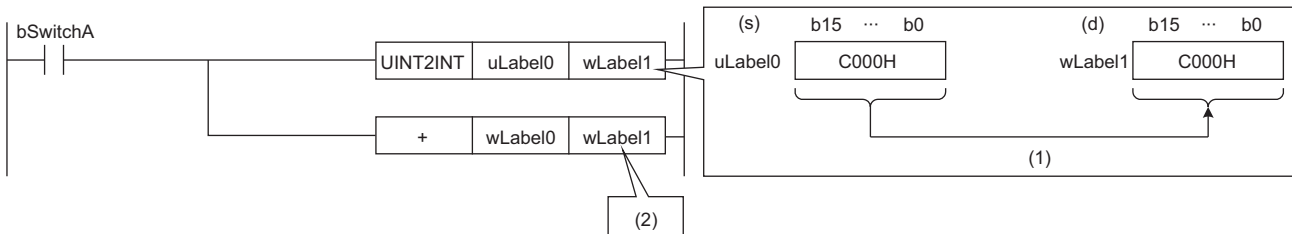
機能

- (s)で指定した符号なしBIN16ビットデータ(ANY16_U)を, 符号付きBIN16ビットデータ(ANY16_S)に変換して, (d)で指定したラベルに格納します。
- 下記にUINT2INT(P)命令の使用例を示します。

例

+命令はオペランドにANY16_Sを指定する必要があるため, +命令を実行する前にUINT2INT命令を使用して, ANY16_UのuLabel0をANY16_SのwLabel1へ変換します。

値としてはuLabel0をそのままwLabel1へ格納します。



bSwitchA: ビット

wLabel0, wLabel1: ワード[符号付き]

uLabel0: ワード[符号なし]/ビット列[16ビット]

(1): そのまま格納します。

(2): +命令のオペランドのデータ型に合わせてから演算します。

エラー

演算エラーはありません。

符号なしBIN16ビットデータ→符号付きBIN32ビットデータ変換

UINT2DINT(P)

指定した符号なしBIN16ビットデータを符号付きBIN32ビットデータに変換します。

ラダー	ST
	ENO:=UINT2DINT(EN,s,d); ENO:=UINT2DINTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
UINT2DINT	
UINT2DINTP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータまたは，BINデータが格納されているデバイス	0~65535	符号なしBIN16ビット	ANY16_U
(d)	BINデータを格納する先頭デバイス	—	符号付きBIN32ビット	ANY32_S
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

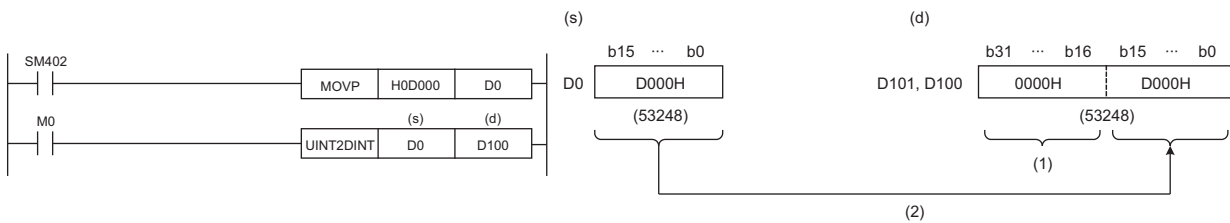
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定された符号なしBIN16ビットデータを，符号付きBIN32ビットデータに変換して，(d)で指定されたデバイスに格納します。

例

M0がONしたとき，D0に格納された符号なしBIN16ビットデータを，符号付きBIN32ビットデータに変換して，D100~D101へ格納します。



(1): 0を格納

(2): 下位16ビットへ格納

エラー

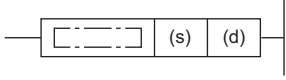
演算エラーはありません。

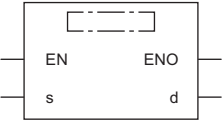
符号なしBIN16ビットデータ→符号なしBIN32ビットデータ変換

12

UINT2UDINT(P)

指定した符号なしBIN16ビットデータを符号なしBIN32ビットデータに変換します。

ラダー	ST
	ENO:=UINT2UDINT(EN,s,d); ENO:=UINT2UDINTP(EN,s,d);

FBD/LD


■実行条件

命令	実行条件
UINT2UDINT	
UINT2UDINTP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータまたは, BINデータが格納されているデバイス	0~65535	符号なしBIN16ビット	ANY16_U
(d)	BINデータを格納する先頭デバイス	—	符号なしBIN32ビット	ANY32_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

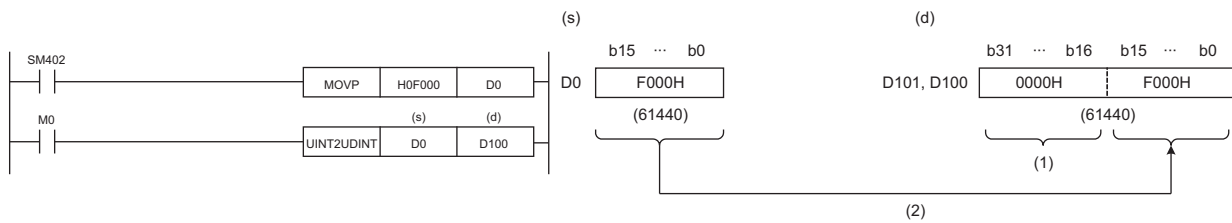
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定された符号なしBIN16ビットデータを，符号なしBIN32ビットデータに変換して，(d)で指定されたデバイスに格納します。

例

M0がONしたとき，D0に格納された符号なしBIN16ビットデータを，符号なしBIN32ビットデータに変換して，D100~D101へ格納します。



(1): 0を格納

(2): 下位16ビットへ格納

エラー

演算エラーはありません。

符号付きBIN32ビットデータ→符号付きBIN16ビットデータ変換

DINT2INT(P)

指定した符号付きBIN32ビットデータを符号付きBIN16ビットデータに変換します。

ラダー	ST
	<pre>ENO:=DINT2INT(EN,s,d); ENO:=DINT2INTP(EN,s,d);</pre>

FBD/LD

■実行条件

命令	実行条件
DINT2INT	
DINT2INTP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータまたは、BINデータが格納されている先頭デバイス	-32768~32767	符号付きBIN32ビット	ANY32_S
(d)	BINデータを格納するデバイス	—	符号付きBIN16ビット	ANY16_S
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

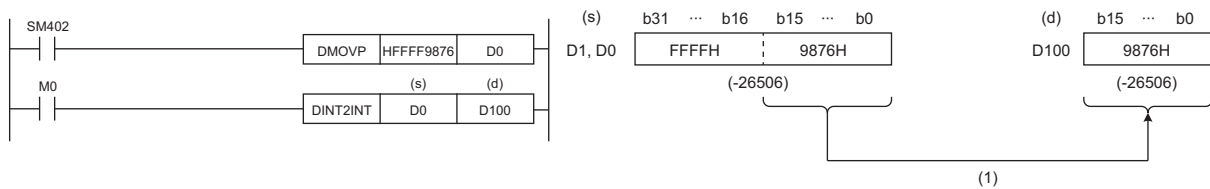
オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□		LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- (s)で指定された符号付きBIN32ビットデータを，符号付きBIN16ビットデータに変換して，(d)で指定されたデバイスに格納します。

例

M0がONしたとき，D0～D1に格納された符号付きBIN32ビットデータを，符号付きBIN16ビットデータに変換して，D100へ格納します。



(1): 下位16ビットへ格納

エラー

エラーコード (SD0)	内容
3281H	(s)で設定された符号付きBIN32ビットデータが-32768～32767の範囲外の時。

符号付きBIN32ビットデータ→符号なしBIN16ビットデータ変換

DINT2UINT(P)

指定した符号付きBIN32ビットデータを符号なしBIN16ビットデータに変換します。

ラダー	ST
	ENO:=DINT2UINT(EN,s,d); ENO:=DINT2UINTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
DINT2UINT	
DINT2UINTP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータまたは、BINデータが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
(d)	BINデータを格納するデバイス	—	符号なしBIN16ビット	ANY16_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

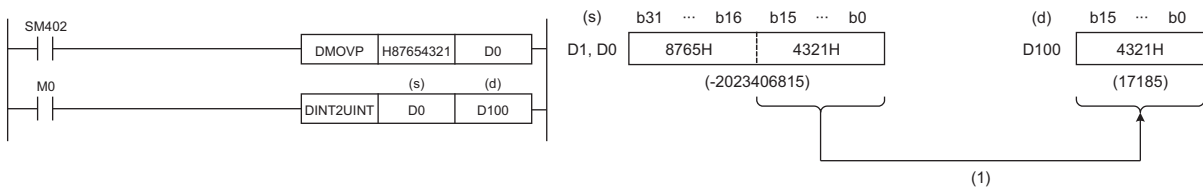
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- (s)で指定された符号付きBIN32ビットデータを，符号なしBIN16ビットデータに変換して，(d)で指定されたデバイスに格納します。

例

M0がONしたとき，D0～D1に格納された符号付きBIN32ビットデータを，符号なしBIN16ビットデータに変換して，D100へ格納します。



(1): 下位16ビットへ格納

エラー

演算エラーはありません。

符号付きBIN32ビットデータ→符号なしBIN32ビットデータ変換

DINT2UDINT(P)

指定した符号付きBIN32ビットデータを符号なしBIN32ビットデータに変換します。

ラダー	ST
	ENO:=DINT2UDINT(EN,s,d); ENO:=DINT2UDINTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
DINT2UDINT	
DINT2UDINTP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータまたは、BINデータが格納されているラベル	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
(d)	BINデータを格納するラベル	—	符号なしBIN32ビット	ANY32_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

Point

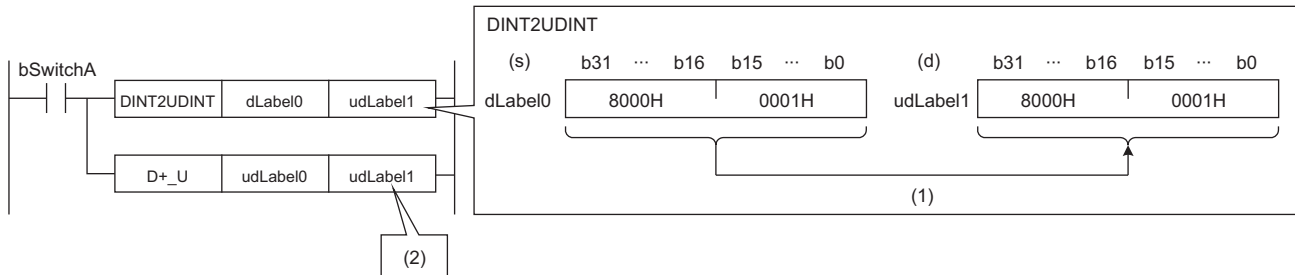
DINT2UDINT(P)命令はラベルでプログラミングする場合に、命令のオペランドに指定可能なデータ型と、指定されたラベルのデータ型を一致させるために使用します。
デバイスでプログラミングする場合は、DINT2UDINT(P)命令を使用する必要はありません。

機能

- (s)で指定した符号付きBIN32ビットデータ(ANY32_S)を, 符号なしBIN32ビットデータ(ANY32_U)に変換して, (d)で指定したラベルに格納します。
- 下記にDINT2UDINT(P)命令の使用例を示します。

例

D+_U命令はオペランドにANY32_Uを指定する必要があるため, D+_U命令を実行する前にDINT2UDINT命令を使用して, ANY32_SのdLabel0をANY32_UのudLabel1へ変換します。
値としてはdLabel0をそのままudLabel1へ格納します。



bSwitchA: ビット

dLabel0: ダブルワード[符号付き]

udLabel0, udLabel1: ダブルワード[符号なし]/ビット列[32ビット]

(1): そのまま格納します。

(2): D+_U命令のオペランドのデータ型に合わせてから演算します。

エラー

演算エラーはありません。

符号なしBIN32ビットデータ→符号付きBIN16ビットデータ変換

UDINT2INT(P)

指定した符号なしBIN32ビットデータを符号付きBIN16ビットデータに変換します。

ラダー	ST
	ENO:=UDINT2INT(EN,s,d); ENO:=UDINT2INTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
UDINT2INT	
UDINT2INTP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータまたは、BINデータが格納されている先頭デバイス	0~4294967295	符号なしBIN32ビット	ANY32_U
(d)	BINデータを格納するデバイス	—	符号付きBIN16ビット	ANY16_S
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—

機能

- 例

1)

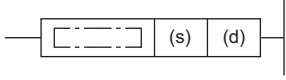
12 基本命令

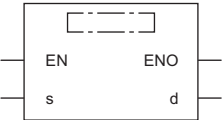
12.6 データ変換命令

符号なしBIN32ビットデータ→符号なしBIN16ビットデータ変換

UDINT2UINT(P)

指定した符号なしBIN32ビットデータを符号なしBIN16ビットデータに変換します。

ラダー	ST
	ENO:=UDINT2UINT(EN,s,d); ENO:=UDINT2UINTP(EN,s,d);

FBD/LD


■実行条件

命令	実行条件
UDINT2UINT	
UDINT2UINTP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータまたは, BINデータが格納されている先頭デバイス	0~4294967295	符号なしBIN32ビット	ANY32_U
(d)	BINデータを格納するデバイス	—	符号なしBIN16ビット	ANY16_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

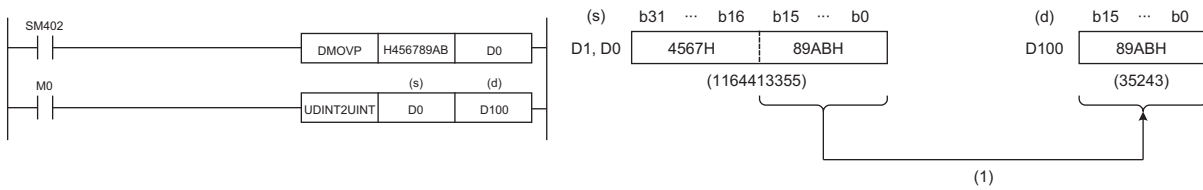
オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	○	—	—	—	—

機能

- (s)で指定された符号なしBIN32ビットデータを，符号なしBIN16ビットデータに変換して，(d)で指定されたデバイスに格納します。

例

M0がONしたとき，D0～D1に格納された符号なしBIN32ビットデータを，符号なしBIN16ビットデータに変換して，D100へ格納します。



(1): 下位16ビットへ格納

エラー

演算エラーはありません。

符号なしBIN32ビットデータ→符号付きBIN32ビットデータ変換

UDINT2DINT(P)

指定した符号なしBIN32ビットデータを符号付きBIN32ビットデータに変換します。

ラダー	ST
	ENO:=UDINT2DINT(EN,s,d); ENO:=UDINT2DINTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
UDINT2DINT	
UDINT2DINTP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータまたは、BINデータが格納されているラベル	0~4294967295	符号なしBIN32ビット	ANY32_U
(d)	BINデータを格納するラベル	—	符号付きBIN32ビット	ANY32_S
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	—	—	—	—

Point

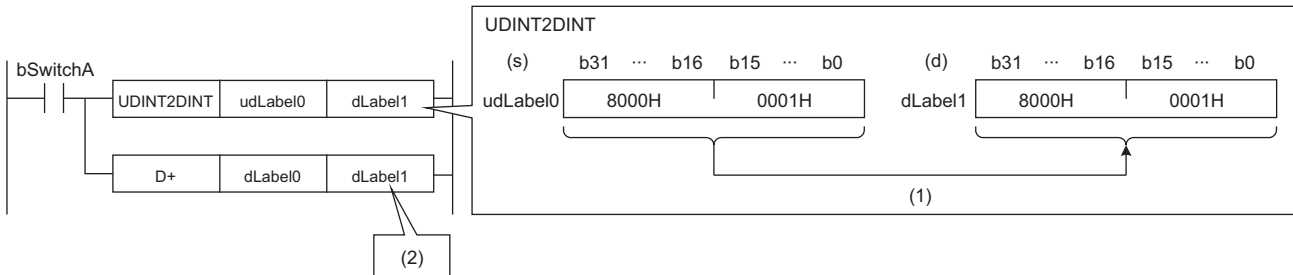
UDINT2DINT(P)命令はラベルでプログラミングする場合に、命令のオペランドに指定可能なデータ型と、指定されたラベルのデータ型を一致させるために使用します。
 デバイスでプログラミングする場合は、UDINT2DINT(P)命令を使用する必要はありません。

機能

- (s)で指定した符号なしBIN32ビットデータ(ANY32_U)を, 符号付きBIN32ビットデータ(ANY32_S)に変換して, (d)で指定したラベルに格納します。
- 下記にUDINT2DINT(P)命令の使用例を示します。

例

D+命令はオペランドにANY32_Sを指定する必要があるため, D+命令を実行する前にUDINT2DINT命令を使用して, ANY32_UのudLabel0をANY32_SのdLabel1へ変換します。
値としてはudLabel0をそのままdLabel1へ格納します。



bSwitchA: ビット

dLabel0, dLabel1: ダブルワード[符号付き]

udLabel0: ダブルワード[符号なし]/ビット列[32ビット]

(1): そのまま格納します。

(2): D+命令のオペランドのデータ型に合わせてから演算します。

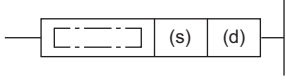
エラー

演算エラーはありません。

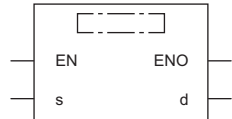
BIN16ビットデータ→グレイコードデータ変換

GRY(P)(_U)

指定したBIN16ビットデータをBIN16ビットグレイコードデータに変換します。

ラダー	ST
	ENO:=GRY(EN,s,d); ENO:=GRYP(EN,s,d);
	ENO:=GRY_U(EN,s,d); ENO:=GRYP_U(EN,s,d);

FBD/LD



■実行条件

命令	実行条件
GRY GRY_U	
GRYP GRYP_U	

設定データ

■内容、範囲、データ型

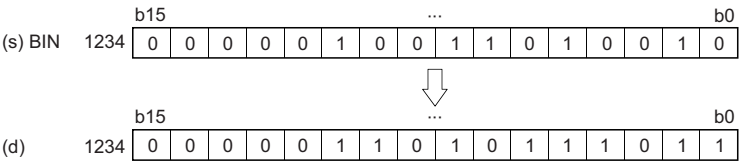
オペランド		内容	範囲	データ型	データ型(ラベル)
(s)	GRY(P)	BINデータまたは、BINデータが格納されているデバイス	0~32767	符号付きBIN16ビット	ANY16_S
	GRY(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
(d)	GRY(P)	変換後のグレイコードデータを格納するデバイス	—	符号付きBIN16ビット	ANY16_S
	GRY(P)_U			符号なしBIN16ビット	ANY16_U
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- (s)で指定されたBIN16ビットデータをBIN16ビットグレイコードデータに変換して、(d)で指定されたデバイスに格納します。



エラー

エラーコード (SD0)	内容
3281H	GRY(P)命令使用時、(s)で指定した値が0~32767以外するとき。

BIN32ビットデータ→グレイコードデータ変換

DGRY(P)(_U)

指定したBIN32ビットデータをBIN32ビットグレイコードデータに変換します。

ラダー	ST
	ENO:=DGRY(EN,s,d); ENO:=DGRYP(EN,s,d);
	ENO:=DGRY_U(EN,s,d); ENO:=DGRYP_U(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
DGRY DGRY_U	
DGRYP DGRYP_U	

設定データ

■内容, 範囲, データ型

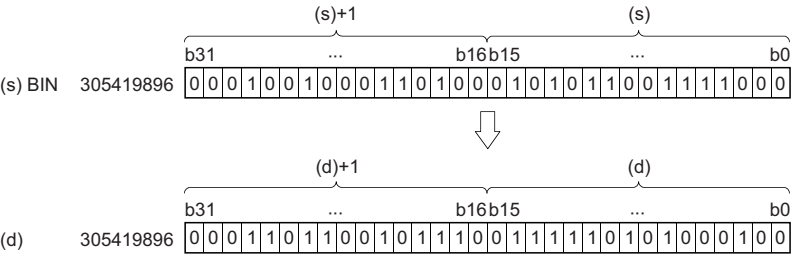
オペランド		内容	範囲	データ型	データ型(ラベル)
(s)	DGRY(P)	BINデータまたは、BINデータが格納されている先頭デバイス	0~2147483647	符号付きBIN32ビット	ANY32_S
	DGRY(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U
(d)	DGRY(P)	変換後のグレイコードデータを格納する先頭デバイス	—	符号付きBIN32ビット	ANY32_S
	DGRY(P)_U			符号なしBIN32ビット	ANY32_U
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定されたBIN32ビットデータをBIN32ビットグレイコードデータに変換して、(d)で指定されたデバイスに格納します。



(s)+1: 上位16ビット
(s): 下位16ビット

エラー

エラーコード (SD0)	内容
3281H	DGRY(P)命令使用時、(s)で指定した値が0~2147483647以外るとき。

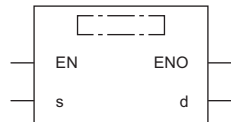
BIN16ビットグレイコードデータ→BIN16ビットデータ変換

GBIN(P)(_U)

指定したBIN16ビットグレイコードデータをBIN16ビットデータに変換します。

ラダー	ST
	ENO:=GBIN(EN,s,d); ENO:=GBINP(EN,s,d);
	ENO:=GBIN_U(EN,s,d); ENO:=GBINP_U(EN,s,d);

FBD/LD



■実行条件

命令	実行条件
GBIN GBIN_U	
GBINP GBINP_U	

設定データ

■内容，範囲，データ型

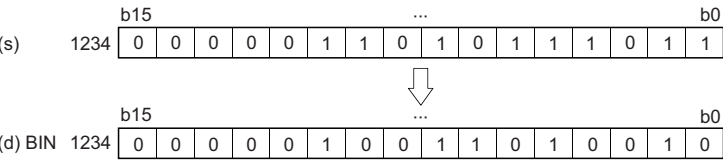
オペランド		内容	範囲	データ型	データ型(ラベル)
(s)	GBIN(P)	グレイコードデータ，またはグレイコードデータが格納されているデバイス	0~32767	符号付きBIN16ビット	ANY16_S
	GBIN(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
(d)	GBIN(P)	変換後のBINデータを格納するデバイス	—	符号付きBIN16ビット	ANY16_S
	GBIN(P)_U			符号なしBIN16ビット	ANY16_U
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

機能

- (s)で指定されたデバイスに格納されているBIN16ビットグレイコードデータを，BIN16ビットデータに変換して，(d)で指定されたデバイスに格納します。



エラー

エラーコード (SD0)	内容
3281H	GBIN(P)命令使用時，(s)で指定した値が0~32767以外るとき。

BIN32ビットグレイコードデータ→BIN32ビットデータ変換

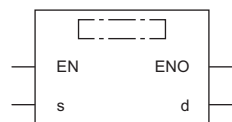
12

DGBIN(P)(_U)

指定したBIN32ビットグレイコードデータをBIN32ビットデータに変換します。

ラダー	ST
	ENO:=DGBIN(EN,s,d); ENO:=DGBINP(EN,s,d);
	ENO:=DGBIN_U(EN,s,d); ENO:=DGBINP_U(EN,s,d);

FBD/LD



■実行条件

命令	実行条件
DGBIN DGBIN_U	
DGBINP DGBINP_U	

設定データ

■内容，範囲，データ型

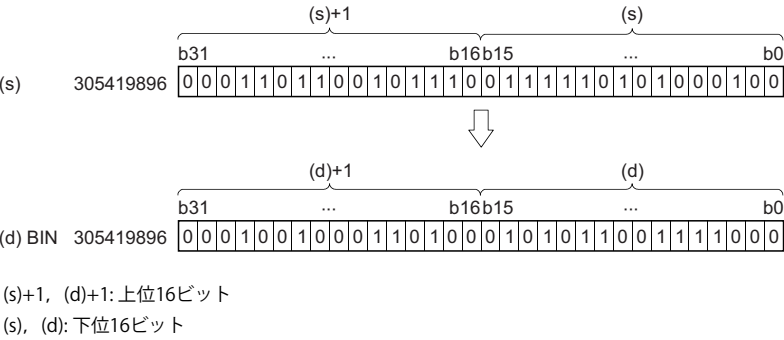
オペランド		内容	範囲	データ型	データ型(ラベル)
(s)	DGBIN(P)	グレイコードデータ，またはグレイコードデータが格納されている先頭デバイス	0~2147483647	符号付きBIN32ビット	ANY32_S
	DGBIN(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U
(d)	DGBIN(P)	変換後のBINデータを格納する先頭デバイス	—	符号付きBIN32ビット	ANY32_S
	DGBIN(P)_U			符号なしBIN32ビット	ANY32_U
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ	K, H		E	\$		
(s)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	○	○	—	—	—

機能

- (s)で指定されたデバイスに格納されているBIN32ビットグレイコードデータを，BIN32ビットデータに変換して，(d)で指定されたデバイスに格納します。



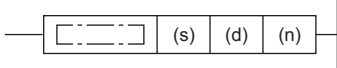
エラー

エラーコード (SD0)	内容
3281H	DGBIN(P)命令使用時，(s)で指定した値が0~2147483647以外るとき。

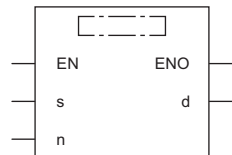
BIN16ビットデータブロック→BCD4桁データ変換ブロック

BKBCD(P)

指定したデバイスから(n)点のBINデータ(0~9999)をBCD変換します。

ラダー	ST
	ENO:=BKBCD(EN,s,n,d); ENO:=BKBCDP(EN,s,n,d);

FBD/LD



■実行条件

命令	実行条件
BKBCD	
BKBCDP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(d)	変換後のBCDデータを格納する先頭デバイス	—	BCD4桁	ANY16 ^{*1}
(n)	変数データ数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

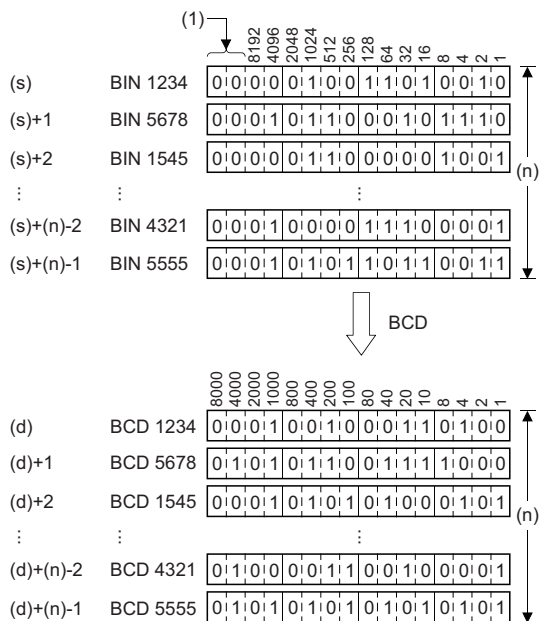
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- (s)で指定したデバイスから、(n)点分のBIN16ビットデータ(0~9999)をBCD変換して、(d)で指定したデバイス以降に格納します。



(1): 必ず0にしてください。

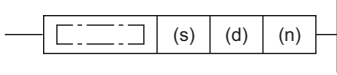
エラー

エラーコード (SD0)	内容
2821H	(s), (d)のデバイスが重複しているとき。
3281H	(s)のデバイスから(n)点分のデータが0~9999以外するとき。

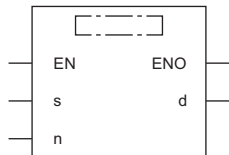
ブロックBCD4桁データ→ブロックBIN16ビット変換データ

BKBIN(P)

指定したデバイスから(n)点のBCDデータ(0~9999)をBIN変換します。

ラダー	ST
	ENO:=BKBIN(EN,s,n,d); ENO:=BKBINP(EN,s,n,d);

FBD/LD



■実行条件

命令	実行条件
BKBIN	
BKBINP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BCDデータが格納されている先頭デバイス	—	BCD4桁	ANY16 ^{*1}
(d)	変換後のBINデータを格納する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(n)	変数データ数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

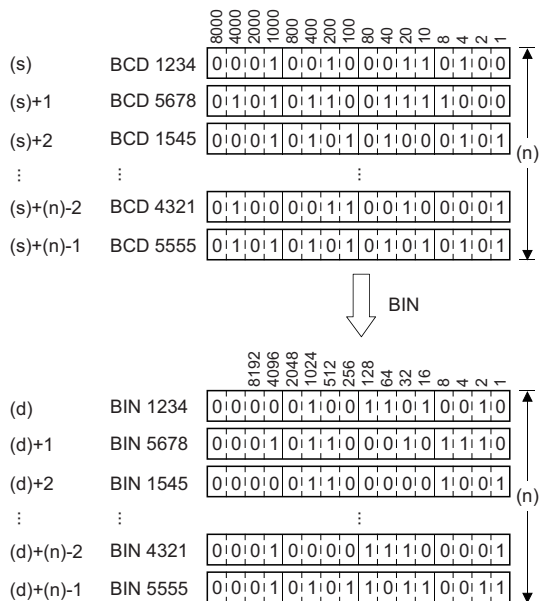
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- (s)で指定されたデバイスから、(n)点分のBCDデータ(0~9999)をBIN16ビット変換して、(d)で指定されたデバイス以降に格納します。



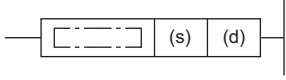
エラー

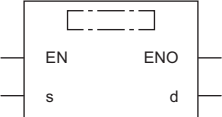
エラーコード (SD0)	内容
2821H	(s), (d)のデバイスが重複しているとき。
3281H	(s)のデバイスから(n)点分のデータが0~9999以外するとき。

10進アスキーデータ→BIN16ビットデータ変換

DABIN(P)(_U)

10進アスキーデータを、BIN16ビットデータに変換します。

ラダー	ST
	ENO:=DABIN(EN,s,d); ENO:=DABINP(EN,s,d);
	ENO:=DABIN_U(EN,s,d); ENO:=DABINP_U(EN,s,d);

FBD/LD


■実行条件

命令	実行条件
DABIN DABIN_U	
DABINP DABINP_U	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BIN値に変換するアスキーデータ、またはアスキーデータが格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	DABIN(P) DABIN(P)_U	変換結果を格納するデバイス	符号付きBIN16ビット 符号なしBIN16ビット	ANY16_S ANY16_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

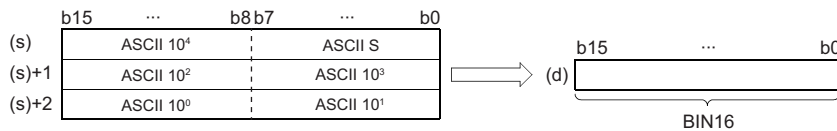
機能

- (s)で指定されたデバイス番号以降に格納されている10進アスキーデータを、BIN16ビットデータに変換して、(d)で指定されたデバイスに格納します。
- SM705(変換桁数切換)の状態によって、(s)に設定する10進アスキーデータの設定方法が異なります。

SM705の状態	(s)の設定方法	参照先
OFF	固定桁数(符号+数値部5桁)で設定	448ページ SM705(変換桁数切換)がOFFの場合の(s)の設定方法
ON	任意の桁数(最大: 符号+数値部5桁)で設定	449ページ SM705(変換桁数切換)がONの場合の(s)の設定方法

■SM705(変換桁数切換)がOFFの場合の(s)の設定方法

(s)~(s)+2に、10進アスキーデータを固定桁数で設定します。



ASCII S: 符号のアスキーコード

ASCII 10⁴: 万の位のアスキーコード

ASCII 10³: 千の位のアスキーコード

ASCII 10²: 百の位のアスキーコード

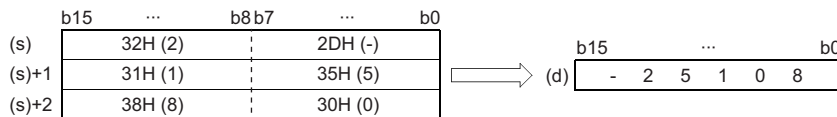
ASCII 10¹: 十の位のアスキーコード

ASCII 10⁰: 一の位のアスキーコード

- (s)~(s)+2で指定するアスキーデータは、DABIN(P)命令の場合は-32768~32767、DABIN(P)_U命令の場合は0~65535の範囲内です。
- (s)+3以降のデータは無視されます。
- 符号データには、変換するデータが正のとき20H(スペース)を設定し、負のとき2DH(-)を設定します。(20H, 2DH以外を設定した場合は、正のデータとして処理します。)
- 各位に設定するアスキーコードは、30H~39Hの範囲内です。
- 各位に設定するアスキーコードが20Hまたは00Hのときは、30Hとして処理します。

例

DABIN(P)命令において、(s)に“-25108”を設定した場合



■SM705(変換桁数切換)がONの場合の(s)の設定方法

(s)には、任意の桁数(00H(NULLコード)まで)の10進アスキーデータを設定します。ただし、整数部が最大桁数(5桁)の場合は、00H(NULLコード)の設定は不要です。

(s)の設定方法を下記に示します。

(s)に設定する値	(s)~(s)+2の内容	(s)に設定する値	(s)~(s)+2の内容																																																
0 - 正の数(数値部1桁)	(s)の上位バイトに00Hを設定します。 (s)+1以降は無視されます。 <table><tr><td></td><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(s)</td><td colspan="2">00H</td><td>:</td><td colspan="2">ASCII 10⁰</td></tr><tr><td>(s)+1</td><td colspan="5"></td></tr><tr><td>(s)+2</td><td colspan="5"></td></tr></table>		b15	...	b8 b7	...	b0	(s)	00H		:	ASCII 10 ⁰		(s)+1						(s)+2						- 正の数(数値部2桁) - 負の数(数値部1桁)	(s)+1の下位バイトに00Hを設定します。 (s)+1の上位バイト以降は無視されます。 <table><tr><td></td><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(s)</td><td colspan="2">ASCII 10⁰</td><td>:</td><td colspan="2">ASCII 10¹ / 2DH (-)</td></tr><tr><td>(s)+1</td><td colspan="2"></td><td>:</td><td colspan="2">00H</td></tr><tr><td>(s)+2</td><td colspan="5"></td></tr></table>		b15	...	b8 b7	...	b0	(s)	ASCII 10 ⁰		:	ASCII 10 ¹ / 2DH (-)		(s)+1			:	00H		(s)+2					
	b15	...	b8 b7	...	b0																																														
(s)	00H		:	ASCII 10 ⁰																																															
(s)+1																																																			
(s)+2																																																			
	b15	...	b8 b7	...	b0																																														
(s)	ASCII 10 ⁰		:	ASCII 10 ¹ / 2DH (-)																																															
(s)+1			:	00H																																															
(s)+2																																																			
⋮																																																			
負の数(数値部4桁)	(s)+2の上位バイトに00Hを設定します。 <table><tr><td></td><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(s)</td><td colspan="2">ASCII 10³</td><td>:</td><td colspan="2">2DH (-)</td></tr><tr><td>(s)+1</td><td colspan="2">ASCII 10¹</td><td>:</td><td colspan="2">ASCII 10²</td></tr><tr><td>(s)+2</td><td colspan="2">00H</td><td>:</td><td colspan="2">ASCII 10⁰</td></tr></table>		b15	...	b8 b7	...	b0	(s)	ASCII 10 ³		:	2DH (-)		(s)+1	ASCII 10 ¹		:	ASCII 10 ²		(s)+2	00H		:	ASCII 10 ⁰		正の数(数値部5桁)	(s)+2の上位バイト以降は無視されます。最大桁数のため、00Hの設定は不要です。 <table><tr><td></td><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(s)</td><td colspan="2">ASCII 10³</td><td>:</td><td colspan="2">ASCII 10⁴</td></tr><tr><td>(s)+1</td><td colspan="2">ASCII 10¹</td><td>:</td><td colspan="2">ASCII 10²</td></tr><tr><td>(s)+2</td><td colspan="2"></td><td>:</td><td colspan="2">ASCII 10⁰</td></tr></table>		b15	...	b8 b7	...	b0	(s)	ASCII 10 ³		:	ASCII 10 ⁴		(s)+1	ASCII 10 ¹		:	ASCII 10 ²		(s)+2			:	ASCII 10 ⁰	
	b15	...	b8 b7	...	b0																																														
(s)	ASCII 10 ³		:	2DH (-)																																															
(s)+1	ASCII 10 ¹		:	ASCII 10 ²																																															
(s)+2	00H		:	ASCII 10 ⁰																																															
	b15	...	b8 b7	...	b0																																														
(s)	ASCII 10 ³		:	ASCII 10 ⁴																																															
(s)+1	ASCII 10 ¹		:	ASCII 10 ²																																															
(s)+2			:	ASCII 10 ⁰																																															
負の数(数値部5桁)	(s)+3以降は無視されます。最大桁数のため、00Hの設定は不要です。 <table><tr><td></td><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(s)</td><td colspan="2">ASCII 10⁴</td><td>:</td><td colspan="2">2DH (-)</td></tr><tr><td>(s)+1</td><td colspan="2">ASCII 10²</td><td>:</td><td colspan="2">ASCII 10³</td></tr><tr><td>(s)+2</td><td colspan="2">ASCII 10⁰</td><td>:</td><td colspan="2">ASCII 10¹</td></tr></table>		b15	...	b8 b7	...	b0	(s)	ASCII 10 ⁴		:	2DH (-)		(s)+1	ASCII 10 ²		:	ASCII 10 ³		(s)+2	ASCII 10 ⁰		:	ASCII 10 ¹		ASCII 10 ⁰ : 一の位のアスキーコード ASCII 10 ¹ : 十の位のアスキーコード ⋮ ASCII 10 ⁴ : 万の位のアスキーコード																									
	b15	...	b8 b7	...	b0																																														
(s)	ASCII 10 ⁴		:	2DH (-)																																															
(s)+1	ASCII 10 ²		:	ASCII 10 ³																																															
(s)+2	ASCII 10 ⁰		:	ASCII 10 ¹																																															

- (s)~(s)+2で指定するアスキーデータは、DABIN(P)命令の場合は-32768~32767、DABIN(P)_U命令の場合は0~65535の範囲内です。
- 変換するデータが負の数の場合は、(s)+0の下位バイトに符号データとして2DH(-)を設定します。変換するデータが0または正の数の場合は符号データは設定せず、最上位桁のアスキーコードを設定します。
- 各位に設定するアスキーコードは、30H~39Hの範囲内です。
- 正の数で数値部5桁の場合、(s)+2の上位バイト以降のデータは無視されます。負の数で数値部5桁の場合、(s)+3以降のデータは無視されます。
- 各位に設定するアスキーコードが20Hのときは、30Hとして処理されます。00Hのときは、10進アスキーデータの終端として処理されます。
- 下記の場合、(d)には0が格納されます。
 - 1文字目が00H(NULL)
 - 1文字目が2DH(-)で、2文字目が00H(NULL)

エラー

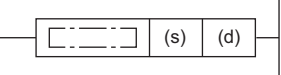
エラーコード (SD0)	内容
3281H	(s)~(s)+2に変換できない不正なデータを入力したとき。 1文字目のアスキーコードが、2DH, 30H~39H, 20H, 00H以外。^{*1} 2文字目以降のアスキーコードが、30H~39H, 20H, 00H以外。 - DABIN(P)命令使用時、アスキーデータが-32768~32767以外。 - DABIN(P)_U命令使用時、アスキーデータが0~65535以外。

^{*1} SM705(変換桁数切換)がOFFの場合は、1文字目のアスキーコードに何を設定してもエラーになりません。

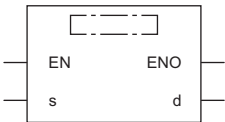
10進アスキーデータ→BIN32ビットデータ変換

DDABIN(P)(_U)

10進アスキーデータを，BIN32ビットデータに変換します。

ラダー	ST	
	ENO:=DDABIN(EN,s,d); ENO:=DDABINP(EN,s,d);	ENO:=DDABIN_U(EN,s,d); ENO:=DDABINP_U(EN,s,d);

FBD/LD



■実行条件

命令	実行条件
DDABIN DDABIN_U	
DDABINP DDABINP_U	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BIN値に変換するアスキーデータ，またはアスキーデータが格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	変換結果を格納する先頭デバイス	—	符号付きBIN32ビット	ANY32_S
			符号なしBIN32ビット	ANY32_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	—	—	○	—	—	—	○	—	—	○	—
(d)	○	○	○	○	○	○	○	—	—	—	—

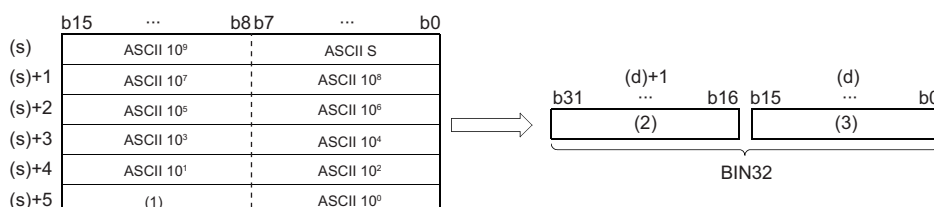
機能

- (s)で指定されたデバイス番号以降に格納されている10進アスキーデータを、BIN32ビットデータに変換して、(d)で指定されたデバイス番号に格納します。
- SM705(変換桁数切換)の状態によって、(s)に設定する10進アスキーデータの設定方法が異なります。

SM705の状態	(s)の設定方法	参照先
OFF	固定桁数(符号+数値部10桁)で設定	451ページ SM705(変換桁数切換)がOFFの場合の(s)の設定方法
ON	任意の桁数(最大: 符号+数値部10桁)で設定	452ページ SM705(変換桁数切換)がONの場合の(s)の設定方法

■SM705(変換桁数切換)がOFFの場合の(s)の設定方法

(s)~(s)+5に、10進アスキーデータを固定桁数で設定します。



ASCII S: 符号のアスキーコード

ASCII 10⁰: 一の位のアスキーコード

ASCII 10¹: 十の位のアスキーコード

ASCII 10²: 百の位のアスキーコード

ASCII 10³: 千の位のアスキーコード

ASCII 10⁴: 万の位のアスキーコード

ASCII 10⁵: 十万の位のアスキーコード

ASCII 10⁶: 百万の位のアスキーコード

ASCII 10⁷: 千万の位のアスキーコード

ASCII 10⁸: 億の位のアスキーコード

ASCII 10⁹: 十億の位のアスキーコード

(1): 無視します。

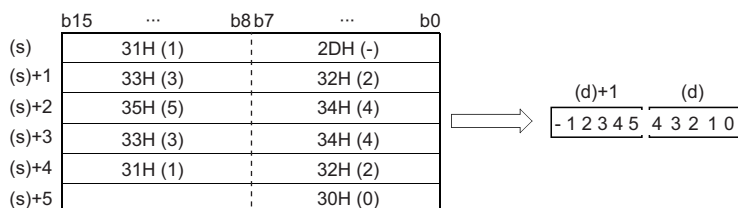
(2): 上位16ビット

(3): 下位16ビット

- (s)~(s)+5で指定するアスキーデータは、DDABIN(P)の場合は-2147483648~2147483647、DDABIN(P)_Uの場合は0~4294967295の範囲内です。また、(s)+5の上位バイトおよび(s)+6以降に格納されているデータは無視されます。
- 符号データには、変換するデータが正のとき20Hを設定し、負のとき2DHを設定します。(20H、2DH以外を設定した場合は、正のデータとして処理します。)
- 各位に設定するアスキーコードは、30H~39Hの範囲内です。
- 各位に設定するアスキーコードが20Hまたは00Hのときは、30Hとして処理します。

例

DDABIN(P)命令において、(s)に"-1234543210"を設定した場合



■SM705(変換桁数切換)がONの場合の(s)の設定方法

(s)には、任意の桁数(00H(NULLコード)まで)の10進アスキーデータを設定します。ただし、整数部が最大桁数(10桁)の場合は、00H(NULLコード)の設定は不要です。

(s)の設定方法を下記に示します。

(s)に設定する値 0 - 正の数(数値部1桁)	(s)~(s)+5の内容 (s)の上位バイトに00Hを設定します。 (s)+1以降は無視されます。 <table><tr><td></td><td>b15</td><td>...</td><td>b8</td><td>b7</td><td>...</td><td>b0</td></tr><tr><td>(s)</td><td colspan="2">00H</td><td colspan="2"></td><td colspan="2">ASCII 10⁰</td></tr><tr><td>(s)+1</td><td colspan="6"></td></tr><tr><td>(s)+2</td><td colspan="6"></td></tr><tr><td>(s)+3</td><td colspan="6"></td></tr><tr><td>(s)+4</td><td colspan="6"></td></tr><tr><td>(s)+5</td><td colspan="6"></td></tr></table>		b15	...	b8	b7	...	b0	(s)	00H				ASCII 10 ⁰		(s)+1							(s)+2							(s)+3							(s)+4							(s)+5							(s)に設定する値 - 正の数(数値部2桁) - 負の数(数値部1桁)	(s)~(s)+5の内容 (s)+1の下位バイトに00Hを設定します。 (s)+1の上位バイト以降は無視されます。 <table><tr><td></td><td>b15</td><td>...</td><td>b8</td><td>b7</td><td>...</td><td>b0</td></tr><tr><td>(s)</td><td colspan="2">ASCII 10⁰</td><td colspan="2"></td><td colspan="2">ASCII 10¹ / 2DH (-)</td></tr><tr><td>(s)+1</td><td colspan="2"></td><td colspan="2"></td><td colspan="2">00H</td></tr><tr><td>(s)+2</td><td colspan="6"></td></tr><tr><td>(s)+3</td><td colspan="6"></td></tr><tr><td>(s)+4</td><td colspan="6"></td></tr><tr><td>(s)+5</td><td colspan="6"></td></tr></table>		b15	...	b8	b7	...	b0	(s)	ASCII 10 ⁰				ASCII 10 ¹ / 2DH (-)		(s)+1					00H		(s)+2							(s)+3							(s)+4							(s)+5						
	b15	...	b8	b7	...	b0																																																																																															
(s)	00H				ASCII 10 ⁰																																																																																																
(s)+1																																																																																																					
(s)+2																																																																																																					
(s)+3																																																																																																					
(s)+4																																																																																																					
(s)+5																																																																																																					
	b15	...	b8	b7	...	b0																																																																																															
(s)	ASCII 10 ⁰				ASCII 10 ¹ / 2DH (-)																																																																																																
(s)+1					00H																																																																																																
(s)+2																																																																																																					
(s)+3																																																																																																					
(s)+4																																																																																																					
(s)+5																																																																																																					
:																																																																																																					
- 正の数(数値部9桁) - 負の数(数値部8桁)	(s)+4の上位バイトに00Hを設定します。 (s)+5は無視されます。 <table><tr><td></td><td>b15</td><td>...</td><td>b8</td><td>b7</td><td>...</td><td>b0</td></tr><tr><td>(s)</td><td colspan="2">ASCII 10⁷</td><td colspan="2"></td><td colspan="2">ASCII 10⁸ / 2DH (-)</td></tr><tr><td>(s)+1</td><td colspan="2">ASCII 10⁵</td><td colspan="2"></td><td colspan="2">ASCII 10⁶</td></tr><tr><td>(s)+2</td><td colspan="2">ASCII 10³</td><td colspan="2"></td><td colspan="2">ASCII 10⁴</td></tr><tr><td>(s)+3</td><td colspan="2">ASCII 10¹</td><td colspan="2"></td><td colspan="2">ASCII 10²</td></tr><tr><td>(s)+4</td><td colspan="2">00H</td><td colspan="2"></td><td colspan="2">ASCII 10⁰</td></tr><tr><td>(s)+5</td><td colspan="6"></td></tr></table>		b15	...	b8	b7	...	b0	(s)	ASCII 10 ⁷				ASCII 10 ⁸ / 2DH (-)		(s)+1	ASCII 10 ⁵				ASCII 10 ⁶		(s)+2	ASCII 10 ³				ASCII 10 ⁴		(s)+3	ASCII 10 ¹				ASCII 10 ²		(s)+4	00H				ASCII 10 ⁰		(s)+5							負の数(数値部9桁)	(s)+5の下位バイトに00Hを設定します。 (s)+5の上位バイトは無視されます。 <table><tr><td></td><td>b15</td><td>...</td><td>b8</td><td>b7</td><td>...</td><td>b0</td></tr><tr><td>(s)</td><td colspan="2">ASCII 10⁸</td><td colspan="2"></td><td colspan="2">2DH (-)</td></tr><tr><td>(s)+1</td><td colspan="2">ASCII 10⁶</td><td colspan="2"></td><td colspan="2">ASCII 10⁷</td></tr><tr><td>(s)+2</td><td colspan="2">ASCII 10⁴</td><td colspan="2"></td><td colspan="2">ASCII 10⁵</td></tr><tr><td>(s)+3</td><td colspan="2">ASCII 10²</td><td colspan="2"></td><td colspan="2">ASCII 10³</td></tr><tr><td>(s)+4</td><td colspan="2">ASCII 10⁰</td><td colspan="2"></td><td colspan="2">ASCII 10¹</td></tr><tr><td>(s)+5</td><td colspan="2"></td><td colspan="2"></td><td colspan="2">00H</td></tr></table>		b15	...	b8	b7	...	b0	(s)	ASCII 10 ⁸				2DH (-)		(s)+1	ASCII 10 ⁶				ASCII 10 ⁷		(s)+2	ASCII 10 ⁴				ASCII 10 ⁵		(s)+3	ASCII 10 ²				ASCII 10 ³		(s)+4	ASCII 10 ⁰				ASCII 10 ¹		(s)+5					00H	
	b15	...	b8	b7	...	b0																																																																																															
(s)	ASCII 10 ⁷				ASCII 10 ⁸ / 2DH (-)																																																																																																
(s)+1	ASCII 10 ⁵				ASCII 10 ⁶																																																																																																
(s)+2	ASCII 10 ³				ASCII 10 ⁴																																																																																																
(s)+3	ASCII 10 ¹				ASCII 10 ²																																																																																																
(s)+4	00H				ASCII 10 ⁰																																																																																																
(s)+5																																																																																																					
	b15	...	b8	b7	...	b0																																																																																															
(s)	ASCII 10 ⁸				2DH (-)																																																																																																
(s)+1	ASCII 10 ⁶				ASCII 10 ⁷																																																																																																
(s)+2	ASCII 10 ⁴				ASCII 10 ⁵																																																																																																
(s)+3	ASCII 10 ²				ASCII 10 ³																																																																																																
(s)+4	ASCII 10 ⁰				ASCII 10 ¹																																																																																																
(s)+5					00H																																																																																																
正の数(数値部10桁)	(s)+5は無視されます。最大桁数のため、00Hの設定は不要です。 <table><tr><td></td><td>b15</td><td>...</td><td>b8</td><td>b7</td><td>...</td><td>b0</td></tr><tr><td>(s)</td><td colspan="2">ASCII 10⁸</td><td colspan="2"></td><td colspan="2">ASCII 10⁹</td></tr><tr><td>(s)+1</td><td colspan="2">ASCII 10⁶</td><td colspan="2"></td><td colspan="2">ASCII 10⁷</td></tr><tr><td>(s)+2</td><td colspan="2">ASCII 10⁴</td><td colspan="2"></td><td colspan="2">ASCII 10⁵</td></tr><tr><td>(s)+3</td><td colspan="2">ASCII 10²</td><td colspan="2"></td><td colspan="2">ASCII 10³</td></tr><tr><td>(s)+4</td><td colspan="2">ASCII 10⁰</td><td colspan="2"></td><td colspan="2">ASCII 10¹</td></tr><tr><td>(s)+5</td><td colspan="6"></td></tr></table>		b15	...	b8	b7	...	b0	(s)	ASCII 10 ⁸				ASCII 10 ⁹		(s)+1	ASCII 10 ⁶				ASCII 10 ⁷		(s)+2	ASCII 10 ⁴				ASCII 10 ⁵		(s)+3	ASCII 10 ²				ASCII 10 ³		(s)+4	ASCII 10 ⁰				ASCII 10 ¹		(s)+5							負の数(数値部10桁)	(s)+5の上位バイトは無視されます。最大桁数のため、00Hの設定は不要です。 <table><tr><td></td><td>b15</td><td>...</td><td>b8</td><td>b7</td><td>...</td><td>b0</td></tr><tr><td>(s)</td><td colspan="2">ASCII 10⁹</td><td colspan="2"></td><td colspan="2">2DH (-)</td></tr><tr><td>(s)+1</td><td colspan="2">ASCII 10⁷</td><td colspan="2"></td><td colspan="2">ASCII 10⁸</td></tr><tr><td>(s)+2</td><td colspan="2">ASCII 10⁵</td><td colspan="2"></td><td colspan="2">ASCII 10⁶</td></tr><tr><td>(s)+3</td><td colspan="2">ASCII 10³</td><td colspan="2"></td><td colspan="2">ASCII 10⁴</td></tr><tr><td>(s)+4</td><td colspan="2">ASCII 10¹</td><td colspan="2"></td><td colspan="2">ASCII 10²</td></tr><tr><td>(s)+5</td><td colspan="2"></td><td colspan="2"></td><td colspan="2">ASCII 10⁰</td></tr></table>		b15	...	b8	b7	...	b0	(s)	ASCII 10 ⁹				2DH (-)		(s)+1	ASCII 10 ⁷				ASCII 10 ⁸		(s)+2	ASCII 10 ⁵				ASCII 10 ⁶		(s)+3	ASCII 10 ³				ASCII 10 ⁴		(s)+4	ASCII 10 ¹				ASCII 10 ²		(s)+5					ASCII 10 ⁰	
	b15	...	b8	b7	...	b0																																																																																															
(s)	ASCII 10 ⁸				ASCII 10 ⁹																																																																																																
(s)+1	ASCII 10 ⁶				ASCII 10 ⁷																																																																																																
(s)+2	ASCII 10 ⁴				ASCII 10 ⁵																																																																																																
(s)+3	ASCII 10 ²				ASCII 10 ³																																																																																																
(s)+4	ASCII 10 ⁰				ASCII 10 ¹																																																																																																
(s)+5																																																																																																					
	b15	...	b8	b7	...	b0																																																																																															
(s)	ASCII 10 ⁹				2DH (-)																																																																																																
(s)+1	ASCII 10 ⁷				ASCII 10 ⁸																																																																																																
(s)+2	ASCII 10 ⁵				ASCII 10 ⁶																																																																																																
(s)+3	ASCII 10 ³				ASCII 10 ⁴																																																																																																
(s)+4	ASCII 10 ¹				ASCII 10 ²																																																																																																
(s)+5					ASCII 10 ⁰																																																																																																

ASCII 10⁰: 一の位のアスキーコード
 ASCII 10¹: 十の位のアスキーコード
 :
 ASCII 10⁹: 十億の位のアスキーコード

- (s)~(s)+5で指定するアスキーデータは、DDABIN(P)命令の場合は-2147483648~2147483647、DDABIN(P)_U命令の場合は0~4294967295の範囲内です。
- 変換するデータが負の数の場合は、(s)+0の下位バイトに符号データとして2DH(-)を設定します。変換するデータが0または正の数の場合は符号データは設定せず、最上位桁のアスキーコードを設定します。
- 各位に設定するアスキーコードは、30H~39Hの範囲内です。
- 正の数で数値部10桁の場合、(s)+5以降に格納されているデータは無視されます。負の数で数値部10桁の場合、(s)+5の上位バイト以降に格納されているデータは無視されます。
- 各位に設定するアスキーコードが20Hのときは、30Hとして処理されます。00Hのときは、10進アスキーデータの終端として処理されます。
- 下記の場合、(d)には0が格納されます。
 - 1文字目が00H(NULL)
 - 1文字目が2DH(-)で、2文字目が00H(NULL)

エラーコード (SD0)	内容
3281H	(s)~(s)+5に変換できない不正なデータを入力したとき。 • 1文字目のアスキーコードが, 2DH, 30H~39H, 20H, 00H以外。*1 • 2文字目以降のアスキーコードが, 30H~39H, 20H, 00H以外。 • DDABIN(P)命令使用時, アスキーデータが-2147483648~2147483647以外。 • DDABIN(P)_U命令使用時, アスキーデータが0~4294967295以外。

*1 SM705(変換桁数切換)がOFFの場合は, 1文字目のアスキーコードに何を設定してもエラーになりません。

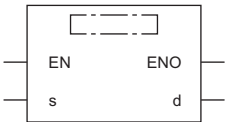
16進アスキーデータ→BIN16ビットデータ変換

HABIN(P)

16進アスキーデータを，BIN16ビットデータに変換します。

ラダー	ST
	ENO:=HABIN(EN,s,d); ENO:=HABINP(EN,s,d);

FBD/LD



■実行条件

命令	実行条件
HABIN	
HABINP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BIN値に変換するアスキーデータ，またはアスキーデータが格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	変換結果を格納するデバイス	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

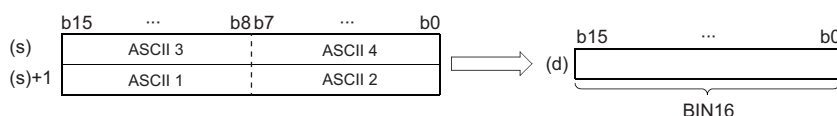
機能

- (s)で指定されたデバイス番号以降に格納されている16進アスキーデータを、BIN16ビットデータに変換して、(d)で指定されたデバイスに格納します。
- SM705(変換桁数切換)の状態によって、(s)に設定する16進アスキーデータの設定方法が異なります。

SM705の状態	(s)の設定方法	参照先
OFF	固定桁数(4桁)で設定	455ページ SM705(変換桁数切換)がOFFの場合の(s)の設定方法
ON	任意の桁数(最大4桁)で設定	455ページ SM705(変換桁数切換)がONの場合の(s)の設定方法

■SM705(変換桁数切換)がOFFの場合の(s)の設定方法

- (s)~(s)+1に、16進アスキーデータを4桁固定で設定します。

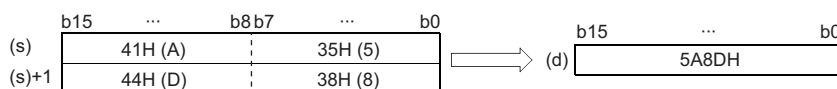


ASCII □: □桁目のアスキーコード

- (s)~(s)+1で指定するアスキーデータは、0000H~FFFFHの範囲内です。
- (s)+2以降のデータは無視されます。
- 各位に設定するアスキーコードは、30H~39Hおよび41H~46Hの範囲内です。

例

(s)に5A8DHが指定された場合



■SM705(変換桁数切換)がONの場合の(s)の設定方法

(s)には、任意の桁数(00H(NULLコード)まで)の16進アスキーデータを設定します。ただし、最大桁数(4桁)の場合は、00H(NULLコード)の設定は不要です。

(s)の設定方法を下記に示します。

(s)に設定する値	(s)~(s)+1の内容	(s)に設定する値	(s)~(s)+1の内容
• 0H~FH	• (s)+0の上位バイトに00Hを設定します。 • (s)+1以降は無視されます。	• 10H~FFH	• (s)+1の下位バイトに00Hを設定します。 • (s)+1の上位バイト以降は無視されます。
• 100H~FFFFH	• (s)+1の上位バイトに00Hを設定します。 • (s)+2以降は無視されます。	• 1000H~FFFFH	• (s)+2以降は無視されます。

ASCII □: □桁目のアスキーコード

- (s)~(s)+1で指定するアスキーデータは、0000H~FFFFHの範囲内です。
- (s)+2以降のデータは無視されます。
- 各位に設定するアスキーコードは、30H~39Hおよび41H~46Hの範囲内です。
- 各位に設定するアスキーコードが00Hのときは、16進アスキーデータの終端として処理されます。

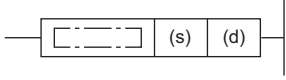
エラー

エラーコード (SD0)	内容
3281H	(s)~(s)+1に変換できない不正なデータを入力したとき。 • 各桁のアスキーコードが、30H~39H、41H~46H以外。

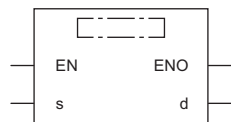
16進アスキーデータ→BIN32ビットデータ変換

DHABIN(P)

16進アスキーデータを，BIN32ビットデータに変換します。

ラダー	ST
	ENO:=DHABIN(EN,s,d); ENO:=DHABINP(EN,s,d);

FBD/LD



■実行条件

命令	実行条件
DHABIN	
DHABINP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BIN値に変換するアスキーデータ，またはアスキーデータが格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	変換結果を格納する先頭デバイス	—	符号付きBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	○	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

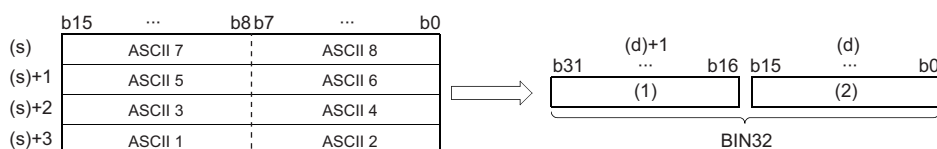
機能

- (s)で指定されたデバイス番号以降に格納されている16進アスキーデータを、BIN32ビットデータに変換して、(d)で指定されたデバイス番号に格納します。
- SM705(変換桁数切換)の状態によって、(s)に設定する16進アスキーデータの設定方法が異なります。

SM705の状態	(s)の設定方法	参照先
OFF	固定桁数(8桁)で設定	457ページ SM705(変換桁数切換)がOFFの場合の(s)の設定方法
ON	任意の桁数(最大8桁)で設定	458ページ SM705(変換桁数切換)がONの場合の(s)の設定方法

■SM705(変換桁数切換)がOFFの場合の(s)の設定方法

- (s)~(s)+3に、16進アスキーデータを8桁固定で設定します。



ASCII □: □桁目のアスキーコード

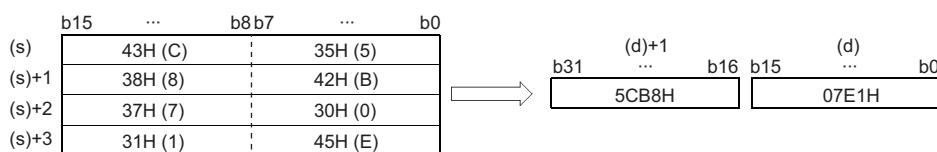
(1): 上位16ビット

(2): 下位16ビット

- (s)~(s)+3で指定するアスキーデータは、00000000H~FFFFFFFFHの範囲内です。
- (s)+4以降のデータは無視されます。
- 各位に設定するアスキーコードは、30H~39Hおよび41H~46Hの範囲内です。

例

(s)に5CB807E1Hが指定された場合



■SM705(変換桁数切換)がONの場合の(s)の設定方法

(s)には、任意の桁数(00H(NULLコード)まで)の16進アスキーデータを設定します。ただし、最大桁数(8桁)の場合は、00H(NULLコード)の設定は不要です。

(s)の設定方法を下記に示します。

(s)に設定する値	(s)~(s)+3の内容	(s)に設定する値	(s)~(s)+3の内容																																																		
• 0H~FH	• (s)の上位バイトに00Hを設定します。 • (s)+1以降は無視されます。 <table><tr><td>b15</td><td>...</td><td>b8b7</td><td>...</td><td>b0</td></tr><tr><td>(s)</td><td>00H</td><td></td><td>ASCII 1</td><td></td></tr><tr><td>(s)+1</td><td colspan="4"></td></tr><tr><td>(s)+2</td><td colspan="4"></td></tr><tr><td>(s)+3</td><td colspan="4"></td></tr></table>	b15	...	b8b7	...	b0	(s)	00H		ASCII 1		(s)+1					(s)+2					(s)+3					• 10H~FFH	• (s)+1の下部バイトに00Hを設定します。 • (s)+1の上位バイト以降は無視されます。 <table><tr><td>b15</td><td>...</td><td>b8b7</td><td>...</td><td>b0</td></tr><tr><td>(s)</td><td>ASCII 1</td><td></td><td>ASCII 2</td><td></td></tr><tr><td>(s)+1</td><td></td><td></td><td>00H</td><td></td></tr><tr><td>(s)+2</td><td colspan="4"></td></tr><tr><td>(s)+3</td><td colspan="4"></td></tr></table>	b15	...	b8b7	...	b0	(s)	ASCII 1		ASCII 2		(s)+1			00H		(s)+2					(s)+3				
b15	...	b8b7	...	b0																																																	
(s)	00H		ASCII 1																																																		
(s)+1																																																					
(s)+2																																																					
(s)+3																																																					
b15	...	b8b7	...	b0																																																	
(s)	ASCII 1		ASCII 2																																																		
(s)+1			00H																																																		
(s)+2																																																					
(s)+3																																																					
⋮																																																					
• 1000000H~FFFFFH	• (s)+3の上位バイトに00Hを設定します。 • (s)+4以降は無視されます。 <table><tr><td>b15</td><td>...</td><td>b8b7</td><td>...</td><td>b0</td></tr><tr><td>(s)</td><td>ASCII 6</td><td></td><td>ASCII 7</td><td></td></tr><tr><td>(s)+1</td><td>ASCII 4</td><td></td><td>ASCII 5</td><td></td></tr><tr><td>(s)+2</td><td>ASCII 2</td><td></td><td>ASCII 3</td><td></td></tr><tr><td>(s)+3</td><td>00H</td><td></td><td>ASCII 1</td><td></td></tr></table>	b15	...	b8b7	...	b0	(s)	ASCII 6		ASCII 7		(s)+1	ASCII 4		ASCII 5		(s)+2	ASCII 2		ASCII 3		(s)+3	00H		ASCII 1		• 10000000H~FFFFFH	• (s)+4以降は無視されます。 <table><tr><td>b15</td><td>...</td><td>b8b7</td><td>...</td><td>b0</td></tr><tr><td>(s)</td><td>ASCII 7</td><td></td><td>ASCII 8</td><td></td></tr><tr><td>(s)+1</td><td>ASCII 5</td><td></td><td>ASCII 6</td><td></td></tr><tr><td>(s)+2</td><td>ASCII 3</td><td></td><td>ASCII 4</td><td></td></tr><tr><td>(s)+3</td><td>ASCII 1</td><td></td><td>ASCII 2</td><td></td></tr></table>	b15	...	b8b7	...	b0	(s)	ASCII 7		ASCII 8		(s)+1	ASCII 5		ASCII 6		(s)+2	ASCII 3		ASCII 4		(s)+3	ASCII 1		ASCII 2	
b15	...	b8b7	...	b0																																																	
(s)	ASCII 6		ASCII 7																																																		
(s)+1	ASCII 4		ASCII 5																																																		
(s)+2	ASCII 2		ASCII 3																																																		
(s)+3	00H		ASCII 1																																																		
b15	...	b8b7	...	b0																																																	
(s)	ASCII 7		ASCII 8																																																		
(s)+1	ASCII 5		ASCII 6																																																		
(s)+2	ASCII 3		ASCII 4																																																		
(s)+3	ASCII 1		ASCII 2																																																		

ASCII 口: 口桁目のアスキーコード

- (s)~(s)+3で指定するアスキーデータは、00000000H~FFFFFFFFHの範囲内です。
- (s)+4以降のデータは無視されます。
- 各位に設定するアスキーコードは、30H~39Hおよび41H~46Hの範囲内です。
- 各位に設定するアスキーコードが00Hのときは、16進アスキーデータの終端として処理されます。

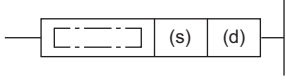
エラー

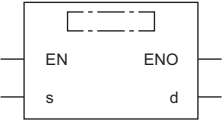
エラーコード (SD0)	内容
3281H	(s)~(s)+3に変換できない不正なデータを入力したとき。 • 各桁のアスキーコードが、30H~39H、41H~46H以外。

10進アスキーデータ→BCD4桁データ変換

DABCD(P)

10進アスキーデータを、BCD4桁データに変換します。

ラダー	ST
	ENO:=DABCD(EN,s,d); ENO:=DABCDP(EN,s,d)

FBD/LD


■実行条件

命令	実行条件
DABCD	
DABCDP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BCD値に変換するアスキーデータ、またはアスキーデータが格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	変換結果を格納するデバイス	—	BCD4桁	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

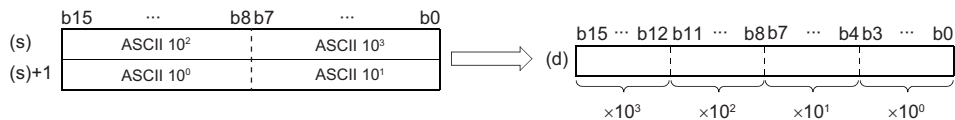
機能

- (s)で指定されたデバイス番号以降に格納されている10進アスキーデータを、BCD4桁データに変換して、(d)で指定されたデバイスに格納します。
- SM705(変換桁数切換)の状態によって、(s)に設定する10進アスキーデータの設定方法が異なります。

SM705の状態	(s)の設定方法	参照先
OFF	固定桁数(4桁)で設定	460ページ SM705(変換桁数切換)がOFFの場合の(s)の設定方法
ON	任意の桁数(最大4桁)で設定	461ページ SM705(変換桁数切換)がONの場合の(s)の設定方法

■SM705(変換桁数切換)がOFFの場合の(s)の設定方法

- (s)~(s)+1に、10進アスキーデータを4桁固定で設定します。



ASCII 10³: 千の位のアスキーコード

ASCII 10²: 百の位のアスキーコード

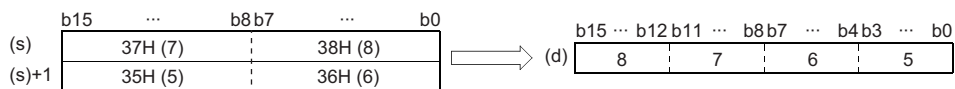
ASCII 10¹: 十の位のアスキーコード

ASCII 10⁰: 一の位のアスキーコード

- (s)~(s)+1で指定するアスキーデータは、0~9999の範囲内です。
- (s)+2以降のデータは無視されます。
- 各位に設定するアスキーコードは、30H~39Hの範囲内です。
- 各位に設定するアスキーコードが20Hまたは00Hのときは、30Hとして処理します。

例

(s)に8765が指定された場合



■SM705(変換桁数切換)がONの場合の(s)の設定方法

(s)には、任意の桁数(00H(NULLコード)まで)の10進アスキーデータを設定します。ただし、最大桁数(4桁)の場合は、00H(NULLコード)の設定は不要です。

(s)の設定方法を下記に示します。

(s)に設定する値	(s)~(s)+1の内容	(s)に設定する値	(s)~(s)+1の内容
0~9	(s)+0の上位バイトに00Hを設定します。 (s)+1以降は無視されます。 b15 ... b8 b7 ... b0 (s) 00H ... ASCII 10⁰ (s)+1	10~99	(s)+1の下位バイトに00Hを設定します。 (s)+1の上位バイト以降は無視されます。 b15 ... b8 b7 ... b0 (s) ASCII 10⁰ ... ASCII 10¹ (s)+1 ... 00H
100~999	(s)+1の上位バイトに00Hを設定します。 (s)+2以降は無視されます。 b15 ... b8 b7 ... b0 (s) ASCII 10¹ ... ASCII 10² (s)+1 00H ... ASCII 10⁰	1000~9999	(s)+2以降は無視されます。 b15 ... b8 b7 ... b0 (s) ASCII 10² ... ASCII 10³ (s)+1 ASCII 10⁰ ... ASCII 10¹

ASCII 10³: 千の位のアスキーコード

ASCII 10²: 百の位のアスキーコード

ASCII 10¹: 十の位のアスキーコード

ASCII 10⁰: 一の位のアスキーコード

- (s)~(s)+1で指定するアスキーデータは、0~9999の範囲内です。
- (s)+2以降のデータは無視されます。
- 各位に設定するアスキーコードは、30H~39Hの範囲内です。
- 各位に設定するアスキーコードが20Hのときは、30Hとして処理されます。00Hのときは、10進アスキーデータの終端として処理されます。

エラー

エラーコード (SD0)	内容
3281H	(s)に変換できない不正なデータを入力したとき。 • データ中に0~9以外の文字がある。

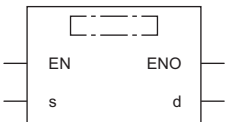
10進アスキーデータ→BCD8桁データ変換

DDABCD(P)

10進アスキーデータを，BCD8桁データに変換する。

ラダー	ST
	ENO:=DDABCD(EN,s,d); ENO:=DDABCDP(EN,s,d);

FBD/LD



■実行条件

命令	実行条件
DDABCD	
DDABCDP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BCD値に変換するアスキーデータ，またはアスキーデータが格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	変換結果を格納する先頭デバイス	—	BCD8桁	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

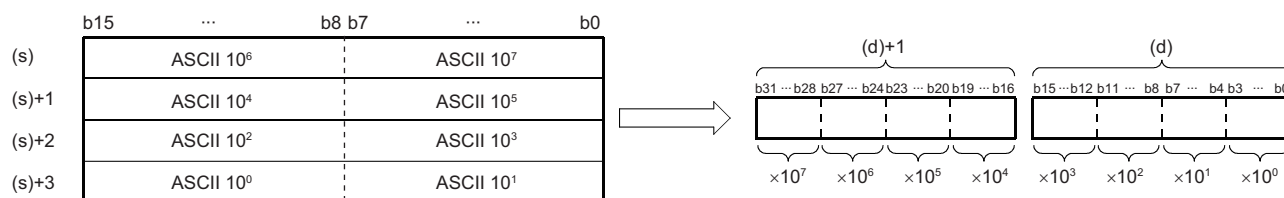
機能

- (s)で指定されたデバイス番号以降に格納されている10進アスキーデータを、BCD8桁データに変換して、(d)で指定されたデバイス番号に格納します。
- SM705(変換桁数切換)の状態によって、(s)に設定する10進アスキーデータの設定方法が異なります。

SM705の状態	(s)の設定方法	参照先
OFF	固定桁数(8桁)で設定	463ページ SM705(変換桁数切換)がOFFの場合の(s)の設定方法
ON	任意の桁数(最大8桁)で設定	464ページ SM705(変換桁数切換)がONの場合の(s)の設定方法

■SM705(変換桁数切換)がOFFの場合の(s)の設定方法

- (s)~(s)+3に、10進アスキーデータを8桁固定で設定します。



ASCII 10⁷: 千万の位のアスキーコード

ASCII 10⁶: 百万の位のアスキーコード

ASCII 10⁵: 十万の位のアスキーコード

ASCII 10⁴: 万の位のアスキーコード

ASCII 10³: 千の位のアスキーコード

ASCII 10²: 百の位のアスキーコード

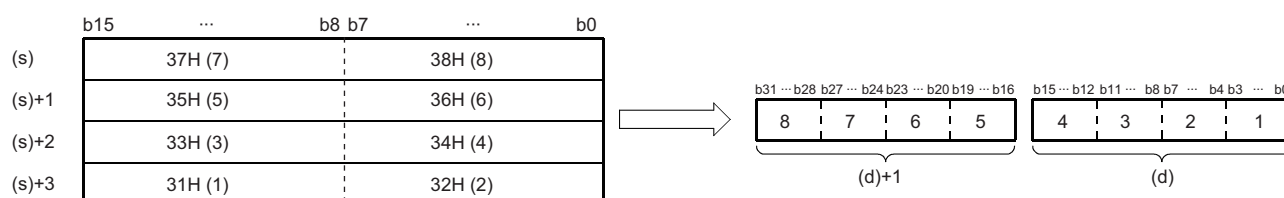
ASCII 10¹: 十の位のアスキーコード

ASCII 10⁰: 一の位のアスキーコード

- (s)~(s)+3で指定するアスキーデータは、0~99999999の範囲内です。
- (s)+4以降のデータは無視されます。
- 各位に設定するアスキーコードは、30H~39Hの範囲内です。
- 各位に設定するアスキーコードが20Hまたは00Hのときは、30Hとして処理します。

例

(s)に87654321が指定された場合



■SM705(変換桁数切換)がONの場合の(s)の設定方法

(s)には、任意の桁数(00H(NULLコード)まで)の10進アスキーデータを設定します。ただし、最大桁数(8桁)の場合は、00H(NULLコード)の設定は不要です。

(s)の設定方法を下記に示します。

(s)に設定する値	(s)~(s)+3の内容	(s)に設定する値	(s)~(s)+3の内容																								
• 0~9	• (s)の上位バイトに00Hを設定します。 • (s)+1以降は無視されます。 b15 ... b8b7 ... b0 <table><tr><td>(s)</td><td>00H</td><td>ASCII 10⁰</td></tr><tr><td>(s)+1</td><td></td><td></td></tr><tr><td>(s)+2</td><td></td><td></td></tr><tr><td>(s)+3</td><td></td><td></td></tr></table>	(s)	00H	ASCII 10 ⁰	(s)+1			(s)+2			(s)+3			• 10~99	• (s)+1の下位バイトに00Hを設定します。 • (s)+1の上位バイト以降は無視されます。 b15 ... b8b7 ... b0 <table><tr><td>(s)</td><td>ASCII 10⁰</td><td>ASCII 10¹</td></tr><tr><td>(s)+1</td><td></td><td>00H</td></tr><tr><td>(s)+2</td><td></td><td></td></tr><tr><td>(s)+3</td><td></td><td></td></tr></table>	(s)	ASCII 10 ⁰	ASCII 10 ¹	(s)+1		00H	(s)+2			(s)+3		
(s)	00H	ASCII 10 ⁰																									
(s)+1																											
(s)+2																											
(s)+3																											
(s)	ASCII 10 ⁰	ASCII 10 ¹																									
(s)+1		00H																									
(s)+2																											
(s)+3																											
⋮																											
• 1000000~9999999 9	• (s)+3の上位バイトに00Hを設定します。 • (s)+4以降は無視されます。 b15 ... b8b7 ... b0 <table><tr><td>(s)</td><td>ASCII 10⁵</td><td>ASCII 10⁶</td></tr><tr><td>(s)+1</td><td>ASCII 10³</td><td>ASCII 10⁴</td></tr><tr><td>(s)+2</td><td>ASCII 10¹</td><td>ASCII 10²</td></tr><tr><td>(s)+3</td><td>00H</td><td>ASCII 10⁰</td></tr></table>	(s)	ASCII 10 ⁵	ASCII 10 ⁶	(s)+1	ASCII 10 ³	ASCII 10 ⁴	(s)+2	ASCII 10 ¹	ASCII 10 ²	(s)+3	00H	ASCII 10 ⁰	• 10000000~99999999 999	• (s)+4以降は無視されます。 b15 ... b8b7 ... b0 <table><tr><td>(s)</td><td>ASCII 10⁶</td><td>ASCII 10⁷</td></tr><tr><td>(s)+1</td><td>ASCII 10⁴</td><td>ASCII 10⁵</td></tr><tr><td>(s)+2</td><td>ASCII 10²</td><td>ASCII 10³</td></tr><tr><td>(s)+3</td><td>ASCII 10⁰</td><td>ASCII 10¹</td></tr></table>	(s)	ASCII 10 ⁶	ASCII 10 ⁷	(s)+1	ASCII 10 ⁴	ASCII 10 ⁵	(s)+2	ASCII 10 ²	ASCII 10 ³	(s)+3	ASCII 10 ⁰	ASCII 10 ¹
(s)	ASCII 10 ⁵	ASCII 10 ⁶																									
(s)+1	ASCII 10 ³	ASCII 10 ⁴																									
(s)+2	ASCII 10 ¹	ASCII 10 ²																									
(s)+3	00H	ASCII 10 ⁰																									
(s)	ASCII 10 ⁶	ASCII 10 ⁷																									
(s)+1	ASCII 10 ⁴	ASCII 10 ⁵																									
(s)+2	ASCII 10 ²	ASCII 10 ³																									
(s)+3	ASCII 10 ⁰	ASCII 10 ¹																									

ASCII 10⁷: 千万の位のアスキーコード

ASCII 10⁶: 百万の位のアスキーコード

ASCII 10⁵: 十万の位のアスキーコード

ASCII 10⁴: 万の位のアスキーコード

ASCII 10³: 千の位のアスキーコード

ASCII 10²: 百の位のアスキーコード

ASCII 10¹: 十の位のアスキーコード

ASCII 10⁰: 一の位のアスキーコード

- (s)~(s)+3で指定するアスキーデータは、0~99999999の範囲内です。
- (s)+4以降のデータは無視されます。
- 各位に設定するアスキーコードは、30H~39Hの範囲内です。
- 各位に設定するアスキーコードが20Hのときは、30Hとして処理されます。00Hのときは、10進アスキーデータの終端として処理されます。

エラー

エラーコード (SD0)	内容
3281H	(s)に変換できない不正なデータを入力したとき。 ・データ中に0~9以外の文字がある。

10進文字列→BIN16ビットデータ変換

VAL(P)(_U)

文字列をBIN16ビットデータに変換します。

ラダー	ST
	ENO:=VAL(EN,s,d1,d2); ENO:=VALP(EN,s,d1,d2);
	ENO:=VAL_U(EN,s,d1,d2); ENO:=VALP_U(EN,s,d1,d2);

FBD/LD

■実行条件

命令	実行条件
VAL VAL_U	
VALP VALP_U	

設定データ

■内容、範囲、データ型

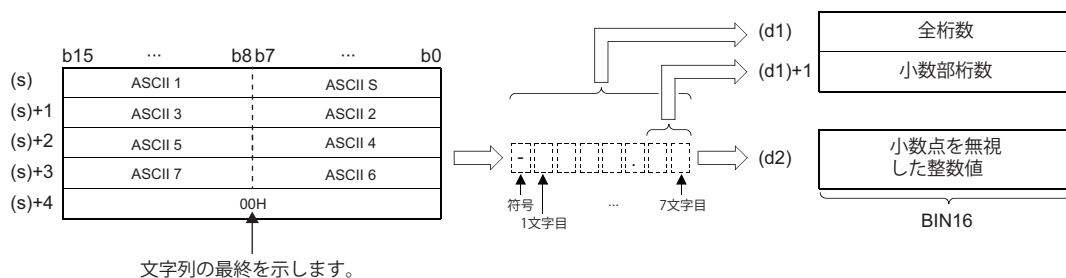
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータに変換する文字列または、文字列が格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d1)	VAL(P)	—	符号付きBIN16ビット	ANY16_S_ARRAY (要素数: 2)
	VAL(P)_U		符号なしBIN16ビット	ANY16_U_ARRAY (要素数: 2)
(d2)	VAL(P)	—	符号付きBIN16ビット	ANY16_S
	VAL(P)_U		符号なしBIN16ビット	ANY16_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d1)	○	—	○	—	—	—	—	○	—	—	—	—
(d2)	○	○	○	○	○	—	—	○	—	—	—	—

機能

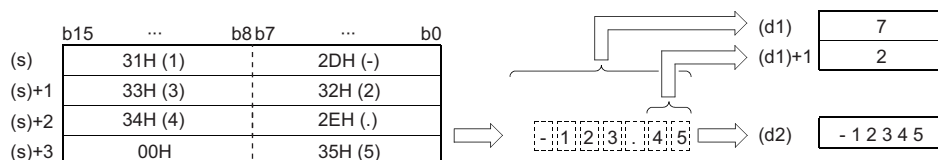
- (s)で指定されたデバイス番号以降に格納されている文字列を、BIN16ビットデータに変換して、桁数を(d1)に、BINデータを(d2)に格納します。文字列→BIN変換では、(s)で指定されたデバイス番号から、00Hが格納されているデバイス番号までのデータを、文字列として処理します。



ASCII S: 符号のアスキーコード
ASCII □: □文字目のアスキーコード

例

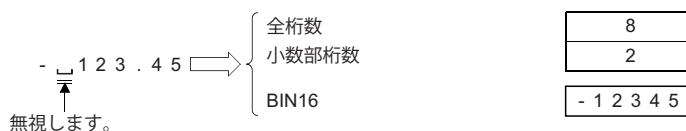
(s)以降に「-123.45」の文字列が指定された場合(符号付きを指定した場合)



- (s)で指定する文字列の全文字数は、2~8文字です。
- (s)で指定する文字列のうち、小数部となる文字数は、0~5文字です。ただし、「全桁数-3」以下で指定してください。
- BIN値に変換できる数値の文字列の範囲は、小数点を無視した値で符号付きを指定した場合は-32768~32767、符号なしを指定した場合は0~65535となります。また、符号と小数点を除く数値の文字列は、30H~39Hの範囲内のみ指定できます。(小数点を無視した値...「-12345.6」の場合は、「-123456」となります。)
- 符号には、正の数値を表す場合は20Hを設定し、負の数値を表す場合は2DHを設定します。
- 小数点には、2EHを設定します。
- (d1)に格納する全桁数は、数値を表すすべての文字(符号、小数点を含む)の数を格納します。(d1)+1に格納する小数部桁数は、2EH(.)以降の小数部を表す文字数を格納します。(d2)に格納するBIN16ビットデータは、小数点を無視した文字列をBIN値に変換して格納します。
- (s)で指定された文字列で、符号と最初の0以外の数値との間に、20H(スペース)か30H(0)が存在する場合は、20H、30Hを無視してBIN値に変換します。

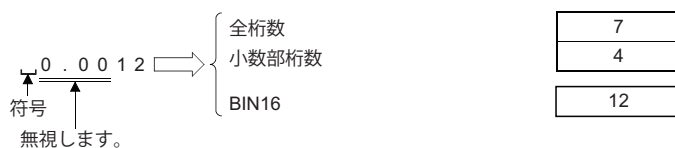
例

符号と最初の0以外の数値との間に20Hが存在する場合(符号付きを指定した場合)



例

符号と最初の0以外の数値との間に30Hが存在する場合



エラーコード (SD0)	内容
2820H	(s)で指定したデバイス番号から、該当デバイスの最終デバイス番号までの間に、00Hが設定されていないとき。
3281H	(s)に変換できない不正なデータを入力したとき。 • 文字数が、2~8以外。 • 小数部文字数が、0~5以外。 • 全文字数と、小数部文字数の関係が、下記範囲外。 全文字数-3≥小数部文字数 • VAL(P)命令使用時、符号に20H、2DH以外のアスキーコードが設定されている。 • VAL(P)_U命令使用時、符号に20H以外のアスキーコードが設定されている。 • 各数字の桁に30H~39H、および2EH(小数点)以外のアスキーコードが設定されている。 • 小数点が複数設定されている。 変換したBIN値の値が、各命令で変換できる範囲を超えたとき。
3285H	(s)で指定する文字列の文字数が、16383文字を超えるととき。

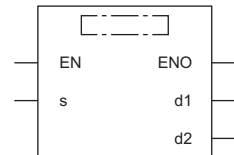
10進文字列→BIN32ビットデータ変換

DVAL(P)(_U)

文字列をBIN32ビットデータに変換します。

ラダー	ST	
	ENO:=DVAL(EN,s,d1,d2); ENO:=DVALP(EN,s,d1,d2);	ENO:=DVAL_U(EN,s,d1,d2); ENO:=DVALP_U(EN,s,d1,d2);

FBD/LD



■実行条件

命令	実行条件
DVAL DVAL_U	
DVALP DVALP_U	

設定データ

■内容、範囲、データ型

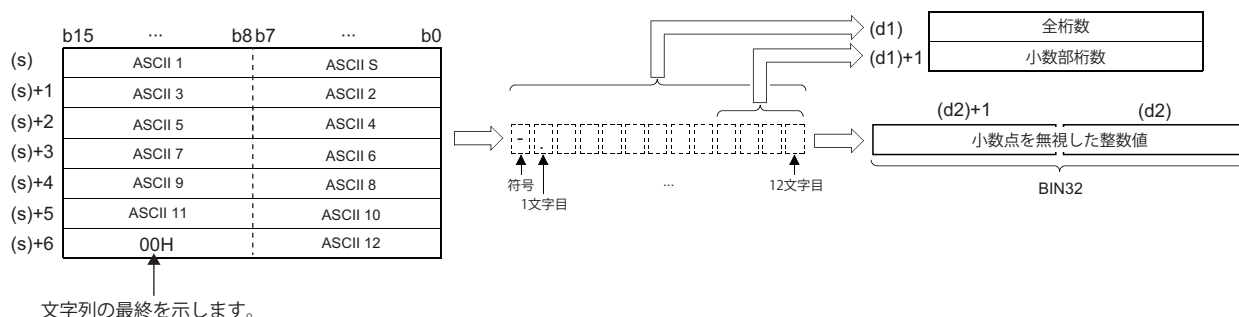
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータに変換する文字列または、文字列が格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d1)	変換後のBINデータの桁数を格納する先頭デバイス	—	符号付きBIN16ビット	ANY16_S_ARRAY (要素数: 2)
			符号なしBIN16ビット	ANY16_U_ARRAY (要素数: 2)
(d2)	変換後のBINデータを格納する先頭デバイス	—	符号付きBIN32ビット	ANY32_S
			符号なしBIN32ビット	ANY32_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d1)	○	—	○	—	—	—	—	○	—	—	—	—
(d2)	○	○	○	○	○	○	○	○	—	—	—	—

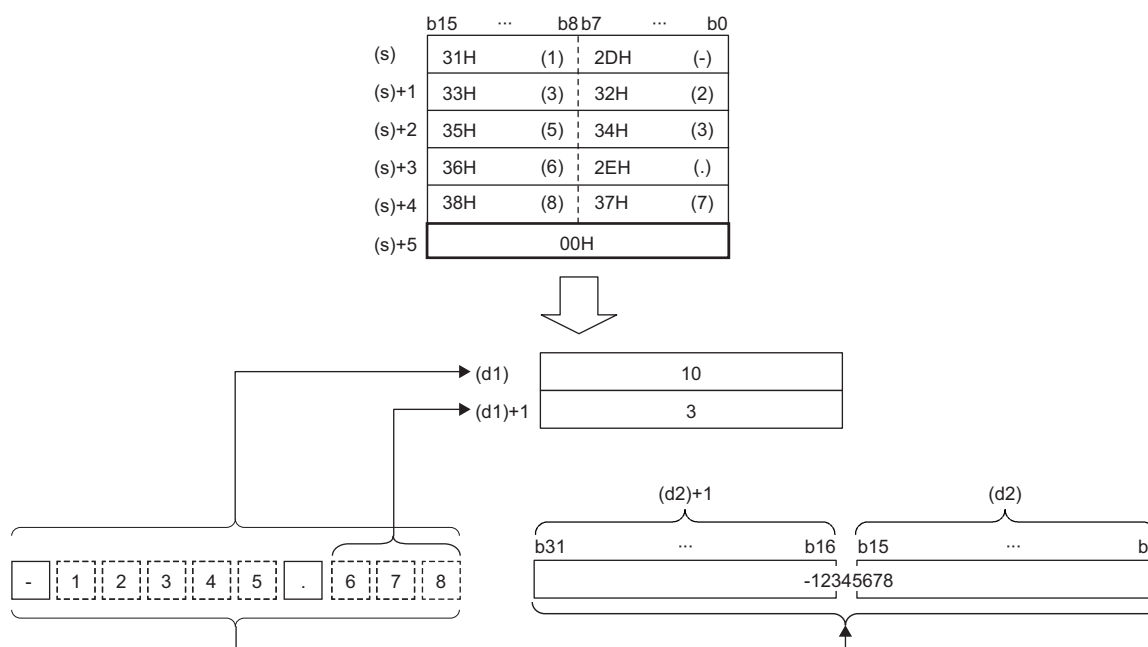
機能

- (s)で指定されたデバイス番号以降に格納されている文字列を、BIN32ビットデータに変換して、桁数を(d1)に、BINデータを(d2)に格納します。文字列→BIN変換では、(s)で指定されたデバイス番号から、00Hが格納されているデバイス番号までのデータを、文字列として処理します。



例

(s)以降に「-12345.678」の文字列が指定された場合(符号付きを指定した場合)

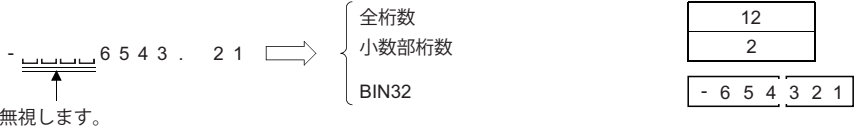


- (s)で指定する文字列の全文字数は、2~13文字です。
- (s)で指定する文字列のうち、小数部となる文字数は、0~10文字です。ただし、「全桁数-3」以下になるようにしてください。
- BIN値に変換できる数値の文字列の範囲は下記になります。また、符号と小数点を除く数値の文字列は、30H~39Hの範囲内のみ指定できます。(小数点を無視した値...「-12345.6」の場合は、「-123456」となります。)
- 小数点を無視した値で符号付きを指定した場合: -2147483648~2147483647
- 小数点を無視した値で符号なしを指定した場合: 0~4294967295
- 符号には、正の数値を表す場合は20Hを設定し、負の数値を表す場合は2DHを設定します。
- 小数点には、2EHを設定します。
- (d1)に格納する全桁数は、数値を表すすべての文字(符号、小数点を含む)の数を格納します。(d1)+1に格納する小数部桁数は、2EH(.)以降の小数部を表す文字数を格納します。(d2)に格納するBIN32ビットデータは、小数点を無視した文字列をBIN値に変換して格納します。

- ・ (s)で指定された文字列で、符号と最初の0以外の数値との間に、20H(スペース)か30H(0)が存在する場合は、20H、30Hを無視してBIN値に変換します。

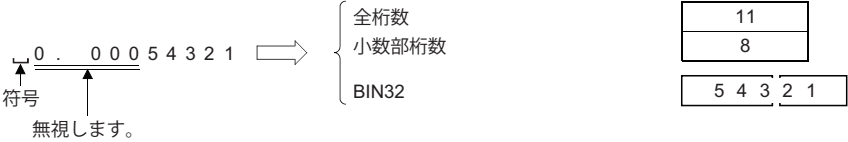
例

符号と最初の0以外の数値との間に20Hが存在する場合(符号付きを指定した場合)



例

符号と最初の"0"以外の数値との間に30Hが存在する場合



エラー

エラーコード (SD0)	内容
2820H	(s)で指定したデバイス番号から、該当デバイスの最終デバイス番号までの間に、00Hが設定されていないとき。
3281H	(s)に変換できない不正なデータを入力したとき。 ・ 文字列の文字数が、2~13以外。 ・ 文字列の小数部文字数が、0~10以外。 ・ 全文字数と、小数部文字数の関係が、下記範囲外。 全文字数-3≧小数部文字数 ・ DVAL(P)命令使用時、符号に20H、2DH以外のアスキーコードが設定されている。 ・ DVAL(P)_U命令使用時、符号に20H以外のアスキーコードが設定されている。 ・ 各数字の桁に30H~39H、および2EH(小数点)以外のアスキーコードが設定されている。 ・ 小数点が複数設定されている。 変換したBIN値の値が、各命令で変換できる範囲を超えたとき。
3285H	(s)で指定する文字列の文字数が、16383文字を超えたとき。

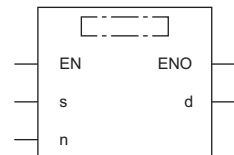
16進アスキー→16進データBIN変換

ASC2INT(P)

16進アスキーデータを，BIN値に変換する。

ラダー	ST
	ENO:=ASC2INT(EN,s,n,d); ENO:=ASC2INTP(EN,s,n,d);

FBD/LD



■実行条件

命令	実行条件
ASC2INT	
ASC2INTP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	BINデータに変換する文字列が格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE ^{*1}
(d)	変換後のBINデータを格納する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(n)	格納する文字数	0~16383	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

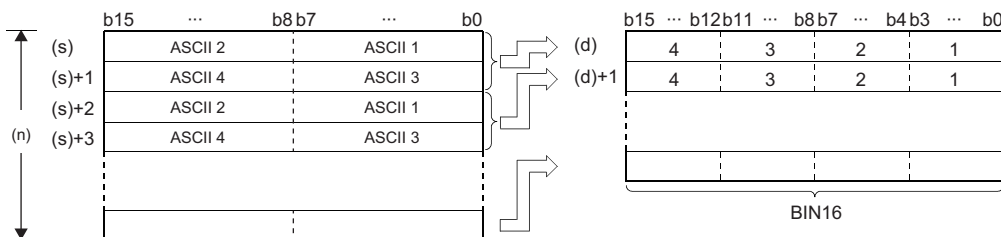
*1 ラベルで設定する場合は，動作に必要な領域が確保できるよう配列を定義し，その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- (s)で指定されたデバイス番号以降、(n)で指定された文字数分に格納されている16進アスキーデータをBIN値に変換して、(d)で指定されたデバイス番号以降に格納します。

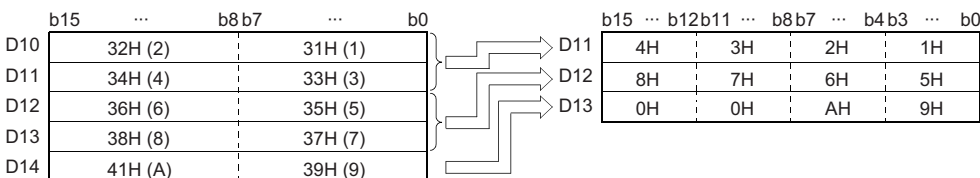


ASCII □: □桁目のアスキーコード

1~4: 1~4桁目

(n): (n)で指定された文字数分

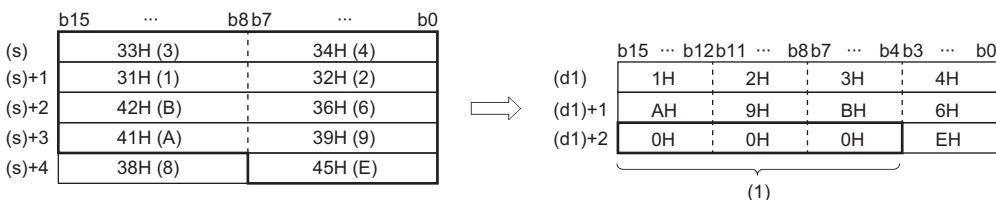
- (n)に文字数を設定すると、(s)で指定する文字列の範囲、および(d)で指定するBINデータを格納するデバイスの範囲が自動的に決まります。
- 変換するアスキーデータが格納されているデバイス範囲と、変換したBINデータを格納するデバイス範囲が重複している場合でも、正常に処理します。



- (n)で指定された文字数が4の倍数でない場合、変換したBIN値を格納するデバイス番号のうち、最終デバイス番号の指定文字数分以降の桁数には、自動的に0Hを格納します。

例

(n)の文字数が9の場合



(1): 自動的に0Hを格納します。

- (n)で指定する文字数が「0」の場合は、変換処理を行いません。
- (s)で指定できるアスキーコードは30H~39H, 41H~46Hの範囲内です。

エラー

エラーコード (SD0)	内容
3281H	(s)に16進数値の文字列以外の文字(30H~39H, 41H~46H以外のアスキーコード)が設定されている。
3285H	(n)に指定可能な範囲外のデータを入力したとき。 • 指定した文字数が0~16383以外。

単精度実数→BCD分解

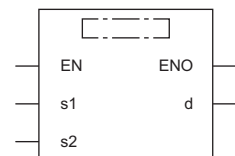
12

EMOD(P)

単精度実数データをBCD型浮動小数点フォーマットに分解します。

ラダー	ST
	ENO:=EMOD(EN,s1,s2,d); ENO:=EMODP(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
EMOD	
EMODP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	単精度実数データまたは，単精度実数データが格納されている先頭デバイス	$0, 2^{-126} \leq (s1) < 2^{128}$	単精度実数	ANYREAL_32
(s2)	小数部桁データ	0~7	符号付きBIN16ビット	ANY16
(d)	BCD分解したデータを格納する先頭デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 5)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	○	○	○	○	○	—	○	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

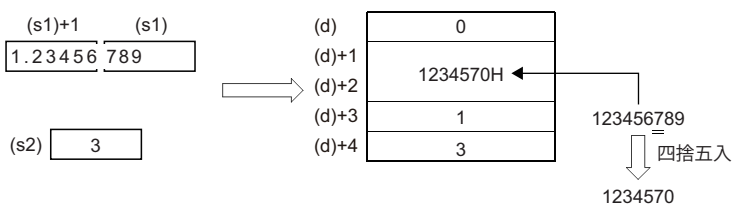
- (s1)で指定されたデバイスに格納されている単精度実数データを、(s2)で指定されたデバイスに格納されている小数部桁数に基づいてBCD型浮動小数点フォーマットに分解して、(d)で指定されたデバイス番号以降に格納します。



- (d)の符号および(d)+3の指数部符号には、正のとき0、負のとき1が格納されます。
- (d)+4のBCD指数には、0~38が格納されます。
- (s2)は(s1)の単精度実数データの小数部桁数を指定します。上図の例では、下記のようになります。

3.25427
 \uparrow
 (s2)=3

- (d)+1, (d)+2に格納されているBCDの有効桁は7桁目を四捨五入した6桁です。



- (s2)の小数部桁数には、0~7が設定できます。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

39ページ 注意事項

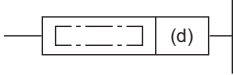
エラー

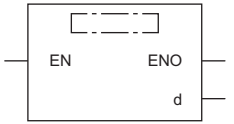
エラーコード (SD0)	内容
3281H	(s2)で指定した小数部桁数が0~7以外るとき。
3282H	(s1)に設定したデバイス、ラベルの内容が-0、非正規化数、非数、±∞のとき。

BIN16ビットデータ2の補数(符号反転)

NEG(P)

BIN16ビットデバイスの符号を反転します。

ラダー	ST
	$ENO := NEG(EN, d);$ $ENO := NEGP(EN, d);$

FBD/LD


■実行条件

命令	実行条件
NEG	
NEGP	

設定データ

■内容、範囲、データ型

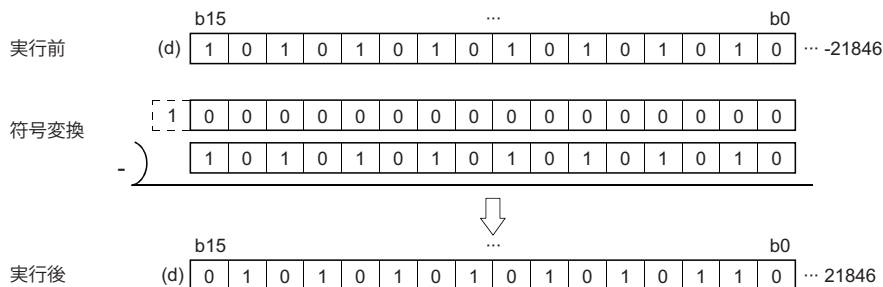
オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	2の補数を行うデータが格納されているデバイス	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □		LT, LST, LC	LZ		K, H	E	\$	
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- (d)で指定されたBIN16ビットデバイスの符号を反転して、(d)で指定されたデバイスに格納します。
- 正負の符号を反転するときに使用します。



エラー

演算エラーはありません。

BIN32ビットデータ2の補数(符号反転)

DNEG(P)

BIN32ビットデバイスの符号を反転します。

ラダー	ST
	ENO:=DNEG(EN,d); ENO:=DNEGP(EN,d);

FBD/LD

■実行条件

命令	実行条件
DNEG	
DNEGP	

設定データ

■内容, 範囲, データ型

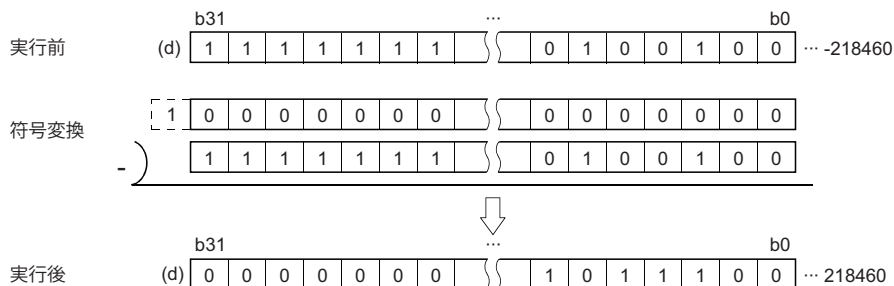
オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	2の補数を行うデータが格納されている先頭デバイス	—	符号付きBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □		LT, LST, LC	LZ		K, H	E	\$	
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (d)で指定されたBIN32ビットデバイスの符号を反転して, (d)で指定されたデバイスに格納します。
- 正負の符号を反転するときに使用します。



エラー

演算エラーはありません。

8→256ビットデコード

12

DECO(P)

指定されたデバイスの下位(n)ビットをデコードします。

ラダー	ST
	ENO:=DECO(EN,s,n,d); ENO:=DECOP(EN,s,n,d);

FBD/LD

■実行条件

命令	実行条件
DECO	
DECOP	

設定データ

■内容、範囲、データ型

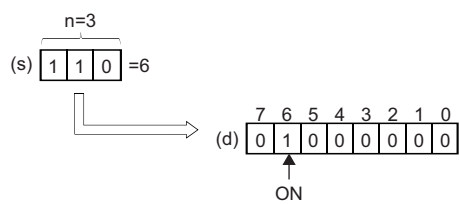
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	デコードデータまたは、デコードデータが格納されているデバイス番号	—	ビット/符号付きBIN16ビット	ANY_ELEMENTARY
(d)	デコード結果を格納するデバイス	—	ビット/ワード	ANY_ELEMENTARY
(n)	有効ビット長	1~8	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□	G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	○	—	—	○	○	—	—	—

機能

- (s)の下位(n)ビットで指定されたBIN値に対応した、(d)のビット位置をONします。



- (n)は1~8が指定できます。
- (n)=0のときは無処理となり、(d)で指定されたデバイスの内容は変化しません。
- ビットデバイスは1ビット、ワードデバイスは16ビットとして扱われます。

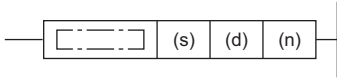
エラー

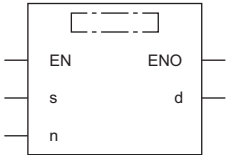
エラーコード (SD0)	内容
3281H	(n)が0~8以外のとき。

256→8ビットエンコード

ENCO(P)

2の(n)乗ビットのデータをエンコードします。

ラダー	ST
	ENO:=ENCO(EN,s,n,d); ENO:=ENCOP(EN,s,n,d);

FBD/LD


■実行条件

命令	実行条件
ENCO	
ENCOP	

設定データ

■内容, 範囲, データ型

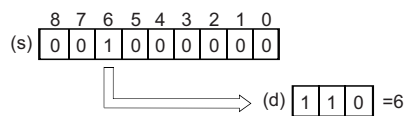
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	エンコードデータが格納されているデバイス	—	ビット/ワード	ANY_ELEMENTARY
(d)	エンコード結果を格納するデバイス番号	—	ビット/符号付きBIN16 ビット	ANY_ELEMENTARY
(n)	有効ビット長	1~8	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	○	—	○	—	—	—	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—
(n)	○	○	○	○	○	—	—	○	—	—	—

機能

- (s)の $2^{(n)}$ ビットのデータから1になっているビットに対応したBIN値を、(d)に格納します。



- (n)は1~8が指定できます。
- (n)=0のときは無処理となり、(d)で指定されたデバイスの内容は変化しません。
- ビットデバイスは1ビット、ワードデバイスは16ビットとして扱われます。
- 複数のビットが1のときは上位のビット位置で処理されます。

エラー

エラーコード (SD0)	内容
3281H	(n)が0~8以外するとき。
	(s)から $2^{(n)}$ ビット分のデータがすべて0のとき。

7セグメントデコード

SEG(P)

デバイスの下位4ビットで指定された0~Fのデータを7セグメント表示データにデコードします。

ラダー	ST
	ENO:=SEG(EN,s,d); ENO:=SEGP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
SEG	
SEGP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	デコードデータまたは, デコードデータが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(d)	デコード結果を格納するデバイス	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

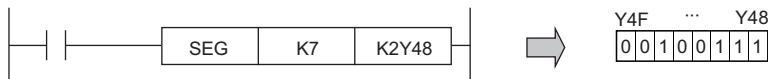
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- (s)の下位4ビットで指定された0~Fのデータを，7セグメント表示データにデコードして，(d)に格納します。
- ビットデバイスするとき(d)は，7セグメント表示データを格納する先頭デバイスを示し，ワードデバイスでは格納するデバイス番号を示します。

例

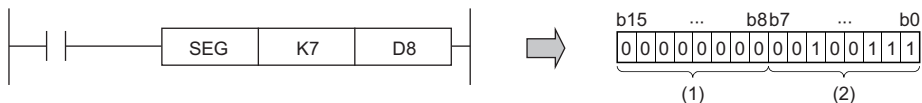
ビットデバイスの場合



Y48~Y4Fは次のデータが出力されるまで変化しません。

例

ワードデバイスの場合



(1): 上位8ビットは必ず0になります。

(2): 下位8ビットに7セグメント表示データを格納します。

- 7セグメントデコード表は下記になります。

(s)		7セグメントの構成	(d)								表示データ
16進数	ビットパターン		b7	b6	b5	b4	b3	b2	b1	b0	
0	0000		0	0	1	1	1	1	1	1	0
1	0001		0	0	0	0	0	1	1	0	1
2	0010		0	1	0	1	1	0	1	1	2
3	0011		0	1	0	0	1	1	1	1	3
4	0100		0	1	1	0	0	1	1	0	4
5	0101		0	1	1	0	1	1	0	1	5
6	0110		0	1	1	1	1	1	0	1	6
7	0111		0	0	1	0	0	1	1	1	7
8	1000		0	1	1	1	1	1	1	1	8
9	1001		0	1	1	0	1	1	1	1	9
A	1010		0	1	1	1	0	1	1	1	A
B	1011		0	1	1	1	1	1	0	0	b
C	1100		0	0	1	1	1	0	0	1	c
D	1101		0	1	0	1	1	1	1	0	d
E	1110		0	1	1	1	1	0	0	1	e
F	1111		0	1	1	1	0	0	0	1	f

エラー

演算エラーはありません。

12

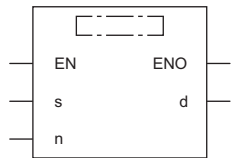
BIN16ビットデータの4ビット分離

DIS(P)

指定されたBIN16ビットデータの下位(n)桁分のデータを指定されたデバイスに格納します。

ラダー	ST
	ENO:=DIS(EN,s,n,d); ENO:=DISP(EN,s,n,d);

FBD/LD



■実行条件

命令	実行条件
DIS	
DISP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	分離するデータが格納されているデバイス	—	符号付きBIN16ビット	ANY16
(d)	分離されたデータを格納する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(n)	分離数	1~4	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

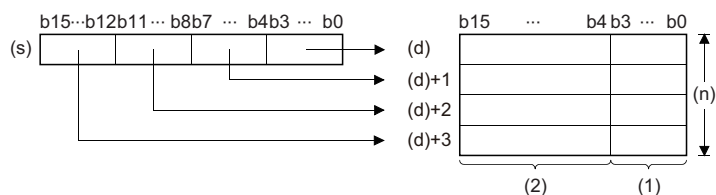
*1 ラベルで設定する場合は，動作に必要な領域が確保できるよう配列を定義し，その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

機能

- (s)で指定されたBIN16ビットデータの低位(n)桁(1桁4ビット)分のデータを、(d)で指定されたデバイスから(n)点分の低位4ビットに格納します。



(1) 格納エリア

(2) 0が格納されます。

- (d)で指定されたデバイスから(n)点分の上位12ビットは0になります。
- (n)は1~4が指定できます。
- (n)=0のときは無処理となり、(d)のデバイスから(n)点の内容は変化しません。

エラー

エラーコード (SD0)	内容
3281H	(n)が0~4以外のとき。

BIN16ビットデータの4ビット結合

UNI(P)

指定されたデバイスから(n)点分のBIN16ビットデータの低位4ビットを16ビットデバイスに結合します。

ラダー	ST
	ENO:=UNI(EN,s,n,d); ENO:=UNIP(EN,s,n,d);

FBD/LD

■実行条件

命令	実行条件
UNI	
UNIP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	結合するデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(d)	結合されたデータを格納するデバイス	—	符号付きBIN16ビット	ANY16
(n)	結合数	1~4	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

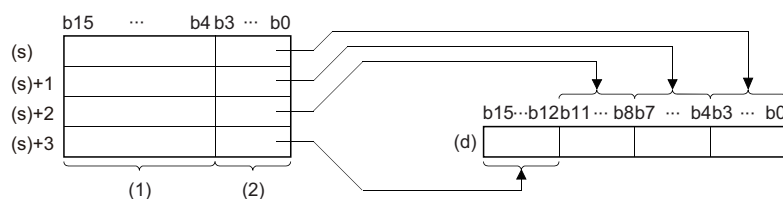
*1 ラベルで設定する場合は，動作に必要な領域が確保できるよう配列を定義し，その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- (s)で指定されたデバイスから(n)点分のBIN16ビットデータの下位4ビットを、(d)で指定されたBIN16ビットデバイスに結合します。



(1): 無視されます。

(2): 結合データ

- (d)で指定されたデバイスの上位(4-n)の桁のビットは0になります。
- (n)は1~4が指定できます。
- (n)=0のときは無処理となり、(d)のデバイスの内容は変化しません。

エラー

エラーコード (SD0)	内容
3281H	(n)が0~4以外のとき。

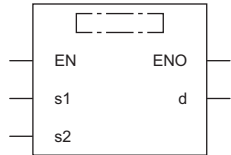
任意データのビット分離

NDIS(P)

指定データの各ビットを，任意のビットずつに分離します。

ラダー	ST
	ENO:=NDIS(EN,s1,s2,d); ENO:=NDISP(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
NDIS	
NDISP	

設定データ

■内容，範囲，データ型

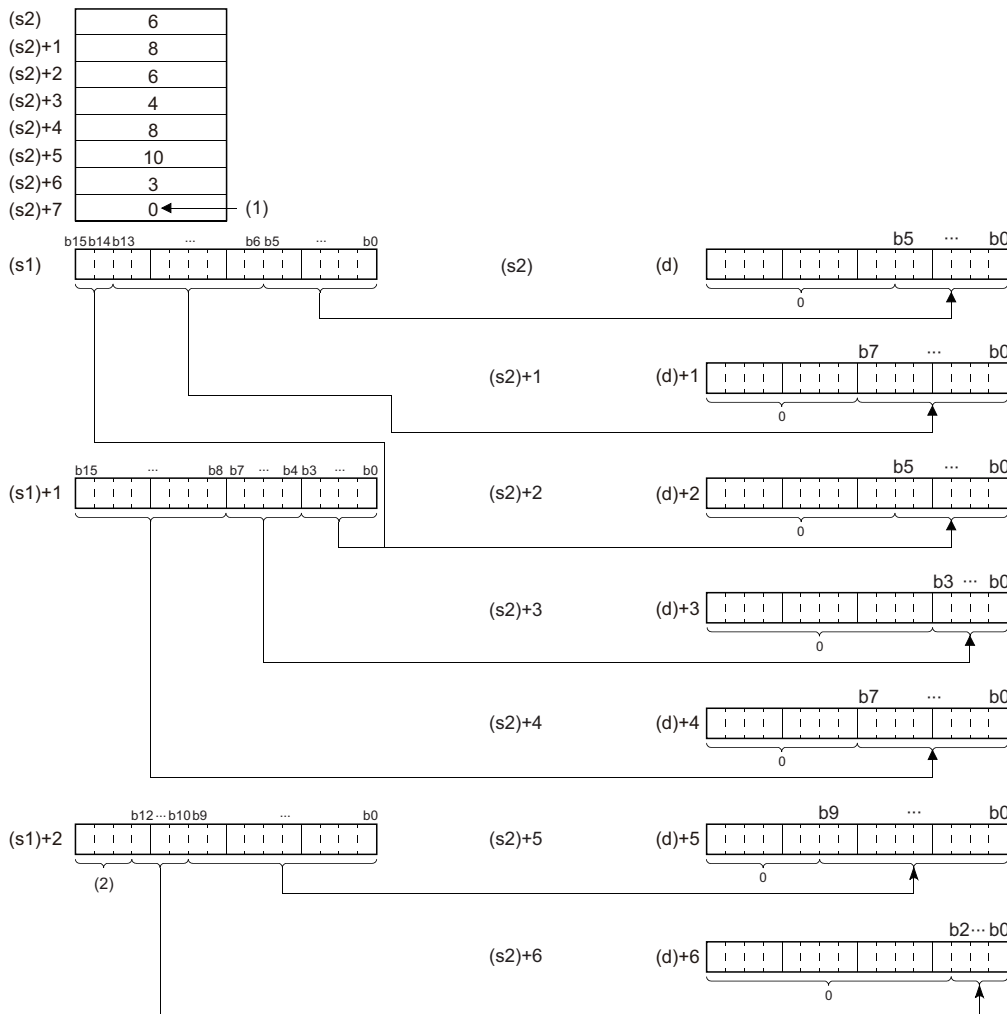
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	分離するデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(d)	分離したデータを格納する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(s2)	分離単位を格納する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は，動作に必要な領域が確保できるよう配列を定義し，その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—

- (s1)で指定されたデバイス番号以降に格納しているデータの各ビットを、(s2)で指定されたビット分ずつ分離して、(d)で指定されたデバイス番号以降に格納します。



(s2)~(s2)+6: (s2)~(s2)+6の指定ビット分

(1): 設定終了指定

(2): 無視されます。

- (s2)で指定する分離ビット数は、1~16ビットの範囲内で指定することができます。
- (s2)による指定デバイス番号から、「0」を格納しているデバイス番号までを分離ビット数として処理します。
- 分離するデータのデバイス範囲((s1)~(s1)の終了範囲)と、分離したデータを格納するデバイス範囲((d)~(d)の終了範囲)は、重複しないようにしてください。重複した場合、正しい演算結果が得られないことがあります。
- (s1), (s2), (d)で指定するデバイス番号は、重複させないでください。

エラー

エラーコード (SD0)	内容
2821H	(s1), (s2)のデバイスが重複しているとき。 (s1), (d)のデバイスが重複しているとき。 (s2), (d)のデバイスが重複しているとき。
3281H	(s2)に変換できない不正なデータを入力したとき。 ・ 指定した分離ビット数指定が1~16ビットの範囲内で設定されていない。 ・ 指定したデバイスから、該当デバイス/ラベルの割付範囲を超えるまでの間に、0が設定されていない。

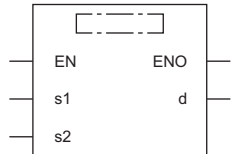
任意データのビット結合

NUNI(P)

指定データの各ビットを，任意のビット分ずつ結合します。

ラダー	ST
	ENO:=NUNI(EN,s1,s2,d); ENO:=NUNIP(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
NUNI	
NUNIP	

設定データ

■内容，範囲，データ型

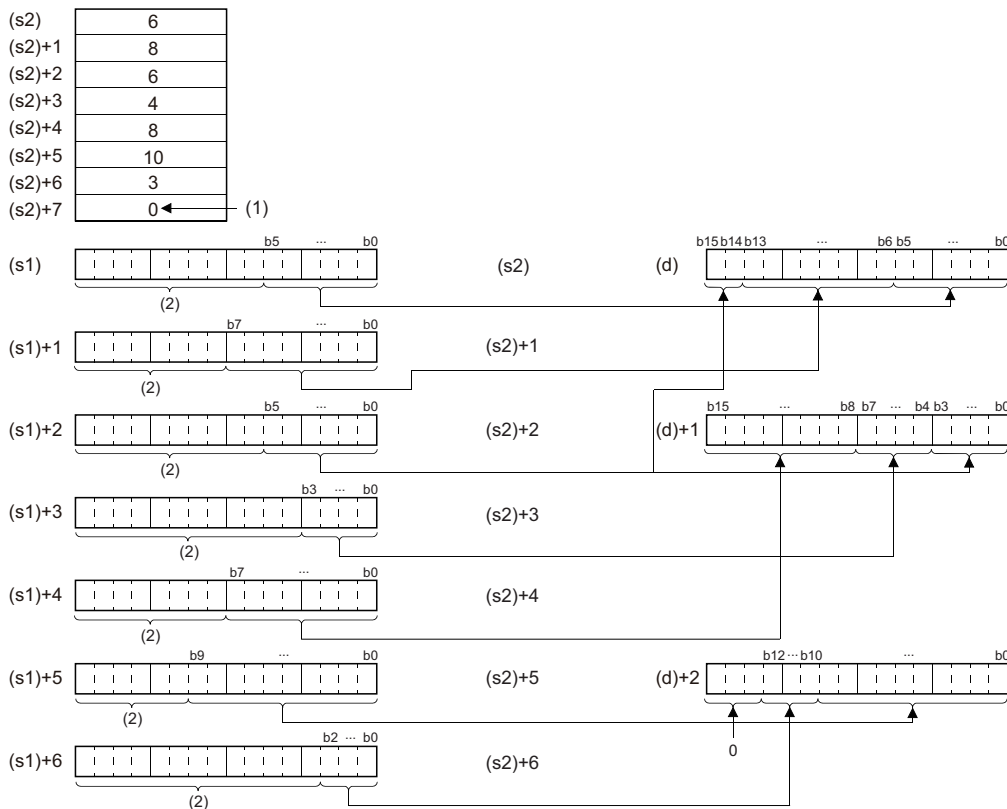
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	結合するデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(d)	結合したデータを格納する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(s2)	結合単位を格納する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は，動作に必要な領域が確保できるよう配列を定義し，その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	—

- (s1)で指定されたデバイス番号以降に格納しているデータの各ビットを、(s2)で指定されたビット分ずつ結合して、(d)で指定されたデバイス番号以降に格納します。



(s2)~(s2)+6: (s2)~(s2)+6の指定ビット分

- (1): 設定終了指定
(2): 無視されます。

- (s2)で指定する結合ビット数は、1~16ビットの範囲内で指定することができます。
- (s2)による指定デバイス番号から、「0」を格納しているデバイス番号までを結合ビット数として処理します。
- 結合するデータのデバイス範囲((s1)~(s1)の終了範囲)と、結合したデータを格納するデバイス範囲((d)~(d)の終了範囲)は、重複しないようにしてください。重複した場合、正しい演算結果が得られないことがあります。
- (s1), (s2), (d)で指定するデバイス番号は、重複させないでください。

エラー

エラーコード (SD0)	内容
2821H	(s1), (s2)のデバイスが重複しているとき。 (s1), (d)のデバイスが重複しているとき。 (s2), (d)のデバイスが重複しているとき。
3281H	(s2)に変換できない不正なデータを入力したとき。 - 指定した結合ビット数指定が1~16ビットの範囲内で設定されていない。 - 指定したデバイスから、該当デバイス/ラベルの割付範囲を超えるまでの間に、0が設定されていない。

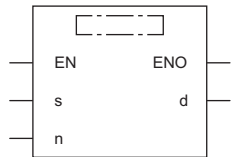
バイト単位データ分離

WTOB(P)

BIN16ビットデータを，(n)バイトに分離します。

ラダー	ST
	ENO:=WTOB(EN,s,n,d); ENO:=WTOBP(EN,s,n,d);

FBD/LD



■実行条件

命令	実行条件
WTOB	
WTOBP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	バイト単位で分離するデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(d)	バイト単位で分離した結果を格納する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(n)	分離するバイトデータの個数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

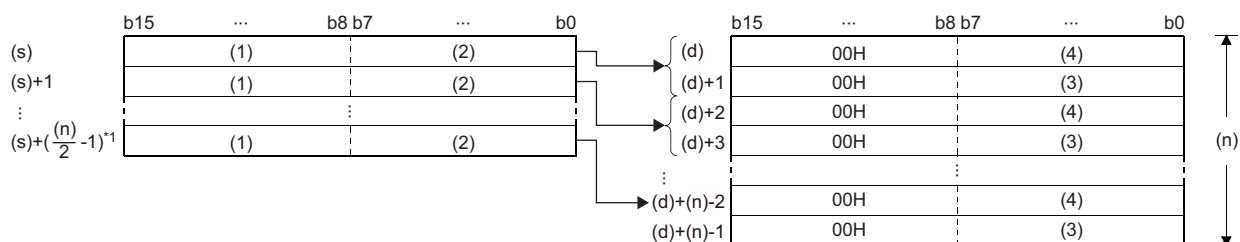
*1 ラベルで設定する場合は，動作に必要な領域が確保できるよう配列を定義し，その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- ・ (s)で指定されたデバイス番号以降に格納されているBIN16ビットデータを, (n)バイトに分離して, (d)で指定されたデバイス番号以降に格納します。

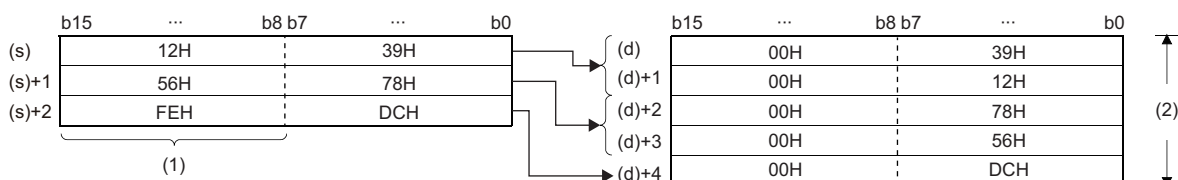


- (1): 上位バイト
 (2): 下位バイト
 (3): 上位バイトのデータ
 (4): 下位バイトのデータ

*1 小数点以下は切り上げになります。

例

(n)=5の場合は, (s)~(s)+2の下位8ビットまでのデータを(d)~(d)+4に格納します。



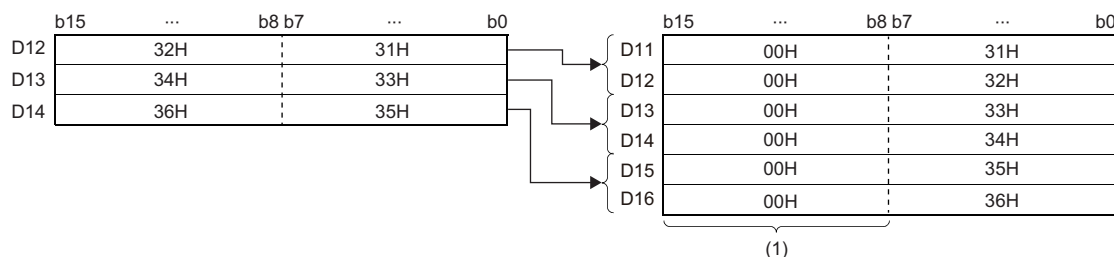
(1): (n)=5の場合は無視されます。

(2): (n)=5の場合

- ・ (n)でバイト数を設定することにより, (s)で指定するBIN16ビットデータの範囲, および(d)で指定するバイトデータを格納するデバイスの範囲が自動的に決まります。
- ・ (n)で指定するバイト数が0の場合は, 処理を行いません。
- ・ (d)で指定されたバイトデータ格納デバイスの上位8ビットには, 自動的に00Hを格納します。

例

D12~D14をD11~D16の下位8ビットに格納する場合



(1): 自動的に00Hを格納します。

- ・ 分離するデータが格納されているデバイス範囲と, 分離するデータを格納するデバイス範囲が重複している場合でも, 正常に処理します。

分離するデータが格納されているデバイス範囲	分離するデータを格納するデバイス範囲
$(s) \sim (s) + (\frac{n}{2}) - 1$	$(d) + 0 \sim (d) + (n) - 1$

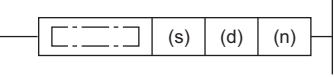
エラー

演算エラーはありません。

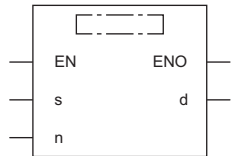
バイト単位データ結合

BTOW(P)

BIN16ビットデータの低位8ビットを，ワード単位に結合します。

ラダー	ST
	ENO:=BTOW(EN,s,n,d); ENO:=BTOWP(EN,s,n,d);

FBD/LD



■実行条件

命令	実行条件
BTOW	
BTOWP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	バイト単位で結合するデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(d)	バイト単位で結合した結果を格納する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(n)	結合するバイトデータの個数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

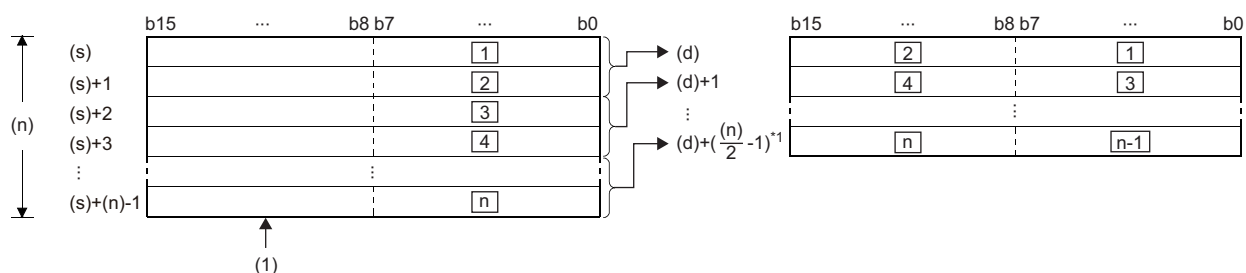
*1 ラベルで設定する場合は，動作に必要な領域が確保できるよう配列を定義し，その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□≠□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□≠G□, J□≠□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	—	—	○	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

機能

- (s)で指定されたデバイス番号以降に格納されている、(n)ワード分のBIN16ビットデータの下位8ビットを、ワード単位に結合して、(d)で指定されたデバイス番号以降に格納します。
- (s)で指定されたデバイス番号以降に格納されている、(n)ワードのデータの上位8ビットは無視されます。また(n)が奇数の場合は、(n)バイト目のデータを格納したデバイスの上位8ビットに0を格納します。



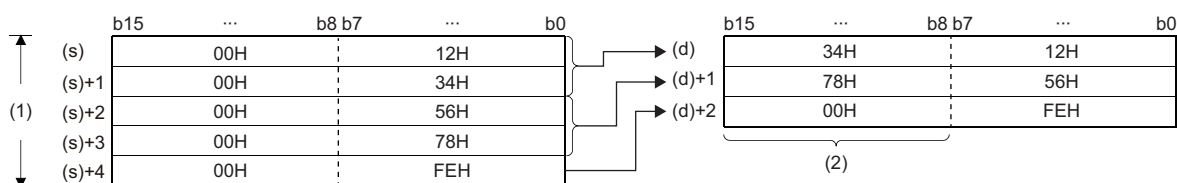
□: 0バイト目のデータ

(1) 上位バイトは無視します。

*1 小数点以下は切り上げになります。

例

(n)=5の場合は、(s)~(s)+4の下位8ビットまでのデータを結合し(d)~(d)+2に格納します。



(1) (n)=5の場合

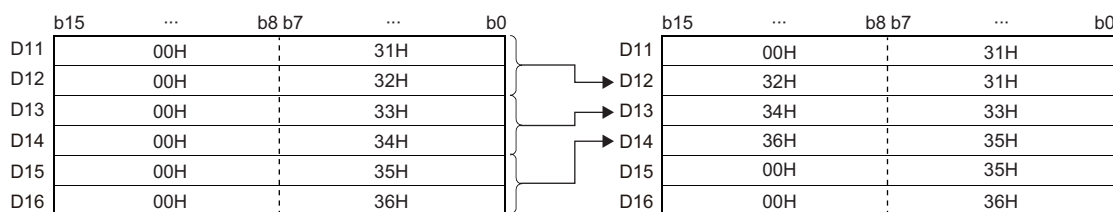
(2) 00Hが格納されます。

- (n)でバイト数を設定することにより、(s)で指定するバイトデータの範囲、および(d)で指定する結合データを格納するデバイスの範囲が自動的に決まります。
- (n)で指定するバイト数が0の場合は、処理を行いません。
- (s)で指定されたバイトデータ格納デバイスの上位8ビットを無視して、下位8ビットが対象となります。
- 結合するデータが格納されているデバイス範囲と、結合されたデータを格納するデバイス範囲が重複している場合でも、正常に処理します。

結合するデータが格納されているデバイス範囲	結合されたデータを格納するデバイス範囲
(s)+0~(s)+(n)-1	(d)~(d)+(n/2-1)

例

D11~D16の下位8ビットをD12~D14に格納する場合



エラー

演算エラーはありません。

12.7 データ転送命令

BIN16ビットデータ転送

MOV(P)

指定されたデバイスのBIN16ビットデータを転送します。

ラダー	ST
	ENO:=MOV(EN,s,d); ENO:=MOV(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
MOV	
MOV(EN,s,d)	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	転送元のデータまたは、データが格納されているデバイス番号	-32768~32767	符号付きBIN16ビット	ANY16
(d)	転送先のデバイス番号	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口		LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

BIN32ビットデータ転送

DMOV(P)

指定されたデバイスのBIN32ビットデータを転送します。

ラダー	ST
	ENO:=DMOV(EN,s,d); ENO:=DMOV(EN,s,d)

FBD/LD

■実行条件

命令	実行条件
DMOV	
DMOV(P)	

設定データ

■内容，範囲，データ型

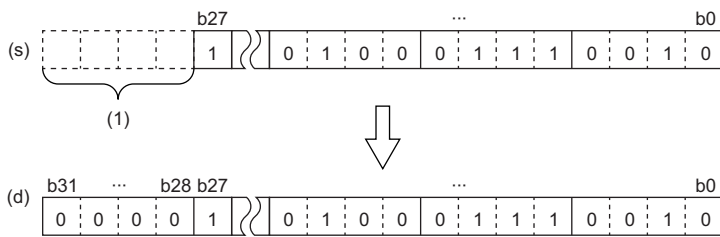
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	転送元のデータまたは，データが格納されているデバイス番号	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(d)	転送先のデバイス番号	—	符号付きBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定されたBIN32ビットデータを、(d)で指定されたデバイスに転送します。
- (s)が桁指定したビットデバイスの場合、桁指定で指定されたビットが転送の対象になります。転送対象外のビットには0が指定されているものとして転送します。



(1): (s)が32ビット未満の場合、0が指定されているものとして転送します。

エラー

演算エラーはありません。

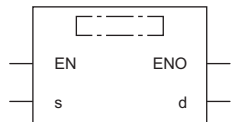
BIN16ビットデータ否定転送

CML(P)

指定されたBIN16ビットデータをビットごとに反転し、転送します。

ラダー	ST
	ENO:=CML(EN,s,d); ENO:=CMLP(EN,s,d);

FBD/LD



■実行条件

命令	実行条件
CML	
CMLP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	反転させるデータまたは、データが格納されているデバイス番号	-32768~32767	符号付きBIN16ビット	ANY16
(d)	反転結果を格納するデバイス番号	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

BIN32ビットデータ否定転送

DCML(P)

指定されたBIN32ビットデータをビットごとに反転し、転送します。

ラダー	ST
	ENO:=DCML(EN,s,d); ENO:=DCMLP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
DCML	
DCMLP	

設定データ

■内容，範囲，データ型

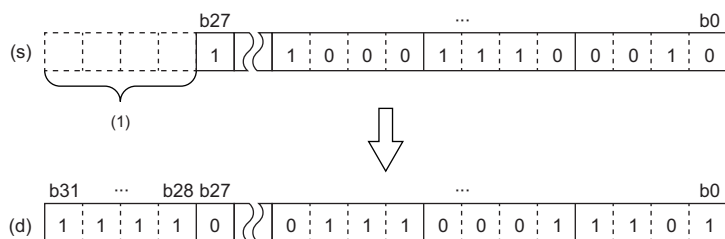
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	反転させるデータまたは，データが格納されているデバイス番号	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(d)	反転結果を格納するデバイス番号	—	符号付きBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

性能

- (s)で指定されたBIN32ビットデータをビットごとに反転して、その結果を(d)で指定されたデバイスに転送します。
- (s)が桁指定したビットデバイスの場合、桁指定で指定されたビットが対象になります。対象外のビットには0が指定されているものとして反転して、転送します。



(1): (s)が32ビット未満の場合、0が指定されているものとして反転します。

1)

演算エラーはありません。

1ビットデータ反転転送

CMLB(P)

指定したビットデータを反転し、転送します。

ラダー	ST
	ENO:=CMLB(EN,s,d); ENO:=CMLBP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
CMLB	
CMLBP	

設定データ

■内容，範囲，データ型

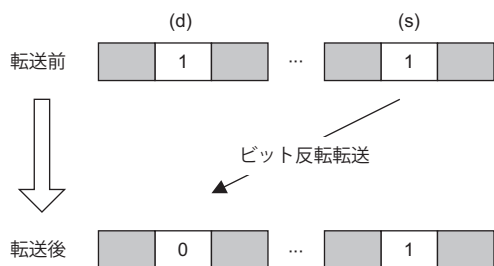
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	反転させるデータまたは、データが格納されているデバイス番号	—	ビット	ANY_BOOL
(d)	転送先のデバイス番号	—	ビット	ANY_BOOL
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	—	—	○	—	○	—	—	—	—
(d)	○	○	○	—	—	○	—	○	—	—	—	—

機能

- (s)で指定されたビットデータを反転し、その結果を(d)で指定されたデバイスへ格納します。



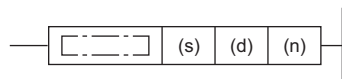
エラー

演算エラーはありません。

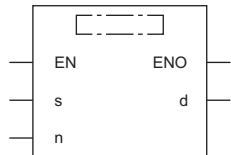
BIN16ビットデータブロック転送(16ビット)

BMOV(P)

指定されたデバイスから(n)点(0~65535)のBIN16ビットデータを一括転送します。

ラダー	ST
	ENO:=BMOV(EN,s,n,d); ENO:=BMOV(P)(EN,s,n,d);

FBD/LD



■実行条件

命令	実行条件
BMOV	
BMOV(P)	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	転送するデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1*2}
(d)	転送先の先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1*2}
(n)	転送数	0~65535	符号なしBIN16ビット	ANY16 ^{*2}
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

*2 INT型/WORD型以外のデータ型も使用できます。(BOOL型/POINTER型/STRUCT型(構造体)を除く)なお、オプション設定の“命令引数のデータ型チェックを行う”で“はい”を選択すると、使用できるデータ型はINT型/WORD型(ANY16の範囲)のみとなります。

オプション設定については、下記を参照してください。

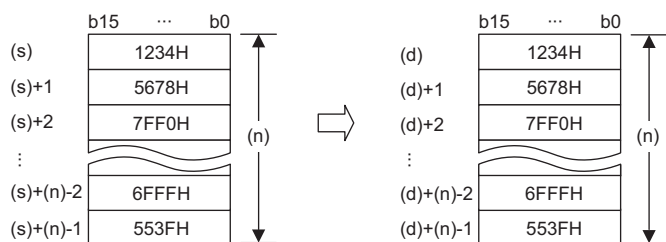
 GX Works3 オペレーティングマニュアル

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	—	○	—	○	—	—	—	—
(d)	○	○	○	○	—	○	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

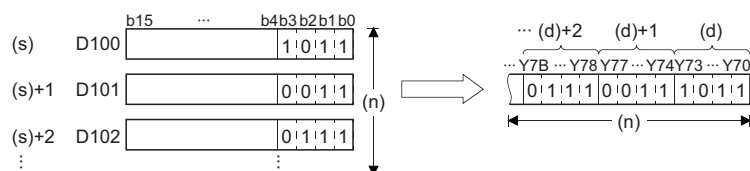
- (s)で指定されたデバイスから(n)点分のBIN16ビットデータを、(d)で指定されたデバイスへ一括転送します。



- 転送元と転送先のデバイスが重複している場合も転送を行うことができます。デバイス番号の小さい方への転送の場合は (s)から、デバイス番号の大きい方への転送の場合は(s)+(n)-1から転送します。
- (s)がワードデバイスで(d)がビットデバイスの場合、ワードデバイスはビットデバイスの桁指定で指定されたビット数が対象になります。

例

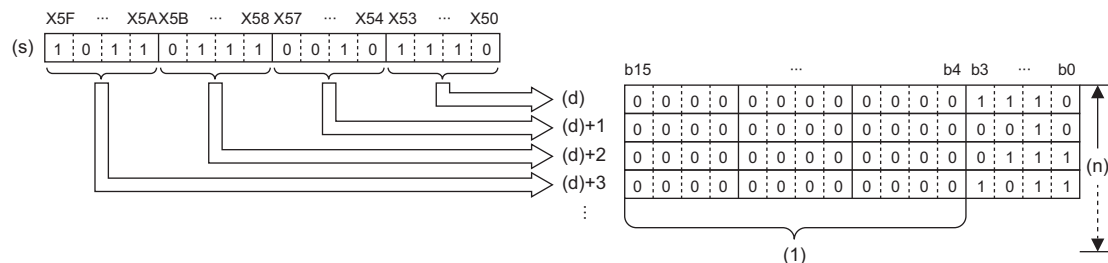
(d)にK1Y70が指定された場合、(s)で指定されたワードデバイスの下位4ビットが対象になります。



- (s)が桁指定したビットデバイスで、(d)がワードデバイスの場合、ワードデバイスはビットデバイスの桁指定で指定されたビット数が対象になります。

例

(s)にK1X50が指定された場合は、(d)のワードデバイスの下位4ビットに転送します。



(1) 0が格納されます。

- (s), (d)が両方ともビットデバイスの場合は、必ず(s), (d)の桁数を同一にしてください。
- (s), (d)にリンクダイレクトデバイス, ユニットアクセスデバイス, CPUバッファメモリアccessデバイスを使用する場合、(s)か(d)の片方のみで指定してください。

エラー

エラーコード (SD0)	内容
3290H	(s), (d)の両方でリンクダイレクトデバイス, ユニットアクセスデバイス, CPUバッファメモリアccessデバイスを指定したとき。

BIN16ビットデータブロック転送(32ビット)

BMOVL(P)

指定されたデバイスから(n)点(1~4294967295)のBIN16ビットデータを一括転送します。

ラダー	ST
	ENO:=BMOVL(EN,s,n,d); ENO:=BMOVLP(EN,s,n,d);

FBD/LD

■実行条件

命令	実行条件
BMOVL	
BMOVLP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	転送するデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(d)	転送先の先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(n)	転送数	0~4294967295	符号なしBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

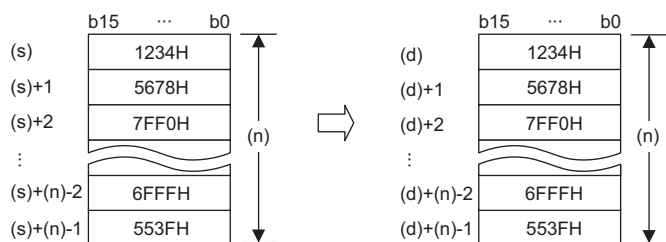
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	○	○	○	○	—	○	—	○	○	—	—	—	—
(d)	○	○	○	○	—	○	—	○	○	—	—	—	—
(n)	○	○	○	○	○	○	○	○	○	○	—	—	—

機能

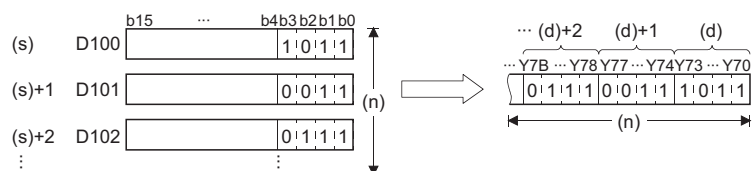
- (s)で指定されたデバイスから(n)点のBIN16ビットデータを(d)で指定されたデバイスから(n)点分、一括転送します。



- 転送元と転送先のデバイスが重複している場合も転送を行うことができます。デバイス番号の小さい方への転送の場合は (s) から、デバイス番号の大きい方への転送の場合は (s)+(n)-1 から転送します。
- (s) がワードデバイスで (d) がビットデバイスの場合、ワードデバイスはビットデバイスの桁指定で指定されたビット数が対象になります。

例

(d)にK1Y70が指定された場合、(s)で指定されたワードデバイスの下位4ビットが対象になります。



- (s), (d)が両方ともビットデバイスの場合は、必ず(s), (d)の桁数を同一にしてください。
- (s), (d)にリンクダイレクトデバイス, ユニットアクセスデバイス, CPUバッファメモリアクセスデバイスを使用する場合, (s)か(d)の片方のみで指定してください。

エラー

エラーコード (SD0)	内容
3290H	(s), (d)の両方でリンクダイレクトデバイス, ユニットアクセスデバイス, CPUバッファメモリアクセスデバイスを指定したとき。

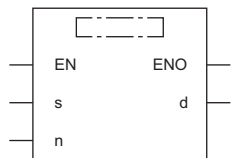
同一BIN16ビットデータブロック転送(16ビット)

FMOV(P)

BIN16ビットデータを指定されたデバイスから(n)点(0~65535)に転送します。

ラダー	ST
	ENO:=FMOV(EN,s,n,d); ENO:=FMOVP(EN,s,n,d);

FBD/LD



■実行条件

命令	実行条件
FMOV	
FMOVP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	転送するデータまたは, 転送するデータが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(d)	転送先の先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(n)	転送数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

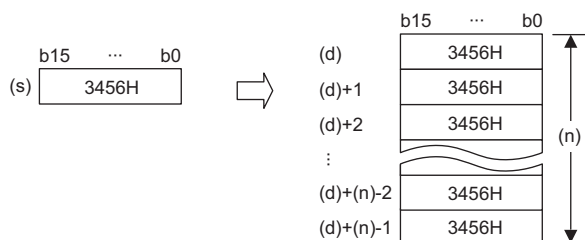
*1 ラベルで設定する場合は, 動作に必要な領域が確保できるよう配列を定義し, その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

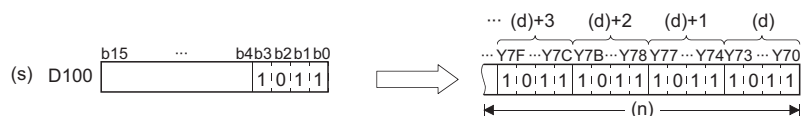
- (s)で指定されたデバイスのBIN16ビットデータと同一のデータを、(d)で指定されたデバイスへ(n)点分転送します。



- (s)がワードデバイスで(d)がビットデバイスの場合、ワードデバイスはビットデバイスの桁指定で指定されたビット数が対象になります。

例

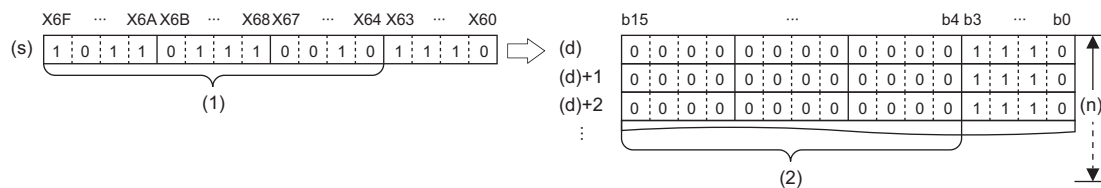
(d)にK1Y70が指定された場合、(s)で指定されたワードデバイスの下位4ビットが対象になります。



- (s)が桁指定したビットデバイスで、(d)がワードデバイスの場合、ワードデバイスはビットデバイスの桁指定で指定されたビット数が対象になります。

例

(s)にK1X60が指定された場合は、(d)のワードデバイスの下位4ビットに転送します。



(1) 無視されます。

(2) 0が格納されます。

- (s), (d)が両方ともビットデバイスの場合は、必ず(s), (d)の桁数を同一にしてください。

エラー

演算エラーはありません。

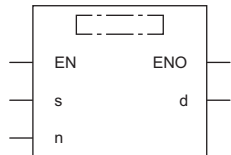
同一BIN16ビットデータブロック転送(32ビット)

FMOVL(P)

BIN16ビットデータを指定されたデバイスから(n)点(1~4294967295)に転送します。

ラダー	ST
	ENO:=FMOVL(EN,s,n,d); ENO:=FMOVLP(EN,s,n,d);

FBD/LD



■実行条件

命令	実行条件
FMOVL	
FMOVLP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	転送するデータまたは，転送するデータが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(d)	転送先の先頭デバイス	—	符号付きBIN16ビット	ANY16*1
(n)	転送数	0~4294967295	符号なしBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

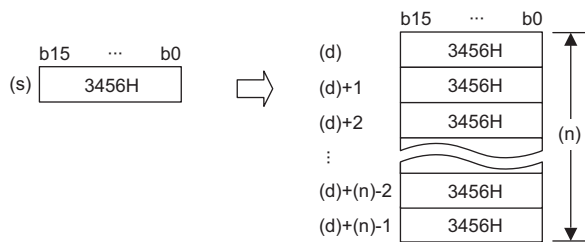
*1 ラベルで設定する場合は，動作に必要な領域が確保できるよう配列を定義し，その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ	K, H		E	\$		
(s)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	○	○	○	○	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	○	○	○	○	—	—	—	

機能

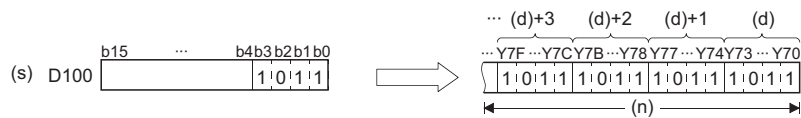
- (s)で指定されたデバイスのBIN16ビットデータと同一のデータを、(d)で指定されたデバイスへ(n)点分転送します。



- (s)がワードデバイスで(d)がビットデバイスの場合、ワードデバイスはビットデバイスの桁指定で指定されたビット数が対象になります。

例

(d)にK1Y70が指定された場合、(s)で指定されたワードデバイスの下位4ビットが対象になります。



- (s), (d)が両方ともビットデバイスの場合は、必ず(s), (d)の桁数を同一にしてください。

エラー

演算エラーはありません。

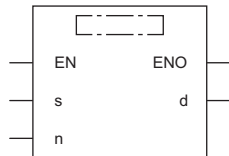
同一BIN32ビットデータブロック転送(16ビット)

DFMOV(P)

BIN32ビットデータを，指定したデバイスから(n)点(1~65535)に転送します。

ラダー	ST
	ENO:=DFMOV(EN,s,n,d); ENO:=DFMOV(P)(EN,s,n,d);

FBD/LD



■実行条件

命令	実行条件
DFMOV	
DFMOV(P)	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	転送するデータまたは，転送するデータが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(d)	転送先の先頭デバイス	—	符号付きBIN32ビット	ANY32 ^{*1}
(n)	転送数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

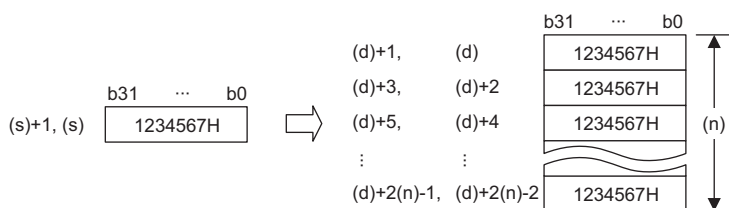
*1 ラベルで設定する場合は，動作に必要な領域が確保できるよう配列を定義し，その配列型ラベルの要素を指定してください。

■使用可能デバイス

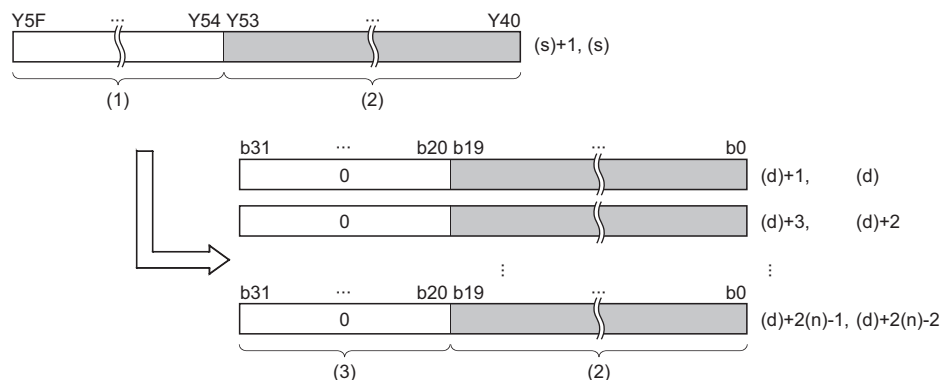
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	—	○	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- (s)で指定されたデバイスのBIN32ビットデータと同一のデータを、(d)で指定されたデバイスへ(n)点分転送します。

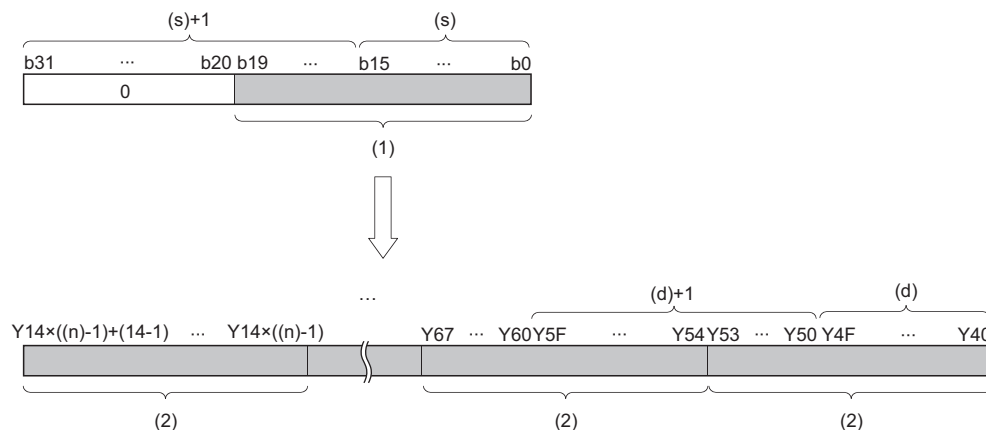


- (s)に桁指定が指定された場合、転送するデータは桁指定分のみとなります。(s)にK5Y40が指定された場合は、(s)のワードデバイスの下位20ビット(5桁分)が対象となります。



- (1) 無視されます。
 (2) 20ビット分(5桁分)のデータ
 (3) 0が格納されます。

- (d)に桁指定が指定された場合、(d)で指定された桁指定分のデータを転送します。(d)にK5Y40が指定された場合、(s)のデバイスの下位20ビットが対象となります。(s)、(d)の両方で桁指定が指定された場合も、桁数に関係なく(d)で指定された桁指定分のデータを転送します。



- (1): (d)で指定したデータ(20ビット分(5桁分))
 (2): 20ビット分(5桁分)のデータ

- (n)で指定された値が0の場合は無処理となります。

エラー

演算エラーはありません。

同一BIN32ビットデータブロック転送(32ビット)

DFMOVL(P)

BIN32ビットデータを，指定したデバイスから(n)点(1~4294967295)に転送します。

ラダー	ST
	ENO:=DFMOVL(EN,s,n,d); ENO:=DFMOVLP(EN,s,n,d);

FBD/LD

■実行条件

命令	実行条件
DFMOVL	
DFMOVLP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	転送するデータまたは，転送するデータが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(d)	転送先の先頭デバイス	—	符号付きBIN32ビット	ANY32*1
(n)	転送数	0~4294967295	符号なしBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

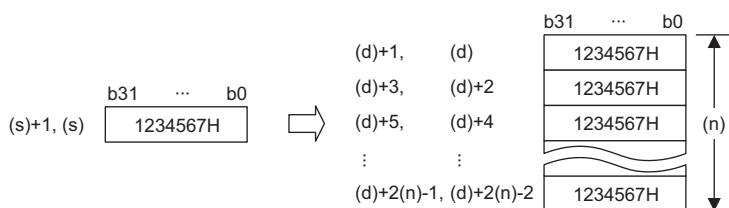
*1 ラベルで設定する場合は，動作に必要な領域が確保できるよう配列を定義し，その配列型ラベルの要素を指定してください。

■使用可能デバイス

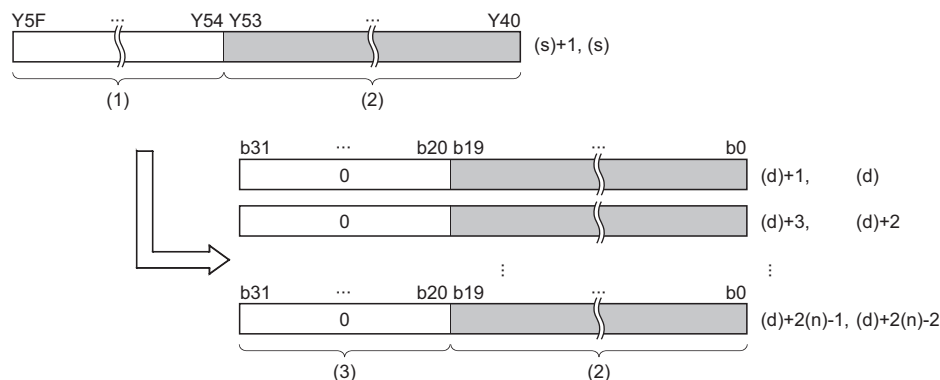
オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	—	○	—	○	○	—	—	—	—
(n)	○	○	○	○	○	○	○	○	○	○	—	—	—

機能

- (s)で指定されたデバイスのBIN32ビットデータと同一のデータを、(d)で指定されたデバイスへ(n)点分転送します。

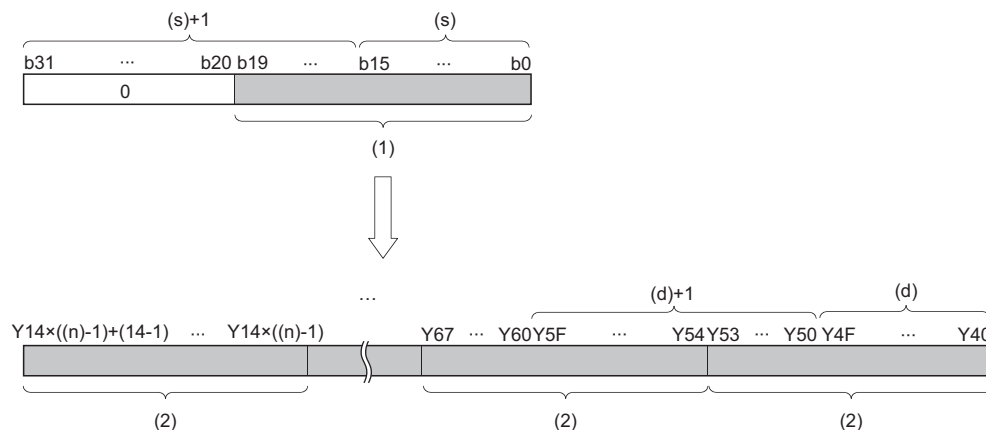


- (s)に桁指定が指定された場合、転送するデータは桁指定分のみとなります。(s)にK5Y40が指定された場合は、(s)のワードデバイスの下位20ビット(5桁分)が対象となります。



- (1) 無視されます。
 (2) 20ビット分(5桁分)のデータ
 (3) 0が格納されます。

- (d)に桁指定が指定された場合、(d)で指定された桁指定分のデータを転送します。(d)にK5Y40が指定された場合、(s)のデバイスの下位20ビットが対象となります。(s)、(d)の両方で桁指定が指定された場合も、桁数に関係なく(d)で指定された桁指定分のデータを転送します。



- (1): (d)で指定したデータ(20ビット分(5桁分))
 (2): 20ビット分(5桁分)のデータ

- (n)で指定された値が0の場合は無処理となります。

エラー

演算エラーはありません。

BIN16ビットデータ交換

XCH(P)

指定されたBIN16ビットデータを交換します。

ラダー	ST
	ENO:=XCH(EN,d1,d2); ENO:=XCHP(EN,d1,d2);

FBD/LD

■実行条件

命令	実行条件
XCH	
XCHP	

設定データ

■内容，範囲，データ型

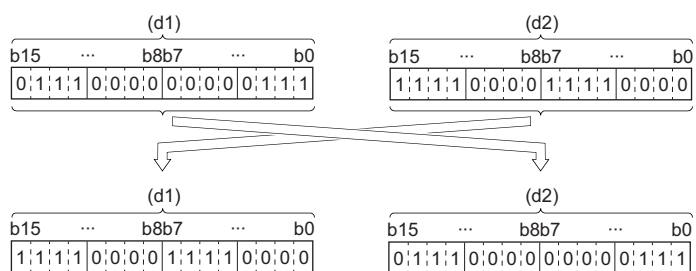
オペランド	内容	範囲	データ型	データ型(ラベル)
(d1)	交換するデータが格納されているデバイス	—	符号付きBIN16ビット	ANY16
(d2)	交換するデータが格納されているデバイス	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ			K, H	E	\$	
(d1)	○	○	○	○	○	—	—	○	—	—	—	—	—
(d2)	○	○	○	○	○	—	—	○	—	—	—	—	—

機能

- (d1)と(d2)のBIN16ビットデータを交換します。



エラー

演算エラーはありません。

BIN32ビットデータ交換

DXCH(P)

指定されたBIN32ビットデータを交換します。

ラダー	ST
	ENO:=DXCH(EN,d1,d2); ENO:=DXCHP(EN,d1,d2);

FBD/LD

■実行条件

命令	実行条件
DXCH	
DXCHP	

設定データ

■内容，範囲，データ型

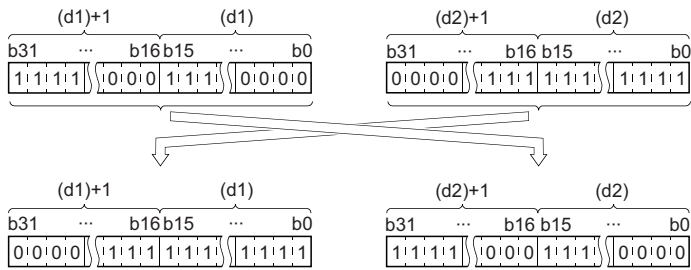
オペランド	内容	範囲	データ型	データ型(ラベル)
(d1)	交換するデータが格納されている先頭デバイス	—	符号付きBIN32ビット	ANY32
(d2)	交換するデータが格納されている先頭デバイス	—	符号付きBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(d1)	○	○	○	○	○	○	○	—	—	—	—
(d2)	○	○	○	○	○	○	○	—	—	—	—

機能

- (d1)と(d2)のBIN32ビットデータを交換します。



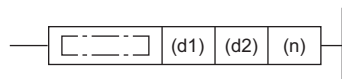
エラー

演算エラーはありません。

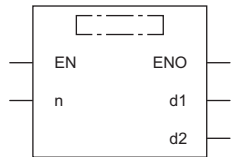
BIN16ビットデータブロック交換

BXCH(P)

指定されたデバイスから(n)点のBIN16ビットデータを交換します。

ラダー	ST
	ENO:=BXCH(EN,n,d1,d2); ENO:=BXCHP(EN,n,d1,d2);

FBD/LD



■実行条件

命令	実行条件
BXCH	
BXCHP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d1)	交換するデータが格納されている先頭デバイス	—	ワード	ANY16 ^{*1}
(d2)	交換するデータが格納されている先頭デバイス	—	ワード	ANY16 ^{*1}
(n)	交換数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

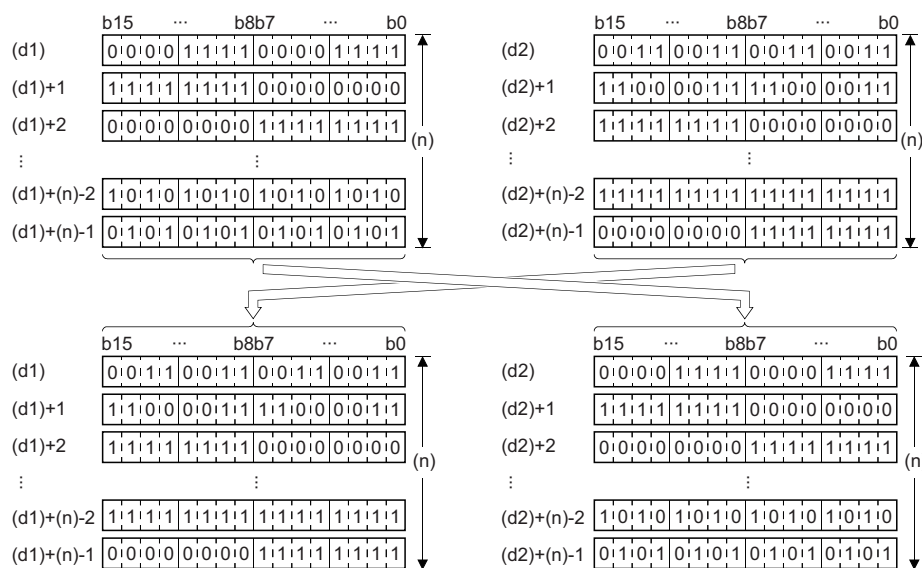
*1 ラベルで設定する場合は，動作に必要な領域が確保できるよう配列を定義し，その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□		LT, LST, LC	LZ		K, H	E	\$	
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- (d1)で指定されたデバイスと、(d2)で指定されたデバイスから(n)点分のBIN16ビットデータを交換します。



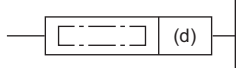
エラー

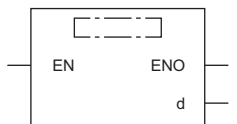
エラーコード (SD0)	内容
2821H	(d1), (d2)のデバイスが重複しているとき。

BIN16ビットデータ上下バイト交換

SWAP(P)

指定したデバイスの上下各8ビットの値を交換します。

ラダー	ST
	ENO:=SWAP(EN,d); ENO:=SWAPP(EN,d);

FBD/LD


■実行条件

命令	実行条件
SWAP	
SWAPP	

設定データ

■内容、範囲、データ型

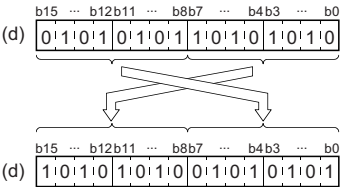
オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	上下各8ビットの値を交換するデータが格納される先頭デバイス	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード	間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□			K, H	E	\$	
(d)	○	○	○	○	○	—	—	—	—	—

機能

- (d)で指定したデバイスが、上下各8ビットの値を交換します。



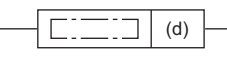
エラー

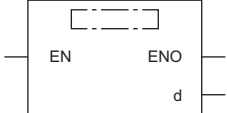
演算エラーはありません。

BIN32ビットデータ上下バイト交換

DSWAP(P)

指定したデバイスの上下各8ビットの値を交換します。

ラダー	ST
	ENO:=DSWAP(EN,d); ENO:=DSWAPP(EN,d);

FBD/LD


■実行条件

命令	実行条件
DSWAP	
DSWAPP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	上下各8ビットの値を交換するデータが格納される先頭デバイス	—	符号付きBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

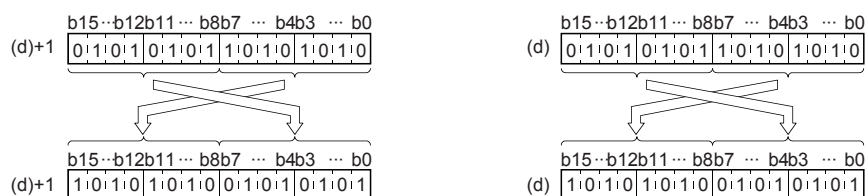
■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(d)	○*1	○	○	○	○	○	○	—	—	—	—

*1 FX, FYは使用できません。

機能

- (d)で指定したデバイスが、上下各8ビットの値を交換します。



エラー

演算エラーはありません。

1ビットデータ転送

MOVB(P)

指定した1ビットデータを転送します。

ラダー	ST
	ENO:=MOVB(EN,s,d); ENO:=MOVBP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
MOVB	
MOVBP	

設定データ

■内容、範囲、データ型

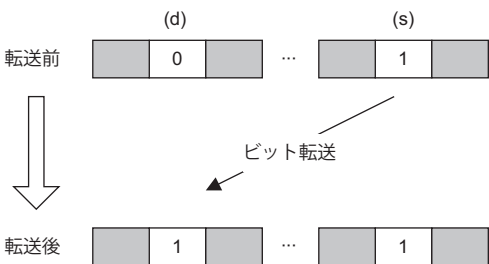
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	転送するデータが格納されているデバイス番号	—	ビット	ANY_BOOL
(d)	転送先のデバイス番号	—	ビット	ANY_BOOL
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	—	—	○	—	○	—	—	—	—
(d)	○	○	○	—	—	○	—	○	—	—	—	—

機能

- ・ (s)で指定されたビットデータを(d)へ転送します。



エラー

演算エラーはありません。

nビットデータ転送

12

BLKMOVB(P)

(n)点分のビットデータを一括転送します。

ラダー	ST
	ENO:=BLKMOVB(EN,s,n,d); ENO:=BLKMOVBP(EN,s,n,d);

FBD/LD

■実行条件

命令	実行条件
BLKMOVB	
BLKMOVBP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	転送元のブロックデータ	—	ビット	ANY_BOOL ^{*1}
(d)	転送先のブロックデータ	—	ビット	ANY_BOOL ^{*1}
(n)	転送数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

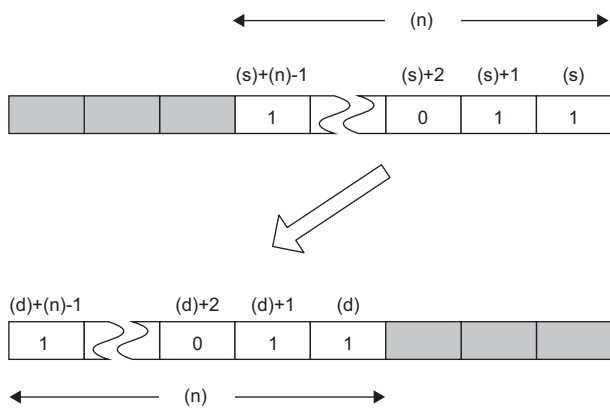
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	○	—	○	—	—	—	○	—	—	—	—
(d)	○	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

機能

- (s)から(n)点分のビットデータを, (d)から(n)点分のビットデータへ一括転送します。
- 転送元と転送先が重複している場合も, 転送を行うことができます。



エラー

演算エラーはありません。

第4部 応用命令

この部は下記の章構成となります。

13 プログラム制御

14 データ処理

15 デバッグ, 故障診断

16 文字列処理

17 実数処理

18 乱数

19 デバイス操作

20 タイマ, カウンタ

21 時計

22 ユニットアクセス

23 PID演算命令

24 PID制御命令

25 アドオン

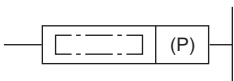
13 プログラム制御

13.1 プログラム分岐命令

ポインタ分岐

CJ, SCJ, JMP

- CJ: 同一プログラムファイル内の指定されたポインタ番号のプログラムを実行します。
- SCJ: 次のスキャンから同一プログラムファイル内の指定されたポインタ番号のプログラムを実行します。
- JMP: 無条件に同一プログラムファイル内の指令されたポインタ番号のプログラムを実行します。

ラダー	ST
	対応していません。

FBD/LD

対応していません。

■実行条件

命令	実行条件
CJ SCJ	
JMP	常時実行

設定データ

■内容, 範囲, データ型

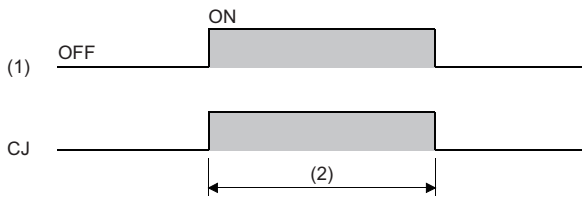
オペランド	内容	範囲	データ型	データ型(ラベル)
(P)	ジャンプ先のポインタ番号	—	デバイス名	POINTER

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (P)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(P)	—	—	—	—	—	—	—	—	—	—	—	○

■CJ

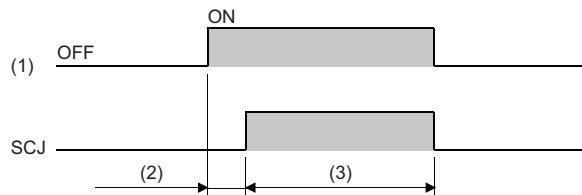
- 実行指令がONのとき、同一プログラムファイル内の指定されたポインタ番号のプログラムを実行します。
- 実行指令がOFFのときは、次のステップのプログラムを実行します。



- (1): 実行指令
(2): 毎スキャン実行

■SCJ

- 実行指令がOFF→ONに変化した次のスキャンから、同一プログラムファイル内の指定されたポインタ番号のプログラムを実行します。
- 実行指令がOFFおよびON→OFFに変化したときは、次のステップのプログラムを実行します。



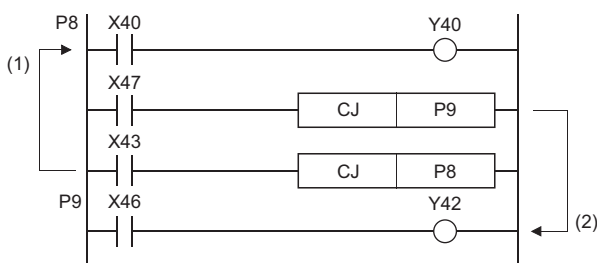
- (1): 実行指令
(2): 1スキャン
(3): 毎スキャン実行

■JMP

- 無条件に、同一プログラムファイル内の指定されたポインタ番号のプログラムを実行します。

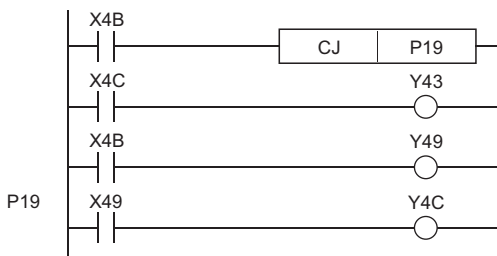
注意事項

- タイマのコイルをONした後、CJ命令、SCJ命令、JMP命令でコイルのONしているタイマをジャンプさせた場合、正常な計測ができなくなります。
- CJ命令、SCJ命令、JMP命令でOUT命令をジャンプさせるとスキャンタイムは短くなります。
- CJ命令、SCJ命令、JMP命令で後ろにジャンプさせるとスキャンタイムは短くなります。
- CJ命令、SCJ命令、JMP命令は、実行中のステップより若いステップへジャンプできます。ただし、ウォッチドッグタイマがタイムアップしないように、ループから抜け出す方法を考慮してください。



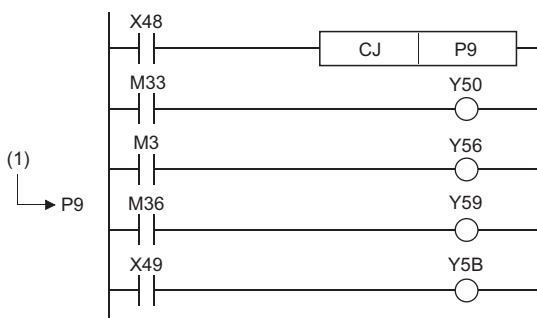
- (1) X43がONの間、ループします。
(2) X47をONすると、ループから抜け出します。

- CJ命令、SCJ命令、JMP命令で飛ばされたデバイスは変化しません。



- X4BがONすると、P19のラベルへジャンプします。
CJ命令を実行中にX4B、X4CがON/OFFしても、Y43、Y49は変化しません。

- ラベル(P□)は1ステップを占有します。



- (1) 1ステップを占有します。

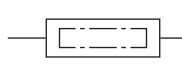
- ジャンプ命令は同一プログラムファイル内のポインタ番号のみ指定可能です。

エラー

エラーコード (SD0)	内容
3340H	(P)で指定したポインタ番号がEND命令以前にないとき。
	(P)で指定したポインタ番号が、同一プログラムでラベルとして使用されていないとき。
	(P)で指定したポインタ番号が、他のプログラムにあるグローバルポインタのとき。

GOEND

同一プログラムファイル内のFENDまたはEND命令へジャンプします。

ラダー	ST
	対応していません。

FBD/LD
対応していません。

■実行条件

命令	実行条件
GOEND	

機能

- 同一プログラムファイル内のFENDまたはEND命令へジャンプします。

エラー

エラーコード (SD0)	内容
3341H	CALL(P)命令, ECALL(P)命令を実行後, RET命令を実行する前にGOEND命令を実行したとき。
3351H	割込みポインタ(I)による割込みプログラム中で, IRET命令を実行する前にGOEND命令を実行したとき。

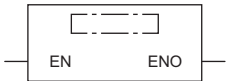
13.2 プログラム実行制御命令

割込禁止，割込許可

DI, EI

- DI: 割込みプログラムの実行を禁止します。
- EI: 割込み禁止状態を解除します。

ラダー	ST
	ENO:=DI(EN); ENO:=EI(EN);

FBD/LD


■実行条件

命令	実行条件
DI EI	

機能

■DI

- 割込みプログラムの起動要因が発生しても，EI命令が実行されるまで割込みプログラムの実行を禁止します。
- DI(指定優先度以下の割込禁止)命令使用時の，DI(割込禁止)命令の動作については，下記を参照してください。

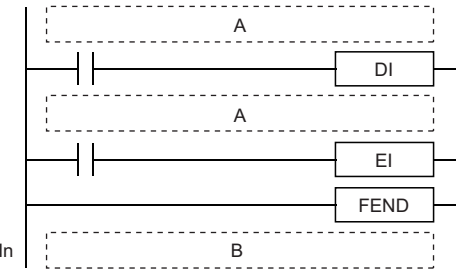
☞ 536ページ DI

- DI(割込禁止)命令は割込みプログラム中には実行できません。実行した場合は無処理となります。

■EI

- DI(割込禁止)命令実行時の割込み禁止状態を解除し，IMASK命令によって許可された割込みポインタ番号の割込みプログラムを実行可能状態にします。IMASK命令非実行時は，I32~I43が割込み禁止になります。
- 電源投入時またはコントローラをリセット，RUN移行した場合はEI命令を実行した状態になります。初期実行タイププログラムの設定がある場合は，RUN後1スキャン目の初期実行タイププログラムの実行中はDI命令を実行した状態になり，2スキャン目のプログラム実行時からEI命令を実行した状態になります。
- DI(指定優先度以下の割込禁止)命令使用時のEI命令の動作については，下記を参照してください。

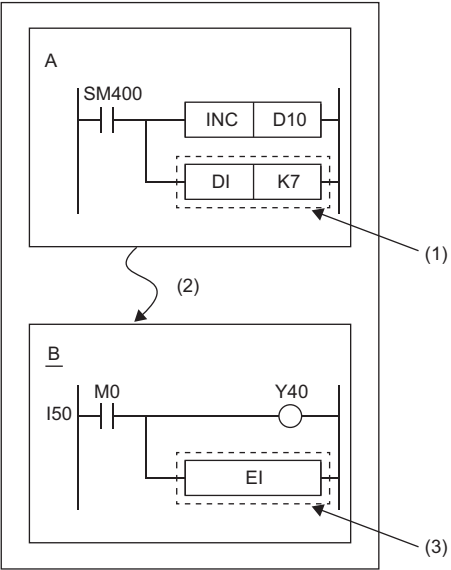
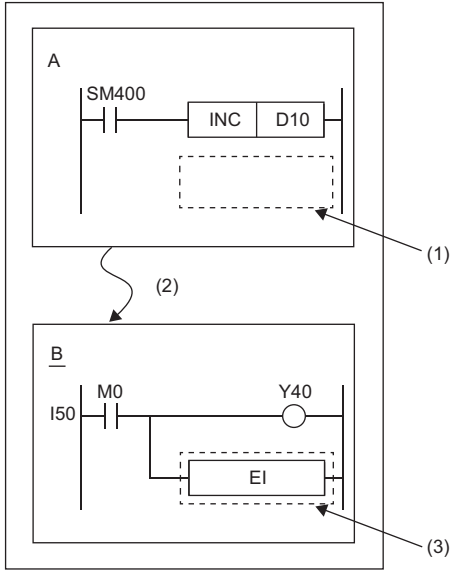
☞ 536ページ DI



A: シーケンスプログラム
B: 割込みプログラム

DI~EI命令間で割込み要因が発生しても，DI~EI命令間の処理を終了するまで，割込みプログラムは待たされます。

- 割込みプログラム中でのEI命令の動作は、EI命令実行前のDI(指定優先度以下の割込禁止)命令の実行により変わります。DI(指定優先度以下の割込禁止)命令実行後の割込みプログラム中のEI命令は実行可能です。割込みプログラム中でのEI命令の実行可否は下記となります。

DI(指定優先度以下の割込禁止)命令の実行後、割込みプログラムでEI命令を実行した場合	DI(指定優先度以下の割込禁止)命令の未実行で、割込みプログラムでEI命令を実行した場合
 A: メインルーチンプログラム B: 割込みプログラム (1): DI(指定優先度以下の割込禁止)命令の実行 (2): 割込み発生 (3): 実行可	 A: メインルーチンプログラム B: 割込みプログラム (1): DI(指定優先度以下の割込禁止)命令の未実行 (2): 割込み発生 (3): 実行不可(無処理)

- 初期実行タイププログラム中のEI命令、DI命令は実行可能です。ただし、初期実行タイププログラム中にEI命令、DI命令を実行した場合、2スキャン目が始まるまでは指定優先度の設定が有効です。2スキャン目のプログラム実行時からEI命令を実行した状態になります。

Point

割込み禁止区間は最小限るようにプログラミングする、もしくはDI(指定優先度以下の割込禁止)命令を使用してネットワーク通信周期同期割込みを禁止の対象外としてください。DI命令を使用するとネットワーク通信周期同期割込みが妨げられ、下記の機能の誤動作の原因となります。

- モーション機能
- ネットワーク周期通信機能

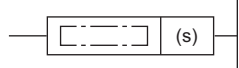
エラー

エラーコード (SD0)	内容
3332H	DI(割込禁止)命令、DI(指定優先度以下の割込禁止)命令によるネスティングが16重を超えた場合。

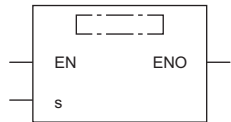
指定優先度以下の割込禁止

DI

指定した優先度以下の割込みプログラムの実行を禁止します。

ラダー	ST
	ENO:=DI_1(EN,s);

FBD/LD



(□には、DI_1が入ります。)

■実行条件

命令	実行条件
DI	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	割込みを禁止する優先度	1~32	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

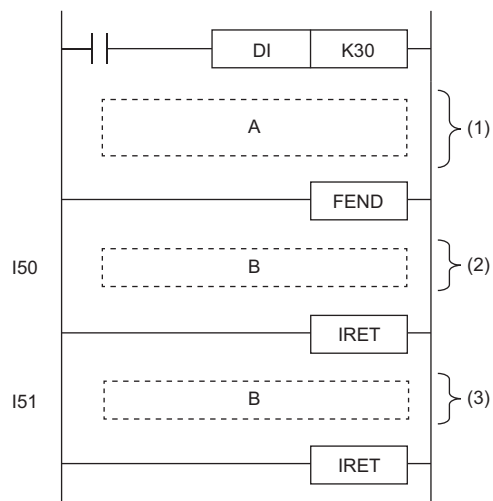
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	—	○	—	○	—	—	○	○	—	—	—

- ・ (s)で指定された割込み優先度以下の割込みポインタ番号の割込みプログラムを禁止します。

割込み優先度設定

I No.	優先度
I50	29
I51	30



A: シーケンスプログラム

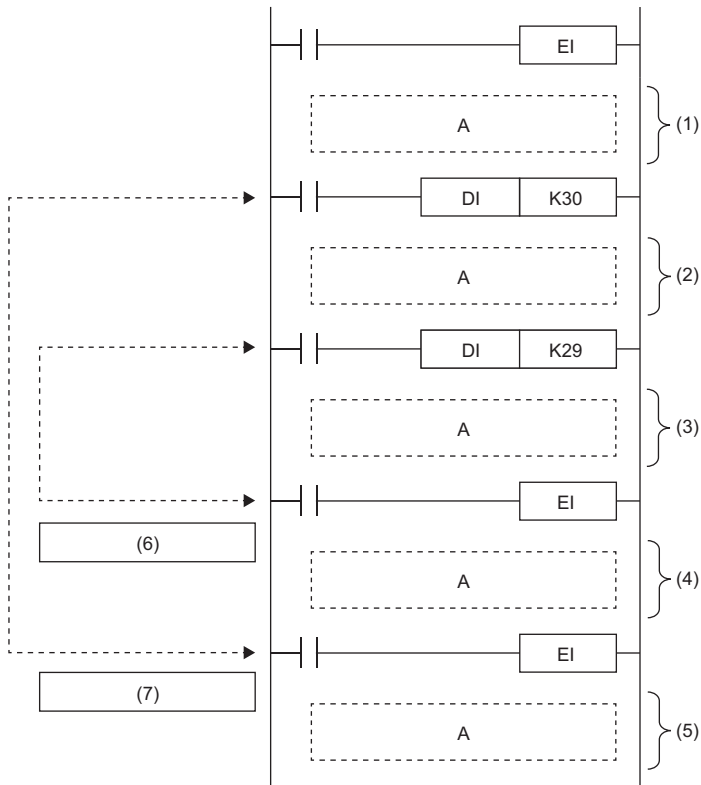
B: 割込みプログラム

(1): 優先度30以下の割込み禁止区間(優先度29以上の割込み許可区間)

(2): 優先度29の割込みのため実行可能

(3): 優先度30の割込みのため実行禁止

- EI命令の実行で、対となるDI命令で禁止した優先度の割込みを許可します。ただし、DI(割込禁止)命令のみで割込み禁止にしている場合、EI命令を1回実行するのみで、すべての優先度の割込みを許可します。

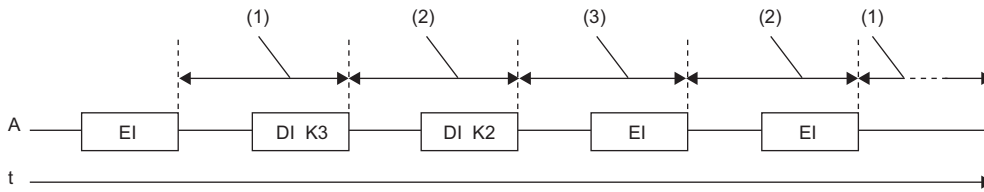


A: シーケンスプログラム

- (1): すべての優先度の割込み許可区間
- (2): 優先度30以下の割込み禁止区間(優先度29以上の割込み許可区間)
- (3): 優先度29以下の割込み禁止区間(優先度28以上の割込み許可区間)
- (4): 優先度30以下の割込み禁止区間(優先度29以上の割込み許可区間)
- (5): すべての優先度の割込み許可区間
- (6): [DI K29]に対応するEI命令
- (7): [DI K30]に対応するEI命令

- DI(指定優先度以下の割込禁止)命令の多重実行時、指定する割込み禁止優先度が現在禁止している割込み禁止優先度以下の場合、現在の割込み禁止優先度は変更されません。
- DI命令のネスティングは16重まで可能です。
- 割込みポイントの割込み優先度はパラメータにより設定可能です。(📖使用するコントローラのユーザーズマニュアル)
- 割込み禁止となっている優先度は、SD758(割込み禁止優先度設定値)にて確認可能です。
- 割込みプログラム中にDI(指定優先度以下の割込禁止)命令を実行し、割込み禁止優先度が変更となった場合はSD758の値も変更されます。

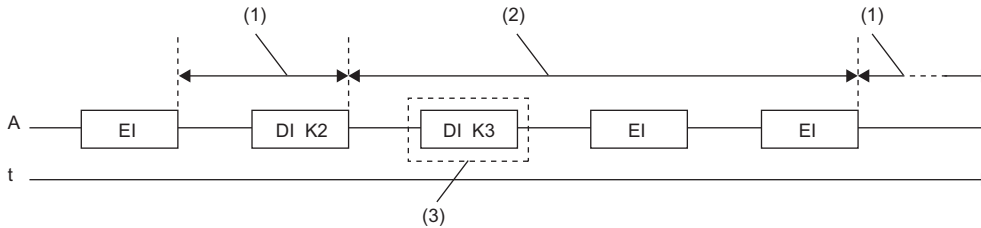
- DI(割込禁止)命令, DI(指定優先度以下の割込禁止)命令, およびEI命令実行時の割込み禁止区間は下記になります。
- DI(指定優先度以下の割込禁止)命令を多重実行した場合(現在禁止している割込み優先度より高い優先度の割込みを禁止する指定をした場合)



A: スキャン実行タイププログラム

- (1): すべての優先度の割込み許可区間
- (2): 優先度3以下の割込み禁止区間(優先度2以上の割込み許可区間)
- (3): 優先度2以下の割込み禁止区間(優先度1以上の割込み許可区間)

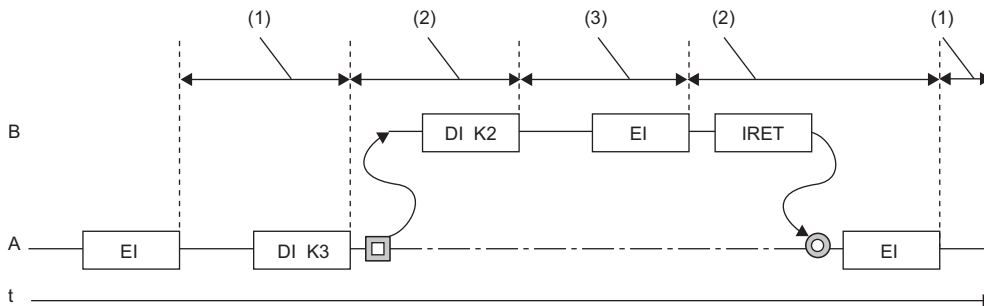
- DI(指定優先度以下の割込禁止)命令を多重実行した場合(現在禁止している割込み優先度より低い優先度の割込みを禁止する指定をした場合)



A: スキャン実行タイププログラム

- (1): すべての優先度の割込み許可区間
- (2): 優先度2以下の割込み禁止区間(優先度1以上の割込み許可区間)
- (3): すでに優先度2以下の割込みが禁止されているため、割込み禁止優先度は変更されません。

- 割込みプログラムでDI(指定優先度以下の割込禁止)命令を実行した場合

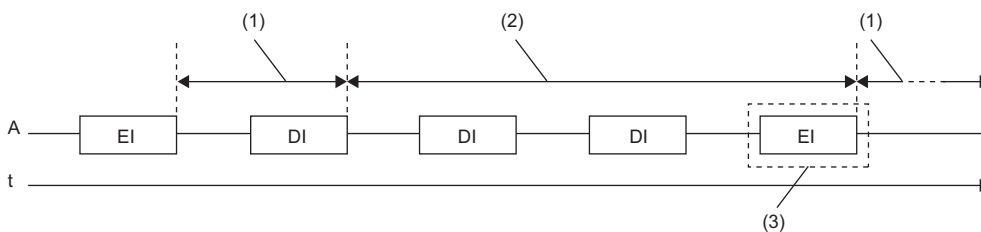


A: スキャン実行タイププログラム

B: 割込みプログラム

- (1): すべての優先度の割込み許可区間
- (2): 優先度3以下の割込み禁止区間(優先度2以上の割込み許可区間)
- (3): 優先度2以下の割込み禁止区間(優先度1以上の割込み許可区間)

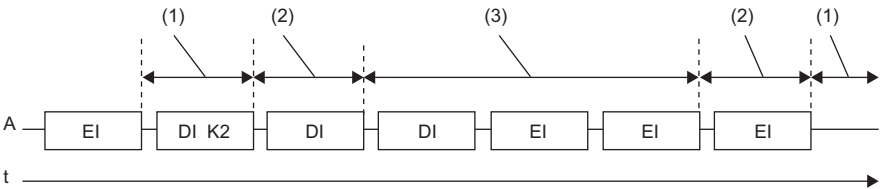
- DI(割込禁止)命令のみ実行した場合



A: スキャン実行タイププログラム

- (1): すべての優先度の割込み許可区間
- (2): 優先度1以下の割込み禁止区間(すべての割込み禁止区間)
- (3): DI(割込禁止)命令で割込み禁止としているため、EI命令を1回実行することですべての優先度の割込みが許可されます。

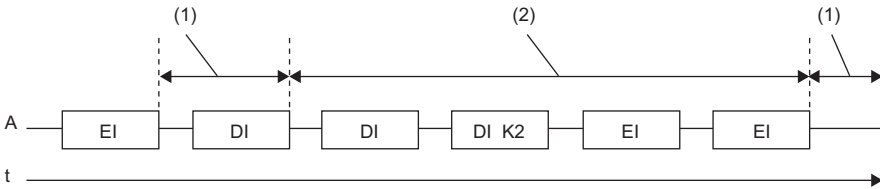
・ DI(指定優先度以下の割込禁止)命令とDI(割込禁止)命令を実行した場合(DI(指定優先度以下の割込禁止)命令→DI(割込禁止)命令の順で実行した場合)



A: スキャン実行タイププログラム

- (1): すべての優先度の割込み許可区間
(2): 優先度2以下の割込み禁止区間(優先度1以上の割込み許可区間)
(3): 優先度1以下の割込み禁止区間(すべての割込み禁止区間)

・ DI(指定優先度以下の割込禁止)命令とDI(割込禁止)命令を実行した場合(DI(割込禁止)命令→DI(指定優先度以下の割込禁止)命令の順で実行した場合)



A: スキャン実行タイププログラム

- (1): すべての優先度の割込み許可区間
(2): 優先度1以下の割込み禁止区間(すべての割込み禁止区間)

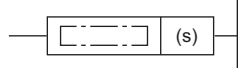
・ 初期実行タイププログラム中のDI命令は実行可能です。ただし、初期実行タイププログラム中にDI命令を実行した場合、2スキャン目が始まるまでは指定優先度の設定が有効です。2スキャン目のプログラム実行時からEI命令を実行した状態になります。

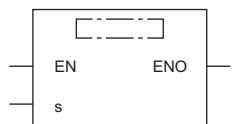
エラー

エラーコード (SD0)	内容
3285H	(s)で指定した優先度が範囲外であるとき。
3332H	DI(割込禁止)命令、DI(指定優先度以下の割込禁止)命令によるネスティングが16重を超えた場合。

IMASK

指定された割込みポイント番号の割込みプログラムを実行許可状態/実行禁止状態にします。

ラダー	ST
	ENO:=IMASK(EN,s);

FBD/LD


■実行条件

命令	実行条件
IMASK	常時実行

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	割込みマスクデータまたは、割込みマスクデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 16)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	—	—	○		—	—	—	—

機能

- (s)で指定されたデバイスから16点分のビットパターンにより、指定された割込みポイント番号の割込みプログラムを実行許可状態または実行禁止状態にします。
- 1(ON): 割込みプログラムの実行許可状態
- 0(OFF): 割込みプログラムの実行禁止状態
- 各ビットに対応する割込みポイント番号は下記のとおりです。

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
(s)	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
(s)+1	I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
(s)+2	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
(s)+3	I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
(s)+4	I79	I78	I77	I76	I75	I74	I73	I72	I71	I70	I69	I68	I67	I66	I65	I64
(s)+5	I95	I94	I93	I92	I91	I90	I89	I88	I87	I86	I85	I84	I83	I82	I81	I80
(s)+6	I111	I110	I109	I108	I107	I106	I105	I104	I103	I102	I101	I100	I99	I98	I97	I96
(s)+7	I127	I126	I125	I124	I123	I122	I121	I120	I119	I118	I117	I116	I115	I114	I113	I112
(s)+8	I143	I142	I141	I140	I139	I138	I137	I136	I135	I134	I133	I132	I131	I130	I129	I128
(s)+9	I159	I158	I157	I156	I155	I154	I153	I152	I151	I150	I149	I148	I147	I146	I145	I144
(s)+10	I175	I174	I173	I172	I171	I170	I169	I168	I167	I166	I165	I164	I163	I162	I161	I160
(s)+11	I191	I190	I189	I188	I187	I186	I185	I184	I183	I182	I181	I180	I179	I178	I177	I176
(s)+12	I207	I206	I205	I204	I203	I202	I201	I200	I199	I198	I197	I196	I195	I194	I193	I192
(s)+13	I223	I222	I221	I220	I219	I218	I217	I216	I215	I214	I213	I212	I211	I210	I209	I208
(s)+14	I239	I238	I237	I236	I235	I234	I233	I232	I231	I230	I229	I228	I227	I226	I225	I224
(s)+15	I255	I254	I253	I252	I251	I250	I249	I248	I247	I246	I245	I244	I243	I242	I241	I240

- 電源投入時またはコントローラをリセットした場合、I0~I31、I44~I1023の割込みプログラムは実行許可状態になり、I32~I43の割込みプログラムは実行禁止状態になります。
- (s)~(s)+15のデバイスの状態は、SD1400~SD1463(IMASK命令マスクパターン)に格納されます。

Point

IMASK命令では、I0~I255の割込みポイントを一括で、実行許可状態または実行禁止状態にできます。
I256~I1023の割込みポイントを実行許可状態または実行禁止状態にする場合は、プログラムでIMASK命令をSIMASK命令へ代替してください。
SIMASK命令の詳細については、下記を参照してください。
 543ページ SIMASK

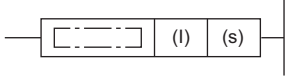
エラー

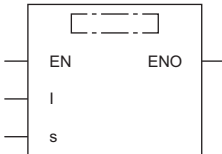
演算エラーはありません。

指定割込ポイントの禁止/許可

SIMASK

指定した割込みポイント番号の割込みプログラムを，実行許可状態/実行禁止状態にします。

ラダー	ST
	ENO:=SIMASK(EN,I,S);

FBD/LD


■実行条件

命令	実行条件
SIMASK	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(I)	割込みを許可/禁止設定する割込みポイントNo.	I0~I1023	デバイス名	POINTER ^{*1}
(S)	指定した割込みポイントNo.の許可/禁止	0: 禁止 1: 許可	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 デバイス(I)に割り付けたラベルのみ使用できます。

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他 (I)
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 口		LT, LST, LC	LZ		K, H	E	\$	
(I)	—	—	—	—	—	—	—	—	—	—	—	○
(S)	—	—	○	—	○	—	—	○	○	—	—	—

機能

- (I)で指定した割込みポイント番号の割込みプログラムを，(S)で指定したデータに従い，実行許可状態または実行禁止状態にします。
 - (S)が1の場合: 割込みプログラムの実行許可状態
 - (S)が0の場合: 割込みプログラムの実行禁止状態
- 電源投入時またはコントローラをリセットした場合，I0~I31，I44~I1023の割込みプログラムは実行許可状態になり，I32~I43の割込みプログラムは実行禁止状態になります。
- 割込みポイントの実行許可状態/実行禁止状態は，SD1400~SD1463(IMASK命令マスクパターン)に格納されます。

Point

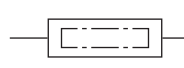
(I)はインデックス修飾ができます。インデックス修飾したSIMASK命令を使用することで，I0~I1023の割込みポイントを実行許可状態または実行禁止状態にできます。

エラー

エラーコード (SD0)	内容
3285H	(l)で指定された割込みポインタNo.が、割込みポインタNo.の範囲を超えるとき。
	(s)の値が、割込み禁止(0)/割込み許可(1)以外るとき。

IRET

割込みプログラムの処理の終了を示します。

ラダー	ST
	対応していません。

FBD/LD
対応していません。

■実行条件

命令	実行条件
IRET	常時実行

機能

- 割込みポインタ(I)による割込みプログラムの処理の終了を示します。
- IRET命令実行後、シーケンスプログラムへ処理を戻します。

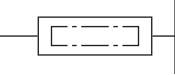
エラー

エラーコード (SD0)	内容
3350H	割込みNo.に対応するポインタが存在しないとき。
3351H	割込み発生後IRET命令を実行する前に、END命令、FEND命令、GOEND命令、STOP命令を実行したとき。
3352H	メインルーチンプログラム、またはサブルーチンプログラム中でIRET命令を実行したとき。
3353H	定周期実行タイププログラム中でIRET命令を実行したとき。
3354H	イベント実行タイププログラム中でIRET命令を実行したとき。

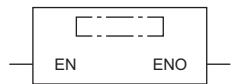
WDTリセット

WDT(P)

ウォッチドッグタイマをリセットします。

ラダー	ST
	ENO:=WDT(EN); ENO:=WDTP(EN);

FBD/LD

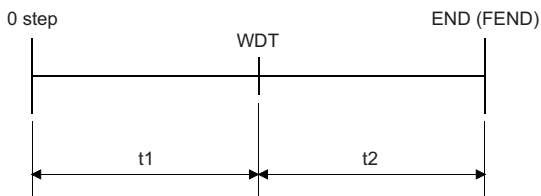


■実行条件

命令	実行条件
WDT	
WDTP	

機能

- プログラム中でウォッチドッグタイマをリセットします。
- スキャンタイムが、条件によりウォッチドッグタイマの設定値を超える場合に使用します。スキャンタイムが、毎スキャンウォッチドッグタイマの設定値を超える場合は、エンジニアリングツールのパラメータ設定でウォッチドッグタイマの設定値を変更してください。
- 0ステップからWDT(P)命令までのt1と、WDT(P)命令からEND(FEND)命令までのt2は、ウォッチドッグタイマの設定値を超えないようにしてください。



- WDT(P)命令は1スキャン中に2回以上使用できますが、異常発生時の出力OFFまでの時間がかかりますので注意してください。
- WDT(P)命令を実行しても、特殊レジスタに格納されるスキャンタイムの値はクリアされません。したがって、特殊レジスタのスキャンタイム値は、パラメータで設定したウォッチドッグタイマの設定値よりも大きくなる場合があります。

エラー

演算エラーはありません。

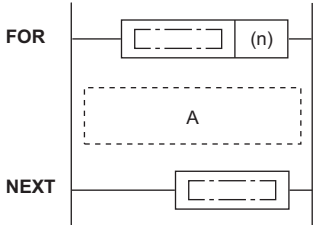
13.3 構造化命令

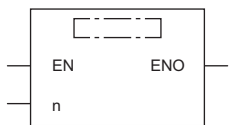
13

FOR~NEXT

FOR, NEXT

FOR~NEXT命令間の処理を(n)回実行します。

ラダー	ST
 A: くり返しプログラム	対応していません。

FBD/LD	
[FOR] 	[NEXT] 

■実行条件

命令	実行条件
FOR NEXT	常時実行

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(n)	FOR~NEXT命令間の繰返し回数	1~32767	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

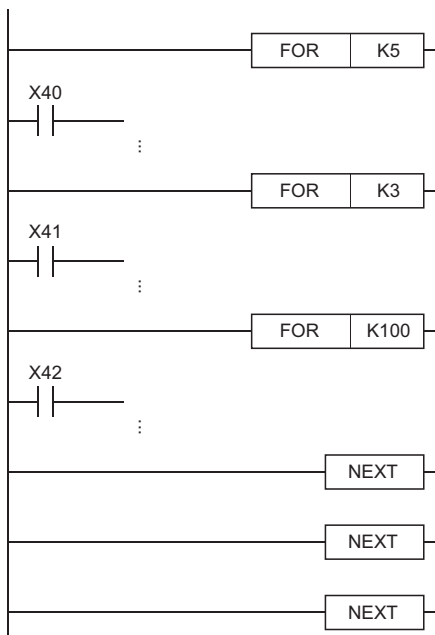
■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □		LT, LST, LC	LZ		K, H	E	\$	
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- FOR~NEXT命令間の処理を無条件に(n)回実行すると、NEXT命令の次ステップの処理を行います。
- (n)は1~32767で指定できます。-32768~0を指定した場合は、(n)=1と同一の処理を行います。
- FOR~NEXT命令間の処理を実行したくないときは、CJ命令、SCJ命令でジャンプさせてください。

- FOR命令のネスティングは16重まで可能です。

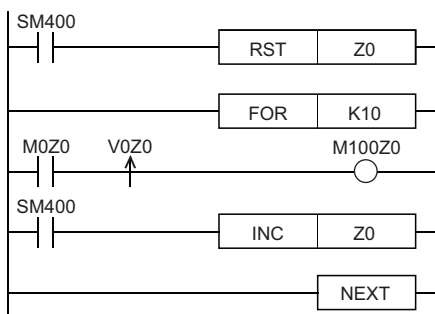


注意事項

- FOR~NEXT命令間を繰り返し実行中に、途中で終了させる場合は、BREAK命令を使用してください。

☞ 550ページ BREAK(P)

- FOR~NEXT命令間で、インデックス修飾されたプログラムのパルス演算を行う場合は、EGP命令、EGF命令を使用してください。ただし、動作出力側に立上り命令、立下り命令は使用できません。



- FOR~NEXT命令間の外からFOR~NEXT命令内に、FOR~NEXT命令間の中からFOR~NEXT命令外に、JMP命令などで分岐できず、変換エラーになります。
- FOR~NEXT命令内のJMP命令などで指定するポインタデバイス/ラベルが、インデックス修飾をした場合、分岐できず、変換エラーになります。
- FOR~NEXT命令間の外から、FOR~NEXT命令内にCALL(P)命令で分岐できず、変換エラーになります。ただし、CALL(P)命令で分岐先に指定するポインタデバイス/ラベルをインデックス修飾し、インデックス修飾解決後にFOR~NEXT内に存在するポインタを指定した場合は、CALL(P)命令実行時にエラー(エラーコード: 3340H)になります。CALL(P)命令の詳細は下記を参照してください。

☞ 552ページ CALL(P)

- FOR~NEXT命令内でネスト構造になっている場合は、JMP命令などで指定するポインタデバイス/ラベルが実行元のJMP命令などと同一階層にある場合のみ記述できます。異なる階層に記述した場合は変換エラーになります。
- FOR命令、NEXT命令をFBD/LD言語で使った場合、EN(実行条件)に常時ON(TRUE/1/SM400)以外の変数や定数を指定すると変換エラーになります。

エラー

演算エラーはありません。

13

Point

分かりやすいプログラムを作成するために、対になるFOR命令とNEXT命令は一つのプログラムブロック内で使用してください。

FOR~NEXT強制終了

BREAK(P)

FOR~NEXT命令による繰り返し処理を強制的に終了し、FOR~NEXT命令以降の処理に実行を移します。

ラダー	ST
	対応していません。

FBD/LD

■実行条件

命令	実行条件
BREAK	
BREAKP	

設定データ

■内容，範囲，データ型

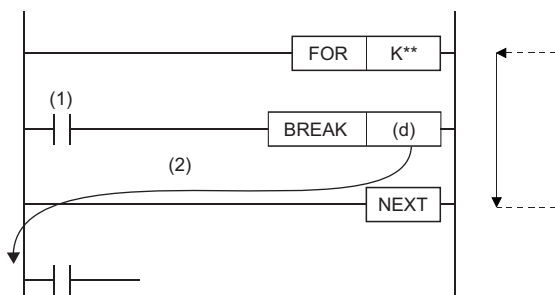
オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	繰返し残数を格納するデバイス番号	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

機能

- FOR~NEXT命令による繰返し処理を強制的に終了し、FOR~NEXT命令以降の処理に実行を移します。



BREAK命令を実行していない場合は、FOR命令で指定した回数分、FOR命令に戻ります。

(1): 強制終了条件

(2): 強制終了条件成立時

- (d)には強制終了した時点の、FOR~NEXT命令における繰返し処理実行回数の残数を格納します。ただし、繰返し処理の残数には、BREAK(P)命令実行時の回数も含まれています。
- BREAK(P)命令は、FOR~NEXT命令間でのみ使用できます。
- BREAK(P)命令は、1つのネスティングに対してのみ使用できます。多重ネスティングを強制終了させる場合は、ネスティング分のBREAK(P)命令を実行させてください。

Point

他シリーズのプログラムを流用する場合、分岐先ポインタに記述した強制終了条件が成立した場合の処理をNEXT命令終了直後に記述してください。

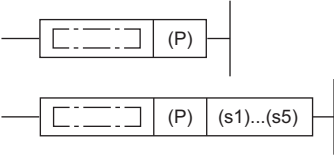
エラー

演算エラーはありません。

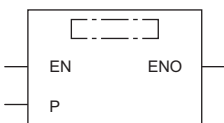
サブルーチンプログラムコール

CALL(P)

指定されたポインタのサブルーチンプログラムを実行します。

ラダー	ST
	対応していません。

FBD/LD



■実行条件

命令	実行条件
CALL	
CALLP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(P)	サブルーチンプログラムの先頭ポインタ番号	—	デバイス名	POINTER
(s1)~(s5)*1	サブルーチンプログラムに引数として渡すデバイス番号	-2147483648~2147483647	ビット/符号付きBIN16 ビット/符号付きBIN32 ビット	ANY
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 FBD/LDでは指定できません。

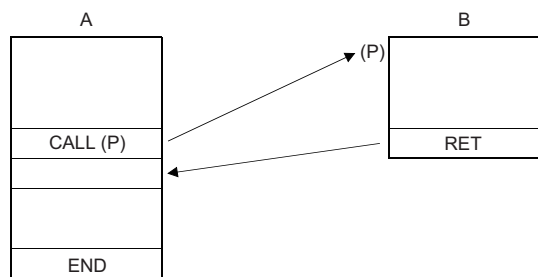
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (P)
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(P)	—	—	—	—	—	—	—	—	—	—	—	○
(s1)~(s5)	○*1	○	○	○	○	○	○	○	○	—	—	—

*1 F以外が使用できます。

機能

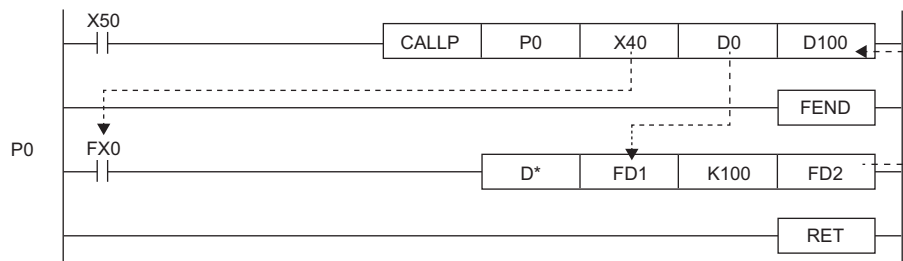
- CALL(P)命令を実行すると、ポインタ(P)のサブルーチンプログラムを実行します。CALL(P)命令は、同一プログラムファイル内のポインタで指定されたサブルーチンプログラムと、グローバルポインタで指定されたサブルーチンプログラムを実行できます。



A: メインルーチンプログラム

B: サブルーチンプログラム

- サブルーチンプログラムでファンクションデバイス(FX, FY, FD)使用時は、(s1)~(s5)でファンクションデバイスに対応するデバイスを指定します。(s1)~(s5)で指定されたデバイスの内容は下記になります。



- サブルーチンプログラム実行前にビットデータの内容がFXに、ワードデータの内容がFDに転送されます。
- サブルーチンプログラム実行後は、FY, FDの内容が対応するデバイスに転送されます。
- FX, FY, FDのデータ転送サイズは、(s1)~(s5)で指定されたデータの種類により異なります。

■デバイスが指定された場合

FX, FYのファンクションデバイスの転送単位は、ビット単位となります。FDのファンクションデバイスの転送単位は、4ワード単位となります。引数で指定されたデバイスの種類により、取り扱うデータサイズが異なります。ファンクションデバイスで指定されたデバイスは、データサイズ分確保できるようにしてください。データサイズ分確保できない場合はエラーとなります。

各ファンクションデバイスのデータサイズは下記になります。

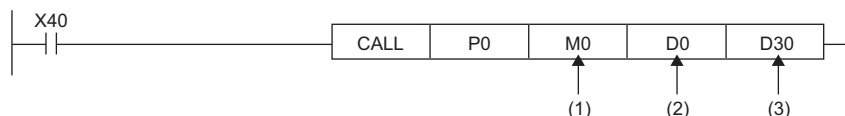
ファンクションデバイス	使用デバイス	(s1)~(s5)からファンクションデバイスに転送した場合のデータサイズ	ファンクションデバイスから(s1)~(s5)に転送した場合のデータサイズ
FX, FY	ビットデバイス	1ビット	1ビット
	ワードデバイスのビット指定時	1ビット	1ビット
FD	ビットデバイスの桁指定時 ^{*1}	4ワード	桁指定のサイズまたは4ワード ^{*2}
	ワードデバイス	4ワード	4ワード

*1 ビットデバイスの桁指定時で、(s1)~(s5)で指定されたデバイス番号が16の倍数ではない場合でも指定できます。

*2 デバイス番号が16の倍数かつ、桁指定がK4またはK8の場合に4ワードサイズを転送します。

例

デバイスを指定した場合、下記の動作になります。



- (1) M0をFX0へ転送します。
- (2) D0~D3をFD1へ転送します。
- (3) D30~D33をFD2へ転送します。

■ラベルが指定された場合

(s1)~(s5)からファンクションデバイスに転送した場合、FX, FYのファンクションデバイスの転送単位は、ビット単位となります。FDのファンクションデバイスの転送単位は、4ワード単位となります。

ファンクションデバイスから(s1)~(s5)に転送した場合、FX, FYのファンクションデバイスの転送単位は、ビット単位となります。FDのファンクションデバイスの転送単位は、ラベルの割付けサイズ単位または桁指定のサイズ単位となります。

各ファンクションデバイスのデータサイズは下記になります。

ファンクションデバイス	ラベルの割付けサイズ	(s1)~(s5)からファンクションデバイスに転送した場合のデータサイズ	ファンクションデバイスから(s1)~(s5)に転送した場合のデータサイズ
FX, FY	1ビット	1ビット	1ビット
FD	2ビット以上	4ワード*1	ラベルの割付けサイズ*2*3/桁指定のサイズ*4

*1 指定されたラベル/構造体の割付けサイズが4ワード未満場合、4ワードに満たないエリアは0を転送します。

*2 指定されたラベル/構造体の割付けサイズが4ワード以上の場合は、4ワード転送します。

*3 下記のラベルを指定した場合、送信するサイズが異なります。当てはまらないラベルを指定した場合は指定したラベルの割付けサイズで転送します。

(1) 配列型ラベル(要素を指定している場合を含む)、構造体型ラベルを指定した場合は、指定したラベル全体のサイズを転送します。

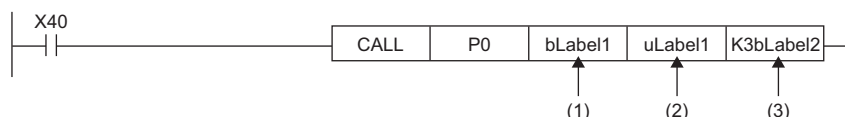
(2) 構造体型ラベルのメンバを指定した場合は、指定したメンバのサイズを転送します。

*4 桁指定がK1~K3の場合は1ワード、K5~K7の場合は2ワード、K4またはK8の場合はラベルの割付けサイズで転送します。

例

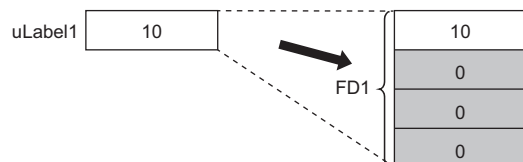
(s1)~(s5)からファンクションデバイスに転送した場合、下記のように動作します。

ラベル名	ラベルの種類
bLabel1	ビット型ラベル
uLabel1	ワード型ラベル
bLabel2	配列ビット型ラベル

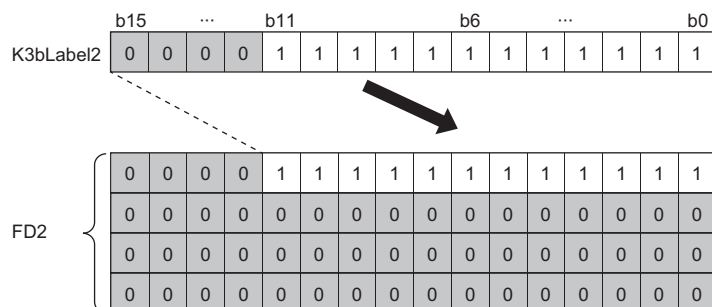


(1) bLabel1(1ビット分)をFX0へ転送します。

(2) uLabel1(1ワード分)をFD1へ転送します。割付けサイズが4ワード未満のため、4ワードに満たないエリアは0を転送します。



(3) K3bLabel2の1ワード分をFD2へ転送します。割付けサイズが4ワード未満のため、4ワードに満たないエリアは0を転送します。



・CALL(P)命令では、(s1)~(s5)まで使用できます。

・サブルーチンプログラムで使用しているファンクションデバイス数と、CALL(P)命令の引数は、同数にしてください。また、ファンクションデバイスとCALL(P)命令の引数の種類は同一にしてください。

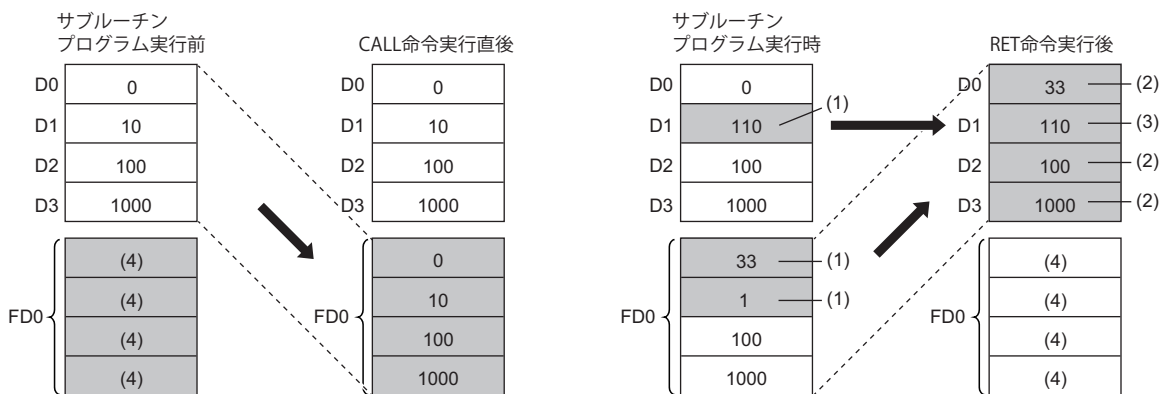
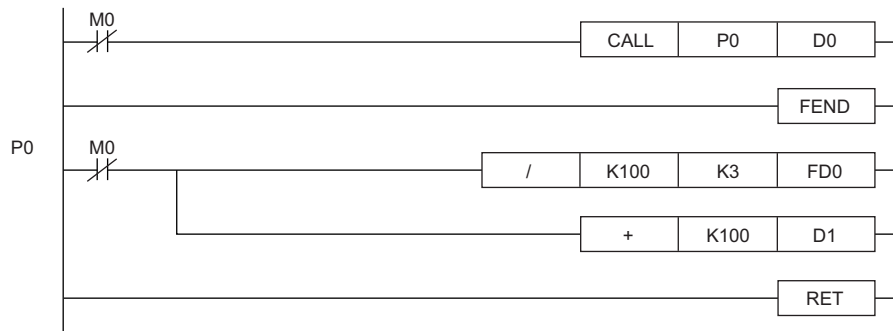
・CALL(P)命令で引数のデバイスにタイマまたはカウンタを使用した場合は、現在値のみ授受します。

・CALL(P)命令にて引数で指定されたデバイス番号は、重複しないでください。重複している場合は正常に演算ができません。

- CALL(P)命令の引数に使用したデバイスを、サブルーチンプログラムで使用しないでください。使用した場合は、正常な演算ができません。

例

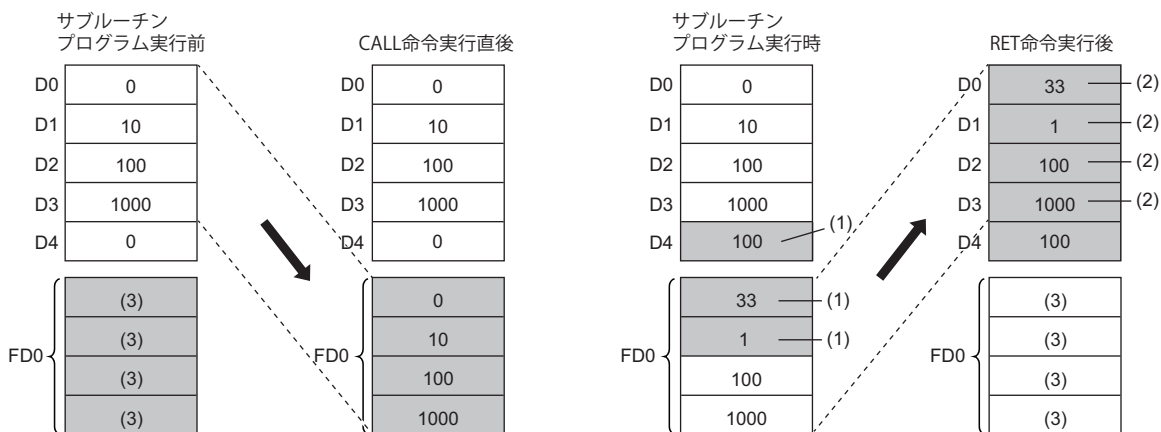
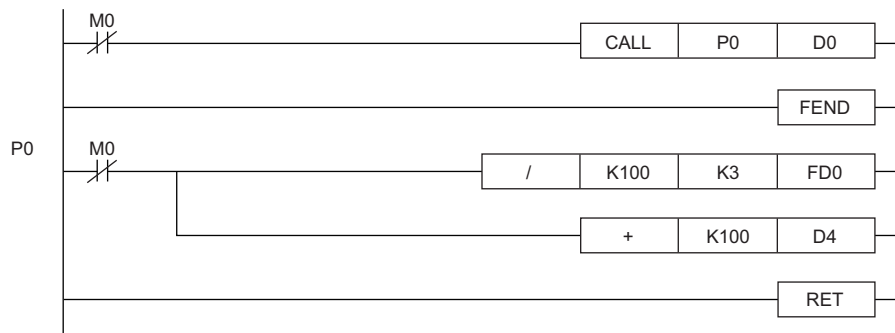
[間違った演算]サブルーチンプログラムのFD0にD0を指定し、サブルーチンプログラムでD1を使用した場合の動作



- (1): サブルーチンプログラムの実行結果が格納されます。
- (2): ファンクションデバイスの値に置き換わります。
- (3): D1はファンクションデバイスの値が反映されません。
- (4): 不定になります。

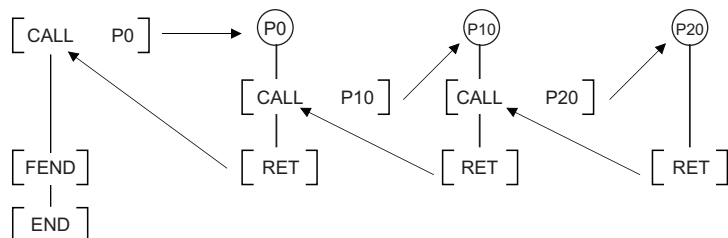
例

[正しい演算] サブルーチンプログラムのFD0にD0を指定し、サブルーチンプログラムでD4を使用した場合の動作



- (1): サブルーチンプログラムの実行結果が格納されます。
- (2): ファンクションデバイスの値に置き換わります。
- (3): 不定になります。

• CALL(P)命令のネスティングは、16重まで可能です。ただしネスティングの16重は、CALL(P)命令、FCALL(P)命令、ECALL命令(P)、EFCALL(P)命令、XCALL命令の合計です。



• サブルーチンプログラム内でONさせたデバイスは、サブルーチンプログラムが非実行でも保持されます。サブルーチンプログラム実行時にONさせたデバイスは、FCALL(P)命令でOFFさせることができます。

注意事項

- FBD/LDプログラムはサブルーチンプログラムとして作成できません。
- FBD/LDでは、引数を持つサブルーチンプログラムの実行はできません。
- (P)で指定したポインタをインデックス修飾した場合、インデックス修飾解決後にFOR~NEXT内に存在するポインタを指定した場合はエラー (エラーコード: 3340H) となります。なお、インデックス修飾せず、直接FOR~NEXT内に存在するポインタを指定した場合は、変換エラーとなります。

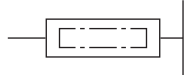
エラー

エラーコード (SD0)	内容
2820H	(s1)~(s5)で引数に指定したデバイスがデータサイズ分確保できないとき。
3330H	17重目のネスティングが実行されたとき。
3340H	(P)で指定したポインタのサブルーチンプログラムが存在しないとき。 または(P)で指定したポインタをインデックス修飾し、インデックス修飾解決後にFOR~NEXT内に存在するポインタが指定されたとき。
3341H	CALL(P)命令実行後、RET命令を実行する前にEND命令、FEND命令、GOEND命令、STOP命令を実行したとき。
3342H	CALL(P)命令を実行する前にRET命令を実行したとき。

サブルーチンプログラムからのリターン

RET

サブルーチンプログラムの終了を示します。

ラダー	ST
	対応していません。

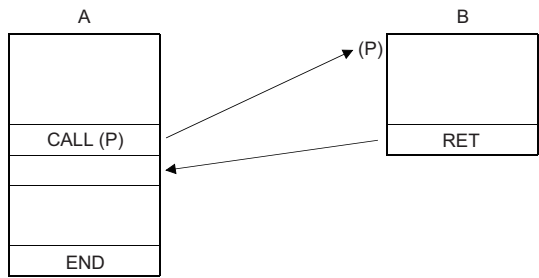
FBD/LD
対応していません。

■実行条件

命令	実行条件
RET	常時実行

機能

- サブルーチンプログラムの終了を示します。
- RET命令を実行すると、サブルーチンプログラムをコールしたCALL(P)命令、FCALL(P)命令、ECALL(P)命令、EFCALL(P)命令、XCALL命令の次ステップに戻ります。



A: メインルーチンプログラム
B: サブルーチンプログラム

エラー

エラーコード (SD0)	内容
3341H	CALL(P)命令、FCALL(P)命令、ECALL(P)命令、EFCALL(P)命令、XCALL命令実行後、RET命令を実行する前にEND命令、FEND命令、GOEND命令、STOP命令を実行したとき。
3342H	CALL(P)命令、FCALL(P)命令、ECALL(P)命令、EFCALL(P)命令、XCALL命令を実行する前にRET命令を実行したとき。

サブルーチンプログラムの出力OFFコール

FCALL(P)

指定されたポインタのサブルーチンプログラムの非実行処理を行います。

ラダー	ST
	対応していません。

FBD/LD

■実行条件

命令	実行条件
FCALL	
FCALLP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(P)	サブルーチンプログラムの先頭ポインタ番号	—	デバイス名	POINTER
(s1)...(s5)*1	サブルーチンプログラムに引数として渡すデバイス番号	-2147483648~2147483647	ビット/符号付きBIN16 ビット/符号付きBIN32 ビット	ANY
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 FBD/LDでは指定できません。

■使用可能デバイス

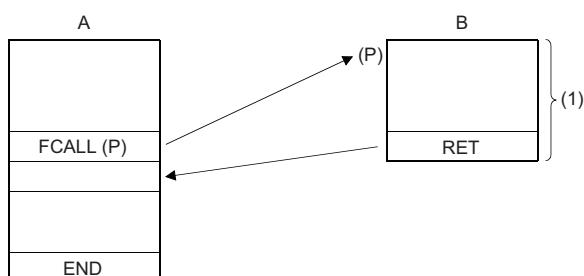
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (P)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(P)	—	—	—	—	—	—	—	—	—	—	—	○
(s1)...(s5)	○*1	○	○	○	○	○	○	○	○	—	—	—

*1 F以外が使用できます。

機能

- FCALL(P)命令を実行すると、ポインタ(P)のサブルーチンプログラムの非実行処理^{*2}を行います。FCALL(P)命令は、同一プログラムファイル内のポインタで指定されたサブルーチンプログラムと、グローバルポインタで指定されたサブルーチンプログラムを非実行にできます。

*2 非実行処理は、各コイル命令を条件設定がOFF状態と同一の処理を行うことです。



A: メインルーチンプログラム

B: サブルーチンプログラム

(1): FCALL(P)命令の指令がON→OFF時に非実行処理を行います。

- 非実行処理後の各コイル命令の演算結果は、条件接点のON/OFFに関係なく下記になります。

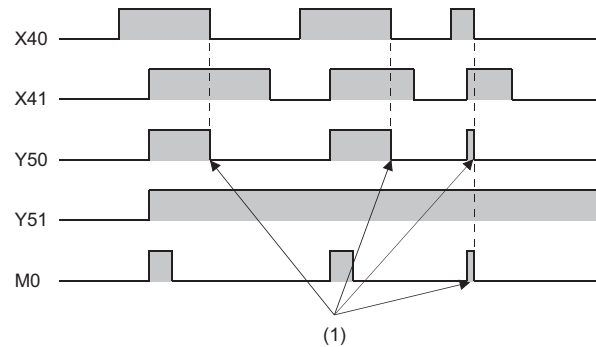
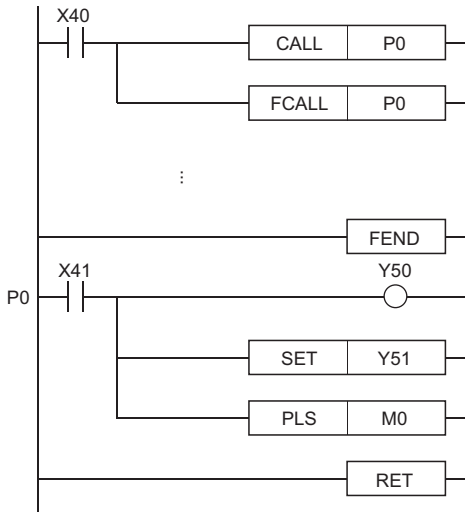
演算で使用するデバイス	演算結果(デバイスの状態)
高速タイマ, 低速タイマ	0となります。
高速積算タイマ, 低速積算タイマ, カウンタ	現在の状態を保持します。
OUT命令中のデバイス	強制的にOFFとなります。
SET命令, RST命令中のデバイス, SFT(P)命令中のデバイス, 基本/応用命令中のデバイス	現在の状態を保持します。
PLS命令, パルス化命令(□P)	条件接点OFFと同一処理となります。

- FCALL(P)命令は、CALL(P)命令と組み合わせて使用します。FCALL(P)命令をCALL(P)命令と組み合わせて実行しない場合は、実行指令をOFFしてもサブルーチンプログラムの非実行処理が行われないので、各コイル命令の出力状態を保持します。

- ・実行指令がOFFするとサブルーチンプログラムの非実行処理を行うため、OUT命令、PLS命令(パルス化命令を含む)を強制的にOFFすることができます。

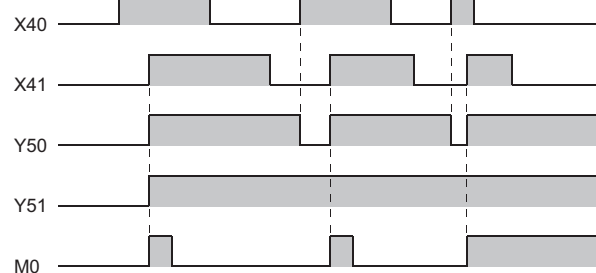
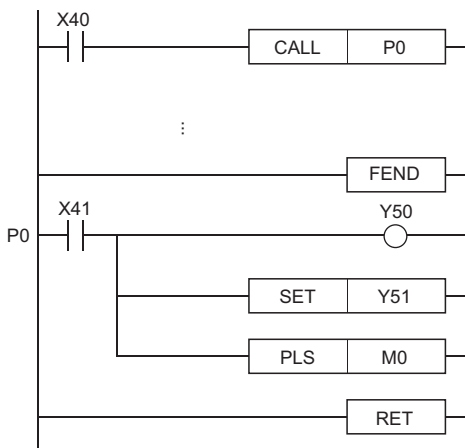
例

FCALL(P)命令を使用するとき

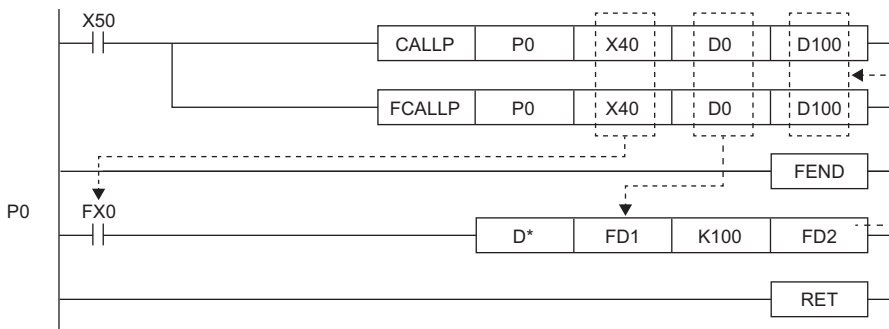


(1): FCALL命令による強制OFF

FCALL(P)命令を使用しないとき



- ・サブルーチンプログラムでファンクションデバイス(FX, FY, FD)使用時は、(s1)~(s5)でファンクションデバイスに対応するデバイスを指定します。(s1)~(s5)で指定されたデバイスの内容は下記になります。



- ・サブルーチンプログラム実行前に、ビットデータの内容がFXに、ワードデータの内容がFDに転送されます。
- ・サブルーチンプログラム実行後は、FY, FDの内容が対応するデバイスに転送されます。
- ・FX, FY, FDのデータ転送サイズは、(s1)~(s5)で指定されたデータの種類により異なります。

■デバイスが指定された場合

FX, FYのファンクションデバイスの転送単位は、ビット単位となります。FDのファンクションデバイスの転送単位は、4ワード単位となります。引数で指定されたデバイスの種類により、取り扱うデータサイズが異なります。ファンクションデバイスで指定されたデバイスは、データサイズ分確保できるようにしてください。データサイズ分確保できない場合はエラーとなります。

各ファンクションデバイスのデータサイズは下記になります。

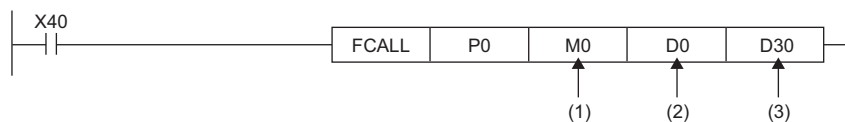
ファンクションデバイス	使用デバイス	(s1)~(s5)からファンクションデバイスに転送した場合のデータサイズ	ファンクションデバイスから(s1)~(s5)に転送した場合のデータサイズ
FX, FY	ビットデバイス	1ビット	1ビット
	ワードデバイスのビット指定時	1ビット	1ビット
FD	ビットデバイスの桁指定時 ^{*1}	4ワード	桁指定のサイズまたは4ワード ^{*2}
	ワードデバイス	4ワード	4ワード

*1 ビットデバイスの桁指定時で、(s1)~(s5)で指定されたデバイス番号が16の倍数ではない場合でも指定できます。

*2 デバイス番号が16の倍数かつ、桁指定がK4またはK8の場合に4ワードサイズを転送します。

例

デバイスを指定した場合、下記の動作になります。



(1) M0をFX0へ転送します。

(2) D0~D3をFD1へ転送します。

(3) D30~D33をFD2へ転送します。

■ラベルが指定された場合

(s1)~(s5)からファンクションデバイスに転送した場合、FX, FYのファンクションデバイスの転送単位は、ビット単位となります。FDのファンクションデバイスの転送単位は、4ワード単位となります。

ファンクションデバイスから(s1)~(s5)に転送した場合、FX, FYのファンクションデバイスの転送単位は、ビット単位となります。FDのファンクションデバイスの転送単位は、ラベルの割付けサイズ単位または桁指定のサイズ単位となります。

各ファンクションデバイスのデータサイズは下記になります。

ファンクションデバイス	ラベルの割付けサイズ	(s1)~(s5)からファンクションデバイスに転送した場合のデータサイズ	ファンクションデバイスから(s1)~(s5)に転送した場合のデータサイズ
FX, FY	1ビット	1ビット	1ビット
FD	2ビット以上	4ワード ^{*1}	ラベルの割付けサイズ ^{*2*3} /桁指定のサイズ ^{*4}

*1 指定されたラベル/構造体の割付けサイズが4ワード未満場合、4ワードに満たないエリアは0を転送します。

*2 指定されたラベル/構造体の割付けサイズが4ワード以上の場合は、4ワード転送します。

*3 下記のラベルを指定した場合、送信するサイズが異なります。当てはまらないラベルを指定した場合は指定したラベルの割付けサイズで転送します。

(1) 配列型ラベル(要素を指定している場合を含む)、構造体型ラベルを指定した場合は、指定したラベル全体のサイズを転送します。

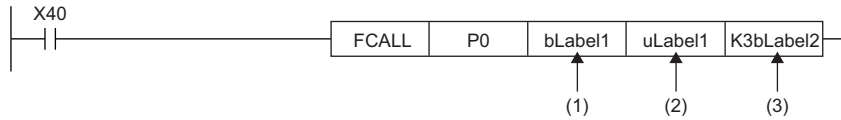
(2) 構造体型ラベルのメンバを指定した場合は、指定したメンバのサイズを転送します。

*4 桁指定がK1~K3の場合は1ワード、K5~K7の場合は2ワード、K4またはK8の場合はラベルの割付けサイズで転送します。

例

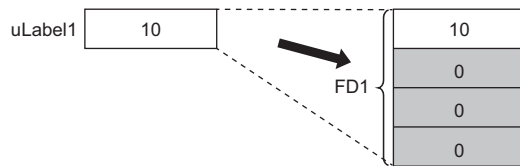
(s1)~(s5)からファンクションデバイスに転送した場合、下記のように動作します。

ラベル名	ラベルの種類
bLabel1	ビット型ラベル
uLabel1	ワード型ラベル
bLabel2	配列ビット型ラベル

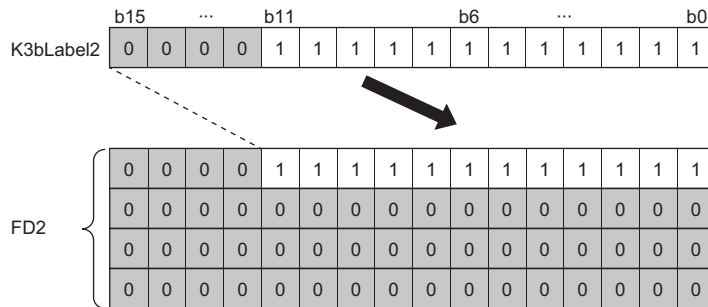


(1) bLabel1(1ビット分)をFX0へ転送します。

(2) uLabel1(1ワード分)をFD1へ転送します。割付けサイズが4ワード未満のため、4ワードに満たないエリアは0を転送します。

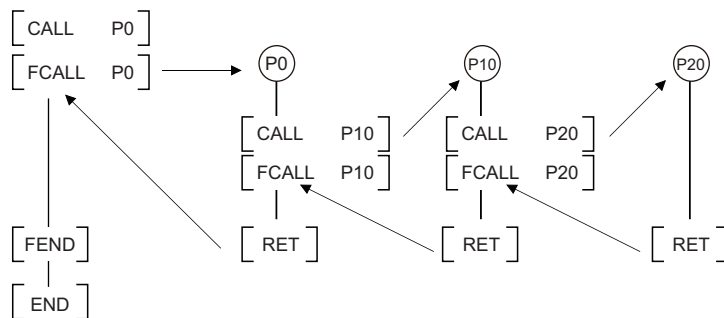


(3) K3bLabel2の1ワード分をFD2へ転送します。割付けサイズが4ワード未満のため、4ワードに満たないエリアは0を転送します。



• FCALL(P)命令では、(s1)~(s5)まで使用できます。

• FCALL(P)命令のネスティングは、16重まで可能です。ただし、ネスティングの16重は、CALL(P)命令、FCALL(P)命令、ECALL(P)命令、EFCALL(P)命令、XCALL命令の合計です。



注意事項

- FBD/LDプログラムはサブルーチンプログラムとして作成できません。
- FBD/LDでは、引数を持つサブルーチンプログラムの実行はできません。

エラー

エラーコード (SD0)	内容
2820H	(s1)~(s5)で引数に指定したデバイスがデータサイズ分確保できないとき。
3330H	17重目のネスティングが実行されたとき。
3340H	FCALL(P)命令で指定したポインタのサブルーチンプログラムが存在しないとき。 または(P)で指定したポインタをインデックス修飾し、インデックス修飾解決後にFOR~NEXT内に存在するポインタが指定されたとき。
3341H	FCALL(P)命令実行後、RET命令を実行する前にEND命令、FEND命令、GOEND命令、STOP命令を実行したとき。
3342H	FCALL(P)命令を実行する前にRET命令を実行したとき。

プログラムファイル間サブルーチンコール

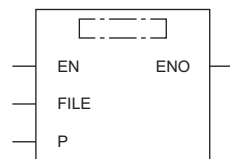
ECALL(P)

指定プログラムファイル名の指定されたポインタのサブルーチンプログラムを実行します。

ラダー	ST
	対応していません。

FILE: ファイル名

FBD/LD



FILE: ファイル名

■実行条件

命令	実行条件
ECALL	
ECALLP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(ファイル名)	コールするプログラムファイル名	—	Unicode文字列	ANYSTRING_DOUBLE
(P)	サブルーチンプログラムの先頭ポインタ番号	—	デバイス名	POINTER
(s1)...(s5)*1	サブルーチンプログラムに引数として渡すデバイス番号	-2147483648~2147483647	ビット/符号付きBIN16 ビット/符号付きBIN32 ビット	ANY
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 FBD/LDでは指定できません。

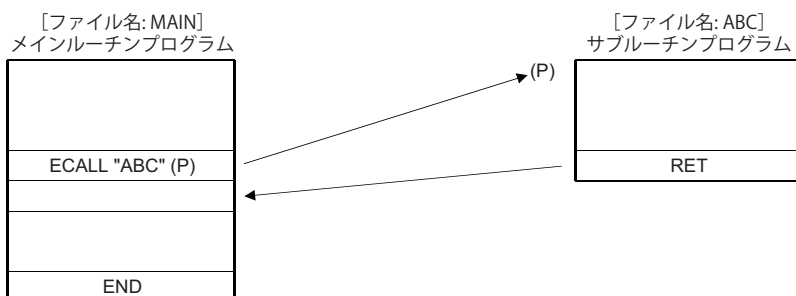
■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他 (P)
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(ファイル名)	—	—	○	—	—	—	—	○	—	—	—	○	—
(P)	—	—	—	—	—	—	—	—	—	—	—	—	○
(s1)...(s5)	○*1	○	○	○	○	○	○	○	○	○	—	—	—

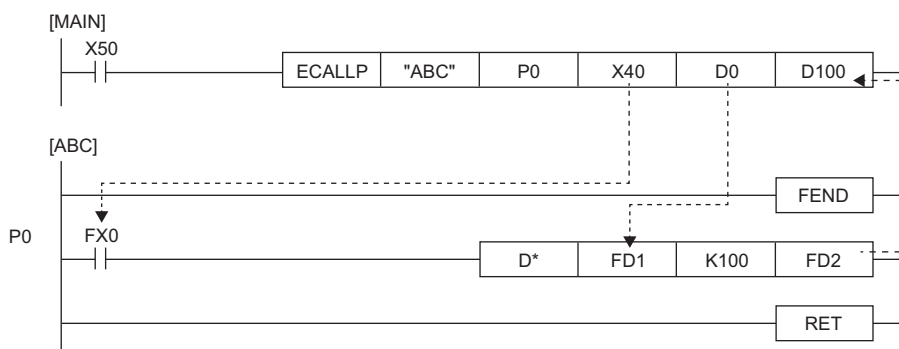
*1 F以外が使用できます。

機能

- ECALL(P)命令を実行すると、指定プログラムファイル名のポインタ(P)のサブルーチンプログラムを実行します。
ECALL(P)命令では、他のプログラムファイルのローカルポインタを使用したサブルーチンプログラムをコールすることもできます。



- ファイル名には、プログラムメモリに格納されているプログラムファイルのみを指定できます。
- ファイル名には拡張子(.PRG)を指定する必要はありません。(PRGのファイルのみを対象とします。)
- サブルーチンプログラムでファンクションデバイス(FX, FY, FD)使用時は、(s1)~(s5)でファンクションデバイスに対応するデバイスを指定します。(s1)~(s5)で指定されたデバイスの内容は下記になります。



- サブルーチンプログラム実行前に、ビットデータの内容がFXに、ワードデータの内容がFDに転送されます。
- サブルーチンプログラム実行後は、FY, FDの内容が対応するデバイスに転送されます。
- FX, FY, FDのデータ転送サイズは、(s1)~(s5)で指定されたデータの種類により異なります。

■デバイスが指定された場合

FX, FYのファンクションデバイスの転送単位は、ビット単位となります。FDのファンクションデバイスの転送単位は、4ワード単位となります。引数で指定されたデバイスの種類により、取り扱うデータサイズが異なります。ファンクションデバイスで指定されたデバイスは、データサイズ分確保できるようにしてください。データサイズ分確保できない場合はエラーとなります。

各ファンクションデバイスのデータサイズは下記になります。

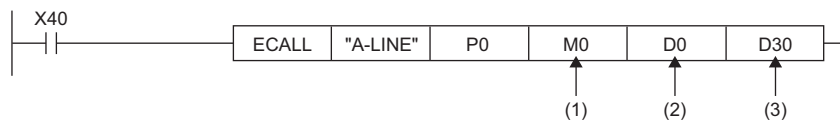
ファンクションデバイス	使用デバイス	(s1)~(s5)からファンクションデバイスに転送した場合のデータサイズ	ファンクションデバイスから(s1)~(s5)に転送した場合のデータサイズ
FX, FY	ビットデバイス	1ビット	1ビット
	ワードデバイスのビット指定時	1ビット	1ビット
FD	ビットデバイスの桁指定時 ^{*1}	4ワード	桁指定のサイズまたは4ワード ^{*2}
	ワードデバイス	4ワード	4ワード

*1 ビットデバイスの桁指定時で、(s1)~(s5)で指定されたデバイス番号が16の倍数ではない場合でも指定できます。

*2 デバイス番号が16の倍数かつ、桁指定がK4またはK8の場合に4ワードサイズを転送します。

例

デバイスを指定した場合、下記の動作になります。



(1) M0をFX0へ転送します。

(2) D0~D3をFD1へ転送します。

(3) D30~D33をFD2へ転送します。

■ラベルが指定された場合

(s1)~(s5)からファンクションデバイスに転送した場合、FX, FYのファンクションデバイスの転送単位は、ビット単位となります。FDのファンクションデバイスの転送単位は、4ワード単位となります。

ファンクションデバイスから(s1)~(s5)に転送した場合、FX, FYのファンクションデバイスの転送単位は、ビット単位となります。FDのファンクションデバイスの転送単位は、ラベルの割付けサイズ単位または桁指定のサイズ単位となります。

各ファンクションデバイスのデータサイズは下記になります。

ファンクションデバイス	ラベルの割付けサイズ	(s1)~(s5)からファンクションデバイスに転送した場合のデータサイズ	ファンクションデバイスから(s1)~(s5)に転送した場合のデータサイズ
FX, FY	1ビット	1ビット	1ビット
FD	2ビット以上	4ワード*1	ラベルの割付けサイズ*2*3/桁指定のサイズ*4

*1 指定されたラベル/構造体の割付けサイズが4ワード未満場合、4ワードに満たないエリアは0を転送します。

*2 指定されたラベル/構造体の割付けサイズが4ワード以上の場合、4ワード転送します。

*3 下記のラベルを指定した場合、送信するサイズが異なります。当てはまらないラベルを指定した場合は指定したラベルの割付けサイズで転送します。

(1) 配列型ラベル(要素を指定している場合を含む)、構造体型ラベルを指定した場合は、指定したラベル全体のサイズを転送します。

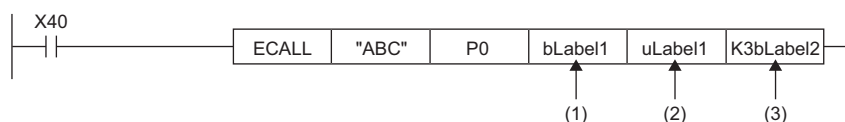
(2) 構造体型ラベルのメンバを指定した場合は、指定したメンバのサイズを転送します。

*4 桁指定がK1~K3の場合は1ワード、K5~K7の場合は2ワード、K4またはK8の場合はラベルの割付けサイズで転送します。

例

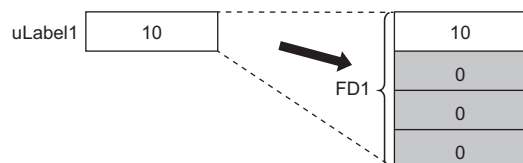
(s1)~(s5)からファンクションデバイスに転送した場合、下記のように動作します。

ラベル名	ラベルの種類
bLabel1	ビット型ラベル
uLabel1	ワード型ラベル
bLabel2	配列ビット型ラベル

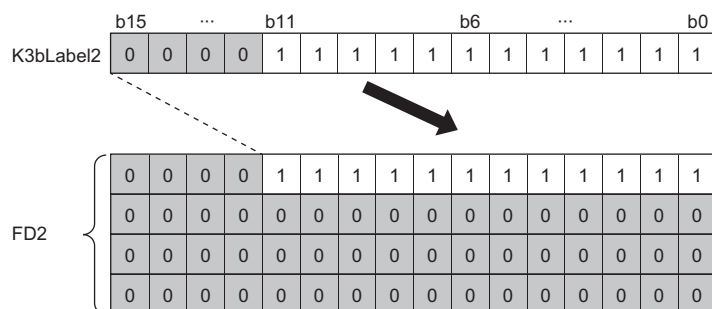


(1) bLabel1(1ビット分)をFX0へ転送します。

(2) uLabel1(1ワード分)をFD1へ転送します。割付けサイズが4ワード未満のため、4ワードに満たないエリアは0を転送します。



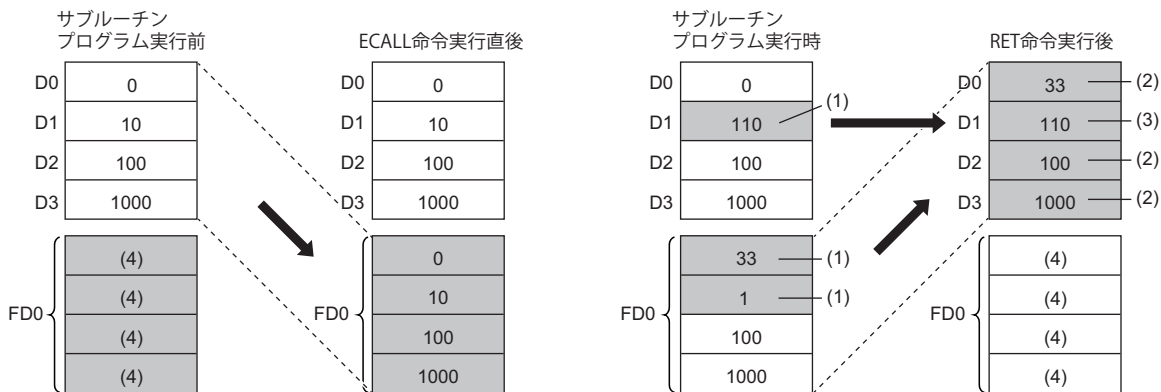
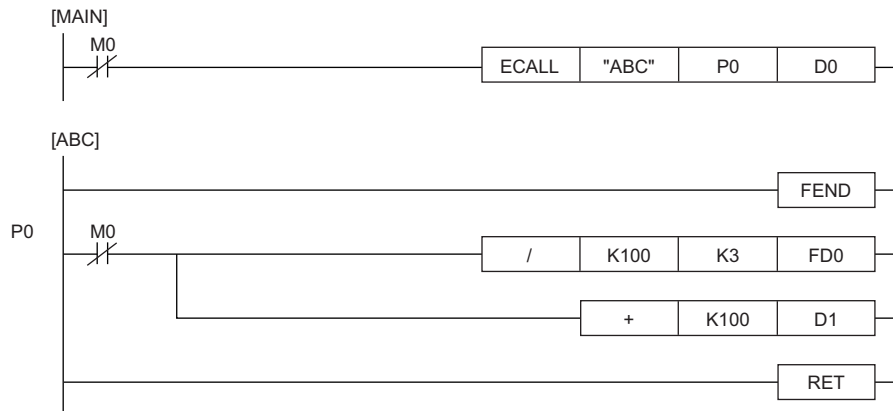
(3) K3bLabel2の1ワード分をFD2へ転送します。割付けサイズが4ワード未満のため、4ワードに満たないエリアは0を転送します。



- ECALL(P)命令では、(s1)~(s5)まで使用できます。
- ECALL(P)命令の引数に使用したデバイスを、サブルーチンプログラムで使用しないでください。ECALL(P)命令の引数に使用したデバイスを、サブルーチンプログラムで使用した場合には、正常な演算ができません。

例

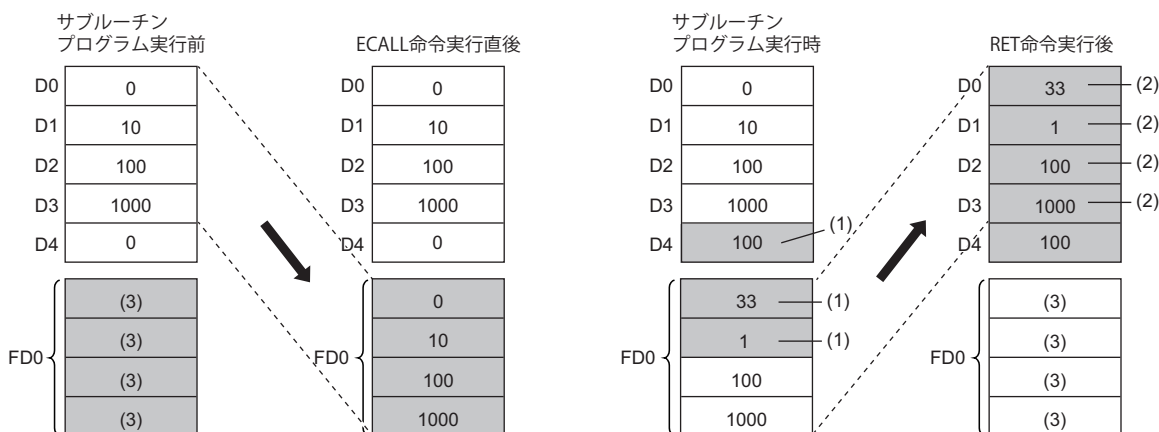
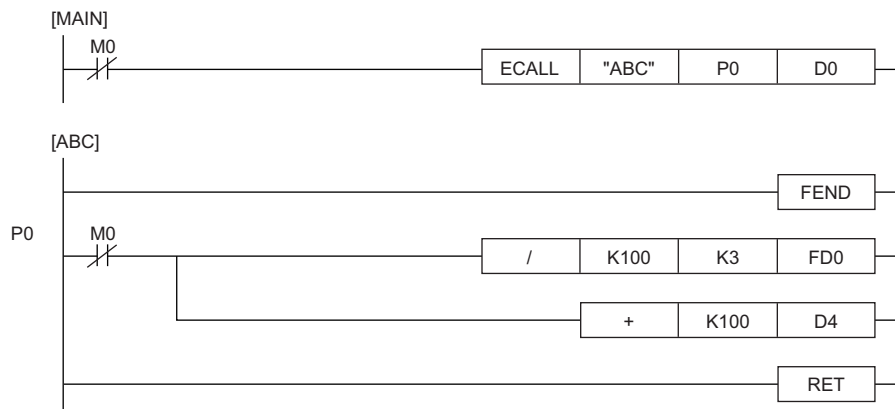
[間違った演算]サブルーチンプログラムのFD0にD0を指定し、サブルーチンプログラムでD1を使用した場合の動作



- (1): サブルーチンプログラムの実行結果が格納されます。
 (2): ファンクションデバイスの値に置き換わります。
 (3): D1はファンクションデバイスの値が反映されません。
 (4): 不定になります。

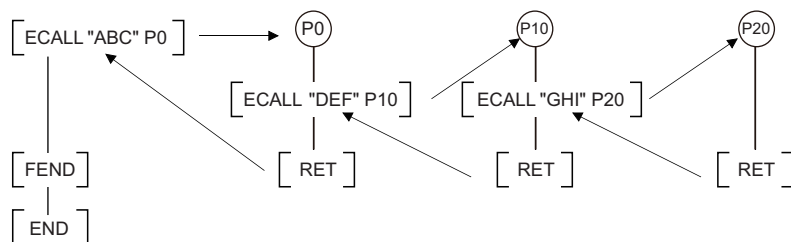
例

[正しい演算] サブルーチンプログラムのFD0にD0を指定し、サブルーチンプログラムでD4を使用した場合の動作



- (1): サブルーチンプログラムの実行結果が格納されます。
 (2): ファンクションデバイスの値に置き換わります。
 (3): 不定になります。

- ECALL(P)命令で引数で指定されたデバイス番号は、重複しないでください。重複している場合は、正常に演算ができません。
- ECALL(P)命令のネスティングは16重まで可能です。ただし、ネスティングの16重は、CALL(P)命令、FCALL(P)命令、ECALL(P)命令、EFCALL(P)命令、XCALL命令の合計です。



- サブルーチンプログラム内でONさせたデバイスは、サブルーチンプログラムが非実行でも保持されます。サブルーチンプログラム実行時にONさせたデバイスは、EFCALL(P)命令でOFFさせることができます。

注意事項

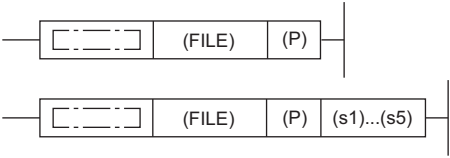
- FBD/LDプログラムはサブルーチンプログラムとして作成できません。
- FBD/LDでは、引数を持つサブルーチンプログラムの実行はできません。
- (P)で指定したポインタをインデックス修飾した場合、インデックス修飾解決後にFOR~NEXT内に存在するポインタを指定した場合はエラー(エラーコード: 3340H)となります。なお、インデックス修飾せず、直接FOR~NEXT内に存在するポインタを指定した場合は、変換エラーとなります。

エラーコード (SD0)	内容
2820H	(s1)~(s5)で引数に指定したデバイスがデータサイズ分確保できないとき。
2840H	(ファイル名)で指定のファイルが存在しないとき。
2841H	(ファイル名)で指定のファイルが実行できないとき。
2842H	(ファイル名)で指定したファイル名のプログラム種別が対応していないプログラムのとき。
3330H	17重目のネスティングが実行されたとき。
3340H	(P)で指定したポインタの、サブルーチンプログラムが存在しないとき。 または(P)で指定したポインタをインデックス修飾し、インデックス修飾解決後にFOR~NEXT内に存在するポインタが指定されたとき。
3341H	ECALL(P)命令実行後、RET命令を実行する前にEND命令、FEND命令、GOEND命令、STOP命令を実行したとき。
3342H	ECALL(P)命令を実行する前にRET命令を実行したとき。

プログラムファイル間サブルーチンプログラム出力OFFコール

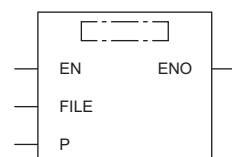
EFCALL(P)

指定プログラムファイル名の指定されたポインタのサブルーチンプログラムの非実行処理を行います。

ラダー	ST
	対応していません。

FILE: ファイル名

FBD/LD



FILE: ファイル名

■実行条件

命令	実行条件
EFCALL	
EFCALLP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(ファイル名)	非実行処理するプログラムファイル名	—	Unicode文字列	ANYSTRING_DOUBLE
(P)	サブルーチンプログラムの先頭ポインタ番号	—	デバイス名	POINTER
(s1)~(s5)*1	サブルーチンプログラムに引数として渡すデバイス番号	-2147483648~2147483647	ビット/符号付きBIN16 ビット/符号付きBIN32 ビット	ANY
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 FBD/LDでは指定できません。

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他 (P)
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(ファイル名)	—	—	○	—	—	—	—	○	—	—	○	—	—
(P)	—	—	—	—	—	—	—	—	—	—	—	—	○
(s1)...(s5)	○*1	○	○	○	○	○	○	○	○	○	—	—	—

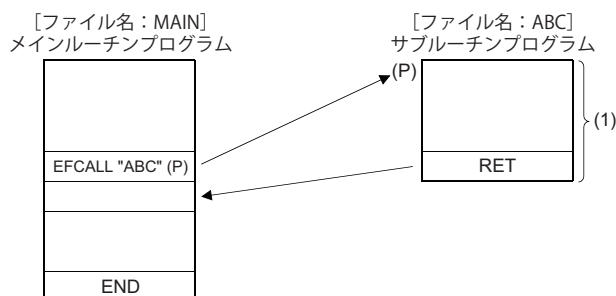
*1 F以外が使用できます。

機能

- EFCALL(P)命令を実行すると、(P)で指定されたポインタのサブルーチンプログラムの非実行処理^{*2}を行います。
EFCALL(P)命令では、他のプログラムファイルのローカルポインタを使用したサブルーチンプログラムをコールすることもできます。

*2 非実行処理は、各コイル命令を条件接点がOFF状態と同一の処理を行うことです。

(1) EFCALL(P)命令の指令がON→OFF時に非実行処理を行います。



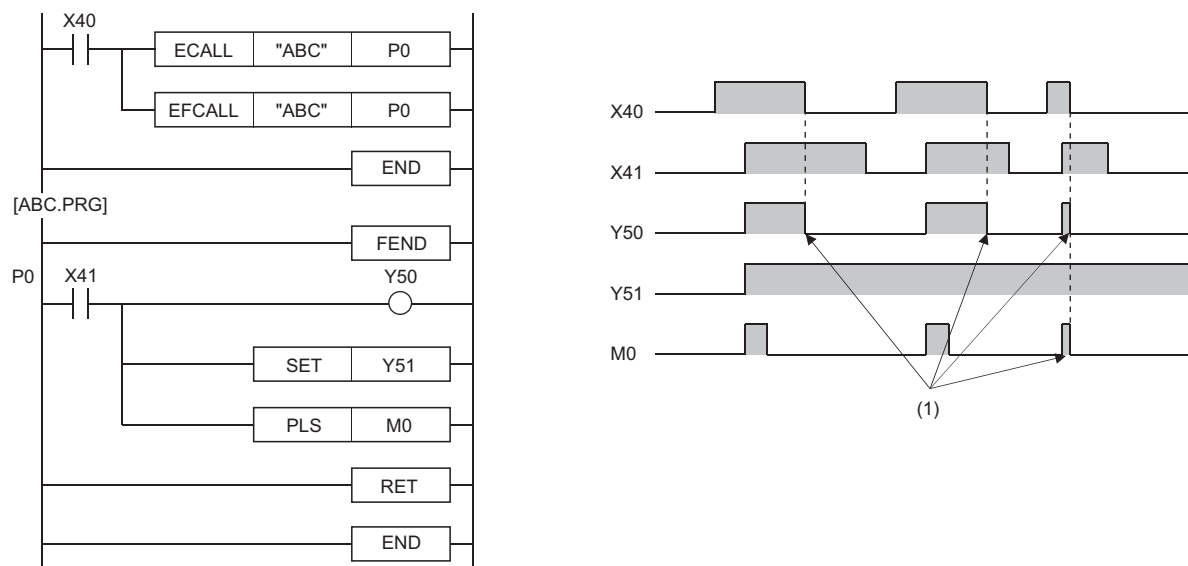
- 非実行処理後の各コイル命令の演算結果は、条件接点のON/OFFに関係なく下記になります。

演算で使用するデバイス	演算結果(デバイスの状態)
高速タイマ、低速タイマ	0となります。
高速積算タイマ、低速積算タイマ、カウンタ	現在の状態を保持します。
OUT命令中のデバイス	強制的にOFFとなります。
SET命令、RST命令中のデバイス、SFT(P)命令中のデバイス、基本/応用命令中のデバイス	現在の状態を保持します。
PLS命令、パルス化命令(□P)	条件接点OFFと同一処理となります。

- EFCALL(P)命令は、ECALL(P)命令と組み合わせて使用します。ECALL(P)命令と組み合わせて実行しない場合は、実行指令をOFFしてもサブルーチンプログラムの非実行処理が行われないので、各コイル命令の出力状態を保持します。
- 実行指令がOFFするとサブルーチンプログラムの非実行処理を行うため、OUT命令、PLS命令(パルス化命令を含む)を強制的にOFFすることができます。

例

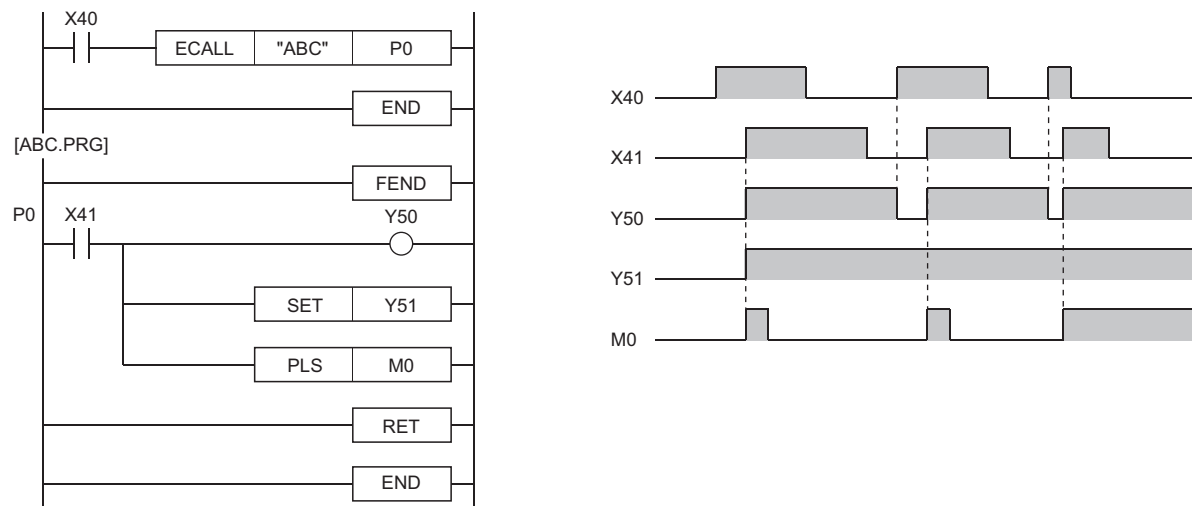
EFCALL命令を使用するとき



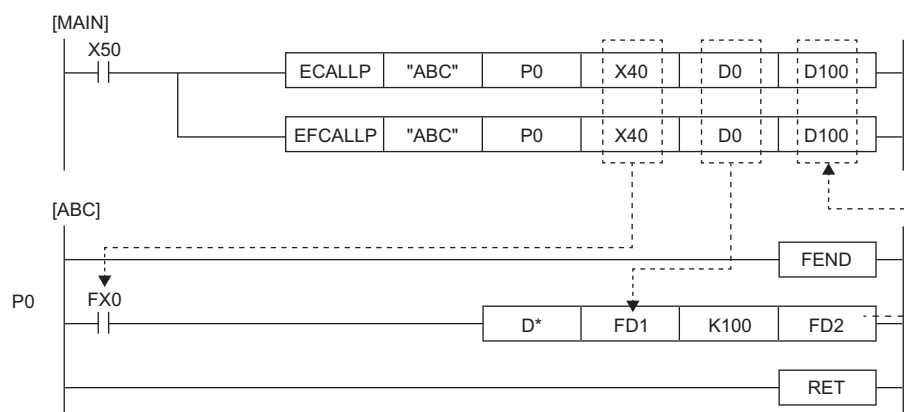
(1): EFCALL命令による強制OFF

例

EFCALL命令を使用しないとき



- ファイル名には、プログラムメモリに格納されているプログラムファイルのみを指定できます。
- ファイル名には拡張子(.PRG)を指定する必要はありません。(.PRGのファイルのみを対象とします。)
- サブルーチンプログラムでファンクションデバイス(FX, FY, FD)使用時は、(s1)~(s5)でファンクションデバイスに対応するデバイスを指定します。(s1)~(s5)で指定されたデバイスの内容は下記になります。



- サブルーチンプログラム実行前に、ビットデータの内容がFXに、ワードデータの内容がFDに転送されます。
- サブルーチンプログラム実行後は、FY, FDの内容が対応するデバイスに転送されます。
- FX, FY, FDのデータ転送サイズは、(s1)~(s5)で指定されたデータの種類により異なります。

■デバイスが指定された場合

FX, FYのファンクションデバイスの転送単位は、ビット単位となります。FDのファンクションデバイスの転送単位は、4ワード単位となります。引数で指定されたデバイスの種類により、取り扱うデータサイズが異なります。ファンクションデバイスで指定されたデバイスは、データサイズ分確保できるようにしてください。データサイズ分確保できない場合はエラーとなります。

各ファンクションデバイスのデータサイズは下記になります。

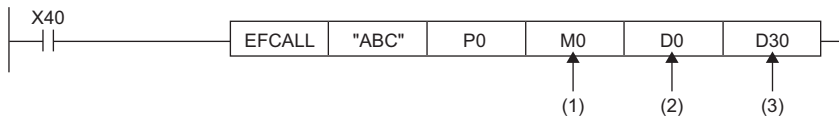
ファンクションデバイス	使用デバイス	(s1)~(s5)からファンクションデバイスに転送した場合のデータサイズ	ファンクションデバイスから(s1)~(s5)に転送した場合のデータサイズ
FX, FY	ビットデバイス	1ビット	1ビット
	ワードデバイスのビット指定時	1ビット	1ビット
FD	ビットデバイスの桁指定時 ^{*1}	4ワード	桁指定のサイズまたは4ワード ^{*2}
	ワードデバイス	4ワード	4ワード

*1 ビットデバイスの桁指定時で、(s1)~(s5)で指定されたデバイス番号が16の倍数ではない場合でも指定できます。

*2 デバイス番号が16の倍数かつ、桁指定がK4またはK8の場合に4ワードサイズを転送します。

例

デバイスを指定した場合、下記の動作になります。



- (1) M0をFX0へ転送します。
 (2) D0~D3をFD1へ転送します。
 (3) D30~D33をFD2へ転送します。

■ラベルが指定された場合

(s1)~(s5)からファンクションデバイスに転送した場合、FX, FYのファンクションデバイスの転送単位は、ビット単位となります。FDのファンクションデバイスの転送単位は、4ワード単位となります。

ファンクションデバイスから(s1)~(s5)に転送した場合、FX, FYのファンクションデバイスの転送単位は、ビット単位となります。FDのファンクションデバイスの転送単位は、ラベルの割付けサイズ単位または桁指定のサイズ単位となります。

各ファンクションデバイスのデータサイズは下記になります。

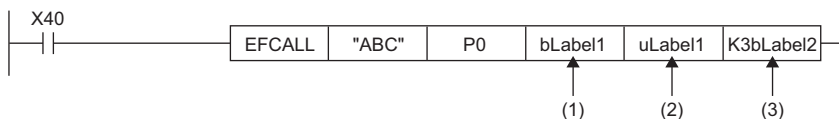
ファンクションデバイス	ラベルの割付けサイズ	(s1)~(s5)からファンクションデバイスに転送した場合のデータサイズ	ファンクションデバイスから(s1)~(s5)に転送した場合のデータサイズ
FX, FY	1ビット	1ビット	1ビット
FD	2ビット以上	4ワード*1	ラベルの割付けサイズ*2*3/桁指定のサイズ*4

- *1 指定されたラベル/構造体の割付けサイズが4ワード未満場合、4ワードに満たないエリアは0を転送します。
 *2 指定されたラベル/構造体の割付けサイズが4ワード以上の場合は、4ワード転送します。
 *3 下記のラベルを指定した場合、送信するサイズが異なります。当てはまらないラベルを指定した場合は指定したラベルの割付けサイズで転送します。
 (1) 配列型ラベル(要素を指定している場合を含む)、構造体型ラベルを指定した場合は、指定したラベル全体のサイズを転送します。
 (2) 構造体型ラベルのメンバを指定した場合は、指定したメンバのサイズを転送します。
 *4 桁指定がK1~K3の場合は1ワード、K5~K7の場合は2ワード、K4またはK8の場合はラベルの割付けサイズで転送します。

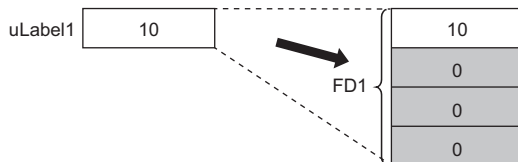
例

(s1)~(s5)からファンクションデバイスに転送した場合、下記のように動作します。

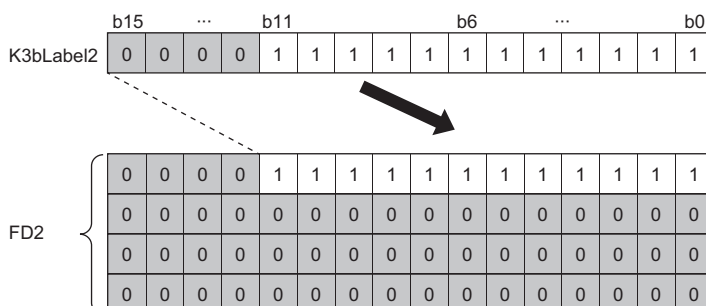
ラベル名	ラベルの種類
bLabel1	ビット型ラベル
uLabel1	ワード型ラベル
bLabel2	配列ビット型ラベル



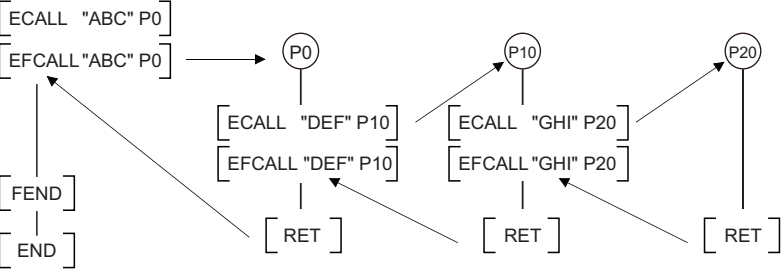
- (1) bLabel1(1ビット分)をFX0へ転送します。
 (2) uLabel1(1ワード分)をFD1へ転送します。割付けサイズが4ワード未満のため、4ワードに満たないエリアは0を転送します。



- (3) K3bLabel2の1ワード分をFD2へ転送します。割付けサイズが4ワード未満のため、4ワードに満たないエリアは0を転送します。



- EFCALL(P)命令では、(s1)~(s5)まで使用できます。
- サブルーチンプログラムで使用しているファンクションデバイス数と、EFCALL(P)命令の引数は、同数にする必要があります。またファンクションデバイスとEFCALL(P)命令の引数の種類は同一にしてください。
- EFCALL(P)命令のネスティングは16重まで可能です。ただし、ネスティングの16重は、CALL(P)命令、FCALL(P)命令、ECALL(P)命令、EFCALL(P)命令、XCALL命令の合計です。



注意事項

- FBD/LDプログラムはサブルーチンプログラムとして作成できません。
- FBD/LDでは、引数を持つサブルーチンプログラムの実行はできません。
- (P)で指定したポインタをインデックス修飾した場合、インデックス修飾解決後にFOR~NEXT内に存在するポインタを指定した場合はエラー (エラーコード: 3340H) となります。なお、インデックス修飾せず、直接FOR~NEXT内に存在するポインタを指定した場合は、変換エラーとなります。

エラー

エラーコード (SD0)	内容
2820H	(s1)~(s5)で引数に指定したデバイスがデータサイズ分確保できないとき。
2840H	(ファイル名)で指定のファイルが存在しないとき。
2841H	(ファイル名)で指定のファイルが実行できないとき。
3330H	17重目のネスティングが実行されたとき。
3340H	(P)で指定したポインタの、サブルーチンプログラムが存在しないとき。 または(P)で指定したポインタをインデックス修飾し、インデックス修飾解決後にFOR~NEXT内に存在するポインタが指定されたとき。
3341H	EFCALL(P)命令実行後、RET命令を実行する前にEND命令、FEND命令、GOEND命令、STOP命令を実行したとき。
3342H	EFCALL(P)命令を実行する前にRET命令を実行したとき。

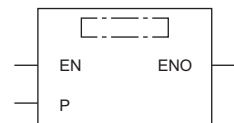
サブルーチンプログラムコール

XCALL

サブルーチンプログラムの実行と非実行処理を行います。条件が成立時はサブルーチンCALLし、条件がON→OFF時はサブルーチンをFCALLします。

ラダー	ST
	対応していません。

FBD/LD



■実行条件

命令	実行条件
XCALL	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(P)	サブルーチンプログラムの先頭ポインタ番号	—	デバイス名	POINTER
(s1)...(s5)*1	サブルーチンプログラムに引数として渡すデバイス番号	-2147483648~2147483647	ビット/符号付きBIN16 ビット/符号付きBIN32 ビット	ANY
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 FBD/LDでは指定できません。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他 (P)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(P)	—	—	—	—	—	—	—	—	—	—	○
(s1)...(s5)	○*1	○	○	○	○	○	○	○	—	—	—

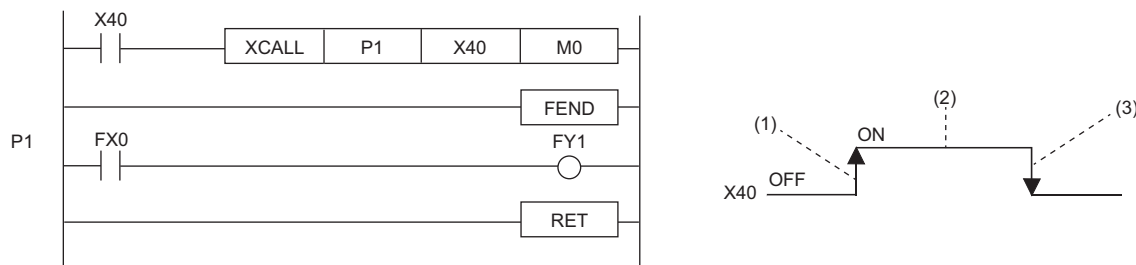
*1 F以外が使用できます。

機能

- XCALL命令は、サブルーチンプログラムの実行と非実行処理を行う命令です。
- サブルーチンプログラムの実行は、各コイル命令を条件接点のON/OFF状態より演算します。
- サブルーチンプログラムの非実行処理は、各コイル命令を条件接点がOFF状態と同一の処理を行います。
- 非実行処理後の各コイル命令の演算結果は、条件接点のON/OFFに関係なく下記の表となります。

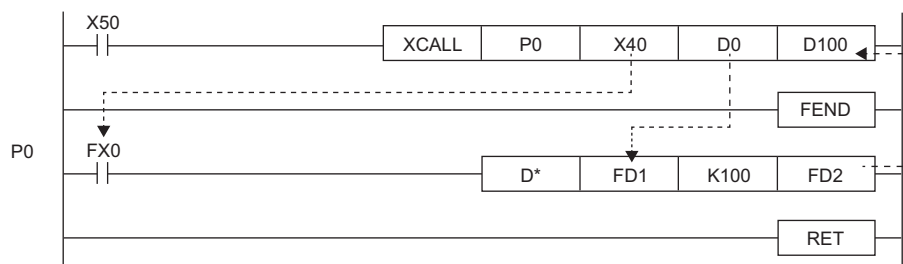
演算で使用するデバイス	演算結果(デバイスの状態)
高速タイマ、低速タイマ	0となります。
高速積算タイマ、低速積算タイマ、カウンタ	現在の状態を保持します。
OUT命令中のデバイス	強制的にOFFとなります。
SET命令、RST命令中のデバイス、SFT(P)命令中のデバイス、基本/応用命令中のデバイス	現在の状態を保持します。
PLS命令、パルス化命令(POP)	条件接点OFFと同一処理となります。

- XCALL命令の動作は下記になります。



- (1) X40の立上り(OFF→ON): P1のサブルーチンプログラムを実行します。
- (2) X40のON中: P1のサブルーチンプログラムを実行します。(X40のON中には、X40の立上りを含みません。)
- (3) X40の立下り(ON→OFF): P1のサブルーチンプログラムの非実行処理を行います。

- サブルーチンプログラムでファンクションデバイス(FX, FY, FD)使用時は、(s1)~(s5)でファンクションデバイスに対応するデバイスを指定します。(s1)~(s5)で指定されたデバイスの内容は下記になります。



- サブルーチンプログラム実行前に、ビットデータの内容がFXに、ワードデータの内容がFDに転送されます。
- サブルーチンプログラム実行後は、FY, FDの内容が対応するデバイスに転送されます。
- FX, FY, FDのデータ転送サイズは、(s1)~(s5)で指定されたデータの種類により異なります。

■デバイスが指定された場合

FX, FYのファンクションデバイスの転送単位は、ビット単位となります。FDのファンクションデバイスの転送単位は、4ワード単位となります。引数で指定されたデバイスの種類により、取り扱うデータサイズが異なります。ファンクションデバイスで指定されたデバイスは、データサイズ分確保できるようにしてください。データサイズ分確保できない場合はエラーとなります。

各ファンクションデバイスのデータサイズは下記になります。

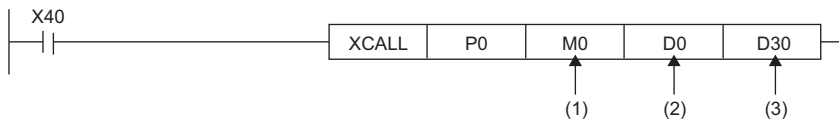
ファンクションデバイス	使用デバイス	(s1)~(s5)からファンクションデバイスに転送した場合のデータサイズ	ファンクションデバイスから(s1)~(s5)に転送した場合のデータサイズ
FX, FY	ビットデバイス	1ビット	1ビット
	ワードデバイスのビット指定時	1ビット	1ビット
FD	ビットデバイスの桁指定時 ^{*1}	4ワード	桁指定のサイズまたは4ワード ^{*2}
	ワードデバイス	4ワード	4ワード

*1 ビットデバイスの桁指定時で、(s1)~(s5)で指定されたデバイス番号が16の倍数ではない場合でも指定できます。

*2 デバイス番号が16の倍数かつ、桁指定がK4またはK8の場合に4ワードサイズを転送します。

例

デバイスを指定した場合、下記の動作になります。



- (1) M0をFX0へ転送します。
 (2) D0~D3をFD1へ転送します。
 (3) D30~D33をFD2へ転送します。

■ラベルが指定された場合

(s1)~(s5)からファンクションデバイスに転送した場合、FX, FYのファンクションデバイスの転送単位は、ビット単位となります。FDのファンクションデバイスの転送単位は、4ワード単位となります。

ファンクションデバイスから(s1)~(s5)に転送した場合、FX, FYのファンクションデバイスの転送単位は、ビット単位となります。FDのファンクションデバイスの転送単位は、ラベルの割付けサイズ単位または桁指定のサイズ単位となります。

各ファンクションデバイスのデータサイズは下記になります。

ファンクションデバイス	ラベルの割付けサイズ	(s1)~(s5)からファンクションデバイスに転送した場合のデータサイズ	ファンクションデバイスから(s1)~(s5)に転送した場合のデータサイズ
FX, FY	1ビット	1ビット	1ビット
FD	2ビット以上	4ワード*1	ラベルの割付けサイズ*2*3/桁指定のサイズ*4

*1 指定されたラベル/構造体の割付けサイズが4ワード未満場合、4ワードに満たないエリアは0を転送します。

*2 指定されたラベル/構造体の割付けサイズが4ワード以上の場合は、4ワード転送します。

*3 下記のラベルを指定した場合、送信するサイズが異なります。当てはまらないラベルを指定した場合は指定したラベルの割付けサイズで転送します。

(1) 配列型ラベル(要素を指定している場合を含む)、構造体型ラベルを指定した場合は、指定したラベル全体のサイズを転送します。

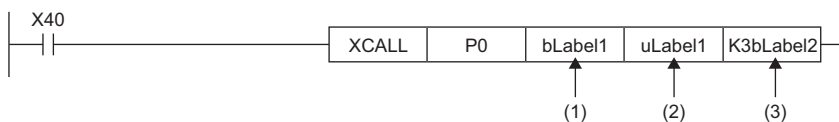
(2) 構造体型ラベルのメンバを指定した場合は、指定したメンバのサイズを転送します。

*4 桁指定がK1~K3の場合は1ワード、K5~K7の場合は2ワード、K4またはK8の場合はラベルの割付けサイズで転送します。

例

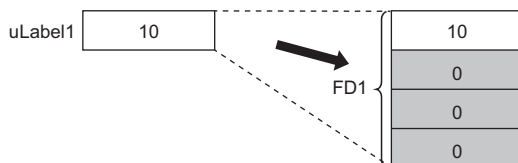
(s1)~(s5)からファンクションデバイスに転送した場合、下記のように動作します。

ラベル名	ラベルの種類
bLabel1	ビット型ラベル
uLabel1	ワード型ラベル
bLabel2	配列ビット型ラベル

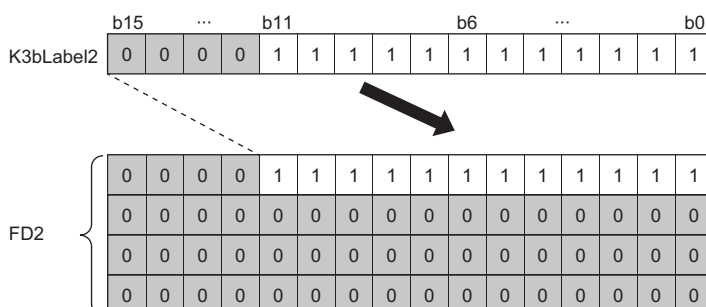


(1) bLabel1(1ビット分)をFX0へ転送します。

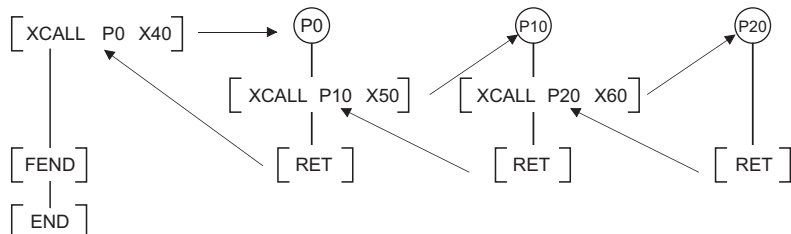
(2) uLabel1(1ワード分)をFD1へ転送します。割付けサイズが4ワード未満のため、4ワードに満たないエリアは0を転送します。



(3) K3bLabel2の1ワード分をFD2へ転送します。割付けサイズが4ワード未満のため、4ワードに満たないエリアは0を転送します。



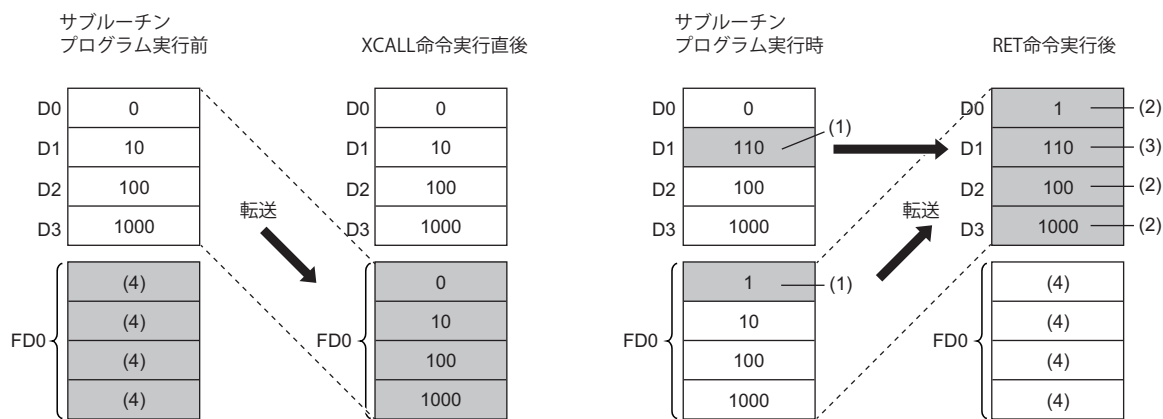
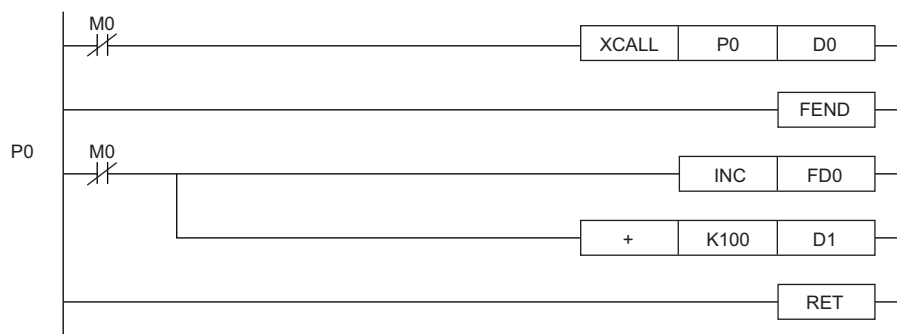
- XCALL命令では、(s1)~(s5)まで使用できます。
- サブルーチンプログラムで使用しているファンクションデバイス数と、XCALL命令の引数は、同数にする必要があります。またファンクションデバイスとXCALL命令の引数の種類は同一にしてください。
- XCALL命令で引数で指定されたデバイス番号は、重複しないください。重複している場合は、正常に演算ができません。
- XCALL命令のネスティングは16重まで可能です。ただし、ネスティングの16重は、CALL(P)命令、FCALL(P)命令、ECALL(P)命令、EFCALL(P)命令、XCALL命令の合計です。



- XCALL命令の引数に使用したデバイスを、サブルーチンプログラムで使用しないでください。XCALL命令の引数に使用したデバイスをサブルーチンプログラムで使用した場合には、正常な演算ができません。

例

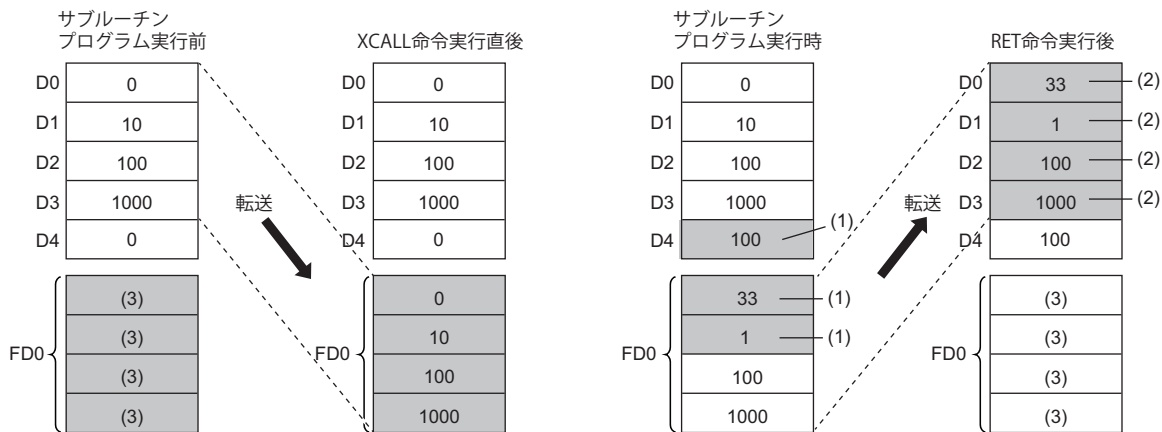
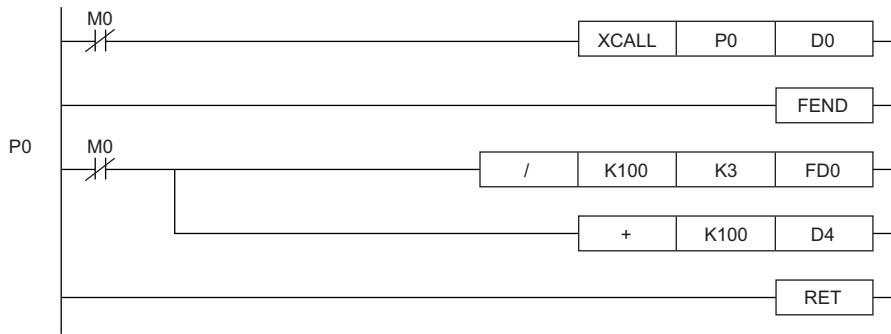
[間違った演算] サブルーチンプログラムのFD0にD0を指定し、サブルーチンプログラムでD1を使用した場合の動作



- (1): サブルーチンプログラムの実行結果が格納されます。
- (2): ファンクションデバイスの値に置き換わります。
- (3): D1はファンクションデバイスの値が反映されません。
- (4): 不定になります。

例

[正しい演算] サブルーチンプログラムのFD0にD0を指定し、サブルーチンプログラムでD4を使用した場合の動作



- (1): サブルーチンプログラムの実行結果が格納されます。
 (2): ファンクションデバイスの値に置き換わります。
 (3): 不定になります。

注意事項

- FBD/LDプログラムはサブルーチンプログラムとして作成できません。
- FBD/LDでは、引数を持つサブルーチンプログラムの実行はできません。
- (P)で指定したポインタをインデックス修飾した場合、インデックス修飾解決後にFOR~NEXT内に存在するポインタを指定した場合はエラー(エラーコード: 3340H)となります。なお、インデックス修飾せず、直接FOR~NEXT内に存在するポインタを指定した場合は、変換エラーとなります。

エラー

エラーコード (SD0)	内容
2820H	(s1)~(s5)で引数に指定したデバイスがデータサイズ分確保できないとき。
3330H	17重目のネスティングが実行されたとき。
3340H	(P)で指定したポインタのサブルーチンプログラムが存在しないとき。 または(P)で指定したポインタをインデックス修飾し、インデックス修飾解決後にFOR~NEXT内に存在するポインタが指定されたとき。
3341H	XCALL命令実行後、RET命令を実行する前にEND命令、FEND命令、GOEND命令、STOP命令を実行したとき。
3342H	XCALL命令を実行する前にRET命令を実行したとき。

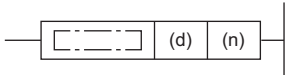
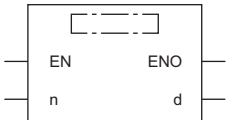
14 データ処理

14.1 ローテーション命令

BIN16ビットデータの右ローテーション

ROR(P), RCR(P)

- ROR(P): BIN16ビットデータをキャリフラグを含めずに(n)ビット右へ回転させます。
- RCR(P): BIN16ビットデータをキャリフラグを含めて(n)ビット右へ回転させます。

ラダー	ST*1
	ENO:=RORP(EN,n,d); ENO:=RCR(EN,n,d); ENO:=RCRP(EN,n,d);
FBD/LD*1	
	

*1 ROR命令はST, FBD/LDでは対応していません。汎用ファンクションのRORを使用してください。

📖 1287ページ ROR(_E)

■実行条件

命令	実行条件
ROR RCR	
RORP RCRP	

設定データ

■内容, 範囲, データ型

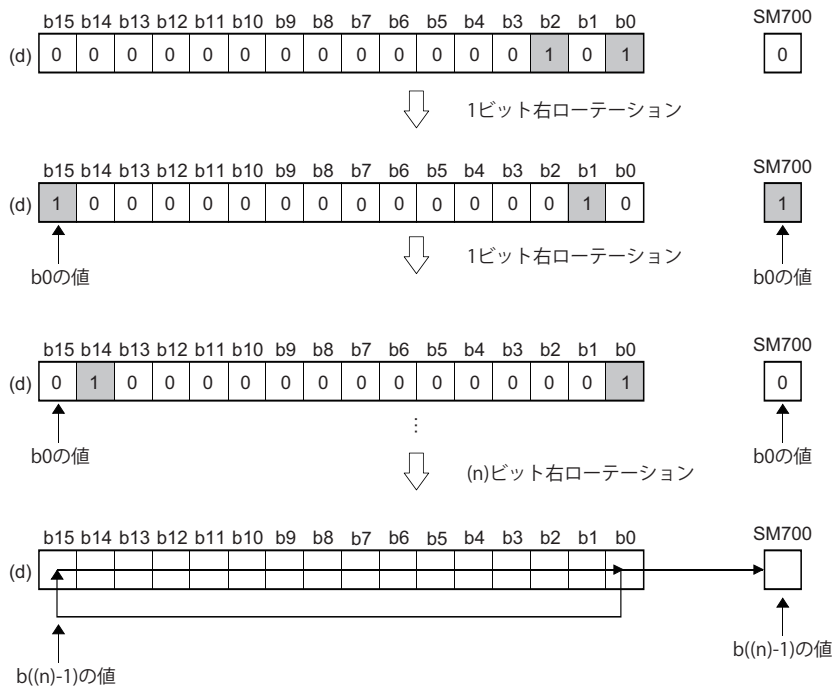
オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	ローテーションするデバイス	—	符号付きBIN16ビット	ANY16
(n)	ローテーションさせるビット数	0~15	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

■ROR(P)

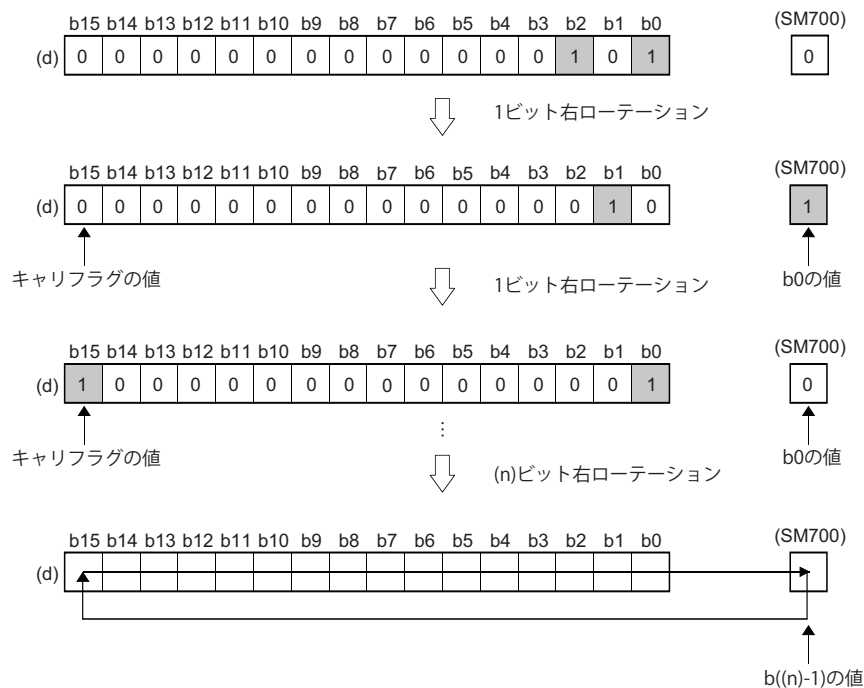
- (d)で指定されたデバイスのBIN16ビットデータを、SM700(キャリフラグ)を含めないで(n)ビット右へ回転させます。SM700(キャリフラグ)は、ROR(P)命令の実行前の状態によりONまたはOFFになっています。



- (d)にビットデバイスが指定された場合は、桁指定で指定されたデバイス範囲で回転します。このとき実際に回転するビット数は、 $(n) \div (\text{桁指定で指定された点数})$ の余りになります。たとえば、 $(n)=15$ 、(桁指定で指定された点数)=12ビットのときは、 $15 \div 12 = 1$ 余り3となるため、3ビット右へ回転します。
- (n)は0~15を指定します。(n)に16以上の値が指定された場合は、 $(n) \div 16$ の余りの値で回転します。たとえば $(n)=18$ のときは、 $18 \div 16 = 1$ 余り2となるため、2ビット右へ回転します。

■RCR(P)

- (d)で指定されたデバイスのBIN16ビットデータを、SM700(キャリフラグ)を含めて(n)ビット右へ回転させます。SM700(キャリフラグ)は、RCR(P)命令の実行前の状態によりONまたはOFFになっています。



- (d)にビットデバイスが指定された場合は、桁指定で指定されたデバイス範囲で回転します。このとき実際に回転するビット数は、 $(n) \div (\text{桁指定で指定された点数})$ の余りになります。たとえば、 $(n)=15$ 、(桁指定で指定された点数)=12ビットのときは、 $15 \div 12 = 1$ 余り3となるため、3ビット右へ回転します。
- (n)は0~15を指定します。(n)に16以上の値が指定された場合は、 $(n) \div 16$ の余りの値で回転します。たとえば $(n)=18$ のときは、 $18 \div 16 = 1$ 余り2となるため、2ビット右へ回転します。

エラー

演算エラーはありません。

BIN16ビットデータの左ローテーション

ROL(P), RCL(P)

- ROL(P): BIN16ビットデータをキャリフラグを含めずに(n)ビット左へ回転させます。
- RCL(P): BIN16ビットデータをキャリフラグを含めて(n)ビット左へ回転させます。

ラダー	ST*1
	ENO:=ROLP(EN,n,d); ENO:=RCL(EN,n,d); ENO:=RCLP(EN,n,d);

FBD/LD*1

*1 ROL命令はST, FBD/LDでは対応していません。汎用ファンクションのROLを使用してください。
 1285ページ ROL(_E)

■実行条件

命令	実行条件
ROL RCL	
ROLP RCLP	

設定データ

■内容, 範囲, データ型

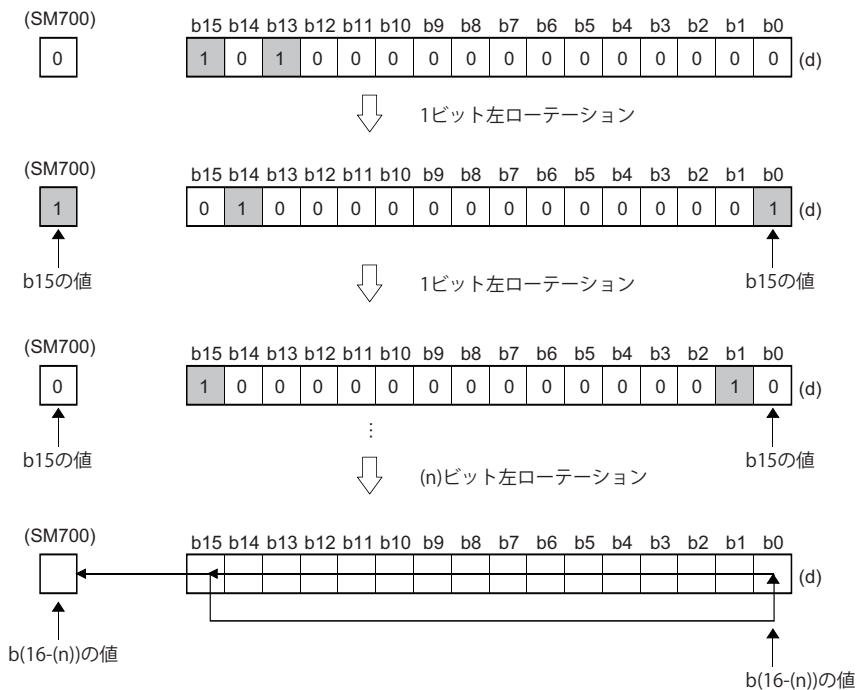
オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	ローテーションするデバイス	—	符号付きBIN16ビット	ANY16
(n)	ローテーションさせるビット数	0~15	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(d)	○	○	○	○	○	—	—	○	—	—	—
(n)	○	○	○	○	○	—	—	○	—	—	—

■ROL(P)

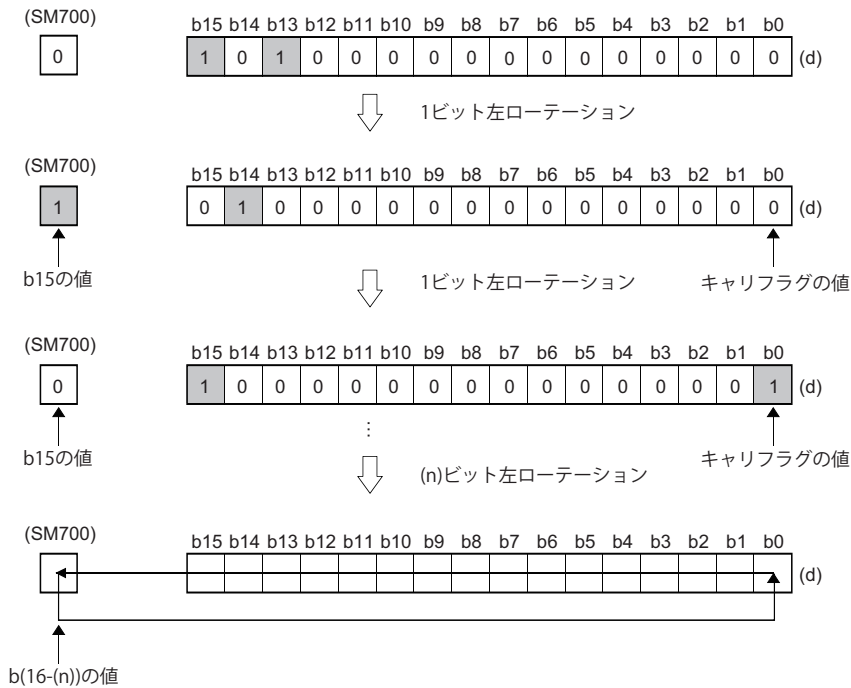
- (d)で指定されたデバイスのBIN16ビットデータを、SM700(キャリフラグ)を含めないで(n)ビット左へ回転させます。SM700(キャリフラグ)は、ROL(P)命令の実行前の状態によりONまたはOFFになっています。



- (d)にビットデバイスが指定された場合は、桁指定で指定されたデバイス範囲で回転します。このとき実際に回転するビット数は、 $(n) \div (\text{桁指定で指定された点数})$ の余りになります。たとえば、 $(n)=15$ 、(桁指定で指定された点数)=12ビットのときは、 $15 \div 12 = 1$ 余り3となるため、3ビット左へ回転します。
- (n)は0~15を指定します。(n)に16以上の値が指定された場合は、 $(n) \div 16$ の余りの値で回転します。たとえば $(n)=18$ のときは、 $18 \div 16 = 1$ 余り2となるため、2ビット左へ回転します。

■RCL(P)

- (d)で指定されたデバイスのBIN16ビットデータを，SM700(キャリフラグ)を含めて(n)ビット左へ回転させます。SM700(キャリフラグ)は，RCL(P)命令の実行前の状態によりONまたはOFFになっています。



- (d)にビットデバイスが指定された場合は，桁指定で指定されたデバイス範囲で回転します。このとき実際に回転するビット数は， $(n) \div (\text{桁指定で指定された点数})$ の余りになります。たとえば， $(n)=15$ ，(桁指定で指定された点数)=12ビットのときは， $15 \div 12 = 1$ 余り3となるため，3ビット左へ回転します。
- (n)は0~15を指定します。(n)に16以上の値が指定された場合は， $(n) \div 16$ の余りの値で回転します。たとえば $(n)=18$ のときは， $18 \div 16 = 1$ 余り2となるため，2ビット左へ回転します。

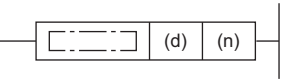
エラー

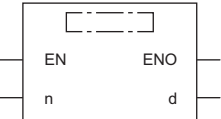
演算エラーはありません。

BIN32ビットデータの右ローテーション

DROR(P), DRCR(P)

- DROR(P): BIN32ビットデータをキャリフラグを含めずに(n)ビット右へ回転させます。
- DRCR(P): BIN32ビットデータをキャリフラグを含めて(n)ビット右へ回転させます。

ラダー	ST*1
	ENO:=DROR(EN,n,d); ENO:=DRCR(EN,n,d); ENO:=DRCRP(EN,n,d);

FBD/LD*1


*1 DROR命令はST, FBD/LDでは対応していません。汎用ファンクションのRORを使用してください。
1287ページ ROR(_E)

■実行条件

命令	実行条件
DROR DRCR	
DRORP DRCRP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	ローテーションする先頭デバイス	—	符号付きBIN32ビット	ANY32
(n)	ローテーションさせるビット数	0~31	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

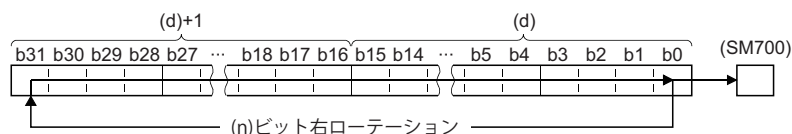
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(d)	○	○	○	○	○	○	○	○	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	○	—	—	—

機能

■DROR(P)

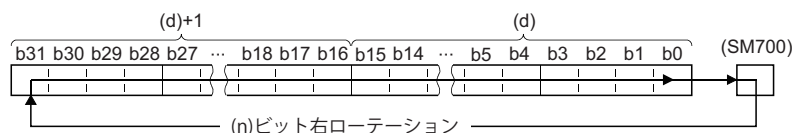
- (d)で指定されたデバイスのBIN32ビットデータを、SM700(キャリフラグ)を含めないで(n)ビット右へ回転させます。SM700(キャリフラグ)は、DROR(P)命令の実行前の状態によりONまたはOFFになっています。



- (d)にビットデバイスが指定された場合は、桁指定で指定されたデバイス範囲で回転します。このとき実際に回転するビット数は、 $(n) \div (\text{桁指定で指定された点数})$ の余りになります。たとえば、 $(n)=31$ 、(桁指定で指定された点数)=24ビットのときは、 $31 \div 24 = 1$ 余り7となるため、7ビット右へ回転します。
- (n)は0~31を指定します。(n)に32以上の値が指定された場合は、 $(n) \div 32$ の余りの値で回転します。たとえば $(n)=34$ のときは、 $34 \div 32 = 1$ 余り2となるため、2ビット右へ回転します。

■DRCR(P)

- (d)で指定されたデバイスのBIN32ビットデータを、SM700(キャリフラグ)を含めて(n)ビット右へ回転させます。SM700(キャリフラグ)は、DRCR(P)命令の実行前の状態によりONまたはOFFになっています。



- (d)にビットデバイスが指定された場合は、桁指定で指定されたデバイス範囲で回転します。このとき実際に回転するビット数は、 $(n) \div (\text{桁指定で指定された点数})$ の余りになります。たとえば、 $(n)=31$ 、(桁指定で指定された点数)=24ビットのときは、 $31 \div 24 = 1$ 余り7となるため、7ビット右へ回転します。
- (n)は0~31を指定します。(n)に32以上の値が指定された場合は、 $(n) \div 32$ の余りの値で回転します。たとえば $(n)=34$ のときは、 $34 \div 32 = 1$ 余り2となるため、2ビット右へ回転します。

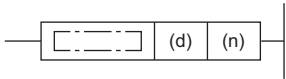
エラー

演算エラーはありません。

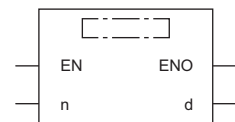
BIN32ビットデータの左ローテーション

DROL(P), DRCL(P)

- DROL(P): BIN32ビットデータをキャリフラグを含めずに(n)ビット左へ回転させます。
- DRCL(P): BIN32ビットデータをキャリフラグを含めて(n)ビット左へ回転させます。

ラダー	ST*1
	ENO:=DROL(EN,n,d); ENO:=DRCL(EN,n,d); ENO:=DRCLP(EN,n,d);

FBD/LD*1



*1 DROL命令はST, FBD/LDでは対応していません。汎用ファンクションのROLを使用してください。

📖 1285ページ ROL(_E)

■実行条件

命令	実行条件
DROL DRCL	
DROLP DRCLP	

設定データ

■内容, 範囲, データ型

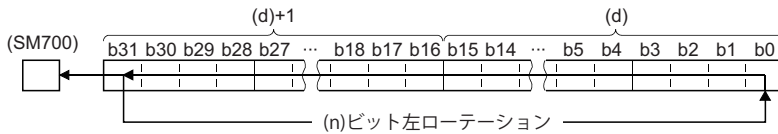
オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	ローテーションする先頭デバイス	—	符号付きBIN32ビット	ANY32
(n)	ローテーションさせるビット数	0~31	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LZ		K, H	E	\$	
(d)	○	○	○	○	○	○	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

■DROL(P)

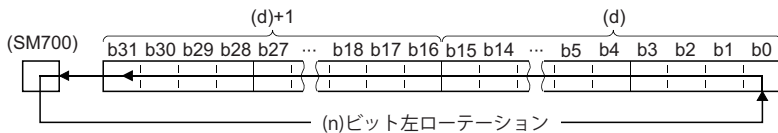
- (d)で指定されたデバイスのBIN32ビットデータを、SM700(キャリフラグ)を含めないで(n)ビット左へ回転させます。SM700(キャリフラグ)は、DROL(P)命令の実行前の状態によりONまたはOFFになっています。



- (d)にビットデバイスが指定された場合は、桁指定で指定されたデバイス範囲で回転します。このとき実際に回転するビット数は、 $(n) \div (\text{桁指定で指定された点数})$ の余りになります。たとえば、 $(n)=31$ 、(桁指定で指定された点数)=24ビットのときは、 $31 \div 24 = 1$ 余り7となるため、7ビット左へ回転します。
- (n)は0~31を指定します。(n)に32以上の値が指定された場合は、 $(n) \div 32$ の余りの値で回転します。たとえば $(n)=34$ のときは、 $34 \div 32 = 1$ 余り2となるため、2ビット左へ回転します。

■DRCL(P)

- (d)で指定されたデバイスのBIN32ビットデータを、SM700(キャリフラグ)を含めて(n)ビット左へ回転させます。SM700(キャリフラグ)は、DRCL(P)命令の実行前の状態によりONまたはOFFになっています。



- (d)にビットデバイスが指定された場合は、桁指定で指定されたデバイス範囲で回転します。このとき実際に回転するビット数は、 $(n) \div (\text{桁指定で指定された点数})$ の余りになります。たとえば、 $(n)=31$ 、(桁指定で指定された点数)=24ビットのときは、 $31 \div 24 = 1$ 余り7となるため、7ビット左へ回転します。
- (n)は0~31を指定します。(n)に32以上の値が指定された場合は、 $(n) \div 32$ の余りの値で回転します。たとえば $(n)=34$ のときは、 $34 \div 32 = 1$ 余り2となるため、2ビット左へ回転します。

エラー

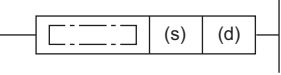
演算エラーはありません。

14.2 データテーブル操作命令

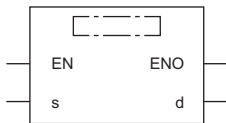
データテーブルからの先入データリード

FIFR(P)

テーブルに最初に格納したデータを指定されたデバイスに格納します。

ラダー	ST
	ENO:=FIFR(EN,s,d); ENO:=FIFRP(EN,s,d);

FBD/LD



■実行条件

命令	実行条件
FIFR	
FIFRP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	テーブルからの読出しデータを格納するデバイス	—	符号付きBIN16ビット	ANY16
(d)	テーブルの先頭デバイス	—	ワード	ANY16*1
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

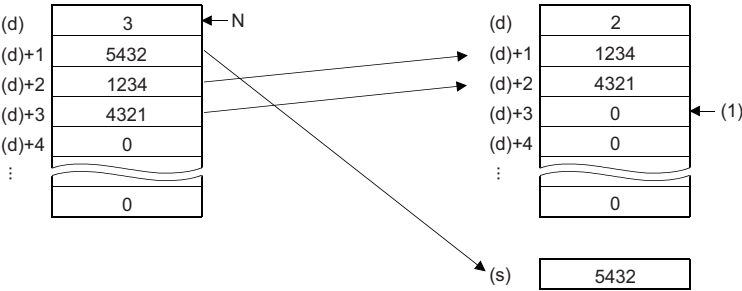
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□		LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- (d)で指定されたテーブルの最先入データ(d)+1を，(s)で指定されたデバイスに格納します。FIFR(P)命令実行後はデータテーブルのデータが1個ずつ前詰めされます。



- N: データ数
(1): 0が格納されます。
- (d)に格納されている値が0のとき，FIFR(P)命令を実行しないようにインタロックをとってください。
 - データ数(d)には，0~65535が格納されます。

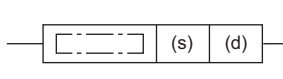
エラー

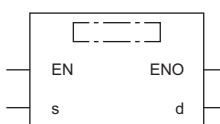
エラーコード (SD0)	内容
3285H	(d)の値が0でFIFR(P)命令を実行したとき。

データテーブルからの後入データリード

FPOP(P)

テーブルの最後に格納したデータを指定されたデバイスに格納します。

ラダー	ST
	ENO:=FPOP(EN,s,d); ENO:=FPOPP(EN,s,d);

FBD/LD


■実行条件

命令	実行条件
FPOP	
FPOPP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	テーブルからの読出しデータを格納するデバイス	—	符号付きBIN16ビット	ANY16
(d)	テーブルの先頭デバイス	—	ワード	ANY16*1
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

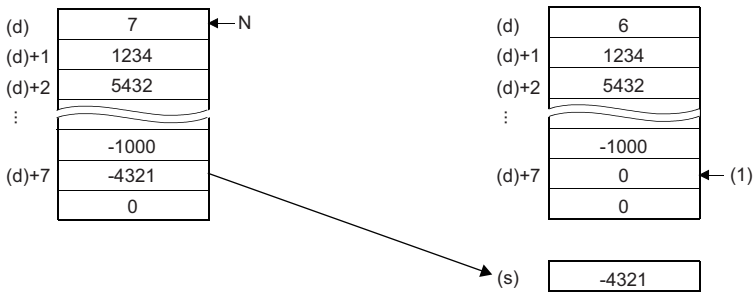
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC		K, H	E	\$	
(s)	○	○	○	○	○	—	○	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—

機能

- (d)で指定されたテーブルの最後に格納したデータを(s)で指定されたデバイスに格納します。FPOP(P)命令実行後は、FPOP(P)命令で読み出したデータを格納していたデバイスが0となります。



- N: データ数
(1): 0が格納されます。
- (d)に格納されている値が0のとき、FPOP(P)命令を実行しないようにインタロックをとってください。
 - データ数(d)には、0~65535が格納されます。

エラー

エラーコード (SD0)	内容
3285H	(d)の値が0でFPOP(P)命令を実行したとき。

データテーブルへのデータライト

FIFW(P)

BIN16ビットデータを、指定されたデータテーブルに格納します。

ラダー	ST
	ENO:=FIFW(EN,s,d); ENO:=FIFWP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
FIFW	
FIFWP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	テーブルへの書き込みデータまたは、データが格納されているデバイス番号	—	符号付きBIN16ビット	ANY16
(d)	テーブルの先頭デバイス	—	ワード	ANY16 ^{*1}
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

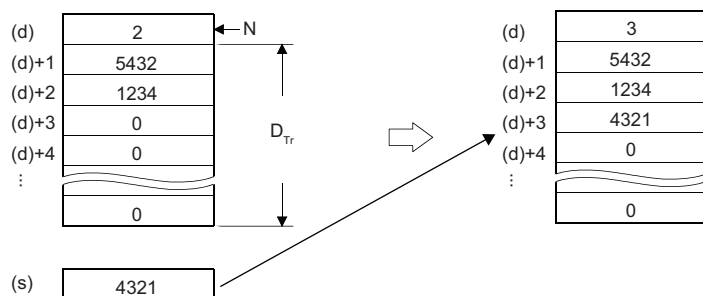
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- (s)で指定されたBIN16ビットデータを，(d)で指定されたデータテーブルに格納します。(d)にはテーブルに格納されているデータ数を格納し，(d)+1以降に(s)で指定されたデータを順次格納していきます。



N: データ数

D_{Tr} : データテーブル範囲(ユーザで管理)

- FIFW(P)命令をはじめて実行する場合は，(d)で指定するデバイスの値をクリアしてください。
- (d)+1以降にデータが順次格納されますので，データテーブルの範囲については注意してください。
- データ数(d)には，0~65535が格納されます。

エラー

エラーコード (SD0)	内容
3285H	(d)の値がFFFFHでFIFW(P)命令を実行したとき。

データテーブルのデータ挿入

FINS(P)

BIN16ビットデータを指定されたデータテーブルの(n)番目に挿入します。

ラダー	ST
	ENO:=FINS(EN,s,n,d); ENO:=FINS(EN,s,n,d);

FBD/LD

■実行条件

命令	実行条件
FINS	
FINS(EN)	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	挿入データが格納されているデバイス	—	符号付きBIN16ビット	ANY16
(d)	テーブルの先頭デバイス	—	ワード	ANY16*1
(n)	挿入するテーブル位置	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

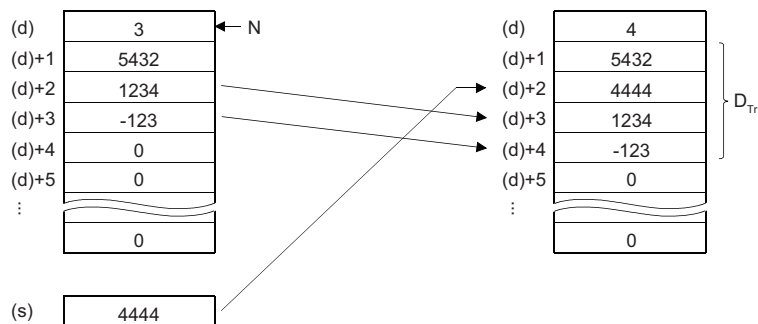
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

機能

- (s)で指定されたBIN16ビットデータを(d)で指定されたデータテーブルの(n)番目に挿入します。FINS(P)命令実行後は、データテーブルの(n)番目からのデータが1個ずつ繰り下がります。



N: データ数

D_{Tr} : データテーブル範囲(ユーザで管理)

(n)=2の場合は, (d)+2に挿入されます。

- データ数(d)には, 0~65535が格納されます。

エラー

エラーコード (SD0)	内容
3285H	(n)の値が0でFINS(P)命令を実行したとき。
	(d)の値がFFFFHでFINS(P)命令を実行したとき。
	(n)の値に(d)のデータ数+1を超えた値を指定してFINS(P)命令を実行したとき。

データテーブルのデータ削除

FDEL(P)

データテーブルの(n)番目のデータを削除します。

ラダー	ST
	ENO:=FDEL(EN,s,n,d); ENO:=FDELP(EN,s,n,d);

FBD/LD

■実行条件

命令	実行条件
FDEL	
FDELP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	削除データを格納するデバイス	—	符号付きBIN16ビット	ANY16
(d)	テーブルの先頭デバイス	—	ワード	ANY16*1
(n)	削除するテーブル位置	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

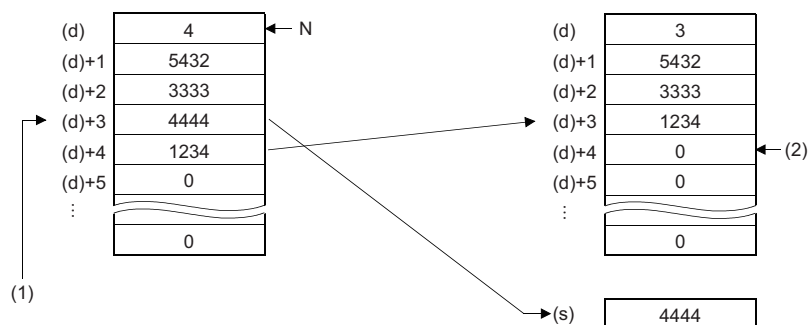
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- (d)で指定されたデータテーブルの(n)番目のデータを削除し、(s)で指定されたデバイスに格納します。FDEL(P)命令実行後は、データテーブルの(n)+1番目以降のデータが1個ずつ前詰めされます。



N: データ数

(1): (n)=3の場合は、(d)+3が対象になります。

(2): 0が格納されます。

- データ数(d)には、0~65535が格納されます。

エラー

エラーコード (SD0)	内容
3285H	(n)の値が0でFDEL(P)命令を実行したとき。
	(d)の値が0でFDEL(P)命令を実行したとき。
	(n)の値に(d)のデータ数+1を超えた値を指定してFDEL(P)命令を実行したとき。

14.3 データリード/ライト命令

データメモリのデータリード/ライト

データメモリへのデータライト命令は、任意のデバイスデータをデータメモリへ書き込む命令です。
演算で使用する固定値や演算結果をデータメモリへ書き込むことにより、バッテリー低下時のデータ消失を防ぐことができます。
また、データメモリへ書き込まれたデータは、データメモリからのデータリード命令を使用して任意のタイミングで読み出すことが可能です。

■実行方法

デバイスデータのデータメモリへの書込みは、SP.DEVST命令で行います。
また、データメモリに書き込まれたデバイスデータは、S(P).DEVLD命令で指定デバイスに読み出します。

■設定方法

SP.DEVST命令、S(P).DEVLD命令を使用する場合は、デバイスデータ格納用ファイル設定が必要です。

🔗 [CPUパラメータ]⇒[ファイル設定]⇒[デバイスデータ格納用ファイル設定]

☐ デバイスデータ格納用ファイル設定	
☐ 使用有無設定	使用する
容量	1 Kワード
ファイル名	DEVSTORE

項目	内容
容量	1K~512K(ワード)
ファイル名	DEVSTORE(固定)

データメモリにデバイスデータ格納用ファイル作成に必要な空き容量が無い場合、エラー (エラーコード: 21A1H)が発生します。
また、STOP→RUN時に実際のデバイスデータ格納用ファイルがパラメータ設定と一致しているかチェックを行い、不一致の場合はエラー (エラーコード: 21A0H)が発生します。

データメモリからのデータリード

S(P).DEVLD

データメモリ上のデバイスデータ格納用ファイルからデータを読み出します。

ラダー	ST
	ENO:=S_DEVLD(EN,s,n,d); ENO:=SP_DEVLD(EN,s,n,d);

FBD/LD

■実行条件

命令	実行条件
S.DEVLD	
SP.DEVLD	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	デバイスデータ格納用ファイルからの読み出しオフセット(1点16ビットの単位で指定)	0~524287	符号なしBIN32ビット	ANY32
(d)	読み出したデータを格納するデバイス	—	ワード	ANY16 ^{*1}
(n)	読み出し点数	1~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

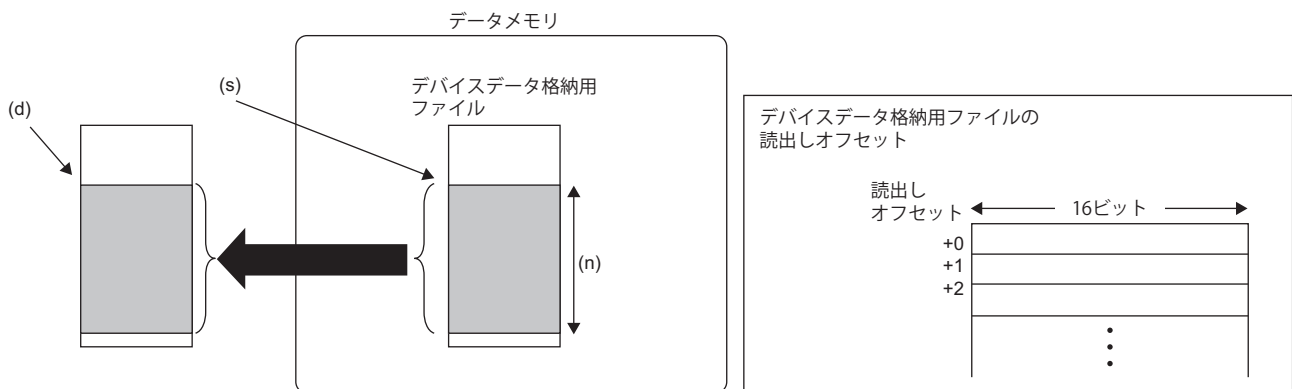
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口¥口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口¥G口, J口¥口	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—
(n)	○	—	○	—	—	—	—	○	—	—	—

機能

- データメモリ上のデバイスデータ格納用ファイルの(s)で指定した読出しオフセットから、(n)で指定された点数分のデバイスデータを読み出して、(d)で指定したデバイスに格納します。(s)は、デバイスデータ格納用ファイル先頭からのオフセットで、ワードオフセット(16ビットごとに+1する単位)にて指定します。



- S(P).DEVLD命令を使用する場合は、デバイスデータ格納用ファイル設定が必要です。(📖 602ページ 設定方法)
- デバイスデータ格納用ファイルへの書込みは、SP.DEVST命令で行います。(📖 605ページ SP.DEVST)

エラー

エラーコード (SD0)	内容
2840H	パラメータにデバイスデータ格納用ファイルの設定をしていないとき。
3285H	(n)の設定データが0のとき。

データメモリへのデータライト

SP.DEVST

指定した点数のデバイスデータを、データメモリ上のデバイスデータ格納用ファイルに書き込みます。

ラダー	ST
	ENO:=SP_DEVST(EN,s1,s2,n,d);

FBD/LD

■実行条件

命令	実行条件
SP.DEVST	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	デバイスデータ格納用ファイルの書き込みオフセット(1点16ビットの単位で指定)	0~524287	符号なしBIN32ビット	ANY32
(s2)	書き込む先頭デバイス	—	ワード	ANY16 ^{*1}
(n)	書き込み点数	1~65535	符号なしBIN16ビット	ANY16
(d)	(d): 完了デバイス, (d)+1: 異常完了デバイス	—	ビット	ANYBIT_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

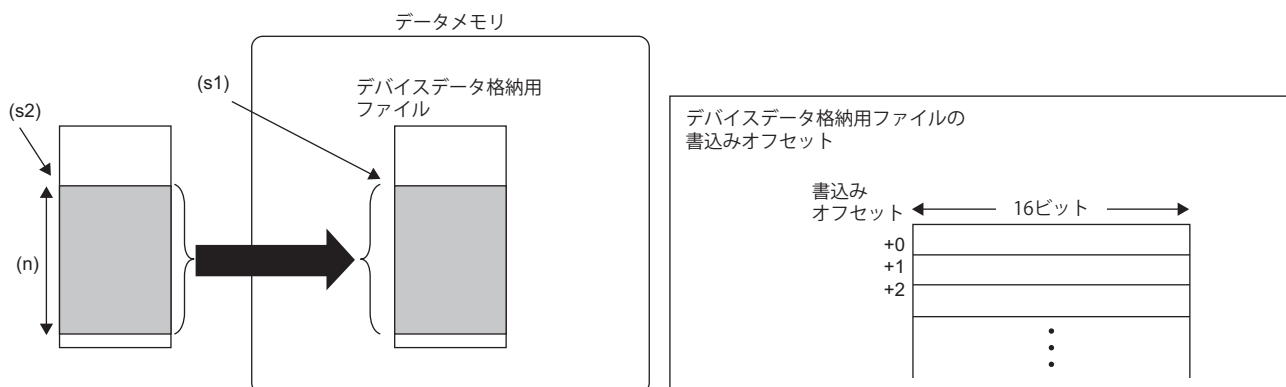
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□		LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	○	—	○	○	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	—	○	—	—	—	—	○	○	—	—	—
(d)	○	—	○	—	—	—	—	○	—	—	—	—

機能

- (s2)で指定したデバイスから(n)で指定した点数のデバイスデータを、データメモリ上のデバイスデータ格納用ファイルの(s1)で指定した書き込みオフセットに書き込みます。(s1)はデバイスデータ格納用ファイル先頭からのオフセットで、ワードオフセット(16ビットごとに+1する単位)にて指定します。



- 完了デバイス(d)は、SP.DEVST命令の処理完了を検出したEND命令実行時に自動的にONし、次のスキャンのEND命令にてOFFするので、SP.DEVST命令の実行完了フラグとして使用します。
- SP.DEVST命令が異常完了時には、異常完了デバイス(d)+1が完了デバイス(d)と同じタイミングでON/OFFするので、SP.DEVST命令の異常完了フラグとして使用します。
- SP.DEVST命令の実行中はSM753(ファイルアクセス中)がONします。すでにSM753がONしている場合は、SP.DEVST命令を実行することができません。(実行した場合は無処理となります。)
- SP.DEVST命令実行時にエラーを検出した場合、完了デバイス(d)、異常完了デバイス(d)+1、SM753はONしません。
- SP.DEVST命令を使用する場合は、デバイスデータ格納用ファイル設定が必要です。(602ページ 設定方法)
- デバイスデータ格納用ファイルに書き込まれたデバイスデータは、S(P).DEVLD命令で指定デバイスに読み出します。(603ページ S(P).DEVLD)

注意事項

- データメモリへ書き込む値は、SP.DEVST命令実行時の値となります。
- SP.DEVST命令の実行により、SD4124、SD4125(内蔵メモリ書換え回数指標)が増加します。コントローラのデータメモリへの書き込み回数には上限があり、データメモリ書き込み回数指標が5万回を超えると、エラー(エラーコード: 1080H)となります。
- 不要な命令実行によるデータメモリ書き込み回数の増加を防止するため、SD771(データメモリへの書き込み命令実行回数指定)を設定することで、1日における書き込み回数を制限することができます。書き込み回数の制限は、デフォルトで36回です。必要に応じてSD771で、制限回数を変更してください。設定された書き込み回数を超えると、エラー(エラーコード: 3291H)となります。なお、1日あたりのデータメモリへの書き込み命令実行回数は、下記のタイミングで0に初期化します。
 - 電源OFF→ON時、リセット→リセット解除時
 - 時間の進行による、時計データの日付(年、月、日)変更時
 - 時計データを変更する機能による、コントローラ内部の時計データの年、月、日の変更時
- デバイスデータ格納用ファイルに書き込むタイミングは、END命令実行時になります。SP.DEVST命令が実行された直後のEND命令実行時にデバイスデータ格納用ファイルに書き込みます。したがって、書き込み点数によって、デバイスデータ格納用ファイルへの書き込みが複数のスキャンに跨る場合もありますので、完了デバイスにて書き込みが完了しているか判断してください。

エラー

エラーコード (SD0)	内容
2840H	パラメータにデバイスデータ格納用ファイルの設定をしていないとき。
3285H	(n)の設定データが0のとき。
3291H	SP.DEVST命令実行時、当日の書き込み回数がSD771で指定している値を超えたとき。 SP.DEVST命令実行時、SD771に1~32767の範囲外の値が設定されているとき。

14.4 ファイル操作命令

指定ファイルからのデータリード

SP.FREAD

SDメモ리카ードの指定ファイルからデバイスデータを読み出します。

ラダー	ST
	ENO:=SP_FREAD(EN,U,s1,s2,d1,d2,d3);

FBD/LD

■実行条件

命令	実行条件
SP.FREAD	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(U)	ダミー	—	デバイス名	ANY16
(s1)	ドライブ指定	2(固定)*1	ワード	ANY16
(d1)	コントロールデータが格納されている先頭デバイス	608ページ コントロールデータ(d1)	ワード	ANY16_ARRAY (要素数: 8)
(s2)	ファイル名が格納されている先頭デバイス	—	Unicode文字列	ANYSTRING_DOUBLE
(d2)	読み出したデータを格納する先頭デバイス	—	ワード	ANY16*2
(d3)	処理完了にてONするビットデバイス (ただし、異常完了時には(d3)+1もON)	—	ビット	ANYBIT_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 SDメモ리카ードのドライブ2のみ設定可能です。

*2 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数				その他 (U)
	X, Y, M, L, SM, F, B, SB, FX, FY	J○¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U○¥□, J○¥□	Z	LT, LST, LC	LZ	K, H	E	\$		
(U)	—	—	—	—	—	—	—	—	—	—	—	○
(s1)	○	—	○	—	—	—	○	○	—	—	—	—
(d1)	—	—	○	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	○	—	—
(d2)	—	—	○	—	—	—	○	—	—	—	—	—
(d3)	○	—	○	—	—	—	—	—	—	—	—	—

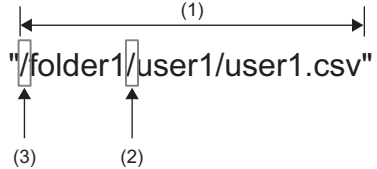
■コントロールデータ(d1)

オペランド: (d1)				
デバイス	項目	内容	設定範囲	セット側
+0	実行/完了タイプ	実行タイプを指定します。 ■00**H: バイナリ読出し ・0000H: BIN16ビットデータ ・0001H: BIN32ビットデータ ■01**H: CSV形式変換読出し ・0100H: 10進数(16ビットデータ) ・0110H: 10進数(32ビットデータ) ・0120H: 16進数(16ビットデータ) ・0121H: 16進数(32ビットデータ) ・0130H: 文字列(アスキーデータ) ・0140H: 浮動小数点実数(単精度実数) ・0141H: 浮動小数点実数(倍精度実数)	0000H 0001H 0100H 0110H 0120H 0121H 0130H 0140H 0141H	ユーザ
+1	完了ステータス	命令完了時に完了ステータスを格納します。 ・0000H: 正常完了 ・0000H以外: 異常完了(エラーコード)(図 657 ページ ファイル操作命令で発生するエラーコード)	—	システム
+2	要求読出しデータ数	読み出したいデータ数を指定します。処理単位および設定範囲は、(d1)+0の実行/完了タイプにより異なります。 ■(d1)+0で「バイナリ読出し」を指定したとき ・BIN16ビットデータ指定時: ワード単位(1~65535) ^{*1*2} ・BIN32ビットデータ指定時: ダブルワード単位(1~32767) ■(d1)+0で「CSV形式変換読出し」を指定したとき ・10進数(16ビットデータ)指定時: 要素数(1~65535) ^{*1*2} ・10進数(32ビットデータ)指定時: 要素数(1~32767) ・16進数(16ビットデータ)指定時: 要素数(1~65535) ・16進数(32ビットデータ)指定時: 要素数(1~32767) ・文字列(アスキーデータ)指定時: 要素数(1~1023) ・浮動小数点実数(単精度実数)指定時: 要素数(1~32767) ・浮動小数点実数(倍精度実数)指定時: 要素数(1~16383)	1~65535	ユーザ
+3	最大読出しデータ数	■(d1)+0で「0130H: 文字列(アスキーデータ)」を指定したとき ・要素内の文字数の合計サイズ(ワード単位) ■(d1)+0で「0130H: 文字列(アスキーデータ)」以外を指定したとき ・0固定	0, 1~65535	ユーザ
+4 +5	ファイル位置	■(d1)+0で「バイナリ読出し」を指定したとき ・00000000H: ファイルの先頭から ・00000001H~FFFFFFFH: 指定位置から(単位は(d1)+7のデータタイプ指定によります。) ・FFFFFFFH: 指定不可 ■(d1)+0で「CSV形式変換読出し」を指定したとき ・00000000H: ファイルの先頭から ・00000001H~FFFFFFFH: 指定行から ・FFFFFFFH: 前回の読出し位置から継続	00000000H~ FFFFFFFH	ユーザ
+6	列数指定	(d1)+0で「バイナリ読出し」を指定したときは、必ず0を指定してください。 (d1)+0で「CSV形式変換読出し」を指定したときは、読出しを行う列数を指定してください。 ・0: 列はなし。1行になります。 ・0以外: 指定数の列になります。	0000H~FFFFH (0~65535)	ユーザ
+7	データタイプ指定	・0: ワード ・1: 偶数バイト ^{*1} ・2: (d1)+0で指定したデータ型の単位 ・3: 奇数バイト ^{*1*2} 「0: ワード」および「1: 偶数バイト」、「3: 奇数バイト」は、(d1)+0で「0000H: BIN16ビットデータ」または「0100H: 10進数(16ビットデータ)」を指定した場合のみ指定できます。	0, 1, 2, 3	ユーザ

*1 データタイプ指定(d1)+7に、「1: 偶数バイト」または「3: 奇数バイト」を指定している場合、要求読出しデータ数(d1)+2の設定範囲は1~32767となります。

*2 奇数バイト(d1)+7指定時は読み出したい奇数バイトデータ数に1バイト加算しワード単位にします。ワード単位にしたデータ数を要求読出しデータ数(d1)+2に指定してください。

■ファイル名(s2)

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+0~+0	ファイル名文字列	ファイルが格納されているフォルダパス、ファイル名の文字列を指定します。 - フォルダパス+ファイル名(拡張子を含む)は253文字以内にしてください。 - フォルダパスは244文字以内にしてください。(ファイルの区切り文字は含まない。) - フォルダパスの階層数は10以内にしてください。 - ファイル名の拡張子を省略する場合は「.」(ピリオド)から省略してください。 - ファイル名(拡張子を含まない)は1文字以上60文字以内、拡張子はピリオド+3文字以内にしてください。ファイル名(拡張子を含まない)が61文字以上のときは、拡張子があっても無視され、「.BIN」または「.CSV」になります。  (1): 最大253文字 (2): フォルダパス、ファイルの区切り文字は"/"または"¥"を使用する。 (3): 省略可能。省略した場合、(1)の最大は252文字となる。	Unicode文字列	ユーザ

■読み出したデータ(d2)

オペランド: (d2)				
デバイス	項目	内容	設定範囲	セット側
+0	読み出し結果データ数	(d1)+2で指定したデータ数に対して、実際の読み出したデータ数をセットします。値の単位は、(d1)+7のデータタイプ指定によります。	—	システム
+1~+0	読み出しデータ	読み出したデータが格納されます。	—	システム

機能

- 指定ファイルからデータを読み出します。コントロールデータの実行/完了タイプにより、ファイルの読み出し形式を指定します。
- 読み出しの対象は、SDメモ리카ードのみです。
- 処理完了(d3)のビットデバイスは、SP.FREAD命令の処理完了を検出したスキンのEND命令実行時に自動的にONし、次のスキンのEND命令にてOFFします。
- SP.FREAD命令が異常完了時には、異常完了(d3)+1のデバイスが、処理完了(d3)のデバイスと同じタイミングでON/OFFします。
- SP.FREAD命令実行中は、SM753(ファイルアクセス中)がONします。
- SM753がON中は、SP.FREAD命令は実行できません。(実行した場合は無処理となります。)
- 命令実行時にエラーを検出した場合は、処理完了(d3)、異常完了(d3)+1およびSM753はONしません。
- 奇数バイト指定時は、下記のようにデータが格納されます。(指定した最後のb8~b15には、命令実行前のデータが格納されています。)

	b15	...	b8 b7	...	b0
(d2)					
(d2)+1					
(d2)+2					
(d2)+3					
(d2)+4					
(d2)+5					

- データの要求読出しデータ数(d1)+2, ファイル位置(d1)+4, (d1)+5および, 読出し結果データ数(d2)+0の処理単位は, 実行/完了タイプ(d1)+0およびデータタイプ指定(d1)+7の組合せにより指定します。

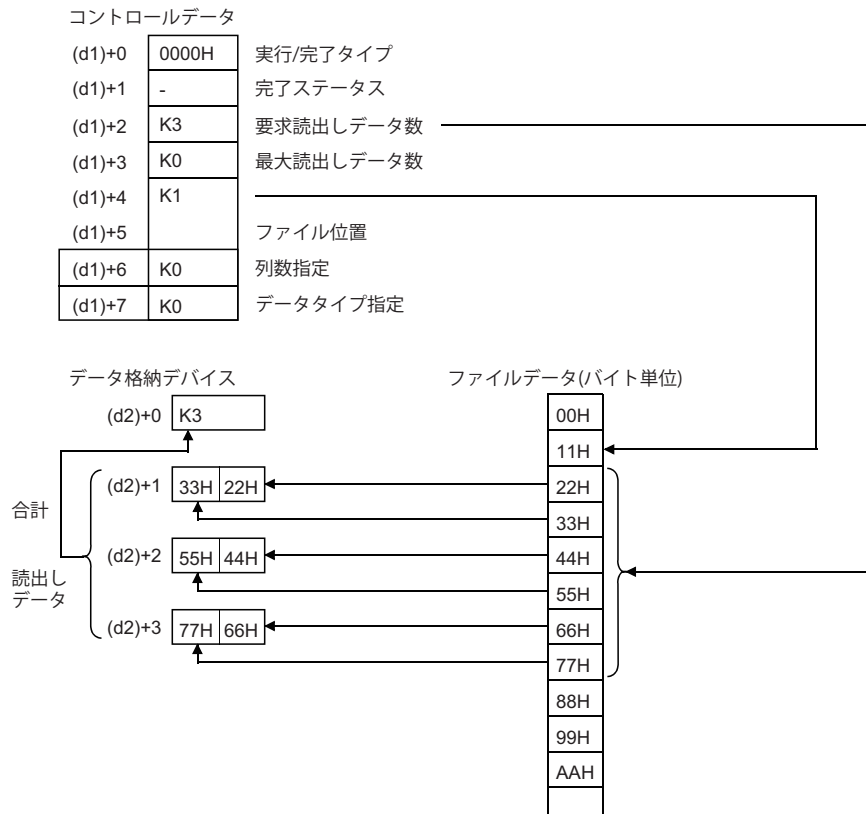
実行/完了タイプ (d1)+0	データタイプ指定 (d1)+7	処理単位および設定範囲		
		要求読出しデータ数 (d1)+2	ファイル位置 (d1)+4, (d1)+5	読出し結果データ数 (d2)+0
バイナリ読出し	0000H: BIN16ビットデータ	0: ワード	ワード(1~65535)	ワード (00000000H~7FFFFFFFH)
		1: 偶数バイト	ワード(1~32767)	バイト (00000000H~FFFFFFFEH)
		2: 実行/完了タイプで指定したデータ型の単位	ワード(1~65535)	ワード (00000000H~7FFFFFFFH)
		3: 奇数バイト	ワード(1~32767)	バイト (00000000H~FFFFFFFEH)
	0001H: BIN32ビットデータ	0: ワード 1: 偶数バイト	(指定不可)	
		2: 実行/完了タイプで指定したデータ型の単位	ダブルワード(1~32767)	ダブルワード (00000000H~3FFFFFFFH)
		3: 奇数バイト	(指定不可)	
CSV形式変換読出し	0100H: 10進数(16ビットデータ)	0: ワード	要素数(1~65535)	行数
		1: 偶数バイト	要素数(1~32767)	行数
		2: 実行/完了タイプで指定したデータ型の単位	要素数(1~65535)	行数
		3: 奇数バイト	要素数(1~32767)	行数
	0110H: 10進数(符号付き32ビットデータ)	0: ワード 1: 偶数バイト	(指定不可)	
		2: 実行/完了タイプで指定したデータ型の単位	要素数(1~32767)	行数
		3: 奇数バイト	(指定不可)	
	0120H: 16進数(16ビットデータ)	0: ワード 1: 偶数バイト	(指定不可)	
		2: 実行/完了タイプで指定したデータ型の単位	要素数(1~65535)	行数
		3: 奇数バイト	(指定不可)	
	0121H: 16進数(32ビットデータ)	0: ワード 1: 偶数バイト	(指定不可)	
		2: 実行/完了タイプで指定したデータ型の単位	要素数(1~32767)	行数
		3: 奇数バイト	(指定不可)	
	0130H: 文字列(アスキーデータ)	0: ワード 1: 偶数バイト	(指定不可)	
		2: 実行/完了タイプで指定したデータ型の単位	要素数(1~1023)	行数
		3: 奇数バイト	(指定不可)	
	0140H: 浮動小数点実数(単精度実数)	0: ワード 1: 偶数バイト	(指定不可)	
		2: 実行/完了タイプで指定したデータ型の単位	要素数(1~32767)	行数
		3: 奇数バイト	(指定不可)	
	0141H: 浮動小数点実数(倍精度実数)	0: ワード 1: 偶数バイト	(指定不可)	
		2: 実行/完了タイプで指定したデータ型の単位	要素数(1~16383)	行数
		3: 奇数バイト	(指定不可)	

■バイナリ読出し時

- 対象ファイルの拡張子を省略した場合、拡張子は「.BIN」となります。
 - 存在しないファイルが指定された場合はエラーとなります。
 - 既存のファイルの大きさよりも、大きいファイル位置が指定された場合は、0点での読出しとなり、正常完了します。
- バイナリ読出しの例を下記に示します。

例

バイナリ読出し(BIN16ビットデータ)の例



■CSV形式変換読出し時

- CSV形式のファイルの要素(Excel®のセル)を行方向に順に読み出していき、デバイスに格納します。
- 対象ファイルの拡張子を省略した場合、拡張子は「.CSV」となります。
- 存在しないファイルが指定された場合はエラーとなります。
- 既存のファイルの大きさよりも大きいファイル位置が指定された場合は、0点での読出しとなり、正常完了します。
- ファイルの先頭から要求読出しデータ数(d1)+2で指定された分のデータが読み込まれます。指定データ数分のデータを読み込む前に、ファイルの最終データに達した場合は、読出し可能分のデータを読み出します。
- 指定列数が0の場合、CSV形式ファイルの行区切りを無視して読み出します。
- CSVファイル内の文字列データと読出し後にデバイスに格納される値は、実行/完了タイプにより決定します。

実行/完了タイプ	CSV内データ(1要素)	デバイス格納値	備考
0100H: 10進数(16ビットデータ)	-32768~-1	-32768~-1 (32768~65535)	符号付き16ビットデータと見なすと-32768~-1となり、符号なし16ビットデータと見なすと32768~65535となります。デバイス格納値としては同一です。
	0~32767	0~32767	—
	32768~65535	-32768~-1 (32768~65535)	符号付き16ビットデータと見なすと-32768~-1となり、符号なし16ビットデータと見なすと32768~65535となります。デバイス格納値としては同一です。
	・上記以外の数値 ・アルファベットや記号を含む文字列	0	変換できないため0が格納されます。
0110H: 10進数(32ビットデータ)	-2147483648~-1	-2147483648~-1 (2147483648~4294967295)	符号付き32ビットデータと見なすと-2147483648~-1となり、符号なし32ビットデータと見なすと2147483648~4294967295となります。デバイス格納値としては同一です。
	0~2147483647	0~2147483647	—
	2147483648~4294967295	-2147483648~-1 (2147483648~4294967295)	符号付き32ビットデータと見なすと-2147483648~-1となり、符号なし32ビットデータと見なすと2147483648~4294967295となります。デバイス格納値としては同一です。
	・上記以外の数値 ・アルファベットや記号を含む文字列	0	変換できないため0が格納されます。
0120H: 16進数(16ビットデータ)	0H~FFFFH	0H~FFFFH	—
	・上記以外の数値 ・A~F以外のアルファベットや記号を含む文字列	0000H	変換できないため0が格納されます。
0121H: 16進数(32ビットデータ)	0H~FFFFFFFFH	0H~FFFFFFFFH	—
	・上記以外の数値 ・A~F以外のアルファベットや記号を含む文字列	00000000H	変換できないため0が格納されます。
0130H: 文字列(アスキーデータ)	文字列(1999文字まで)	文字列(1999文字まで)	末尾にNULL(00H)が追加されます。CSV内の文字列が偶数バイトの場合、次の1ワード分に0000Hが格納されます。 CSV内の文字列中に00Hが含まれている場合、この00Hは無視されます。
	文字列(2000文字以上)		1要素内の文字数が1999文字を超える場合は、1999文字までが1要素として読み出されます。2000文字目以降の文字は読み出されず、次の要素の読出しが行われます。
0140H: 浮動小数点実数(単精度実数)	$-2^{128} < \text{データ} \leq -2^{126}$, 0, $2^{-126} \leq \text{データ} < 2^{128}$ の範囲の値	左記範囲の値	小数点形式/指数形式のいずれの形式でも変換されます。
	上記以外の数値	0	変換できないため0が格納されます。
0141H: 浮動小数点実数(倍精度実数)	$-2^{1024} < \text{データ} \leq -2^{1022}$, 0, $2^{-1022} \leq \text{データ} < 2^{1024}$ の範囲の値	左記範囲の値	小数点形式/指数形式のいずれの形式でも変換されます。
	上記以外の数値	0	変換できないため0が格納されます。

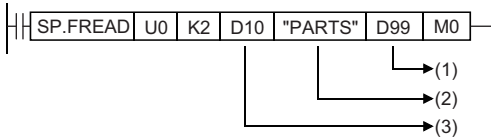
例

CSV形式変換読出し(文字列(アスキーデータ))を指定した場合
[CSV形式で保存したデータ]

PARTS.CSV

No.		Name		Value1		Value2	CR	LF
AA_0001	,	Prts_A	,	100	,	200	CR	LF
BB_0002	,	Prts_B	,	300	,	400	CR	LF

[デバイスに読み込まれるデータ]



[コントロールデータ]

D10	H0130
D11	H0000
D12	K6
D13	K100
D14	K2
D15	K2
D16	K3
D17	K2

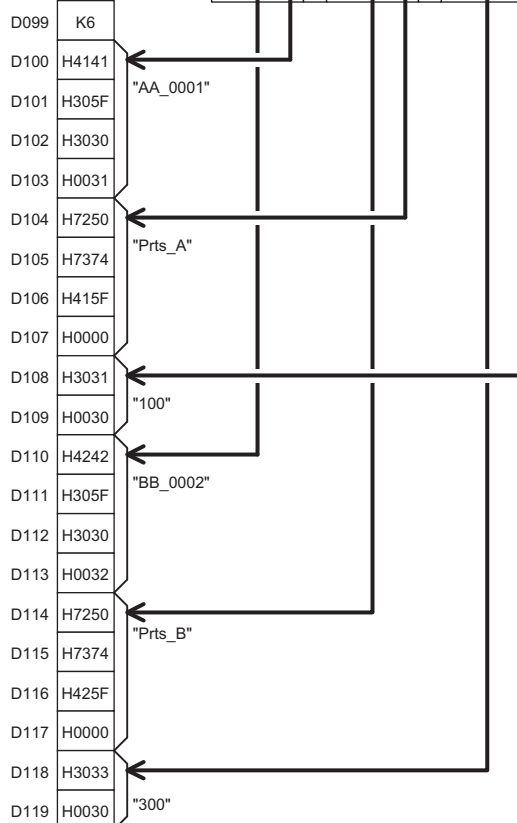
左記の囲んだ箇所の読出しを行います。
(PARTS.CSVの2行目から3列ずつ、合計6
要素の読出し)

- (1): 読み込まれたデータ
(2): ファイル名
(3): コントロールデータ

D10: 実行/完了タイプ
D11: 完了ステータス
D12: 要求読出しデータ数
D13: 最大読出しデータ数
D14, D15: ファイル位置
D16: 列数指定
D17: データタイプ指定

PARTS.CSV

No.		Name		Value1		Value2	CR	LF
AA_0001	,	Prts_A	,	100	,	200	CR	LF
BB_0002	,	Prts_B	,	300	,	400	CR	LF



D99: 読出し結果データ数
D100~D103: 2行目第1列の文字列
D104~D107: 2行目第2列の文字列
D108~D109: 2行目第3列の文字列
D110~D113: 3行目第1列の文字列
D114~D117: 3行目第2列の文字列
D118~D119: 3行目第3列の文字列

例

CSV形式変換読出し(10進数(16ビットデータ))指定時に指定列数が0の場合

[Excelで作成したデータ]

	A	B	C
1	大／小項目		測定値
2	長さ	1	3
3	温度	-21	
4			



[CSV形式で保存したデータ]

大／小項目	,	,	測定値	CR	LF
長さ	,	1	,	3	CR LF
温度	,	-21	,		CR LF



[デバイスに読み込まれるデータ]

SP.FREAD	U0	K2	D10	"ABCDE"	D99	M0

(1)
(2)
(3)

- (1): 読み込まれたデータ
- (2): ファイル名
- (3): コントロールデータ

[コントロールデータ]

D10	0100H
D11	-
D12	K9
D13	K0
D14	K0
D15	K0
D16	K0
D17	K0

- D10: 実行/完了タイプ
- D11: 完了ステータス
- D12: 要求読出しデータ数
- D13: 最大読出しデータ数
- D14, D15: ファイル位置
- D16: 列数指定
- D17: データタイプ指定

[読み込まれたデータ]

D99	K9
D100	K0
D101	K0
D102	K0
D103	K0
D104	K1
D105	K3
D106	K0
D107	K-21
D108	K0

- D99: 読出し結果データ数
- D100: 「大／小項目」は数値以外のデータであるため、変換データ(0)が格納されます。
- D101: 「」は数値以外のデータであるため、変換データ(0)が格納されます。
- D102: 「測定値」は数値以外のデータであるため、変換データ(0)が格納されます。
- D103: 「長さ」は数値以外のデータであるため、変換データ(0)が格納されます。
- D104: 「1」は数値であるため、バイナリ値に変換されます。
- D105: 「3」は数値であるため、バイナリ値に変換されます。
- D106: 「温度」は数値以外のデータであるため、変換データ(0)が格納されます。
- D107: 「-21」は数値であるため、バイナリ値に変換されます。
- D108: 「」は数値以外のデータであるため、変換データ(0)が格納されます。

- 列数がそれぞれの行で異なる場合でも、行を無視して読み出します。

例

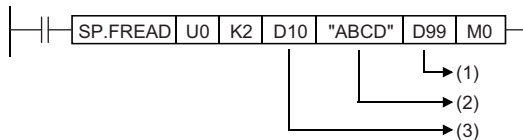
読み出したときの列数がそれぞれの行で異なる場合

[CSV形式で保存したデータ]

大／小項目	,	,	測定値	,	余分	CR	LF
長さ	CR	LF					
温度	,	-21	,	CR	LF		



[デバイスに読み込まれるデータ]



- (1): 読み込まれたデータ
- (2): ファイル名
- (3): コントロールデータ

[コントロールデータ]

D10	0100H
D11	-
D12	K7
D13	K0
D14	K0
D15	K0
D16	K0
D17	K0

- D10: 実行/完了タイプ
- D11: 完了ステータス
- D12: 要求読み出しデータ数
- D13: 最大読み出しデータ数
- D14, D15: ファイル位置
- D16: 列数指定
- D17: データタイプ指定

[読み込まれたデータ]

D99	K7
D100	K0
D101	K0
D102	K0
D103	K0
D104	K0
D105	K0
D106	K-21

- D99: 読み出し結果データ数
- D100: 「大／小項目」は数値以外のデータであるため、変換データ(0)が格納されます。
- D101: 「」は数値以外のデータであるため、変換データ(0)が格納されます。
- D102: 「測定値」は数値以外のデータであるため、変換データ(0)が格納されます。
- D103: 「余分」は数値以外のデータであるため、変換データ(0)が格納されます。
- D104: 「長さ」は数値以外のデータであるため、変換データ(0)が格納されます。
- D105: 「温度」は数値以外のデータであるため、変換データ(0)が格納されます。
- D106: 「-21」は数値であるため、バイナリ値に変換されます。

Point

Excelでは、列数がそれぞれの行で異なるファイルは作成されません。CSVファイルを修正した場合に発生します。

- 指定列数が0以外の場合、指定された列数の表としてCSV形式のファイルを読み出します。指定列数を超える要素は無視されます。

例

CSV形式変換読出し(10進数(16ビットデータ))指定時に指定列数が0以外の場合((d1)+6が2の場合)

[Excelで作成したデータ]

	A	B	C
1	大／小項目		測定値
2	長さ	1	3
3	温度	-21	
4			



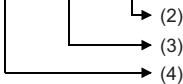
[CSV形式で保存したデータ]

大／小項目	,	,	測定値	CR	LF
長さ	,	1	,	3	CR LF
温度	,	-21		CR	LF



[デバイスに読み込まれるデータ]

SP.FREAD	U0	K2	D10	"ABCD"	D99	M0



(1): 指定列数を超える要素は無視されます。

- (2): 読み込まれたデータ
- (3): ファイル名
- (4): コントロールデータ

[コントロールデータ]

D10	0100H
D11	-
D12	K6
D13	K0
D14	K0
D15	K0
D16	K2
D17	K0

- D10: 実行/完了タイプ
- D11: 完了ステータス
- D12: 要求読出しデータ数
- D13: 最大読出しデータ数
- D14, D15: ファイル位置
- D16: 列数指定
- D17: データタイプ指定

[読み込まれたデータ]

D99	K6
D100	K0
D101	K0
D102	K0
D103	K1
D104	K0
D105	K-21

- D99: 読出し結果データ数
- D100: 「大／小項目」は数値以外のデータであるため、変換データ(0)が格納されます。
- D101: 「」は数値以外のデータであるため、変換データ(0)が格納されます。
- D102: 「長さ」は数値以外のデータであるため、変換データ(0)が格納されます。
- D103: 「1」は数値であるため、バイナリ値に変換されます。
- D104: 「温度」は数値以外のデータであるため、変換データ(0)が格納されます。
- D105: 「-21」は数値であるため、バイナリ値に変換されます。

- 列数がそれぞれの行で異なる場合でも、指定列数を超える要素は無視され、指定列数に足りない列は0が補われます。

例

読み出したときの列数がそれぞれの行で異なる場合

[CSV形式で保存したデータ]

大／小項目		,	,	測定値		,	余分	CR	LF
長さ	CR	LF							
温度	,	-21	,	CR	LF				

(1)

(1): 指定列数を超える要素は無視されます。



[デバイスに読み込まれるデータ]

SP.FREAD	U0	K2	D10	"ABCD"	D99	M0

(2)
(3)
(4)

(2): 読み込まれたデータ

(3): ファイル名

(4): コントロールデータ

[コントロールデータ]

D10	0100H
D11	-
D12	K6
D13	K0
D14	K0
D15	K0
D16	K2
D17	K0

D10: 実行/完了タイプ

D11: 完了ステータス

D12: 要求読出しデータ数

D13: 最大読出しデータ数

D14, D15: ファイル位置

D16: 列数指定

D17: データタイプ指定

[読み込まれたデータ]

D99	K6
D100	K0
D101	K0
D102	K0
D103	K0
D104	K0
D105	K-21

D99: 読出し結果データ数

D100: 「大/小項目」は数値以外のデータであるため、変換データ(0)が格納されます。

D101: 「」は数値以外のデータであるため、変換データ(0)が格納されます。

D102: 「長さ」は数値以外のデータであるため、変換データ(0)が格納されます。

D103: 該当する要素が無いため、変換データ(0)が補われます。

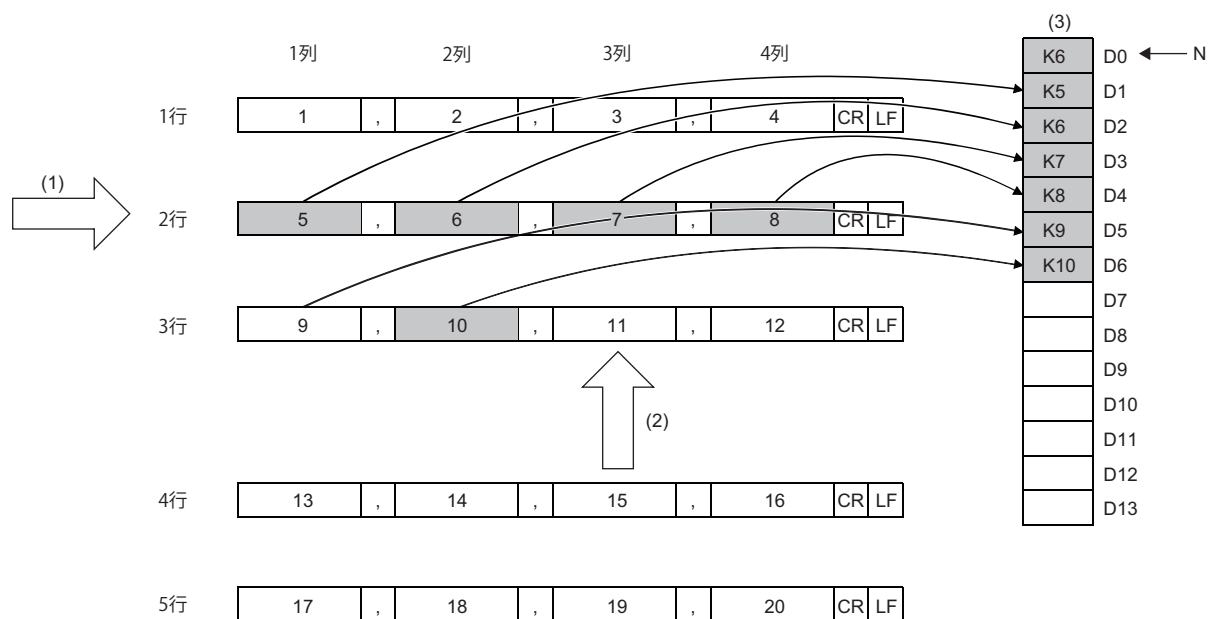
D104: 「温度」は数値以外のデータであるため、変換データ(0)が格納されます。

D105: 「-21」は数値であるため、バイナリ値に変換されます。

- CSV形式変換読み出しでは、データを複数回に分けて読み出すことが可能です。

[読み出しを開始したい行を指定します]

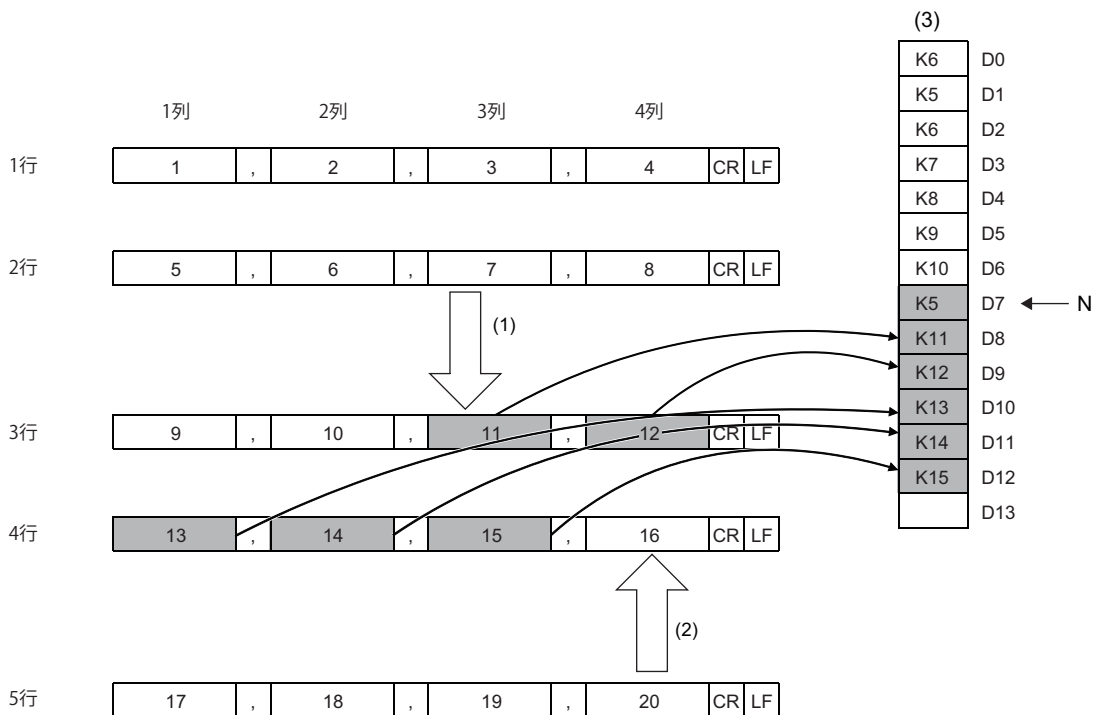
- 実行/完了タイプ: CSV形式変換読み出し(10進数(BIN16ビットデータ))
- 列数指定: 4H
- データタイプ指定: ワード
- ファイル位置: 2H
- 読み出し先頭デバイス: D0
- 読み出し結果データ数: 6H



- (1): 開始行
 (2): 次回開始位置
 (3): デバイス格納データ(読み出しデータ)
 N: データ数

[前回読出し位置の続きから読み出します]

- 実行/完了タイプ: CSV形式変換読出し(10進数(BIN16ビットデータ))
- 列数指定: 4H
- データタイプ指定: ワード
- ファイル位置: FFFFFFFFH(前回の読出し位置から継続)
- 読出し先頭デバイス: D7
- 読出し結果データ数: 5H



(1): 開始行

(2): 次回開始位置

(3): デバイス格納データ(読出しデータ)

N: データ数

Point

- 前回の読出し位置から継続して読み出しを行う場合は、「実行/完了タイプ」、「列数指定」、「データタイプ指定」を前回と異なった設定にすると、前回の読出し位置から正常に追加ができません。
- 前回の読出し位置から継続してデータを読み出ししている途中で、他の設定のSP.FREAD命令やSP.FWRITE命令を実行すると、前回の読出し位置から正常に追加ができません。

注意事項

- SP.FREAD命令は、割込みプログラムで実行しないでください。割込みプログラムで実行した場合、誤動作の原因となります。
- CSV形式変換読出し時、複数の要素数を読み出す場合は、読出しデータエリアには各要素の合計サイズ分のデバイスをあらかじめ確保した上で命令を実行してください。読出しデータは(d2)+1から格納されるため、(d2)として確保が必要なワード数は((各要素の合計ワード数)+1)ワードとなります。
- CSV形式変換読出し(文字列(アスキーデータ))指定時は、合計サイズ(ワード単位)を、(d1)+3の最大読出しデータ数に設定してください。
[例] CSVファイルから、1要素の文字数が100文字の文字列を、100要素読み出す場合
 $(100(\text{文字})+2(\text{NULL})) \times 100(\text{要素}) = 10200 \text{ バイト} = 5100 \text{ ワード}$
したがって、最大読出しデータ数((d1)+3)には5100を設定し、(d2)は5101ワード分の領域を確保してください。
- SP.FREAD命令はファイル読出しが発生するため、命令を実行するとスキャンタイムが延びることがあります。
- SP.FREAD命令はBOM付きの文字コードに対応していません。BOM付きの文字コードを使用したファイルを実行した場合、意図しないデータが読み出されます。EXCELで作成したBOM付きのCSVファイルの内容を読み出す場合は、テキストエディタなどで文字コードをBOMなしに変更してください。

エラー

エラーコード (SD0)	内容
2820H	読み出したデータのサイズが、読出しデバイスサイズを超えた場合。
3285H	ドライブ指定(s1)によって指定されたドライブが、SDメモリカード以外のとき。 コントロールデータ(d1)以降に設定された値が、設定範囲外のとき。 608ページ コントロールデータ(d1) (s2)で指定したファイル名文字列が読み出せないとき。 - 指定したファイル名文字列の文字数が範囲を超えている。 - 使用できない値を設定した。 - 指定したファイル名文字列の末尾に区切り文字を指定した。 - 指定したファイル名文字列のファイル名(拡張子を含まない)が0文字のとき。 608ページ コントロールデータ(d1)
3297H	(d1)+0の実行/完了タイプと(d1)+7のデータタイプ指定が、指定できない組合せになっているとき。 608ページ コントロールデータ(d1)

SP.FREAD命令が異常完了した場合、完了ステータス(d1)+1で指定したデバイスにエラーコードが格納されます。(命令が演算エラーになった場合は格納されません。)

完了ステータス(d1)+1に格納されるエラーコードについては、下記を参照してください。

657ページ ファイル操作命令で発生するエラーコード

指定ファイルへのデータライト

SP.FWRITE

デバイスデータをSDメモ리카ードの指定ファイルに書き込みます。

ラダー	ST
	ENO:=SP_FWRITE(EN,U,s1,s2,s3,d1,d2);

FBD/LD

■実行条件

命令	実行条件
SP.FWRITE	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(U)	ダミー	—	デバイス名	ANY16
(s1)	ドライブ指定	2(固定)*1	ワード	ANY16
(d1)	コントロールデータが格納されている先頭デバイス	622ページ コントロールデータ(d1)	ワード	ANY16_ARRAY (要素数: 8)
(s2)	ファイル名が格納されている先頭デバイス	—	Unicode文字列	ANYSTRING_DOUBLE
(s3)	データが格納されている先頭デバイス	—	ワード	ANY16*2
(d2)	処理完了にてONするビットデバイス	—	ビット	ANYBIT_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 SDメモ리카ードのドライブ2のみ設定可能です。

*2 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(U)	—	—	—	—	—	—	—	—	—	—	—	○
(s1)	○	—	○	—	—	—	—	○	○	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	○	—
(s3)	—	—	○	—	—	—	—	○	—	—	—	—
(d2)	○	—	○	—	—	—	—	—	—	—	—	—

■コントロールデータ(d1)

オペランド: (d1)				
デバイス	項目	内容	設定範囲	セット側
+0	実行/完了タイプ	実行タイプを指定します。 ■00**H: バイナリ書込み ・ 0000H: BIN16ビットデータ ・ 0001H: BIN32ビットデータ ■01**H: CSV形式変換書込み ・ 0100H: 10進数(符号付き16ビットデータ) ・ 0101H: 10進数(符号なし16ビットデータ) ・ 0110H: 10進数(符号付き32ビットデータ) ・ 0111H: 10進数(符号なし32ビットデータ) ・ 0120H: 16進数(16ビットデータ) ・ 0121H: 16進数(32ビットデータ) ・ 0130H: 文字列(アスキーデータ) ・ 0140H: 浮動小数点実数(単精度実数) ・ 0141H: 浮動小数点実数(倍精度実数)	0000H 0001H 0100H 0101H 0110H 0111H 0120H 0121H 0130H 0140H 0141H	ユーザ
+1	完了ステータス	命令完了時に完了ステータスを格納します。 ・ 0000H: 正常完了 ・ 0000H以外: 異常完了(エラーコード)( 657ページ ファイル操作命令で発生するエラーコード)	—	システム
+2	書込み結果データ数	(s3)で指定するデータに対して、実際に書き込んだデータ数が入ります。 値の単位は、(d1)+7のデータタイプ指定によります。	—	システム
+3	使用用途設定エリア	b15 ... b0 0 1/0 b0: 書込み開始位置設定*3 (d1)+0で「バイナリ書込み」を指定したときは、必ず0を指定してください。 (d1)+0で「CSV形式変換書込み」、(d1)+4、(d1)+5で「ファイルの最後に追加」を指定したときは、書込み開始位置を指定してください。 ・ 0: ファイルの最後に追加 ・ 1: ファイルの最後の改行コードをカンマに変換して追加(最後の行の続きから書込み)	左記	ユーザ
+4 +5	ファイル位置	■(d1)+0で「バイナリ書込み」を指定したとき ・ 00000000H: ファイルの先頭から ・ 00000001H~FFFFFFEH: 指定位置から(単位は(d1)+7のデータタイプ指定によります。) ・ FFFFFFFFH: ファイルの最後に追加 ■(d1)+0で「CSV形式変換書込み」を指定したとき ・ 00000000H~FFFFFFEH: ファイルの先頭から ・ FFFFFFFFH: ファイルの最後に追加	00000000H~ FFFFFFFHH	ユーザ
+6	列数指定	(d1)+0で「バイナリ書込み」を指定したときは、必ず0を指定してください。 (d1)+0で「CSV形式変換書込み」を指定したときは、書込みを行う列数を指定してください。 ・ 0: 列はなし。1行になります。 ・ 0以外: 指定数の列になります。	0000H~FFFFH (0~65535)	ユーザ
+7	データタイプ指定	・ 0: ワード ・ 1: 偶数バイト*1 ・ 2: (d1)+0で指定したデータ型の単位 ・ 3: 奇数バイト*1*2 「0: ワード」および「1: 偶数バイト」、「3: 奇数バイト」は、(d1)+0で「0000H: BIN16ビットデータ」または「0100H: 10進数(符号付き16ビットデータ)」を指定した場合のみ指定できます。	0, 1, 2, 3	ユーザ

*1 データタイプ指定(d1)+7に、「1: 偶数バイト」または「3: 奇数バイト」を指定している場合、要求書込みデータ数(s3)+0の設定範囲は1~32767となります。

*2 奇数バイト(d1)+7指定時は書き込みたい奇数バイトデータ数に1バイト加算しワード単位にします。ワード単位にしたデータ数を要求書込みデータ数(s3)+0に指定してください。

*3 実行/完了タイプ(d1)+0で"CSV形式変換書込み"かつファイル位置(d1)+4、(d1)+5で"ファイルの最後に追加"を指定した場合のみ有効です。それ以外の場合はエラーとなります。

■ファイル名(s2)

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+0~+□	ファイル名文字列	ファイルが格納されているフォルダパス、ファイル名の文字列を指定します。 - フォルダパス+ファイル名(拡張子を含む)は253文字以内にしてください。 - フォルダパスは244文字以内にしてください。(ファイルの区切り文字は含まない。) - フォルダパスの階層数は10以内にしてください。 - ファイル名の拡張子を省略する場合は「.」(ピリオド)から省略してください。 - ファイル名(拡張子を含まない)は1文字以上60文字以内、拡張子はピリオド+3文字以内にしてください。ファイル名(拡張子を含まない)が61文字以上のときは、拡張子があっても無視され、「.BIN」または「.CSV」になります。 "/folder1/user1/user1.csv" (1) (2) (3) (1): 最大253文字 (2): フォルダパス、ファイルの区切り文字は"/"または"¥"を使用する。 (3): 省略可能。省略した場合、(1)の最大は252文字となる。	Unicode文字列	ユーザ

■書込みデータ(s3)

オペランド: (s3)				
デバイス	項目	内容	設定範囲	セット側
+0	要求書込みデータ数	書込みを要求するデータ数を指定します。 (d1)+0の実行/完了タイプの設定により、処理単位および設定範囲が異なります。 ■(d1)+0で「バイナリ書込み」を指定したとき - BIN16ビットデータ指定時: ワード単位(1~65535)*1*2 - BIN32ビットデータ指定時: ダブルワード単位(1~32767) ■(d1)+0で「CSV形式変換書込み」を指定したとき: 要素数 10進数(符号付き16ビットデータ)指定時: ワード単位(1~65535)*1*2 10進数(符号なし16ビットデータ)指定時: ワード単位(1~65535) 10進数(符号付き32ビットデータ)指定時: ダブルワード単位(1~32767) 10進数(符号なし32ビットデータ)指定時: ダブルワード単位(1~32767) 16進数(16ビットデータ)指定時: ワード単位(1~65535) 16進数(32ビットデータ)指定時: ダブルワード単位(1~32767) - 文字列(アスキーデータ)指定時: 要素数(1~1023) - 浮動小数点実数(単精度実数)指定時: ダブルワード単位(1~32767) - 浮動小数点実数(倍精度実数)指定時: 4ワード単位(1~16383)	1~65535	ユーザ
+1~+□	書込みデータ	書込みを要求するデータを格納します。	0000H~FFFFH	ユーザ

*1 データタイプ指定(d1)+7に、「1: 偶数バイト」または「3: 奇数バイト」を指定している場合、要求書込みデータ数(s3)+0の設定範囲は1~32767となります。

*2 奇数バイト(d1)+7指定時は書き込みたい奇数バイトデータ数に1バイト加算しワード単位にします。ワード単位にしたデータ数を要求書込みデータ数(s3)+0に指定してください。

機能

- 指定されたデータ数のデータを指定されたファイルに書き込みます。コントロールデータの実行/完了タイプにより、ファイルの書き込み形式を指定します。
- 書き込みの対象は、SDメモ리카ードのみです。
- 処理完了(d2)のビットデバイスは、SP.FWRITE命令の処理完了を検出したスキンのEND命令実行時に自動的にONし、次のスキンのEND命令にてOFFします。
- SP.FWRITE命令が異常完了時には、異常完了(d2)+1のデバイスが、処理完了(d2)のデバイスと同じタイミングでON/OFFします。
- SP.FWRITE命令実行中は、SM753(ファイルアクセス中)がONします。
- SM753がON中は、SP.FWRITE命令は実行できません。(実行した場合は無処理となります。)
- 命令実行時に検出したエラーについては、処理完了(d2)、異常完了(d2)+1、およびSM753はONしません。
- 要求書き込みデータ数((s3)+0)、ファイル位置((d1)+4, (d1)+5)、および書き込み結果データ数((d1)+2)の処理単位は、実行/完了タイプ(d1)+0およびデータタイプ指定(d1)+7の組合せにより指定します。

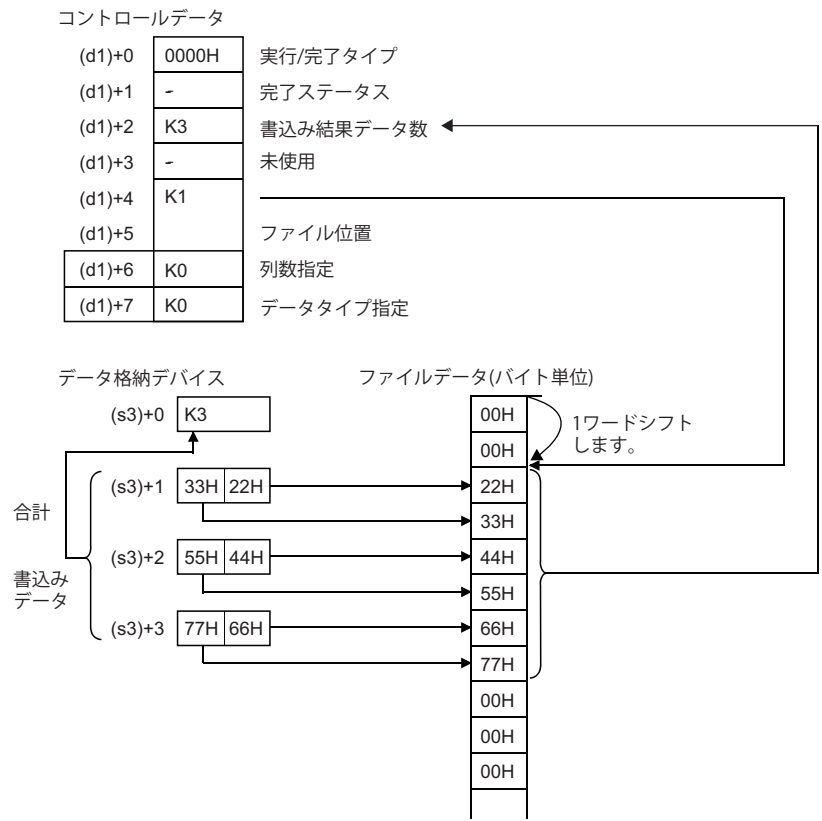
実行/完了タイプ (d1)+0		データタイプ指定 (d1)+7	処理単位および設定範囲		
			要求書き込みデータ数 (s3)+0	ファイル位置 (d1)+4, (d1)+5	書き込み結果データ数 (d1)+2
バイナリ書き込み	0000H: BIN16ビットデータ	0: ワード	ワード(1~65535)	ワード (00000000H~7FFFFFFFH, FFFFFFFFH)	ワード
		1: 偶数バイト	ワード(1~32767)	バイト (00000000H~FFFFFFFH)	バイト
		2: 実行/完了タイプで指定したデータ型の単位	ワード(1~65535)	ワード (00000000H~7FFFFFFFH, FFFFFFFFH)	ワード
		3: 奇数バイト	ワード(1~32767)	バイト (00000000H~FFFFFFFH)	バイト
	0001H: BIN32ビットデータ	0: ワード 1: 偶数バイト	(指定不可)		
		2: 実行/完了タイプで指定したデータ型の単位	ダブルワード(1~32767)	ダブルワード (00000000H~3FFFFFFFH, FFFFFFFFH)	ダブルワード
		3: 奇数バイト	(指定不可)		

実行/完了タイプ (d1)+0		データタイプ指定 (d1)+7	処理単位および設定範囲		
			要求書込みデータ数 (s3)+0	ファイル位置 (d1)+4, (d1)+5	書込み結果データ数 (d1)+2
CSV形式変換 書込み	0100H: 10進数(符号付き16 ビットデータ)	0: ワード	ワード(1~65535)	先頭/末尾 ^{*3}	ワード
		1: 偶数バイト	ワード(1~32767)	先頭/末尾 ^{*3}	バイト
		2: 実行/完了タイプで指定 したデータ型の単位	ワード(1~65535)	先頭/末尾 ^{*3}	ワード
		3: 奇数バイト	ワード(1~32767)	先頭/末尾 ^{*3}	バイト
	0101H: 10進数(符号なし16 ビットデータ)	0: ワード	(指定不可)		
		1: 偶数バイト	(指定不可)		
		2: 実行/完了タイプで指定 したデータ型の単位	ワード(1~65535)	先頭/末尾 ^{*3}	ワード
		3: 奇数バイト	(指定不可)		
	0110H: 10進数(符号付き32 ビットデータ)	0: ワード	(指定不可)		
		1: 偶数バイト	(指定不可)		
		2: 実行/完了タイプで指定 したデータ型の単位	ダブルワード(1~32767)	先頭/末尾 ^{*3}	ダブルワード
		3: 奇数バイト	(指定不可)		
	0111H: 10進数(符号なし32 ビットデータ)	0: ワード	(指定不可)		
		1: 偶数バイト	(指定不可)		
		2: 実行/完了タイプで指定 したデータ型の単位	ダブルワード(1~32767)	先頭/末尾 ^{*3}	ダブルワード
		3: 奇数バイト	(指定不可)		
	0120H: 16進数(16ビットデー タ)	0: ワード	(指定不可)		
		1: 偶数バイト	(指定不可)		
		2: 実行/完了タイプで指定 したデータ型の単位	ワード(1~65535)	先頭/末尾 ^{*3}	ワード
		3: 奇数バイト	(指定不可)		
	0121H: 16進数(32ビットデー タ)	0: ワード	(指定不可)		
		1: 偶数バイト	(指定不可)		
		2: 実行/完了タイプで指定 したデータ型の単位	ダブルワード(1~32767)	先頭/末尾 ^{*3}	ダブルワード
		3: 奇数バイト	(指定不可)		
	0130H: 文字列(アスキーデー タ)	0: ワード	(指定不可)		
		1: 偶数バイト	(指定不可)		
		2: 実行/完了タイプで指定 したデータ型の単位	要素数(1~1023)	先頭/末尾 ^{*3}	要素数
		3: 奇数バイト	(指定不可)		
	0140H: 浮動小数点実数(単精 度実数)	0: ワード	(指定不可)		
		1: 偶数バイト	(指定不可)		
		2: 実行/完了タイプで指定 したデータ型の単位	ダブルワード(1~32767)	先頭/末尾 ^{*3}	ダブルワード
		3: 奇数バイト	(指定不可)		
	0141H: 浮動小数点実数(倍精 度実数)	0: ワード	(指定不可)		
		1: 偶数バイト	(指定不可)		
		2: 実行/完了タイプで指定 したデータ型の単位	4ワード(1~16383)	先頭/末尾 ^{*3}	4ワード
		3: 奇数バイト	(指定不可)		

*3 設定値は、00000000H~FFFFFFFFH(ファイルの先頭から)、またはFFFFFFFFH(ファイルの最後に追加)のいずれかとなります。

■バイナリ書込み時

- 対象ファイルの拡張子を省略した場合は、拡張子は「.BIN」となります。
- 存在しないファイルが指定された場合は、該当ファイルを新規に作成して、先頭からデータを追加保存します。
- 存在するファイルが指定された場合は、存在するファイルの先頭から保存します。データを書いている途中で、既存の大きさを超えた場合、超えた部分のデータは追加保存されます。
- 既存のファイルの大きさよりも、大きいファイル位置が指定された場合は、0点での書込みとなり正常完了します。
- データを追加保存している途中で、媒体から空きエリアがなくなってしまった場合はエラーとなります。このとき、書込みおよび追加保存に成功している部分は、そのまま書き込まれた状態になり、追加保存できる分だけ追加保存した後でエラー終了します。
- バイナリ書込み(BIN16ビットデータ)の場合で要求書込みデータ数、ファイル位置が指定された場合のデータの書込み方法は下記になります。

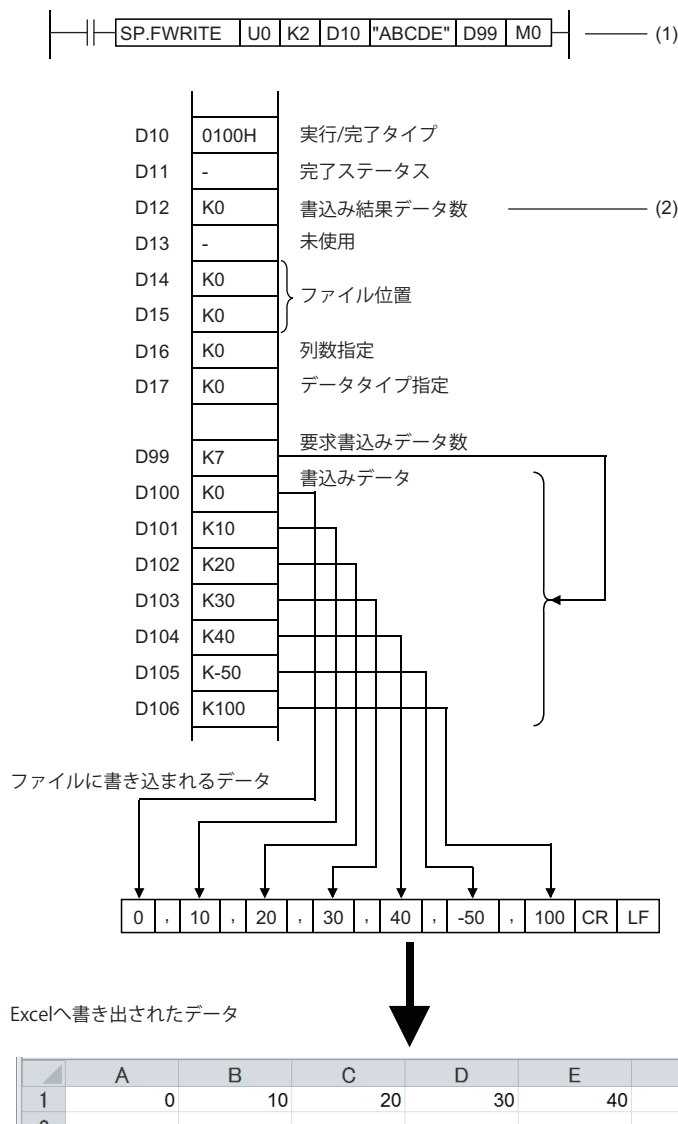


■CSV形式変換書込み時

- 拡張子を省略した場合は、拡張子は「.CSV」となります。
- 存在するファイルが指定された場合は、下記ようになります。
- (d1)+4, (d1)+5でFFFFFFFFH以外を設定すると、ファイルの内容をすべて削除し先頭からデータを保存します。
- (d1)+4, (d1)+5でFFFFFFFFHを設定すると、ファイルの最後からデータを保存します。このとき、(d1)+3で1を指定した場合、ファイルの最後の改行コードをカンマに変換して、データを保存します。ただし、ファイルの最後が改行コード以外の場合はカンマへの変換を行わず、ファイルの最後からデータを保存します。
- 存在しないファイルが指定された場合は、該当ファイルを新規に作成して、先頭からデータを追加保存します。
- データを追加保存している途中で、媒体から空きエリアがなくなった場合はエラーとなります。このとき、追加保存に成功している部分は、そのまま書き込まれた状態となり、追加保存できる分だけ追加保存した後でエラー終了します。
- 指定列数が0の場合、1行のCSV形式ファイルとして保存します。

例

CSV形式変換書込み時に指定列数が0の場合

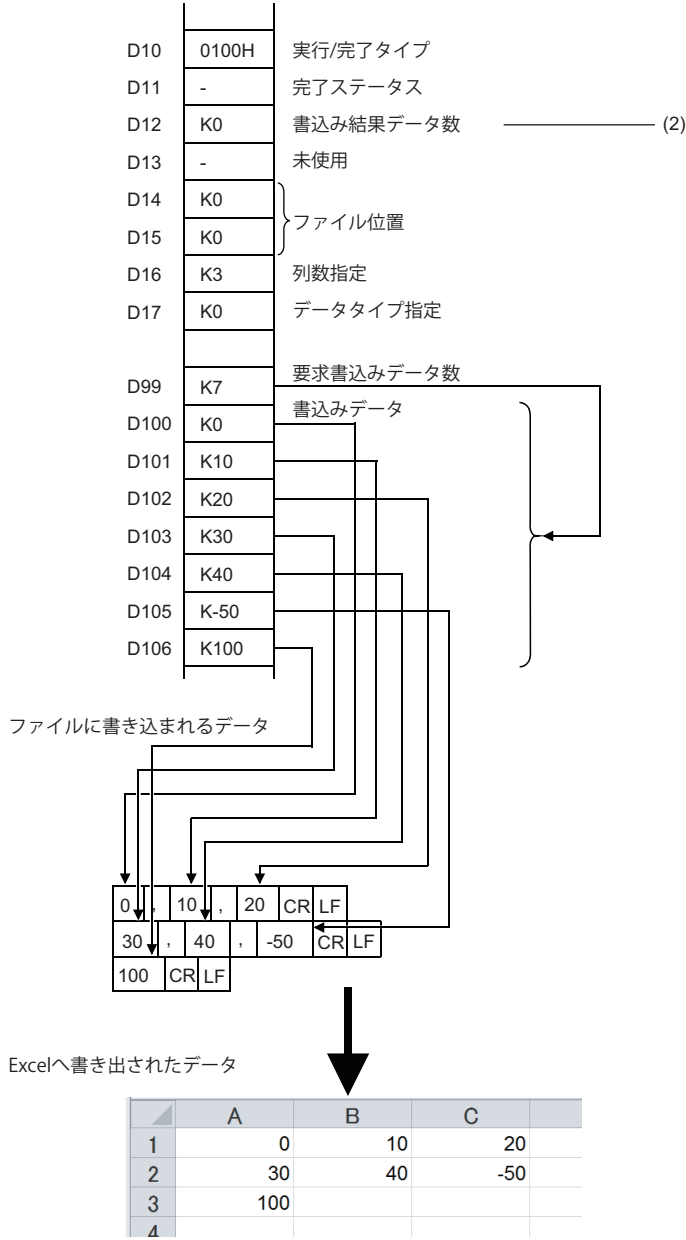
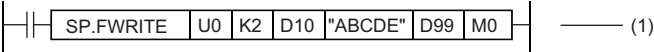


- (1) ワード単位指定とします。
 (2) 正常完了すると、書き込みデータ数と同じになります。

- 指定列数が0以外の場合、指定された列数の表としてCSV形式のファイルを保存します。

例

CSV形式変換書き込み時に指定列数が0以外の場合

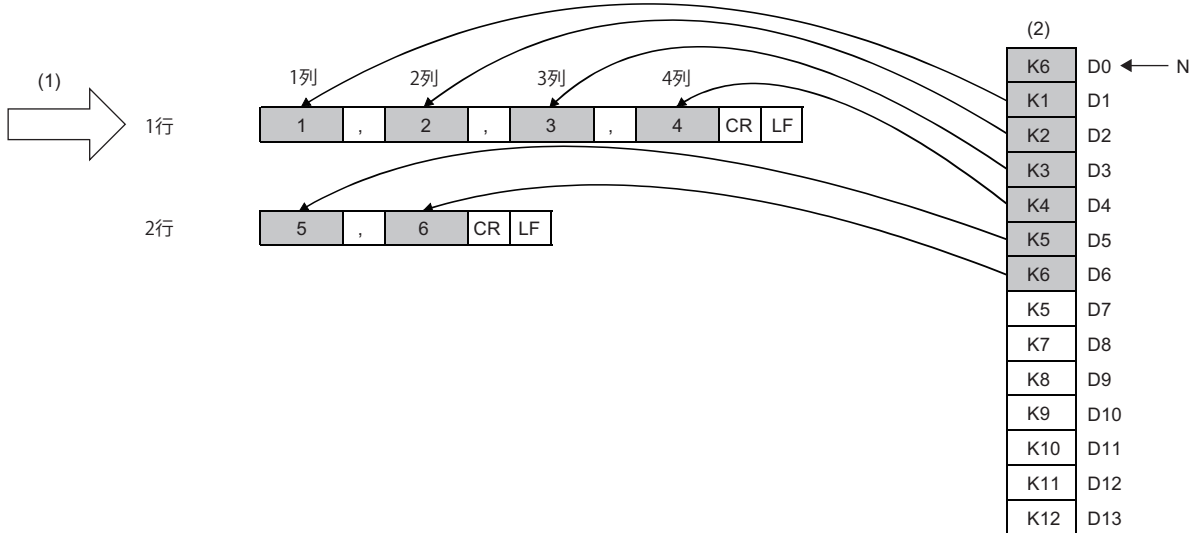


- (1) ワード単位指定とします。
(2) 正常完了すると、書き込みデータ数と同じになります。

- データを追加する場合は下記ようになります。

[書き込みを行うファイルを指定](ファイルが存在していても、削除して再度新規作成します。)

- 実行/完了タイプ: CSV形式変換書き込み(10進数(符号付き16ビットデータ))
- ファイル位置: 00000000H(ファイルの先頭から)
- 列数指定: 4H
- データタイプ指定: ワード
- 書き込み先頭デバイス: D0
- 要求書き込みデータ数: 6H



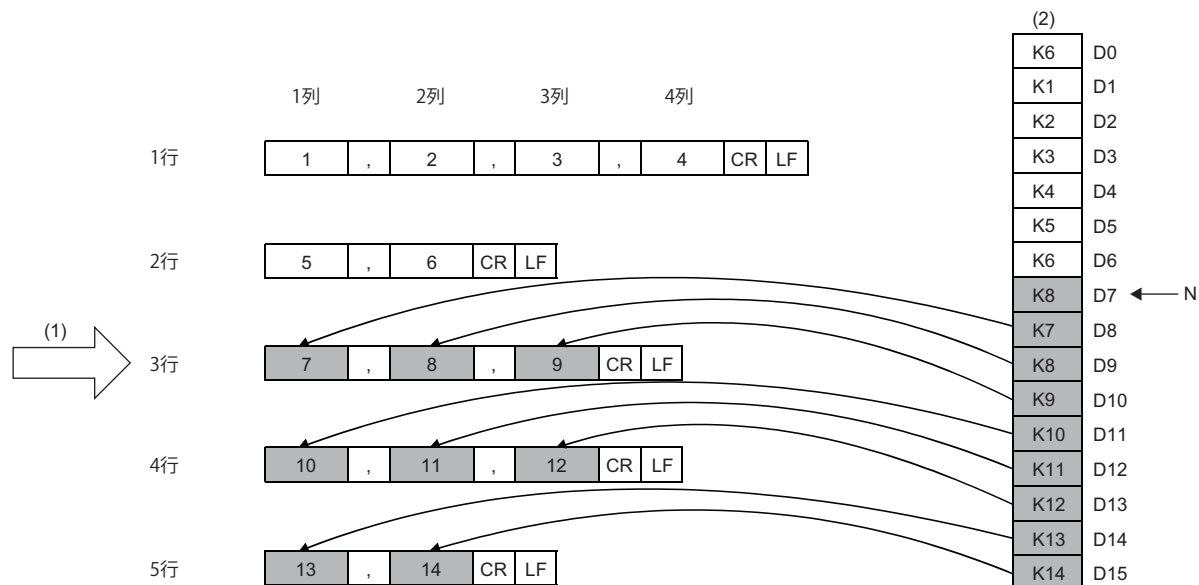
(1): 開始行

(2): デバイス格納データ(書き込みデータ)

N: データ数

[ファイルの最後に追加]

- 実行/完了タイプ: CSV形式変換書き込み(10進数(符号付き16ビットデータ))
- 使用用途設定エリア: 0H
- ファイル位置: FFFFFFFFH(ファイルの最後に追加)
- 列数指定: 3H
- データタイプ指定: ワード
- 書き込み先頭デバイス: D7
- 要求書き込みデータ数: 8H



(1): 今回開始位置

(2): デバイス格納データ(書き込みデータ)

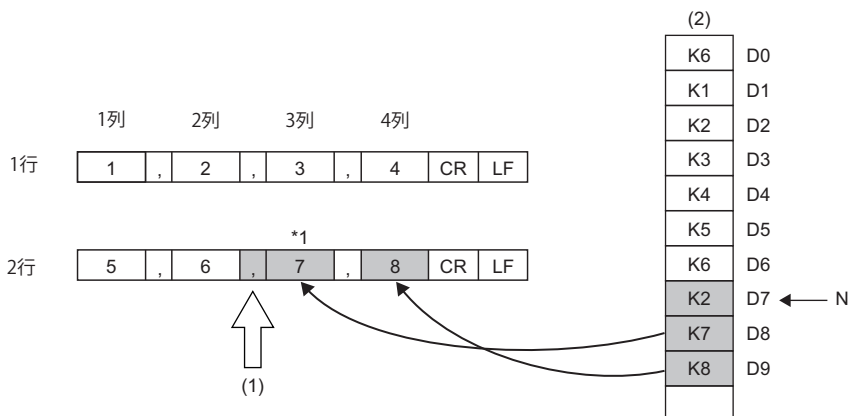
N: データ数

Point

- 「要求書き込みデータ数」は、「列数指定」の整数倍に設定してください。整数倍ではない場合、列数がバラバラになります。
- 最後のデータの後ろには必ず改行コードが入るので、使用用途設定エリアが0の場合、新しい行の先頭から追加されます。
- ファイルの最後に追加する場合、「列数指定」を前回の書き込み時から変更すると、列数がずれます。

[ファイルの最後の行の続きから追加]

- 実行/完了タイプ: CSV形式変換書込み(10進数(符号付き16ビットデータ))
- 使用用途設定エリア: 1H
- ファイル位置: FFFFFFFFH(ファイルの最後に追加)
- データタイプ指定: ワード
- 書込み先頭デバイス: D7
- 要求書込みデータ数: 2H



(1): 今回開始位置

(2): デバイス格納データ(書込みデータ)

N: データ数

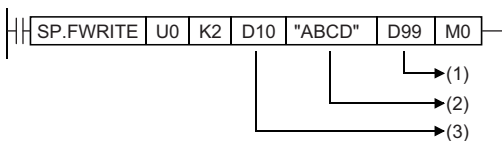
*1 使用用途設定エリアで1を設定した場合、ファイル末尾の改行コードがカンマに変換され、最後の行の続きからデータを追加できます。

- ・実行/完了タイプで「文字列(アスキーデータ)」を指定した場合の例を下記に示します。

例

CSV形式書込み(文字列(アスキーデータ))の場合

[ファイルに書き込まれるデータ]

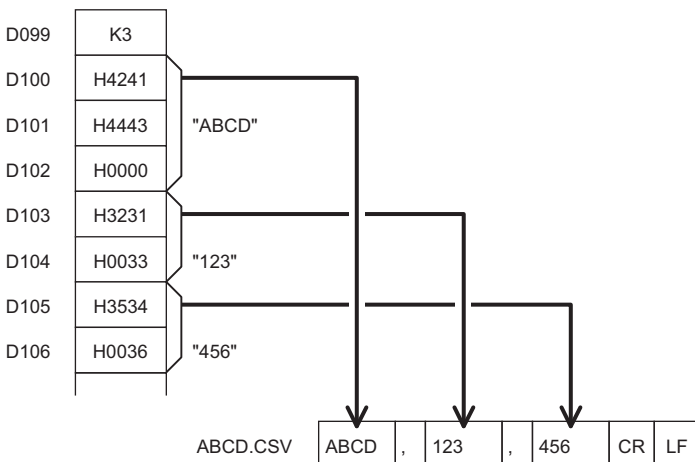


- (1): 書き込むデータ
- (2): ファイル名
- (3): コントロールデータ

[コントロールデータ]

D10	H0130
D11	H0000
D12	-
D13	-
D14	K0
D15	K0
D16	K0
D17	K2

- D10: 実行/完了タイプ: 文字列(アスキーデータ)
- D11: 完了ステータス
- D12: 書込み結果データ数
- D13: (未使用)
- D14, D15: ファイル位置
- D16: 列数指定
- D17: データタイプ指定



- D99: 要求書込みデータ数
- D100~D102: 1行目第1列に書き込む文字列
- D103~D104: 1行目第2列に書き込む文字列
- D105~D106: 1行目第3列に書き込む文字列

	A	B	C	D
1	ABCD	123	456	
2				

Point

- ・1要素の文字列の終端には必ず00H(NULL)を設定してください。文字列が偶数バイトの場合は、次の1ワードに0000H(2バイト分のNULL)を設定してください。
- ・1要素の最大文字数は1999文字です。この文字数を超過00H(NULL)が格納されていない場合は、2000文字目以降は書き込まず、次の要素の書込みが行われます。
- ・1回の命令実行で書き込み可能な要素数の最大は1023です。

- 実行/完了タイプ((d1)+0)に「0140H: 浮動小数点実数(単精度実数)」または「0141H: 浮動小数点実数(倍精度実数)」を設定した場合に、(s3)+1以降に設定する値と、CSVファイルに書き込まれる内容を下記に示します。

実行/完了タイプ((d1)+0)	書き込みデータ((s3)+1以降)に設定する値	CSVファイルに書き込まれる内容
0140H: 浮動小数点実数(単精度実数)	$-2^{128} < \text{データ} \leq -2^{126}$, 0, $2^{-126} \leq \text{データ} < 2^{128}$ の範囲の値	左記の値(小数部桁数0~7)が指数形式で書き込まれます。
	上記以外の値	変換できないため0が書き込まれます。
0141H: 浮動小数点実数(倍精度実数)	$-2^{1024} < \text{データ} \leq -2^{1022}$, 0, $2^{-1022} \leq \text{データ} < 2^{1024}$ の範囲の値	左記の値(小数部桁数0~15)が指数形式で書き込まれます。
	上記以外の値	変換できないため0が書き込まれます。

- SDメモ리카ードにCSV形式のファイルを書き込む場合のファイルサイズ(合計バイト数)の算出方法は下記になります。

[合計バイト数]=[最終行以外の合計バイト数]+[最終行のバイト数]

([各行のバイト数]=[列数*2]+1+[行内の各データ値のバイト数の合計*3])

- *2 最終行以外は、指定した列数となります。最終行の列数は、書き込みデータ数により指定した列数と異なる場合があるため、下記のよう

- 最終行を除いた行数を算出します。(最終行を除いた行数=要求書き込みデータ数-列数(余りは切捨て))
- 最終行の列数を算出します。(最終行の列数=要求書き込みデータ数-(最終行を除いた行数×列数))

- *3 各データ値のバイト数は下記の表のように算出します。

データ値の符号	各データ値のバイト数	バイト数の範囲	例
正	桁数	1~5(ワード指定時) 1~3(バイト指定時)	• 12345→5バイト • 67→2バイト
負	桁数+1	2~6(ワード指定時) 2~4(バイト指定時)	• -12345→6バイト • -67→3バイト

注意事項

- SP.FWRITE命令は、割込みプログラムで実行しないでください。割込みプログラムで実行した場合、誤動作の原因となります。
- SP.FWRITE命令はファイル書き込みが発生するため、命令を実行するとスキャンタイムが延びることがあります。
- SP.FWRITE命令はBOM付きの文字コードに対応していません。EXCELで作成したBOM付きのCSVファイルに対して書き込む場合は、テキストエディタ等で文字コードをBOMなしに変更してください。

エラー

エラーコード (SD0)	内容
2820H	要求書き込みデータ数(s3)+0で指定する値が設定範囲外の時、または(s3)+1以降の該当デバイス/ラベルの割付範囲を超えるとき。
3285H	ドライブ指定(s1)によって指定されたドライブが、SDメモリカード以外の時。 コントロールデータ(d1)以降に設定された値が、設定範囲外の時。 622ページ コントロールデータ(d1) (s2)で指定したファイル名文字列が読み出せないとき。 - 指定したファイル名文字列の文字数が範囲を超えている。 - 使用できない値を設定した。 - 指定したファイル名文字列の末尾に区切り文字を指定した。 - 指定したファイル名文字列のファイル名(拡張子を含まない)が0文字の時。 622ページ コントロールデータ(d1)
3297H	(d1)+0の実行/完了タイプと(d1)+7のデータタイプ指定が、指定できない組合せになっているとき。 (d1)の実行/完了タイプと(d1)+3の書き込み開始位置設定と(d1)+4のファイル位置指定が、指定できない組合せになっているとき。 622ページ コントロールデータ(d1)

SP.FWRITE命令が異常完了した場合、完了ステータス(d1)+1で指定したデバイスにエラーコードが格納されます。(命令が演算エラーになった場合は格納されません。)

完了ステータス(d1)+1に格納されるエラーコードについては、下記を参照してください。

657ページ ファイル操作命令で発生するエラーコード

指定ファイルの削除

SP.FDELETE

SDメモリカード内の指定したファイル，またはフォルダを削除します。

ラダー	ST
	ENO:=SP_FDELETE(EN,U,s1,s2,d1,d2);

FBD/LD

■実行条件

命令	実行条件
SP.FDELETE	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(U)	ダミー	—	デバイス名	ANY16
(s1)	ドライブ指定	2(固定)*1	ワード	ANY16
(d1)	コントロールデータが格納されている先頭デバイス	≡ 636ページ コントロールデータ(d1)	ワード	ANY16_ARRAY (要素数: 2)
(s2)	ファイル名/フォルダ名が格納されている先頭デバイス	—	Unicode文字列	ANYSTRING_DOUBLE
(d2)	処理完了にてONするビットデバイス	—	ビット	ANYBIT_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 SDメモリカードのドライブ2のみ設定可能です。

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他 (U)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥ G□, J□¥□	Z	LT, LST, LC	LZ	K, H		E	\$		
(U)	—	—	—	—	—	—	—	—	—	—	—	—	○
(s1)	○	—	○	—	—	—	—	○	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	○	—	—
(d2)	○	—	○	—	—	—	—	—	—	—	—	—	—

■コントロールデータ(d1)

オペランド: (d1)				
デバイス	項目	内容	設定範囲	セット側
+0	使用用途設定エリア	b15 ... b1 b0 0 1/0 1/0 b0: 対象種別設定 指定対象がファイルかフォルダかを指定します。 • 0: ファイル指定 • 1: フォルダ指定 b1: 空フォルダ削除設定 フォルダ削除時に、空フォルダのみ削除するかどうかを指定します。 • 0: 空フォルダでない場合も削除する • 1: 空フォルダの場合のみ削除する	左記	ユーザ
+1	完了ステータス	命令完了時に完了ステータスを格納します。 • 0000H: 正常完了 • 0000H以外: 異常完了(エラーコード)(657ページ ファイル操作命令で発生するエラーコード)	—	システム

■ファイル名/フォルダ名(s2)

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+0~+0	ファイル名/フォルダ名	ファイルを指定する場合、削除するファイルが格納されているフォルダパス+ファイル名を指定します。 フォルダを指定する場合、削除するフォルダのフォルダパスを指定します。 • ファイル名に拡張子が存在する場合、省略しないでください。 • ファイル名はピリオドと拡張子を含む64文字以内にしてください。 • フォルダパス+ファイル名(ピリオド、拡張子を含む)は253文字以内にしてください。 • フォルダパスは244文字以内にしてください。(ファイルの区切り文字は含まない。) • ファイル名、もしくはフォルダパスは区切り文字を除いて1文字以上指定してください。 • ファイル指定時、文字列の末尾に区切り文字を指定しないでください。 • 文字列の末尾、または各区切り文字の直前に半角ピリオドを指定しないでください。 • フォルダパスの階層数は10以内にしてください。 ■ファイルを指定する場合 (1) "/folder1/user1/user1.csv" (3) (2) (1): 最大253文字 (2): フォルダパス、ファイルの区切り文字は"/"または"¥"を使用する。 (3): 省略可能。省略した場合、(1)の最大は252文字となる。 ■フォルダを指定する場合 (4) "/folder1/user1/subfolder1/" (6) (5) (7) (4): 最大244文字 (5): フォルダパスの区切り文字は"/"または"¥"を使用する。 (6): 省略可能。省略した場合、(4)の最大は243文字となる。 (7): 省略可能	Unicode文字列	ユーザ

機能

- (s1)で指定したドライブの(s2)で指定したファイル，またはフォルダを削除します。
- SP.FDELETE命令実行中はSM753(ファイルアクセス中)がONします。SM753がON中は，SP.FDELETE命令は実行できません。(実行した場合は無処理となります。)
- 処理完了(d2)のビットデバイスは，SP.FDELETE命令の処理完了を検出したスキンのEND命令実行時に自動的にONし，次のスキンのEND命令にてOFFします。
- SP.FDELETE命令が異常完了時には，異常完了(d2)+1のデバイスが，処理完了(d2)のデバイスと同じタイミングでON/OFFします。
- 命令実行時に演算エラーを検出した場合，処理完了(d2)，異常完了(d2)+1はONしません。

注意事項

- SP.FDELETE命令は，割込みプログラムで実行しないでください。割込みプログラムで実行した場合，誤動作の原因となります。
- 命令実行中にコントローラの状態をRUN→STOPに変更しても，本命令は処理を継続します。
- 命令が処理の途中で異常完了した場合，すでに削除されたファイル，またはフォルダは復元されません。
- 削除対象ファイルのファイルサイズやファイル数が大きくなる場合，命令が完了するまでの時間が長くなります。
- SP.FDELETE命令で操作中のファイルには，他機能からアクセスしないでください。(ファイルが破損，またはエラーとなる場合があります。)
- 他機能からアクセスしているファイル/フォルダへの操作は行わないでください。

エラー

エラーコード (SD0)	内容
3285H	ドライブ指定(s1)によって指定されたドライブが，SDメモリカード以外のとき。 (s2)で指定したファイル名/フォルダ名文字列が読み出せないとき。 • ファイル名文字列が1文字以上指定されていない。 • 指定したファイル名文字列に254文字以上を指定した。 • 指定したフォルダパスに245文字以上を指定した。 • 指定したフォルダパスに11以上の階層を指定した。 • 指定したファイル名文字列の末尾，または各区切り文字の直前に半角ピリオドを指定した。 • ファイル指定時，指定したファイル名文字列の末尾に区切り文字を指定した。 (s2)で指定したフォルダパス内にルートフォルダ直下のシステムフォルダ(\$MELPRJ\$)を指定したとき。

SP.FDELETE命令が異常完了した場合，完了ステータス(d1)+1で指定したデバイスにエラーコードが格納されます。(命令が演算エラーになった場合は格納されません。)

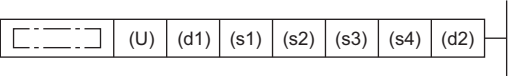
完了ステータス(d1)+1に格納されるエラーコードについては，下記を参照してください。

📖 657ページ ファイル操作命令で発生するエラーコード

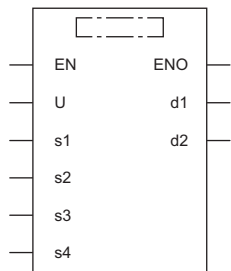
指定ファイルのコピー

SP.FCOPY

SDメモリカード内の指定したファイルまたはフォルダをコピーします。フォルダ指定の場合は、フォルダごと、または指定フォルダ内のすべてのファイル、サブフォルダをコピーします。

ラダー	ST
	ENO:=SP_FCOPY(EN,U,s1,s2,s3,s4,d1,d2);

FBD/LD



■実行条件

命令	実行条件
SP.FCOPY	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(U)	ダミー	—	デバイス名	ANY16
(d1)	コントロールデータが格納されている先頭デバイス	639ページ コントロールデータ(d1)	ワード	ANY16_ARRAY (要素数: 2)
(s1)	コピー元ドライブ指定	2(固定)*1	ワード	ANY16
(s2)	コピー元ファイル名/フォルダ名が格納されている先頭デバイス	—	Unicode文字列	ANYSTRING_DOUBLE
(s3)	コピー先ドライブ指定	2(固定)*1	ワード	ANY16
(s4)	コピー先フォルダパスを格納する先頭デバイス	—	Unicode文字列	ANYSTRING_DOUBLE
(d2)	処理完了にてONするビットデバイス	—	ビット	ANYBIT_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

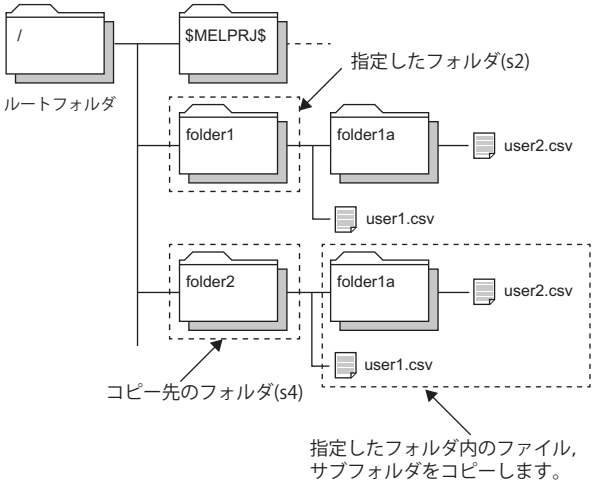
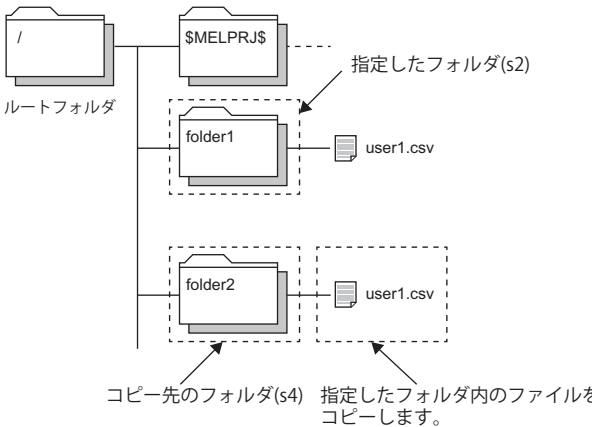
*1 SDメモリカードのドライブ2のみ設定可能です。

■使用可能デバイス

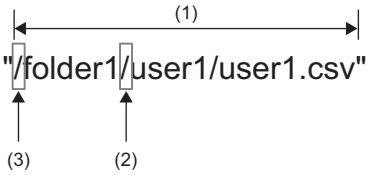
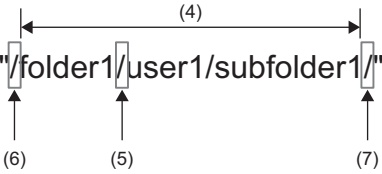
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(U)	—	—	—	—	—	—	—	—	—	—	—	○
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s1)	○	—	○	—	—	—	—	○	○	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	○	—
(s3)	○	—	○	—	—	—	—	○	○	—	—	—
(s4)	—	—	○	—	—	—	—	○	—	—	○	—
(d2)	○	—	○	—	—	—	—	○	—	—	—	—

■コントロールデータ(d1)

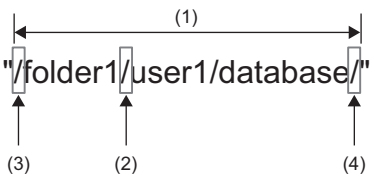
オペランド: (d1)				
デバイス	項目	内容	設定範囲	セット側
+0	使用用途設定 エリア	b15 ... b2 b1 b0 0 1/0 1/0 1/0 ■b0, b2: 対象種別設定 b0: 指定対象がファイルかフォルダかを指定します。 0: ファイル指定 1: フォルダ指定 b2: コピー対象を指定します。(b0が1(フォルダ指定)の場合に有効となります。) 0: 指定フォルダごとコピーする 1: 指定フォルダ内のすべてのファイル、サブフォルダをコピーする ・b2が0(指定フォルダごとコピーする)の場合の動作 指定したフォルダをフォルダごと(ファイル、サブフォルダも含む)コピーします。 指定したフォルダ(s2) コピー先のフォルダ(s4) 指定したフォルダのフォルダ構成のまま フォルダごとコピーします。	左記	ユーザ

オペランド: (d1)				
デバイス	項目	内容	設定範囲	セット側
+0	使用用途設定 エリア	・ b2が1(指定フォルダ内のすべてのファイル、サブフォルダをコピーする)の場合の動作(指定したフォルダにサブフォルダが存在する場合) 指定したフォルダ内のファイル、サブフォルダをコピーします。  指定したフォルダ(s2) 指定したフォルダ内のファイル、サブフォルダをコピーします。 コピー先のフォルダ(s4) ・ b2が1(指定フォルダ内のすべてのファイル、サブフォルダをコピーする)の場合の動作(指定したフォルダにサブフォルダが存在しない(ファイルのみ)場合) 指定したフォルダ内のファイルをコピーします。  指定したフォルダ(s2) 指定したフォルダ内のファイルをコピーします。 コピー先のフォルダ(s4) ■b1: 上書き設定 コピー先にコピー元と同名のファイルやフォルダが存在する場合の動作を指定します。 ・ 0: 上書きしない ・ 1: 上書きする b0を1(フォルダ指定)および、b1を0(上書きしない)と指定した場合、コピー先と同名のファイルやフォルダは、コピーをスキップします。(異常完了しません。)	左記	ユーザ
+1	完了ステータス	命令完了時に完了ステータスを格納します。 ・ 0000H: 正常完了 ・ 0000H以外: 異常完了(エラーコード)( 657ページ ファイル操作命令で発生するエラーコード)	—	システム

■コピー元ファイル名/フォルダ名(s2)

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+0~+□	ファイル名/フォルダ名	ファイルを指定する場合、コピー元のファイルが格納されているフォルダパス+ファイル名を指定します。 フォルダを指定する場合、コピー元のフォルダパスを指定します。 - ファイル名に拡張子が存在する場合、省略しないでください。 - ファイル名はピリオドと拡張子を含む64文字以内にしてください。 - フォルダパス+ファイル名(ピリオド, 拡張子を含む)は253文字以内にしてください。 - フォルダパスは244文字以内にしてください。(ファイルの区切り文字は含まない。) - ファイル名, もしくはフォルダパスは区切り文字を除いて1文字以上指定してください。 - ファイル指定時, 文字列の末尾に区切り文字を指定しないでください。 - 文字列の末尾, または各区切り文字の直前に半角ピリオドを指定しないでください。 - フォルダパスの階層数は10以内にしてください。 ■ファイルを指定する場合  (1): 最大253文字 (2): フォルダパス, ファイルの区切り文字は"/"または"¥"を使用する。 (3): 省略可能。省略した場合, (1)の最大は252文字となる。 ■フォルダを指定する場合  (4): 最大244文字 (5): フォルダパスの区切り文字は"/"または"¥"を使用する。 (6): 省略可能。省略した場合, (4)の最大は243文字となる。 (7): 省略可能	Unicode文字列	ユーザ

■コピー先フォルダパス(s4)

オペランド: (s4)				
デバイス	項目	内容	設定範囲	セット側
+0~+□	フォルダパス	コピー先のフォルダパスを指定します。 - フォルダパスは244文字以内にしてください。(ファイルの区切り文字は含まない。) - フォルダパスは区切り文字を除いて1文字以上指定してください。 - 文字列の末尾, または各区切り文字の直前に半角スペースを指定しないでください。 - 文字列の末尾, または各区切り文字の直前に半角ピリオドを指定しないでください。 - フォルダパスの階層数は10以内にしてください。  (1): 最大244文字 (2): フォルダパスの区切り文字は"/"または"¥"を使用する。 (3): 省略可能。省略した場合, (1)の最大は243文字となる。 (4): 省略可能	Unicode文字列	ユーザ

機能

- ・(s1)で指定したドライブの(s2)で指定したファイルまたはフォルダを、(s3)で指定したドライブの(s4)で指定したフォルダにコピーします。(s2)にフォルダを指定したとき、コピー先に同名のファイルおよびサブフォルダが存在する場合、(d1)のビット1の値(上書き設定)が0でも異常完了となりません。(コピーをスキップします。)
- ・(s4)で指定したコピー先のフォルダがない場合、フォルダは自動で作成されます。
- ・SP.FCOPY命令実行中はSM753(ファイルアクセス中)がONします。SM753がON中は、SP.FCOPY命令は実行できません。(実行した場合は無処理となります。)
- ・処理完了(d2)のビットデバイスは、SP.FCOPY命令の処理完了を検出したスキャンのEND命令実行時に自動的にONし、次のスキャンのEND命令にてOFFします。
- ・SP.FCOPY命令が異常完了時には、異常完了(d2)+1のデバイスが、処理完了(d2)のデバイスと同じタイミングでON/OFFします。
- ・命令実行時に演算エラーを検出した場合、処理完了(d2)、異常完了(d2)+1はONしません。

注意事項

- ・SP.FCOPY命令は、割込みプログラム中では実行しないでください。割込みプログラムで実行した場合、誤動作の原因となります。
- ・命令実行中にコントローラの状態をRUN→STOPに変更しても、本命令は処理を継続します。
- ・コピー後のフォルダパス+ファイル名およびフォルダパスが文字数制限を超えないようにしてください。(ファイルがアクセス不可、またはエラーとなる場合があります。)
- ・命令が処理の途中で異常完了した場合、ファイル/フォルダが処理中の状態が残ることがあります。
- ・コピー対象ファイルのファイルサイズやファイル数が大きくなる場合、命令が完了するまでの時間が長くなります。
- ・SP.FCOPY命令で操作中のファイルには、他機能からアクセスしないでください。(ファイルが破損、またはエラーとなる場合があります。)
- ・他機能からアクセスしているファイル/フォルダへの操作は行わないでください。

エラー

エラーコード (SD0)	内容
3285H	ドライブ指定(s1)、(s3)によって指定されたドライブが、SDメモ리카ード以外のとき。 (s2)で指定したファイル名/フォルダ名文字列が読み出せないとき。 ・ファイル名文字列が1文字以上指定されていない。 ・指定したファイル名文字列に254文字以上を指定した。 ・指定したフォルダパスに245文字以上を指定した。 ・指定したフォルダパスに11以上の階層を指定した。 ・ファイル指定時、指定したファイル名文字列の末尾に区切り文字を指定した。 ・指定したファイル名文字列の末尾、または各区切り文字の直前に半角ピリオドを指定した。 (s4)で指定したフォルダパス文字列が読み出せないとき。 ・フォルダパス文字列が1文字以上指定されていない。 ・指定したフォルダパス文字列に245文字以上を指定した。 ・指定したフォルダパス文字列に11以上の階層を指定した。 ・指定したフォルダパス文字列の末尾、または各区切り文字の直前に半角スペースを指定した。 ・指定したフォルダパス文字列の末尾、または各区切り文字の直前に半角ピリオドを指定した。 (s2)、(s4)で指定したフォルダパス内にルートフォルダ直下のシステムフォルダ(\$MELPRJ\$)を指定したとき。

SP.FCOPY命令が異常完了した場合、完了ステータス(d1)+1で指定したデバイスにエラーコードが格納されます。(命令が演算エラーになった場合は格納されません。)

完了ステータス(d1)+1に格納されるエラーコードについては、下記を参照してください。

📖 657ページ ファイル操作命令で発生するエラーコード

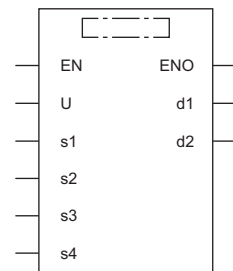
指定ファイルの移動

SP.FMOVE

SDメモリカード内の指定したファイルまたはフォルダを移動します。フォルダ指定の場合は、フォルダごと、または指定フォルダ内のすべてのファイル、サブフォルダを移動します。

ラダー	ST
	ENO:=SP_FMOVE(EN,U,s1,s2,s3,s4,d1,d2);

FBD/LD



■実行条件

命令	実行条件
SP.FMOVE	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(U)	ダミー	—	デバイス名	ANY16
(d1)	コントロールデータが格納されている先頭デバイス	644ページ コントロールデータ(d1)	ワード	ANY16_ARRAY (要素数: 2)
(s1)	移動元ドライブ指定	2(固定)* ¹	ワード	ANY16
(s2)	移動元ファイル名/フォルダ名が格納されている先頭デバイス	—	Unicode文字列	ANYSTRING_DOUBLE
(s3)	移動先ドライブ指定	2(固定)* ¹	ワード	ANY16
(s4)	移動先フォルダパスを格納するデバイス	—	Unicode文字列	ANYSTRING_DOUBLE
(d2)	処理完了にてONするビットデバイス	—	ビット	ANYBIT_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 SDメモリカードのドライブ2のみ設定可能です。

■使用可能デバイス

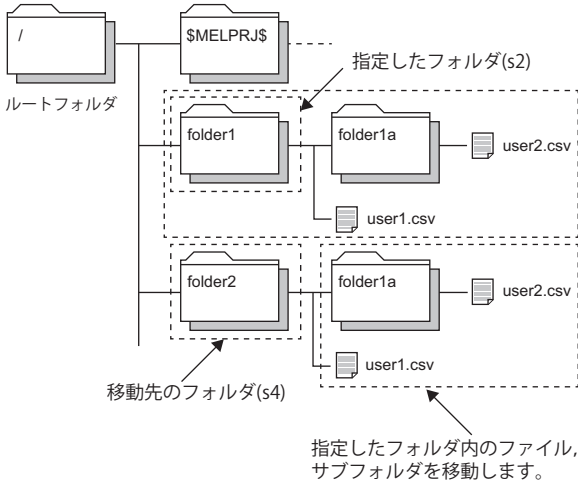
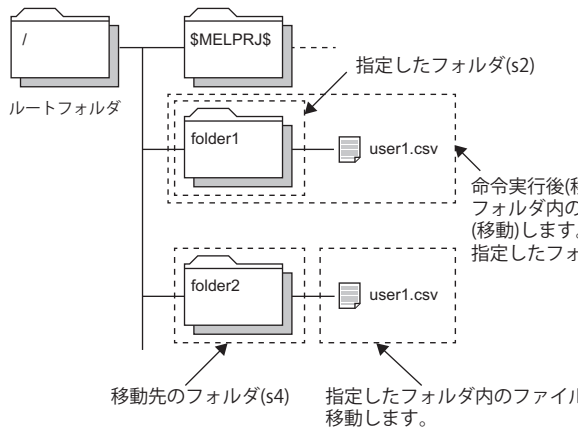
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(U)	—	—	—	—	—	—	—	—	—	—	—	○
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s1)	○	—	○	—	—	—	—	○	○	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	○	—
(s3)	○	—	○	—	—	—	—	○	○	—	—	—
(s4)	—	—	○	—	—	—	—	○	—	—	○	—
(d2)	○	—	○	—	—	—	—	○	—	—	—	—

■コントロールデータ(d1)

オペランド: (d1)				
デバイス	項目	内容	設定範囲	セット側
+0	使用用途設定 エリア	b15 ... b2 b1 b0 0 1/0 1/0 1/0 ■b0, b2: 対象種別設定 b0: 指定対象がファイルかフォルダかを指定します。 0: ファイル指定 1: フォルダ指定 b2: 移動対象を指定します。(b0が1(フォルダ指定)の場合に有効となります。) 0: 指定フォルダごと移動する 1: 指定フォルダ内のすべてのファイル、サブフォルダを移動する ・b2が0(指定フォルダごと移動する)の場合の動作 指定したフォルダをフォルダごと(ファイル、サブフォルダも含む)移動します。 / \$MELPRJS ルートフォルダ folder1 folder1a user1.csv user2.csv folder2 folder1 folder1a user1.csv user2.csv 指定したフォルダ(s2) 移動先のフォルダ(s4) 命令実行後(移動後)は指定したフォルダごと削除(移動)します。 指定したフォルダのフォルダ構成のままフォルダごと移動します。	左記	ユーザ

左記

ユーザ

オペランド:(d1)				
デバイス	項目	内容	設定範囲	セット側
+0	使用用途設定 エリア	・b2が1(指定フォルダ内のすべてのファイル、サブフォルダを移動する)の場合の動作(指定したフォルダにサブフォルダが存在する場合) 指定したフォルダ内のファイル、サブフォルダを移動します。  指定したフォルダ(s2) 移動先のフォルダ(s4) 指定したフォルダ内のファイル、サブフォルダを移動します。 命令実行後(移動後)は指定したフォルダ内のファイル、サブフォルダは削除(移動)します。指定したフォルダは削除します。 ・b2が1(指定フォルダ内のすべてのファイル、サブフォルダを移動する)の場合の動作(指定したフォルダにサブフォルダが存在しない(ファイルのみ)場合) 指定したフォルダ内のファイルを移動します。  指定したフォルダ(s2) 移動先のフォルダ(s4) 指定したフォルダ内のファイルを移動します。 命令実行後(移動後)は指定したフォルダ内のファイルは削除(移動)します。指定したフォルダは削除します。 ■b1: 上書き設定 移動先に移動元と同名のファイルやフォルダが存在する場合の動作を指定します。 ・0: 上書きしない ・1: 上書きする	左記	ユーザ
+1	完了ステータス	命令完了時に完了ステータスを格納します。 ・0000H: 正常完了 ・0000H以外: 異常完了(エラーコード)( 657ページ ファイル操作命令で発生するエラーコード)	—	システム

■ファイル名/フォルダ名(s2)

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+0~+□	ファイル名/フォルダ名	ファイルを指定する場合、移動元のファイルが格納されているフォルダパス+ファイル名を指定します。 フォルダを指定する場合、移動元のフォルダのフォルダパスを指定します。 - ファイル名に拡張子が存在する場合、省略しないでください。 - ファイル名はピリオドと拡張子を含む64文字以内にしてください。 - フォルダパス+ファイル名(ピリオド, 拡張子を含む)は253文字以内にしてください。 - フォルダパスは244文字以内にしてください。(ファイルの区切り文字は含まない。) - ファイル名, もしくはフォルダパスは区切り文字を除いて1文字以上指定してください。 - ファイル指定時, 文字列の末尾に区切り文字を指定しないでください。 - 文字列の末尾, または各区切り文字の直前に半角ピリオドを指定しないでください。 - フォルダパスの階層数は10以内にしてください。 ■ファイルを指定する場合 (1): 最大253文字 (2): フォルダパス, ファイルの区切り文字は"/"または"¥"を使用する。 (3): 省略可能。省略した場合, (1)の最大は252文字となる。 ■フォルダを指定する場合 (4): 最大244文字 (5): フォルダパスの区切り文字は"/"または"¥"を使用する。 (6): 省略可能。省略した場合, (4)の最大は243文字となる。 (7): 省略可能	Unicode文字列	ユーザ

■移動先フォルダパス(s4)

オペランド: (s4)				
デバイス	項目	内容	設定範囲	セット側
+0~+□	フォルダパス	移動先のフォルダパスを指定します。 - フォルダパスは244文字以内にしてください。(ファイルの区切り文字は含まない。) - フォルダパスは区切り文字を除いて1文字以上指定してください。 - 文字列の末尾, または各区切り文字の直前に半角スペースを指定しないでください。 - 文字列の末尾, または各区切り文字の直前に半角ピリオドを指定しないでください。 - フォルダパスの階層数は10以内にしてください。 (1): 最大244文字 (2): フォルダパス, ファイルの区切り文字は"/"または"¥"を使用する。 (3): 省略可能。省略した場合, (1)の最大は243文字となる。 (4): 省略可能	Unicode文字列	ユーザ

機能

- (s1)で指定したドライブの(s2)で指定したファイルまたはフォルダを、(s3)で指定したドライブの(s4)で指定したフォルダに移動します。(s4)で指定した移動先のフォルダがない場合、フォルダは自動で作成されます。
- SP.FMOVE命令実行中はSM753(ファイルアクセス中)がONします。SM753がON中は、SP.FMOVE命令は実行できません。(実行した場合は無処理となります。)
- 処理完了(d2)のビットデバイスは、SP.FMOVE命令の処理完了を検出したスキンのEND命令実行時に自動的にONし、次のスキンのEND命令にてOFFする。
- SP.FMOVE命令が異常完了時には、異常完了(d2)+1のデバイスが、処理完了(d2)のデバイスと同じタイミングでON/OFFします。
- 命令実行時に演算エラーを検出した場合、処理完了(d2)、異常完了(d2)+1はONしません。

注意事項

- SP.FMOVE命令は、割込みプログラム中では実行しないでください。割込みプログラムで実行した場合、誤動作の原因となります。
- 命令実行中にコントローラの状態をRUN→STOPに変更しても、本命令は処理を継続します。
- 移動後のフォルダパス+ファイル名およびフォルダパスが文字数の制限を超えないようにしてください。(ファイルが破損、またはエラーとなる場合があります。)
- 上書きが発生する場合、命令が完了するまでの時間が延びる場合があります。
- 上書きが発生する場合、操作対象ファイルと同等サイズの空き容量を確保してください。
- 命令が処理の途中で異常完了した場合、ファイル/フォルダが処理途中の状態が残ることがあります。
- 上書きが発生し、移動対象ファイルのファイルサイズやファイル数が大きくなる場合、命令が完了するまでの時間が長くなります。
- SP.FMOVE命令で操作中のファイルには、他機能からアクセスしないでください。(ファイルがアクセス不可、またはエラーとなる場合があります。)
- 他機能からアクセスしているファイル/フォルダへの操作は行わないでください。

エラー

エラーコード (SD0)	内容
3285H	ドライブ指定(s1)によって指定されたドライブが、SDメモリカード以外るとき。 (s2)で指定したファイル名/フォルダ名文字列が読み出せないとき。 • ファイル名文字列が1文字以上指定されていない。 • 指定したファイル名文字列に254文字以上を指定した。 • 指定したフォルダパス245文字以上を指定した。 • 指定したフォルダパスに11以上の階層を指定した。 • ファイル指定時、指定したファイル名文字列の末尾に区切り文字を指定した。 • 指定したファイル名文字列の末尾、または各区切り文字の直前に半角ピリオドを指定した。 (s4)で指定したフォルダパス文字列が読み出せないとき。 • フォルダパス文字列が1文字以上指定されていない。 • 指定したフォルダパス文字列に245文字以上を指定した。 • 指定したフォルダパス文字列に11以上の階層を指定した。 • 指定したフォルダパス文字列の末尾、または各区切り文字の直前に半角スペースを指定した。 • 指定したフォルダパス文字列の末尾、または各区切り文字の直前に半角ピリオドを指定した。 (s2)、(s4)で指定したフォルダパス内にルートフォルダ直下のシステムフォルダ(\$MELPRJ\$)を指定したとき。

SP.FMOVE命令が異常完了した場合、完了ステータス(d1)+1で指定したデバイスにエラーコードが格納されます。(命令が演算エラーになった場合は格納されません。)

完了ステータス(d1)+1に格納されるエラーコードについては、下記を参照してください。

📖 657ページ ファイル操作命令で発生するエラーコード

指定ファイル名の変更

SP.FRENAME

SDメモリカード内の指定したファイル名、またはフォルダ名を変更します。

ラダー	ST
	ENO:=SP_FRENAME(EN,U,s1,s2,s3,d1,d2);

FBD/LD

■実行条件

命令	実行条件
SP.FRENAME	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(U)	ダミー	—	デバイス名	ANY16
(s1)	ドライブ指定	2(固定)*1	ワード	ANY16
(d1)	コントロールデータが格納されている先頭デバイス	649ページ コントロールデータ(d1)	ワード	ANY16_ARRAY (要素数: 2)
(s2)	変更するファイル名/フォルダ名が格納されている先頭デバイス	—	Unicode文字列	ANYSTRING_DOUBLE
(s3)	変更後のファイル名/フォルダ名が格納される先頭デバイス	—	Unicode文字列	ANYSTRING_DOUBLE
(d2)	処理完了にてONするビットデバイス	—	ビット	ANYBIT_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 SDメモリカードのドライブ2のみ設定可能です。

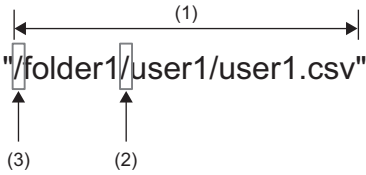
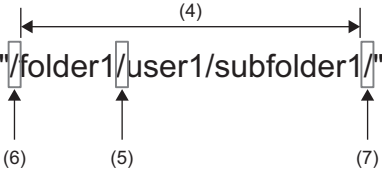
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(U)	—	—	—	—	—	—	—	—	—	—	—	○
(s1)	○	—	○	—	—	—	—	○	○	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	○	—
(s3)	—	—	○	—	—	—	—	○	—	—	○	—
(d2)	○	—	○	—	—	—	—	○	—	—	—	—

■コントロールデータ(d1)

オペランド: (d1)				
デバイス	項目	内容	設定範囲	セット側
+0	使用用途設定エリア	b15 ... b0 0 1/0 b0: 対象種別設定 指定対象がファイルかフォルダかを指定します。 • 0: ファイル指定 • 1: フォルダ指定	左記	ユーザ
+1	完了ステータス	命令完了時に完了ステータスを格納します。 • 0000H: 正常完了 • 0000H以外: 異常完了(エラーコード)( 657ページ ファイル操作命令で発生するエラーコード)	—	システム

■変更するファイル名/フォルダ名(s2)

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+0~+□	ファイル名/フォルダ名	ファイルを指定する場合、名称変更するファイルが格納されているフォルダパス+ファイル名を指定します。 フォルダを指定する場合、名称変更するフォルダのフォルダパスを指定します。 - ファイル名に拡張子が存在する場合、省略しないでください。 - ファイル名はピリオドと拡張子を含む64文字以内にしてください。 - フォルダパス+ファイル名(ピリオド、拡張子を含む)は253文字以内にしてください。 - フォルダパスは244文字以内にしてください。(ファイルの区切り文字は含まない。) - ファイル名、もしくはフォルダパスは区切り文字を除いて1文字以上指定してください。 - ファイル指定時、文字列の末尾に区切り文字を指定しないでください。 - 文字列の末尾、または各区切り文字の直前に半角ピリオドを指定しないでください。 - フォルダパスの階層数は10以内にしてください。 ■ファイルを指定する場合  (1): 最大253文字 (2): フォルダパス、ファイルの区切り文字は"/"または"¥"を使用する。 (3): 省略可能。省略した場合、(1)の最大は252文字となる。 ■フォルダを指定する場合  (4): 最大244文字 (5): フォルダパスの区切り文字は"/"または"¥"を使用する。 (6): 省略可能。省略した場合、(4)の最大は243文字となる。 (7): 省略可能	Unicode文字列	ユーザ

■変更後のファイル名/フォルダ名(s3)

オペランド: (s3)				
デバイス	項目	内容	設定範囲	セット側
+0~+□	ファイル名/フォルダ名	変更後のファイル名、フォルダ名の文字列を指定します。 - フォルダパスは指定しないでください。 - ファイル名(ピリオド、拡張子を含む)は64文字以内にしてください。 - フォルダ名は243文字以内にしてください。 - ファイル名、もしくはフォルダ名は1文字以上指定してください。 - 区切り文字は指定しないでください。 - 文字列の末尾に半角ピリオドを指定しないでください。	Unicode文字列	ユーザ

機能

- ・(s1)で指定したドライブの(s2)で指定したファイル名、またはフォルダ名を(s3)で指定したファイル名、またはフォルダ名に変更します。(s3)で指定したファイル名、またはフォルダ名が既に存在している場合、異常完了します。
- ・SP.FRENAME命令実行中はSM753(ファイルアクセス中)がONします。SM753がON中は、SP.FRENAME命令は実行できません。(実行した場合は無処理となります。)
- ・処理完了(d2)のビットデバイスは、SP.FRENAME命令の処理完了を検出したスキンのEND命令実行時に自動的にONし、次のスキンのEND命令にてOFFします。
- ・SP.FRENAME命令が異常完了時には、異常完了(d2)+1のデバイスが、処理完了(d2)のデバイスと同じタイミングでON/OFFします。
- ・命令実行時に演算エラーを検出した場合、処理完了(d2)、異常完了(d2)+1はONしません。

注意事項

- SP.FRENAME命令は、割込みプログラム中では実行しないでください。割込みプログラムで実行した場合、誤動作の原因となります。
- 命令実行中にコントローラの状態をRUN→STOPに変更しても、本命令は処理を継続します。
- 変更後のフォルダパス+ファイル名およびフォルダパスが文字数の制限を超えないようにしてください。(ファイルがアクセス不可、またはエラーとなる場合があります。)
- SP.FRENAME命令で操作中のファイルには、他機能からアクセスしないでください。(ファイルが破損、またはエラーとなる場合があります。)
- 他機能からアクセスしているファイル/フォルダへの操作は行わないでください。

エラー

エラーコード (SD0)	内容
3285H	ドライブ指定(s1)によって指定されたドライブが、SDメモリカード以外るとき。 (s2)で指定したファイル名/フォルダ名文字列が読み出せないとき。 • ファイル名文字列が1文字以上指定されていない。 • 指定したファイル名文字列に254文字以上を指定した。 • 指定したフォルダパス245文字以上を指定した。 • 指定したフォルダパスに11以上の階層を指定した。 • ファイル指定時、指定したファイル名文字列の末尾に区切り文字を指定した。 • 指定したファイル名文字列の末尾、または各区切り文字の直前に半角ピリオドを指定した。 (s3)で指定したファイル名/フォルダ名文字列が読み出せないとき。 • ファイル名/フォルダ名文字列が1文字以上指定されていない。 • 指定したファイル名文字列に253文字以上を指定した。 • 指定したフォルダ名文字列に244文字以上を指定した。 • 指定したファイル名文字列に区切り文字を指定した。 • 指定したファイル名文字列の末尾に半角ピリオドを指定した。 (s2)で指定したフォルダパス内にルートフォルダ直下のシステムフォルダ(\$MELPRJ\$)を指定したとき。 (s3)で指定したファイル名/フォルダ名に\$MELPRJ\$を指定したとき。

SP.FRENAME命令が異常完了した場合、完了ステータス(d1)+1で指定したデバイスにエラーコードが格納されます。(命令が演算エラーになった場合は格納されません。)

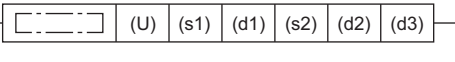
完了ステータス(d1)+1に格納されるエラーコードについては、下記を参照してください。

📖 657ページ ファイル操作命令で発生するエラーコード

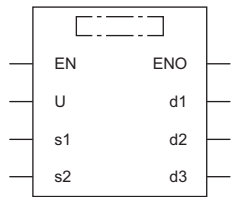
指定ファイルの状態取得

SP.FSTATUS

SDメモリカード内の指定したファイル，またはフォルダの状態を取得します。

ラダー	ST
	ENO:=SP_FSTATUS(EN,U,s1,s2,d1,d2,d3);

FBD/LD



■実行条件

命令	実行条件
SP.FSTATUS	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(U)	ダミー	—	デバイス名	ANY16
(s1)	ドライブ指定	2(固定)*1	ワード	ANY16
(d1)	コントロールデータが格納されている先頭デバイス	 653ページ コントロールデータ(d1)	ワード	ANY16_ARRAY (要素数: 2)
(s2)	ファイル名/フォルダ名が格納されている先頭デバイス	—	Unicode文字列	ANYSTRING_DOUBLE
(d2)	ファイル状態を格納する先頭デバイス	—	ワード	ANY16_ARRAY (要素数: 10)
(d3)	処理完了にてONするビットデバイス	—	ビット	ANYBIT_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 SDメモリカードのドライブ2のみ設定可能です。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(U)	—	—	—	—	—	—	—	—	—	—	—	○
(s1)	○	—	○	—	—	—	—	○	○	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	○	—
(d2)	—	—	○	—	—	—	—	○	—	—	—	—
(d3)	○	—	○	—	—	—	—	○	—	—	—	—

■コントロールデータ(d1)

オペランド: (d1)				
デバイス	項目	内容	設定範囲	セット側
+0	使用用途設定エリア	b15 ... b0 0 1/0 b0: 対象種別設定 指定対象がファイルかフォルダかを指定します。 • 0: ファイル指定 • 1: フォルダ指定	左記	ユーザ
+1	完了ステータス	命令完了時に完了ステータスを格納します。 • 0000H: 正常完了 • 0000H以外: 異常完了(エラーコード)( 657ページ ファイル操作命令で発生するエラーコード)	—	システム

■ファイル名/フォルダ名(s2)

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+0~+□	ファイル名/フォルダ名	ファイルを指定する場合、状態を取得するファイルが格納されているフォルダパス+ファイル名を指定します。 フォルダを指定する場合、状態を取得するフォルダのフォルダパスを指定します。 ・ファイル名に拡張子が存在する場合、省略しないでください。 ・ファイル名はピリオドと拡張子を含む64文字以内にしてください。 ・フォルダパス+ファイル名(ピリオド、拡張子を含む)は253文字以内にしてください。 ・フォルダパスは244文字以内にしてください。(ファイルの区切り文字は含まない。) ・ファイル名、もしくはフォルダパスは区切り文字を除いて1文字以上指定してください。 ・ファイル指定時、文字列の末尾に区切り文字を指定しないでください。 ・文字列の末尾、または各区切り文字の直前に半角ピリオドを指定しないでください。 ・フォルダパスの階層数は10以内にしてください。 ■ファイルを指定する場合 (1) "/folder1/user1/user1.csv" (3)(2) (1): 最大253文字 (2): フォルダパス, ファイルの区切り文字は"/"または"¥"を使用する。 (3): 省略可能。省略した場合, (1)の最大は252文字となる。 ■フォルダを指定する場合 (4) "/folder1/user1/subfolder1/" (6)(5)(7) (4): 最大244文字 (5): フォルダパスの区切り文字は"/"または"¥"を使用する。 (6): 省略可能。省略した場合, (4)の最大は243文字となる。 (7): 省略可能	Unicode文字列	ユーザ

■ファイル状態(d2)

オペランド: (d2)			
デバイス	項目	範囲	セット側
+0	ファイル属性 ■(s2)でファイルを指定時 bit0: 読み取り専用ファイルの場合ONする bit1~bit4: リザーブ(0固定) bit5: リザーブ(1固定) bit6~15: リザーブ(0固定) ■(s2)でフォルダを指定時 bit0~3: リザーブ(0固定) bit4: フォルダの場合ONする bit5~15: リザーブ(0固定)	左記	システム
+1	リザーブ	0	
+2~+3	■(s2)でファイルを指定時 ファイルサイズ(バイト単位) ■(s2)でフォルダを指定時 0固定	0~4294967294*1	
+4	最終更新日時: 年	0, 1980~2079*1	
+5	最終更新日時: 月	0~12	
+6	最終更新日時: 日	0~31	
+7	最終更新日時: 時	0~23	
+8	最終更新日時: 分	0~59	
+9	最終更新日時: 秒		

*1 コントローラ以外の環境(OS)でアクセスされた場合、取得する値の範囲はその環境(OS)に依存します。

機能

- (s1)で指定したドライブの(s2)で指定したファイル、またはフォルダの状態を取得し、(d2)以降のデバイスに格納します。(s2)にフォルダを指定した場合、(d2)+2~(d2)+3には0を格納します。
- SP.FSTATUS命令実行中はSM753(ファイルアクセス中)がONします。SM753がON中は、SP.FSTATUS命令は実行できません。(実行した場合は無処理となります。)
- 処理完了(d3)のビットデバイスは、SP.FSTATUS命令の処理完了を検出したスキンのEND命令実行時に自動的にONし、次のスキンのEND命令にてOFFします。
- SP.FSTATUS命令が異常完了時には、異常完了(d3)+1のデバイスが、処理完了(d3)のデバイスと同じタイミングでON/OFFします。
- 命令実行時に演算エラーを検出した場合、処理完了(d3)、異常完了(d3)+1はONしません。

注意事項

- SP.FSTATUS命令は、割込みプログラム中では実行しないでください。割込みプログラムで実行した場合、誤動作の原因となります。
- 命令実行中にコントローラの状態をRUN→STOPに変更しても、本命令は処理を継続します。

エラー

エラーコード (SD0)	内容
3285H	ドライブ指定(s1)によって指定されたドライブが、SDメモ리카ード以外のとき。 (s2)で指定したファイル名/フォルダ名文字列が読み出せないとき。 • ファイル名文字列が1文字以上指定されていない。 • 指定したファイル名文字列に254文字以上を指定した。 • 指定したフォルダパスに245文字以上を指定した。 • 指定したフォルダパスに11以上の階層を指定した。 • ファイル指定時、指定したファイル名文字列の末尾に区切り文字を指定した。 • 指定したファイル名文字列の末尾、または各区切り文字の直前に半角ピリオドを指定した。

SP.FSTATUS命令が異常完了した場合、完了ステータス(d1)+1で指定したデバイスにエラーコードが格納されます。(命令が演算エラーになった場合は格納されません。)

完了ステータス(d1)+1に格納されるエラーコードについては、下記を参照してください。

📖 657ページ ファイル操作命令で発生するエラーコード

ファイル操作命令で発生するエラーコード

ファイル操作命令の完了ステータスに格納されるエラーコードの一覧を示します。

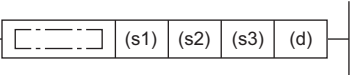
エラーコード	エラー内容	処置方法
8000H	SM606(SDメモリカード強制使用停止指示)がONしている。	SM606がONの場合、SM606をOFFし、SDメモリカード強制使用停止解除を行ってください。
	SDメモリカードが装着されていない。	SDメモリカードを装着してください。
	SDメモリカードがマウントされていない。	SDメモリカードをマウントしてください。
8001H	SDメモリカードへのアクセスに失敗した。	- ファイル名文字列の指定に誤りがないか確認してください。 - 他機能でファイルにアクセスしていないか確認してください。 - ノイズ対策を施してください。 - コントローラをリセット後、RUNしてください。再度同じエラーを表示した場合は、SDメモリカードのハードウェア異常の恐れがあります。最寄りの三菱電機システムサービス株式会社または当社の支社、代理店にご相談ください。
	SDメモリカードがプロテクトされている。	SDメモリカードのライトプロテクトスイッチを無効(書込み許可)としてください。
	ファイルが読取専用になっている。	読み取り専用の設定を解除してください。
	SDメモリカードに格納するファイルのサイズが、SDメモリカードの空き容量もしくは、最大容量を超えている。	- SDメモリカード内の不要なファイルを削除し、空き容量を確保してください。 - ファイルサイズが、最大容量以下となるように修正してください。
	ファイル名文字列に使用できない文字が含まれている。	ファイル名文字列の指定に誤りがないか確認してください。
8002H	指定したファイルまたはフォルダが存在しない。	- 指定したファイルまたはフォルダが存在するか、確認してください。 - 指定したフォルダパスが存在するか、確認してください。
	SDメモリカードの空き容量が不足している。	SDメモリカード内の不要なファイルを削除し、空き容量を確保してください。
8003H	ファイルから読み出したデータ数の合計が、(d1)+3(最大読出しデータ数)を超えた。	(d1)+2(要求読出しデータ数)または(d1)+3(最大読出しデータ数)を調整してください。
8004H	削除対象フォルダが空フォルダではない。	コントロールデータの設定を確認してください。
8005H	同名のファイルまたはフォルダが存在している。	ファイル名またはフォルダ名を変更してください。
8006H	- 命令実行後のフォルダパスが244文字を超えるフォルダが存在している。 - 命令実行後のファイルパスが253文字を超えるファイルが存在している。	命令実行後のフォルダパスまたはファイルパスが、最大文字数を超えないように設定してください。
	コピー先または移動先に、下記のフォルダ/サブフォルダを指定している。 - コピー元と同一のフォルダ/サブフォルダ - 移動元と同一のフォルダ/サブフォルダ	コピー先または移動先のパスを変更してください。

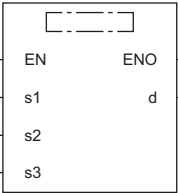
14.5 データ制御命令

BIN16ビットデータ上下限リミット制御

LIMIT(P)(_U)

指定した入力値(BIN16ビット値)が上下限リミット値の範囲内か否かにより、出力値を制御します。

ラダー	ST*1	
	ENO:=LIMITP(EN,s1,s2,s3,d);	ENO:=LIMITP_U(EN,s1,s2,s3,d);

FBD/LD*1


*1 LIMIT命令, LIMIT_U命令はST, FBD/LDでは対応していません。汎用ファンクションのLIMITを使用してください。
📖 1297ページ LIMIT(_E)

■実行条件

命令	実行条件
LIMIT LIMIT_U	
LIMITP LIMITP_U	

設定データ

■内容、範囲、データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(s1)	LIMIT(P)	下限リミット値(最小出力限界値)	-32768~32767	符号付きBIN16ビット	ANY16_S
	LIMIT(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
(s2)	LIMIT(P)	上限リミット値(最大出力限界値)	-32768~32767	符号付きBIN16ビット	ANY16_S
	LIMIT(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
(s3)	LIMIT(P)	上下限リミット制御により制御する入力値	-32768~32767	符号付きBIN16ビット	ANY16_S
	LIMIT(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
(d)	LIMIT(P)	上下限リミット制御により制御した出力値を格納するデバイス	—	符号付きBIN16ビット	ANY16_S
	LIMIT(P)_U			符号なしBIN16ビット	ANY16_U
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

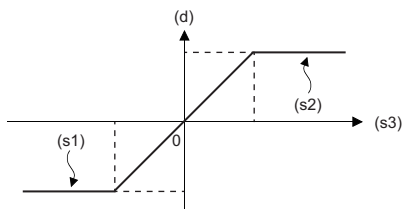
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(s3)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- ・(s3)で指定した入力値(BIN16ビット値)が、(s1)、(s2)で指定した上下限リミット値の範囲により、(d)で指定したデバイスに格納する出力値を制御します。出力値は下記のように制御されます。

条件	出力値に格納される値
下限リミット値(s1)>入力値(s3)のとき	下限リミット値(s1)
上限リミット値(s2)<入力値(s3)のとき	上限リミット値(s2)
下限リミット値(s1)≤入力値(s3)≤上限リミット値(s2)のとき	入力値(s3)



- ・上限リミット値のみによる制御を行う場合は、(s1)で指定する下限リミット値に設定データ範囲の最小値を設定します。
- ・下限リミット値のみの制御を行う場合は、(s2)で指定する上限リミット値に設定データ範囲の最大値を設定します。

エラー

エラーコード (SD0)	内容
3285H	(s1)で指定した下限リミット値が、(s2)で指定した上限リミット値よりも大きいとき。

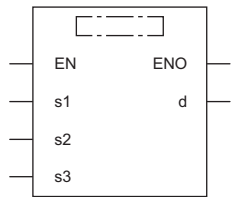
BIN32ビットデータ上下限リミット制御

DLIMIT(P)(_U)

指定した入力値(BIN32ビット値)が上下限リミット値の範囲内か否かにより、出力値を制御します。

ラダー	ST*1	
	ENO:=DLIMITP(EN,s1,s2,s3,d);	ENO:=DLIMITP_U(EN,s1,s2,s3,d);

FBD/LD*1



*1 DLIMIT命令, DLIMIT_U命令はST, FBD/LDでは対応していません。汎用ファンクションのLIMITを使用してください。
1297ページ LIMIT(_E)

■実行条件

命令	実行条件
DLIMIT DLIMIT_U	
DLIMITP DLIMITP_U	

設定データ

■内容、範囲、データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(s1)	DLIMIT(P)	下限リミット値(最小出力限界値)を格納するデバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	DLIMIT(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U
(s2)	DLIMIT(P)	上限リミット値(最大出力限界値)を格納するデバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	DLIMIT(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U
(s3)	DLIMIT(P)	上下限リミット制御により制御する入力値を格納するデバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	DLIMIT(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U
(d)	DLIMIT(P)	上下限リミット制御により制御した出力値を格納する先頭デバイス	—	符号付きBIN32ビット	ANY32_S
	DLIMIT(P)_U			符号なしBIN32ビット	ANY32_U
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

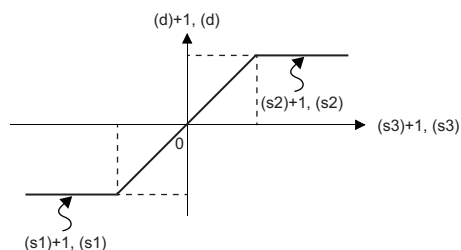
■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC		K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(s3)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

機能

- (s3)で指定した入力値(BIN32ビット値)が, (s1), (s2)で指定した上下限リミット値の範囲により, (d)で指定したデバイスに格納する出力値を制御します。出力値は下記のように制御されます。

条件	出力値に格納される値
下限リミット値((s1), (s1)+1)>入力値((s3), (s3)+1)のとき	下限リミット値((s1), (s1)+1)
上限リミット値((s2), (s2)+1)<入力値((s3), (s3)+1)のとき	上限リミット値((s2), (s2)+1)
下限リミット値((s1), (s1)+1)≤入力値((s3), (s3)+1)≤上限リミット値((s2), (s2)+1)のとき	入力値((s3), (s3)+1)



- 上限リミット値のみの制御を行う場合は, (s1)で指定する下限リミット値に設定データ範囲の最小値を設定します。
- 下限リミット値のみの制御を行う場合は, (s2)で指定する上限リミット値に設定データ範囲の最大値を設定します。

エラー

エラーコード (SD0)	内容
3285H	(s1)で指定した下限リミット値が, (s2)で指定した上限リミット値よりも大きいとき。

BIN16ビットデータ不感帯制御

BAND(P)(_U)

指定した入力値(BIN16ビット値)が不感帯の上下限範囲内か否かにより，出力値を制御します。

ラダー	ST	
	ENO:=BAND(EN,s1,s2,s3,d); ENO:=BANDP(EN,s1,s2,s3,d);	ENO:=BAND_U(EN,s1,s2,s3,d); ENO:=BANDP_U(EN,s1,s2,s3,d);

FBD/LD

■実行条件

命令	実行条件
BAND BAND_U	
BANDP BANDP_U	

設定データ

■内容，範囲，データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(s1)	BAND(P)	不感帯(無出力帯域)の下限値	-32768~32767	符号付きBIN16ビット	ANY16_S
	BAND(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
(s2)	BAND(P)	不感帯(無出力帯域)の上限値	-32768~32767	符号付きBIN16ビット	ANY16_S
	BAND(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
(s3)	BAND(P)	不感帯制御により制御する入力値	-32768~32767	符号付きBIN16ビット	ANY16_S
	BAND(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
(d)	BAND(P)	不感帯制御により制御した出力値を格納するデバイス	—	符号付きBIN16ビット	ANY16_S
	BAND(P)_U			符号なしBIN16ビット	ANY16_U
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

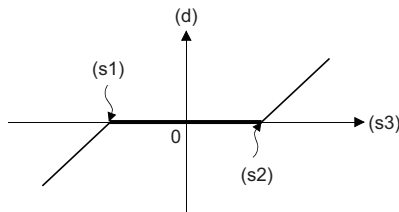
■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	—
(s3)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

機能

- (s3)で指定した入力値(BIN16ビット値)が、(s1)、(s2)で指定した不感帯の上下限範囲により、(d)で指定したデバイスに格納する出力値を制御します。出力値は、下記のように制御されます。

条件	出力値に格納される値
不感帯下限値(s1)>入力値(s3)のとき	入力値(s3)-不感帯下限値(s1)
不感帯上限値(s2)<入力値(s3)のとき	入力値(s3)-不感帯上限値(s2)
不感帯下限値(s1)≤入力値(s3)≤不感帯上限値(s2)のとき	0



- BAND(P)命令の場合、演算結果が-32768~32767を超えるときは、下記の例のようになります。

例

(s1)が10で、(s3)が-32768のとき、出力値=-32768-10=8000H-000AH=7FF6H=32758となります。

- BAND(P)_U命令の場合、演算結果が0~65535を超えるときは、下記の例のようになります。

例

(s1)が100で、(s3)が50のとき、出力値=50-100=0032H-0064H=FFCEH=65486となります。

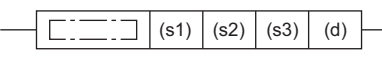
エラー

エラーコード (SD0)	内容
3285H	(s1)で指定した下限値が、(s2)で指定した上限値よりも大きいとき。

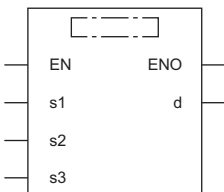
BIN32ビットデータ不感帯制御

DBAND(P)(_U)

指定した入力値(BIN32ビット値)が不感帯の上下限範囲内か否かにより、出力値を制御します。

ラダー	ST	
	ENO:=DBAND(EN,s1,s2,s3,d); ENO:=DBANDP(EN,s1,s2,s3,d);	ENO:=DBAND_U(EN,s1,s2,s3,d); ENO:=DBANDP_U(EN,s1,s2,s3,d);

FBD/LD



■実行条件

命令	実行条件
DBAND DBAND_U	
DBANDP DBANDP_U	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	DBAND(P)	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	DBAND(P)_U		符号なしBIN32ビット	ANY32_U
(s2)	DBAND(P)	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	DBAND(P)_U		符号なしBIN32ビット	ANY32_U
(s3)	DBAND(P)	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	DBAND(P)_U		符号なしBIN32ビット	ANY32_U
(d)	DBAND(P)	—	符号付きBIN32ビット	ANY32_S
	DBAND(P)_U		符号なしBIN32ビット	ANY32_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

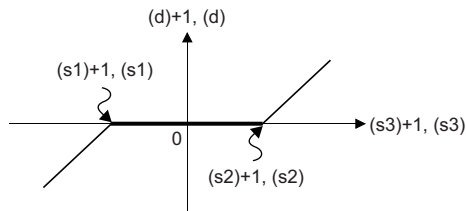
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	—	—	—
(s3)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (s3)で指定した入力値(BIN32ビット値)が、(s1)、(s2)で指定した不感帯の上下限範囲により、(d)で指定したデバイスに格納する出力値を制御します。出力値は、下記のように制御されます。

条件	出力値に格納される値
不感帯下限値((s1), (s1)+1)>入力値((s3), (s3)+1)のとき	入力値((s3), (s3)+1)-不感帯下限値((s1), (s1)+1)
不感帯上限値((s2), (s2)+1)<入力値((s3), (s3)+1)のとき	入力値((s3), (s3)+1)-不感帯上限値((s2), (s2)+1)
不感帯下限値((s1), (s1)+1)≤入力値((s3), (s3)+1)≤不感帯上限値((s2), (s2)+1)のとき	0



- DBAND(P)命令の場合、演算結果が-2147483648~2147483647を超えるときは、下記の例のようになります。

例

(s1), (s1)+1が1000で、(s3), (s3)+1が-2147483648のとき、出力値=-2147483648-1000=80000000H-000003E8H=7FFFFC18H=2147482648となります。

- DBAND(P)_U命令の場合、演算結果が0~4294967295を超えるときは、下記の例のようになります。

例

(s1), (s1)+1が100で、(s3), (s3)+1が50のとき、出力値=50-100=00000032H-00000064H=FFFFFFCEH=4294967246となります。

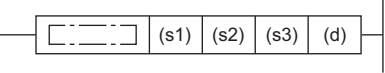
エラー

エラーコード (SD0)	内容
3285H	(s1)で指定した下限値が、(s2)で指定した上限値よりも大きいとき。

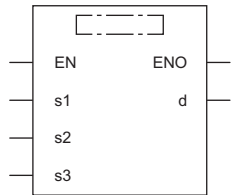
BIN16ビットデータゾーン制御

ZONE(P)(_U)

指定した入力値(BIN16ビット値)にバイアス値を付加します。

ラダー	ST	
	ENO:=ZONE(EN,s1,s2,s3,d); ENO:=ZONEP(EN,s1,s2,s3,d);	ENO:=ZONE_U(EN,s1,s2,s3,d); ENO:=ZONEP_U(EN,s1,s2,s3,d);

FBD/LD



■実行条件

命令	実行条件
ZONE ZONE_U	
ZONEP ZONEP_U	

設定データ

■内容、範囲、データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(s1)	ZONE(P)	入力値に加算する負のバイアス値	-32768~32767	符号付きBIN16ビット	ANY16_S
	ZONE(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
(s2)	ZONE(P)	入力値に加算する正のバイアス値	-32768~32767	符号付きBIN16ビット	ANY16_S
	ZONE(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
(s3)	ZONE(P)	ゾーン制御を行うための入力値	-32768~32767	符号付きBIN16ビット	ANY16_S
	ZONE(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
(d)	ZONE(P)	ゾーン制御により制御した出力値を格納するデ バイス	—	符号付きBIN16ビット	ANY16_S
	ZONE(P)_U			符号なしBIN16ビット	ANY16_U
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

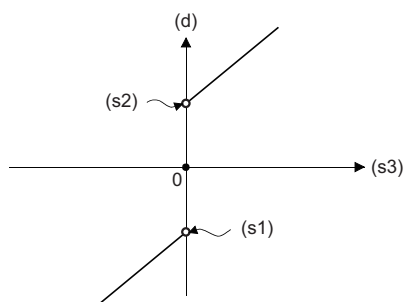
■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	—
(s3)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

機能

- (s3)で指定した入力値(BIN16ビット値)に、(s1)または(s2)で指定したバイアス値を付加して、(d)で指定したデバイス番号に格納します。バイアス値は下記のように制御されます。

条件	出力値に格納される値
入力値(s3)<0のとき	入力値(s3)+負のバイアス値(s1)
入力値(s3)=0のとき	0
入力値(s3)>0のとき	入力値(s3)+正のバイアス値(s2)



- ZONE(P)命令の場合、演算結果が-32768~32767を超えるときは、下記の例のようになります。

例

(s1)が-100で、(s3)が-32768のとき、出力値=-32768+(-100)=8000H-FF9CH=7F9CH=32668となります。

- ZONE(P)_U命令の場合、演算結果が0~65535を超えるときは、下記の例のようになります。

例

(s2)が100で、(s3)が65535のとき、出力値=65535+100=FFFFH-0064H=0063H=99となります。

- ZONE(P)_U命令の場合、(s1)はダミーデータ扱いとして使用しません。

エラー

演算エラーはありません。

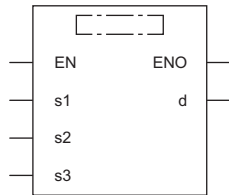
BIN32ビットデータゾーン制御

DZONE(P)(_U)

指定した入力値(BIN32ビット値)にバイアス値を付加します。

ラダー	ST	
	ENO:=DZONE(EN,s1,s2,s3,d); ENO:=DZONEP(EN,s1,s2,s3,d);	ENO:=DZONE_U(EN,s1,s2,s3,d); ENO:=DZONEP_U(EN,s1,s2,s3,d);

FBD/LD



■実行条件

命令	実行条件
DZONE DZONE_U	
DZONEP DZONEP_U	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	DZONE(P) DZONE(P)_U	入力値に加算する負のバイアス値を格納するデバイス 0~4294967295	符号付きBIN32ビット 符号なしBIN32ビット	ANY32_S ANY32_U
(s2)	DZONE(P) DZONE(P)_U	入力値に加算する正のバイアス値を格納するデバイス 0~4294967295	符号付きBIN32ビット 符号なしBIN32ビット	ANY32_S ANY32_U
(s3)	DZONE(P) DZONE(P)_U	ゾーン制御を行うための入力値を格納するデバイス 0~4294967295	符号付きBIN32ビット 符号なしBIN32ビット	ANY32_S ANY32_U
(d)	DZONE(P) DZONE(P)_U	ゾーン制御により制御した出力値を格納する先頭デバイス	符号付きBIN32ビット 符号なしBIN32ビット	ANY32_S ANY32_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

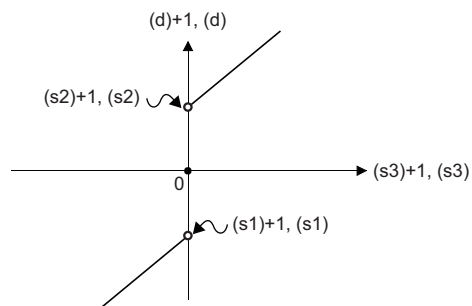
■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(s3)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

機能

- (s3)で指定した入力値(BIN32ビット値)に、(s1)または(s2)で指定したバイアス値を付加して、(d)で指定したデバイス番号に格納します。バイアス値は下記のように制御されます。

条件	出力値に格納される値
入力値((s3), (s3)+1)<0のとき	入力値((s3), (s3)+1)+負のバイアス値(s1), (s1)+1
入力値((s3), (s3)+1)=0のとき	0
入力値((s3), (s3)+1)>0のとき	入力値((s3), (s3)+1)+正のバイアス値(s2), (s2)+1



- DZONE(P)命令の場合、演算結果が-2147483648~2147483647を超えるときは、下記の例のようになります。

例

(s1), (s1)+1が-1000で、(s3), (s3)+1が-2147483648のとき、出力値=-2147483648+(-1000)=80000000H-FFFFFC18H=7FFFC18H=2147482648となります。

- DZONE(P)_U命令の場合、演算結果が0~4294967295を超えるときは、下記の例のようになります。

例

(s2), (s2)+1が1000で、(s3), (s3)+1が4294967295のとき、出力値=4294967295+1000=FFFFFFFFH-00003E8H=000003E7H=999となります。

- DZONE(P)_U命令の場合、(s1), (s1)+1はダミーデータ扱いとして使用しません。

エラー

演算エラーはありません。

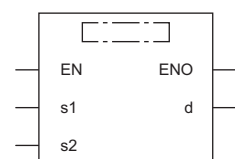
BIN16ビット単位スケーリング(ポイント別座標データ)

SCL(P)(_U)

スケーリング用変換データ(16ビットデータ単位)に対し、指定した入力値でスケーリング(ポイント別座標データ)を行います。

ラダー	ST	
	ENO:=SCL(EN,s1,s2,d); ENO:=SCLP(EN,s1,s2,d);	ENO:=SCL_U(EN,s1,s2,d); ENO:=SCLP_U(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
SCL SCL_U	
SCLP SCLP_U	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	SCL(P)	-32768~32767	符号付きBIN16ビット	ANY16_S
	SCL(P)_U	0~65535	符号なしBIN16ビット	ANY16_U
(s2)	SCL(P)	—	符号付きBIN16ビット*1	ANY16_S*2
	SCL(P)_U	—	符号なしBIN16ビット*1	ANY16_U*2
(d)	SCL(P)	—	符号付きBIN16ビット	ANY16_S
	SCL(P)_U	—	符号なしBIN16ビット	ANY16_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 (s2)の座標ポイント数は、符号なしBIN16ビットとなります。

*2 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

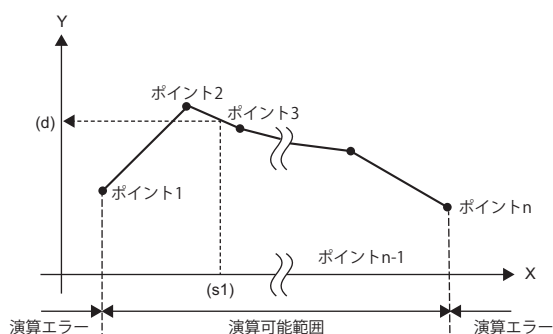
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- (s2)で指定したスケーリング用変換データ(16ビットデータ単位)に対し、(s1)で指定した入力値からスケーリングを行い、(d)で指定されたデバイス番号に演算結果を格納します。スケーリング変換は、(s2)で指定したデバイス以降に格納したスケーリング用変換データに基づいて行います。

設定項目(nは(s2)で指定する座標ポイント数を示します。)		デバイス割付
座標ポイント数		(s2)
ポイント1	X座標	(s2)+1
	Y座標	(s2)+2
ポイント2	X座標	(s2)+3
	Y座標	(s2)+4
⋮		
ポイントn	X座標	(s2)+2n-1
	Y座標	(s2)+2n



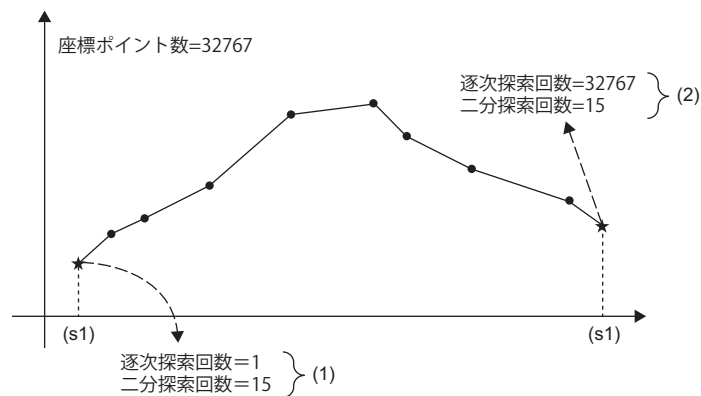
- 演算結果が整数値にならない場合は、小数点以下1桁目を四捨五入します。
- スケーリング用変換データのX座標データは、昇順に設定してください。
- (s1)は、スケーリング用変換データ範囲内((s2)のデバイス値)で設定してください。
- 複数のポイントが同一X座標を指定している場合は、ポイントNo.が最も大きいポイントのY座標の値を出力します。
- (s2)で指定するスケーリング用変換データの座標ポイント数は1~65535にしてください。

注意事項

- 探索方法と探索回数範囲は、SM755(スケーリングデータチェック設定)のON/OFFにより異なります。

SM755	探索方法	探索回数範囲
OFF	逐次探索	1≤回数≤65535
ON	二分探索	1≤回数≤16

- スケーリング用変換データが昇順にソートされている場合、SM755の状態により探索方法が異なるため、処理速度も異なります。探索回数により処理速度が決まり、探索回数が少ないほど処理速度が速くなります。
 - 逐次探索の処理速度が速くなる場合
座標ポイント数が最大で、(s1)が座標ポイント1~15の間にある場合、逐次探索回数≤15となるため、逐次探索の処理速度が速くなります。
 - 二分探索の処理速度が速くなる場合
最大探索回数が16回のため、(s1)が座標ポイント17以降にある場合、二分探索回数≤逐次探索回数となり、二分探索の処理速度が速くなります。



- (1) 逐次探索回数<二分探索回数のため、逐次探索の処理速度が速くなります。
(2) 逐次探索回数>二分探索回数のため、二分探索の処理速度が速くなります。

エラー

エラーコード (SD0)	内容
3285H	スケーリング用変換データの(s1)より前のポイントのX座標データが昇順に設定されていないとき。 (ただし、SM755がON時には、本エラーを検出しません。)
	(s1)で指定した入力値が、設定したスケーリング用変換データの範囲外の時。
	(s2)のデバイスから座標ポイント数が1~65535の範囲外の時。

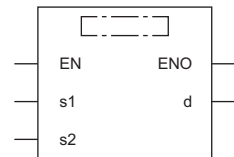
BIN32ビット単位スケーリング(ポイント別座標データ)

DSCL(P)(_U)

スケーリング用変換データ(32ビットデータ単位)に対し、指定した入力値でスケーリング(ポイント別座標データ)を行います。

ラダー	ST	
	ENO:=DSCL(EN,s1,s2,d); ENO:=DSCLP(EN,s1,s2,d);	ENO:=DSCL_U(EN,s1,s2,d); ENO:=DSCLP_U(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
DSCL DSCL_U	
DSCLP DSCLP_U	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	DSCL(P)	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	DSCL(P)_U	0~4294967295	符号なしBIN32ビット	ANY32_U
(s2)	DSCL(P)	—	符号付きBIN32ビット*1	ANY32_S*2
	DSCL(P)_U	—	符号なしBIN32ビット*1	ANY32_U*2
(d)	DSCL(P)	—	符号付きBIN32ビット	ANY32_S
	DSCL(P)_U	—	符号なしBIN32ビット	ANY32_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 (s2)+0~(s2)+1の座標ポイント数は、符号なしBIN32ビットとなります。

*2 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

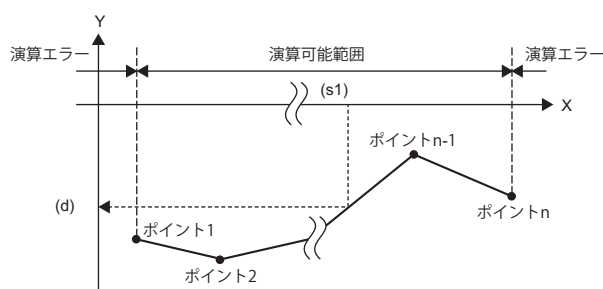
■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口		LT, LST, LC	LZ		K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- ・(s2)で指定したスケーリング用変換データ(32ビットデータ単位)に対し、(s1)で指定した入力値からスケーリングを行い、(d)で指定されたデバイス番号に演算結果を格納します。スケーリング変換は、(s2)で指定したデバイス以降に格納したスケーリング用変換データに基づいて行います。

設定項目(nは(s2)で指定する座標ポイント数を示します。)		デバイス割付
座標ポイント数		(s2)+1, (s2)
ポイント1	X座標	(s2)+3, (s2)+2
	Y座標	(s2)+5, (s2)+4
ポイント2	X座標	(s2)+7, (s2)+6
	Y座標	(s2)+9, (s2)+8
⋮		
ポイントn	X座標	(s2)+4n-1, (s2)+4n-2
	Y座標	(s2)+4n+1, (s2)+4n



- ・演算結果が整数値にならない場合は、小数点以下1桁目を四捨五入します。
- ・スケーリング用変換データのX座標データは、昇順に設定してください。
- ・(s1)は、スケーリング用変換データ範囲内((s2), (s2)+1)のデバイス値で設定してください。
- ・複数のポイントが同一X座標を指定している場合は、ポイントNo.が最も大きいポイントのY座標の値を出力します。
- ・(s2)で指定するスケーリング用変換データの座標ポイント数は1~4294967295にしてください。

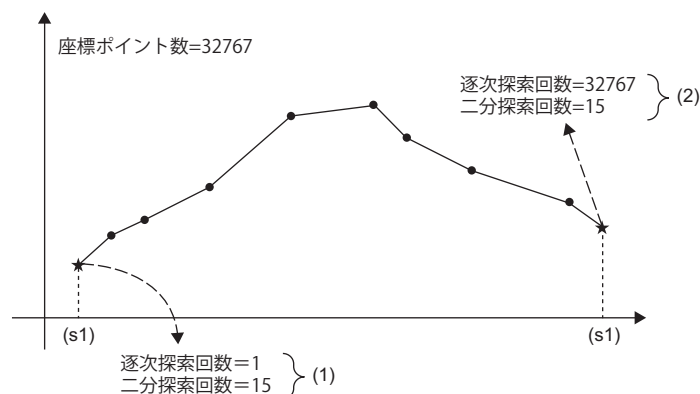
注意事項

- ・探索方法と探索回数範囲は、SM755(スケーリングデータチェック設定)のON/OFFにより異なります。

SM755	探索方法	探索回数範囲
OFF	逐次探索	1≤回数≤4294967295
ON	二分探索	1≤回数≤32

- ・スケーリング用変換データが昇順にソートされている場合、SM755の状態により探索方法が異なるため、処理速度も異なります。探索回数により処理速度が決まり、探索回数が少ないほど処理速度が速くなります。

- ・逐次探索の処理速度が速くなる場合
座標ポイント数が最大で、(s1)が座標ポイント1~15の間にある場合、逐次探索回数≤15となるため、逐次探索の処理速度が速くなります。
- ・二分探索の処理速度が速くなる場合
最大探索回数が32回のため、(s1)が座標ポイント33以降にある場合、二分探索回数≤逐次探索回数となり、二分探索の処理速度が速くなります。



- (1) 逐次探索回数<二分探索回数のため、逐次探索の処理速度が速くなります。
- (2) 逐次探索回数>二分探索回数のため、二分探索の処理速度が速くなります。

エラーコード (SD0)	内容
3285H	スケーリング用変換データの(s1)より前のポイントのX座標データが昇順に設定されていないとき。 (ただし、SM755がON時には、本エラーを検出しません。)
	(s1)で指定した入力値が、設定したスケーリング用変換データの範囲外の時。
	(s2)のデバイスから座標ポイント数が1~4294967295の範囲外の時。

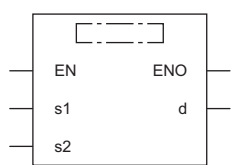
BIN16ビット単位スケーリング(X/Y別座標データ)

SCL2(P)(_U)

スケーリング用変換データ(16ビットデータ単位)に対し、指定した入力値でスケーリング(X/Y別座標データ)を行います。

ラダー	ST	
	ENO:=SCL2(EN,s1,s2,d); ENO:=SCL2P(EN,s1,s2,d);	ENO:=SCL2_U(EN,s1,s2,d); ENO:=SCL2P_U(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
SCL2 SCL2_U	
SCL2P SCL2P_U	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	SCL2(P)	-32768~32767	符号付きBIN16ビット	ANY16_S
	SCL2(P)_U	0~65535	符号なしBIN16ビット	ANY16_U
(s2)	SCL2(P)	—	符号付きBIN16ビット*1	ANY16_S*2
	SCL2(P)_U		符号なしBIN16ビット*1	ANY16_U*2
(d)	SCL2(P)	—	符号付きBIN16ビット	ANY16_S
	SCL2(P)_U		符号なしBIN16ビット	ANY16_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 (s2)の座標ポイント数は、符号なしBIN16ビットとなります。
*2 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

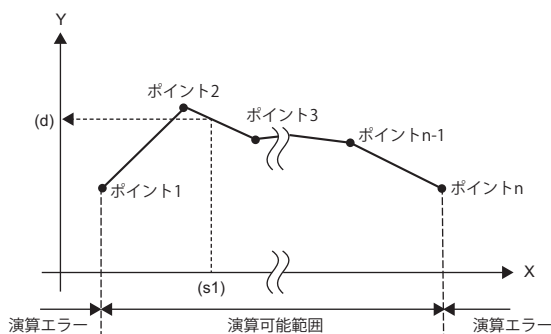
■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J○¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U○¥G□, J○¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	—	—	—
(s2)	—	—	○	—	—	—	—	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—

機能

- (s2)で指定したスケーリング用変換データ(16ビットデータ単位)に対し、(s1)で指定した入力値からスケーリングを行い、(d)で指定されたデバイス番号に演算結果を格納します。スケーリング変換は、(s2)で指定したデバイス以降に格納したスケーリング用変換データに基づいて行います。

設定項目(nは(s2)で指定する座標ポイント数を示します。)		デバイス割付
座標ポイント数		(s2)
X座標	ポイント1	(s2)+1
	ポイント2	(s2)+2
	⋮	⋮
	ポイントn	(s2)+n
Y座標	ポイント1	(s2)+n+1
	ポイント2	(s2)+n+2
	⋮	⋮
	ポイントn	(s2)+2n



- 演算結果が整数値にならない場合は、小数点以下1桁目を四捨五入します。
- スケーリング用変換データのX座標データは、昇順に設定してください。
- (s1)は、スケーリング用変換データ範囲内((s2)のデバイス値)で設定してください。
- 複数のポイントが同一X座標を指定している場合は、ポイントNo.が最も大きいポイントのY座標の値を出力します。
- スケーリング用変換データの座標ポイント数は1~65535にしてください。

注意事項

スケーリング用変換データが昇順にソートされている場合、SM755(スケーリングデータチェック設定)の状態により探索方法が異なるため、処理速度も異なります。詳細については、SCL(P)(U)命令を参照してください。

☞ 670ページ SCL(P)(U)

エラー

エラーコード (SD0)	内容
3285H	X座標データが昇順に設定されていないとき。
	(s1)で指定した入力値が、設定したスケーリング用変換データの範囲外のとき。
	(s2)のデバイスから座標ポイント数が1~65535の範囲外のとき。

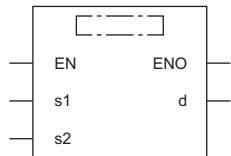
BIN32ビット単位スケーリング(X/Y別座標データ)

DSCL2(P)(_U)

スケーリング用変換データ(32ビットデータ単位)に対し、指定した入力値でスケーリング(X/Y別座標データ)を行います。

ラダー	ST	
	ENO:=DSCL2(EN,s1,s2,d); ENO:=DSCL2P(EN,s1,s2,d);	ENO:=DSCL2_U(EN,s1,s2,d); ENO:=DSCL2P_U(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
DSCL2 DSCL2_U	
DSCL2P DSCL2P_U	

設定データ

■内容、範囲、データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(s1)	DSCL2(P)	スケーリングを行う入力値または、入力値が格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	DSCL2(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U
(s2)	DSCL2(P)	スケーリング用変換データが格納されている先頭デバイス	—	符号付きBIN32ビット*1	ANY32_S*2
	DSCL2(P)_U			符号なしBIN32ビット*1	ANY32_U*2
(d)	DSCL2(P)	スケーリングにより制御した出力値を格納する先頭デバイス	—	符号付きBIN32ビット	ANY32_S
	DSCL2(P)_U			符号なしBIN32ビット	ANY32_U
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

*1 (s2)~(s2)+1の座標ポイント数は、符号なしBIN32ビットとなります。

*2 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

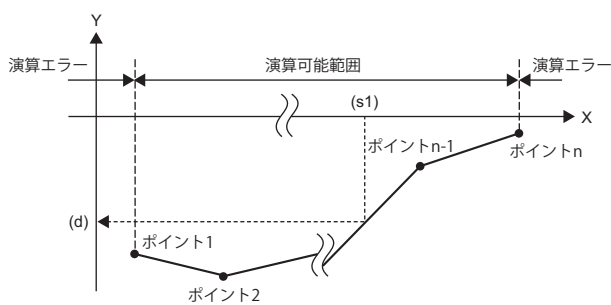
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—

機能

- (s2)で指定したスケーリング用変換データ(32ビットデータ単位)に対し、(s1)で指定した入力値からスケーリングを行い、(d)で指定されたデバイス番号に演算結果を格納します。スケーリング変換は、(s2)で指定したデバイス以降に格納したスケーリング用変換データに基づいて行います。

設定項目(nは(s2)で指定する座標ポイント数を示します。)		デバイス割付
座標ポイント数		(s2)+1, (s2)
X座標	ポイント1	(s2)+3, (s2)+2
	ポイント2	(s2)+5, (s2)+4
	⋮	⋮
	ポイントn	(s2)+2n+1, (s2)+2n
Y座標	ポイント1	(s2)+2n+3, (s2)+2n+2
	ポイント2	(s2)+2n+5, (s2)+2n+4
	⋮	⋮
	ポイントn	(s2)+4n+1, (s2)+4n



- 演算結果が整数値にならない場合は、小数点以下1桁目を四捨五入します。
- スケーリング用変換データのX座標データは、昇順に設定してください。
- (s1)は、スケーリング用変換データ範囲内((s2)~(s2)+1のデバイス値)で設定してください。
- 複数のポイントが同一X座標を指定している場合は、ポイントNo.が最も大きいポイントのY座標の値を出力します。
- スケーリング用変換データの座標ポイント数は1~4294967295にしてください。

注意事項

スケーリング用変換データが昇順にソートされている場合、SM755(スケーリングデータチェック設定)の状態により探索方法が異なるため、処理速度も異なります。詳細については、DSCL(P)(_U)命令を参照してください。

673ページ DSCL(P)(_U)

エラー

エラーコード (SD0)	内容
3285H	X座標データが昇順に設定されていないとき。
	(s1)で指定した入力値が、設定したスケーリング用変換データの範囲外の時。
	(s2)のデバイスから座標ポイント数が1~4294967295の範囲外の時。

14.6 データ処理命令

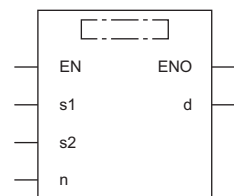
BIN16ビットデータサーチ

SERDATA(P)

(s1)で指定したBIN16ビットデータを，(s2)で指定したBIN16ビットデータから(n)点分をサーチします。

ラダー	ST
	ENO:=SERDATA(EN,s1,s2,n,d); ENO:=SERDATAP(EN,s1,s2,n,d);

FBD/LD



■実行条件

命令	実行条件
SERDATA	
SERDATAP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	サーチデータまたは，サーチデータが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(s2)	サーチされるデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(d)	サーチ結果を格納する先頭デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 2)
(n)	サーチ数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

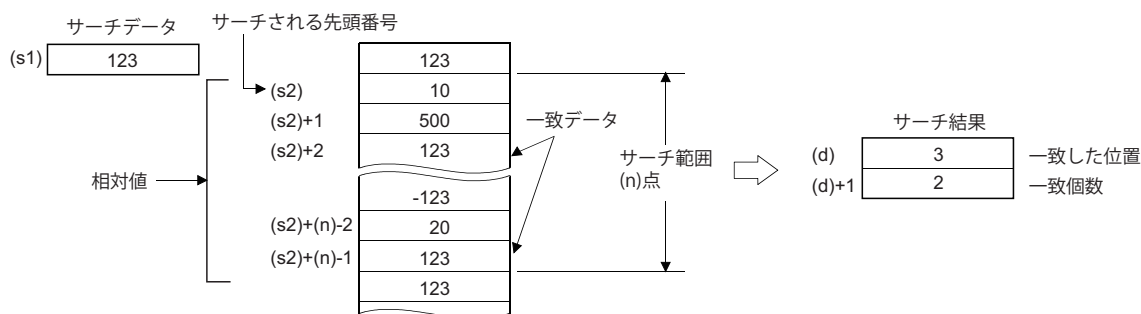
*1 ラベルで設定する場合は，動作に必要な領域が確保できるよう配列を定義し，その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数				その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ	K, H		E	\$			
(s1)	○	○	○	○	○	—	—	○	○	—	—	—		
(s2)	—	—	○	—	—	—	—	○	—	—	—	—		
(d)	—	—	○	○	○	—	—	○	—	—	—	—		
(n)	○	○	○	○	○	—	—	○	○	—	—	—		

機能

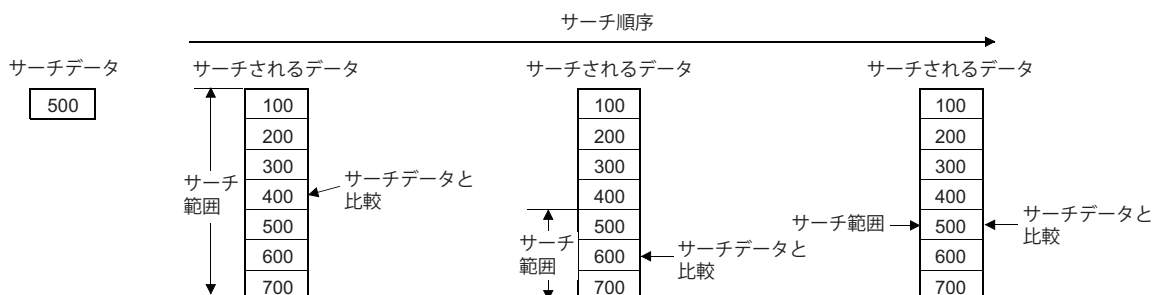
- (s1)で指定したデバイスのBIN16ビットデータをキーワードとして、(s2)で指定したデバイスからBIN16ビット単位で(n)点分をサーチします。キーワードと一致した個数を(d)+1で指定されたデバイスに、最初に一致したデバイス番号が(s2)からの相対値を(d)で指定されたデバイスに格納します。



- (n)が0の場合は、無処理となります。
- サーチした結果一致データが見つからなかった場合、(d), (d)+1で指定されたデバイスは0になります。

Point

- SERDATA(P)命令でサーチされるデータが昇順にソートされている場合、SM702(サーチ方法)*1をONすると二分探索法によるサーチにより処理を早くできます。サーチされるデータが昇順にソートされていない場合にSM702をONすると、正常なサーチ結果が得られません。二分探索法によるサーチの例を示します。



*1 SM702はサーチ方法設定用の特殊リレーです。

[SM702がOFFの場合]: 逐次探索法(線形探索法)になります。

サーチされるデータの先頭からサーチデータと比較していく方法です。

[SM702がONの場合]: 二分探索法になります。

昇順にソートされているデータに対し、探索範囲の中央の値を調べて、その値と探したい値の大小によってどちらか一方に探索範囲を絞り込みます。これを繰り返しながら、求めるデータを検索する方法です。

エラー

演算エラーはありません。

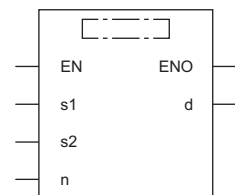
BIN32ビットデータサーチ

DSERDATA(P)

(s1)で指定したBIN32ビットデータを，(s2)で指定したBIN32ビットデータから(n)点分をサーチします。

ラダー	ST
	ENO:=DSERDATA(EN,s1,s2,n,d); ENO:=DSERDATAP(EN,s1,s2,n,d);

FBD/LD



■実行条件

命令	実行条件
DSERDATA	
DSERDATAP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	サーチデータまたは，サーチデータが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(s2)	サーチされるデータが格納されている先頭デバイス	—	符号付きBIN32ビット	ANY32*1
(d)	サーチ結果を格納する先頭デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 2)
(n)	サーチ数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

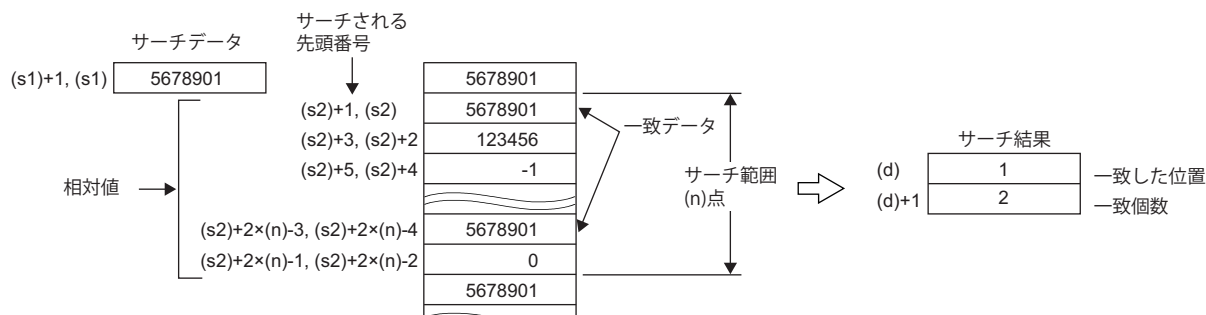
*1 ラベルで設定する場合は，動作に必要な領域が確保できるよう配列を定義し，その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—
(d)	—	—	○	○	○	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

機能

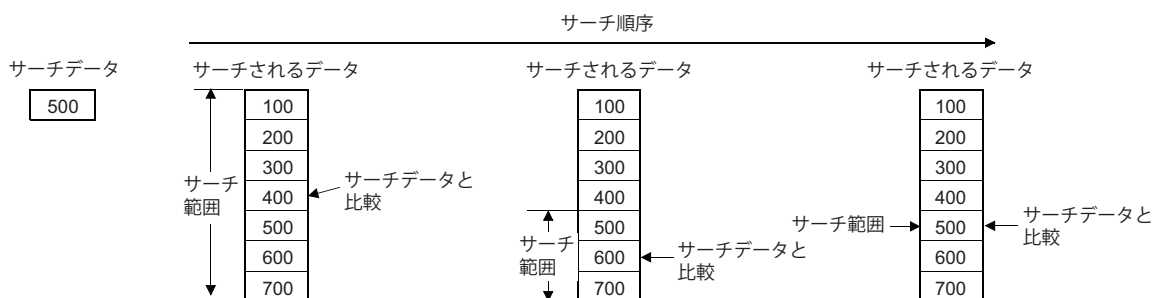
- (s1)で指定したデバイスのBIN32ビットデータをキーワードとして、(s2)で指定したデバイスからBIN32ビット単位で(n)点分(16ビット単位で $2 \times (n)$ 点分)をサーチします。キーワードと一致した個数を(d)+1で指定したデバイスに、最初に一致したデバイス番号が(s2)からの相対値を(d)で指定したデバイスに格納します。



- (n)が0の場合は、無処理となります。
- サーチした結果一致データが見つからなかった場合、(d), (d)+1で指定したデバイスは0になります。

Point

- DSERDATA(P)命令でサーチされるデータが昇順にソートされている場合、SM702(サーチ方法)^{*1}をONすると二分探索法によるサーチにより処理を早くできます。サーチされるデータが昇順にソートされていない場合にSM702をONすると、正常なサーチ結果が得られません。二分探索法によるサーチの例を示します。



^{*1} SM702はサーチ方法設定用の特殊リレーです。

[SM702がOFFの場合]: 逐次探索法(線形探索法)になります。

サーチされるデータの先頭からサーチデータと比較していく方法です。

[SM702がONの場合]: 二分探索法になります。

昇順にソートされているデータに対し、探索範囲の中央の値を調べて、その値と探したい値の大小によってどちらか一方に探索範囲を絞り込みます。これを繰り返しながら、求めるデータを検索する方法です。

エラー

演算エラーはありません。

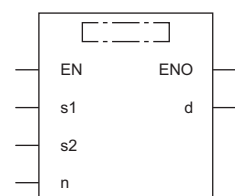
BIN16ビットデータサーチ(最小・同一・最大)

SERMM(P)

(s1)で指定したBIN16ビットデータから(n)点分、(s2)で指定したBIN16ビットデータと同一データおよび最大値、最小値を検索します。

ラダー	ST
	ENO:=SERMM(EN,s1,s2,n,d); ENO:=SERMMP(EN,s1,s2,n,d);

FBD/LD



■実行条件

命令	実行条件
SERMM	
SERMMP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	同一データ、最大値、最小値を検索する先頭デバイス	—	符号付きBIN16ビット	ANY16
(s2)	同一データを検索する値またはその格納先デバイス	—	符号付きBIN16ビット	ANY16
(d)	同一データ、最大値、最小値を検索後それらの個数が格納される先頭デバイス	—	符号なしBIN16ビット	ANY16_ARRAY (要素数: 5)
(n)	同一データ、最大値、最小値を検索する個数	0~65535	符号なしBIN16ビット	ANY16_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	○*1	○	○	○	○	—	—	○	○	—	—	—
(d)	—	—	○	○	—	—	—	○	—	—	—	—
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—

*1 FX, FYは使用できません。

機能

- (s1)を先頭とする(n)個のデータに対し、(s2)のBIN16ビットデータと同一のデータのものを検索し、結果を(d)~(d)+4に格納します。
- 同一データがあった場合、(d)を先頭とする5点のデバイスには、同一データの個数、初回/最終位置や最大値、最小値の位置が格納されます。
- 同一データがなかった場合、(d)を先頭とする5点のデバイスには、同一データの個数、初回/最終位置や最大値、最小値の位置が格納されます。ただし、(d)を先頭とする3点のデバイス(同一データの個数、初回/最終位置)には、0が格納されます。
- データの中に最小値、最大値が複数あるときは、それぞれ後側の位置が格納されます。
- (n)に0を指定した場合、無処理となります。
- 検索結果テーブルの構成とデータ例を下記に示します。(n=10)

被検索デバイス (s1)	被検索データ (s1)の値(例)	比較データ (s2)の値(例)	データの位置	検索結果		
				最大値(d)+4	一致(d)	最小値(d)+3
(s1)	100	100	0	—	○(初回)	—
(s1)+1	111		1	—	—	—
(s1)+2	100		2	—	○	—
(s1)+3	98		3	—	—	—
(s1)+4	123		4	—	—	—
(s1)+5	66		5	—	—	○
(s1)+6	100		6	—	○(最終)	—
(s1)+7	95		7	—	—	—
(s1)+8	210		8	○	—	—
(s1)+9	88		9	—	—	—

- 上記の例による検索結果テーブルを下記に示します。

デバイス番号	内容	検索結果項目
(d)	3	同一データの個数
(d)+1	0	同一データの位置(初回)
(d)+2	6	同一データの位置(最終)
(d)+3	5	最小値の最終位置
(d)+4	8	最大値の最終位置

エラー

演算エラーはありません。

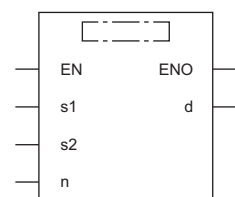
BIN32ビットデータサーチ(最小・同一・最大)

DSERMM(P)

(s1)で指定したBIN32ビットデータから(n)点分、(s2)で指定したBIN32ビットデータと同一データおよび最大値、最小値を検索します。

ラダー	ST
	ENO:=DSERMM(EN,s1,s2,n,d); ENO:=DSERMMP(EN,s1,s2,n,d);

FBD/LD



■実行条件

命令	実行条件
DSERMM	
DSERMMP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	同一データ、最大値、最小値を検索する先頭デバイス	—	符号付きBIN32ビット	ANY32
(s2)	同一データを検索する値またはその格納先デバイス	—	符号付きBIN32ビット	ANY32
(d)	同一データ、最大値、最小値を検索後それらの個数が格納される先頭デバイス	—	符号なしBIN32ビット	ANY32_ARRAY (要素数: 5)
(n)	同一データ、最大値、最小値を検索する個数	0~65535	符号なしBIN16ビット	ANY16_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(s2)	○*1	○	○	○	○	○	○	○	—	—	—
(d)	—	—	○	○	—	—	○	—	—	—	—
(n)	○*1	○	○	○	○	—	○	○	—	—	—

*1 FX, FYは使用できません。

機能

- (s1)+1, (s1)を先頭とする(n)個のデータに対し, (s2)+1, (s2)のBIN32ビットデータと同一のデータのものを検索し, 結果を[(d)+1, (d)]~[(d)+9, (d)+8]に格納します。
- 同一データがあった場合, (d)+1, (d)を先頭とする5点のデバイスには, 同一データの個数, 初回/最終位置や最大値, 最小値の位置が格納されます。
- 同一データがなかった場合, (d)+1, (d)を先頭とする5点のデバイスには, 同一データの個数, 初回/最終位置や最大値, 最小値の位置が格納されます。ただし, (d)+1, (d)を先頭とする3点のデバイス(同一データの個数, 初回/最終位置)には, 0が格納されます。
- データの中に最小値, 最大値が複数あるときは, それぞれ後側の位置が格納されます。
- (n)に0を指定した場合, 無処理となります。
- 検索結果テーブルの構成とデータ例を下記に示します。(n=10)

被検索デバイス (s1)	被検索データ (s1)の値(例)	比較データ (s2)の値(例)	データの位置	検索結果		
				最大値 (d)+9, (d)+8	一致(d)	最小値 (d)+7, (d)+6
(s1)+1, (s1)	100000	100000	0	—	○(初回)	—
(s1)+3, (s1)+2	110100		1	—	—	—
(s1)+5, (s1)+4	100000		2	—	○	—
(s1)+7, (s1)+6	98000		3	—	—	—
(s1)+9, (s1)+8	123000		4	—	—	—
(s1)+11, (s1)+10	66000		5	—	—	○
(s1)+13, (s1)+12	100000		6	—	○(最終)	—
(s1)+15, (s1)+14	95000		7	—	—	—
(s1)+17, (s1)+16	910000		8	○	—	—
(s1)+19, (s1)+18	910000		9	○	—	—

- 上記の例による検索結果テーブルを下記に示します。

デバイス番号	内容	検索結果項目
(d)+1, (d)	3	同一データの個数
(d)+3, (d)+2	0	同一データの位置(初回)
(d)+5, (d)+4	6	同一データの位置(最終)
(d)+7, (d)+6	5	最小値の最終位置
(d)+9, (d)+8	9	最大値の最終位置

エラー

演算エラーはありません。

BIN16ビットデータビットチェック

SUM(P)

指定したデバイスのBIN16ビットデータのうち、1になっているビットの総数を格納します。

ラダー	ST
	ENO:=SUM(EN,s,d); ENO:=SUMP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
SUM	
SUMP	

設定データ

■内容、範囲、データ型

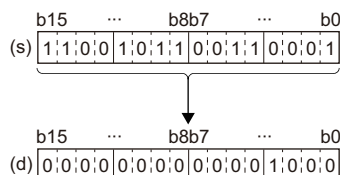
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	1になっているビットの総数をカウントするデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(d)	ビットの総数を格納するデバイス	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□		LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- ・(s)で指定されたデバイスのBIN16ビットデータのうち、1になっているビットの総数を(d)で指定されたデバイスに格納します。



1の総数がBINで格納されます。(左記の例では8個です。)

エラー

演算エラーはありません。

BIN32ビットデータビットチェック

DSUM(P)

指定したデバイスのBIN32ビットデータのうち、1になっているビットの総数を格納します。

ラダー	ST
	ENO:=DSUM(EN,s,d); ENO:=DSUMP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
DSUM	
DSUMP	

設定データ

■内容、範囲、データ型

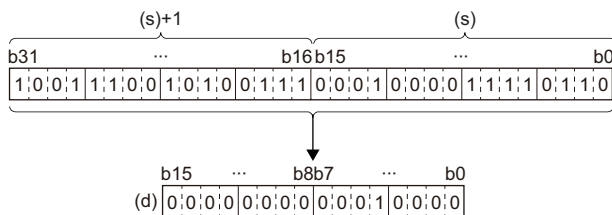
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	1になっているビットの総数をカウントする先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(d)	ビットの総数を格納するデバイス	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—

機能

- (s)で指定されたデバイスのBIN32ビットデータのうち、1になっているビットの総数を(d)で指定されたデバイスに格納します。



1の総数がBINで格納されます。(左記の例では16個です。)

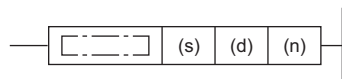
エラー

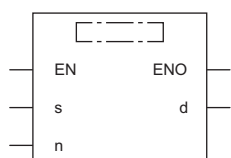
演算エラーはありません。

BIN16ビットデータ最大値検索

MAX(P)(_U)

指定したデバイスから(n)点のBIN16ビットデータから最大値を検索します。

ラダー	ST*1	
	ENO:=MAXP(EN,s,n,d);	ENO:=MAXP_U(EN,s,n,d);

FBD/LD*1


*1 MAX命令, MAX_U命令はST, FBD/LDでは対応していません。汎用ファンクションのMAXを使用してください。
1295ページ MAX(_E), MIN(_E)

■実行条件

命令	実行条件
MAX MAX_U	
MAXP MAXP_U	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	MAX(P)	—	符号付きBIN16ビット	ANY16_S*1
	MAX(P)_U		符号なしBIN16ビット	ANY16_U*1
(d)	MAX(P)	—	符号付きBIN16ビット	ANY16_S_ARRAY (要素数: 3)
	MAX(P)_U		符号なしBIN16ビット	ANY16_U_ARRAY (要素数: 3)
(n)	サーチデータ数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

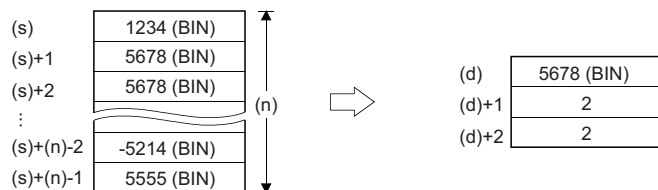
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	—	—	○	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

機能

- (s)で指定されたデバイスから(n)点のBIN16ビットデータから最大値を検索して、(d)で指定されたデバイスに最大値を格納します。(s)で指定されたデバイスから検索し、最初に検出した最大値が格納されているデバイス番号が(s)から何点目かを(d)+1に格納し、最大値の個数を(d)+2に格納します。



(d): 最大値

(d)+1: 位置

(d)+2: 個数

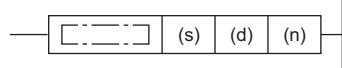
エラー

演算エラーはありません。

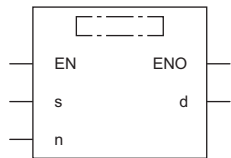
BIN32ビットデータ最大値検索

DMAX(P)(_U)

指定したデバイスから(n)点のBIN32ビットデータから最大値を検索します。

ラダー	ST*1	
	ENO:=DMAXP(EN,s,n,d);	ENO:=DMAXP_U(EN,s,n,d);

FBD/LD*1



*1 DMAX命令, DMAX_U命令はST, FBD/LDでは対応していません。汎用ファンクションのMAXを使用してください。
1295ページ MAX(_E), MIN(_E)

■実行条件

命令	実行条件
DMAX DMAX_U	
DMAXP DMAXP_U	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	DMAX(P) DMAX(P)_U	—	符号付きBIN32ビット	ANY32_S*1
			符号なしBIN32ビット	ANY32_U*1
(d)	DMAX(P) DMAX(P)_U	—	符号付きBIN32ビット	—*2 (ANY32_S_ARRAY)
			符号なしBIN32ビット	—*2 (ANY32_U_ARRAY)
(n)	サーチデータ数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

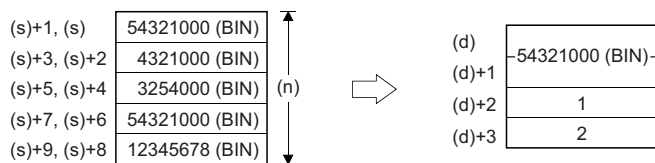
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。
*2 使用するプログラム言語にかかわらず、デバイスで指定します。ラベルは指定しないでください。

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口		LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- (s)で指定されたデバイスから(n)点のBIN32ビットデータから最大値を検索して、(d)、(d)+1で指定されたデバイスに最大値を格納します。(s)で指定したデバイスから検索し、最初に検出した最大値に格納されているデバイス番号が(s)から何点目かを(d)+2に格納し、最大値の個数を(d)+3に格納します。



(d), (d)+1:最大値

(d)+2: 位置

(d)+3: 個数

エラー

演算エラーはありません。

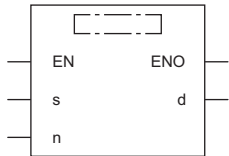
BIN16ビットデータ最小値検索

MIN(P)(_U)

指定したデバイスから(n)点のBIN16ビットデータから最小値を検索します。

ラダー	ST*1	
	ENO:=MINP(EN,s,n,d);	ENO:=MINP_U(EN,s,n,d);

FBD/LD*1



*1 MIN命令, MIN_U命令はST, FBD/LDでは対応していません。汎用ファンクションのMINを使用してください。
📖 1295ページ MAX(_E), MIN(_E)

■実行条件

命令	実行条件
MIN MIN_U	
MINP MINP_U	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	MIN(P)	—	符号付きBIN16ビット	ANY16_S*1
	MIN(P)_U		符号なしBIN16ビット	ANY16_U*1
(d)	MIN(P)	—	符号付きBIN16ビット	ANY16_S_ARRAY (要素数: 3)
	MIN(P)_U		符号なしBIN16ビット	ANY16_U_ARRAY (要素数: 3)
(n)	サーチデータ数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

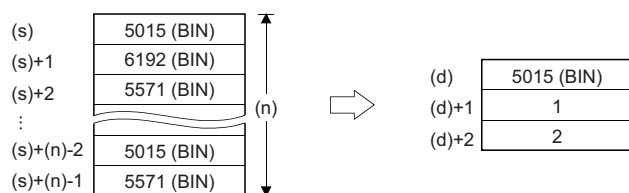
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- (s)で指定されたデバイスから(n)点までのBIN16ビットデータから最小値を検索して、(d)で指定されたデバイスに最小値を格納します。(s)で指定されたデバイスから検索し、最初に検出した最小値が格納されているデバイス番号が(s)から何点目かを(d)+1に格納し、最小値の個数を(d)+2に格納します。



(d): 最小値

(d)+1: 位置

(d)+2: 個数

エラー

演算エラーはありません。

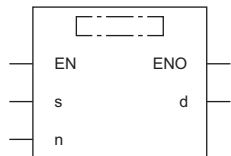
BIN32ビットデータ最小値検索

DMIN(P)(_U)

指定したデバイスから(n)点のBIN32ビットデータから最小値を検索します。

ラダー	ST*1	
	ENO:=DMINP(EN,s,n,d);	ENO:=DMINP_U(EN,s,n,d);

FBD/LD*1



*1 DMIN命令, DMIN_U命令はST, FBD/LDでは対応していません。汎用ファンクションのMINを使用してください。
📖 1295ページ MAX(_E), MIN(_E)

■実行条件

命令	実行条件
DMIN DMIN_U	
DMINP DMINP_U	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	DMIN(P)	—	符号付きBIN32ビット	ANY32_S*1
	DMIN(P)_U		符号なしBIN32ビット	ANY32_U*1
(d)	DMIN(P)	—	符号付きBIN32ビット	—*2 (ANY32_S_ARRAY)
	DMIN(P)_U		符号なしBIN32ビット	—*2 (ANY32_U_ARRAY)
(n)	サーチデータ数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は, 動作に必要な領域が確保できるよう配列を定義し, その配列型ラベルの要素を指定してください。

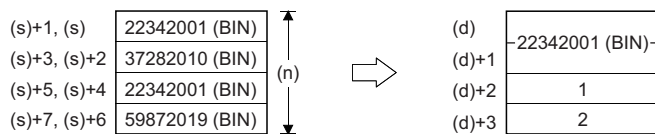
*2 使用するプログラム言語にかかわらず, デバイスで指定します。ラベルは指定しないでください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

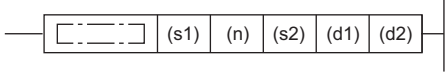
- (s)で指定されたデバイスから(n)点までのBIN32ビットデータから最小値を検索して、(d)、(d)+1で指定されたデバイスに最小値を格納します。(s)で指定されたデバイスから検索し、最初に検出した最小値に格納されているデバイス番号が(s)から何点目かを(d)+2に格納し、最小値の個数を(d)+3に格納します。



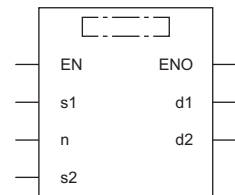
BIN16ビットデータソート

SORTD(_U)

指定した(n)点分のBIN16ビットデータを昇順/降順にソート(並び替え)します。

ラダー	ST	
	ENO:=SORTD(EN,s1,n,s2,d1,d2);	ENO:=SORTD_U(EN,s1,n,s2,d1,d2);

FBD/LD



■実行条件

命令	実行条件
SORTD SORTD_U	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	SORTD	—	符号付きBIN16ビット	ANY16_S*1
	SORTD_U		符号なしBIN16ビット	ANY16_U*1
(n)	ソートデータ数	0~65535	符号なしBIN16ビット	ANY16
(s2)	1回の実行で比較するデータ数	1~65535	符号なしBIN16ビット	ANY16
(d1)	ソート完了でONさせるビットデバイス番号	—	ビット	ANY_BOOL
(d2)	システム使用デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

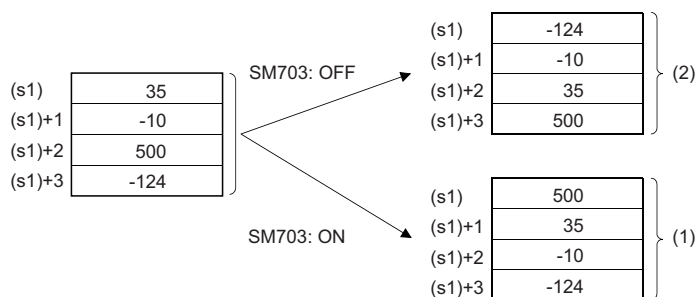
■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, J□¥□	Z	LT, LST, LC		K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	○	○	—	—	—
(d1)	○	—	○*1	—	—	—	○	—	—	—	—
(d2)	—	—	○	—	—	—	○	—	—	—	—

*1 T, C, STは使用できません。

機能

- (s1)から(n)点分のBIN16ビットデータを、昇順または降順にソート(並び替え)します。ソート順は、SM703(ソート順)がOFF時に昇順でソートし、ON時に降順でソートします。



(1): 降順にソートします。

(2): 昇順にソートします。

- SORTD(_U)命令によるソートは、数スキャンが必要です。実行完了までのスキャン回数は、ソート実行完了までの最大実行回数を(s2)で指定された1回の実行で比較するデータ数で割った値(小数点以下は切り上げ)になります。(s2)の値を多くすると、ソート完了までのスキャン回数は少なくなりますが、スキャンタイムは延びます。
- ソートの実行完了までの最大実行回数は、 $(n) \times (n-1) \div 2$ (回)で算出します。たとえば、(n)=10の場合は、 $10 \times (10-1) \div 2 = 45$ 回となります。このとき(s2)=2にすると、ソート完了までに $45 \div 2 = 22.5$ となり、23スキャンかかります。
- (d1)で指定されたデバイス(完了デバイス)は、SORTD(_U)命令実行開始でOFFし、ソートが完了するとONします。ソート完了後、(d1)で指定されたデバイスはONのまま保持されますので、必要に応じてOFFしてください。
- (d2)で指定されたデバイスから2点は、SORTD(_U)命令実行時にシステムが使用します。(d2)で指定されたデバイスから2点は変更しないでください。変更した場合は、エラーとなる場合があります。(エラーコード: 3285H)
- ソート実行中に(n)を変更した場合は、変更後のソートデータ数でソートします。
- ソート実行中に実行指令をOFFした場合は、ソートを中断します。再度実行指令をONした場合は、ソートを最初からやり直します。
- ソートの実行完了後、連続して次のソートを行う場合は、実行指令を一度OFFした後、再度実行指令をONする必要があります。

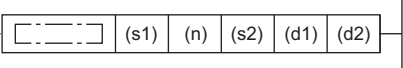
エラー

エラーコード (SD0)	内容
2821H	(s1)から(n)点分のデバイス範囲と(d2)から2点分のデバイス範囲が重複しているとき。
3285H	(s2)が0のとき。
	2スキャン目以降で、システムで使用する(d2)の値が(n)以上の値であるとき。
	2スキャン目以降で、システムで使用する(d2)の値が、 $(d2) < (d2)+1$ であるとき。

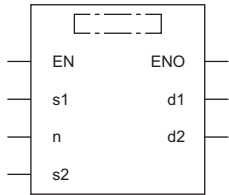
BIN32ビットデータソート

DSORTD(_U)

指定した(n)点分のBIN32ビットデータを昇順/降順にソート(並び替え)します。

ラダー	ST	
	ENO:=DSORTD(EN,s1,n,s2,d1,d2);	ENO:=DSORTD_U(EN,s1,n,s2,d1,d2);

FBD/LD



■実行条件

命令	実行条件
DSORTD DSORTD_U	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	DSORTD	—	符号付きBIN32ビット	ANY32_S ^{*1}
	DSORTD_U		符号なしBIN32ビット	ANY32_U ^{*1}
(n)	ソートデータ数	0~65535	符号なしBIN16ビット	ANY16
(s2)	1回の実行で比較するデータ数	1~65535	符号なしBIN16ビット	ANY16
(d1)	ソート完了でONさせるビットデバイス番号	—	ビット	ANY_BOOL
(d2)	システム使用デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

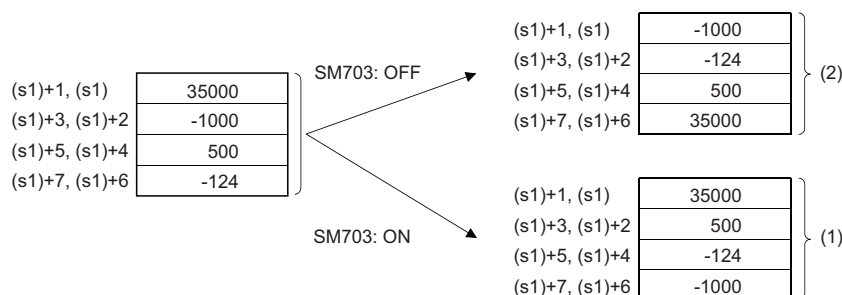
■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	○	○	—	—	—
(d1)	○	—	○ ^{*1}	—	—	—	○	—	—	—	—
(d2)	—	—	○	—	—	—	○	—	—	—	—

*1 T, C, STは使用できません。

機能

- (s1)から(n)点分のBIN32ビットデータを昇順または降順にソート(並び替え)します。ソート順は、SM703(ソート順)がOFF時に昇順でソートし、ON時に降順でソートする。



(1): 降順にソートします。

(2): 昇順にソートします。

- DSORTD(U)命令によるソートは、数スキャンが必要です。実行完了までのスキャン回数は、ソート実行完了までの最大実行回数を(s2)で指定された1回の実行で比較するデータ数で割った値(小数点以下は切り上げ)になります。(s2)の値を多くすると、ソート完了までのスキャン回数は少なくなりますが、スキャンタイムは延びます。
- ソートの実行完了までの最大実行回数は、 $(n) \times (n-1) \div 2$ (回)で算出します。たとえば、 $(n)=10$ の場合は、 $10 \times (10-1) \div 2 = 45$ 回となります。このとき(s2)=2にすると、ソート完了までに $45 \div 2 = 22.5$ となり、23スキャンかかります。
- (d1)で指定されたデバイス(完了デバイス)は、DSORTD(U)命令開始実行でOFFし、ソートが完了するとONします。ソート完了後、(d1)で指定されたデバイスはONのまま保持されますので、必要に応じてOFFしてください。
- (d2)で指定されたデバイスから2点は、DSORTD(U)命令実行時にシステムが使用します。(d2)で指定されたデバイスから2点は変更しないでください。変更した場合は、エラーとなる場合があります。(エラーコード: 3285H)
- ソート実行中に(n)を変更した場合は、変更後のソートデータ数でソートします。
- ソート実行中に実行指令をOFFした場合は、ソートを中断します。再度実行指令をONした場合は、ソートを最初からやり直します。
- ソートの実行完了後、連続して次のソートを行う場合は、実行指令を一度OFFした後、再度実行指令をONする必要があります。

エラー

エラーコード (SD0)	内容
2821H	(s1)から2×(n)点分のデバイス範囲と(d2)から2点分のデバイス範囲が重複しているとき。
3285H	(s2)が0のとき。
	2スキャン目以降で、システムで使用する(d2)の値が(n)以上の値であるとき。
	2スキャン目以降で、システムで使用する(d2)の値が、 $(d2) < (d2)+1$ であるとき。

BIN16ビットデータ合計値算出

WSUM(P)(_U)

指定したデバイスから(n)点のBIN16ビットデータをすべて加算する。

ラダー	ST	
	ENO:=WSUM(EN,s,n,d); ENO:=WSUMP(EN,s,n,d);	ENO:=WSUM_U(EN,s,n,d); ENO:=WSUMP_U(EN,s,n,d);

FBD/LD

■実行条件

命令	実行条件
WSUM WSUM_U	
WSUMP WSUMP_U	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	WSUM(P)	—	符号付きBIN16ビット	ANY16_S*1
	WSUM(P)_U		符号なしBIN16ビット	ANY16_U*1
(d)	WSUM(P)	—	符号付きBIN32ビット	ANY32_S
	WSUM(P)_U		符号なしBIN32ビット	ANY32_U
(n)	データ个数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

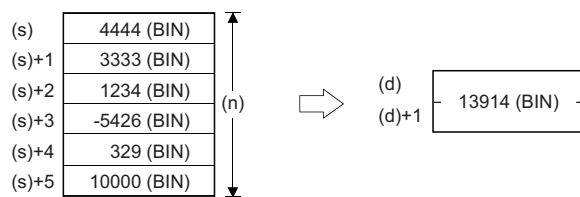
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	—	—	○	—	—	—	○	—	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	—	—	—

機能

- (s)で指定されたデバイスから(n)点分のBIN16ビットデータをすべて加算し、(d)で指定されたデバイスに格納します。



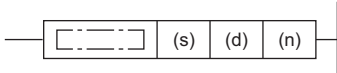
エラー

演算エラーはありません。

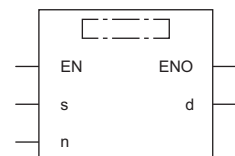
BIN32ビットデータ合計値算出

DWSUM(P)(_U)

指定したデバイスから(n)点のBIN32ビットデータをすべて加算します。

ラダー	ST	
	ENO:=DWSUM(EN,s,n,d); ENO:=DWSUMP(EN,s,n,d);	ENO:=DWSUM_U(EN,s,n,d); ENO:=DWSUMP_U(EN,s,n,d);

FBD/LD



■実行条件

命令	実行条件
DWSUM DWSUM_U	
DWSUMP DWSUMP_U	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	DWSUM(P) DWSUM(P)_U	—	符号付きBIN32ビット	ANY32_S ^{*1}
			符号なしBIN32ビット	ANY32_U ^{*1}
(d)	DWSUM(P) DWSUM(P)_U	—	符号付きBIN64ビット	ANY32_ARRAY (要素数: 2)
			符号なしBIN64ビット	
(n)	データ個数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

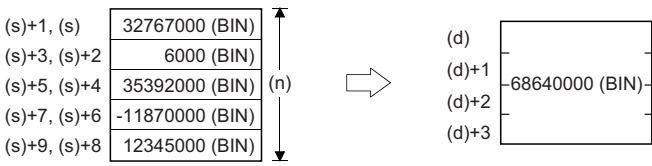
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- (s)で指定されたデバイスから(n)点分のBIN32ビットデータをすべて加算し、(d)で指定されたデバイスに格納します。



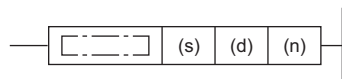
エラー

演算エラーはありません。

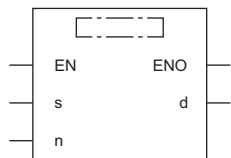
BIN16ビットデータ平均値算出

MEAN(P)(_U)

指定したデバイスから(n)点分(16ビットデータ単位)の平均値を算出します。

ラダー	ST	
	ENO:=MEAN(EN,s,n,d); ENO:=MEANP(EN,s,n,d);	ENO:=MEAN_U(EN,s,n,d); ENO:=MEANP_U(EN,s,n,d);

FBD/LD



■実行条件

命令	実行条件
MEAN MEAN_U	
MEANP MEANP_U	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	MEAN(P) MEAN(P)_U	—	符号付きBIN16ビット	ANY16_S*1
			符号なしBIN16ビット	ANY16_U*1
(d)	MEAN(P) MEAN(P)_U	—	符号付きBIN16ビット	ANY16_S
			符号なしBIN16ビット	ANY16_U
(n)	データ数またはデータ数が格納されたデバイス番号	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

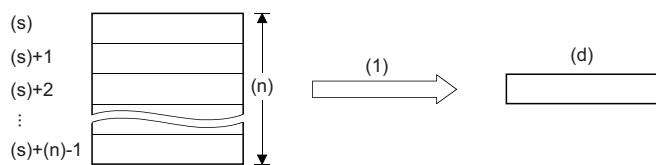
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

機能

- (s)で指定されたデバイスから(n)点分のBIN16ビットデータの平均値を算出し、(d)で指定されたデバイスに格納します。



(1): 平均値

- 算出結果が整数値にならない場合は、小数点以下は切り捨てとなります。
- (n)で指定した値が0の場合は、無処理となります。

エラー

演算エラーはありません。

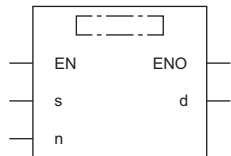
BIN32ビットデータ平均値算出

DMEAN(P)(_U)

指定したデバイスから(n)点分(32ビットデータ単位)の平均値を算出します。

ラダー	ST	
	ENO:=DMEAN(EN,s,n,d); ENO:=DMEANP(EN,s,n,d);	ENO:=DMEAN_U(EN,s,n,d); ENO:=DMEANP_U(EN,s,n,d);

FBD/LD



■実行条件

命令	実行条件
DMEAN DMEAN_U	
DMEANP DMEANP_U	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	DMEAN(P)	—	符号付きBIN32ビット	ANY32_S ^{*1}
	DMEAN(P)_U		符号なしBIN32ビット	ANY32_U ^{*1}
(d)	DMEAN(P)	—	符号付きBIN32ビット	ANY32_S
	DMEAN(P)_U		符号なしBIN32ビット	ANY32_U
(n)	データ数またはデータ数が格納されたデバイス番号	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

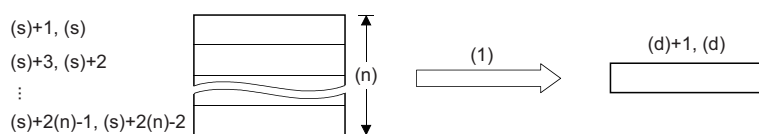
*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J○¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U○¥G□, J○¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—
(n)	○	○	○	○	○	—	—	○	—	—	—

機能

- (s)で指定されたデバイスから(n)点分のBIN32ビットデータの平均値を算出し、(d)で指定されたデバイスに格納します。



(1): 平均値

- 算出結果が整数値にならない場合は、小数点以下は切り捨てとなります。
- (n)で指定した値が0の場合は、無処理となります。

エラー

演算エラーはありません。

BIN16ビット平方根算出

SQRT(P)

指定したBIN16ビットデータの平方根を演算します。

ラダー	ST*1
	ENO:=SQRTP(EN,s,d);

FBD/LD*1

*1 SQRT命令はST言語、FBD/LD言語で対応していません。汎用ファンクションのSQRTを使用してください。(P1255ページ SQRT(_E))

■実行条件

命令	実行条件
SQRT	
SQ RTP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	平方根を演算するデータを格納するデバイス	0~65535	符号なしBIN16ビット	ANY16
(d)	演算した平方根が格納されるデバイス	—	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	○*1	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○*1	○	○	○	○	—	—	○	—	—	—	—	—

*1 FX, FYは使用できません。

機能

- (s)で指定したBIN16ビットデータの平方根を演算し、演算結果を(d)に格納します。演算結果は小数点が切り捨てられて整数となります。

$\sqrt{(s)} \rightarrow (d)$

エラー

演算エラーはありません。

BIN32ビット平方根算出

DSQRT(P)

指定したBIN32ビットデータの平方根を演算します。

ラダー	ST
	ENO:=DSQRT(EN,s,d); ENO:=DSQRTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
DSQRT	
DSQRTP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	平方根を演算するデータを格納するデバイス	0~4294967295	符号なしBIN32ビット	ANY32
(d)	演算した平方根が格納されるデバイス	—	符号なしBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	○*1	○	○	○	○	○	○	○	—	—	—
(d)	○*1	○	○	○	○	○	○	—	—	—	—

*1 FX, FYは使用できません。

機能

- (s)で指定したBIN32ビットデータの平方根を演算し、演算結果を(d)に格納します。演算結果は小数点が切り捨てられて整数となります。

$\sqrt{(s)+1}$, $(s) \rightarrow (d)$

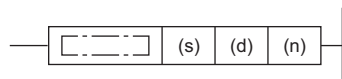
エラー

演算エラーはありません。

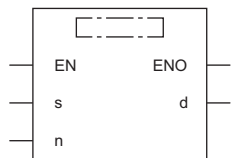
CRC演算

CRC(P)

(s)で指定したデバイスを先頭に(n)点の8ビットデータ(バイト単位)について、CRC値を生成し(d)へ格納します。

ラダー	ST
	ENO:=CRC(EN,s,n,d); ENO:=CRCP(EN,s,n,d);

FBD/LD



■実行条件

命令	実行条件
CRC	
CRCP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	CRC値の生成対象となるデータを格納する先頭デバイス	—	符号付きBIN16ビット	ANY16
(d)	生成されたCRC値の格納先デバイス	—	符号付きBIN16ビット	ANY16
(n)	CRC値を求める8ビットデータ(バイト単位)数、または8ビットデータ(バイト単位)数が格納されたデバイス	0~65535	符号なしBIN16ビット	ANY16_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口		LT, LST, LC	LZ		K, H	E	\$	
(s)	○*1	—	○	—	—	—	—	○	—	—	—	—
(d)	○*1	—	○	—	—	—	—	○	—	—	—	—
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—

*1 FX, FYは使用できません。

機能

- ・ (s)で指定したデバイスを先頭に(n)点の8ビットデータ(バイト単位)について、CRC値を生成し(d)へ格納します。CRC値(CRC-16)の生成多項式には「 $X^{16}+X^{15}+X^2+1$ 」を使用しています。この命令には、算出する際に使用するモードとして、16ビット変換モードと8ビット変換モードがあります。変換モードは、SM772(CCD/CRC命令変換モード指定)のON/OFFによって選択できます。
- ・ (n)に0を指定した場合、無処理となります。
- ・ 各変換モードの動作を下記に示します。

■16ビット変換モード(SM772=OFF時)

(s)のデバイスの上位8ビット(バイト単位)と下位8ビット(バイト単位)に対してCRC演算を行います。演算結果は(d)に指定したデバイス1点の16ビットに格納します。

例

(n)=6の場合

16ビット変換モードの場合、下記の網掛け部の6バイトが演算対象となります。CRC値が「A57BH」となるため、「A57BH」を(d)のデバイスへ格納します。

	10進	16進	
		上位	下位
D0	24932	61H	64H
D1	4219	10H	7BH
D2	-1333	FAH	CBH
D3	-1	FFH	FFH
D4	32761	7FH	F9H
D5	10000	27H	10H

■8ビット変換モード(SM772=ON時)

(s)のデバイスの下位8ビット(下位バイト)のみにに対してCRC演算を行います。演算結果は(d)に指定したデバイスから2点を使用し、(d)に下位8ビット(バイト単位)、(d)+1に上位8ビット(バイト単位)を格納します。

例

(n)=6の場合

8ビット変換モードの場合、下記の網掛け部の6バイトが演算対象となります。CRC値は「BDA1H」となるため、「A1H」を(d)へ、「BDH」を(d)+1へ格納します。

	10進	16進	
		上位	下位
D0	24932	61H	64H
D1	4219	10H	7BH
D2	-1333	FAH	CBH
D3	-1	FFH	FFH
D4	32761	7FH	F9H
D5	10000	27H	10H

エラー

エラーコード (SD0)	内容
3285H	(s)、(d)にビットデバイスの桁指定で、4桁指定以外を指定したとき。

15 デバッグ, 故障診断

15.1 デバッグ, 故障診断命令

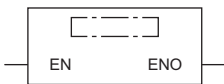
エラー表示または, アナンシェータリセット

LEDR

コントローラのアナンシェータ表示や運転を続行できる自己診断エラーの表示をリセットします。

ラダー	ST
	ENO:=LEDR(EN);

FBD/LD



■実行条件

命令	実行条件
LEDR	

機能

- コントローラのアナンシェータ表示や運転を続行できる自己診断エラーの表示をリセットします。1回の命令実行で、エラー表示とアナンシェータを両方リセットします。
- 自己診断エラー発生時の動作は、下記のとおりです。
 - 運転を続行できる自己診断エラーが発生しているとき。
コントローラが運転を続行できる自己診断エラーのエラー表示をしているときには、コントローラ前面のERROR LEDが消灯します。
このときSM0, SM1(最新自己診断エラー), SD0(最新自己診断エラーコード)の内容はリセットされませんので、プログラムにてリセットしてください。
 - バッテリーエラーが発生しているとき。
バッテリーを交換後にLEDR命令を実行すると、コントローラ前面のBATTERY LEDが消灯します。このときSM51(バッテリー低下ラッチ)もOFFされます。
- アナンシェータ(F)がONしているときの動作は、下記のとおりです。
 - USER LEDが消灯します。
 - SD62(アナンシェータ番号), SD63(アナンシェータ個数), SD64~79(アナンシェータ検出番号テーブル)をすべてクリアします。

実行前	実行後
SD62 200	SD62 0
SD63 15	SD63 0
SD64 200	SD64 0
SD65 99	SD65 0
SD66 5	SD66 0
SD67 255	SD67 0
⋮	⋮
SD78 83	SD78 0
SD79 0	SD79 0

(1): すべてクリアします。

エラー

演算エラーはありません。

16 文字列処理

16.1 文字列処理命令

文字列比較

LD\$□, AND\$□, OR\$□

文字列データをa接点扱いで比較演算します。

ラダー	ST
	ENO:=LDSTRING_□(EN,s1,s2); ENO:=ANDSTRING_□(EN,s1,s2); ENO:=ORSTRING_□(EN,s1,s2); (□には, EQ, NE, GT, LE, LT, GEが入ります。)*1

(□には, \$=, \$<>, \$>, \$<=, \$<, \$>=が入ります。)

FBD/LD



(□には, LDSTRING_, ANDSTRING_, ORSTRING_とEQ, NE, GT, LE, LT, GEの組合せが入ります。)*1

*1 EQは=, NEは<>, GTは>, LEは<=, LTは<, GEは>=です。

■実行条件

命令	実行条件
LD\$□, AND\$□, OR\$□	常時実行

設定データ

■内容, 範囲, データ型

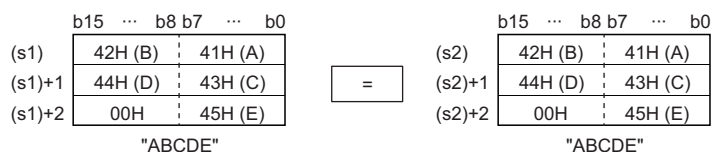
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	比較データまたは, 比較データが格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(s2)	比較データまたは, 比較データが格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC		K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	○	—
(s2)	—	—	○	—	—	—	○	—	—	○	—

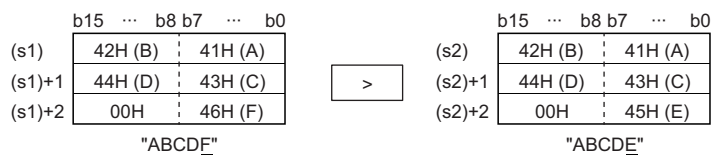
機能

- (s1)で指定された文字列データと、(s2)で指定された文字列データを、a接点扱いで比較演算します。
- 比較演算は、文字列のアスキーコードを文字列の最初から1文字ずつ比較します。
- (s1), (s2)の文字列は、指定されたデバイス番号から00Hが格納されているデバイス番号までが対象となります。
- すべての文字列が一致した場合に、比較結果が一致となります。



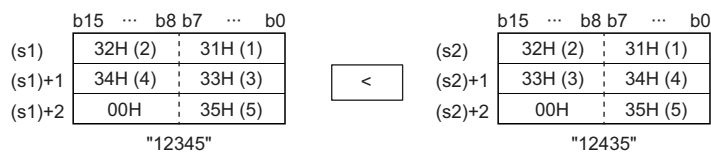
□内命令記号(ラダー, FBD/LD)	比較演算結果
\$=, EQ	導通状態(ENOはON)
\$<>, NE	非導通状態(ENOはOFF)
\$>, GT	非導通状態(ENOはOFF)
\$<=, LE	導通状態(ENOはON)
\$<, LT	非導通状態(ENOはOFF)
\$>=, GE	導通状態(ENOはON)

- 異なった文字列の場合は、キャラクターコードの大きい文字列が大きくなります。



□内命令記号(ラダー, FBD/LD)	比較演算結果
\$=, EQ	非導通状態(ENOはOFF)
\$<>, NE	導通状態(ENOはON)
\$>, GT	導通状態(ENOはON)
\$<=, LE	非導通状態(ENOはOFF)
\$<, LT	非導通状態(ENOはOFF)
\$>=, GE	導通状態(ENOはON)

- 異なった文字列の場合は、最初に異なったキャラクターコードの大小で、文字列の大小が決まります。



□内命令記号(ラダー, FBD/LD)	比較演算結果
\$=, EQ	非導通状態(ENOはOFF)
\$<>, NE	導通状態(ENOはON)
\$>, GT	非導通状態(ENOはOFF)
\$<=, LE	導通状態(ENOはON)
\$<, LT	導通状態(ENOはON)
\$>=, GE	非導通状態(ENOはOFF)

- (s1)と(s2)での文字列データの長さが異なる場合は、長い文字列データが大きくなります。

	b15	...	b8	b7	...	b0		b15	...	b8	b7	...	b0
(s1)	32H (2)					31H (1)	>	(s2)	32H (2)				31H (1)
(s1)+1	34H (4)					33H (3)		(s2)+1	34H (4)				33H (3)
(s1)+2	36H (6)					35H (5)		(s2)+2	36H (6)				35H (5)
(s1)+3	00H					37H (7)		(s2)+3	00H				00H
	"1234567"							"123456"					

□内命令記号(ラダー, FBD/LD)	比較演算結果
\$=, EQ	非導通状態(ENOはOFF)
\$<>, NE	導通状態(ENOはON)
\$>, GT	導通状態(ENOはON)
\$<=, LE	非導通状態(ENOはOFF)
\$<, LT	非導通状態(ENOはOFF)
\$>=, GE	導通状態(ENOはON)

- (s1), (s2)の文字列が16383文字を超える場合は、非導通(ENOはOFF)となります。
- LDSTRING_□命令をFBD/LDで使用した場合、ENに左母線、または常時ONの変数部品/定数部品を指定してください。
- ORSTRING_□命令をFBD/LDで使用した場合、ENにTRUEを指定するとENOはONします。ENは実行条件にはなりません。

エラー

演算エラーはありません。

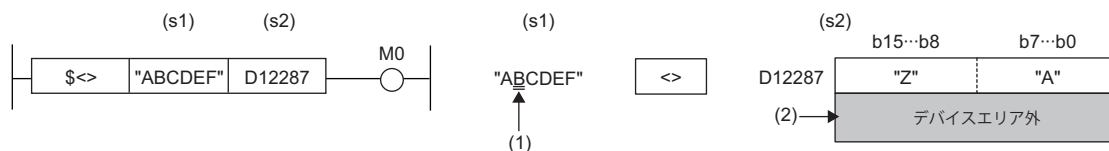
Point

文字列比較命令では、文字列の比較を行いながら、下記のチェックを行っています。

- デバイスエリアの範囲を超えたかのチェック
- 文字列が16383文字以内かのチェック

そのため、デバイスエリア内に00Hが存在しない場合、または文字列が16383文字を超える場合に、文字の不一致を検出すると、非導通(ENOはOFF)にならずに比較演算結果が出力されます。

下記の例は、デバイスエリアの最終デバイスNo.がD12287の場合の演算結果を示します。



(1) (s1)の2文字目が(s2)と異なり、(s1)≠(s2)となるので、演算結果は導通(ENOはON)になります。

(2) D12287以降はデバイスエリア外となるため、D12287までで文字列データ比較処理を実施します。文字列の不一致を検出しているので、条件が成立し、処理が終了します。

文字列の結合

\$+(P) [オペランドが2個の場合]

文字列データを連結します。

ラダー	ST
	対応していません。 (☞ 721ページ \$+(P) [オペランドが3個の場合])
FBD/LD	
対応していません。 (☞ 721ページ \$+(P) [オペランドが3個の場合])	

■実行条件

命令	実行条件
\$+	
\$+P	

設定データ

■内容、範囲、データ型

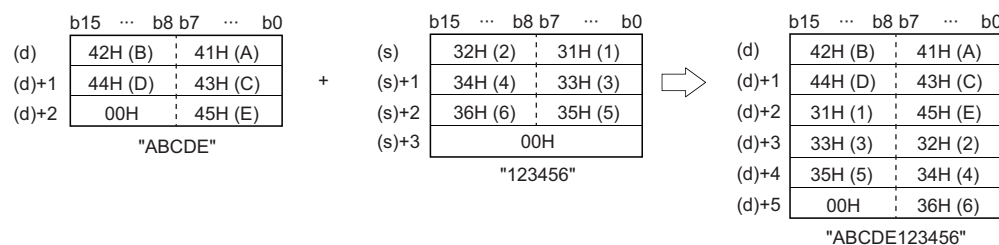
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	連結データまたは、データが格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	連結されるデータが格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	—	—	○	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	○	—	—	—	—

機能

- (d)で指定されたデバイス番号以降に格納されている文字列データの後ろに、(s)で指定されたデバイス番号以降に格納されている文字列を連結して、(d)で指定されたデバイス番号以降に格納します。



- 文字列の結合は、(d)で指定された文字列の終了を示す00Hを無視し、(d)の最終文字に続けて(s)で指定された文字列を連結します。

エラー

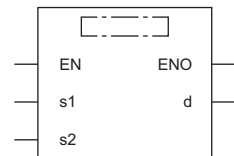
エラーコード (SD0)	内容
2820H	(s)で指定したデバイス番号以降，該当デバイス/ラベルの割付範囲に00Hが存在しないとき。
	(d)で指定したデバイス番号以降，該当デバイス/ラベルの割付範囲に00Hが存在しないとき。
2821H	(s)と(d)で指定された文字列の格納デバイス番号が重複しているとき。
3285H	(s)の文字列が16383文字を超えるとき。
	(d)の文字列が16383文字を超えるとき。
3286H	(s)+(d)の文字列が16383文字を超えるとき。
	(d)で指定されたデバイス番号以降，該当デバイス/ラベルの割付範囲に結合した文字列がすべて格納できないとき。

\$+(P) [オペランドが3個の場合]

文字列データを連結します。

ラダー	ST
	<pre>ENO:=STRINGPLUS(EN,s1,s2,d); ENO:=STRINGPLUSP(EN,s1,s2,d);</pre>

FBD/LD



(□には、STRINGPLUS、STRINGPLUSPが入ります。)

■実行条件

命令	実行条件
\$+	
\$+P	

設定データ

■内容、範囲、データ型

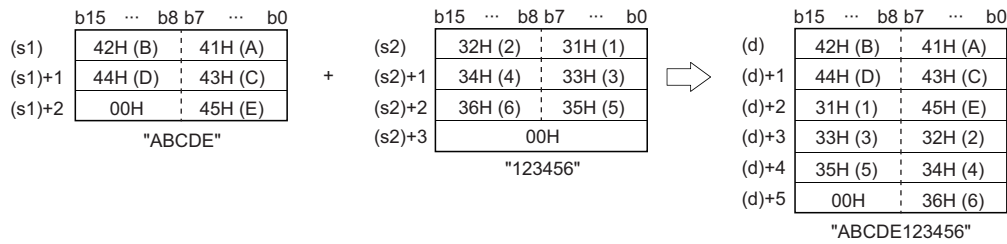
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	連結データまたは、データが格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(s2)	連結されるデータまたは、連結されるデータが格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	連結結果を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	○	—
(s2)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- (s1)で指定されたデバイス番号以降に格納されている文字列データの後ろに、(s2)で指定されたデバイス番号以降に格納されている文字列を連結して、(d)で指定されたデバイス番号以降に格納します。



- 文字列の結合は、(s1)で指定された文字列の終了を示す00Hを無視し、(s1)の最終文字に続けて(s2)で指定された文字列を連結します。

エラー

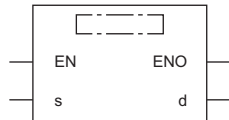
エラーコード (SD0)	内容
2820H	(s1)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に00Hが存在しないとき。 (s2)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に00Hが存在しないとき。
2821H	(s2)と(d)で指定された文字列の格納デバイス番号が重複しているとき。
3285H	(s1)の文字列が16383文字を超えるとき。 (s2)の文字列が16383文字を超えるとき。
3286H	(d)に格納する文字列が出力可能な範囲外のデータとなったとき。 • 結合した文字列が16383文字を超える • 指定されたデバイス番号以降、該当デバイス/ラベルの割付範囲に結合した文字列がすべて格納できない

\$MOV(P)

文字列データを、指定したデバイス番号以降に転送します。

ラダー	ST
	ENO:=STRINGMOV(EN,s,d); ENO:=STRINGMOVP(EN,s,d);

FBD/LD



(□には、STRINGMOV、STRINGMOVPが入ります。)

■実行条件

命令	実行条件
\$MOV	
\$MOVP	

設定データ

■内容、範囲、データ型

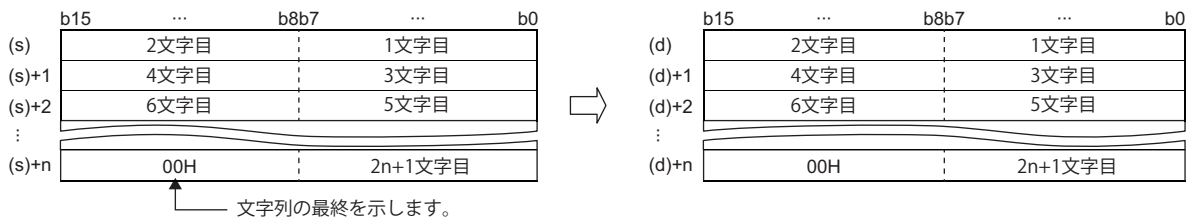
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	転送文字列(最大255文字)または、文字列が格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	転送文字列を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

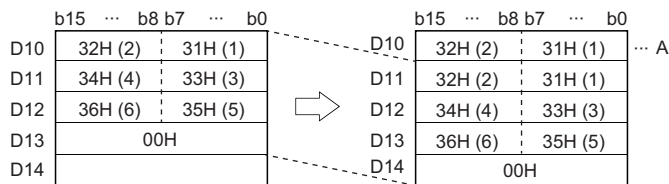
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□≠□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□≠G□, J□≠□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	—	○	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

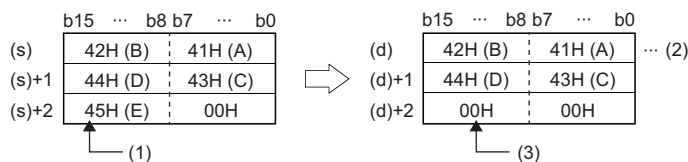
- ・ (s)で指定された文字列データを、(d)で指定されたデバイス番号以降に転送します。文字列の転送では、(s)で指定された文字列、またはデバイス番号から00Hが格納されているデバイス番号までの文字列を一度に転送します。



- ・ 転送する文字列データが格納されているデバイス範囲(s)~(s)+nと、転送された文字列データを格納するデバイス範囲(d)~(d)+nが重複している場合でも、正常に処理します。たとえば、D10~D13に格納されていた文字列をD11~D14に転送すると、下記ようになります。

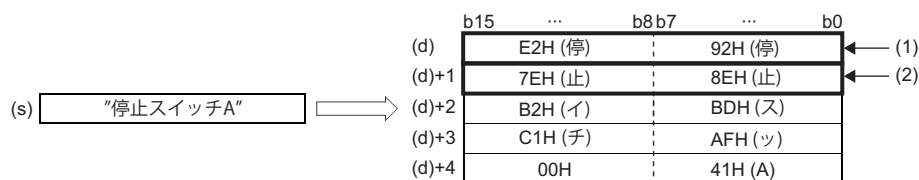


- ・ (s)+nの下位バイトに00Hが格納されている場合は、(d)+nの上位バイト、下位バイトともに00Hを格納します。



- (1): 上位バイトは転送されません。
- (2): 転送前の文字列のままになります。
- (3): 上位バイトには自動的に00Hを格納します。

- ・ (s)に漢字などの2バイトデータが指定された場合は、シフトJISコードに変換されます。\$MOV(P)命令を実行すると、(d)には上位バイトと下位バイトが逆に格納されます。



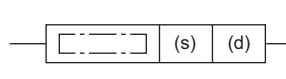
- (1) シフトJISコードは、「92E2H」になります。
- (2) シフトJISコードは、「8E7EH」になります。

エラー

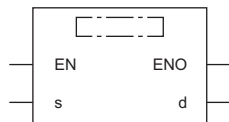
エラーコード (SD0)	内容
2820H	(s)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に00Hが存在しないとき。
3285H	(s)の文字列が16383文字を超えるとき。
3286H	(d)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に指定した文字列がすべて格納できないとき。

\$MOV(P)_WS

Unicode文字列データを，指定したデバイス番号以降に転送します。

ラダー	ST
	ENO:=STRINGMOV_WS(EN,s,d); ENO:=STRINGMOVP_WS(EN,s,d);

FBD/LD



(□には，STRINGMOV_WS，STRINGMOVP_WSが入ります。)

■実行条件

命令	実行条件
\$MOV_WS	
\$MOVP_WS	

設定データ

■内容，範囲，データ型

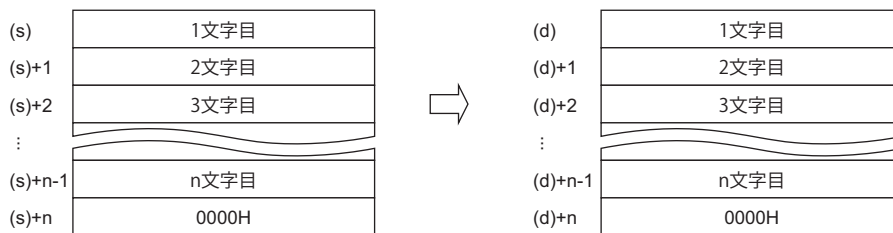
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	転送Unicode文字列(最大255文字)または，Unicode文字列が格納されている先頭デバイス	—	Unicode文字列	ANYSTRING_DOUBLE
(d)	転送Unicode文字列を格納する先頭デバイス	—	Unicode文字列	ANYSTRING_DOUBLE
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	—	—	○	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—	—

機能

- (s)で指定されたUnicode文字列データを、(d)で指定されたデバイス番号以降に転送します。Unicode文字列の転送では、(s)で指定されたUnicode文字列、または、デバイス番号から0000Hが格納されているデバイス番号までのUnicode文字列を一度に転送します。



- 転送するUnicode文字列データが格納されているデバイス範囲(s)~(s)+nと、転送された文字列データを格納するデバイス範囲(d)~(d)+nが重複している場合でも、正常に処理します。たとえば、D10~D13に格納されていた文字列をD11~D14に転送すると、下記ようになります。



A: 転送前の値の状態

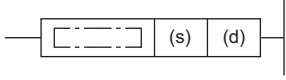
エラー

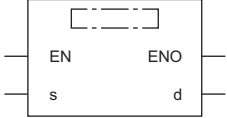
エラーコード (SD0)	内容
2820H	(s)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に0000Hが存在しないとき。
3285H	(s)のUnicode文字列が16383文字を超えるととき。
3286H	(d)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に指定したUnicode文字列がすべて格納できないとき。

BIN16ビットデータ→10進アスキー変換

BINDA(P)(_U)

BIN16ビットデータを，10進数アスキーコードに変換します。

ラダー	ST
	ENO:=BINDA(EN,s,d); ENO:=BINDAP(EN,s,d)
	ENO:=BINDA_U(EN,s,d); ENO:=BINDAP_U(EN,s,d)

FBD/LD


■実行条件

命令	実行条件
BINDA BINDA_U	
BINDAP BINDAP_U	

設定データ

■内容，範囲，データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(s)	BINDA(P)	アスキー変換を行うBINデータ	-32768~32767	符号付きBIN16ビット	ANY16_S
	BINDA(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
(d)		変換結果を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□≠□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□≠G□, J□≠□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- (s)で指定されたBIN16ビットデータを，10進数で表現したときの数値をアスキーコードに変換し，(d)で指定されたデバイス番号以降に格納します。
- SM705(変換桁数切換)の状態によって，(d)に格納される10進アスキーデータの形式が異なります。

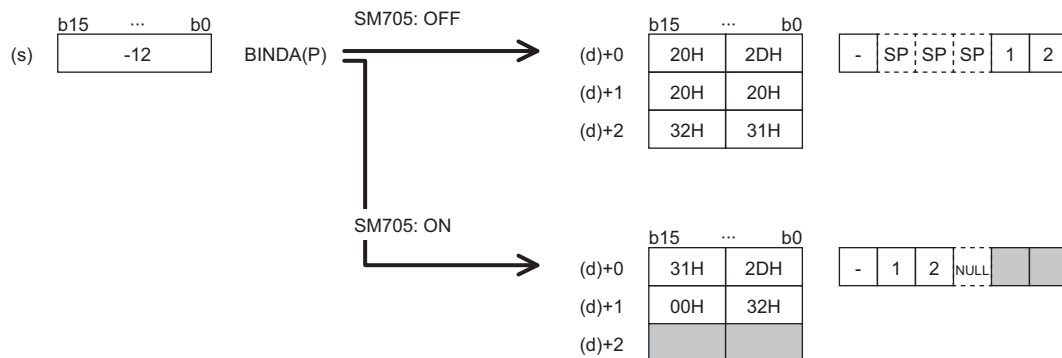
SM705の状態	(d)の格納形式	参照先
OFF	固定桁数(符号+5桁)で格納	729ページ SM705(変換桁数切換)がOFFの場合の動作
ON	(s)の値によって各桁を前詰めで格納	730ページ SM705(変換桁数切換)がONの場合の動作

■動作の概要

SM705(変換桁数切換)がOFFの場合とONの場合のそれぞれについての動作を下記に示します。

例

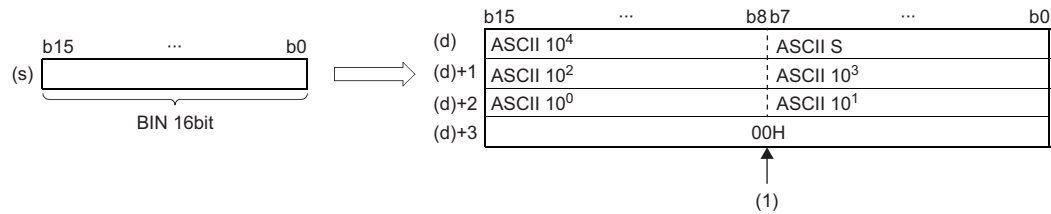
(s)に数値「-12」が格納されている状態で，BINDA(P)命令を実行した場合



- SM705(変換桁数切換)がOFFの場合は固定桁数となります。1文字目は符号で，上記の例では2DH(-)となります。(s)が0または正の数の場合は，1文字目は20H(スペース)となります。)数値部は右詰めで格納されます。数値部が5桁に満たない場合，上位桁のアスキーコードは20H(スペース)となります。
- SM705(変換桁数切換)がONの場合は前詰めとなります。数値部が5桁に満たない場合，終端に00Hが格納されます。

■SM705(変換桁数切換)がOFFの場合の動作

(d)~(d)+2には、10進アスキーデータが固定桁数で格納されます。



ASCII S: 符号のアスキーコード^{*1}

ASCII 10⁴: 万の位のアスキーコード^{*2}

ASCII 10³: 千の位のアスキーコード^{*3}

ASCII 10²: 百の位のアスキーコード^{*4}

ASCII 10¹: 十の位のアスキーコード^{*5}

ASCII 10⁰: 一の位のアスキーコード

(1): (d)+3には、SM701(出力文字数切換)がOFFのときは00Hが格納され、ONのときは変化しません。

*1 0または正の数の場合は20H(スペース)が格納され、負の数の場合は2DH(-)が格納されます。

*2 数値部が4桁以下の場合、ASCII 10⁴には20H(スペース)が格納されます。

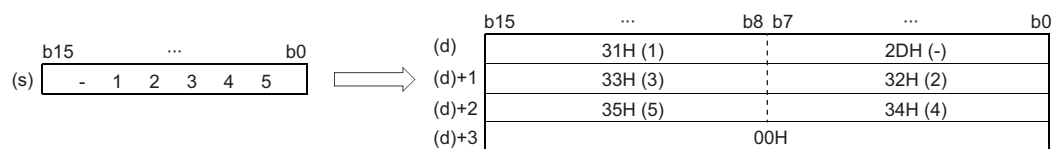
*3 数値部が3桁以下の場合、ASCII 10³には20H(スペース)が格納されます。

*4 数値部が2桁以下の場合、ASCII 10²には20H(スペース)が格納されます。

*5 数値部が1桁の場合、ASCII 10¹には20H(スペース)が格納されます。

例

BINDA(P)命令において、(s)に-12345を指定した場合



■SM705(変換桁数切換)がONの場合の動作

(d)には、10進アスキーデータが前詰めで格納されます。
(s)の値と(d)に格納される値の例を下記に示します。

(s)の値	(d)~(d)+2の内容	(s)の値	(d)~(d)+2の内容																																													
0 - 正の数(数値部1桁)	(d)の上位バイトに00Hが格納されます。 (d)+1~(d)+2は変化しません。 <table><tr><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(d)</td><td colspan="2">00H</td><td>:</td><td>ASCII 10⁰</td></tr><tr><td>(d)+1</td><td colspan="4"></td></tr><tr><td>(d)+2</td><td colspan="4"></td></tr></table>	b15	...	b8 b7	...	b0	(d)	00H		:	ASCII 10 ⁰	(d)+1					(d)+2					- 正の数(数値部2桁) - 負の数(数値部1桁)	(d)+1に00Hが格納されます。 (d)+2は変化しません。 <table><tr><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(d)</td><td colspan="2">ASCII 10⁰</td><td>:</td><td>ASCII 10¹ / 2DH (-)</td></tr><tr><td>(d)+1</td><td colspan="4">00H</td></tr><tr><td>(d)+2</td><td colspan="4"></td></tr></table>	b15	...	b8 b7	...	b0	(d)	ASCII 10 ⁰		:	ASCII 10 ¹ / 2DH (-)	(d)+1	00H				(d)+2									
b15	...	b8 b7	...	b0																																												
(d)	00H		:	ASCII 10 ⁰																																												
(d)+1																																																
(d)+2																																																
b15	...	b8 b7	...	b0																																												
(d)	ASCII 10 ⁰		:	ASCII 10 ¹ / 2DH (-)																																												
(d)+1	00H																																															
(d)+2																																																
⋮																																																
- 正の数(数値部5桁) - 負の数(数値部4桁)	(d)+2の上位バイトに00Hが格納されます。 <table><tr><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(d)</td><td colspan="2">ASCII 10³</td><td>:</td><td>ASCII 10⁴ / 2DH (-)</td></tr><tr><td>(d)+1</td><td colspan="2">ASCII 10¹</td><td>:</td><td>ASCII 10²</td></tr><tr><td>(d)+2</td><td colspan="2">00H</td><td>:</td><td>ASCII 10⁰</td></tr></table>	b15	...	b8 b7	...	b0	(d)	ASCII 10 ³		:	ASCII 10 ⁴ / 2DH (-)	(d)+1	ASCII 10 ¹		:	ASCII 10 ²	(d)+2	00H		:	ASCII 10 ⁰	負の数(数値部5桁)	(1): (d)+3にはSM701(出力文字数切換)がOFF時のみ00Hが格納されます。 <table><tr><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(d)</td><td colspan="2">ASCII 10⁴</td><td>:</td><td>2DH (-)</td></tr><tr><td>(d)+1</td><td colspan="2">ASCII 10²</td><td>:</td><td>ASCII 10³</td></tr><tr><td>(d)+2</td><td colspan="2">ASCII 10⁰</td><td>:</td><td>ASCII 10¹</td></tr><tr><td>(d)+3</td><td colspan="4">00H</td></tr></table> (1) ↑	b15	...	b8 b7	...	b0	(d)	ASCII 10 ⁴		:	2DH (-)	(d)+1	ASCII 10 ²		:	ASCII 10 ³	(d)+2	ASCII 10 ⁰		:	ASCII 10 ¹	(d)+3	00H			
b15	...	b8 b7	...	b0																																												
(d)	ASCII 10 ³		:	ASCII 10 ⁴ / 2DH (-)																																												
(d)+1	ASCII 10 ¹		:	ASCII 10 ²																																												
(d)+2	00H		:	ASCII 10 ⁰																																												
b15	...	b8 b7	...	b0																																												
(d)	ASCII 10 ⁴		:	2DH (-)																																												
(d)+1	ASCII 10 ²		:	ASCII 10 ³																																												
(d)+2	ASCII 10 ⁰		:	ASCII 10 ¹																																												
(d)+3	00H																																															

ASCII 10⁴: 万の位のアスキーコード
ASCII 10³: 千の位のアスキーコード
ASCII 10²: 百の位のアスキーコード
ASCII 10¹: 十の位のアスキーコード
ASCII 10⁰: 一の位のアスキーコード

- 演算桁数が最大桁数(符号+5桁)未満の場合、SM701(出力文字数切換)のON/OFFに関わらず、文字列の終端に00Hを格納します。文字列の終端が下位バイトの場合、上位バイトにも00Hを格納します。
- 演算桁数が最大桁数(符号+5桁)の場合、SM701(出力文字数切換)がOFFのときは(d)+3に00Hを格納します。SM701(出力文字数切換)がONのときは(d)+3は変化しません。

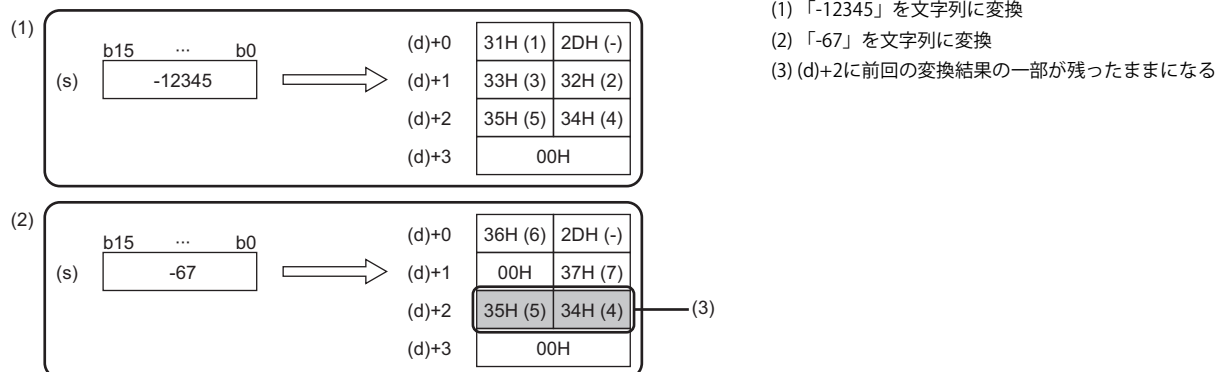
エラー

演算エラーはありません。

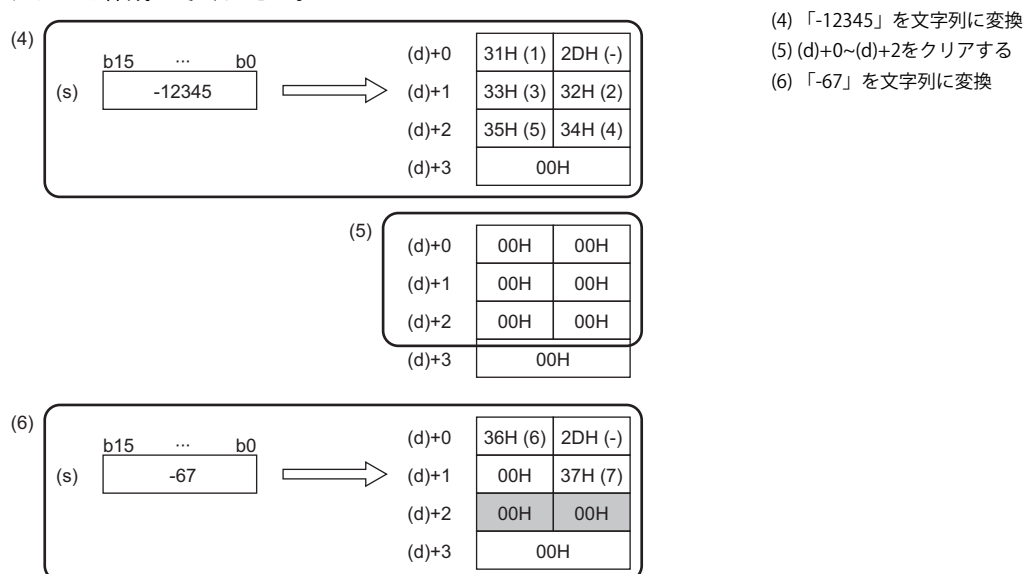
注意事項

SM705(変換桁数切換)がONのとき、(d)には演算結果を有効桁数分だけ格納します。このため、BINDA(P)(_U)命令を連続して実行し、演算結果を同一デバイスに格納する場合、(d)の一部に前回の演算結果が上書きされずに残っている可能性があります。

[例] (s)が「-12345」のときにBINDA(P)命令を実行し、さらに(s)が「-67」のときにBINDA(P)命令を実行する例



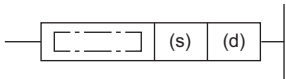
これを回避したい場合は、格納エリア(d)+0~(d)+2をすべてクリアしておいてからBINDA(P)(_U)命令を実行するようにプログラムを作成してください。



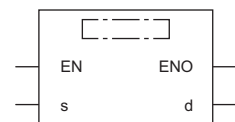
BIN32ビットデータ→10進アスキー変換

DBINDA(P)(_U)

BIN32ビットデータを，10進数アスキーコードに変換します。

ラダー	ST	
	ENO:=DBINDA(EN,s,d); ENO:=DBINDAP(EN,s,d);	ENO:=DBINDA_U(EN,s,d); ENO:=DBINDAP_U(EN,s,d);

FBD/LD



■実行条件

命令	実行条件
DBINDA DBINDA_U	
DBINDAP DBINDAP_U	

設定データ

■内容，範囲，データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(s)	DBINDA(P)	アスキー変換を行うBINデータ	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	DBINDA(P)_U		0~4294967295	符号なしBIN32ビット	ANY32_U
(d)		変換結果を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- (s)で指定されたBIN32ビットデータを，10進数で表現したときの数値をアスキーコードに変換し，(d)で指定されたデバイス番号以降に格納します。
- SM705(変換桁数切換)の状態によって，(d)に格納される10進アスキーデータの形式が異なります。

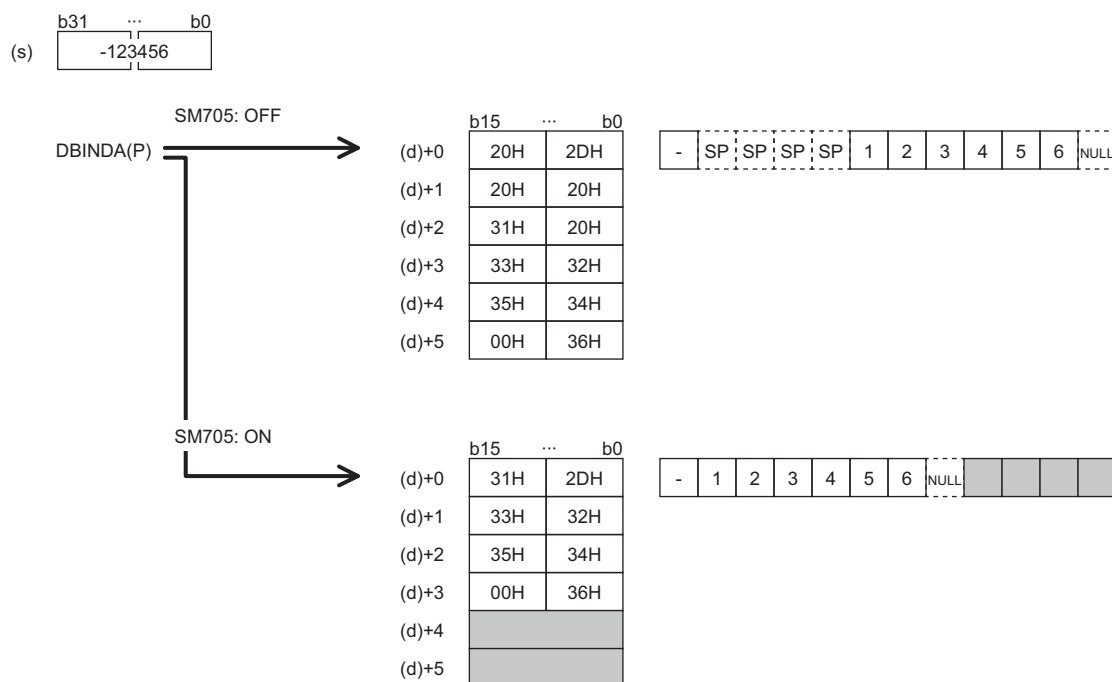
SM705の状態	(d)の格納形式	参照先
OFF	固定桁数(符号+10桁)で格納	734ページ SM705(変換桁数切換)がOFFの場合の動作
ON	(s)の値によって各桁を前詰めで格納	735ページ SM705(変換桁数切換)がONの場合の動作

■動作の概要

SM705(変換桁数切換)がOFFの場合とONの場合のそれぞれについての動作を下記に示します。

例

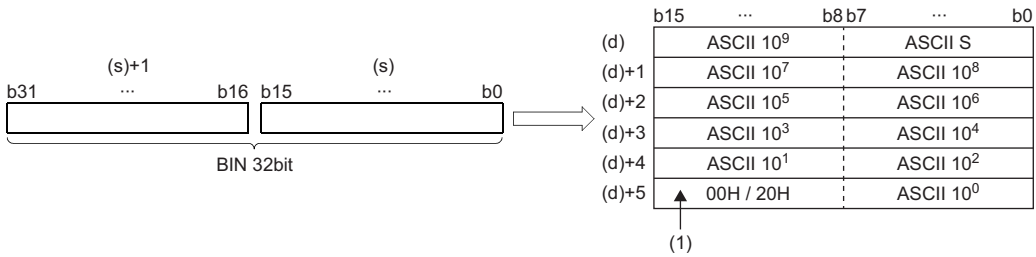
(s)に数値「-123456」が格納されている状態で，DBINDA(P)命令を実行した場合



- SM705(変換桁数切換)がOFFの場合は固定桁数となります。1文字目は符号で，上記の例では2DH(-)となります。(s)が0または正の数の場合は，1文字目は20H(スペース)となります。)数値部は右詰めで格納されます。数値部が10桁に満たない場合，上位桁のアスキーコードは20H(スペース)となります。
- SM705(変換桁数切換)がONの場合は前詰めとなります。数値部が10桁に満たない場合，終端に00Hが格納されます。

■SM705(変換桁数切換)がOFFの場合の動作

(d)~(d)+5には、10進アスキーデータが固定桁数で格納されます。



ASCII S: 符号のアスキーコード*1

ASCII 10⁹: 十億の位のアスキーコード*2

ASCII 10⁸: 億の位のアスキーコード*3

⋮

ASCII 10¹: 十の位のアスキーコード*4

ASCII 10⁰: 一の位のアスキーコード

(1): (d)+5の上位バイトには、SM701(出力文字数切換)がOFFのときは00Hが格納され、ONのときは20H(スペース)が格納されます。

*1 0または正の数の場合は20H(スペース)が格納され、負の数の場合は2DH(-)が格納されます。

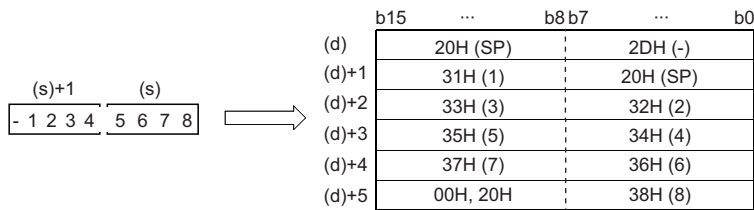
*2 数値部が9桁以下の場合、ASCII 10⁹には20H(スペース)が格納されます。

*3 数値部が8桁以下の場合、ASCII 10⁸には20H(スペース)が格納されます。

*4 数値部が1桁の場合、ASCII 10¹には20H(スペース)が格納されます。

例

(s)に-12345678を指定した場合(符号付きを指定した場合)



■SM705(変換桁数切換)がONの場合の動作

(d)には、10進アスキーデータが前詰めで格納されます。

(s)の値と(d)に格納される値の例を下記に示します。

(s)の値	(d)~(d)+2の内容	(s)の値	(d)~(d)+2の内容																																																																																				
0 - 正の数(数値部1桁)	(d)の上位バイトに00Hが格納されます。 (d)+1以降は変化しません。 <table><tr><td></td><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(d)</td><td colspan="2">00H</td><td colspan="3">ASCII 10⁰</td></tr><tr><td>(d)+1</td><td colspan="5"></td></tr><tr><td>(d)+2</td><td colspan="5"></td></tr><tr><td>(d)+3</td><td colspan="5"></td></tr><tr><td>(d)+4</td><td colspan="5"></td></tr><tr><td>(d)+5</td><td colspan="5"></td></tr></table>		b15	...	b8 b7	...	b0	(d)	00H		ASCII 10 ⁰			(d)+1						(d)+2						(d)+3						(d)+4						(d)+5						- 正の数(数値部2桁) - 負の数(数値部1桁)	(d)+1に00Hが格納されます。 (d)+2以降は変化しません。 <table><tr><td></td><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(d)</td><td colspan="2">ASCII 10⁰</td><td colspan="3">ASCII 10¹ / 2DH (-)</td></tr><tr><td>(d)+1</td><td colspan="5">00H</td></tr><tr><td>(d)+2</td><td colspan="5"></td></tr><tr><td>(d)+3</td><td colspan="5"></td></tr><tr><td>(d)+4</td><td colspan="5"></td></tr><tr><td>(d)+5</td><td colspan="5"></td></tr></table>		b15	...	b8 b7	...	b0	(d)	ASCII 10 ⁰		ASCII 10 ¹ / 2DH (-)			(d)+1	00H					(d)+2						(d)+3						(d)+4						(d)+5					
	b15	...	b8 b7	...	b0																																																																																		
(d)	00H		ASCII 10 ⁰																																																																																				
(d)+1																																																																																							
(d)+2																																																																																							
(d)+3																																																																																							
(d)+4																																																																																							
(d)+5																																																																																							
	b15	...	b8 b7	...	b0																																																																																		
(d)	ASCII 10 ⁰		ASCII 10 ¹ / 2DH (-)																																																																																				
(d)+1	00H																																																																																						
(d)+2																																																																																							
(d)+3																																																																																							
(d)+4																																																																																							
(d)+5																																																																																							
:																																																																																							
- 正の数(数値部9桁) - 負の数(数値部8桁)	(d)+4の上位バイトに00Hが格納されます。 (d)+5以降は変化しません。 <table><tr><td></td><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(d)</td><td colspan="2">ASCII 10⁷</td><td colspan="3">ASCII 10⁸ / 2DH (-)</td></tr><tr><td>(d)+1</td><td colspan="2">ASCII 10⁵</td><td colspan="3">ASCII 10⁶</td></tr><tr><td>(d)+2</td><td colspan="2">ASCII 10³</td><td colspan="3">ASCII 10⁴</td></tr><tr><td>(d)+3</td><td colspan="2">ASCII 10¹</td><td colspan="3">ASCII 10²</td></tr><tr><td>(d)+4</td><td colspan="2">00H</td><td colspan="3">ASCII 10⁰</td></tr><tr><td>(d)+5</td><td colspan="5"></td></tr></table>		b15	...	b8 b7	...	b0	(d)	ASCII 10 ⁷		ASCII 10 ⁸ / 2DH (-)			(d)+1	ASCII 10 ⁵		ASCII 10 ⁶			(d)+2	ASCII 10 ³		ASCII 10 ⁴			(d)+3	ASCII 10 ¹		ASCII 10 ²			(d)+4	00H		ASCII 10 ⁰			(d)+5						- 正の数(数値部10桁) - 負の数(数値部9桁)	(d)+5に00Hが格納されます。 <table><tr><td></td><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(d)</td><td colspan="2">ASCII 10⁸</td><td colspan="3">ASCII 10⁹ / 2DH (-)</td></tr><tr><td>(d)+1</td><td colspan="2">ASCII 10⁶</td><td colspan="3">ASCII 10⁷</td></tr><tr><td>(d)+2</td><td colspan="2">ASCII 10⁴</td><td colspan="3">ASCII 10⁵</td></tr><tr><td>(d)+3</td><td colspan="2">ASCII 10²</td><td colspan="3">ASCII 10³</td></tr><tr><td>(d)+4</td><td colspan="2">ASCII 10⁰</td><td colspan="3">ASCII 10¹</td></tr><tr><td>(d)+5</td><td colspan="5">00H</td></tr></table>		b15	...	b8 b7	...	b0	(d)	ASCII 10 ⁸		ASCII 10 ⁹ / 2DH (-)			(d)+1	ASCII 10 ⁶		ASCII 10 ⁷			(d)+2	ASCII 10 ⁴		ASCII 10 ⁵			(d)+3	ASCII 10 ²		ASCII 10 ³			(d)+4	ASCII 10 ⁰		ASCII 10 ¹			(d)+5	00H				
	b15	...	b8 b7	...	b0																																																																																		
(d)	ASCII 10 ⁷		ASCII 10 ⁸ / 2DH (-)																																																																																				
(d)+1	ASCII 10 ⁵		ASCII 10 ⁶																																																																																				
(d)+2	ASCII 10 ³		ASCII 10 ⁴																																																																																				
(d)+3	ASCII 10 ¹		ASCII 10 ²																																																																																				
(d)+4	00H		ASCII 10 ⁰																																																																																				
(d)+5																																																																																							
	b15	...	b8 b7	...	b0																																																																																		
(d)	ASCII 10 ⁸		ASCII 10 ⁹ / 2DH (-)																																																																																				
(d)+1	ASCII 10 ⁶		ASCII 10 ⁷																																																																																				
(d)+2	ASCII 10 ⁴		ASCII 10 ⁵																																																																																				
(d)+3	ASCII 10 ²		ASCII 10 ³																																																																																				
(d)+4	ASCII 10 ⁰		ASCII 10 ¹																																																																																				
(d)+5	00H																																																																																						
負の数(数値部10桁)	(d)+5の上位バイト(1)には、SM701(出力文字数切換)がOFFのときは00Hが、ONのときは20H(スペース)が格納されます。 <table><tr><td></td><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(d)</td><td colspan="2">ASCII 10⁹</td><td colspan="3">2DH (-)</td></tr><tr><td>(d)+1</td><td colspan="2">ASCII 10⁷</td><td colspan="3">ASCII 10⁸</td></tr><tr><td>(d)+2</td><td colspan="2">ASCII 10⁵</td><td colspan="3">ASCII 10⁶</td></tr><tr><td>(d)+3</td><td colspan="2">ASCII 10³</td><td colspan="3">ASCII 10⁴</td></tr><tr><td>(d)+4</td><td colspan="2">ASCII 10¹</td><td colspan="3">ASCII 10²</td></tr><tr><td>(d)+5</td><td colspan="2">↑ 00H / 20H</td><td colspan="3">ASCII 10⁰</td></tr></table> (1)		b15	...	b8 b7	...	b0	(d)	ASCII 10 ⁹		2DH (-)			(d)+1	ASCII 10 ⁷		ASCII 10 ⁸			(d)+2	ASCII 10 ⁵		ASCII 10 ⁶			(d)+3	ASCII 10 ³		ASCII 10 ⁴			(d)+4	ASCII 10 ¹		ASCII 10 ²			(d)+5	↑ 00H / 20H		ASCII 10 ⁰			ASCII 10 ⁹ : 十億の位のアスキーコード ASCII 10 ⁸ : 一億の位のアスキーコード : ASCII 10 ¹ : 十の位のアスキーコード ASCII 10 ⁰ : 一の位のアスキーコード																																											
	b15	...	b8 b7	...	b0																																																																																		
(d)	ASCII 10 ⁹		2DH (-)																																																																																				
(d)+1	ASCII 10 ⁷		ASCII 10 ⁸																																																																																				
(d)+2	ASCII 10 ⁵		ASCII 10 ⁶																																																																																				
(d)+3	ASCII 10 ³		ASCII 10 ⁴																																																																																				
(d)+4	ASCII 10 ¹		ASCII 10 ²																																																																																				
(d)+5	↑ 00H / 20H		ASCII 10 ⁰																																																																																				

- 演算桁数が最大桁数(符号+10桁)未満の場合、SM701(出力文字数切換)のON/OFFに関わらず、文字列の終端に00Hを格納します。文字列の終端が下位バイトの場合、上位バイトにも00Hを格納します。
- 演算桁数が最大桁数(符号+10桁)の場合、SM701(出力文字数切換)がOFFのときは(d)+5の上位バイトに00Hを格納します。SM701(出力文字数切換)がONのときは(d)+5の上位バイトに20H(スペース)を格納します。

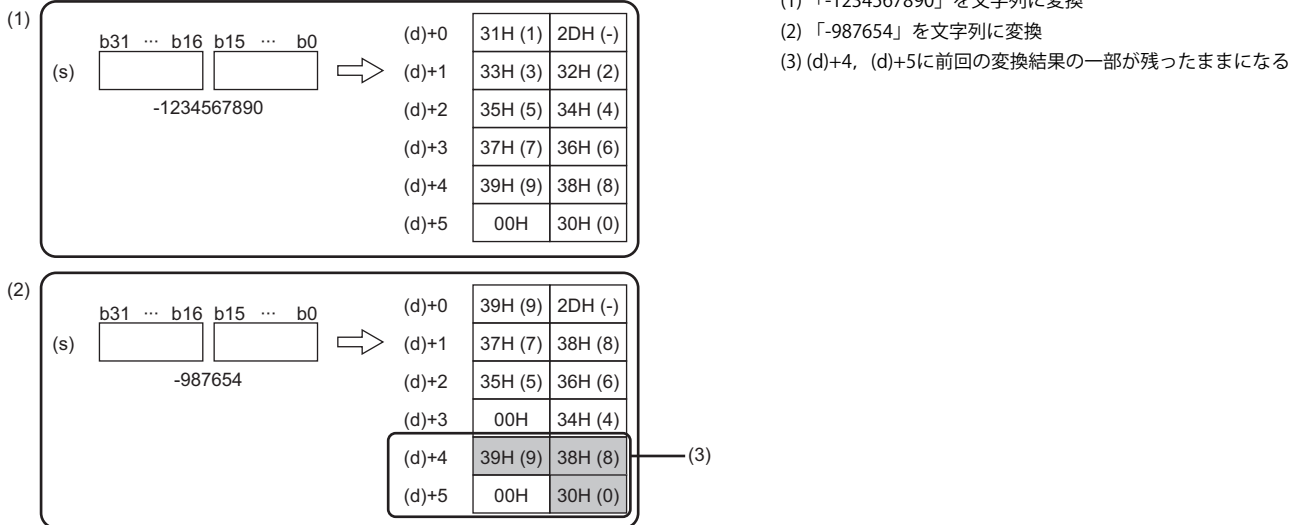
エラー

演算エラーはありません。

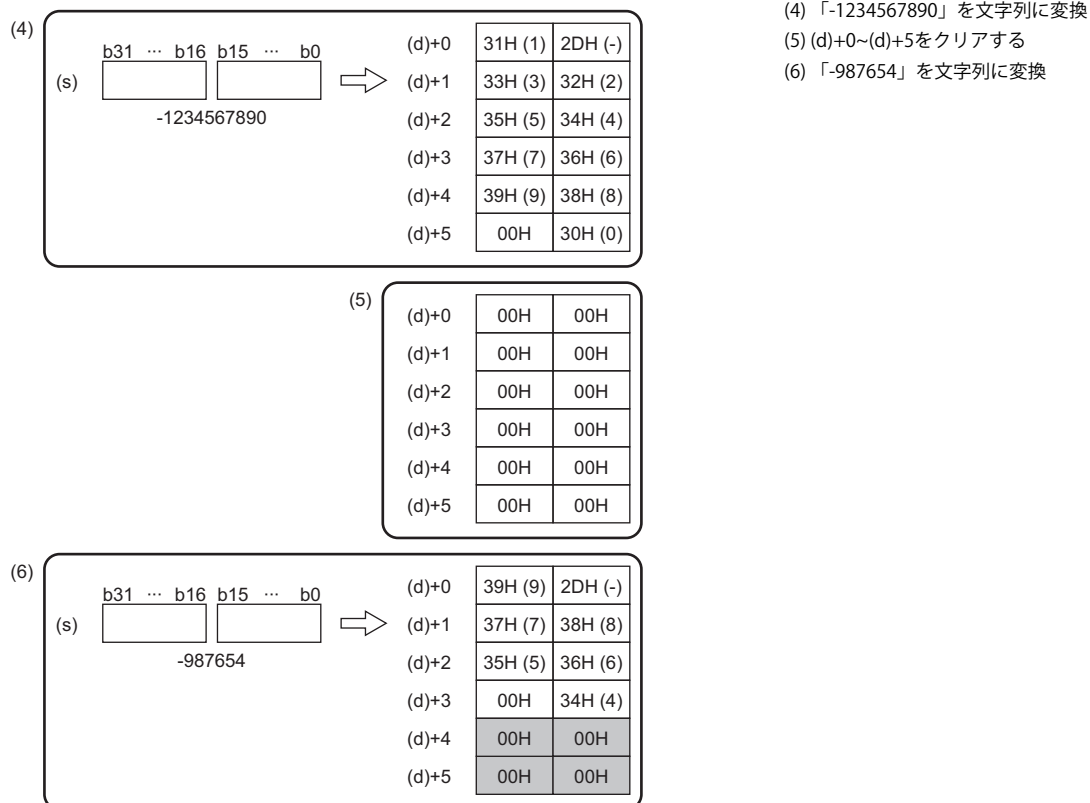
注意事項

SM705(変換桁数切換)がONのとき、(d)には演算結果を有効桁数分だけ格納します。このため、DBINDA(P)(_U)命令を連続して実行し、演算結果を同一デバイスに格納する場合、(d)の一部に前回の演算結果が上書きされずに残っている可能性があります。

[例] (s)が「-1234567890」のときにDBINDA(P)命令を実行し、さらに(s)が「-987654」のときにDBINDA(P)命令を実行する例



これを回避したい場合は、格納エリア(d)+0~(d)+5をすべてクリアしておいてからDBINDA(P)(_U)命令を実行するようにプログラムを作成してください。



BIN16ビットデータ→16進アスキー変換

BINHA(P)

BIN16ビットデータを，16進数アスキーコードに変換します。

ラダー	ST
	ENO:=BINHA(EN,s,d); ENO:=BINHAP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
BINHA	
BINHAP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	アスキー変換を行うBINデータ	-32768~32767	符号付きBIN16ビット	ANY16
(d)	変換結果を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□≠□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□≠G□, J□≠□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- (s)で指定されたBIN16ビットデータを，16進数で表現したときの数値をアスキーコードに変換し，(d)で指定されたデバイス番号以降に格納します。
- SM705(変換桁数切換)の状態によって，(d)に格納される16進アスキーデータの形式が異なります。

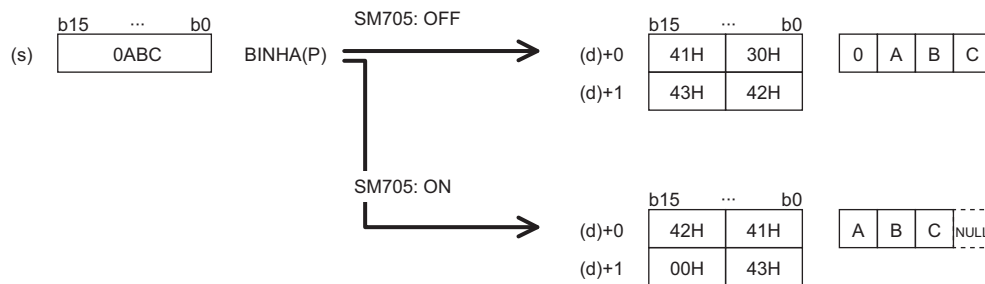
SM705の状態	(d)の格納形式	参照先
OFF	固定桁数(4桁)で格納	739ページ SM705(変換桁数切換)がOFFの場合の動作
ON	(s)の値によって各桁を前詰めで格納	739ページ SM705(変換桁数切換)がONの場合の動作

■動作の概要

SM705(変換桁数切換)がOFFの場合とONの場合のそれぞれについての動作を下記に示します。

例

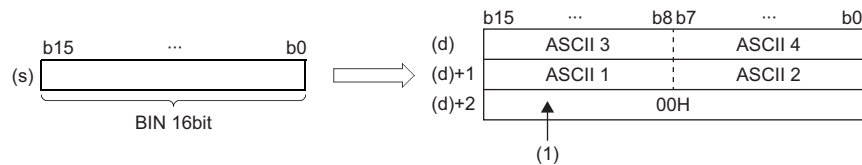
(s)にBIN16ビットデータ「0ABCH」が格納されている状態で，BINHA(P)命令を実行した場合



- SM705(変換桁数切換)がOFFの場合は固定桁数となります。「0ABC」の4桁がアスキーデータに変換されて格納されます。
- SM705(変換桁数切換)がONの場合は前詰めとなります。「0ABC」の先頭の0を省略した「ABC」がアスキーデータに変換されて格納され，終端に00Hが格納されます。

■SM705(変換桁数切換)がOFFの場合の動作

(d)には、16進アスキーデータが固定桁数(4桁)で格納されます。



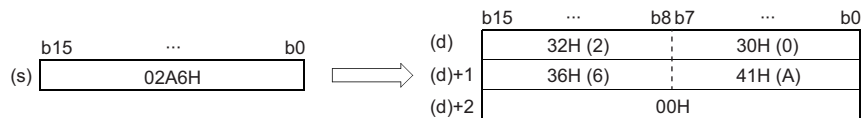
ASCII 口: 口桁目のアスキーコード

(1): (d)+2には、SM701(出力文字数切換)がOFFのときは00Hが格納され、ONのときは変化しません。

- (d)に格納する演算結果は、4桁の16進数として処理します。したがって、有効桁数の左側の0は「0」として処理します。(0パディングします。)

例

(s)に02A6Hを指定した場合



■SM705(変換桁数切換)がONの場合の動作

(d)には、有効桁数の左側の0を省略した桁数(最大4桁)の16進アスキーデータが前詰めで格納されます。

(s)の値と(d)に格納される値の例を下記に示します。

(s)の値	(d)~(d)+1の内容	(s)の値	(d)~(d)+1の内容
• 0H~FH	• (d)の上位バイトに00Hが格納されます。 • (d)+1以降は変化しません。	• 10H~FFH	• (d)+1に00Hが格納されます。
• 100H~FFFH	• (d)+1の上位バイトに00Hが格納されます。	• 1000H~FFFFH	• (1): SM701(出力文字数切換)がOFFのときは(d)+2に00Hが格納されます。SM701(出力文字数切換)がONのときは(d)+2は変化しません。

ASCII 口: 口桁目のアスキーコード

- 桁数が最大桁数(4桁)未満の場合、SM701(出力文字数切換)のON/OFFに関わらず、文字列の終端に00Hを格納します。文字列の終端が下位バイトの場合、上位バイトにも00Hを格納します。
- 桁数が最大桁数(4桁)の場合、SM701(出力文字数切換)がOFFのときは(d)+2に00Hを格納します。SM701(出力文字数切換)がONのときは(d)+2は変化しません。

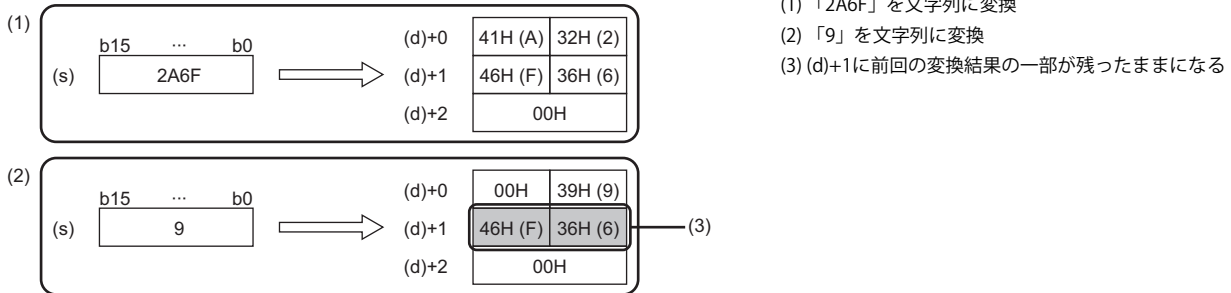
エラー

演算エラーはありません。

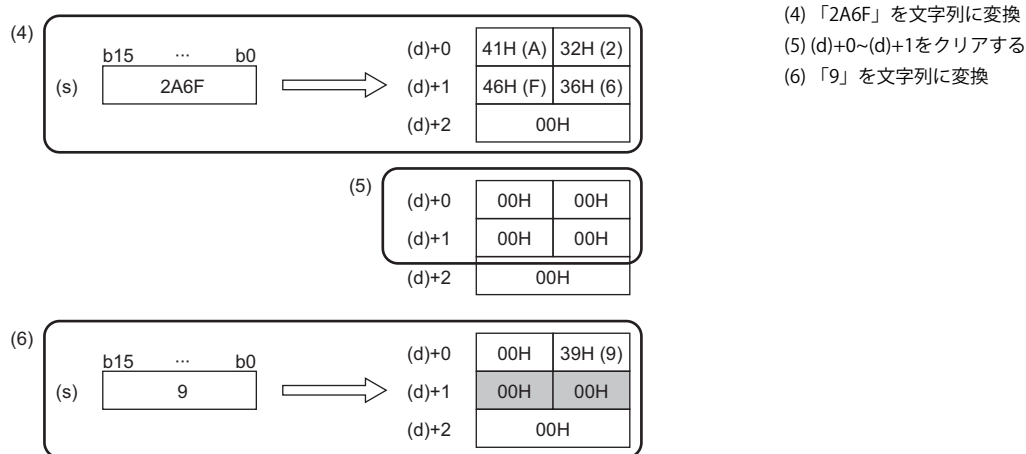
注意事項

SM705(変換桁数切換)がONのとき、(d)には演算結果を有効桁数分だけ格納します。このため、BINHA(P)命令を連続して実行し、演算結果を同一デバイスに格納する場合、(d)の一部に前回の演算結果が上書きされずに残っている可能性があります。

[例] (s)が「2A6F」のときにBINHA(P)命令を実行し、さらに(s)が「9」のときにBINHA(P)命令を実行する例



これを回避したい場合は、格納エリア(d)+0~(d)+1をすべてクリアしておいてからBINHA(P)命令を実行するようにプログラムを作成してください。



BIN32ビットデータ→16進アスキー変換

DBINHA(P)

BIN32ビットデータを，16進数アスキーコードに変換します。

ラダー	ST
	ENO:=DBINHA(EN,s,d); ENO:=DBINHAP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
DBINHA	
DBINHAP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	アスキー変換を行うBINデータ	-2147483648~2147483647	符号付きBIN32ビット	ANY32
(d)	変換結果を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□≠□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□≠G□, J□≠□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- (s)で指定されたBIN32ビットデータを，16進数で表現したときの数値をアスキーコードに変換し，(d)で指定されたデバイス番号以降に格納します。
- SM705(変換桁数切換)の状態によって，(d)に格納される10進アスキーデータの形式が異なります。

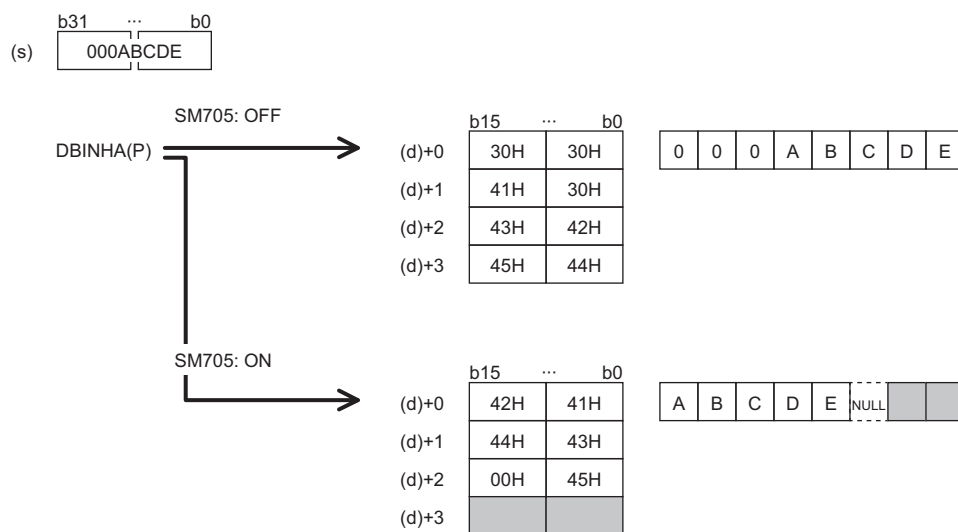
SM705の状態	(d)の格納形式	参照先
OFF	固定桁数(8桁)で格納	743ページ SM705(変換桁数切換)がOFFの場合の動作
ON	(s)の値によって各桁を前詰めで格納	743ページ SM705(変換桁数切換)がONの場合の動作

■動作の概要

SM705(変換桁数切換)がOFFの場合とONの場合のそれぞれについての動作を下記に示します。

例

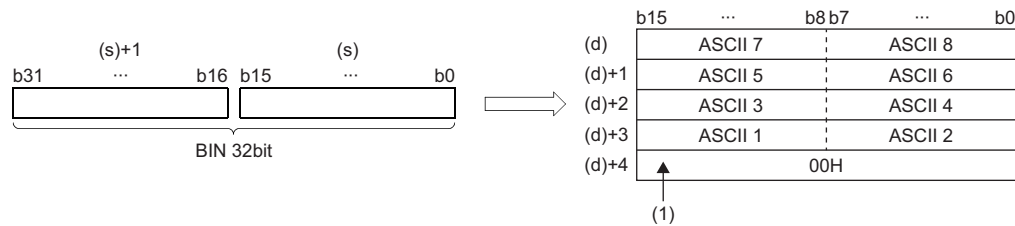
(s)にBIN32ビットデータ「000ABCDEH」が格納されている状態で，DBINHA(P)命令を実行した場合



- SM705(変換桁数切換)がOFFの場合は固定桁数となります。「000ABCDE」の8桁がアスキーデータに変換されて格納されます。
- SM705(変換桁数切換)がONの場合は前詰めとなります。「000ABCDE」の有効桁数部分「ABCDE」のみがアスキーデータに変換されて格納され，終端に00Hが格納されます。

■SM705(変換桁数切換)がOFFの場合の動作

(d)~(d)+3には、16進アスキーデータが固定桁数(8桁)で格納されます。



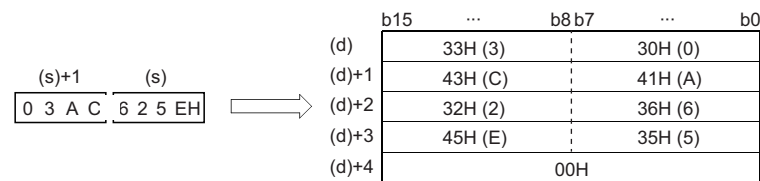
ASCII □: □桁目のアスキーコード

(1): (d)+4には、SM701(出力文字数切換)がOFFのときは00Hが格納され、ONのときは変化しません。

- (d)に格納する演算結果は、8桁の16進数として処理します。したがって、有効桁数の左側の0は「0」として処理します。(0パディングします。)

例

(s)に03AC625EHを指定した場合



■SM705(変換桁数切換)がONの場合の動作

(d)には、有効桁数の左側の0を省略した桁数(最大8桁)の16進アスキーデータが前詰めで格納されます。

(s)の値と(d)に格納される値の例を下記に示します。

(s)の値	(d)~(d)+3の内容	(s)の値	(d)~(d)+3の内容
• 0H~FH	• (d)の上位バイトに00Hが格納されます。 • (d)+1以降は変化しません。 b15 ... b8 b7 ... b0 (d) 00H ASCII 1 (d)+1 (d)+2 (d)+3	• 10H~FFH	• (d)+1に00Hが格納されます。 • (d)+2以降は変化しません。 b15 ... b8 b7 ... b0 (d) ASCII 1 ASCII 2 (d)+1 00H (d)+2 (d)+3
• 1000000H~FFFFFFH	• (d)+3の上位バイトに00Hが格納されます。 b15 ... b8 b7 ... b0 (d) ASCII 6 ASCII 7 (d)+1 ASCII 4 ASCII 5 (d)+2 ASCII 2 ASCII 3 (d)+3 00H ASCII 1	• 1000000H~FFFFFFH	• (1): SM701(出力文字数切換)がOFFのときは(d)+4に00Hが格納されます。SM701(出力文字数切換)がONのときは(d)+4は変化しません。 b15 ... b8 b7 ... b0 (d) ASCII 7 ASCII 8 (d)+1 ASCII 5 ASCII 6 (d)+2 ASCII 3 ASCII 4 (d)+3 ASCII 1 ASCII 2 (d)+4 00H (1)

ASCII □: □桁目のアスキーコード

- 桁数が最大桁数(8桁)未満の場合、SM701(出力文字数切換)のON/OFFに関わらず、文字列の終端に00Hを格納します。文字列の終端が下位バイトの場合、上位バイトにも00Hを格納します。
- 桁数が最大桁数(8桁)の場合、SM701(出力文字数切換)がOFFのときは(d)+4に00Hを格納します。SM701(出力文字数切換)がONのときは(d)+4は変化しません。

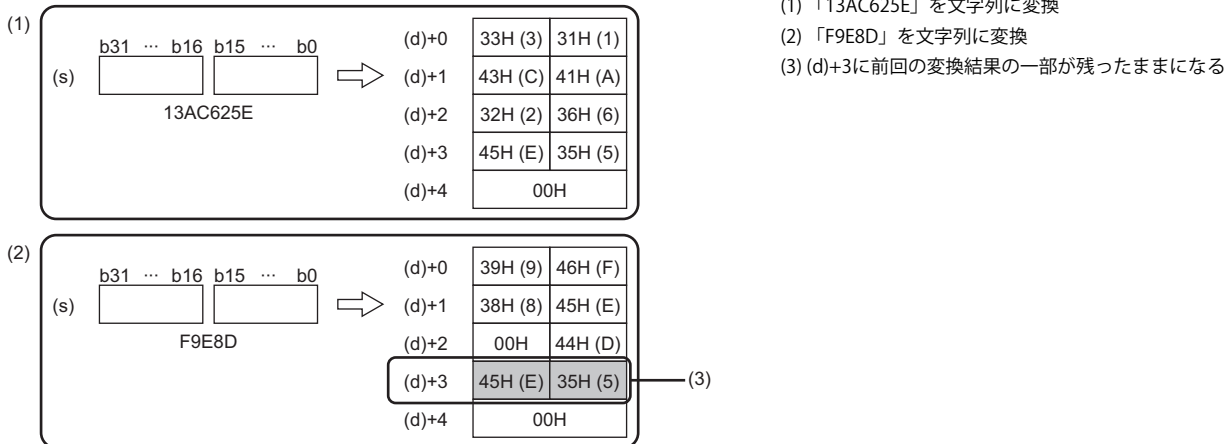
エラー

演算エラーはありません。

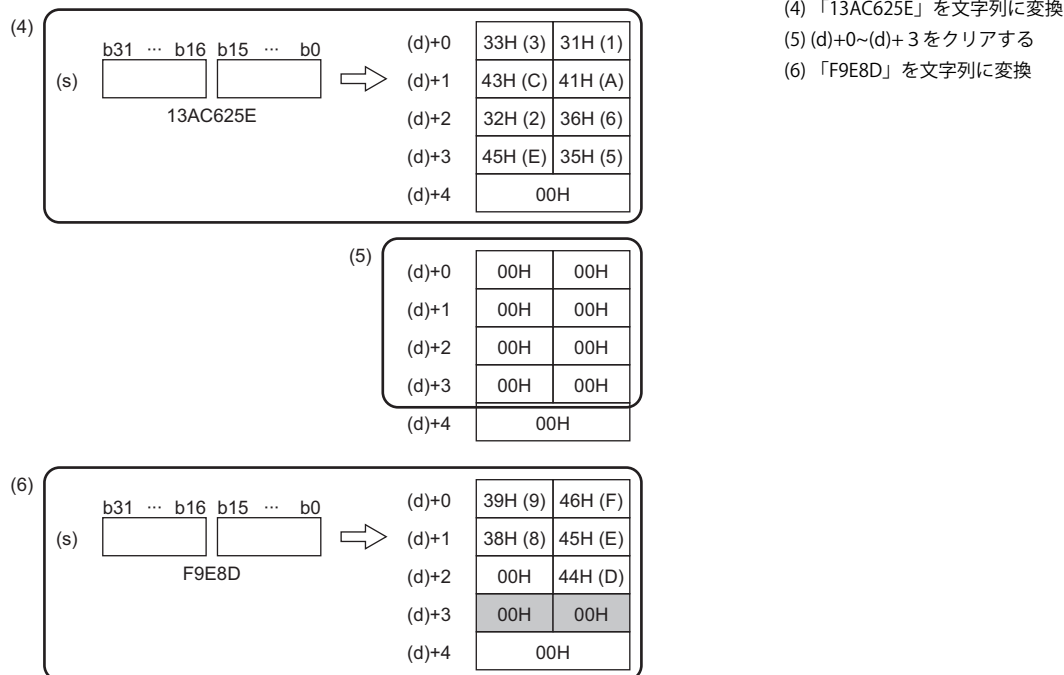
注意事項

SM705(変換桁数切換)がONのとき、(d)には演算結果を有効桁数分だけ格納します。このため、DBINHA(P)命令を連続して実行し、演算結果を同一デバイスに格納する場合、(d)の一部に前回の演算結果が上書きされずに残っている可能性があります。

[例] (s)が「13AC625E」のときにDBINHA(P)命令を実行し、さらに(s)が「F9E8D」のときにDBINHA(P)命令を実行する例



これを回避したい場合は、格納エリア(d)+0~(d)+3をすべてクリアしておいてからDBINHA(P)命令を実行するようにプログラムを作成してください。



BIN16ビットデータ→文字列変換

STR(P)(_U)

BIN16ビットデータを，指定した位置に小数点を付加して文字列に変換します。

ラダー	ST
	ENO:=STR(EN,s1,s2,d); ENO:=STRP(EN,s1,s2,d);

FBD/LD

■実行条件

命令	実行条件
STR STR_U	
STRP STRP_U	

設定データ

■内容，範囲，データ型

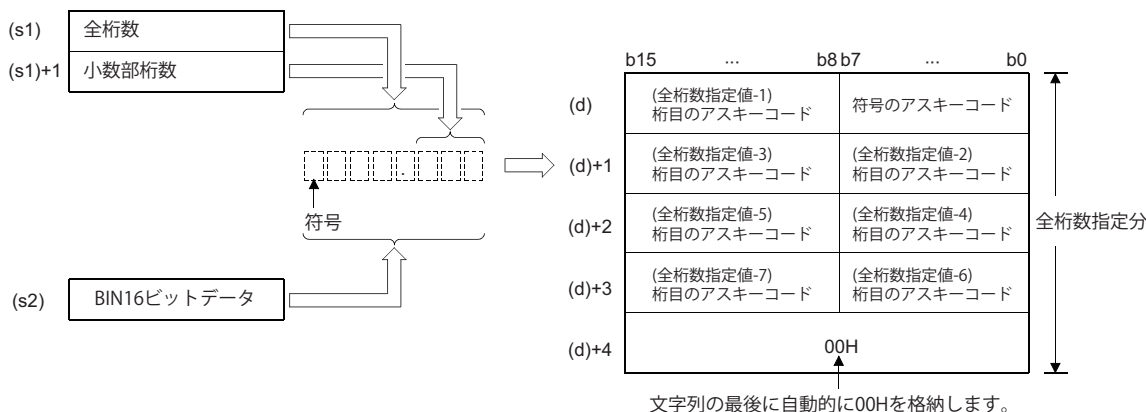
オペランド		内容	範囲	データ型	データ型(ラベル)
(s1)	STR(P)	変換する数値の桁数が格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16_S_ARRAY (要素数: 2)
	STR(P)_U			符号なしBIN16ビット	ANY16_U_ARRAY (要素数: 2)
(s2)	STR(P)	変換するBINデータ	-32768~32767	符号付きBIN16ビット	ANY16_S
	STR(P)_U		0~65535	符号なしBIN16ビット	ANY16_U
(d)		変換後の文字列を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	—	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—

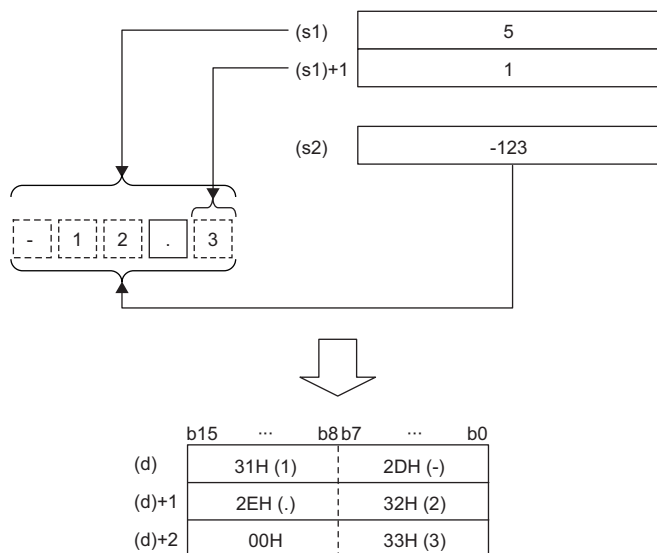
機能

- (s2)で指定されたBIN16ビットデータを、(s1)で指定された位置に小数点を付加して文字列に変換し、(d)で指定されたデバイス番号以降に格納します。



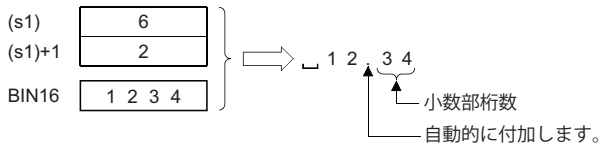
例

(s2)のデータ「-123」を、小数部桁数が1桁の値「-12.3」と見なして文字列に変換する場合

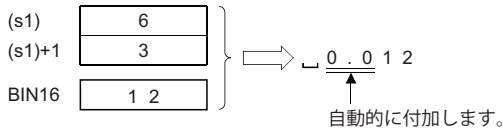


- (d)に小数部を格納する場合は、(s1)で指定できる全桁数は、4~8桁です。(s1)+1で指定できる小数部桁数は、1~5桁です。ただし、小数部桁数≤(全桁数-3)となるように設定してください。
- (d)に小数部を格納しない場合は、(s1)で指定できる全桁数は、2~8桁です。(s1)+1で指定できる小数部桁数は、設定できません。

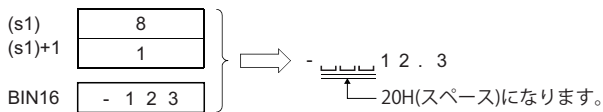
- 変換後の文字列データは、下記のように(d)以降のデバイス番号に格納します。
- 符号には、BIN16ビットデータが正のとき20H(スペース)を格納し、負のとき2DH(-)を格納します。
- 小数部桁数を0以外に設定した場合は、指定された桁数+1桁目に自動的に2EH(.)を格納します。小数部桁数が0のときは、2EH(.)を格納しません。



- 小数部桁数の値が、BIN16ビットデータの桁数より大きい場合は、自動的に0を付加し、右詰めで「0.□□□□」と変換します。



- 全桁数の値から、符号、小数点分を除いた桁数が、BIN16ビットデータの桁数よりも大きい場合、符号と数値の間に、20H(スペース)を格納します。BIN16ビットデータの桁数の方が大きい場合は、エラーとなります。



- 変換した文字列の最後には、自動的に00Hを格納します。

エラー

エラーコード (SD0)	内容
3281H	(s1)に、変換できない不正なデータを入力したとき。 - 全桁数指定が、2~8以外 (s1)+1の小数部桁数指定が、0~5以外 (s1)+1の小数部桁数指定が、1~5の場合に、(s1)で指定する全桁数と(s1)+1で指定する小数部桁数の指定値の関係が、下記以外 全桁数-3≥小数部桁数 (s1)で指定した桁数が、(s2)で指定したBIN16ビットデータの桁数+2より小さい [(s1)の桁数]<[(s2)の符号を含まないBIN16ビットデータの桁数+符号分(+もしくは-)の桁数+小数点分(.)の桁数]
3286H	(d)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に変換した文字列がすべて格納できないとき。

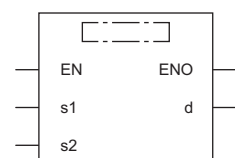
BIN32ビットデータ→文字列変換

DSTR(P)(_U)

BIN32ビットデータを，指定した位置に小数点を付加して文字列に変換します。

ラダー	ST	
	ENO:=DSTR(EN,s1,s2,d); ENO:=DSTRP(EN,s1,s2,d);	ENO:=DSTR_U(EN,s1,s2,d); ENO:=DSTRP_U(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
DSTR DSTR_U	
DSTRP DSTRP_U	

設定データ

■内容，範囲，データ型

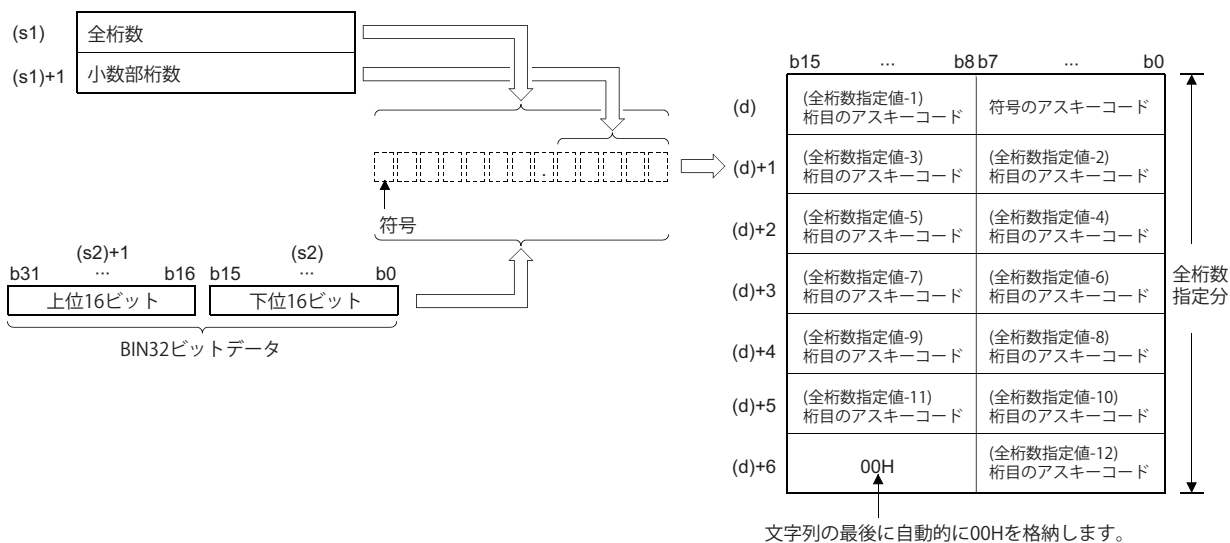
オペランド		内容	範囲	データ型	データ型(ラベル)
(s1)	DSTR(P)	変換する数値の桁数が格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16_S_ARRAY (要素数: 2)
	DSTR(P)_U			符号なしBIN16ビット	ANY16_U_ARRAY (要素数: 2)
(s2)	DSTR(P)	変換するBINデータ	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
	DSTR(P)_U			符号なしBIN32ビット	ANY32_U
(d)		変換後の文字列を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□≠□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□≠G□, J□≠□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	—	—	○	—	—	—	—	—
(s2)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—

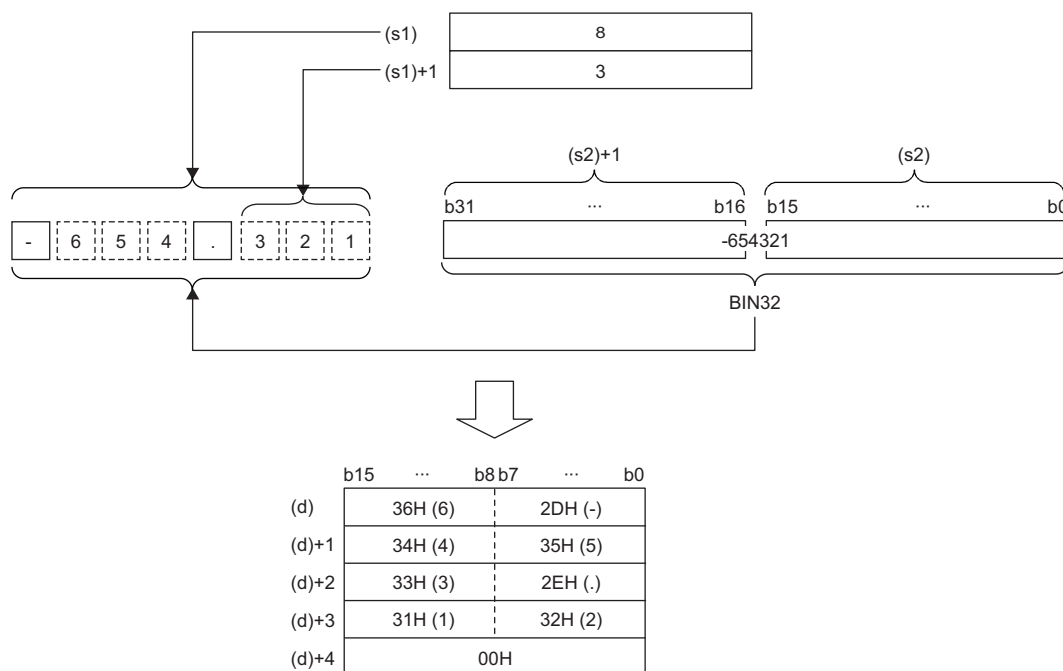
機能

- (s2)で指定されたBIN32ビットデータを、(s1)で指定された位置に小数点を付加して文字列に変換し、(d)で指定されたデバイス番号以降に格納します。



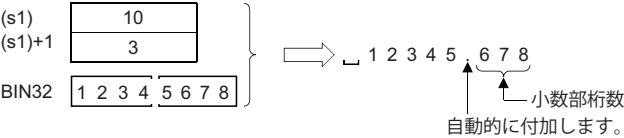
例

(s2)のデータ「-654321」を、小数部桁数が3桁の値「-654.321」と見なして文字列に変換する場合

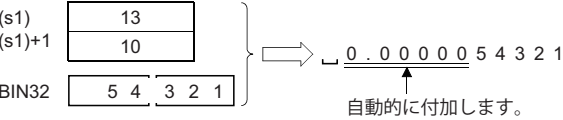


- (d)に小数部を格納する場合は、(s1)で指定できる全桁数は、4~13桁です。(s1)+1で指定できる小数部桁数は、1~10桁です。ただし、小数部桁数≤(全桁数-3)となるように設定してください。
- (d)に小数部を格納しない場合は、(s1)で指定できる全桁数は、2~13桁です。(s1)+1で指定できる小数部桁数は、設定できません。

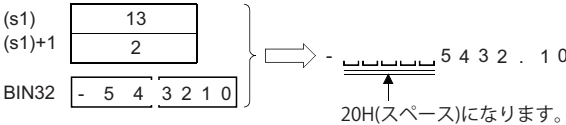
- 変換後の文字列データは、下記のように(d)以降のデバイス番号に格納します。
- 符号には、BIN32ビットデータが正のとき20H(スペース)を格納し、負のとき2DH(-)を格納します。
- 小数部桁数を0以外に設定した場合は、指定された桁数+1桁目に自動的に2EH(.)を格納します。小数部桁数が0のときは、2EH(.)を格納しません。



- 小数部桁数の値が、BIN32ビットデータの桁数より大きい場合は、自動的に0を付加し、右詰めで「0.□□□□」と変換します。



- 全桁数の値から、符号、小数点分を除いた桁数が、BIN32ビットデータの桁数よりも大きい場合、符号と数値の間に、20H(スペース)を格納します。BIN32ビットデータの桁数の方が大きい場合は、エラーとなります。



- 変換した文字列の最後には、自動的に00Hを格納します。

エラー

エラーコード (SD0)	内容
3281H	(s1)に、変換できない不正なデータを入力したとき。 - 全桁数指定が、2~13以外 (s1)+1の小数部桁数指定が、0~10以外 (s1)+1の小数部桁数指定が、0~10の場合に、(s1)で指定する全桁数と(s1)+1で指定する小数部桁数の指定値の関係が、下記以外 全桁数-3≧小数部桁数 (s1)で指定した桁数が、(s2)で指定したBIN32ビットデータの桁数+2より小さい [(s1)の桁数]<[(s2)の符号を含まないBIN32ビットデータの桁数+符号分(+もしくは-)の桁数+小数点分(.)の桁数]
3286H	(d)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に変換した文字列がすべて格納できないとき。

BCD4桁データ→10進アスキーコード変換

BCDDA(P)

BCD4桁データをアスキーコードに変換します。

ラダー	ST
	ENO:=BCDDA(EN,s,d); ENO:=BCDDAP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
BCDDA	
BCDDAP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	アスキー変換を行うBCDデータ	0~9999	BCD4桁	ANY16
(d)	変換結果を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—

機能

- (s)で指定されたBCD4桁データの各桁の数値をアスキーコードに変換し、(d)で指定されたデバイス番号以降に格納します。
- SM705(変換桁数切換)の状態によって、(d)に格納される16進アスキーデータの形式が異なります。

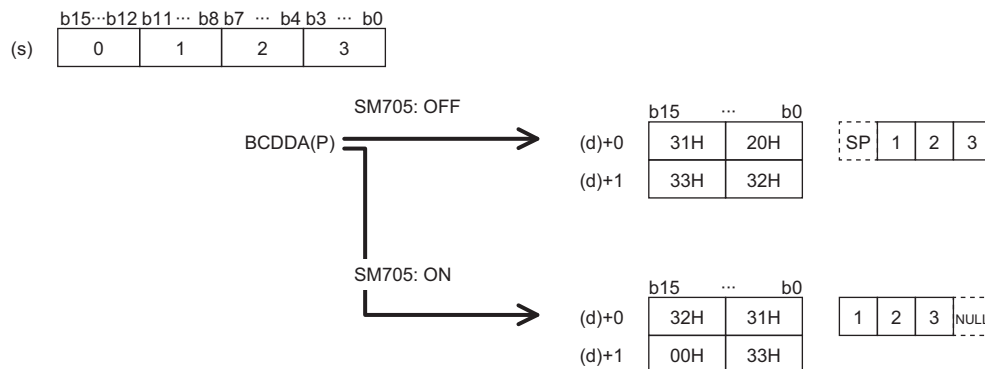
SM705の状態	(d)の格納形式	参照先
OFF	固定桁数(4桁)で格納	753ページ SM705(変換桁数切換)がOFFの場合の動作
ON	(s)の値によって各桁を前詰めで格納	753ページ SM705(変換桁数切換)がONの場合の動作

■動作の概要

SM705(変換桁数切換)がOFFの場合とONの場合のそれぞれについての動作を下記に示します。

例

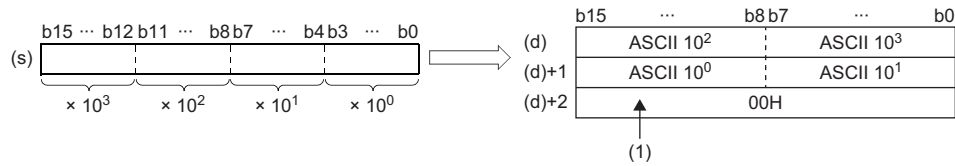
(s)にBCD4桁データ「0123」が格納されている状態で、BCDDA(P)命令を実行した場合



- SM705(変換桁数切換)がOFFの場合は固定桁数となります。「0123」の先頭の0は20H(スペース)に変換されて格納されます。
- SM705(変換桁数切換)がONの場合は前詰めとなります。「0123」の先頭の0を省略した「123」がアスキーデータに変換されて格納され、終端に00Hが格納されます。

■SM705(変換桁数切換)がOFFの場合の動作

(d)~(d)+1には、10進アスキーデータが固定桁数(4桁)で格納されます。



ASCII 10³: 千の位のアスキーコード

ASCII 10²: 百の位のアスキーコード

ASCII 10¹: 十の位のアスキーコード

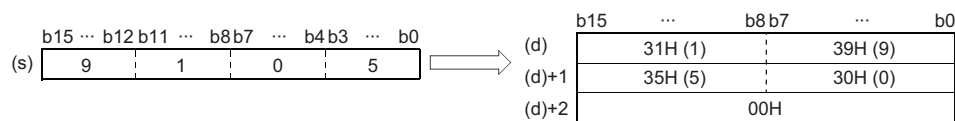
ASCII 10⁰: 一の位のアスキーコード

(1): (d)+2には、SM701(出力文字数切換)がOFFのときは00Hが格納され、ONのときは変化しません。

- (d)に格納する演算結果で、有効桁数の左側の0には20H(スペース)を格納します。(0サプレスします。)たとえば「0050」の場合、「00」はそれぞれ20H(スペース)となり、「50」が有効桁数となります。

例

(s)に9105を指定した場合



■SM705(変換桁数切換)がONの場合の動作

(d)には、有効桁数の左側の0を省略した桁数(最大4桁)の10進アスキーデータが格納されます。

(s)の値と(d)に格納される値の例を下記に示します。

(s)の値	(d)~(d)+1の内容	(s)の値	(d)~(d)+1の内容
• 0H~9H	• (d)の上位バイトに00Hが格納されます。 • (d)+1以降は変化しません。 b15 ... b8 b7 ... b0 (d) 00H ASCII 10⁰ (d)+1	• 10H~99H	• (d)+1に00Hが格納されます。 b15 ... b8 b7 ... b0 (d) ASCII 10⁰ ASCII 10¹ (d)+1 00H
• 100H~999H	• (d)+1の上位バイトに00Hが格納されます。 b15 ... b8 b7 ... b0 (d) ASCII 10¹ ASCII 10² (d)+1 00H ASCII 10⁰	• 1000H~9999H	• (1): SM701(出力文字数切換)がOFFのときは(d)+2に00Hが格納されます。SM701(出力文字数切換)がONのときは(d)+2は変化しません。 b15 ... b8 b7 ... b0 (d) ASCII 10² ASCII 10³ (d)+1 ASCII 10⁰ ASCII 10¹ (d)+2 00H (1)

ASCII 10³: 千の位のアスキーコード

ASCII 10²: 百の位のアスキーコード

ASCII 10¹: 十の位のアスキーコード

ASCII 10⁰: 一の位のアスキーコード

- 桁数が最大桁数(4桁)未満の場合、SM701(出力文字数切換)のON/OFFに関わらず、文字列の終端に00Hを格納します。文字列の終端が下位バイトの場合、上位バイトにも00Hを格納します。
- 桁数が最大桁数(4桁)の場合、SM701(出力文字数切換)がOFFのときは(d)+2に00Hを格納します。SM701(出力文字数切換)がONのときは(d)+2は変化しません。

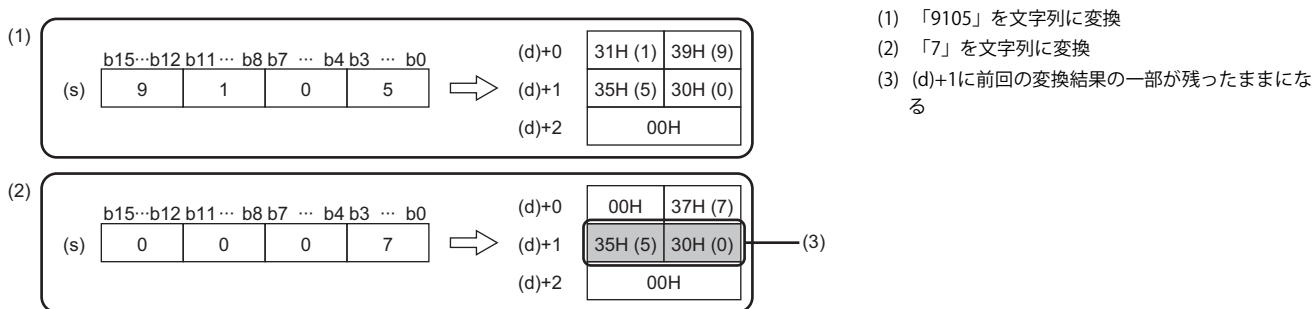
エラー

エラーコード (SD0)	内容
3281H	(s)のデータが0~9999以外するとき。
3286H	(d)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に変換した文字列がすべて格納できないとき。

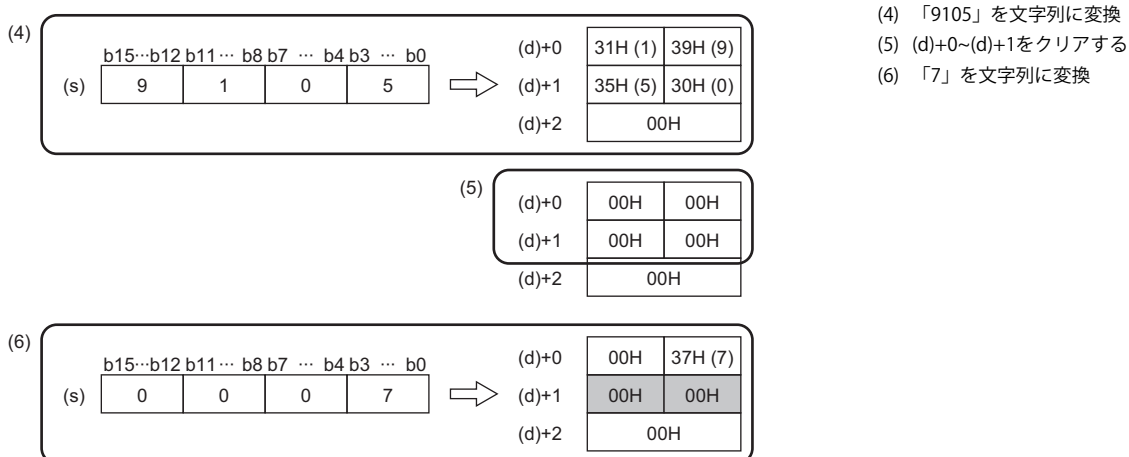
注意事項

SM705(変換桁数切換)がONのとき、(d)には演算結果を有効桁数分だけ格納します。このため、BCDDA(P)命令を連続して実行し、演算結果を同一デバイスに格納する場合、(d)の一部に前回の演算結果が上書きされずに残っている可能性があります。

[例] (s)が「9105H」のときにBCDDA(P)命令を実行し、さらに(s)が「0007H」のときにBCDDA(P)命令を実行する例



これを回避したい場合は、格納エリア(d)+0~(d)+1をすべてクリアしておいてからBCDDA(P)命令を実行するようにプログラムを作成してください。



BCD8桁データ→10進アスキーコード変換

DBCDDA(P)

BCD8桁データをアスキーコードに変換します。

ラダー	ST
	ENO:=DBCDDA(EN,s,d); ENO:=DBCDDAP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
DBCDDA	
DBCDDAP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	アスキー変換を行うBCDデータ	0~99999999	BCD8桁	ANY32
(d)	変換結果を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- (s)で指定されたBCD8桁データの各桁の数値をアスキーコードに変換し、(d)で指定されたデバイス番号以降に格納します。
- SM705(変換桁数切換)の状態によって、(d)に格納される10進アスキーデータの形式が異なります。

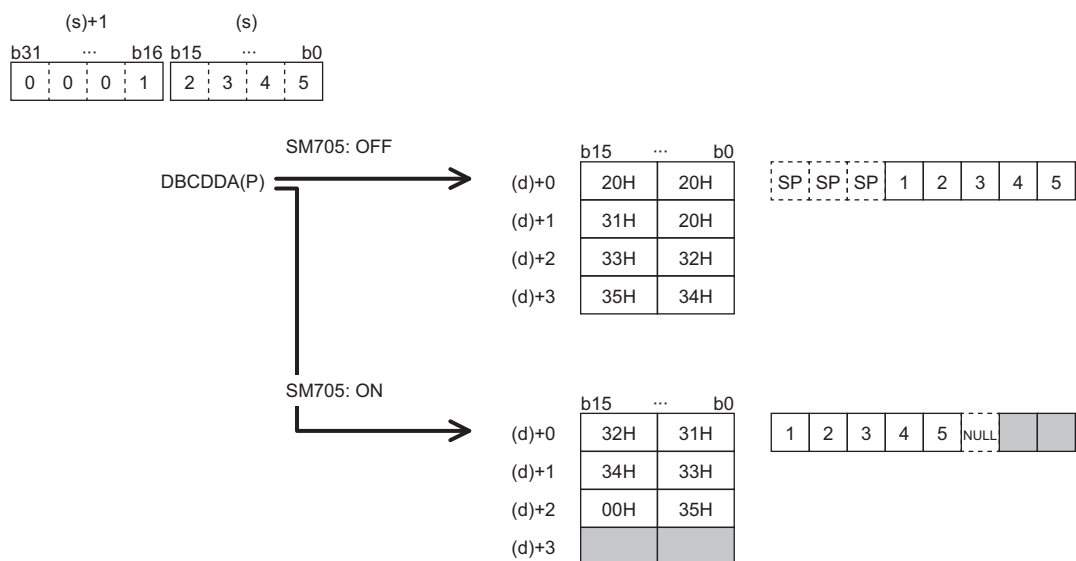
SM705の状態	(d)の格納形式	参照先
OFF	固定桁数(8桁)で格納	757ページ SM705(変換桁数切換)がOFFの場合の動作
ON	(s)の値によって各桁を前詰めで格納	758ページ SM705(変換桁数切換)がONの場合の動作

■動作の概要

SM705(変換桁数切換)がOFFの場合とONの場合のそれぞれについての動作を下記に示します。

例

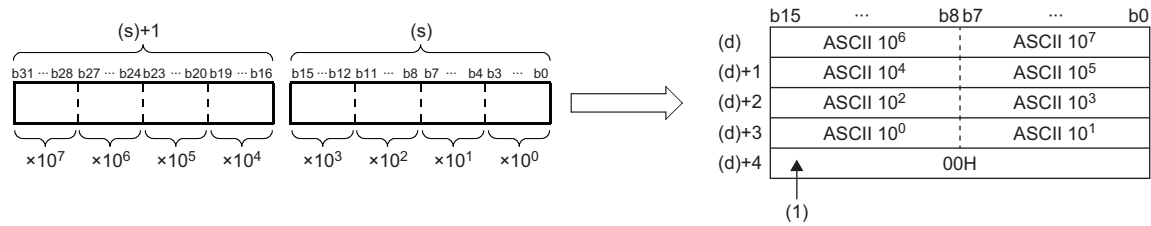
(s)にBCD8桁データ「00012345」が格納されている状態で、DBCDDA(P)命令を実行した場合



- SM705(変換桁数切換)がOFFの場合は固定桁数となります。「00012345」の0は20H(スペース)に変換されて格納されます。
- SM705(変換桁数切換)がONの場合は前詰めとなります。「00012345」の先頭の0を省略した「12345」がアスキーデータに変換されて格納され、終端に00Hが格納されます。

■SM705(変換桁数切換)がOFFの場合の動作

(d)~(d)+3には、10進アスキーデータが固定桁数(8桁)で格納されます。



ASCII 10⁷: 千万の位のアスキーコード

ASCII 10⁶: 百万の位のアスキーコード

⋮

ASCII 10¹: 十の位のアスキーコード

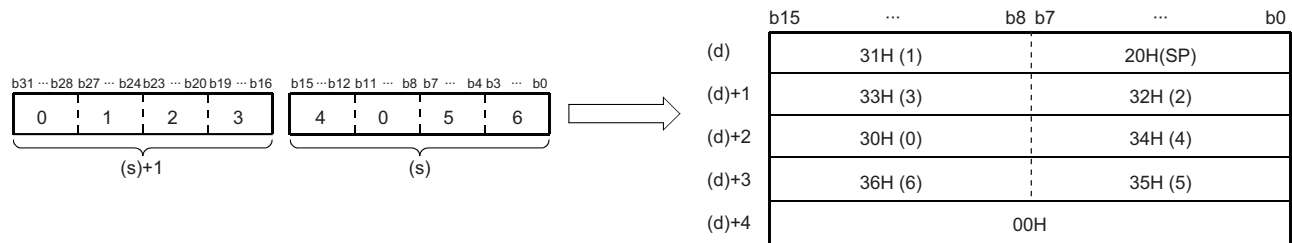
ASCII 10⁰: 一の位のアスキーコード

(1): (d)+4には、SM701(出力文字数切換)がOFFのときは00Hが格納され、ONのときは変化しません。

- (d)に格納する演算結果で、有効桁数の左側の0には20H(スペース)を格納します。(0サプレスします。)たとえば「00012098」の場合、「000」はそれぞれ20H(スペース)となり、「12098」が有効桁数となります。

例

(s)に01234056を指定した場合



■SM705(変換桁数切換)がONの場合の動作

(d)には、有効桁数の左側の0を省略した桁数(最大8桁)の10進アスキーデータが格納されます。

(s)の値と(d)に格納される値の例を下記に示します。

(s)の値	(d)~(d)+3の内容	(s)の値	(d)~(d)+3の内容																																																							
• 0H~9H	• (d)の上位バイトに00Hが格納されます。 • (d)+1以降は変化しません。 <table><tr><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(d)</td><td>00H</td><td>:</td><td>ASCII 10⁰</td><td></td></tr><tr><td>(d)+1</td><td colspan="4"></td></tr><tr><td>(d)+2</td><td colspan="4"></td></tr><tr><td>(d)+3</td><td colspan="4"></td></tr></table>	b15	...	b8 b7	...	b0	(d)	00H	:	ASCII 10 ⁰		(d)+1					(d)+2					(d)+3					• 10H~99H	• (d)+1に00Hが格納されます。 • (d)+2以降は変化しません。 <table><tr><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(d)</td><td>ASCII 10⁰</td><td>:</td><td>ASCII 10¹</td><td></td></tr><tr><td>(d)+1</td><td colspan="4">00H</td></tr><tr><td>(d)+2</td><td colspan="4"></td></tr><tr><td>(d)+3</td><td colspan="4"></td></tr></table>	b15	...	b8 b7	...	b0	(d)	ASCII 10 ⁰	:	ASCII 10 ¹		(d)+1	00H				(d)+2					(d)+3									
b15	...	b8 b7	...	b0																																																						
(d)	00H	:	ASCII 10 ⁰																																																							
(d)+1																																																										
(d)+2																																																										
(d)+3																																																										
b15	...	b8 b7	...	b0																																																						
(d)	ASCII 10 ⁰	:	ASCII 10 ¹																																																							
(d)+1	00H																																																									
(d)+2																																																										
(d)+3																																																										
:																																																										
• 1000000H~9999999H	• (d)+3の上位バイトに00Hが格納されます。 <table><tr><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(d)</td><td>ASCII 10⁵</td><td>:</td><td>ASCII 10⁶</td><td></td></tr><tr><td>(d)+1</td><td>ASCII 10³</td><td>:</td><td>ASCII 10⁴</td><td></td></tr><tr><td>(d)+2</td><td>ASCII 10¹</td><td>:</td><td>ASCII 10²</td><td></td></tr><tr><td>(d)+3</td><td>00H</td><td>:</td><td>ASCII 10⁰</td><td></td></tr></table>	b15	...	b8 b7	...	b0	(d)	ASCII 10 ⁵	:	ASCII 10 ⁶		(d)+1	ASCII 10 ³	:	ASCII 10 ⁴		(d)+2	ASCII 10 ¹	:	ASCII 10 ²		(d)+3	00H	:	ASCII 10 ⁰		• 10000000H~99999999H	• (1): SM701(出力文字数切換)がOFFのときは(d)+4に00Hが格納されます。SM701(出力文字数切換)がONのときは(d)+4は変化しません。 <table><tr><td>b15</td><td>...</td><td>b8 b7</td><td>...</td><td>b0</td></tr><tr><td>(d)</td><td>ASCII 10⁶</td><td>:</td><td>ASCII 10⁷</td><td></td></tr><tr><td>(d)+1</td><td>ASCII 10⁴</td><td>:</td><td>ASCII 10⁵</td><td></td></tr><tr><td>(d)+2</td><td>ASCII 10²</td><td>:</td><td>ASCII 10³</td><td></td></tr><tr><td>(d)+3</td><td>ASCII 10⁰</td><td>:</td><td>ASCII 10¹</td><td></td></tr><tr><td>(d)+4</td><td colspan="4">00H</td></tr></table> (1)↑	b15	...	b8 b7	...	b0	(d)	ASCII 10 ⁶	:	ASCII 10 ⁷		(d)+1	ASCII 10 ⁴	:	ASCII 10 ⁵		(d)+2	ASCII 10 ²	:	ASCII 10 ³		(d)+3	ASCII 10 ⁰	:	ASCII 10 ¹		(d)+4	00H			
b15	...	b8 b7	...	b0																																																						
(d)	ASCII 10 ⁵	:	ASCII 10 ⁶																																																							
(d)+1	ASCII 10 ³	:	ASCII 10 ⁴																																																							
(d)+2	ASCII 10 ¹	:	ASCII 10 ²																																																							
(d)+3	00H	:	ASCII 10 ⁰																																																							
b15	...	b8 b7	...	b0																																																						
(d)	ASCII 10 ⁶	:	ASCII 10 ⁷																																																							
(d)+1	ASCII 10 ⁴	:	ASCII 10 ⁵																																																							
(d)+2	ASCII 10 ²	:	ASCII 10 ³																																																							
(d)+3	ASCII 10 ⁰	:	ASCII 10 ¹																																																							
(d)+4	00H																																																									

ASCII 10⁷: 千万の位のアスキーコード

ASCII 10⁶: 百万の位のアスキーコード

⋮

ASCII 10¹: 十の位のアスキーコード

ASCII 10⁰: 一の位のアスキーコード

- 桁数が最大桁数(8桁)未満の場合、SM701(出力文字数切換)のON/OFFに関わらず、文字列の終端に00Hを格納します。文字列の終端が下位バイトの場合、上位バイトにも00Hを格納します。
- 桁数が最大桁数(8桁)の場合、SM701(出力文字数切換)がOFFのときは(d)+4に00Hを格納します。SM701(出力文字数切換)がONのときは(d)+4は変化しません。

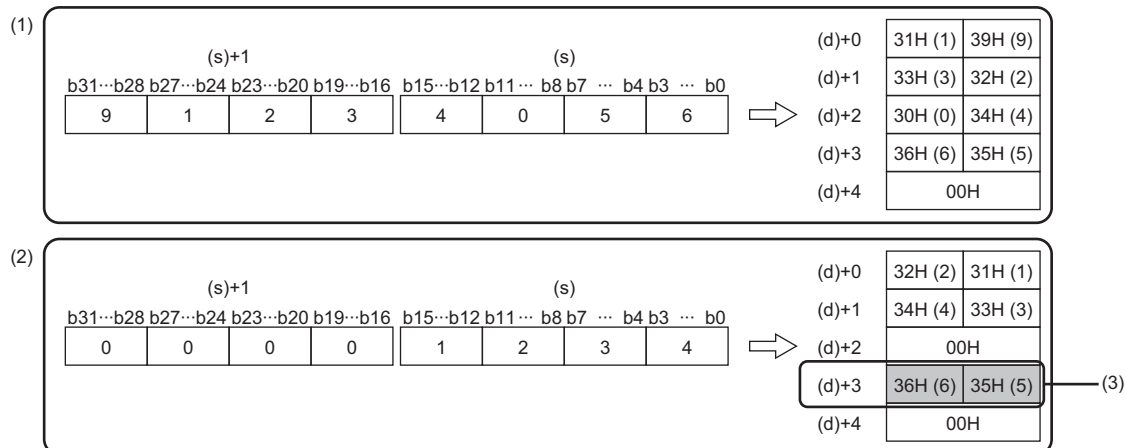
エラー

エラーコード (SD0)	内容
3281H	(s)のデータが0~999999999以外のとき。
3286H	(d)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に変換した文字列がすべて格納できないとき。

注意事項

SM705(変換桁数切換)がONのとき、(d)には演算結果を有効桁数分だけ格納します。このため、DBCDDA(P)命令を連続して実行し、演算結果を同一デバイスに格納する場合、(d)の一部に前回の演算結果が上書きされずに残っている可能性があります。

[例] (s)が「91234056H」のときにDBCDDA(P)命令を実行し、さらに(s)が「00001234H」のときにDBCDDA(P)命令を実行する例

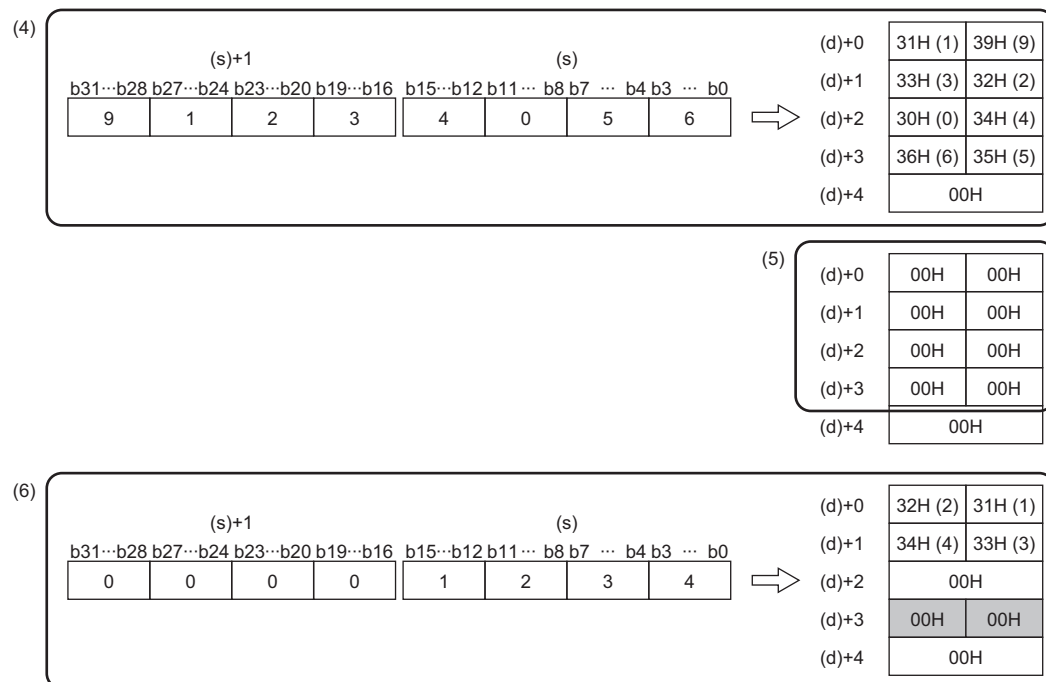


(1) 「91234056」を文字列に変換

(2) 「1234」を文字列に変換

(3) (d)+3に前回の変換結果の一部が残ったままになる

これを回避したい場合は、格納エリア(d)+0~(d)+3をすべてクリアしておいてからDBCDDA(P)命令を実行するようにプログラムを作成してください。



(4) 「91234056」を文字列に変換

(5) (d)+0~(d)+3をクリアする

(6) 「1234」を文字列に変換

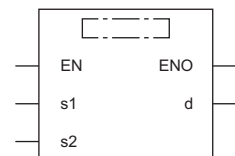
単精度実数→文字列変換

ESTR(P)

単精度実数データを、表示指定にしたがって文字列に変換します。

ラダー	ST
	ENO:=ESTR(EN,s1,s2,d); ENO:=ESTRP(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
ESTR	
ESTRP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	変換する単精度実数または、データが格納されている先頭デバイス	$0, 2^{-126} < (s1) < 2^{128}$	単精度実数	ANYREAL_32
(s2)	変換する数値の表示指定が格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 3)
(d)	変換後の文字列を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□≠□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□≠G□, J□≠□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	○	○	○	○	○	—	○	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

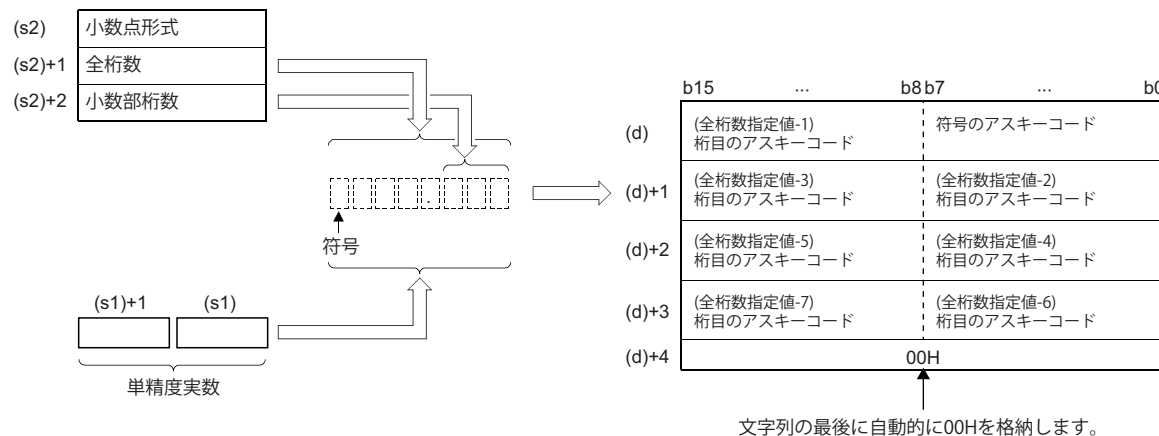
機能

- (s1)で指定されたデバイスに格納されている単精度実数データを、(s2)で指定されたデバイス番号以降に格納されている表示指定にしたがって文字列に変換し、(d)で指定されたデバイス番号以降に格納します。
- (s2)で指定された表示指定により変換後のデータは異なります。

(s2)	0: 小数点形式 1: 指数形式
(s2)+1	全桁数
(s2)+2	小数部桁数

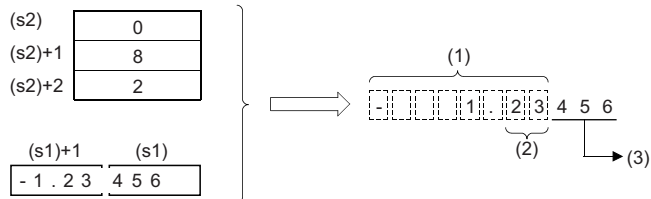
■小数点形式

- (s2)に0を指定した場合は、小数点形式となります。



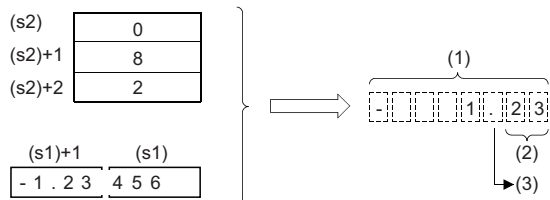
- (d)に小数部を格納する場合は、(s2)+1で指定できる全桁数は、全桁数(最大24) $\geq$ (整数部桁数^{*1} + 小数部桁数 + 2) となります。(s2)+2で指定できる小数部桁数は、0~7桁です。ただし、小数部桁数 $\leq$ (全桁数 - 3) となるように設定してください。
- *1 (s1)で指定する32ビット浮動小数点型実数データの整数部桁数を示します。
- (d)に小数部を格納しない場合は、(s2)+1で指定できる全桁数は、全桁数(最大24) $\geq$ (整数部桁数^{*2} + 1) となります。(s2)+2で指定できる小数部桁数は、設定できません。
- *2 (s1)で指定する32ビット浮動小数点型実数データの整数部桁数を示します。

- 変換後の文字列データは、下記のように(d)以降のデバイス番号に格納します。
- 符号には、単精度実数データが正のとき20H(スペース)を格納し、負のとき2DH(-)を格納します。
- 小数部桁数の範囲に、単精度実数データの小数部が収まらない場合は、下位小数部が四捨五入されます。



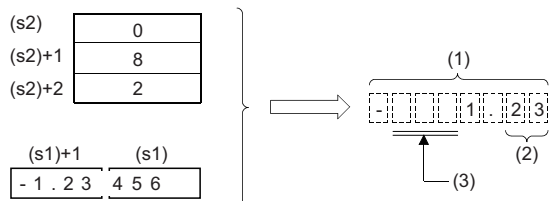
- (1): 全桁数
(2): 小数部桁数
(3): 四捨五入されます。

- 小数部桁数を0以外に設定した場合は、指定された小数部桁数+1桁目に、自動的に2EH(.)を格納します。小数部桁数が0のときは、2EH(.)を格納しません。



- (1): 全桁数
(2): 小数部桁数
(3): 自動的に付加します。

- 全桁数から符号、小数点、小数部を除いた桁数が単精度実数データの整数部よりも大きい場合、符号と整数部の間に20H(スペース)を格納します。



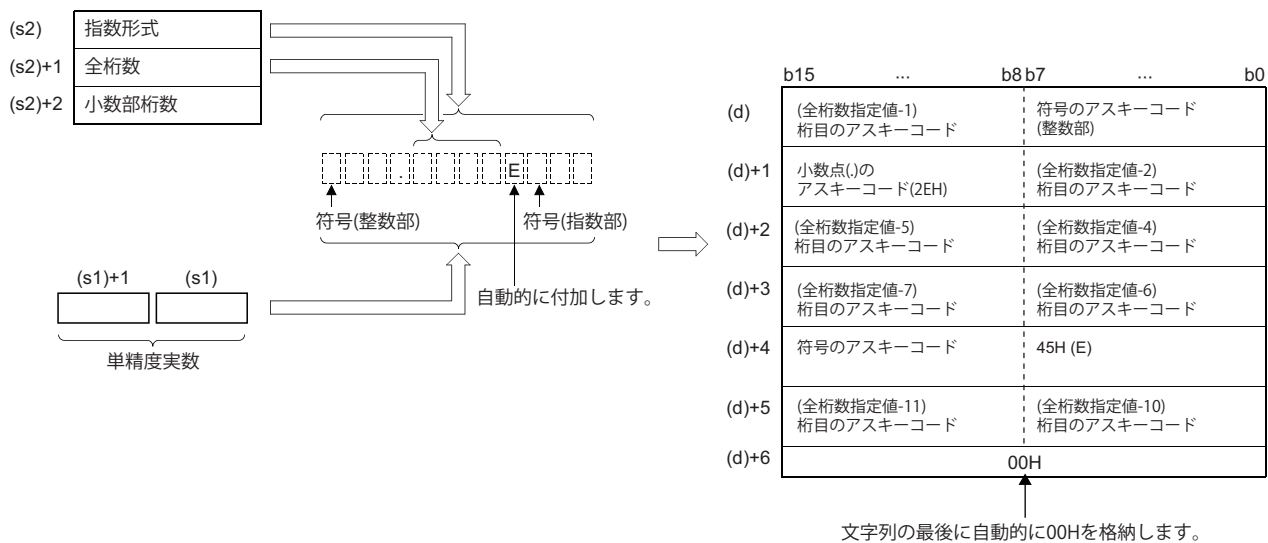
- (1): 全桁数
(2): 小数部桁数
(3): 20H(スペース)になります。

- 変換した文字列の最後には、自動的に00Hを格納します。

- (s1)で指定できる32ビット浮動小数点型実数データの整数部桁数は、1~16桁となる。

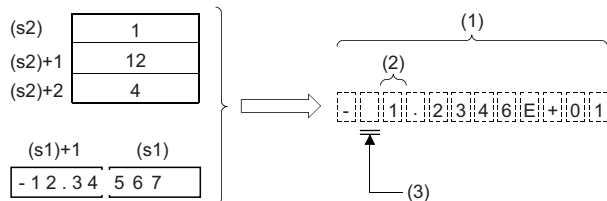
■指数形式

- (s2)に1を指定した場合は、指数形式となります。

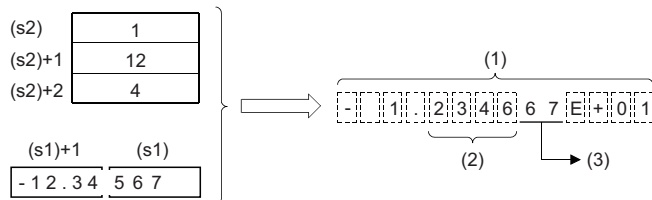


- (d)に小数部を格納する場合は、(s2)+1で指定できる全桁数は、桁数(最大24)≥(小数部桁数+7)となります。(s2)+2で指定できる小数部桁数は、0~7桁です。ただし、小数部桁数≤(全桁数-7)となるように設定してください。
- (d)に小数部を格納しない場合は、(s2)+1で指定できる全桁数は、桁数(最大24)≥6となります。(s2)+2で指定できる小数部桁数は、設定できません。

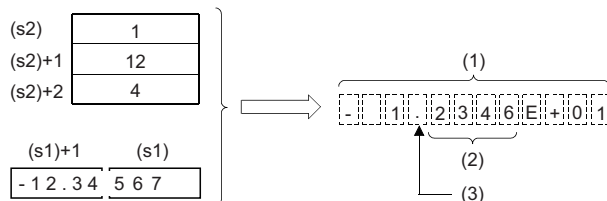
- 変換後の文字列データは、下記のように(d)以降のデバイス番号に格納します。
- 整数部の符号には、単精度実数データが正のとき20H(スペース)を格納し、負のとき2DH(-)を格納します。
- 整数部は1桁固定です。整数部と符号との間に20H(スペース)を格納します。



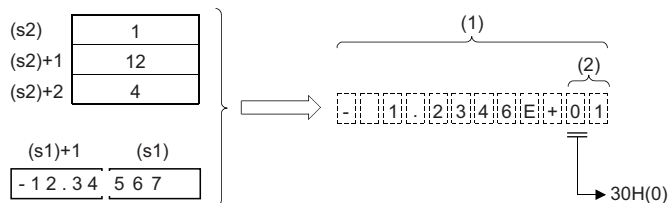
- (1): 全桁数
 (2): 1桁固定
 (3): 20H(スペース)になります。
- 小数部桁数の範囲に、単精度実数データの小数部が収まらない場合は、下位小数部が四捨五入されます。



- (1): 全桁数
 (2): 小数部桁数
 (3): 切り捨てられます。
- 小数部桁数を0以外に設定した場合は、指定された小数部桁数+1桁目に、自動的に2EH(.)を格納します。小数部桁数が0のときは、2EH(.)を格納しません。



- (1): 全桁数
 (2): 小数部桁数
 (3): 自動的に付加します。
- 指数部の符号には、指数が正のとき2BH(+)を格納し、負のとき2DH(-)を格納します。
 - 指数部は2桁固定です。指数部が1桁の場合は、指数部の符号との間に30H(0)を格納します。



- (1): 全桁数
 (2): 2桁固定
- 変換した文字列の最後には、自動的に00Hを格納します。
 - エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

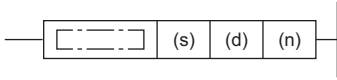
エラーコード (SD0)	内容
3281H	(s1)が下記範囲内でないとき。 $0, 2^{-126} \leq (s1) < 2^{128}$ (s2)に変換できない不正なデータを設定したとき。 ・(s2)で指定する形式指定が0, 1以外るとき。 ・小数点形式で, (s2)+1で指定する全桁数指定が下記範囲外になっているとき。 小数部桁数が0のとき: 全桁数$\geq$整数部桁数^{*1}+1 小数部桁数が0以外るとき: 全桁数$\geq$整数部桁数^{*1}+小数点桁数+2 ・小数点形式で, (s2)+1で指定する全桁数指定と(s2)+2で指定する小数部桁指定が下記範囲外になっているとき。 小数部桁数が0のとき: 全桁数$\geq$2 小数部桁数が0以外るとき: 全桁数$\geq$(小数点桁数+3) ・指数形式で, (s2)+1で指定する全桁数指定が下記範囲外になっているとき。 小数部桁数が0のとき: 全桁数$\geq$6 小数部桁数が0以外るとき: 全桁数$\geq$(小数部桁数+7) ・指数形式で, (s2)+2で指定する小数部桁数指定が下記範囲外になっているとき。 小数点形式のとき: 小数部桁数$\leq$(全桁数-3) 指数形式のとき: 小数部桁数$\leq$(全桁数-7) ・小数点形式の場合, (s1)に指定できる32ビット浮動小数点実数データの整数部桁数が16桁を超える場合。 ・(s2)+2で指定する小数部桁数が, 0~7桁以外るとき。 全桁数が24を超える値を指定したとき。
3282H	(s1)で指定したデバイスの内容が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3286H	(d)で指定したデバイス番号以降, 該当デバイス/ラベルの割付範囲に変換した文字列がすべて格納できないとき。

*1 (s1)で指定する32ビット浮動小数点型実数データの整数部桁数を示します。

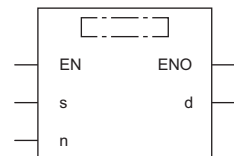
16進BINデータ→16進アスキーコード変換

INT2ASC(P)

BIN16ビットデータを，16進数アスキーコードに変換し，任意の範囲に格納します。

ラダー	ST
	ENO:=INT2ASC(EN,s,n,d); ENO:=INT2ASCP(EN,s,n,d)

FBD/LD



■実行条件

命令	実行条件
INT2ASC	
INT2ASCP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	文字列に変換するBINデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
(d)	変換後の文字列を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
(n)	格納する文字数	0~16383	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

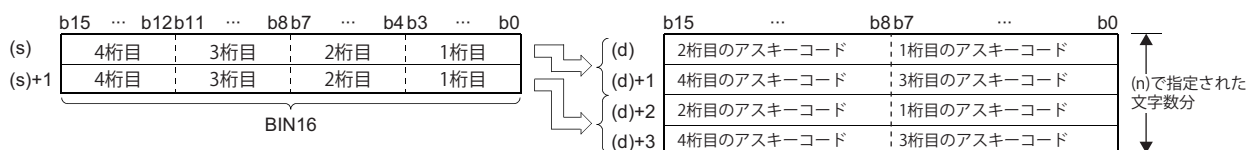
*1 ラベルで設定する場合は，動作に必要な領域が確保できるよう配列を定義し，その配列型ラベルの要素を指定してください。

■使用可能デバイス

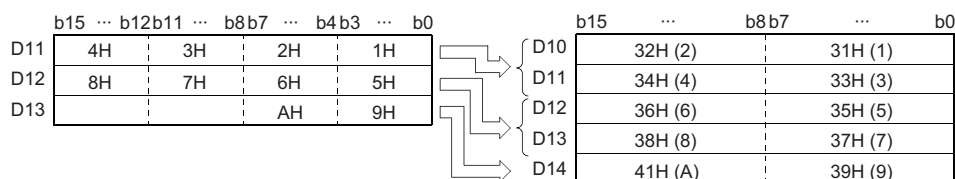
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

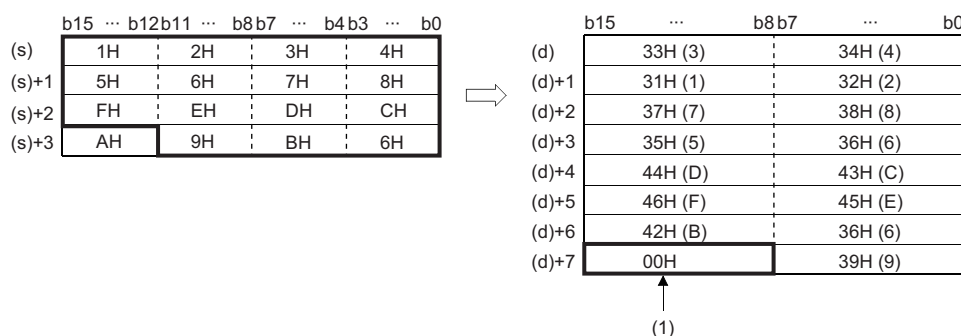
- (s)で指定されたデバイス番号以降に格納されているBIN16ビットデータを、16進数扱いでアスキー変換し、(d)で指定したデバイス番号以降に(n)で指定した文字数分を格納します。



- (n)で文字数を設定することにより、(s)で指定するBINデータの範囲、および(d)で指定する文字列の格納デバイスの範囲が自動的に決まります。
- 変換するBINデータが格納されているデバイス範囲と、変換したアスキーデータを格納するデバイス範囲が重複している場合でも、正常に処理します。



- (n)で指定された文字数が奇数の場合、文字列を格納するデバイス範囲の最終デバイス番号の上位8ビットには、自動的に00Hを格納します。



(1): 自動的に00Hを格納します。

- (n)で指定する文字数が0の場合は、変換処理を行いません。

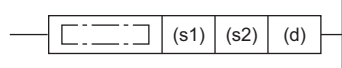
エラー

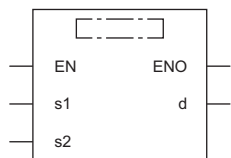
エラーコード (SD0)	内容
3285H	(n)に指定可能な範囲外のデータを入力したとき。 • 指定した文字数が0~16383以外

複数データの文字列変換結合

SPF(P)

複数のデータを指定した表示形式に従い文字列に変換し、任意の文字列と連結して1つの文字列として出力します。

ラダー	ST
	ENO:=SPF(EN,s1,s2,d); ENO:=SPFP(EN,s1,s2,d);

FBD/LD


■実行条件

命令	実行条件
SPF	
SPFP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	文字列データおよび表示形式，またはそれらが格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(s2)	変換するデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16
(d)	変換および連結された文字列を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	○	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- (s1)で指定された表示形式に従って、(s2)で指定されたデバイス番号以降に格納したデータを文字列に変換します。さらに、(s1)に格納された任意の文字列と連結し、1つの文字列として(d)で指定されたデバイス番号以降に格納します。なお、任意の文字列を省略し、(s2)で指定したデバイス番号以降に格納したデータのみで結合することもできます。
- 表示形式は、“%”を先頭に記載し[フラグ]、[最小フィールド幅]、[精度]、[取得データサイズ]、[変換指定子]の順で指定子を組み合わせ、連結したい任意の文字列とともに(s1)格納します。使用できる表示形式選択は下記があります。異なる順番で表示形式を使用した場合、エラーとなります。

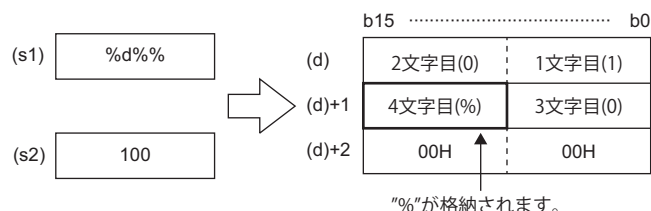
表示形式	指定子 ^{*1}	説明	詳細
フラグ	-, +, "(半角スペース), #, 0	指定する指定子に応じて、(d)に出力する文字列に修飾をします。	771ページ フラグ
最小フィールド幅	任意の数値	変換した文字列の符号、小数点、指数部および“(半角スペース)を含む、(d)に出力する全体の出力文字数を指定します。記載省略時は取得データの文字数分を出力します。	772ページ 最小フィールド幅
精度	“(ピリオド)数値	(s2)に格納されたデータから、[変換指定子]によって(d)に出力する文字数を指定します。	773ページ 精度
取得データサイズ	w, q	(s2)に格納されたデータの何ワード分変換し、(d)に出力するか指定します。記載省略時は1ワード単位で取得します。	775ページ 取得データサイズ
変換指定子	d, u, x, X, o, s, c, f, e, E, g, G	(d)に出力する文字列の出力形式を指定します。	776ページ 変換指定子

*1 大文字/小文字は区別します。

- [フラグ]、[最小フィールド幅]、[精度]、[取得データサイズ]は省略できます。
- “%”を文字として出力する場合のみ、[変換指定子]を省略できます。(s1)に“%%”を格納し、文字列として出力すると、(d)に“%”が文字列として格納されます。なお、“%%”の後続に格納したデータは、次の表示形式まで文字として出力されます。

例

(s1)に“%d%%”を格納し、(s2)に指定したデバイスに“100”を格納した場合、(d)で指定したデバイスには“100%”が格納されます。



例

(s1)に“%%ABC%d”を格納し、(s2)に指定したデバイスに“100”を格納した場合、(d)で指定したデバイスには“%ABC100”が格納されます。

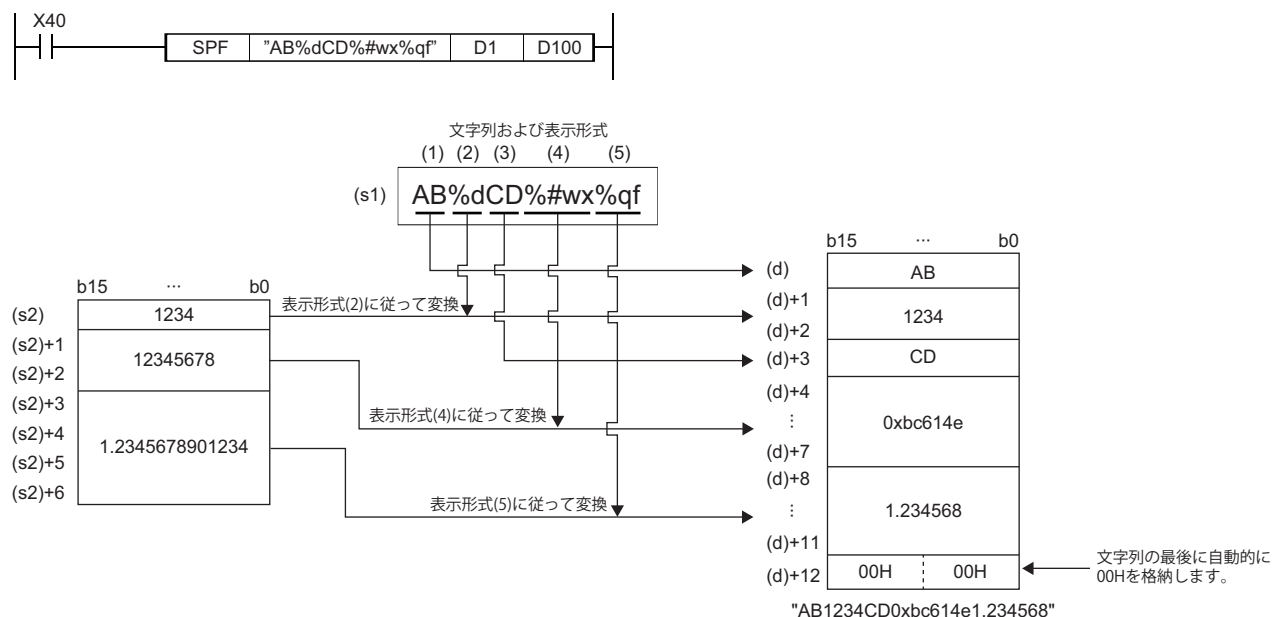


- 変換後文字列(d)が偶数バイトで終わる場合、文字列の最後の次のデバイス(上位8ビット、下位8ビット)にそれぞれNULLコード“00H”が格納されます。また、変換後文字列(d)が奇数バイトで終わる場合、文字列の最後のデバイス(上位8ビット)にNULLコード“00H”が格納されます。

例

下記が格納されている状態で、SPF(P)命令を実行した場合、D100以降に格納される文字列は、“AB1234CD0xbc614e1.234568”となります。

オペランド	格納されているデータ	
(s1)	AB%dCD%#wx%qf(文字列) ・格納された文字列は、下記の表示形式を指定しています。 AB(文字列を出力)%d(D1の表示形式)CD(文字列を出力)%#wx(D2~D3の表示形式)%qf(D4~D7の表示形式)	
(s2)	D1	1234(10進数)
	D2~D3	12345678(10進数)
	D4~D7	1.2345678901234(倍精度実数)
(d)	D100	



- (1): 文字列ABを出力します。
- (2): 表示形式は[取得データサイズ]=(記載省略), [変換指定子]=dとなります。(s2)から1ワード単位でデータ取得し、符号付き10進数で出力します。
- (3): 文字列CDを出力します。
- (4): 表示形式は[フラグ]=#, [取得データサイズ]=w, [変換指定子]=xとなります。(s2)+1から2ワード単位でデータを取得し、先頭に"0x"を付加し、符号なし16進数で出力します。
- (5): 表示形式は[精度]=(記載省略), [取得データサイズ]=q, [変換指定子]=fとなります。(s2)+3から4ワード単位でデータを取得し、倍精度実数の小数表現で出力します。また、[精度]の記載を省略したため、小数点以下6桁で出力し、出力文字以下1桁目の数値を四捨五入します。

■フラグ

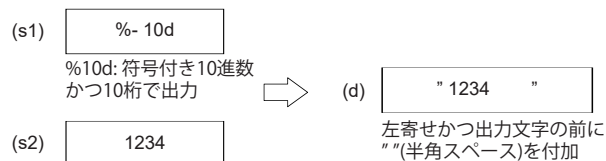
条件によって、(d)に出力する文字列に修飾をします。

指定子	説明
(記載省略)	右寄せで出力します。
-	[最小フィールド幅]が(s2)から取得して変換した文字数より大きい場合、左に寄せて出力します。 (s1) <code>%-10d</code> %10d: 符号付き10進数 かつ10桁で出力 ⇒ (d) <code>"1234 "</code> 左寄せで出力 (s2) <code>1234</code>
+	0または正の数を入力する場合、"+"を付加します。
" "(半角スペース)	0または正の数を入力する場合、" "(半角スペース)を付加します。
#	設定した[変換指定子]によって、下記の出力形式で出力します。 • "o": 先頭に"0"を付加します。 • "x", "X": 先頭に"0x"または"0X"を付加します。 • "f", "e", "E": 実数出力時に必ず小数点を出力します。 (s1) <code>%#.0wf</code> %.0wf: 単精度実数 小数表現かつ小数点 以下0桁で出力 ⇒ (d) <code>"1."</code> "."(ピリオド)を出力 (s2) <code>1.234</code> (s2)+1 • "g", "G": 実数出力時に必ず小数点を出力し、末尾に有効桁数分"0"を格納します。 (s1) <code>%#wg</code> %wg: 単精度実数 小数表現かつ小数点 以下の"0"未出力 ⇒ (d) <code>"1.00000"</code> [精度]が省略されたため 末尾の"0"を 有効桁数(6桁)まで出力 (s2) <code>1.0</code> (s2)+1
0	[最小フィールド幅]が(s2)から取得して変換した文字数より大きい場合、出力文字列の前に"0"を出力します。 (s1) <code>%#06x</code> %6x: 16進数 かつ6桁で出力 ⇒ (d) <code>"0x00ff"</code> 出力文字列の前に "0"を出力 (s2) <code>255</code>

[フラグ]は複数指定できます。使用する順番に制約はありません。ただし、"+"および" "(半角スペース)を同時に使用した場合、"+"の指定子の内容を優先し出力します。また、同じ指定子を使用した場合、2つ目以降の指定は無視されます。"-および"0"の指定子を同時に指定した場合、"0"の指定子は無視されます。

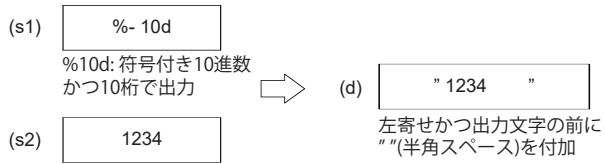
例

"-", " "(半角スペース)を使用した場合



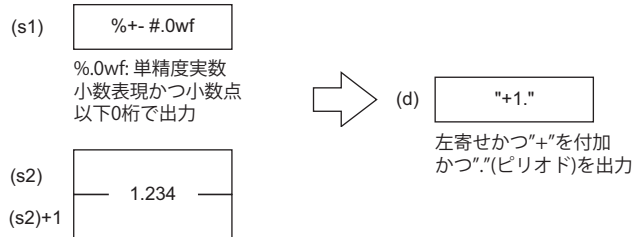
例

"#", "-"を使用した場合



例

"+", "-", "(半角スペース)および"#"を使用した場合



■最小フィールド幅

変換した文字列の符号, 小数点, 指数部および"(半角スペース)を含む, (d)に出力する全体の出力文字数を指定します。

指定子	説明
数値	- 記載省略時は取得データの文字数分を出力します。 (s1) <code>%d</code> %d: 符号付き 10進数出力 (d) <code>"1234"</code> 数値の記載を省略したため, 出力文字数分(4桁)を出力 (s2) <code>1234</code> [最小フィールド幅]が(s2)から取得して変換した文字数以下の場合, [最小フィールド幅]の指定は無視されます。 (s1) <code>%3d</code> %d: 符号付き 10進数出力 (d) <code>"1234"</code> 出力文字数分(4桁)を出力 (s2) <code>1234</code> [最小フィールド幅]が(s2)から取得して変換した文字数より大きい場合, 数値の前に"(半角スペース)を出力します。 (s1) <code>%10d</code> %d: 符号付き 10進数出力 (d) <code>" 1234 "</code> 出力文字の前に "(半角スペース)を出力 (s2) <code>1234</code>

■精度

(s2)に格納されたデータから、[変換指定子]によって(d)に出力する文字数を指定します。

精度を表すには"."(ピリオド)の後ろに数値を記載します。"."(ピリオド)だけを記述して数値を省略すると、".0"を指定した場合と同じ動作をします。

指定子	説明
“(ピリオド)数値	■[変換指定子]が“c”の場合 [精度]の指定は無視されます。 ■[変換指定子]が“d”, “u”, “x”, “X”, “o”の場合 出力文字数を指定します。 - 指定した数値が(s2)から取得して変換した文字数以下の場合、[精度]の指定は無視されます。 (s1) %.3d %d: 符号付き 10進数出力 (d) 1234 出力文字数分(4桁)を出力 (s2) 1234 - 指定した数値が(s2)から取得して変換した文字数より大きい場合、出力文字列の前に“0”を出力します。 (s1) %.5d %d: 符号付き 10進数出力 (d) 01234 出力文字列の前に“0”を出力 (s2) 1234 [精度]の指定が“0”の場合、かつ(s2)に指定する変換データが“0”の場合、なにも出力されません。 (s1) %.0d %d: 符号付き 10進数出力 (d) (s2) 0 ■[変換指定子]が“f”, “e”, “E”の場合 小数点以下の出力文字数を指定します。 - 数値の記載省略時は小数点以下6桁で出力し、出力文字以下1桁目の数値を四捨五入します。 (s1) %.3wf %wf: 単精度実数 小数表現出力 (d) 1.235 数値の記載を省略したため、 小数点以下3桁で出力 (s2) 1.2345 (s2)+1 - 指定した数値が(s2)から取得して変換した文字数より大きい場合、“0”を付加して出力します。ただし、(s2)に指定する変換データによって、“0”以外の文字が出力される場合があります。なお、小数表現であれば出力文字列の後ろ、指数表現であれば出力文字列の後ろかつ“e”, “E”の前に0を付加します。 (s1) %+-#.0wf %we: 単精度実数 指数表現出力 (d) 1.23400e+00 出力文字列の後ろかつ “e”の前に“0”を出力 (s2) 1.234 (s2)+1

指定子	説明																																			
“(ピリオド)数値	■[変換指定子]が“g”, “G”の場合 有効桁数を指定します。 • 数値の記載省略時は有効桁数6桁を出力し、出力文字以下1桁目の数値を四捨五入します。 (s1) <table><tr><td>%5d</td></tr></table> %d: 符号付き 10進数出力 ⇒ (d) <table><tr><td>"01234"</td></tr></table> 出力文字列の前に“0”を出力 (s2) <table><tr><td>1234</td></tr></table> • [精度]を“.0”に指定した場合, “.1”に指定した場合と等しくなります。 (s2)に1.234を指定した場合は下記のように出力されます。 (s1) <table><tr><td>%0qg</td></tr></table> %qg: 倍精度実数 指数表現出力 ⇒ (d) <table><tr><td>"1"</td></tr></table> 有効数字1桁で出力 (s2) <table><tr><td>—</td></tr><tr><td>—</td></tr><tr><td>1.234</td></tr><tr><td>—</td></tr></table> (s2)+1 (s2)+2 (s2)+3 (s2)に0.1234を指定した場合は下記のように出力されます。 (s1) <table><tr><td>%0qg</td></tr></table> %qg: 倍精度実数 指数表現出力 ⇒ (d) <table><tr><td>"0.1"</td></tr></table> 有効数字1桁で出力 (s2) <table><tr><td>—</td></tr><tr><td>—</td></tr><tr><td>0.1234</td></tr><tr><td>—</td></tr></table> (s2)+1 (s2)+2 (s2)+3 ■[変換指定子]が“s”の場合 (s2)に格納した文字データ順に、指定した出力文字数分出力します。 • 出力文字数が(s2)で取得した文字データの文字数より大きい場合、(s2)で取得した文字列データの文字数分を出力します。 (s1) <table><tr><td>%10s</td></tr></table> ⇒ (d) <table><tr><td>"ABCDE"</td></tr></table> (s2)より取得した 文字データの 文字数分(5文字)を出力 (s2) <table><tr><td>b15 …… b8</td><td>b7 …… b1</td></tr><tr><td>42H(B)</td><td>41H(A)</td></tr><tr><td>44H(D)</td><td>43H(C)</td></tr><tr><td>00H</td><td>45H(E)</td></tr></table> (s2)+1 (s2)+2 • 出力文字数が(s2)で取得した文字データの文字数より小さい場合、出力文字数の文字数分を出力します。 (s1) <table><tr><td>%3s</td></tr></table> ⇒ (d) <table><tr><td>"ABC"</td></tr></table> 出力文字数分(3文字)を出力 (s2) <table><tr><td>b15 …… b8</td><td>b7 …… b1</td></tr><tr><td>42H(B)</td><td>41H(A)</td></tr><tr><td>44H(D)</td><td>43H(C)</td></tr><tr><td>00H</td><td>45H(E)</td></tr></table> (s2)+1 (s2)+2	%5d	"01234"	1234	%0qg	"1"	—	—	1.234	—	%0qg	"0.1"	—	—	0.1234	—	%10s	"ABCDE"	b15 …… b8	b7 …… b1	42H(B)	41H(A)	44H(D)	43H(C)	00H	45H(E)	%3s	"ABC"	b15 …… b8	b7 …… b1	42H(B)	41H(A)	44H(D)	43H(C)	00H	45H(E)
%5d																																				
"01234"																																				
1234																																				
%0qg																																				
"1"																																				
—																																				
—																																				
1.234																																				
—																																				
%0qg																																				
"0.1"																																				
—																																				
—																																				
0.1234																																				
—																																				
%10s																																				
"ABCDE"																																				
b15 …… b8	b7 …… b1																																			
42H(B)	41H(A)																																			
44H(D)	43H(C)																																			
00H	45H(E)																																			
%3s																																				
"ABC"																																				
b15 …… b8	b7 …… b1																																			
42H(B)	41H(A)																																			
44H(D)	43H(C)																																			
00H	45H(E)																																			

■取得データサイズ

(s2)に格納されたデータの何ワード分変換し、(d)に出力するか指定します。

指定子	説明
(記載省略)	(s2)を1ワード単位で取得します。 (s1) %d %d: 符号付き 10進数出力 ⇒ (d) "722" (s2) 02D2H ← 1ワード分のデータを変換 (s2)+1 4996H
w	(s2)を2ワード単位で取得します。 (s1) %wd %d: 符号付き 10進数出力 ⇒ (d) "1234567890" (s2) 02D2H ← 2ワード分のデータを変換 (s2)+1 4996H
q	(s2)を4ワード単位で取得します。

■変換指定子

(d)に出力する文字列の出力形式を指定します。

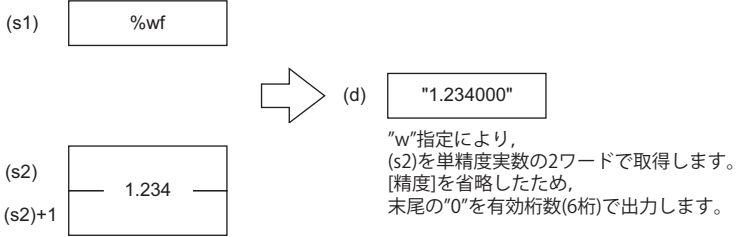
指定子	説明																																								
d	符号付き10進数で出力します。																																								
u	符号なし10進数で出力します。																																								
x	符号なし16進数で出力します。10~15の数は"a"~"f"で表示します。																																								
X	符号なし16進数で出力します。10~15の数は"A"~"F"で表示します。																																								
o	符号なし8進数で出力します。																																								
s	(s2)に格納された文字列データの00H(NULL)まで取得し、出力します。00H(NULL)がない場合、エラー (エラーコード: 2820H)になります。 (s1) %s ⇒(d) "ABCD" (s2) <table><tr><td>b15</td><td>b8</td><td>b7</td><td>b1</td></tr><tr><td>42H(B)</td><td></td><td>41H(A)</td><td></td></tr><tr><td>(s2)+1</td><td>44H(D)</td><td></td><td>43H(C)</td></tr><tr><td>(s2)+2</td><td>45H(E)</td><td>00H</td><td></td></tr><tr><td>(s2)+3</td><td>47H(G)</td><td></td><td>46H(F)</td></tr></table> ← NULLまで取得 00H(NULL)の格納位置が下位1バイト(b1~b7)の場合、次の変換データは後続のデバイスに指定したデータから有効になります。 (s1) %s%c ⇒(d) "ABCD F" (s2) <table><tr><td>b15</td><td>b8</td><td>b7</td><td>b1</td></tr><tr><td>42H(B)</td><td></td><td>41H(A)</td><td></td></tr><tr><td>(s2)+1</td><td>44H(D)</td><td></td><td>43H(C)</td></tr><tr><td>(s2)+2</td><td>45H(E)</td><td>00H</td><td></td></tr><tr><td>(s2)+3</td><td>47H(G)</td><td>46H(F)</td><td></td></tr></table> ← %cで指定する変換データ	b15	b8	b7	b1	42H(B)		41H(A)		(s2)+1	44H(D)		43H(C)	(s2)+2	45H(E)	00H		(s2)+3	47H(G)		46H(F)	b15	b8	b7	b1	42H(B)		41H(A)		(s2)+1	44H(D)		43H(C)	(s2)+2	45H(E)	00H		(s2)+3	47H(G)	46H(F)	
b15	b8	b7	b1																																						
42H(B)		41H(A)																																							
(s2)+1	44H(D)		43H(C)																																						
(s2)+2	45H(E)	00H																																							
(s2)+3	47H(G)		46H(F)																																						
b15	b8	b7	b1																																						
42H(B)		41H(A)																																							
(s2)+1	44H(D)		43H(C)																																						
(s2)+2	45H(E)	00H																																							
(s2)+3	47H(G)	46H(F)																																							
c	(s2)に格納された文字列データを1ワード単位でアスキーコード(10進数)として取得し、文字変換して出力します。 (s1) %c ⇒(d) "A" アスキーコードを文字で出力 (s2) 65 (s2)に格納した変換データは下位1バイト分(b1~b7)のみ有効になります。 (s1) %c ⇒(d) "A" 下位1バイト分の文字("A")のみ出力 (s2) 16961 "AB"を示すアスキーコード(10進数)																																								
f	小数表現で出力します。単精度実数時は7桁、倍精度実数時は15桁を最大有効桁数とします。																																								
e	指数表現で出力し、指数部は"e"で表示します。 単精度実数時は7桁、倍精度実数時は15桁を最大有効桁数とします。																																								
E	指数表現で出力し、指数部は"E"で表示します。 単精度実数時は7桁、倍精度実数時は15桁を最大有効桁数とします。																																								

指定子	説明
g	数値の大きさに応じて小数表現および指数表現で出力します。指数部は"e"で表示します。ただし、(s2)に指定した変換データの指数部の値が[精度]に指定した数値以上の場合は、指数表現で出力します。 - 数値の絶対値が0.0001以上: 小数表現 - 数値の0.0001未満: 指数表現 単精度実数時は7桁、倍精度実数時は15桁を最大有効桁数とします。 ■(s2)に指定した変換データが0.0001以上、かつ小数表現で出力する場合 (s1) %wg [精度]が省略されたため、有効桁数6桁で出力 (d) "0.0001" (s2)の指数部-4が[精度]の有効桁数よりも小さいため 小数表現で出力 (s2) 0.0001 (1e-4) (s2)+1 ■(s2)に指定した変換データが0.0001以上、かつ指数表現で出力する場合 (s1) %qg [精度]が省略されたため、有効桁数6桁で出力 (d) "1.23457e+06" (s2)の指数部6が[精度]の有効桁数6桁以上のため 指数表現で出力 (s2) 1234567.12 (1.23456712e+6) (s2)+1 (s2)+2 (s2)+3
G	数値の大きさに応じて小数表現および指数表現で出力します。指数部は"E"で表示します。ただし、(s2)に指定した変換データの指数部の値が[精度]に指定した数値以上の場合は、指数表現で出力します。 - 数値の絶対値が0.0001以上: 小数表現 - 数値の0.0001未満: 指数表現 単精度実数時は7桁、倍精度実数時は15桁を最大有効桁数とします。 (s1) %wG (d) "-9.99E-05" 指数表現で出力 (s2) -0.0000999 (s2)+1

[変換指定子]で"f", "e", "E", "g", "G"を使用する場合, [取得データサイズ]の指定子によって出力形式を, 単精度実数または倍精度実数から選択できます。[取得データサイズ]が"w"指定時は単精度実数, "q"指定時は倍精度実数で出力します。

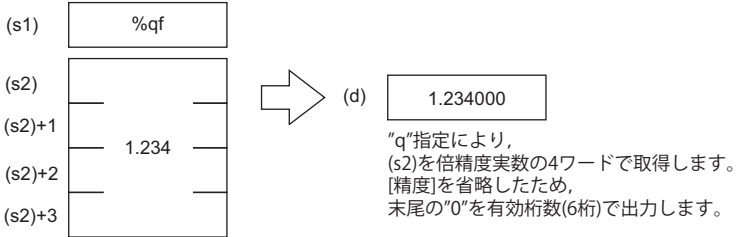
例

単精度実数で出力する場合



例

倍精度実数で出力する場合

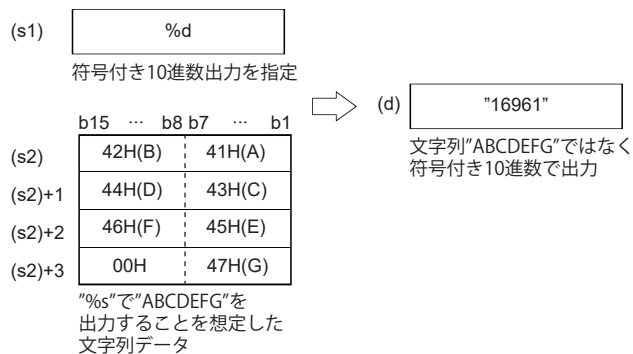


注意事項

表示形式の組み合わせと変換したいデータの形式は必ず一致させてください。意図しない文字列が出力されます。

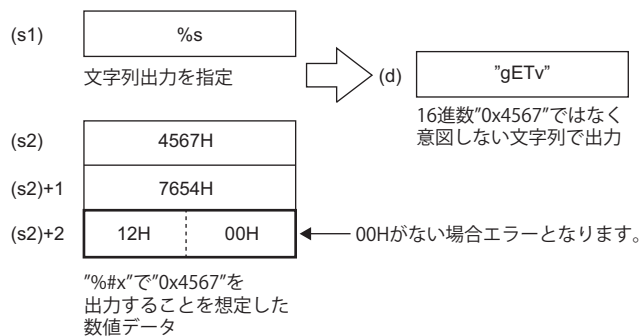
例

変換するデータを文字列データ、表示形式を数値データとした場合



例

変換するデータを数値データ、表示形式を文字列データとした場合

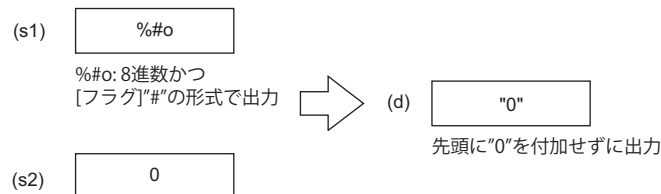


■フラグ

- [変換指定子]が"s"の場合、[フラグ]の指定は無視されます。
- [最小フィールド幅]が(s2)から取得して変換した文字数以下の場合、"- "の指定子は無視されます。
- [変換指定子]が"u", "x", "X", "o", "s", "c"の場合、"+ "と" "(半角スペース)の指定子は無視されます。
- [変換指定子]が"d", "u", "s", "c"の場合、または[変換指定子]に"o", "x", "X"を指定し、(s2)に"0"を指定した場合、"# "の指定子は無視されます。

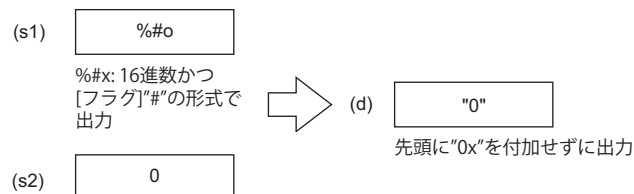
例

[変換指定子]に"o"を指定し、[フラグ]に"# "を指定した場合、下記のように出力されます。



例

[変換指定子]に"x"を指定し、[フラグ]に"# "を指定した場合、下記のように出力されます。



- "+ "および" "(半角スペース)使用時、0 または正の数以外を出力する場合、無視されます。
- "+ "が指定されていない、かつ負の数を出力する場合、先頭に"- "が付与されます。

■最小フィールド値

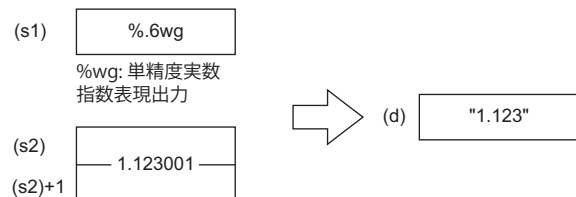
[最小フィールド幅]が255文字を超えて指定された場合、エラー (エラーコード: 329CH)になります。

■精度

- [精度]が255文字を超えて指定した場合、エラー (エラーコード: 329CH)になります。
- [変換指定子]が"g", "G"の場合、小数点以下の末尾の"0"は出力されません。また、小数点以下に数値がない場合、"."(ピリオド)も表示されません。

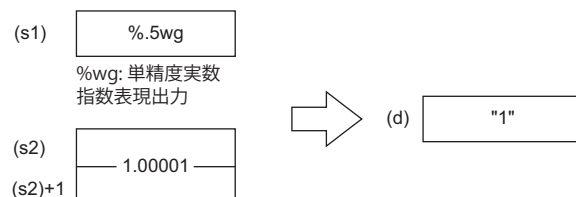
例

有効桁数6桁の場合、小数点以下の末尾の0(小数点以下第4位と第5位の0)は未出力になります。



例

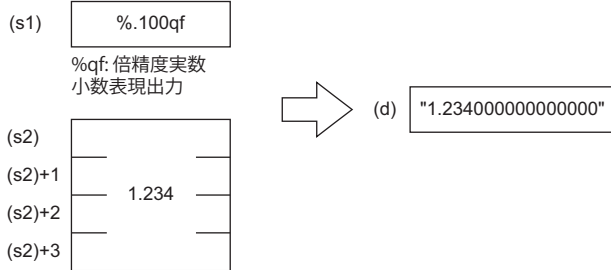
有効桁数5桁の場合、小数点以下がすべて0(数値がない)になるため、"."(ピリオド)は未出力になります。



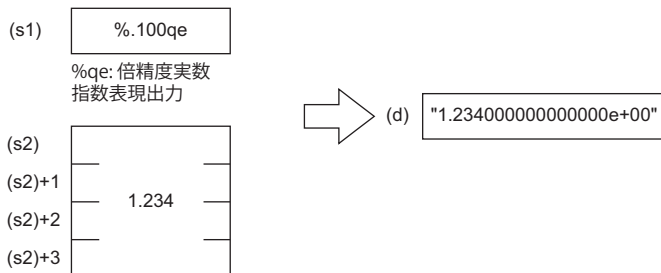
- [変換指定子]が"f", "e", "E", "g", "G"の場合、[精度]で指定した数値が出力結果の最大有効桁数より大きい場合、最大有効桁数で出力されます。

例

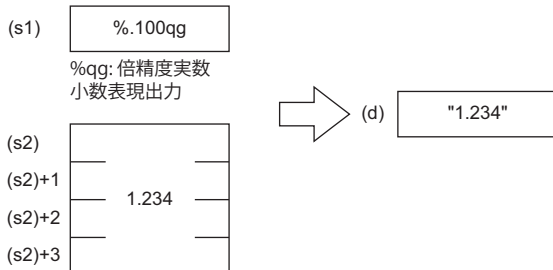
[変換指定子]に"f"を指定し、[精度]で指定した数値が倍精度実数の最大有効桁数より大きい場合、倍精度実数の最大有効桁数15桁で出力されます。

**例**

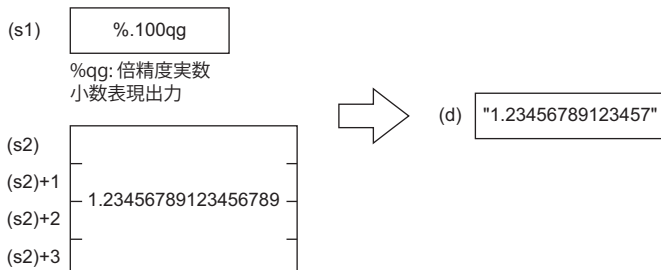
[変換指定子]に"e"を指定し、[精度]で指定した数値が倍精度実数の最大有効桁数より大きい場合、倍精度実数の最大有効桁数15桁で出力されます。

**例**

[変換指定子]に"g"を指定し、[精度]で指定した数値が倍精度実数の最大有効桁数より大きい、かつ(s2)に格納されている数値の桁数が最大有効桁数以下の場合、小数点以下の末尾の"0"は出力されません。

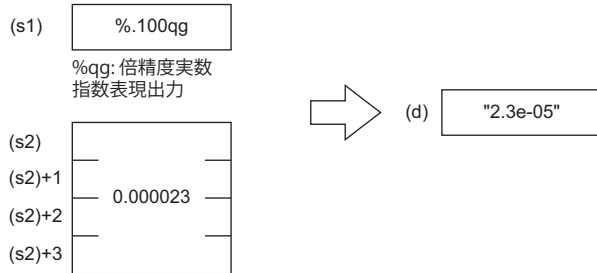
**例**

[変換指定子]に"g"を指定し、[精度]で指定した数値が倍精度実数の最大有効桁数より大きい、かつ(s2)に格納されている数値の桁数が最大有効桁数以上の場合、16桁目を四捨五入して最大有効桁数15桁で出力されます。



例

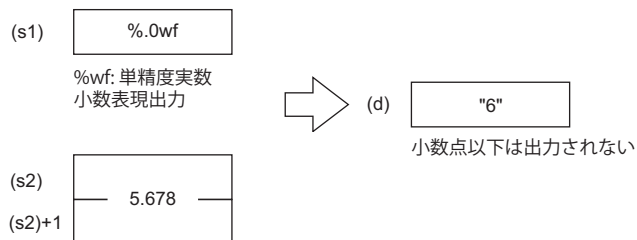
[変換指定子]に"q"を指定し、[精度]で指定した数値が倍精度実数の最大有効桁数より大きい、かつ指数表現で出力する場合、小数点以下の末尾"0"は出力されません。



- [変換指定子]に"f", "e", "E"を指定時、[精度]の指定が".0", かつ[フラグ]に"#"を指定しない場合、小数点以下は出力されません。

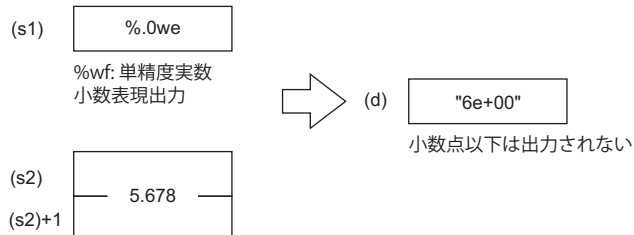
例

[変換指定子]が"f"の場合、[フラグ]を".0"に指定すると下記のように表現されます。



例

[変換指定子]が"e"の場合、[フラグ]を".0"に指定すると下記のように表現されます。



- [変換指定子]が"d", "f", "e", "E", "g", "G"の場合、変換した文字列の符号は[精度]で指定された出力文字数に含まれません。

■取得データサイズ

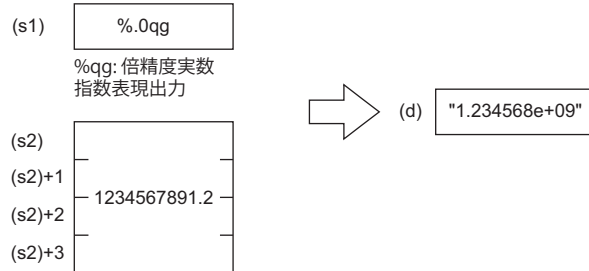
- [変換指定子]が"s", "c"の場合, [取得データサイズ]の指定は無視されます。
- [変換指定子]が"f", "e", "E", "g", "G"の場合, [取得データサイズ]は必ず"w"または"q"を指定してください。記載を省略した場合, エラー (エラーコード: 3299H)となります。

■変換指定子

指数表現で出力した場合, 指数は2桁以上で出力されます。

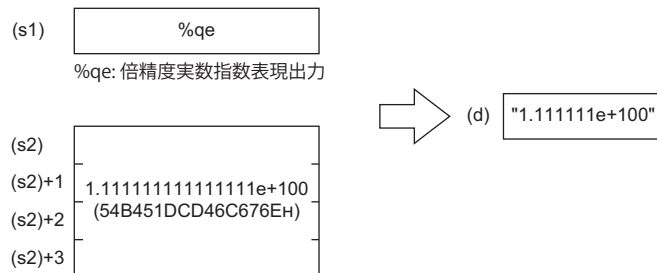
例

(s2)に格納されたデータを指数表現した場合の指数部の桁数が2桁の場合, 下記のように指数部は2桁で出力されます。



例

(s2)に格納されたデータを指数表現した場合の指数部の桁数が3桁の場合, 下記のように指数部は3桁で出力されます。



エラー

エラーコード (SD0)	内容
2820H	(s2)で指定したデバイス番号から, 該当デバイスの最終デバイス番号までの間に, 00Hが存在しないとき。 ([変換指定子]に"s"を指定した場合)
	(s2)+[取得データサイズ]で指定したサイズが, (s2)の最終デバイス番号を超えるととき。 ([変換指定子]に"s"以外を指定した場合)
	(s1)で指定したデバイス番号から, 該当デバイスの最終デバイス番号までの間に, 00Hが存在しないとき。
2821H	(s1)と(d)で指定された文字列の格納デバイス番号が重複しているとき。
3285H	(s1)の文字列が, 255文字を超えるととき。
3298H	(s1)に, 下記のように表示形式を使用したとき。 • 使用可能な指定子以外を使用 • 表示形式を指定された形式以外の並びで使用 • [精度], [取得データサイズ]を1つの表示形式内で複数回使用
3299H	(s1)に表示形式の指定がないとき。
329AH	(s2)に指定された文字列が255文字を超えるととき。 ([変換指定子]に"s"を指定した場合)
3286H	(d)に格納する文字列が出力可能な範囲外のデータとなったとき。 • 結合した文字列が255文字を超えた。 • 指定したデバイス番号以降, 該当デバイス/ラベルの割付範囲に変換した文字列がすべて格納できないとき。
329CH	[最小フィールド幅]または[精度]が, 255文字を超えて指定されたとき。

Unicode文字列→シフトJIS文字列変換

WS2SJIS(P)

Unicode文字列をシフトJIS文字列に変換します。

ラダー	ST
	ENO:=WS2SJIS(EN,s,d); ENO:=WS2SJISP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
WS2SJIS	
WS2SJISP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	変換する文字列が格納されている先頭デバイス	—	Unicode文字列	ANYSTRING_DOUBLE
(d)	変換後の文字列を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

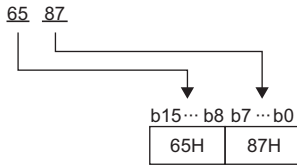
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- (s)で指定したUnicode文字列をシフトJIS文字列に変換し、(d)へ格納します。
- (s)のUnicode文字列は、リトルエンディアンで指定してください。

例

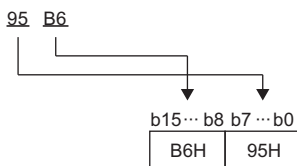
Unicode文字列の「6587H」は、「6587H」で指定します。



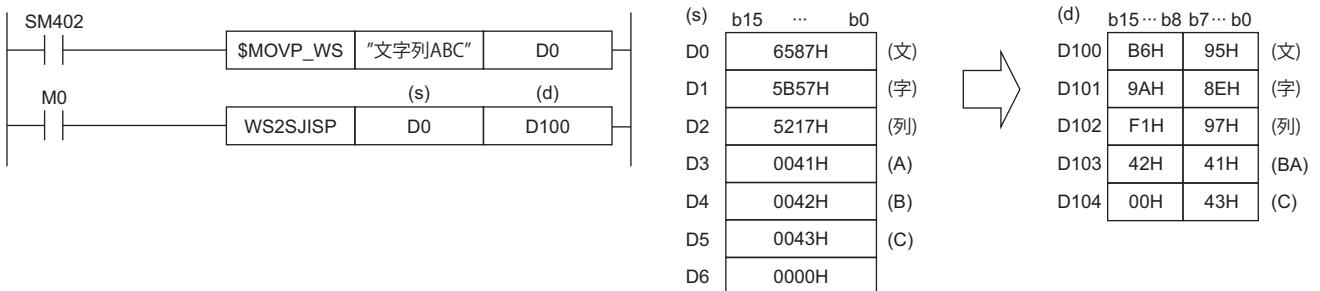
- (d)のシフトJIS文字列は、ビッグエンディアンで格納します。

例

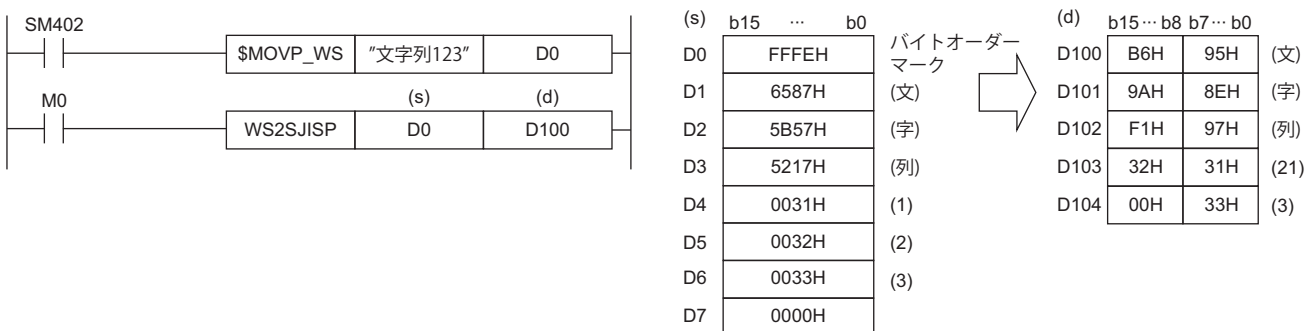
シフトJIS文字列の「95B6H」は、「B695H」で指定します。



- バイトオーダーマークが無い場合のUnicode→シフトJIS変換の動作は下記になります。



- バイトオーダーマーク(FFFEH)が付加されている場合のUnicode→シフトJIS変換の動作は下記になります。



エラー

エラーコード (SD0)	内容
2821H	(s), (d)で指定した範囲が重複している場合。
3281H	(s)で指定した文字列に、バイトオーダーマークのFFFEH(ビッグエンディアン)が付加されている場合。 (s)で指定した範囲に、変換できない文字コードが含まれている場合。
3285H	(s)で指定した文字列が、16383文字を超える場合。 ^{*1}
3286H	(d)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に変換した文字列がすべて格納できないとき。

*1 漢字などのシフトJISコードで表現した場合に2バイトデータになる文字は、1文字あたり2とカウントしてください。

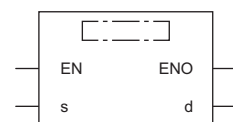
シフトJIS文字列→Unicode文字列変換(バイトオーダーマークなし)

SJIS2WS(P)

シフトJIS文字列をUnicode文字列に変換します。

ラダー	ST
	ENO:=SJIS2WS(EN,s,d); ENO:=SJIS2WSP(EN,s,d);

FBD/LD



■実行条件

命令	実行条件
SJIS2WS	
SJIS2WSP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	変換する文字列が格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	変換後の文字列を格納する先頭デバイス	—	Unicode文字列	ANYSTRING_DOUBLE
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

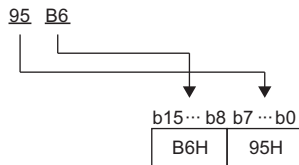
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- ・ (s)で指定したシフトJIS文字列をUnicode文字列に変換し、(d)へ格納します。
- ・ (s)のシフトJIS文字列は、ビッグエンディアンで指定してください。

例

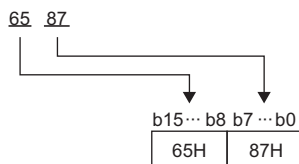
シフトJIS文字列の「95B6H」は、「B695H」で指定します。



- ・ (d)のUnicode文字列は、リトルエンディアンで格納されます。

例

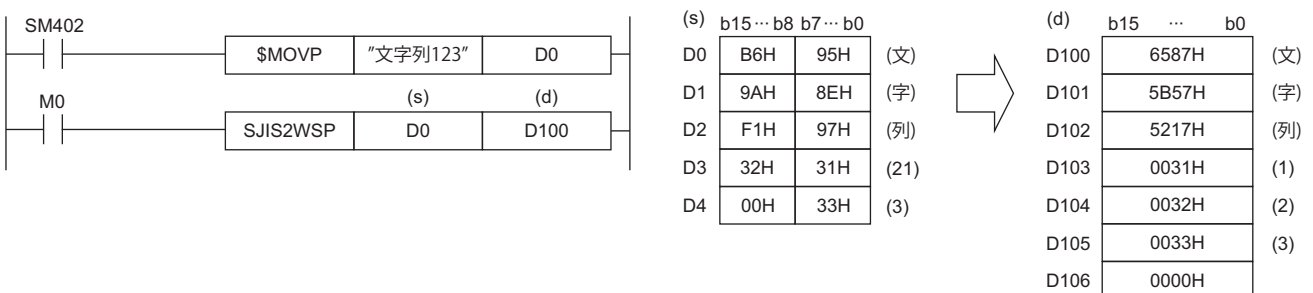
Unicode文字列の「6587H」は、「6587H」で指定します。



- ・ SJIS2WS(P)命令では、(d)の先頭にバイトオーダーマークを付加しません。バイトオーダーマークを付加したい場合は、SJIS2WSB(P)命令を使用してください。

788ページ SJIS2WSB(P)

- ・ シフトJIS→Unicode変換の動作は下記になります。



エラー

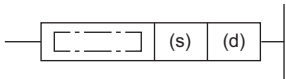
エラーコード (SD0)	内容
2821H	(s), (d)で指定した範囲が重複している場合。
3281H	(s)で指定した範囲に、変換できない文字コードが含まれている場合。
3285H	(s)で指定した文字列が、16383文字を超える場合。 ^{*1}
3286H	(d)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に変換した文字列がすべて格納できないとき。

*1 漢字などのシフトJISコードで表現した場合に2バイトデータになる文字は、1文字あたり2とカウントしてください。

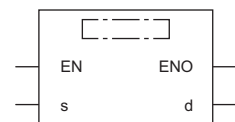
シフトJIS文字列→Unicode変換(バイトオーダーマークあり)

SJIS2WSB(P)

シフトJIS文字列をUnicode文字列に変換し、先頭にバイトオーダーマークを付加します。

ラダー	ST
	ENO:=SJIS2WSB(EN,s,d); ENO:=SJIS2WSBP(EN,s,d);

FBD/LD



■実行条件

命令	実行条件
SJIS2WSB	
SJIS2WSBP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	変換する文字列が格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	変換後の文字列を格納する先頭デバイス	—	Unicode文字列	ANYSTRING_DOUBLE
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

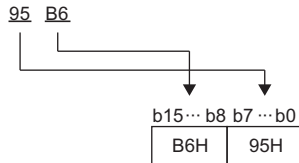
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- (s)で指定したシフトJIS文字列をUnicode文字列に変換し、先頭にバイトオーダーマークを付加して(d)へ格納します。
- (s)のシフトJIS文字列は、ビッグエンディアンで指定してください。

例

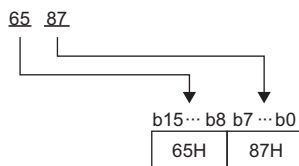
シフトJIS文字列の「95B6H」は、「B695H」で指定します。



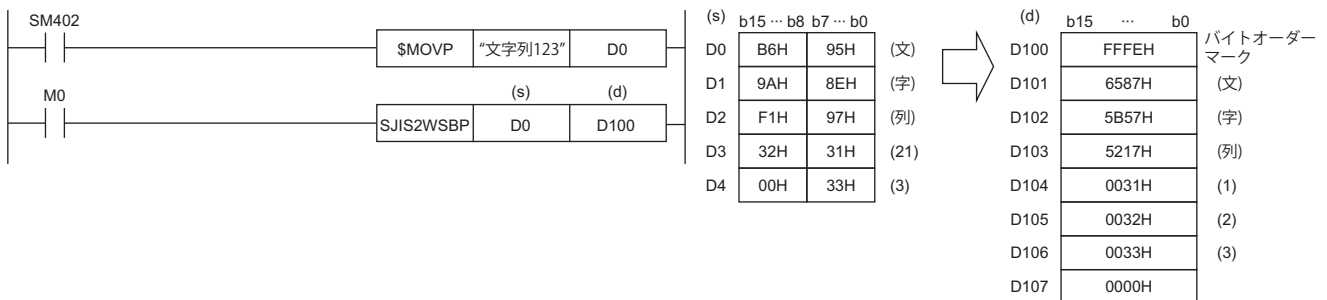
- (d)のUnicode文字列は、リトルエンディアンで格納されます。

例

Unicode文字列の「6587H」は、「6587H」で指定します。



- シフトJIS→Unicode変換の動作は下記になります。



エラー

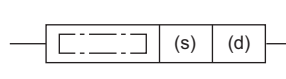
エラーコード (SD0)	内容
2821H	(s), (d)で指定した範囲が重複している場合。
3281H	(s)で指定した範囲に、変換できない文字コードが含まれている場合。
3285H	(s)で指定した文字列が、16383文字を超える場合。 ^{*1}
3286H	(d)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に変換した文字列がすべて格納できないとき。

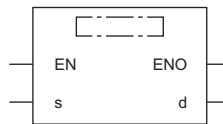
*1 漢字などのシフトJISコードで表現した場合に2バイトデータになる文字は、1文字あたり2とカウントしてください。

文字列の長さ検出

LEN(P)

指定した文字列の長さを検出します。

ラダー	ST*1
	ENO:=LENP(EN,s,d);

FBD/LD*1


*1 LEN命令はST, FBD/LDでは対応していません。汎用ファンクションのLENを使用してください。
📖 1306ページ LEN(_E)

■実行条件

命令	実行条件
LEN	
LENP	

設定データ

■内容, 範囲, データ型

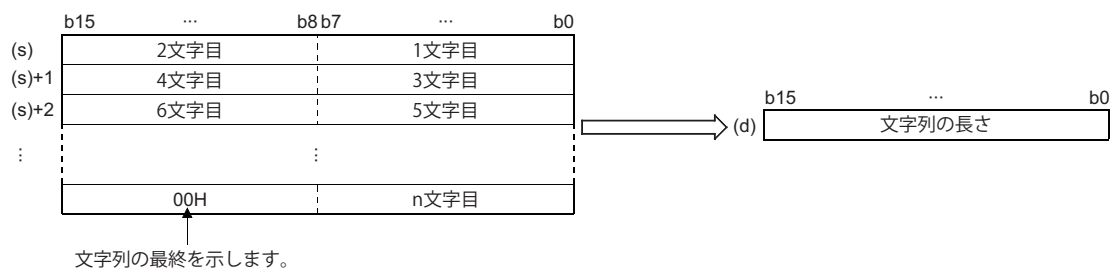
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	文字列または, 文字列が格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	検出した文字列の長さを格納するデバイス番号	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

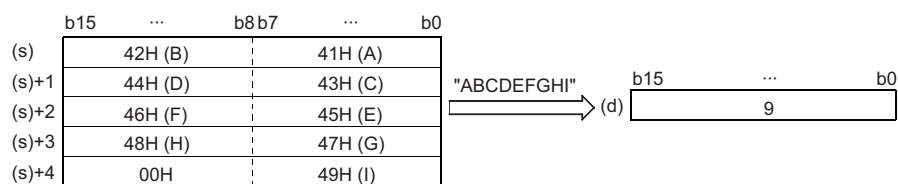
機能

- (s)で指定された文字列の長さを検出し、(d)で指定されたデバイス番号以降に格納します。(s)で指定されたデバイス番号から、00Hが格納されているデバイス番号までのデータを文字列として処理します。



例

(s)以降に「ABCDEFGHI」が格納されている場合



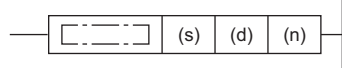
エラー

エラーコード (SD0)	内容
2820H	(s)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に00Hが設定されていないとき。
3285H	(s)の文字列が16383文字を超えるとき。

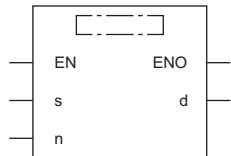
文字列の右側からの取出し

RIGHT(P)

文字列データの右から(n)文字分のデータを取り出します。

ラダー	ST*1
	ENO:=RIGHTP(EN,s,n,d);

FBD/LD*1



*1 RIGHT命令はST, FBD/LDでは対応していません。汎用ファンクションのRIGHTを使用してください。
📖 1308ページ LEFT(_E), RIGHT(_E)

■実行条件

命令	実行条件
RIGHT	
RIGHTP	

設定データ

■内容、範囲、データ型

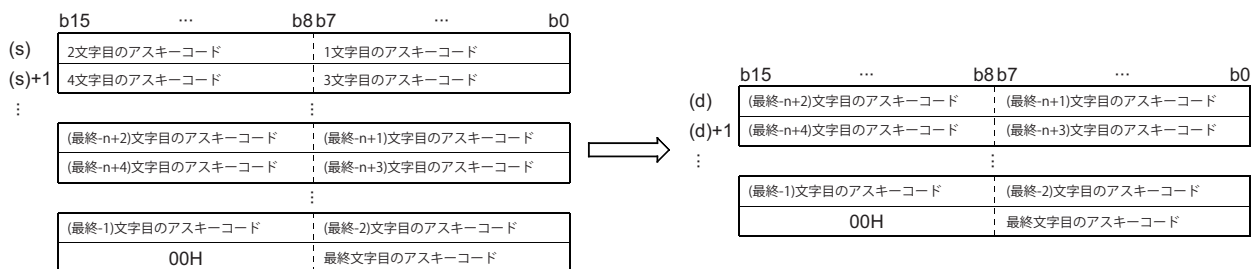
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	文字列または、文字列が格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	(s)の右から(n)文字分の文字列を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
(n)	取り出す文字数	1~16383	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□≠□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□≠G□, J□≠□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

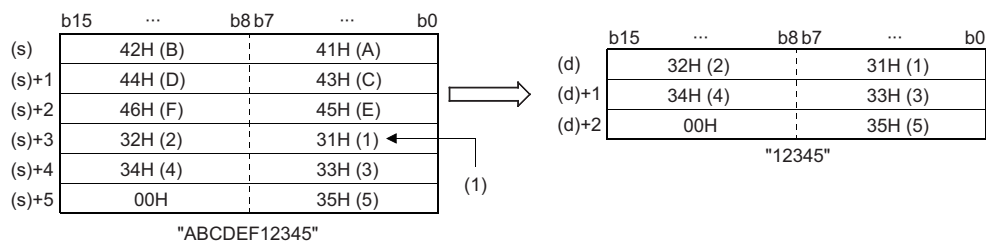
機能

- (s)で指定されたデバイス番号以降に格納されている文字列データに対して、文字列の右(文字列の最終)から(n)文字分のデータを、(d)で指定されたデバイス番号以降に格納します。



例

(n)=5の場合



(1): 5文字目のアスキーコード

- 文字列の最終を示すNULLコード(00H)が、文字列データの最後に自動付加されます。
- (n)で指定する文字数が0の場合は、(d)にNULLコード(00H)が格納されます。

エラー

エラーコード (SD0)	内容
2820H	(s)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に00Hが設定されていないとき。
3285H	(s)に指定可能な範囲外のデータを設定したとき。 • (s)の文字列が16383文字を超える • (s)の文字列が0文字 (n)が(s)で指定の文字数を超えているとき。
3286H	(d)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に取り出した文字列がすべて格納できないとき。

文字列の左側からの抽出

LEFT(P)

文字列データの左から(n)文字分のデータを(d)で指定されたデバイス番号以降に格納します。

ラダー	ST*1
	ENO:=LEFTP(EN,s,n,d);

FBD/LD*1

*1 LEFT命令はST, FBD/LDでは対応していません。汎用ファンクションのLEFTを使用してください。
📖 1308ページ LEFT(_E), RIGHT(_E)

■実行条件

命令	実行条件
LEFT	
LEFTP	

設定データ

■内容、範囲、データ型

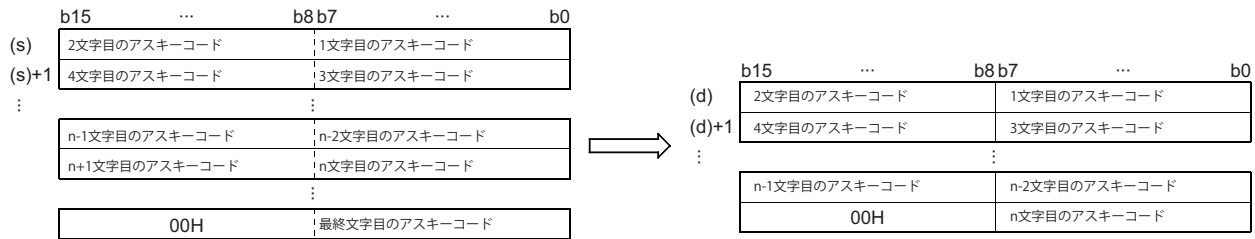
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	文字列または、文字列が格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	(s)の左から(n)文字分の文字列を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
(n)	取り出す文字数	1~16383	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	○	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

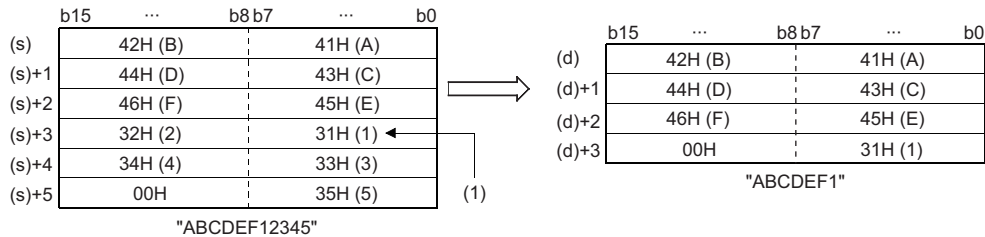
機能

- (s)で指定されたデバイス番号以降に格納されている文字列データに対して、文字列の左(文字列の先頭)から(n)文字分のデータを、(d)で指定されたデバイス番号以降に格納します。



例

(n)=7の場合



(1): 7文字目のアスキーコード

- 文字列の最終を示すNULLコード(00H)が、文字列データの最後に自動付加されます。
- (n)で指定する文字数が0の場合は、(d)にNULLコード(00H)が格納されます。

エラー

エラーコード (SD0)	内容
2820H	(s)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に00Hが設定されていないとき。
3285H	(s)に指定可能な範囲外のデータを設定したとき。 • (s)の文字列が16383文字を超える • (s)の文字列が0文字 (n)が(s)で指定の文字数を超えているとき。
3286H	(d)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に取り出した文字列がすべて格納できないとき。

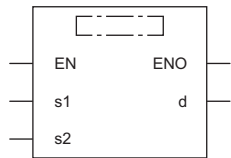
文字列中の任意取出し

MIDR(P)

文字列データ中の任意位置のデータを(d)で指定されたデバイス番号以降に格納します。

ラダー	ST
	ENO:=MIDR(EN,s1,s2,d); ENO:=MIDRP(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
MIDR	
MIDRP	

設定データ

■内容，範囲，データ型

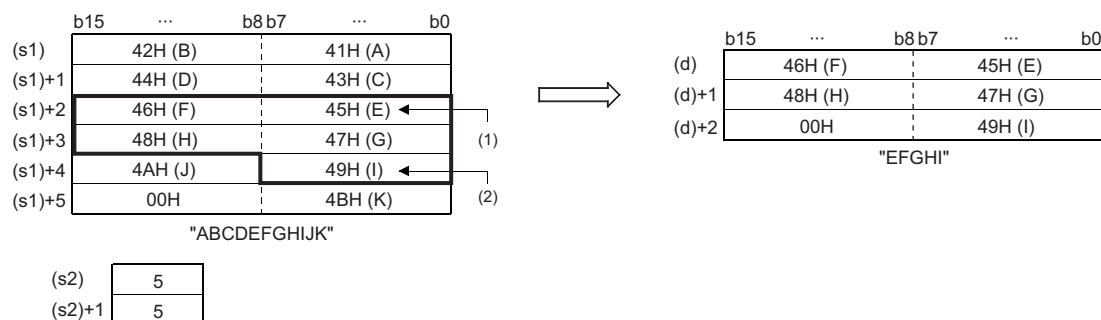
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	文字列または、文字列が格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	演算結果の文字列データを格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
(s2)	先頭文字の位置および文字数を格納する先頭デバイス (s2): 先頭文字の位置, (s2)+1: 文字数	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□≠□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□≠G□, J□≠□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	—	—	—	—

機能

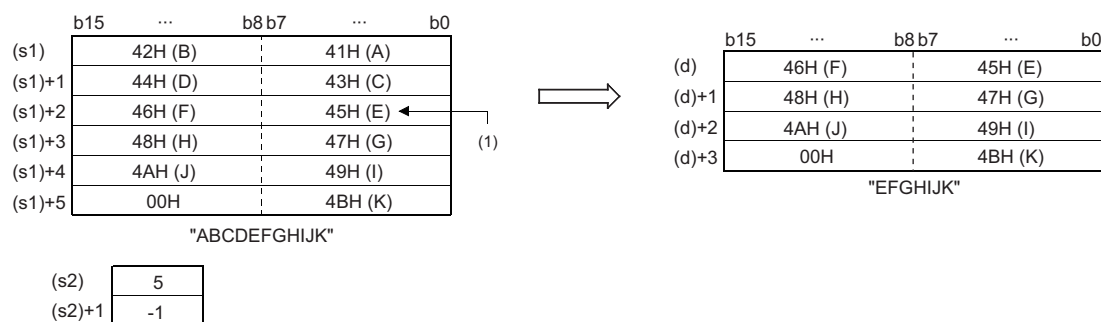
- (s1)で指定されたデバイス番号以降に格納されている文字列データに対して、(s2)で指定された位置から(s2)+1で指定された文字分のデータを、(d)で指定されたデバイス番号以降に格納します。



(1): 5文字目の位置(s2)

(2): 5文字目のアスキーコード(s2)+1

- 文字列の最終を示すNULLコード(00H)が、文字列データの最後に自動付加されます。
- (s2)+1で指定する文字数が0の場合は処理を行いません。
- (s2)+1で指定する文字数が-1の場合は、(s1)で指定された最終文字データまでを(d)で指定されたデバイス以降に格納します。



(1): 5文字目の位置(s2)

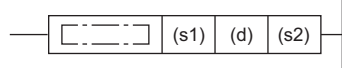
エラー

エラーコード (SD0)	内容
2820H	(s1)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に00Hが存在しないとき。
3285H	(s1)の文字列が16383文字を超えるととき。 (s2)に指定可能な範囲外のデータを設定したとき。 • (s2)の値が0以下 • (s2)+1の値が有効値(-1, 0, 1以上)以外 • (s2)の値が(s1)の文字数を超えている • (s2)と(s2)+1を加算した値が(s1)の文字数+1を超えている
3286H	(d)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に取り出した文字列がすべて格納できないとき。

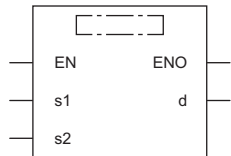
文字列中の任意置換え

MIDW(P)

文字列データ中の任意位置のデータを任意の文字列で置き換えます。

ラダー	ST
	ENO:=MIDW(EN,s1,s2,d); ENO:=MIDWP(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
MIDW	
MIDWP	

設定データ

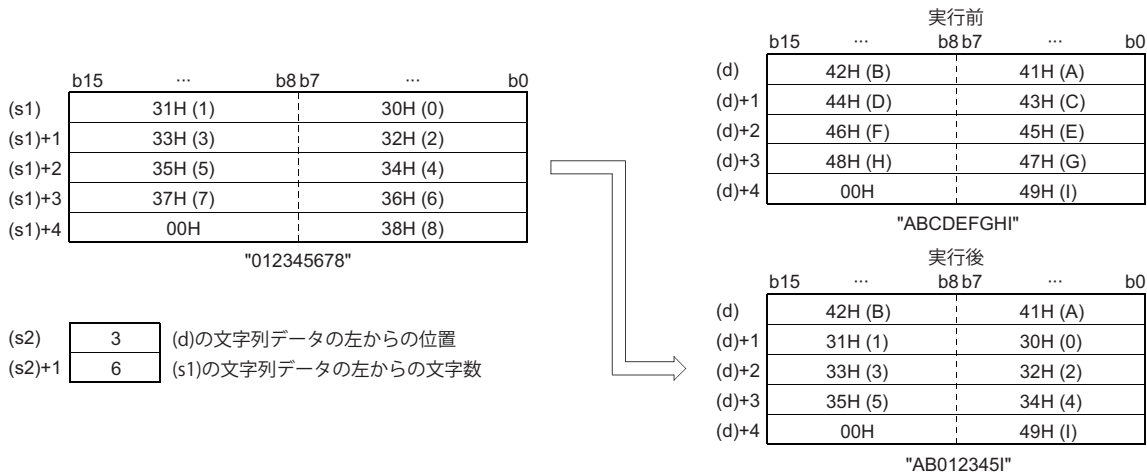
■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	文字列または、文字列が格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	演算結果の文字列データを格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
(s2)	先頭文字の位置および文字数を格納する先頭デバイス (s2): 先頭文字の位置, (s2)+1: 文字数	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

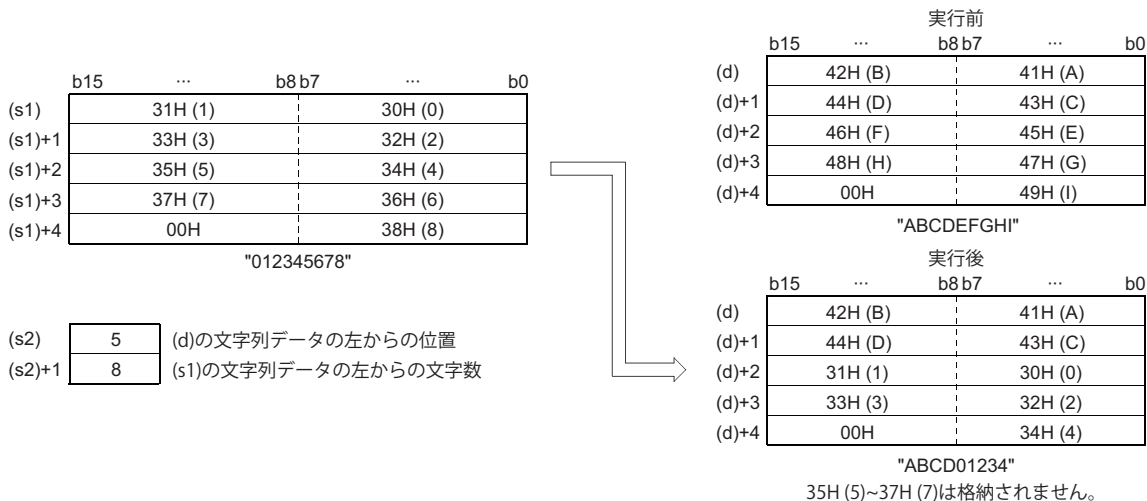
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□≠□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□≠G□, J□≠□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	—	—	—	—

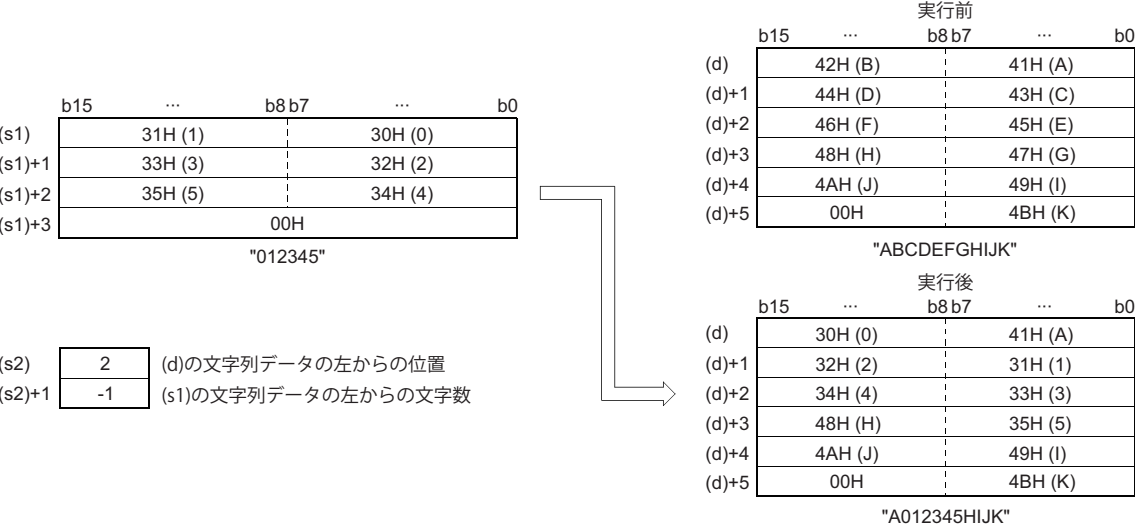
- (s1)で指定されたデバイス番号以降に格納されている文字列データに対して、(s2)+1で指定された文字分のデータを、(d)で指定されたデバイス番号以降に格納されている文字列データの、(s2)で指定された位置以降に格納します。



- 文字列の最終を示すNULLコード(00H)が、文字列データの最後に自動付加されます。
- (s2)+1で指定する文字数が0の場合は処理を行いません。
- (s2)+1で指定された文字数が、(d)で指定された文字列データの最終文字を超えてしまった場合は、(d)の最終文字目までデータを格納します。



- (s2)+1で指定する文字数が-1の場合は、(s1)で指定された最終文字データまでを(d)で指定されたデバイス以降に格納します。



エラー

エラーコード (SD0)	内容
2820H	(s1)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に00Hが存在しないとき。
	(d)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に00Hが存在しないとき。
3285H	(s1)の文字列が16383文字を超えるとき。
	(d)の文字列が16383文字を超えるとき。
	(s2)に指定可能な範囲外のデータを設定したとき。
	• (s2)の値が0以下 • (s2)+1の値が有効値(-1, 0, 1以上)以外 • (s2)の値が(d)の文字数を超えている • (s2)+1の値が(s1)の文字数を超えているとき。

INSTR(P)

文字列データ中の指定文字列を検索します。

ラダー	ST
	ENO:=INSTR(EN,s1,s2,s3,d); ENO:=INSTRP(EN,s1,s2,s3,d);

FBD/LD

■実行条件

命令	実行条件
INSTR	
INSTRP	

設定データ

■内容、範囲、データ型

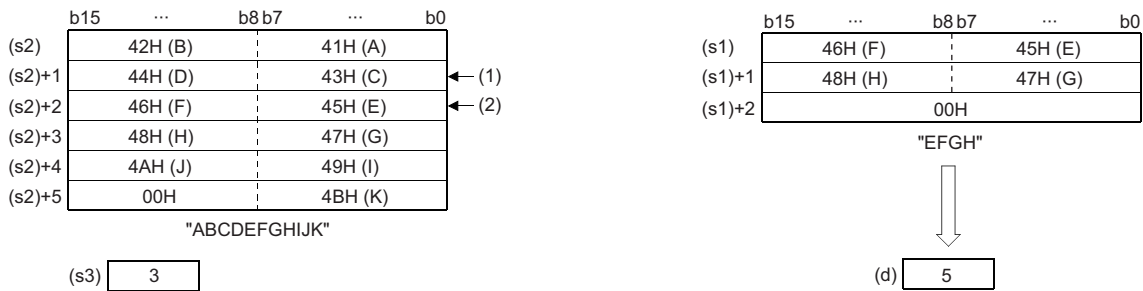
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	検索文字列または、検索文字列が格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(s2)	検索される文字列または、検索される文字列が格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	検索した結果を格納するデバイス	—	符号付きBIN16ビット	ANY16
(s3)	検索開始位置	1~16383	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	○	—
(s2)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—
(s3)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- (s2)で指定されたデバイス番号以降に格納されている文字列データの左の(s3)文字目から、(s1)で指定されたデバイス番号以降に格納されている文字列データを検索し、検索結果を(d)で指定されたデバイスに格納します。検索結果は、(s2)で指定された文字列データの先頭文字から何文字目かを格納します。



- (1): 検索開始位置(s3): 3文字目
(2): 先頭文字から5文字目

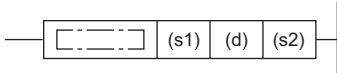
- 一致した文字列データが無い場合は、(d)に0を格納します。

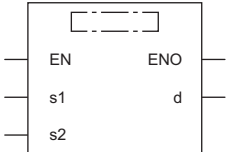
エラー

エラーコード (SD0)	内容
2820H	(s1)で指定のデバイス以降、該当デバイス/ラベルの割付範囲にNULLコード(00H)が無いとき。
	(s2)で指定のデバイス以降、該当デバイス/ラベルの割付範囲にNULLコード(00H)が無いとき。
3285H	(s1)に指定可能な範囲外のデータを設定したとき。 • 設定した文字列が0文字 • 設定した文字列が16383文字を超える
	(s2)に設定した文字列が16383文字を超えるとき。
	(s3)に指定可能な範囲外のデータを設定したとき。 • (s3)の値が(s2)の文字数を超えている • (s3)の値が負の数または0

STRINS(P)

任意の文字列データを，文字列データ任意挿入位置に挿入します。

ラダー	ST
	ENO:=STRINS(EN,s1,s2,d); ENO:=STRINSP(EN,s1,s2,d);

FBD/LD


■実行条件

命令	実行条件
STRINS	
STRINSP	

設定データ

■内容，範囲，データ型

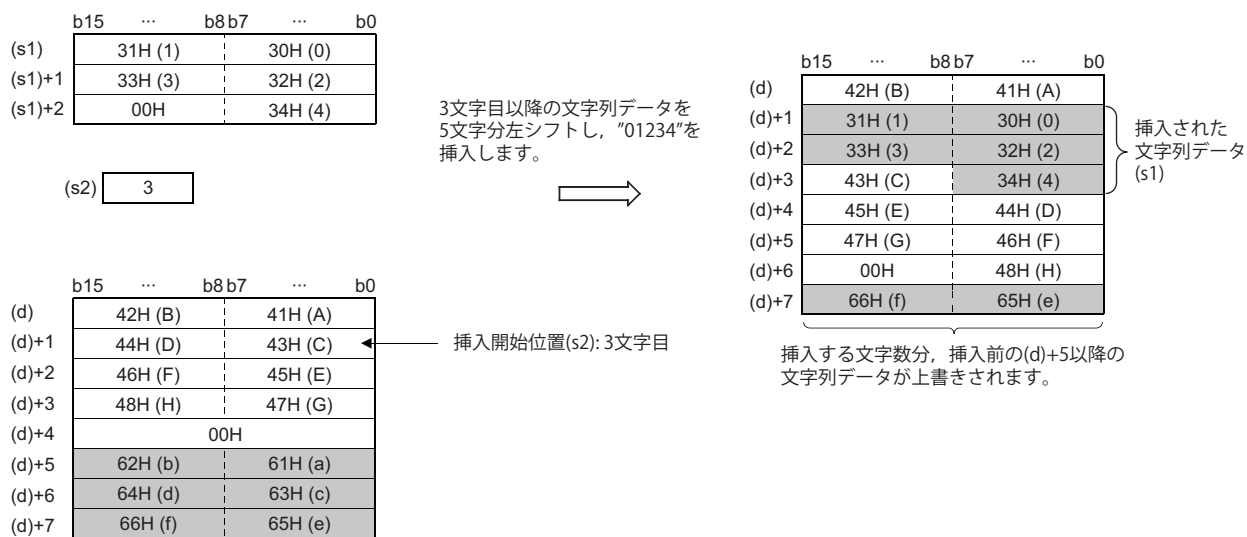
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	挿入文字列または，挿入文字列が格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	挿入文字列を格納する先頭デバイス	—	文字列	ANYSTRING_SINGLE
(s2)	挿入位置(バイト単位)	1~16383	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	○	—	—	—	—
(s2)	○	○	○	○	○	—	○	○	—	—	—

機能

- (s1)で指定された文字列データを、(d)で指定された文字列データの先頭から(s2)文字目(挿入位置)に挿入します。



- 挿入後の文字列(s1)+(d)が偶数の場合、文字列の最後の次のデバイス(1ワード)にNULLコード(00H)を格納します。
- 挿入後の文字列(s1)+(d)が奇数の場合、文字列の最後のデバイス(上位8ビット)にNULLコード(00H)を格納します。
- (s2)に(d)の文字数+1を指定した場合、(d)の文字列の最後に(s1)の文字列を結合します。

エラー

エラーコード (SD0)	内容
2820H	(s1)で指定のデバイス以降、該当デバイス/ラベルの割付範囲にNULLコード(00H)が無いとき。 (d)で指定のデバイス以降、該当デバイス/ラベルの割付範囲にNULLコード(00H)が無いとき。
2821H	文字列(s1)と文字列(d)のデバイスが一部でも重複しているとき。 挿入後の文字列(s1)+(d)が、(s1)の文字列格納デバイスと重複しているとき。
3285H	(s1)で指定した文字列の文字数が16383文字を超えているとき。 (s2)で指定可能な範囲外のデータを設定したとき。 - 指定した値が文字列(d)の文字数+1を超えている。 - 指定した値が下記範囲内でない。 $1 \leq (s2) \leq 16383$ (d)で指定した文字列の文字数が16383文字を超えているとき。
3286H	挿入後の文字列(s1)+(d)が出力可能な範囲外のデータとなったとき。 - 挿入後の文字列の文字数が16383文字を超えている。 (d)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に挿入後の文字列がすべて格納できないとき。

STRDEL(P)

文字列データの指定された位置より，(n)文字分を削除します。

ラダー	ST
	ENO:=STRDEL(EN,s,n,d); ENO:=STRDELP(EN,s,n,d);

FBD/LD

■実行条件

命令	実行条件
STRDEL	
STRDELP	

設定データ

■内容，範囲，データ型

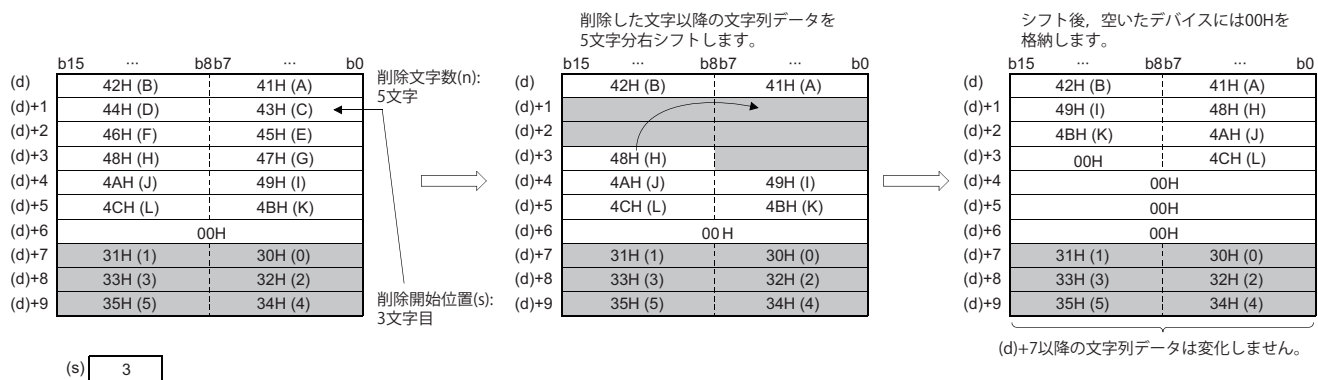
オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	削除文字列が格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(s)	削除開始位置	1~16383	符号付きBIN16ビット	ANY16
(n)	削除文字数	0~16384-(s)	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(s)	—	○	○	○	○	—	—	○	○	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- (d)で指定された文字列データの先頭から(s)文字目で指定された位置(削除開始位置)より, (n)文字分を削除します。



- 削除後, 文字列(d)が偶数の場合, 文字列の最後の次のデバイス(1ワード)にNULLコード(00H)を格納します。
- 削除後, 文字列(d)が奇数の場合, 文字列の最後のデバイス(上位8ビット)にNULLコード(00H)を格納します。
- 削除した文字列以降の文字列を(n)文字分右シフト後, 空いたデバイスにはNULLコード(00H)を格納します。

エラー

エラーコード (SD0)	内容
2820H	(d)で指定のデバイス以降, 該当デバイス/ラベルの割付範囲にNULLコード(00H)が無いとき。
3285H	(d)で指定した文字列の文字数が16383文字を超えているとき。 (s)に指定可能な範囲外のデータを設定したとき。 • 指定した値が下記範囲内でない $1 \leq (s) \leq 16383$ • 指定した値が文字列(d)の文字数を超えている (n)に指定可能な範囲外のデータを設定したとき。 • 指定した値が文字列(d)の(s)から最終文字までの文字数を超えている • 指定した値が負数

17.1 浮動小数点命令

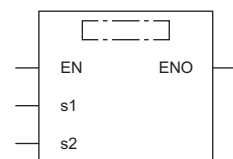
単精度実数比較

LDE□, ANDE□, ORE□

単精度実数をa接点扱いで比較演算します。

ラダー	ST
	<pre> ENO:=LDE_□(EN,s1,s2); ENO:=ANDE_□(EN,s1,s2); ENO:=ORE_□(EN,s1,s2); (□には, EQ, NE, GT, LE, LT, GEが入ります。)*1 </pre>
(□には, E=, E<>, E>, E<=, E<, E>=が入ります。)	

FBD/LD



(□には, LDE_, ANDE_, ORE_と, EQ, NE, GT, LE, LT, GEの組合せが入ります。)*1

*1 EQは=, NEは<>, GTは>, LEは<=, LTは<, GEは>=です。

■実行条件

命令	実行条件
LDE□, ANDE□, ORE□	常時実行

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	比較データまたは, 比較データが格納されている先頭デバイス	$0, 2^{-126} \leq (s1) < 2^{128}$	単精度実数	ANYREAL_32
(s2)	比較データまたは, 比較データが格納されている先頭デバイス	$0, 2^{-126} \leq (s2) < 2^{128}$	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	—	—	○	○	○	○	○	—	○	—	—
(s2)	—	—	○	○	○	○	○	—	○	—	—

機能

- (s1)で指定されたデバイスの単精度実数と、(s2)で指定されたデバイスの単精度実数をa接点扱いで比較演算します。
- 各命令の比較演算結果は下記ようになります。

命令記号(ラダー, FBD/LD)	条件	比較演算結果
E=, EQ	(s1)=(s2)	導通状態(ENOはON)
E<>, NE	(s1)≠(s2)	
E>, GT	(s1)>(s2)	
E<=, LE	(s1)≤(s2)	
E<, LT	(s1)<(s2)	
E>=, GE	(s1)≥(s2)	
E=, EQ	(s1)≠(s2)	非導通状態(ENOはOFF)
E<>, NE	(s1)=(s2)	
E>, GT	(s1)≤(s2)	
E<=, LE	(s1)>(s2)	
E<, LT	(s1)≥(s2)	
E>=, GE	(s1)<(s2)	

- (s1), (s2)の内容が設定データの範囲外の場合は、非導通(ENOはOFF)になります。
- LDE_□命令をFBD/LDで使用した場合、ENに左母線、または常時ONの変数部品/定数部品を指定してください。
- ORE_□命令をFBD/LDで使用した場合、ENにTRUEを指定するとENOはONします。ENは実行条件にはなりません。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

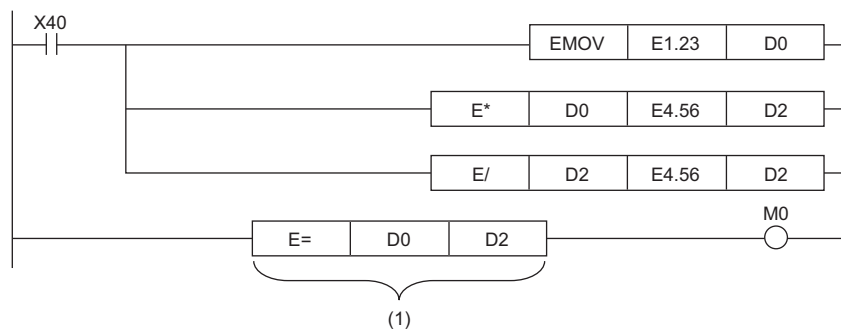
📖 39ページ 注意事項

エラー

演算エラーはありません。

Point

E=命令を使用した場合、誤差により等しくならない場合がありますので注意してください。



(1): 等しくならない場合があります。

倍精度実数比較

LDED□, ANDED□, ORED□

倍精度実数をa接点扱いで比較演算します。

ラダー	ST
	ENO:=LDED_□(EN,s1,s2); ENO:=ANDED_□(EN,s1,s2); ENO:=ORED_□(EN,s1,s2); (□には, EQ, NE, GT, LE, LT, GEが入ります。)*1
(□には, ED=, ED<>, ED>, ED<=, ED<, ED>=が入ります。)	

FBD/LD
(□には, LDED_, ANDED_, ORED_と, EQ, NE, GT, LE, LT, GEの組合せが入ります。)*1

*1 EQは=, NEは<>, GTは>, LEは<=, LTは<, GEは>=です。

■実行条件

命令	実行条件
LDED□, ANDED□, ORED□	常時実行

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	比較データまたは, 比較データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s1) < 2^{1024}$	倍精度実数	ANYREAL_64
(s2)	比較データまたは, 比較データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s2) < 2^{1024}$	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	—	—	○	—	—	○	—	○	—	—	—
(s2)	—	—	○	—	—	○	—	○	—	—	—

機能

- (s1)で指定されたデバイスの倍精度実数と、(s2)で指定されたデバイスの倍精度実数をa接点扱いで比較演算します。
- 各命令の比較演算結果は下記ようになります。

命令記号(ラダー, FBD/LD)	条件	比較演算結果
ED=, EQ	(s1)=(s2)	導通状態(ENOはON)
ED<>, NE	(s1)≠(s2)	
ED>, GT	(s1)>(s2)	
ED<=, LE	(s1)≤(s2)	
ED<, LT	(s1)<(s2)	
ED>=, GE	(s1)≥(s2)	
ED=, EQ	(s1)≠(s2)	非導通状態(ENOはOFF)
ED<>, NE	(s1)=(s2)	
ED>, GT	(s1)≤(s2)	
ED<=, LE	(s1)>(s2)	
ED<, LT	(s1)≥(s2)	
ED>=, GE	(s1)<(s2)	

- (s1), (s2)の内容が設定データの範囲外の場合は、非導通(ENOはOFF)になります。
- LDED_□命令をFBD/LDで使用した場合、ENに左母線、または常時ONの変数部品/定数部品を指定してください。
- ORED_□命令をFBD/LDで使用した場合、ENにTRUEを指定するとENOはONします。ENは実行条件にはなりません。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

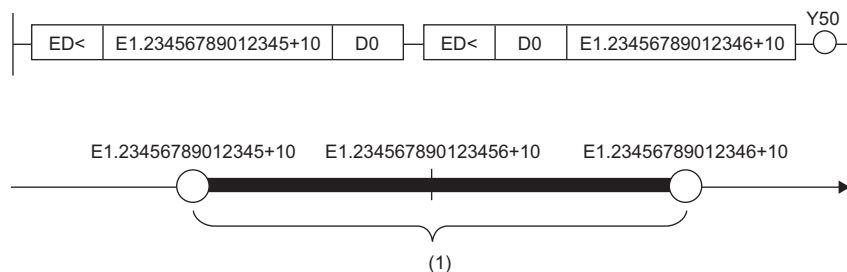
演算エラーはありません。

注意事項

エンジニアリングツールで入力できる実数の桁数は最大15桁であるため、本項で示す命令では有効桁数16桁以上の実数との比較を行うことはできません。本項の命令で有効桁数16桁以上の実数との一致、不一致を判断する場合は、比較する実数前後の近似値との大小比較によって判断をする必要があります。

例

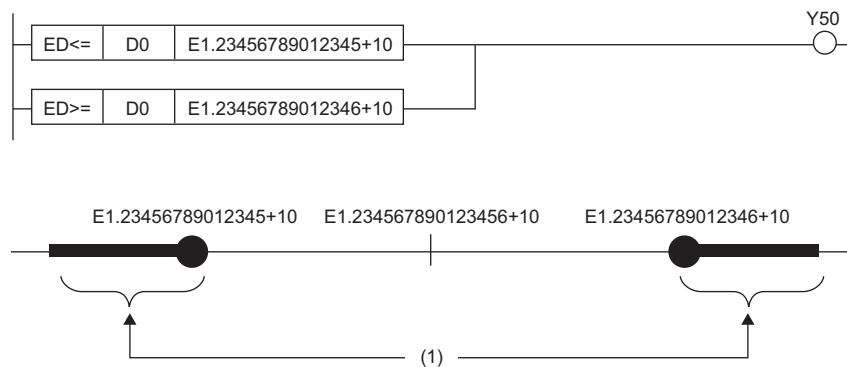
E1.234567890123456+10(有効桁数16桁)と倍精度実数の一致を判断する場合



(1): D0~D3がこの範囲内であるかチェックします。(境界値は範囲に含みません。)

例

E1.234567890123456+10(有効桁数16桁)と倍精度実数の不一致を判断する場合

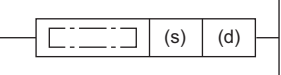


(1): D0~D3がこの範囲内であるかチェックします。(境界値は範囲に含みます。)

単精度実数加算

E+(P) [オペランドが2個の場合]

単精度実数を加算します。

ラダー	ST
	対応していません。 (参照 814ページ E+(P) [オペランドが3個の場合])

FBD/LD
対応していません。 (参照 814ページ E+(P) [オペランドが3個の場合])

■実行条件

命令	実行条件
E+	
E+P	

設定データ

■内容，範囲，データ型

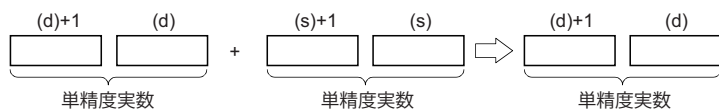
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	加算データまたは，加算データが格納されている先頭デバイス	$0, 2^{-126} \leq (s) < 2^{128}$	単精度実数	ANYREAL_32
(d)	加算されるデータが格納されている先頭デバイス	$0, 2^{-126} \leq (d) < 2^{128}$	単精度実数	ANYREAL_32

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

機能

- (d)で指定された単精度実数と、(s)で指定された単精度実数の加算を行い、加算結果を(d)で指定されたデバイスに格納します。



- (s), (d)には、 $0, 2^{-126} \leq |\text{指定値(格納値)}| < 2^{128}$ の値が指定および格納できます。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

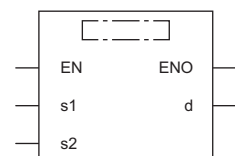
エラーコード (SD0)	内容
3282H	(s)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (d)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ d < 2^{128}$

E+(P) [オペランドが3個の場合]

単精度実数を加算します。

ラダー	ST
	ENO:=EPLUS(EN,s1,s2,d); ENO:=EPLUSP(EN,s1,s2,d);

FBD/LD



(□には、EPLUS、EPLUSPが入ります。)

■実行条件

命令	実行条件
E+	
E+P	

設定データ

■内容、範囲、データ型

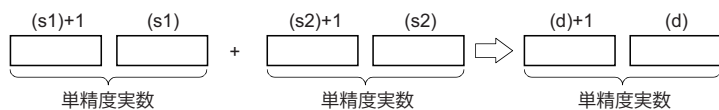
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	加算されるデータまたは、加算されるデータが格納されている先頭デバイス	$0, 2^{-126} \leq (s1) < 2^{128}$	単精度実数	ANYREAL_32
(s2)	加算データまたは、加算データが格納されている先頭デバイス	$0, 2^{-126} \leq (s2) < 2^{128}$	単精度実数	ANYREAL_32
(d)	演算結果を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	○	○	○	○	○	—	○	—	—
(s2)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

機能

- (s1)で指定された単精度実数と，(s2)で指定された単精度実数の加算を行い，加算結果を(d)で指定されたデバイスに格納します。



- (s1), (s2), (d)には， $0, 2^{-126} \leq |\text{指定値(格納値)}| < 2^{128}$ の値が指定および格納できます。

エラー

エラーコード (SD0)	内容
3282H	(s1)で指定したデータが-0，非正規化数，非数， $\pm\infty$ のとき。 (s2)で指定したデータが-0，非正規化数，非数， $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ (d) < 2^{128}$

単精度実数減算

E-(P) [オペランドが2個の場合]

単精度実数を減算します。

ラダー	ST
	対応していません。 (☞ 818ページ E-(P) [オペランドが3個の場合])

FBD/LD
対応していません。 (☞ 818ページ E-(P) [オペランドが3個の場合])

■実行条件

命令	実行条件
E-	
E-P	

設定データ

■内容, 範囲, データ型

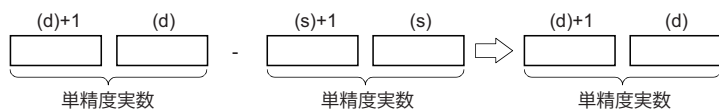
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	減算データまたは, 減算データが格納されている先頭デバイス	$0, 2^{-126} \leq (s) < 2^{128}$	単精度実数	ANYREAL_32
(d)	減算されるデータが格納されている先頭デバイス	$0, 2^{-126} \leq (d) < 2^{128}$	単精度実数	ANYREAL_32

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

機能

- (d)で指定された単精度実数と、(s)で指定された単精度実数の減算を行い、減算結果を(d)で指定されたデバイスに格納します。



- (s), (d)には、 $0, 2^{-126} \leq |\text{指定値(格納値)}| < 2^{128}$ の値が指定および格納できます。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

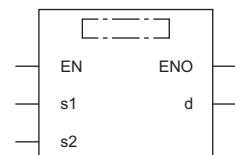
エラーコード (SD0)	内容
3282H	(s)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (d)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ d < 2^{128}$

E-(P) [オペランドが3個の場合]

単精度実数を減算します。

ラダー	ST
	<pre>ENO:=EMINUS(EN,s1,s2,d); ENO:=EMINUSP(EN,s1,s2,d);</pre>

FBD/LD



(□には、EMINUS、EMINUSPが入ります。)

■実行条件

命令	実行条件
E-	
E-P	

設定データ

■内容、範囲、データ型

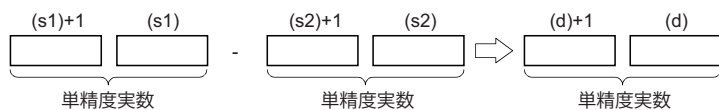
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	減算されるデータまたは、減算されるデータが格納されている先頭デバイス	$0, 2^{-126} \leq (s1) < 2^{128}$	単精度実数	ANYREAL_32
(s2)	減算データまたは、減算データが格納されている先頭デバイス	$0, 2^{-126} \leq (s2) < 2^{128}$	単精度実数	ANYREAL_32
(d)	演算結果を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	—	—	○	○	○	○	○	○	○	—	○	—	—
(s2)	—	—	○	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	○	—	—	—	—

機能

- (s1)で指定された単精度実数と、(s2)で指定された単精度実数の減算を行い、減算結果を(d)で指定されたデバイスに格納します。



- (s1), (s2), (d)には、 $0, 2^{-126} \leq |\text{指定値(格納値)}| < 2^{128}$ の値が指定および格納できます。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

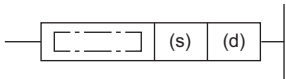
エラー

エラーコード (SD0)	内容
3282H	(s1)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (s2)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えると時。(オーバーフローが発生したとき。) $ d < 2^{128}$

倍精度実数加算

ED+(P) [オペランドが2個の場合]

倍精度実数を加算します。

ラダー	ST
	対応していません。 (参照 822ページ ED+(P) [オペランドが3個の場合])

FBD/LD
対応していません。 (参照 822ページ ED+(P) [オペランドが3個の場合])

■実行条件

命令	実行条件
ED+	
ED+P	

設定データ

■内容, 範囲, データ型

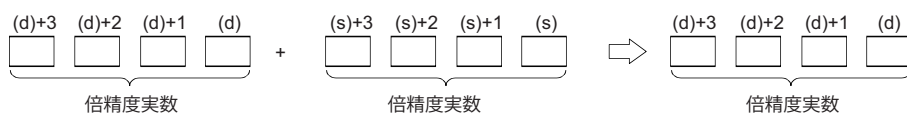
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	加算データまたは, 加算データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	加算されるデータが格納されている先頭デバイス	$0, 2^{-1022} \leq (d) < 2^{1024}$	倍精度実数	ANYREAL_64

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (d)で指定された倍精度実数と、(S)で指定された倍精度実数の加算を行い、加算結果を(d)で指定されたデバイスに格納します。



- (s), (d)には、 $0, 2^{-1022} \leq |\text{指定値(格納値)}| < 2^{1024}$ の値が指定および格納できます。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

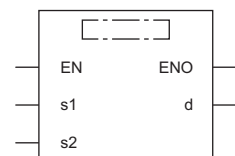
エラーコード (SD0)	内容
3282H	(s)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (d)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ (\text{d}) < 2^{1024}$

ED+(P) [オペランドが3個の場合]

倍精度実数を加算します。

ラダー	ST
	ENO:=EDPLUS(EN,s1,s2,d); ENO:=EDPLUSP(EN,s1,s2,d);

FBD/LD



(□には、EDPLUS、EDPLUSPが入ります。)

■実行条件

命令	実行条件
ED+	
ED+P	

設定データ

■内容、範囲、データ型

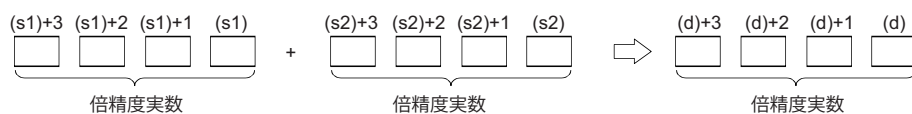
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	加算されるデータまたは、加算されるデータが格納されている先頭デバイス	$0, 2^{-1022} \leq (s1) < 2^{1024}$	倍精度実数	ANYREAL_64
(s2)	加算データまたは、加算データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s2) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	演算結果を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	○	—	○	—	○	—	—
(s2)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (s1)で指定された倍精度実数と、(s2)で指定された倍精度実数の加算を行い、加算結果を(d)で指定されたデバイスに格納します。



- (s1), (s2), (d)には、 $0, 2^{-1022} \leq |\text{指定値(格納値)}| < 2^{1024}$ の値が指定および格納できます。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

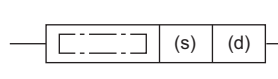
エラー

エラーコード (SD0)	内容
3282H	(s1)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (s2)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ d < 2^{1024}$

倍精度実数減算

ED-(P) [オペランドが2個の場合]

倍精度実数を減算します。

ラダー	ST
	対応していません。 (826ページ ED-(P) [オペランドが3個の場合])
FBD/LD	
対応していません。 (826ページ ED-(P) [オペランドが3個の場合])	

■実行条件

命令	実行条件
ED-	
ED-P	

設定データ

■内容，範囲，データ型

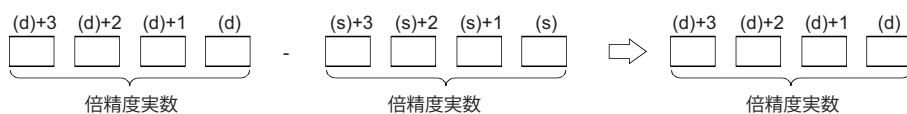
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	減算データまたは、減算データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	減算されるデータが格納されている先頭デバイス	$0, 2^{-1022} \leq (d) < 2^{1024}$	倍精度実数	ANYREAL_64

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (d)で指定された倍精度実数と、(s)で指定された倍精度実数の減算を行い、減算結果を(d)で指定されたデバイスに格納します。



- (s), (d)には、 $0, 2^{-1022} \leq |\text{指定値(格納値)}| < 2^{1024}$ の値が指定および格納できます。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

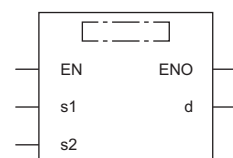
エラーコード (SD0)	内容
3282H	(s)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (d)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ (\text{d}) < 2^{1024}$

ED-(P) [オペランドが3個の場合]

倍精度実数を減算します。

ラダー	ST
	ENO:=EDMINUS(EN,s1,s2,d); ENO:=EDMINUSP(EN,s1,s2,d);

FBD/LD



(□には、EDMINUS、EDMINUSPが入ります。)

■実行条件

命令	実行条件
ED-	
ED-P	

設定データ

■内容、範囲、データ型

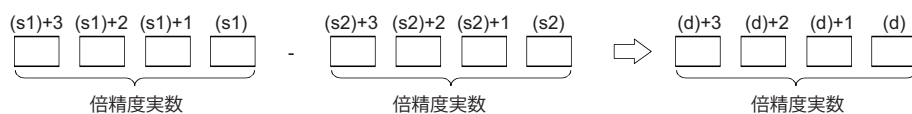
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	減算されるデータまたは、減算されるデータが格納されている先頭デバイス	$0, 2^{-1022} \leq (s1) < 2^{1024}$	倍精度実数	ANYREAL_64
(s2)	減算データまたは、減算データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s2) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	演算結果を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	○	—	○	—	○	—	—
(s2)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (s1)で指定された倍精度実数と、(s2)で指定された倍精度実数の減算を行い、減算結果を(d)で指定されたデバイスに格納します。



- (s1), (s2), (d)には、 $0, 2^{-1022} \leq |\text{指定値(格納値)}| < 2^{1024}$ の値が指定および格納できます。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

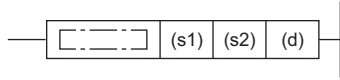
エラー

エラーコード (SD0)	内容
3282H	(s1)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (s2)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ d < 2^{1024}$

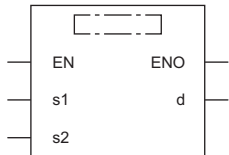
単精度実数乗算

E*(P)

単精度実数を乗算します。

ラダー	ST
	ENO:=EMULTI(EN,s1,s2,d); ENO:=EMULTIP(EN,s1,s2,d);

FBD/LD



(□には、EMULTI、EMULTIPが入ります。)

■実行条件

命令	実行条件
E*	
E*P	

設定データ

■内容、範囲、データ型

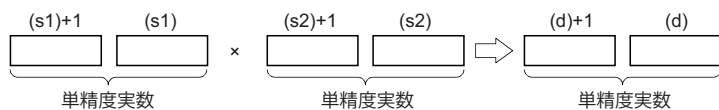
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	乗算されるデータまたは、乗算されるデータが格納されている先頭デバイス	$0, 2^{-126} \leq (s1) < 2^{128}$	単精度実数	ANYREAL_32
(s2)	乗算データまたは、乗算データが格納されている先頭デバイス	$0, 2^{-126} \leq (s2) < 2^{128}$	単精度実数	ANYREAL_32
(d)	演算結果を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	—	—	○	○	○	○	○	○	○	—	○	—	—
(s2)	—	—	○	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	○	—	—	—	—

機能

- (s1)で指定された単精度実数と、(s2)で指定された単精度実数の乗算を行い、乗算結果を(d)で指定されたデバイスに格納します。



- (s1), (s2), (d)には、 $0, 2^{-126} \leq |\text{指定値(格納値)}| < 2^{128}$ の値が指定および格納できます。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

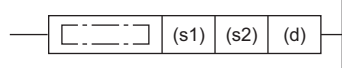
エラー

エラーコード (SD0)	内容
3282H	(s1)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (s2)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ d < 2^{128}$

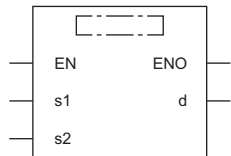
単精度実数除算

E/(P)

単精度実数を除算します。

ラダー	ST
	ENO:=EDIVISION(EN,s1,s2,d); ENO:=EDIVISIONP(EN,s1,s2,d);

FBD/LD



(□には、EDIVISION、EDIVISIONPが入ります。)

■実行条件

命令	実行条件
E/	
E/P	

設定データ

■内容、範囲、データ型

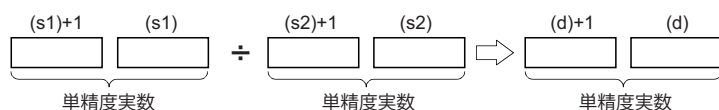
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	除算されるデータまたは、除算されるデータが格納されている先頭デバイス	$0, 2^{-126} \leq (s1) < 2^{128}$	単精度実数	ANYREAL_32
(s2)	除算データまたは、除算データが格納されている先頭デバイス	$0, 2^{-126} \leq (s2) < 2^{128}$	単精度実数	ANYREAL_32
(d)	演算結果を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□≠□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□≠G□, J□≠□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	—	—	○	○	○	○	○	○	○	—	○	—	—
(s2)	—	—	○	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	○	—	—	—	—

機能

- (s1)で指定された単精度実数と、(s2)で指定された単精度実数の除算を行い、除算結果を(d)で指定されたデバイスに格納します。



- (s1), (s2), (d)には、 $0, 2^{-126} \leq |\text{指定値(格納値)}| < 2^{128}$ の値が指定および格納できます。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3280H	(s2)で指定したデータ(除数)が0のとき。
3282H	(s1)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (s2)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ d < 2^{128}$

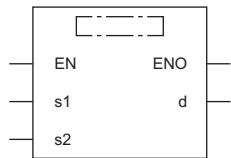
倍精度実数乗算

ED*(P)

倍精度実数を乗算します。

ラダー	ST
	ENO:=EDMULTI(EN,s1,s2,d); ENO:=EDMULTIP(EN,s1,s2,d);

FBD/LD



(□には、EDMULTI、EDMULTIPが入ります。)

■実行条件

命令	実行条件
ED*	
ED*P	

設定データ

■内容、範囲、データ型

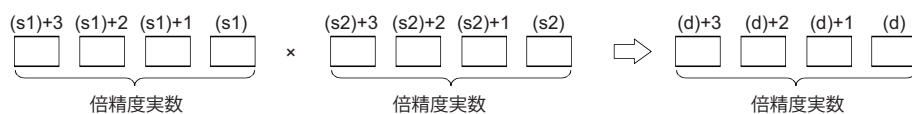
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	乗算されるデータまたは、乗算されるデータが格納されている先頭デバイス	$0, 2^{-1022} \leq (s1) < 2^{1024}$	倍精度実数	ANYREAL_64
(s2)	乗算データまたは、乗算データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s2) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	演算結果を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	○	—	○	—	○	—	—
(s2)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (s1)で指定された倍精度実数と、(s2)で指定された倍精度実数の乗算を行い、乗算結果を(d)で指定されたデバイスに格納します。



- (s1), (s2), (d)には、 $0, 2^{-1022} \leq |\text{指定値(格納値)}| < 2^{1024}$ の値が指定および格納できます。
- 演算結果が-0またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s1)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (s2)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ d < 2^{1024}$

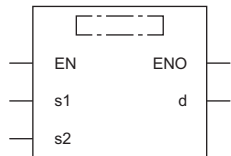
倍精度実数除算

ED/(P)

倍精度実数を除算します。

ラダー	ST
	ENO:=EDDIVISION(EN,s1,s2,d); ENO:=EDDIVISIONP(EN,s1,s2,d);

FBD/LD



(□には、EDDIVISION、EDDIVISIONPが入ります。)

■実行条件

命令	実行条件
ED/	
ED/P	

設定データ

■内容、範囲、データ型

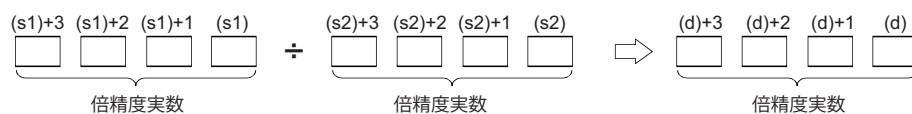
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	除算されるデータまたは、除算されるデータが格納されている先頭デバイス	$0, 2^{-1022} \leq (s1) < 2^{1024}$	倍精度実数	ANYREAL_64
(s2)	除算データまたは、除算データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s2) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	演算結果を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	○	—	○	—	○	—	—
(s2)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (s1)で指定された倍精度実数と、(s2)で指定された倍精度実数の除算を行い、除算結果を(d)で指定されたデバイスに格納します。



- (s1), (s2), (d)には、 $0, 2^{-1022} \leq |\text{指定値(格納値)}| < 2^{1024}$ の値が指定および格納できます。
- 演算結果が-0またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3280H	(s2)で指定したデータ(除数)が0のとき。
3282H	(s1)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (s2)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ (d) < 2^{1024}$

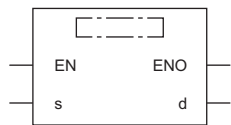
符号付きBIN16ビットデータ→単精度実数変換

INT2FLT(P)

符号付きBIN16ビットデータを単精度実数に変換します。

ラダー	ST
	ENO:=INT2FLT(EN,s,d); ENO:=INT2FLTP(EN,s,d);

FBD/LD



■実行条件

命令	実行条件
INT2FLT	
INT2FLTP	

設定データ

■内容，範囲，データ型

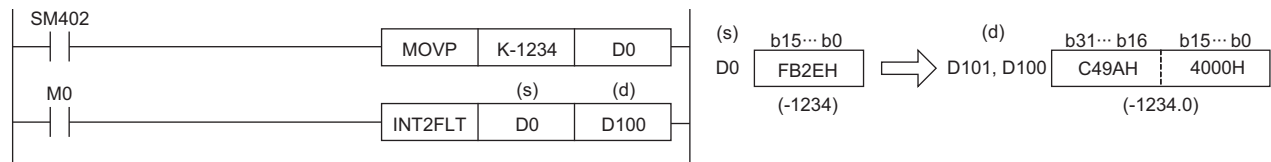
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	単精度実数に変換する整数データまたは，整数データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16_S
(d)	変換した単精度実数を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ				K, H	E	\$	
(s)	○	○	○	○	○	—	—	○		○	—	—	—	—
(d)	—	—	○	○	○	○	○	○	○	—	—	—	—	—

機能

- ・ (s)で指定した符号付きBIN16ビットデータを単精度実数に変換し，(d)へ格納します。



エラー

演算エラーはありません。

符号なしBIN16ビットデータ→単精度実数変換

UINT2FLT(P)

符号なしBIN16ビットデータを単精度実数に変換します。

ラダー	ST
	ENO:=UINT2FLT(EN,s,d); ENO:=UINT2FLTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
UINT2FLT	
UINT2FLTP	

設定データ

■内容、範囲、データ型

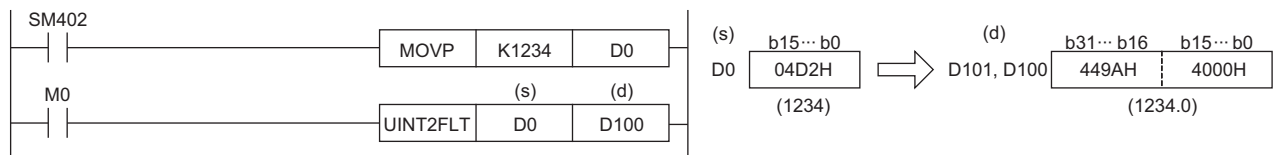
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	単精度実数に変換する整数データまたは、整数データが格納されているデバイス	0~65535	符号なしBIN16ビット	ANY16_U
(d)	変換した単精度実数を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口¥口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口¥G口, J口¥口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定した符号なしBIN16ビットデータを単精度実数に変換し、(d)へ格納します。



エラー

演算エラーはありません。

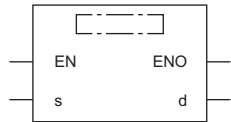
符号付きBIN32ビットデータ→単精度実数変換

DINT2FLT(P)

符号付きBIN32ビットデータを単精度実数に変換します。

ラダー	ST
	ENO:=DINT2FLT(EN,s,d); ENO:=DINT2FLTP(EN,s,d);

FBD/LD



■実行条件

命令	実行条件
DINT2FLT	
DINT2FLTP	

設定データ

■内容，範囲，データ型

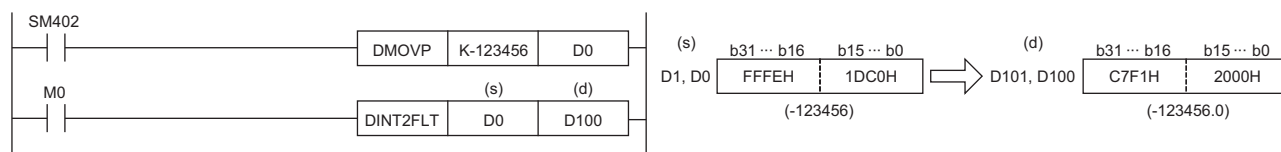
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	単精度実数に変換する整数データまたは，整数データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
(d)	変換した単精度実数を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

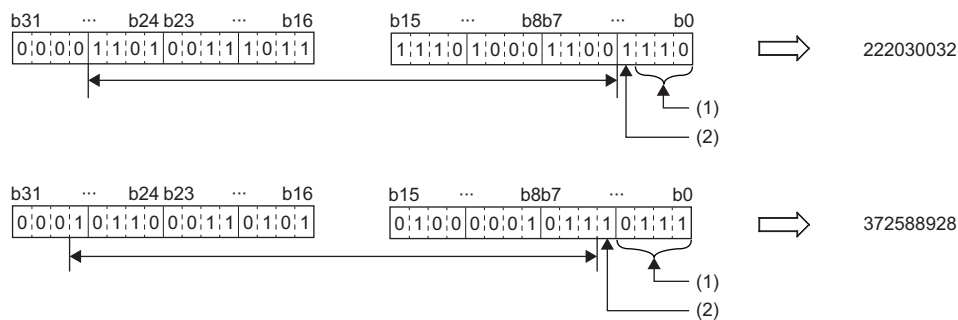
オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	○	—	—	—
(d)	—	—	○	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定した符号付きBIN32ビットデータを単精度実数に変換し、(d)へ格納します。



- 単精度実数は、32ビットの単精度で処理するため、2進数で表現したときは有効桁数が24ビット、10進数で表現したときは約7桁となります。したがって、整数値が-16777216~16777215(24ビットBIN値)の範囲を超えた場合、変換した値に誤差が生じます。変換結果は、整数値の上位ビットから25ビット目を0捨1入し、26ビット目以降を切り捨てた値となります。



- (1): 切り捨てられます。
(2): 0捨1入されます。

エラー

演算エラーはありません。

符号なしBIN32ビットデータ→単精度実数変換

UDINT2FLT(P)

符号なしBIN32ビットデータを単精度実数に変換します。

ラダー	ST
	ENO:=UDINT2FLT(EN,s,d); ENO:=UDINT2FLTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
UDINT2FLT	
UDINT2FLTP	

設定データ

■内容，範囲，データ型

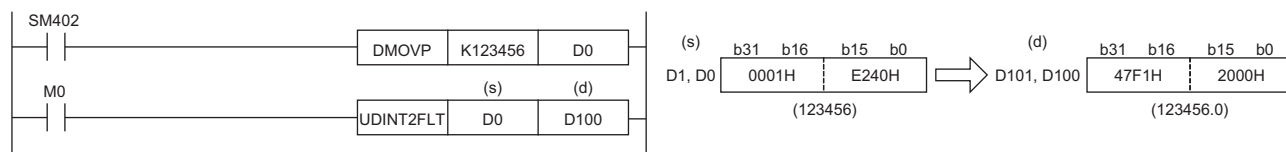
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	単精度実数に変換する整数データまたは、整数データが格納されている先頭デバイス	0~4294967295	符号なしBIN32ビット	ANY32_U
(d)	変換した単精度実数を格納する先頭デバイス	—	単精度実数	ANYREAL_32

■使用可能デバイス

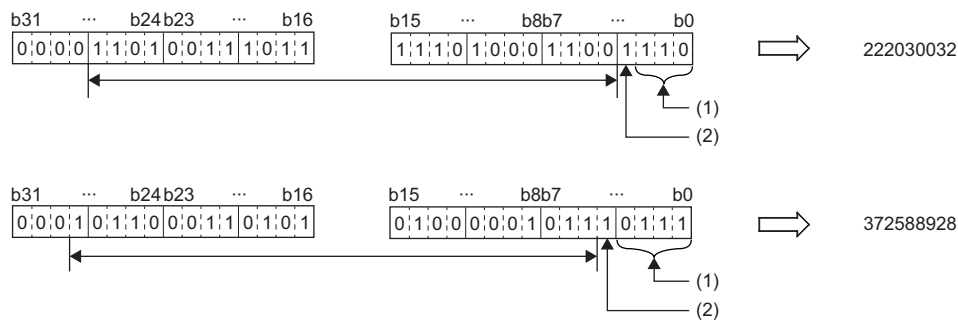
オペランド	ビット		ワード		ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定した符号なしBIN32ビットデータを単精度実数に変換し、(d)へ格納します。



- 単精度実数は、32ビットの単精度で処理するため、2進数で表現したときは有効桁数が24ビット、10進数で表現したときは約7桁となります。したがって、整数値が0~16777215(24ビットBIN値)の範囲を超えた場合、変換した値に誤差が生じます。変換結果は、整数値の上位ビットから25ビット目を0捨1入し、26ビット目以降を切り捨てた値となります。



(1): 切り捨てられます。

(2): 0捨1入されます。

エラー

演算エラーはありません。

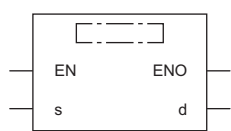
倍精度実数→単精度実数変換

DBL2FLT(P)

倍精度実数を単精度実数に変換します。

ラダー	ST
	ENO:=DBL2FLT(EN,s,d); ENO:=DBL2FLTP(EN,s,d);

FBD/LD



■実行条件

命令	実行条件
DBL2FLT	
DBL2FLTP	

設定データ

■内容，範囲，データ型

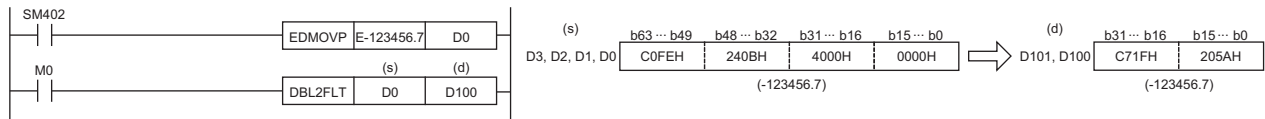
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	単精度実数に変換する倍精度実数データまたは、倍精度実数データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	変換した単精度実数を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	○	○	○	○	—	—	—	—

機能

- ・ (s) で指定した倍精度実数を単精度実数に変換し、(d) へ格納します。



- ・ エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

39ページ 注意事項

エラーコード (SD0)	内容
3282H	(s)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ (\text{d}) < 2^{128}$

符号付きBIN16ビットデータ→倍精度実数変換

INT2DBL(P)

符号付きBIN16ビットデータを倍精度実数に変換します。

ラダー	ST
	ENO:=INT2DBL(EN,s,d); ENO:=INT2DBLP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
INT2DBL	
INT2DBLP	

設定データ

■内容，範囲，データ型

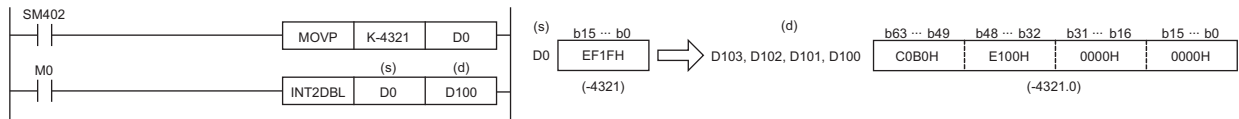
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	倍精度実数に変換する整数データまたは、整数データが格納されているデバイス	-32768~32767	符号付きBIN16ビット	ANY16_S
(d)	変換した倍精度実数を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	—	○	—	○	—	—	○	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- ・ (s)で指定した符号付きBIN16ビットデータを倍精度実数に変換し、(d)へ格納します。



エラー

演算エラーはありません。

符号なしBIN16ビットデータ→倍精度実数変換

UINT2DBL(P)

符号なしBIN16ビットデータを倍精度実数に変換する。

ラダー	ST
	ENO:=UINT2DBL(EN,s,d); ENO:=UINT2DBLP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
UINT2DBL	
UINT2DBLP	

設定データ

■内容、範囲、データ型

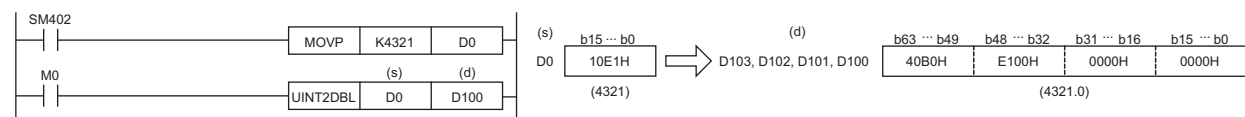
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	倍精度実数に変換する整数データまたは、整数データが格納されているデバイス	0~65535	符号なしBIN16ビット	ANY16_U
(d)	変換した倍精度実数を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	—	○	—	○	—	—	○	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (s)で指定した符号なしBIN16ビットデータを倍精度実数に変換し、(d)へ格納します。



エラー

演算エラーはありません。

符号付きBIN32ビットデータ→倍精度実数変換

DINT2DBL(P)

符号付きBIN32ビットデータを倍精度実数に変換します。

ラダー	ST
	ENO:=DINT2DBL(EN,s,d); ENO:=DINT2DBLP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
DINT2DBL	
DINT2DBLP	

設定データ

■内容，範囲，データ型

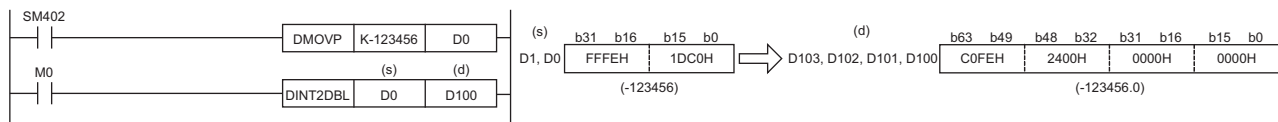
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	倍精度実数に変換する整数データまたは，整数データが格納されている先頭デバイス	-2147483648~2147483647	符号付きBIN32ビット	ANY32_S
(d)	変換した倍精度実数を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	○	—	○	—	○	○	○	○	○	○	—	—	—
(d)	—	—	○	—	—	○	—	○	○	—	—	—	—

機能

- ・ (s)で指定した符号付きBIN32ビットデータを倍精度実数に変換し，(d)へ格納します。



エラー

演算エラーはありません。

符号なしBIN32ビットデータ→倍精度実数変換

UDINT2DBL(P)

符号なしBIN32ビットデータを倍精度実数に変換します。

ラダー	ST
	ENO:=UDINT2DBL(EN,s,d); ENO:=UDINT2DBLP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
UDINT2DBL	
UDINT2DBLP	

設定データ

■内容，範囲，データ型

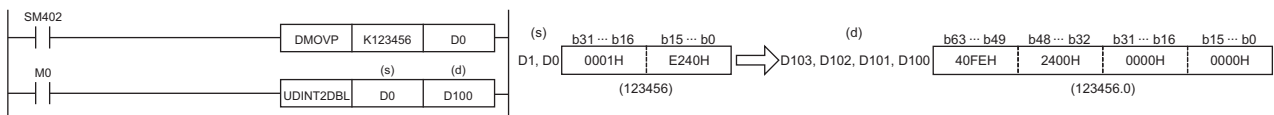
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	倍精度実数に変換する整数データまたは，整数データが格納されている先頭デバイス	0~4294967295	符号なしBIN32ビット	ANY32_U
(d)	変換した倍精度実数を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口¥口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口¥G口, J口¥口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	—	○	—	○	○	○	○	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (s)で指定した符号なしBIN32ビットデータを倍精度実数に変換し，(d)へ格納します。



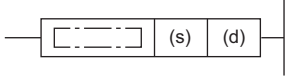
エラー

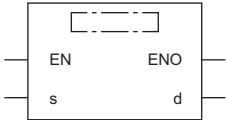
演算エラーはありません。

文字列→単精度実数変換

EVAL(P)

文字列を単精度実数に変換します。

ラダー	ST
	<pre>ENO:=EVAL(EN,s,d); ENO:=EVALP(EN,s,d);</pre>

FBD/LD


■実行条件

命令	実行条件
EVAL	
EVALP	

設定データ

■内容、範囲、データ型

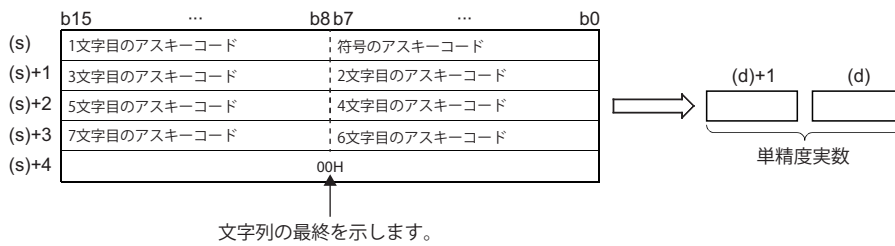
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	単精度実数データに変換する文字列データまたは、文字列データが格納されている先頭デバイス	—	文字列	ANYSTRING_SINGLE
(d)	変換した単精度実数データを格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	—	—	○	—	—	—	○	—	—	○	—
(d)	—	—	○	○	○	○	○	—	—	—	—

機能

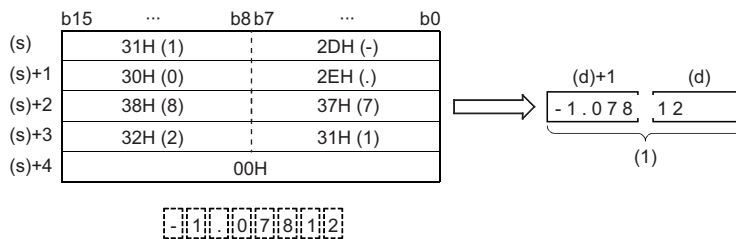
- (s)で指定されたデバイス番号以降に格納されている文字列を単精度実数データに変換し、(d)で指定されたデバイスに格納します。
- 指定する文字列は小数点形式、指数形式どちらでも単精度実数に変換することができます。



- 文字列中に20H(スペース)が含まれていた場合は、20Hを無視して変換します。
- 文字列は最大24文字まで設定できます。文字列中の20H(スペース)、30H(0)も1文字としてカウントします。

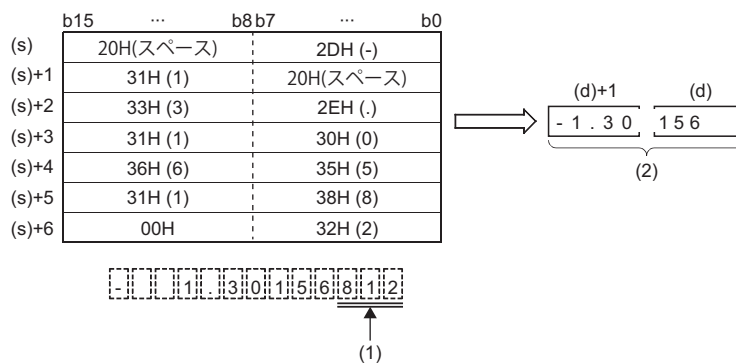
■小数点形式の場合

- (s)で指定した文字列が小数点形式の場合は、下記のように実行されます。



(1): 単精度実数

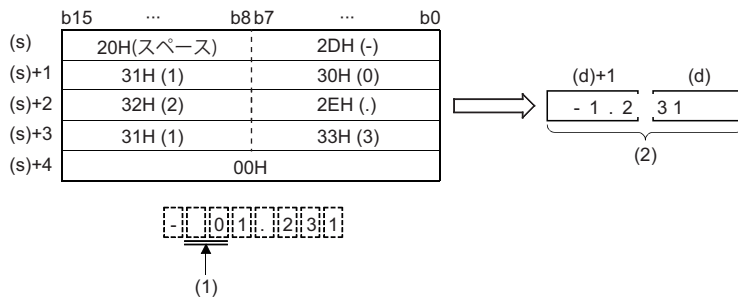
- (s)で指定された文字列において、単精度実数に変換する文字列は、符号、小数点、指数部を除いて6桁が有効となり、7桁目以降はカットされて変換されます。



(1): 切り捨てられます。

(2): 単精度実数

- 小数点形式で符号に2BH(+)を指定するか、符号を省略すると、正の値として変換します。また、符号に2DH(-)を指定すると、負の値として変換します。
- (s)で指定された文字列で、最初の0以外の数値の間に、20H(スペース)か30H(0)が存在する場合は、20H、30Hを無視して変換します。

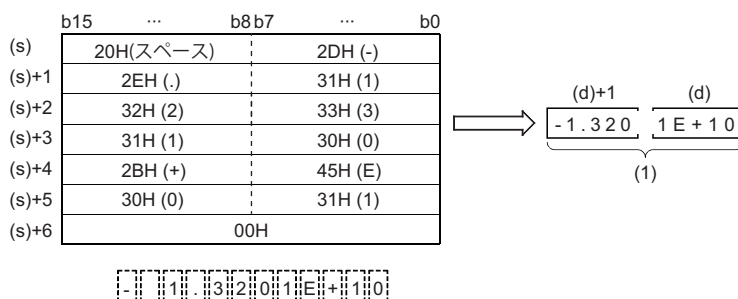


(1): 無視します。

(2): 単精度実数

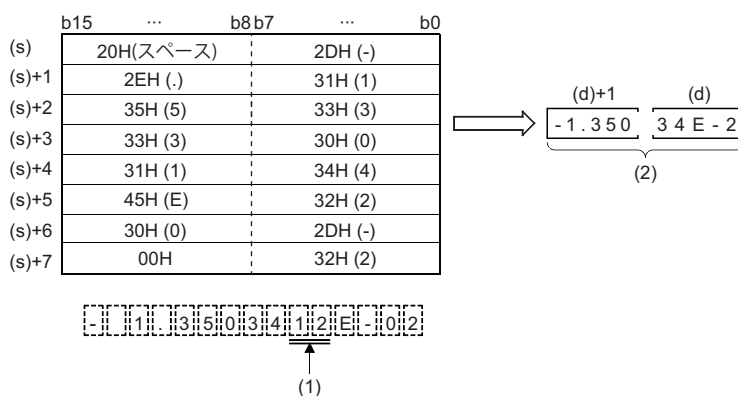
■指数形式の場合

- (s)で指定した文字列が指数形式の場合は、下記のように実行されます。



(1): 単精度実数

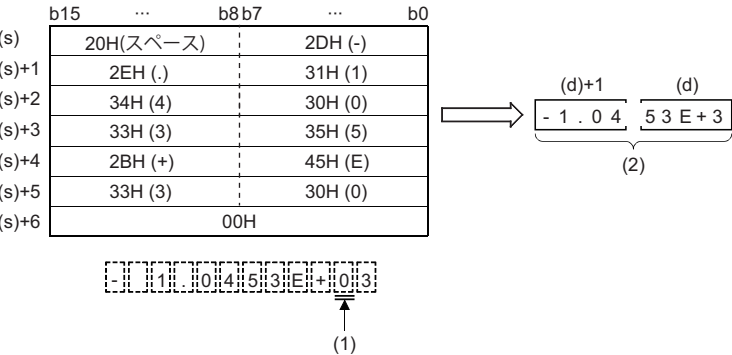
- (s)で指定された文字列において、単精度実数に変換する文字列は、符号、小数点、指数部を除いて6桁が有効となり、7桁目以降はカットされて変換されます。



(1): 切り捨てられます。

(2): 単精度実数

- 指数形式で指数部の符号に2BH(+)を指定するか、符号を省略すると正の値として変換します。指数部の符号に2DH(-)を指定すると、負の値として変換します。
- (s)で指定された文字列で、最初の0以外の数値の間に、20H(スペース)か30H(0)が存在する場合は、20H、30Hを無視して変換します。
- 指数形式の文字列で、「E」と数値の間に30H(0)が存在する場合は、30Hを無視して変換します。



- (1): 無視します。
 (2): 単精度実数
- 文字列中に“20H”(スペース)が含まれていた場合は、“20H”を無視して変換します。
 - 文字列は最大24文字まで設定できます。文字列中の“20H”(スペース)、“30H”(0)も1文字としてカウントします。

エラー

エラーコード (SD0)	内容
2820H	(s)から該当デバイス範囲内に00Hが無いとき。
3281H	(s)に変換できない不正なデータを設定したとき。 - 整数部、小数部に、30H(0)~39H(9)以外の文字がある - 指定した文字列の中に2EH(.)が2つ以上ある - 指定した文字列の指数部に、45H(E)、65H(e)、2BH(+), 2DH(-)以外の文字がある - 指定した文字列の中に45H(E)、65H(e)の指数部が複数ある - 指定した文字列の中に、指数部の数値を3桁以上記載している - 指定した文字列の中に2BH(+), 2DH(-)の指数部の符号が複数ある - 指定した文字列の中に2BH(+), 2DH(-)の符号が小数点形式の場合は正数部に複数ある、指数形式の場合は仮数部に複数ある
3283H	(d)が下記範囲を超えると時。(オーバーフローが発生したとき。) $ d < 2^{128}$
3285H	(s)以降の文字数が0、または24文字を超えているとき。

BCDフォーマットデータ→単精度実数データ変換

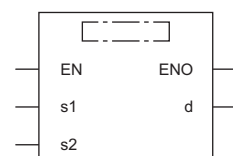
17

EREXP(P)

BCD型浮動小数点フォーマットデータを指定された小数部桁数に基づいて単精度実数データに変換します。

ラダー	ST
	ENO:=EREXP(EN,s1,s2,d); ENO:=EREXPP(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
EREXP	
EREXPP	

設定データ

■内容、範囲、データ型

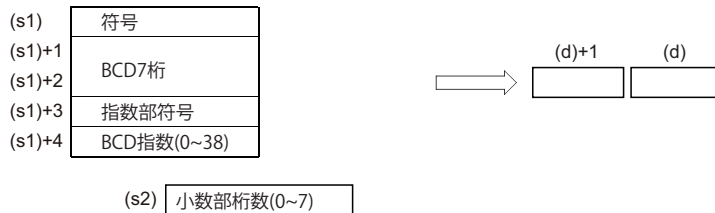
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	BCD型浮動小数点フォーマットのデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 5)
(s2)	小数部桁数データ	0~7	符号付きBIN16ビット	ANY16
(d)	単精度実数を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□	G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	○	○	○	○	○	○	—	○	○	—	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

機能

- (s1)で指定されたデバイス番号以降に格納されているBCD型浮動小数点フォーマットデータを、(s2)で指定されたデバイスに格納されている小数部桁数に基づいて単精度実数データに変換し、(d)で指定されたデバイス番号以降に格納します。



(s1): 符号(正のとき0, 負のとき1)

(s1)+1, (s1)+2: BCD7桁

(s1)+3: 指数部符号(正のとき0, 負のとき1)

(s1)+4: BCD指数(0-38)

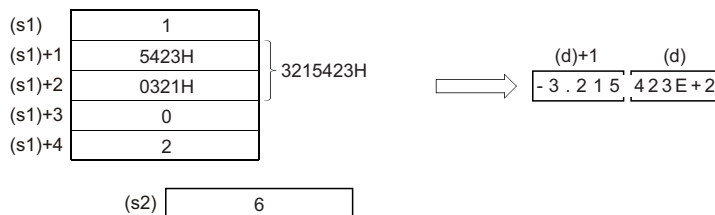
(s2): 小数部桁数(0-7)

(d)+1, (d): 単精度実数

- (s1)の符号および(s1)+3の指数部符号には、正のとき0, 負のとき1を設定します。
- (s1)+4のBCD指数には、0~38が設定できます。
- (s2)の小数部桁数には、0~7が設定できます。

例

(s2)に6を設定した場合



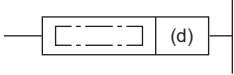
エラー

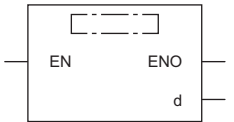
エラーコード (SD0)	内容
3281H	(s1)で指定する形式指定が0, 1以外のとき。 (s1)+1, (s1)+2の各桁に0~9以外の値があるとき。 (s1)+3で指定する指数部符号が0, 1以外のとき。 (s1)+4で指定する指数データが0~38以外のとき。 (s2)で指定した小数部桁数が0~7以外のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ d < 2^{128}$

単精度実数符号反転

ENEG(P)

単精度実数データの符号を反転します。

ラダー	ST
	ENO:=ENEG(EN,d); ENO:=ENEGP(EN,d);

FBD/LD


■実行条件

命令	実行条件
ENEG	
ENEGP	

設定データ

■内容, 範囲, データ型

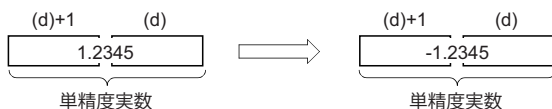
オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	符号反転する単精度実数データが格納されている先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	—	—	○	○	○	○	○	○	—	—	—	—

機能

- (d)で指定されたデバイスの単精度実数の符号を反転して、(d)で指定されたデバイスに格納します。



- 正負の符号を反転するときに使用します。

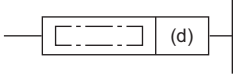
エラー

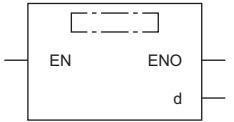
エラーコード (SD0)	内容
3282H	(d)で指定した内容が-0, 非正規化数, 非数, ±∞のとき。

倍精度実数符号反転

EDNEG(P)

倍精度実数データの符号を反転します。

ラダー	ST
	ENO:=EDNEG(EN,d); ENO:=EDNEGP(EN,d);

FBD/LD


■実行条件

命令	実行条件
EDNEG	
EDNEGP	

設定データ

■内容、範囲、データ型

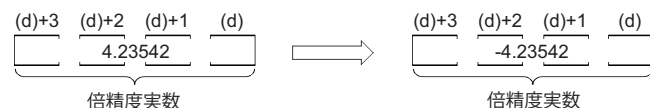
オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	符号反転する倍精度実数が格納されている先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □		LT, LST, LC	LZ		K, H	E	\$	
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (d)で指定されたデバイスの倍精度実数データの符号を反転して、(d)で指定されたデバイスに格納します。



- 正負の符号を反転するときに使用します。

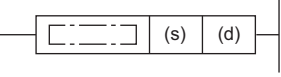
エラー

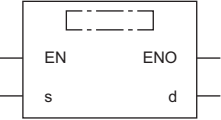
エラーコード (SD0)	内容
3282H	(d)で指定した内容が-0, 非正規化数, 非数, ±∞のとき。

単精度実数データ転送

EMOV(P)

単精度実数データを指定されたデバイスへ転送します。

ラダー	ST
	ENO:=EMOV(EN,s,d); ENO:=EMOVP(EN,s,d)

FBD/LD


■実行条件

命令	実行条件
EMOV	
EMOVP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	転送するデータまたは、転送するデータが格納されている先頭デバイス	$0, 2^{-126} \leq (s) < 2^{128}$	単精度実数	ANYREAL_32
(d)	転送先のデータを格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	—	—	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	—	—	—	—

機能

- (s)で指定されたデバイスに格納されている単精度実数データを、(d)で指定されたデバイスへ転送します。



エラー

演算エラーはありません。

倍精度実数データ転送

EDMOV(P)

倍精度実数データを指定されたデバイスへ転送します。

ラダー	ST
	ENO:=EDMOV(EN,s,d); ENO:=EDMOV(P,EN,s,d);

FBD/LD

■実行条件

命令	実行条件
EDMOV	
EDMOV(P)	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	転送するデータまたは、転送するデータが格納されている先頭デバイス	$0, 2^{-1022} \leq (s) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	転送先のデータを格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- ・ (s)で指定されたデバイスに格納されている倍精度実数データを、(d)で指定されたデバイスへ転送します。



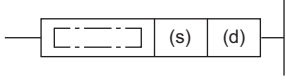
エラー

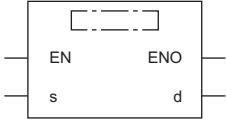
演算エラーはありません。

単精度実数SIN演算

SIN(P)

単精度実数で指定した角度のSIN(正弦)値を演算します。

ラダー	ST*1
	ENO:=SINP(EN,s,d);

FBD/LD*1


*1 SIN命令はST, FBD/LDでは対応していません。汎用ファンクションのSINを使用してください。

1260ページ SIN(_E)

■実行条件

命令	実行条件
SIN	
SINP	

設定データ

■内容、範囲、データ型

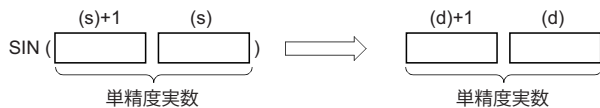
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	SIN(正弦)演算を行う角度データまたは、角度データが格納されている先頭デバイス	$0, 2^{-126} \leq (s) < 2^{128}$	単精度実数	ANYREAL_32
(d)	演算結果を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定された角度のSIN(正弦)値を演算し、(d)で指定されたデバイス番号に演算結果を格納します。



- (s)で指定する角度は、ラジアン単位(角度 $\times\pi\div180$)で設定します。
- 演算結果が-0、またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0、非正規化数、非数、 $\pm\infty$ のとき。

Point

角度 $\leftrightarrow$ ラジアン変換については、RAD(P)命令、DEG(P)命令を参照してください。

895ページ RAD(P)

897ページ DEG(P)

単精度実数COS演算

COS(P)

単精度実数で指定した角度のCOS(余弦)値を演算します。

ラダー	ST*1
	ENO:=COSP(EN,s,d);

FBD/LD*1

*1 COS命令はST, FBD/LDでは対応していません。汎用ファンクションのCOSを使用してください。

📖 1261ページ COS(_E)

■実行条件

命令	実行条件
COS	
COSP	

設定データ

■内容, 範囲, データ型

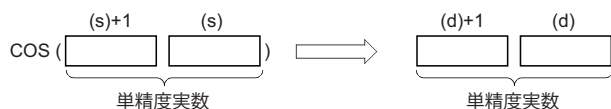
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	COS(余弦)演算を行う角度データまたは、角度データが格納されている先頭デバイス	$0, 2^{-126} \leq (s) < 2^{128}$	単精度実数	ANYREAL_32
(d)	演算結果を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定された角度のCOS(余弦)値を演算し、(d)で指定されたデバイス番号に演算結果を格納します。



- (s)で指定する角度は、ラジアン単位(角度 $\times\pi\div180$)で設定します。
- 演算結果が-0、またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0、非正規化数、非数、 $\pm\infty$ のとき。

Point

角度 $\leftrightarrow$ ラジアン変換については、RAD(P)命令、DEG(P)命令を参照してください。

895ページ RAD(P)

897ページ DEG(P)

単精度実数TAN演算

TAN(P)

単精度実数で指定した角度のTAN(正接)値を演算します。

ラダー	ST*1
	ENO:=TANP(EN,s,d);

FBD/LD*1

*1 TAN命令はST, FBD/LDでは対応していません。汎用ファンクションのTANを使用してください。

📖 1262ページ TAN(_E)

■実行条件

命令	実行条件
TAN	
TANP	

設定データ

■内容, 範囲, データ型

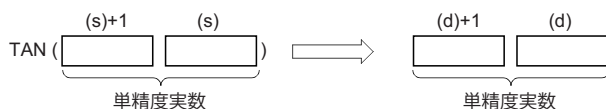
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	TAN(正接)演算を行う角度データまたは、角度データが格納されている先頭デバイス	$0, 2^{-126} \leq (s) < 2^{128}$	単精度実数	ANYREAL_32
(d)	演算結果を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定された角度のTAN(正接)値を演算し、(d)で指定されたデバイス番号に演算結果を格納します。



- (s)で指定する角度は、ラジアン単位(角度 $\times\pi\div180$)で設定します。
- 演算結果が-0、またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

 39ページ 注意事項

注意事項

(s)で指定する角度が、 $\pi/2$ ラジアン、 $(3/2)\pi$ ラジアンの場合、ラジアン値に演算誤差が生じるため、エラーは発生しません。

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0、非正規化数、非数、 $\pm\infty$ のとき。

Point

角度 $\leftrightarrow$ ラジアン変換については、RAD(P)命令、DEG(P)命令を参照してください。

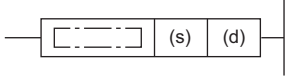
 895ページ RAD(P)

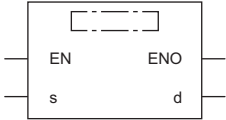
 897ページ DEG(P)

単精度実数 SIN^{-1} 演算

ASIN(P)

単精度実数で指定した SIN 値から角度を演算します。

ラダー	ST*1
	ENO:=ASINP(EN,s,d);

FBD/LD*1


*1 ASIN命令はST, FBD/LDでは対応していません。汎用ファンクションのASINを使用してください。

📖 1263ページ ASIN(_E)

■実行条件

命令	実行条件
ASIN	
ASINP	

設定データ

■内容、範囲、データ型

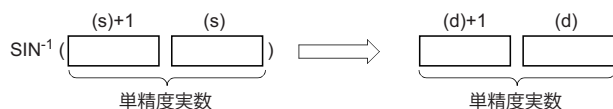
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	SIN^{-1} (逆正弦)演算を行う SIN 値または、 SIN 値が格納されている先頭デバイス	-1.0~1.0	単精度実数	ANYREAL_32
(d)	演算結果を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定された角度のSIN(正弦)値から角度を演算し、(d)で指定されたデバイス番号に演算結果を格納します。



- (s)で指定するSIN値は、-1.0~1.0の範囲で設定できます。
- (d)に格納する角度(演算結果)は、ラジアン単位で格納します。
- 演算結果が-0、またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0、非正規化数、非数、 $\pm\infty$ のとき。
3285H	(s)で指定した値が-1.0~1.0の範囲外のとき。

Point

ラジアン↔角度変換については、RAD(P)命令、DEG(P)命令を参照してください。

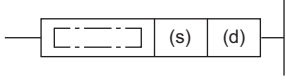
📖 895ページ RAD(P)

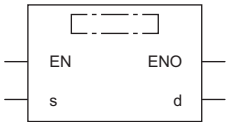
📖 897ページ DEG(P)

単精度実数 COS^{-1} 演算

ACOS(P)

単精度実数で指定したCOS値から角度を演算します。

ラダー	ST ^{*1}
	ENO:=ACOSP(EN,s,d)

FBD/LD ^{*1}


*1 ACOS命令はST, FBD/LDでは対応していません。汎用ファンクションのACOSを使用してください。

1264ページ ACOS(_E)

■実行条件

命令	実行条件
ACOS	
ACOSP	

設定データ

■内容、範囲、データ型

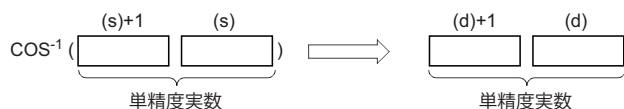
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	COS^{-1} (逆余弦)演算を行うCOS値または、COS値が格納されている先頭デバイス	-1.0~1.0	単精度実数	ANYREAL_32
(d)	演算結果を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定された角度のCOS(余弦)値から角度を演算し、(d)で指定されたデバイス番号に演算結果を格納します。



- (s)で指定するCOS値は、-1.0~1.0の範囲で設定できます。
- (d)に格納する角度(演算結果)は、ラジアン単位で格納します。
- 演算結果が-0、またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

☞ 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0、非正規化数、非数、±∞のとき。
3285H	(s)で指定した値が-1.0~1.0の範囲外のとき。

Point

ラジアン↔角度変換については、RAD(P)命令、DEG(P)命令を参照してください。

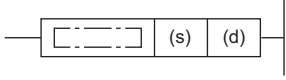
☞ 895ページ RAD(P)

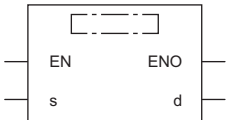
☞ 897ページ DEG(P)

単精度実数TAN⁻¹演算

ATAN(P)

単精度実数で指定したTAN値から角度を演算します。

ラダー	ST ^{*1}
	ENO:=ATANP(EN,s,d);

FBD/LD ^{*1}


*1 ATAN命令はST, FBD/LDでは対応していません。汎用ファンクションのATANを使用してください。

📖 1265ページ ATAN(_E)

■実行条件

命令	実行条件
ATAN	
ATANP	

設定データ

■内容、範囲、データ型

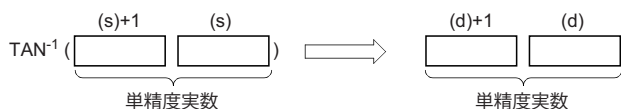
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	TAN ⁻¹ (逆正接)演算を行うTAN値または、TAN値が格納されている先頭デバイス	$0, 2^{-126} \leq (s) < 2^{128}$	単精度実数	ANYREAL_32
(d)	演算結果を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□	G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定された角度のTAN(正接)値から角度を演算し、(d)で指定されたデバイス番号に演算結果を格納します。



- (d)に格納する角度(演算結果)は、ラジアン単位で格納します。
- 演算結果が-0、またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0、非正規化数、非数、 $\pm\infty$ のとき。

Point

ラジアン↔角度変換については、RAD(P)命令、DEG(P)命令を参照してください。

📖 895ページ RAD(P)

📖 897ページ DEG(P)

倍精度実数SIN演算

SIND(P)

倍精度実数で指定した角度のSIN(正弦)値を演算します。

ラダー	ST*1
	ENO:=SINDP(EN,s,d);
FBD/LD*1	

*1 SIND命令はST, FBD/LDでは対応していません。汎用ファンクションのSINを使用してください。

1260ページ SIN(_E)

■実行条件

命令	実行条件
SIND	
SINDP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	SIN(正弦)演算を行う角度データまたは、角度データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	演算結果を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- Diagram illustrating the conversion of a floating-point number to a fixed-point number (integer) for the `SIN` function:
- Left side (Floating-point representation): `SIN` function takes four arguments, each in a box. Above the boxes are labels: $(s)+3$, $(s)+2$, $(s)+1$, and (s) . A bracket below the boxes is labeled "倍精度実数" (Double precision real number).
- Right side (Fixed-point representation): The same `SIN` function structure, but the arguments are labeled $(d)+3$, $(d)+2$, $(d)+1$, and (d) . A bracket below the boxes is labeled "倍精度実数" (Double precision real number).
- An arrow points from the left side to the right side, indicating the conversion process.

- 👉 39ページ 注意事項

17-

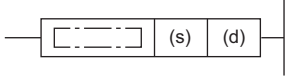
Point

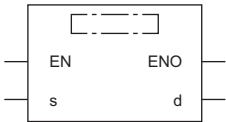
➡ 899ページ RADD(P)
➡ 901ページ DEGD(P)

倍精度実数COS演算

COSD(P)

倍精度実数で指定した角度のCOS(余弦)値を演算します。

ラダー	ST*1
	ENO:=COSDP(EN,s,d);

FBD/LD*1


*1 COSD命令はST, FBD/LDでは対応していません。汎用ファンクションのCOSを使用してください。

1261ページ COS(_E)

■実行条件

命令	実行条件
COSD	
COSDP	

設定データ

■内容、範囲、データ型

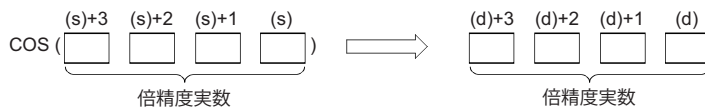
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	COS(余弦)演算を行う角度データまたは、角度データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	演算結果を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (s)で指定された角度のCOS(余弦)値を演算し、(d)で指定されたデバイス番号に演算結果を格納します。



- (s)で指定する角度は、ラジアン単位(角度 $\times\pi\div180$)で設定します。
- 演算結果が-0、またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0、非正規化数、非数、 $\pm\infty$ のとき。

Point

角度 $\leftrightarrow$ ラジアン変換については、RADD(P)命令、DEGD(P)命令を参照してください。

📖 899ページ RADD(P)

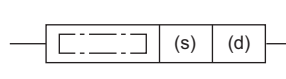
📖 901ページ DEGD(P)

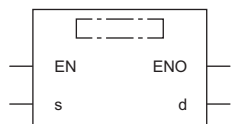
倍精度実数TAN演算

17

TAND(P)

倍精度実数で指定した角度のTAN(正接)値を演算します。

ラダー	ST*1
	ENO:=TANDP(EN,s,d);

FBD/LD*1


*1 TAND命令はST, FBD/LDでは対応していません。汎用ファンクションのTANを使用してください。
1262ページ TAN(_E)

■実行条件

命令	実行条件
TAND	
TANDP	

設定データ

■内容、範囲、データ型

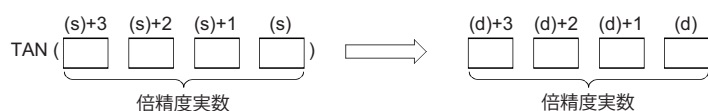
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	TAN(正接)演算を行う角度データまたは、角度データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	演算結果を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—	—

機能

- (s)で指定された角度のTAN(正接)値を演算し、(d)で指定されたデバイス番号に演算結果を格納します。



- (s)で指定する角度は、ラジアン単位(角度 $\times\pi\div180$)で設定します。
- 演算結果が-0またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

 39ページ 注意事項

注意事項

(s)で指定する角度が、 $\pi/2$ ラジアン、 $(3/2)\pi$ ラジアンの場合、ラジアン値に演算誤差が生じるため、エラーは発生しません。

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0、非正規化数、非数、 $\pm\infty$ のとき。

Point

角度 $\leftrightarrow$ ラジアン変換については、RADD(P)命令、DEGD(P)命令を参照してください。

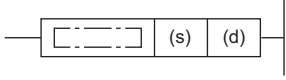
 899ページ RADD(P)

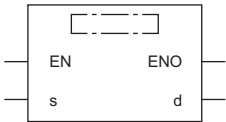
 901ページ DEGD(P)

倍精度実数 SIN^{-1} 演算

ASIND(P)

倍精度実数で指定した SIN 値から角度を演算します。

ラダー	ST*1
	ENO:=ASINDP(EN,s,d)

FBD/LD*1


*1 ASIND命令はST, FBD/LDでは対応していません。汎用ファンクションのASINを使用してください。

1263ページ ASIN(_E)

■実行条件

命令	実行条件
ASIND	
ASINDP	

設定データ

■内容, 範囲, データ型

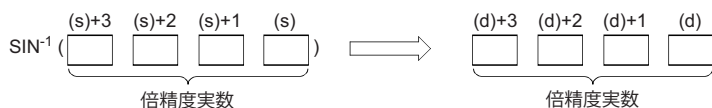
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	SIN^{-1} (逆正弦)演算を行う SIN 値または、 SIN 値が格納されている先頭デバイス	-1.0~1.0	倍精度実数	ANYREAL_64
(d)	演算結果を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (s)で指定された角度のSIN(正弦)値から角度を演算し、(d)で指定されたデバイス番号に演算結果を格納します。



- (s)で指定するSIN値は、-1.0~1.0の範囲で設定できます。
- (d)に格納する角度(演算結果)は、ラジアン単位で格納します。
- 演算結果が-0またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

🔍 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3285H	(s)で指定した値が-1.0~1.0の範囲外のとき。

Point

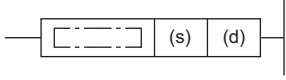
ラジアン↔角度変換については、RADD(P)命令、DEGD(P)命令を参照してください。

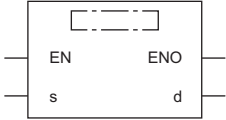
🔍 899ページ RADD(P)

🔍 901ページ DEGD(P)

ACOSD(P)

倍精度実数で指定したCOS値から角度を演算します。

ラダー	ST*1
	ENO:=ACOSDP(EN,s,d);

FBD/LD*1


*1 ACOSD命令はST, FBD/LDでは対応していません。汎用ファンクションのACOSを使用してください。
[1264ページ ACOS\(_E\)](#)

■実行条件

命令	実行条件
ACOSD	
ACOSDP	

設定データ

■内容、範囲、データ型

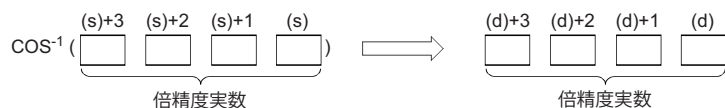
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	COS ⁻¹ (逆余弦)演算を行うCOS値または、COS値が格納されている先頭デバイス	-1.0~1.0	倍精度実数	ANYREAL_64
(d)	演算結果を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (s)で指定されたCOS(余弦)値から角度を演算し、(d)で指定されたデバイス番号に演算結果を格納します。



- (s)で指定するCOS値は、-1.0~1.0の範囲で設定できます。
- (d)に格納する角度(演算結果)は、ラジアン単位で格納します。
- 演算結果が-0またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

🔍 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0、非正規化数、非数、±∞のとき。
3285H	(s)で指定した値が-1.0~1.0の範囲外のとき。

Point

ラジアン↔角度変換については、RADD(P)命令、DEGD(P)命令を参照してください。

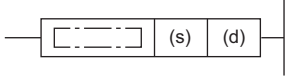
🔍 899ページ RADD(P)

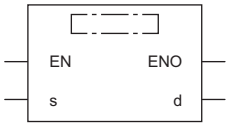
🔍 901ページ DEGD(P)

倍精度実数TAN⁻¹演算

ATAND(P)

倍精度実数で指定したTAN値から角度を演算します。

ラダー	ST ^{*1}
	ENO:=ATANDP(EN,s,d);

FBD/LD ^{*1}


*1 ATAND命令はST, FBD/LDでは対応していません。汎用ファンクションのATANを使用してください。

📖 1265ページ ATAN(_E)

■実行条件

命令	実行条件
ATAND	
ATANDP	

設定データ

■内容, 範囲, データ型

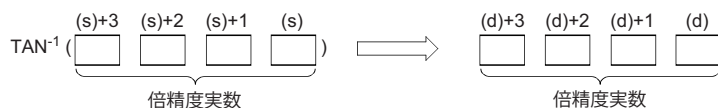
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	TAN ⁻¹ (逆正接)演算を行うTAN値または、TAN値が格納されている先頭デバイス	$0, 2^{-1022} \leq (s) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	演算結果を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (s)で指定された角度のTAN(正接)値から角度を演算し、(d)で指定されたデバイス番号に演算結果を格納します。



- (d)に格納する角度(演算結果)は、ラジアン単位で格納します。
- 演算結果が-0またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。

Point

ラジアン↔角度変換については、RADD(P)命令、DEGD(P)命令を参照してください。

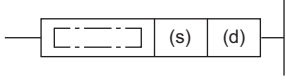
📖 899ページ RADD(P)

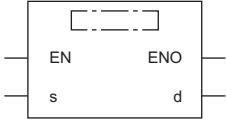
📖 901ページ DEGD(P)

BCD型SIN演算

BSIN(P)

BCD値で指定した角度のSIN(正弦)値を演算します。

ラダー	ST
	ENO:=BSIN(EN,s,d); ENO:=BSINP(EN,s,d);

FBD/LD


■実行条件

命令	実行条件
BSIN	
BSINP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	SIN(正弦)演算を行うデータまたは、データが格納されているデバイス	0~9999	BCD4桁	ANY16
(d)	演算結果を格納する先頭デバイス	—	BCD4桁	ANY16_ARRAY (要素数: 3)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- (s)で指定された値(角度)のSIN(正弦)値を演算し、(d)で指定されたデバイスに演算結果の符号を、(d)+1、(d)+2で指定されたデバイスに演算結果を格納します。

SIN (s) =

(d): 符号

(d)+1: 整数部

(d)+2: 小数部

- (s)で指定する値は、0~360°(DEG.単位)をBCD値で設定します。
- (d)に格納する演算結果の符号は、演算結果が正のとき0を格納し、負のとき1を格納します。
- (d)+1、(d)+2に格納する演算結果は、BCD値で-1.000~1.000の範囲内の値です。
- 演算結果は、小数部5桁目を四捨五入した値となります。

エラー

エラーコード (SD0)	内容
3285H	(s)に指定可能な範囲外のデータを設定したとき。 • 指定したデータがBCD値でない • 指定したデータが0~360の範囲内でない

BCD型COS演算

17

BCOS(P)

BCD値で指定した角度のCOS(余弦)値を演算します。

ラダー	ST
	<pre>ENO:=BCOS(EN,s,d); ENO:=BCOSP(EN,s,d);</pre>

FBD/LD

■実行条件

命令	実行条件
BCOS	
BCOSP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	COS(余弦)演算を行うデータまたは、データが格納されているデバイス	0~9999	BCD4桁	ANY16
(d)	演算結果を格納する先頭デバイス	—	BCD4桁	ANY16_ARRAY (要素数: 3)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- (s)で指定された値(角度)のCOS(余弦)値を演算し、(d)で指定されたワードデバイスに演算結果の符号を、(d)+1、(d)+2で指定されたワードデバイスに演算結果を格納します。

COS (s) =

(d): 符号

(d)+1: 整数部

(d)+2: 小数部

- (s)で指定する値は、0~360°(DEG.単位)をBCD値で設定します。
- (d)に格納する演算結果の符号は、演算結果が正のとき0を格納し、負のとき1を格納します。
- (d)+1、(d)+2に格納する演算結果は、BCD値で-1.000~1.000の範囲内の値です。
- 演算結果は、小数部5桁目を四捨五入した値となります。

エラー

エラーコード (SD0)	内容
3285H	(s)に指定可能な範囲外のデータを設定したとき。 • 指定したデータがBCD値でない • 指定したデータが0~360の範囲内でない

BCD型TAN演算

BTAN(P)

BCD値で指定した角度のTAN(正接)値を演算します。

ラダー	ST
	ENO:=BTAN(EN,s,d); ENO:=BTANP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
BTAN	
BTANP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	TAN(正接)演算を行うデータまたは、データが格納されているデバイス	0~9999	BCD4桁	ANY16
(d)	演算結果を格納する先頭デバイス	—	BCD4桁	ANY16_ARRAY (要素数: 3)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—

機能

- (s)で指定された値(角度)のTAN(正接)値を演算し、(d)で指定されたデバイスに演算結果の符号を、(d)+1、(d)+2で指定されたデバイスに演算結果を格納します。

TAN (s) =

(d): 符号

(d)+1: 整数部

(d)+2: 小数部

- (s)で指定する値は、0~360°(DEG.単位)をBCD値で設定します。
- (d)に格納する演算結果の符号は、演算結果が正のとき0を格納し、負のとき1を格納します。
- (d)+1、(d)+2に格納する演算結果は、BCD値で-57.2901~57.2903の範囲内の値です。
- 演算結果は、小数部5桁目を四捨五入した値となります。

エラー

エラーコード (SD0)	内容
3285H	(s)に指定可能な範囲外のデータを設定したとき。 • 指定したデータがBCD値でない • 指定したデータが0~360の範囲内でない • 指定したデータが90°, 270°

BCD型 SIN^{-1} 演算

BASIN(P)

BCD値で指定した値の SIN^{-1} (逆正弦)値を演算します。

ラダー	ST
	ENO:=BASIN(EN,s,d); ENO:=BASINP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
BASIN	
BASINP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	SIN^{-1} (逆正弦)演算を行うデータが格納されている先頭デバイス	0~9999	BCD4桁	ANY16_ARRAY (要素数: 3)
(d)	演算結果を格納するデバイス	—	BCD4桁	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□	G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	—	○	—	—	—	—
(d)	○	○	○	○	○	○	—	—	○	—	—	—	—

機能

- (s)で指定された値の SIN^{-1} (逆正弦)値を演算し、(d)で指定されたデバイスに演算結果(角度)を格納します。

$$\text{SIN}^{-1} \left(\begin{array}{c} \text{(s)} \\ \boxed{} \end{array} \begin{array}{c} \text{(s)+1} \\ \boxed{} \end{array} \begin{array}{c} \text{(s)+2} \\ \boxed{} \end{array} \right) = \text{(d)}$$

(s): 符号

(s)+1: 整数部

(s)+2: 小数部

- (s)には、演算するデータの符号を設定します。演算データが正のときは0を格納し、負のときは1を格納します。
- (s)+1, (s)+2には、演算するデータの整数部、小数部をそれぞれBCD値で格納します。(0~1.0000を設定できます。)
- (d)に格納する演算結果は、BCD値で0~90°, 270~360°(DEG.単位)の範囲内の値です。
- 演算結果は、小数部を四捨五入した値となります。

エラー

エラーコード (SD0)	内容
3285H	(s)に指定可能な範囲外のデータを設定したとき。 • 指定したデータがBCD値でない • 指定したデータが-1.0000~1.0000の範囲内でない

BCD型 $\cos^{-1}$ 演算

BACOS(P)

BCD値で指定した値の $\cos^{-1}$ (逆余弦)値を演算します。

ラダー	ST
	ENO:=BACOS(EN,s,d); ENO:=BACOSP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
BACOS	
BACOSP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	$\cos^{-1}$ (逆余弦)演算を行うデータが格納されている先頭デバイス	0~9999	BCD4桁	ANY16_ARRAY (要素数: 3)
(d)	演算結果を格納するデバイス	—	BCD4桁	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

機能

- (s)で指定された値の COS^{-1} (逆余弦)値を演算し、(d)で指定されたデバイスに演算結果(角度)を格納します。

$\text{COS}^{-1} \left(\overset{(s)}{\boxed{}} \overset{(s)+1}{\boxed{}} . \overset{(s)+2}{\boxed{}} \right) = (d)$

(s): 符号

(s)+1: 整数部

(s)+2: 小数部

- (s)には、演算するデータの符号を設定します。演算データが正のときは0を格納し、負のときは1を格納します。
- (s)+1, (s)+2には、演算するデータの整数部、小数部をそれぞれBCD値で格納します。(0~1.0000を設定できます。)
- (d)に格納する演算結果は、BCD値で0~180°(DEG.単位)の範囲内の値です。
- 演算結果は、小数部を四捨五入した値となります。

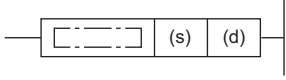
エラー

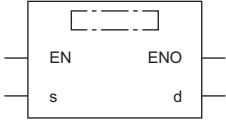
エラーコード (SD0)	内容
3285H	(s)に指定可能な範囲外のデータを設定したとき。 • 指定したデータがBCD値でない • 指定したデータが-1.0000~1.0000の範囲内でない

BCD型TAN⁻¹演算

BATAN(P)

BCD値で指定した値のTAN⁻¹(逆正接)値を演算する。

ラダー	ST
	ENO:=BATAN(EN,s,d); ENO:=BATANP(EN,s,d);

FBD/LD


■実行条件

命令	実行条件
BATAN	
BATANP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	TAN ⁻¹ (逆正接)演算を行うデータが格納されている先頭デバイス	0~9999	BCD4桁	ANY16_ARRAY (要素数: 3)
(d)	演算結果を格納するデバイス	—	BCD4桁	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—	—

機能

- (s)で指定された値のTAN-1(逆正接)値を演算し、(d)で指定されたデバイスに演算結果(角度)を格納します。

$\text{TAN}^{-1} \left(\overset{(s)}{\boxed{}} \overset{(s)+1}{\boxed{}} . \overset{(s)+2}{\boxed{}} \right) = (d)$

(s): 符号

(s)+1: 整数部

(s)+2: 小数部

- (s)には、演算するデータの符号を設定します。演算データが正のときは0を格納し、負のときは1を格納します。
- (s)+1, (s)+2には、演算するデータの整数部、小数部をそれぞれBCD値で格納します。(0~9999.9999を設定できます。)
- (d)に格納する演算結果は、BCD値で0~90°, 270~360°(DEG.単位)の範囲内の値です。
- 演算結果は、小数部を四捨五入した値となります。

エラー

エラーコード (SD0)	内容
3285H	(s)に指定したデータがBCD値でないとき。

単精度実数角度→ラジアン変換

RAD(P)

角度の大きさの単位を、単精度実数で指定した度単位からラジアン単位に変換する。

ラダー	ST
	ENO:=RAD(EN,s,d); ENO:=RADP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
RAD	
RADP	

設定データ

■内容、範囲、データ型

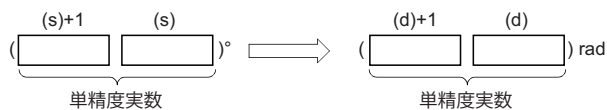
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	ラジアン単位に変換する角度または、角度が格納されている先頭デバイス	$0, 2^{-126} \leq (s) < 2^{128}$	単精度実数	ANYREAL_32
(d)	ラジアン単位に変換した値を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

機能

- 角度の大きさの単位を、(s)で指定された度単位(DEG.単位)からラジアン単位に変換し、(d)で指定されたデバイス番号に格納します。



- 度単位→ラジアン単位変換は、下記のように行います。

$$\text{ラジアン単位} = \text{度単位} \times \frac{\pi}{180}$$

- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ (d) < 2^{128}$

単精度実数ラジアン→角度変換

DEG(P)

角度の大きさの単位を、単精度実数で指定したラジアン単位から度単位に変換します。

ラダー	ST
	ENO:=DEG(EN,s,d); ENO:=DEGP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
DEG	
DEGP	

設定データ

■内容、範囲、データ型

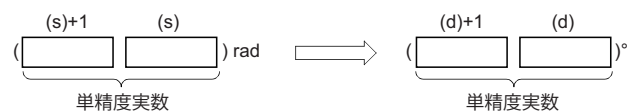
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	度単位に変換するラジアン角度または、ラジアン角度が格納されている先頭デバイス	$0, 2^{-126} \leq (s) < 2^{128}$	単精度実数	ANYREAL_32
(d)	度単位に変換した値を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

機能

- 角度の大きさの単位を、(s)で指定されたラジアン単位から度単位(DEG.単位)に変換し、(d)で指定されたデバイス番号に格納します。



- ラジアン単位→度単位変換は、下記のように行います。

$$\text{度単位} = \text{ラジアン単位} \times \frac{180}{\pi}$$

- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ (d) < 2^{128}$

倍精度実数角度→ラジアン変換

RADD(P)

角度の大きさの単位を、倍精度実数で指定した度単位からラジアン単位に変換します。

ラダー	ST
	<pre>ENO:=RADD(EN,s,d); ENO:=RADDP(EN,s,d);</pre>

FBD/LD

■実行条件

命令	実行条件
RADD	
RADDP	

設定データ

■内容、範囲、データ型

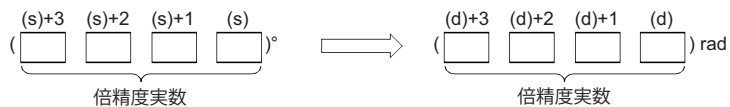
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	ラジアン単位に変換する角度または、角度が格納されている先頭デバイス	$0, 2^{-1022} \leq (s) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	ラジアン単位に変換した値を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—	—

機能

- 角度の大きさの単位を、(s)で指定された度単位(DEG.単位)からラジアン単位に変換し、(d)で指定されたデバイス番号に格納します。



- 度単位→ラジアン単位変換は、下記のように行います。

$$\text{ラジアン単位} = \text{度単位} \times \frac{\pi}{180}$$

- 演算結果が-0またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

☞ 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ (d) < 2^{1024}$

倍精度実数ラジアン→角度変換

DEGD(P)

角度の大きさの単位を，倍精度実数で指定したラジアン単位から度単位に変換します。

ラダー	ST
	<pre>ENO:=DEGD(EN,s,d); ENO:=DEGDP(EN,s,d);</pre>

FBD/LD

■実行条件

命令	実行条件
DEGD	
DEGDP	

設定データ

■内容，範囲，データ型

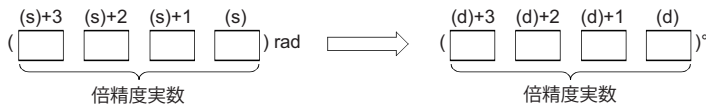
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	度単位に変換するラジアン角度または，ラジアン角度が格納されている先頭デバイス	$0, 2^{-1022} \leq (s) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	度単位に変換した値を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—	—

機能

- 角度の大きさの単位を、(s)で指定されたラジアン単位から度単位(DEG.単位)に変換し、(d)で指定されたデバイス番号に格納します。



- ラジアン単位→度単位変換は、下記のように行います。

$$\text{度単位} = \text{ラジアン単位} \times \frac{180}{\pi}$$

- 演算結果が-0またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

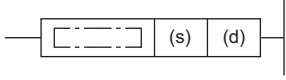
エラー

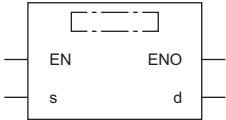
エラーコード (SD0)	内容
3282H	(s)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ d < 2^{1024}$

単精度実数平方根

ESQRT(P)

単精度実数で指定した値の平方根を演算します。

ラダー	ST
	ENO:=ESQRT(EN,s,d); ENO:=ESQRT(EN,s,d);

FBD/LD


■実行条件

命令	実行条件
ESQRT	
ESQRT(EN,s,d)	

設定データ

■内容, 範囲, データ型

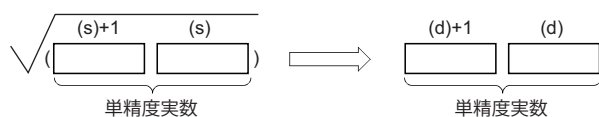
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	平方根演算を行うデータまたは、データが格納されている先頭デバイス	$0, 2^{-126} \leq (s) < 2^{128}$	単精度実数	ANYREAL_32
(d)	演算結果を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(s)	—	—	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	—	—	—	—

機能

- (s)で指定された値の平方根を演算し、(d)で指定されたデバイス番号に演算結果を格納します。



- (s)で指定する値は、正の数のみ設定できます。(負の数では演算できません。)
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3285H	(s)で指定した値が負の数のとき。

EDSQRT(P)

倍精度実数で指定した値の平方根を演算します。

ラダー	ST
	ENO:=EDSQRT(EN,s,d); ENO:=EDSQRT(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
EDSQRT	
EDSQRTP	

設定データ

■内容，範囲，データ型

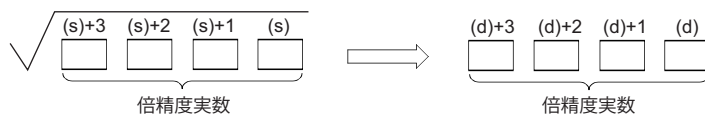
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	平方根演算を行うデータまたは、データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	演算結果を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (s)で指定された値の平方根を演算し、(d)で指定されたデバイス番号に演算結果を格納します。



- (s)で指定する値は、正の数のみ設定できます。(負の数では演算できません。)
- 演算結果が-0またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3285H	(s)で指定した値が負の数のとき。

単精度実数指数演算

EXP(P)

単精度実数で指定した値の指数を演算します。

ラダー	ST*1
	ENO:=EXPP(EN,s,d);

FBD/LD*1

*1 EXP命令はST, FBD/LDでは対応していません。汎用ファンクションのEXPを使用してください。
 1259ページ EXP(_E)

■実行条件

命令	実行条件
EXP	
EXPP	

設定データ

■内容, 範囲, データ型

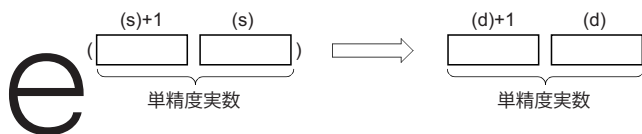
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	指数演算を行うデータまたは、データが格納されている先頭デバイス	$0, 2^{-126} \leq (s) < 2^{128}$	単精度実数	ANYREAL_32
(d)	演算結果を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定された値の指数を演算し、(d)で指定されたデバイスに演算結果を格納します。



- 指数演算では、底(e)を「2.71828」として演算します。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

🔍 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ d < 2^{128}$

Point

- EXP(P)命令は自然対数での演算を行います。常用対数での値を求める場合は、常用対数に0.43429で割った値を(s)に指定してください。

$$10^X = e^{\frac{X}{0.43429}}$$

倍精度実数指数演算

EXPDP(P)

倍精度実数で指定した値の指数を演算します。

ラダー	ST*1
	ENO:=EXPDP(EN,s,d);

FBD/LD*1

*1 EXPD命令はST, FBD/LDでは対応していません。汎用ファンクションのEXPを使用してください。
 1259ページ EXP(_E)

■実行条件

命令	実行条件
EXPDP	
EXPDP	

設定データ

■内容, 範囲, データ型

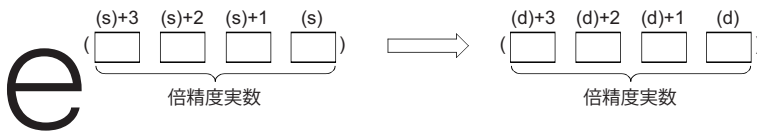
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	指数演算を行うデータまたは、データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	演算結果を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (s)で指定された値の指数を演算し、(d)で指定されたデバイスに演算結果を格納します。



- 指数演算では、底(e)を「2.71828」として演算します。
- 演算結果が-0またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき。) $ d < 2^{1024}$

Point

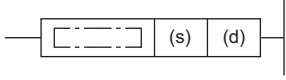
- EXPD(P)命令では、自然対数での演算を行います。常用対数での値を求める場合は、常用対数に0.43429で割った値を(s)に指定してください。

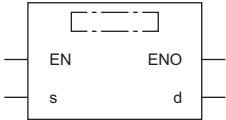
$$10^X = e^{\frac{X}{0.43429}}$$

単精度実数自然対数演算

LOG(P)

単精度実数で指定した値の自然対数(e)を底としたときの対数を演算します。

ラダー	ST*1
	ENO:=LOGP(EN,s,d);

FBD/LD*1


*1 LOG命令はST, FBD/LDでは対応していません。汎用ファンクションのLOGを使用してください。

📖 1257ページ LOG(_E)

■実行条件

命令	実行条件
LOG	
LOGP	

設定データ

■内容, 範囲, データ型

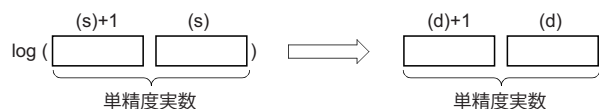
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	自然対数演算を行うデータまたは、データが格納されている先頭デバイス	$0, 2^{-126} \leq (s) < 2^{128}$	単精度実数	ANYREAL_32
(d)	演算結果を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定された値の自然対数eを底としたときの対数を演算し、(d)で指定されたデバイスに演算結果を格納します。



- (s)で指定する値は、正の数のみ設定できます。(負の数では演算できません。)
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

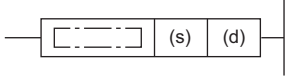
エラー

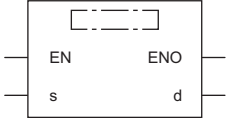
エラーコード (SD0)	内容
3282H	(s)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3285H	(s)に指定可能な範囲外のデータを設定したとき。 • 指定した値が負の数 • 指定した値が0

倍精度実数自然対数演算

LOGD(P)

倍精度実数で指定した値の自然対数(e)を底としたときの対数を演算します。

ラダー	ST*1
	ENO:=LOGDP(EN,s,d);

FBD/LD*1


*1 LOGD命令はST, FBD/LDでは対応していません。汎用ファンクションのLOGを使用してください。

1257ページ LOG(_E)

■実行条件

命令	実行条件
LOGD	
LOGDP	

設定データ

■内容、範囲、データ型

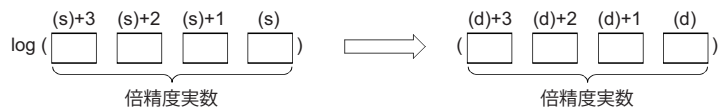
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	自然対数演算を行うデータまたは、データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	演算結果を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (s)で指定された値の自然対数eを底としたときの対数を演算し、(d)で指定されたデバイスに演算結果を格納します。



- (s)で指定する値は、正の数のみ設定できます。(負の数では演算できません。)
- 演算結果が-0またはアンダーフローが発生したとき、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

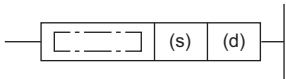
エラーコード (SD0)	内容
3282H	(s)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3285H	(s)に指定可能な範囲外のデータを設定したとき。 • 指定した値が負の数 • 指定した値が0

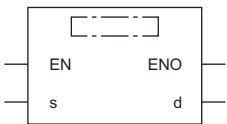
BCD4桁平方根

17

BSQRT(P)

BCD値(4桁)で指定した値の平方根を演算します。

ラダー	ST
	ENO:=BSQRT(EN,s,d); ENO:=BSQRT(EN,s,d);

FBD/LD


■実行条件

命令	実行条件
BSQRT	
BSQRT(EN,s,d)	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	平方根演算を行うデータまたは、データが格納されているデバイス	0~9999	BCD4桁	ANY16
(d)	演算結果を格納する先頭デバイス	—	BCD4桁	ANY16_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- (s)で指定したBCD4桁データの平方根を演算し、(d)で指定したデバイスに演算結果を格納します。

$$\sqrt{\boxed{(s)}} = \boxed{(d)}.\boxed{(d)+1}$$

(d): 整数部

(d)+1: 小数部

- (s)で指定する値は、BCD値で最大4桁(0~9999)です。
- (d)の演算結果は、それぞれBCD値で0~9999.9999を格納します。
- 演算結果は小数部5桁目を切り捨てた値となります。

エラー

エラーコード (SD0)	内容
3285H	(s)で指定したデータがBCD値以外るとき。

BDSQRT(P)

BCD値(8桁)で指定した値の平方根を演算します。

ラダー	ST
	ENO:=BDSQRT(EN,s,d); ENO:=BDSQRTP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
BDSQRT	
BDSQRTP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	平方根演算を行うデータまたは、データが格納されている先頭デバイス	0~99999999	BCD8桁	ANY32
(d)	演算結果を格納する先頭デバイス	—	BCD4桁	ANY16_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- (s)で指定したBCD8桁データの平方根を演算し、(d)で指定したデバイスに演算結果を格納します。

$$\sqrt{((s)+1 \text{ (s)})} = \text{(d)} \text{ (d)+1}$$

(s)+1, (s): 2ワードデータ

(d): 整数部

(d)+1: 小数部

- (s)で指定する値は、BCD値で最大8桁(0~99999999)です。
- (d)の演算結果は、それぞれBCD値で0~9999.9999を格納します。
- 演算結果は小数部5桁目を切り捨てた値となります。

エラー

エラーコード (SD0)	内容
3285H	(s)で指定したデータがBCD値以外るとき。

単精度実数べき乗演算

POW(P)

単精度実数のべき乗を演算します。

ラダー	ST
	ENO:=POW(EN,s1,s2,d); ENO:=POWP(EN,s1,s2,d);

FBD/LD

■実行条件

命令	実行条件
POW	
POWP	

設定データ

■内容、範囲、データ型

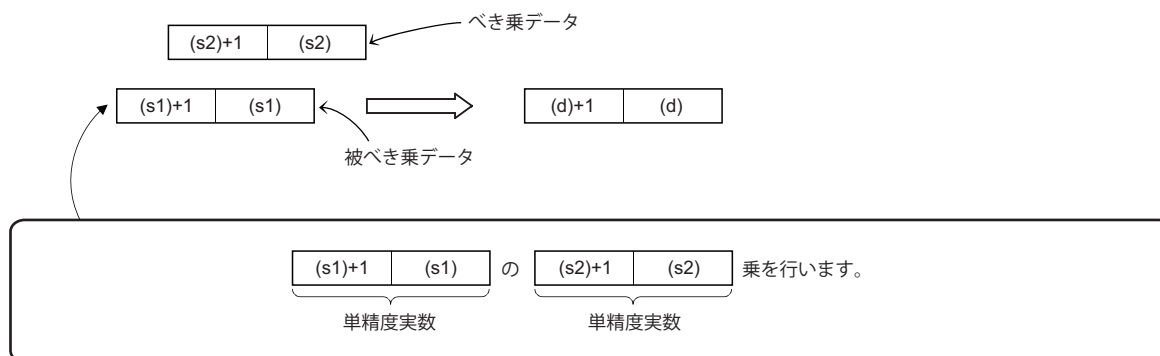
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	被べき乗データ, または被べき乗データを格納している先頭デバイス	$0, 2^{-126} \leq (s1) < 2^{128}$	単精度実数	ANYREAL_32
(s2)	べき乗データまたは, データが格納されている先頭デバイス	$0, 2^{-126} \leq (s2) < 2^{128}$	単精度実数	ANYREAL_32
(d)	演算結果を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	—	—	○	○	○	○	○	○	○	—	○	—	—
(s2)	—	—	○	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	○	—	—	—	—

機能

- (s1)で指定した単精度実数を(s2)で指定した単精度実数でべき乗演算して、(d)で指定したデバイスに演算結果を格納します。



- (s1), (s2)に指定できる値、および格納できる値は0, $2^{-126} \leq |\text{設定値(格納値)}| < 2^{128}$ です。
- 演算結果が-0またはアンダーフローが発生したときは、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s1)または(s2)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき) $ d < 2^{128}$

倍精度実数べき乗演算

POWD(P)

倍精度実数のべき乗を演算します。

ラダー	ST
	ENO:=POWD(EN,s1,s2,d); ENO:=POWDP(EN,s1,s2,d);

FBD/LD

■実行条件

命令	実行条件
POWD	
POWDP	

設定データ

■内容、範囲、データ型

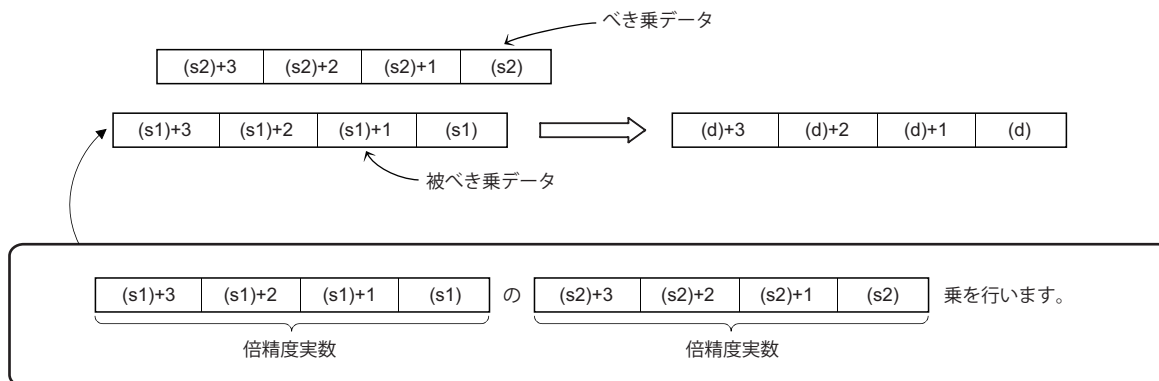
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	被べき乗データ, または被べき乗データを格納している先頭デバイス	$0, 2^{-1022} \leq (s1) < 2^{1024}$	倍精度実数	ANYREAL_64
(s2)	べき乗データまたは, データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s2) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	演算結果を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	○	—	○	—	○	—	—
(s2)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (s1)で指定した倍精度実数を(s2)で指定した倍精度実数でべき乗演算して、(d)で指定したデバイスに演算結果を格納します。



- (s1), (s2)に指定できる値, および格納できる値は0, $2^{-1022} \leq |\text{設定値(格納値)}| < 2^{1024}$ です。
- 演算結果が-0またはアンダーフローが発生したときは, 演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合, 丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は, 下記を参照してください。

📖 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s1)または(s2)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき) $ (d) < 2^{128}$

単精度実数常用対数演算

LOG10(P)

単精度実数で指定した値の常用対数(10を底とした対数)を演算します。

ラダー	ST
	ENO:=LOG10(EN,s,d); ENO:=LOG10P(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
LOG10	
LOG10P	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	常用対数演算を行うデータまたは、データが格納されている先頭デバイス	$0, 2^{-126} \leq (s) < 2^{128}$	単精度実数	ANYREAL_32
(d)	演算結果を格納する先頭デバイス	—	単精度実数	ANYREAL_32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	○	○	○	○	○	—	○	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

機能

- (s)で指定された値の常用対数(10を底とした対数)を演算し、(d)で指定されたデバイス番号に演算結果を格納します。



- (s)で指定する値は、正の数のみ設定できます。(負の数では演算できません。)
- 演算結果が-0またはアンダーフローが発生したときは、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

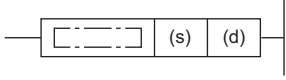
エラー

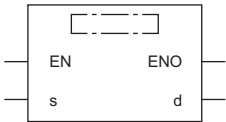
エラーコード (SD0)	内容
3282H	(s)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3285H	(s)に指定可能な範囲外のデータを設定したとき。 • 指定した値が負の数 • 指定した値が0

倍精度実数常用対数演算

LOG10D(P)

倍精度実数で指定した値の常用対数(10を底とした対数)を演算します。

ラダー	ST
	ENO:=LOG10D(EN,s,d); ENO:=LOG10DP(EN,s,d);

FBD/LD


■実行条件

命令	実行条件
LOG10D	
LOG10DP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	常用対数演算を行うデータまたは、データが格納されている先頭デバイス	$0, 2^{-1022} \leq (s) < 2^{1024}$	倍精度実数	ANYREAL_64
(d)	演算結果を格納する先頭デバイス	—	倍精度実数	ANYREAL_64
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	○	—	○	—	○	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

機能

- (s)で指定された値の常用対数(10を底とした対数)を演算し、(d)で指定されたデバイス番号に演算結果を格納します。

$$\log_{10} \left(\underbrace{\left((s)+3 \cdot (s)+2 \cdot (s)+1 \cdot (s) \right)}_{\text{倍精度実数}} \right) \longrightarrow \left(\underbrace{(d)+3 \cdot (d)+2 \cdot (d)+1 \cdot (d)}_{\text{倍精度実数}} \right)$$

- (s)で指定する値は、正の数のみ設定できます。(負の数では演算できません。)
- 演算結果が-0またはアンダーフローが発生したときは、演算結果は0となります。
- エンジニアリングツールから入力値を設定する場合、丸め誤差が発生する場合があります。エンジニアリングツールから入力値を設定する場合の注意事項は、下記を参照してください。

📖 39ページ 注意事項

エラー

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0、非正規化数、非数、 $\pm\infty$ のとき。
3285H	(s)に指定可能な範囲外のデータを設定したとき。 • 指定した値が負の数 • 指定した値が0

単精度実数最大値検索

EMAX(P)

単精度実数のブロックデータの最大値を検索する。

ラダー	ST*1
	ENO:=EMAXP(EN,s,n,d);

FBD/LD*1

*1 EMAX命令はST, FBD/LDでは対応していません。汎用ファンクションのMAXを使用してください。
 1295ページ MAX(_E), MIN(_E)

■実行条件

命令	実行条件
EMAX	
EMAXP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	最大値を検索する単精度実数または、単精度実数が格納されている先頭デバイス	$0, 2^{-126} \leq (s) < 2^{128}$	単精度実数	ANYREAL_32*1
(d)	検索結果を格納する先頭デバイス (d)~(d)+1: 最大値 (d)+2: 位置 (d)+3: 個数	—	単精度実数	—*2 (ANY_REAL_32_ARRAY)
(n)	単精度実数ブロックデータの点数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

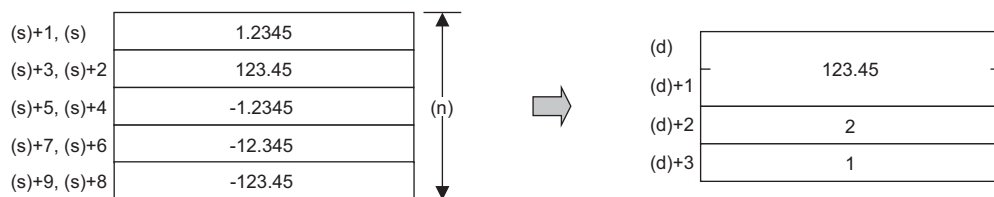
*2 使用するプログラム言語にかかわらず、デバイスで指定します。ラベルは指定しないでください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

機能

- (s)から(n)点分の単精度実数のブロックデータに対して、最大値を検索して、(d)で指定したデバイスへ格納します。最初に最大値を検出したデータが、(s)から何点目かを(d)+2で指定したデバイスへ格納し、最大値と同じ値の個数を、(d)+3で指定したデバイスへ格納します。
- 検索結果(位置)の値については、(s)で指定したブロックデータの先頭を1点目として扱います。



(d), (d)+1: 最大値

(d)+2: 位置

(d)+3: 個数

- (n)で指定した値が0の場合は無処理となります。

エラー

エラーコード (SD0)	内容
3282H	(s)で指定したブロックデータに、単精度実数以外の値が存在する場合。

倍精度実数最大値検索

EDMAX(P)

倍精度実数のブロックデータの最大値を検索します。

ラダー	ST*1
	ENO:=EDMAXP(EN,s,n,d);

FBD/LD*1

*1 EDMAX命令はST, FBD/LDでは対応していません。汎用ファンクションのMAXを使用してください。
 1295ページ MAX(_E), MIN(_E)

■実行条件

命令	実行条件
EDMAX	
EDMAXP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	最大値を検索する倍精度実数または、倍精度実数が格納されている先頭デバイス	$0, 2^{-1022} \leq (s) < 2^{1024}$	倍精度実数	ANYREAL_64*1
(d)	検索結果を格納する先頭デバイス (d)~(d)+3: 最大値 (d)+4: 位置 (d)+5: 個数	—	倍精度実数	—*2 (ANY_REAL_64_ARRAY)
(n)	倍精度実数ブロックデータの点数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

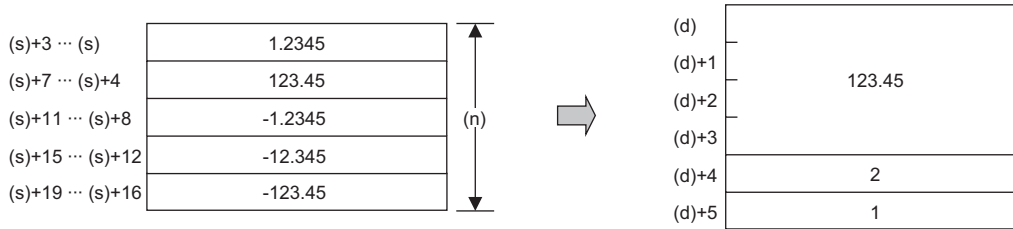
*2 使用するプログラム言語にかかわらず、デバイスで指定します。ラベルは指定しないでください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

機能

- (s)から(n)点分の倍精度実数のブロックデータに対して、最大値を検索して、(d)で指定したデバイスへ格納します。最初に最大値を検出したデータが、(s)から何点目かを(d)+4で指定したデバイスへ格納し、最大値と同じ値の個数を、(d)+5で指定したデバイスへ格納します。
- 検索結果(位置)の値については、(s)で指定したブロックデータの先頭を1点目として扱います。



(d), (d)+1,(d)+2,(d)+3: 最大値

(d)+4: 位置

(d)+5: 個数

- (n)で指定した値が0の場合は無処理となります。

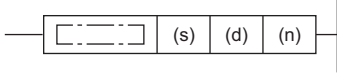
エラー

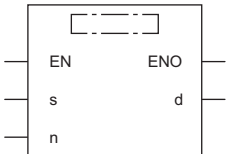
エラーコード (SD0)	内容
3282H	(s)で指定したブロックデータに、倍精度実数以外の値が存在する場合。

単精度実数最小値検索

EMIN(P)

単精度実数のブロックデータの最小値を検索します。

ラダー	ST*1
	ENO:=EMINP(EN,s,n,d);

FBD/LD*1


*1 EMIN命令はST, FBD/LDでは対応していません。汎用ファンクションのMINを使用してください。
 1295ページ MAX(_E), MIN(_E)

■実行条件

命令	実行条件
EMIN	
EMINP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	最小値を検索する単精度実数または、単精度実数が格納されている先頭デバイス	$0, 2^{-126} \leq (s) < 2^{128}$	単精度実数	ANYREAL_32*1
(d)	検索結果を格納する先頭デバイス (d)~(d)+1: 最小値 (d)+2: 位置 (d)+3: 個数	—	単精度実数	__*2 (ANY_REAL_32_ARRAY)
(n)	単精度実数ブロックデータの点数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるように配列を定義し、その配列型ラベルの要素を指定してください。

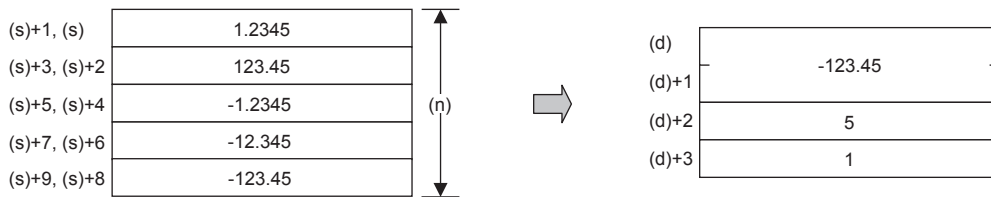
*2 使用するプログラム言語にかかわらず、デバイスで指定します。ラベルは指定しないでください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

機能

- (s)から(n)点分の単精度実数のブロックデータに対して、最小値を検索して、(d)で指定したデバイスへ格納します。最初に最小値を検出したデータが、(s)から何点目かを(d)+2で指定したデバイスへ格納し、最小値と同じ値の個数を、(d)+3で指定したデバイスへ格納します。
- 検索結果(位置)の値については、(s)で指定したブロックデータの先頭を1点目として扱います。



(d), (d)+1: 最小値

(d)+2: 位置

(d)+3: 個数

- (n)で指定した値が0の場合は無処理となります。

エラー

エラーコード (SD0)	内容
3282H	(s)で指定したブロックデータに、単精度実数以外の値が存在する場合。

倍精度実数最小値検索

EDMIN(P)

倍精度実数のブロックデータの最小値を検索します。

ラダー	ST*1
	ENO:=EDMINP(EN,s,n,d);

FBD/LD*1

*1 EDMIN命令はST, FBD/LDでは対応していません。汎用ファンクションのMINを使用してください。

1295ページ MAX(_E), MIN(_E)

■実行条件

命令	実行条件
EDMIN	
EDMINP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	最小値を検索する倍精度実数または、倍精度実数が格納されている先頭デバイス	$0, 2^{-1022} \leq (s) < 2^{1024}$	倍精度実数	ANYREAL_64*1
(d)	検索結果を格納する先頭デバイス (d)~(d)+3: 最小値 (d)+4: 位置 (d)+5: 個数	—	倍精度実数	—*2 (ANY_REAL_64_ARRAY)
(n)	倍精度実数ブロックデータの点数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

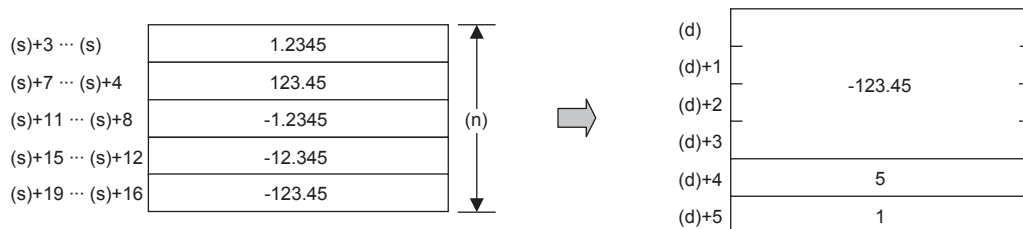
*2 使用するプログラム言語にかかわらず、デバイスで指定します。ラベルは指定しないでください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数				その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$		
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

機能

- (s)から(n)点分の倍精度実数のブロックデータに対して、最小値を検索して、(d)で指定したデバイスへ格納します。最初に最小値を検出したデータが、(s)から何点目かを(d)+4で指定したデバイスへ格納し、最小値と同じ値の個数を、(d)+5で指定したデバイスへ格納します。
- 検索結果(位置)の値については、(s)で指定したブロックデータの先頭を1点目として扱います。



(d), (d)+1, (d)+2, (d)+3: 最小値

(d)+4: 位置

(d)+5: 個数

- (n)で指定した値が0の場合は無処理となります。

エラー

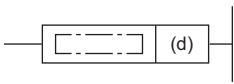
エラーコード (SD0)	内容
3282H	(s)で指定したブロックデータに、倍精度実数以外の値が存在する場合。

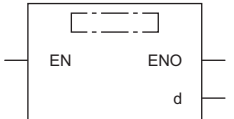
18.1 乱数命令

乱数発生

RND(P)

0~32767未満の乱数を発生させ、指定したデバイスに格納します。

ラダー	ST
	ENO:=RND(EN,d); ENO:=RNDP(EN,d);

FBD/LD


■実行条件

命令	実行条件
RND	
RNDP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	乱数を格納するデバイス	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 0	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 0	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

0~32767の乱数を発生し、(d)で指定されたデバイスに格納します。RND(P)命令は、ある算出式に従って乱数を発生させます。算出式では、前回の算出結果を係数として使用します。

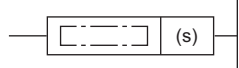
エラー

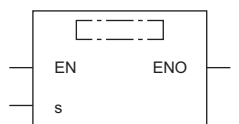
演算エラーはありません。

系列変更

SRND(P)

指定したデバイスに格納されているBIN16ビットデータの内容にしたがって乱数系列を変更します。

ラダー	ST
	ENO:=SRND(EN,s); ENO:=SRNDP(EN,s);

FBD/LD


■実行条件

命令	実行条件
SRND	
SRNDP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	乱数系列データ	-32768~32767	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—

機能

(s)で指定されたデバイスに格納されているBIN16ビットデータの内容に従って、乱数系列を変更します。SRND(P)命令は、乱数の算出パターンを変更することができます。

エラー

演算エラーはありません。

19 デバイス操作

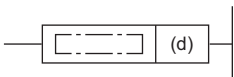
19

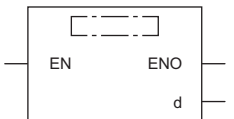
19.1 インデックスレジスタ命令

インデックスレジスタの一括退避

ZPUSH(P)

インデックスレジスタの内容を、指定した領域に退避します。

ラダー	ST
	ENO:=ZPUSH(EN,d); ENO:=ZPUSHP(EN,d);

FBD/LD


■実行条件

命令	実行条件
ZPUSH	
ZPUSHP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	インデックスレジスタを退避する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □		LT, LST, LC	LZ		K, H	E	\$	
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- インデックスレジスタ，ロングインデックスレジスタ，ローカルインデックスレジスタ，ローカルロングインデックスレジスタの内容を，(d)で指定したデバイスまたは配列型ラベルの要素以降に退避します。退避するローカルインデックスレジスタ，ローカルロングインデックスレジスタはZPUSH命令を実行したプログラムのローカルインデックスレジスタ，ローカルロングインデックスレジスタです。
- インデックスレジスタの内容を退避すると，退避回数(d)が+1されます。
- 点数の割り付けに関わらず，
- インデックスレジスタ，ロングインデックスレジスタ，ローカルインデックスレジスタ，ローカルロングインデックスレジスタを48ワード分を退避します。
- データの復帰には，ZPOP(P)命令を使用します。ZPUSH(P)命令とZPOP(P)命令は対で使用し，入れ子構造にすることでスタックとしての使用が可能です。

📖 940ページ ZPOP(P)

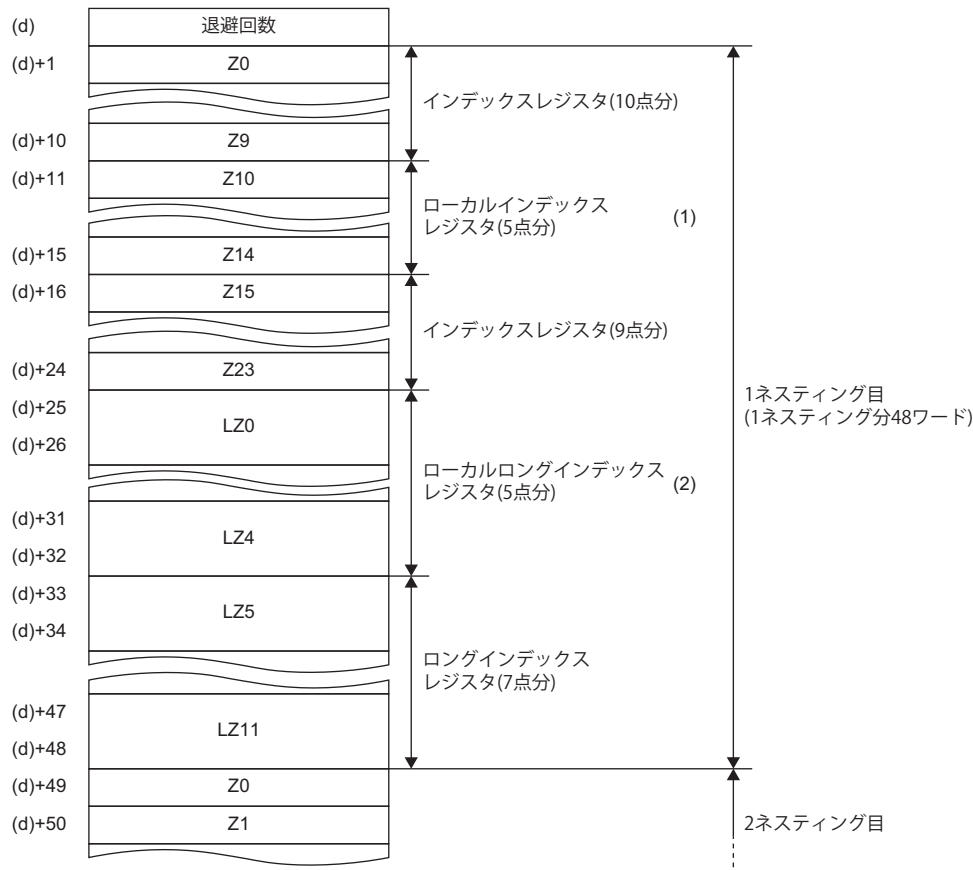
- 入れ子構造は，ZPUSH(P)命令の(d)で指定した領域を，再度ZPUSH(P)命令の(d)に指定することで作成できます。ZPUSH(P)命令を実行するごとに，退避回数(d)は+1されます。
- 入れ子構造とは別の領域をZPUSH(P)命令の(d)に指定した場合，指定した別の領域にインデックスレジスタの内容を退避します。
- 入れ子構造にした場合，ZPUSH(P)命令を実行するごとに退避したデータが追加されていくため，使用回数分の領域をあらかじめ確保しておいてください。

例

エンジニアリングツールでインデックスレジスタ設定のローカル設定を下記のように設定したとき、(d)以降に使用される領域の構成は下記のとおりです。

☞ [CPU/パラメータ]⇒[メモリ/デバイス設定]⇒[インデックスレジスタ設定]

項目			設定値
ローカル設定	点数設定	ローカルインデックスレジスタ(Z)	5点
		ローカルロングインデックスレジスタ(LZ)	5点
	先頭	インデックスレジスタ(Z)	10
		ロングインデックスレジスタ(LZ)	0



(1): Z10から5点分のローカルインデックスレジスタを退避します。

(2): LZ0から5点分のローカルロングインデックスレジスタを退避します。

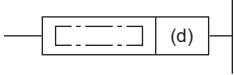
エラー

エラーコード (SD0)	内容
3285H	(d)+0の内容(退避回数)がFFFFHのとき。

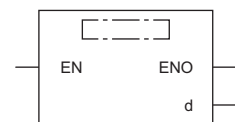
インデックスレジスタの一括復帰

ZPOP(P)

指定した領域に退避されているデータを、インデックスレジスタに読み出します。

ラダー	ST
	ENO:=ZPOP(EN,d); ENO:=ZPOPP(EN,d);

FBD/LD



■実行条件

命令	実行条件
ZPOP	
ZPOPP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	インデックスレジスタを復帰する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 0	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 0	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- (d)で指定したデバイスまたは配列型ラベルの要素以降に退避されているデータを、インデックスレジスタ、ロングインデックスレジスタ、ローカルインデックスレジスタ、ローカルロングインデックスレジスタに読み出します。
- 退避されているインデックスレジスタの内容を読み出すと、退避回数(d)が-1されます。
- 復帰するローカルインデックスレジスタ、ローカルロングインデックスレジスタはZPOP命令を実行したプログラムのローカルインデックスレジスタ、ローカルロングインデックスレジスタです。
- (d)以降に使用される領域の構成については、下記を参照してください。

☞ 937ページ ZPUSH(P)

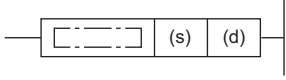
エラー

エラーコード (SD0)	内容
3285H	(d)+0の内容(退避回数)が0のとき。

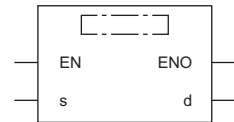
インデックスレジスタ/ロングインデックスレジスタ選択退避

ZPUSH(P)

インデックスレジスタ、ロングインデックスレジスタの内容を、指定した領域に退避します。

ラダー	ST
	ENO:=ZPUSH_2(EN,s,d); ENO:=ZPUSH_2(EN,s,d);

FBD/LD



(□には、ZPUSH_2、ZPUSH_2が入ります。)

■実行条件

命令	実行条件
ZPUSH	
ZPUSHP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	退避するインデックスレジスタ、ロングインデックスレジスタ種別	1~3	符号なしBIN16ビット	ANY16
(d)	インデックスレジスタ、ロングインデックスレジスタを退避する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

^{*1} ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口		LT, LST, LC	LZ		K, H	E	\$	
(s)	○	—	○	○ ^{*1}	○	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

^{*1} U3E口 ¥ G口のみ使用できます。

機能

- (s)で指定した退避するインデックスレジスタ、ロングインデックスレジスタ、ローカルインデックスレジスタ、ローカルロングインデックスレジスタを、(d)で指定したデバイスまたは配列型ラベルの要素以降に退避します。
- インデックスレジスタ、ロングインデックスレジスタ、ローカルインデックスレジスタ、ローカルロングインデックスレジスタの内容を退避すると退避回数(d)が+1されます。
- 退避するローカルインデックスレジスタ、ローカルロングインデックスレジスタはZPUSH命令を実行したプログラムのローカルインデックスレジスタ、ローカルロングインデックスレジスタです。
- 退避したデータの末尾に、退避したインデックスレジスタ、ロングインデックスレジスタ、ローカルインデックスレジスタ、ローカルロングインデックスレジスタの種別を格納します。
- (s)で指定する値と、退避するインデックスレジスタ、ロングインデックスレジスタ、ローカルインデックスレジスタ、ローカルロングインデックスレジスタは下記になります。

(s)の値	退避するZ, LZ
1	Z, LZ全範囲
2	Z全範囲
3	LZ全範囲

- データの復帰には、インデックスレジスタ/ロングインデックスレジスタ選択復帰命令(ZPOP(P)命令)を使用します。インデックスレジスタ/ロングインデックスレジスタ選択退避命令(ZPUSH(P)命令)とインデックスレジスタ/ロングインデックスレジスタ選択復帰命令(ZPOP(P)命令)は対で使用し、入れ子構造にすることでスタックとしての使用が可能です。

945ページ ZPOP(P)

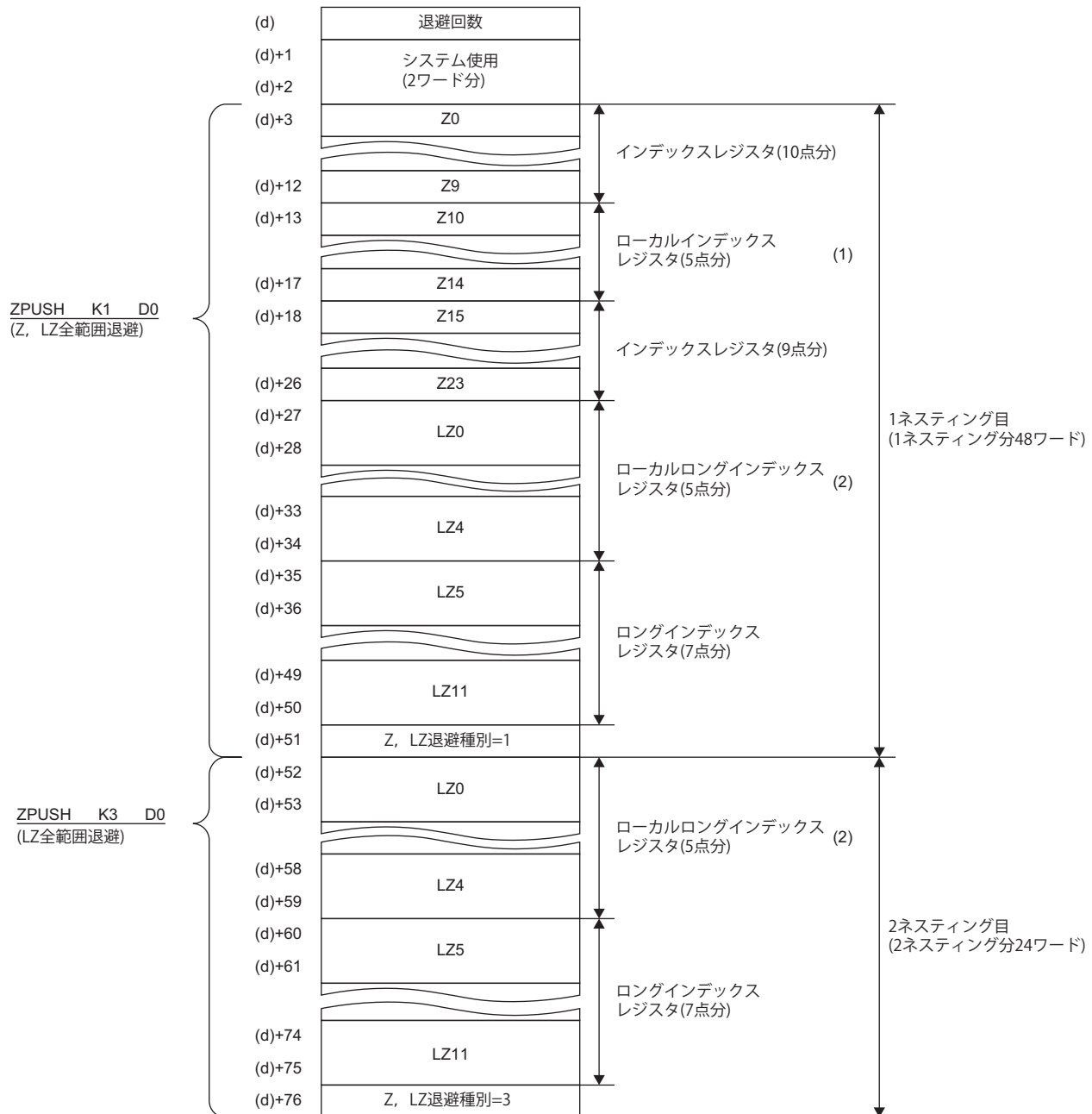
- 入れ子構造は、インデックスレジスタ/ロングインデックスレジスタ選択退避命令(ZPUSH(P)命令)の(d)で指定した領域を、再度インデックスレジスタ/ロングインデックスレジスタ選択退避命令(ZPUSH(P)命令)の(d)に指定することで作成できます。インデックスレジスタ/ロングインデックスレジスタ選択退避命令(ZPUSH(P)命令)を実行するごとに、退避回数(d)は+1されます。
- 入れ子構造とは別の領域をインデックスレジスタ/ロングインデックスレジスタ選択退避命令(ZPUSH(P)命令)の(d)に指定した場合、指定した別の領域にインデックスレジスタ、ロングインデックスレジスタの内容を退避します。
- 入れ子構造にした場合、インデックスレジスタ/ロングインデックスレジスタ選択退避命令(ZPUSH(P)命令)を実行するごとに退避したデータが追加されていくため、SD300, SD302によりインデックスレジスタ、ロングインデックスレジスタ点数を確認し、使用回数分の領域をあらかじめ確保しておいてください。

例

エンジニアリングツールでインデックスレジスタ設定のローカル設定を下記のように設定したとき、(d)以降に使用される領域の構成は下記のとおりです。

🔗 [CPU/パラメータ]⇒[メモリ/デバイス設定]⇒[インデックスレジスタ設定]

項目			設定値
ローカル設定	点数設定	ローカルインデックスレジスタ(Z)	5点
		ローカルロングインデックスレジスタ(LZ)	5点
	先頭	インデックスレジスタ(Z)	10
		ロングインデックスレジスタ(LZ)	0



(1): Z10から5点分のローカルインデックスレジスタを退避します。

(2): LZ0から5点分のローカルロングインデックスレジスタを退避します。

注意事項

ZPUSH(P)命令の(d)+1, (d)+2はシステムで使用するため、値を変更しないでください。

また、(d)で指定した領域内に格納されるZ, LZ退避種別もシステムで使用するため、値を変更しないでください。
誤動作の原因となります。

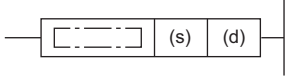
エラー

エラーコード (SD0)	内容
3285H	(s)に指定可能な範囲外のデータを入力したとき。 指定した値が1~3以外
	(d)で指定した領域のシステムが使用するエリアに格納された値が書き換えられたとき。
	退避回数(d)がFFFFHのとき。

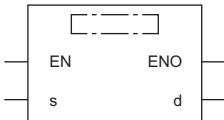
インデックスレジスタ/ロングインデックスレジスタ選択復帰

ZPOP(P)

指定した領域に退避されているデータを、インデックスレジスタ、ロングインデックスレジスタに読み出します。

ラダー	ST
	ENO:=ZPOP_2(EN,s,d); ENO:=ZPOPP_2(EN,s,d);

FBD/LD



(□には、ZPOP_2, ZPOPP_2が入ります。)

■実行条件

命令	実行条件
ZPOP	
ZPOPP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	ダミー	—	符号なしBIN16ビット	ANY16
(d)	インデックスレジスタを復帰する先頭デバイス	—	符号付きBIN16ビット	ANY16 ^{*1}
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □		LT, LST, LC	LZ		K, H	E	\$	
(s)	○	—	○	○ ^{*1}	○	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

*1 U3E□ ¥ G□のみ使用できます。

機能

- (d)で指定したデバイスまたは配列型ラベルの要素以降に退避されているデータを、インデックスレジスタ、ロングインデックスレジスタ、ローカルインデックスレジスタ、ローカルロングインデックスレジスタに読み出します。
- 退避されているインデックスレジスタ、ロングインデックスレジスタの内容を読み出すと、退避回数(d)が-1されます。
- 復帰するローカルインデックスレジスタ、ローカルロングインデックスレジスタはZPOP命令を実行したプログラムのローカルインデックスレジスタ、ローカルロングインデックスレジスタです。
- (s)で指定したデータはダミーデータ扱いとし、無視します。
- (d)以降に使用される領域の構成については、下記を参照してください。

941ページ ZPUSH(P)

注意事項

(d)+1, (d)+2はシステムで使用するため、値を変更しないでください。
また、(d)で指定した領域に格納されるZ, LZ退避種別もシステムで使用するため、値を変更しないでください。
誤動作の原因となります。

エラー

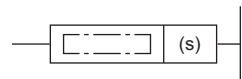
エラーコード (SD0)	内容
3285H	(d)+0の退避回数が0のとき。
	(d)で指定した領域のシステムが使用するエリアに格納された値が書き換えられたとき。
	Z, LZ退避種別に1~3以外の値が設定されているとき。

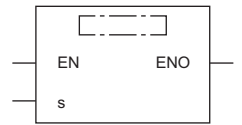
19.2 ファイルレジスタ操作命令

ファイルレジスタのブロックNo.切換え

RSET(P)

プログラム中で使用するファイルレジスタのブロックNo.を変更します。

ラダー	ST
	ENO:=RSET(EN,s); ENO:=RSETP(EN,s);

FBD/LD


■実行条件

命令	実行条件
RSET	
RSETP	

設定データ

■内容, 範囲, データ型

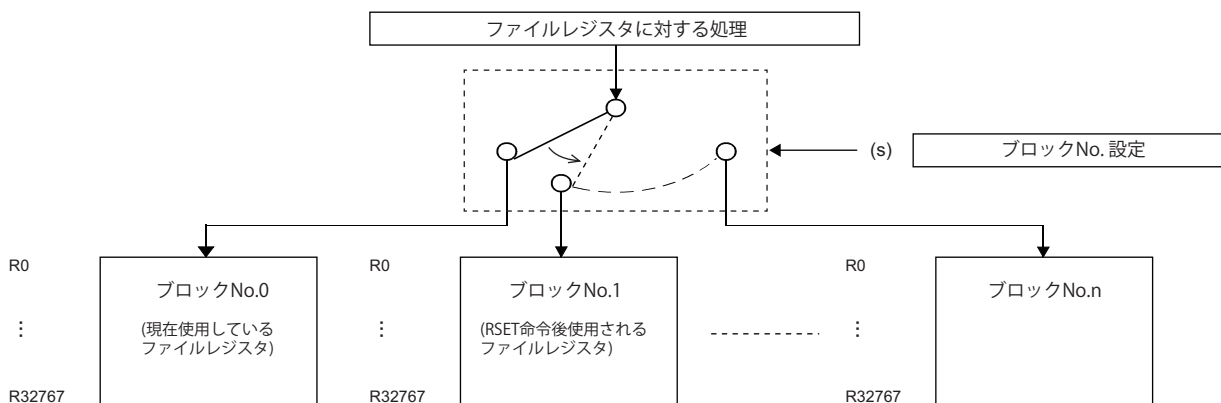
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	切り換えるブロックNo.データまたは, ブロックNo.データが格納されているデバイス番号	0~32767	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口		LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—

機能

- プログラム中で使用するファイルレジスタのブロックNo.を(s)で指定されたデバイスに格納されているブロックNo.に変更します。ブロックNo.変更後、シーケンスプログラムで使用しているすべてのファイルレジスタは、変更後のブロックNo.のファイルレジスタを対象にして処理します。



注意事項

エンジニアリングツールからファイルレジスタの"ブロックNo.退避/復帰設定"を"退避/復帰しない"に設定した場合、割込みプログラム内で本命令でファイルレジスタのブロックNo.を切り換えしないでください。切り換えた場合、割込みプログラムから復帰して即時反映されず、次のプログラム開始時までブロックNo.が切り換わらないことがあります。ファイルレジスタの制約事項については、下記を参照してください。

📖 58ページ ファイルレジスタ使用時の制約事項

エラー

エラーコード (SD0)	内容
2820H	ファイルレジスタが存在しないとき。
3285H	(s)で指定したブロックNo.が存在しないとき。

19.3 ファイルレジスタの1バイト単位のリード/ライト命令

ファイルレジスタの1バイトデータリード

ZRRDB(P)

指定した通しバイトNo.のファイルレジスタの内容を読み出します。

ラダー	ST
	ENO:=ZRRDB(EN,s,d); ENO:=ZRRDBP(EN,s,d);
FBD/LD	

■実行条件

命令	実行条件
ZRRDB	
ZRRDBP	

設定データ

■内容, 範囲, データ型

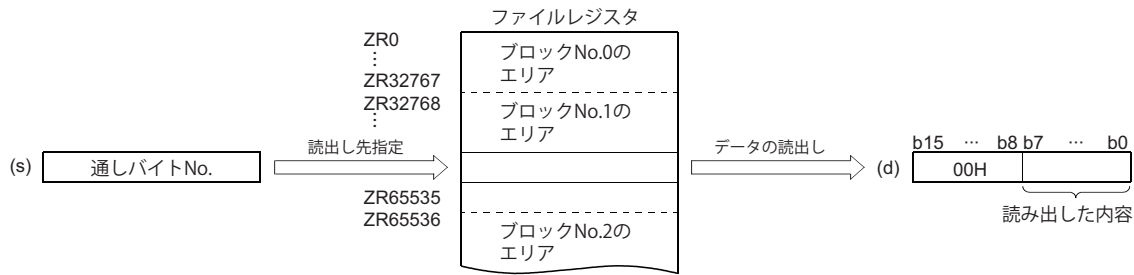
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	読み出すファイルレジスタの通しバイトNo.	0~4294967295	符号なしBIN32ビット	ANY32
(d)	読み出されたデータを格納するデバイス番号	—	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

機能

- ブロックNo.を意識することなく、(s)で指定された通しバイトNo.のファイルレジスタの内容を読み出し、(d)で指定されたデバイスの下位8ビットに格納します。(d)で指定された上位8ビットは00Hになります。



- 通しバイトNo.に対応するファイルレジスタの番号は下記のとおりです。

	b15 ... b8	b7 ... b0
ZR0	No.1	No.0
ZR1	No.3	No.2
ZR2	No.5	No.4
...	...	...
ZR2500	No.5001	No.5000
ZR2501	No.5003	No.5002
ZR2502	No.5005	No.5004
ZR2503	No.5007	No.5006

(1) points to the right column (b7...b0).
(2) points to the left column (b15...b8).

- (1): 奇数番号指定時のデータ
(2): 偶数番号指定時のデータ

例

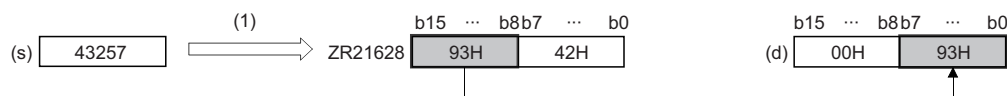
(s)=23560を指定した場合は、ZR11780の下位8ビットのデータを読み出します。



- (1): 読出し先指定

例

(s)=43257を指定した場合は、ZR21628の上位8ビットのデータを読み出します。



- (1): 読出し先指定

エラー

エラーコード (SD0)	内容
2820H	指定可能な範囲を超えたデバイス番号(通しバイトNo.)を指定したとき。

ファイルレジスタの1バイトデータライト

ZRWRB(P)

指定したデバイスの下位ビットの内容を通しバイトNo.のファイルレジスタに書き込みます。

ラダー	ST
	<pre>ENO:=ZRWRB(EN,s1,s2); ENO:=ZRWRBP(EN,s1,s2);</pre>

FBD/LD

■実行条件

命令	実行条件
ZRWRB	
ZRWRBP	

設定データ

■内容，範囲，データ型

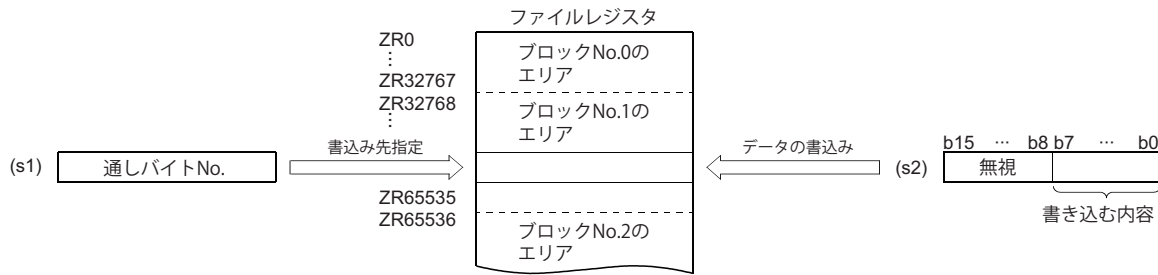
オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	書き込むファイルレジスタの通しバイトNo.	0~4294967295	符号なしBIN32ビット	ANY32
(s2)	書き込むデータが格納されているデバイス番号	-32768~32767	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	○	—	—	—

機能

- ブロックNo.を意識することなく、(s2)で指定されたデバイスに格納されている下位8ビットの内容を、(s1)で指定された通しバイトNo.のファイルレジスタに書き込みます。(s2)で指定されたデバイスの上位8ビットのデータは無視されます。



- 通しバイトNo.に対応するファイルレジスタの番号は下記のとおりです。

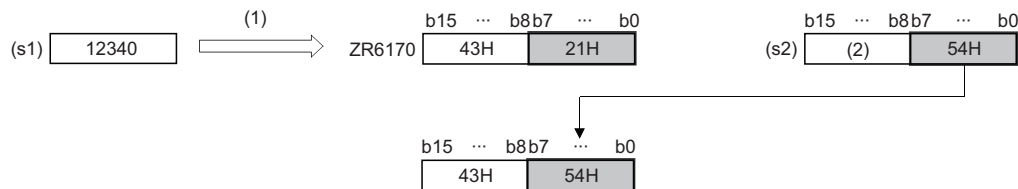
	b15 ... b8	b7 ... b0
ZR0	No.1	No.0
ZR1	No.3	No.2
ZR2	No.5	No.4
...	...	...
ZR2500	No.5001	No.5000
ZR2501	No.5003	No.5002
ZR2502	No.5005	No.5004
ZR2503	No.5007	No.5006

(1) points to the right column (b7...b0) and (2) points to the left column (b15...b8).

- (1): 奇数番号指定時のデータ
(2): 偶数番号指定時のデータ

例

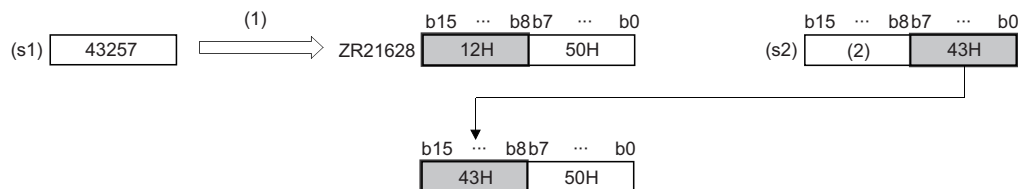
(s1)=12340が指定された場合は、ZR11170の下位8ビットに書き込みます。



- (1): 書き込み先指定
(2): 無視

例

(s1)=43257が指定された場合は、ZR21628の上位8ビットに書き込みます。



- (1): 書き込み先指定
(2): 無視

エラー

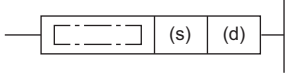
エラーコード (SD0)	内容
2820H	指定可能な範囲を超えたデバイス番号(通しバイトNo.)を指定したとき。

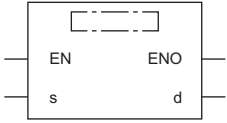
19.4 間接アドレスリード命令

間接アドレス読出し

ADRSET(P)

指定したデバイスの間接アドレスを読み出します。

ラダー	ST
	ENO:=ADRSET(EN,s,d); ENO:=ADRSETP(EN,s,d); ENO:=_ADRSET(EN,s,d);

FBD/LD


■実行条件

命令	実行条件
ADRSET	
ADRSETP	

設定データ

■内容, 範囲, データ型

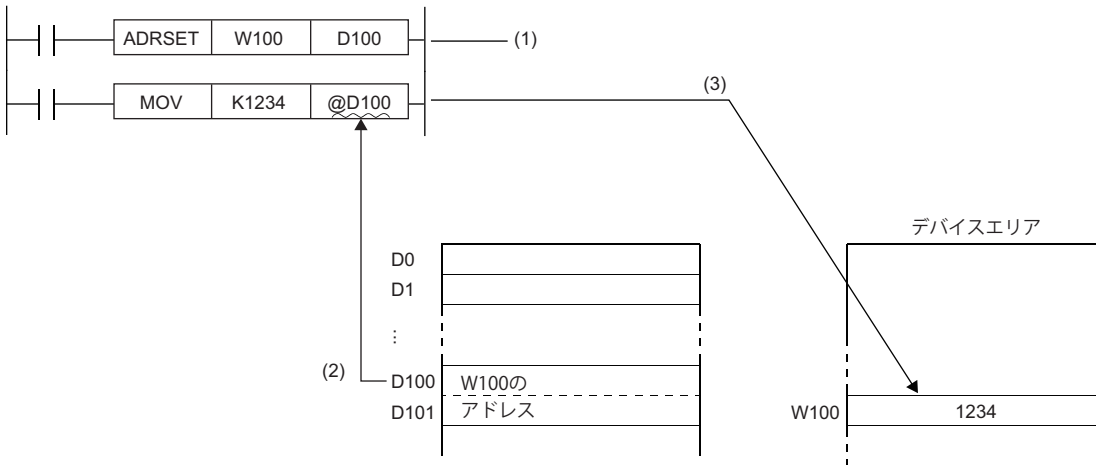
オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	間接アドレスを読み出すデバイス番号	—	デバイス名	ANY_ELEMENTARY
(d)	(s)で指定されたデバイスの間接アドレスを格納するデバイス番号	—	符号付きBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	—	○	—	—	○	—	○	—	—	—	—
(d)	○	—	○	—	—	—	—	○	—	—	—	—

機能

- (s)で指定したデバイスの間接アドレスを、(d)で指定したデバイスに格納します。(d)+0, (d)+1のデバイスに格納されたアドレスは、プログラムでデバイスの間接アドレスを行う場合に使用します。



- (1) W100のアドレスをD100, D101に格納します。
(2) D100, D101の内容(W100のアドレス)を指定します。
(3) W100に1234を書き込みます。

- (s)にはビットデバイスの桁指定、ワードデバイスのビット指定はできません。
- デバイスの間接指定については、下記を参照してください。

使用するコントローラのユーザーズマニュアル

- ラベルの間接指定には対応していません。(d)にラベルを指定して使用する場合は、下記の方法で使用してください。

ラベル	内容
デバイスを指定したグローバルラベルの場合	間接アドレスとして使用する場合は、グローバルラベルに指定したデバイスの間接指定を使用してください。 ■STの例 ADRSET(TRUE, intVar, gvAddr); // intVarの間接アドレスをgvAddrへ読み出す。 INC (TRUE, @D0); // gvAddrへ指定したデバイスD0の間接指定で使用する。
自動割付けのグローバルラベル/ローカルラベルの場合	間接アドレスをデバイスへ転送し、転送先デバイスの間接指定を使用してください。 ■STの例 ADRSET(TRUE, intVar, lvAddr); // intVarの間接アドレスをlvAddrへ読み出す。 DMOV(TRUE, lvAddr, D0); // lvAddrに読み出した間接アドレスをデバイスへ転送する。 INC (TRUE, @D0); // 転送したデバイスの間接指定で使用する。

エラー

演算エラーはありません。

20 タイマ, カウンタ

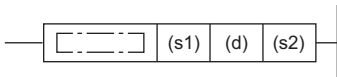
20

20.1 特殊カウンタ命令

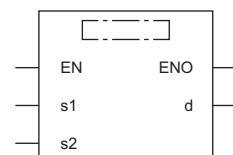
1相入力アップ/ダウンカウンタ

UDCNT1

指定したカウンタの現在値を更新します。

ラダー	ST
	ENO:=UDCNT1(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
UDCNT1	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	(s1)+0: カウント入力用の入力番号 (s1)+1: アップ/ダウン設定用 OFFの場合は, アップカウント(現在値を増加方向にカウント) ONの場合は, ダウンカウント(現在値を減少方向にカウント)	—	ビット	ANYBIT_ARRAY* ¹ (要素数: 2)
(d)	UDCNT1命令でカウントを行うカウンタ番号(デバイス名)	—	デバイス名	ANY16* ²
(s2)	設定値	-32768~32767	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 デバイス(X)に割り付けたラベルのみ使用できます。

*2 デバイス(C)に割り付けたラベルのみ使用できます。

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ	K, H		E	\$		
(s1)	○*1	—	—	—	—	—	—	—	—	—	—	—	
(d)	—	—	○*2	—	—	—	—	—	—	—	—	—	
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	

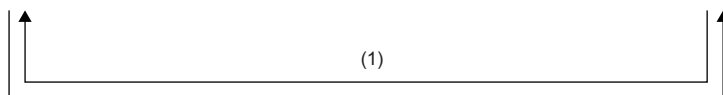
*1 Xのみ使用できます。ただし, 入出力点数(入出力ユニットとのアクセス可能点数)の範囲でのみ使用できます。

*2 Cのみ使用できます。

機能

- ・(s1)で指定された入力のOFF→ON時に、(d)で指定されたカウンタの現在値を更新します。
- ・カウント方向は、(s1)+1で指定された入力のON/OFFにより決まります。
 - ・OFF: アップカウント(現在値を増加方向にカウント)
 - ・ON: ダウンカウント(現在値を減少方向にカウント)
- ・カウント処理は、下記のように行います。
 - ・アップカウント中に、現在値が(s2)で指定された設定値と等しくなったとき、(d)で指定されたカウンタの接点がONします。ただし、(d)で指定されたカウンタの接点がONしても、現在値のカウントは継続します。
 - ・ダウンカウント中に、現在値が設定値-1のとき、(d)で指定されたカウンタの接点がOFFします。
 - ・(d)で指定されたカウンタはリングカウンタです。現在値が32767のときにアップカウントすると、現在値は-32768になります。また、現在値が-32768のときにダウンカウントすると、現在値は32767になります。現在値のカウント処理内容は下記になります。

-32768 → -32767 -----▶ -2 → -1 → 0 → 1 → 2 -----▶ 32766 → 32767



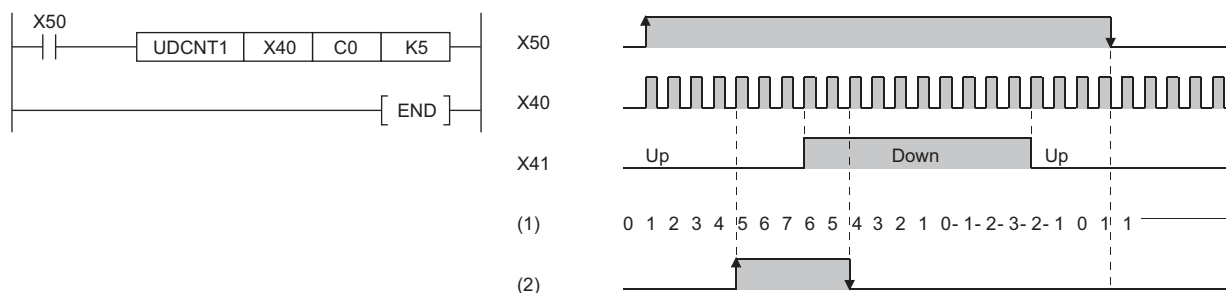
(1): アップカウント時

(2): ダウンカウント時

- ・UDCNT1命令を使用したカウント処理の動作は下記になります。

例

X50がONしてからのX40のOFF→ON回数をC0(アップ/ダウンカウンタ)によりカウントするプログラムと動作は下記のようにになります。



(1): C0現在値

(2): C0接点

- ・UDCNT1命令によるカウント処理は、実行指令のOFF→ON時にカウントを開始し、ON→OFF時にカウントを中止します。再度、実行指令をOFF→ONすると、中止時の現在値からカウントを再開します。
- ・(d)で指定されたカウンタの現在値のクリアおよび接点のOFFは、RST命令で実行します。

Point

- ・UDCNT1命令は、引数のデバイスデータをコントローラのワークエリアに登録し、実際のカウント動作はシステム割込みで処理します。(コントローラのワークエリアに登録されたデバイスデータは、実行指令をOFFまたは、STOP→RUNにてクリアします。)このためカウントできるパルスは、ONおよびOFF時間がコントローラの割込み間隔以上が必要です。コントローラの割込み間隔は1msです。
- ・UDCNT1命令によるカウント中(実行指令がON中)は、設定値の変更はできません。設定値を変更する場合は、実行指令をOFFにしてから行ってください。
- ・UDCNT1命令で指定しているカウンタは、他の命令で使用できません。他の命令で使用了場合は正常なカウントができなくなります。
- ・UDCNT1命令は、実行している全プログラムで6回まで使用できます。7回目以降は、無処理となります。

エラー

演算エラーはありません。

2相入力アップ/ダウンカウンタ

UDCNT2

指定したA相パルスとB相パルスの状態により、カウンタの現在値を更新します。

ラダー	ST
	ENO:=UDCNT2(EN,s1,s2,d);

FBD/LD

■実行条件

命令	実行条件
UDCNT2	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	(s1)+0: カウント入力用の入力番号(A相/パルス)	—	ビット	ANYBIT_ARRAY* ¹ (要素数: 2)
	(s1)+1: カウント入力用の入力番号(B相/パルス)			
(d)	UDCNT2命令でカウントを行うカウンタ番号(デバイス名)	—	デバイス名	ANY16* ²
(s2)	設定値	-32768~32767	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 デバイス(X)に割り付けたラベルのみ使用できます。

*2 デバイス(C)に割り付けたラベルのみ使用できます。

■使用可能デバイス

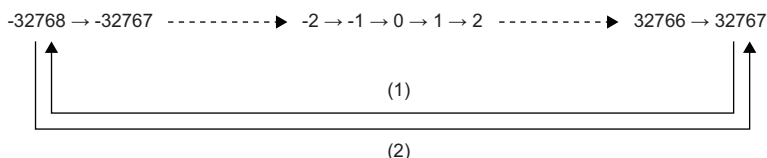
オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○* ¹	—	—	—	—	—	—	—	—	—	—	—	—
(d)	—	—	○* ²	—	—	—	—	—	—	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	○	—	—	—

*1 Xのみ使用できます。ただし、入出力点数(入出力ユニットとのアクセス可能点数)の範囲でのみ使用できます。

*2 Cのみ使用できます。

機能

- (s1)で指定された入力(A相パルス)と、(s1)+1で指定された入力(B相パルス)の状態により、(d)で指定されたカウンタの現在値を更新します。
- カウント方向は、下記のように決まっています。
 - (s1)がON時に(s1)+1がOFF→ONの場合は、アップカウント(現在値を増加方向にカウント)します。
 - (s1)がON時に(s1)+1がON→OFFの場合は、ダウンカウント(現在値を減少方向にカウント)します。
 - (s1)がOFF時にはカウントを行いません。
- カウント処理は、下記のように行います。
 - アップカウント中に、現在値が(s2)で指定された設定値と等しくなったとき、(d)で指定されたカウンタの接点がONします。ただし、(d)で指定されたカウンタの接点がONしても、現在値のカウントは継続します。
 - ダウンカウント中に、現在値が設定値-1のとき、(d)で指定されたカウンタの接点がOFFします。
 - (d)で指定されたカウンタはリングカウンタです。現在値が32767のときにアップカウントすると、現在値は-32768になります。また、現在値が-32768のときにダウンカウントすると、現在値は32767になります。現在値のカウント処理内容は下記になります。

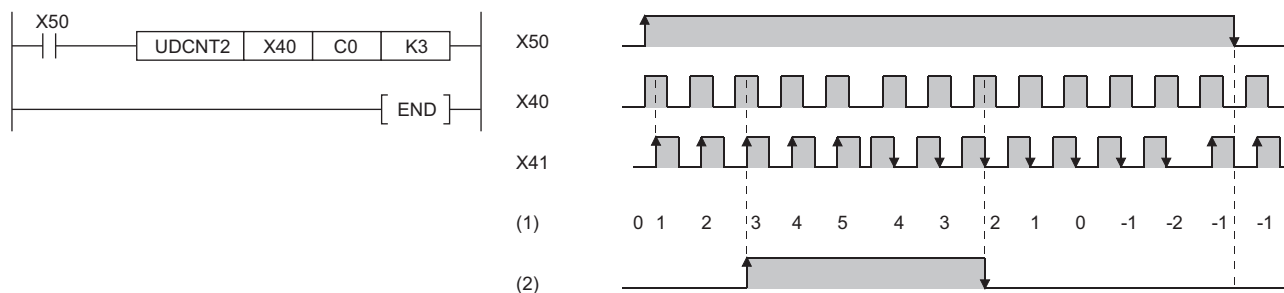


- (1): アップカウント時
(2): ダウンカウント時

- UDCNT2命令を使用したカウント処理の動作は下記になります。

例

X50がONしてからのX40, X41の状態をC0(アップ/ダウンカウンタ)によりカウントするプログラムと動作は下記のようになります。



- (1): C0現在値
(2): C0接点

- UDCNT2命令によるカウント処理は、実行指令のOFF→ON時にカウントを開始し、ON→OFF時にカウントを中止します。再度、実行指令をOFF→ONすると、中止時の現在値からカウントを再開します。
- (d)で指定されたカウンタの現在値のクリアおよび接点のOFFは、RST命令で実行します。

Point

- UDCNT2命令は、引数のデバイスデータをコントローラのワークエリアに登録し、実際のカウント動作はシステム割込みで処理します。(コントローラのワークエリアに登録されたデバイスデータは、実行指令をOFFまたは、STOP→RUNにてクリアします。)このためカウントできるパルスは、ONおよびOFF時間がコントローラの割込み間隔以上必要です。コントローラの割込み間隔は1msです。
- UDCNT2命令によるカウント中(実行指令がON中)は、設定値の変更はできません。設定値を変更する場合は、実行指令をOFFにしてから行ってください。
- UDCNT2命令で指定しているカウンタは、他の命令で使用できません。他の命令で使用了場合は正常なカウントができなくなります。
- UDCNT2命令は、実行している全プログラムで5回まで使用できます。6回目以降は、無処理となります。

エラー

演算エラーはありません。

20.2 特殊タイマ命令

ティーチングタイマ

TTMR

計測指令のON中の時間を秒単位で測定して、乗数を掛けた値を格納します。

ラダー	ST
	ENO:=TTMR(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
TTMR	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	(d)+0: 測定値格納デバイス (d)+1: コントローラのシステム用	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 2)
(s)	測定値の乗数	0~2	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(s)	—	○	○	○	○	—	—	○	○	—	—	—

■コントロールデータ

オペランド: (d)			
デバイス	内容	設定範囲	セット側
+0	測定値格納デバイス	—	システム
+1	コントローラのシステム用	—	システム

機能

- 実行指令がONしている時間を秒単位で測定し、(s)で指定された乗数を掛けた値を(d)で指定されたデバイスに格納します。
- 実行指令がOFF→ONしたとき、(d)+0, (d)+1で指定されたデバイスをクリアします。
- (s)で指定できる乗数は下記になります。

(s)	乗数
0	1
1	10
2	100

- (s)で指定された値が0~2以外の場合は、無処理となります。

Point

- TTMR命令実行時に、時間を計測します。TTMR命令をJMP命令などで飛ばさないでください。飛ばした場合は、正確な測定ができなくなります。
- TTMR命令実行中は、(s)で指定の乗数を変更しないでください。(s)で指定の乗数を変更した場合は、正確な値になりません。
- (d)+1で指定されるデバイスは、コントローラのシステムで使用するので、値を変更しないでください。値を変更した場合は、(d)で指定されたデバイスに格納された値が正確な値になりません。

エラー

演算エラーはありません。

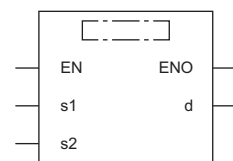
STMR

下記の4種類のタイマ出力を行います。

- ・ オフディレイタイマ出力
- ・ オフ後ワンショットタイマ出力
- ・ オン後ワンショットタイマ出力
- ・ オンディレイ+オフディレイタイマ出力

ラダー	ST
	ENO:=STMR(EN,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
STMR	常時実行

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	タイマデバイスまたはタイマ型ラベル	—	デバイス名	ANY16
(s2)	設定値	0~32767	符号付きBIN16ビット	ANY16
(d)	(d)+0: オフディレイタイマ出力 (d)+1: オフ後ワンショットタイマ出力 (d)+2: オン後ワンショットタイマ出力 (d)+3: オンディレイ+オフディレイタイマ出力	—	ビット	ANYBIT_ARRAY (要素数: 4)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数				その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□	G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$		
(s1)	—	—	○*1	—	—	—	—	—	—	—	—	—	—	—
(s2)	○	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	—	—	—	—	—	—	—	—	—	—	—	—	—

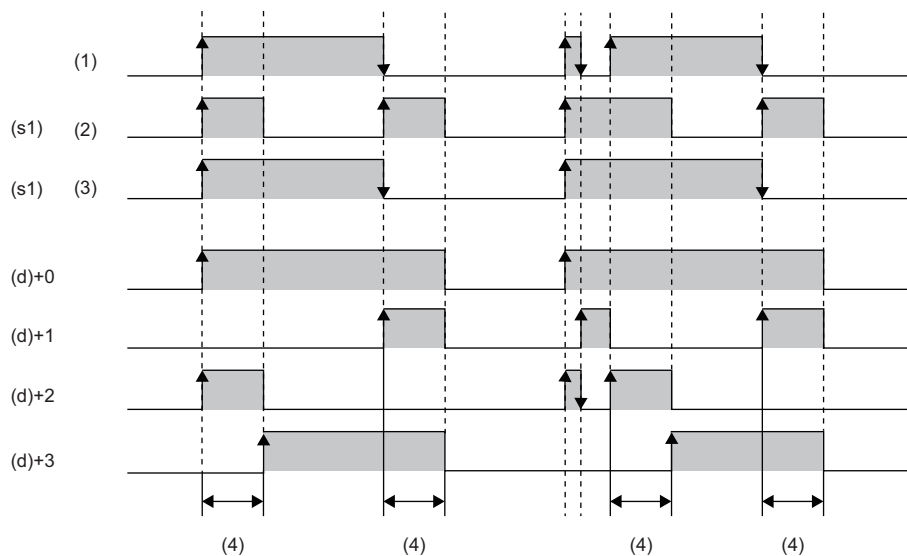
*1 Tのみ使用できます。

機能

- (d)で指定されたデバイスから4点分を使用し、4種類のタイマ出力を行います。

設定データ	内容
(d) +0	オフディレイタイマ出力: STMR命令の指令の立上りでONし、指令の立下り後(s2)で指定された時間経過するとOFFします。
+1	オフ後ワンショットタイマ出力: STMR命令の指令の立下りでONし、(s2)で指定された時間経過するとOFFします。
+2	オン後ワンショットタイマ出力: STMR命令の指令の立上りでONし、(s2)で指定された時間経過、またはSTMR命令の指令がOFFするとOFFします。
+3	オンディレイ+オフディレイタイマ出力: タイマコイルの立下りでONし、STMR命令の指令の立下り後、(s2)で指定された時間経過するとOFFします。

- (s1)で指定されたタイマのコイルは、STMR命令の指令の立上りおよび立下りでONし、現在値の計測を開始します。
- タイマのコイルは、(s2)で指定された設定値まで計測し、タイムアップするとOFFします。
- タイマのコイルは、タイムアップ前にSTMR命令の指令がOFFすると、ONのまま保持されます。このときタイマの計測は継続しています。再度STMR命令の指令がONすると、現在値を0にした後、計測再開します。
- タイマの接点は、STMR命令の指令の立上りでONし、タイマのコイルが立下りした後、STMR命令の指令の立下りでOFFします。タイマの接点はシステム用のため、使用することはできません。



- (1): STMR命令の指令
- (2): コイル
- (3): 接点
- (4): (s2)で指定の設定値

- STMR命令で指定したタイマの現在値の計測は、STMR命令の指令のON/OFF状態に関係なく実行します。JMP命令などでSTMR命令を飛ばした場合は、正常な計測ができません。
- (d)で指定されたタイマの計測単位は、低速タイマと同一です。
- (s2)の設定値は、0~32767が指定できます。0~32767以外の場合は無処理となります。
- (s1)で指定されたタイマは、OUT命令を使用しないでください。STMR命令とOUT命令で同一タイマデバイスまたはタイマ型ラベルを使用した場合は、正常な動作を行いません。

注意事項

RUN中の回路ブロック変更、またはRUN中のPC書込の書込み範囲内にSTMR命令がある場合は、STMR命令が動作します。詳細については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

エラー

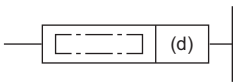
演算エラーはありません。

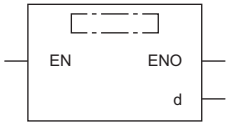
21.1 時計用命令

時計データの読出し

DATERD(P)

コントローラの時計素子より「年，月，日，時，分，秒，曜日」を読み出します。

ラダー	ST
	ENO:=DATERD(EN,d); ENO:=DATERDP(EN,d);

FBD/LD


■実行条件

命令	実行条件
DATERD	
DATERDP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	読み出した時計データを格納する先頭デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 7)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- コントローラの時計素子より「年，月，日，時，分，秒，曜日」を読み出し，(d)で指定されたデバイス以降にBIN値で格納します。

(データ)	(d)	(d)+1	(d)+2	(d)+3	(d)+4	(d)+5	(d)+6
(内容)	年(西暦)	月	日	時(24時間計)	分	秒	曜日
(設定範囲)	1980~2079	1~12	1~31	0~23	0~59	0~59	0~6

- (d)の「年」は，西暦4桁で格納します。
- (d)+6の「曜日」は，「日~土」を「0~6」で格納します。

曜日	日	月	火	水	木	金	土
格納データ	0	1	2	3	4	5	6

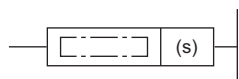
- うるう年は，自動的に補正します。

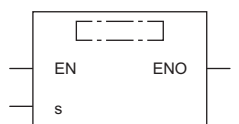
エラー

演算エラーはありません。

DATEWR(P)

指定したデバイス以降に格納している時計データを，コントローラの時計素子に書き込みます。

ラダー	ST
	ENO:=DATEWR(EN,s); ENO:=DATEWRP(EN,s);

FBD/LD


■実行条件

命令	実行条件
DATEWR	
DATEWRP	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	時計素子に書き込む時計データを格納している先頭デバイス	機能参照	符号付きBIN16ビット	ANY16_ARRAY (要素数: 7)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	○	—	—	—	—

機能

- (s)で指定したデバイス以降に格納している時計データを、コントローラの時計素子に書き込みます。

(データ)	(d)	(d)+1	(d)+2	(d)+3	(d)+4	(d)+5	(d)+6
(内容)	年(西暦)	月	日	時(24時間計)	分	秒	曜日
(設定範囲)	1980~2079	1~12	1~31	0~23	0~59	0~59	0~6

- 各項目の設定はBIN値で設定します。
- (s)の「年」は、西暦4桁を1980~2079で設定します。
- (s)+1の「月」は、1~12(1月~12月)で設定します。
- (s)+2の「日」は、1~31(1日~31日)で設定します。
- (s)+3の「時」は、0~23(0時~23時)で設定します。(24時間計として設定します。)
- (s)+4の「分」は、0~59(0分~59分)で設定します。
- (s)+5の「秒」は、0~59(0秒~59秒)で設定します。
- (s)+6の「曜日」は、「日~土」を「0~6」で設定します。

曜日	日	月	火	水	木	金	土
格納データ	0	1	2	3	4	5	6

エラー

エラーコード (SD0)	内容
3285H	(s)~(s)+6の各項目に設定範囲外のデータを設定したとき。
3295H	(s)~(s)+6に夏時間開始時刻から1時間未満を設定したとき。

Point

時計データを変更すると、イベント履歴に“時計設定”(イベントコード: 24000)が保存されます。したがって、本命令を実行した場合、イベント履歴に“時計設定”が保存されます。

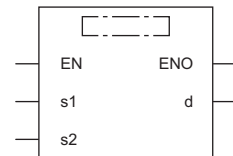
時計データの加算

DATE+(P)

時間データを加算します。

ラダー	ST
	ENO:=DATEPLUS(EN,s1,s2,d); ENO:=DATEPLUSP(EN,s1,s2,d);

FBD/LD



(□には、DATEPLUS、DATEPLUSPが入ります。)

■実行条件

命令	実行条件
DATE+	
DATE+P	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	加算される時刻データが格納されている先頭デバイス	機能参照	符号付きBIN16ビット	ANY16_ARRAY (要素数: 3)
(s2)	加算時間(時刻)データが格納されている先頭デバイス	機能参照	符号付きBIN16ビット	ANY16_ARRAY (要素数: 3)
(d)	加算結果時刻(時間)データを格納する先頭デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 3)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—

機能

- (s1)で指定された時刻データに、(s2)で指定された時間データを加算し、加算結果を(d)で指定されたデバイス番号以降に格納します。

(s1)	hour	(0...23)		(s2)	hour	(0...23)		(d)	hour	(0...23)
(s1)+1	minute	(0...59)	+	(s2)+1	minute	(0...59)	⇒	(d)+1	minute	(0...59)
(s1)+2	second	(0...59)		(s2)+2	second	(0...59)		(d)+2	second	(0...59)

例

6時32分40秒に7時48分10秒を加算した場合

(s1)	6		(s2)	7		(d)	14
(s1)+1	32	+	(s2)+1	48	⇒	(d)+1	20
(s1)+2	40		(s2)+2	10		(d)+2	50

- 演算結果の時間が24時を超えた場合は、24時間を引いた値が演算結果となります。たとえば、14時20分30秒に20時20分20秒を加算した場合、34時40分50秒ではなく、10時40分50秒になります。

(s1)	14		(s2)	20		(d)	10
(s1)+1	20	+	(s2)+1	20	⇒	(d)+1	40
(s1)+2	30		(s2)+2	20		(d)+2	50

エラー

エラーコード (SD0)	内容
3285H	(s1), (s2)のデータが範囲外の時。

時計データの減算

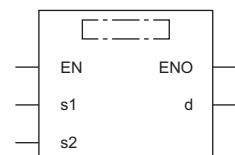
21

DATE-(P)

時間データを減算します。

ラダー	ST
	ENO:=DATEMINUS(EN,s1,s2,d); ENO:=DATEMINUSP(EN,s1,s2,d);

FBD/LD



(□には、DATEMINUS、DATEMINUSPが入ります。)

■実行条件

命令	実行条件
DATE-	
DATE-P	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	減算される時刻データが格納されている先頭デバイス	機能参照	符号付きBIN16ビット	ANY16_ARRAY (要素数: 3)
(s2)	減算時間(時刻)データが格納されている先頭デバイス	機能参照	符号付きBIN16ビット	ANY16_ARRAY (要素数: 3)
(d)	減算結果時刻(時間)データを格納する先頭デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 3)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- (s1)で指定された時刻データから，(s2)で指定された時間データを減算し，減算結果を(d)で指定されたデバイス番号以降に格納します。

(s1)	hour	(0...23)		(s2)	hour	(0...23)		(d)	hour	(0...23)
(s1)+1	minute	(0...59)	-	(s2)+1	minute	(0...59)	⇒	(d)+1	minute	(0...59)
(s1)+2	second	(0...59)		(s2)+2	second	(0...59)		(d)+2	second	(0...59)

例

10時40分20秒から3時50分10秒を減算した場合

(s1)	10		(s2)	3		(d)	6
(s1)+1	40	-	(s2)+1	50	⇒	(d)+1	50
(s1)+2	20		(s2)+2	10		(d)+2	10

- 演算結果の時間が負の数になった場合は，そのデータに+24した値が演算結果となります。たとえば，4時50分32秒から10時間42分12秒を減算した場合，-6時8分20秒ではなく，18時8分20秒になります。

(s1)	4		(s2)	10		(d)	18
(s1)+1	50	-	(s2)+1	42	⇒	(d)+1	8
(s1)+2	32		(s2)+2	12		(d)+2	20

エラー

エラーコード (SD0)	内容
3285H	(s1)，(s2)のデータが範囲外の時。

時間データの変換(時分秒→秒)

TIME2SEC(P)

時間データ(時, 分, 秒)を秒のデータに換算します。

ラダー	ST
	ENO:=TIME2SEC(EN,s,d); ENO:=TIME2SECP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
TIME2SEC	
TIME2SECP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	変換前の時計データが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 3)
(d)	変換後の時計データを格納する先頭デバイス	—	符号付きBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

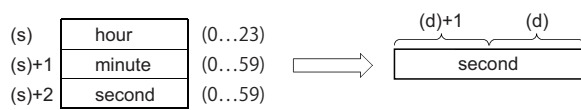
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	○*1	○	○	○	○	○	○	○	—	—	—	—

*1 FX, FYは使用できません。

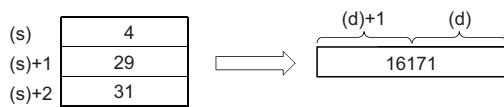
機能

- (s)で指定されたデバイス番号以降に格納されている時間データを，秒に換算して換算結果を(d)で指定されたデバイスに格納します。



例

(s)に4時間29分31秒が指定された場合



エラー

エラーコード (SD0)	内容
3285H	(s)のデータが範囲外の時。

時間データの変換(秒→時分秒)

SEC2TIME(P)

秒のデータを時間データ(時, 分, 秒)に換算します。

ラダー	ST
	ENO:=SEC2TIME(EN,s,d); ENO:=SEC2TIMEP(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
SEC2TIME	
SEC2TIMEP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	変換前の時計データが格納されている先頭デバイス	0~86399	符号付きBIN32ビット	ANY32
(d)	変換後の時計データを格納する先頭デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 3)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

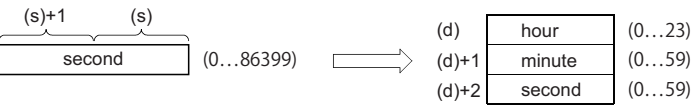
■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□		LT, LST, LC	LZ		K, H	E	\$	
(s)	○*1	○	○	○	○	○	○	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

*1 FX, FYは使用できません。

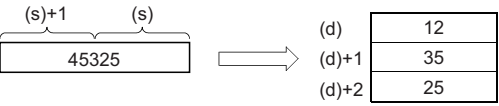
機能

- (s)で指定されたデバイス番号以降に格納されている秒のデータを、時、分、秒に換算して、換算結果を(d)で指定されたデバイス以降に格納します。



例

(s)に45325秒が指定された場合



エラー

エラーコード (SD0)	内容
3285H	(s)のデータが範囲外の時。

日時データの変換(日時→秒)

DATE2SEC(P)(_U)

日時データを秒に換算します。

ラダー	ST
	ENO:=DATE2SEC(EN,s,d); ENO:=DATE2SECP(EN,s,d); ENO:=DATE2SEC_U(EN,s,d); ENO:=DATE2SECP_U(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
DATE2SEC DATE2SEC_U	
DATE2SECP DATE2SECP_U	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	変更前の日時データを格納している先頭デバイス	機能詳細を参照。	符号付きBIN16ビット	ANY16_ARRAY (要素数:6)
(d)	DATE2SEC(P)	0~2145916799	符号付きBIN32ビット	ANY32
	DATE2SEC(P)_U	0~3155759999	符号なしBIN32ビット	ANY32_U
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

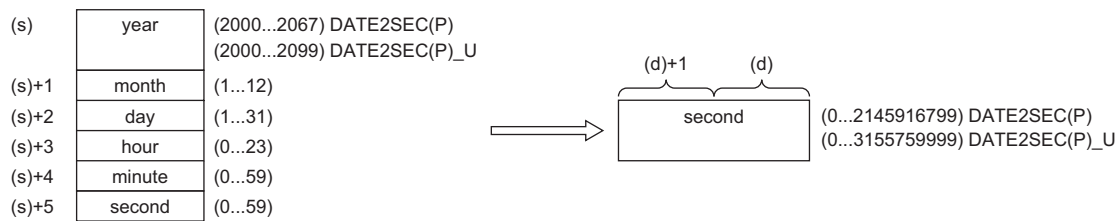
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	○*1	○	○	○	○	○	○	○	—	—	—	—	—

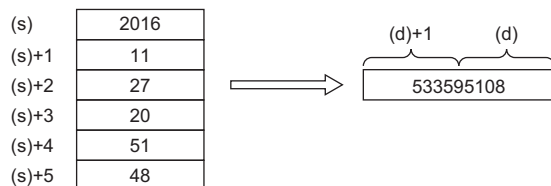
*1 FX, FYは使用できません。

機能

- (s)で指定したデバイス番号以降に格納している日時データを秒に換算して、換算結果が(d)で指定したデバイスに格納されます。2000年1月1日0時0分0秒が起点(0秒)に相当します。



- 年のデータ範囲は、DATE2SEC(P)では2000~2067、DATE2SEC(P)_Uのときは2000~2099となります。たとえば、2016年11月27日20時51分48秒を指定した場合は、次のようになります。



- 本命令の動作は時計データの変更(タイムゾーン、夏時間による時刻補正含む)には影響されません。たとえば、夏時間期間中の換算結果と、夏時間期間外の換算結果は同じになります。

エラー

エラーコード (SD0)	内容
3285H	(s)のデータが範囲外の時。 実際に存在しない日時を指定したとき。(例:2016年2月30日)

日時データの変換(秒→日時)

SEC2DATE(P)(_U)

秒のデータを日時に換算します。

ラダー	ST
	ENO:=SEC2DATE(EN,s,d); ENO:=SEC2DATEP(EN,s,d); ENO:=SEC2DATE_U(EN,s,d); ENO:=SEC2DATEP_U(EN,s,d);

FBD/LD

■実行条件

命令	実行条件
SEC2DATE SEC2DATE_U	
SEC2DATEP SEC2DATEP_U	

設定データ

■内容, 範囲, データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(s)	SEC2DATE(P)	変換前の秒データを格納している先頭デバイス	0~2145916799	符号付きBIN32ビット	ANY32
	SEC2DATE(P)_U		0~3155759999	符号なしBIN32ビット	ANY32_U
(d)		変換後の日時データが格納される先頭デバイス	機能詳細を参照。	符号付きBIN16ビット	ANY16_ARRAY (要素数:7)
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○*1	○	○	○	○	○	○	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

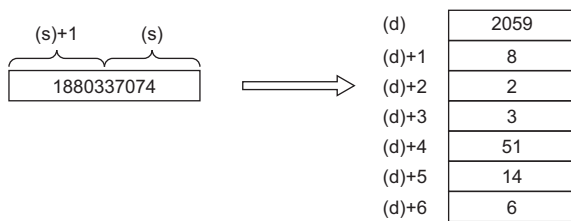
*1 FX, FYは使用できません。

機能

- (s)で指定したデバイス番号以降に格納している秒データを日時に換算して、換算結果が(d)で指定したデバイス以降に格納されます。2000年1月1日0時0分0秒が起点(0秒)に相当します。



たとえば、1880337074秒を指定した場合は、次のようになります。



- (d)+6の曜日は、"日~土"が"0~6"で格納されます。

曜日	日	月	火	水	木	金	土
格納データ	0	1	2	3	4	5	6

- 本命令の動作は時計データの変更(タイムゾーン、夏時間による時刻補正含む)には影響されません。たとえば、夏時間期間中の換算結果と、夏時間期間外の換算結果は同じになります。

エラー

エラーコード (SD0)	内容
3285H	(s)のデータが範囲外の時。

日付比較

LDDT□, ANDDT□, ORDT□

指定した日付データの比較, または日付データと現在日付を比較します。

ラダー	ST
	ENO:=LDDT_□(EN,s1,s2,s3); ENO:=ANDDT_□(EN,s1,s2,s3); ENO:=ORDT_□(EN,s1,s2,s3); (□には, EQ, NE, GT, LE, LT, GEが入ります。)*1
(□には, DT=, DT<>, DT>, DT<=, DT<, DT>=が入ります。)	

FBD/LD
(□には, LDDT_, ANDDT_, ORDT_と, EQ, NE, GT, LE, LT, GEの組合せが入ります。)*1

*1 EQは=, NEは<>, GTは>, LEは<=, LTは<, GEは>=です。

■実行条件

命令	実行条件
LDDT□, ANDDT□, ORDT□	常時実行

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	比較されるデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY_DT*1
(s2)	比較されるデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY_DT*1
(s3)	比較対象を示す値または, 比較対象が格納されたデータ数	0001H~0007H, 8001H~8007H	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は, 動作に必要な領域が確保できるよう配列を定義し, その配列型ラベルの要素を指定してください。

■使用可能デバイス

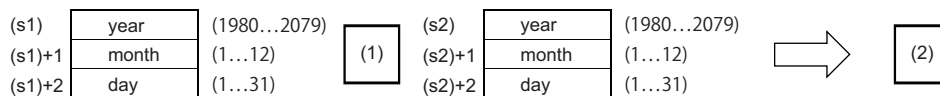
オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—
(s3)	○	○	○	○	○	—	○	○	—	—	—

機能

- ・(s1), (s2)で指定した日付データ、または、(s1)で指定した日付データと現在の日付を比較します。(s3)により比較対象を選択することができます。

- ・任意の日付データとの比較

(s1)で指定した日付データと、(s2)で指定した日付データとの比較を(s3)の条件に従ってa接点扱いで行います。

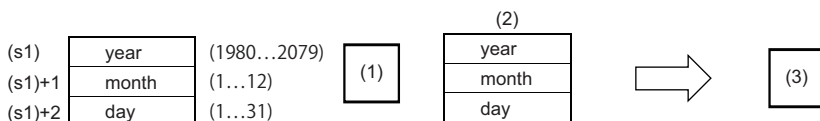


(1)比較演算子

(2)比較演算結果

- ・現在の日付データとの比較

(s1)で指定した日付データと、現在の日付データとの比較を(s3)の条件に従ってa接点扱いで行います。(s2)で指定した日付データはダミーデータ扱いとし、無視します。



(1)比較演算子

(2)現在の日付データ

(3)比較演算結果

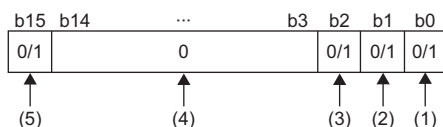
- ・各項目の設定はBIN値で設定します。

- ・(s1), (s2)の「年」は、西暦4桁を1980~2079で設定します。

- ・(s1)+1, (s2)+1の「月」は、1~12(1月~12月)で設定します。

- ・(s1)+2, (s2)+2の「日」は、1~31(1日~31日)で設定します。

- ・(s3)に下記の値を指定することにより、比較対象を詳細に設定することができます。(s3)のビット構成は下記になります。



(1) 比較対象に「日」を追加します。

(2) 比較対象に「月」を追加します。

(3) 比較対象に「年」を追加します。

(4) 0を指定します。0以外を指定すると、演算結果に関係なく非導通になります。

(5) 15ビット目をON(1)すると、(s1)と現在の日付を0~2ビット目で指定した条件に従って比較します。

- ・比較対象日付(0~2ビット目)が0の場合は、比較対象の日付データ(年、月、日)の比較は行いません。1の場合は、比較対象の日付データ(年、月、日)について比較します。

- ・比較演算対象(15ビット目)が0の場合は、(s1)で指定した日付データと、(s2)で指定した日付データとの比較を行います。1の場合は、(s1)で指定した日付データと、現在の日付データとの比較を行います。(s2)で指定された日付データは無視します。

- ・比較対象ビットの処理内容は下記になります。

任意の日付データとの比較時の(s3)値	現在の日付データとの比較時の(s3)値	比較対象日付	処理内容
0001H	8001H	日	(s1)+2のみを比較します。
0002H	8002H	月	(s1)+1のみを比較します。
0003H	8003H	月、日	(s1)+1, (s1)+2を比較します。
0004H	8004H	年	(s1)のみを比較します。
0005H	8005H	年、日	(s1), (s1)+2を比較します。
0006H	8006H	年、月	(s1), (s1)+1を比較します。
0007H	8007H	年、月、日	(s1), (s1)+1, (s1)+2すべてを比較します。
0001H~0007H, 8001H~8007H以外		なし	(s1), (s1)+1, (s1)+2すべてを比較しません。(非導通になります。)

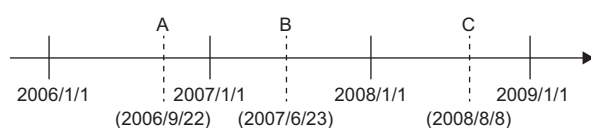
- ・比較対象となるデバイスに格納されたデータが日付データとして認識できない場合は、命令実行後にSM709(DT, TM命令不正データ検出フラグ)がONし、非導通になります。日付データとして認識できない場合でも、設定範囲内であればSM709はONしません。(s1)~(s1)+2, または(s2)~(s2)+2が該当デバイス/ラベルの割付範囲を超えている場合も、SM709がONし、非導通になります。SM709は、一度ONするとリセットおよび電源OFF時までON状態を保持するので、必要に応じてOFFしてください。

- 各命令の比較演算結果は下記になります。

命令記号(ラダー, FBD/LD)	条件	比較演算結果
DT=, EQ	(s1)=(s2)	導通状態(ENOはON)
DT<>, NE	(s1)≠(s2)	
DT>, GT	(s1)>(s2)	
DT<=, LE	(s1)≤(s2)	
DT<, LT	(s1)<(s2)	
DT>=, GE	(s1)≥(s2)	
DT=, EQ	(s1)≠(s2)	非導通状態(ENOはOFF)
DT<>, NE	(s1)=(s2)	
DT>, GT	(s1)≤(s2)	
DT<=, LE	(s1)>(s2)	
DT<, LT	(s1)≥(s2)	
DT>=, GE	(s1)<(s2)	

例

A, B, Cそれぞれの日付の比較を示します。



- 上記に示した日付A, B, Cの比較演算結果は下記のようにになります。同じ条件で比較した場合でも、選択した比較対象によって比較演算結果が変わります。

比較対象	比較条件*1		
	A<B	B<C	A<C
日	導通	非導通	非導通
月	非導通	導通	非導通
月, 日	非導通	導通	非導通
年	導通	導通	導通
年, 日	導通	導通	導通
年, 月	導通	導通	導通
年, 月, 日	導通	導通	導通
なし	非導通	非導通	非導通

*1 FBD/LDの場合、導通はENOがON、非導通はENOがOFFを示します。

- 比較する日付が実在しない場合でも、設定範囲内の日付であれば下記の条件に従って比較演算します。

- 日付A: 2006/02/30(実在しませんが、日付の設定が可能です。)
- 日付B: 2007/03/29
- 日付C: 2008/02/31(実在しませんが、日付の設定が可能です。)

比較対象	比較条件*2		
	A<B	B<C	A<C
日	非導通	導通	導通
月	導通	非導通	非導通
月, 日	導通	非導通	導通
年	導通	導通	導通
年, 日	導通	導通	導通
年, 月	導通	導通	導通
年, 月, 日	導通	導通	導通
なし	非導通	非導通	非導通

*2 FBD/LDの場合、導通はENOがON、非導通はENOがOFFを示します。

- LDDT_□命令をFBD/LDで使用した場合、ENに左母線、または常時ONの変数部品/定数部品を指定してください。
- ORDT_□命令をFBD/LDで使用した場合、ENにTRUEを指定するとENOはONします。ENは実行条件にはなりません。

エラー

演算エラーはありません。

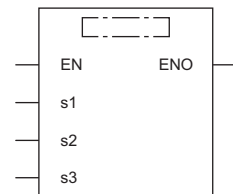
時刻比較

LDTM□, ANDTM□, ORTM□

指定した時刻データの比較, または指定した時刻データと現在時刻を比較します。

ラダー	ST
	<pre> ENO:=LDTM_□(EN,s1,s2,s3); ENO:=ANDTM_□(EN,s1,s2,s3); ENO:=ORTM_□(EN,s1,s2,s3); (□には, EQ, NE, GT, LE, LT, GEが入ります。)*1 </pre>
(□には, TM=, TM<>, TM>, TM<=, TM<, TM>=が入ります。)	

FBD/LD



(□には, LDTM_, ANDTM_, ORTM_と, EQ, NE, GT, LE, LT, GEの組合せが入ります。)*1

*1 EQは=, NEは<>, GTは>, LEは<=, LTは<, GEは>=です。

■実行条件

命令	実行条件
LDTM□, ANDTM□, ORTM□	常時実行

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	比較されるデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY_TM*1
(s2)	比較されるデータが格納されている先頭デバイス	—	符号付きBIN16ビット	ANY_TM*1
(s3)	比較対象を示す値または, 比較対象が格納されたデータ数	0001H~0007H, 8001H~8007H	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

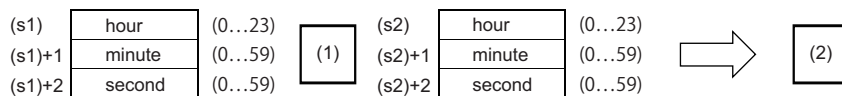
*1 ラベルで設定する場合は, 動作に必要な領域が確保できるよう配列を定義し, その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	○	—	—	—	—
(s3)	○	○	○	○	○	—	○	○	—	—	—

機能

- (s1), (s2)で指定した時刻データ、または、(s1)で指定した時刻データと現在の時刻を比較します。(s3)により比較対象を選択することができます。
- 任意の時刻データとの比較
- (s1)で指定した時刻データと、(s2)で指定した時刻データとの比較を(s3)の条件に従ってa接点扱いで行います。

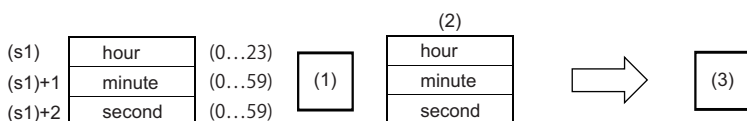


(1)比較演算子

(2)比較演算結果

- 現在の時刻データとの比較

(s1)で指定した時刻データと、現在の時刻データとの比較を(s3)の条件に従ってa接点扱いで行います。(s2)で指定した時刻データはダミーデータ扱いとし、無視します。

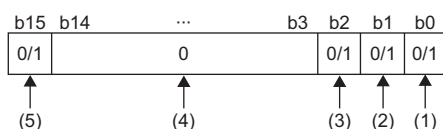


(1)比較演算子

(2)現在の時刻データ

(3)比較演算結果

- 各項目の設定はBIN値で設定します。
- (s1), (s2)の「時」は、0~23(0時~23時)で24時間計として設定します。
- (s1)+1, (s2)+1の「分」は、0~59(0分~59分)で設定します。
- (s1)+2, (s2)+2の「秒」は、0~59(0秒~59秒)で設定します。
- (s3)に下記の値を指定することにより、比較対象を詳細に設定することができます。(s3)のビット構成は下記になります。



- (1) 比較対象に「秒」を追加します。
- (2) 比較対象に「分」を追加します。
- (3) 比較対象に「時」を追加します。
- (4) 0を指定します。0以外を指定すると、演算結果に関係なく非導通になります。
- (5) 15ビット目をON(1)すると、(s1)と現在の時刻を0~2ビット目で指定した条件に従って比較します。

- 比較対象時刻(0~2ビット目)が0の場合は、比較対象の時刻データ(時、分、秒)の比較は行いません。1の場合は、比較対象の時刻データ(時、分、秒)について比較します。
- 比較演算対象(15ビット目)が0の場合は、(s1)で指定した時刻データと、(s2)で指定した時刻データとの比較を行います。1の場合は、(s1)で指定した時刻データと、現在の時刻データとの比較を行います。(s2)で指定された時刻データは無視します。
- 比較対象ビットの処理内容は下記になります。

任意の時刻データとの比較時の(s3)値	現在の時刻データとの比較時の(s3)値	比較対象時刻	処理内容
0001H	8001H	秒	(s1)+2のみを比較します。
0002H	8002H	分	(s1)+1のみを比較します。
0003H	8003H	分、秒	(s1)+1, (s1)+2を比較します。
0004H	8004H	時	(s1)のみを比較します。
0005H	8005H	時、秒	(s1), (s1)+2を比較します。
0006H	8006H	時、分	(s1), (s1)+1を比較します。
0007H	8007H	時、分、秒	(s1), (s1)+1, (s1)+2すべてを比較します。
0001H~0007H, 8001H~8007H以外		なし	(s1), (s1)+1, (s1)+2すべてを比較しません。(非導通になります。)

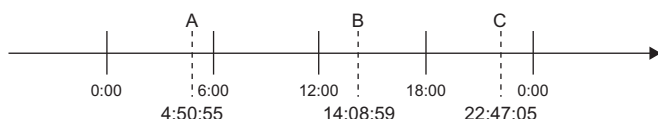
- 比較対象となるデバイスに格納されたデータが時刻データとして認識できない場合は、命令実行後にSM709(DT, TM命令不正データ検出フラグ)がONし、非導通(ENOはOFF)になります。(s1)~(s1)+2, または(s2)~(s2)+2が該当デバイス/ラベルの割付範囲を超えている場合も、SM709がONし、非導通(ENOはOFF)になります。SM709は、一度ONするとリセット/電源OFF時までON状態を保持するので、必要に応じOFFしてください。

- 各命令の比較演算結果は下記になります。

命令記号(ラダー, FBD/LD)	条件	比較演算結果
TM=, EQ	(s1)=(s2)	導通状態(ENOはON)
TM<>, NE	(s1)≠(s2)	
TM>, GT	(s1)>(s2)	
TM<=, LE	(s1)≤(s2)	
TM<, LT	(s1)<(s2)	
TM>=, GE	(s1)≥(s2)	
TM=, EQ	(s1)≠(s2)	非導通状態(ENOはOFF)
TM<>, NE	(s1)=(s2)	
TM>, GT	(s1)≤(s2)	
TM<=, LE	(s1)>(s2)	
TM<, LT	(s1)≥(s2)	
TM>=, GE	(s1)<(s2)	

例

A, B, Cそれぞれの時刻の比較を示します。



- 上記に示した時刻A, B, Cの比較演算結果は下記のようにになります。同じ条件で比較した場合でも、選択した比較対象によって比較演算結果が変わります。

比較対象	比較条件*1		
	A<B	B<C	A<C
秒	導通	非導通	非導通
分	非導通	導通	非導通
分, 秒	非導通	導通	非導通
時	導通	導通	導通
時, 秒	導通	導通	導通
時, 分	導通	導通	導通
時, 分, 秒	導通	導通	導通
なし	非導通	非導通	非導通

*1 FBD/LDの場合、導通はENOがON、非導通はENOがOFFを示します。

- LDTM_□命令をFBD/LDで使用了した場合、ENに左母線、または常時ONの変数部品/定数部品を指定してください。
- ORTM_□命令をFBD/LDで使用了した場合、ENにTRUEを指定するとENOはONします。ENは実行条件にはなりません。

エラー

演算エラーはありません。

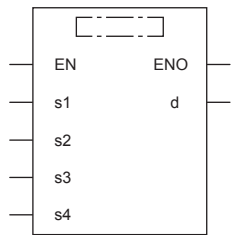
時刻比較出力

TCMP(P)

(s1), (s2), (s3)で指定した比較する時刻データと(s4)で指定した時刻データを比較し, その結果(小, 一致, 大)に応じて(d), (d)+1, (d)+2のいずれかがONします。

ラダー	ST
	ENO:=TCMP(EN,s1,s2,s3,s4,d); ENO:=TCMPP(EN,s1,s2,s3,s4,d);

FBD/LD



■実行条件

命令	実行条件
TCMP	
TCMPP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	比較する時刻データ(時)を格納する先頭デバイス	0~23	符号付きBIN16ビット	ANY16
(s2)	比較する時刻データ(分)を格納する先頭デバイス	0~59	符号付きBIN16ビット	ANY16
(s3)	比較する時刻データ(秒)を格納する先頭デバイス	0~59	符号付きBIN16ビット	ANY16
(s4)	比較する時刻データ(時, 分, 秒)を格納する先頭デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 3)
(d)	比較結果が格納される先頭デバイス	—	ビット	ANYBIT_ARRAY (要素数: 3)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

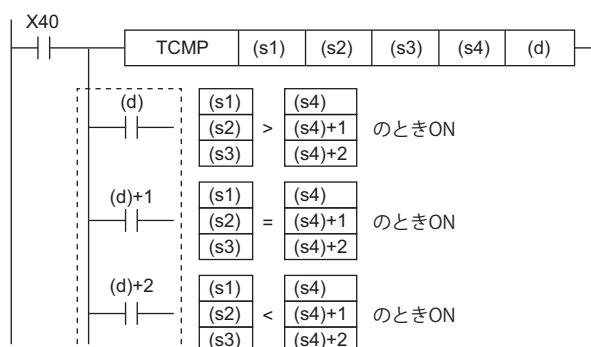
オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	○*1	○	○	○	○	—	—	○	○	—	—	—
(s2)	○*1	○	○	○	○	—	—	○	○	—	—	—
(s3)	○*1	○	○	○	○	—	—	○	○	—	—	—
(s4)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	○	—	○*2	—	—	—	—	—	—	—	—	—

*1 FX, FYは使用できません。

*2 T, ST, Cは使用できません。

機能

- (s1), (s2), (s3)で指定した比較する時刻データと(s4)で指定した時刻データを比較し、その結果(小, 一致, 大)に応じて (d), (d)+1, (d)+2のいずれかがONします。



• (s1): 時, (s2): 分, (s3): 秒

• (s4): 時, (s4)+1: 分, (s4)+2: 秒

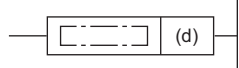
エラー

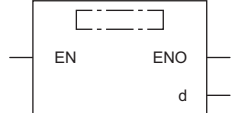
エラーコード (SD0)	内容
3285H	(s1), (s4)で指定した値が下記範囲外の時。 0~23
	(s2), (s3), (s4)+1, (s4)+2で指定した値が下記範囲外の時。 0~59

拡張時計データの読出し

S(P).DATERD

コントローラ内部の時計素子から1/1000秒を含む時計データを読み出します。

ラダー	ST
	ENO:=S_DATERD(EN,d); ENO:=SP_DATERD(EN,d);

FBD/LD


■実行条件

命令	実行条件
S.DATERD	
SP.DATERD	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(d)	読み出した時計データを格納する先頭デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 8)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(d)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- コントローラの時計素子より「年，月，日，時，分，秒，曜日，1/1000秒」を読み出し，(d)で指定されたデバイス以降にBIN値で格納します。

(データ)	(d)	(d)+1	(d)+2	(d)+3	(d)+4	(d)+5	(d)+6	(d)+7
(内容)	年(西暦)	月	日	時(24時間計)	分	秒	曜日	1/1000秒
(設定範囲)	1980~2079	1~12	1~31	0~23	0~59	0~59	0~6	0~999

- (d)の「年」は，西暦4桁を格納します。
- (d)+6の「曜日」は，「日~土」を「0~6」で格納します。

曜日	日	月	火	水	木	金	土
格納データ	0	1	2	3	4	5	6

- うるう年は，自動的に補正します。

注意事項

- S(P).DATERD命令は，コントローラに間違った時計データを設定している場合でも，時計データを読み出してデバイスに格納します。(例: 2月30日)DATEWR(P)命令やエンジニアリングツールで時計データを設定時は，正しい時計データを設定してください。
- 1/1000秒の時計データ読み出し時の誤差は，最大2msです。(コントローラ内部の時計素子で記憶しているデータと，S(P).DATERD命令により読み出したデータとの誤差。)

エラー

演算エラーはありません。

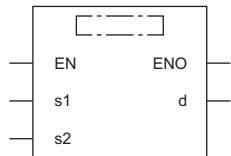
拡張時計データの加算

S(P).DATE+

時間データを加算します。

ラダー	ST
	ENO:=S_DATEPLUS(EN,s1,s2,d); ENO:=SP_DATEPLUS(EN,s1,s2,d);

FBD/LD



(□には、S_DATEPLUS、SP_DATEPLUSが入ります。)

■実行条件

命令	実行条件
S.DATE+	
SP.DATE+	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	加算される時刻データが格納されている先頭デバイス	機能参照	符号付きBIN16ビット	ANY16_ARRAY (要素数: 5)
(s2)	加算する時間(時刻)データの格納されている先頭デバイス	機能参照	符号付きBIN16ビット	ANY16_ARRAY (要素数: 5)
(d)	加算結果時刻(時間)データを格納する先頭デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 5)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—

機能

- (s1)で指定された時刻データに、(s2)で指定された時間データを加算し、加算結果を(d)で指定されたデバイス番号以降に格納します。

(s1)	hour	(0...23)		(s2)	hour	(0...23)		(d)	hour	(0...23)
(s1)+1	minute	(0...59)		(s2)+1	minute	(0...59)		(d)+1	minute	(0...59)
(s1)+2	second	(0...59)	+	(s2)+2	second	(0...59)	⇒	(d)+2	second	(0...59)
(s1)+3	—			(s2)+3	—			(d)+3	—	
(s1)+4	1/1000second	(0...999)		(s2)+4	1/1000second	(0...999)		(d)+4	1/1000second	(0...999)

例

6時32分40秒875に、7時48分10秒500を加算した場合

(s1)	6		(s2)	7		(d)	14
(s1)+1	32		(s2)+1	48		(d)+1	20
(s1)+2	40	+	(s2)+2	10	⇒	(d)+2	51
(s1)+3	—		(s2)+3	—		(d)+3	—
(s1)+4	875		(s2)+4	500		(d)+4	375

- 演算結果の時間が24時を超えた場合は、24時間を引いた値が演算結果となります。たとえば、14時20分30秒875に、20時20分20秒500を加算した場合、34時40分51秒375ではなく、10時40分51秒375になります。

(s1)	14		(s2)	20		(d)	10
(s1)+1	20		(s2)+1	20		(d)+1	40
(s1)+2	30	+	(s2)+2	20	⇒	(d)+2	51
(s1)+3	—		(s2)+3	—		(d)+3	—
(s1)+4	875		(s2)+4	500		(d)+4	375

Point

- (s1)+3, (s2)+3, (d)+3のデバイスは演算には使用しません。
- S(P).DATERD命令で読み出した時計データを、そのまま加算できます。

(d)	時
(d)+1	分
(d)+2	秒
(d)+3	曜日
(d)+4	1/1000秒

S(P).DATERD命令で読み出した場合、「秒」と「1/1000秒」の間に「曜日」が入ります。
S(P).DATE+命令では、演算しないため、そのまま加算が可能です。

エラー

エラーコード (SD0)	内容
3285H	(s1), (s2)の設定データが範囲外の時。

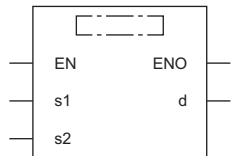
拡張時計データの減算

S(P).DATE-

時間データを減算します。

ラダー	ST
	ENO:=S_DATEMINUS(EN,s1,s2,d); ENO:=SP_DATEMINUS(EN,s1,s2,d);

FBD/LD



(□には、 S_DATEMINUS、 SP_DATEMINUSが入ります。)

■実行条件

命令	実行条件
S.DATE-	
SP.DATE-	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	減算される時刻データが格納されている先頭デバイス	機能参照	符号付きBIN16ビット	ANY16_ARRAY (要素数: 5)
(s2)	減算する時間(時刻)データの格納されている先頭デバイス	機能参照	符号付きBIN16ビット	ANY16_ARRAY (要素数: 5)
(d)	減算結果時刻(時間)データを格納する先頭デバイス	—	符号付きBIN16ビット	ANY16_ARRAY (要素数: 5)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—	—

機能

- (s1)で指定された時刻データから、(s2)で指定された時間データを減算し、減算結果を(d)で指定されたデバイス番号以降に格納します。

(s1)	hour	(0...23)	(s2)	hour	(0...23)	(d)	hour	(0...23)
(s1)+1	minute	(0...59)	(s2)+1	minute	(0...59)	(d)+1	minute	(0...59)
(s1)+2	second	(0...59)	(s2)+2	second	(0...59)	(d)+2	second	(0...59)
(s1)+3	—		(s2)+3	—		(d)+3	—	
(s1)+4	1/1000second	(0...999)	(s2)+4	1/1000second	(0...999)	(d)+4	1/1000second	(0...999)

例

10時40分20秒875から、3時50分10秒500を減算した場合

(s1)	10	(s2)	3	(d)	6
(s1)+1	40	(s2)+1	50	(d)+1	50
(s1)+2	20	(s2)+2	10	(d)+2	10
(s1)+3	—	(s2)+3	—	(d)+3	—
(s1)+4	875	(s2)+4	500	(d)+4	375

- 演算結果の時間が負の数になった場合は、そのデータに+24した値が演算結果となります。たとえば、4時50分32秒875から10時42分12秒500を減算した場合、-6時8分20秒375ではなく、18時8分20秒375になります。

(s1)	4	(s2)	10	(d)	18
(s1)+1	50	(s2)+1	42	(d)+1	8
(s1)+2	32	(s2)+2	12	(d)+2	20
(s1)+3	—	(s2)+3	—	(d)+3	—
(s1)+4	875	(s2)+4	500	(d)+4	375

Point

- (s1)+3, (s2)+3, (d)+3のデバイスは演算には使用しません。
- S(P).DATERD命令で読み出した時計データを、そのまま減算できます。

(d)	時
(d)+1	分
(d)+2	秒
(d)+3	曜日
(d)+4	1/1000秒

S(P).DATERD命令で読み出した場合、「秒」と「1/1000秒」の間に「曜日」が入ります。
S(P).DATE-命令では演算しないため、そのまま減算が可能です。

エラー

エラーコード (SD0)	内容
3285H	(s1), (s2)の設定データが範囲外の時。

21.2 タイミング計測命令

タイミングパルス発生

DUTY

ユーザ用タイミングクロックを指定したスキャン分ONさせ、指定したスキャン分OFFします。

ラダー	ST
	ENO:=DUTY(EN,n1,n2,d);

FBD/LD

■実行条件

命令	実行条件
DUTY	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(n1)	ONさせるスキャン数	0~65535	符号なしBIN16ビット	ANY16
(n2)	OFFさせるスキャン数	0~65535	符号なしBIN16ビット	ANY16
(d)	動作させるユーザ用タイミングクロック特殊リレーデバイス番号	SM420~SM424	ビット	ANY_BOOL*1
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 SM420~SM424に割り付けたラベルのみ使用できます。

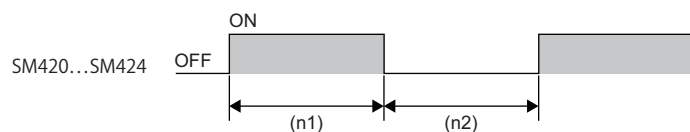
■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(n1)	○	○	○	○	○	—	—	○	○	—	—	—	—
(n2)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○*1	—	—	—	—	—	—	○	—	—	—	—	—

*1 SM420~SM424のみ使用できます。

機能

- (d)で指定されたSM420~SM424(ユーザ用タイミングクロック)を(n1)で指定されたスキャン分ONし, (n2)で指定されたスキャン分OFFします。



(n1): (n1)スキャン

(n2): (n2)スキャン

- スキャン実行タイププログラムではSM420~SM424を使用します。
- (n1), (n2)が0に設定された場合は, (n1)=0, (n2)≥0だとSM420~SM424はOFFのまま, (n1)>0, (n2)=0だとSM420~SM424はONのままとなります。
- DUTY命令実行時に(n1), (n2), (d)で指定されたデータをシステムに登録し, タイミングパルスのON/OFFはEND処理で行います。

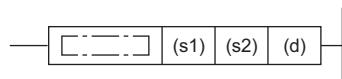
エラー

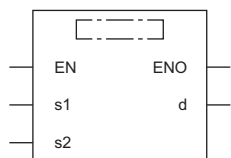
エラーコード (SD0)	内容
2820H	(d)で指定のデバイスが, SM420~SM424以外のとき。

指定データの時間計測

TIMCHK

デバイスのON時間を計測し、設定時間以上連続してONであれば指定デバイスをONします。

ラダー	ST
	ENO:=TIMCHK(EN,s1,s2,d);

FBD/LD


■実行条件

命令	実行条件
TIMCHK	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	計測した現在値を格納するデバイス(単位: 100ms)	—	符号付きBIN16ビット	ANY16
(s2)	計測の設定値, または計測の設定値を格納しているデバイス(単位: 100ms)	0~32767	符号付きBIN16ビット	ANY16
(d)	タイムアップ時にONさせるデバイス	—	ビット	ANY_BOOL
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(s2)	○	○	○	○	○	—	○	○	—	—	—
(d)	○	—	○	—	—	—	○	—	—	—	—

機能

- 条件のデバイスのON時間を計測し、(s2)で指定されたデバイスで設定した時間以上連続してONであれば、(d)で指定したデバイスをONさせます。
- (s1)で指定されたデバイスの現在値の0クリアと、(d)で指定されたデバイスのOFFは、実行指令の立上り時に行います。(s1)で指定されたデバイスの現在値と、(d)で指定したデバイスのON状態は、実行指令がOFFしても状態を保持します。
- 計測した現在値は100ms単位で格納されます。また、計測の設定値は100ms単位で設定します。
- (s2)に0を指定した場合、(s1)で指定されたデバイスの現在値の0クリアと、(d)で指定されたデバイスのOFFを、実行指令の立上り時に行います。
- (s2)に0~32767以外を指定した場合、実行指令がONした次のスキャンに(d)がONします。

エラー

演算エラーはありません。

22 ユニットアクセス

22

22.1 ユニットアクセス命令

I/Oリフレッシュ

RFS(P)

指定したデバイスからn点分をリフレッシュして、外部入力の取込み、または出力ユニットへの出力を行う。

ラダー	ST
	ENO:=RFS(EN,s,n); ENO:=RFSP(EN,s,n);

FBD/LD

■実行条件

命令	実行条件
RFS	
RFSP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	リフレッシュする先頭デバイス	—	ビット	ANY_BOOL ^{*1}
(n)	リフレッシュ点数	1~4096	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 デバイス(X, Y)に割り付けたラベルのみ使用できます。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード			間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	○ ^{*1}	—	—	—	—	—	—	—	—	—	—	—	—
(n)	—	○	○	○	○	—	—	○	○	○	—	—	—

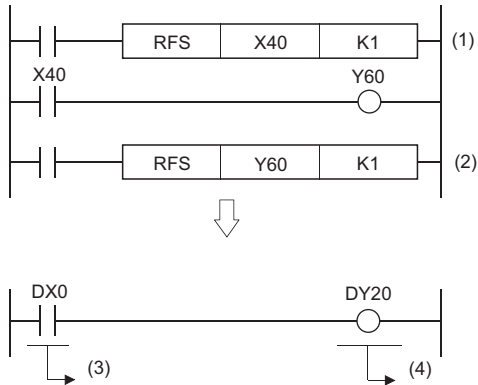
*1 X, Yのみ使用できます。

機能

- 1スキャン途中に該当するデバイスのみをリフレッシュして、外部入力の取込み、または出力ユニットへの出力を行う機能です。
- 入力の取込みおよび外部への出力は、プログラムのEND命令実行後にのみ一括で行うため、1スキャン中に外部にパルス信号を出力することはできません。I/Oリフレッシュ命令を実行すると、プログラム実行途中に該当する入力(X)または出力(Y)を強制的にリフレッシュするため、1スキャン中にパルス信号を外部に出力することができます。
- 入力(X)または出力(Y)を1点単位でリフレッシュする場合は、ダイレクトアクセス入力(DX)、ダイレクトアクセス出力(DY)を使用してください。

例

RFS命令によるプログラムをダイレクトアクセスによるプログラムにした場合



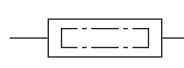
- (1)X40のリフレッシュを行います。
 (2)Y60のリフレッシュを行います。
 (3)ダイレクトアクセス入力
 (4)ダイレクトアクセス出力

エラー

エラーコード (SD0)	内容
2820H	(s)で指定したデバイスから(n)点の範囲が近接I/Oの範囲を超えたとき。

COM(P)

I/Oリフレッシュやネットワークユニットのリンクリフレッシュ、デバイス/ラベルアクセスサービス処理などを行います。

ラダー	ST
	ENO:=COM(EN); ENO:=COMP(EN);

FBD/LD



■実行条件

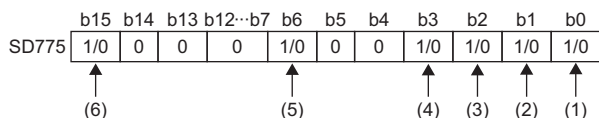
命令	実行条件
COM	
COMP	

機能

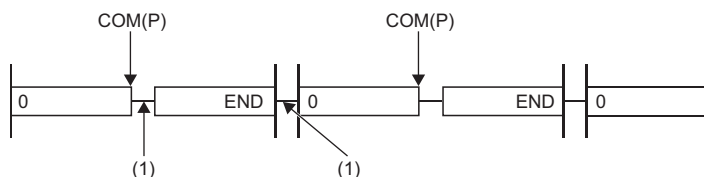
- COM(P)命令は、シーケンスプログラム実行途中の任意のタイミングで、I/Oリフレッシュなどを実施する場合に使用します。
- COM(P)命令を実行すると、下記の処理を行います。
 - I/Oリフレッシュ
 - CC-Linkユニットのリンクリフレッシュ
 - CC-Link IEコントローラネットワークユニットのリンクリフレッシュ
 - CC-Link IEフィールドネットワークユニットのリンクリフレッシュ
 - インテリジェント機能ユニットのリフレッシュ
 - デバイス/ラベルアクセスサービス処理(エンジニアリングツール、GOTまたはその他の外部機器との通信)
- SM775(COM命令実行時リフレッシュ処理選択)をOFFすると、I/Oリフレッシュを除く処理を行います。

処理内容	SM775 OFF時	SM775 ON時
I/Oリフレッシュ	非実行	実行/非実行の選択が可能
CC-Linkユニットのリンクリフレッシュ	実行	
CC-Link IEコントローラネットワークユニットのリンクリフレッシュ		
CC-Link IEフィールドネットワークユニットのリンクリフレッシュ		
インテリジェント機能ユニットのリフレッシュ		
デバイス/ラベルアクセスサービス処理(エンジニアリングツール, GOTまたはその他の外部機器との交信)		

- 処理を選択する場合は、SD775(COM命令実行時リフレッシュ処理選択)のb0~b3, b15で実行/非実行を選び、SM775をONします。

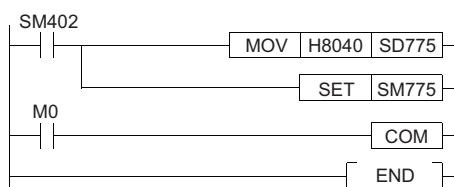


- (1) I/Oリフレッシュ
 - (2) CC-Linkユニットのリンクリフレッシュ
 - (3) CC-Link IEコントローラネットワークユニットのリンクリフレッシュ
 - (4) インテリジェント機能ユニットのリフレッシュ
 - (5) CC-Link IEフィールドネットワークユニットのリンクリフレッシュ
 - (6) デバイス/ラベルアクセスサービス処理(エンジニアリングツール, GOT, またはその他外部機器との通信)
- COM(P)命令実行時、指定されたリフレッシュ処理を行います。



(1): 指定した処理

- 下記のプログラム例では、M0がONしたとき、CC-Link IEフィールドネットワークユニットのリンクリフレッシュを実行します。



注意事項

- COM(P)命令はプログラム中に何回でも使用できます。ただし、SD775で選択されている処理の時間分ずつプログラムのスキャンタイムが延びます。
- COM(P)命令の実行中は、割込み許可状態となります。割込みプログラムなどでリフレッシュデータなどを使用する場合は、データの泣き別れが発生する可能性があります。
- EthernetポートにEthernetが接続されているときに、COM(P)命令でデバイス/ラベルアクセスサービス処理を行った場合、COM(P)命令の処理時間が延びる可能性があります。

Point

COM(P)命令は割込みプログラムでは使用できません。

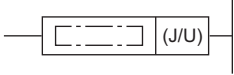
エラー

演算エラーはありません。

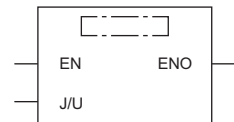
ユニットリフレッシュ

S(P).ZCOM

指定されたユニットのリフレッシュ処理を行います。

ラダー	ST
	ENO:=S_ZCOM(EN,J); ENO:=S_ZCOM(EN,U); ENO:=SP_ZCOM(EN,J); ENO:=SP_ZCOM(EN,U);

FBD/LD



■実行条件

命令	実行条件
S.ZCOM	
SP.ZCOM	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(J)* ¹	自局のネットワークNo.	1~239	デバイス名	ANY16
(U)* ¹	ユニットの先頭入出力番号(16進数4桁で表したときの上3桁)	0H~FFH		
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 CC-Link IE TSN Plusマスタ・ローカルユニットでは、(U)指定時はRJ71GN11-EIP(CC-Link IE TSN部)のリンクリフレッシュ (リフレッシュ設定時)とRJ71GN11-EIP(EtherNet/IP部)のリフレッシュ (リフレッシュ設定時)が動作し、(J)指定時はRJ71GN11-EIP(CC-Link IE TSN部)のリンクリフレッシュ (リフレッシュ設定時)のみ動作します。

■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他 (J/U)
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口		LT, LST, LC	LZ		K, H	E	\$	
(J/U)	—	—	—	—	—	—	—	—	—	—	—	○

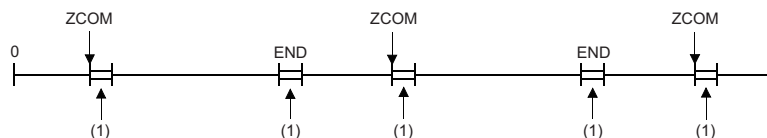
・ S(P).ZCOM命令は、シーケンスプログラムの実行途中に、任意のタイミングでリフレッシュする場合に使用します。

S(P).ZCOM命令によるリフレッシュの対象は下記になります。

- ・ CC-Link IE TSNマスタ・ローカルユニットのリンクリフレッシュ (リフレッシュ設定時)
- ・ CC-Link IEコントローラネットワークユニットのリンクリフレッシュ (リフレッシュ設定時)
- ・ CC-Link IEフィールドネットワークマスタ・ローカルユニットのリンクリフレッシュ (リフレッシュ設定時)
- ・ CC-Linkユニットのリンクリフレッシュ (リフレッシュ設定時)
- ・ インテリジェント機能ユニットのリフレッシュ (リフレッシュ設定時)

機能

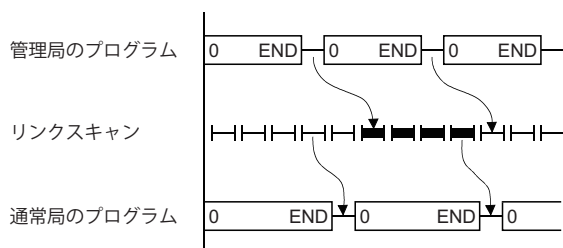
- S(P).ZCOM命令を実行すると、コントローラはシーケンスプログラムの処理を一時中断し、(J/U)で指定されたユニットのリフレッシュ処理を行います。



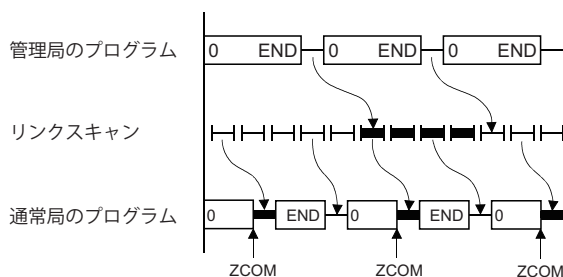
(1): リフレッシュ処理

- CC-Link IEコントローラネットワークユニットのリフレッシュ処理を行う場合は下記になります。
- S(P).ZCOM命令は、自局のシーケンスプログラムのスキャンタイムが他局のスキャンタイムより長い場合に、他局からのデータを確実に取り込むために使

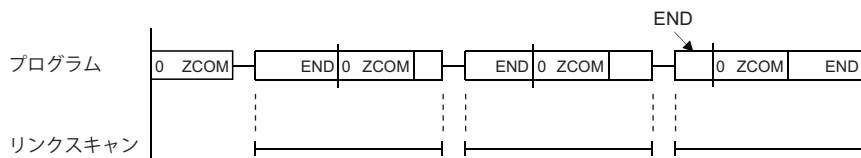
[S(P).ZCOM命令を使用しない場合]



[S(P).ZCOM命令を使用した場合]



- シーケンスプログラムのスキャンタイムよりもリンクスキャンタイムの方が長い場合は、S(P).ZCOM命令を使用してもデータの送受信は早くなりません。



注意事項

- S(P).ZCOM命令はプログラム中に何回でも使用できます。ただし、リフレッシュ時間分ずつプログラムのスキャンタイムが延びます。
- S(P).ZCOM命令実行中、割り込み許可状態になります。割り込みプログラムなどでリフレッシュデータを使用する場合は、データの泣き別れが発生することがあります。

エラー

エラーコード (SD0)	内容
2800H	指定した先頭入出力番号が0~FFH以外のとき。
2801H	先頭入出力番号で指定した位置にユニットが存在しないとき。
2804H	指定したネットワークNo.が1~239以外のとき。
2805H	指定したネットワークNo.が自局に接続されていないとき。

Point

- S(P).ZCOM命令は割り込みプログラムでは使用できません。
- 外部機器との交信のみを実行したい場合は、COM(P)命令を使用してください。

ユニットからの1ワード/2ワードデータリード(16ビット指定)

FROM(P), DFROM(P)

• FROM(P):

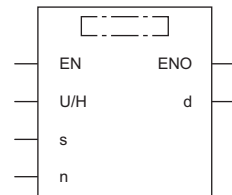
指定されたユニット内のバッファメモリアドレスからnワードのデータを読み出します。

• DFROM(P):

指定されたユニット内のバッファメモリアドレスからn×2ワードのデータを読み出します。

ラダー	ST
	ENO:=FROM(EN,U/H,s,n,d); ENO:=FROMP(EN,U/H,s,n,d); ENO:=DFROM(EN,U/H,s,n,d); ENO:=DFROMP(EN,U/H,s,n,d);

FBD/LD



■実行条件

命令	実行条件
FROM DFROM	
FROMP DFROMP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(U/H)	ユニットおよびコントローラの先頭入出力番号 (16進数4桁で表したときの上3桁)	0H~FFH, 3E0H	符号なしBIN16ビット	ANY16
(s)	読み出すデータが格納されているバッファメモリ およびCPUメモリの先頭アドレス	0~65535	符号なしBIN16ビット	ANY16
(d)	FROM(P) DFROM(P)	読み出したデータを格納する先頭デバイス	符号付きBIN16ビット 符号付きBIN32ビット	ANY16* ¹ ANY32* ¹
(n)	読出しデータ数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

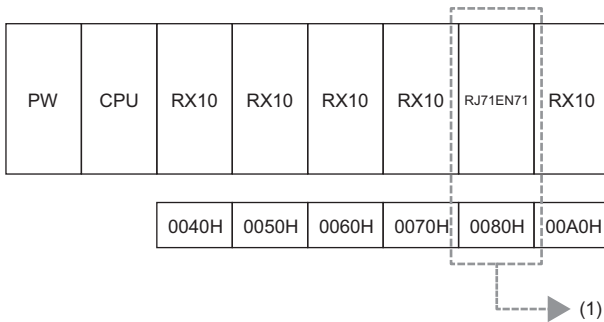
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 0	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 0	Z	LT, LST, LC	LZ		K, H	E	\$	
(U/H)	○	○	○	○	○	—	—	○	○	—	—	○
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	—	○	—	—	○* ¹	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

*1 DFROM(P)命令のみ使用できます。

機能

- (U/H)は、ユニットおよびコントローラ先の入出力番号を、16進数4桁で表したときの上3桁で指定します。



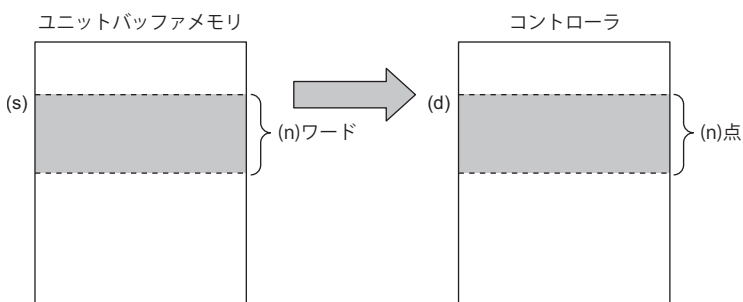
(1): 読み出すユニットの先頭入出力番号はK8またはH8を指定

コントローラ先の入出力番号は、下記の表のように指定します。

コントローラ	先頭入出力番号
1号機	3E0H

FROM(P)

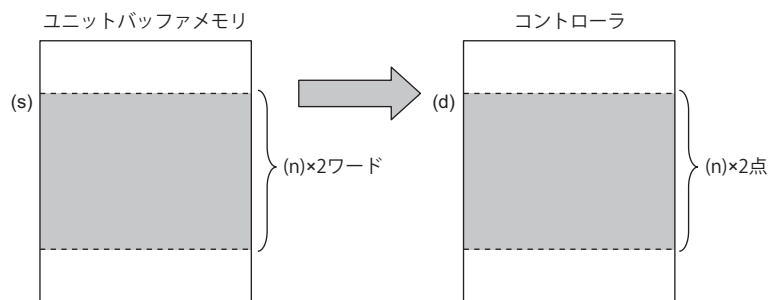
- (U/H)で指定したユニット内のバッファメモリの(s)で指定したアドレスから、(n)ワード分のデータを読み出し、(d)で指定したデバイス以降に格納します。
- ユニットからのワードデータリード



- 読み出しデータ(n)が0のときは、無処理となります。
- 命令実行時、対象ユニットの異常や処理中などでアクセスできない場合は、無処理となります。

DFROM(P)

- (U/H)で指定したユニット内のバッファメモリの(s)で指定したアドレスから、(n)×2ワード分のデータを読み出し、(d)で指定したデバイス以降に格納します。
- ユニットからのダブルワードデータリード

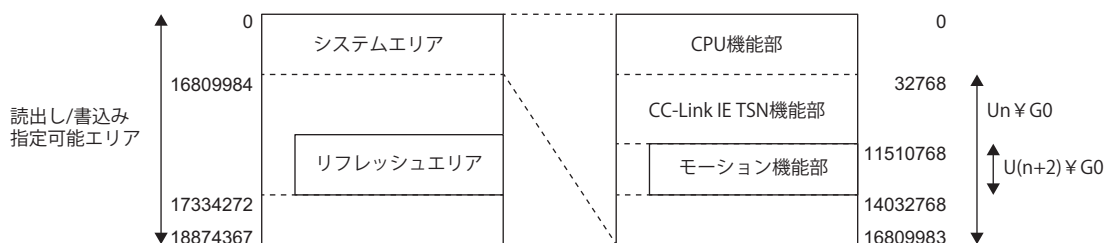


- 読み出しデータ(n)が0のときは、無処理となります。
- 命令実行時、対象ユニットの異常や処理中などでアクセスできない場合は、無処理となります。

エラーコード (SD0)	内容
2820H	(U)で指定の入出力番号がバッファメモリを持たないユニットであるとき。
2823H	(H)で指定の入出力番号がバッファメモリを持たないユニットであるとき。 (s)で指定したアドレスがバッファメモリの範囲外の時。 (s)で指定したアドレスから(n)点分が、バッファメモリの範囲外の時。(FROM(P)命令の場合) (s)で指定したアドレスから2×(n)点分が、バッファメモリの範囲外の時。(DFROM(P)命令の場合)

Point

- ユニットのデータの読出しは、ユニットアクセスデバイスを使用して行うこともできます。(使用コントローラのユーザーズマニュアル)
- CPUバッファメモリの読出し/書込み可能エリアのリフレッシュエリアについて、リフレッシュ設定しない場合は、読出し/書込み指定可能エリアとして使用できます。また、リフレッシュ設定した場合でも、リフレッシュ送信範囲以降は読出し/書込み指定可能エリアとして使用できます。オペランド(U/H)にCC-Link IE TSN機能部またはモーション機能部の先頭入出力番号Un ¥ G0, U(n+2) ¥ G0を指定した場合、下記の範囲にアクセスできます。

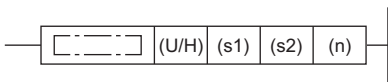


- CPUバッファメモリからのデータの読出しは、CPUバッファメモリアクセスデバイスが使用できます。(使用コントローラのユーザーズマニュアル)
- FROM(P)命令、DFROM(P)命令では、64Kバイト以下のバッファメモリアドレスからデータを読み出すことができます。64Kバイトを超えるバッファメモリアドレスからデータを読み出す場合は、FROMD(P)命令、DFROMD(P)命令を使用します。(1011ページ FROMD(P), DFROMD(P))

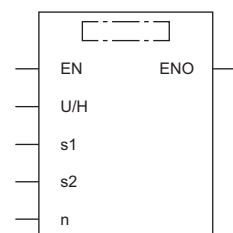
ユニットへの1ワード/2ワードデータライト(16ビット指定)

TO(P), DTO(P)

- TO(P): 指定したデバイスからn点のデータをユニット/自号機コントローラ内のバッファメモリに書き込みます。
- DTO(P): 指定したデバイスからn×2点のデータをユニット/自号機コントローラ内のバッファメモリに書き込みます。

ラダー	ST
	ENO:=TO(EN,U/H,s1,s2,n); ENO:=TOP(EN,U/H,s1,s2,n); ENO:=DTO(EN,U/H,s1,s2,n); ENO:=DTOP(EN,U/H,s1,s2,n);

FBD/LD



■実行条件

命令	実行条件
TO DTO	
TOP DTOP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(U/H)	ユニットおよびコントローラの先頭入出力番号 (16進数4桁で表したときの上3桁)	0H~FFH, 3E0H	符号なしBIN16ビット	ANY16
(s1)	データを書き込むバッファメモリおよびCPUメモリの先頭アドレス	0~65535	符号なしBIN16ビット	ANY16
(s2)	TO(P) DTO(P)	-32768~32767	符号付きBIN16ビット	ANY16 ^{*1}
		-2147483648~2147483647	符号付きBIN32ビット	ANY32 ^{*1}
(n)	書き込みデータ数	0~65535	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

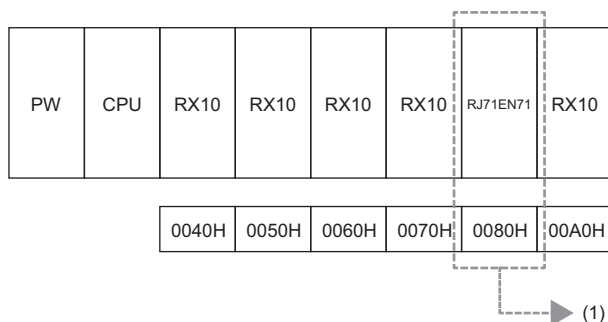
■使用可能デバイス

オペランド	ビット		ワード		Z	ダブルワード		間接指定	定数			その他 (U)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□		LT, LST, LC	LZ		K, H	E	\$	
(U/H)	○	○	○	○	○	—	—	○	○	—	—	○
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	—	○	—	—	○ ^{*1}	—	○	○	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

*1 DTO(P)命令のみ使用できます。

機能

- (U/H)は、ユニットおよびコントローラの手元入出力番号を、16進数4桁で表したときの上3桁で指定します。



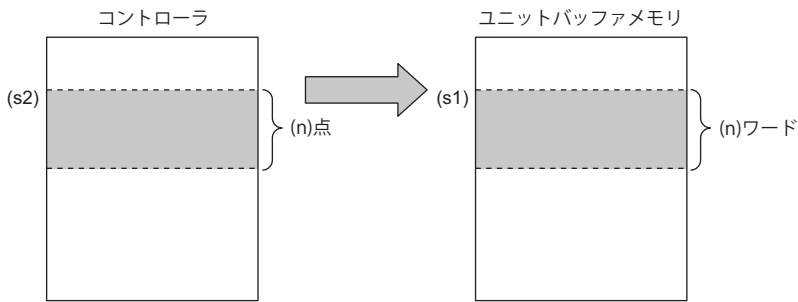
(1): 書き込むユニットの手元入出力番号はK8またはH8を指定

コントローラの手元入出力番号は、下記の表のように指定します。

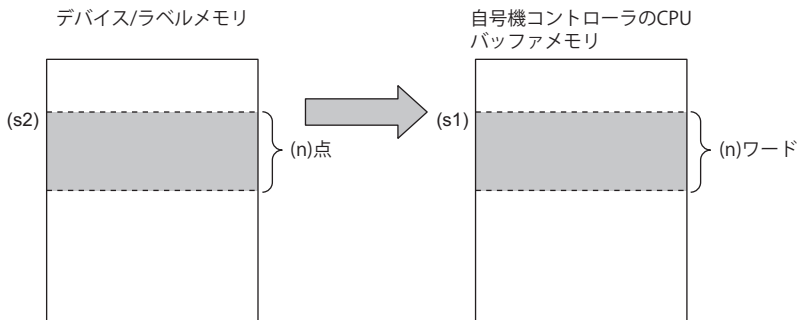
コントローラ	手元入出力番号
1号機	3E0H

■TO(P)

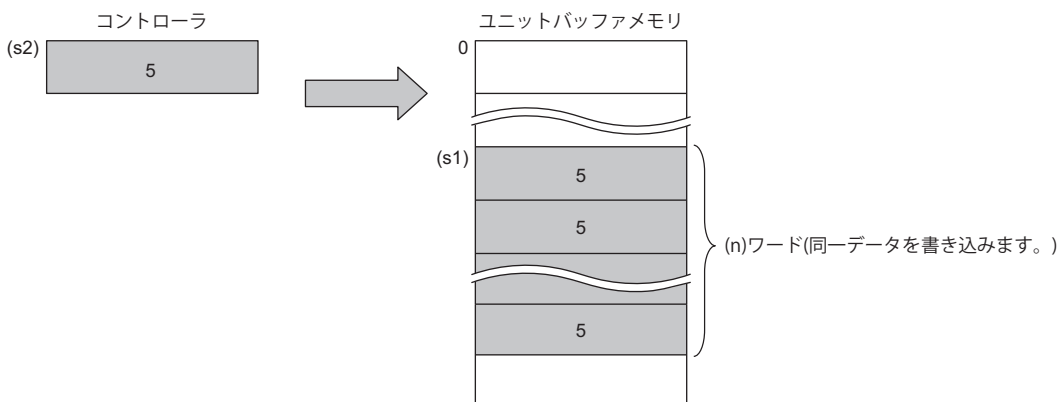
- (s2)で指定したデバイスから(n)点分のデータを、(U/H)で指定したユニットまたは自号機コントローラ内のバッファメモリの(s1)で指定したアドレス以降に書き込みます。
- ユニットへのワードデータライト



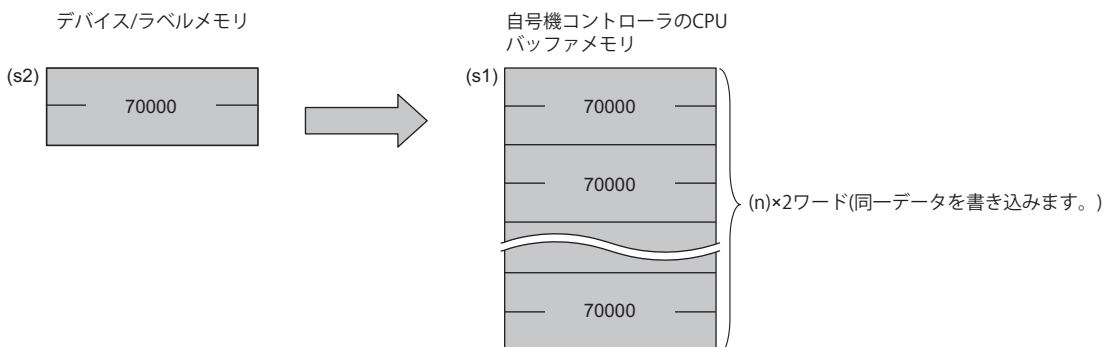
- 自号機コントローラへのワードデータライト



- (s2)に定数を指定した場合は、同一データ((s2)で指定した値)を、指定したバッファメモリアドレスから(n)ワード分には書き込みます。
- ユニットへのワードデータライト



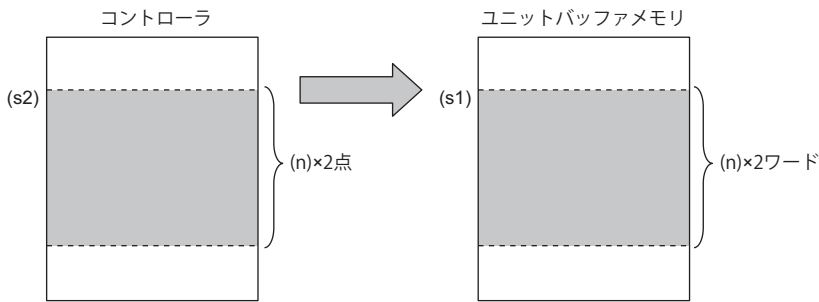
- 自号機コントローラへのワードデータライト



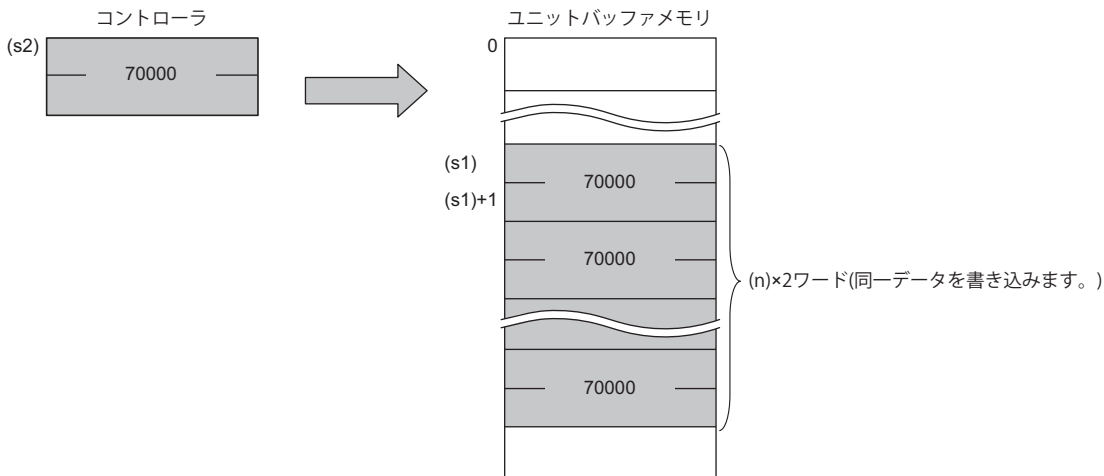
- 命令実行時、対象ユニットの異常や処理中などでアクセスできない場合は、無処理となります。

■DTO(P)

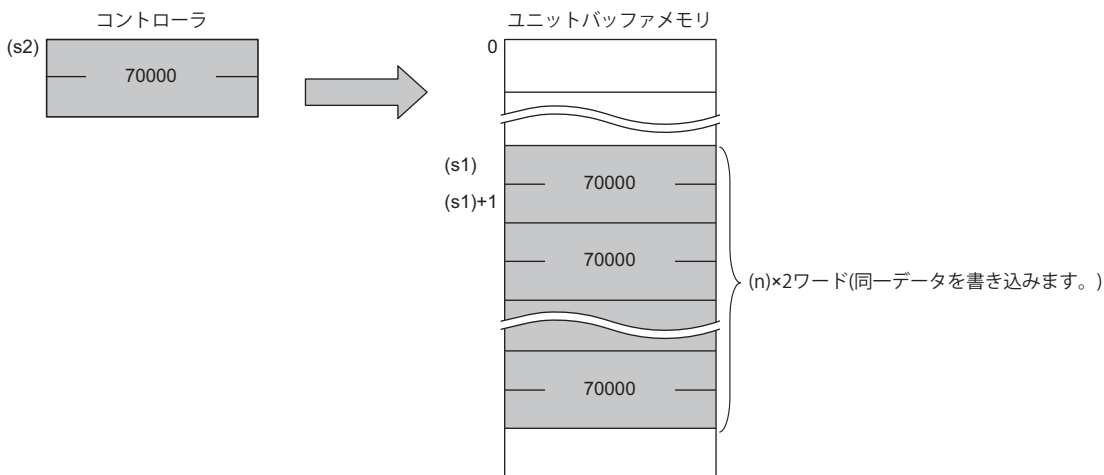
- (s2)で指定したデバイスから(n)×2点分のデータを、(U/H)で指定したユニットまたは自号機コントローラ内のバッファメモリの(s1)で指定したアドレス以降に書き込みます。
- ユニットへのダブルワードデータライト



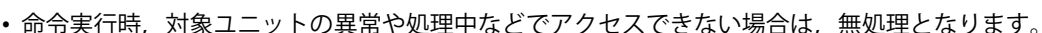
- 自号機コントローラへのダブルワードデータライト



- (s2)に定数を指定した場合は、同一データ((s2)で指定した値)を、指定したバッファメモリアドレスから(n)×2ワード分には書き込みます。
- ユニットへのダブルワードデータライト



- デバイス/ラベルメモリ



ユニットからの1ワード/2ワードデータリード(32ビット指定)

FROMD(P), DFROMD(P)

• FROMD(P):

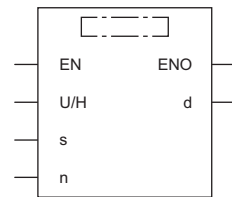
指定されたユニット内のバッファメモリアドレスからnワードのデータを読み出します。

• DFROMD(P):

指定されたユニット内のバッファメモリアドレスからn×2ワードのデータを読み出します。

ラダー	ST
	<pre> ENO:=FROMD(EN,U/H,s,n,d); ENO:=FROMDP(EN,U/H,s,n,d); ENO:=DFROMD(EN,U/H,s,n,d); ENO:=DFROMDP(EN,U/H,s,n,d); </pre>

FBD/LD



■実行条件

命令	実行条件
FROMD DFROMD	
FROMDP DFROMDP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(U/H)	ユニットおよびコントローラの手元入出力番号 (16進数4桁で表したときの上3桁)	0H~FFH, 3E0H	符号なしBIN16ビット	ANY16
(s)	読み出すデータが格納されているバッファメモリおよびCPUメモリの先頭アドレスを格納する 先頭デバイス	0~4294967295	符号なしBIN32ビット	ANY32
(d)	FROMD(P)	—	符号付きBIN16ビット	ANY16 ^{*1}
	DFROMD(P)		符号付きBIN32ビット	ANY32 ^{*1}
(n)	読み出しデータ数	0~4294967295	符号なしBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

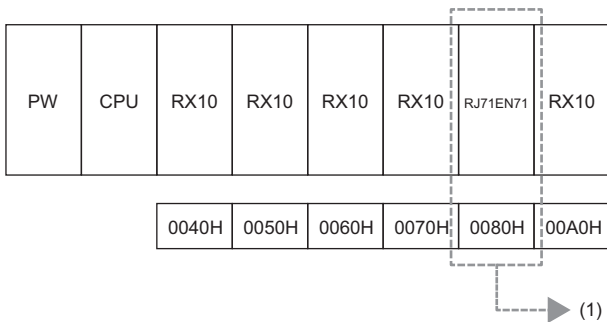
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(U/H)	○	○	○	○	○	—	—	○	○	—	—	○
(s1)	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	—	○	—	—	○ ^{*1}	—	○	○	—	—	—
(n)	○	○	○	○	○	○	○	○	○	—	—	—

*1 DFROMD(P)命令のみ使用できます。

機能

- (U/H)は、ユニットおよびコントローラの手元入出力番号を、16進数4桁で表したときの上3桁で指定します。



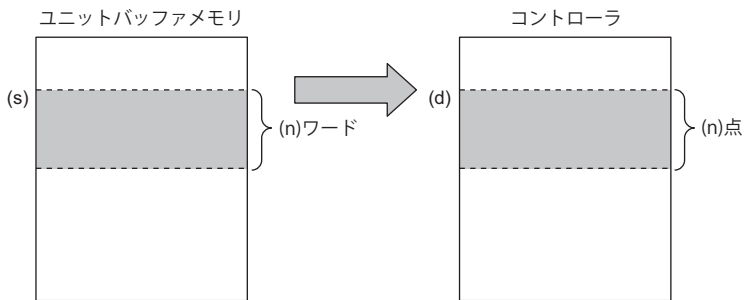
(1): 読み出すユニットの手元入出力番号はK8またはH8を指定

コントローラの手元入出力番号は、下記の表のように指定します。

コントローラ	手元入出力番号
1号機	3E0H

■FROMD(P)

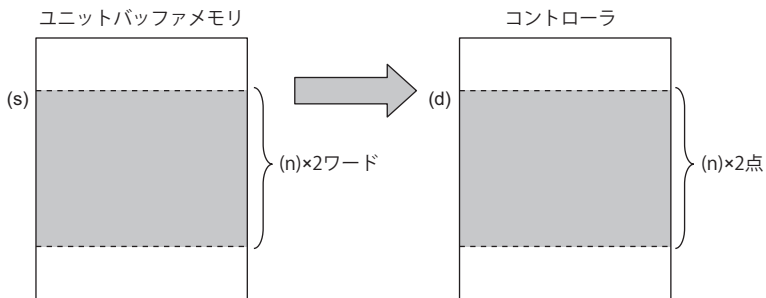
- (U/H)で指定したユニット内のバッファメモリの(s)で指定したアドレスから、(n)ワード分のデータを読み出し、(d)で指定したデバイス以降に格納します。
- ユニットからのワードデータリード



- 読み出しデータ(n)が0のときは、無処理となります。
- 命令実行時、対象ユニットの異常や処理中などでアクセスできない場合は、無処理となります。

■DFROMD(P)

- (U/H)で指定したユニット内のバッファメモリの(s)で指定したアドレスから、(n)×2ワード分のデータを読み出し、(d)で指定したデバイス以降に格納します。
- ユニットからのダブルワードデータリード



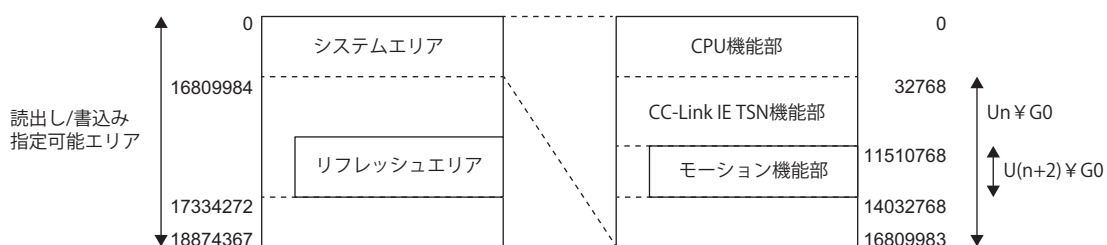
- 読み出しデータ(n)が0のときは、無処理となります。
- 命令実行時、対象ユニットの異常や処理中などでアクセスできない場合は、無処理となります。

エラー

エラーコード (SD0)	内容
2820H	(U)で指定の入出力番号がバッファメモリを持たないユニットであるとき。
	(s)で指定したアドレスがCPUバッファメモリの範囲外の時。
	(s)で指定したアドレスから(n)点分が、CPUバッファメモリの範囲外の時。(FROMD(P)命令の場合)
	(s)で指定したアドレスから2×(n)点分が、CPUバッファメモリの範囲外の時。(DFROMD(P)命令の場合)
2823H	(H)で指定の入出力番号がバッファメモリを持たないユニットであるとき。
	(s)で指定したアドレスがバッファメモリの範囲外の時。
	(s)で指定したアドレスから(n)点分が、バッファメモリの範囲外の時。(FROMD(P)命令の場合)
	(s)で指定したアドレスから2×(n)点分が、バッファメモリの範囲外の時。(DFROMD(P)命令の場合)

Point

- CPUバッファメモリの読出し/書込み可能エリアのリフレッシュエリアについて、リフレッシュ設定しない場合は、読出し/書込み指定可能エリアとして使用できます。また、リフレッシュ設定した場合でも、リフレッシュ送信範囲以降は読出し/書込み指定可能エリアとして使用できます。オペランド(U/H)にCC-Link IE TSN機能部またはモーション機能部の先頭入出力番号 $Un \neq G0$, $U(n+2) \neq G0$ を指定した場合、下記の範囲にアクセスできます。



- CPUバッファメモリからのデータの読出しは、CPUバッファメモリアクセスデバイスが使用できます。
(使用するコントローラのユーザーズマニュアル)
- FROMD(P)命令、DFROMD(P)命令では、64Kバイトを超えるバッファメモリアドレスからデータを読み出すことができます。

ユニットへの1ワード/2ワードデータライト(32ビット指定)

TOD(P), DTOD(P)

- TOD(P): 指定したデバイスからn点のデータをユニット/自号機コントローラ内のバッファメモリに書き込みます。
- DTOD(P): 指定したデバイスからn×2点のデータをユニット/自号機コントローラ内のバッファメモリに書き込みます。

ラダー	ST
	ENO:=TOD(EN,U/H,s1,s2,n); ENO:=TODP(EN,U/H,s1,s2,n); ENO:=DTOD(EN,U/H,s1,s2,n); ENO:=DTODP(EN,U/H,s1,s2,n);

FBD/LD

■実行条件

命令	実行条件
TOD DTOD	
TODP DTODP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(U/H)	ユニットおよびコントローラの先頭入出力番号 (16進数4桁で表したときの上3桁)	0H~FFH, 3E0H	符号なしBIN16ビット	ANY16
(s1)	データを書き込むバッファメモリおよびCPUメモリの先頭アドレスを格納する先頭デバイス	0~4294967295	符号なしBIN32ビット	ANY32
(s2)	TOD(P)	-32768~32767	符号付きBIN16ビット	ANY16 ^{*1}
	DTOD(P)	-2147483648~2147483647	符号付きBIN32ビット	ANY32 ^{*1}
(n)	書き込みデータ数	0~4294967295	符号なしBIN32ビット	ANY32
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

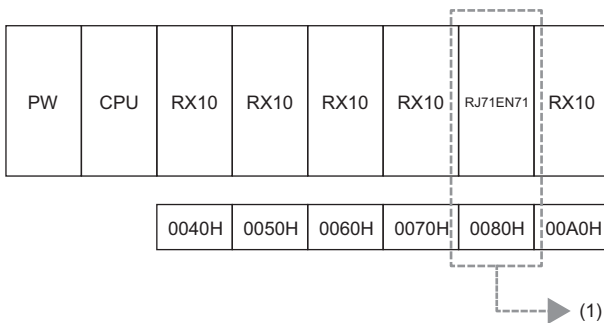
■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(U/H)	○	○	○	○	○	—	—	○	○	—	—	○
(s1)	○	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	—	○	—	—	○ ^{*1}	—	○	○	—	—	—
(n)	○	○	○	○	○	○	○	○	○	—	—	—

*1 DTOD(P)命令のみ使用できます。

機能

- (U/H)は、ユニットおよびコントローラの手元入出力番号を、16進数4桁で表したときの上3桁で指定します。



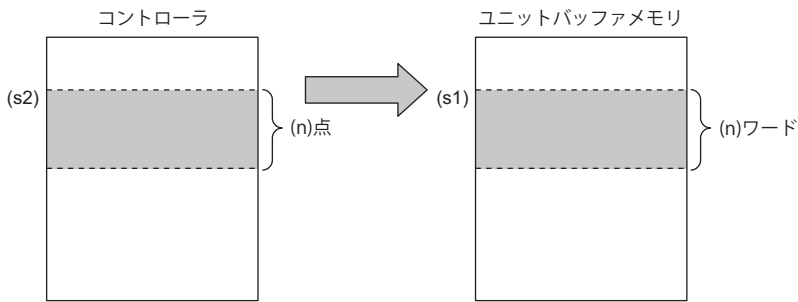
(1): 書き込むユニットの手元入出力番号はK8またはH8を指定

コントローラの手元入出力番号は、下記の表のように指定します。

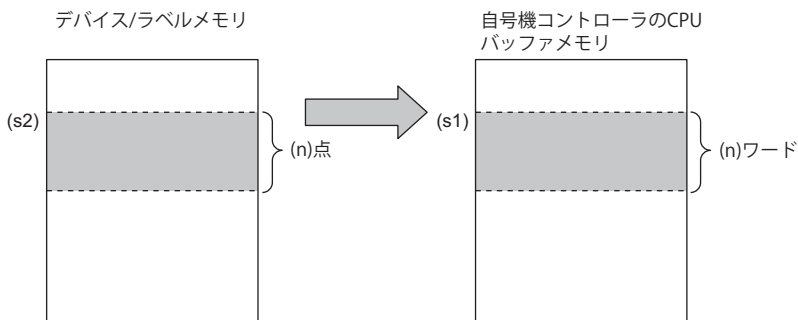
コントローラ	手元入出力番号
1号機	3E0H

■TOD(P)

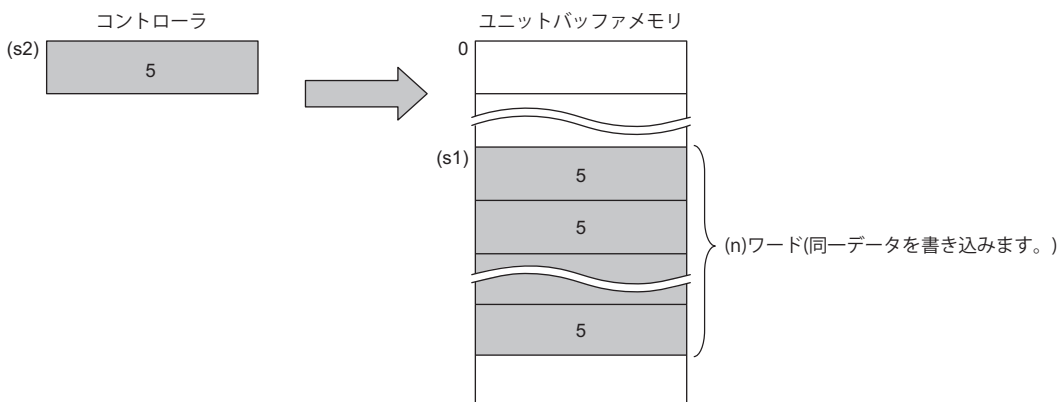
- (s2)で指定したデバイスから(n)点分のデータを、(U/H)で指定したユニットまたは自号機コントローラ内のバッファメモリの(s1)で指定したアドレス以降に書き込みます。
- ユニットへのワードデータライト



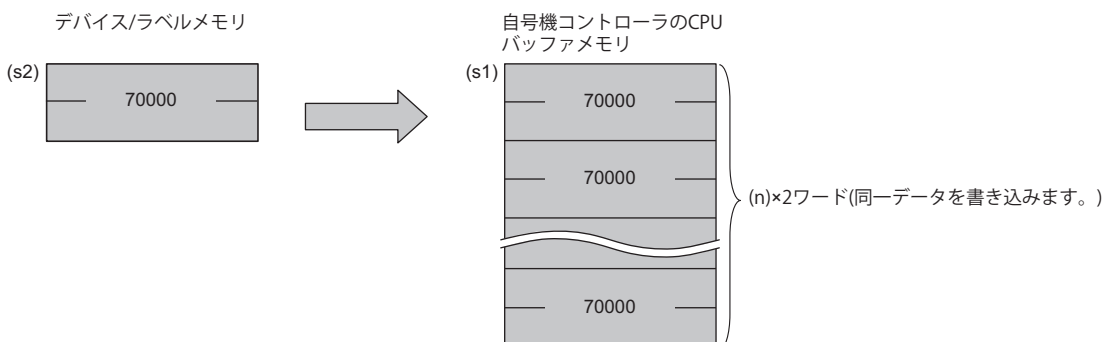
- 自号機コントローラへのワードデータライト



- (s2)に定数を指定した場合は、同一データ((s2)で指定した値)を指定したバッファメモリアドレスから(n)ワード分には書き込みます
- ユニットへのワードデータライト



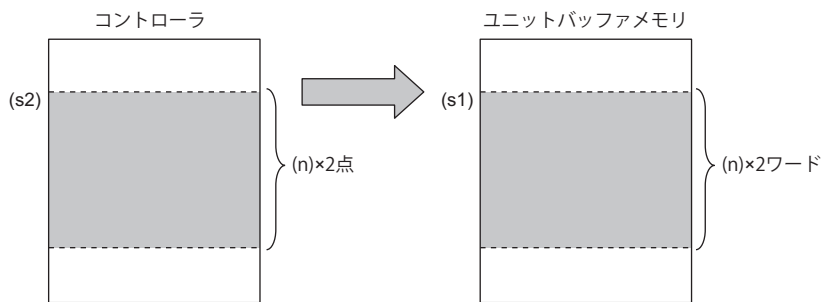
- 自号機コントローラへのワードデータライト



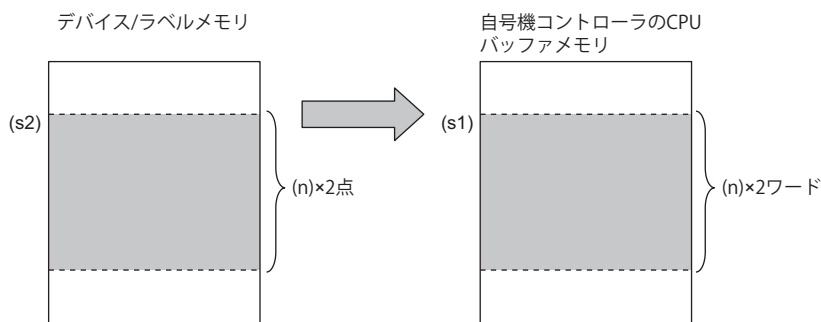
- 命令実行時、対象ユニットの異常や処理中などでアクセスできない場合は、無処理となります。

■DTOD(P)

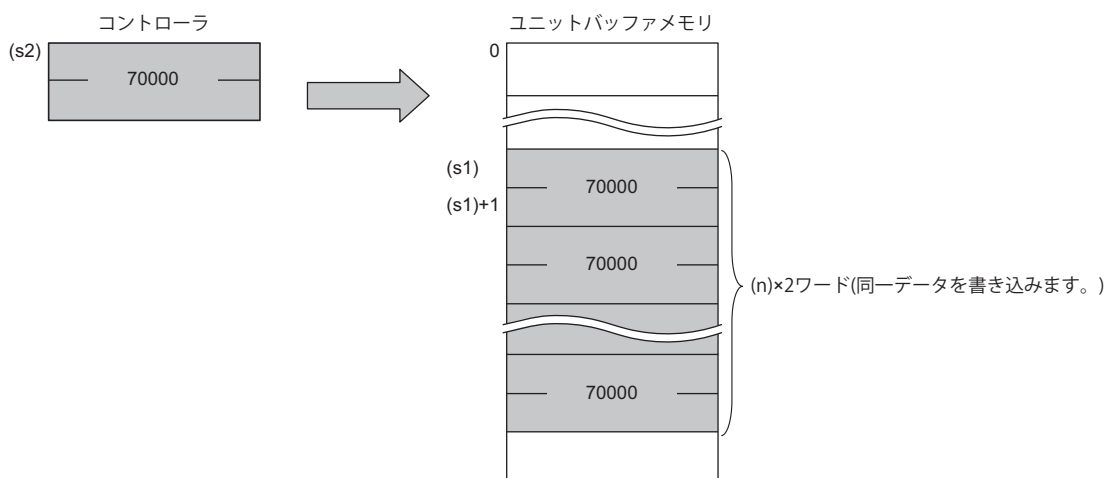
- (s2)で指定したデバイスから(n)×2点分のデータを、(U/H)で指定したユニットまたは自号機コントローラ内のバッファメモリの(s1)で指定したアドレス以降に書き込みます。
- ユニットへのダブルワードデータライト



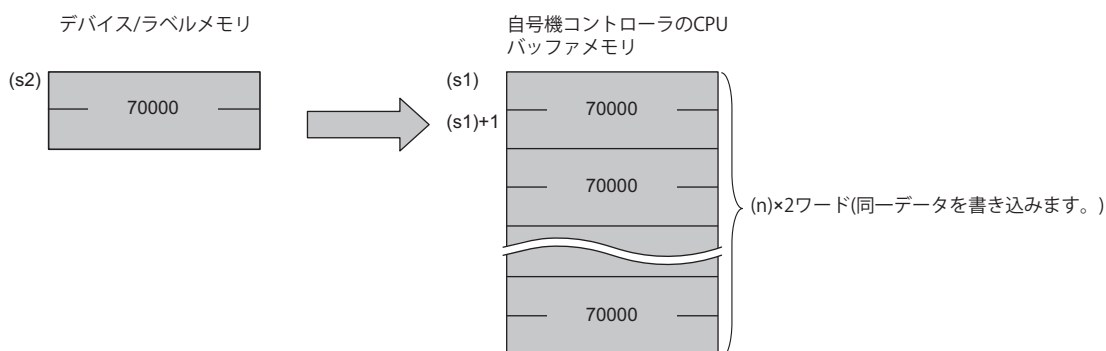
- 自号機コントローラへのダブルワードデータライト



- (s2)に定数が指定された場合は、同一データ((s2)で指定した値)を指定したバッファメモリアドレスから(n)×2ワード分には書き込みます。
- ユニットへのダブルワードデータライト



- 自号機コントローラへのダブルワードデータライト

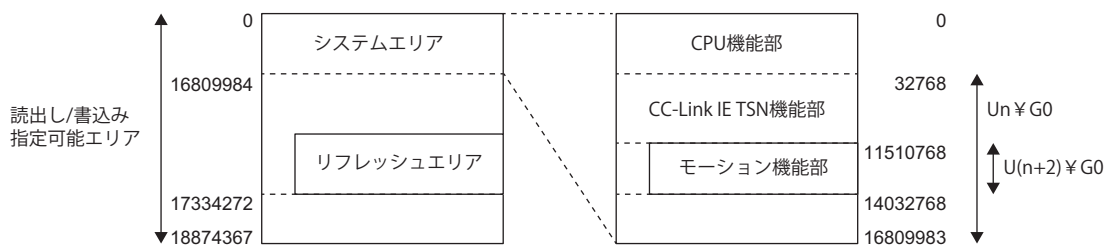


- 命令実行時、対象ユニットの異常や処理中などでアクセスできない場合は、無処理となります。

エラーコード (SD0)	内容
2820H	(U)で指定の入出力番号がバッファメモリを持たないユニットであるとき。
	(s1)で指定したアドレスがCPUバッファメモリの範囲外のとき。
	(s1)で指定したアドレスから(n)点分が、CPUバッファメモリの範囲外のとき。(TOD(P)命令の場合)
	(s1)で指定したアドレスから2×(n)点分が、CPUバッファメモリの範囲外のとき。(DTOD(P)命令の場合)
2823H	(H)で指定の入出力番号がバッファメモリを持たないユニットであるとき。
	(s1)で指定したアドレスがバッファメモリの範囲外のとき。
	(s1)で指定したアドレスから(n)点分が、バッファメモリの範囲外のとき。(TOD(P)命令の場合)
	(s1)で指定したアドレスから2×(n)点分が、バッファメモリの範囲外のとき。(DTOD(P)命令の場合)

Point

- CPUバッファメモリの読出し/書込み可能エリアのリフレッシュエリアについて、リフレッシュ設定しない場合は、読出し/書込み指定可能エリアとして使用できます。また、リフレッシュ設定した場合でも、リフレッシュ送信範囲以降は読出し/書込み指定可能エリアとして使用できます。オペランド(U/H)にCC-Link IE TSN機能部またはモーション機能部の先頭入出力番号Un ¥ G0, U(n+2) ¥ G0を指定した場合、下記の範囲にアクセスできます。

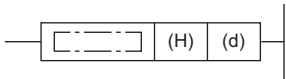


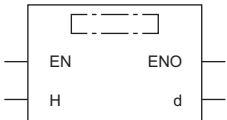
- CPUバッファメモリへのデータの書込みは、CPUバッファメモリアクセスデバイスが使用できます。(使用コントローラのユーザズマニュアル)
- TOD(P)命令, DTOD(P)命令では、64Kバイトを超えるバッファメモリアドレスにデータを書き込むことができます。

ユニット形名読出し

TYPERD(P)

指定したスロットのユニット形名を読み出します。

ラダー	ST
	ENO:=TYPERD(EN,H,d); ENO:=TYPERDP(EN,H,d);

FBD/LD


■実行条件

命令	実行条件
TYPERD	
TYPERDP	

設定データ

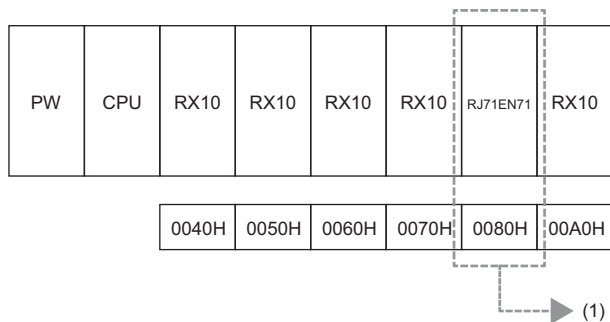
■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(H)	ユニット形名を読み出すユニットの先頭入出力番号(16進数4桁で表したときの3桁)	0H~FFH, 3E0H	符号なしBIN16ビット	ANY16
(d)	(d)+0: 命令実行結果, (d)+1~(d)+9: ユニット形名	—	ワード	ANY16_ARRAY (要素数: 10)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥□	G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(H)	○	—	○	—	—	—	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	—	○	—	—	—	—

- (H)で指定したスロットのユニット形名を読み出し、(d)で指定したデバイス以降へ格納します。対象ユニットは下記のとおりです。
 - コントローラ
 - 入力ユニット
 - 出力ユニット
 - 入出力混合ユニット
 - インテリジェント機能ユニット
- (H)には読出し対象ユニットの先頭入出力番号を、16進数4桁で表したときの3桁で指定します。



(1): 読み出すユニットの先頭入出力番号はK8またはH8を指定

- 2スロットを占有しているユニットを指定する場合、読出し対象として指定するスロットは、各ユニットのマニュアルに記載している入出力占有点数を参照してください。
- コントローラの入出力番号は、下記の表のように指定します。

コントローラ	先頭入出力番号
1号機	3E0H

- (d)+0には命令実行結果を格納し、(d)+1~(d)+9にはユニット形名を格納します。(d)に格納する値は下記のとおりです。

条件	(d)+0	(d)+1~(d)+9
読出し対象ユニットが形名を保持している場合	0	ユニットが保持している形名
読出し対象ユニットが形名を保持していない場合	1	ユニット種別と点数を組み合わせた文字列
読出し対象スロットが空きスロットの場合	-1	0000H
読出し対象ユニットがオンライン交換中の場合		
(H)で指定した入出力番号がユニットの先頭番号でない場合		
読出し対象ユニットが再起動中の場合		

- ・ 読出し対象ユニットが形名を保持している場合、(d)+1以降に格納するユニット形名は下記のとおりです。
 - ・ 9ワード使用します。
 - ・ ASCII文字で格納します。
 - ・ 18文字目は00Hを格納します。
 - ・ 文字数が17文字に満たない場合、残りの文字にも00Hを格納します。
 - ・ ユニットが保持している形名を格納します。(定格銘板に記載されている形名とは異なる場合がありますので注意してください)

例

格納する形名の例を示します。

対象ユニット	格納する形名の例
コントローラ	MXR300-16
入出力ユニット	INPUT_16
ネットワークユニット	RJ71GP21-SX

Point

I/O割付設定で設定されている形名と、実装されているユニットの形名が異なっている場合、実装されているユニットが保持している形名を格納します。

- ・ 読出し対象ユニットが形名を保持していない場合、(d)+1以降に格納する文字列は下記のとおりです。
 - ・ 9ワード使用します。
 - ・ ASCII文字で格納します。
 - ・ 18文字目は00Hを格納します。
 - ・ 文字数が17文字に満たない場合、残りの文字にも00Hを格納します。
 - ・ 「ユニット種別を示す文字列」と「点数を示す文字列」を組み合わせた文字列を格納します。

例

格納する文字列の例を示します。

対象ユニット	格納する文字列の例
入力ユニット	INPUT_16
出力ユニット	OUTPUT_32
入出力混合ユニット	MIXED_64
インテリジェント機能ユニット	INTELLIGENT_128

点数を示す文字列の例を示します。

点数	点数を示す文字列の例
16点	_16
32点	_32
48点	_48
64点	_64
128点	_128
256点	_256
512点	_512
1024点	_1024

Point

I/O割付設定で設定されている点数と、実装されているユニットの点数が異なっている場合、実装されているユニットの点数を格納します。

- ・コントローラのCC-Link IE TSN機能部・モーション機能部の先頭入出力番号を指定してユニット形名を読み出した場合、下記のように形名を格納されていません。コントローラのシステム構成の詳細は使用するコントローラのユーザーズマニュアルを確認してください。

例

コントローラに対して各機能のユニット形名を読み出した場合の例を示します。

指定するシステム	先頭入出力番号	(d)+0	(d)+1~(d)+9
CPU機能部	3E00H	0	MXR300-16
CC-Link IE TSN機能部・モーション機能部	0000H	-1	0000H
	0010H		
	0020H		
	0030H		

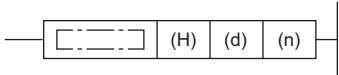
エラー

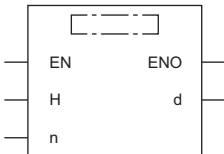
エラーコード (SD0)	内容
2800H	(H)の指定が0~FFH, 3E0H以外のとき。
2810H	読み出し対象ユニットが故障などの原因で交信できないとき。

ユニット固有情報読出し

UNIINFRD(P)

指定した点数分のユニット情報を読み出します。

ラダー	ST
	ENO:=UNIINFRD(EN,H,n,d); ENO:=UNIINFRDP(EN,H,n,d);

FBD/LD


■実行条件

命令	実行条件
UNIINFRD	
UNIINFRDP	

設定データ

■内容, 範囲, データ型

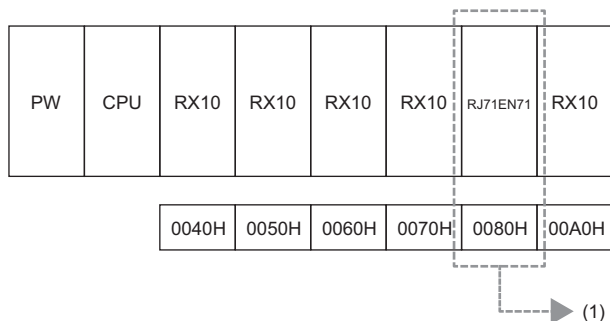
オペランド	内容	範囲	データ型	データ型(ラベル)
(H)	ユニット情報を読み出すユニットの先頭入出力番号(16進数4桁で表したときの3桁)	0H~FFH	符号なしBIN16ビット	ANY16
(d)	ユニット情報を格納する先頭デバイス	—	ワード	ANY16 ^{*1}
(n)	読出しデータ点数	0~256	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(H)	○	—	○	—	—	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(n)	○	—	○	—	—	—	—	○	○	—	—	—

- (H)で指定したユニットから、(n)で指定した点数分のユニット情報を、(d)で指定したデバイス以降に格納します。I/O割付でユニット種別と点数を変更しても、実装のユニット状態が読み出されます。
- (H)は、ユニット情報を読み出すユニットの先頭入出力番号を、16進数4桁で表したときの3桁で指定します。
- 読み出すユニットの先頭入出力番号を指定しなかった場合、ユニット装着状態のみON、ユニット装着状態以外がOFFのユニット情報を格納します。



(1): 読み出すユニットの先頭入出力番号はK8またはH8を指定

- ユニット情報の詳細は下記のとおりです。

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
(d)																
(d)+1																

デバイス	ビット	項目名	内容
(d)	b0	入出力点数	0000: 16点
	b1		0001: 32点
	b2		0010: 48点
	b3		0011: 64点 0100: 128点 0101: 256点 0110: 512点 0111: 1024点
	b4	ユニット種別	0000: 入力ユニット, または未設定
	b5		0010: 出力ユニット
	b6		0100: 入出力混合ユニット(別番号)
	b7		0110: 入出力混合ユニット(同一番号)
	b8	シリーズ種別	1000: インテリジェント機能ユニット
	b9		011: MELSEC iQ-Rシリーズのユニット
	b10		111: 不明
	b11	グループ番号	000: 1号機管理
	b12		
	b13		
	b14	スロットの先頭判定	0: スロットの先頭ではない 1: スロットの先頭である
	b15	ユニット装着状態	0: ユニット未装着 1: ユニットが装着されている*2
(d)+1	b0	エラー判定	00: 正常
	b1		01: 軽度異常 10: 中度異常 11: 重度異常
	b2	ユニット準備完了状態	0: 準備未完了 1: 準備完了
	b3	空き	0: 固定
	b4	空き	0: 固定
	b5	空き	0: 固定
	b6	空き	0: 固定
	b7		
	b8	空き	0: 固定
	b9	空き	0: 固定
	b10	外部供給電源状態	0: 正常運転中, または外部供給電源なし 1: 電源断発生
	b11	ヒューズ断発生有無	0: 正常運転中 1: ヒューズ断発生
	b12	空き	0: 固定
	b13	空き	0: 固定
	b14	空き	0: 固定
	b15	ユニットアクセス可否	0: アクセス不可*1 1: アクセス可

*1 ユニットの脱着などが該当します。

*2 32点以上占有するユニットの場合、(d)+0, (d)+1に情報が格納され、以降はユニット装着状態のみONの情報が格納されます。

- 下記のプログラム例では、M0がONしたとき、入出力番号が0080H以降のユニット情報をD0以降へ格納します。32点ユニットのユニット情報を読み出すため、(n)にはK2を指定します。

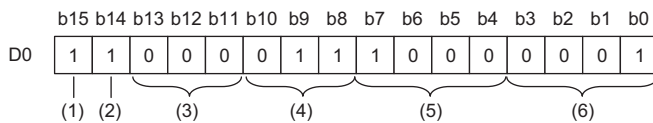
PW	CPU	RX10	RX10	RX10	RX10	RJ71EN71	RX10

0040H	0050H	0060H	0070H	0080H	00A0H

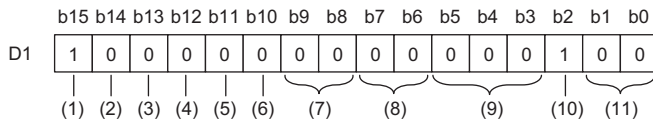


[読み出し結果]

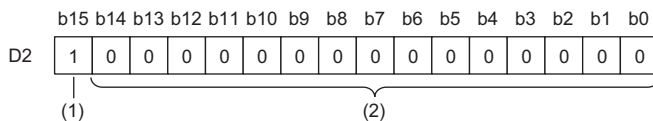
- RJ71EN71のユニット情報を読み出します。



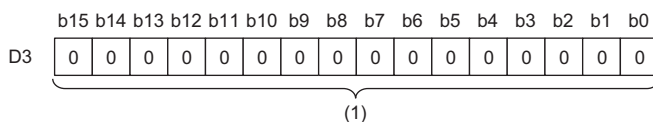
- (1) ユニットが装着されている
- (2) スロットの先頭である
- (3) 1号機管理
- (4) MELSEC iQ-Rシリーズのユニット
- (5) インテリジェント機能ユニット
- (6) 32点ユニット



- (1) ユニットアクセス可
- (2) 0固定
- (3) 0固定
- (4) 0固定
- (5) ヒューズ断エラーなし
- (6) 0固定
- (7) 0固定
- (8) 0固定
- (9) 0固定
- (10) ユニット準備完了
- (11) 正常(ユニットエラーなし)



- (1) 32点ユニットの後半16点としてユニットが接続されています。
- (2) D0, D1に情報が格納されるため、すべて0となります。



- (1) D0, D1に情報が格納されるため、すべて0となります。

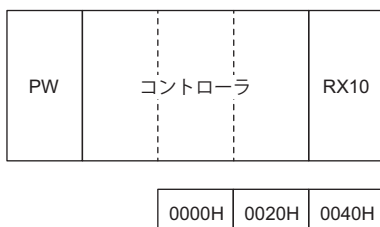
- ・コントローラのCC-Link IE TSN機能部・モーション機能部の先頭入出力番号を指定してユニット情報を読み出した場合は、下記の3項目をユニット情報に格納します。下記項目以外はOFFのユニット情報を格納します。

デバイス	ビット	項目名	内容
(d)	b0	入出力点数	0000: 16点
	b1		0001: 32点
	b2		0010: 48点
	b3		0011: 64点
			0100: 128点
	b14	スロットの先頭判定	0101: 256点
			0110: 512点
0111: 1024点			
b15	ユニット装着状態	0: スロットの先頭ではない 1: スロットの先頭である	
		0: ユニット未装着 1: ユニットが装着されている	

例

M0がONしたとき、入出力番号が0000H~003FHのユニット情報をD0以降へ格納します。各機能の入力点数はCC-Link IE TSN機能部が32点、モーション機能部が32点です。

CC-Link IE TSN機能部・モーション機能部の先頭入出力番号を0000Hに指定した場合



それぞれ下記を指します。

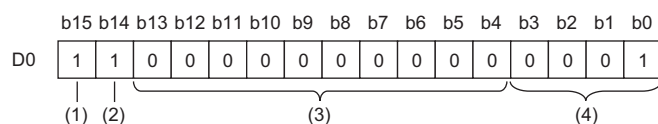
0000H: CC-Link TSN機能部

0020H: モーション機能部



[読み出し結果]

■[0000H: ネットワーク通信機能の先頭入出力番号]

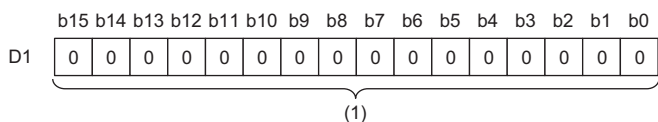


(1) ユニットが装着されている

(2) スロットの先頭である

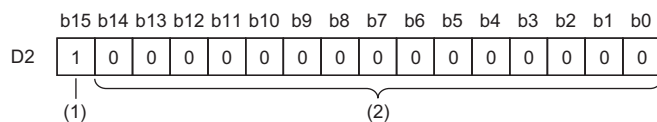
(3) すべて0になります。

(4) 32点ユニット

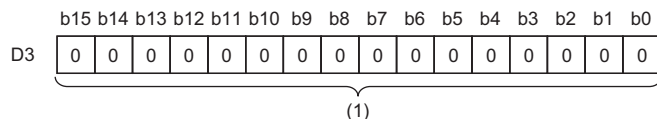


(1) すべて0になります。

■[0010H: ネットワーク通信機能の先頭でない入出力番号]

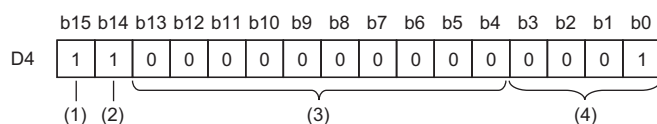


- (1) ユニットが装着されている
(2) すべて0になります。

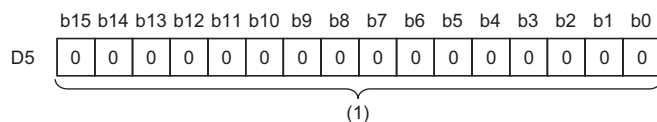


- (1) すべて0になります。

■[0020H: モーション機能部の先頭入出力番号]

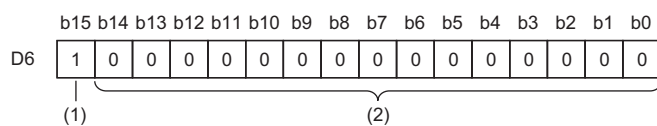


- (1) ユニットが装着されている
(2) スロットの先頭である
(3) すべて0になります。
(4) 32点ユニット

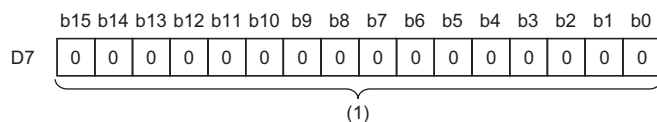


- (1) すべて0になります。

■[0030H: モーション機能部の先頭でない入出力番号]



- (1) ユニットが装着されている
(2) すべて0になります。



- (1) すべて0になります。

エラー

エラーコード (SD0)	内容
2800H	(H)が0~FFH以外るとき。
3285H	(n)が0~256以外るとき。
	(H)と(n)の合計が257以上のとき。

PID演算命令およびPID制御命令の使い分け

PID制御を行うための命令には、下記の種類があります。

種類	使用	参照先(概要)	参照先(命令詳細)
PID演算命令	オートチューニングを使用した制御を実施する場合	1031ページ 概要	1043ページ PID演算命令
PID制御命令	MELSEC-QシリーズやMELSEC-Lシリーズと同様のPID制御を実施する場合	1047ページ 概要	1056ページ PID制御命令(不完全微分), 1068ページ PID制御命令(完全微分)

PID演算命令とPID制御命令の比較を示します。

項目	PID演算命令	PID制御命令
PID演算方式	不完全微分	不完全微分/完全微分
サンプリング周期/サンプリングタイム	1~32767ms	10~60000ms
制御ループ数	1ループ/命令	最大32ループ
オートチューニング	あり(リミットサイクル法/ステップ応答法)	なし

23.1 概要

PID演算命令を使用した、PID制御の概要を説明します。

PID演算命令

PID演算命令は、目標値(SV)に近づけるために、測定値(PV)からP動作(比例動作)、I動作(積分動作)、D動作(微分動作)を組み合わせ、出力値(MV)を演算する命令です。

■警報出力機能

入力(測定値)変化量や出力(値)変化量について警報出力をONできます。

■出力値の上下限設定

出力の上下限值を設定し、PID制御の積分項の増大を抑えることができます。

■オートチューニング機能

比例ゲイン(K_p)、積分時間(T_i)、微分時間(T_D)を自動的に設定することができます。リミットサイクル法、またはステップ応答法が選択できます。

■PID演算命令の演算方式

速度形・測定値微分形で演算を行っています。

PID演算命令の基本演算式[参考]

本命令は、速度形・測定値微分形の演算式によりPID演算を行います。

(s3)+1(動作設定(ACT))のビット0の内容によって、正動作または逆動作の演算式が実行されます。

演算に必要とされている各値は、(s3)以降に指定した制御用データの内容で演算します。

・ 演算式

正動作/逆動作 ((s3)+1のビット0)	演算式
正動作 (OFF)	$\Delta MV = K_p \{ (EV_n - EV_{n-1}) + \frac{T_s}{T_i} EV_n + D_n \}$ $EV_n = PV_{nf} - SV$ $D_n = \frac{T_D}{T_s + K_D \cdot T_D} (-2PV_{nf-1} + PV_{nf} + PV_{nf-2}) + \frac{K_D \cdot T_D}{T_s + K_D \cdot T_D} \cdot D_{n-1}$ $MV_n = \Sigma \Delta MV$
逆動作 (ON)	$\Delta MV = K_p \{ (EV_n - EV_{n-1}) + \frac{T_s}{T_i} EV_n + D_n \}$ $EV_n = SV - PV_{nf}$ $D_n = \frac{T_D}{T_s + K_D \cdot T_D} (2PV_{nf-1} - PV_{nf} - PV_{nf-2}) + \frac{K_D \cdot T_D}{T_s + K_D \cdot T_D} \cdot D_{n-1}$ $MV_n = \Sigma \Delta MV$

演算式中の記号の意味は下記のとおりです。

記号	意味
EV_n	今回のサンプル時の偏差
EV_{n-1}	1周期前の偏差
SV	目標値
PV_{nf}	今回のサンプル時の測定値(フィルタ後)
PV_{nf-1}	1周期前の測定値(フィルタ後)
PV_{nf-2}	2周期前の測定値(フィルタ後)
ΔMV	出力変化量
MV_n	今回の操作量
D_n	今回の微分項
D_{n-1}	1周期前の微分項
K_p	比例ゲイン
T_s	サンプリング周期
T_i	積分定数
T_D	微分定数
K_D	微分ゲイン

今回のサンプル時の測定値(フィルタ後)(PV_{nf})は、読み込んだ測定値を下式で演算した値です。入力データのフィルタ係数が設定されていない場合は、入力データの測定値(PV)と同一の値となります。

$$PV_{nf} = PV_n + L(PV_{nf-1} - PV_n)$$

PV_{nf} : 今回のサンプル時の測定値, L : フィルタ係数, PV_{nf-1} : 1周期前の測定値(フィルタ後)

制御用データ

PID演算命令の制御用データの内容について説明します。

サンプリングタイム:(s3)

設定範囲: 1~32767[ms]

PID演算を行うための周期(ms)を設定します。

- ・オートチューニング(リミットサイクル法)時
(シーケンサの演算周期)<(サンプリングタイム)になるように設定します。
- ・オートチューニング(ステップ応答法)時
1000ms以上に設定します。

■最大誤差について

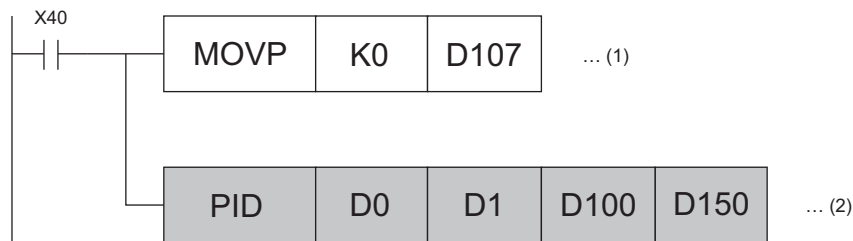
サンプリングタイム(T_S)の最大誤差は、 $-(1\text{演算周期}+1\text{ms})\sim+(1\text{演算周期})$ となります。

- ・サンプリングタイム(T_S)が小さな値の場合

最大誤差の変動が問題になることがあります。コンスタントスキャンを設定して実行するか、タイマ割込みルーチン内にプログラムしてください。

- ・シーケンサの1演算周期よりも短い場合

PID演算エラー (11A6H)が発生しますが、サンプリングタイム(T_S)=演算周期としてPID演算が実行されます。このような場合、PID演算命令をタイマ割込み内で使用し、PID演算命令の実行直前に(s3)+7をクリアしてから使用してください。



(1) (s3)+7をリセット(初回の割込みルーチン実行時に、内部処理用レジスタをパルス化指令でクリアします。)

(2) PID演算を実行します。

動作設定: (s3)+1

■動作方向(正動作/逆動作): (s3)+1 ビット0

設定範囲: OFF=正動作/ON=逆動作

正動作または逆動作を選択します。

- ・オートチューニング(リミットサイクル法)の場合

オートチューニングは、正動作または逆動作のPID制御したい方向を設定する必要があります。

- ・オートチューニング(ステップ応答法)の場合

オートチューニングは、正動作または逆動作のいずれかで実行しても、完了時に自動で設定が行われます。

[正動作((s3)+1 ビット0=OFF)]

[逆動作((s3)+1 ビット0=ON)]

■警報設定(入力変化量, 出力変化量): (s3)+1 ビット1, 2

設定範囲: OFF=警報なし/ON=警報有効

入力変化量, 出力変化量のチェックができます。チェックした結果は, (s3)+24で確認ができます。(1039ページ 警報出力のフラグ動作: (s3)+24)

- ・入力変化量((s3)+1 ビット1)

入力変化量警報を使用する場合, 下記のビットをONし, チェックしたい値を設定する必要があります。

設定項目			内容	設定範囲
動作設定(ACT)	(s3)+1	ビット1	入力変化量警報	ON: 使用する OFF: 使用しない
入力変化量警報設定値	(s3)+20		入力変化量(増側)警報設定値	0~32767
	(s3)+21		入力変化量(減側)警報設定値	

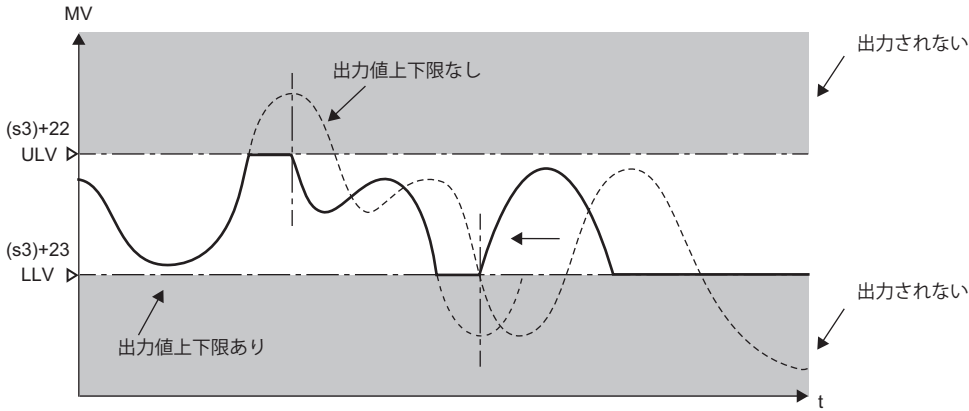
- ・出力変化量((s3)+1 ビット2)

出力変化量警報を使用する場合, 下記のビットをONし, チェックしたい値を設定する必要があります。

設定項目			内容	設定範囲
動作設定(ACT)	(s3)+1	ビット2	出力変化量警報	ON: 使用する OFF: 使用しない
		ビット5	出力値上下限設定	必ずOFFに設定
出力変化量警報設定値	(s3)+22		出力変化量(増側)警報設定値	0~32767
	(s3)+23		出力変化量(減側)警報設定値	

■出力値上下限設定: (s3)+1 ビット5

出力値上下限設定によって出力値は、下記ようになります。



MV: 出力値
ULV: 出力上限値
LLV: 出力下限値
t: 時間

出力値上下限設定は、PID制御の積分項の増大を抑える効果があります。本機能を使用する場合は、(s3)+1のビット2を必ずOFFに設定してください。

設定項目			内容	設定範囲
動作設定(ACT)	(s3)+1	ビット2	出力変化量警報	必ずOFFに設定
		ビット5	出力値上下限設定	ON: 使用する OFF: 使用しない

入力フィルタ: (s3)+2

設定範囲: 0~99[%]

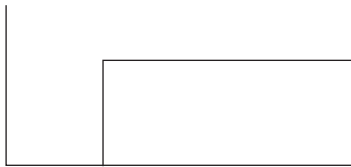
PID制御: 比例動作, 積分動作, 微分動作

入力フィルタ(α)は, 測定値(PV)のノイズによる変動を低減させるためのソフトウェアフィルタです。このフィルタの入力フィルタ(α)を制御対象の特性とそのノイズレベルにあわせて設定することによって, ノイズの影響を抑えることができます。

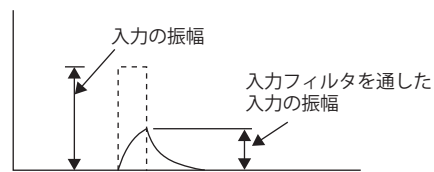
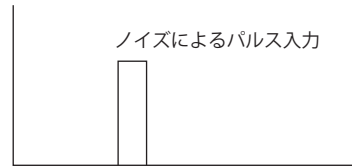
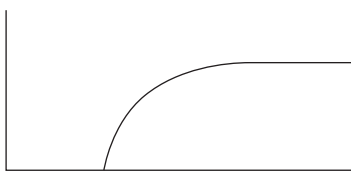
- 小さいと, フィルタとしての効果が少なくなります。
- 大きすぎると, 入力の応答が悪くなります。

入力フィルタ(α)は, 目標値(SV)に作用するため, 比例動作, 積分動作, 微分動作に影響します。

実際の測定値
(PV)



入力フィルタを
通した測定値
(PV)



比例ゲイン: (s3)+3

設定範囲: 1~32767[%]

PID制御: 比例動作

出力値(MV)は、比例動作で偏差(目標値(SV)と測定値(PV)の差)に比例して増加します。この割合を比例ゲイン(K_p)といい、下記の関係式で表現されます。

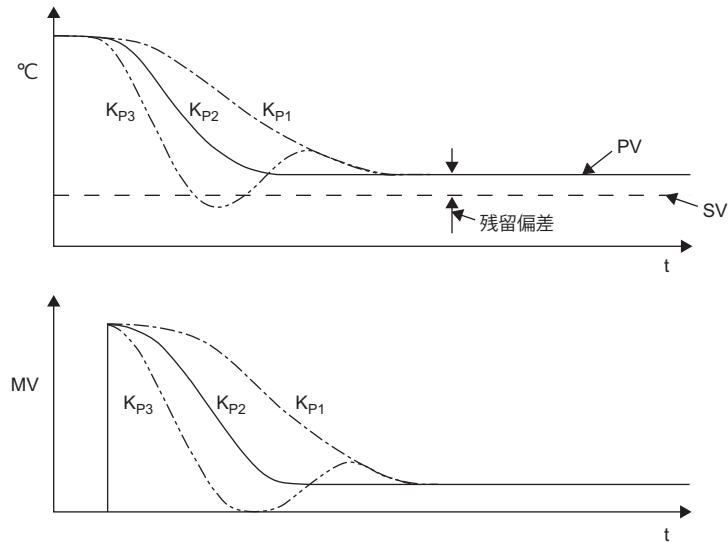
出力値(MV)=比例ゲイン(K_p)×偏差(EV)

また、比例ゲイン(K_p)の逆数を比例帯といいます。

比例ゲイン(K_p)が大きくなるにつれて、目標値(SV)に測定値(PV)を近づけようとする動きが強くなります。

例

冷房(正動作)の場合での比例動作(P動作)



比例ゲイン(K_p): $K_{p3} > K_{p2} > K_{p1}$

°C: 温度

SV: 目標値

PV: 測定値

MV: 出力

t: 時間

積分時間: (s3)+4

設定範囲: 0~32767[×100ms](0の場合は, ∞として扱う。(積分なし))

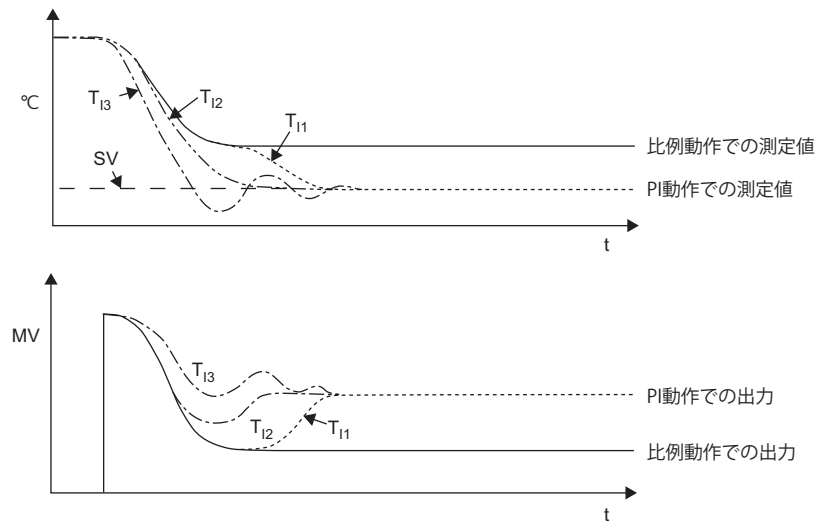
PID制御: 積分動作

積分動作で偏差が生じてから, 積分動作の出力が比例動作の出力になるまでの時間を積分時間といい, T_I で表します。

T_I を小さくすると, 積分動作が強くなります。

例

冷房(正動作)の場合でのPI動作



積分時間(T_I): $0 < T_{I3} < T_{I2} < T_{I1}$

°C: 温度

SV: 目標値

MV: 出力

t: 時間

微分ゲイン: (s3)+5

設定範囲: 0~200[%]

PID制御: 微分動作

微分動作による出力にフィルタをかけます。微分ゲイン(K_D)は, 微分動作のみに影響します。

- 微分ゲイン(K_D)を小さくすると, 外乱などによる測定値(PV)の変化に対し, 瞬時に応答します。
- 微分ゲイン(K_D)を大きくすると, 外乱などによる測定値(PV)の変化に対し, 長時間かけて応答します。

Point

最初は微分ゲイン(K_D)を0に設定し, 入力フィルタ(α)にて調整してください。出力の変化が外乱に対して応答が良すぎる場合は, 微分ゲイン(K_D)の値を大きくしてください。

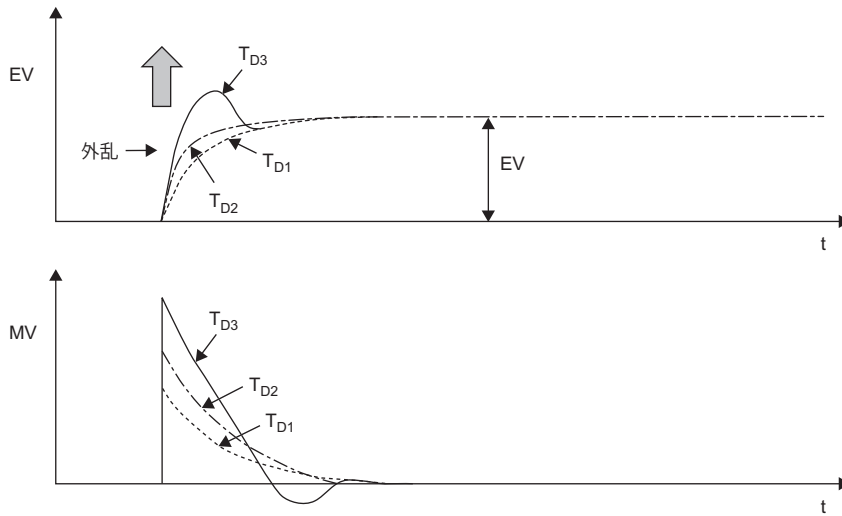
微分時間(T_D): (s3)+6

設定範囲: 0~32767[$\times 10\text{ms}$](0の場合は、微分なしとなる。)

PID制御: 微分動作

測定値(PV)の外乱などによる変動に対して、敏感に反応し変動を最小限に抑えるために使用します。

- ・ 微分時間(T_D)を大きくすると、外乱などで制御対象が大きく変動するのを防ぐ動きが強くなります。



微分時間(T_D): $T_{D3} > T_{D2} > T_{D1}$

EV: 偏差

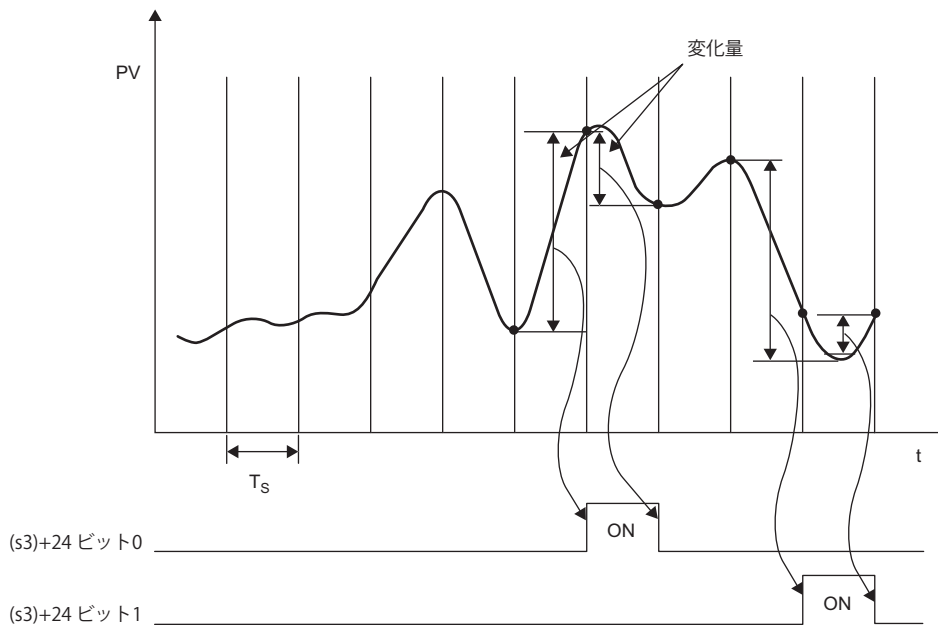
MV: 出力

t: 時間

警報出力のフラグ動作: (s3)+24

設定された入出力変化量を超えたとき、警報フラグとして(s3)+24の各ビットがPID演算命令実行直後にONします。

- ・ 入力変化量警報((s3)+1のビット1がON)の場合



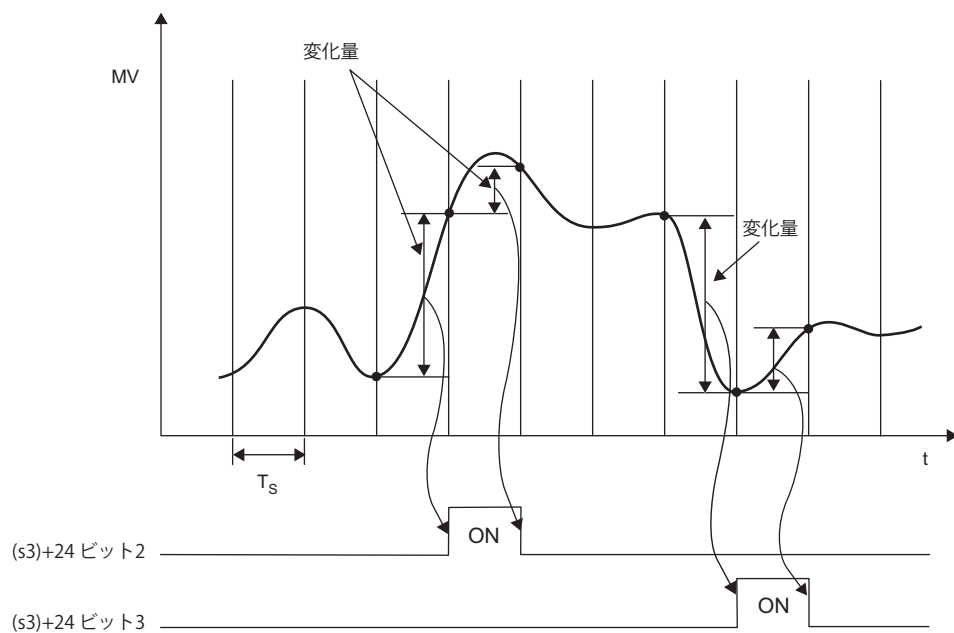
PV: 測定値

T_s : サンプルングタイム

t: 時間

(s3)+24 ビット0, ビット1: 警報フラグ

- 出力変化量警報((s3)+1のビット2がON)の場合



MV : 出力
 T_s : サンプルタイム
 t : 時間
 (s3)+24 ビット2, ビット3: 警報フラグ

オートチューニング

PID制御において良好な制御結果を得るためには、制御対象に合った各定数(制御用データ)の最適値を求める必要があります。オートチューニング機能は、PID制御を最適にするために重要な定数である比例ゲイン、積分時間、微分時間を自動的に設定するものです。

PID演算命令のオートチューニング機能は、リミットサイクル法とステップ応答法の2種類の方法があります。それぞれの手順を説明します。

Point

オートチューニングは、システムが安定した状態から開始してください。安定していない状態から開始すると、正しくオートチューニングできない場合があります。

リミットサイクル法

リミットサイクル法の概要は、下記を参照してください。

☞ 1934ページ リミットサイクル法の概要

■オートチューニング(リミットサイクル法)で設定される制御用データ

- 比例ゲイン(K_p): (s3)+3
- 積分時間(T_i): (s3)+4
- 微分時間(T_D): (s3)+6

■オートチューニング手順

1. 正動作/逆動作の設定

(s3)+1(動作設定(ACT))のビット0で、正動作か逆動作かを設定します。

2. リミットサイクル法の選択

(s3)+1(動作設定(ACT))のビット6をONにして、リミットサイクル法を選択します。
(OFF時は、ステップ応答法で動作します。)

3. オートチューニング実行フラグのON

(s3)+1(動作設定(ACT))のビット4をONします。

4. 入力フィルタの設定

(s3)+2(入力フィルタ(α))を設定します。

5. サンプリングタイムの設定

(s3)(サンプリングタイム(T_s))を設定します。

6. 出力値上限(ULV)の設定

(s3)+26(出力値上限(ULV))に、出力値(MV)の最大出力値を設定します。

7. 出力値下限(LLV)の設定

(s3)+27(出力値下限(LLV))に、出力値(MV)の最小出力値を設定します。

8. PV値スレッシュホールド(ヒステリシス)幅(SHPV)の設定

(s3)+25(PV値スレッシュホールド(ヒステリシス)幅(SHPV))を設定します。

9. 目標値(SV)の設定

PID演算命令の(s1)に目標値(SV)を設定します。

10. オートチューニングの開始

PID演算命令の起動接点をONすると、測定値(PV)により、オートチューニングが始まります。

チューニング完了時点で、(s3)+1(動作設定(ACT))のオートチューニングフラグ(ビット4, ビット6)がOFFします。

ステップ応答法

ステップ応答法の概要は、下記を参照してください。

📖 1935ページ ステップ応答法の概要

■オートチューニング(ステップ応答法)で設定される制御用データ

- ・動作方向(正動作/逆動作): (s3)+1 ビット0
- ・比例ゲイン(K_p): (s3)+3
- ・積分時間(T_i): (s3)+4
- ・微分時間(T_D): (s3)+6

■オートチューニング手順

1. オートチューニング用出力値の設定

出力値(MV)にオートチューニング用出力値を転送します。

オートチューニング用出力値は、出力機器に対して出力可能最大値 $\times 0.5 \sim 1$ の値に設定してください。

2. オートチューニングで設定されないデータの設定

オートチューニングで設定されない下記の項目をシステムに応じて設定します。

設定項目	備考
(s1)	目標値(SV)
(s3)	サンプリングタイム(T_S)
(s3)+2	入力フィルタ(α)
(s3)+5	微分ゲイン(K_D)
その他	

*1 目標値(SV)の設定値と測定値(PV)との差について

オートチューニング開始時の測定値と目標値との差が150以上ないと、正しくオートチューニングができません。そのため、150未満の場合は、オートチューニング用に目標値を設定してください。

オートチューニング完了後、目標値を再設定してください。

*2 サンプリングタイム(T_S)の設定時間について

オートチューニングする際のサンプリングタイム(T_S)は、必ず1000ms以上にしてください。

また、このサンプリングタイムは、出力変化周期よりも十分長い時間にしてください。

3. オートチューニングの開始

(s3)+1(動作設定(ACT))のビット4をONにすると、オートチューニングが開始されます。

オートチューニング開始時の測定値から目標値への変化量が1/3以上変化するとオートチューニングが完了し、(s3)+1(動作設定(ACT))のビット4が自動的にOFFになります。

オートチューニング時の注意事項

- ・測定値(PV)が変化しない場合のプログラム対策

アナログ入力断線などの要因により、測定値(PV)が正常に変化しない場合は、オートチューニングが終了しません。入力値やチューニング開始からの経過時間を監視するシーケンスを導入することによって検知し、回避させてください。

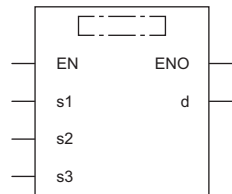
23.2 PID演算命令

PID

(s1), (s2), (s3)で設定した値からPID演算を行い、サンプリングタイムごとに演算結果を(d)に格納します。

ラダー	ST
	ENO:=PID(EN,s1,s2,s3,d);

FBD/LD



■実行条件

命令	実行条件
PID	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	目標値(SV)を格納するデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(s2)	測定値(PV)を格納するデバイス	-32768~32767	符号付きBIN16ビット	ANY16
(s3)	制御用データを格納する先頭デバイス	—	符号付きBIN16ビット	ANY16
(d)	出力値(MV)を格納するデバイス	-32768~32767	符号付きBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード		ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □, U3E□ ¥ (H)G□	Z	LT, LST, LC	LZ	K, H	E	\$	
(s1)	—	—	○	○	—	—	—	○	—	—	—
(s2)	—	—	○	○	—	—	—	○	—	—	—
(s3)	—	—	○	—	—	—	—	○	—	—	—
(d)	—	—	○	○	—	—	—	○	—	—	—

機能

目標値(s1), 測定値(s2), 制御用データ(s3)~(s3)+6を設定し、プログラムを実行すると、サンプリングタイム(s3)ごとに演算結果(MV)を出力値(d)に格納します。詳細は、下記を参照してください。

📖 1031ページ 概要

■各引数の設定項目

設定項目		内容	占有点数
(s1)	目標値(SV)	目標値(SV)を設定します。 ■オートチューニング: リミットサイクル法の場合 オートチューニング用の目標値とPID制御を行う際の目標値が異なった場合、バイアス値を加算した値を設定し、オートチューニングフラグがOFFになった時点で、実際の目標値を設定してください。	1点
(s2)	測定値(PV)	PID演算の入力値を設定します。	1点
(s3)	制御用データ*1	■オートチューニング: リミットサイクル法の場合 (s3)に指定した先頭デバイスから29点分のデバイスを占有します。	29点
		■オートチューニング: ステップ応答法の場合 (s3)に指定した先頭デバイスから25点分のデバイスを占有します。	25点
(d)	出力値(MV)	■PID制御(通常処理)の場合 命令実行前にユーザ側で初期出力値をセットします。命令実行後は演算結果が格納されます。 ■オートチューニング: リミットサイクル法の場合 オートチューニング中に自動的に出力上限値(ULV)または出力下限値(LLV)が出力され、オートチューニング終了後に所定の出力値(MV)が設定されます。 ■オートチューニング: ステップ応答法の場合 命令実行前にユーザ側でステップ出力値をセットします。オートチューニング中は、出力値(MV)はPID命令側で変更しません。	1点

*1 オートチューニングを使用しない場合、ステップ応答法を使用する場合と同じ点数を占有します。

■制御用データの設定項目

オペランド: (s3)				
デバイス	項目		内容	備考
+0	サンプリングタイム(T_S)		1~32767[ms]	演算周期より短い値での実行はできません。
+1	動作設定(ACT)	ビット0	0: 正動作 1: 逆動作	動作方向指定
		ビット1	0: 入力変化量警報なし 1: 入力変化量警報有効	—
		ビット2	0: 出力変化量警報なし 1: 出力変化量警報有効	ビット2とビット5は同時にONしないでください。
		ビット3	使用不可	—
		ビット4	0: オートチューニング不動作 1: オートチューニング実行	—
		ビット5	0: 出力値上下限設定なし 1: 出力値上下限設定有効	ビット2とビット5は同時にONしないでください。
		ビット6	0: ステップ応答法 1: リミットサイクル法	オートチューニングモード選択
		ビット7~ビット15	使用不可	—
+2	入力フィルタ定数(α)		0~99[%]	0を指定した場合、入力フィルタなしとなります。
+3	比例ゲイン(K_P)		1~32767[%]	—
+4	積分時間(T_I)		1~32767[$\times 100$ ms]	0を指定した場合、 ∞ として扱います。(積分なし)
+5	微分ゲイン(K_D)		0~200[%]	0を指定した場合、微分ゲインなしとなります。
+6	微分時間(T_D)		1~32767[$\times 10$ ms]	0を指定した場合、微分なしとなります。
+7~+19	PID演算の内部処理で占有するため、データ変更禁止			
+20 ^{*1}	入力変化量(増側)警報設定値		0~32767	動作設定(ACT): (s3)+1 ビット1=1のとき有効
+21 ^{*1}	入力変化量(減側)警報設定値		0~32767	動作設定(ACT): (s3)+1 ビット1=1のとき有効
+22 ^{*1}	出力変化量(増側)警報設定値		0~32767	動作設定(ACT): (s3)+1 ビット2=1, ビット5=0のとき有効
	出力上限設定値		-32768~32767	動作設定(ACT): (s3)+1 ビット2=0, ビット5=1のとき有効
+23 ^{*1}	出力変化量(減側)警報設定値		0~32767	動作設定(ACT): (s3)+1 ビット2=1, ビット5=0のとき有効
	出力下限設定値		-32768~32767	動作設定(ACT): (s3)+1 ビット2=0, ビット5=1のとき有効
+24 ^{*1}	警報出力	ビット0	0: 入力変化量(増側)未オーバ 1: 入力変化量(増側)オーバ	動作設定(ACT): (s3)+1 ビット1=1またはビット2=1のとき有効
		ビット1	0: 入力変化量(減側)未オーバ 1: 入力変化量(減側)オーバ	—
		ビット2	0: 出力変化量(増側)未オーバ 1: 出力変化量(増側)オーバ	—
		ビット3	0: 出力変化量(減側)未オーバ 1: 出力変化量(減側)オーバ	—
+25 ^{*2}	PV値スレッシュホールド(ヒステリシス)幅(SHPV)		測定値(PV)のふらつきに応じて設定	動作設定(ACT): (s3)+1 ビット6=1(リミットサイクル法を選択)のとき占有
+26 ^{*2}	出力値上限(ULV)		出力値(MV)の最大出力値(ULV)設定	
+27 ^{*2}	出力値下限(LLV)		出力値(MV)の最小出力値(LLV)設定	
+28 ^{*2}	チューニングサイクル終了からPID制御開始までのウェイト設定パラメータ(K_W)		-50~32717[%]	

*1 (s3)+1の動作設定(ACT)のビット1=1, ビット2=1またはビット5=1のときに占有されます。

*2 (s3)+1の動作設定(ACT)のビット6=1のとき(リミットサイクル法の選択時)に占有されます。

エラー

エラーコード (SD0)	内容
11A0H	サンプリングタイム(T_S)に範囲外($T_S \leq 0$)の値を指定したとき。
11A1H	入力フィルタ定数(α)に範囲外($\alpha < 0$ または $100 \leq \alpha$)の値を指定したとき。
11A2H	比例ゲイン(K_P)に範囲外($K_P < 0$)の値を指定したとき。
11A3H	積分時間(T_I)に範囲外($T_I < 0$)の値を指定したとき。
11A4H	微分ゲイン(K_D)に範囲外($K_D < 0$ または $201 \leq K_D$)の値を指定したとき。
11A5H	微分時間(T_D)に範囲外($T_D < 0$)の値を指定したとき。
11A6H	サンプリングタイム(T_S)に演算周期以下の値を指定したとき。
11A7H	測定値変化量(ΔPV)がオーバーフローしたとき。
11A8H	偏差(EV)がオーバーフローしたとき。
11A9H	積分計算値がオーバーフローしたとき。
11AAH	微分ゲイン(K_D)の値がオーバーフローしたとき。
11ABH	微分計算値がオーバーフローしたとき。
11ACH	PID演算結果がオーバーフローしたとき。
11ADH	出力上限設定値に出力下限設定値より小さな値を指定したとき。
11AEH	入力変化量警報設定値または出力変化量警報設定値に0より小さな値を指定したとき。
11AFH	■ステップ応答法 オートチューニング結果不良のとき。 ・オートチューニング(ステップ応答法)開始時の偏差が150以下になった。 ・オートチューニング(ステップ応答法)終了時の偏差が開始時の偏差の1/3以上で終了した。
11B0H	■ステップ応答法 オートチューニング動作方向不一致のとき。 ・オートチューニング(ステップ応答法)開始時の目標値と測定値の関係から推測する動作方向と、出力値の動作方向が不一致になった。
11B1H	■ステップ応答法 オートチューニング動作不良のとき。 ・入力値(PV)が正常に変化せず、オートチューニング(ステップ応答法)が正しく動作しなかった。
11B2H	■リミットサイクル法 オートチューニング(リミットサイクル法)用の出力値上限(ULV)に出力値下限(LLV)以下の値を指定したとき。
11B3H	■リミットサイクル法 オートチューニング(リミットサイクル法)用のPV値スレッシュホールド(ヒステリシス)幅(SHPV)に範囲外($SHPV < 0$)の値を指定したとき。
11B4H	■リミットサイクル法 オートチューニング(リミットサイクル法)で使用するシステムエリアが書き換えられたとき。
11B5H	■リミットサイクル法 オートチューニング(リミットサイクル法)の測定時間が超過したことで、 τ および τ_{on} の時間が正常に取得できなかったとき。 ($\tau_{on} > \tau$, $\tau_{on} < 0$, $\tau < 0$)
11B6H	■リミットサイクル法 オートチューニング(リミットサイクル法)で算出された比例ゲイン(K_P)がオーバーフローしたとき。
11B7H	■リミットサイクル法 オートチューニング(リミットサイクル法)で算出された積分時間(T_I)が0~32767以外の結果になったとき。
11B8H	■リミットサイクル法 オートチューニング(リミットサイクル法)で算出された微分時間(T_D)が0~32767以外の結果になったとき。

PID制御を行うための命令です。

- PID演算命令
- PID制御命令

それぞれの使い分けや比較については、下記を参照してください。

📖 1031ページ PID演算命令およびPID制御命令の使い分け

24.1 概要

PID制御命令における演算方式，PID制御手順，その他の機能について説明します。

Point

PID制御命令には，不完全微分と完全微分それぞれの命令があります。

不完全微分は，微分項の入力に一次遅れフィルタを入れたPID制御で，下記のような場合に使用すると有効です。

- 高周波ノイズの影響を受けやすい制御
- 完全微分方式でステップ状変化があったときに，操作端を動作させるだけの有効なエネルギーが与えられないとき

完全微分は，微分項の入力をそのまま使用するPID制御です。

演算方式

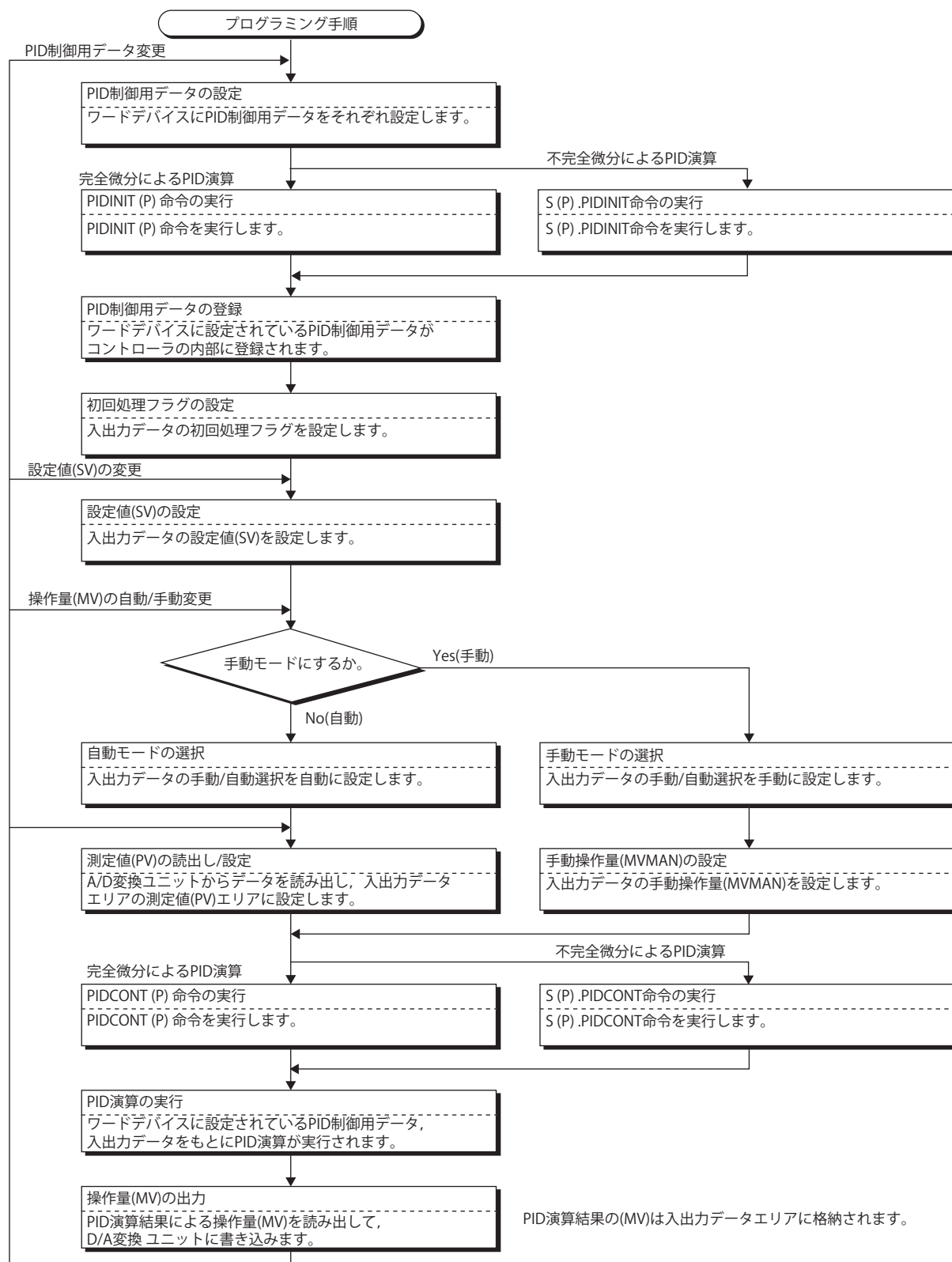
PID制御命令のPID制御における演算方式は速度形・測定値微分形です。

速度形・測定値微分形では下記の制御を行います。

演算方式	内容
速度形演算	PID演算で操作量(MV)の変化分を計算する方式です。実際の操作量はサンプリング周期ごとに計算した操作量の変化分が累積された値となります。
測定値微分形	PID演算で微分項に測定値(PV)を使用して演算する方式です。微分項に偏差を使用していないため，設定値変更による偏差の変化時に微分動作による出力の急変を軽減できます。

PID制御手順

PID制御を行うためのデバイスを使用したプログラミング手順について示します。



- PID制御用データは、プログラムのスキャンごとに登録または変更できます。ただし、登録または変更した場合、必ずPIDINIT命令を実行してください。PIDINIT命令を実行しないと、登録または変更したデータがPIDCONT命令実行時に反映されません。
- PIDPRMW命令で1ループごとにPID制御用データを変更する場合は、PIDINIT命令を実行する必要はありません。

PID制御用データ

PID演算における基準値を設定するためのデータです。PIDCONT命令によるPID演算を開始する前に、PIDINIT命令でコントローラ内部に登録します。また、ワードデバイスの任意の番号に設定できます。ただし、使用ループ分のすべてのデータを連続したデバイス番号に設定する必要があります。

PID制御データの割付けは、下記を参照してください。

- 不完全微分: 1058ページ S(P).PIDINIT
- 完全微分: 1070ページ PIDINIT(P)

PID制御用データには、全ループ共通で設定するデータと、各ループに設定するデータの2種類があります。

データの種類	項目	内容	設定範囲	設定データが指定範囲外のと きの処理
共通設定データ	使用ループ数	PID演算を実行させるループ数の設定です。	1~32	エラーとなり、全ループPID演算を 実行しません。
	1スキャンの実行ループ数	サンプリング周期に達したループが複数あったとき、1回のPID演算で何ループを実行させるかの設定です。	1~32	
各ループの設定データ	演算式選択	PID演算式の正動作か逆動作かの選択です。	0: 正動作 1: 逆動作	エラーとなり、該当ループのPID演算 を実行しません。
	サンプリング周期(T_S)	PID演算を行う周期の設定です。	1~6000(単位: 10ms)	
	比例定数(K_P)	PID演算の比です。	1~10000(単位: 0.01)	
	積分定数(T_I)	積分動作(D動作)の効果の大きさを表す定数です。積分定数を大きくすると、操作量の変化がゆるやかになります。	1~32767(単位: 100ms)	
	微分定数(T_D)	微分動作(D動作)の効果の大きさを表す定数です。微分定数を大きくすると、制御対象のわずかな変化で大きな操作量の変化になります。	0~30000(単位: 10ms)	
	フィルタ係数(α)	測定値(A/D変換ユニットからの入力)に対してフィルタをどの程度かけるかの設定です。0に近くなるほどフィルタは効かなくなります。	1~100	
	操作量下限値(MVLL)	自動モード時、PID演算で算出した操作量の下限値の設定です。操作量が操作量下限値(MVLL)未満のときは、操作量下限値(MVLL)が操作量になります。	-50~2050 ^{*1} -32768~32767 ^{*2}	
	操作量上限値(MVHL)	自動モード時、PID演算で算出した操作量の上限値の設定です。操作量が操作量上限値(MVHL)を超えるときは、操作量上限値(MVHL)が操作量になります。	-50~2050 ^{*1} -32768~32767 ^{*2}	
	操作量変化率リミット値(ΔMVL)	前回と今回の操作量における変化量の制限値の設定です。操作量の変化量が制限値を超えるときは、アラーム用デバイスのb1が1となります。 ただし、操作量の変化量は制限しません。操作量の変化量が制限値を超える場合でも、そのまま操作量の変化量として使用し、操作量を算出します。	0~2000 ^{*1} 0~32767 ^{*2}	
	測定値変化率リミット値(ΔPVL)	前回と今回の測定値における変化量の制限値の設定です。測定値の変化量が制限値を超えるときは、アラーム用デバイスのb0が1となります。 ただし、測定値の変化量は制限しません。測定値の変化量が制限値を超える場合でも、そのまま測定値の変化量として使用し、PID演算を行います。	0~2000 ^{*1} 0~32767 ^{*2}	
	微分ゲイン(K_D) ^{*3}	微分動作に対して時間幅(動作遅れ)を持たせる設定です。値が大きいほど時間幅は小さくなり、完全微分に動作が近づきます。	0~32767(単位: 0.01)	エラーとなり、該当ループのPID演算 を実行しません。

*1 "PIDリミット制限あり"設定時です。

*2 "PIDリミット制限なし"設定時です。

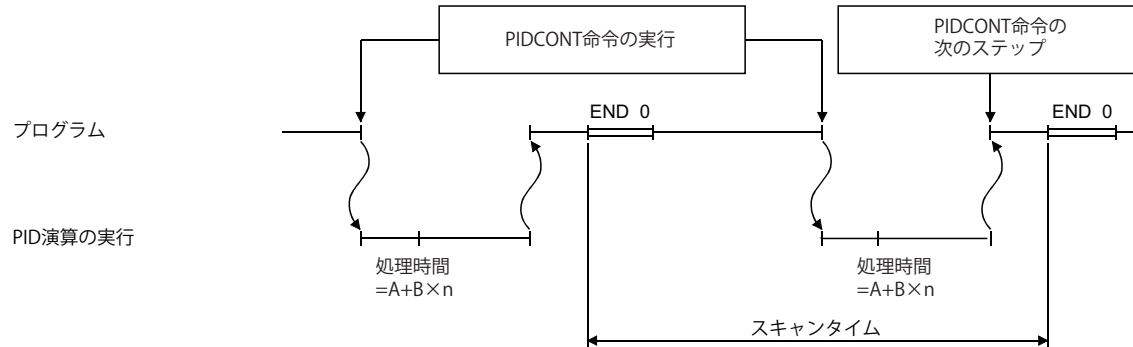
*3 不完全微分時のみです。

■使用ループ数と1スキンの実行ループ数

使用ループ数とは、PID演算を実行させるループ数です。PIDCONT命令実行時に設定されたループ数分のサンプリング周期の計測を行い、サンプリング周期に達したループのPID演算を行います。

PIDCONT命令を実行した場合の処理時間は、PID演算を実行するループ数に比例して増えていきます。

1スキンの実行ループ数は、サンプリング周期に達したループが複数あったとき、1スキンで何ループのPID演算を行うかの設定です。1スキンの実行ループ数を指定しておくと、PIDCONT命令実行時にサンプリング周期に達しているループ数が多い場合でも、設定された1スキンの実行ループ数分のみを実行し、残りを次スキンで実行します。



A: サンプリング周期の計測などの固定時間

B: ループのPID演算を行うための時間

n: ループ数

Point

サンプリング周期に達したループが、1スキンの実行ループ数より多い場合の優先順位は、下記ようになります。

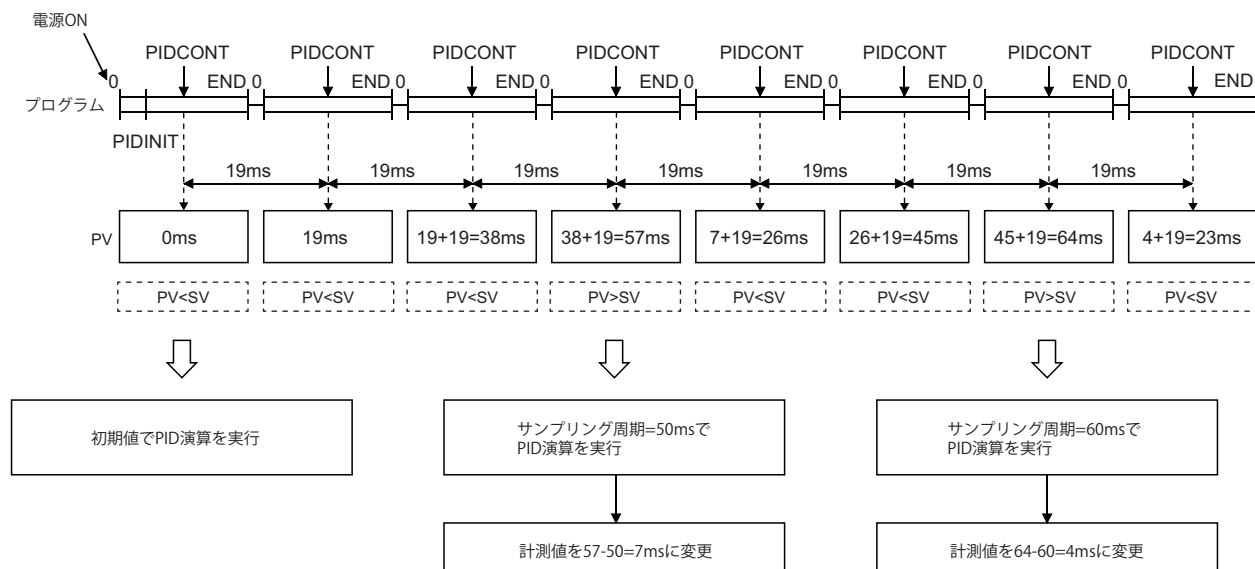
- 優先順位はループNo.の若い番号が高くなっています。
- 前のスキンでPID演算を実行していないループと実行したループがある場合は、前のスキンでPID演算を実行しなかったループを優先します。

■サンプリング周期

サンプリング周期は、PID演算を行う周期です。PIDCONT命令実行ごとに1スキャン分の計測時間を前回までの計測時間に加算します。加算された値が設定されたサンプリング周期以上になったとき、該当ループのPID演算を行います。PID演算で使用するためのサンプリング周期は、10ms単位の実行値を使用します。

例

サンプリング周期の設定値が50msの場合



PV: 計測値

SV: 設定値

制約事項

サンプリング周期の計測は、PIDCONT命令実行時に行うため、プログラムのスキャンタイムよりも小さい値をサンプリング周期には設定できません。スキャンタイムよりも小さい値を設定した場合は、スキャンタイム値でPID演算します。

入出力データ

入出力データとは、設定値(SV)、測定値(PV)などPID演算を行うために設定する入力データと、演算結果などの出力データです。また、ワードデバイスの任意の番号を指定できます。ただし、使用ループ分のすべてのデータを連続したデバイス番号に設定する必要があります。

入出力データの割付けは、下記を参照してください。

- 不完全微分: 1061ページ S(P).PIDCONT
- 完全微分: 1072ページ PIDCONT(P)

入出力データエリアには、各ループに割り付けられている下記の項目と、PID演算を行うためにシステムで使用するワークエリアがあります。

項目		内容	設定範囲	備考
設定値	SV	PID制御の目標値です。	0~2000 ^{*1} -32768~32767 ^{*2}	“PIDリミット制限あり”では、設定値(SV)の値が指定範囲外の場合は、下記の値に変換されます。 • 0未満のときは、0になります。 • 2000を超えるときは、2000になります。
測定値	PV	制御対象からA/D変換ユニットにフィードバックされたデータです。	-50~2050 ^{*1} -32768~32767 ^{*2}	“PIDリミット制限あり”では、測定値(PV)の値が指定範囲外の場合は、下記の値に変換されます。 • -50未満のときは、-50になります。 • 2050を超えるときは、2050になります。
自動操作量	MV	PID演算で算出した操作量です。D/A変換ユニットから制御対象へ出力します。	-50~2050 ^{*1} -32768~32767 ^{*2}	—
フィルタ後の測定値	PVf	下記の演算式で算出した測定値です。 $PV_{fn}=PV_n+\alpha(PV_{fn-1}-PV_n)$	-50~2050 ^{*1} -32768~32767 ^{*2}	—
手動操作量	MV _{MAN}	手動時にD/A変換ユニットから出力するデータを格納します。	-50~2050 ^{*1} -32768~32767 ^{*2}	“PIDリミット制限あり”では、手動操作量(MV _{MAN})の値が設定範囲外の場合は、下記の値に変換されます。 • -50未満のときは、-50になります。 • 2050を超えるときは、2050になります。
手動/自動選択	MAN/ AUTO	D/A変換ユニットへの出力データを手動操作量にするか、自動操作量にするかを選択します。手動時には、自動操作量は変化しません。	0: 自動操作量 1: 手動操作量	0, 1以外のときはエラーとなり、該当ループのPID演算は実行されません。
アラーム	ALARM	自動操作量(MV)、測定値(PV)の変化率がリミット値の範囲内か範囲外かの判別用です。1度セットすると、リセットするまで保持されます。  (2) (1) (1) 測定値(PV)がリミット範囲外の場合、b0が1になります。 (2) 自動操作量(MV)がリミット範囲外の場合、b1が1になります。	—	—

*1 “PIDリミット制限あり”設定時です。

*2 “PIDリミット制限なし”設定時です。

その他の機能

PID制御命令によるPID制御では、下記に示すバンプレス切換え、操作量上下限リミッタ制御を自動的に行います。

バンプレス切換え

バンプレスとは、モード切換え(手動 $\leftrightarrow$ 自動)時に操作量(MV)を連続的に制御させる機能です。モード切換えを行うと、下記のように自動モードの操作量エリアと手動モードの操作量エリア間でデータを転送します。

- 手動モードから自動モードへの切換え時: 手動モードの操作量を自動モードの操作量エリアに転送します。
- 自動モードから手動モードへの切換え時: 自動モードの操作量を手動モードの操作量エリアに転送します。

モード切換えは、入出力データエリアで行います。(1053ページ 入出力データ)

Point

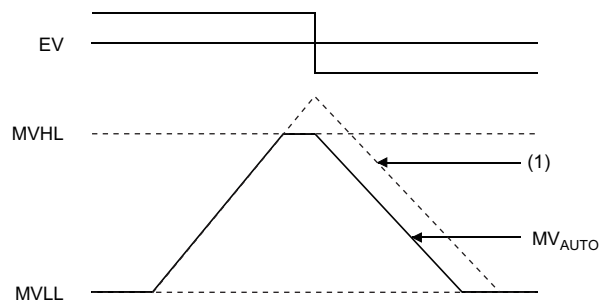
PID制御の自動モード、手動モードは下記のようになっています。

- 自動モード: PID制御命令でPID演算を行って算出した操作量により、制御対象の制御を行うモードです。
- 手動モード: PID制御命令でPID演算を行わずに算出した操作量により、制御対象の制御を行うモードです。
手動モードに設定されているループは、サンプリング周期ごとに測定値(PV)を設定値エリアに格納します。

操作量上下限リミッタ制御

操作量上下限リミッタ制御とは、PID演算で算出した操作量の上限または下限を制限する機能です。この機能は自動モード時のみ有効で、手動モード時は実行されません。

操作量上限値(MVHL)、操作量下限値(MVLL)を設定することで、PID演算で算出した操作量を下限値から上限値の範囲内に制限できます。操作量上下限リミッタ機能では、下記の動作になります。



(1): リミッタ制御がかからなかった場合の MV_{AUTO}

操作量上限値(MVHL)、操作量下限値(MVLL)は、各ループに対して-50~2050または任意の範囲が設定できます。デフォルト値は下記になります。

- 操作量上限値: 2000
- 操作量下限値: 0

操作量上限値(MVHL)、操作量下限値(MVLL)は、(上限値)<(下限値)と設定するとエラーになります。

手動モード時の測定値の設定値への転送機能

手動モード時もPIDCONT命令を実行します。手動モード時には、SM792(PIDバンプレス処理(完全微分用)), SM794(PIDバンプレス処理(不完全微分用))のON/OFFにより、PIDCONT命令実行時にA/D変換ユニットから取り込んだ測定値を設定値に転送するか選択できます。デフォルトはOFFです。なお、設定値および測定値は、PIDCONT命令で入出力データエリアに指定したデバイスまたはラベルに格納されます。

SM792/SM794	動作内容
OFF	• PIDCONT命令実行時に、測定値を設定値に転送します。 • 手動モード→自動モードに切換え時、手動モード時の操作量出力を継続できます。 • 自動モードに切り換わった後に設定値を変更すると、出力していた操作量から設定値への制御ができます。
ON	• PIDCONT命令実行時に、測定値を設定値に転送しません。 • 手動モード→自動モードに切り換えると、手動モード時の操作量出力から設定値への制御ができます。 • 自動モードへの切換え前に、設定値を格納してください。

Point

SM792(PIDバンプレス処理(完全微分用)), SM794(PIDバンプレス処理(不完全微分用))のON/OFFにより、手動モード→自動モード切換え時には、下記の違いがあります。

- OFF時は、測定値を設定値に転送するので、モード切換え時に測定値と設定値に差がなく、操作量の急激な変化が発生しません。一方、切換え後の設定値は、自動モードでの目標となる値と異なっているため、設定値を目標となる値までプログラムにより段階的に変化させてください。
- ON時は、測定値を設定値に転送しないので、モード切換え時に測定値と設定値に差があります。切換え時にその差が大きい場合、操作量の急激な変化が発生することがあるため、測定値が設定値に十分に近づいたときに切り換えるようなシステムで使用してください。プログラムにより設定値を段階的に変化させる必要がなく、すぐに自動モードで実施できます。

PID制御用データ、入出力データの設定範囲の変更機能

PID制御用データ、入出力データの設定範囲を設定により任意の範囲に変更できます。

設定を有効にする場合には、SD792~SD793(PIDリミット制限設定(完全微分用)), SD794~SD795(PIDリミット制限設定(不完全微分用))の該当ループに対するビットをONしてください。

不完全微分	完全微分	動作内容																
SD794	SD792	b15b14b13b12b11b10b9b8b7b6b5b4b3b2b1b0 <table><tr><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td></tr></table> b0: ループ1 b1: ループ2 ⋮ b14: ループ15 b15: ループ16	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0			
SD795	SD793	b15b14b13b12b11b10b9b8b7b6b5b4b3b2b1b0 <table><tr><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td><td>1/0</td></tr></table> b0: ループ17 b1: ループ18 ⋮ b14: ループ31 b15: ループ32	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0			

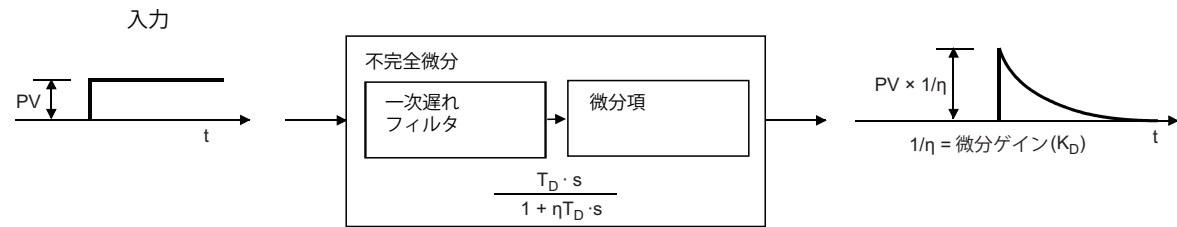
0: PIDリミット制限あり(デフォルト)

1: PIDリミット制限なし

24.2 PID制御命令(不完全微分)

不完全微分は、微分項の入力に一次遅れフィルタを入れたPID制御です。不完全微分は下記の場合に有効です。

- ・高周波ノイズの影響を受けやすい制御の場合
- ・完全微分方式で、ステップ状変化があったときに操作端を動作させるだけの有効なエネルギーが与えられない場合

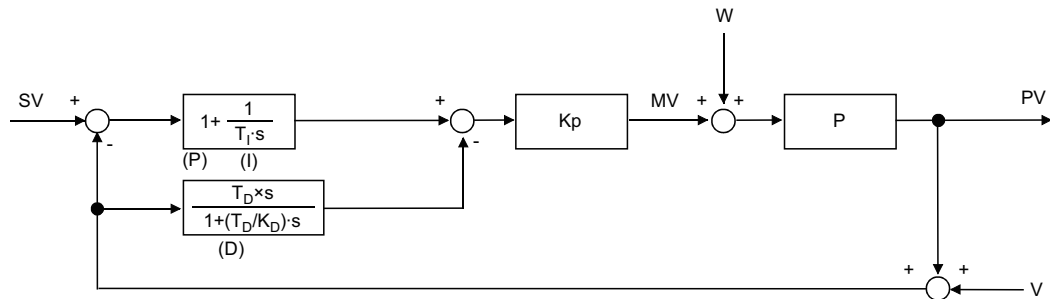


不完全微分によるPID制御命令の性能仕様を示します。

項目			"PIDリミット制限あり"時	"PIDリミット制限なし"時
PID制御ループ数			—	最大32ループ
サンプリング周期			T_S	0.01~60.00s
PID演算方式			—	測定値微分形不完全微分(正動作/逆動作)
PID定数設定範囲	比例定数	K_P	0.01~100.00	
	積分定数	T_I	0.1~3000.0s	
	微分定数	T_D	0.00~300.00s	
	微分ゲイン	K_D	0.00~300.00	
設定値設定範囲			SV	0~2000
測定値設定範囲			PV	-50~2050
操作量出力範囲			MV	-32768~32767

PID演算のブロック図と演算式を示します。

- ・不完全微分のPID演算ブロック



K_P : ゲイン
W: 外乱
P: 制御対象
V: 検出雑音

・ 演算式

正動作/逆動作	演算式
正動作	$EV_n = PV_{fn} - SV$ $\Delta MV = K_p \{ (EV_n - EV_{n-1}) + \frac{T_s}{T_i} \cdot EV_n + D_n \}$ $D_n = \frac{T_D}{T_s + \frac{T_D}{K_D}} (PV_{fn} - 2PV_{fn-1} + PV_{fn-2}) + \frac{\frac{T_D}{K_D}}{T_s + \frac{T_D}{K_D}} \cdot D_{n-1}$ $MV_n = \Sigma \Delta MV$
逆動作	$EV_n = SV - PV_{fn}$ $\Delta MV = K_p \{ (EV_n - EV_{n-1}) + \frac{T_s}{T_i} \cdot EV_n + D_n \}$ $D_n = \frac{T_D}{T_s + \frac{T_D}{K_D}} (-PV_{fn} + 2PV_{fn-1} - PV_{fn-2}) + \frac{\frac{T_D}{K_D}}{T_s + \frac{T_D}{K_D}} \cdot D_{n-1}$ $MV_n = \Sigma \Delta MV$

演算式中の記号の意味は下記のとおりです。

記号	意味
EV_n	今回のサンプル時の偏差
EV_{n-1}	1周期前の偏差
SV	設定値
PV_{fn}	今回のサンプル時の測定値(フィルタ後)
PV_{fn-1}	1周期前の測定値(フィルタ後)
PV_{fn-2}	2周期前の測定値(フィルタ後)
ΔMV	出力変化量
MV_n	今回の操作量
D_n	今回の微分項
D_{n-1}	1周期前の微分項
T_s	サンプリング周期
K_p	比例定数
T_i	積分定数
T_D	微分定数
K_D	微分ゲイン

今回のサンプル時の測定値(フィルタ後)(PV_{fn})は、入力データの測定値を下式で演算した値です。入力データのフィルタ係数が設定されていない場合は、入力データの測定値(PV)と同一の値となります。

$$PV_{fn} = PV_n + \alpha(PV_{fn-1} - PV_n)$$

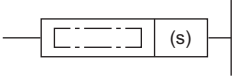
PV_n : 今回のサンプル時の測定値, α : フィルタ係数, PV_{fn-1} : 1周期前の測定値(フィルタ後)

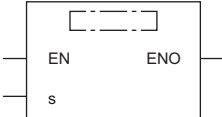
なお, PV_{fn} は入出力データエリアに格納されます。(1053ページ 入出力データ)

PID制御用データの設定

S(P).PIDINIT

指定したデバイス番号以降に設定されている使用ループ数分のPID制御データを、一括でコントローラ内部に登録します。

ラダー	ST
	ENO:=S_PIDINIT(EN,s); ENO:=SP_PIDINIT(EN,s);

FBD/LD


■実行条件

命令	実行条件
S.PIDINIT	
SP.PIDINIT	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	PID制御用データが設定されている先頭デバイス	—	ワード	ANY16 ^{*1}
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は，動作に必要な領域が確保できるよう配列を定義し，その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—

機能

- (s)で指定したデバイス番号以降に設定されている使用ループ数分のPID制御用データを一括でコントローラ内部に登録し、PID制御を可能にします。(1050ページ PID制御用データ)
- PID制御用データは、下記のように割り付けられています。

項目	PID制御用データ	
全ループ共通	(s)+0	使用ループ数
	(s)+1	1スキャンの実行ループ数
No.1ループ用(14ワード)	(s)+2	演算式選択
	(s)+3	サンプリング周期(T_S)
	(s)+4	比例定数(K_P)
	(s)+5	積分定数(T_I)
	(s)+6	微分定数(T_D)
	(s)+7	フィルタ係数(α)
	(s)+8	操作量下限値(MVLL)
	(s)+9	操作量上限値(MVHL)
	(s)+10	操作量変化率リミット値(ΔMVL)
	(s)+11	測定値変化率リミット値(ΔPVL)
	(s)+12	0固定(0以外を指定した場合はエラー)
	(s)+13	微分ゲイン(K_D)
	(s)+14	0固定(0以外を指定した場合はエラー)
	(s)+15	0固定(0以外を指定した場合はエラー)
⋮	⋮	⋮
No.nループ用(14ワード)*1	(s)+(m+0)	演算式選択
	(s)+(m+1)	サンプリング周期(T_S)
	(s)+(m+2)	比例定数(K_P)
	(s)+(m+3)	積分定数(T_I)
	(s)+(m+4)	微分定数(T_D)
	(s)+(m+5)	フィルタ係数(α)
	(s)+(m+6)	操作量下限値(MVLL)
	(s)+(m+7)	操作量上限値(MVHL)
	(s)+(m+8)	操作量変化率リミット値(ΔMVL)
	(s)+(m+9)	測定値変化率リミット値(ΔPVL)
	(s)+(m+10)	0固定(0以外を指定した場合はエラー)
	(s)+(m+11)	微分ゲイン(K_D)
	(s)+(m+12)	0固定(0以外を指定した場合はエラー)
	(s)+(m+13)	0固定(0以外を指定した場合はエラー)

*1 $m=(n-1) \times 14 + 2$

(1)「0」固定になります。「0」以外を指定した場合は、エラーになります。

- PID制御用データの設定で使用するデバイスの点数は、下式により算出します。

デバイス点数 = $2 + 14 \times n$ (n: 使用ループ数)

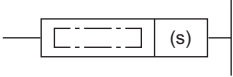
- 各データはBIN値で設定してください。
- 使用ループ分のデバイス点数が、指定したデバイスの最終デバイス番号を超えるとエラーとなり処理しません。
- 1スキャン中に複数箇所S(P).PIDINIT命令を実行した場合は、S(P).PIDCONT命令に1番近くで実行したS(P).PIDINIT命令の設定値が有効になります。
- S(P).PIDINIT命令の実行は、必ずS(P).PIDCONT命令実行前に行ってください。S(P).PIDINIT命令を実行していない場合、PID制御を行うことはできません。

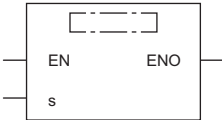
エラー

エラーコード (SD0)	内容
3285H	(s)に指定可能な範囲外のデータを設定したとき。 • PID制御用データに設定した値が、設定可能範囲外の時。 • (使用ループ数)<(1スキャンの実行ループ数)の時。 • (操作量上限値)<(操作量下限値)の時。 • PID制御用データの「0」固定のエリアが「0」でない時。

S(P).PIDCONT

実行指令のONで、サンプリング周期の計測とPID演算を行います。

ラダー	ST
	ENO:=S_PIDCONT(EN,s); ENO:=SP_PIDCONT(EN,s);

FBD/LD


■実行条件

命令	実行条件
S.PIDCONT	
SP.PIDCONT	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	入出力データエリアに割り付けられている先頭デバイス	—	ワード	ANY16 ^{*1}
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

^{*1} ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—

機能

- S(P).PIDCONT命令実行時に、サンプリング周期の計測とPID演算を行います。
- S(P).PIDCONT命令は、(s)で指定したデバイス番号以降に設定されている入出力データエリアの設定値(SV)、測定値(PV)をもとにPID演算を行い、演算結果を入出力データエリアの自動操作量(MV)エリアに格納します。
- PID演算は、サンプリング周期の設定時間経過後の最初のS(P).PIDCONT命令実行時に行います。
- PID制御中は、必ず制御指令をONさせて、S(P).PIDCONT命令を毎スキャン実行させるようにしてください。毎スキャン実行させない場合は、正常なサンプリング周期でPID演算ができなくなります。また、1スキャン中にS(P).PIDCONT命令を複数回実行させることもできません。1スキャン中に複数回S(P).PIDCONT命令を実行させた場合は、正常なサンプリング周期でPID演算ができなくなります。
- S(P).PIDCONT命令は割り込みプログラムに記述して使用することはできません。S(P).PIDCONT命令を割り込みプログラムに記述した場合は、正常なサンプリング周期でPID演算ができなくなります。
- (s)には、入出力データエリアに指定されているデバイス番号の先頭を指定します。(1053ページ 入出力データ)
- 入出力データエリアとしてファイルレジスタを指定した場合は、ファイルレジスタに対してメモリプロテクトをかけないでください。メモリプロテクトがかかっているとエラーにはなりませんが、正常なPID演算を行うことができなくなります。
- 入出力データは、下記のように割り付けられています。

項目		入出力データ	
ライト		(s)+0	初回処理フラグ
リード/ライト禁止		(s)+1	PID制御用ワークエリア
		⋮	(ユーザ使用不可)
		(s)+9	
No.1ループ用入出力データ エリア (23ワード)	ライト	(s)+10	設定値(SV)
		(s)+11	測定値(PV)
	リード	(s)+12	自動操作量(MV)
		(s)+13	フィルタ後の測定値(PVf)
	ライト	(s)+14	手動操作量(MV _{MAN})
		(s)+15	手動/自動選択(MAN/AUTO)
	リード/ライト	(s)+16	アラーム(ALARM)
	リード/ライト禁止	(s)+17	No.1ループ用ワークエリア
		⋮	(ユーザ使用不可)
		(s)+32	
⋮	⋮	⋮	⋮
No.nループ用入出力データ エリア (23ワード)*1	ライト	(s)+(m+0)	設定値(SV)
		(s)+(m+1)	測定値(PV)
	リード	(s)+(m+2)	自動操作量(MV)
		(s)+(m+3)	フィルタ後の測定値(PVf)
	ライト	(s)+(m+4)	手動操作量(MV _{MAN})
		(s)+(m+5)	手動/自動選択(MAN/AUTO)
	リード/ライト	(s)+(m+6)	アラーム(ALARM)
	リード/ライト禁止	(s)+(m+7)	No.nループ用ワークエリア
		⋮	(ユーザ使用不可)
		(s)+(m+22)	

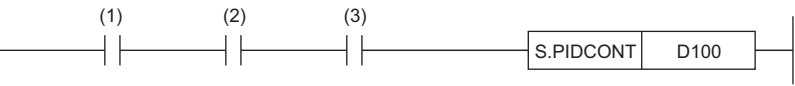
*1 $m=(n-1) \times 23 + 10$

- 入出力データの設定で使用するデバイスの点数は、下式により算出します。

デバイス点数 = $10 + 23 \times n$ (n: 使用ループ数)

- 各データは、BIN値で設定してください。
- 初回処理フラグは、PID演算開始時の処理方法の設定です。
- 初回演算処理時には、設定されたサンプリング周期に達しているとみなして演算します。
- 初回処理フラグが0の場合、使用ループ数分のPID演算処理を1スキャンで一括に処理します。また、0以外の場合は、使用ループ数分のPID演算処理を数スキャンに分割して処理し、初回処理を完了したループから順次サンプリングを開始します。1スキャンあたりの処理ループ数は、設定されている1スキャンの実行ループ数分となります。

- 入出力データエリアで「ライト」となっているデータは、お客様がプログラムで書き込んでください。「リード」となっているデータは、お客様がプログラムで読み出して使用します。「リード/ライト禁止」および「リード」となっているデータに書き込みを行うと正常な演算が行えなくなるため、絶対に行わないでください。ただし、初期状態から制御を開始する場合は、プログラムでデータクリアする必要があります。
- 使用ループ分のデバイス点数が、指定したデバイスの最終デバイス番号を超えるとエラーとなり処理しません。
- 手動モードにより手動操作量(MV_{MAN})の出力を行っている場合でも、S(P).PIDCONT命令を毎スキャン実行してください。S(P).PIDCONT命令を実行しないと、バンプレス機能を行うことができません。
- S(P).PIDCONT命令は、測定値(PV)の取込みを行うためのA/D変換ユニット、および操作量(MV)の出力を行うためのD/A変換ユニットが正常なときのみ実行するように、各ユニットのREADY信号によりインタロックを行うようにしてください。



- (1): 制御指令
- (2): A/D変換ユニットのREADY信号
- (3): D/A変換ユニットのREADY信号

各ユニット異常時に実行すると、正常な測定値(PV)の取込み、または正常な操作量(MV)の出力が行えないため、PID演算も正常に行うことができません。

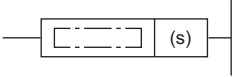
エラー

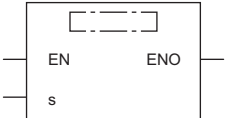
エラーコード (SD0)	内容
3285H	(s)で指定した入出力データエリアに設定したデータの値が、設定可能範囲外のとき。
3292H	S(P).PIDCONT命令実行前に、S(P).PIDINIT命令を実行していないとき。

指定ループNo.の演算停止

S(P).PIDSTOP

指定したループNo.のPID演算を停止します。

ラダー	ST
	ENO:=S_PIDSTOP(EN,s); ENO:=SP_PIDSTOP(EN,s);

FBD/LD


■実行条件

命令	実行条件
S.PIDSTOP	
SP.PIDSTOP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	停止するループNo.	1~32	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	○	—	—	—

機能

- (s)で指定されたループNo.のPID演算を停止します。S(P).PIDSTOP命令で停止させたループは、S(P).PIDINIT命令を実行しても、PID演算を再開しません。
- 停止中は演算データを保持します。

エラー

エラーコード (SD0)	内容
3285H	(s)に指定可能な範囲外のデータを指定したとき。 • 指定したループNo.が存在しない • 指定した値が1~32以外
3292H	S(P).PIDSTOP命令実行前に、S(P).PIDINIT命令、S(P).PIDCONT命令を実行していないとき。

指定ループNo.の演算開始

S(P).PIDRUN

指定したループNo.の演算を開始します。

ラダー	ST
	ENO:=S_PIDRUN(EN,s); ENO:=SP_PIDRUN(EN,s);

FBD/LD

■実行条件

命令	実行条件
S.PIDRUN	
SP.PIDRUN	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	停止するループNo.	1~32	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	○	—	—	—

機能

- (s)で指定したループNo.のPID演算を開始します。S(P).PIDSTOP命令でPID演算を停止したループNo.を再度実行させるための命令です。
- すでにPID演算実行中のループNo.に対してS(P).PIDRUN命令を実行した場合は、無処理となります。

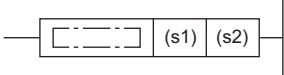
エラー

エラーコード (SD0)	内容
3285H	(s)で指定したループNo.が存在しないとき。 (s)が1~32以外するとき。
3292H	S(P).PIDRUN命令実行前に、S(P).PIDINIT命令、S(P).PIDCONT命令を実行していないとき。

指定ループNo.のパラメータ変更

S(P).PIDPRMW

指定ループNo.の演算パラメータを、指定したデバイス番号以降に格納されているPID制御用データに変更します。

ラダー	ST
	ENO:=S_PIDPRMW(EN,s1,s2); ENO:=SP_PIDPRMW(EN,s1,s2);

FBD/LD


■実行条件

命令	実行条件
S.PIDPRMW	
SP.PIDPRMW	

設定データ

■内容，範囲，データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	変更するループNo.	1~32	符号なしBIN16ビット	ANY16
(s2)	変更するPID制御用データが格納されている先頭デバイス	—	ワード	ANY16*1
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s1)	○	—	○	—	—	—	—	○	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	—

機能

- (s1)で指定したループNo.の演算パラメータを、(s2)で指定したデバイス番号以降に格納されているPID制御用データに変更します。
- (s2)で指定したデバイス番号以降のPID制御用データの構成は下記のとおりです。

PID制御用データ	
(s2)+0	演算式選択
(s2)+1	サンプリング周期(T_S)
(s2)+2	比例定数(K_P)
(s2)+3	積分定数(T_I)
(s2)+4	微分定数(T_D)
(s2)+5	フィルタ係数(α)
(s2)+6	操作量下限値(MVLL)
(s2)+7	操作量上限値(MVHL)
(s2)+8	操作量変化率リミット値(ΔMVL)
(s2)+9	測定値変化率リミット値(ΔPVL)
(s2)+10	0
(s2)+11	微分ゲイン(K_D)
(s2)+12	0
(s2)+13	0

エラー

エラーコード (SD0)	内容
3285H	(s1)に指定可能な範囲外のデータを設定したとき。 • 指定したループNo.が存在しない • 指定した値が1~32以外 (s2)に指定可能な範囲外のデータを設定したとき。 • PID制御用データが設定可能範囲外 • PID制御用データの(s2)+10, (s2)+12, (s2)+13が0でない
3292H	S(P).PIDPRMW命令実行前に、S(P).PIDINIT命令を実行していないとき。

24.3 PID制御命令(完全微分)

完全微分は、微分項の入力をそのまま使用するPID制御です。

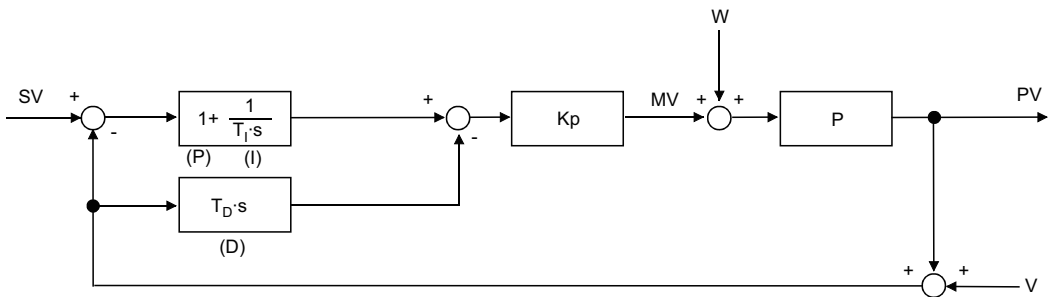


完全微分によるPID制御命令の性能仕様を示します。

項目			"PIDリミット制限あり"時	"PIDリミット制限なし"時
PID制御ループ数			最大32ループ	
サンプリング周期			T_S	
PID演算方式			測定値微分形完全微分(正動作/逆動作)	
PID定数設定範囲	比例定数	K_P	0.01~100.00	
	積分定数	T_I	0.1~3000.0s	
	微分定数	T_D	0.00~300.00s	
設定値設定範囲			SV	0~2000
測定値設定範囲			PV	-50~2050
操作量出力範囲			MV	-32768~32767

PID演算のブロック図と演算式を示します。

- 完全微分のPID演算ブロック



K_P : ゲイン
W: 外乱
P: 制御対象
V: 検出雑音

・ 演算式

正動作/逆動作	演算式
正動作	$EV_n = PV_{fn} - SV$ $\Delta MV = K_p \{ (EV_n - EV_{n-1}) + \frac{T_s}{T_i} \cdot EV_n + D_n \}$ $D_n = \frac{T_D}{T_s} (PV_{fn} - 2PV_{fn-1} + PV_{fn-2})$ $MV_n = \Sigma \Delta MV$
逆動作	$EV_n = SV - PV_{fn}$ $\Delta MV = K_p \{ (EV_n - EV_{n-1}) + \frac{T_s}{T_i} \cdot EV_n + D_n \}$ $D_n = \frac{T_D}{T_s} (-PV_{fn} + 2PV_{fn-1} - PV_{fn-2})$ $MV_n = \Sigma \Delta MV$

演算式中の記号の意味は下記のとおりです。

記号	意味
EV_n	今回のサンプル時の偏差
EV_{n-1}	1周期前の偏差
SV	設定値
PV_{fn}	今回のサンプル時の測定値(フィルタ後)
PV_{fn-1}	1周期前の測定値(フィルタ後)
PV_{fn-2}	2周期前の測定値(フィルタ後)
ΔMV	出力変化量
MV_n	今回の操作量
D_n	今回の微分項
T_s	サンプリング周期
K_p	比例定数
T_i	積分定数
T_D	微分定数

今回のサンプル時の測定値(フィルタ後)(PV_{fn})は、入力データの測定値を下式で演算した値です。入力データのフィルタ係数が設定されていない場合は、入力データの測定値(PV)と同一の値となります。

$$PV_{fn} = PV_n + \alpha(PV_{fn-1} - PV_n)$$

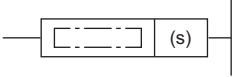
PV_n : 今回のサンプル時の測定値, α : フィルタ係数, PV_{fn-1} : 1周期前の測定値(フィルタ後)

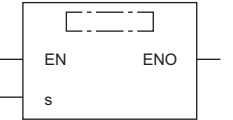
なお, PV_{fn} は入出力データエリアに格納されます。(1053ページ 入出力データ)

PID制御用データの設定

PIDINIT(P)

指定したデバイス番号以降に設定されている使用ループ数分のPID制御データを、一括でコントローラ内部に登録します。

ラダー	ST
	ENO:=PIDINIT(EN,s); ENO:=PIDINITP(EN,s);

FBD/LD


■実行条件

命令	実行条件
PIDINIT	
PIDINITP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	PID制御用データが設定されている先頭デバイス	—	ワード	ANY16*1
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード				ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—	—

機能

- (s)で指定したデバイス番号以降に設定されている使用ループ数分のPID制御用データを一括でコントローラ内部に登録し、PID制御を可能にします。(1050ページ PID制御用データ)
- PID制御用データは、下記のように割り付けられています。

項目	PID制御用データ	
全ループ共通	(s)+0	使用ループ数
	(s)+1	1スキャンの実行ループ数
No.1ループ用(10ワード)	(s)+2	演算式選択
	(s)+3	サンプリング周期(T_S)
	(s)+4	比例定数(K_P)
	(s)+5	積分定数(T_I)
	(s)+6	微分定数(T_D)
	(s)+7	フィルタ係数(α)
	(s)+8	操作量下限値(MVLL)
	(s)+9	操作量上限値(MVHL)
	(s)+10	操作量変化率リミット値(ΔMVL)
	(s)+11	測定値変化率リミット値(ΔPVL)
⋮	⋮	⋮
No.nループ用(10ワード)*1	(s)+(m+0)	演算式選択
	(s)+(m+1)	サンプリング周期(T_S)
	(s)+(m+2)	比例定数(K_P)
	(s)+(m+3)	積分定数(T_I)
	(s)+(m+4)	微分定数(T_D)
	(s)+(m+5)	フィルタ係数(α)
	(s)+(m+6)	操作量下限値(MVLL)
	(s)+(m+7)	操作量上限値(MVHL)
	(s)+(m+8)	操作量変化率リミット値(ΔMVL)
	(s)+(m+9)	測定値変化率リミット値(ΔPVL)

*1 $m=(n-1) \times 10 + 2$

- PID制御用データの設定で使用するデバイスの点数は、下式により算出します。

デバイス点数 $=2+10 \times n$ (n: 使用ループ数)

- 各データは、BIN値で設定してください。
- 使用ループ分のデバイス点数が、指定したデバイスの最終デバイス番号を超えるとエラーとなり処理しません。
- 1スキャン中に複数箇所でのPIDINIT(P)命令を実行した場合は、PIDCONT(P)命令に1番近くで実行したPIDINIT(P)命令の設定値が有効になります。
- PIDINIT(P)命令の実行は、必ずPIDCONT(P)命令実行前に行ってください。PIDINIT(P)命令を実行していない場合、PID制御を行うことはできません。

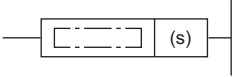
エラー

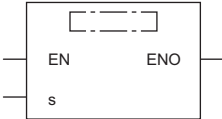
エラーコード (SD0)	内容
3285H	(s)に指定可能な範囲外のデータを設定したとき。 • PID制御用データに設定した値が、設定可能範囲外 • (使用ループ数) < (1スキャンの実行ループ数) • (操作量上限値) < (操作量下限値)

PID演算

PIDCONT(P)

実行指令のONで、サンプリング周期の計測とPID演算を行います。

ラダー	ST
	ENO:=PIDCONT(EN,s); ENO:=PIDCONT(EN,s);

FBD/LD


■実行条件

命令	実行条件
PIDCONT	
PIDCONT(EN,s)	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	入出力データエリアに割り付けられている先頭デバイス	—	ワード	ANY16 ^{*1}
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- PIDCONT(P)命令実行時に、サンプリング周期の計測とPID演算を行います。
- PIDCONT(P)命令は、(s)で指定したデバイス番号以降に設定されている入出力データエリアの設定値(SV)、測定値(PV)をもとにPID演算を行い、演算結果を入出力データエリアの自動操作量(MV)エリアに格納します。
- PID演算は、サンプリング周期の設定時間経過後の最初のPIDCONT(P)命令実行時に行います。
- PID制御中は、必ず制御指令をONさせて、PIDCONT(P)命令を毎スキャン実行させるようにしてください。毎スキャン実行させない場合は、正常なサンプリング周期でPID演算ができなくなります。また、1スキャン中にPIDCONT(P)命令を複数回実行させることもできません。1スキャン中に複数回PIDCONT(P)命令を実行させた場合は、正常なサンプリング周期でPID演算ができなくなります。
- PIDCONT(P)命令は割り込みプログラムに記述して使用することはできません。PIDCONT(P)命令を割り込みプログラムに記述した場合は、正常なサンプリング周期でPID演算ができなくなります。
- (s)には、入出力データエリアに指定されているデバイス番号の先頭を指定します。(1053ページ 入出力データ)
- 入出力データエリアとしてファイルレジスタを指定した場合は、ファイルレジスタに対してメモリプロテクトをかけないでください。メモリプロテクトがかかっているとエラーにはなりませんが、正常なPID演算を行うことができなくなります。
- 入出力データは、下記のように割り付けられています。

項目		入出力データ	
ライト		(s)+0	初回処理フラグ
リード/ライト禁止		(s)+1	PID制御用ワークエリア
		⋮	(ユーザ使用不可)
		(s)+9	
No.1ループ用入出力データ エリア (18ワード)	ライト	(s)+10	設定値(SV)
		(s)+11	測定値(PV)
	リード	(s)+12	自動操作量(MV)
		(s)+13	フィルタ後の測定値(PVf)
	ライト	(s)+14	手動操作量(MV _{MAN})
		(s)+15	手動/自動選択(MAN/AUTO)
	リード/ライト	(s)+16	アラーム(ALARM)
	リード/ライト禁止	(s)+17	No.1ループ用ワークエリア
		⋮	(ユーザ使用不可)
		(s)+27	
⋮	⋮	⋮	⋮
No.nループ用入出力データ エリア (18ワード)*1	ライト	(s)+(m+0)	設定値(SV)
		(s)+(m+1)	測定値(PV)
	リード	(s)+(m+2)	自動操作量(MV)
		(s)+(m+3)	フィルタ後の測定値(PVf)
	ライト	(s)+(m+4)	手動操作量(MV _{MAN})
		(s)+(m+5)	手動/自動選択(MAN/AUTO)
	リード/ライト	(s)+(m+6)	アラーム(ALARM)
	リード/ライト禁止	(s)+(m+7)	No.nループ用ワークエリア
		⋮	(ユーザ使用不可)
		(s)+(m+17)	

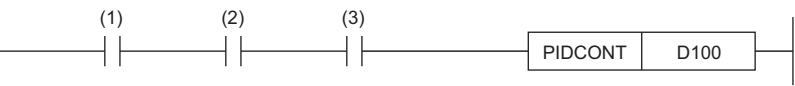
*1 $m=(n-1) \times 18 + 10$

- 入出力データの設定で使用するデバイスの点数は、下式により算出します。

デバイス点数 = $10 + 18 \times n$ (n: 使用ループ数)

- 各データは、BIN値で設定してください。
- 初回処理フラグは、PID演算開始時の処理方法の設定です。
- 初回演算処理時には、設定されたサンプリング周期に達しているとみなして演算します。
- 初回処理フラグが0の場合、使用ループ数分のPID演算処理を1スキャンで一括に処理します。また、0以外の場合は、使用ループ数分のPID演算処理を数スキャンに分割して処理し、初回処理を完了したループから順次サンプリングを開始します。1スキャンあたりの処理ループ数は、設定されている1スキャンの実行ループ数分となります。

- 入出力データエリアで「ライト」となっているデータは、お客様がプログラムで書き込んでください。「リード」となっているデータは、お客様がプログラムで読み出して使用します。「リード/ライト禁止」および「リード」となっているデータに書き込みを行うと正常な演算が行えなくなるため、絶対に行わないでください。ただし、初期状態から制御を開始する場合は、プログラムでデータクリアする必要があります。
- 手動モードにより手動操作量(MV_{MAN})の出力を行っている場合でも、PIDCONT(P)命令を毎スキャン実行してください。PIDCONT(P)命令を実行しないと、バンプレス機能を行うことができません。
- PIDCONT(P)命令は、測定値(PV)の取込みを行うためのA/D変換ユニット、および操作量(MV)の出力を行うためのD/A変換ユニットが正常なときのみ実行するように、各ユニットのREADY信号によりインタロックを行うようにしてください。各ユニット異常時に実行すると、正常な測定値(PV)の取込み、または正常な操作量(MV)の出力が行えないため、PID演算も正常に行うことができません。



- (1): 制御指令
- (2): A/D変換ユニットのREADY信号
- (3): D/A変換ユニットのREADY信号

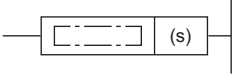
エラー

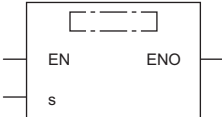
エラーコード (SD0)	内容
3285H	(s)で指定した入出力データエリアに設定したデータの値が、設定可能範囲外のとき。
3292H	PIDCONT(P)命令実行前に、PIDINIT(P)命令を実行していないとき。

指定ループNo.の演算停止

PIDSTOP(P)

(s)で指定されたループNo.のPID演算を停止します。

ラダー	ST
	ENO:=PIDSTOP(EN,s); ENO:=PIDSTOPP(EN,s);

FBD/LD


■実行条件

命令	実行条件
PIDSTOP	
PIDSTOPP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	停止するループNo.	1~32	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 0	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 0	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	—	○	—	—	—	—	○	○	—	—	—

機能

- (s)で指定されたループNo.のPID演算を停止します。PIDSTOP(P)命令で停止させたループは、PIDINIT(P)命令を実行しても、PID演算を再開しません。
- 停止中は演算データを保持します。

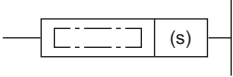
エラー

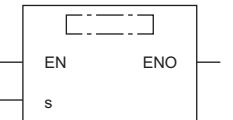
エラーコード (SD0)	内容
3285H	(s)に指定可能な範囲外のデータを設定したとき。 ・ 指定したループNo.が存在しない ・ 指定した値が1~32以外
3292H	PIDSTOP(P)命令実行前に、PIDINIT(P)命令、PIDCONT(P)命令を実行していないとき。

指定ループNo.の演算開始

PIDRUN(P)

指定したループNo.の演算を開始します。

ラダー	ST
	ENO:=PIDRUN(EN,s); ENO:=PIDRUNP(EN,s);

FBD/LD


■実行条件

命令	実行条件
PIDRUN	
PIDRUNP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s)	停止するループNo.	1~32	符号なしBIN16ビット	ANY16
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(s)	○	—	○	—	—	—	—	○	○	—	—	—

機能

- (s)で指定したループNo.のPID演算を開始します。PIDSTOP(P)命令でPID演算を停止したループNo.を再度実行させるための命令です。
- すでにPID演算実行中のループNo.に対してPIDRUN(P)命令を実行した場合は、無処理となります。

エラー

エラーコード (SD0)	内容
3285H	(s)に指定可能な範囲外のデータを設定したとき。 • 指定したループNo.が存在しない • 指定した値が1~32以外
3292H	PIDRUN(P)命令実行前に、PIDINIT(P)命令、PIDCONT(P)命令を実行していないとき。

指定ループNo.のパラメータ変更

PIDPRMW(P)

指定したループNo.の演算パラメータを、指定したデバイス番号以降に格納されているPID制御用データに変更します。

ラダー	ST
	ENO:=PIDPRMW(EN,s1,s2); ENO:=PIDPRMWP(EN,s1,s2);

FBD/LD

■実行条件

命令	実行条件
PIDPRMW	
PIDPRMWP	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	変更するループNo.	1~32	符号なしBIN16ビット	ANY16
(s2)	変更するPID制御用データが格納されている先頭デバイス	—	ワード	ANY16*1
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	○	—	○	—	—	—	—	○	○	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—

機能

- (s1)で指定したループNo.の演算パラメータを、(s2)で指定したデバイス番号以降に格納されているPID制御用データに変更します。
- (s2)で指定したデバイス番号以降のPID制御用データの構成は下記のとおりです。

PID制御用データ	
(s2)+0	演算式選択
(s2)+1	サンプリング周期(T_S)
(s2)+2	比例定数(K_P)
(s2)+3	積分定数(T_I)
(s2)+4	微分定数(T_D)
(s2)+5	フィルタ係数(α)
(s2)+6	操作量下限値(MVLL)
(s2)+7	操作量上限値(MVHL)
(s2)+8	操作量変化率リミット値(ΔMVL)
(s2)+9	測定値変化率リミット値(ΔPVL)

エラー

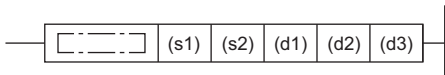
エラーコード (SD0)	内容
3285H	(s1)に指定可能な範囲外のデータを設定したとき。 指定したループNo.が存在しない 指定した値が1~32以外
	(s2)に指定したPID制御用データが設定可能範囲外のとき。
3292H	PIDPRMW(P)命令実行前に、PIDINIT(P)命令を実行していないとき。

25.1 アドオン用命令

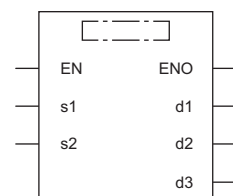
アドオン実行命令

ADDONEXE(P)

インストールされたアドオンの実行を指示する。

ラダー	ST
	ENO:= ADDONEXE (EN,s1,s2,d1,d2,d3); ENO:= ADDONEXEP (EN,s1,s2,d1,d2,d3);

FBD/LD



■実行条件

命令	実行条件
ADDONEXE	
ADDONEXEP	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(s1)	アドオンID (SQタイプのみ対応)*1	1~64	ダブルワード	ANY32_S
(s2)	要求データが格納されているデバイスの先頭番号	—	ワード	ANY16
(d1)	応答データが格納されているデバイスの先頭番号	—	ワード	ANY16
(d2)	処理完了でONにするビットデバイスを指定します。*2	—	ビット	ANYBIT_ARRAY
(d3)	命令完了時に完了ステータスを格納するデバイスを指定します。	—	ワード	ANY16

*1 アドオンタイプについては、使用するコントローラのユーザーズマニュアルを確認してください。

*2 異常完了時には(d2)+1もONします。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (P)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(s1)	—	—	○*1	—	—	—	—	○	○	—	—	○
(s2)	—	—	○*1	—	—	—	—	○	—	—	—	○
(d1)	—	—	○*1	—	—	—	—	○	—	—	—	○
(d2)	○*2	—	○*3	—	—	—	—	—	—	—	—	○
(d3)	—	—	○*1	—	—	—	—	○	—	—	—	○

*1 FDは使用できません。

*2 FX, FYは使用できません。

*3 T, ST, C, FDは使用できません

オペランド(s2)では下記のデバイスを使用します。

オペランド	項目	内容	設定範囲	セット側
(s2)	応答データ許容数	応答データの許容するワード数を指定します。	0~8192	ユーザ
(s2)+1	要求データ長	要求データ長のワード数を指定します。	0~8192	ユーザ
(s2)+2	要求データ	要求データを指定します。	—	—

オペランド(d1)では下記のデバイスを使用します。

オペランド	項目	内容	設定範囲	セット側
(d1)	応答データ長	応答データ長のワード数を指定します。	0~8192	システム
(d1)+1	応答データ	応答データを指定します。	—	システム

機能

- インストールされたアドオンの実行を指示します。
- (s1)で指定したアドオンIDに紐づくアドオンの関数に、(s2)で指定したデバイス以降に格納されている要求データを引き渡し、応答データを(d1)+1で指定したデバイス以降に格納します。ただし、(s2)で指定した応答データ許容数よりも実際に受信した応答データが大きい場合は、応答データ許容数分のみ格納し、残りの応答データを切り捨てます。(専用命令は正常完了します。)この場合、応答データ長(d1)は実際に格納したデータ数になります。
- 本命令の完了確認は(d2)と(d2)+1で行えます。

オペランド	項目	内容
(d2)	完了デバイス	命令実行開始でOFFし、アドオン実行が完了するとONします。 完了後、ONのまま保持されるので、必要に応じてOFFしてください。
(d2)+1	完了デバイス	命令実行開始でOFFし、正常完了時はOFFしたまま変化しません。 異常完了時はONします。完了後、ONのまま保持されるので、必要に応じてOFFしてください。

- 本命令の実行結果は(d3)で確認できます。

オペランド	項目	内容
(d3)	完了ステータス	命令実行開始で0000Hを格納し、正常完了時はそのまま変化しません。 異常完了時はエラーに応じた完了ステータスが格納されます。完了後、ステータスはそのまま保持されるので、必要に応じてクリアしてください。

注意事項

- 実行する処理に応じて(s2)を設定してください。(d1)の内容は、実行する処理によって異なります。
- 要求データおよび応答データが不要な場合でも、オペランドを指定してください。
- 本命令の実行が完了するまで、指定している各要求データを変更しないでください。
- アドオンIDの指定は、ラベル「Type_AddonLibraryName_System.Md.AddonMemory.AddonID」を指定してください。ラベルの詳細については、使用するコントローラのユーザズマニュアルを確認してください。

エラー

エラーコード (SD0)	内容
3285H	(s1)で指定したアドオンIDが設定範囲外の時。
	(s2)で指定した応答データ許容数が設定範囲外の時。
	(s2+1)で指定した要求データ長が設定範囲外の時。

本命令の異常完了時には、(d3)で指定されたデバイスに下記のエラーコードが格納します。

エラーコード	内容
1000H	(s1)で指定したアドオンIDに対応したアドオンが存在しない。

MEMO

第5部 ネットワーク命令

この部は下記の章構成となります。

MX-Rモデルでインテリジェント機能ユニットを使用する場合の専用命令については、下記を参照してください。

 MELSEC iQ-R プログラミングマニュアル(ユニット専用命令編)

26 Ethernet用命令

26 Ethernet用命令

MX-Rモデルでインテリジェント機能ユニットを使用する場合の専用命令については、下記を参照してください。
MELSEC iQ-R プログラミングマニュアル(ユニット専用命令編)

26.1 オープン/クローズ処理命令

Point

下記の命令を実行中のコネクションは、重複して下記の命令を実行しても要求は無視されます。
この時、後から実行した命令の実行条件が成立している間は、先に実行している命令が完了した後で要求が実行されます。また、要求が無視されている間、SM699(専用命令未実行フラグ)はONになります。

- SOCOPEN命令
- SOCCLOSE命令
- SOCSND命令
- SOCRCV命令
- ECPRTCL命令

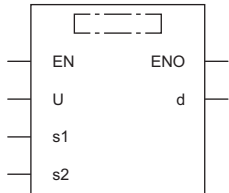
コネクションの確立

GP.SOCOPEN, SP.SOCOPEN

指定したコネクションのオープン処理を行います。

ラダー	ST
	ENO:=GP_SOCOPEN(EN,U,s1,s2,d); ENO:=SP_SOCOPEN(EN,U,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
GP.SOCOPEN SP.SOCOPEN	

設定データ

■内容、範囲、データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(U)	GP.SOCOPEN	対象の先頭入出力番号 (先頭入出力番号の上3桁を指定)	3E0H	符号なしBIN16ビット	ANY16
	SP.SOCOPEN	ダミー	—	文字列	ANYSTRING_SINGLE
(s1)		コネクション番号	1~64	符号なしBIN16ビット	ANY16
(s2)	GP.SOCOPEN	コントロールデータを格納する先頭デバイス 参照	コントロールデータ 参照	デバイス	ANY16
	SP.SOCOPEN			ワード	ANY16_ARRAY (要素数: 10)
(d)		命令完了にて1スキャンONするデバイス 異常完了時は(d)+1もONします。	—	ビット	ANYBIT_ARRAY (要素数: 2)
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド		ビット		ワード				ダブルワード		間接指定	定数			その他 (U)
		X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z		LT, LST, LC	LZ		K, H	E	\$	
(U)	GP.SOCOPEN	—	—	○	—	—	—	—	—	○	○	—	—	○
	SP.SOCOPEN	—	—	○	—	—	—	—	—	○	—	—	○	○
(s1)		—	—	○	—	—	—	—	—	○	○	—	—	—
(s2)		—	—	○	—	—	—	—	—	○	—	—	—	—
(d)		○	—	○	—	—	—	—	—	○	—	—	—	—

■コントロールデータ

オペランド: (s2)										
デバイス	項目	内容	設定範囲	セット側						
+0	実行/完了タイプ	コネクションのオープン処理時に、エンジニアリングツールによるパラメータ設定値を使用するか、コントロールデータの(s2)+2~(s2)+9の設定値を使用するか指定します。 ・0000H: エンジニアリングツールの"オープン設定"で設定した内容でオープン処理を行います。 ・8000H: コントロールデータの(s2)+2~(s2)+9で指定した内容でオープン処理を行います。*1	0000H 8000H	ユーザ						
+1	完了ステータス	命令完了時に完了ステータスを格納します。 ・0000H: 正常完了 ・0000H以外: 異常完了(エラーコード)	—	システム						
+2	使用用途設定エリア	コネクションの使用用途を指定します。 b15 b14 ... b10 b9 b8 b7 ... b0 (s2)+2 <table><tr><td>(3)</td><td>0</td><td>(4)</td><td>(2)</td><td>(1)</td><td>0</td></tr></table> (1) 通信方式(プロトコル)(ビット8, 7) ・00: TCP ・01: TLS ・10: UDP ・11:DTLS (2) ソケット通信機能の手順有無(ビット9) ・1: 無手順(固定) (3) オープン方式(ビット15, 14) ・00: ActiveオープンまたはUDP ・10: Unpassiveオープン ・11: Fullpassiveオープン (4) 通信プロトコル設定(ビット10) ・0: 通信プロトコル支援機能を使用しない(ソケット通信機能を使用する) ・1: 通信プロトコル支援機能を使用する	(3)	0	(4)	(2)	(1)	0	—	ユーザ
(3)	0	(4)	(2)	(1)	0					
+3	自局ポート番号	自局のポート番号を指定します。	1~4999, 5010~65534 (0001H~1387H, 1392H~FFFEH)	ユーザ						

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+4	相手機器IPアドレス ^{*2}	相手機器のIPアドレスを指定します。	00000001H~FFF	ユーザ
+5		ブロードキャスト通信を行う場合はFFFFFFFHを指定します。	FFFFFFH	
+6	相手機器ポート番号 ^{*2}	相手機器のポート番号を指定します。	1~65535	ユーザ
		全ポート番号を対象に受信を行う場合、FFFFFFHを指定します。 ^{*3}	(0001H~FFFFFFH)	
+7~+9	—	使用禁止	—	システム

*1 コントロールデータをにない設定項目は下記の設定になります。

- ・生存確認: KeepAlive (TCP/TLSの場合), UDP (UDP/DTLSの場合)
- ・コネクション監視タイマの設定: 無効

*2 Unpassiveオープン時は相手機器IPアドレス, 相手機器ポート番号は無視されます。

*3 ポート番号を65535(FFFFFFH)を指定した場合, データの送信できません。データを送信する場合は1~65534(1~FFFEH)を指定してください。

Point

ポート番号0001H~03FFHは一般的に予約されたポート番号(WELL KNOWN PORT NUMBERS)であり, F000H~FFFEHは他の機能で動的に使用する番号であるため, 自局ポート番号0400H~1387H, 1392H~FFFEHを使用することを推奨します。

機能

- ・(s1)で指定したコネクションのオープン処理を行います。オープン処理で使用する設定値は, (s2)+0で選択します。
- ・本命令実行および正常/異常完了は, 設定データで指定した完了デバイス(d), 完了時の状態表示デバイス(d)+1で確認できます。

・完了デバイス(d)

本命令が完了したスキャンのEND処理でONし, 次のEND処理でOFFします。

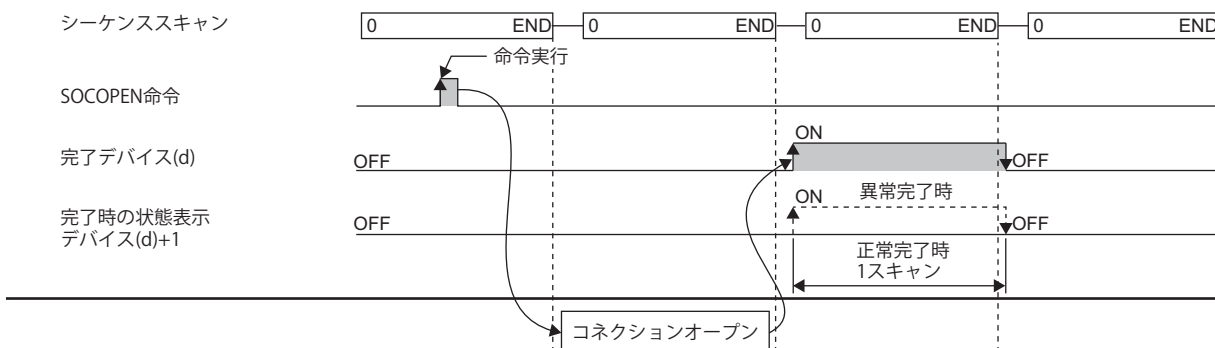
・完了時の状態表示デバイス(d)+1

本命令の完了したときの状態により, ON/OFFします。

正常完了時: OFFのまま変化しません。

異常完了時: 本命令が完了したスキャンのEND処理でONし, 次のEND処理でOFFします。

- ・本命令の実行タイミングは下記となります。



- ・パラメータ上で設定していないコネクション(プロトコル欄が空欄になっているコネクション)をオープンして使用できます。この場合は(s2)+0を8000Hにして, コントロールデータの(s2)+1~(s2)+9でオープンの内容を指定してください。

Point

本命令は暗号化通信プロトコル(TLSとDTLS)に対応しています。詳細は使用するコントローラのユーザーズマニュアルを確認してください。

注意事項

- Ethernetポートにおけるコネクション確立時の注意事項を下記に記載します。コネクション状態の確認方法は使用するコントローラのユーザズマニュアルを確認してください。

項目	内容
TCP/IPまたはTLSプロトコル、かつFullPassiveモードでコネクション確立する場合	本命令を実行した後、相手機器接続構成設定で設定されていないIPアドレスの相手機器からコネクションの要求を受けた場合、正常応答を返した後にコネクションを切断する動作になります。相手機器から見れば一度コネクションが確立して切断される動作になります。 コネクション確立が成功したかは下記命令の完了ステータスを確認してください。成功した場合、0000Hが格納されます。失敗した場合、C1B1Hが格納されます。 • SOCSND命令 • SOCRCV命令 • SOCCSET命令 • ECPRTCL命令
TCP/IPまたはTLSプロトコル、かつFullPassiveモード/UnPassiveモードでコネクション確立後、再度コネクション確立する場合	本命令を実行した後、相手機器からのコネクション要求によりコネクションを確立した状態で、再度相手機器からコネクション要求を受けた場合、これを受け付けず、コネクションが切断されます。(たとえば、コネクション確立後にSOCCLOSE命令を実行する前に相手機器がリセットされ、相手機器から再度コネクションオープンを要求される場合があります。) コントローラは再コネクション要求を受付可能になるまで数百ms程度かかるため、相手機器は切断を確認後1~2秒間隔をあけてから、コネクションを要求してください。 また、相手機器で再コネクション要求が成功しているかは、相手機器側のコネクション状態がオープン状態になっているかを確認してください。
TCP/IPプロトコルとTLSプロトコルでコネクション確立する場合	本命令が正常完了することがありますが、コネクションは確立しておらず、データの送受信は保証されません。 相手機器から正常完了後に切断されたかは、相手機器側のコネクション状態のオープン状態を確認してください。 各ケースの動作は下記ようになります。 ■Activeモード(コントローラ): TCP/IP, Passiveモード: TLSの場合 TCP/IPプロトコルのActiveモードのコントローラから、TLSプロトコルのPassiveモードでオープンされている相手機器に対して本命令を実行すると、命令が正常完了し、その後相手機器から切断されます。 ■Activeモード: TLS, Passiveモード(コントローラ): TCP/IPの場合 TCP/IPプロトコルのPassiveモードのコントローラで本命令を実行し、TLSプロトコルのActiveモードの相手機器からコネクションの要求を受けた場合、命令が正常完了し、その後相手機器から切断されます。

- Ethernet機能のアドオン及びEthernet命令を同時に実行しないでください。同時に使用した場合、誤動作を起こす可能性があります。

エラー

エラーコード (SD0)	内容
3285H	(s1)で指定するコネクション番号が範囲外のとき。

異常完了時は、完了時の状態表示デバイス(d)+1がONになり、完了ステータス(s2)+1にエラーコードが格納されます。完了ステータス(s2)+1に格納されるエラーコードについては、下記を参照してください。

📖使用するコントローラのユーザズマニュアル

Point

TCP/TLS/DTLS Passive方式のコネクションにSOCOPEN命令を実行すると接続待ちになり、相手機器からの接続要求により接続が完了するまで命令実行中になります。

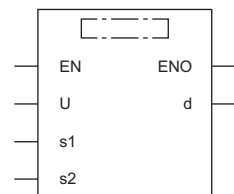
コネクションの切断

GP.SOCCLOSE, SP.SOCCLOSE

指定したコネクションに対するクローズ処理を行います。

ラダー	ST
	ENO:=GP_SOCCLOSE(EN,U,s1,s2,d); ENO:=SP_SOCCLOSE(EN,U,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
GP.SOCCLOSE SP.SOCCLOSE	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(U)	GP.SOCCLOSE 対象の先頭入出力番号 (先頭入出力番号の上3桁を指定)	3E0H	符号なしBIN16ビット	ANY16
	SP.SOCCLOSE ダミー	—	文字列	ANYSTRING_SINGLE
(s1)	コネクション番号	1~64	符号なしBIN16ビット	ANY16
(s2)	GP.SOCCLOSE コントロールデータを格納する先頭デバイス	コントロールデータ参照	デバイス	ANY16
	SP.SOCCLOSE		ワード	ANY16_ARRAY (要素数: 2)
(d)	命令完了にて1スキャンONするデバイス 異常完了時は(d)+1もONします。	—	ビット	ANYBIT_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

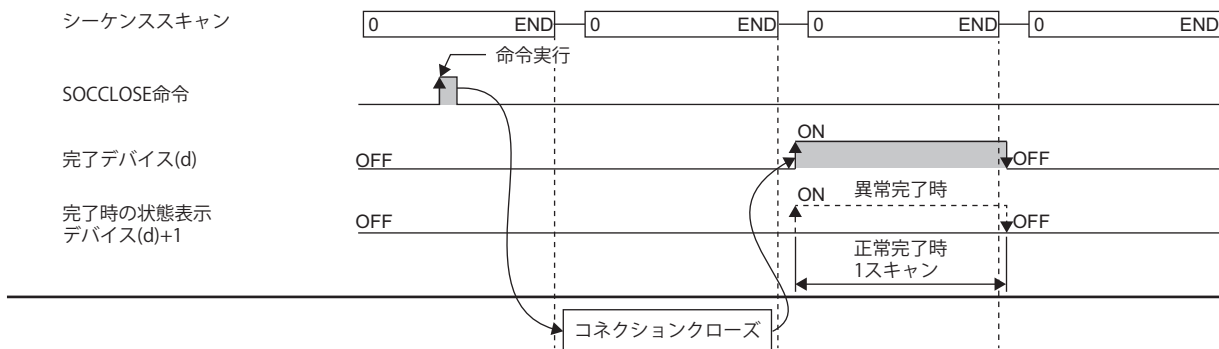
オペランド		ビット		ワード			ダブルワード		間接指定	定数				その他 (U)
		X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$		
(U)	GP.SOCCLOSE	—	—	○	—	—	—	—	○	○	—	—	○	○
	SP.SOCCLOSE	—	—	○	—	—	—	—	○	—	—	○	○	○
(s1)		—	—	○	—	—	—	—	○	○	—	—	—	—
(s2)		—	—	○	—	—	—	—	○	—	—	—	—	—
(d)		○	—	○	—	—	—	—	○	—	—	—	—	—

■コントロールデータ

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+0	システムエリア	—	—	—
+1	完了ステータス	命令完了時に完了ステータスを格納します。 ・0000H: 正常完了 ・0000H以外: 異常完了(エラーコード)	—	システム

機能

- ・(s1)で指定した接続に対するクローズ処理を行います(接続の切断)。
 - ・本命令実行および正常/異常完了は、設定データで指定した完了デバイス(d)，完了時の状態表示デバイス(d)+1で確認できます。
 - ・完了デバイス(d)
- 本命令が完了したスキャンのEND処理でONし、次のEND処理でOFFします。
- ・完了時の状態表示デバイス(d)+1
- 本命令の完了したときの状態により、ON/OFFします。
- 正常完了時: OFFのまま変化しません。
- 異常完了時: 本命令が完了したスキャンのEND処理でONし、次のEND処理でOFFします。
- ・本命令の実行タイミングは下記となります。



エラー

エラーコード (SD0)	内容
3285H	(s1)で指定する接続番号が範囲外のとき。

異常完了時は、完了時の状態表示デバイス(d)+1がONになり、完了ステータス(s2)+1にエラーコードが格納されます。完了ステータス(s2)+1に格納されるエラーコードについては、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

26.2 ソケット通信用命令

Point

下記の命令を実行中のコネクションは、重複して下記の命令を実行しても要求は無視されます。

この時、後から実行した命令の実行条件が成立している間は、先に実行している命令が完了した後で要求が実行されます。また、要求が無視されている間、SM699(専用命令未実行フラグ)はONになります。

- SOCOPEN命令
- SOCCLOSE命令
- SOCSND命令
- SOCRCV命令
- ECPRTCL命令

受信データのEND処理時読出し

GP.SOCRCV, SP.SOCRCV

命令実行後のEND処理にて、指定したコネクションの受信データをソケット通信受信データエリアから読み出します。

ラダー	ST
	ENO:=GP_SOCRCV(EN,U,s1,s2,d1,d2); ENO:=SP_SOCRCV(EN,U,s1,s2,d1,d2);

FBD/LD

■実行条件

命令	実行条件
GP.SOCRCV SP.SOCRCV	

設定データ

■内容、範囲、データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(U)	GP.SOCRCV	対象の先頭入出力番号 (先頭入出力番号の上3桁を指定)	3E0H	符号なしBIN16ビット	ANY16
	SP.SOCRCV	ダミー	—	文字列	ANYSTRING_SINGLE
(s1)		コネクション番号	1~64	符号なしBIN16ビット	ANY16
(s2)	GP.SOCRCV	コントロールデータを格納する先頭デバイス	コントロールデータ参照	デバイス	ANY16
	SP.SOCRCV			ワード	ANY16_ARRAY (要素数: 2)
(d1)	GP.SOCRCV	受信データを格納する先頭デバイス	—	デバイス	ANY16*1
	SP.SOCRCV			ワード	
(d2)		命令完了にて1スキャンONするデバイス 異常完了時は(d2)+1もONします。	—	ビット	ANYBIT_ARRAY (要素数: 2)
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド		ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
		X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 0	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 0	Z	LT, LST, LC	LZ		K, H	E	\$	
(U)	GP.SOCRCV	—	—	○	—	—	—	—	○	○	—	—	○
	SP.SOCRCV	—	—	○	—	—	—	—	○	—	—	○	○
(s1)	GP.SOCRCV	○	—	○	—	—	—	—	○	○	—	—	—
	SP.SOCRCV	—	—	○	—	—	—	—	○	○	—	—	—
(s2)		—	—	○	—	—	—	—	○	—	—	—	—
(d1)		—	—	○	—	—	—	—	○	—	—	—	—
(d2)		○	—	○	—	—	—	—	○	—	—	—	—

■コントロールデータ

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+0	システムエリア	—	—	—
+1	完了ステータス	命令完了時に完了ステータスを格納します。 • 0000H: 正常完了 • 0000H以外: 異常完了(エラーコード)	—	システム

オペランド: (d1)				
デバイス	項目	内容	設定範囲	セット側
+0	受信データ長	ソケット通信受信データエリアから読み出したデータのデータ長が格納されます。 (バイト数)	0~10238	システム
+1~+0	受信データ	ソケット通信受信データエリアから読み出したデータが若いアドレスから順次格納 されます。*1	—	システム

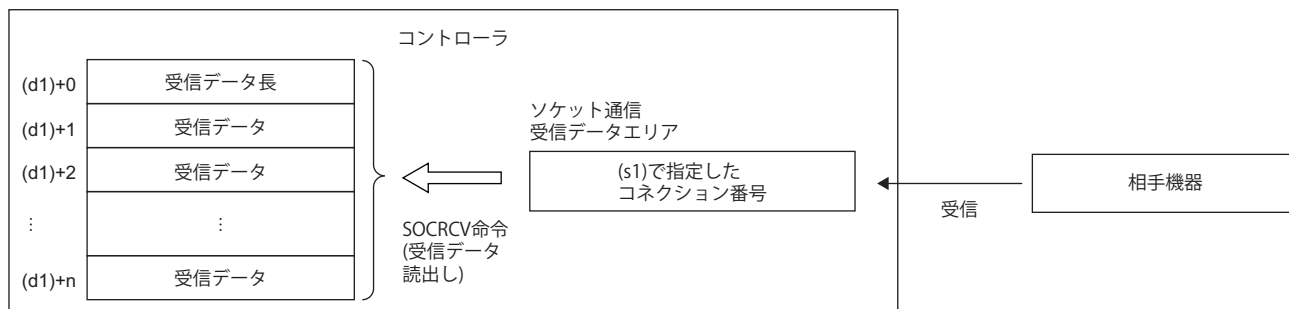
*1 受信したデータは、バイト単位で下位バイト側から順に格納されます。奇数バイト分のデータを受信したときは、最終データの格納エリアの下位バイトに最後の受信データが格納されます。

Point

- 本命令実行時、受信データのソケット通信受信データエリアからの読出しはEND処理で行います。そのため、本命令を実行するとスキャンタイムが延びます。
- 奇数バイトのデータを受信した場合、最後の受信データを格納したデバイスの上位バイトには無効なデータが入ります。

機能

- 本命令実行後のEND処理で、(s1)で指定した接続の受信データをソケット通信受信データエリア(各接続にある相手機器から受信したデータを格納するエリア)から読み出します。



- 本命令実行および正常/異常完了は、設定データで指定した完了デバイス(d2), 完了時の状態表示デバイス(d2)+1で確認できます。

- 完了デバイス(d2)

本命令が完了したスキャンのEND処理でONし、次のEND処理でOFFします。

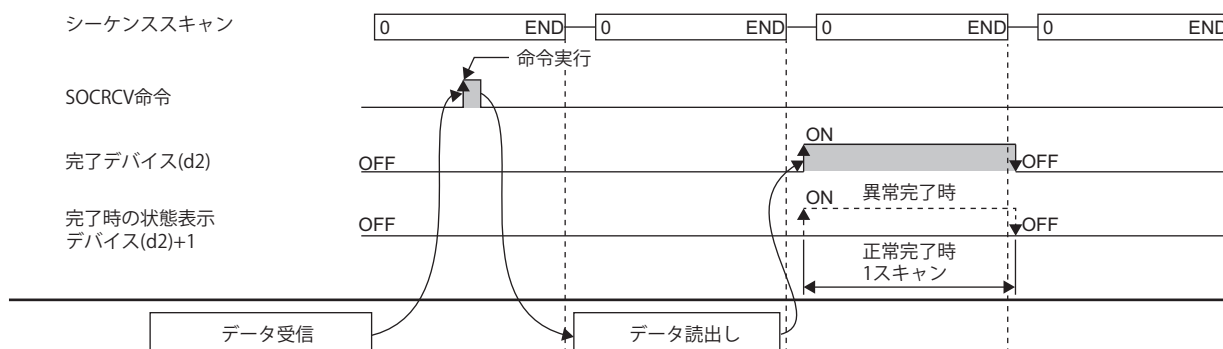
- 完了時の状態表示デバイス(d2)+1

本命令の完了したときの状態により、ON/OFFします。

正常完了時: OFFのまま変化しません。

異常完了時: 本命令が完了したスキャンのEND処理でONし、次のEND処理でOFFします。

- 本命令の実行タイミングは下記となります。



Point

本命令は暗号化通信プロトコル(TLSとDTLS)に対応しています。詳細は使用するコントローラのユーザーズマニュアルを確認してください。

注意事項

同一コネクションに対する受信データの読出しを行う場合、SOCRCVS命令と併用しないでください。

エラー

エラーコード (SD0)	内容
3285H	(s1)で指定するコネクション番号が範囲外の時。

異常完了時は、完了時の状態表示デバイス(d2)+1がONになり、完了ステータス(s2)+1にエラーコードが格納されます。

完了ステータス(s2)+1に格納されるエラーコードについては、下記を参照してください。

 使用するコントローラのユーザーズマニュアル

Point

- 膨大なデータを受信しないために、SOCRMODE命令で受信データサイズを設定することで、受信するデータ量を抑えることができます。
- 本命令の完了デバイスをb接点で実行指示に接続することで、データを連続して受信した場合でも連続して読み出すことができます。

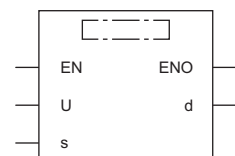
受信データの命令実行時読出し

G.SOCRCVS, S.SOCRCVS

指定したコネクションの受信データをソケット通信受信データエリアから読み出します。

ラダー	ST
	ENO:=G_SOCRCVS(EN,U,s,d); ENO:=S_SOCRCVS(EN,U,s,d);

FBD/LD



■実行条件

命令	実行条件
G.SOCRCVS S.SOCRCVS	

設定データ

■内容, 範囲, データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(U)	G.SOCRCVS	対象の先頭入出力番号 (先頭入出力番号の上3桁を指定)	3E0H	符号なしBIN16ビット	ANY16
	S.SOCRCVS	ダミー	—	文字列	ANYSTRING_SINGLE
(s)		コネクション番号	1~64	符号なしBIN16ビット	ANY16
(d)	G.SOCRCVS	受信データを格納する先頭デバイス	—	デバイス	ANY16 ^{*1}
	S.SOCRCVS			ワード	
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド		ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
		X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(U)	G.SOCRCVS	—	—	○	—	—	—	—	○	○	—	—	○
	S.SOCRCVS	—	—	○	—	—	—	—	○	—	—	○	○
(s)	G.SOCRCVS	○	—	○	—	—	—	—	○	○	—	—	—
	S.SOCRCVS	—	—	○	—	—	—	—	○	○	—	—	—
(d)		—	—	○	—	—	—	—	○	—	—	—	—

■コントロールデータ

オペランド: (d)				
デバイス	項目	内容	設定範囲	セット側
+0	受信データ長	ソケット通信受信データエリアから読み出したデータのデータ長が格納されます。(バイト数)	0~10238	システム
+1~+□	受信データ	ソケット通信受信データエリアから読み出したデータが若いアドレスから順次格納されます。 ^{*1}	—	システム

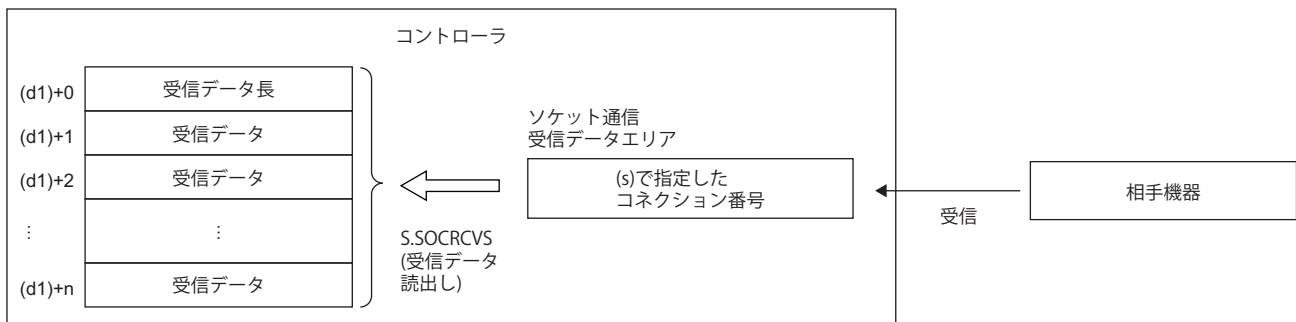
*1 受信したデータは、バイト単位で下位バイト側から順に格納されます。奇数バイト分のデータを受信したときは、最終データの格納エリアの下位バイトに最後の受信データが格納されます。

Point

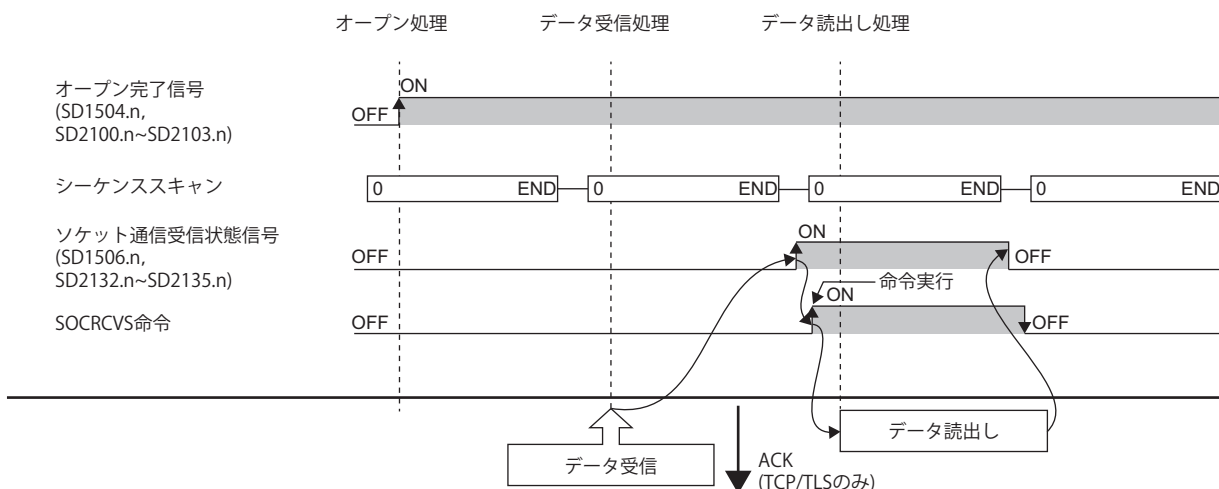
- 受信データサイズのデフォルトは2046バイトです。2047バイト以上のデータを受信する場合は、SOCRMODE命令により、受信データサイズを変更してください。
- 奇数バイトのデータを受信した場合、最後の受信データを格納したデバイスの上位バイトには無効なデータが入ります。

機能

- (s)で指定したコネクションの受信データをソケット通信受信データエリア(コネクション毎にある相手機器から受信したデータを格納するエリア)から読み出します。



- 本命令を使用した受信処理のタイミングは下記になります。



Point

本命令は暗号化通信プロトコル(TLSとDTLS)に対応しています。詳細は使用するコントローラのユーザーズマニュアルを確認してください。

注意事項

同一コネクションに対する受信データの読出しをする場合、SOCRCV命令と併用しないでください。

エラー

エラーコード (SD0)	内容
3285H	(s)で指定するコネクション番号が範囲外の時。

Point

膨大なデータを受信しないために、SOCRMODE命令で受信データサイズを設定することで、受信するデータ量を抑えることができます。

GP.SOCSND, SP.SOCSND

指定した接続の相手機器へ、データを送信します。

ラダー	ST
	ENO:=GP_SOCSND(EN,U,s1,s2,s3,d); ENO:=SP_SOCSND(EN,U,s1,s2,s3,d);

FBD/LD

■実行条件

命令	実行条件
GP.SOCSND SP.SOCSND	

設定データ

■内容, 範囲, データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(U)	GP.SOCSND	対象の先頭入出力番号 (先頭入出力番号の上3桁を指定)	3E0H	符号なしBIN16ビット	ANY16
	SP.SOCSND	ダミー	—	文字列	ANYSTRING_SINGLE
(s1)		コネクション番号	1~64	符号なしBIN16ビット	ANY16
(s2)	GP.SOCSND	コントロールデータを格納する先頭デバイス	コントロールデータ参照	デバイス	ANY16
	SP.SOCSND			ワード	ANY16_ARRAY (要素数: 2)
(s3)	GP.SOCSND	送信データを格納する先頭デバイス	—	デバイス	ANY16 ^{*1}
	SP.SOCSND			ワード	
(d)		命令完了にて1スキャンONするデバイス 異常完了時は(d)+1もONします。	—	ビット	ANYBIT_ARRAY (要素数: 2)
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド		ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
		X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(U)	GP.SOCSND	—	—	○	—	—	—	—	○	○	—	—	○
	SP.SOCSND	—	—	○	—	—	—	—	○	—	—	○	○
(s1)	GP.SOCSND	○	—	○	—	—	—	—	○	○	—	—	—
	SP.SOCSND	—	—	○	—	—	—	—	○	○	—	—	—
(s2)		—	—	○	—	—	—	—	○	—	—	—	—
(s3)		—	—	○	—	—	—	—	○	—	—	—	—
(d)		○	—	○	—	—	—	—	○	—	—	—	—

■コントロールデータ

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+0	システムエリア	—	—	—
+1	完了ステータス	命令完了時に完了ステータスを格納します。 • 0000H: 正常完了 • 0000H以外: 異常完了(エラーコード)	—	システム

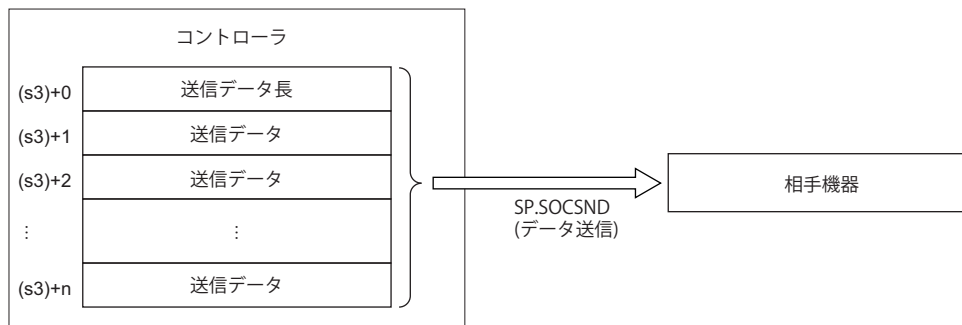
オペランド: (s3)				
デバイス	項目	内容	設定範囲	セット側
+0	送信データ長	送信データ長を指定します。(バイト数)	1~10238	ユーザ
+1~+□	送信データ	送信データを指定します。*1	—	ユーザ

*1 送信データは、バイト単位で下位バイト側から順に送信されます。奇数バイト分のデータを送信したときは、最終データの格納エリアの下位バイトに最後の送信データを格納してください。

Point

TCPの場合、送信データ長を相手機器の最大ウィンドウサイズ(受信バッファ)以下にしてください。
 TLSの場合、暗号化前の送信データ長を相手機器の最大ウィンドウサイズ(受信バッファ)以下にしてください。
 相手機器の最大ウィンドウサイズを超えるデータの送信はできません。

- (s1)で指定した接続の相手機器へ、(s3)で設定したデータを送信します。



- 本命令の実行および正常/異常完了は、設定データで指定した完了デバイス(d)、完了時の状態表示デバイス(d)+1で確認できます。

- 完了デバイス(d)

本命令が完了したスキャンのEND処理でONし、次のEND処理でOFFします。

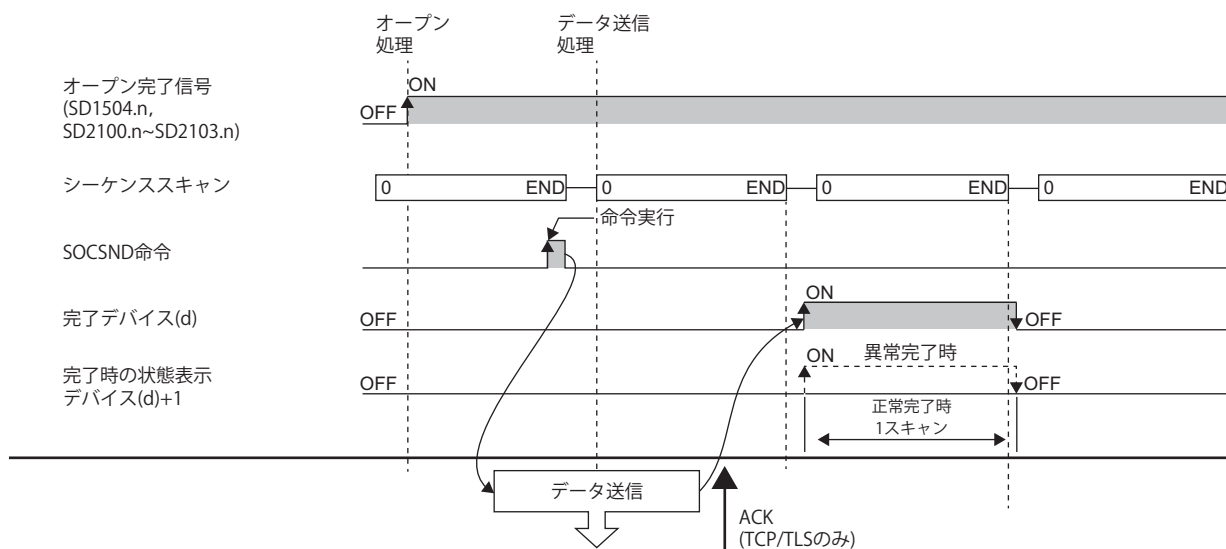
- 完了時の状態表示デバイス(d)+1

本命令の完了したときの状態により、ON/OFFします。

正常完了時: OFFのまま変化しません。

異常完了時: 本命令が完了したスキャンのEND処理でONし、次のEND処理でOFFします。

- 本命令を使用した送信処理のタイミングは下記になります。



完了デバイスON後も、送信が続く場合があります。送信完了の確認が必要な場合は、受信側で完了を確認するなどしてください。

注意事項

本命令を実行後、送信データが相手機器に到達し、データ送信が正常に完了したことを確認する、到達確認手段を自局と相手機器の双方に作成してください。

コントローラがTCP/IPまたはTLSプロトコルで相手機器とコネクションを確立後、本命令を実行して相手機器にデータ送信を行った場合、相手機器への到達を確認することなく本命令実行を完了し、完了ステータス((S2)+1)は正常完了になります。例えば、正常にデータ送信ができたことを相手機器に通知するプログラムにより到達を相手機器に通知します。

エラー

エラーコード (SD0)	内容
3285H	(s1)で指定するコネクション番号が範囲外の時。
3293H	(s3)で指定する送信データ長が範囲外の時。

異常完了時は、完了時の状態表示デバイス(d)+1がONになり、完了ステータス(s2)+1にエラーコードが格納されます。

完了ステータス(s2)+1に格納されるエラーコードについては、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

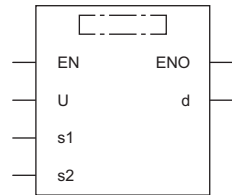
コネクション情報の読出し

GP.SOCCINF, SP.SOCCINF

(s1)で指定したコネクションのコネクション情報を読み出し、(d)で指定されたデバイス以降に格納します。

ラダー	ST
	ENO:=GP_SOCCINF(EN,U,s1,s2,d); ENO:=SP_SOCCINF(EN,U,s1,s2,d);

FBD/LD



■実行条件

命令	実行条件
GP.SOCCINF SP.SOCCINF	

設定データ

■内容, 範囲, データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(U)	GP.SOCCINF	対象の先頭入出力番号 (先頭入出力番号の上3桁を指定)	3E0H	符号なしBIN16ビット	ANY16
	SP.SOCCINF	ダミー	—	文字列	ANYSTRING_SINGLE
(s1)		コネクション番号	1~64	符号なしBIN16ビット	ANY16
(s2)	GP.SOCCINF	コントロールデータを格納する先頭デバイス	コントロールデータ参照	デバイス	ANY16
	SP.SOCCINF			ワード	ANY16_ARRAY (要素数: 2)
(d)		コネクション情報を格納する先頭デバイス	—	ワード	ANY16_ARRAY (要素数: 5)
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド		ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
		X, Y, M, L, SM, F, B, SB, FX, FY	J0 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U0 ¥ G0, J0 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(U)	GP.SOCCINF	—	—	○	—	—	—	—	○	○	—	—	○
	SP.SOCCINF	—	—	○	—	—	—	—	○	—	—	○	○
(s1)	GP.SOCCINF	○	—	○	—	—	—	—	○	○	—	—	—
	SP.SOCCINF	—	—	○	—	—	—	—	○	○	—	—	—
(s2)		—	—	○	—	—	—	—	○	—	—	—	—
(d)		—	—	○	—	—	—	—	○	—	—	—	—

■コントロールデータ

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+0	システムエリア	—	—	—
+1	完了ステータス	命令完了時に完了ステータスを格納します。 • 0000H: 正常完了 • 0000H以外: 異常完了(エラーコード)	—	システム

オペランド: (d)				
デバイス	項目	内容	設定範囲	セット側
+0 +1	相手機器IPアドレス	相手機器のIPアドレスを格納します。	00000001H~FFFFFFFFH 00000000H: 交信相手なし (FFFFFFFFH: 一斉同報通信)	システム
+2	相手機器ポート番号	相手機器のポート番号を格納します。	0001H~FFFFH(FFFFH: 一斉同報通信)	システム
+3	自局ポート番号	自局ポート番号を格納します。	0001H~1387H 1392H~FFFEH	システム
+4	コネクション使用用途	コネクションの使用用途を格納します。 b15 b14 ... b10 b9 b8 b7 ... b0 (d)+4 (3) 0 (4)(2) (1) 0 (1) 通信方式(プロトコル)(ビット8, 7) • 00: TCP • 01: TLS • 10: UDP • 11: DTLS (2) ソケット通信機能の手順有無(ビット9) • 1: 無手順(固定) (3) オープン方式(ビット15, 14) • 00: ActiveオープンまたはUDP • 10: Unpassiveオープン • 11: Fullpassiveオープン (4) 通信プロトコル設定(ビット10) • 0: 通信プロトコル支援機能を使用しない。(ソケット通信機能を使用する。) • 1: 通信プロトコル支援機能を使用する。	—	システム

機能

(s1)で指定したコネクションのコネクション情報を読み出します。

Point

本命令は暗号化通信プロトコル(TLSとDTLS)に対応しています。詳細は使用するコントローラのユーザーズマニュアルを確認してください。

TCP/TLS/DTLSのPassive方式において、コネクションがオープン状態かつ、相手機器とコネクション未確立の状態で本命令を実行すると、下記のように相手機器の情報以外のデータが格納されます。また、コネクションが非オープン状態の場合は0Hが格納されます。

デバイス	項目	格納されるデータ	
		コネクションがオープン状態かつ、相手機器とコネクション未確立の状態	コネクションが非オープン状態
d+0 d+1	相手機器IPアドレス	00000000H	00000000H
d+2	相手機器ポート番号	0000H	0000H
d+3	自局ポート番号	オープン時のデータ	0000H
d+4	コネクション使用用途	オープン時のデータ	0000H

エラー

エラーコード (SD0)	内容
3285H	(s1)で指定するコネクション番号が範囲外の時。

異常完了時は、完了ステータス(s2)+1にエラーコードが格納されます。

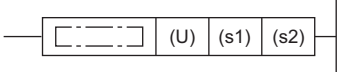
完了ステータス(s2)+1に格納されるエラーコードについては、下記を参照してください。

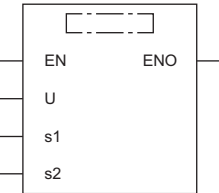
📖使用するコントローラのユーザーズマニュアル

コネクションの発信先変更(UDP/IP)

GP.SOCCSET, SP.SOCCSET

指定したコネクションの発信相手IPアドレス, 発信相手ポート番号を変更します。

ラダー	ST
	ENO:=GP_SOCCSET(EN,U,s1,s2); ENO:=SP_SOCCSET(EN,U,s1,s2);

FBD/LD


■実行条件

命令	実行条件
GP.SOCCSET SP.SOCCSET	

設定データ

■内容, 範囲, データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(U)	GP.SOCCSET 対象の先頭入出力番号 (先頭入出力番号の上3桁を指定)	3E0H	符号なしBIN16ビット	ANY16
	SP.SOCCSET ダミー	—	文字列	ANYSTRING_SINGLE
(s1)	コネクション番号	1~64	符号なしBIN16ビット	ANY16
(s2)	GP.SOCCSET コントロールデータを格納する先頭デバイス	コントロールデータ参照	デバイス	ANY16
	SP.SOCCSET		ワード	ANY16_ARRAY (要素数: 5)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド		ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
		X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ		K, H	E	\$	
(U)	GP.SOCCSET	—	—	○	—	—	—	—	○	○	—	—	○
	SP.SOCCSET	—	—	○	—	—	—	—	○	—	—	○	○
(s1)	GP.SOCCSET	○	—	○	—	—	—	—	○	○	—	—	—
	SP.SOCCSET	—	—	○	—	—	—	—	○	○	—	—	—
(s2)		—	—	○	—	—	—	—	○	—	—	—	—

■コントロールデータ

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+0	システムエリア	—	—	—
+1	完了ステータス	命令完了時に完了ステータスを格納します。 ・ 0000H: 正常完了 ・ 0000H以外: 異常完了(エラーコード)	—	システム
+2 +3	相手機器IPアドレス	相手機器のIPアドレスを格納します。	00000001H~FFFFFFFFH (FFFFFFFFH: 一斉同報通信)	ユーザ
+4	相手機器ポート番号	相手機器のポート番号を格納します。	0001H~FFFFH (FFFFH: 一斉同報通信)	ユーザ

機能

UDP通信時、(s1)で指定した接続の通信相手IPアドレス、通信相手ポート番号を変更します。

Point

- ・ 本命令を使用すると、接続をクローズしなくても通信相手を変更できます。
- ・ 命令実行後、相手機器のIPアドレスとポート番号が変更されるタイミングが、次のスキャン以降になる場合があります。SOCCINF命令を使用し、本命令で指定した値になっているか確認してください。

注意事項

SOCSND命令実行中に本命令で通信相手を変更しないようにしてください。

エラー

エラーコード (SD0)	内容
3285H	(s1)で指定する接続番号が範囲外のとき。

異常完了時は、完了ステータス(s2)+1にエラーコードが格納されます。

完了ステータス(s2)+1に格納されるエラーコードについては、下記を参照してください。

📖使用するコントローラのユーザズマニュアル

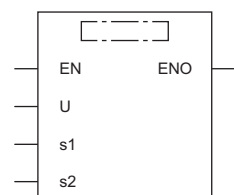
コネクションの受信モード変更

GP.SOCRMODE, SP.SOCRMODE

指定したコネクションに対して、TCP/TLS受信モード(UDP/DTLS通信のコネクションに対しては無効)および受信データサイズを変更します。

ラダー	ST
	ENO:=GP_SOCRMODE(EN,U,s1,s2); ENO:=SP_SOCRMODE(EN,U,s1,s2);

FBD/LD



■実行条件

命令	実行条件
GP.SOCRMODE SP.SOCRMODE	

設定データ

■内容, 範囲, データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(U)	GP.SOCRMODE	対象の先頭入出力番号 (先頭入出力番号の上3桁を指定)	3E0H	符号なしBIN16ビット	ANY16
	SP.SOCRMODE	ダミー	—	文字列	ANYSTRING_SINGLE
(s1)		コネクション番号	1~64	符号なしBIN16ビット	ANY16
(s2)	GP.SOCRMODE	コントロールデータを格納する先頭デバイス	コントロールデータ参照	デバイス	ANY16
	SP.SOCRMODE			ワード	ANY16_ARRAY (要素数: 4)
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド		ビット		ワード				ダブルワード		間接指定	定数			その他 (U)
		X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(U)	GP.SOCRMODE	—	—	○	—	—	—	—	○	○	—	—	○	○
	SP.SOCRMODE	—	—	○	—	—	—	—	○	—	—	○	○	○
(s1)	GP.SOCRMODE	○	—	○	—	—	—	—	○	○	—	—	—	—
	SP.SOCRMODE	—	—	○	—	—	—	—	○	○	—	—	—	—
(s2)		—	—	○	—	—	—	—	○	—	—	—	—	—

■コントロールデータ

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+0	システムエリア	—	—	—
+1	完了ステータス	命令完了時に完了ステータスを格納します。 ・ 0000H: 正常完了 ・ 0000H以外: 異常完了(エラーコード)	—	システム
+2	TCP/TLS受信モード*1	TCP/TLS受信モードを格納します。 ・ 0: TCP/TLS通常受信モード ・ 1: TCP/TLS固定長受信モード	0, 1	ユーザ
+3	受信データサイズ	ソケット通信受信データサイズを格納します。(バイト数)	1~10238	ユーザ

*1 UDP/DTLS通信のコネクションに対しては無効となります。

機能

- ・ (s1)で指定したコネクションに対して、UDP/DTLS通信のコネクション以外のTCP/TLS受信モードおよび受信データサイズを変更します。
- ・ TCP/TLSのコネクションに対して(s2)+2で指定したモードとなります。

Point

- ・ 本命令は暗号化通信プロトコル(TLS)に対応しています。詳細は使用するコントローラのユーザーズマニュアルを確認してください。
- ・ 接続機器からのデータ受信は、本命令を実行してから100ms以上間隔を空けてから開始してください。

■TCP/TLS通常受信モード

データを受信すると、データをソケット通信受信データエリアに格納して、SD1506, SD2132~SD2135(ソケット通信受信状態信号)をONします。

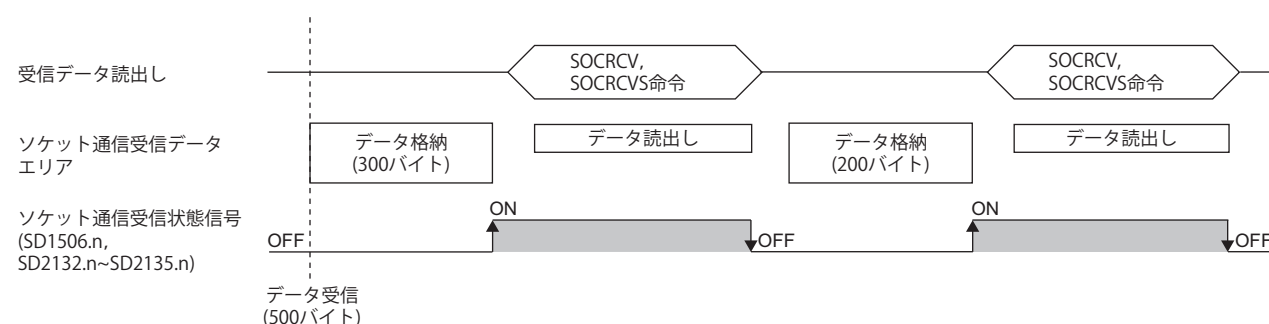
受信したデータが指定された受信データサイズを超える場合は、超えた分の受信データは次の受信データとなります。

その後、SOCRCV命令/SOCRCVS命令でソケット通信受信データエリアからデータを読み出すまでにデータを受信した場合は、OS内受信データに格納されます。

また、SOCRCV命令/SOCRCVS命令でソケット通信受信データエリアからデータを取り出したとき、OS内受信データに格納されたデータがあればソケット通信受信データエリアに格納し、SD1506, SD2132~SD2135(ソケット通信受信状態信号)をONします。

例

受信データサイズを300バイトにして、500バイト受信した場合



■TCP/TLS固定長受信モード

データを受信すると、データをソケット通信受信データエリアに格納します。指定された受信データサイズに達していない場合は、SD1506, SD2132~SD2135(ソケット通信受信状態信号)はONしません。

受信データが受信データサイズに達するまで受信を繰り返し、受信データサイズに達した時点でSD1506, SD2132~SD2135(ソケット通信受信状態信号)がONになります。

受信したデータが指定された受信データサイズを超える場合は、超えた分の受信データは次の受信データになります。

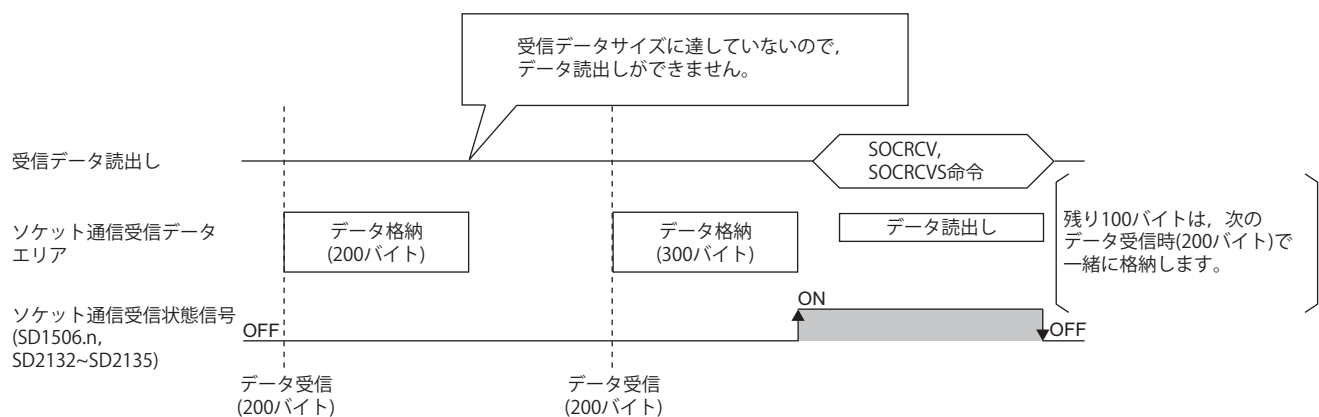
その後、SOCRCV命令/SOCRCVS命令でソケット通信受信データエリアからデータを読み出すまでにデータを受信した場合は、OS内受信データに格納されます。

SOCRCV命令/SOCRCVS命令でソケット通信受信データエリアからデータを取り出したとき、OS内受信データに格納されたデータがあれば、ソケット通信受信データエリアに格納しますが、指定された受信データサイズに達していない場合は、SD1506, SD2132~SD2135(ソケット通信受信状態信号)はONしません。

その後、受信データが受信データサイズに達するまで受信を繰り返し、受信データサイズに達した時点でSD1506, SD2132~SD2135(ソケット通信受信状態信号)がONになります。

例

受信データサイズを300バイトにして、200バイト連続受信した場合



Point

• デバイスの有効活用

SOCRCV命令/SOCRCVS命令で使用する受信データの格納用デバイスは、最大で5119ワード(10238バイト)を確保しなければなりませんが、受信データサイズを抑えることでデバイスを有効活用できます。

• 受信データの分割防止

回線種類などにより、相手機器からのデータが分割されて到達する場合に、TCP/TLS固定長受信モードで受信データサイズを指定して受信することで、データが分割されることを防ぐことができます。

• 受信データの結合防止

シーケンスプログラムの受信処理の遅れなどにより、相手機器によっては、分けて送信したデータが結合されてしまう場合があります。

TCP/TLS固定長受信モードで受信データサイズを指定して受信することで、正しく分けて受信することができます。

- (s2)+3で、1回のSOCRCV命令/SOCRCVS命令で読み出す受信データサイズを指定します。UDP/DTLSの場合、受信したデータが指定された受信データサイズを超える場合は、超えた分の受信データは次の受信データになります。

エラー

エラーコード (SD0)	内容
3285H	(s1)で指定するコネクション番号が範囲外の時。

異常完了時は、完了ステータス(s2)+1にエラーコードが格納されます。

完了ステータス(s2)+1に格納されるエラーコードについては、下記を参照してください。

 使用するコントローラのユーザズマニュアル

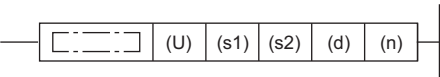
Point

TCP/TLS固定長受信モードで、受信状態信号がONしないときに、現時点で受信している分のデータをSOCDATA命令で読み出すことができ、相手機器からの送信データが不足しているかを確認できます。

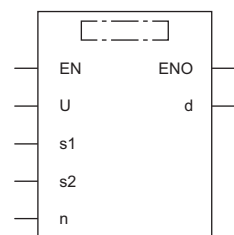
ソケット通信受信データリード

G(P).SOCRDATA, S(P).SOCRDATA

(s1)で指定した接続のソケット通信受信データエリアから(n)で指定したワード分のデータを読み出し、(d)で指定されたデバイス以降に格納します。

ラダー	ST
	ENO:=G_SOCRDATA(EN,U,s1,s2,n,d); ENO:=GP_SOCRDATA(EN,U,s1,s2,n,d); ENO:=S_SOCRDATA(EN,U,s1,s2,n,d); ENO:=SP_SOCRDATA(EN,U,s1,s2,n,d);

FBD/LD



■実行条件

命令	実行条件
G.SOCRDATA S.SOCRDATA	
GP.SOCRDATA SP.SOCRDATA	

設定データ

■内容、範囲、データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(U)	G(P).SOCRDATA	対象の先頭入出力番号 (先頭入出力番号の上3桁を指定)	3E0H	符号なしBIN16ビット	ANY16
	S(P).SOCRDATA	ダミー	—	文字列	ANYSTRING_SINGLE
(s1)		コネクション番号	1~64	符号なしBIN16ビット	ANY16
(s2)	G(P).SOCRDATA	コントロールデータを格納する先頭デバイス	コントロールデータ参照	デバイス	ANY16
	S(P).SOCRDATA			ワード	ANY16_ARRAY (要素数: 2)
(d)		読み出したデータを格納する先頭デバイス	—	ワード	ANY16 ^{*1}
(n)		読出しデータ数	1~5120	符号なしBIN16ビット	ANY16
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

■使用可能デバイス

オペランド		ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
		X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(U)	G(P).SOCRDATA	—	—	○	—	—	—	—	○	○	—	—	○
	S(P).SOCRDATA	—	—	○	—	—	—	—	○	—	—	○	○
(s1)	G(P).SOCRDATA	○	—	○	—	—	—	—	○	○	—	—	—
	S(P).SOCRDATA	—	—	○	—	—	—	—	○	○	—	—	—
(s2)		—	—	○	—	—	—	—	○	—	—	—	—
(d)		—	—	○	—	—	—	—	○	—	—	—	—
(n)		○	—	○	—	—	—	—	○	○	—	—	—

■コントロールデータ

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+0	システムエリア	—	—	—
+1	完了ステータス	命令完了時に完了ステータスを格納します。 • 0000H: 正常完了 • 0000H以外: 異常完了(エラーコード)	—	システム

機能

(s1)で指定した接続のソケット通信受信データエリアから、(n)で指定したワード分のデータを読み出し、(d)で指定されたデバイス以降に格納します。読出しデータ(n)が0のときは無処理です。

Point

- 本命令は暗号化通信プロトコル(TLSとDTLS)に対応しています。詳細は使用するコントローラのユーザーズマニュアルを確認してください。
- 読出しデータ数を1ワードにすると、受信データ長を読み出すことができます。これにより、SOCRCV命令/SOCRCVS命令実行時に受信データを格納するデバイスを変更することができます。
- 本命令で今回受信するデータをチェックし、SOCRMODE命令で次回に受信するデータサイズを指定してから、SOCRCV命令/SOCRCVS命令で今回分を読み出すことができます。これにより、今回の受信したデータをもとに、次回の受信するデータサイズを指定することができます。

エラー

エラーコード (SD0)	内容
3285H 3293H	(s1)で指定する接続番号が範囲外のとき。
	(n)で指定する読出しデータ数が範囲の最大値を超えるとき。

異常完了時は、完了ステータス(s2)+1にエラーコードが格納されます。

完了ステータス(s2)+1に格納されるエラーコードについては、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

26.3 通信プロトコル支援機能命令

Point

下記の命令を実行中のコネクションは、重複して下記の命令を実行しても要求は無視されます。

この時、後から実行した命令の実行条件が成立している間は、先に実行している命令が完了した後で要求が実行されます。また、要求が無視されている間、SM699(専用命令未実行フラグ)はONになります。

- SOCOPEN命令
- SOCCLOSE命令
- SOCSND命令
- SOCRCV命令
- ECPRTCL命令

登録プロトコル実行

GP.ECPRTCL, SP.ECPRTCL

エンジニアリングツールで登録したプロトコルを実行します。

ラダー	ST
	<pre>ENO:=GP_ECPRTCL(EN,U,s1,s2,s3,d); ENO:=SP_ECPRTCL(EN,U,s1,s2,s3,d);</pre>

FBD/LD

■実行条件

命令	実行条件
GP.ECPRTCL SP.ECPRTCL	

設定データ

■内容、範囲、データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(U)	GP.ECPRTCL	対象の先頭入出力番号 (先頭入出力番号の上3桁を指定)	3E0H	符号なしBIN16ビット	ANY16
	SP.ECPRTCL	ダミー	—	文字列	ANYSTRING_SINGLE
(s1)		コネクション番号	1~64	符号なしBIN16ビット	ANY16
(s2)		連続実行するプロトコルの数	1~8	符号なしBIN16ビット	ANY16
(s3)	GP.ECPRTCL	コントロールデータを格納する先頭デバイス	コントロールデータ参照	デバイス	ANY16_ARRAY (要素数: 18)
	SP.ECPRTCL			ワード	ANY16_ARRAY (要素数: 2)
(d)		命令完了にて1スキャンONするデバイス 異常完了時は(d)+1もONします。	—	ビット	ANYBIT_ARRAY (要素数: 2)
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

■使用可能デバイス

オペランド		ビット		ワード				ダブルワード		間接指定	定数			その他 (U)
		X, Y, M, L, SM, F, B, SB, FX, FY	J□¥□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□¥G□, J□¥□	Z	LT, LST, LC	LZ			K, H	E	\$	
(U)	GP.ECPRTCL	—	—	○	—	—	—	—	○	○	—	—	○	○
	SP.ECPRTCL	—	—	○	—	—	—	—	○	—	—	○	○	○
(s1)		○	—	○*1	—	—	—	—	○*1	○	—	—	—	—
(s2)		○	—	○*1	—	—	—	—	○*1	○	—	—	—	—
(s3)	GP.ECPRTCL	—	—	○*1	—	—	—	—	○*1	—	—	—	—	—
	SP.ECPRTCL	○*2	—	○*1	—	—	—	—	○*1	—	—	—	—	—
(d)		○*1	—	○	—	—	—	—	○*1	—	—	—	—	—

*1 ローカルデバイスおよびプログラムごとに設定したファイルレジスタは使用できません。

*2 16ビット(4桁)の桁指定の場合かつ、開始ビットデバイスが16の倍数(10H)の場合のみ使用できます。

■コントロールデータ

オペランド: (s3)				
デバイス	項目	内容	設定範囲	セット側
+0	実行数結果	本命令で実行したプロトコル数を格納します。エラーの発生したプロトコルも、実行数に含みます。設定データ、コントロールデータの設定内容に誤りがある場合は「0」を格納します。	0, 1~8	システム
+1	完了ステータス	命令完了時に完了ステータスを格納します。 プロトコルを複数実行する場合は、最後に実行したプロトコルの実行結果を格納します。 ・0: 正常完了 ・0以外: 異常完了(エラーコード)	—	システム
+2	実行プロトコル番号指定1	1番目に実行するプロトコルのプロトコル番号を指定します。	1~128	ユーザ
+3	実行プロトコル番号指定2	2番目に実行するプロトコルのプロトコル番号を指定します。	0, 1~128	ユーザ
+4	実行プロトコル番号指定3	3番目に実行するプロトコルのプロトコル番号を指定します。	0, 1~128	ユーザ
+5	実行プロトコル番号指定4	4番目に実行するプロトコルのプロトコル番号を指定します。	0, 1~128	ユーザ
+6	実行プロトコル番号指定5	5番目に実行するプロトコルのプロトコル番号を指定します。	0, 1~128	ユーザ
+7	実行プロトコル番号指定6	6番目に実行するプロトコルのプロトコル番号を指定します。	0, 1~128	ユーザ
+8	実行プロトコル番号指定7	7番目に実行するプロトコルのプロトコル番号を指定します。	0, 1~128	ユーザ
+9	実行プロトコル番号指定8	8番目に実行するプロトコルのプロトコル番号を指定します。	0, 1~128	ユーザ
+10	照合一致 受信パケット番号1	1番目に実行したプロトコルの通信タイプに受信が含まれる場合は、照合一致した受信パケット番号を格納します。通信タイプが「送信のみ」の場合は、「0」を格納します。1番目のプロトコル実行時にエラーが発生した場合は「0」を格納します。	0, 1~16	システム
+11	照合一致 受信パケット番号2	2番目に実行したプロトコルの通信タイプに受信が含まれる場合は、照合一致した受信パケット番号を格納します。通信タイプが「送信のみ」の場合は、「0」を格納します。2番目のプロトコル実行時にエラーが発生した場合は「0」を格納します。実行したプロトコル数が2未満のときは「0」を格納します。	0, 1~16	システム
+12	照合一致 受信パケット番号3	3番目に実行したプロトコルの通信タイプに受信が含まれる場合は、照合一致した受信パケット番号を格納します。通信タイプが「送信のみ」の場合は、「0」を格納します。3番目のプロトコル実行時にエラーが発生した場合は「0」を格納します。実行したプロトコル数が3未満のときは「0」を格納します。	0, 1~16	システム
+13	照合一致 受信パケット番号4	4番目に実行したプロトコルの通信タイプに受信が含まれる場合は、照合一致した受信パケット番号を格納します。通信タイプが「送信のみ」の場合は、「0」を格納します。4番目のプロトコル実行時にエラーが発生した場合は「0」を格納します。実行したプロトコル数が4未満のときは「0」を格納します。	0, 1~16	システム
+14	照合一致 受信パケット番号5	5番目に実行したプロトコルの通信タイプに受信が含まれる場合は、照合一致した受信パケット番号を格納します。通信タイプが「送信のみ」の場合は、「0」を格納します。5番目のプロトコル実行時にエラーが発生した場合は「0」を格納します。実行したプロトコル数が5未満のときは「0」を格納します。	0, 1~16	システム
+15	照合一致 受信パケット番号6	6番目に実行したプロトコルの通信タイプに受信が含まれる場合は、照合一致した受信パケット番号を格納します。通信タイプが「送信のみ」の場合は、「0」を格納します。6番目のプロトコル実行時にエラーが発生した場合は「0」を格納します。実行したプロトコル数が6未満のときは「0」を格納します。	0, 1~16	システム
+16	照合一致 受信パケット番号7	7番目に実行したプロトコルの通信タイプに受信が含まれる場合は、照合一致した受信パケット番号を格納します。通信タイプが「送信のみ」の場合は、「0」を格納します。7番目のプロトコル実行時にエラーが発生した場合は「0」を格納します。実行したプロトコル数が7未満のときは「0」を格納します。	0, 1~16	システム
+17	照合一致 受信パケット番号8	8番目に実行したプロトコルの通信タイプに受信が含まれる場合は、照合一致した受信パケット番号を格納します。通信タイプが「送信のみ」の場合は、「0」を格納します。8番目のプロトコル実行時にエラーが発生した場合は「0」を格納します。実行したプロトコル数が8未満のときは「0」を格納します。	0, 1~16	システム

機能

- エンジニアリングツールで登録したプロトコルを実行します。(s1)で指定されたコネクションを使用し、実行するプロトコルは、(s3)で指定されたデバイス以降のコントロールデータに従います。
- 1回の命令実行で、(s2)で指定した数(最大8)のプロトコルを連続実行します。
- プロトコルを実行した数は、実行数結果(s3)+0に格納します。
- プロトコルの実行状態は、通信プロトコル支援機能実行状態確認用エリア(Un ¥ G350~Un ¥ G669)で確認することができます。(使用使用するコントローラのユーザーズマニュアル)
- 本命令実行および正常/異常完了は、設定データで指定した完了デバイス(d), 完了時の状態表示デバイス(d)+1で確認できます。

• 完了デバイス(d)

本命令が完了したスキャンのEND処理でONし、次のEND処理でOFFします。

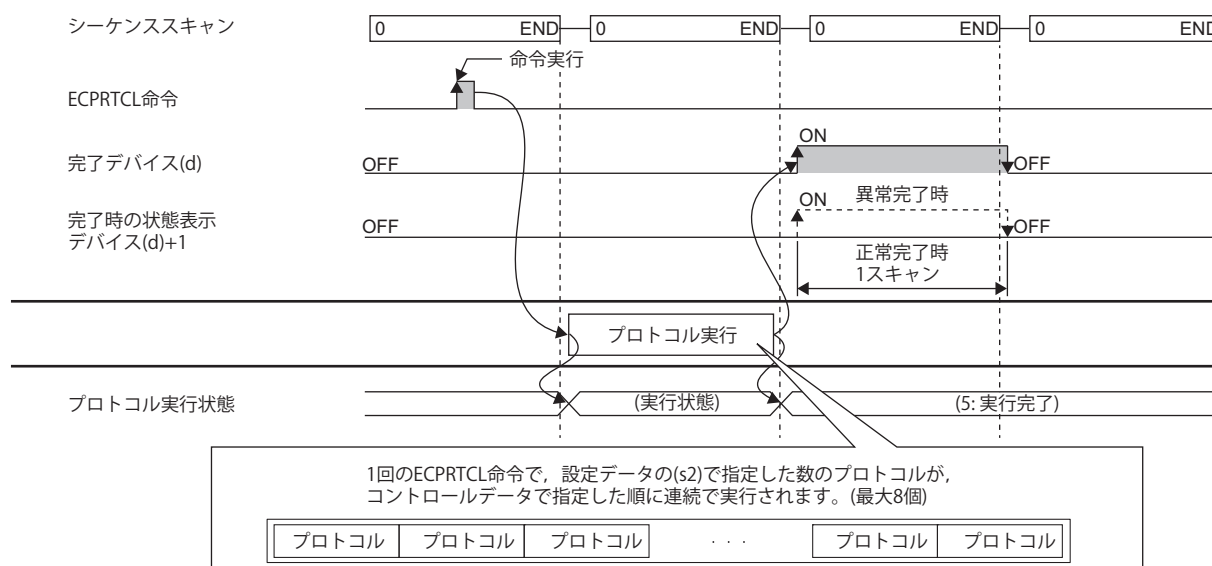
• 完了時の状態表示デバイス(d)+1

本命令の完了したときの状態により、ON/OFFします。

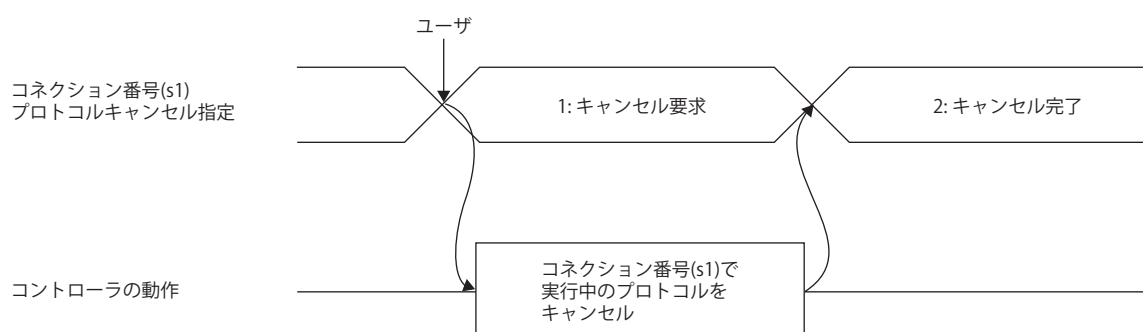
正常完了時: OFFのまま変化しません。

異常完了時: 本命令が完了したスキャンのEND処理でONし、次のEND処理でOFFします。異常完了時には、(s3)+1の完了ステータスにエラーコードを格納します。

- 本命令の実行タイミングは下記となります。

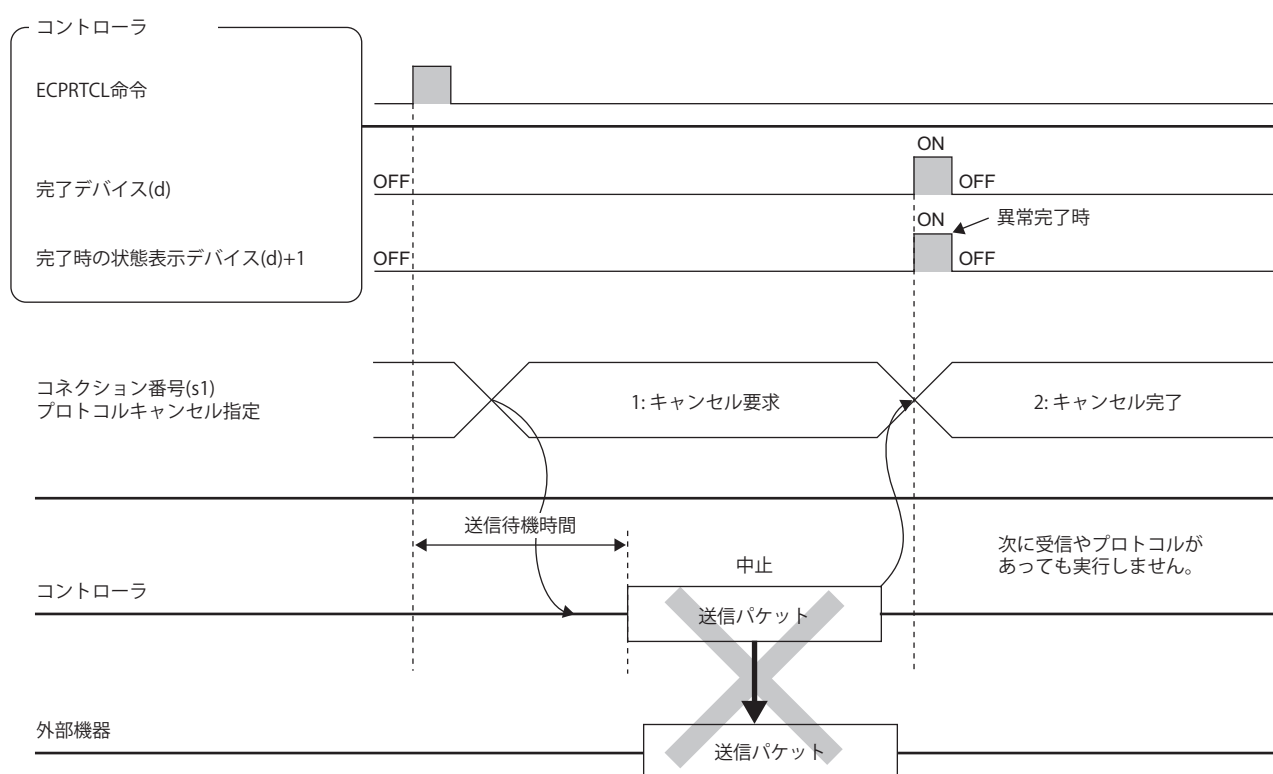


- プロトコルキャンセル要求をセットすることで、プロトコルの実行がキャンセルできます。プロトコルキャンセル指定は、通信プロトコル支援機能実行状態確認用エリア(Un ¥ G350~Un ¥ G669)を使用します。( 使用するコントローラのユーザーズマニュアル)

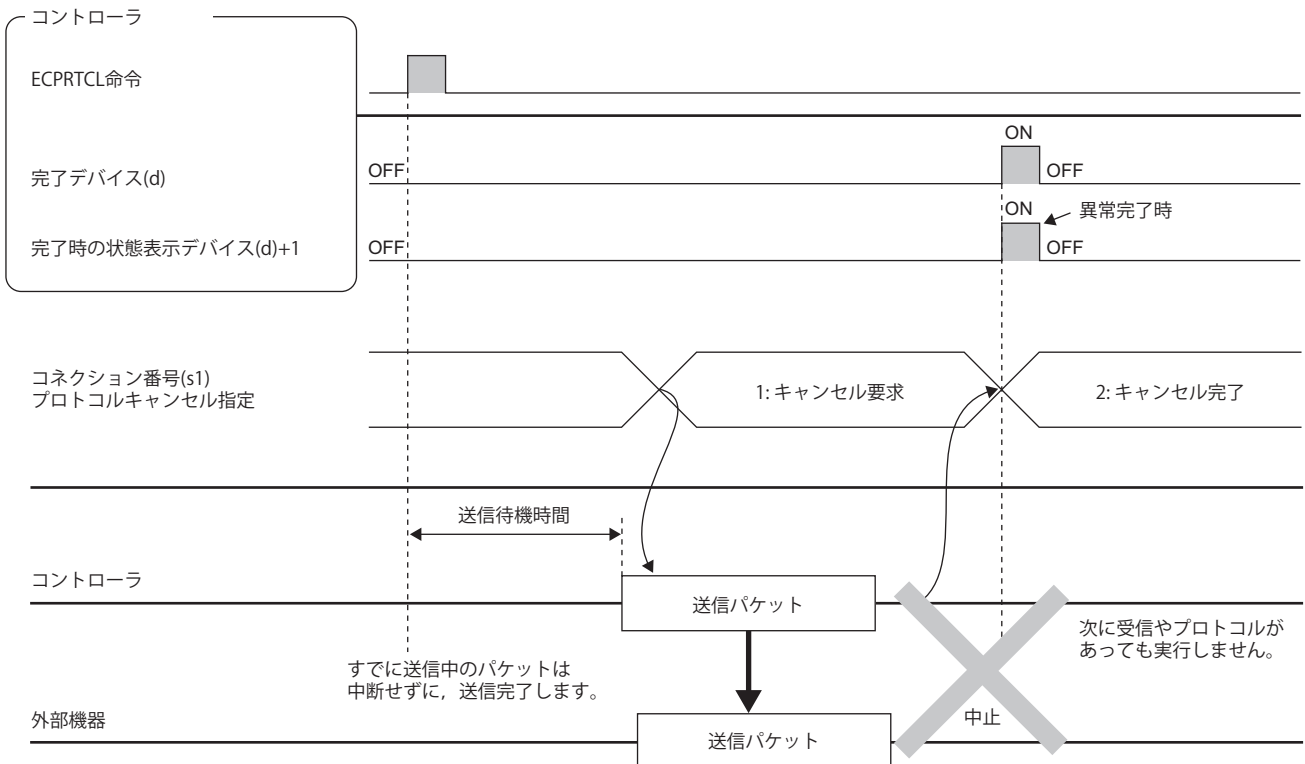


- 各タイミングにおける、プロトコルキャンセルの動作は下記になります。

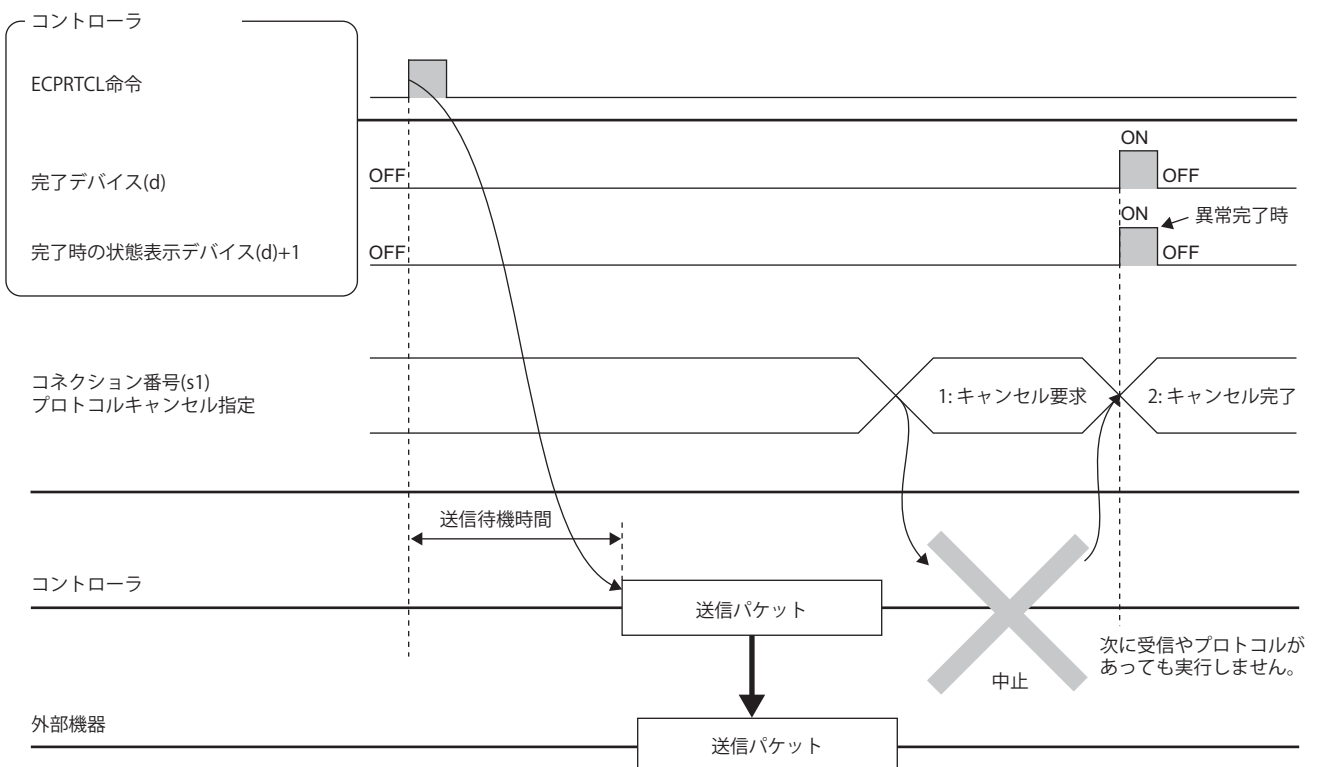
- 送信前にキャンセル要求があった場合
プロトコル実行状態が「1: 送信待ち」の場合の動作は下記になります。



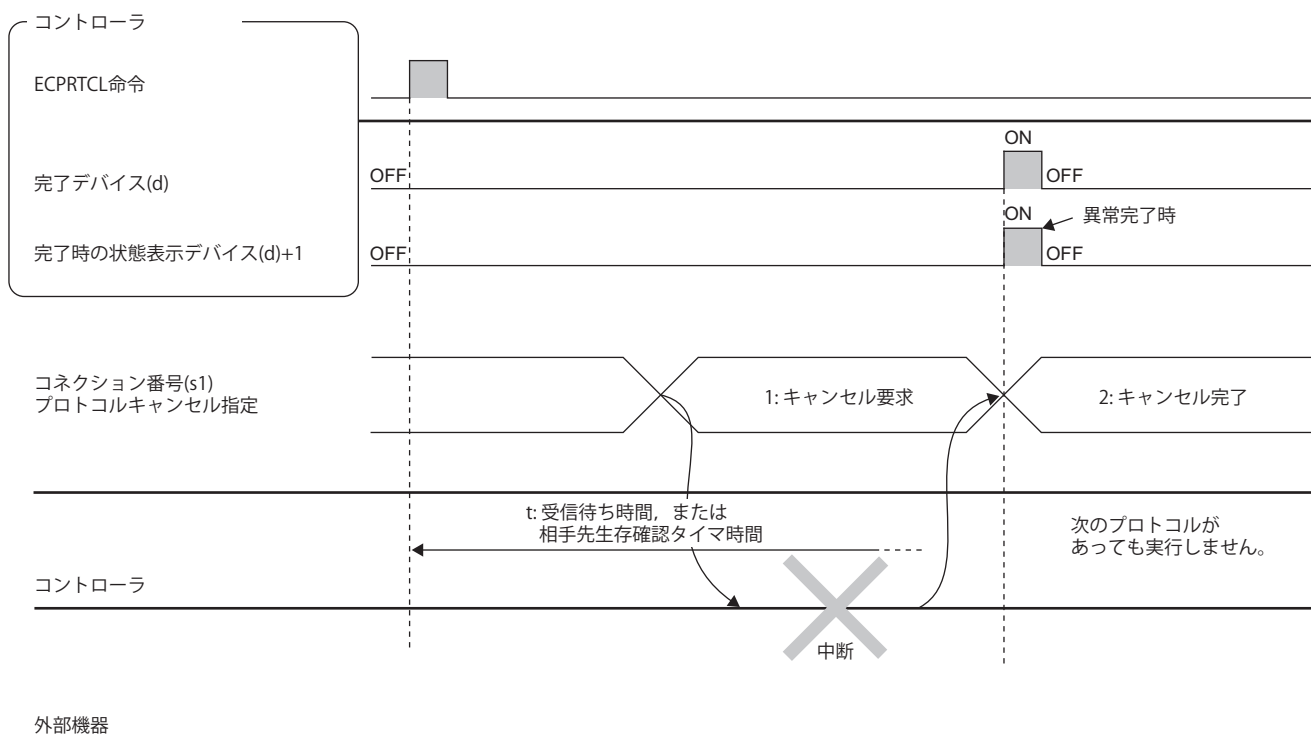
- 送信完了前にキャンセル要求があった場合
 プロトコル実行状態が「2: 送信中」で、送信が完了していない場合の動作は下記になります。



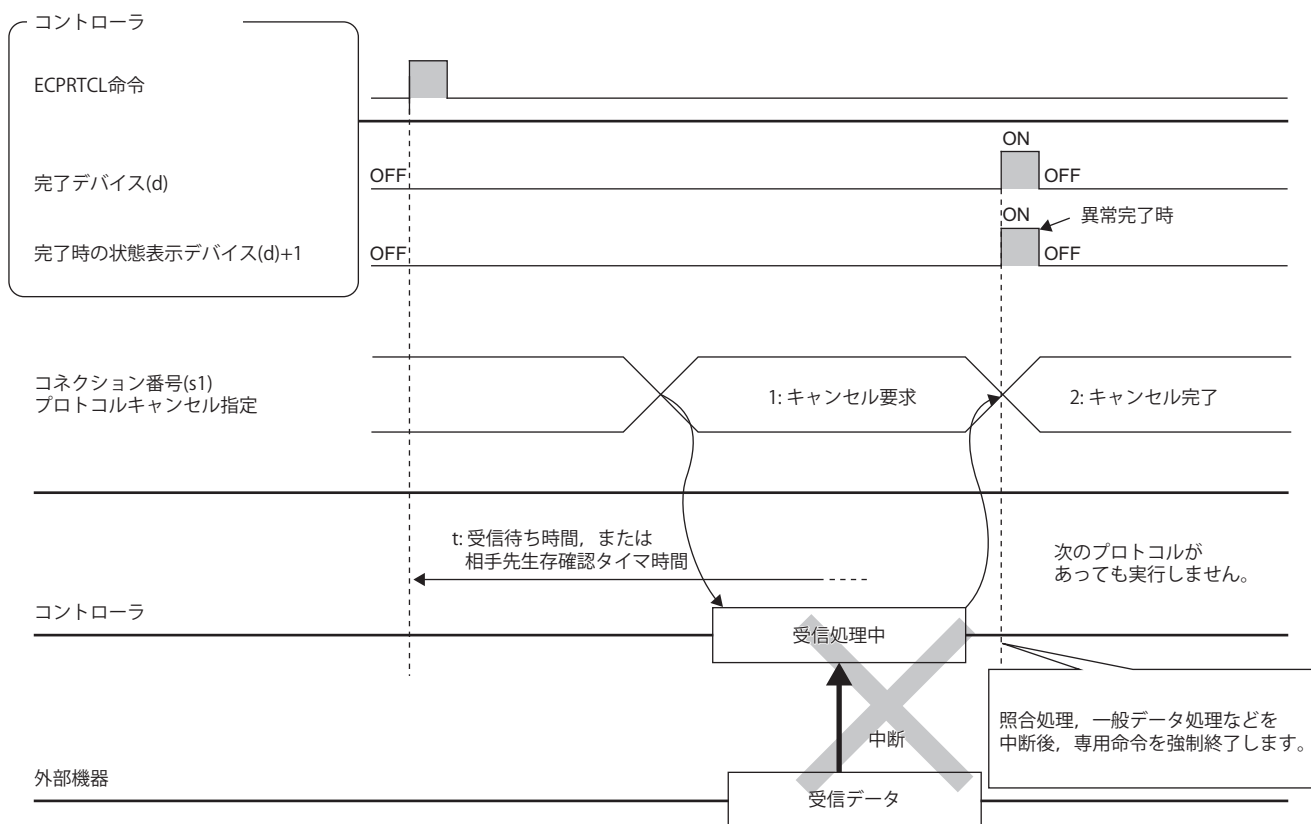
- 送信完了時にキャンセル要求があった場合
 プロトコル実行状態が「2: 送信中」で、送信が完了している場合の動作を下記に示します。



- 受信待ち中にキャンセル要求があった場合
 プロトコル実行状態が「3: 受信データ待ち」の場合の動作を下記に示します。



- 受信処理中にキャンセル要求があった場合
 プロトコル実行状態が「4: 受信処理中」の場合の動作を下記に示します。



エラーコード (SD0)	内容
3285H	(s1)が範囲外の時。

異常完了時は、完了時の状態表示デバイス(d)+1がONになり、完了ステータス(s3)+1にエラーコードが格納されます。
完了ステータス(s3)+1に格納されるエラーコードについては、下記を参照してください。

📖使用するコントローラのユーザズマニュアル

注意事項

- 複数のプロトコルを実行する場合において、m番目のプロトコルでエラーが発生すると、m+1番目以降のプロトコルは実行せずに、命令を異常完了します。
- 変換なし変数を含むプロトコルを実行する場合、1パケット中に使用されている変数のデータ長の合計が1920バイトを超える場合、数スキャンにわたってCPUデバイス値を取得する可能性があるため、命令起動時から実行完了までは変換なし変数に指定したCPUデバイス値を変更しないでください。
- プロトコルキャンセルについて
 - 本命令は異常完了し、(s3)+1の完了ステータスにプロトコルキャンセル要求エラー (C404H)を格納します。
 - 実行中のプロトコルがない状態で、キャンセル要求をセットされた場合、コントローラは無処理でキャンセル完了します。
 - 通信プロトコルを使用しない場合は、キャンセル要求をセットされても無視します。
 - 複数のプロトコルを連続実行する場合において、n番目のプロトコル実行中にキャンセル要求されると、コントローラはn番目のプロトコルを強制終了し、またn+1番目以降のプロトコルを実行しません。実行数結果((s3)+0)に実行中のプロトコル番号nを、照合一致受信パケット番号1~(n-1)に照合一致した受信パケット番号を、完了ステータス((s3)+1)にはプロトコルキャンセル要求エラー (C404H)を格納します。
 - コントローラは、キャンセル要求があるかを定期処理で確認します。そのため、キャンセル要求指示からキャンセル処理が行われるまでに時間のかかる場合があります。
- 本命令単体では、接続のオープン/クローズを行わないため、SOCOPEN命令/SOCCLOSE命令による接続のオープン/クローズ処理が必要となります。

📖 1084ページ GP.SOCOPEN, SP.SOCOPEN

📖 1088ページ GP.SOCCLOSE, SP.SOCCLOSE

- 同一接続に対して同一命令を実行した場合、先に実行中の命令が完了するまで、あとの命令は無視され実行されません。
- 受信待ち時間を「0:無限待ち」に設定している場合、プロトコル設定で指定しているデータが受信されるまで本命令は完了しません。
- 通信タイプ"送信のみ"を選択しTCP/IPプロトコルで相手機器にデータ送信を行った場合、相手機器への到達を確認することなく本命令実行を完了し、(S3)+1の完了ステータスは正常完了になります。送信データが相手機器に到達したことを確認するために通信タイプ"送信&受信"を選択して、相手機器からの応答を待って命令が完了するようにしてください。

26.4 SLMPフレーム送信命令

SLMPフレーム送信

G(P).SLMPSND, J(P).SLMPSND, SP.SLMPSND

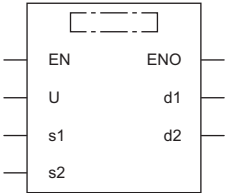
SLMP対応機器に対してSLMPの伝文を送信します。

Point

- G(P).SLMPSND命令は、Ethernet用ポートとCC-Link IE TSN用ポートの両方で使用できます。
- J(P).SLMPSND命令は、CC-Link IE TSN用ポートでのみ使用できます。
- SP.SLMPSND命令は、Ethernet用ポートでのみ使用できます。

ラダー	ST
	ENO:=G_SLMPSND(EN, U, s1, s2, d1, d2) ENO:=GP_SLMPSND(EN, U, s1, s2, d1, d2) ENO:=J_SLMPSND(EN, U, s1, s2, d1, d2) ENO:=JP_SLMPSND(EN, U, s1, s2, d1, d2) ENO:=SP_SLMPSND(EN, U, s1, s2, d1, d2)

FBD/LD



■実行条件

命令	実行条件
G.SLMPSND J.SLMPSND	
GP.SLMPSND JP.SLMPSND SP.SLMPSND	

設定データ

■内容、範囲、データ型

オペランド		内容	範囲	データ型	データ型(ラベル)
(U)	G(P).SLMPSND	対象の先頭入出力番号 (先頭入出力番号の上3桁を指定)	Ethernetポート:3E0H CC-Link IE TSNポート:00H~FEH	符号なしBIN16ビット	ANY16
	J(P).SLMPSND	自局ネットワークNo.	1~239	符号なしBIN16ビット	ANY16
	SP.SLMPSND	ダミー	—	文字列	ANYSTRING_SINGLE
(s1)	G(P).SLMPSND	コントロールデータを格納する先頭デバイス	📖 1122ページ コントロールデータ	デバイス	ANY16* ¹
	J(P).SLMPSND				
	SP.SLMPSND			ワード	
(s2)		要求フレームを格納する先頭デバイス	📖 1123ページ 要求フレーム	ワード	ANY16* ¹
(d1)		応答フレームを格納する先頭デバイス	📖 1123ページ 応答フレーム	ワード	ANY16_ARRAY* ²
(d2)		命令完了にて1スキャンONするデバイス 異常完了時は(d2)+1もONします。	—	ビット	ANYBIT_ARRAY* ² (要素数: 2)
EN		実行条件	—	ビット	BOOL
ENO		実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

*2 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義してください。

■使用可能デバイス

オペランド		ビット		ワード			ダブルワード		間接指定	定数				その他	
		X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	U	J	
(U)	G(P).SLMPSND	—	—	○	—	—	—	—	○	○	—	—	○	—	
	J(P).SLMPSND	—	—	—	—	—	—	—	—	—	—	—	—	○	
	SP.SLMPSND	—	—	○	—	—	—	—	○	—	—	○	○	—	
(s1)		—	—	○	—	—	—	—	○	—	—	—	—	—	
(s2)		—	—	○	—	—	—	—	○	—	—	—	—	—	
(d1)		—	—	○	—	—	—	—	○	—	—	—	—	—	
(d2)		○	—	○*1	—	—	—	—	○	—	—	—	—	—	

*1 T, ST, C, FDは使用できません。

■コントロールデータ

オペランド: (s1)				
デバイス	項目	内容	設定範囲	セット側
+0	実行・異常時完了タイプ	b15 b7 b0 (3) 0 (2) 0 (1) (1)実行タイプ(ビット0) 0: 到達確認なし(自局から要求伝文を送信したことで完了となります。)*¹ 1: 到達確認あり(相手機器から応答伝文を受信したことで完了となります。) (2)異常時完了タイプ(ビット7) 異常完了時のデータのセット状態を指定します。 0: (s1)+13以降に異常完了時のデータをセットしない。((s1)+13以降をクリアする。) 1: (s1)+13以降に異常完了時のデータをセットする。 (3)命令タイプ(ビット15) 0: オペランド(U)が00H~FEH指定の場合 1: オペランド(U)が3E0H指定の場合	—	ユーザ
+1	完了ステータス	命令完了時の状態が格納されます。 0000H: 正常 0000H以外: 異常(エラーコード)	—	システム
+2	自局使用チャンネル	自局が使用するチャンネルを指定します。チャンネルにより要求伝文にシリアル番号*⁴を付加する/付加しないが決まるため、用途に応じてチャンネルを指定してください。 SLMP SNDC命令と併用する場合、シリアル番号を付加するチャンネルは共有しているので重複しないように設定してください。 1: シリアル番号を付加しないチャンネル 2~9: シリアル番号を付加するチャンネル 10~17: 局番拡張フレーム	• 1~17* ²	ユーザ
+3	相手機器IPアドレス(第3, 4オクテット)	相手機器のIPアドレス(第3, 4オクテット)を指定します。 b15 b8 b7 b0 3 4 3, 4: IPアドレスのオクテットを示します。	(s1)+3, (s1)+4 合わせて, 00000001H~FFF FFFFEH (1~4294967294)	ユーザ
+4	相手機器IPアドレス(第1, 2オクテット)	相手機器のIPアドレス(第1, 2オクテット)を指定します。 b15 b8 b7 b0 1 2 1, 2: IPアドレスのオクテットを示します。		ユーザ
+5	相手機器ポート番号	相手機器のポート番号を指定します。	1~65534 (1~FFFFEH)	ユーザ
+6	要求先ネットワーク番号	0000H固定	0000H	ユーザ
+7	要求先局番	00FFH固定	00FFH	ユーザ
+8	要求先ユニットI/O番号	要求先のユニットI/O番号を指定します。 03FFH: 自局/管理CPU 03E0H: マルチCPU1号機 03E1H: マルチCPU2号機 03E2H: マルチCPU3号機 03E3H: マルチCPU4号機	03FFH, 03E0H~03E3H	ユーザ
+9	要求先マルチドロップ局番	相手機器のマルチドロップ番号を指定します。	0000H~FFFEH* ³	ユーザ
+10	再送回数* ⁵	(s1)+0で指定した実行タイプが"1: 到達確認あり"のときに有効になります。 ■命令実行時 (s1)+11で指定した監視時間内に応答伝文の受信が完了しない場合に、要求伝文を再送させる回数を指定します。 0~15(回) ■命令完了時 要求伝文の再送を行った回数(結果)が格納されます。 0~15(回)	0~15	ユーザ/システム
+11	到達監視時間	処理完了までの監視時間を指定します。監視時間内に完了しない場合は、(s1)+10で指定した回数まで再送されます。 0: 10秒 1~32767: 1~32767秒	0~32767	ユーザ

オペランド: (s1)				
デバイス	項目	内容	設定範囲	セット側
+12	時計セットフラグ	(s)+13以降のデータ有効/無効状態が格納されます。なお、正常完了した場合、(s1)+13以降のデータはクリアされます。 ・0: 無効 ・1: 有効	—	システム
+13	時計データ	上位8ビット: 月(01H~12H) 下位8ビット: 年(00H~99H)西暦下2桁	—	システム
+14		上位8ビット: 時(00H~23H) 下位8ビット: 日(01H~31H)		
+15		上位8ビット: 秒(00H~59H) 下位8ビット: 分(00H~59H)		
+16		上位8ビット: 年(00H~99H)西暦上2桁 下位8ビット: 曜日(00H(日)~06H(土))		
+17	異常検出機器IPアドレス(第3, 4オクテット)	異常を検出した機器のIPアドレス(第3, 4オクテット)が格納されます。 b15 b8 b7 b0 3 4 3, 4: IPアドレスのオクテットを示します。	—	システム
+18	異常検出機器IPアドレス(第1, 2オクテット)	異常を検出した機器のIPアドレス(第1, 2オクテット)が格納されます。 b15 b8 b7 b0 1 2 1, 2: IPアドレスのオクテットを示します。	—	システム

- *1 (s1)+0の到達確認なしに設定した場合、受信データは設定されません。(s1)+0の到達確認なしは、下記の場合に設定してください。
- ・応答伝文が返信されないコマンドを使用する場合
 - ・応答伝文を参照しない場合
- *2 オペランド(U)が3E0Hを指定した場合、1~9を指定できます。
- *3 オペランド(U)が3E0Hを指定した場合、0000Hに固定されます。
- *4 同一のSLMP対応機器に複数の要求伝文を送信する場合に付加します。付加するシリアル番号は、システムで自動的に決定されます。シリアル番号については、下記を参照してください。
[📖 SLMPリファレンスマニュアル](#)
- *5 要求伝文の送信処理実行時、相手機器がネットワーク上に存在していないと、要求伝文が送信されない場合があるが、命令完了時の再送回数はカウントされます。この時、送信された要求伝文の個数と、命令完了時の再送回数は一致しません。

■要求フレーム

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+0	要求データ長	監視タイマから要求データまでのデータ長を指定します。(バイト単位)	1~2000	ユーザ
+1	監視タイマ	要求伝文を受信した相手機器が、アクセス先へ処理を要求してから応答が返るまでの待ち時間を設定するタイマです。(単位: 250ms) ・0: 無限待ち ・1~65535: 1~65535×250ms	0~65535	ユーザ
+2~+0	要求データ	SLMPの伝文の要求データを格納します。	—	ユーザ

■応答フレーム

オペランド: (d1)				
デバイス	項目	内容	設定範囲	セット側
+0	応答データ長	終了コードから応答データまでのデータ長が格納されます。(バイト単位)	2~2000	システム
+1	終了コード	コマンド処理結果が格納されます。 正常終了時は0が格納されます。異常終了時は相手機器でセットされたエラーコードが格納されます。	—	システム
+2~+0	応答データ*1	要求データに対する実行結果がセットされます。 要求データの詳細については、下記マニュアルを参照してください。 📖 SLMPリファレンスマニュアル	—	システム

- *1 応答データは、バイト単位で下位バイト側から順に格納されます。奇数バイト分の応答データを受信したときは、最終データの格納エリアの下位バイトに最後の応答データが格納されます。

機能

- (s2)で指定したデバイス以降の要求フレームを、コントロールデータの相手機器IPアドレスで指定した相手機器へ送信します。相手機器より応答伝文を受信すると(d1)で指定したデバイスに格納されます。
- 本命令は、UDPで通信します。相手機器の設定もUDPに設定してください。
- 本命令は、バイナリコードで交信します。相手機器の設定も、バイナリコードに合わせてください。
- 本命令実行および正常/異常完了は、設定データで指定した完了デバイス(d2)+0, 完了時の状態表示デバイス(d2)+1で確認できます。
- 完了デバイス(d2)+0
本命令が完了したスキャンのEND処理でONし、次のEND処理でOFFします。
- 完了時の状態表示デバイス(d2)+1
本命令が完了したときの状態により、ON/OFFします。
- 正常完了時: OFFのまま変化しません。
- 異常完了時: 本命令が完了したスキャンのEND処理でONし、次のEND処理でOFFします。
- 奇数バイトの応答データを受信した場合、最後の応答データを格納したデバイスの上位バイトには無効なデータが入ります。

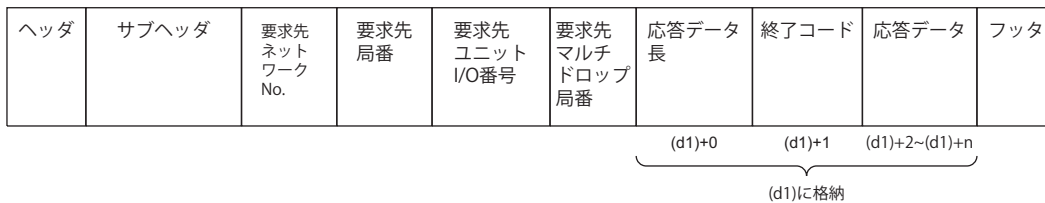
■伝文フォーマット(3E/4Eフレーム)

3E/4Eフレームの要求伝文と正常/異常終了時の応答伝文を示します。

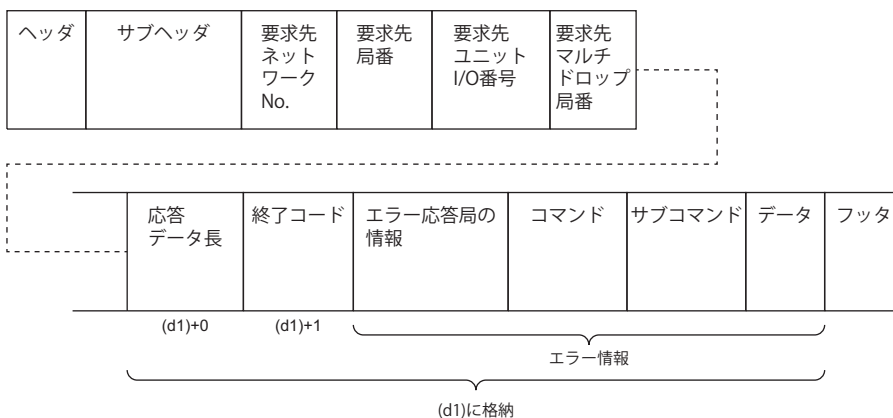
- 要求伝文



- 応答伝文(正常終了時)



- 応答伝文(異常終了時)



エラー応答局の情報には、ネットワークNo., 局番, 要求先ユニットI/O番号, マルチドロップ局番が格納されます。

■伝文フォーマット(局番拡張フレーム)

局番拡張フレームの要求伝文と正常/異常終了時の応答伝文を示します。

・ 要求伝文

ヘッダ	サブヘッダ	要求先 ネット ワーク No.	要求先 局番	要求先 ユニット I/O番号	要求先 マルチ ドロップ 局番	固定値 00	要求先 拡張局番	要求デー タ長	監視タイマ	要求デー タ	フッタ
(s1)+3~(s1)+5		(s1)+6	(s1)+7	(s1)+8	(s1)+9	(s2)+0			(s2)+1	(s2)+2~(s2)+n	
(s1)で指定(一部除く)								(s2)で指定			

・ 応答伝文(正常終了時)

ヘッダ	サブヘッダ	要求先 ネット ワーク No.	要求先 局番	要求先 ユニット I/O番号	要求先 マルチ ドロップ 局番	固定値 00	要求先 拡張局番	応答データ 長	終了コード	応答データ	フッタ
								(d1)+0	(d1)+1	(d1)+2~(d1)+n	
(d1)に格納											

・ 応答伝文(異常終了時)

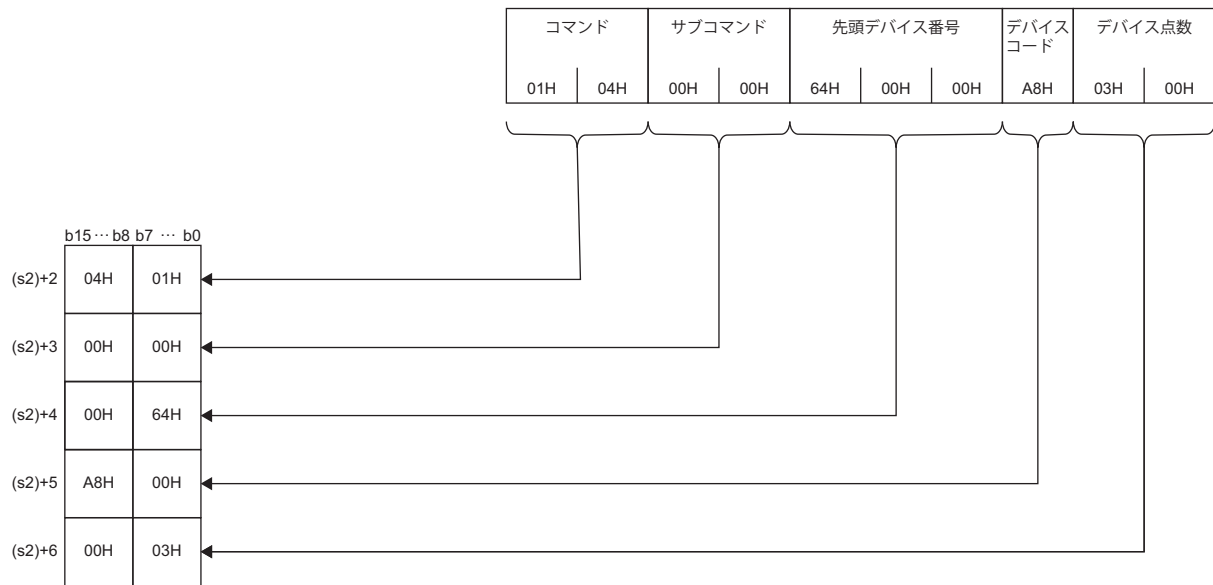
ヘッダ	サブヘッダ	要求先 ネット ワーク No.	要求先 局番	要求先 ユニット I/O番号	要求先 マルチ ドロップ 局番	固定値 00	要求先 拡張局番				
								応答 データ長	終了コード	コマンド	サブコマンド
								(d1)+0	(d1)+1		
エラー情報											
(d1)に格納											

エラー応答局の情報には、ネットワークNo., 局番, 要求先ユニットI/O番号, マルチドロップ局番, 00(固定値), 拡張局番が格納されます。

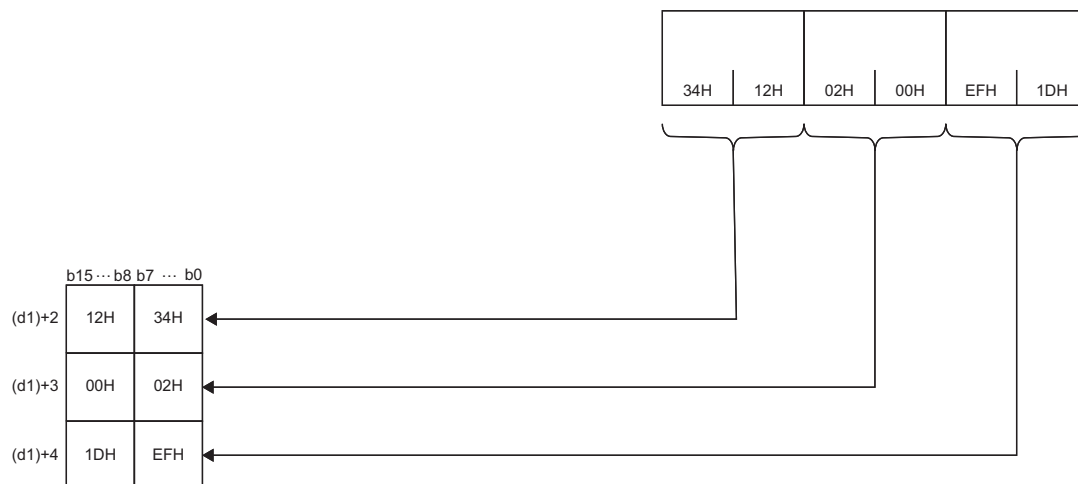
例

D100~D102の値を読み出す「Read(コマンド: 0401H)」(ワード単位読出し)を送信する場合

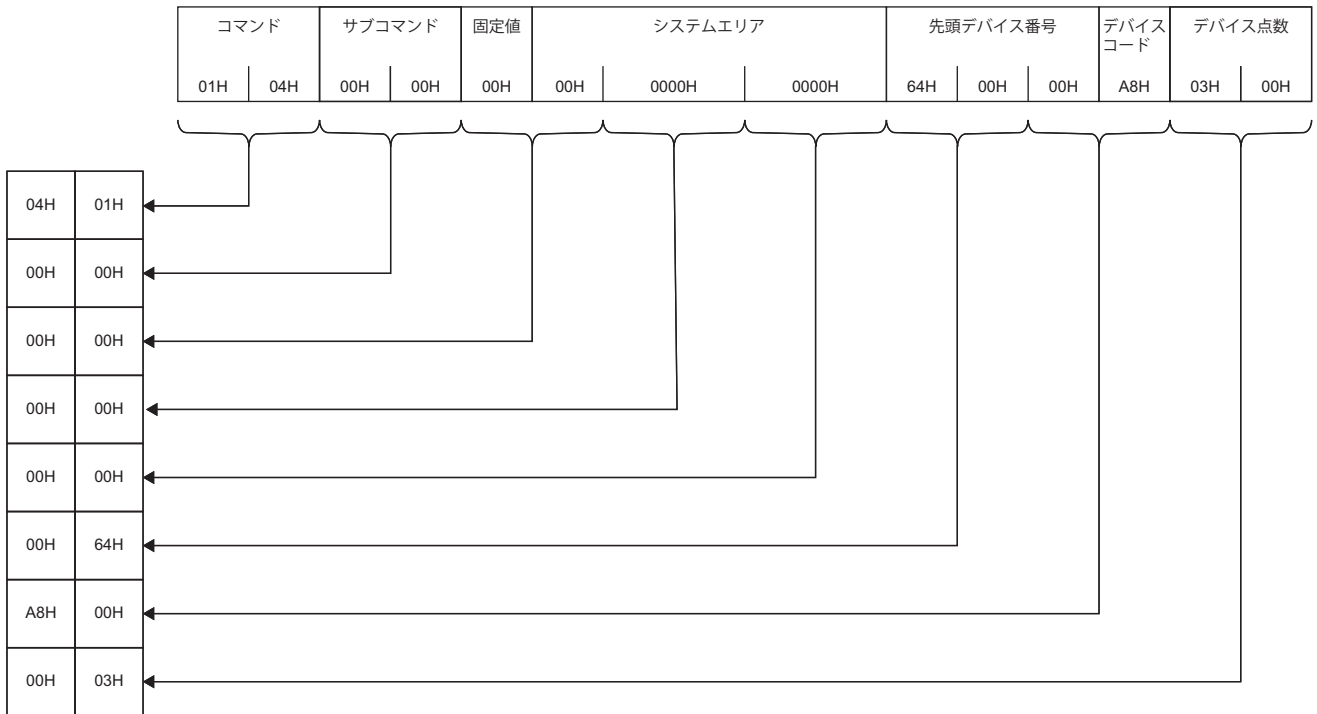
- 要求データを(s2)+2以降に格納する場合(3E/4Eフレーム)



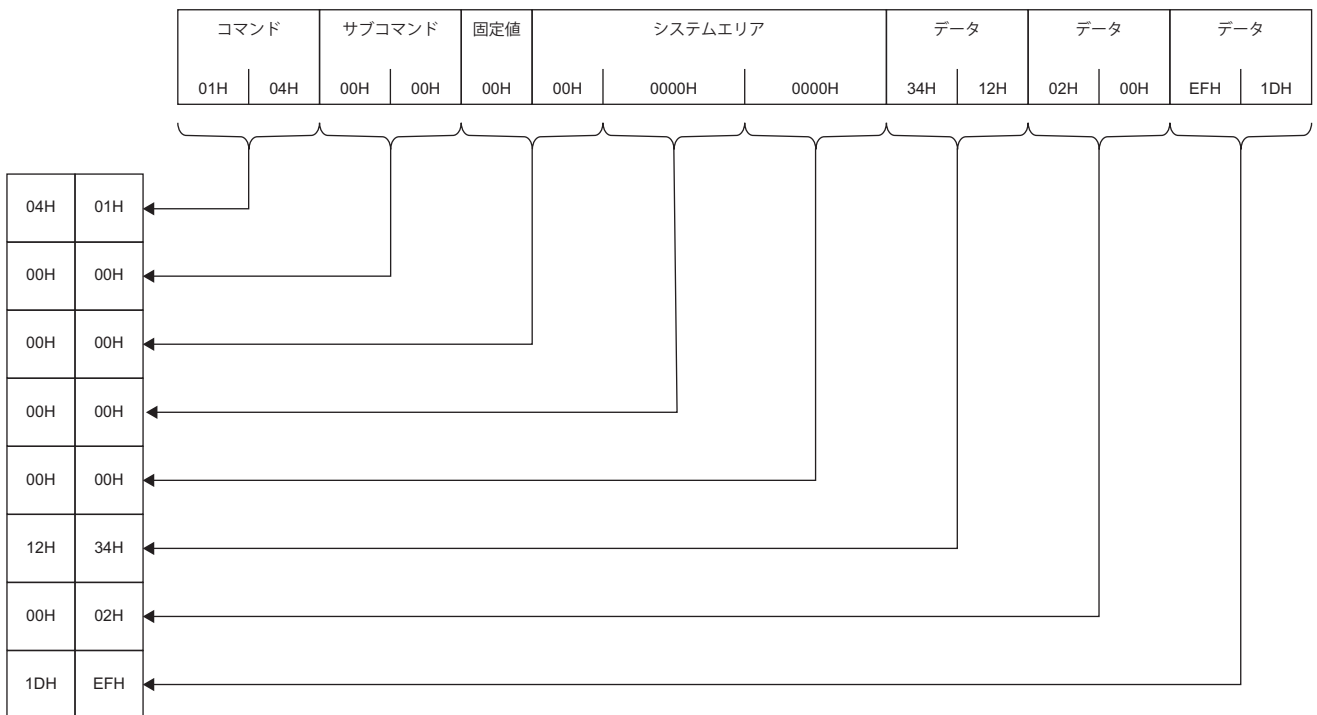
- 応答データを(d1)+2以降に格納する場合(3E/4Eフレーム)



・ 要求データを(s2)+2以降に格納する場合(局番拡張フレーム)

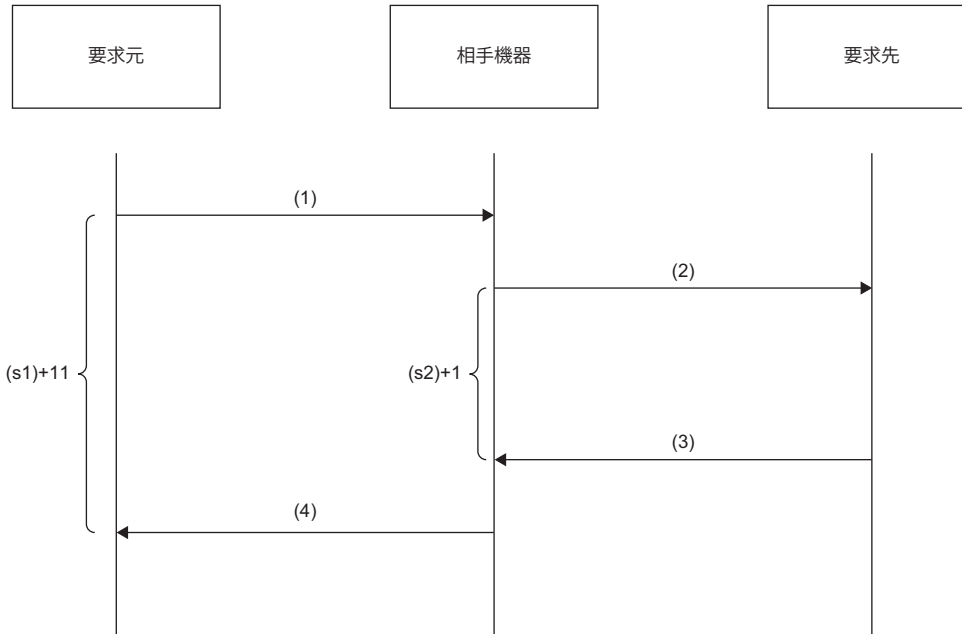


・ 応答データを(d1)+2以降に格納する場合(局番拡張フレーム)



注意事項

- 複数のSLMPSND命令を同時に実行する場合、SLMPSND命令のチャンネルが重複しないように設定してください。
SLMPSND命令を実行中に同一チャンネルを設定したSLMPSND命令を実行した場合、後から実行した命令は無視されます。先に実行している命令が完了した時、後から実行した命令の実行条件が成立している場合は、続けて後から実行した命令の要求が実行されます。また後から実行した命令の要求が無視されている間、SM699(専用命令未実行フラグ)がONします。
- コントロールデータの到達監視時間 $((s1)+11)$ および要求フレームの監視タイマ $((s2)+1)$ は、到達監視時間 $\geq$ 監視タイマとなるよう設定してください。



- (1)要求伝文
(2)相手機器から要求先への処理要求
(3)要求先から相手機器への処理応答
(4)応答伝文

Point

本命令は、対象機器が異常応答を返した場合でも正常完了します。本命令が正常完了した場合、応答フレームの終了コードにより、正常応答か異常応答かを判断します。異常応答の場合は、使用しているSLMP対応機器のマニュアルで確認および処置を行ってください。

エラー

エラーコード (SD0)	内容
3285H	(s1)+2で指定する自局使用チャンネルが範囲外の時。 (s2)+0で指定する要求データ長が0、または2000を超えるとき。

異常完了時は、完了時の状態表示デバイス(d2)+1がONになり、完了ステータス(s1)+1にエラーコードが格納されます。完了ステータス(s1)+1に格納されるエラーコードについては、下記を参照してください。

📖使用するコントローラのユーザズマニュアル

暗号化したSLMPフレーム送信

G(P).SLMPSNDC

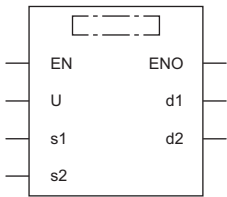
SLMP対応機器に対して暗号化したSLMPの伝文を送信します。

Point

本命令は、Ethernet用ポートでのみ使用できます。

ラダー	ST
	ENO:=G_SLMPSNDC(EN, U, s1, s2, d1, d2) ENO:=GP_SLMPSNDC(EN, U, s1, s2, d1, d2)

FBD/LD



■実行条件

命令	実行条件
G.SLMPSNDC	
GP.SLMPSNDC	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(U)	対象の先頭入出力番号 (先頭入出力番号の上3桁を指定)	3E0H	符号なしBIN16ビット	ANY16
(s1)	コントロールデータを格納する先頭デバイス	1122ページ コントロール データ	デバイス名	ANY16 ^{*1}
(s2)	要求フレームを格納する先頭デバイス	1123ページ 要求フレーム	デバイス名	ANY16 ^{*1}
(d1)	応答フレームを格納する先頭デバイス	1123ページ 応答フレーム	デバイス名	ANY16_ARRAY ^{*2}
(d2)	命令完了にて1スキャンONするデバイス 異常完了時は(d2)+1もONします。	—	ビット	ANYBIT_ARRAY ^{*2} (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義し、その配列型ラベルの要素を指定してください。

*2 ラベルで設定する場合は、動作に必要な領域が確保できるよう配列を定義してください。

■使用可能デバイス

オペランド	ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□ ¥ □	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□ ¥ G□, J□ ¥ □	Z	LT, LST, LC	LZ		K, H	E	\$	
(U)	—	—	○	—	—	—	—	○	○	—	—	○
(s1)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d1)	—	—	○	—	—	—	—	○	—	—	—	—
(d2)	○	—	○*1	—	—	—	—	○	—	—	—	—

*1 T, ST, C, FDは使用できません。

■コントロールデータ

オペランド: (s1)								
デバイス	項目	内容					設定範囲	セット側
+0	実行・異常時完了タイプ	b15 b7 b0 1 0 (2) 0 (1)					―	ユーザ
		(1)実行タイプ(ビット0) ・0: 到達確認なし(自局から要求伝文を送信したことで完了となります。)*1 ・1: 到達確認あり(相手機器から応答伝文を受信したことで完了となります。) (2)異常時完了タイプ(ビット7) 異常完了時のデータのセット状態を指定します。 ・0: (s1)+13以降に異常完了時のデータをセットしない。((s1)+13以降をクリアする。) ・1: (s1)+13以降に異常完了時のデータをセットする。						
+1	完了ステータス	命令完了時の状態が格納されます。 0: 正常 0以外: 異常(エラーコード)					―	システム
+2	自局使用チャンネル	自局が使用するチャンネルを指定します。チャンネルにより要求伝文にシリアル番号 ^{*2} を付加する/付加しないが決まるため、用途に応じてチャンネルを指定してください。 ・1: シリアル番号を付加しないチャンネル ・2~9: シリアル番号を付加するチャンネル					1~9	ユーザ
+3	自局ポート番号	自局のポート番号を指定します。シリアル番号を付加するチャンネルを設定する場合は、自局ポート番号が重複しない様に指定してください。SLMPSND命令と併用する場合、シリアル番号を付加するチャンネルは共有しているので重複しないように設定してください。					1~4999, 5010~65534 (1H~1387H, 1392H~FFFEH)	ユーザ
+4	相手機器IPアドレス(第3, 4オクテット)	b15 b8 b7 b0 3 4 3, 4: IPアドレスのオクテットを示します。					(s1)+3, (s1)+4合わせて、00000001H~FFFFFFFFEH (1~4294967294)	ユーザ
+5	相手機器IPアドレス(第1, 2オクテット)	b15 b8 b7 b0 1 2 1, 2: IPアドレスのオクテットを示します。						ユーザ
+6	相手機器ポート番号	相手機器のポート番号を指定します。					1~65534 (1~FFFEH)	ユーザ
+7	要求先ネットワーク番号	0000H固定					0000H	ユーザ
+8	要求先局番	00FFH固定					00FFH	ユーザ
+9	要求先ユニットI/O番号	要求先のユニットI/O番号を指定します。 ・03FFH: 自局/管理CPU ・03E0H: マルチCPU1号機 ・03E1H: マルチCPU2号機 ・03E2H: マルチCPU3号機 ・03E3H: マルチCPU4号機					03FFH, 03E0H~03E3H	ユーザ
+10	要求先マルチドロップ局番	0000H固定					0000H	ユーザ

オペランド: (s1)				
デバイス	項目	内容	設定範囲	セット側
+11	再送回数 ^{*3*4}	(s1)+0で指定した実行タイプが“1: 到達確認あり”のときに有効になります。 ■命令実行時 (s1)+11で指定した監視時間内に応答伝文の受信が完了しない場合に、要求伝文を再送させる回数を指定します。 ・ 0~15(回) ■命令完了時 要求伝文の再送を行った回数(結果)が格納されます。 ・ 0~15(回)	0~15	ユーザ/システム
+12	到達監視時間	処理完了までの監視時間を指定します。監視時間内に完了しない場合は、(s1)+10で指定した回数まで再送されます。 ・ 0: 10秒 ・ 1~32767: 1~32767秒	0~32767	ユーザ
+13	時計セットフラグ	(s)+13以降のデータ有効/無効状態が格納されます。なお、正常完了した場合、(s1)+13以降のデータはクリアされます。 ・ 0: 無効 ・ 1: 有効	—	システム
+14	時計データ	上位8ビット: 月(01H~12H) 下位8ビット: 年(00H~99H)西暦下2桁	—	システム
+15		上位8ビット: 時(00H~23H) 下位8ビット: 日(01H~31H)		
+16		上位8ビット: 秒(00H~59H) 下位8ビット: 分(00H~59H)		
+17		上位8ビット: 年(00H~99H)西暦上2桁 下位8ビット: 曜日(00H(日)~06H(土))		
+18	異常検出機器IPアドレス(第3, 4オクテット)	異常を検出した機器のIPアドレス(第3, 4オクテット)が格納されます。 b15b8 b7b0 3 4 3, 4: IPアドレスのオクテットを示します。	—	システム
+19	異常検出機器IPアドレス(第1, 2オクテット)	異常を検出した機器のIPアドレス(第1, 2オクテット)が格納されます。 b15b8 b7b0 1 2 1, 2: IPアドレスのオクテットを示します。	—	システム

- *1 (s1)+0の到達確認なしに設定した場合、受信データは設定されません。(s1)+0の到達確認なしは、応答伝文が返信されないコマンドを使用する場合に設定してください。(例えば、応答伝文が返信されないコマンドには、リモートリセットコマンドがあります。)
応答伝文が返信されるコマンドを使用する場合は、必ず到達確認ありに設定してください。到達確認なしに設定すると相手機器でエラーが発生することがあります。
- *2 同一のSLMP対応機器に複数の要求伝文を送信する場合に付加します。付加するシリアル番号は、システムで自動的に決定されます。シリアル番号については、下記を参照してください。
📖 SLMPリファレンスマニュアル
- *3 送信先機器とのDTLSコネクション確立時に送信される、Client Hello等ハンドシェイクの再送回数には適用されません。
- *4 要求伝文の送信処理実行時、相手機器がネットワーク上に存在していないと、要求伝文が送信されない場合があるが、命令完了時の再送回数はカウントされます。この時、送信された要求伝文の個数と、命令完了時の再送回数は一致しません。

■要求フレーム

オペランド: (s2)				
デバイス	項目	内容	設定範囲	セット側
+0	要求データ長	監視タイマから要求データまでのデータ長を指定します。(バイト単位)	1~2000	ユーザ
+1	監視タイマ	要求伝文を受信した相手機器が、アクセス先へ処理を要求してから応答が返るまでの待ち時間を設定するタイマです。(単位: 250ms) ・ 0: 無限待ち ・ 1~65535: 1~65535×250ms	0~65535	ユーザ
+2~+□	要求データ	SLMPの伝文の要求データを格納します。	—	ユーザ

■応答フレーム

オペランド: (d1)				
デバイス	項目	内容	設定範囲	セット側
+0	応答データ長	終了コードから応答データまでのデータ長が格納されます。(バイト単位)	2~2000	システム
+1	終了コード	コマンド処理結果が格納されます。 正常終了時は0が格納されます。異常終了時は相手機器でセットされたエラーコードが格納されます。	—	システム
+2~+□	応答データ ^{*1}	要求データに対する実行結果がセットされます。 (コマンドによっては、応答データを返さないものもあります。)	—	システム

*1 応答データは、バイト単位で下位バイト側から順に格納されます。奇数バイト分の応答データを受信したときは、最終データの格納エリアの下位バイトに最後の応答データが格納されます。

- ・(s2)で指定したデバイス以降の要求フレームを、コントロールデータの相手機器IPアドレスで指定した相手機器へ送信します。相手機器より応答伝文を受信すると(d1)で指定したデバイスに格納されます。
- ・本命令は、DTLSで通信します。相手機器の設定もDTLSに設定してください。
- ・本命令は、バイナリコードで交信します。相手機器の設定も、バイナリコードに合わせてください。
- ・本命令実行および正常/異常完了は、設定データで指定した完了デバイス(d2)+0、完了時の状態表示デバイス(d2)+1で確認できます。

・完了デバイス(d2)+0

本命令が完了したスキャンのEND処理でONし、次のEND処理でOFFします。

・完了時の状態表示デバイス(d2)+1

本命令が完了したときの状態により、ON/OFFします。

正常完了時: OFFのまま変化しません。

異常完了時: 本命令が完了したスキャンのEND処理でONし、次のEND処理でOFFします。

- ・奇数バイトの応答データを受信した場合、最後の応答データを格納したデバイスの上位バイトには無効なデータが入ります。

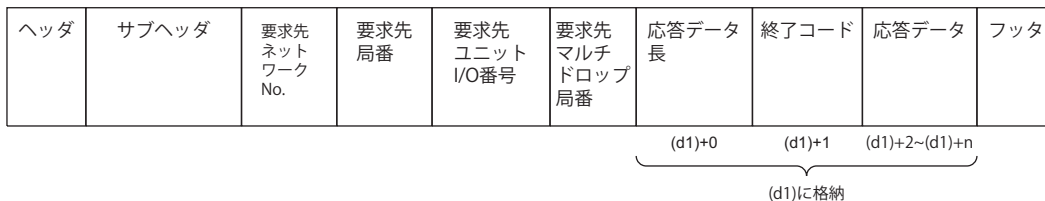
■伝文フォーマット(3E/4Eフレーム)

3E/4Eフレームの要求伝文と正常/異常終了時の応答伝文を示します。

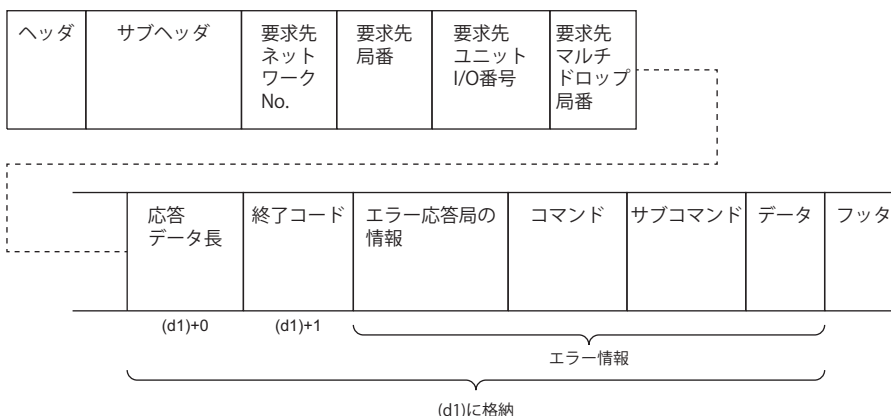
- ・要求伝文



- ・応答伝文(正常終了時)



- ・応答伝文(異常終了時)



エラー応答局の情報には、ネットワークNo., 局番, 要求先ユニットI/O番号, マルチドロップ局番が格納されます。

■伝文フォーマット(局番拡張フレーム)

局番拡張フレームの要求伝文と正常/異常終了時の応答伝文を示します。

• 要求伝文

ヘッダ	サブヘッダ	要求先 ネット ワーク No.	要求先 局番	要求先 ユニット I/O番号	要求先 マルチ ドロップ 局番	固定値 00	要求先 拡張局番	要求デー タ長	監視タイマ	要求デー タ	フッタ
(s1)+3~(s1)+5		(s1)+6	(s1)+7	(s1)+8	(s1)+9	(s2)+0			(s2)+1	(s2)+2~(s2)+n	
(s1)で指定(一部除く)								(s2)で指定			

• 応答伝文(正常終了時)

ヘッダ	サブヘッダ	要求先 ネット ワーク No.	要求先 局番	要求先 ユニット I/O番号	要求先 マルチ ドロップ 局番	固定値 00	要求先 拡張局番	応答データ 長	終了コード	応答データ	フッタ
								(d1)+0	(d1)+1	(d1)+2~(d1)+n	
(d1)に格納											

• 応答伝文(異常終了時)

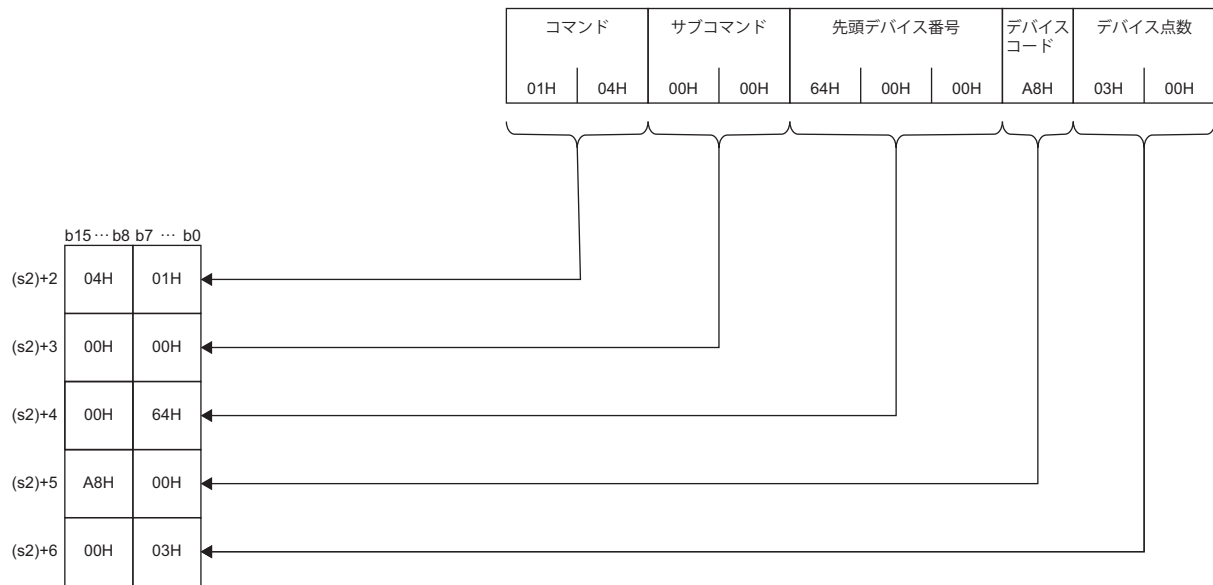
ヘッダ	サブヘッダ	要求先 ネット ワーク No.	要求先 局番	要求先 ユニット I/O番号	要求先 マルチ ドロップ 局番	固定値 00	要求先 拡張局番									
								応答 データ長	終了コード	コマンド	サブコマンド	固定値 00	システムエリア	エラー応答局の 情報	データ	フッタ
		(d1)+0	(d1)+1	エラー情報												
(d1)に格納																

エラー応答局の情報には、ネットワークNo., 局番, 要求先ユニットI/O番号, マルチドロップ局番, 00(固定値), 拡張局番が格納されます。

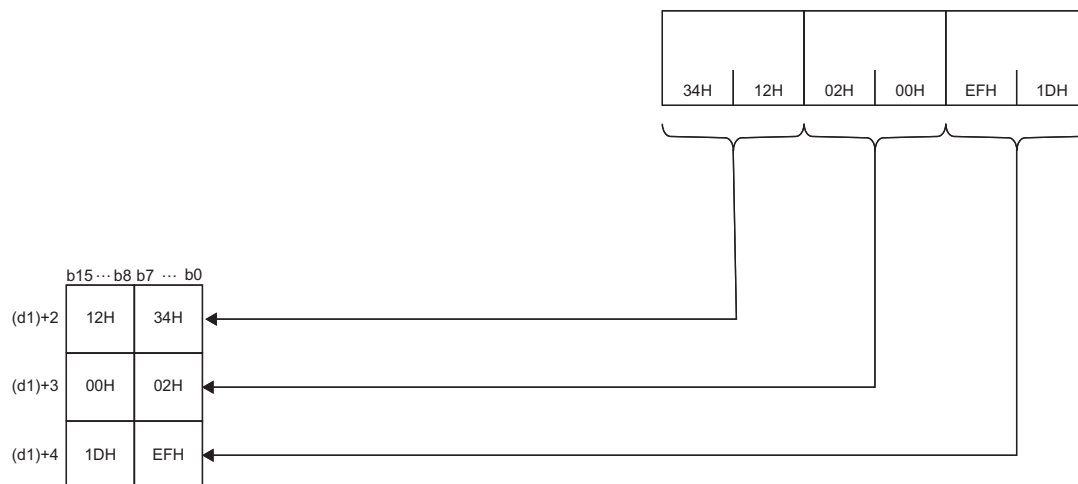
例

D100~D102の値を読み出す「Read(コマンド: 0401H)」(ワード単位読出し)を送信する場合

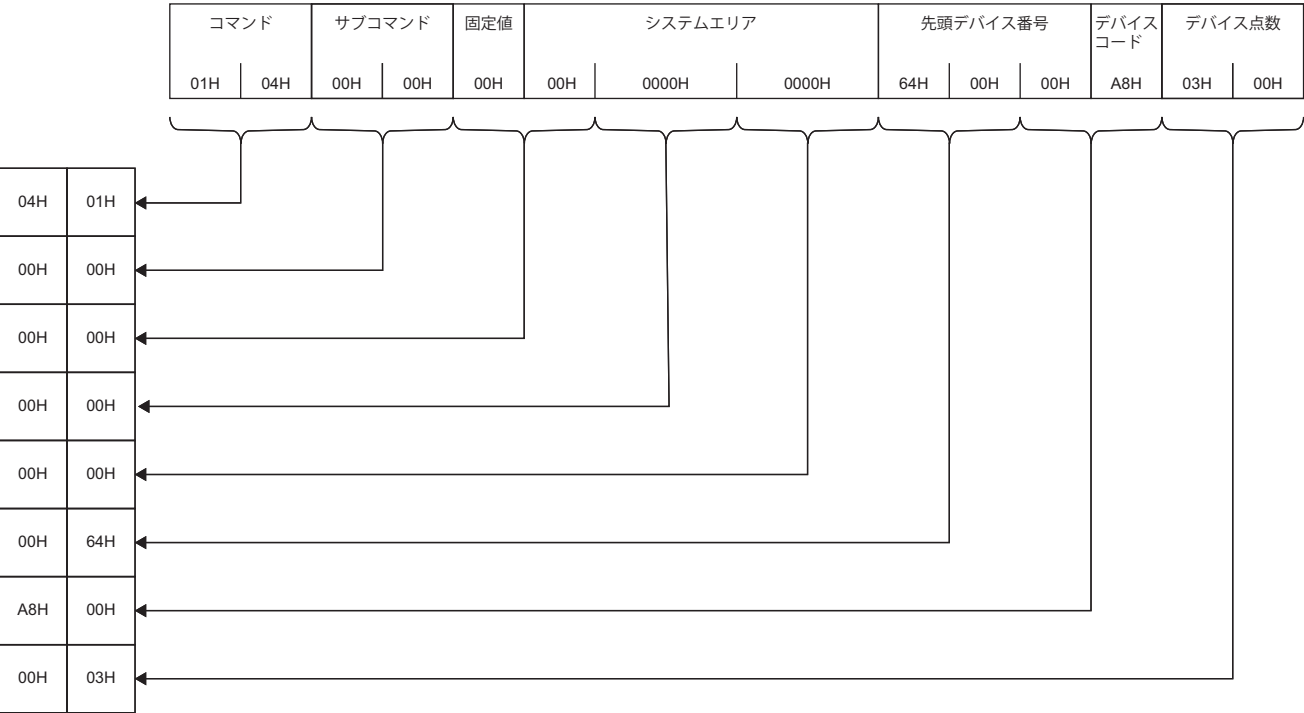
- 要求データを(s2)+2以降に格納する場合(3E/4Eフレーム)



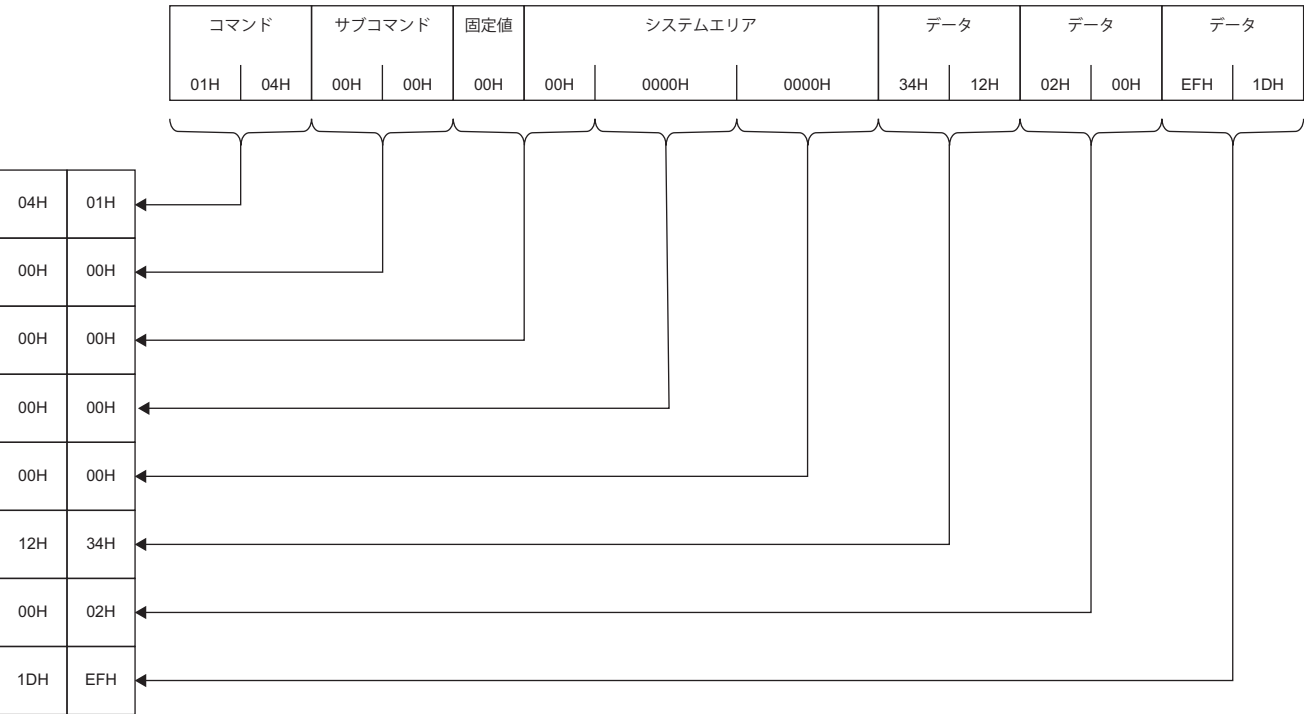
- 応答データを(d1)+2以降に格納する場合(3E/4Eフレーム)



・ 要求データを(s2)+2以降に格納する場合(局番拡張フレーム)

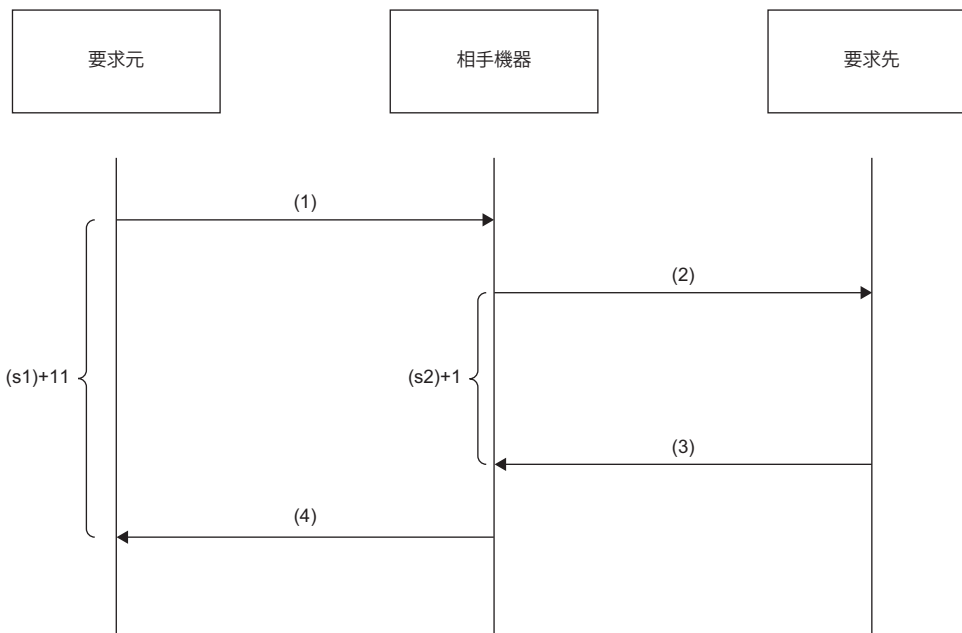


・ 応答データを(d1)+2以降に格納する場合(局番拡張フレーム)



注意事項

- 複数の本命令を同時に実行する場合、本命令のチャンネルが重複しないように設定してください。本命令を実行中に同一チャンネルを設定した本命令を実行した場合、後から実行した本命令は無視されます。先に実行している本命令が完了した時、後から実行した本命令の実行条件が成立している場合は、続けて後から実行した本命令の要求が実行されます。また後から実行した本命令の要求が無視されている間、SM699(専用命令未実行フラグ)がONします。
- コントロールデータの到達監視時間((s1)+11)および要求フレームの監視タイマ((s2)+1)は、到達監視時間>監視タイマとなるよう設定してください。



- (1)要求伝文
 (2)相手機器から要求先への処理要求
 (3)要求先から相手機器への処理応答
 (4)応答伝文

Point

本命令は、対象機器が異常応答を返した場合でも正常完了します。本命令が正常完了した場合、応答フレームの終了コードにより、正常応答か異常応答かを判断します。異常応答の場合は、使用しているSLMP対応機器のマニュアルで確認および処置を行ってください。

エラー

エラーコード (SD0)	内容
1193H	シリアル番号を付加された異なるチャンネルで、同一の自局ポート番号の場合
3285H	(s1)+2で指定する自局使用チャンネルが1~9以外するとき。 (s2)+0で指定する要求データ長が0、または2000を超えるとき。

異常完了時は、完了時の状態表示デバイス(d2)+1がONになり、完了ステータス(s1)+1にエラーコードが格納されます。完了ステータス(s1)+1に格納されるエラーコードについては、下記を参照してください。

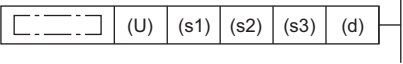
📖使用するコントローラのユーザズマニュアル

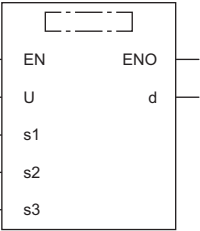
26.5 ファイル転送機能用命令

FTPクライアントファイル送付

GP.FTPCPUT, SP.FTPPUT

(s2)で指定したコントローラのファイルを、(s3)で指定したFTPサーバのフォルダパスへ送付します。

ラダー	ST
	ENO:=GP_FTPCPUT(EN,U,s1,s2,s3,d); ENO:=SP_FTPPUT(EN,U,s1,s2,s3,d);

FBD/LD


■実行条件

命令	実行条件
GP.FTPCPUT SP.FTPPUT	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(U)	GP.FTPCPUT 対象の先頭入出力番号 (先頭入出力番号の上3桁を指定)	3E0H	符号なしBIN16ビット	ANY16
	SP.FTPPUT ダミー	—	文字列	ANYSTRING_SINGLE
(s1)	GP.FTPCPUT コントロールデータを格納する先頭デバイス	コントロールデータ参照	デバイス	ANY16
	SP.FTPPUT		ワード	ANY16_ARRAY (要素数: 4)
(s2)	コントローラに格納しているファイル名(転送元)*1	—	Unicode文字列*2	ANYSTRING_DOUBLE
(s3)	FTPサーバのフォルダパス(転送先)*1	—	Unicode文字列*2	ANYSTRING_DOUBLE
(d)	命令完了にて1スキャンONするデバイス 異常完了時は(d)+1もONします。	—	ビット	ANYBIT_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 Unicode文字列または、Unicode文字列が格納されている先頭デバイスです。

*2 データ型はUnicode文字列ですが、半角英数字、記号、カナ文字、全角文字(シフトJISコード)のみ使用できます。サポートしていない文字は、「_」として扱われます。

■使用可能デバイス

オペランド		ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
		X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(U)	GP.FTPCPUT	—	—	○	—	—	—	—	○	○	—	—	○
	SP.FTPPUT	—	—	○	—	—	—	—	○	—	—	○	○
(s1)		—	—	○	—	—	—	—	○	—	—	—	—
(s2)		—	—	○	—	—	—	—	○	—	—	○	—
(s3)		—	—	○	—	—	—	—	○	—	—	○	—
(d)		○	—	○*1	—	—	—	—	○	—	—	—	—

*1 T, ST, C, FDは使用できません。

■コントロールデータ

オペランド: (s1)				
デバイス	項目	内容	設定範囲	セット側
+0	使用用途設定エリア	b15 ... b3 b2 b1 b0 0 (2) (1) 0 (1)転送完了ファイル削除設定(ビット2) 転送後に転送完了したファイルの扱いを指定します。 • 0: 削除しない • 1: 削除する (2)ファイル転送時の一時ファイル作成設定(ビット3) ファイル転送時の一時ファイル(拡張子を含む転送元ファイル名.TMP)を作成する/しないを指定します。 「0: 作成する」に設定することで、ファイル転送中(ファイル書換え中)にケーブル断または電源断などが発生しても、転送先ファイルが不正なファイル状態とならないようにすることができます。 • 0: 作成する • 1: 作成しない	左記	ユーザ
+1	完了ステータス	命令完了時に完了ステータスが格納されます。 • 0000H: 正常完了 • 0000H以外: 異常完了(エラーコード)	—	システム
+2	処理ファイル総数	本命令で処理するファイルの総数が格納されます。	—	システム
+3	処理完了ファイル数	処理が完了したファイル数が格納されます。	—	システム

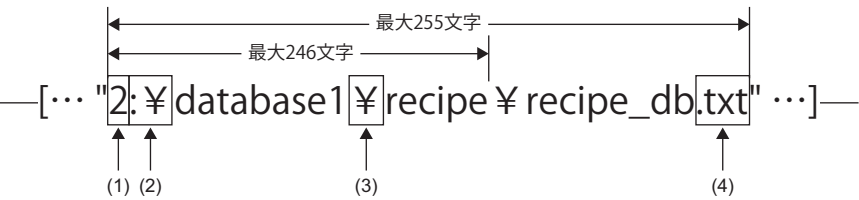
機能

- (s2)で指定したコントローラのファイルを、(s3)で指定したFTPサーバのフォルダパスへ送付します。本命令実行時に、“ユニットパラメータ”の“FTPクライアント設定”で指定したFTPサーバに接続し、ファイル送付後にFTPサーバとの接続を切断します。“FTPクライアント設定”の詳細は、下記を参照してください。

📖使用するコントローラのユーザズマニュアル

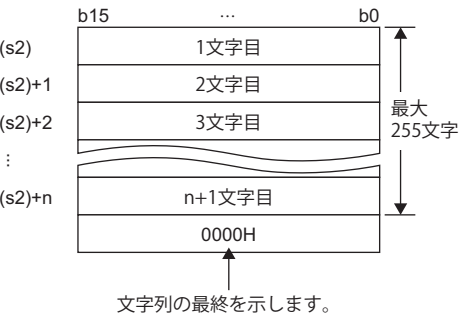
- 本命令で処理するファイルの総数が(s1)+2の「処理ファイル総数」に格納され、処理が完了したファイルの数が(s1)+3の「処理完了ファイル数」に格納されます。ただし、命令実行時にFTPサーバが接続できない状態になった場合、処理完了ファイル数が0になる場合があります。

- (s2)にはコントローラの転送元のドライブNo.2~4, No.6とファイルが格納されているフォルダパス, ファイル名(ピリオド, 拡張子を含む)が64文字以内のファイルをUnicode文字列で指定します。ファイルパスの長さは最大255文字です。ファイル名を除いたパスの長さは, ファイルの区切り文字を含まず最大246文字です。ドライブ, フォルダパス, ファイル名の区切り文字には半角の「¥」または「/」を使用します。^{*1}



- (1) 指定可能なドライブNo.は, 2~4, 6になります。(□使用するコントローラのユーザーズマニュアル)
- (2) ドライブNo.の区切り文字は, 「¥」または「/」を使用します。
- (3) フォルダパス, ファイルの区切り文字は, 半角の「¥」または「/」を使用します。
- (4) 指定するファイル名は拡張子を含めます。

^{*1} FTPクライアントのパスに相対パスは指定できません。



- (s2)のファイル名および拡張子の指定には, ワイルドカード(*, ?)を使用できます。

記号	内容
*	「*」を指定した位置から任意の文字列(なしも含む)のすべてのファイルを対象にします。
?	「?」を指定した位置が任意の文字(なしは含まない)のすべてのファイルを対象にします。(「?」は複数使用できます。)

ピリオドは, 「?」を使用する場合のみ, ワイルドカードの対象になります。

下記の指定方法はエラーとなります。

- ファイル名(ピリオドまで)または拡張子に「*」を2個以上使用する。(例: 「*abc*.txt」)
- ファイル名(ピリオドまで)または拡張子に「*」と「?」が混在している。(例: 「*ab?.txt」)

ワイルドカード指定での転送可能ファイルの数は, 指定された各ファイルのファイル名サイズの合計によって決まります。転送可能ファイルの数, および各ファイル名サイズが下記の式を満たす場合, 転送可能となります。下記の式を満たさない場合にファイル転送機能用命令を実行すると異常完了します。

$$(Fi + NM) + 1 < 65536 \text{ [バイト]}$$

N: ワイルドカード指定に一致するファイルの総数
Fi: ワイルドカード指定に一致するファイル名の合計サイズ
M: 特定情報サイズ(固定値: 6/バイト)

- 本命令実行中に途中の1ファイルの送付でエラーが発生した場合, エラーが発生した時点で転送を中止し, 残りのファイルの転送は実行されません。
- 転送不可ファイルは, ワイルドカード指定の条件に合致していても転送の対象となりません。
- ファイル名, 拡張子を含むファイルパスが255文字を超える場合, ワイルドカード指定の条件に合致していても転送の対象となりません。
- ファイル名(拡張子を含む)が64文字を超える場合, ワイルドカード指定の条件に合致していても転送の対象となりません。

- (s3)には、FTPサーバの転送先のフォルダパスをUnicode文字列で指定します。フォルダパスは、FTPサーバのホームディレクトリからの相対パス、または絶対パスで指定します。フォルダパスの区切り文字には半角の「¥」または「/」を使用します。^{*1}フォルダパスの長さは最大255文字です。ファイル名を除いたパスの長さは、ファイルの区切り文字を含まず最大246文字です。ただし、フォルダパス(末尾の区切り文字を含む)と、(s2)のファイル名部分との合計は、転送先FTPサーバがサポートする最大パス長を超えないように指定してください。末尾の区切り文字は省略可能ですが、省略された場合、末尾には「¥」が設定されていると見なします。なお、存在しないフォルダパスの場合、命令実行時にシステムで自動作成し、ファイル転送を行います。^{*2}

-[... "¥ database1¥ recipe¥" ...]-

 (1) (2)

(1) フォルダパスの区切り文字は、半角の「¥」または「/」を使用します。^{*3}

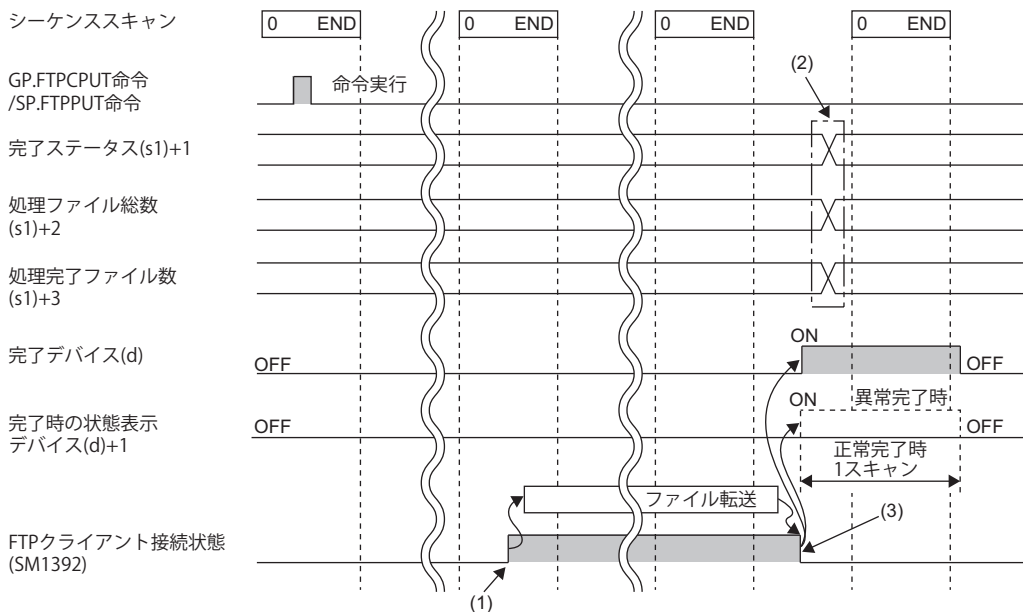
(2) 末尾の区切り文字は省略可能です。

^{*2} FTPサーバのドライブは指定できません。

^{*3} FTPサーバによっては、「¥」を区切り文字として使用できない場合があります。

- (s3)にNULLまたはデバイスに「0000H」のみを指定した場合は、FTPサーバのホームディレクトリ直下にアクセスします。詳細はFTPサーバの仕様に従ってください。
 - 転送先に同名のファイルが存在する場合は、同名のファイルが上書きされます。
 - 送付可能なファイルサイズは、1ファイルにつき最大4294967295バイト(4Gバイト-1バイト)です。
 - 本命令実行および正常/異常完了は、設定データで指定した完了デバイス(d)、完了時の状態表示デバイス(d)+1で確認できます。
 - 完了デバイス(d)
- 本命令が完了したスキャンのEND処理でONし、次のEND処理でOFFします。
- 完了時の状態表示デバイス(d)+1
- 本命令の完了したときの状態により、ON/OFFします。
- 正常完了時: OFFのまま変化しません。
- 異常完了時: 本命令が完了したスキャンのEND処理でONし、次のEND処理でOFFします。異常完了時には、(s1)+1の完了ステータスにエラーコードを格納します。

- 本命令の実行タイミングは下記となります。



- (1) FTPサーバ接続後のEND処理でONします。
 (2) 命令完了時に値を格納します。
 (3) 全ファイルの転送が完了したらOFFします。

- FTPサーバ接続中はSM1392(FTPクライアント接続状態)がONし、切断するとSM1392がOFFします。
- 本命令実行中はSM753(ファイルアクセス中)がONします。SM753がON中は、本命令は実行できません。実行した場合は無処理となります。
- 本命令、またはGP.FTPCGET命令/SP.FTPGET命令を実行中に本命令を実行した場合、後から実行した命令は無視されます。先に実行している命令が完了した時、後から実行している命令の実行条件が成立している場合は、続けて後から実行した命令の要求が実行されます。また後から実行した命令が無視されている間、SM699(専用命令未実行フラグ)がONします。
- (s1)+0のビット3(ファイル転送時の一時ファイル作成設定)の設定に関わらず、転送先に空き容量がない場合は異常完了します。
- ファイル転送を実行中にコントローラの状態をRUN→STOPにしても、完了するまでファイル転送を継続します。

注意事項

- ファイル転送中にケーブル断やコントローラの電源断・リセットが発生した場合、必要に応じてFTPサーバ内に残った不要なファイル(一時ファイルや不完全なファイル)は削除し、再度実行してください。また、ファイル転送中に電源断が発生した場合は、書込み途中のデータが消失することがあるため注意してください。
- 転送完了後の転送元ファイル削除について、(s1)+0のビット2(転送完了ファイル削除設定)を1(削除する)に設定する場合、下記に注意してください。

項目	内容
コントローラのファイルを転送する場合	コントローラの動作に必要なファイルであってもコントローラからファイルが削除されます。削除した場合の動作は保証しません。
転送ファイル名をワイルドカード指定して転送する場合	意図せずに必要なファイルが削除されることがあります。

- ファイルパスワード機能で転送元のファイルがアクセス制限されている場合、本命令が異常完了します。ただし、(s1)+0のビット2(転送完了ファイル削除設定)を0(削除しない)に設定し、ファイルパスワードの設定が「書込み禁止」の場合は本命令を実行できます。
- (s1)+0のビット3(ファイル転送時の一時ファイル作成設定)を0(作成する)に設定する場合、転送先に最大68文字の一時ファイル(ピリオド、拡張子を含む転送元ファイル名.TMP)が作成されるため、転送先フォルダパスと一時ファイルを合わせたパス長が、255文字を超えないようにしてください。ファイル名は、最大64文字を超えないようにしてください。
- サポートされていない文字を使用しないでください。(s2)、(s3)に指定したファイル名またはフォルダ名にサポートされていない文字が含まれる場合、その文字は「_」に変換されて処理されます。ワイルドカード指定時、転送元コントローラに存在するファイル名にサポートされていない文字が含まれる場合、その文字は「_」に変換されて処理されます。そのため、ファイル転送においては、「_」に変換された後のファイル名またはフォルダ名が指定された場合と同等の扱いとなります。

エラー

エラーコード (SD0)	内容
2820H	(s2)、(s3)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に0000Hが存在しないとき。
3285H	(s2)で指定したデバイスに格納されている文字列が255文字を超えるとき。 (s2)または(s3)で指定したファイル名を除いたパス(ファイルの区切り文字は含まない)が246文字を超えるとき。 (s2)で指定したファイル名(ドライブ、フォルダパスを除く)と(s3)で指定した文字列の合計が255文字を超えるとき。 ^{*1*2} (s2)で指定したドライブNo.が転送可能範囲外のとき。 (s2)で指定したファイル名が64文字を超えるとき。 ^{*2}
3296H	(s2)でファイル名が指定されていないとき。 (s2)で指定したファイル名が転送できないファイルのとき。 (s2)のドライブNo.の区切り文字が「:¥」または「:/」以外のとき。 (s2)で指定したファイル名(ピリオドまで)または拡張子に「*」を2個以上指定したとき。 (s2)で指定したファイル名(ピリオドまで)または拡張子に「*」と「?」が混在したとき。 (s3)にワイルドカード指定文字(「*」、「?」)が含まれているとき。
329BH	FTPクライアント設定のパラメータが設定されていない状態で本命令を実行したとき。

*1 (s3)でフォルダパス末尾の区切り文字を省略した場合は最大254文字です。

*2 ファイル転送時の一時ファイル作成設定を「0:作成する」に設定した場合、ファイル名には、拡張子「.TMP」の4文字を含みます。異常完了時は、完了時の状態表示デバイス(d)+1がONになり、完了ステータス(s1)+1にエラーコードが格納されます。

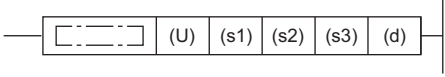
完了ステータス(s1)+1に格納されるエラーコードについては、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

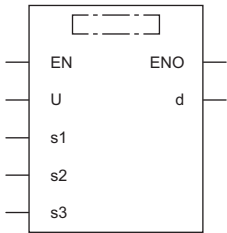
FTPクライアントファイル取得

GP.FTPCGET, SP.FTPGET

(s2)で指定したFTPサーバのファイルを、(s3)で指定したコントローラのフォルダパスに取得します。

ラダー	ST
	ENO:=GP_FTPCGET(EN,U,s1,s2,s3,d); ENO:=SP_FTPGET(EN,U,s1,s2,s3,d);

FBD/LD



■実行条件

命令	実行条件
GP.FTPCGET SP.FTPGET	

設定データ

■内容、範囲、データ型

オペランド	内容	範囲	データ型	データ型(ラベル)
(U)	GP.FTPCGET 対象の先頭入出力番号 (先頭入出力番号の上3桁を指定)	3E0H	符号なしBIN16ビット	ANY16
	SP.FTPGET ダミー	—	文字列	ANYSTRING_SINGLE
(s1)	GP.FTPCGET コントロールデータを格納する先頭デバイス	コントロールデータ参照	デバイス	ANY16
	SP.FTPGET		ワード	ANY16_ARRAY (要素数: 4)
(s2)	FTPサーバに格納しているファイル名(転送元)*1	—	Unicode文字列*2	ANYSTRING_DOUBLE
(s3)	コントローラのフォルダパス(転送先)*1	—	Unicode文字列*2	ANYSTRING_DOUBLE
(d)	命令完了にて1スキャンONするデバイス 異常完了時は(d)+1もONします。	—	ビット	ANYBIT_ARRAY (要素数: 2)
EN	実行条件	—	ビット	BOOL
ENO	実行結果	—	ビット	BOOL

*1 Unicode文字列または、Unicode文字列が格納されている先頭デバイスです。

*2 データ型はUnicode文字列ですが、半角英数字、記号、カナ文字、全角文字(シフトJISコード)のみ使用できます。サポートしていない文字は、「」として扱われます。

■使用可能デバイス

オペランド		ビット		ワード			ダブルワード		間接指定	定数			その他 (U)
		X, Y, M, L, SM, F, B, SB, FX, FY	J口 ¥ 口	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U口 ¥ G口, J口 ¥ 口	Z	LT, LST, LC	LZ		K, H	E	\$	
(U)	GP.FTPCGET	—	—	○	—	—	—	—	○	○	—	—	○
	SP.FTPGET	—	—	○	—	—	—	—	○	—	—	○	○
(s1)		—	—	○	—	—	—	—	○	—	—	—	—
(s2)		—	—	○	—	—	—	—	○	—	—	○	—
(s3)		—	—	○	—	—	—	—	○	—	—	○	—
(d)		○	—	○*1	—	—	—	—	○	—	—	—	—

*1 T, ST, C, FDは使用できません。

■コントロールデータ

オペランド: (s1)				
デバイス	項目	内容	設定範囲	セット側
+0	使用用途設定エリア	b15 ... b3 b2 b1 b0 0 (2) (1) 0 (1)転送完了ファイル削除設定(ビット2) 転送後に転送完了したファイルの扱いを指定します。 • 0: 削除しない • 1: 削除する (2)ファイル転送時の一時ファイル作成設定(ビット3) ファイル転送時の一時ファイル(拡張子を含む転送元ファイル名.TMP)を作成する/しないを指定します。 「0: 作成する」に設定することで、ファイル転送中(ファイル書換え中)にケーブル断または電源断などが発生しても、転送先ファイルが不正なファイル状態とならないようにすることができます。 • 0: 作成する • 1: 作成しない	左記	ユーザ
+1	完了ステータス	命令完了時に完了ステータスが格納されます。 • 0000H: 正常完了 • 0000H以外: 異常完了(エラーコード)	—	システム
+2	処理ファイル総数	本命令で処理するファイルの総数が格納されます。	—	システム
+3	処理完了ファイル数	処理が完了したファイル数が格納されます。	—	システム

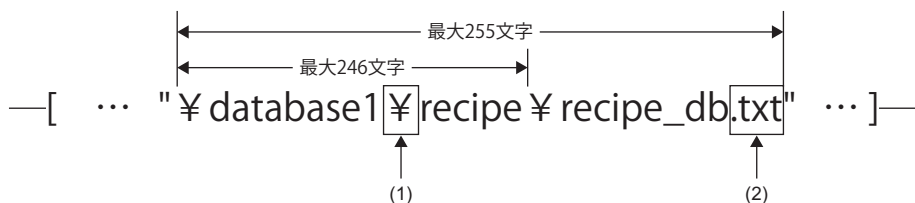
機能

- (s2)で指定したFTPサーバのファイルを、(s3)で指定したコントローラのフォルダパスに取得します。本命令実行時に、“ユニットパラメータ”の“FTPクライアント設定”で指定したFTPサーバに接続し、ファイル取得後にFTPサーバとの接続を切断します。“FTPクライアント設定”の詳細は、下記を参照してください。

使用するコントローラのユーザーズマニュアル

- 本命令で処理するファイルの総数が(s1)+2の「処理ファイル総数」に格納され、処理が完了したファイルの数が(s1)+3の「処理完了ファイル数」に格納されます。ただし、命令実行時に転送先にSDメモリカードを設定した状態で、ファイル転送中にSDメモリカードを抜いた場合、処理完了ファイル数が0になる場合があります。
- (s2)にはFTPサーバの転送元のファイルが格納されているフォルダパス、ファイル名(ピリオド、拡張子を含む)が64文字以内のファイルをUnicode文字列で指定します。ファイルパスの長さは最大255文字です。ファイル名を除いたパスの長さは、ファイルの区切り文字を含まず最大246文字です。ファイルパスは、FTPサーバのホームディレクトリからの相対パスまたは絶対パスで指定します。フォルダパス、ファイル名の区切り文字には半角の「¥」または「/」を使用します。

*1*2

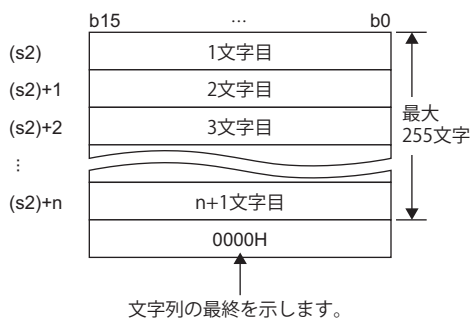


(1) フォルダパス、ファイルの区切り文字は、半角の「¥」または「/」を使用します。^{*2}

(2) 指定するファイル名は拡張子を含めます。

*1 FTPサーバのドライブは指定できません。

*2 FTPサーバによっては、「¥」を区切り文字として使用できない場合があります。



- (s2)のファイル名および拡張子の指定には、ワイルドカード(*, ?)を使用できます。

記号	内容
*	「*」を指定した位置から任意の文字列(なしも含む)のすべてのファイルを対象にします。
?	「?」を指定した位置が任意の文字(なしは含まない)のすべてのファイルを対象にします。(「?」は複数使用できます。)

ピリオドは、「?」を使用する場合のみ、ワイルドカードの対象になります。

下記の指定方法はエラーとなります。

- ファイル名(ピリオドまで)または拡張子に「*」を2個以上使用する。(例: 「*abc*.txt」)
- ファイル名(ピリオドまで)または拡張子に「*」と「?」が混在している。(例: 「*ab?.txt」)

サブフォルダ内のファイル転送はできません。

ワイルドカード指定した取得対象ファイルはFTPサーバの環境によって異なります。FTPサーバの動作仕様を確認してください。

ワイルドカード指定での転送可能ファイルの数は、指定された各ファイルのファイル名サイズの合計によって決まります。転送可能ファイルの数、および各ファイル名サイズが下記の式を満たす場合、転送可能となります。下記の式を満たさない場合にファイル転送機能用命令を実行すると異常完了します。

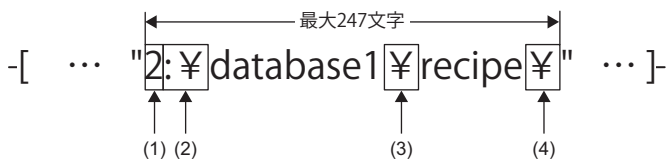
$(Fi + NM) + 1 < 65536$ [バイト]

N: ワイルドカード指定に一致するファイルの総数

Fi: ワイルドカード指定に一致するファイル名の合計サイズ

M: 特定情報サイズ(固定値: 6バイト)

- 本命令実行中に途中の1ファイルの取得でエラーが発生した場合、エラーが発生した時点で転送を中止し、残りのファイルの転送は実行されません。
- 転送不可ファイルは、ワイルドカード指定の条件に合致していても転送の対象となりません。
- ファイル名、拡張子を含むファイルパスが255文字を超える場合、ワイルドカード指定の条件に合致していても転送の対象となりません。
- ファイル名(拡張子を含む)が64文字を超える場合、ワイルドカード指定の条件に合致していても転送の対象となりません。
- (s2)にファイル名のみを指定した場合は、FTPサーバのホームディレクトリ直下にアクセスします。ファイル名のみ指定する場合は、区切り文字から指定してください。詳細はFTPサーバの仕様に従ってください。
- (s3)には、コントローラの転送先のフォルダパスをUnicode文字列で指定します。フォルダパスの区切り文字には半角の「¥」または「/」を使用します。フォルダパスの長さは最大247文字(末尾の区切り文字が省略された場合は、最大246文字)です。ただし、フォルダパス(末尾の区切り文字を含む)と、(s2)のファイル名部分との合計は、コントローラがサポートする最大パス長(255文字)を超えないように指定してください。末尾の区切り文字は省略可能ですが、省略された場合、末尾には「¥」が設定されていると見なします。なお、存在しないフォルダパスの場合、命令実行時にシステムで自動作成し、ファイル転送を行います。^{*3}



- (1) 指定可能なドライブNo.は、2, 4になります。(□□使用するコントローラのユーザーズマニュアル)
- (2) ドライブNo.の区切り文字は、「:¥」または「:/」を使用します。
- (3) フォルダパスの区切り文字は、半角の「¥」または「/」を使用します。
- (4) 末尾の区切り文字は省略可能です。

^{*3} FTPクライアントのパスに相対パスは指定できません。

- (s3)にNULLまたはデバイスに「0000H」のみを指定した場合は、FTPサーバのホームディレクトリ直下(4:¥)にアクセスします。
- 転送先に同名のファイルが存在する場合は、同名のファイルが上書きされます。
- 取得可能なファイルサイズは、1ファイルにつき最大4294967295バイト(4Gバイト-1バイト)です。
- 本命令実行および正常/異常完了は、設定データで指定した完了デバイス(d)、完了時の状態表示デバイス(d)+1で確認できます。

• 完了デバイス(d)

本命令が完了したスキャンのEND処理でONし、次のEND処理でOFFします。

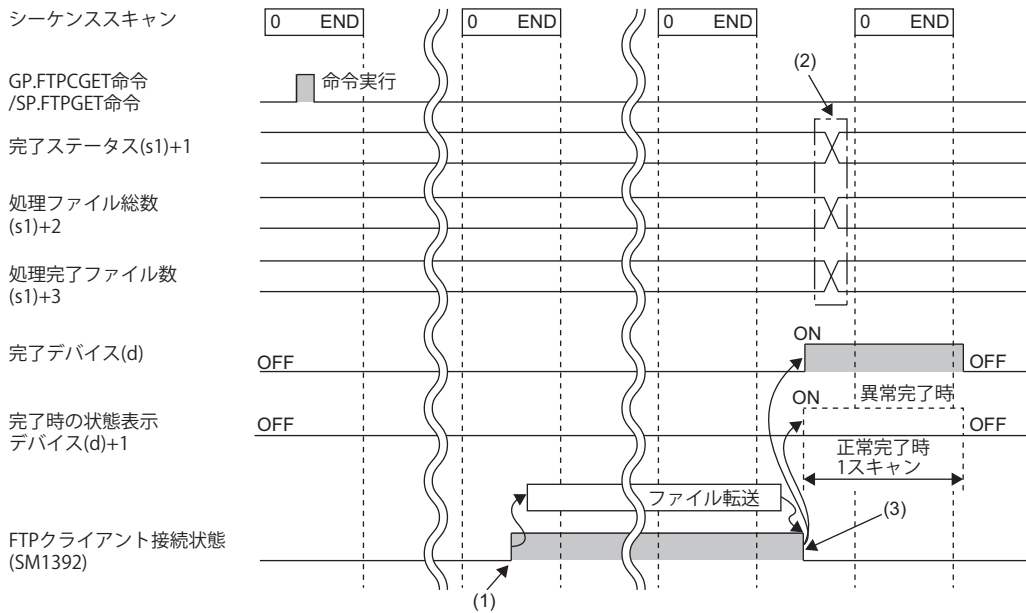
• 完了時の状態表示デバイス(d)+1

本命令の完了したときの状態により、ON/OFFします。

正常完了時: OFFのまま変化しません。

異常完了時: 本命令が完了したスキャンのEND処理でONし、次のEND処理でOFFします。異常完了時には、(s1)+1の完了ステータスにエラーコードを格納します。

- 本命令の実行タイミングは下記となります。



- (1) FTPサーバ接続後のEND処理でONします。
 (2) 命令完了時に値を格納します。
 (3) 全ファイルの転送が完了したらOFFします。

- FTPサーバ接続中はSM1392(FTPクライアント接続状態)がONし、切断するとSM1392がOFFします。
- 本命令実行中はSM753(ファイルアクセス中)がONします。SM753がON中は、本命令は実行できません。実行した場合は無処理となります。
- GP.FTPCPUT命令/SP.FTPPUT命令、または本命令を実行中に本命令を実行した場合、後から実行した命令は無視されます。先に実行している命令が完了した時、後から実行している命令の実行条件が成立している場合は、続けて後から実行した命令の要求が実行されます。また後から実行した命令が無視されている間、SM699(専用命令未実行フラグ)がONします。
- (s1)+0のビット3(ファイル転送時の一時ファイル作成設定)の設定に関わらず、転送先に空き容量がない場合は異常完了します。
- ファイル転送を実行中にコントローラの状態をRUN→STOPにしても、完了するまでファイル転送を継続します。

注意事項

- ファイル転送中にケーブル断やコントローラの電源断・リセットが発生した場合、再度実行してください。ケーブル断や電源断・リセットが発生した場合、不要ファイルは下記のように削除されます。また、ファイル転送中に電源断が発生した場合は、書き込み途中のデータが消失することがあるため注意してください。

不要ファイル	削除動作
一時ファイル(拡張子を含む転送元ファイル名.TMP)	FTPクライアント内に残った一時ファイルは、次の同一フォルダに対するファイル転送開始時に、自動削除されます。
不完全な転送対象ファイル	FTPクライアント内に残った不完全な転送対象ファイルは、次の同一ファイルに対するファイル転送時に上書きされます。

- 転送元のファイルに対し、転送先にファイルパスワード機能でアクセス制限されている同名ファイルが存在する場合、本命令が異常完了します。
- (s1)+0のビット3(ファイル転送時の一時ファイル作成設定)を0(作成する)に設定する場合、コントローラのフォルダパスの長さは最大243文字(末尾の区切り文字が省略された場合、最大242文字)にしてください。また、転送先に最大68文字の一時ファイル(ピリオド、拡張子を含む転送元ファイル名.TMP)が作成されるため、転送先フォルダパスと合わせたパス長が、コントローラがサポートする最大パス長(255文字)を超えないようにしてください。ファイル名は、最大64文字を超えないようにしてください。
- サポートされていない文字を使用しないでください。(s2)、(s3)に指定したファイル名またはフォルダ名にサポートされていない文字が含まれる場合、その文字は「_」に変換されて処理されます。そのため、ファイル転送においては、「_」に変換された後のファイル名またはフォルダ名が指定された場合と同等の扱いとなります。また、ワイルドカード指定時、転送元FTPサーバに存在するファイル名にサポートされていない文字が含まれる場合、正しくファイル取得できません。(サポートされていない文字をどのように扱うのかはFTPサーバの仕様に従います)

エラー

エラーコード(SD0)	内容
2820H	(s2)、(s3)で指定したデバイス番号以降、該当デバイス/ラベルの割付範囲に0000Hが存在しないとき。
3285H	(s2)で指定したデバイスに格納されている文字列が255文字を超えるとき。 (s2)または(s3)で指定したファイル名を除いたパス(ファイルの区切り文字は含まない)が246文字を超えるとき。 (s3)で指定した文字列と(s2)で指定した文字列のファイル名部分との合計が255文字を超えるとき。 ^{*1*2} (s3)で指定したドライブNo.が転送可能範囲外のとき。 (s2)で指定したファイル名が64文字を超えるとき。 ^{*2}
3296H	(s2)でファイル名が指定されていないとき。 (s2)で指定したファイル名が転送できないファイルのとき。 (s3)のドライブNo.の区切り文字が「:¥」または「:/」以外のとき。 (s2)で指定したファイル名(ピリオドまで)または拡張子に「*」を2個以上指定したとき。 (s2)で指定したファイル名(ピリオドまで)または拡張子に「*」と「?」が混在したとき。 (s3)にワイルドカード指定文字(「*」、「?」)が含まれているとき。
329BH	FTPクライアント設定のパラメータが設定されていない状態で本命令を実行したとき。

*1 (s2)でフォルダパス末尾の区切り文字を省略した場合は最大254文字です。

*2 ファイル転送時の一時ファイル作成設定を「0:作成する」に設定した場合、ファイル名には、拡張子「.TMP」の4文字を含みます。異常完了時は、完了時の状態表示デバイス(d)+1がONになり、完了ステータス(s1)+1にエラーコードが格納されます。

完了ステータス(s1)+1に格納されるエラーコードについては、下記を参照してください。

📖使用するコントローラのユーザズマニュアル

MEMO

第6部 汎用FUN

この部は下記の章構成となります。

27 型変換ファンクション

28 単数値変数ファンクション

29 算術演算ファンクション

30 ビットシフトファンクション

31 ビット型ブールファンクション

32 選択ファンクション

33 比較ファンクション

34 文字列ファンクション

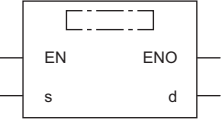
35 時刻データ型ファンクション

27 型変換ファンクション

27.1 BOOL型→WORD型変換

BOOL_TO_WORD(_E)

BOOL型データをWORD型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=BOOL_TO_WORD(s); [EN/ENO付き] d:=BOOL_TO_WORD_E(EN,ENO,s);

設定データ

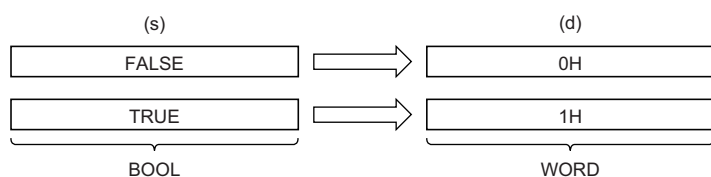
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	BOOL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	WORD

機能

■演算処理

- (s)に入力されたBOOL型のデータを，WORD型のデータに変換して(d)から出力します。
- 入力値がFALSEの場合は，WORD型のデータ値で0Hを出力します。
- 入力値がTRUEの場合は，WORD型のデータ値で1Hを出力します。



- (s)への入力値は，BOOL型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合，(d)から出力されるデータは不定値となります。その場合は，(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

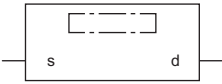
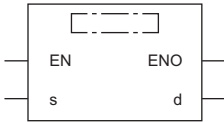
エラー

演算エラーはありません。

27.2 BOOL型→DWORD型変換

BOOL_TO_DWORD(_E)

BOOL型データをDWORD型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=BOOL_TO_DWORD(s); [EN/ENO付き] d:=BOOL_TO_DWORD_E(EN,ENO,s);

設定データ

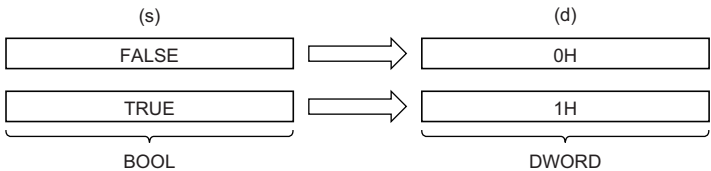
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	BOOL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	DWORD

機能

■演算処理

- (s)に入力されたBOOL型のデータを, DWORD型のデータに変換して(d)から出力します。
- 入力値がFALSEの場合は, DWORD型のデータ値で0Hを出力します。
- 入力値がTRUEの場合は, DWORD型のデータ値で1Hを出力します。



- (s)への入力値は, BOOL型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

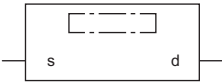
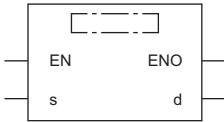
演算エラーはありません。

27.3 BOOL型→INT型変換

27

BOOL_TO_INT(_E)

BOOL型データをINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=BOOL_TO_INT(s); [EN/ENO付き] d:=BOOL_TO_INT_E(EN,ENO,s);

設定データ

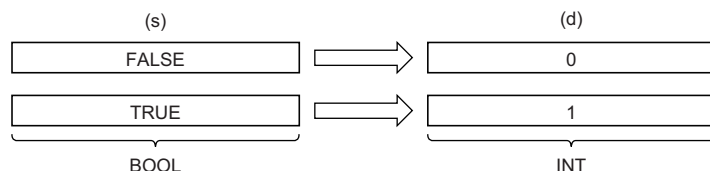
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	BOOL
ENO	出力状態(TRUE: 正常実行, FALSE: 異常または停止)	出力変数	BOOL
d	出力	出力変数	INT

機能

■演算処理

- ・ (s)に入力されたBOOL型のデータを, INT型のデータに変換して(d)から出力します。
- ・ 入力値がFALSEの場合は, INT型のデータ値で0を出力します。
- ・ 入力値がTRUEの場合は, INT型のデータ値で1を出力します。



- ・ (s)への入力値は, BOOL型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

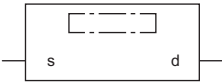
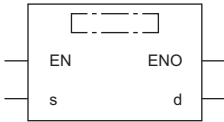
エラー

演算エラーはありません。

27.4 BOOL型→DINT型変換

BOOL_TO_DINT(_E)

BOOL型データをDINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=BOOL_TO_DINT(s); [EN/ENO付き] d:=BOOL_TO_DINT_E(EN,ENO,s);

設定データ

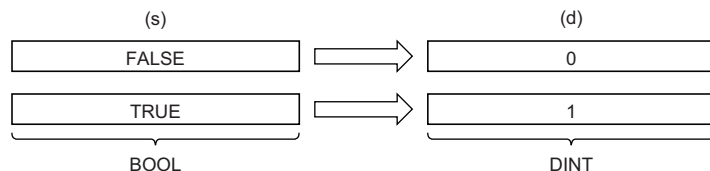
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	BOOL
ENO	出力状態(TRUE: 正常実行, FALSE: 異常または停止)	出力変数	BOOL
d	出力	出力変数	DINT

機能

■演算処理

- ・ (s)に入力されたBOOL型のデータを, DINT型のデータに変換して(d)から出力します。
- ・ 入力値がFALSEの場合は, DINT型のデータ値で0を出力します。
- ・ 入力値がTRUEの場合は, DINT型のデータ値で1を出力します。



- ・ (s)への入力値は, BOOL型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

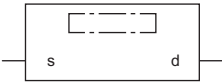
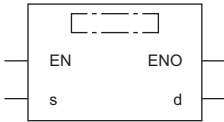
演算エラーはありません。

27.5 BOOL型→TIME型変換

27

BOOL_TO_TIME(_E)

BOOL型データをTIME型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=BOOL_TO_TIME(s); [EN/ENO付き] d:=BOOL_TO_TIME_E(EN,ENO,s);

設定データ

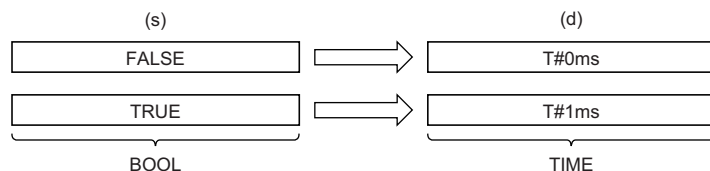
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	BOOL
ENO	出力状態(TRUE: 正常実行, FALSE: 異常または停止)	出力変数	BOOL
d	出力	出力変数	TIME

機能

■演算処理

- ・ (s)に入力されたBOOL型のデータを, TIME型のデータに変換して(d)から出力します。
- ・ 入力値がFALSEの場合は, TIME型のデータ値で0を出力します。
- ・ 入力値がTRUEの場合は, TIME型のデータ値で1を出力します。



- ・ (s)への入力値は, BOOL型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

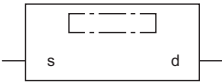
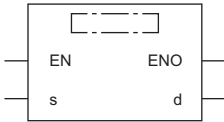
エラー

演算エラーはありません。

27.6 BOOL型→STRING型変換

BOOL_TO_STRING(_E)

BOOL型データをSTRING型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=BOOL_TO_STRING(s); [EN/ENO付き] d:=BOOL_TO_STRING_E(EN,ENO,s);

設定データ

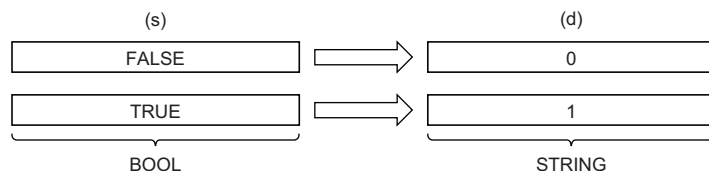
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	BOOL
ENO	出力状態(TRUE: 正常実行, FALSE: 異常または停止)	出力変数	BOOL
d	出力	出力変数	STRING

機能

■演算処理

- ・ (s)に入力されたBOOL型のデータを, STRING型のデータに変換して(d)から出力します。
- ・ 入力値がFALSEの場合は, STRING型のデータ値で0を出力します。
- ・ 入力値がTRUEの場合は, STRING型のデータ値で1を出力します。



- ・ (s)への入力値は, BOOL型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

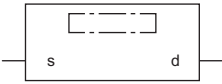
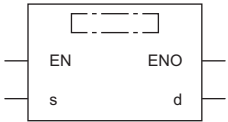
演算エラーはありません。

27.7 WORD型→BOOL型変換

27

WORD_TO_BOOL(_E)

WORD型データをBOOL型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=WORD_TO_BOOL(s); [EN/ENO付き] d:=WORD_TO_BOOL_E(EN,ENO,s);

設定データ

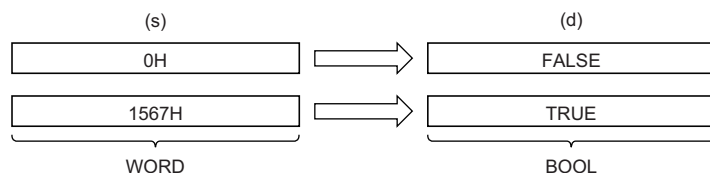
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	WORD
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	BOOL

機能

■演算処理

- ・ (s)に入力されたWORD型のデータを, BOOL型のデータに変換して(d)から出力します。
- ・ 入力値が0Hの場合は, FALSEを出力します。
- ・ 入力値が0H以外の場合は, TRUEを出力します。



- ・ (s)への入力値は, WORD型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

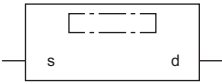
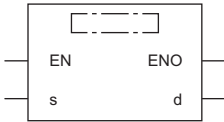
エラー

演算エラーはありません。

27.8 WORD型→DWORD型変換

WORD_TO_DWORD(_E)

WORD型データをDWORD型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=WORD_TO_DWORD(s); [EN/ENO付き] d:=WORD_TO_DWORD_E(EN,ENO,s);

設定データ

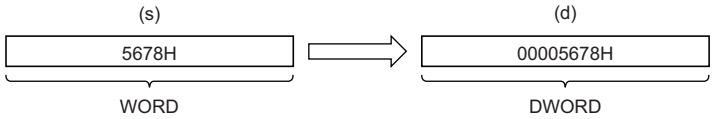
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	WORD
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	DWORD

機能

■演算処理

- (s)に入力されたWORD型のデータを, DWORD型のデータに変換して(d)から出力します。
- データ変換後, 上位16ビットは0となります。



- (s)への入力値は, WORD型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

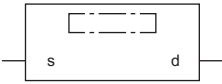
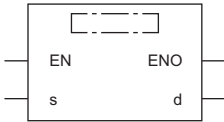
演算エラーはありません。

27.9 WORD型→INT型変換

27

WORD_TO_INT(_E)

WORD型データをINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=WORD_TO_INT(s); [EN/ENO付き] d:=WORD_TO_INT_E(EN,ENO,s);

設定データ

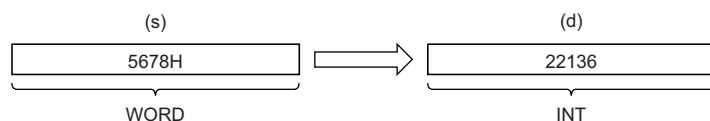
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	WORD
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	INT

機能

■演算処理

- (s)に入力されたWORD型のデータを, INT型のデータに変換して(d)から出力します。



- (s)への入力値は, WORD型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

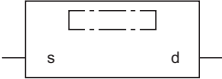
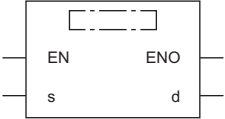
エラー

演算エラーはありません。

27.10 WORD型→DINT型変換

WORD_TO_DINT(_E)

WORD型データをDINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=WORD_TO_DINT(s); [EN/ENO付き] d:=WORD_TO_DINT_E(EN,ENO,s);

設定データ

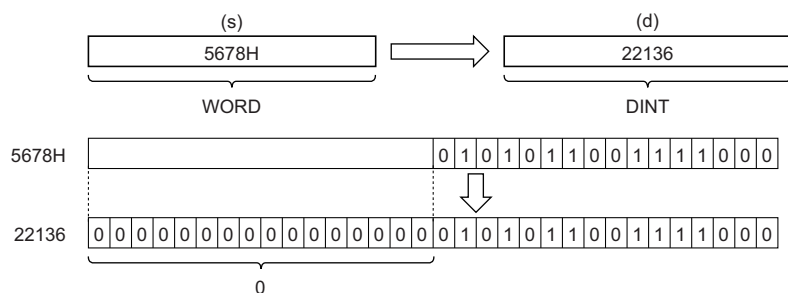
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	WORD
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	DINT

機能

■演算処理

- (s)に入力されたWORD型のデータを，DINT型のデータに変換して(d)から出力します。
- データ変換後，上位16ビットは0となります。



- (s)への入力値は，WORD型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合，(d)から出力されるデータは不定値となります。その場合は，(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

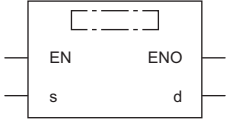
エラー

演算エラーはありません。

27.11 WORD型→TIME型変換

WORD_TO_TIME(_E)

WORD型データをTIME型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=WORD_TO_TIME(s); [EN/ENO付き] d:=WORD_TO_TIME_E(EN,ENO,s);

設定データ

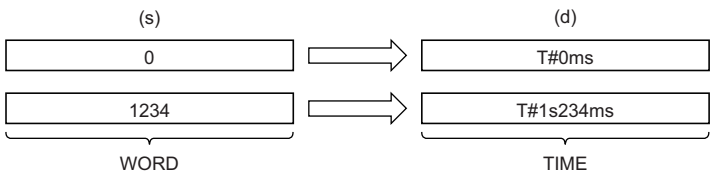
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	WORD
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	TIME

機能

■演算処理

- (s)に入力されたWORD型のデータを, TIME型のデータに変換して(d)から出力します。



- (s)への入力値は, WORD型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

演算エラーはありません。

27.12 WORD型→STRING型変換

WORD_TO_STRING(_E)

WORD型データをSTRING型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=WORD_TO_STRING(s);
[EN/ENO付き]		[EN/ENO付き] d:=WORD_TO_STRING_E(EN,ENO,s);

設定データ

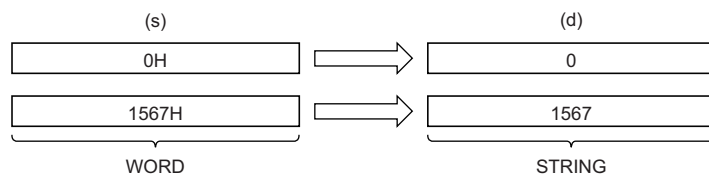
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	WORD
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	STRING(4)

機能

■演算処理

- (s)に入力されたWORD型のデータを, STRING型のデータに変換して(d)から出力します。



- (s)への入力値は, WORD型のデータ値です。
- 文字列の最後には, SM701(出力文字数切換)がOFFの場合, 00Hを格納します。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

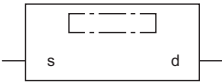
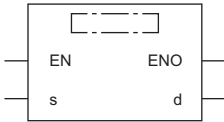
エラー

演算エラーはありません。

27.13 DWORD型→BOOL型変換

DWORD_TO_BOOL(_E)

DWORD型データをBOOL型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=DWORD_TO_BOOL(s); [EN/ENO付き] d:=DWORD_TO_BOOL_E(EN,ENO,s);

設定データ

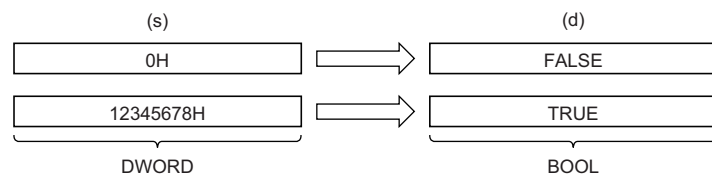
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	DWORD
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	BOOL

機能

■演算処理

- ・ (s)に入力されたDWORD型のデータを, BOOL型のデータに変換して(d)から出力します。
- ・ 入力値が0Hの場合は, FALSEを出力します。
- ・ 入力値が0H以外の場合は, TRUEを出力します。



- ・ (s)への入力値は, DWORD型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

演算エラーはありません。

27.14 DWORD型→WORD型変換

27

DWORD_TO_WORD(_E)

DWORD型データをWORD型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=DWORD_TO_WORD(s);
[EN/ENO付き]		[EN/ENO付き] d:=DWORD_TO_WORD_E(EN,ENO,s);

設定データ

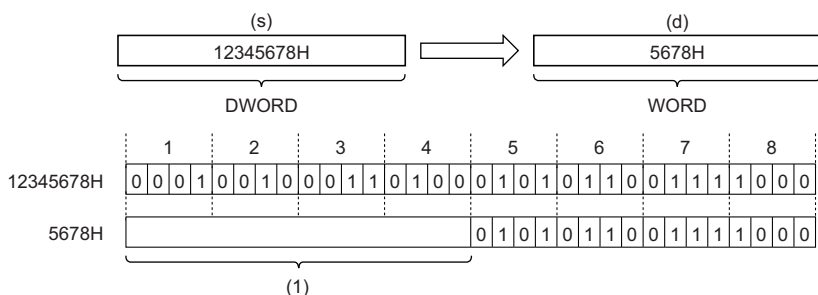
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	DWORD
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	WORD

機能

■演算処理

- (s)に入力されたDWORD型のデータを, WORD型のデータに変換して(d)から出力します。
- 入力値の上位16ビットの情報は捨てられます。(下記(1))



- (s)への入力値は, DWORD型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

Point

DWORD_TO_WORD(_E)を実行時、(s)から入力されたDWORD型データ値の上位16ビットの情報は捨てられます。

エラー

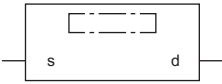
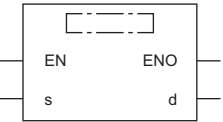
演算エラーはありません。

27.15 DWORD型→INT型変換

27

DWORD_TO_INT(_E)

DWORD型データをINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=DWORD_TO_INT(s); [EN/ENO付き] d:=DWORD_TO_INT_E(EN,ENO,s);

設定データ

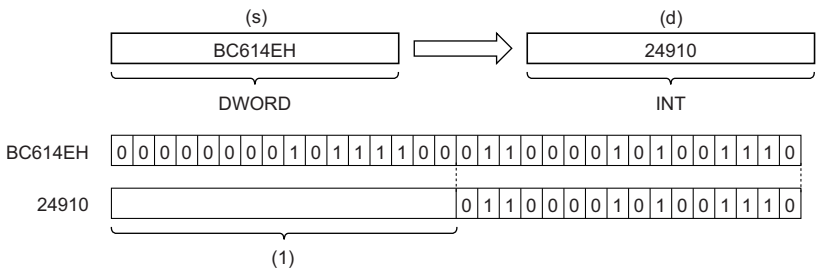
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	DWORD
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	INT

機能

■演算処理

- (s)に入力されたDWORD型のデータを, INT型のデータに変換して(d)から出力します。
- 入力値の上位16ビットの情報は捨てられます。(下記(1))



- (s)への入力値は, DWORD型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

Point

DWORD_TO_INT(_E)を実行時、(s)から入力されたDWORD型データ値の上位16ビットの情報は捨てられます。

エラー

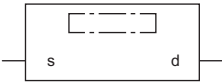
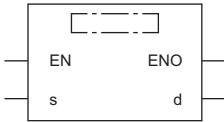
演算エラーはありません。

27.16 DWORD型→DINT型変換

27

DWORD_TO_DINT(_E)

DWORD型データをDINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=DWORD_TO_DINT(s); [EN/ENO付き] d:=DWORD_TO_DINT_E(EN,ENO,s);

設定データ

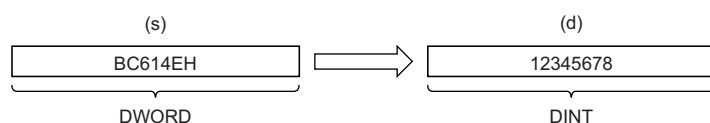
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	DWORD
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	DINT

機能

■演算処理

- (s)に入力されたDWORD型のデータを, DINT型のデータに変換して(d)から出力します。



- (s)への入力値は, DWORD型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

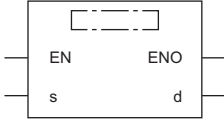
エラー

演算エラーはありません。

27.17 DWORD型→TIME型変換

DWORD_TO_TIME(_E)

DWORD型データをTIME型データへ変換します。

ラダー， FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=DWORD_TO_TIME(s); [EN/ENO付き] d:=DWORD_TO_TIME_E(EN,ENO,s);

設定データ

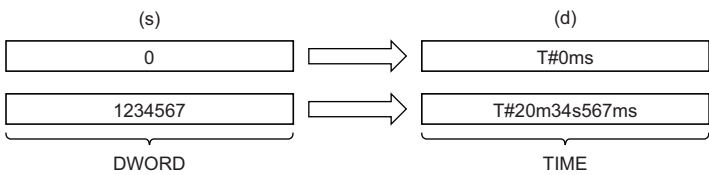
■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	DWORD
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	TIME

機能

■演算処理

- (s)に入力されたDWORD型のデータを，TIME型のデータに変換して(d)から出力します。



- (s)への入力値は，DWORD型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合，(d)から出力されるデータは不定値となります。その場合は，(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

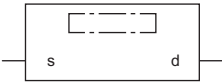
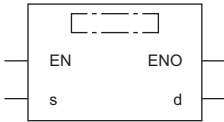
演算エラーはありません。

27.18 DWORD型→STRING型変換

27

DWORD_TO_STRING(_E)

DWORD型データをSTRING型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=DWORD_TO_STRING(s); [EN/ENO付き] d:=DWORD_TO_STRING_E(EN,ENO,s);

設定データ

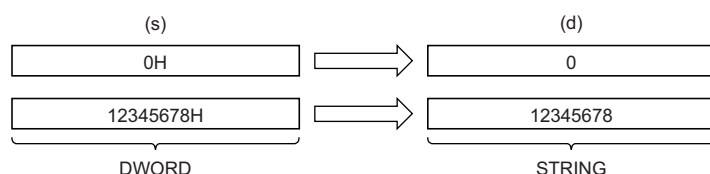
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	DWORD
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	STRING(8)

機能

■演算処理

- (s)に入力されたDWORD型のデータを, STRING型のデータに変換して(d)から出力します。



- (s)への入力値は, DWORD型のデータ値です。
- 文字列の最後には, SM701(出力文字数切換)がOFFの場合, 00Hを格納します。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

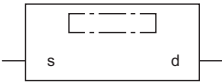
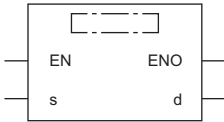
エラー

演算エラーはありません。

27.19 INT型→BOOL型変換

INT_TO_BOOL(_E)

INT型データをBOOL型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=INT_TO_BOOL(s); [EN/ENO付き] d:=INT_TO_BOOL_E(EN,ENO,s);

設定データ

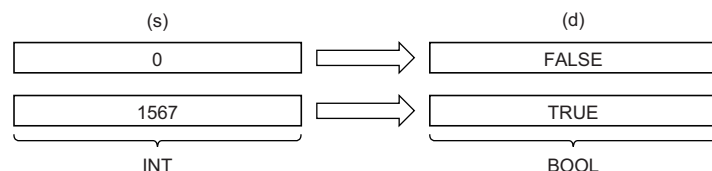
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	BOOL

機能

■演算処理

- ・ (s)に入力されたINT型のデータを, BOOL型のデータに変換して(d)から出力します。
- ・ 入力値が0の場合は, FALSEを出力します。
- ・ 入力値が0以外の場合は, TRUEを出力します。



- ・ (s)への入力値は, INT型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

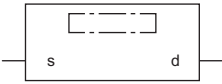
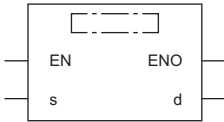
演算エラーはありません。

27.20 INT型→WORD型変換

27

INT_TO_WORD(_E)

INT型データをWORD型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=INT_TO_WORD(s); [EN/ENO付き] d:=INT_TO_WORD_E(EN,ENO,s);

設定データ

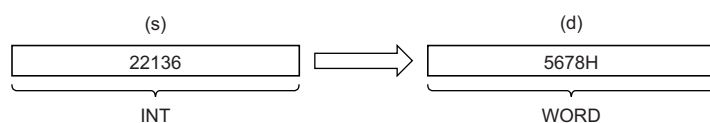
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	WORD

機能

■演算処理

- (s)に入力されたINT型のデータを, WORD型のデータに変換して(d)から出力します。



- (s)への入力値は, INT型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

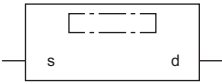
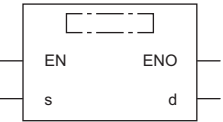
エラー

演算エラーはありません。

27.21 INT型→DWORD型変換

INT_TO_DWORD(_E)

INT型データをDWORD型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 		[EN/ENOなし] d:=INT_TO_DWORD(s);
[EN/ENO付き] 		[EN/ENO付き] d:=INT_TO_DWORD_E(EN,ENO,s);

設定データ

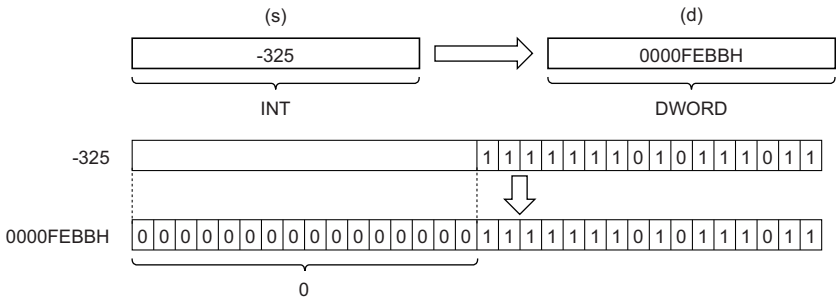
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	DWORD

機能

■演算処理

- (s)に入力されたINT型のデータを, DWORD型のデータに変換して(d)から出力します。
- データ変換後, 上位16ビットは0となります。



- (s)への入力値は, INT型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

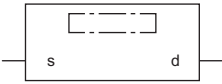
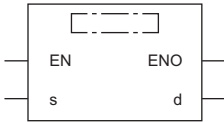
エラー

演算エラーはありません。

27.22 INT型→DINT型変換

INT_TO_DINT(_E)

INT型データをDINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=INT_TO_DINT(s); [EN/ENO付き] d:=INT_TO_DINT_E(EN,ENO,s);

設定データ

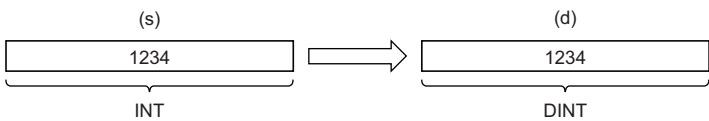
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	DINT

機能

■演算処理

- (s)に入力されたINT型のデータを, DINT型のデータに変換して(d)から出力します。



- (s)への入力値は, INT型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

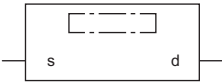
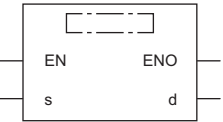
演算エラーはありません。

27.23 INT型→BCD型変換

27

INT_TO_BCD(_E)

INT型データをBCD型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=INT_TO_BCD(s); [EN/ENO付き] d:=INT_TO_BCD_E(EN,ENO,s);
		

設定データ

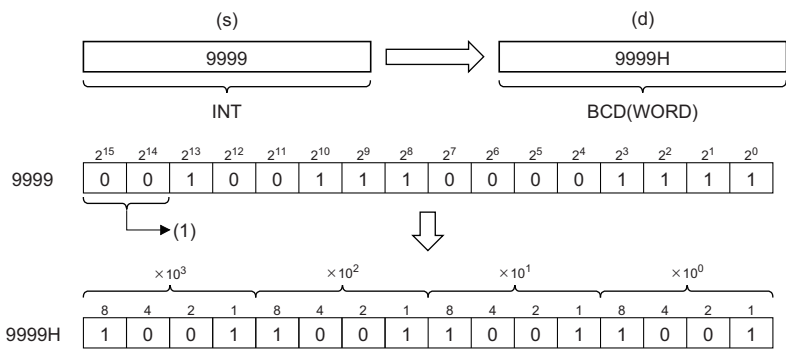
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	WORD

機能

■演算処理

- (s)に入力されたINT型のデータを, BCD型(WORD型)のデータに変換して(d)から出力します。



(1): 必ず0にしてください。

- (s)への入力値は, INT型のデータ値で0~9999の範囲内です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

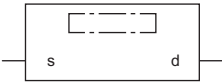
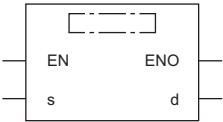
エラーコード (SD0)	内容
3281H	(s)のデータが0~9999以外のとき。

27.24 INT型→REAL型変換

27

INT_TO_REAL(_E)

INT型データをREAL型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=INT_TO_REAL(s); [EN/ENO付き] d:=INT_TO_REAL_E(EN,ENO,s);

設定データ

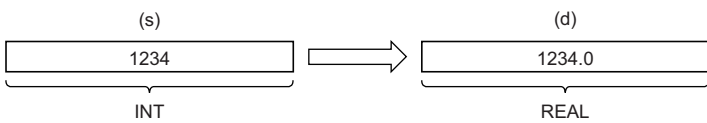
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	REAL

機能

■演算処理

- (s)に入力されたINT型のデータを, REAL型のデータに変換して(d)から出力します。



- (s)への入力値は, INT型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

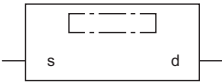
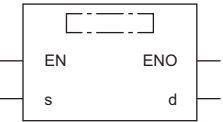
エラー

演算エラーはありません。

27.25 INT型→LREAL型変換

INT_TO_LREAL(_E)

INT型データをLREAL型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=INT_TO_LREAL(s); [EN/ENO付き] d:=INT_TO_LREAL_E(EN,ENO,s);

設定データ

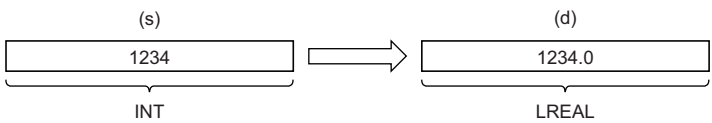
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	LREAL

機能

■演算処理

- (s)に入力されたINT型のデータを, LREAL型のデータに変換して(d)から出力します。



- (s)への入力値は, INT型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

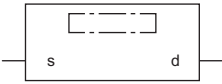
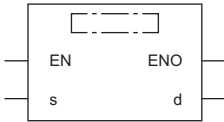
演算エラーはありません。

27.26 INT型→TIME型変換

27

INT_TO_TIME(_E)

INT型データをTIME型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=INT_TO_TIME(s); [EN/ENO付き] d:=INT_TO_TIME_E(EN,ENO,s);

設定データ

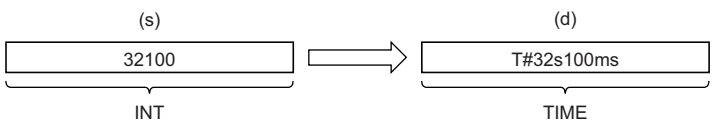
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	TIME

機能

■演算処理

- (s)に入力されたINT型のデータを, TIME型のデータに変換して(d)から出力します。



- (s)への入力値は, INT型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

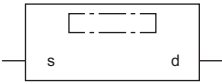
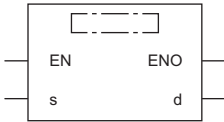
エラー

演算エラーはありません。

27.27 INT型→STRING型変換

INT_TO_STRING(_E)

INT型データをSTRING型データへ変換します。

ラダー， FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=INT_TO_STRING(s); [EN/ENO付き] d:=INT_TO_STRING_E(EN,ENO,s);
		

設定データ

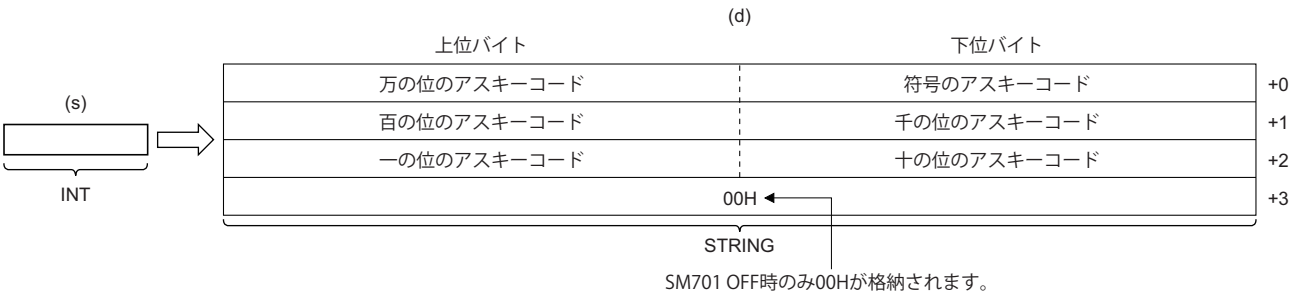
■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	STRING(6)

機能

■演算処理

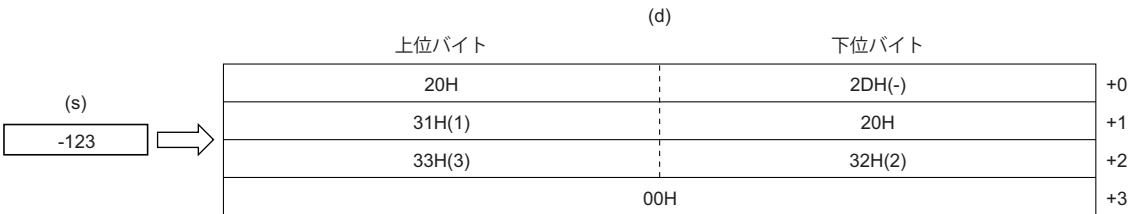
- (s)に入力されたINT型のデータを， STRING型のデータに変換して(d)から出力します。



- (s)への入力値は， INT型のデータ値です。
- 符号データには， 入力された値が正のとき20H(スペース)を格納し， 負のとき2DH(-)を格納します。
- 有効桁数が少ない場合は， 上位桁に20H(スペース)を格納します。

例

-123を入力した場合



- 文字列の最後(4ワード目)には， SM701(出力文字数切換)がOFFの場合， 00Hを格納します。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

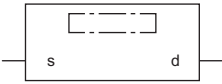
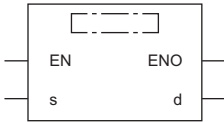
エラー

演算エラーはありません。

27.28 DINT型→BOOL型変換

DINT_TO_BOOL(_E)

DINT型データをBOOL型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=DINT_TO_BOOL(s); [EN/ENO付き] d:=DINT_TO_BOOL_E(EN,ENO,s);

設定データ

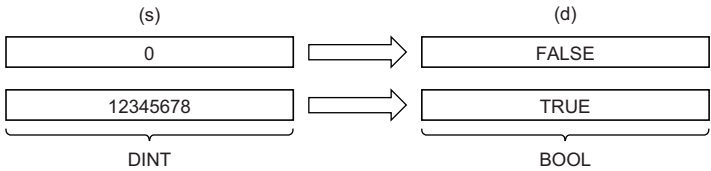
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	DINT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	BOOL

機能

■演算処理

- (s)に入力されたDINT型のデータを, BOOL型のデータに変換して(d)から出力します。
- 入力値が0の場合は, FALSEを出力します。
- 入力値が0以外の場合は, TRUEを出力します。



- (s)への入力値は, DINT型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

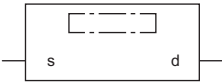
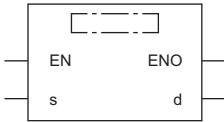
演算エラーはありません。

27.29 DINT型→WORD型変換

27

DINT_TO_WORD(_E)

DINT型データをWORD型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=DINT_TO_WORD(s); [EN/ENO付き] d:=DINT_TO_WORD_E(EN,ENO,s);

設定データ

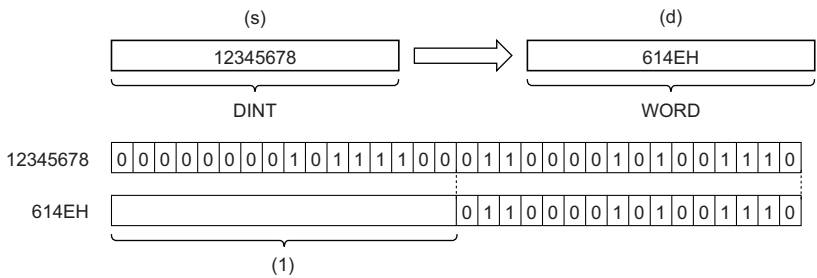
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	DINT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	WORD

機能

■演算処理

- (s)に入力されたDINT型のデータを, WORD型のデータに変換して(d)から出力します。
- 入力値の上位16ビットの情報は捨てられます。(下記(1))



- (s)への入力値は, DINT型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

Point

DINT_TO_WORD(_E)を実行時、(s)から入力されたDINT型データ値の上位16ビットの情報は捨てられます。

エラー

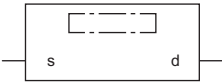
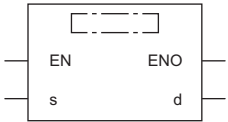
演算エラーはありません。

27.30 DINT型→DWORD型変換

27

DINT_TO_DWORD(_E)

DINT型データをDWORD型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=DINT_TO_DWORD(s); [EN/ENO付き] d:=DINT_TO_DWORD_E(EN,ENO,s);

設定データ

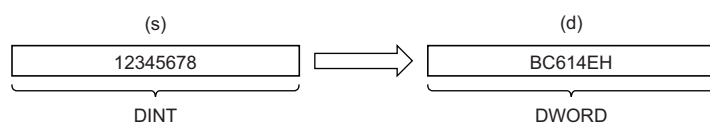
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	DINT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	DWORD

機能

■演算処理

- (s)に入力されたDINT型のデータを, DWORD型のデータに変換して(d)から出力します。



- (s)への入力値は, DINT型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

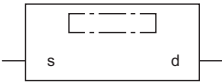
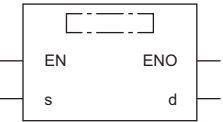
エラー

演算エラーはありません。

27.31 DINT型→INT型変換

DINT_TO_INT(_E)

DINT型データをINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=DINT_TO_INT(s); [EN/ENO付き] d:=DINT_TO_INT_E(EN,ENO,s);

設定データ

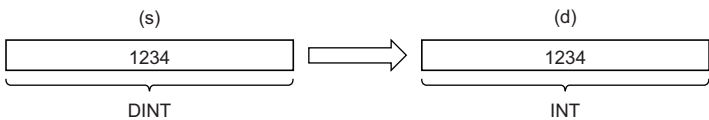
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	DINT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	INT

機能

■演算処理

- (s)に入力されたDINT型のデータを, INT型のデータに変換して(d)から出力します。



- (s)への入力値は, DINT型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

エラーコード (SD0)	内容
3281H	(s)で設定した符号付きBIN32ビットデータが-32768~32767の範囲外のとき。

27.32 DINT型→BCD型変換

DINT_TO_BCD(_E)

DINT型データをBCD型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] <code>d:=DINT_TO_BCD(s);</code> [EN/ENO付き] <code>d:=DINT_TO_BCD_E(EN,ENO,s);</code>

設定データ

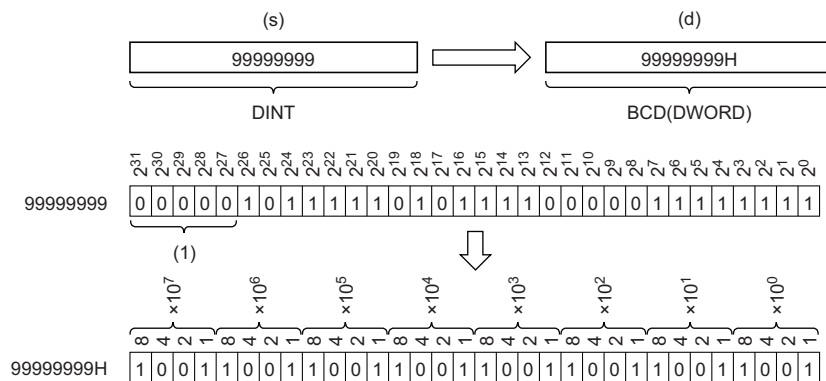
■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	DINT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_BIT

機能

■演算処理

- ・ (s)に入力されたDINT型のデータを、BCD型(DWORD型)のデータに変換して(d)から出力します。



(1): 必ず0にしてください。

- (s)への入力値は、DINT型のデータ値です。(d)がWORDの場合、入力値は0~9999の範囲です。(d)がDWORD型の場合、入力値は0~99999999の範囲内です。
- (d)には、WORD型、DWORD型が指定できます。BOOL型は指定できません。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

• (d)がWORD型の場合

エラーコード (SD0)	内容
3281H	(s)で設定した符号付きBIN32ビットデータが-32768~32767の範囲外の時。
	(s)のデータが0~9999以外の時。

• (d)がDWORD型の場合

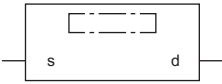
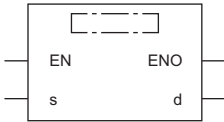
エラーコード (SD0)	内容
3281H	(s)のデータが0~99999999以外の時。

27.33 DINT型→REAL型変換

27

DINT_TO_REAL(_E)

DINT型データをREAL型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=DINT_TO_REAL(s); [EN/ENO付き] d:=DINT_TO_REAL_E(EN,ENO,s);

設定データ

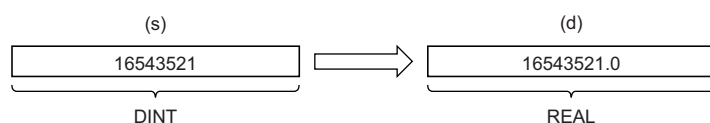
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	DINT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	REAL

機能

■演算処理

- (s)に入力されたDINT型のデータを, REAL型のデータに変換して(d)から出力します。



- (s)への入力値は, DINT型のデータ値です。
- REAL型データは, 32ビットの単精度で処理するため, 有効桁数は約7桁となります。
- 整数値が-16777216~16777215の範囲を超えた場合, 変換した値に誤差が生じます。(丸め誤差)

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

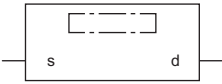
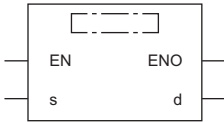
エラー

演算エラーはありません。

27.34 DINT型→LREAL型変換

DINT_TO_LREAL(_E)

DINT型データをLREAL型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=DINT_TO_LREAL(s); [EN/ENO付き] d:=DINT_TO_LREAL_E(EN,ENO,s);

設定データ

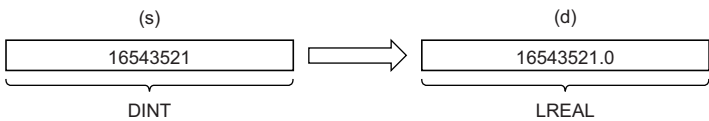
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	DINT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	LREAL

機能

■演算処理

- ・ (s)に入力されたDINT型のデータを, LREAL型のデータに変換して(d)から出力します。



- ・ (s)への入力値は, DINT型のデータ値です。
- ・ LREAL型データは, 64ビットの倍精度で処理するため, 有効桁数は約15桁となります。
- ・ 整数値が-2147483648~2147483647の範囲を超えた場合, 変換した値に誤差が生じます。(丸め誤差)

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

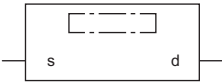
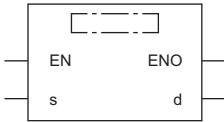
演算エラーはありません。

27.35 DINT型→TIME型変換

27

DINT_TO_TIME(_E)

DINT型データをTIME型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=DINT_TO_TIME(s); [EN/ENO付き] d:=DINT_TO_TIME_E(EN,ENO,s);

設定データ

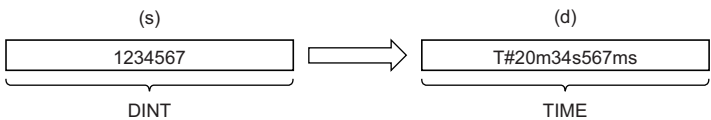
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	DINT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	TIME

機能

■演算処理

- (s)に入力されたDINT型のデータを, TIME型のデータに変換して(d)から出力します。



- (s)への入力値は, DINT型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

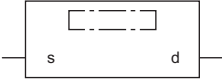
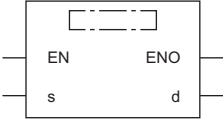
エラー

演算エラーはありません。

27.36 DINT型→STRING型変換

DINT_TO_STRING(_E)

DINT型データをSTRING型データへ変換します。

ラダー， FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=DINT_TO_STRING(s); [EN/ENO付き] d:=DINT_TO_STRING_E(EN,ENO,s);

設定データ

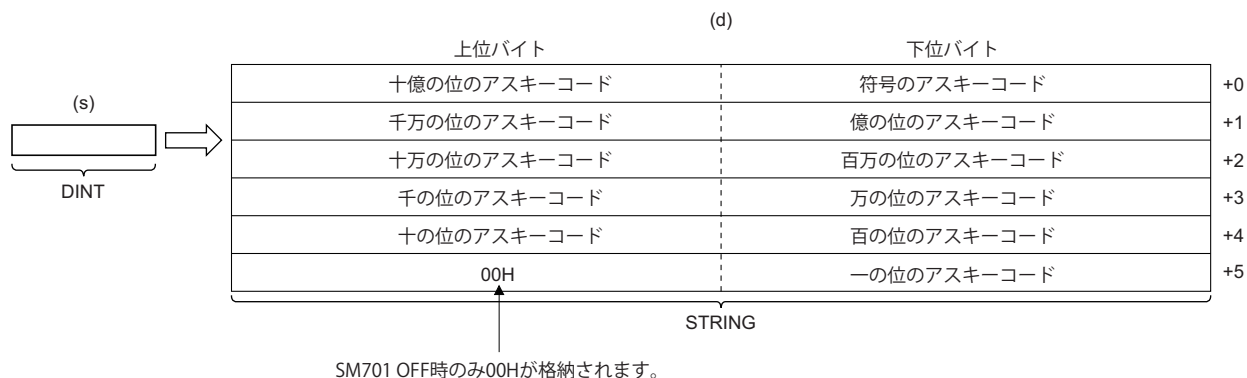
■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	DINT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	STRING(11)

機能

■演算処理

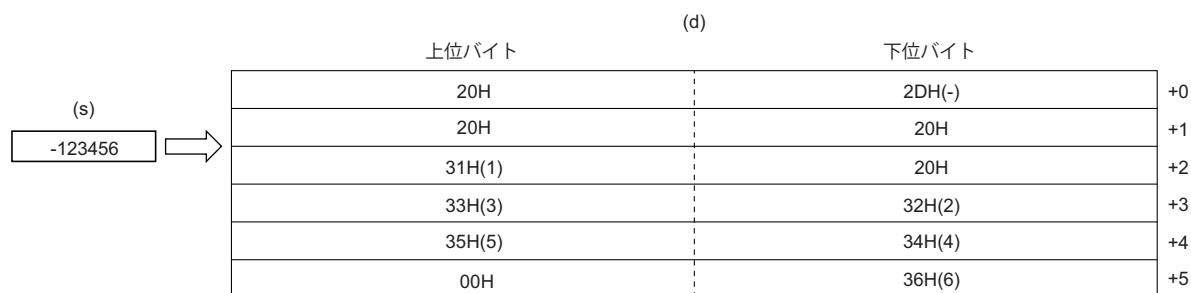
- (s)に入力されたDINT型のデータを，STRING型のデータに変換して(d)から出力します。



- (s)への入力値は，DINT型のデータ値です。
- 符号データには，入力された値が正のとき20H(スペース)を格納し，負のとき2DH(-)を格納します。
- 有効桁数が少ない場合は，上位桁に20H(スペース)を格納します。

例

-123456を入力した場合



- 文字列の最後(6ワード目上位バイト)には，SM701(出力文字数切換)がOFFの場合，00Hを格納します。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合，(d)から出力されるデータは不定値となります。その場合は，(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

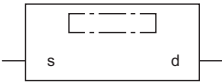
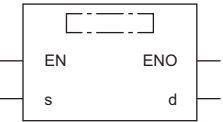
エラー

演算エラーはありません。

27.37 BCD型→INT型変換

BCD_TO_INT(_E)

BCD型データをINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=BCD_TO_INT(s); [EN/ENO付き] d:=BCD_TO_INT_E(EN,ENO,s);
		
[EN/ENO付き]		
		

設定データ

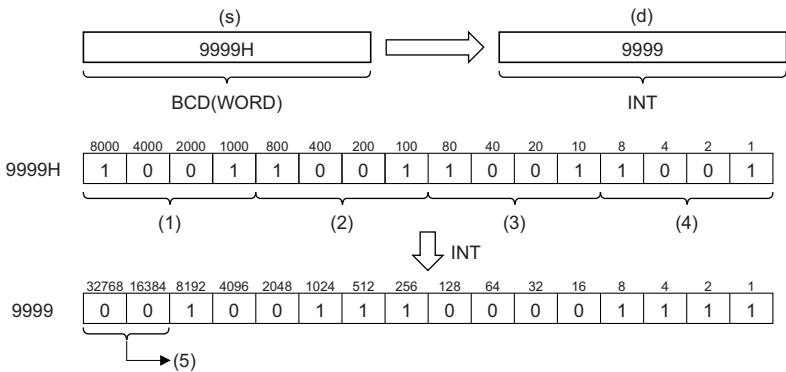
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	WORD
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	INT

機能

■演算処理

- (s)に入力されたBCD型(WORD型)のデータを, INT型のデータに変換して(d)から出力します。



- (s)への入力値は, WORD型のデータ値で0H~9999H(各桁は0~9の範囲)の範囲内です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

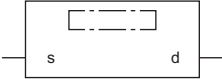
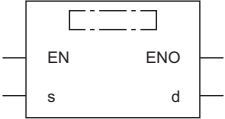
エラーコード (SD0)	内容
3281H	(s)の各桁に0~9以外の値があるとき。

- SM754をONしておくことにより、上記エラーを出さないようにすることができます。範囲外の数値を設定した場合、SM754のON/OFFに関わらず命令は実行されません。

27.38 BCD型→DINT型変換

BCD_TO_DINT(_E)

BCD型データをDINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=BCD_TO_DINT(s); [EN/ENO付き] d:=BCD_TO_DINT_E(EN,ENO,s);

設定データ

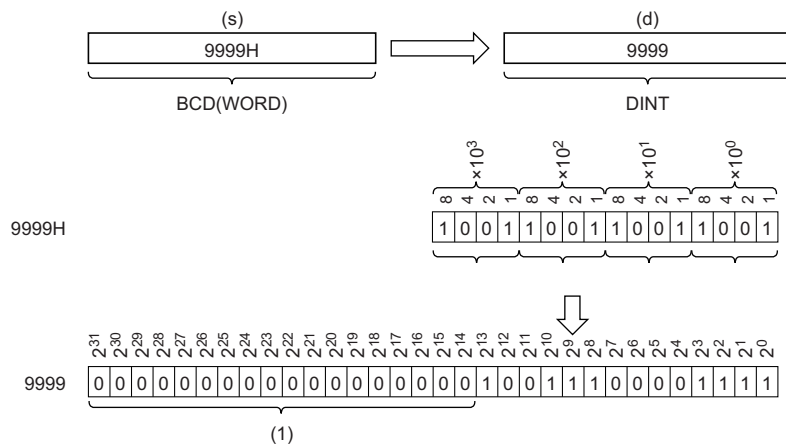
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_BIT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	DINT

機能

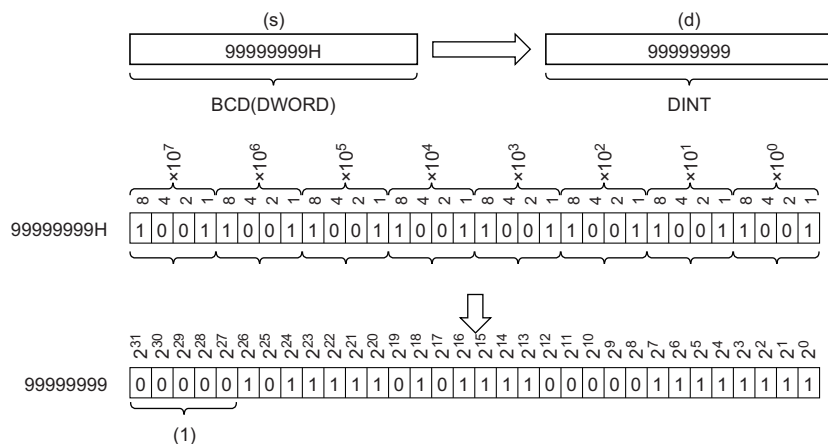
■演算処理

- (s)に入力されたBCD型(WORD型またはDWORD型)のデータを、DINT型のデータに変換して(d)から出力します。
- (s)にWORD型を指定した場合



(1): 必ず0になります。

- (s)にDWORD型を指定した場合



(1): 必ず0になります。

- (s)への入力値は、WORD型データ値で0H~9999H(各桁は0~9の範囲)、DWORD型のデータ値で0H~99999999H(各桁は0~9の範囲)の範囲内です。
- (s)には、WORD型、DWORD型が指定できます。BOOL型は指定できません。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

エラーコード (SD0)	内容
3281H	(s)の各桁に0~9以外の値があるとき。

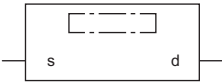
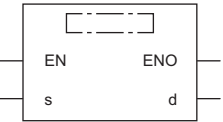
- SM754をONしておくことにより、上記エラーを出さないようにすることができます。なお、範囲外の数値を設定した場合、SM754のON/OFFに関わらず命令は実行されません。

27.39 BCD型→STRING型変換

27

BCD_TO_STRING(_E)

BCD型データをSTRING型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] d:=BCD_TO_STRING(s); [EN/ENO付き] d:=BCD_TO_STRING_E(EN,ENO,s);
		

設定データ

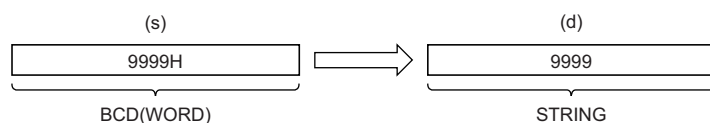
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_BIT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	STRING(8)

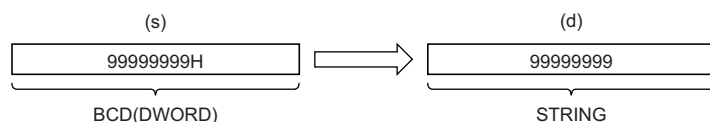
機能

■演算処理

- (s)に入力されたBCD型(WORD型またはDWORD型)のデータを, STRING型のデータに変換して(d)から出力します。
- (s)にWORD型を指定した場合



- (s)にDWORD型を指定した場合



- (s)には, WORD型, DWORD型が指定できます。BOOL型は指定できません。
- 文字列の最後には, SM701(出力文字数切換)がOFFの場合, 00Hを格納します。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

- (s)がWORD型の場合

エラーコード (SD0)	内容
3281H	(s)のデータが0~9999以外るとき。

- (s)がDWORD型の場合

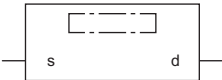
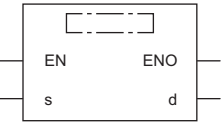
エラーコード (SD0)	内容
3281H	(s)のデータが0~99999999以外るとき。

27.40 REAL型→INT型変換

27

REAL_TO_INT(_E)

REAL型データをINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=REAL_TO_INT(s); [EN/ENO付き] d:=REAL_TO_INT_E(EN,ENO,s);

設定データ

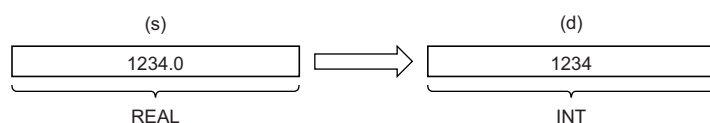
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	REAL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	INT

機能

■演算処理

- (s)に入力されたREAL型のデータを, INT型のデータに変換して(d)から出力します。



- (s)への入力値は, REAL型のデータ値で-32768~32767の範囲内です。
- 変換後のデータは, REAL型データ値の小数点以下1桁目を四捨五入した値となります。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

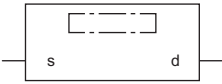
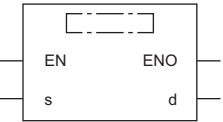
エラーコード (SD0)	内容
3281H	(s)で設定した単精度実数が-32768~32767の範囲外の時。
3282H	• (s)に特殊な数を設定したとき。 • 設定した単精度実数が下記範囲内でない。 $-2^{128} < (s) \leq -2^{-126}$, 0, $2^{-126} \leq (s) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38, 0, E1.17549435-38~E3.40282347+38) • 設定したデバイス, ラベルの内容が-0, 非正規化数, 非数, $\pm\infty$の時。

27.41 REAL型→DINT型変換

27

REAL_TO_DINT(_E)

REAL型データをDINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=REAL_TO_DINT(s); [EN/ENO付き] d:=REAL_TO_DINT_E(EN,ENO,s);

設定データ

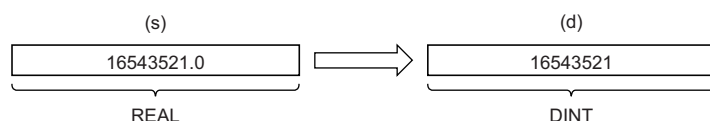
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	REAL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	DINT

機能

■演算処理

- (s)に入力されたREAL型のデータを, DINT型のデータに変換して(d)から出力します。



- (s)への入力値は, REAL型のデータ値で-2147483648~2147483647の範囲内です。
- 変換後のデータは, REAL型データ値の小数点以下1桁目を四捨五入した値となります。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

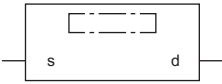
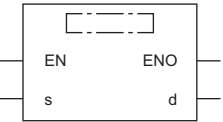
エラーコード (SD0)	内容
3281H	(s)で設定した単精度実数が-2147483648~2147483647の範囲外の時。
3282H	• (s)に特殊な数を設定したとき。 • 設定した単精度実数が下記範囲内でない。 $-2^{128} < (s) \leq -2^{-126}$, 0, $2^{-126} \leq (s) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38, 0, E1.17549435-38~E3.40282347+38) • 設定したデバイス, ラベルの内容が-0, 非正規化数, 非数, $\pm\infty$の時。

27.42 REAL型→LREAL型変換

27

REAL_TO_LREAL(_E)

REAL型データをLREAL型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=REAL_TO_LREAL(s);
	[EN/ENO付き]	[EN/ENO付き] d:=REAL_TO_LREAL_E(EN,ENO,s);
		

設定データ

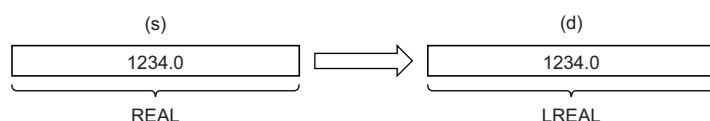
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	REAL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	LREAL

機能

■演算処理

- (s)に入力されたREAL型のデータを, LREAL型のデータに変換して(d)から出力します。



- (s)への入力値は, REAL型のデータ値です。
- REAL型データは, 32ビットの単精度で処理するため, 有効桁数は約7桁となります。
- 整数値が-16777216~16777215の範囲を超えた場合, 変換した値に誤差が生じます。(丸め誤差)

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

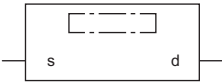
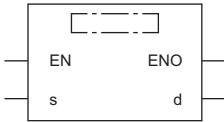
エラーコード (SD0)	内容
3282H	(s)の内容が-0, または下記範囲内でないとき。 $-2^{128} < (s) \leq -2^{126}$, 0 , $2^{-126} \leq (s) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38, 0 , E1.17549435-38~E3.40282347+38)

27.43 REAL型→STRING型変換

27

REAL_TO_STRING(_E)

REAL型データをSTRING型(指数型式)データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=REAL_TO_STRING(s); [EN/ENO付き] d:=REAL_TO_STRING_E(EN,ENO,s);
		

設定データ

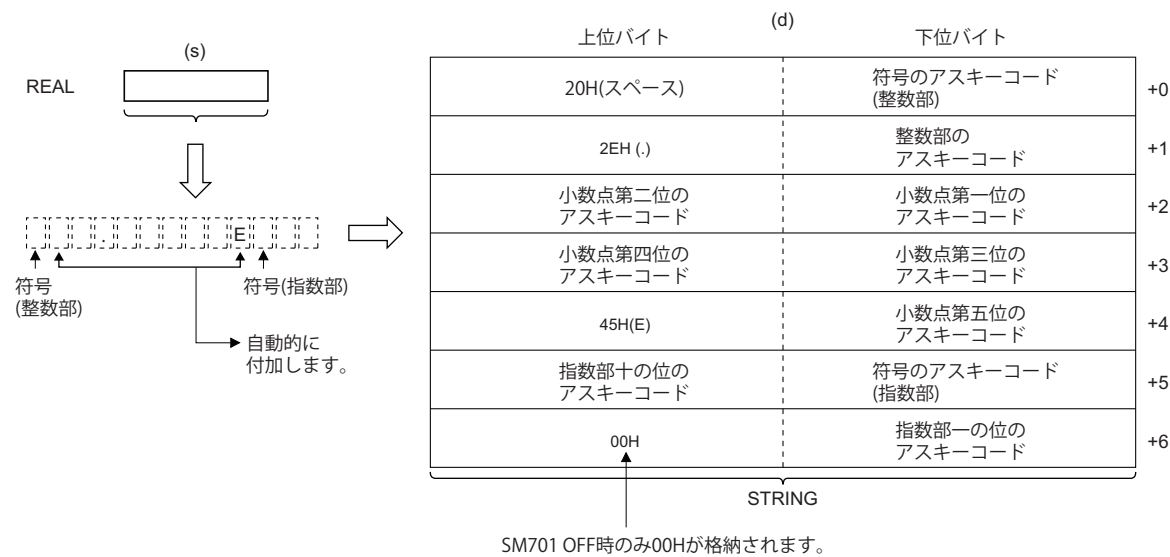
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	REAL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	STRING(13)

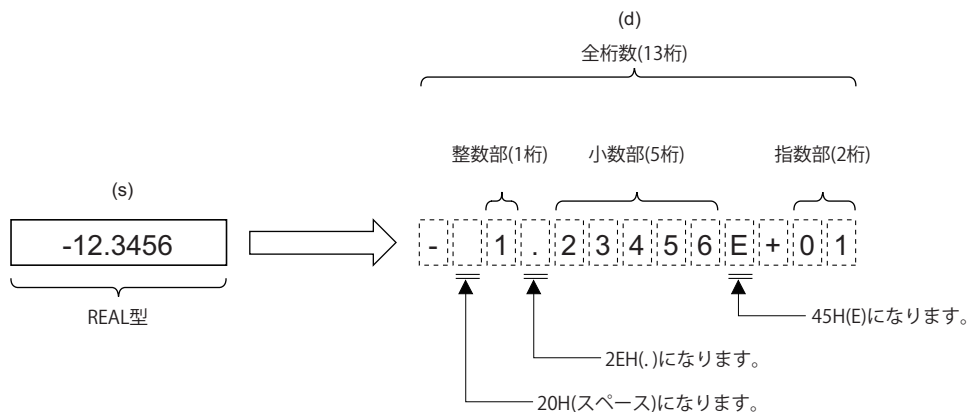
機能

■演算処理

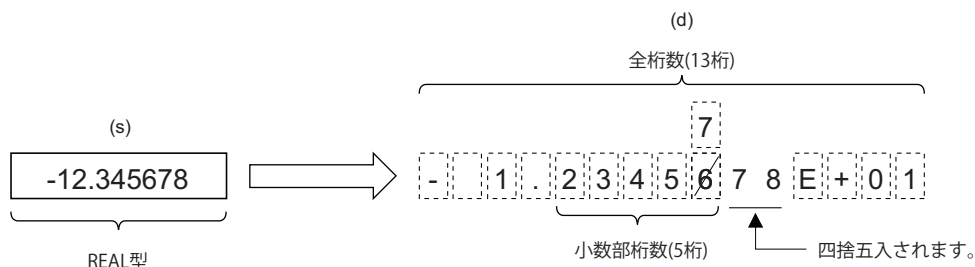
- (s)に入力されたREAL型のデータを, STRING型(指数型式)のデータに変換して(d)から出力します。



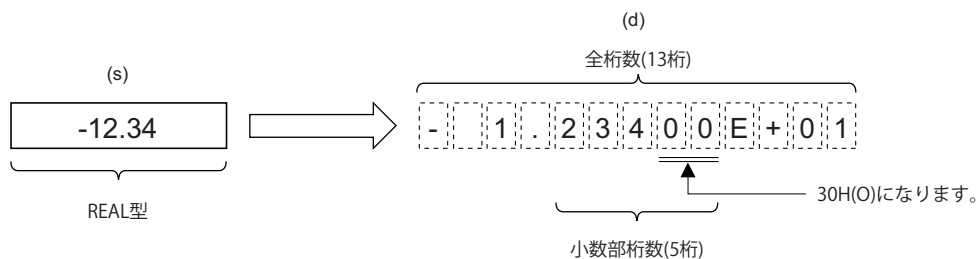
- (s)への入力値は、REAL型のデータ値です。
- 変換後の文字列データは、(d)から下記のように出力されます。
 - 整数部、小数部、指数部の桁数は固定です。(整数部:1桁、小数部:5桁、指数部:2桁)
 - 2バイト目、4バイト目、10バイト目には、それぞれ自動的に「20H」(スペース)、「2EH」(.), 「45H」(E)を格納します。



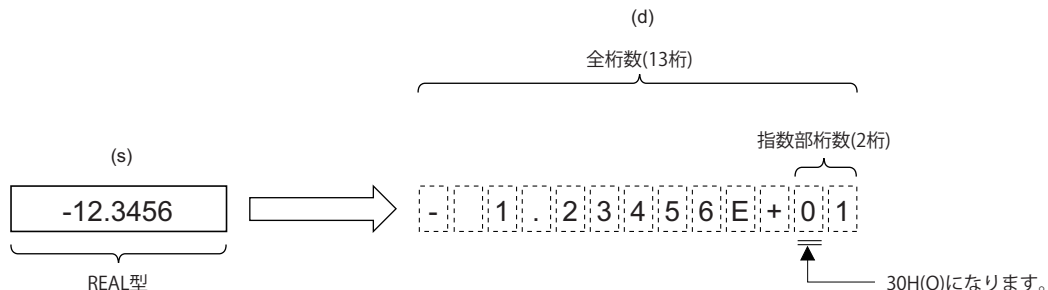
- 符号データ(整数部)には、入力された値が正のとき「20H」(スペース)を格納し、負のとき「2DH」(-)を格納します。
- 小数部の6桁目以降は、四捨五入されます。



- 有効桁数が少ない場合は、小数部に「30H」(0)を格納します。



- 符号データ(指数部)には、指数が正のとき「2BH」(+)を格納し、負のとき「2DH」(-)を格納します。
- 指数部が1桁の場合は、指数部十の位へ「30H」(0)を格納します。



- 文字列の最後(7ワード目)には、自動的に00Hを格納します。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

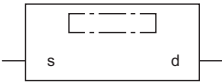
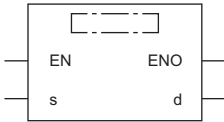
エラー

エラーコード (SD0)	内容
3282H	(s)の内容が下記範囲内でないとき。 $-2^{128} < (s) \leq -2^{126}$, 0, $2^{-126} \leq (s) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38, 0, E1.17549435-38~E3.40282347+38) (s)の内容が-0, 非正規化数, 非数, $\pm\infty$のとき。
3286H	(d)で指定されたラベル, または指定したデバイス番号から該当デバイスの最終デバイス番号までの点数に, 変換後の文字列がすべて格納できないとき。

27.44 LREAL型→INT型変換

LREAL_TO_INT(_E)

LREAL型データをINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=LREAL_TO_INT(s); [EN/ENO付き] d:=LREAL_TO_INT_E(EN,ENO,s);

設定データ

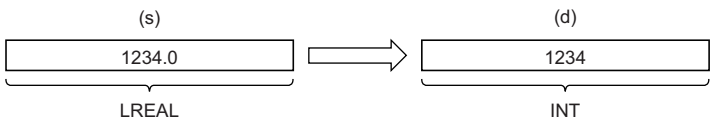
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	LREAL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	INT

機能

■演算処理

- (s)に入力されたLREAL型のデータを, INT型のデータに変換して(d)から出力します。



- (s)への入力値は, LREAL型のデータ値です。
- 変換後のデータは, LREAL型データ値の小数点以下1桁目を四捨五入した値となります。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件		演算結果
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

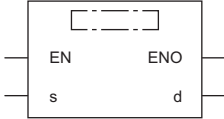
^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラーコード (SD0)	内容
3281H	(s)の内容が-32768~32767以外るとき。
3282H	• (s)の内容が-0, または下記の範囲内でないとき。 $-2^{1024} < (s), (d) \leq -2^{1022}, 0, 2^{1022} \leq (s), (d) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308, 0, E2.2250738585072014-308~E1.7976931348623157+308) • 設定したデバイス, ラベルの内容が-0, 非正規化数, 非数, $\pm\infty$のとき。

27.45 LREAL型→DINT型変換

LREAL_TO_DINT(_E)

LREAL型データをDINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=LREAL_TO_DINT(s); [EN/ENO付き] d:=LREAL_TO_DINT_E(EN,ENO,s);

設定データ

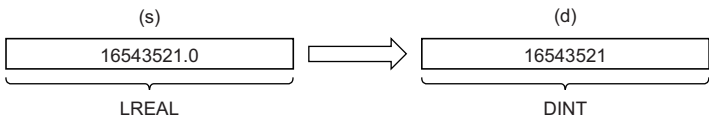
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	LREAL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	DINT

機能

■演算処理

- (s)に入力されたLREAL型のデータを, DINT型のデータに変換して(d)から出力します。



- (s)への入力値は, LREAL型のデータ値です。
- 変換後のデータは, LREAL型データ値の小数点以下1桁目を四捨五入した値となります。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件		演算結果
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

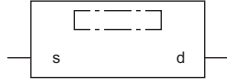
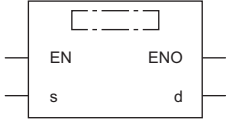
^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラーコード (SD0)	内容
3281H	(s)の内容が-2147483648~2147483647以外するとき。
3282H	• (s)の内容が-0, または下記の範囲内でないとき。 $-2^{1024} < (s), (d) \leq -2^{-1022}, 0, 2^{-1022} \leq (s), (d) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308, 0, E2.2250738585072014-308~E1.7976931348623157+308) • 設定したデバイス, ラベルの内容が-0, 非正規化数, 非数, $\pm\infty$のとき。

27.46 LREAL型→REAL型変換

LREAL_TO_REAL(_E)

LREAL型データをREAL型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=LREAL_TO_REAL(s); [EN/ENO付き] d:=LREAL_TO_REAL_E(EN,ENO,s);

設定データ

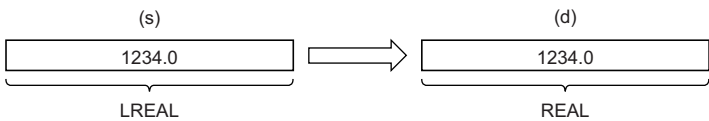
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	LREAL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	REAL

機能

■演算処理

- (s)に入力されたLREAL型のデータを, REAL型のデータに変換して(d)から出力します。



- (s)への入力値は, LREAL型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

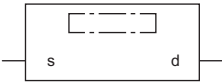
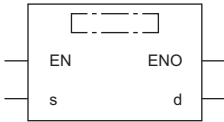
^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラーコード (SD0)	内容
3282H	(s)の内容が-0, または下記範囲内でないとき。 $-2^{1024} < (s), (d) \leq -2^{-1022}, 0, 2^{-1022} \leq (s), (d) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308, 0, E2.2250738585072014-308~E1.7976931348623157+308)
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき) $ (d) < 2^{128}$

27.47 TIME型→BOOL型変換

TIME_TO_BOOL(_E)

TIME型データをBOOL型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=TIME_TO_BOOL(s); [EN/ENO付き] d:=TIME_TO_BOOL_E(EN,ENO,s);

設定データ

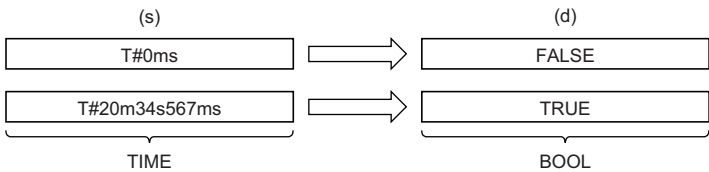
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	TIME
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	BOOL

機能

■演算処理

- (s)に入力されたTIME型のデータを, BOOL型のデータに変換して(d)から出力します。



■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

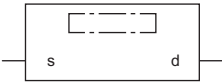
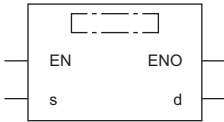
演算エラーはありません。

27.48 TIME型→WORD型変換

27

TIME_TO_WORD(_E)

TIME型データをWORD型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=TIME_TO_WORD(s); [EN/ENO付き] d:=TIME_TO_WORD_E(EN,ENO,s);

設定データ

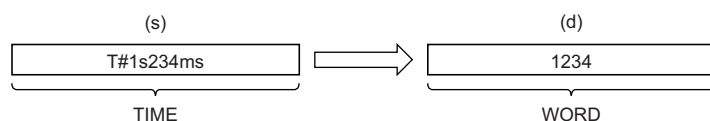
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	TIME
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	WORD

機能

■演算処理

- (s)に入力されたTIME型のデータを, WORD型のデータに変換して(d)から出力します。



- (s)への入力値は, TIME型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

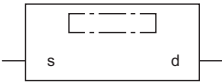
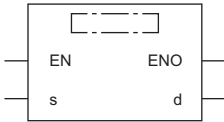
エラー

演算エラーはありません。

27.49 TIME型→DWORD型変換

TIME_TO_DWORD(_E)

TIME型データをDWORD型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=TIME_TO_DWORD(s); [EN/ENO付き] d:=TIME_TO_DWORD_E(EN,ENO,s);

設定データ

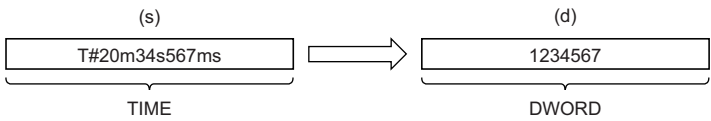
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	TIME
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	DWORD

機能

■演算処理

- (s)に入力されたTIME型のデータを, DWORD型のデータに変換して(d)から出力します。



- (s)への入力値は, TIME型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

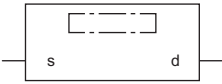
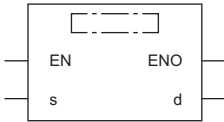
演算エラーはありません。

27.50 TIME型→INT型変換

27

TIME_TO_INT(_E)

TIME型データをINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=TIME_TO_INT(s); [EN/ENO付き] d:=TIME_TO_INT_E(EN,ENO,s);

設定データ

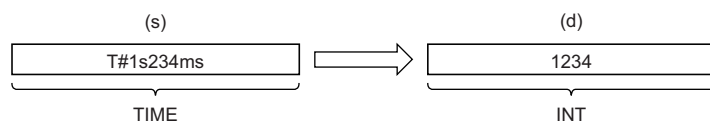
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	TIME
ENO	出力状態(TRUE: 正常実行, FALSE: 異常または停止)	出力変数	BOOL
d	出力	出力変数	INT

機能

■演算処理

- (s)に入力されたTIME型のデータを, INT型のデータに変換して(d)から出力します。



- (s)への入力値は, TIME型のデータ値です。
- INT型へ変換する際, TIME型の上位16ビット(1ワード)分のデータは切り捨てられます。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

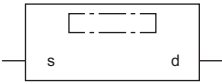
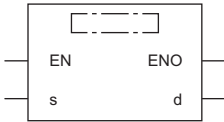
エラー

演算エラーはありません。

27.51 TIME型→DINT型変換

TIME_TO_DINT(_E)

TIME型データをDINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 		[EN/ENOなし] d:=TIME_TO_DINT(s);
[EN/ENO付き] 		[EN/ENO付き] d:=TIME_TO_DINT_E(EN,ENO,s);

設定データ

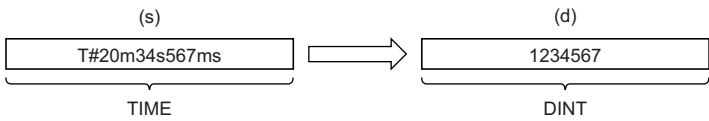
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	TIME
ENO	出力状態(TRUE: 正常実行, FALSE: 異常または停止)	出力変数	BOOL
d	出力	出力変数	DINT

機能

■演算処理

- (s)に入力されたTIME型のデータを, DINT型のデータに変換して(d)から出力します。



- (s)への入力値は, TIME型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件		演算結果
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

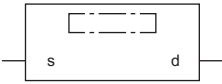
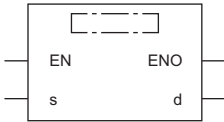
演算エラーはありません。

27.52 TIME型→STRING型変換

27

TIME_TO_STRING(_E)

TIME型データをSTRING型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=TIME_TO_STRING(s); [EN/ENO付き] d:=TIME_TO_STRING_E(EN,ENO,s);

設定データ

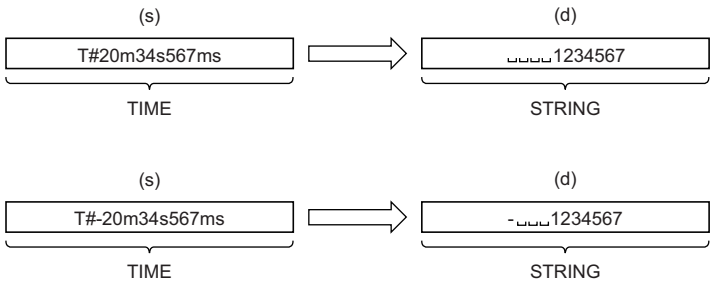
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	TIME
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	STRING STRING(11)

機能

■演算処理

- (s)に入力されたTIME型のデータを, STRING型のデータに変換して(d)から出力します。



- (s)への入力値は, TIME型のデータ値です。
- 文字列の最後には, SM701(出力文字数切替)がOFFの場合, 00Hを格納します。
- 出力に格納する演算結果は, 下記ようになります。
- 1文字目には, BINデータが正のとき20H(スペース)を格納し, 負のとき2DH(-)を格納します。
- 有効桁数の左側には20H(スペース)を格納します。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

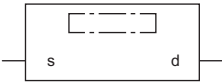
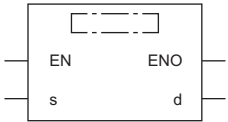
演算エラーはありません。

27.53 STRING型→BOOL型変換

27

STRING_TO_BOOL(_E)

STRING型データをBOOL型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] d:=STRING_TO_BOOL(s); [EN/ENO付き] d:=STRING_TO_BOOL_E(EN,ENO,s);
		

設定データ

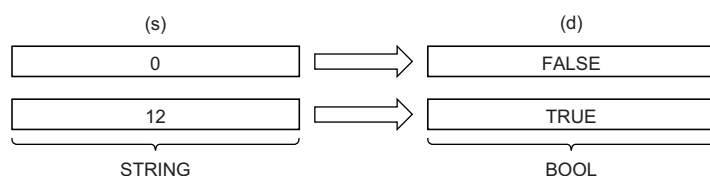
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	STRING(1)
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	BOOL

機能

■演算処理

- (s)に入力されたSTRING型(小数点型式/指数型式)のデータを, BOOL型のデータに変換して(d)から出力します。



■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件		演算結果
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

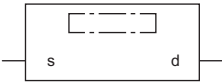
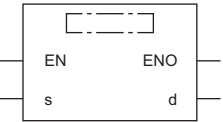
エラー

演算エラーはありません。

27.54 STRING型→WORD型変換

STRING_TO_WORD(_E)

STRING型データをWORD型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=STRING_TO_WORD(s); [EN/ENO付き] d:=STRING_TO_WORD_E(EN,ENO,s);
		

設定データ

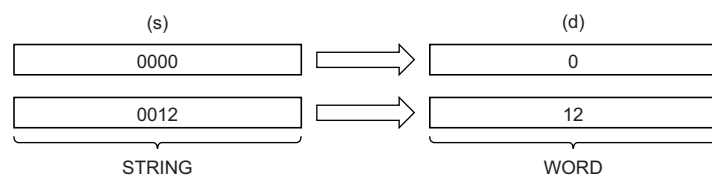
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	STRING(4)
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	WORD

機能

■演算処理

- (s)に入力されたSTRING型のデータを, WORD型のデータに変換して(d)から出力します。



- (s)への入力値は, STRING型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

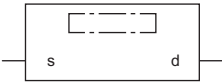
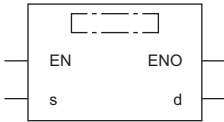
エラーコード (SD0)	内容
3281H	入力の各桁のアスキーコードが, 30H~39H, 41H~46H以外のとき。

27.55 STRING型→DWORD型変換

27

STRING_TO_DWORD(_E)

STRING型データをDWORD型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] d:=STRING_TO_DWORD(s); [EN/ENO付き] d:=STRING_TO_DWORD_E(EN,ENO,s);
		

設定データ

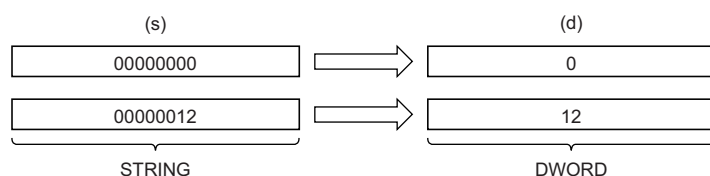
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	STRING(8)
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	DWORD

機能

■演算処理

- (s)に入力されたSTRING型のデータを, DWORD型のデータに変換して(d)から出力します。



- (s)への入力値は, STRING型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

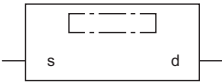
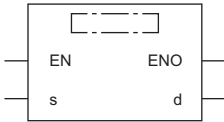
エラー

エラーコード (SD0)	内容
3281H	入力の各桁のアスキーコードが, 30H~39H, 41H~46H以外のとき。

27.56 STRING型→INT型変換

STRING_TO_INT(_E)

STRING型データをINT型データへ変換します。

ラダー， FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=STRING_TO_INT(s); [EN/ENO付き] d:=STRING_TO_INT_E(EN,ENO,s);
		

設定データ

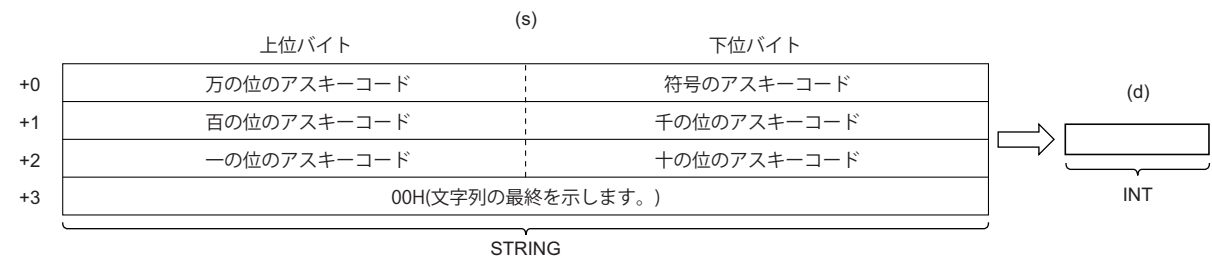
■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	STRING(6)
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	INT

機能

■演算処理

- (s)に入力されたSTRING型のデータを，INT型のデータに変換して(d)から出力します。



- (s)への入力値は，STRING型のデータ値で下記に示す範囲内です。
 - アスキーコードで30H~39H, 20H, 2DH, 00Hの範囲
 - STRING型のデータ値で-32768~32767の範囲

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

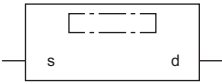
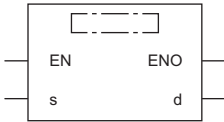
^{*1} ENOからFALSEが出力された場合，(d)から出力されるデータは不定値となります。その場合は，(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラーコード (SD0)	内容
3281H	(s)に変換できない不正なデータを入力したとき。 • 各位のアスキーコードが、30H~39H, 20H, 00H以外。 • アスキーデータが-32768~32767以外。

27.57 STRING型→DINT型変換

STRING_TO_DINT(_E)

STRING型データをDINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=STRING_TO_DINT(s); [EN/ENO付き] d:=STRING_TO_DINT_E(EN,ENO,s);

設定データ

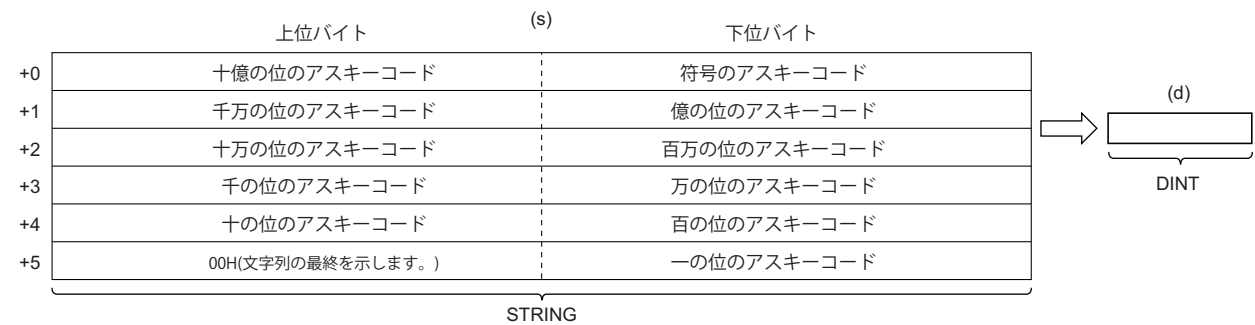
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	STRING(11)
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	DINT

機能

■演算処理

- (s)に入力されたSTRING型のデータを, DINT型のデータに変換して(d)から出力します。



- (s)への入力値は, STRING型のデータ値で下記に示す範囲内です。
 - アスキーコードで30H~39H, 20H, 2DH, 00Hの範囲
 - STRING型のデータ値で-2147483648~2147483647の範囲

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

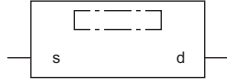
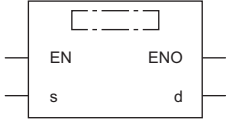
^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラーコード (SD0)	内容
3281H	(s)に変換できない不正なデータを入力したとき。 • 各位のアスキーコードが、30H~39H, 20H, 00H以外。 • アスキーデータが-2147483648~2147483647以外。

27.58 STRING型→BCD型変換

STRING_TO_BCD(_E)

STRING型データをBCD型データへ変換します。

ラダー， FBD/LD		ST
[EN/ENOなし] 		[EN/ENOなし] d:=STRING_TO_BCD(s); [EN/ENO付き] d:=STRING_TO_BCD_E(EN,ENO,s);
[EN/ENO付き] 		

設定データ

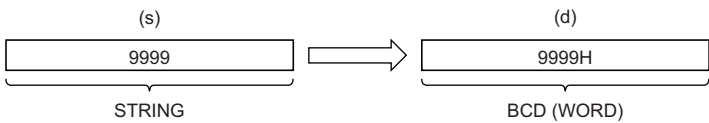
■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行， FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	STRING(8)
ENO	出力状態(TRUE: 正常， FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_BIT

機能

■演算処理

- (s)に入力されたSTRING型4文字分のデータを， BCD型(WORD型)のデータに変換して(d)から出力します。



- 文字列中に20H(スペース)が含まれていた場合は， 20Hを無視して変換します。
- 文字列中の20H(スペース)， 30H(0)も1文字としてカウントします。
- (s)への入力値は， STRING型のデータ値で下記に示す範囲内です。
アスキーコードで30H~39H， 20H， 00Hの範囲
- 入力文字列が4文字に満たない場合は， 文字列の後ろに0を補完した4文字で変換します。このため， 4文字に満たない文字列をBCD変換したい場合は， 0パディングした文字列('1'の場合'0001')を入力してください。
- 文字列が4文字を越える場合は， 左から4文字が変換対象となります。

入力文字列	変換文字列	出力(BCD型)
'1'	'1000'	1000H(4096D)
'12'	'1200'	1200H(4608D)
'123'	'1230'	1230H(4656D)
'1234'	'1234'	1234H(4660D)
'12345'	'1234'	1234H(4660D)

- (d)には， WORD型， DWORD型が指定できます。 BOOL型は指定できません。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

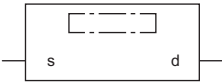
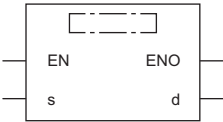
エラー

エラーコード (SD0)	内容
3281H	アスキーコードで30H~39H, 20H, 00H以外の範囲の文字列を入力したとき。

27.59 STRING型→REAL型変換

STRING_TO_REAL(_E)

STRING型データをREAL型データへ変換します。

ラダー， FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=STRING_TO_REAL(s); [EN/ENO付き] d:=STRING_TO_REAL_E(EN,ENO,s);
		

設定データ

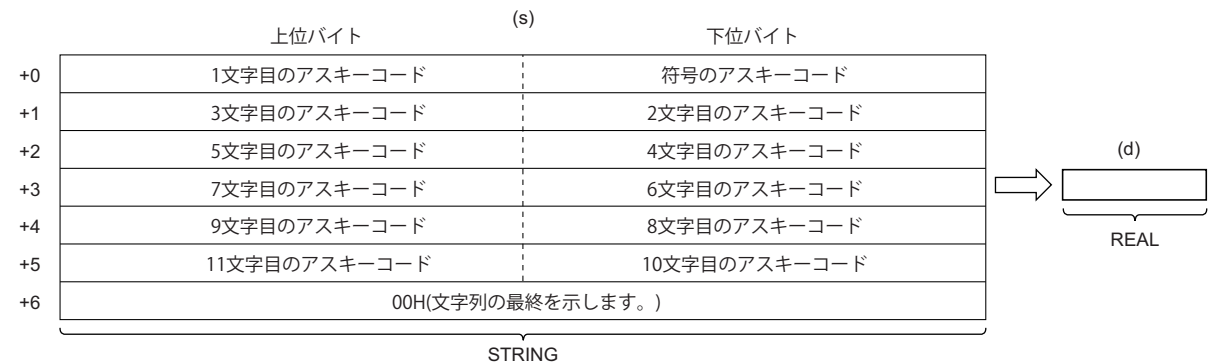
■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	STRING(24)
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	REAL

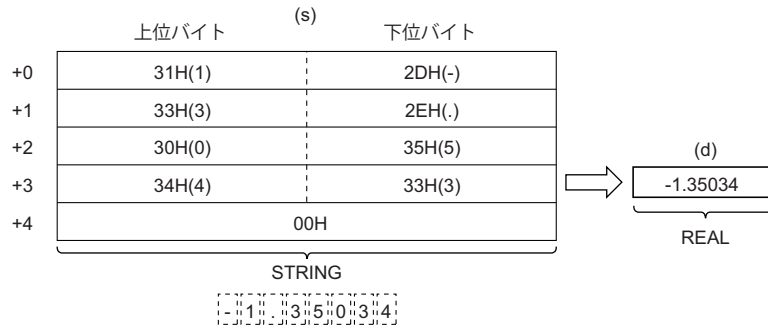
機能

■演算処理

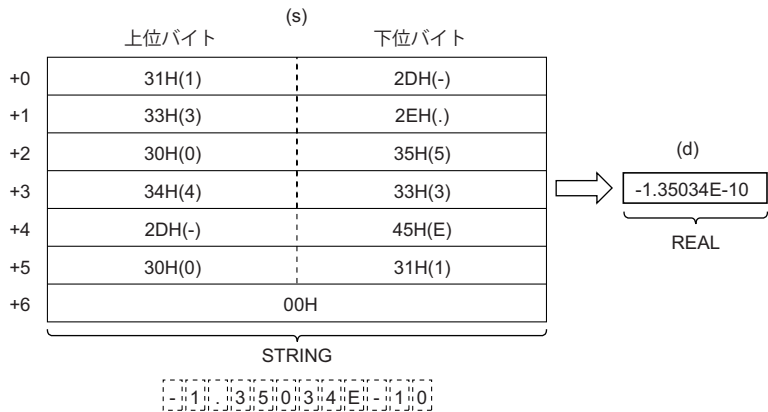
- (s)に入力されたSTRING型(小数点型式/指数型式)のデータを， REAL型のデータに変換して(d)から出力します。



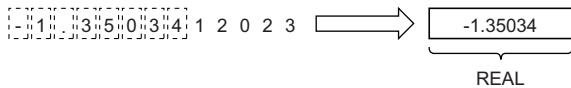
- STRING型データは、小数点型式、指数型式どちらも変換できます。
- 小数点型式の場合



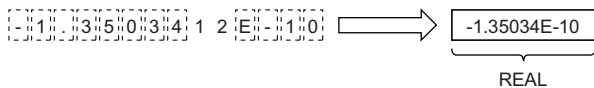
- 指数型式の場合



- STRING型データは、下記のように符号、小数点、指数部を除いて6桁が有効となり、7桁目以降はカットされて変換されます。
- 小数点型式の場合

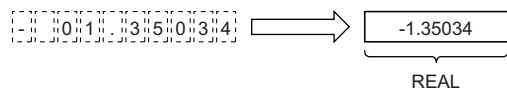


- 指数型式の場合

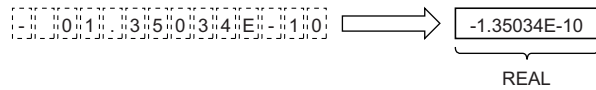


- 小数点型式で符号に2BH(+)を指定するか、符号を省略すると正の値として変換します。また、符号に2DH(-)を指定すると負の値として変換します。
- 指数型式で指数部の符号に2BH(+)を指定するか、符号を省略すると正の値として変換します。指数部の符号に2DH(-)を指定すると負の値として変換します。

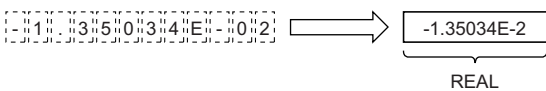
- STRING型データで、最初の0以外の数値の間に20H(スペース)または30H(0)が存在する場合は、下記のように20H、30Hを無視して変換します。
- 小数点型式の場合



- 指数型式の場合



- STRING型データ(指数型式)で「E」と数値の間に30H(0)が存在する場合は下記のように30Hを無視して変換します。



- 文字列中に20H(スペース)が含まれていた場合は、20Hを無視して変換します。
- STRING型データは、最大24文字まで入力できます。文字列中の20H(スペース)、30H(0)も1文字としてカウントします。
- (s)への入力値は、STRING型のデータ値で下記に示す範囲内です。
- アスキーコードで30H~39H、45H、2BH、2DH、2EH、20H、00Hの範囲

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

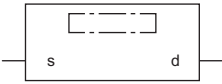
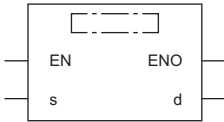
エラーコード (SD0)	内容
2820H	(s)から該当デバイス範囲内に00Hがないとき。
3281H	(s)に変換できない不正なデータを設定したとき。 • 整数部、小数部に、30H(0)~39H(9)以外の文字がある • 指定した文字列の中に2EH(.)が2つ以上ある • 指定した文字列の指数部に、45H(E)、65H(e)、2BH(+), 2DH(-)以外の文字がある • 指定した文字列の中に45H(E)、65H(e)の指数部が複数ある • 指定した文字列の中に、指数部の数値を3桁以上記載している • 指定した文字列の中に2BH(+), 2DH(-)の指数部の符号が複数ある • 指定した文字列の中に2BH(+), 2DH(-)の符号が小数点形式の場合:整数部、指数形式の場合:仮数部に複数ある • (s)以降の文字数が0、または24文字を超えているとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき) $ (d) < 2^{128}$

27.60 STRING型→TIME型変換

27

STRING_TO_TIME(_E)

STRING型データをTIME型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] d:=STRING_TO_TIME(s); [EN/ENO付き] d:=STRING_TO_TIME_E(EN,ENO,s);
		

設定データ

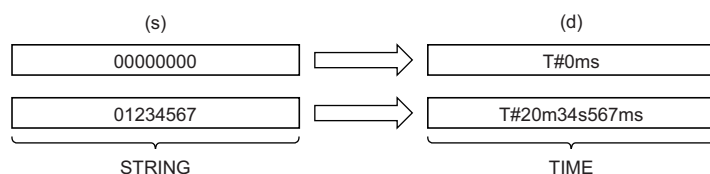
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	STRING(11)
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	TIME

機能

■演算処理

- (s)に入力されたSTRING型のデータを, TIME型のデータに変換して(d)から出力します。



- (s)への入力値は, STRING型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

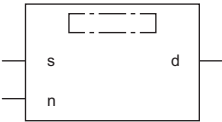
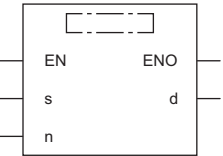
エラーコード (SD0)	内容
3281H	入力の各桁のアスキーコードが、30H~39H, 20H, 00H以外するとき。
	入力のアスキーデータが下記範囲外するとき。 -2147483648~2147483647

27.61 ビット配列→INT型変換

27

BITARR_TO_INT(_E)

ビット配列から指定のビット数をINT型データへ変換します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=BITARR_TO_INT(s,n); [EN/ENO付き] d:=BITARR_TO_INT_E(EN,ENO,s,n);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(BitArr)	入力(要素指定は変数も可)	入力変数	BOOL配列要素
n	4, 8, 12, 16の定数のみ指定可能	入力変数	INT
ENO	出力状態(TRUE: 正常実行, FALSE: 異常または停止)	出力変数	BOOL
d	出力	出力変数	ANY16

機能

■演算処理

- ・ (s)に入力されたビット配列要素を先頭として(n)で指定されたビット数分をANY16型のデータに変換して(d)から出力します。
- ・ 指定されたビット数以上の出力ビットには, 0が設定されます。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

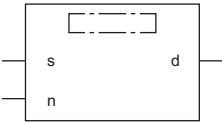
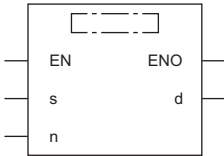
エラー

演算エラーはありません。

27.62 ビット配列→DINT型変換

BITARR_TO_DINT(_E)

ビット配列から指定のビット数をDINT型データへ変換します。

ラダー， FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=BITARR_TO_DINT(s,n) [EN/ENO付き] d:=BITARR_TO_DINT_E(EN,ENO,s,n);
		

設定データ

■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(BitArr)	入力(要素指定は変数も可)	入力変数	BOOL配列要素
n	4, 8, 12, 16, 20, 24, 28, 32の定数のみ指定可能	入力変数	INT
ENO	出力状態(TRUE: 正常実行, FALSE: 異常または停止)	出力変数	BOOL
d	出力	出力変数	ANY32

機能

■演算処理

- (s)に入力されたビット配列要素を先頭として(n)で指定されたビット数分をANY32型のデータに変換して(d)から出力します。
- 指定されたビット数以上の出力ビットには， 0が設定されます。

■演算結果

1. EN/ENOなしファンクション
演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション
実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合， (d)から出力されるデータは不定値となります。その場合は， (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

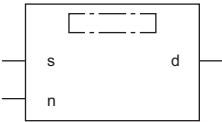
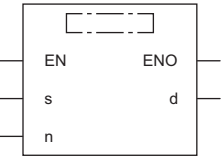
エラー

演算エラーはありません。

27.63 INT型→ビット配列変換

INT_TO_BITARR(_E)

INT型データの低位nビットをビット配列に出力します。

ラダー, FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=INT_TO_BITARR(s,n); [EN/ENO付き] d:=INT_TO_BITARR_E(EN,ENO,s,n);
		

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s	入力	入力変数	ANY16
n	4, 8, 12, 16の定数のみ指定可能	入力変数	INT
ENO	出力状態(TRUE: 正常実行, FALSE: 異常または停止)	出力変数	BOOL
d	出力(要素指定は変数も可)	出力変数	BOOL配列要素

機能

■演算処理

- ・ (s)に指定されたANY16型の低位(n)ビットを(d)に出力します。
- ・ 指定されたビット数以上の出力ビットは変更しません。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

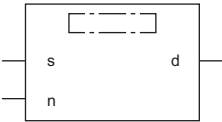
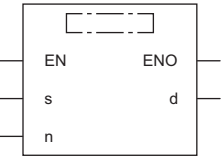
エラー

演算エラーはありません。

27.64 DINT型→ビット配列変換

DINT_TO_BITARR(_E)

DINT型データの低位nビットをビット配列に出力します。

ラダー， FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=DINT_TO_BITARR(s,n);
	[EN/ENO付き]	[EN/ENO付き] d:=DINT_TO_BITARR_E(EN,ENO,s,n);
		

設定データ

■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行， FALSE: 停止)	入力変数	BOOL
s	入力	入力変数	ANY32
n	4, 8, 12, 16, 20, 24, 28, 32の定数のみ指定可能	入力変数	BOOL
ENO	出力状態(TRUE: 正常実行， FALSE: 異常または停止)	出力変数	BOOL
d	出力(要素指定は変数も可)	出力変数	BOOL配列要素

機能

■演算処理

- ・ (s)に指定されたANY32型の低位(n)ビットを(d)に出力します。
- ・ 指定されたビット数以上の出力ビットは変更しません。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合，(d)から出力されるデータは不定値となります。その場合は，(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

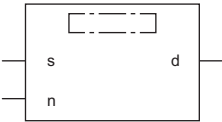
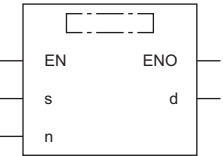
演算エラーはありません。

27.65 ビット配列のコピー

27

CPY_BITARR(_E)

ビット配列を指定ビットサイズ分コピーします。

ラダー, FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] d:=CPY_BITARR(s,n); [EN/ENO付き] d:=CPY_BITARR_E(EN,ENO,s,n);
		

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(BitArrIn)	入力	入力変数	BOOL配列要素
n	4, 8, 12, 16, 20, 24, 28, 32の定数のみ指定可能	入力変数	INT
ENO	出力状態(TRUE: 正常実行, FALSE: 異常または停止)	出力変数	BOOL
d	出力	出力変数	BOOL配列要素

機能

■演算処理

- ・ (s)に指定されたビット配列の(n)ビットを(d)に出力します。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

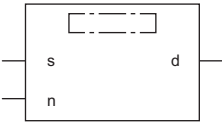
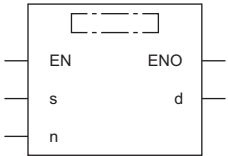
エラー

演算エラーはありません。

27.66 ワードラベルの指定ビット読出し

GET_BIT_OF_INT(_E)

ワードラベルの指定ビットを読み出します。

ラダー *1		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] d:=GET_BIT_OF_INT(s,n); [EN/ENO付き] d:=GET_BIT_OF_INT_E(EN,ENO,s,n);
		

*1 FBD/LDでは対応していません。FBD/LDの場合は、ラベルのビット指定を使用してください。

設定データ

■内容、種別、データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s	入力	入力変数	INT*1
n	0から15の定数のみ指定可能	入力変数	INT
ENO	出力状態(TRUE: 正常実行, FALSE: 異常または停止)	出力変数	BOOL
d	出力	出力変数	BOOL

*1 桁指定したラベルおよびデバイスは使用できません。

機能

■演算処理

- (s)の(n)ビット目を出力します。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE*1	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

演算エラーはありません。

ラベルのビット指定を使用すると、GET_BIT_OF_INTと同じ動作のプログラムを簡潔に作成できます。
下記の例はGET_BIT_OF_INTを使用した場合と同様に、g_int1のビット5(b5)の値をg_bool1に読み出します。
ラダーの場合



STの場合

`g_bool1 := g_int1.5;`

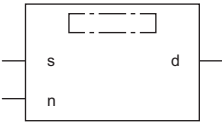
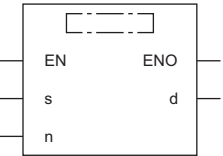
FBD/LDの場合



27.67 ワードラベルの指定ビット書込み

SET_BIT_OF_INT(_E)

ワードラベルの指定ビットに書き込みます。

ラダー *1		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] d:=SET_BIT_OF_INT(s,n); [EN/ENO付き] d:=SET_BIT_OF_INT_E(EN,ENO,s,n);
		

*1 FBD/LDでは対応していません。FBD/LDの場合は、ラベルのビット指定を使用してください。

設定データ

■内容、種別、データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s	入力	入力変数	BOOL
n	0から15の定数のみ指定可能	入力変数	INT
ENO	出力状態(TRUE: 正常実行, FALSE: 異常または停止)	出力変数	BOOL
d	入出力	入出力変数	INT*1

*1 桁指定したラベルおよびデバイスは使用できません。

機能

■演算処理

- (d)の(n)ビット目に(s)で指定されたBOOL値を書き込みます。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE*1	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

注意事項

SET_BIT_OF_INT(_E)をSTで使用する場合、戻り値を変数に代入するプログラムを作成してください。

```
//SET_BIT_OF_INTの戻り値は、変数に代入してから使用します。
g_int1 := SET_BIT_OF_INT(TRUE, 0);
g_bool1 := GET_BIT_OF_INT(g_int1, 0);
```

SET_BIT_OF_INT(_E)の戻り値は入出力を兼ねているため、他の命令、ファンクション、ファンクションブロックの入力引数に直接代入しないで下さい。

//下記のプログラムでは、GET_BIT_OF_INTの第一引数の値は不定値になります。
g_bool1 := GET_BIT_OF_INT(SET_BIT_OF_INT(TRUE, 0), 0);

エラー

演算エラーはありません。

Point

ラベルのビット指定を使用すると、SET_BIT_OF_INTと同じ動作のプログラムを簡潔に作成できます。

下記の例はSET_BIT_OF_INTを使用した場合と同様に、g_int1のビット5(b5)をg_bool1の値に変更します。

ラダーの場合



STの場合

```
g_int1.5 := g_bool1;
```

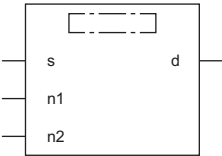
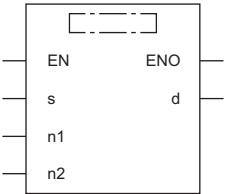
FBD/LDの場合



27.68 ワードラベルの指定ビットコピー

CPY_BIT_OF_INT(_E)

ワードラベルの指定ビットを別のワードラベルの指定ビットにコピーします。

ラダー *1		ST
[EN/ENOなし]		[EN/ENOなし] d:=CPY_BIT_OF_INT(s,n1,n2); [EN/ENO付き] d:=CPY_BIT_OF_INT_E(EN,ENO,s,n1,n2);
		

*1 FBD/LDでは対応していません。FBD/LDの場合は、ラベルのビット指定を使用してください。

設定データ

■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s	入力	入力変数	INT*1
n1	入力変数のビット指定(0から15の定数のみ指定可能)	入力変数	INT
n2	出力変数のビット指定(0から15の定数のみ指定可能)	入力変数	INT
ENO	出力状態(TRUE: 正常実行, FALSE: 異常または停止)	出力変数	BOOL
d	入出力	入出力変数	INT*1

*1 桁指定したラベルおよびデバイスは使用できません。

機能

■演算処理

- (d)の(n2)ビット目に(s)で指定されたワードの(n1)ビット目の値をコピーします。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE*1	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

注意事項

CPY_BIT_OF_INT(_E)をSTで使用する場合、戻り値を変数に代入するプログラムを作成してください。

```
//CPY_BIT_OF_INTの戻り値は、変数に代入してから使用します。
g_int2 := CPY_BIT_OF_INT(g_int1,5,3);
g_bool1 := GET_BIT_OF_INT(g_int2,3);
```

CPY_BIT_OF_INT(_E)の戻り値は入出力を兼ねているため、他の命令、ファンクション、ファンクションブロックの入力引数に直接代入しないで下さい。

//下記のプログラムでは、GET_BIT_OF_INTの第一引数の値は不定値になります。
g_bool1 := GET_BIT_OF_INT(CPY_BIT_OF_INT(g_int1,5,3), 3);

エラー

演算エラーはありません。

Point

ラベルのビット指定を使用すると、CPY_BIT_OF_INTと同じ動作のプログラムを簡潔に作成できます。

下記の例はCPY_BIT_OF_INTを使用した場合と同様に、g_int2のビット3(b3)をg_int1のビット5(b5)の値に変更します。

ラダーの場合



STの場合

```
g_int2.3 := g_int1.5;
```

FBD/LDの場合



27.69 先頭データの取得

GET_BOOL_ADDR, GET_INT_ADDR, GET_WORD_ADDR

指定した配列の先頭データをBOOL型, INT型, またはWORD型データとして出力します。

ラダー, FBD/LD	ST
	d:=GET_BOOL_ADDR(s) d:=GET_INT_ADDR(s); d:=GET_WORD_ADDR(s);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
s	入力	入力変数	ANY
d	出力	出力変数	BOOL/INT/WORD

機能

■演算処理

- GET_BOOL_ADDRは(s)で指定した配列から, 先頭データをBOOL型データとして出力します。
- GET_INT_ADDRは(s)で指定した配列から, 先頭データをINT型データとして出力します。ビット長が32ビット以上のデータ型を指定した場合, 下位16ビットを取得することができます。
- GET_WORD_ADDRは(s)で指定した配列から, 先頭データをWORD型データとして出力します。ビット長が32ビット以上のデータ型を指定した場合, 下位16ビットを取得することができます。

汎用ファンクション	入力データ型	出力データ型
GET_BOOL_ADDR	BOOL ARRAY OF BOOL	BOOL
GET_INT_ADDR	INT	INT
GET_WORD_ADDR	DINT WORD REAL TIME STRING ARRAY OF INT ARRAY OF DINT ARRAY OF WORD ARRAY OF DWORD ARRAY OF REAL ARRAY OF TIME	WORD

■演算結果

演算処理を実行します。(d)からは演算出力値が出力されます。

エラー

演算エラーはありません。

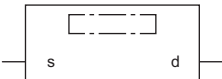
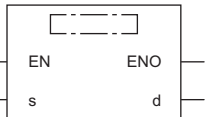
28 単数値変数ファンクション

28

28.1 絶対値

ABS(_E)

入力値の絶対値を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=ABS(s); [EN/ENO付き] d:=ABS_E(EN,ENO,s);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_NUM
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_NUM

機能

■演算処理

- ・ (s)に入力されたINT型/DINT型/REAL/LREAL型データの絶対値を, (d)から(s)と同一のデータ型で出力します。
- ・ 入力値をA, 演算出力値をBとすると, 下記ようになります。

$$B=|A|$$

- ・ (s)への入力値は, INT型/DINT型/REAL型/LREAL型のデータ値です。
- ・ (s)のデータ型がINT型で-32768を入力した場合, (d)から-32768が出力されます。
- ・ (s)のデータ型がDINT型で-2147483648を入力した場合, (d)から-2147483648が出力されます。(演算エラーにはなりません。またABS_Eのときは, 出力変数ENOからTRUEが出力されます。)

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

- (s)がREAL型の場合

エラーコード (SD0)	内容
3282H	(d)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。

- (s)がLREAL型の場合

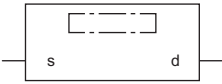
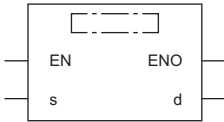
エラーコード (SD0)	内容
3282H	(d)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。

28.2 平方根

28

SQRT(_E)

入力値の平方根を演算します。

ラダー, FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] d:=SQRT(s); [EN/ENO付き] d:=SQRT_E(EN,ENO,s);
		

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_REAL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_REAL

機能

■演算処理

- (s)に入力されたREAL型/LREAL型データの平方根を演算し, 演算結果を(d)に格納します。
- 入力値をA, 演算出力値をBとすると, 下記ようになります。

$$B = \sqrt{A}$$

- (s)への入力値は, REAL型/LREAL型のデータ値で正の数の範囲内です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

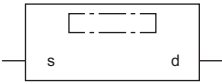
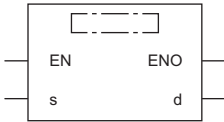
エラー

エラーコード (SD0)	内容
3285H	入力された値が負の数のとき。

28.3 自然対数演算

LN(_E)

入力値の自然対数演算結果を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=LN(s); [EN/ENO付き] d:=LN_E(EN,ENO,s);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_REAL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_REAL

機能

■演算処理

- ・ (s)に入力されたREAL型/LREAL型データの自然対数「e」を底としたときの対数を演算して, (d)から出力します。
- ・ 入力値をA, 演算出力値をBとすると下記になります。

$$B=\log_e A$$

- ・ 自然対数演算では, 底「e」を2.71828として演算します。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

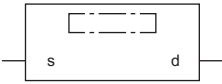
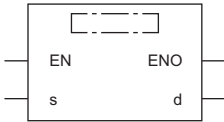
エラーコード (SD0)	内容
3285H	入力された値が0または負の数のとき。

28.4 常用対数演算

28

LOG(_E)

入力値の常用対数(10を底とした対数)の演算結果を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] $d := \text{LOG}(s);$ [EN/ENO付き] $d := \text{LOG\_E}(\text{EN}, \text{ENO}, s);$
		

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_REAL
ENO	出力条件(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_REAL

機能

■演算処理

- ・(s)に入力されたREAL型/LREAL型データの10を底としたときの対数を演算して、(d)から出力します。
- ・入力値をA, 演算出力値をBとすると下記になります。

$$B = \log_{10} A$$

- ・(s)への入力値は, REAL型/LREAL型のデータ値です。
- ・(s)で指定する値は, 正の数のみ設定できます。(負の数は演算できません。)
- ・演算結果が-0またはアンダーフローが発生したときは, 演算結果を0として出力します。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

• (s)がREAL型の場合

エラーコード (SD0)	内容
3282H	(s)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3285H	(s)に指定可能な範囲外のデータを設定したとき。 - 指定した値が負の数 - 指定した値が「0」

• (s)がLREAL型の場合

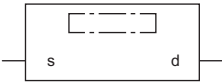
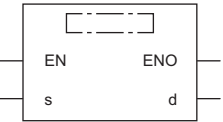
エラーコード (SD0)	内容
3282H	(s)に指定した値が-0, 非正規化数, 非数, $\pm\infty$ のとき。
3285H	(s)に指定可能な範囲外のデータを設定したとき。 - 指定した値が負の数 - 指定した値が「0」

28.5 指数演算

28

EXP(_E)

入力値の指数演算結果を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=EXP(s); [EN/ENO付き] d:=EXP_E(EN,ENO,s);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_REAL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_REAL

機能

■演算処理

- ・ (s)に入力されたREAL型/LREAL型データの指数を演算して, (d)から出力します。
- ・ 入力値をA, 演算出力値をBとすると下記になります。

$$B=e^A$$

- ・ 指数演算では, 底「e」を2.71828として演算します。
- ・ (s)への入力値は, REAL型/LREAL型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

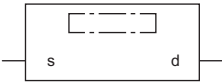
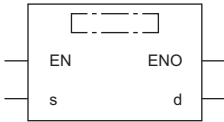
エラー

エラーコード (SD0)	内容
3282H	(s)で指定したデータが-0, 非正規化数, 非数, ±∞のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき) $ d < 2^{128}$

28.6 SIN演算

SIN(_E)

入力値のSIN(正弦)値を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=SIN(s); [EN/ENO付き] d:=SIN_E(EN,ENO,s);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_REAL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_REAL

機能

■演算処理

- (s)に入力されたREAL型データ(角度)のSIN(正弦)値を演算して, (d)から出力します。
- 入力値をA, 演算出力値をBとすると下記になります。

$$B=\sin A$$

- (s)への入力値(角度)は, REAL型のデータ値です。入力値はラジアン単位(角度 $\times\pi/180$)で入力してください。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

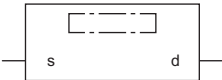
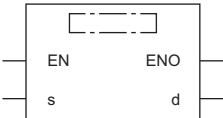
エラーコード (SD0)	内容
3282H	(s)で指定したデータが-0のとき。

28.7 COS演算

28

COS(_E)

入力値のCOS(余弦)値を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=COS(s); [EN/ENO付き] d:=COS_E(EN,ENO,s);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_REAL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_REAL

機能

■演算処理

- (s)に入力されたREAL型データ(角度)のCOS(余弦)値を演算して, (d)から出力します。
- 入力値をA, 演算出力値をBとすると下記になります。

$B = \cos A$

- (s)への入力値(角度)は, REAL型のデータ値です。入力値は, ラジアン単位(角度 $\times \pi/180$)で入力してください。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

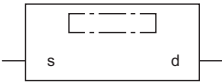
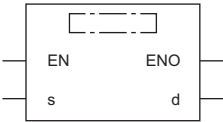
エラー

エラーコード (SD0)	内容
3282H	(s)で指定したデータが-0のとき。

28.8 TAN演算

TAN(_E)

入力値のTAN(正接)値を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=TAN(s); [EN/ENO付き] d:=TAN_E(EN,ENO,s);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_REAL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_REAL

機能

■演算処理

- ・ (s)に入力されたREAL型データ(角度)のTAN(正接)値を演算して, (d)から出力します。
- ・ 入力値をA, 演算出力値をBとすると下記になります。

B=TAN A

- ・ 入力された値が, $\pi/2$ ラジアン, $(3/2)\pi$ ラジアンの場合でも, ラジアン値に演算誤差が生じるためエラーとなりませんので注意してください。
- ・ (s)への入力値(角度)は, REAL型のデータ値です。入力値は, ラジアン単位(角度 $\times\pi/180$)で入力してください。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

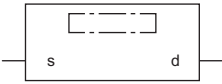
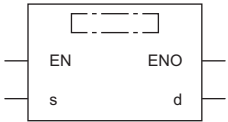
エラーコード (SD0)	内容
3282H	(s)で指定したデータが-0のとき。

28.9 SIN⁻¹演算

28

ASIN(_E)

入力値のSIN⁻¹(逆正弦)値を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=ASIN(s); [EN/ENO付き] d:=ASIN_E(EN,ENO,s);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_REAL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_REAL

機能

■演算処理

- ・ (s)に入力されたREAL型データのSIN⁻¹(逆正弦)値を演算して, (d)から出力します。
- ・ 入力値をA, 演算出力値をBとすると下記になります。

$$B = \text{SIN}^{-1} A$$

- ・ (s)への入力値は, REAL型のデータ値で下記に示す範囲内です。

ASIN(_E): -1.0~1.0

- ・ (d)からの出力値(角度)は, ラジアン単位(角度× $\pi/180$)です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

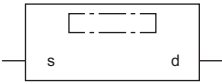
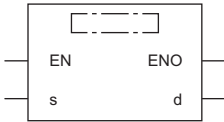
エラー

エラーコード (SD0)	内容
3282H	(s)で指定したデータが-0のとき。
3285H	ASIN(_E)で入力された値が-1.0~1.0以外のとき。

28.10 cos⁻¹演算

ACOS(_E)

入力値のCOS⁻¹(逆余弦)値を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=ACOS(s); [EN/ENO付き] d:=ACOS_E(EN,ENO,s);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_REAL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_REAL

機能

■演算処理

- ・ (s)に入力されたREAL型データのCOS⁻¹(逆余弦)値を演算して, (d)から出力します。
- ・ 入力値をA, 演算出力値をBとすると下記になります。

$$B = \cos^{-1} A$$

- ・ (s)への入力値は, REAL型のデータ値で下記に示す範囲内です。

ACOS(_E): -1.0~1.0

- ・ (d)からの出力値(角度)は, ラジアン単位(角度×π/180)です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

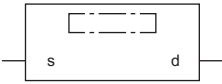
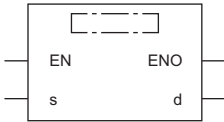
エラーコード (SD0)	内容
3282H	(s)で指定したデータが-0のとき。
3285H	ACOS(_E)で入力された値が-1.0~1.0以外のとき。

28.11 TAN⁻¹演算

28

ATAN(_E)

入力値のTAN⁻¹(逆正接)値を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=ATAN(s); [EN/ENO付き] d:=ATAN_E(EN,ENO,s);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_REAL
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_REAL

機能

■演算処理

- ・(s)に入力されたREAL型データのTAN⁻¹(逆正接)値を演算して、(d)から出力します。
- ・入力値をA, 演算出力値をBとすると下記になります。

$$B = \text{TAN}^{-1} A$$

- ・(s)への入力値は、REAL型のデータ値で下記に示す範囲内です。

$$\text{ATAN\_E}: \pm 1.17549 \times 10^{-38} \sim \pm 3.40282 \times 10^{38}$$

- ・(d)からの出力値(角度)は、ラジアン単位(角度 $\times \pi / 180$)です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

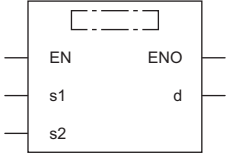
エラーコード (SD0)	内容
3282H	(s)で指定したデータが-0のとき。

29 算術演算ファンクション

29.1 加算

ADD(_E)

入力値の和((s1)+(s2)+...+(s28))を出力します。

ラダー, FBD/LD *1		ST*1
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=ADD(s1,s2); [EN/ENO付き] d:=ADD_E(EN,ENO,s1,s2);

*1 入力変数sは, 2~28の範囲で変更できます。

設定データ

■内容, 種別, データ型

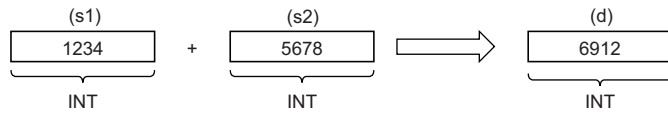
引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1)~s28(IN28)	入力	入力変数	ANY_NUM
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_NUM

■演算処理

- (s1)~(s28)に入力されたINT型/DINT型/WORD型/DWORD型/REAL型/LREAL型データの加算((s1)+(s2)+...+(s28))を行い、演算結果を(d)から(s)と同一のデータ型で出力します。

例

データ型がINT型の場合



- (s1)~(s28)への入力値は、INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型のデータ値です。
- 演算結果にアンダーフロー / オーバーフローが生じた場合は、(d)から下記のように出力されます。

データ型	内容
INT型	アンダーフローおよびオーバーフローが生じた場合でも、演算エラーにはなりません。また、ADD_Eの場合、ENOからTRUEが出力されます。 [例1] $32767+2=-32767$ $(7FFFH)+(0002H)=(8001H)$ 最上位ビットが1になるため、負の値になります。 [例2] $-32767+(-2)=32766$ $(8000H)+(FFFEH)=(7FFE H)$ 最上位ビットが0になるため、正の値になります。
DINT型	アンダーフローおよびオーバーフローが生じた場合でも、演算エラーにはなりません。また、ADD_Eの場合、ENOからTRUEが出力されます。 [例1] $2147483647+2=-2147483647$ $(7FFFFFFFH)+(00000002H)=(80000001H)$ 最上位ビットが1になるため、負の値になります。 [例2] $-2147483648+(-2)=2147483646$ $(80000000H)+(FFFEH)=(7FFFFFFEH)$ 最上位ビットが0になるため、正の値になります。
WORD型	オーバーフローが生じた場合でも、演算エラーにはなりません。また、ADD_Eの場合、ENOからTRUEが出力されます。 [例] $65535+1=0$ $(FFFFH)+(0001H)=(0000H)$
DWORD型	オーバーフローが生じた場合でも、演算エラーにはなりません。また、ADD_Eの場合、ENOからTRUEが出力されます。 [例] $4294967295+1=0$ $(FFFFFFFFH)+(00000001H)=(00000000H)$
REAL型	演算エラーとなり不定値を出力します。
LREAL型	

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

• (s1)~(s28)がREAL型の場合

エラーコード (SD0)	内容
3282H	(s1)~(s28)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
	(d)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき) $ d < 2^{128}$

• (s1)~(s28)がLREAL型の場合

エラーコード (SD0)	内容
3282H	(s1)~(s28)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
	(d)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき) $ d < 2^{1024}$

29.2 乗算

29

MUL(_E)

入力値の積((s1)×(s2)×…×(s28))を出力します。

ラダー, FBD/LD*1	ST*1
[EN/ENOなし] 	[EN/ENOなし] d:=MUL(s1,s2); [EN/ENO付き] d:=MUL_E(EN,ENO,s1,s2);

*1 入力変数sは、2~28の範囲で変更できます。

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1)~s28(IN28)	入力	入力変数	ANY_NUM
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_NUM

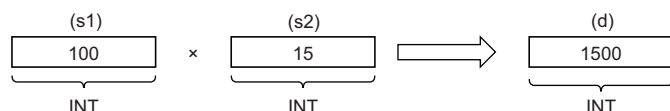
機能

■演算処理

- (s1)~(s28)に入力されたINT型/DINT型/WORD型/DWORD型/REAL型/LREAL型データの乗算((s1)×(s2)×…×(s28))を行い、演算結果を(d)から(s)と同一のデータ型で出力します。

例

データ型がINT型の場合



- (s1)~(s28)への入力値は、INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型のデータ値です。
- 演算結果にアンダーフロー / オーバーフローが生じた場合は、(d)から下記のように出力されます。

データ型	内容
INT型 WORD型	• アンダーフローおよびオーバーフローが生じた場合でも、演算エラーにはなりません。また、MUL_Eの場合、ENOからTRUEが出力されます。 • 演算結果がINT型/WORD型データの範囲を超えている場合でも、INT型/WORD型のデータを出力します。(演算結果はDINT型/DWORD型になりますが、出力は上位16ビットを削除したINT型/WORD型のデータで出力されます。) • 演算結果がINT型/WORD型データの範囲を超えている場合は、INT_TO_DINT/WORD_TO_DWORDにより入力値をDINT型/DWORD型データへ変換後、演算を行ってください。
DINT型 DWORD型	• アンダーフローおよびオーバーフローが生じた場合でも、演算エラーにはなりません。また、MUL_Eの場合、ENOからTRUEが出力されます。 • 演算結果がDINT型/DWORD型データの範囲を超えている場合でも、DINT型/DWORD型のデータを出力します。(演算結果は64ビットデータになりますが、出力は上位32ビットを削除したDINT型/DWORD型のデータで出力されます。) • 演算結果がDINT型/DWORD型データの範囲を超えている場合は、DINT_TO_REALにより入力値をREAL型データへ変換後、演算を行ってください。
REAL型	演算エラーとなり不定値を出力します。
LREAL型	

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

Point

演算結果がデータ型の範囲を超える場合は、入力値のデータ型を変換してから演算を実行してください。

エラー

• (s1)~(s28)がREAL型の場合

エラーコード (SD0)	内容
3282H	(s1)~(s28)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき) $ d < 2^{128}$

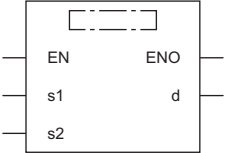
• (s1)~(s28)がLREAL型の場合

エラーコード (SD0)	内容
3282H	(s1)~(s28)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき) $ d < 2^{1024}$

29.3 減算

SUB(_E)

入力値の差((s1)-(s2))を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] d:=SUB(s1,s2); [EN/ENO付き] d:=SUB_E(EN,ENO,s1,s2);
		

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1), s2(IN2)	入力	入力変数	ANY_NUM
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_NUM

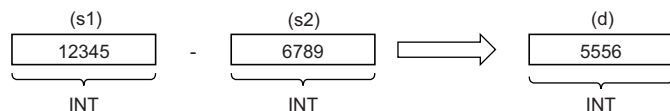
機能

■演算処理

- (s1), (s2)に入力されたINT型/DINT型/WORD型/DWORD型/REAL型/LREAL型データの減算((s1)-(s2))を行い、演算結果を(d)から(s)と同一のデータ型で出力します。

例

データ型がINT型の場合



- (s1), (s2)への入力値は、INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型のデータ値です。
- 演算結果にアンダーフロー / オーバーフローが生じた場合は、(d)から下記のように出力されます。

データ型	内容
INT型	アンダーフローおよびオーバーフローが生じた場合でも、演算エラーにはなりません。また、SUB_Eの場合、ENOからTRUEが出力されます。 [例1] 32767-(-2)=-32767 (7FFFH)-(FFFEH)=(8001H) 最上位ビットが1になるため、負の値になります。 [例2] -32767-2=32766 (8000H)-(0002H)=(7FFEH) 最上位ビットが0になるため、正の値になります。
DINT型	アンダーフローおよびオーバーフローが生じた場合でも、演算エラーにはなりません。また、SUB_Eの場合、ENOからTRUEが出力されます。 [例1] 2147483647-(-2)=-2147483647 (7FFFFFFFH)-(0000FFFEH)=(80000001H) 最上位ビットが1になるため、負の値になります。 [例2] -2147483648-2=2147483646 (80000000H)-(00000002H)=(7FFFFFFEH) 最上位ビットが0になるため、正の値になります。
WORD型	アンダーフローが生じた場合でも、演算エラーにはなりません。また、ADD_Eの場合、ENOからTRUEが出力されます。 [例] 0-1=65535 (0000H)-(0001H)=(FFFFH)
DWORD型	アンダーフローが生じた場合でも、演算エラーにはなりません。また、ADD_Eの場合、ENOからTRUEが出力されます。 [例] 0-1=4294967295 (00000000H)-(00000001H)=(FFFFFFFFH)
REAL型	演算エラーとなり不定値を出力します。
LREAL型	

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

• (s1), (s2)がREAL型の場合

エラーコード (SD0)	内容
3282H	(s1)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
	(s2)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
	(d)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき) $ (d) < 2^{128}$

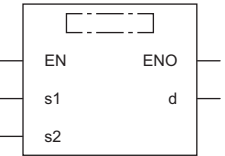
• (s1), (s2)がLREAL型の場合

エラーコード (SD0)	内容
3282H	(s1)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
	(s2)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
	(d)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき) $ (d) < 2^{1024}$

29.4 除算

DIV(_E)

入力値の商((s1)÷(s2))を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] d:=DIV(s1,s2); [EN/ENO付き] d:=DIV_E(EN,ENO,s1,s2);
		

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1)	被除数	入力変数	ANY_NUM
s2(IN2)	除数	入力変数	ANY_NUM
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_NUM

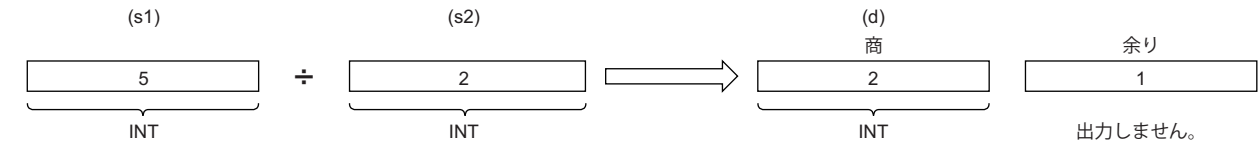
機能

■演算処理

- ・ (s1), (s2)に入力されたINT型/DINT型/WORD型/DWORD型/REAL型/LREAL型データの除算((s1)÷(s2))を行い, 演算結果の商を(d)から(s)と同一のデータ型で出力します。

例

データ型がINT型の場合



- ・ (s1), (s2)への入力値は, INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型のデータ値です。(ただし, (s2)へ入力される値は, 0以外です。)

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

- (s1), (s2)がINT型/WORD型の場合

エラーコード (SD0)	内容
3280H	(s2)で指定した値(除数)が0のとき。

- (s1), (s2)がDINT型/DWORD型の場合

エラーコード (SD0)	内容
3280H	(s2)で指定した値(除数)が0のとき。

- (s1), (s2)がREAL型の場合

エラーコード (SD0)	内容
3280H	(s2)で指定した値(除数)が0のとき。
3282H	(s1)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (s2)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき) $ d < 2^{128}$

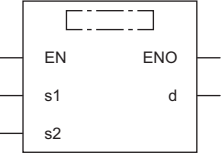
- (s1), (s2)がLREAL型の場合

エラーコード (SD0)	内容
3280H	(s2)で指定した値(除数)が0のとき。
3282H	(s1)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (s2)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3283H	(d)が下記範囲を超えるととき。(オーバーフローが発生したとき) $ d < 2^{1024}$

29.5 剰余

MOD(_E)

入力値の余り((s1)÷(s2))を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] 演算子として記述します。(93ページ 演算子) [EN/ENO付き] d:=MOD_E(EN,ENO,s1,s2);
		

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1)	被除数	入力変数	ANY_INT
s2(IN2)	除数	入力変数	ANY_INT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_INT

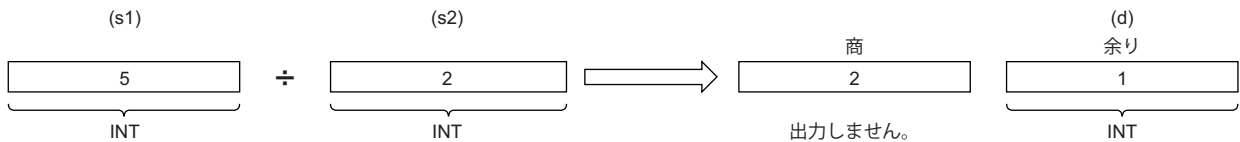
機能

■演算処理

- ・ (s1), (s2)に入力されたINT型/DINT型/WORD型/DWORD型データの除算((s1)÷(s2))を行い, 演算結果の余りを(d)から(s)と同一のデータ型で出力します。

例

データ型がINT型の場合



- ・ (s1), (s2)への入力値は, INT型/DINT型/WORD型/DWORD型のデータ値です。(ただし, (s2)へ入力される値は, 0以外です。)

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

- (s1), (s2)がINT型/WORD型の場合

エラーコード (SD0)	内容
3280H	(s2)で指定した値(除数)が0のとき。

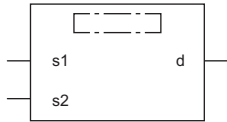
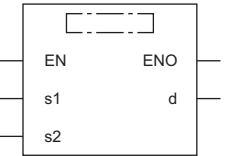
- (s1), (s2)がDINT型/DWORD型の場合

エラーコード (SD0)	内容
3280H	(s2)で指定した値(除数)が0のとき。

29.6 べき乗

EXPT(_E)

入力値のべき乗を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] d:=EXPT(s1,s2); [EN/ENO付き] d:=EXPT_E(EN,ENO,s1,s2);
		

設定データ

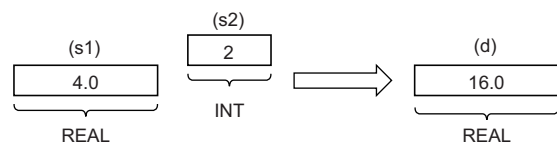
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1)	基数	入力変数	ANY_REAL
s2(IN2)	指数	入力変数	ANY_NUM
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_REAL

機能

■演算処理

- ・(s1)に入力されたREAL型/LREAL型データを(s2)で指定したINT型/DINT型/REAL型/LREAL型でべき乗を行い, 演算結果を(d)から出力します。



■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

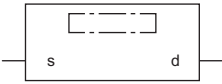
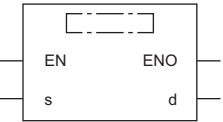
エラー

エラーコード (SD0)	内容
3282H	(s1)または(s2)で指定したデータが-0, 非正規化数, 非数, ±∞のとき。

29.7 代入

MOVE(_E)

入力値の代入を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=MOVE(s); [EN/ENO付き] d:=MOVE_E(EN,ENO,s);

設定データ

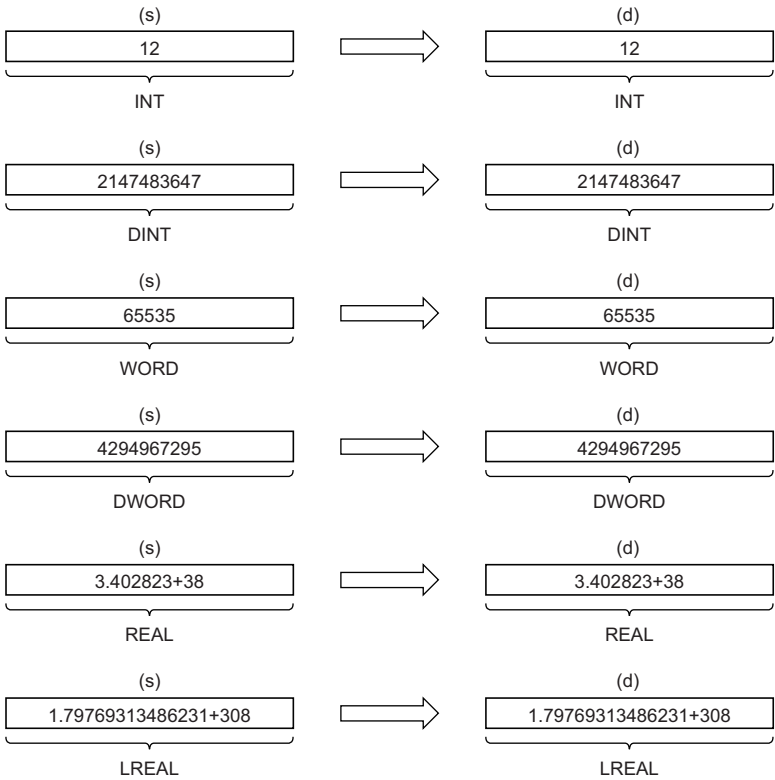
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY

機能

■演算処理

- (s)に指定した変数の値を(d)に指定した変数へ代入します。
- (s), (d)には, BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型/構造体型/配列型が指定することが可能です。また, (s)と(d)は同じデータ型のみ指定可能です。



■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

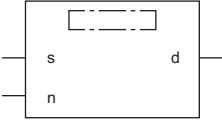
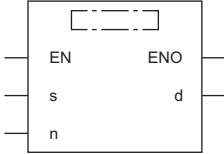
エラーコード (SD0)	内容
2820H	(s)で指定したデバイス番号以降、デバイス/ラベルメモリの各設定エリアの間に00Hが存在しないとき。
3285H	(s)の文字列が16383文字を超えるとき。
3286H	(d)で指定したデバイス番号以降、該当デバイス/ラベルメモリの各設定エリアの最終番号までの点数に、指定した文字列がすべて格納できないとき。

30 ビットシフトファンクション

30.1 nビット左シフト

SHL(_E)

入力値を(n)ビット数分左へシフトして出力します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=SHL(s,n); [EN/ENO付き] d:=SHL_E(EN,ENO,s,n);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_BIT
n(N)	シフトビット数指定	入力変数	ANY_BIT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_BIT

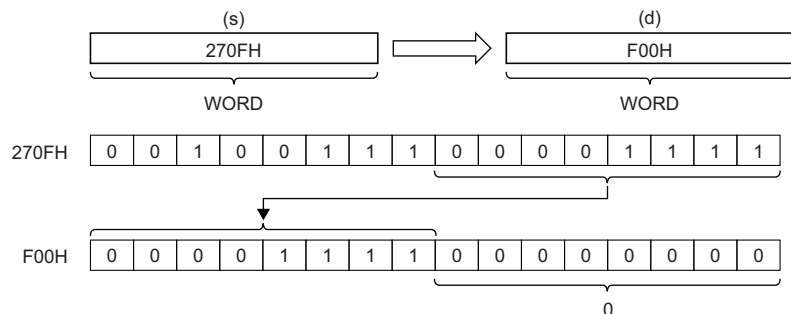
機能

■演算処理

- (s)に入力された16ビットデータ/32ビットデータ(WORD型/DWORD型)を(n)ビット左へシフトして、(d)から(s)と同一のデータ型で出力します。
- 左へシフトする(n)ビット数は、(n)へ入力された値です。

例

(s)のデータが16ビットデータ(WORD型)、(n)への入力値が8の場合



- 最下位から(n)ビット分は0になります。
- (s)への入力値は、16ビットデータ/32ビットデータ(WORD型/DWORD型)のデータです。
- (n)への入力値(シフトビット数指定)は、下記に示す範囲内です。

(s)のデータが16ビットデータ(WORD型)の場合	(s)のデータが32ビットデータ(DWORD型)の場合
(n)の入力値は、0~15の範囲内です。 (n)への入力値の下位4ビット分のデータが使用されます。 [例] 入力値が6の場合: 6 入力値が22の場合: 6	(n)の入力値は、0~31の範囲内です。 (n)への入力値の下位5ビット分のデータが使用されます。 [例] 入力値が6の場合: 6 入力値が22の場合: 22

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

演算エラーはありません。

30.2 nビット右シフト

SHR(_E)

入力値を(n)ビット数分右へシフトして出力します。

ラダー, FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=SHR(s,n);
[EN/ENO付き]		[EN/ENO付き] d:=SHR_E(EN,ENO,s,n);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_BIT
n(N)	シフトビット数指定	入力変数	ANY_BIT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_BIT

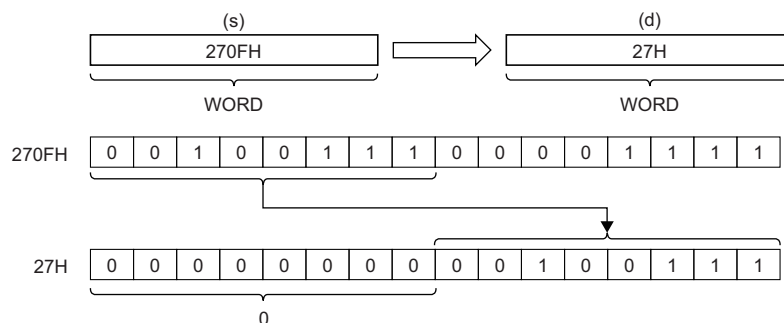
機能

■演算処理

- (s)に入力された16ビットデータ/32ビットデータ(WORD型/DWORD型)を(n)ビット右へシフトして, (d)から(s)と同一のデータ型で出力します。
- 右へシフトする(n)ビット数は, (n)へ入力された値です。

例

(s)のデータが16ビットデータ(WORD型), (n)への入力値が8の場合



- 最上位から(n)ビット分は0になります。
- (s)への入力値は, 16ビットデータ/32ビットデータ(WORD型/DWORD型)のデータです。
- (n)への入力値(シフトビット数指定)は, 下記に示す範囲内です。

(s)のデータが16ビットデータ(WORD型)の場合	(s)のデータが32ビットデータ(DWORD型)の場合
(n)の入力値は, 0~15の範囲内です。 (n)への入力値の下位4ビット分のデータが使用されます。 [例] 入力値が6の場合: 6 入力値が22の場合: 6	(n)の入力値は, 0~31の範囲内です。 (n)への入力値の下位5ビット分のデータが使用されます。 [例] 入力値が6の場合: 6 入力値が22の場合: 22

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件		演算結果	
EN	ENO	(d)	
TRUE(演算実行)	TRUE		演算出力値
FALSE(演算停止)	FALSE*1		不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

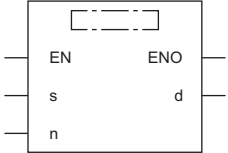
エラー

演算エラーはありません。

30.3 nビット左ローテーション

ROL(_E)

入力値を(n)ビット数分左へ回転して出力します。

ラダー, FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=ROL(s,n); [EN/ENO付き] d:=ROL_E(EN,ENO,s,n);
		

設定データ

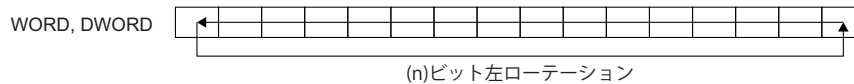
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_BIT
n(N)	シフトビット数指定	入力変数	ANY_BIT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_BIT

機能

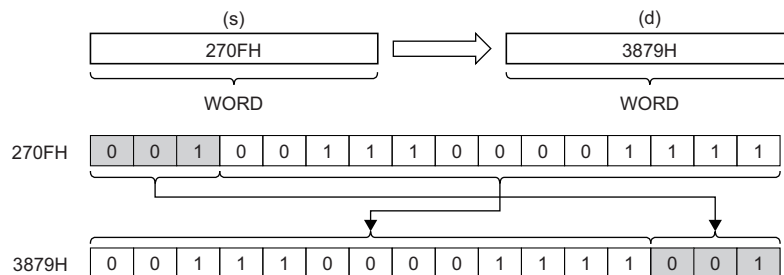
■演算処理

- ・ (s)に入力された16ビットデータ/32ビットデータ(WORD型/DWORD型)を(n)ビット左へ回転(ローテーション)して, (d)から(s)と同一のデータ型で出力します。
- ・ 左へ回転する(n)ビット数は, (n)へ入力された値です。



例

(s)のデータ型が16ビットデータ(WORD型), (n)への入力値が3の場合。(3ビット分左へ回転します。)



- ・ (s)への入力値は, 16ビットデータ/32ビットデータ(WORD型/DWORD型)のデータです。
- ・ (n)への入力値(シフトビット数指定)は, 下記に示す範囲内です。

(s)のデータが16ビットデータ(WORD型)の場合	(s)のデータが32ビットデータ(DWORD型)の場合
(n)の入力値は, 0~15の範囲内です。 (n)への入力値の下位4ビット分のデータが使用されます。 [例] 入力値が6の場合: 6 入力値が22の場合: 6	(n)の入力値は, 0~31の範囲内です。 (n)への入力値の下位5ビット分のデータが使用されます。 [例] 入力値が6の場合: 6 入力値が22の場合: 22

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

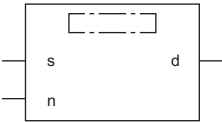
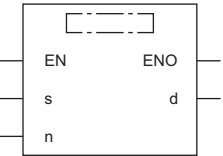
演算エラーはありません。

30.4 nビット右ローテーション

30

ROR(_E)

入力値を(n)ビット数分右へ回転して出力します。

ラダー, FBD/LD	ST
[EN/ENOなし]  [EN/ENO付き] 	[EN/ENOなし] $d := ROR(s, n);$ [EN/ENO付き] $d := ROR\_E(EN, ENO, s, n);$

設定データ

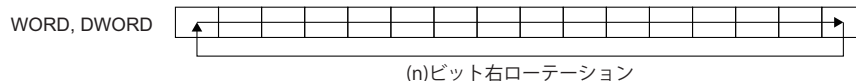
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANY_BIT
n(N)	シフトビット数指定	入力変数	ANY_BIT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_BIT

機能

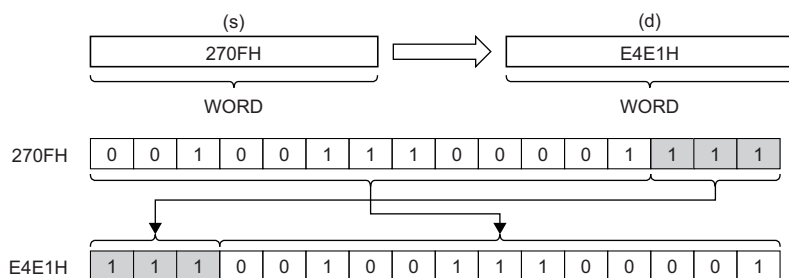
■演算処理

- (s)に入力された16ビットデータ/32ビットデータ(WORD型/DWORD型)を(n)ビット右へ回転(ローテーション)して, (d)から(s)と同一のデータ型で出力します。
- 右へ回転する(n)ビット数は, (n)へ入力された値です。



例

(s)のデータが16ビットデータ(WORD型), (n)への入力値が3の場合。(3ビット分右へ回転します。)



- (s)への入力値は, 16ビットデータ/32ビットデータ(WORD型/DWORD型)のデータです。
- (n)への入力値(シフトビット数指定)は, 下記に示す範囲内です。

(s)のデータが16ビットデータ(WORD型)の場合	(s)のデータが32ビットデータ(DWORD型)の場合
(n)の入力値は, 0~15の範囲内です。 (n)への入力値の下位4ビット分のデータが使用されます。 [例] 入力値が6の場合: 6 入力値が22の場合: 6	(n)の入力値は, 0~31の範囲内です。 (n)への入力値の下位5ビット分のデータが使用されます。 [例] 入力値が6の場合: 6 入力値が22の場合: 22

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件		演算結果	
EN	ENO	(d)	
TRUE(演算実行)	TRUE		演算出力値
FALSE(演算停止)	FALSE ^{*1}		不定値

^{*1} ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

演算エラーはありません。

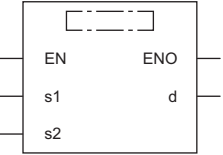
31 ビット型ブールファンクション

31

31.1 論理積，論理和，排他的論理和

AND(_E), OR(_E), XOR(_E)

- AND(_E): 入力値の論理積を出力します。
- OR(_E): 入力値の論理和を出力します。
- XOR(_E): 入力値の排他的論理和を出力します。

ラダー，FBD/LD*1		ST*1
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] 演算子として記述します。(I-93ページ 演算子) [EN/ENO付き] d:=AND_E(EN,ENO,s1,s2); d:=OR_E(EN,ENO,s1,s2); d:=XOR_E(EN,ENO,s1,s2);

*1 入力変数sは，2～28の範囲で変更できます。

設定データ

■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1)~s28(IN28)	入力	入力変数	ANY_BIT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_BIT

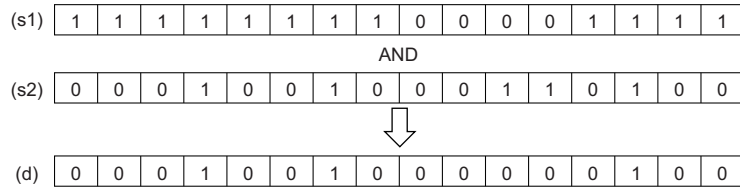
■演算処理

1. AND(_E)

- (s1)~(s28)に入力されたBOOL型/WORD型/DWORD型データをビットごとに論理積を行い、演算結果を(d)から(s)と同一のデータ型で出力します。

例

データ型がWORD型の場合

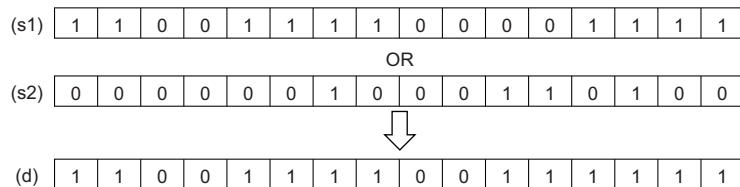


2. OR(_E)

- (s1)~(s28)に入力されたBOOL型/WORD型/DWORD型データをビットごとに論理和を行い、演算結果を(d)から(s)と同一のデータ型で出力します。

例

データ型がWORD型の場合

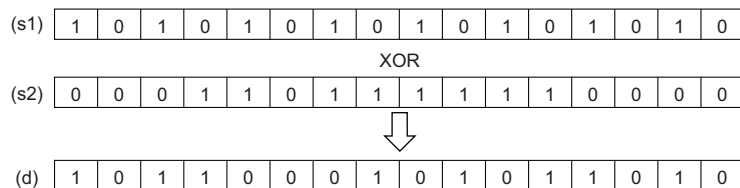


3. XOR(_E)

- (s1)~(s28)に入力されたBOOL型/WORD型/DWORD型データをビットごとに排他的論理和を行い、演算結果を(d)から(s)と同一のデータ型で出力します。

例

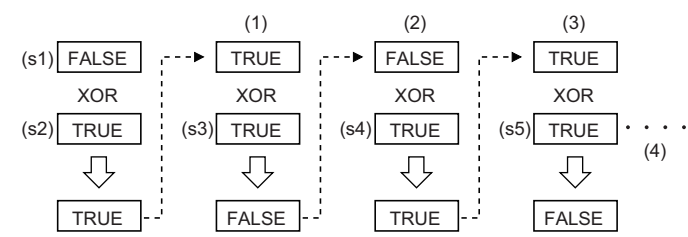
データ型がWORD型の場合



- (s)が3つ以上ある場合、(s1)と(s2)をXORした結果に対して、(s3)をXORします。また、(s4)がある場合は、(s3)をXORした結果に対して(s4)をXORします。以降(s5)、(s6)と(s)の数だけXORしていきます。

例

データ型がBOOL型の場合



- (1): sの数が3つの場合
- (2): sの数が4つの場合
- (3): sの数が5つの場合
- (4): 以降sの数だけXORしていきます。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE*1	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

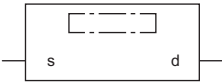
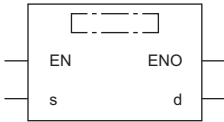
エラー

演算エラーはありません。

31.2 論理否定

NOT(_E)

入力値の論理否定を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] 演算子として記述します。(93ページ 演算子) [EN/ENO付き] d:=NOT_E(EN,ENO,s);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)*1	入力	入力変数	ANY_BIT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_BIT

*1 DXは使用できません。

機能

■演算処理

- ・ (s)に入力されたBOOL型/WORD型/DWORD型データをビットごとに論理否定を行い, 演算結果を(d)から(s)と同一のデータ型で出力します。

例

データ型がWORD型の場合

(s)	0	1	1	0	1	0	1	1	0	0	0	0	1	1	1	1
	NOT															
(d)	1	0	0	1	0	1	0	0	1	1	1	1	0	0	0	0

- ・ (s)への入力値は, BOOL型/WORD型/DWORD型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE*1	不定値

*1 ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

演算エラーはありません。

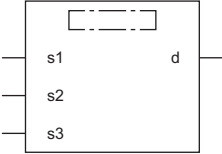
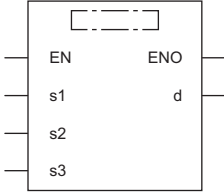
32 選択ファンクション

32

32.1 選択値

SEL(_E)

選択した入力値を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=SEL(s1,s2,s3); [EN/ENO付き] d:=SEL_E(EN,ENO,s1,s2,s3);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(G)	出力条件(TRUE: s3出力, FALSE: s2出力)	入力変数	BOOL
s2(IN0)	入力	入力変数	ANY
s3(IN1)			
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY

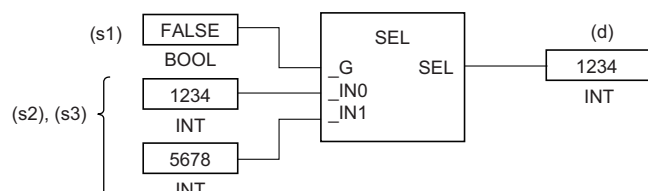
機能

■演算処理

- ・(s1)へ入力された入力値に従い, (s2), (s3)へ入力された値のいずれかを, (d)から(s2), (s3)と同一のデータ型で出力します。
- ・(s1)の入力値がFALSE(=0)の場合, (s2)の入力値を(d)から出力します。
- ・(s1)の入力値がTRUE(=1)の場合, (s3)の入力値を(d)から出力します。

例

(s2), (s3)のデータ型が, INT型の場合(引数名の(s2), (s3)は, (s1)のビット値(0または1)に対応しています。)



- ・(s1)の入力値は, BOOL型のデータ値です。
- ・(s2), (s3)への入力値は, BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型/構造体型/配列型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

- (s2), (s3)がSTRING型の場合

エラーコード (SD0)	内容
2820H	(s2)で指定したラベル, またはデバイス番号から該当デバイスの最終デバイス番号までの間に, 00Hが設定されていないとき。
	(s3)で指定したラベル, またはデバイス番号から該当デバイスの最終デバイス番号までの間に, 00Hが設定されていないとき。
3286H	(d)で指定したラベル, またはデバイス番号から該当デバイスの最終デバイス番号までの点数に, 指定した文字列がすべて格納できないとき。

32.2 最大値, 最小値選択

MAX(_E), MIN(_E)

- MAX(_E): 入力値の最大値を出力します。
- MIN(_E): 入力値の最小値を出力します。

ラダー, FBD/LD ^{*1}		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] $d := \text{MAX}(s1, s2);$ $d := \text{MIN}(s1, s2);$ [EN/ENO付き] $d := \text{MAX\_E}(\text{EN}, \text{ENO}, s1, s2);$ $d := \text{MIN\_E}(\text{EN}, \text{ENO}, s1, s2);$
		

*1 入力変数sは, 2~28の範囲で変更できます。

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1)~s28(IN28)	入力	入力変数	ANY_ELEMENTARY
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_ELEMENTARY

機能

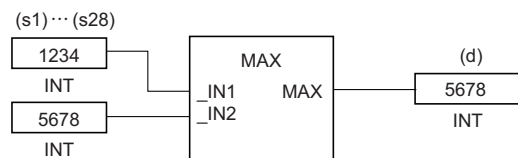
■演算処理

• MAX(_E)

(s1)~(s28)に入力されたBOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型データの最大値を、(d)から(s)と同一のデータ型で出力します。

例

データ型が、INT型の場合

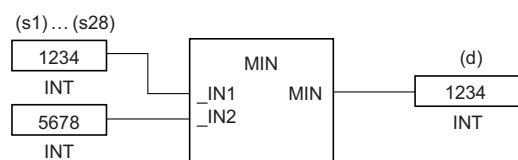


• MIN(_E)

(s1)~(s28)に入力されたBOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型データの最小値を、(d)から(s)と同一のデータ型で出力します。

例

データ型が、INT型の場合



- (s1)~(s28)への入力値は、BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型のデータ値です。
- 文字列の比較を行う場合、比較条件は下記のとおりです。
 - 一致の条件:
 - すべての文字列が一致した場合
 - 大きい文字列の条件:
 - 異なった文字列の場合は、文字コードが大きい文字列
 - 文字列の長さが異なる場合は、文字列が長い文字列
 - 小さい文字列の条件:
 - 異なった文字列の場合は、文字コードが小さい文字列
 - 文字列の長さが異なる場合は、文字列が短い文字列

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

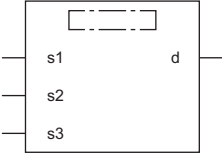
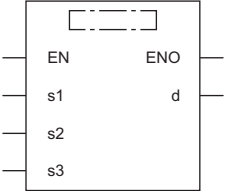
エラー

エラーコード (SD0)	内容
2820H	(s1)~(s28)で指定したデバイス番号以降、デバイス/ラベルメモリの各設定エリアの間に00Hが存在しないとき。
3285H	(s1)~(s28)の文字列が16383文字を超えるとき。
3286H	(d)で指定したデバイス番号以降、該当デバイス/ラベルメモリの各設定エリアの最終番号までの点数に、指定した文字列がすべて格納できないとき。

32.3 上下限リミット制御

LIMIT(_E)

上下限リミット制御された入力値を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし] 		[EN/ENOなし] d:=LIMIT(s1,s2,s3); [EN/ENO付き] d:=LIMIT_E(EN,ENO,s1,s2,s3);
[EN/ENO付き] 		

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(MN)*1	下限リミット値(最小出力限界値)	入力変数	ANY_ELEMENTARY
s2(IN)*1	上下限リミット制御により制御する入力値	入力変数	ANY_ELEMENTARY
s3(MX)*1	上限リミット値(最大出力限界値)	入力変数	ANY_ELEMENTARY
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY_ELEMENTARY

*1 DXは使用できません。

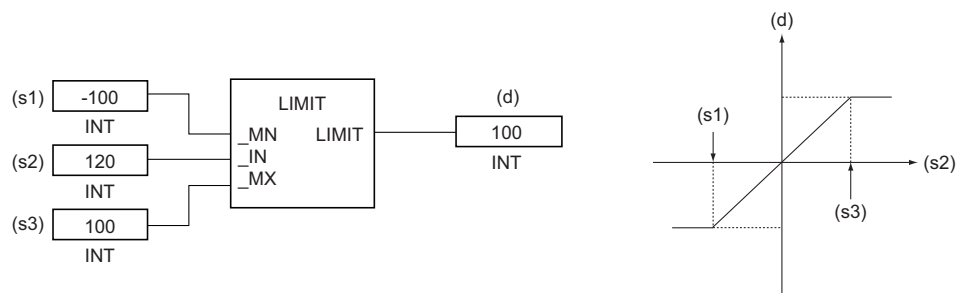
機能

■演算処理

- (s1), (s2), (s3)へ入力されたBOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型データに従い、(d)から(s1), (s2), (s3)と同一のデータ型で出力します。
- (s2)の入力値>(s3)の入力値の場合、(d)から(s3)の入力値を出力します。
- (s2)の入力値<(s1)の入力値の場合、(d)から(s1)の入力値を出力します。
- (s1)の入力値≤(s2)の入力値≤(s3)の入力値の場合、(d)から(s2)の入力値を出力します。

例

データ型が、INT型の場合



- (s1), (s2), (s3)への入力値は、BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型のデータ値です。(ただし、(s1)の入力値<(s3)の入力値)
- 文字列の比較を行う場合、比較条件は下記のとおりです。
 - 一致の条件:
 - すべての文字列が一致した場合
 - 大きい文字列の条件:
 - 異なった文字列の場合は、文字コードが大きい文字列
 - 文字列の長さが異なる場合は、文字列が長い文字列
 - 小さい文字列の条件:
 - 異なった文字列の場合は、文字コードが小さい文字列
 - 文字列の長さが異なる場合は、文字列が短い文字列

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

- (s1), (s2), (s3)がINT型/WORD型の場合

エラーコード (SD0)	内容
3285H	(s1)で指定した下限リミット値が、(s2)で指定した上限リミット値よりも大きいとき。

- (s1), (s2), (s3)がDINT型/DWORD型/TIME型の場合

エラーコード (SD0)	内容
3285H	(s1)で指定した下限リミット値が、(s2)で指定した上限リミット値よりも大きいとき。

• (s1), (s2), (s3)がBOOL型の場合

エラーコード (SD0)	内容
3285H	(s1)で指定した下限リミット値が, (s3)で指定した上限リミット値よりも大きいとき。

• (s1), (s2), (s3)がREAL型の場合

エラーコード (SD0)	内容
3282H	(s1)の内容が下記範囲外のとき。 $-2^{128} < (s1) \leq -2^{126}$, 0, $2^{-126} \leq (s1) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38, 0, E1.17549435-38~E3.40282347+38) (s1)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (s2)の内容が下記範囲外のとき。 $-2^{128} < (s2) \leq -2^{126}$, 0, $2^{-126} \leq (s2) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38, 0, E1.17549435-38~E3.40282347+38) (s2)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (s3)の内容が下記範囲外のとき。 $-2^{128} < (s3) \leq -2^{126}$, 0, $2^{-126} \leq (s3) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38, 0, E1.17549435-38~E3.40282347+38) (s3)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3285H	(s1)で指定した下限リミット値が, (s3)で指定した上限リミット値よりも大きいとき。

• (s1), (s2), (s3)がLREAL型の場合

エラーコード (SD0)	内容
3282H	(s1)の内容が下記範囲外のとき。 $-2^{1024} < (s1) \leq -2^{1022}$, 0, $2^{-1022} \leq (s1) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308, 0, E2.2250738585072014-308~E1.7976931348623157+308) (s1)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (s2)の内容が下記範囲外のとき。 $-2^{1024} < (s2) \leq -2^{1022}$, 0, $2^{-1022} \leq (s2) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308, 0, E2.2250738585072014-308~E1.7976931348623157+308) (s2)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。 (s3)の内容が下記範囲外のとき。 $-2^{1024} < (s3) \leq -2^{1022}$, 0, $2^{-1022} \leq (s3) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308, 0, E2.2250738585072014-308~E1.7976931348623157+308) (s3)で指定したデータが-0, 非正規化数, 非数, $\pm\infty$ のとき。
3285H	(s1)で指定した下限リミット値が, (s3)で指定した上限リミット値よりも大きいとき。

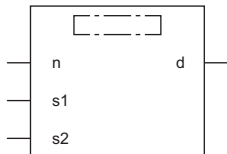
• (s1), (s2), (s3)がSTRING型の場合

エラーコード (SD0)	内容
2820H	(s1), (s2), (s3)で指定したラベル, または指定したデバイス番号から該当デバイスの最終デバイス番号までの間に, 00Hが設定されていないとき。
3285H	(s1)で指定した下限リミット値が, (s3)で指定した上限リミット値よりも大きいとき。 (s1), (s2), (s3)で指定する文字列の文字数が, 16383文字を超えるととき。
3286H	(d)で指定されたラベル, または指定したデバイス番号から該当デバイスの最終デバイス番号までの点数に, 指定した文字列がすべて格納できないとき。

32.4 マルチプレクサ

MUX(_E)

複数入力値の一つを出力します。

ラダー, FBD/LD*1		ST*1
[EN/ENOなし] 		[EN/ENOなし] d:=MUX(n,s1,s2); [EN/ENO付き] d:=MUX_E(EN,ENO,n,s1,s2);

*1 入力変数sは、2~28の範囲で変更できます。

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
n(K)	出力値選択	入力変数	ANY_INT
s1(IN0)~s28(IN27)	入力	入力変数	ANY
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANY

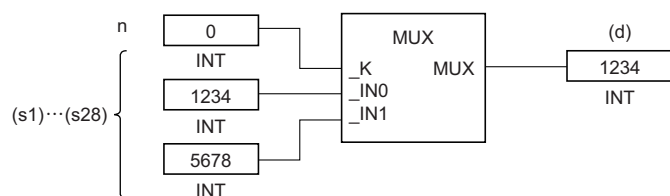
機能

■演算処理

- (n)へ入力された入力値に従い、(s1)~(s28)へ入力された値のいずれかを、(d)から(s)と同一のデータ型で出力します。
- (n)の入力値が0の場合、(s1)へ入力された値を(d)から出力します。
- (n)の入力値が(n)-1の場合、(sn)へ入力された値を(d)から出力します。

例

データ型が、INT型の場合



- (n)に(s)のピン数の範囲外を入力した場合、(d)から不定値が出力されます。(演算エラーにはなりません。またMUX_Eのときは、(ENO)からFALSEが出力されます。)
- (n)への入力値は、INT型/DINT型/WORD型/DWORD型のデータ値で0~27の範囲内です。(ただし、(s)のピン数の範囲内です。)
- (s)への入力値は、BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型/構造体型/配列型のデータ値です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件		演算結果
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

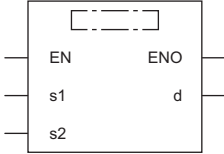
エラーコード (SD0)	内容
2820H	(s1)~(s28)で指定したデバイス番号以降、デバイス/ラベルメモリの各設定エリアの間に00Hが存在しないとき。
3285H	(s1)~(s28)の文字列が16383文字を超えるとき。
3286H	(d)で指定したデバイス番号以降、該当デバイス/ラベルメモリの各設定エリアの最終番号までの点数に、指定した文字列がすべて格納できないとき。

33 比較ファンクション

33.1 比較

GT(_E), GE(_E), EQ(_E), LE(_E), LT(_E)

入力値のデータ比較結果を出力します。

ラダー, FBD/LD *1		ST*1
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] d:=GT(s1,s2); d:=GE(s1,s2); d:=EQ(s1,s2); d:=LE(s1,s2); d:=LT(s1,s2); [EN/ENO付き] d:=GT_E(EN,ENO,s1,s2); d:=GE_E(EN,ENO,s1,s2); d:=EQ_E(EN,ENO,s1,s2); d:=LE_E(EN,ENO,s1,s2); d:=LT_E(EN,ENO,s1,s2);

*1 入力変数sは、2~28の範囲で変更できます。

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1)~s28(IN28)*1	入力	入力変数	ANY_ELEMENTARY
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力(TRUE: 真値, FALSE: 偽値)	出力変数	BOOL

*1 DXは使用できません。

機能

■演算処理

- (s)へ入力された入力値の比較演算を行い、演算結果を(d)からBOOL型で出力します。
- GT(E): $[(s1) > (s2)] \& [(s2) > (s3)] \& \dots \& [(s)_{(n-1)} > (s)_{(n)}]$ の比較を行います。
 - すべて $(s)_{(n-1)} > (s)_{(n)}$ のとき、TRUEを出力します。
 - いずれかが $(s)_{(n-1)} \leq (s)_{(n)}$ のとき、FALSEを出力します。
- GE(E): $[(s1) \geq (s2)] \& [(s2) \geq (s3)] \& \dots \& [(s)_{(n-1)} \geq (s)_{(n)}]$ の比較を行います。
 - すべて $(s)_{(n-1)} \geq (s)_{(n)}$ のとき、TRUEを出力します。
 - いずれかが $(s)_{(n-1)} < (s)_{(n)}$ のとき、FALSEを出力します。
- EQ(E): $[(s1) = (s2)] \& [(s2) = (s3)] \& \dots \& [(s)_{(n-1)} = (s)_{(n)}]$ の比較を行います。
 - すべて $(s)_{(n-1)} = (s)_{(n)}$ のとき、TRUEを出力します。
 - いずれかが $(s)_{(n-1)} \neq (s)_{(n)}$ のとき、FALSEを出力します。
- LE(E): $[(s1) \leq (s2)] \& [(s2) \leq (s3)] \& \dots \& [(s)_{(n-1)} \leq (s)_{(n)}]$ の比較を行います。
 - すべて $(s)_{(n-1)} \leq (s)_{(n)}$ のとき、TRUEを出力します。
 - いずれかが $(s)_{(n-1)} > (s)_{(n)}$ のとき、FALSEを出力します。
- LT(E): $[(s1) < (s2)] \& [(s2) < (s3)] \& \dots \& [(s)_{(n-1)} < (s)_{(n)}]$ の比較を行います。
 - すべて $(s)_{(n-1)} < (s)_{(n)}$ のとき、TRUEを出力します。
 - いずれかが $(s)_{(n-1)} \geq (s)_{(n)}$ のとき、FALSEを出力します。
- (s)への入力値は、INT型/DINT型/REAL型/LREAL型/BOOL型/WORD型/DWORD型/TIME型/STRING型のデータ値です。
WSTRING型のUnicode文字列は指定できません。
- 文字列の比較を行う場合、比較条件は下記のとおりです。

一致の条件:	• すべての文字列が一致した場合
大きい文字列の条件:	• 異なった文字列の場合は、文字コードが大きい文字列 • 文字列の長さが異なる場合は、文字列が長い文字列
小さい文字列の条件:	• 異なった文字列の場合は、文字コードが小さい文字列 • 文字列の長さが異なる場合は、文字列が短い文字列

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

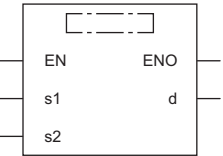
エラー

エラーコード (SD0)	内容
2820H	(s1)~(s28)で指定したデバイス番号以降、デバイス/ラベルメモリの各設定エリアの間に00Hが存在しないとき。
3285H	(s1)~(s28)の文字列が16383文字を超えるとき。
3286H	(d)で指定したデバイス番号以降、該当デバイス/ラベルメモリの各設定エリアの最終番号までの点数に、指定した文字列がすべて格納できないとき。

33.2 比較

NE(_E)

入力値のデータ比較結果を出力します。

ラダー， FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=NE(s1,s2); [EN/ENO付き] d:=NE_E(EN,ENO,s1,s2);
		

設定データ

■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1, s2	入力	入力変数	ANY_ELEMENTARY
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力(TRUE: 真値, FALSE: 偽値)	出力変数	BOOL

機能

■演算処理

- (s)へ入力された入力値の比較演算を行い，演算結果を(d)からBOOL型で出力します。
- NE(_E): [(s1)≠(s2)]の比較を行います。
 - (s1)≠(s2)のときTRUEを出力します。
 - (s1)=(s2)のときFALSEを出力します。
- (s)への入力値は，INT型/DINT型/REAL型/LREAL型/BOOL型/WORD型/DWORD型/TIME型/STRING型のデータ値です。
WSTRING型のUnicode文字列は指定できません。
- 文字列の比較を行う場合，比較条件は下記のとおりです。
 - 一致の条件:
 - すべての文字列が一致した場合
 - 大きい文字列の条件:
 - 異なった文字列の場合は，文字コードが大きい文字列
 - 文字列の長さが異なる場合は，文字列が長い文字列
 - 小さい文字列の条件:
 - 異なった文字列の場合は，文字コードが小さい文字列
 - 文字列の長さが異なる場合は，文字列が短い文字列

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合，(d)から出力されるデータは不定値となります。その場合は，(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラーコード (SD0)	内容
2820H	(s)で指定したデバイス番号以降，デバイス/ラベルメモリの各設定エリアの間に00Hが存在しないとき。
3285H	(s)の文字列が16383文字を超えるとき。
3286H	(d)で指定したデバイス番号以降，該当デバイス/ラベルメモリの各設定エリアの最終番号までの点数に，指定した文字列がすべて格納できないとき。

34 文字列ファンクション

34.1 文字列の長さ検出

LEN(_E)

入力された文字列の長さを検出して出力します。

ラダー, FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] d:=LEN(s); [EN/ENO付き] d:=LEN_E(EN,ENO,s);
		

設定データ

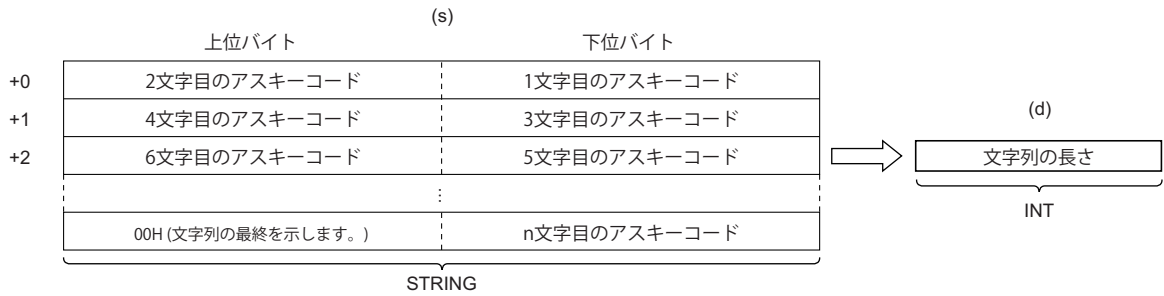
■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANYSTRING_SINGLE
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	INT

機能

■演算処理

- (s)に入力された文字列の長さを検出して, (d)から出力します。



- (s)への入力値は, STRING型のデータ値で0~255バイトの範囲内です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

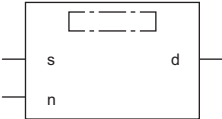
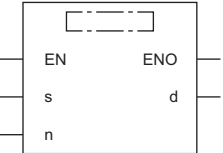
エラー

エラーコード (SD0)	内容
2820H	(s)で指定したデバイス番号以降、デバイス/ラベルメモリの各設定エリアに00Hが存在しないとき。
3285H	(s)の文字列が16383文字を超えるとき。

34.2 文字列の左側，右側からの抽出

LEFT(_E), RIGHT(_E)

- LEFT(_E): 入力された文字列データの左側から指定文字分を出力します。
- RIGHT(_E): 入力された文字列データの右側から指定文字分を出力します。

ラダー， FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] d:=LEFT(s,n); d:=RIGHT(s,n); [EN/ENO付き] d:=LEFT_E(EN,ENO,s,n); d:=RIGHT_E(EN,ENO,s,n);
		

設定データ

■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANYSTRING_SINGLE
n(L)	抽出文字数指定	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANYSTRING_SINGLE

機能

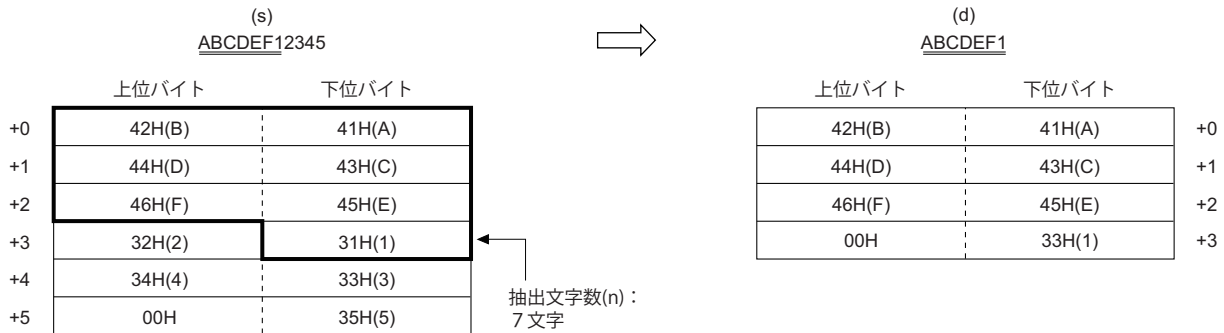
■演算処理

• LEFT(_E)

(s)へ入力された文字列の左側から指定文字数のデータを、(d)から出力します。
抽出する文字数は、(n)への入力値により指定します。

例

(n)への入力値が7の場合

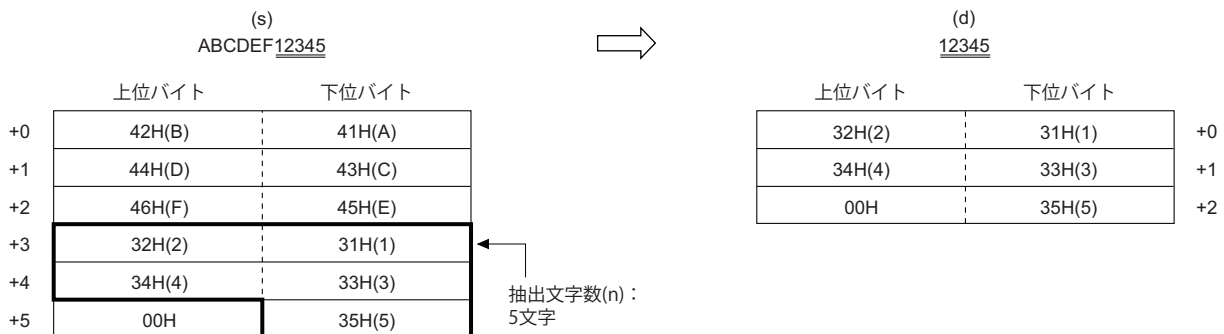


• RIGHT(_E)

(s)へ入力された文字列の右側から指定文字数のデータを、(d)から出力します。
抽出する文字数は、(n)への入力値により指定します。

例

(n)への入力値が5の場合



- (s)への入力値は、STRING型のデータ値で0~255バイトの範囲内です。
- (n)への入力値は、INT型のデータ値で0~255の範囲内です。(ただし、(s)へ入力される文字列の文字数以内に限りです。)

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

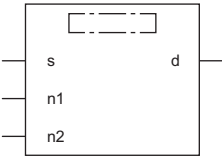
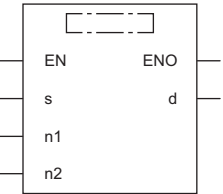
エラーコード (SD0)	内容
2820H	(s)で指定したデバイス番号以降、デバイス/ラベルメモリの各設定エリアに00Hが存在しないとき。
3285H	(s)の文字列が16383文字を超えるとき。
	(s)の文字列が0文字のとき。
	(n)が、(s)で指定する文字列の文字数を超えるとき。

34.3 文字列の抽出

34

MID(_E)

入力された文字列の任意位置から指定文字分を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=MID(s,n1,n2);
		[EN/ENO付き] d:=MID_E(EN,ENO,s,n1,n2);
		

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANYSTRING_SINGLE
n1(L)	抽出文字数指定	入力変数	INT
n2(P)	抽出先頭位置指定	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANYSTRING_SINGLE

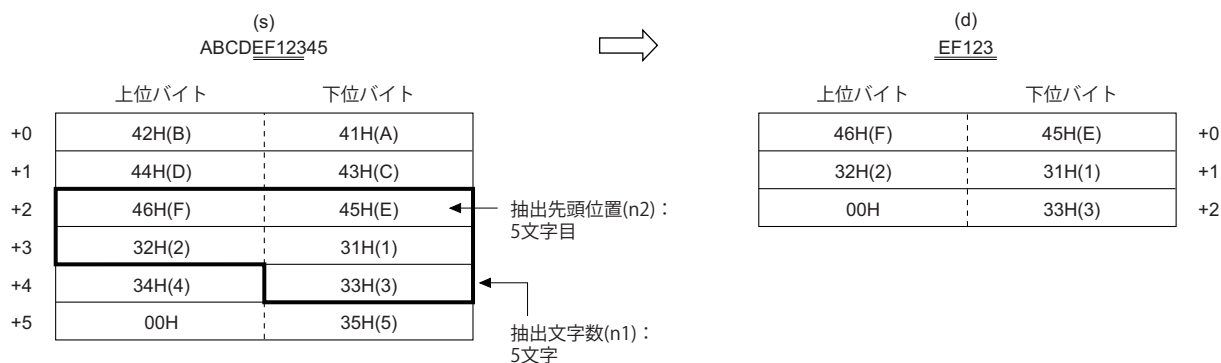
機能

■演算処理

- (s)へ入力された文字列の任意位置から指定文字数分のデータを, (d)から出力します。
- 抽出する文字数は, (n1)への入力値により指定します。
- 抽出する文字列の先頭位置は, (n2)への入力値により指定します。

例

(n1), (n2)への入力値が5の場合



- (s)への入力値は, STRING型のデータ値で0~255バイトの範囲内です。
- (n1)への入力値は, INT型のデータ値で0~255の範囲内です。(ただし, (s)へ入力される文字列の文字数以内に限りします。)
- (n2)への入力値は, INT型のデータ値で1~255の範囲内です。(ただし, (s)へ入力される文字列の文字数以内に限りします。)

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

エラーコード (SD0)	内容
2820H	(s)で指定したデバイス番号以降、デバイス/ラベルメモリの各設定エリアに00Hが存在しないとき。
3285H	(s)の文字列が16383文字を超えるとき。 (n1)、(n2)に指定可能な範囲外のデータを設定したとき。 • (n1)、(n2)の値が0以下 • (n2)の値が有効値(-1, 0, 1以上)以外 • (n1)の値が(s)の文字数を超えている • (n1)と(n2)を加算した値が(s)の文字数を超えている

34.4 文字列の結合

CONCAT(_E)

文字列を結合して出力します。

ラダー, FBD/LD* ¹		ST* ¹
[EN/ENOなし]		[EN/ENOなし] d:=CONCAT(s1,s2);
[EN/ENO付き]		[EN/ENO付き] d:=CONCAT_E(EN,ENO,s1,s2);

*1 入力変数sは、2~28の範囲で変更できます。

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1)~s28(IN28)	入力	入力変数	ANYSTRING_SINGLE
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANYSTRING_SINGLE

機能

■演算処理

- (s1)へ入力された文字列のあとに、入力変数(s2)~(s28)へ入力された文字列を結合して、(d)から出力します。
- 文字列の結合は、(s1)の文字列の終了を示す00Hを無視して、(s2)~(s28)の文字列を続けて結合します。
- 結合した文字列が255バイトを超えている場合は、255バイトまでの文字列を出力します。

(s1) <u>ABCDE</u>		+	(s2) <u>123456</u>		⇒	(d) <u>ABCDE123456</u>	
上位バイト	下位バイト		上位バイト	下位バイト		上位バイト	下位バイト
+0	42H(B)		+0	32H(2)		+0	42H(B)
+1	44H(D)		+1	34H(4)		+1	44H(D)
+2	00H		+2	36H(6)		+2	31H(1)
			+3	00H		+3	33H(3)
						+4	35H(5)
						+5	00H
							41H(A)
							43H(C)
							45H(E)
							32H(2)
							34H(4)
							36H(6)

- 入力変数(s1), (s2)~(s28)への入力値は、STRING型のデータ値で0~255バイトの範囲内です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

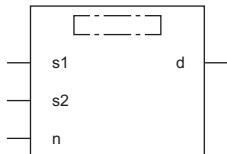
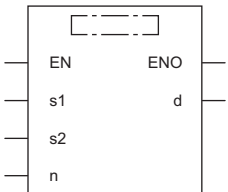
エラーコード (SD0)	内容
2820H	(s1)~(s28)で指定したデバイス番号以降、デバイス/ラベルメモリの各設定エリアの間に00Hが存在しないとき。 (d)で指定したデバイス番号以降、デバイス/ラベルメモリの各設定エリアの間に00Hが存在しないとき。
2821H	(s1)~(s28)と(d)で指定された文字列の格納デバイス番号が重複しているとき。
3285H	(s1)~(s28)の文字列が16383文字を超えるとき。 (s1)~(s28)の文字列が0文字のとき。 (d)の文字列が16383文字を超えるとき。
3286H	(d)で指定されたデバイス番号以降、該当デバイス/ラベルメモリの各設定エリアの最終番号までの点数に結合した文字列がすべて格納できないとき。

34.5 文字列の挿入

34

INSERT(_E)

文字列の間へ文字列を挿入して出力します。

ラダー, FBD/LD	ST
[EN/ENOなし] 	[EN/ENOなし] d:=INSERT(s1,s2,n); [EN/ENO付き] d:=INSERT_E(EN,ENO,s1,s2,n);
[EN/ENO付き] 	

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1), s2(IN2)	入力	入力変数	ANYSTRING_SINGLE
n(P)	挿入先頭位置指定	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANYSTRING_SINGLE

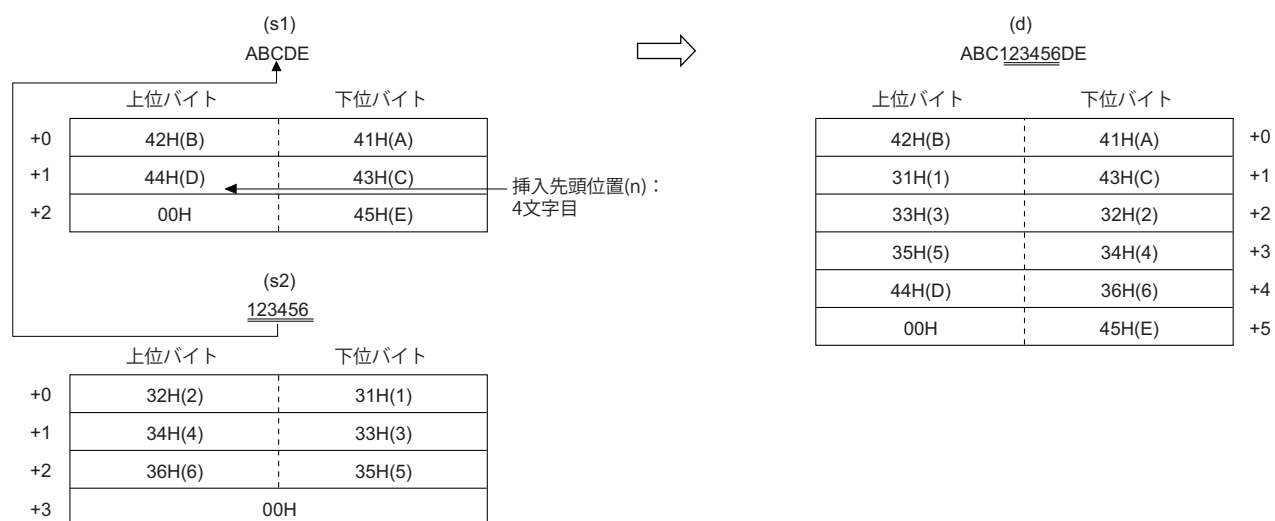
機能

■演算処理

- (s1)へ入力された文字列の先頭からn文字目(挿入先頭位置)に, (s2)へ入力された文字列を挿入して, (d)から出力します。
- (s2)の文字列を(s1)の文字列へ挿入後, (s2)の文字列の終了を示す00Hは無視します。
- 挿入した文字列が255バイトを超えている場合は, 255バイトまでの文字列を出力します。

例

(n)への入力値が, 4の場合



- (s1), (s2)への入力値は, STRING型のデータ値で0~255バイトの範囲内です。
- (n)への入力値は, INT型のデータ値で1~255の範囲内です。(ただし, (s1)へ入力される文字列の文字数以内に限りま。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

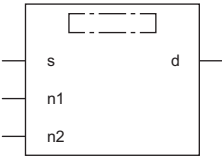
エラーコード (SD0)	内容
2820H	(s1)~(s28)で指定したデバイス番号以降、デバイス/ラベルメモリの各設定エリアの間に00Hが存在しないとき。 (d)で指定したデバイス番号以降、デバイス/ラベルメモリの各設定エリアの間に00Hが存在しないとき。
2821H	(s1)~(s28)と(d)で指定された文字列の格納デバイス番号が重複しているとき。
3285H	(s1)~(s28)の文字列が16383文字を超えるとき。 (s1)~(s28)の文字列が0文字のとき。 (d)の文字列が16383文字を超えるとき。
3286H	(d)で指定されたデバイス番号以降、該当デバイス/ラベルメモリの各設定エリアの最終番号までの点数に結合した文字列がすべて格納できないとき。

34.6 文字列の削除

34

DELETE(_E)

文字列の任意の範囲を削除して出力します。

ラダー, FBD/LD	ST
[EN/ENOなし] 	[EN/ENOなし] d:=DELETE(s,n1,n2); [EN/ENO付き] d:=DELETE_E(EN,ENO,s,n1,n2);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	入力	入力変数	ANYSTRING_SINGLE
n1(L)	削除文字数指定	入力変数	INT
n2(P)	削除先頭位置指定	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANYSTRING_SINGLE

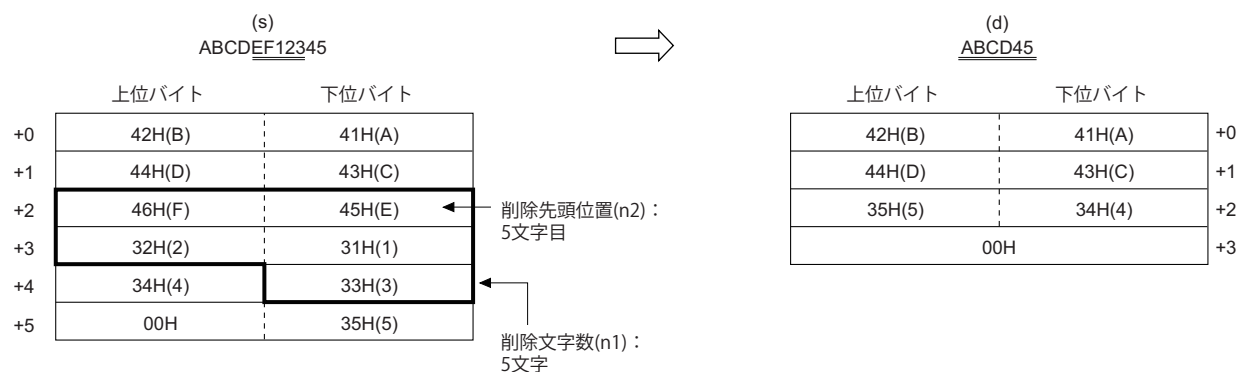
機能

■演算処理

- (s)へ入力された文字列の任意位置から指定文字数分のデータを削除して, 残りの文字列を(d)から出力します。
- 削除する文字数は, (n1)への入力値により指定します。
- 削除する文字列の先頭位置は, (n2)への入力値により指定します。

例

(n1), (n2)への入力値が, 5の場合



- (s)への入力値は, STRING型のデータ値で0~255バイトの範囲内です。
- (n1)への入力値は, INT型のデータ値で0~255の範囲内です。(ただし, (s)へ入力される文字列の文字数以内に限りします。)
- (n2)への入力値は, INT型のデータ値で1~255の範囲内です。(ただし, (s)へ入力される文字列の文字数以内に限りします。)

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

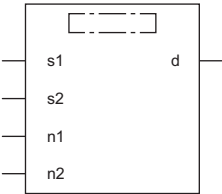
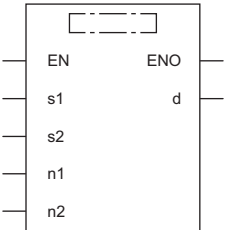
エラー

エラーコード (SD0)	内容
2820H	(s)で指定したラベル、または指定したデバイス番号から該当デバイスの最終デバイス番号までの間に、00Hが設定されていないとき。
3285H	(s)で指定する文字列の文字数が、255文字を超えるとき。
	(n1)で指定された内容が0~255の範囲外のとき。
	(n2)で指定された内容が1~255の範囲外のとき。
	(n1)が、(s)で指定する文字列の文字数を超えるとき。
	(n2)が、(s)で指定する文字列の文字数を超えるとき。
3286H	(d)で指定されたラベル、または指定したデバイス番号から該当デバイスの最終デバイス番号までの点数に、削除後の文字列がすべて格納できないとき。

34.7 文字列の置換

REPLACE(_E)

文字列の任意の範囲を置換して出力します。

ラダー, FBD/LD		ST
[EN/ENOなし]  [EN/ENO付き] 		[EN/ENOなし] d:=REPLACE(s1,s2,n1,n2); [EN/ENO付き] d:=REPLACE_E(EN,ENO,s1,s2,n1,n2);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1), s2(IN2)	入力	入力変数	ANYSTRING_SINGLE
n1(L)	置換文字数指定	入力変数	INT
n2(P)	置換先頭位置指定	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	ANYSTRING_SINGLE

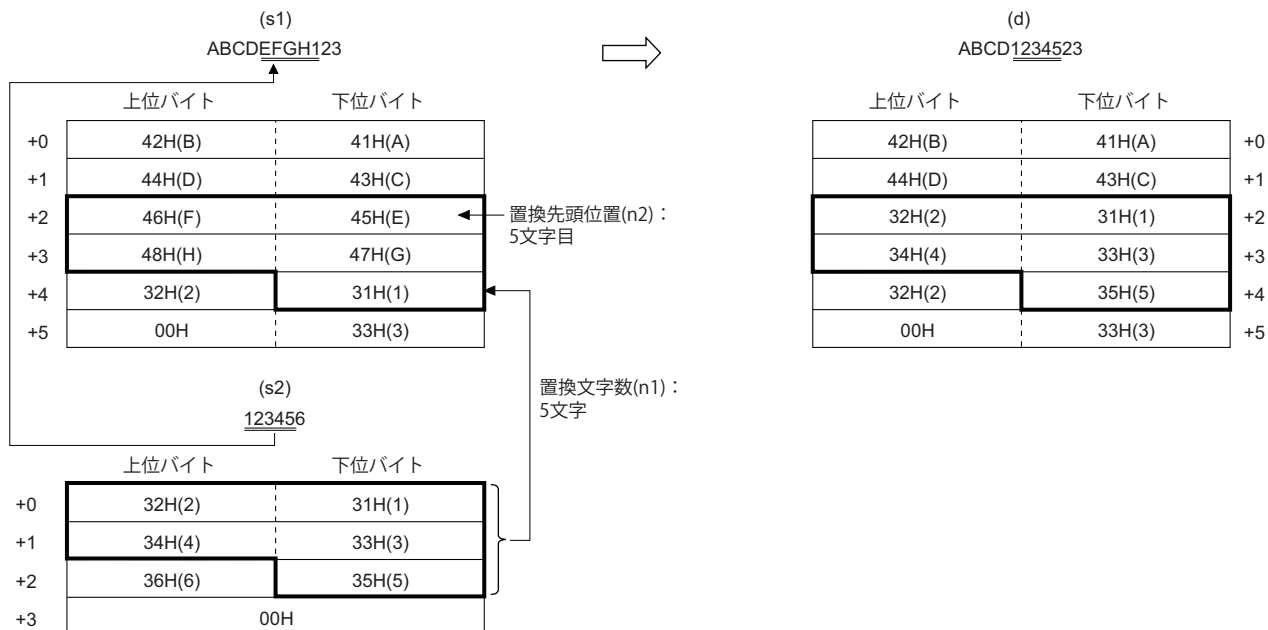
機能

■演算処理

- (s1)へ入力された文字列の任意位置から指定文字数分のデータを、(s2)へ入力された文字列へ置換して、(d)から出力します。
- 置換する文字数は、(n1)への入力値により指定します。
- 置換する文字列の先頭位置は、(n2)への入力値により指定します。

例

(n1), (n2)への入力値が、5の場合



- (s1), (s2)への入力値は、STRING型のデータ値で0~255バイトの範囲内です。
- (n1)への入力値は、INT型のデータ値で0~255の範囲内です。(ただし、(s1)へ入力される文字列の文字数以内に限りま
- (n2)への入力値は、INT型のデータ値で1~255の範囲内です。(ただし、(s1)へ入力される文字列の文字数以内に限りま

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

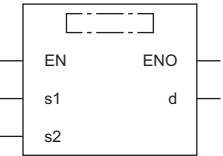
^{*1} ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラーコード (SD0)	内容
2820H	(s1)で指定したラベル, または指定したデバイス番号から該当デバイスの最終デバイス番号までの間に, 00Hが設定されていないとき。
	(s2)で指定したラベル, または指定したデバイス番号から該当デバイスの最終デバイス番号までの間に, 00Hが設定されていないとき。
3285H	(s1)で指定する文字列の文字数が, 255文字を超えるとき。
	(s2)で指定する文字列の文字数が, 255文字を超えるとき。
	(n1)で指定された内容が0~255の範囲外のとき。
	(n2)で指定された内容が1~255の範囲外のとき。
	(n1)が, (s2)で指定する文字列の文字数を超えるとき。
	(n2)が, (s1)で指定する文字列の文字数を超えるとき。
3286H	(d)で指定されたラベル, または指定したデバイス番号から該当デバイスの最終デバイス番号までの点数に削除後の文字列がすべて格納できないとき。

34.8 文字列の検索

FIND(_E)

文字列を検索して検索結果を出力します。

ラダー， FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=FIND(s1,s2); [EN/ENO付き] d:=FIND_E(EN,ENO,s1,s2);
		

設定データ

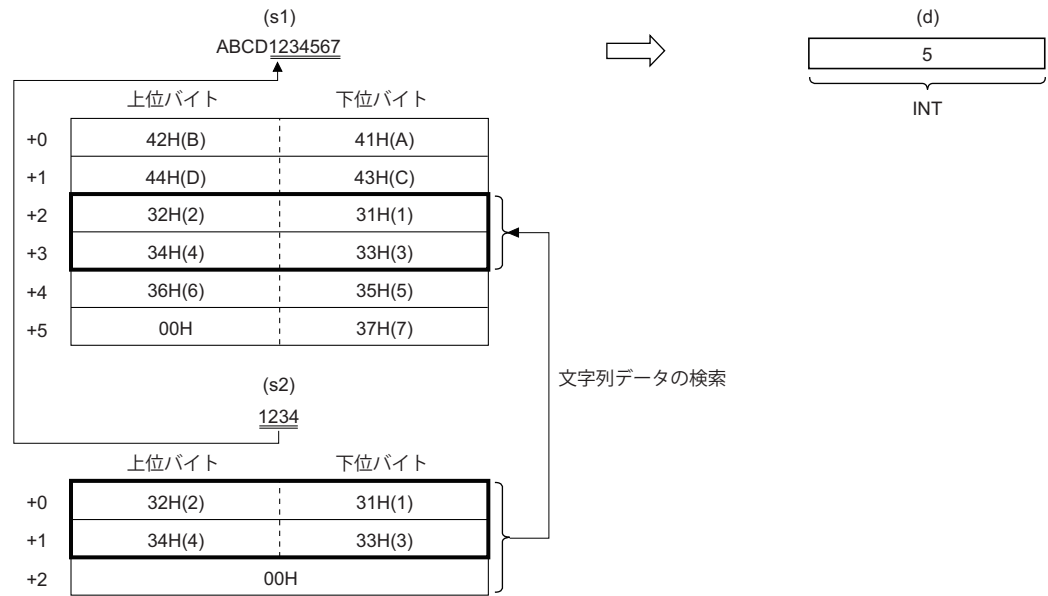
■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1), s2(IN2)	入力	入力変数	ANYSTRING_SINGLE
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	INT

機能

■演算処理

- (s1)へ入力された文字列の最初から，(s2)へ入力された文字列を検索して，検索結果を(d)から出力します。
- 検索結果は，最初に検索した文字列の先頭文字位置を出力します。
- (s1)の文字列から，(s2)の文字列が検索できなかった場合は，0を出力します。



- (s1), (s2)への入力値は，STRING型のデータ値で0~255バイトの範囲内です。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

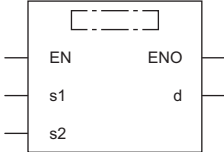
エラーコード (SD0)	内容
2820H	(s1)で指定したデバイス番号以降、デバイス/ラベルメモリの各設定エリアに00Hが存在しないとき。
	(s2)で指定したデバイス番号以降、デバイス/ラベルメモリの各設定エリアに00Hが存在しないとき。
3285H	(s1)の文字列が16383文字を超えるとき。
	(s1)の文字列が0文字のとき。
	(s2)の文字列が16383文字を超えるとき。
	(s2)の文字列が0文字のとき。

35 時刻データ型ファンクション

35.1 加算

ADD_TIME(_E)

入力値(TIME型)の和((s1)+(s2))を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] d:=ADD_TIME(s1,s2); [EN/ENO付き] d:=ADD_TIME_E(EN,ENO,s1,s2);
		

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1), s2(IN2)	入力	入力変数	TIME
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	TIME

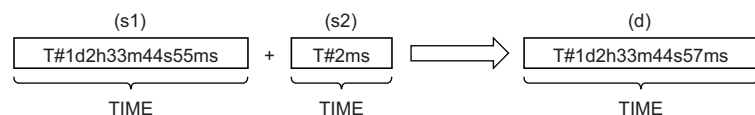
機能

■演算処理

- (s1), (s2)に入力されたTIME型データの加算((s1)+(s2))を行い、演算結果を(d)からTIME型で出力します。

例

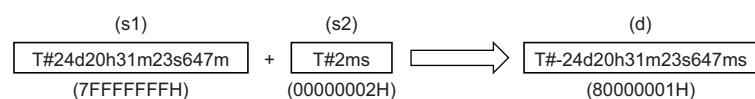
(s1), (s2)への入力値が、T#1d2h33m44s55ms(1日2時間33分44秒55ミリ秒)とT#2ms(2ミリ秒)の場合



- (s1), (s2)への入力値は、TIME型のデータ値です。
- 演算結果にアンダーフロー/オーバーフローが生じても演算エラーにはなりません。(d)へ下記のように出力されます。また、ADD_TIME_Eの場合、出力変数ENOからTRUEが出力されます。

例

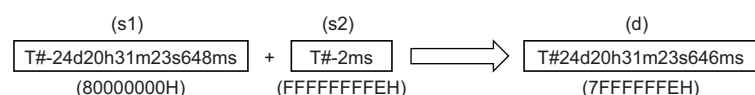
オーバーフロー



最上位ビットが1になるため、負の時間となります。

例

アンダーフロー



最上位ビットが0になるため、正の時間となります。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

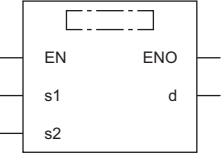
エラー

演算エラーはありません。

35.2 減算

SUB_TIME(_E)

入力値(TIME型)の差((s1)-(s2))を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=SUB_TIME(s1,s2); [EN/ENO付き] d:=SUB_TIME_E(EN,ENO,s1,s2);
		

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1), s2(IN2)	入力	入力変数	TIME
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	TIME

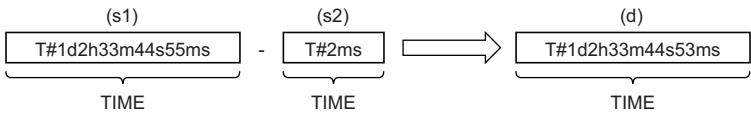
機能

■演算処理

- (s1), (s2)に入力されたTIME型データの減算((s1)-(s2))を行い, 演算結果を(d)からTIME型で出力します。

例

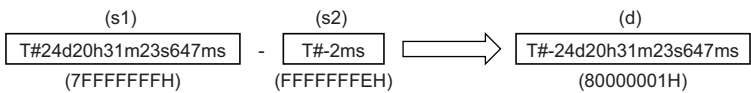
(s1), (s2)への入力値が, T#1d2h33m44s55ms(1日2時間33分44秒55ミリ秒)とT#2ms(2ミリ秒)の場合



- (s1), (s2)への入力値は, TIME型のデータ値です。
- 演算結果にアンダーフロー / オーバーフローが生じても演算エラーにはなりません。(d)へ下記のように出力されます。また, SUB_TIME_Eの場合, 出力変数ENOからTRUEが出力されます。

例

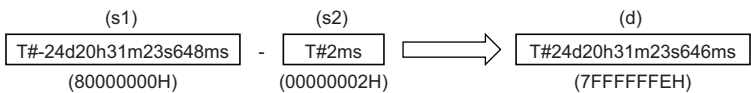
オーバーフロー



最上位ビットが1になるため, 負の時間となります。

例

アンダーフロー



最上位ビットが0になるため, 正の時間となります。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

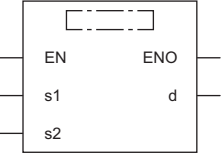
エラー

演算エラーはありません。

35.3 乗算

MUL_TIME(_E)

入力値(TIME型)の積((s1)×(s2))を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] d:=MUL_TIME(s1,s2); [EN/ENO付き] d:=MUL_TIME_E(EN,ENO,s1,s2);
		

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1)	入力	入力変数	TIME
s2(IN2)	入力	入力変数	ANY_NUM
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	TIME

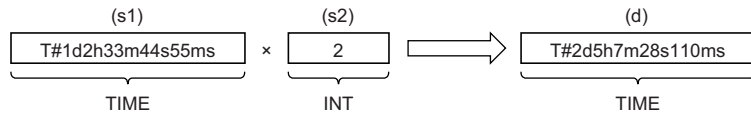
機能

■演算処理

- (s1), (s2)に入力されたTIME型データの乗算((s1)×(s2))を行い、演算結果を(d)からTIME型で出力します。

例

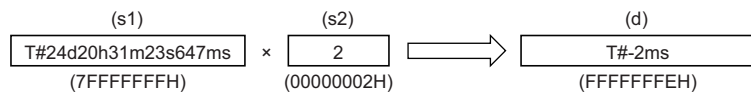
(s1), (s2)への入力値が、T#1d2h33m44s55ms(1日2時間33分44秒55ミリ秒)と2の場合



- (s1)への入力値は、TIME型のデータ値です。
- (s2)への入力値は、INT型/DINT型/REAL型/LREAL型です。
- 演算結果にアンダーフロー / オーバーフローが生じても演算エラーにはなりません。(d)へ下記のように出力されます。また、MUL_TIME_Eの場合、出力変数ENOからTRUEが出力されます。(演算結果は64ビットデータになりますが、出力は上位32ビットを削除した時間型のデータで出力されます。)

例

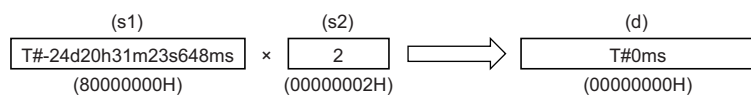
オーバーフロー



最上位ビットが1になるため、負の時間となります。

例

アンダーフロー



最上位ビットが0になるため、正の時間となります。

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

*1 ENOからFALSEが出力された場合、(d)から出力されるデータは不定値となります。その場合は、(d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

- (s2)がLREAL型の場合

エラーコード (SD0)	内容
3282H	(s2)の内容が下記範囲外のととき。 $-2^{1024} < (s2) \leq -2^{1022}$, 0 , $2^{-1022} \leq (s2) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308, 0 , E2.2250738585072014-308~E1.7976931348623157+308)
	(s2)で指定したデータが -0 、非正規化数、非数、 $\pm\infty$ のととき。
3285H	(s2)で設定した単精度実数データが-2147483648~2147483647の範囲外のととき。

35.4 除算

DIV_TIME(_E)

入力値(TIME型)の商((s1)÷(s2))を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] d:=DIV_TIME(s1,s2); [EN/ENO付き] d:=DIV_TIME_E(EN,ENO,s1,s2);
		

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(IN1)	入力	入力変数	TIME
s2(IN2)	入力	入力変数	ANY_NUM
ENO	出力状態(TRUE: 正常, FALSE: 異常)	出力変数	BOOL
d	出力	出力変数	TIME

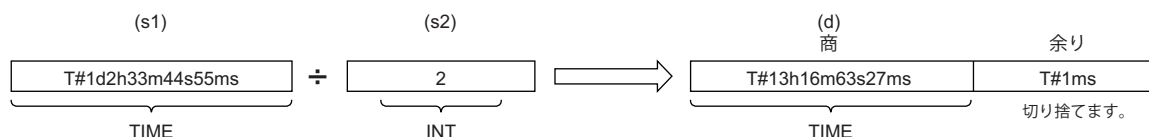
機能

■演算処理

- (s1), (s2)に入力されたTIME型データの除算((s1)÷(s2))を行い, 演算結果の商を(d)からTIME型で出力します。余りは切り捨てます。

例

(s1), (s2)への入力値が, T#1d2h33m44s55ms(1日2時間33分44秒55ミリ秒)と2の場合



- (s1)への入力値は, TIME型のデータ値です。
- (s2)への入力値は, INT型/DINT型/REAL型/LREAL型です。(ただし, (s2)へ入力される値は, 0以外です。)

■演算結果

1. EN/ENOなしファンクション

演算処理を実行します。(d)からは演算出力値が出力されます。

2. EN/ENO付きファンクション

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE ^{*1}	不定値

^{*1} ENOからFALSEが出力された場合, (d)から出力されるデータは不定値となります。その場合は, (d)から出力されたデータを使用しないようなプログラムの処理を行ってください。

エラー

エラーコード (SD0)	内容
3280H	(s2)へ入力される値が0のとき。(0割り)

• (s2)がLREAL型の場合

エラーコード (SD0)	内容
3282H	(s2)の内容が下記範囲外のとき。 $-2^{1024} < (s2) \leq -2^{-1022}$, 0, $2^{-1022} \leq (s2) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308, 0, E2.2250738585072014-308~E1.7976931348623157+308)
3285H	(s2)の内容が-2147483648~2147483647の範囲外のとき。

MEMO

第7部 汎用FB

この部は下記の章構成となります。

36 二安定ファンクションブロック

37 エッジ検出ファンクションブロック

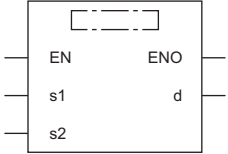
38 カウンタ/タイマファンクションブロック

36 二安定ファンクションブロック

36.1 二安定ファンクションブロック(セット優先)

SR(_E)

2つの入力値を判別し，1(TRUE)または0(FALSE)を出力します。

ラダー， FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] インスタンス名(S1:=s1,R:=s2,Q1:=d); [EN/ENO付き] インスタンス名(EN:=en,ENO:=eno,S1:=s1,R:=s2,Q1:=d);

設定データ

■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(S1)	セット指令	入力変数	BOOL
s2(R)	リセット指令	入力変数	BOOL
ENO	出力状態(TRUE: 正常, FALSE: 異常または停止)	出力変数	BOOL
d(Q1)	出力	出力変数	BOOL

■演算処理

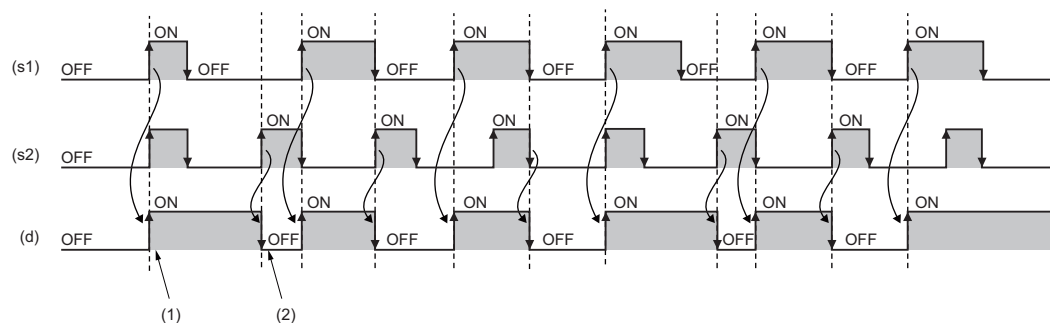
- (s1)がONすると、(d)をSETします。(s1)がOFFの時、(s2)をONすると、(d)をRESETできます。
- (s1)がONのとき、(s2)をONしても、(d)はRESETされません。

■演算結果

1. EN/ENOなしファンクションブロック

演算処理を実行します。(d)からは演算出力値が出力されます。

- タイミングチャート



(1) (s1)のOFF→ONで(d)をONします。

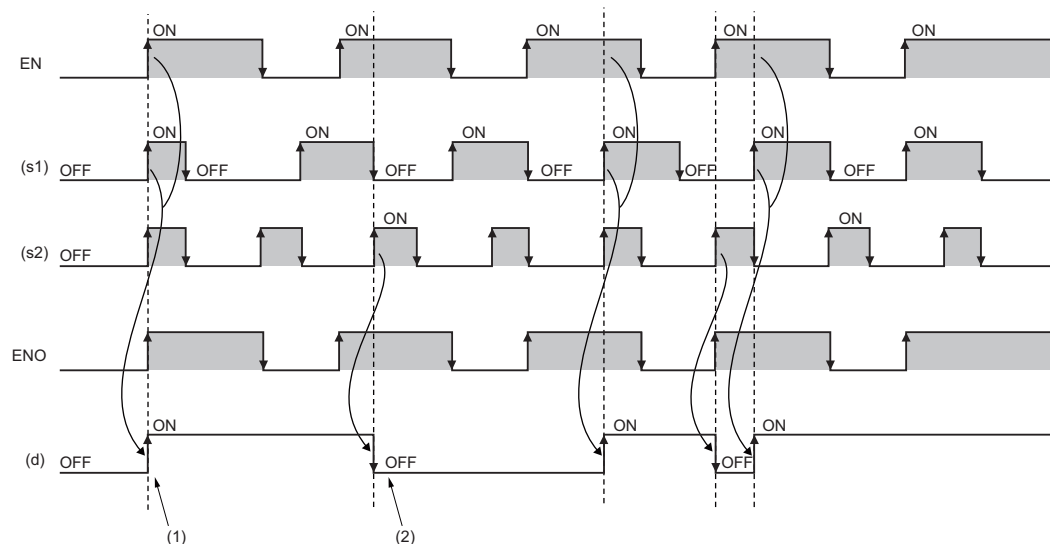
(2) (s1)=OFFかつ(s2)のOFF→ONで(d)をOFFします。

2. EN/ENO付きファンクションブロック

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE	前回出力値

- タイミングチャート



(1) EN=ONかつ(s1)のOFF→ONで(d)をONします。

(2) EN=ONかつ(s1)=OFFかつ(s2)のOFF→ONで(d)をOFFします。

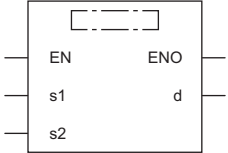
エラー

演算エラーはありません。

36.2 二安定ファンクションブロック(リセット優先)

RS(_E)

2つの入力値を判別し、1(TRUE)または0(FALSE)を出力します。

ラダー， FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] インスタンス名(S:=s1,R1:=s2,Q1:=d); [EN/ENO付き] インスタンス名(EN:=en,ENO:=eno,S:=s1,R1:=s2,Q1:=d);
		

設定データ

■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(S)	セット指令	入力変数	BOOL
s2(R1)	リセット指令	入力変数	BOOL
ENO	出力状態(TRUE: 正常, FALSE: 異常または停止)	出力変数	BOOL
d(Q1)	出力	出力変数	BOOL

■演算処理

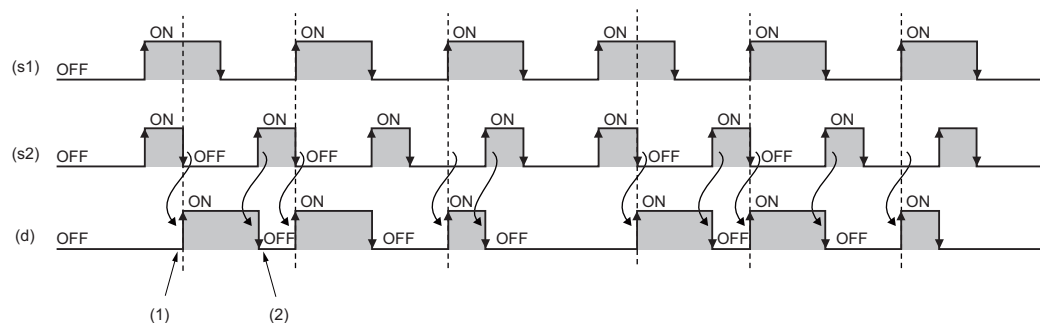
- (s1)がONすると、(d)をSETします。(s2)をONすると、(d)をRESETします。
- (s2)がONのとき、(s1)をONしても、(d)はSETされません。

■演算結果

1. EN/ENOなしファンクションブロック

演算処理を実行します。(d)からは演算出力値が出力されます。

- タイミングチャート



(1) (s1)=ONかつ(s2)のON→OFFで(d)をONします。

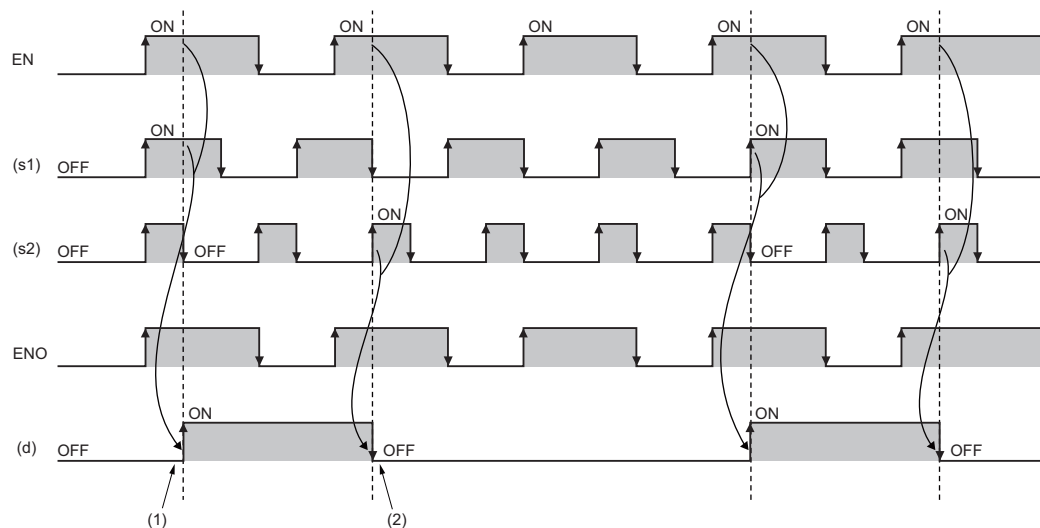
(2) (s2)のOFF→ONで(d)をOFFします。

2. EN/ENO付きファンクションブロック

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE	前回出力値

- タイミングチャート



(1) EN=ONかつ(s1)=ONかつ(s2)のON→OFFで(d)をONします。

(2) EN=ONかつ(s2)のOFF→ONで(d)をOFFします。

エラー

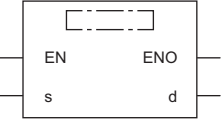
演算エラーはありません。

37 エッジ検出ファンクションブロック

37.1 立ち上りエッジ検出

R_TRIG(_E)

信号の立ち上がりを検出してパルス信号を出力します。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] インスタンス名(CLK:=s,Q:=d); [EN/ENO付き] インスタンス名(EN:= en,ENO:=eno,CLK:=s,Q:=d);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(CLK)	立ち上がりエッジ検出入力	入力変数	BOOL
ENO	出力状態(TRUE: 正常, FALSE: 異常または停止)	出力変数	BOOL
d(Q)	出力	出力変数	BOOL

機能

■演算処理

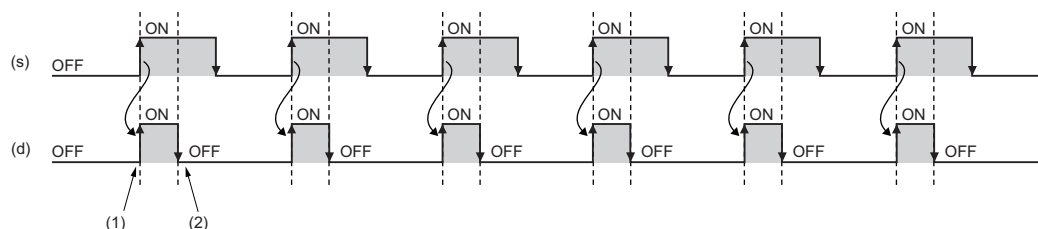
(s)がONすると、(d)を1スキャンのみONします。

■演算結果

1. EN/ENOなしファンクションブロック

演算処理を実行します。(d)からは演算出力値が出力されます。

・ タイミングチャート



(1) (s)の立上りで(d)をONします。

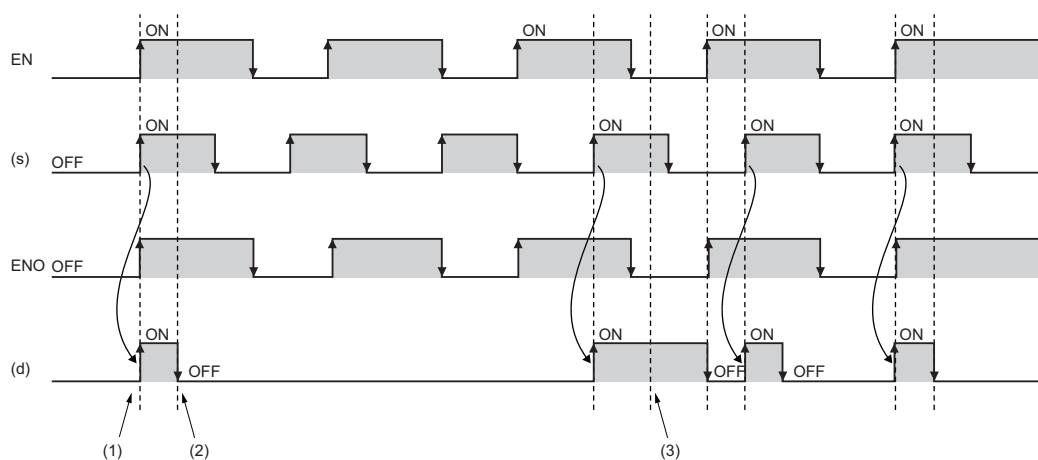
(2) 次スキャンで(d)をOFFします。

2. EN/ENO付きファンクションブロック

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE	前回出力値

・ タイミングチャート



(1) EN=ONかつ(s)の立上りで(d)をONします。

(2) 次スキャンで(d)をOFFします。

(3) EN=OFFの場合は、(d)は前回のスキャンの出力を保持します。

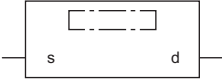
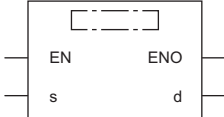
エラー

演算エラーはありません。

37.2 立ち下りエッジ検出

F_TRIG(_E)

信号の立ち下がりを検出してパルス信号を出力します。

ラダー， FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] インスタンス名(CLK:=s,Q:=d); [EN/ENO付き] インスタンス名(EN:=en,ENO:=eno,CLK:=s,Q:=d);
		

設定データ

■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行， FALSE: 停止)	入力変数	BOOL
s(CLK)	立ち下がりエッジ検出入力	入力変数	BOOL
ENO	出力状態(TRUE: 正常， FALSE: 異常または停止)	出力変数	BOOL
d(Q)	出力	出力変数	BOOL

機能

■演算処理

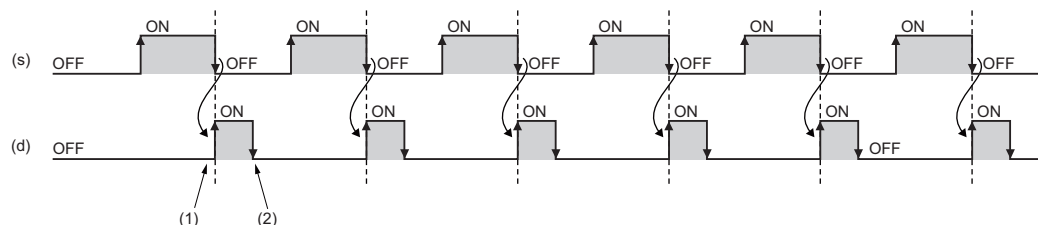
(s)がOFFすると、(d)を1スキャンのみONします。

■演算結果

1. EN/ENOなしファンクションブロック

演算処理を実行します。(d)からは演算出力値が出力されます。

・ タイミングチャート



(1) (s)の立下りで(d)をONします。

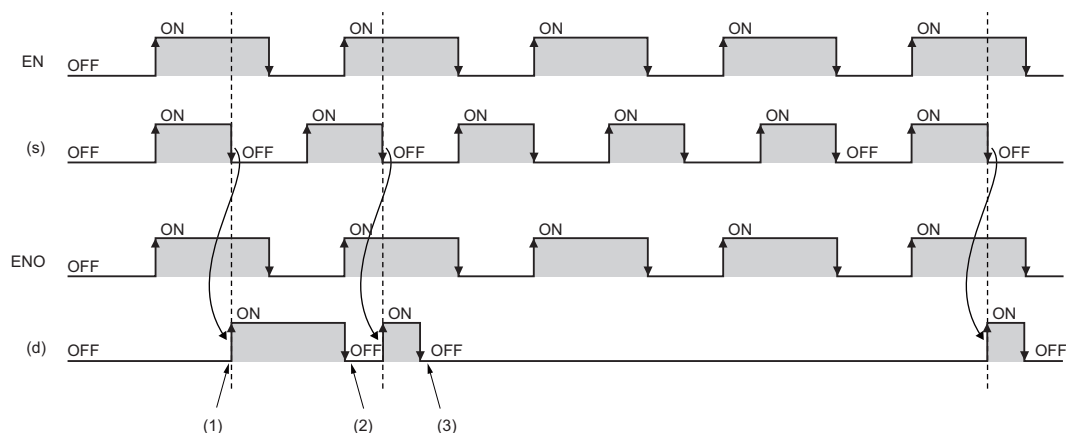
(2) 次スキャンで(d)をOFFします。

2. EN/ENO付きファンクションブロック

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE	前回出力値

・ タイミングチャート



(1) EN=ONかつ(s)の立下りで(d)をONします。

(2) 次スキャンで(d)をOFFします。

(3) EN=OFFの場合は、(d)は前回のスキャンの出力を保持します。

エラー

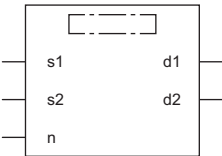
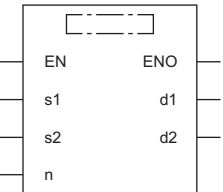
演算エラーはありません。

38 カウンタ/タイマファンクションブロック

38.1 アップカウンタ

CTU(_E)

信号の立ち上がり回数をカウントアップします。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] インスタンス名 (CU:=s1,R:=s2,PV:=n,Q:=d1,CV:=d2); [EN/ENO付き] インスタンス名 (EN:=en,ENO:=eno,CU:=s1,R:=s2,PV:=n,Q:=d1,CV:=d2);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(CU)	カウント信号入力	入力変数	BOOL
s2(R)	カウント値リセット	入力変数	BOOL
n(PV)	カウント最大値	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常または停止)	出力変数	BOOL
d1(Q)	カウント完了	出力変数	BOOL
d2(CV)	カウント値	出力変数	INT

機能

■演算処理

1. カウントアップ

- ・ (s1)がOFF→ONすると, (d2)を加算カウント(+1)します。
- ・ (d2)がカウンタの(n)に達すると(d1)がONし, 加算カウントはストップします。
- ・ (n)はカウンタの最大値を設定します。(s2)をONすると, (d1)はOFFし, (d2)が0にセットされます。

2. カウント最大値

(n)の有効設定範囲は, 0~32767です。

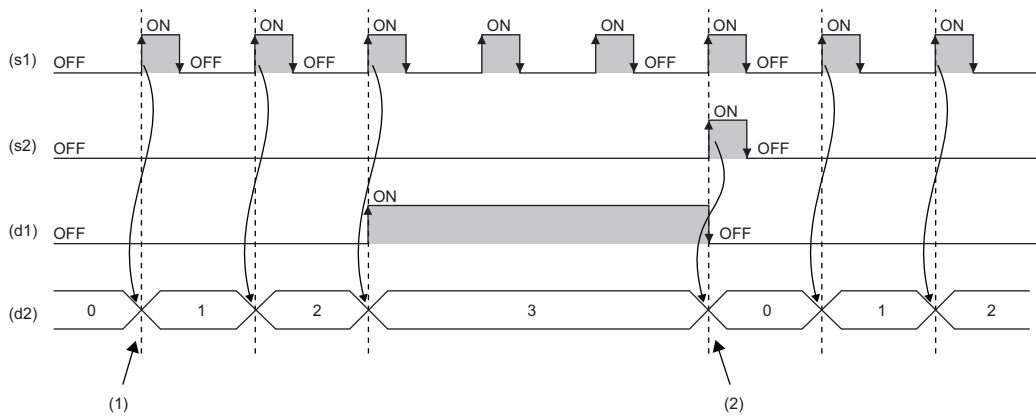
■演算結果

1. EN/ENOなしファンクションブロック

演算処理を実行します。(d1), (d2)からは演算出力値が出力されます。

- ・タイミングチャート

n=3の場合



(1) (s1)のOFF→ONで(d2)をカウントアップします。

(2) (s2)のOFF→ONで(d2)を初期化します。

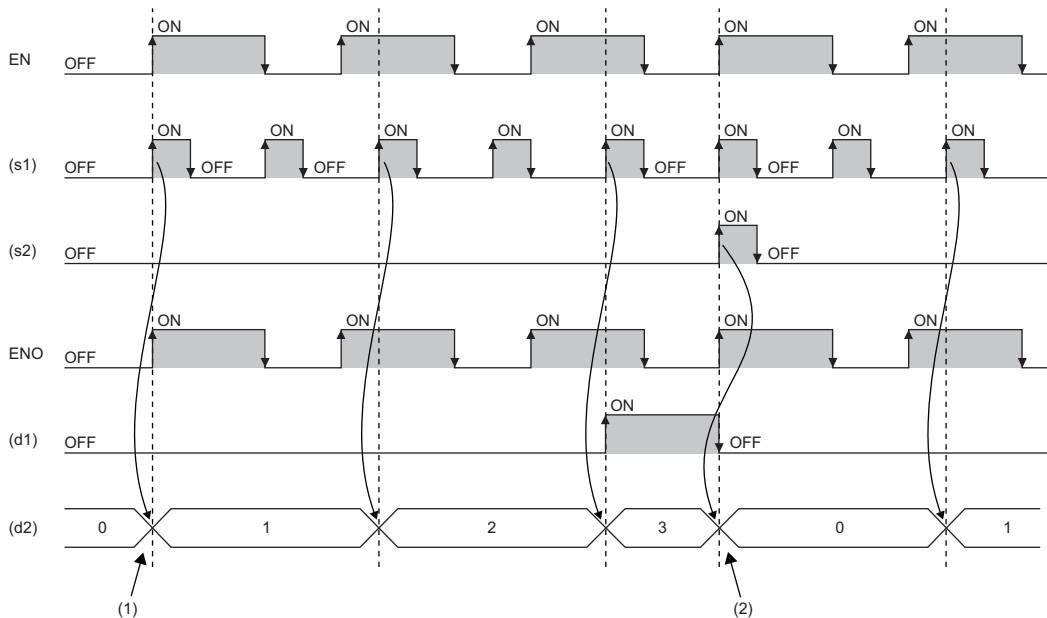
2. EN/ENO付きファンクションブロック

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d1), (d2)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE	前回出力値

- ・タイミングチャート

n=3の場合



(1) EN=ONかつ(s1)のOFF→ONで(d2)をカウントアップします。

(2) EN=ONかつ(s2)のOFF→ONで(d2)を初期化します。

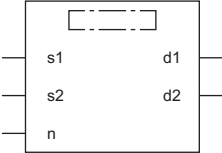
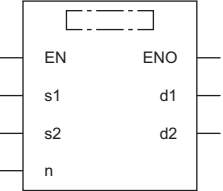
エラー

演算エラーはありません。

38.2 ダウンカウンタ

CTD(_E)

信号の立ち上がり回数をカウントダウンします。

ラダー， FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] インスタンス名(CD:=s1,LD:=s2,PV:=n,Q:=d1,CV:=d2); [EN/ENO付き] インスタンス名(EN:=en,ENO:=eno,CD:=s1,LD:=s2,PV:=n,Q:=d1,CV:=d2);
		

設定データ

■内容，種別，データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(CD)	カウント信号入力	入力変数	BOOL
s2(LD)	カウント値セット	入力変数	BOOL
n(PV)	カウント開始値	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常または停止)	出力変数	BOOL
d1(Q)	カウント完了	出力変数	BOOL
d2(CV)	カウント値	出力変数	INT

機能

■演算処理

1. カウントダウン

- (s1)がOFF→ONすると，(d2)を減算カウント(-1)します。
- (d2)が0の場合，(d1)がONし，減算カウントはストップします。
- (n)はカウント開始値を設定します。(s2)をONすると，(d1)はOFFし，(n)が(d2)にセットされます。

2. カウント開始値

(n)の有効設定範囲は，0~32767です。

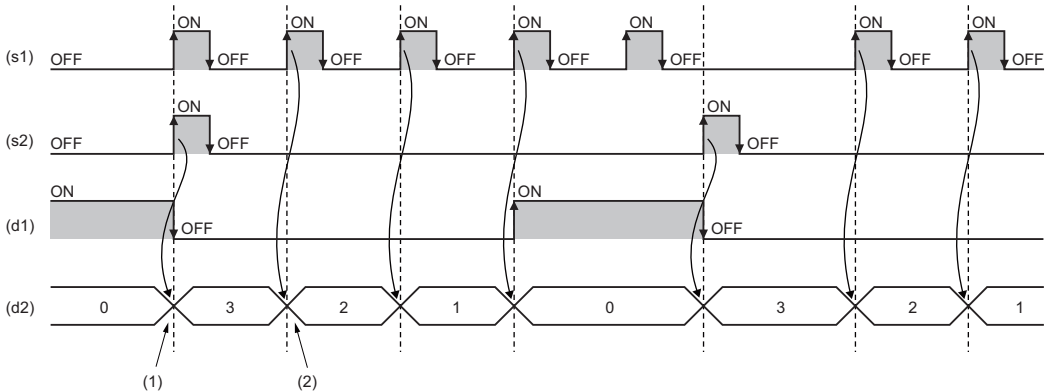
■演算結果

1. EN/ENOなしファンクションブロック

演算処理を実行します。(d1), (d2)からは演算出力値が出力されます。

・ タイミングチャート

n=3の場合



(1) (s2)のOFF→ONで(d2)を初期化します。

(2) (s1)のOFF→ONで(d2)をカウントダウンします。

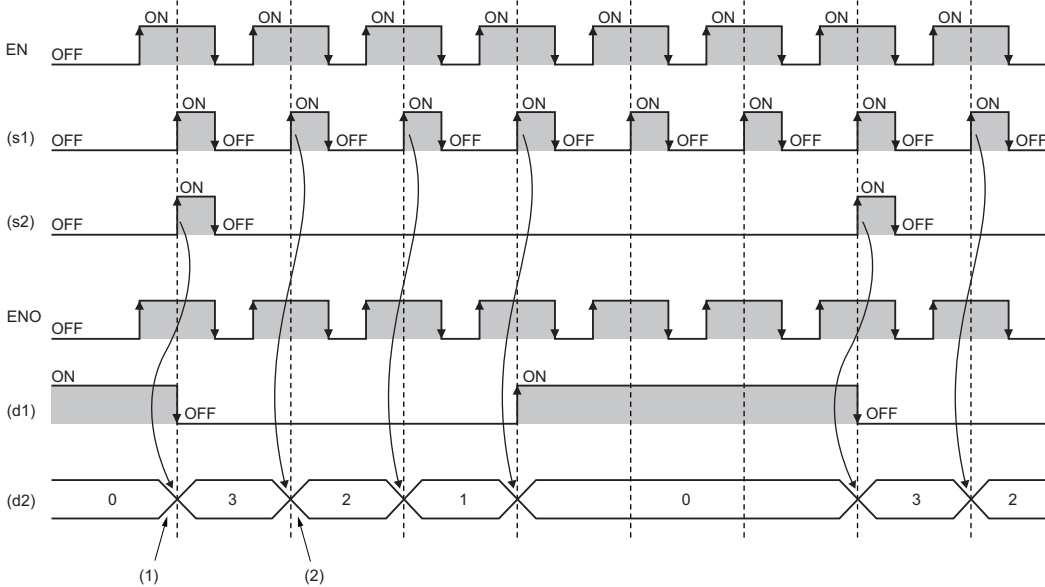
2. EN/ENO付きファンクションブロック

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d1), (d2)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE	前回出力値

・ タイミングチャート

n=3の場合



(1) EN=ONかつ(s2)のOFF→ONで(d2)を初期化します。

(2) EN=ONかつ(s1)のOFF→ONで(d2)をカウントダウンします。

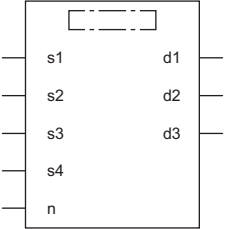
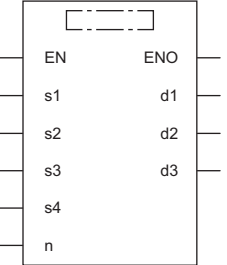
エラー

演算エラーはありません。

38.3 アップダウンカウンタ

CTUD(_E)

信号の立ち上がり回数をカウントアップ/カウントダウンします。

ラダー, FBD/LD		ST
[EN/ENOなし]		[EN/ENOなし] インスタンス名(CU:=s1,CD:=s2,R:=s3,LD:=s4,PV:=n,QU:=d1,QD:=d2,CV:=d3); [EN/ENO付き] インスタンス名 (EN:=en,ENO:=eno,CU:=s1,CD:=s2,R:=s3,LD:=s4,PV:=n,QU:=d1,QD:=d2,CV:=d3);
		

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s1(CU)	カウントアップ信号入力	入力変数	BOOL
s2(CD)	カウントダウン信号入力	入力変数	BOOL
s3(R)	カウント値リセット	入力変数	BOOL
s4(LD)	カウント値セット	入力変数	BOOL
n(PV)	カウント最大値/開始値	入力変数	INT
ENO	出力状態(TRUE: 正常, FALSE: 異常または停止)	出力変数	BOOL
d1(QU)	カウントアップ完了	出力変数	BOOL
d2(QD)	カウントダウン完了	出力変数	BOOL
d3(CV)	現在カウント値	出力変数	INT

■演算処理

(n)には、加算カウントの場合はカウント最大値を設定し、減算カウントの場合はカウント開始値を設定します。(n)の有効範囲は、0~32767です。

1. カウントアップ

- (s1)がOFF→ONすると、(d3)を加算カウント(+1)します。
- (d3)が(n)に達すると(d1)がONし、加算カウントはストップします。
- (s3)をONすると、(d1)はOFFし、(d3)が0にセットされます。

2. カウントダウン

- (s2)がOFF→ONすると、(d3)を減算カウント(-1)します。
- (d3)が0の場合、(d2)がONし、減算カウントはストップします。
- (s4)をONすると、(d2)はOFFし、(n)が(d3)にセットされます。

3. その他

- (s1), (s2)が同時にOFF→ONすると、(s1)が優先され(d3)を加算カウント(+1)します。
- (s3), (s4)を同時にONすると、(s3)が優先され(d3)を0にセットします。

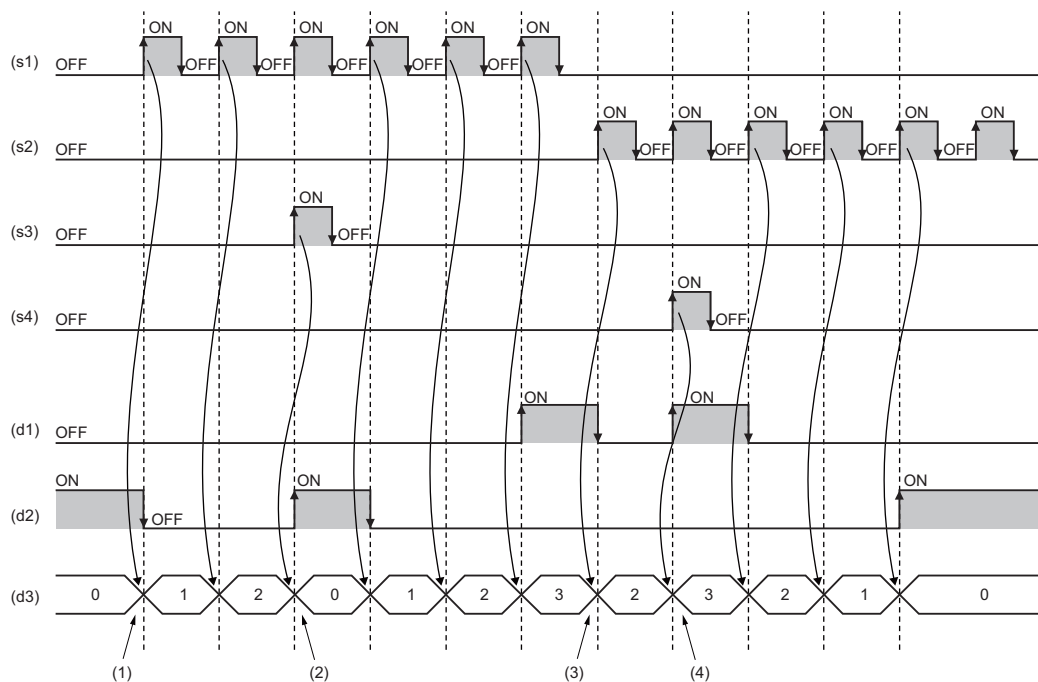
■演算結果

1. EN/ENOなしファンクションブロック

演算処理を実行します。(d1), (d2), (d3)からは演算出力値が出力されます。

- タイミングチャート

n=3の場合



- (1) (s1)のOFF→ONで(d3)をカウントアップします。
- (2) (s3)のOFF→ONで(d3)を初期化します。
- (3) (s2)のOFF→ONで(d3)をカウントダウンします。
- (4) (s4)のOFF→ONで(d3)を初期化します。

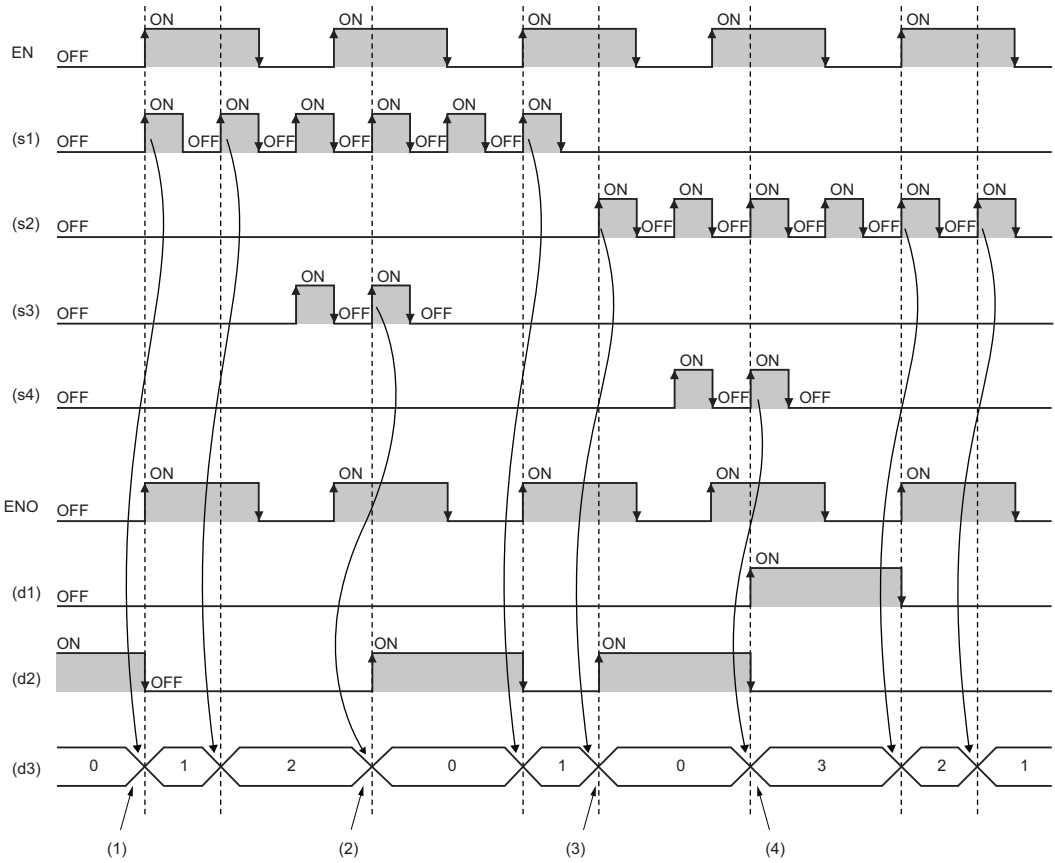
2. EN/ENO付きファンクションブロック

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d1), (d2), (d3)
TRUE(演算実行)	TRUE	演算出力値
FALSE(演算停止)	FALSE	前回出力値

・ タイミングチャート

n=3の場合



- (1) EN=ONかつ(s1)のOFF→ONで(d3)をカウントアップします。
- (2) EN=ONかつ(s3)のOFF→ONで(d3)を初期化します。
- (3) EN=ONかつ(s2)のOFF→ONで(d3)をカウントダウンします。
- (4) EN=ONかつ(s4)のOFF→ONで(d3)を初期化します。

エラー

演算エラーはありません。

38.4 カウンタファンクションブロック

COUNTER_FB_M

実行条件が成立すると、カウントアップを実行します。

ラダー, FBD/LD	ST
	インスタンス名(Coil:=s1,Preset:=s2,ValueIn:=s3,ValueOut:=d1,Status:=d2);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
s1(Coil)	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s2(Preset)	カウンタ設定値	入力変数	INT
s3(ValueIn)	カウンタ初期値	入力変数	INT
d1(ValueOut)	カウンタ現在値	出力変数	ANY16
d2(Status)	出力	出力変数	BOOL

機能

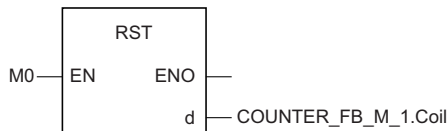
■演算処理

- (s1)の立上がり(OFF→ON)を検出してカウントします。(s1)がONのままではカウントされません。カウントは(s3)の値から開始し、(s2)の値になると、(d2)がONします。現在のカウンタ値は(d1)に格納されます。
- (s2)には0~32767の値が指定可能です。
- (s3)には-32768~32767の値が指定可能です。ただし、負の値を指定した場合0が初期値となります。
- カウンタの現在値(d1)をリセットしたい場合は、FBの(s1)を直接リセットしてください。

例

ラベル名: COUNTER_FB_M_1の場合

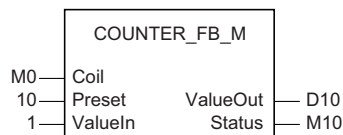
[ラダープログラム]



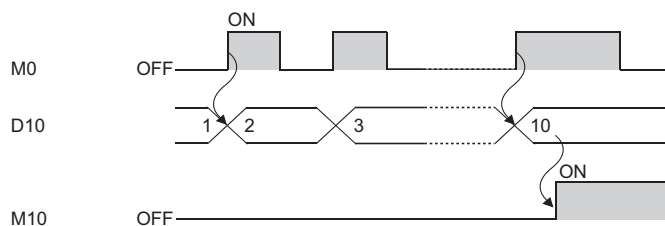
[STプログラム]

RST(M0, COUNTER_FB_M_1.Coil)

[回路例]



[タイミングチャート]



エラー

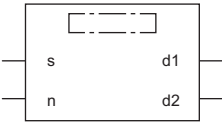
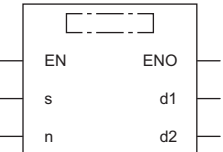
演算エラーはありません。

38.5 パルスタイマ

38

TP(_E)

指定された時間の間、信号をONします。

ラダー, FBD/LD		ST
[EN/ENOなし]	[EN/ENO付き]	[EN/ENOなし] インスタンス名(IN:=s,PT:=n,Q:=d1,ET:=d2); [EN/ENO付き] インスタンス名(EN:=en,ENO:=eno,IN:=s,PT:=n,Q:=d1,ET:=d2);
		

設定データ

■内容、種別、データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	出力開始	入力変数	BOOL
n(PT)	出力時間設定値	入力変数	TIME
ENO	出力状態(TRUE: 正常, FALSE: 異常または停止)	出力変数	BOOL
d1(Q)	出力	出力変数	BOOL
d2(ET)	経過時間	出力変数	TIME

機能

■演算処理

1. 出力

- ・(s)がONすると、(n)で設定された時間の間、(d1)をONします。(d2)には(d1)がONしてからの経過時間がセットされます。
- ・経過時間のカウントにはロングタイマを使用します。

2. 出力の終了

- ・経過時間が設定時間に到達すると(d1)をOFFします。
- ・(d1)がOFF後、(s)がOFFの場合は経過時間をリセットします。
- ・(d1)がONの時に(s)がOFFしても、(d1)はOFFされません。

3. 出力時間の設定

(n)の有効設定範囲はT#1ms~T#2147483msです。ただし、エンジニアリングツールのタイマ時限設定を変更することによって、有効設定範囲は下記となります。

最小値	最大値
タイマ時限のロングタイマ設定値[ms]と同じです。 ただし、ロングタイマ設定値が1ms未満の場合、1msとなります。	下記の条件を満たす値となります。 ただし、出力時間設定値は時間型(32ビット値)のため、最大値は時間型の範囲に収まる値となります。 ・出力時間設定値[ms]≤2147483647[ms]×タイマ時限のロングタイマ設定値[ms] [例] ・タイマ時限のロングタイマ設定値が0.001msの場合:T#1ms~T#2147483ms ・タイマ時限のロングタイマ設定値が1000msの場合:T#1000ms~T#2147483000ms

(n)の設定値は、(d1)がOFF→ON(立上り)となったときの値が使用されます。(d1)がONの時に(n)の値を変更した場合、変更した値が有効になるのは次回出力開始時です。

■演算結果

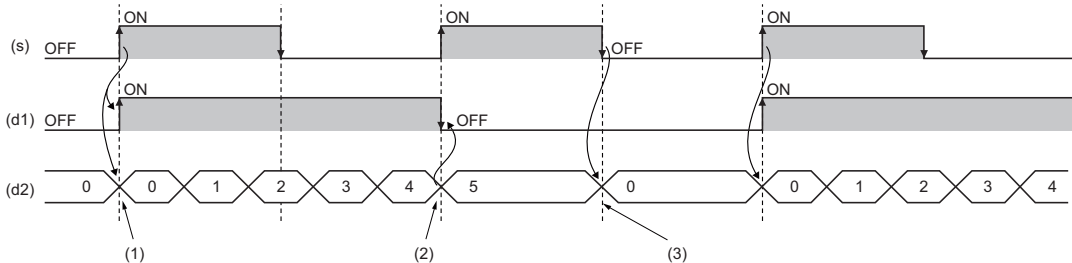
1. EN/ENOなしファンクションブロック

演算結果は下記になります。

演算結果	(d1), (d2)
演算エラーなし	演算出力値
演算エラーあり	不定値

・ タイミングチャート

n=T#5s(5秒)の場合



- (1) (s)のOFF→ONで(d1)をONします。(s)のOFF→ONで(d2)の時間計測を開始します。
- (2) (d2)が(n)で指定した時間に到達すると、(d1)をOFFします。
- (3) (s)=OFFかつ(d1)=OFFにて(d2)を初期化します。

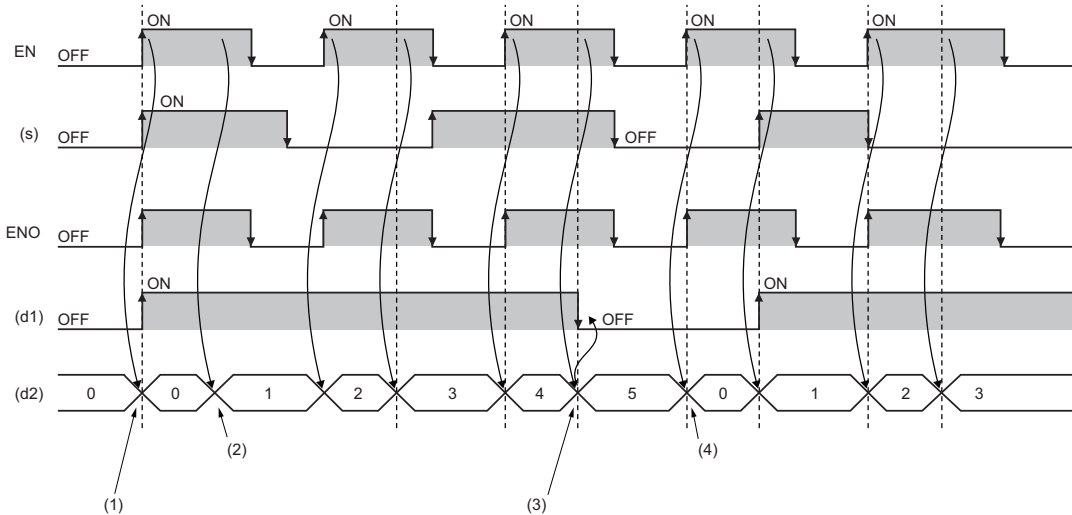
2. EN/ENO付きファンクションブロック

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d1), (d2)
TRUE(演算実行)	TRUE(演算エラーなし)	演算出力値
	FALSE(演算エラーあり)	不定値
FALSE(演算停止)	FALSE	前回出力値

・ タイミングチャート

n=T#5s(5秒)の場合



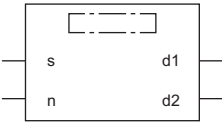
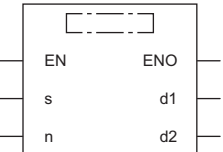
- (1) EN=ONかつ(s)のOFF→ONで(d1)をONします。EN=ONかつ(s)のOFF→ONで(d2)の時間計測を開始します。
- (2) 計測開始後は、EN=ONで計測時間がアップします。
- (3) (d2)が(n)で指定した時間に到達すると、(d1)をOFFします。
- (4) EN=ONかつ(s)=OFFかつ(d1)=OFFにて(d2)を初期化します。

エラーコード (SD0)	内容
3281H	出力時間設定値が有効な範囲を超えているとき。

38.6 オンディレイタイマ

TON(_E)

指定された時間の後、信号をONします。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] インスタンス名(IN:=s,PT:=n,Q:=d1,ET:=d2); [EN/ENO付き] インスタンス名(EN:=en,ENO:=eno,IN:=s,PT:=n,Q:=d1,ET:=d2);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	時間計測	入力変数	BOOL
n(PT)	遅延時間設定値	入力変数	TIME
ENO	出力状態(TRUE: 正常, FALSE: 異常または停止)	出力変数	BOOL
d1(Q)	出力	出力変数	BOOL
d2(ET)	経過時間	出力変数	TIME

機能

■演算処理

1. 出力

- ・ (s)がONすると, (n)で設定された時間の後, (d1)をONします。(d2)には(d1)がONしてからの遅延経過時間がセットされます。
- ・ (s)がOFFすると(d1)をOFFし遅延経過時間もリセットします。
- ・ 経過時間のカウンタにはロングタイマを使用します。

2. 遅延時間の設定

(n)の有効設定範囲はT#1ms~T#2147483msです。ただし, エンジニアリングツールのタイマ時限設定を変更することによって, 有効設定範囲は下記のとおりとなります。

最小値	最大値
タイマ時限のロングタイマ設定値[ms]と同じです。 ただし, ロングタイマ設定値が1ms未満の場合, 1msとなります。	下記の条件を満たす値となります。 ただし, 遅延時間設定値は時間型(32ビット値)のため, 最大値は時間型の範囲に収まる値となります。 ・ 遅延時間設定値[ms]≤2147483647[ms]×タイマ時限のロングタイマ設定値[ms] [例] ・ タイマ時限のロングタイマ設定値が0.001msの場合:T#1ms~T#2147483ms ・ タイマ時限のロングタイマ設定値が1000msの場合:T#1000ms~T#2147483000ms

(n)の設定値は, (d)がOFF→ON(立上り)となったときの値が使用されます。(s)がONの時に(n)の値を変更した場合, 変更した値が有効になるのは, (s)の次回立ち上がり時です。

■演算結果

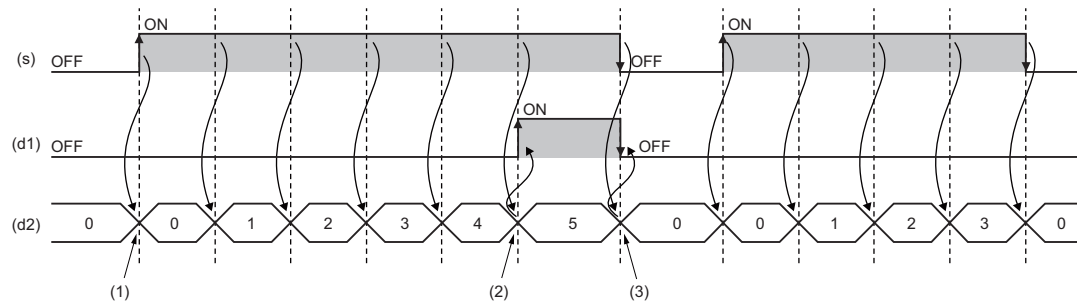
1. EN/ENOなしファンクションブロック

演算結果は下記になります。

演算結果	(d1), (d2)
演算エラーなし	演算出力値
演算エラーあり	不定値

・ タイミングチャート

n=T#5s(5秒)の場合



- (1) (s)のOFF→ONで(d2)の時間計測を開始します。
- (2) (d2)が(n)で指定した時間に到達すると、(d1)をONします。
- (3) (s)のON→OFFかつ(d1)のON→OFFで(d2)を初期化します。

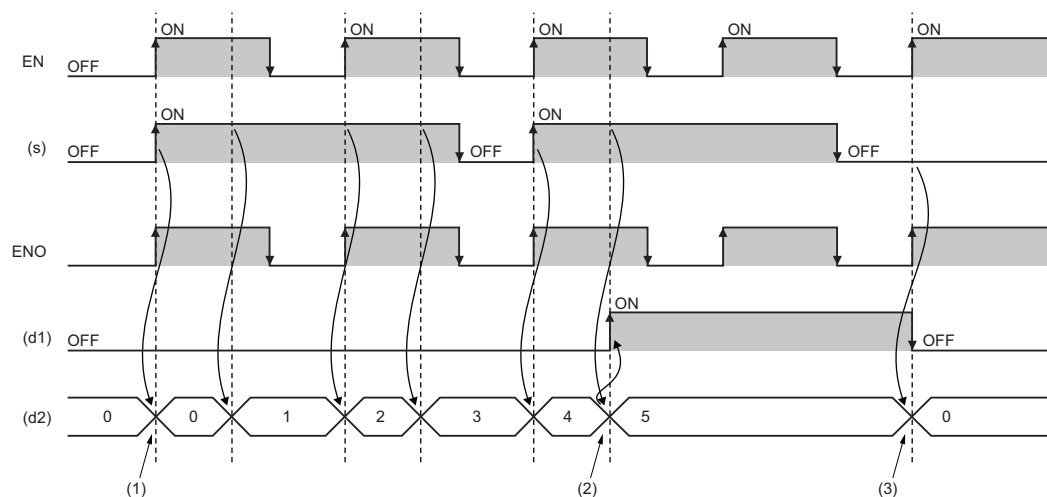
2. EN/ENO付きファンクションブロック

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d1), (d2)
TRUE(演算実行)	TRUE(演算エラーなし)	演算出力値
	FALSE(演算エラーあり)	前回出力値
FALSE(演算停止)	FALSE	前回出力値

・ タイミングチャート

n=T#5s(5秒)の場合



- (1) EN=ONかつ(s)のOFF→ONで(d2)の時間計測を開始します。
- (2) (d2)が(n)で指定した時間に到達すると、(d1)をONします。
- (3) EN=ONかつ(s)のON→OFFかつ(d1)のON→OFFで(d2)を初期化します。

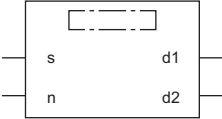
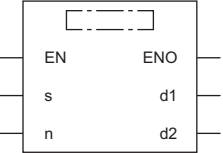
エラー

エラーコード (SD0)	内容
3281H	出力時間設定値が有効な範囲を超えているとき。

38.7 オフディレイタイマ

TOF(_E)

指定された時間の後、信号をOFFします。

ラダー, FBD/LD		ST
[EN/ENOなし] 	[EN/ENO付き] 	[EN/ENOなし] インスタンス名(IN:=s,PT:=n,Q:=d1,ET:=d2); [EN/ENO付き] インスタンス名(EN:=en,ENO:=eno,IN:=s,PT:=n,Q:=d1,ET:=d2);

設定データ

■内容, 種別, データ型

引数	内容	種別	データ型
EN	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s(IN)	時間計測	入力変数	BOOL
n(PT)	遅延時間設定値	入力変数	TIME
ENO	出力状態(TRUE: 正常, FALSE: 異常または停止)	出力変数	BOOL
d1(Q)	出力	出力変数	BOOL
d2(ET)	経過時間	出力変数	TIME

機能

■演算処理

1. 出力

- (s)がONすると, (d1)をONします。
- (s)がON→OFFすると, (n)で設定された時間の後, (d1)をOFFします。(d2)には(d1)がOFFするまでの経過時間がセットされます。
- 経過時間のカウントにはロングタイマを使用します。

2. 遅延時間の設定

(n)の有効設定範囲はT#1ms~T#2147483msです。ただし、エンジニアリングツールのタイマ時限設定を変更することによって、有効設定範囲は下記のとおりとなります。

最小値	最大値
タイマ時限のロングタイマ設定値[ms]と同じです。 ただし、ロングタイマ設定値が1ms未満の場合、1msとなります。	下記の条件を満たす値となります。 ただし、遅延時間設定値は時間型(32ビット値)のため、最大値は時間型の範囲に収まる値となります。 • 遅延時間設定値[ms]≤2147483647[ms]×タイマ時限のロングタイマ設定値[ms] [例] • タイマ時限のロングタイマ設定値が0.001msの場合:T#1ms~T#2147483ms • タイマ時限のロングタイマ設定値が1000msの場合:T#1000ms~T#2147483000ms

(n)の設定値は, (s)がON→OFF(立下り)となったときの値が使用されます。(s)がOFFの時に(n)の値を変更した場合、変更した値が有効になるのは, (s)の次回立ち下がり時です。

■演算結果

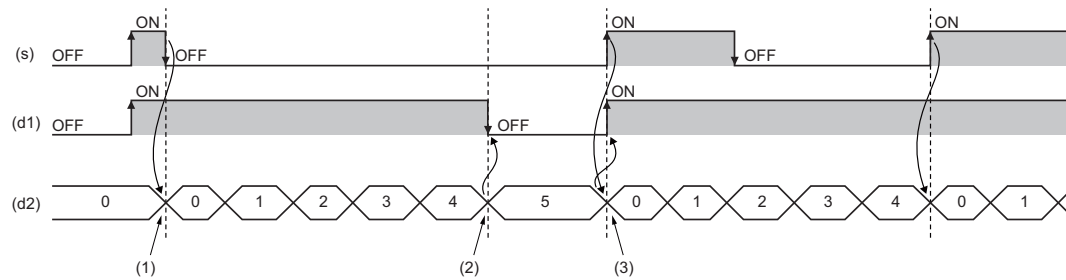
1. EN/ENOなしファンクションブロック

演算結果は下記になります。

演算結果	(d1), (d2)
演算エラーなし	演算出力値
演算エラーあり	不定値

・ タイミングチャート

n=T#5s(5秒)の場合



(1) (s)のON→OFFで(d2)の時間計測を開始します。

(2) (d2)が(n)で指定した時間に到達すると、(d1)をONします。

(3) (s)のOFF→ONで(d2)を初期化します。

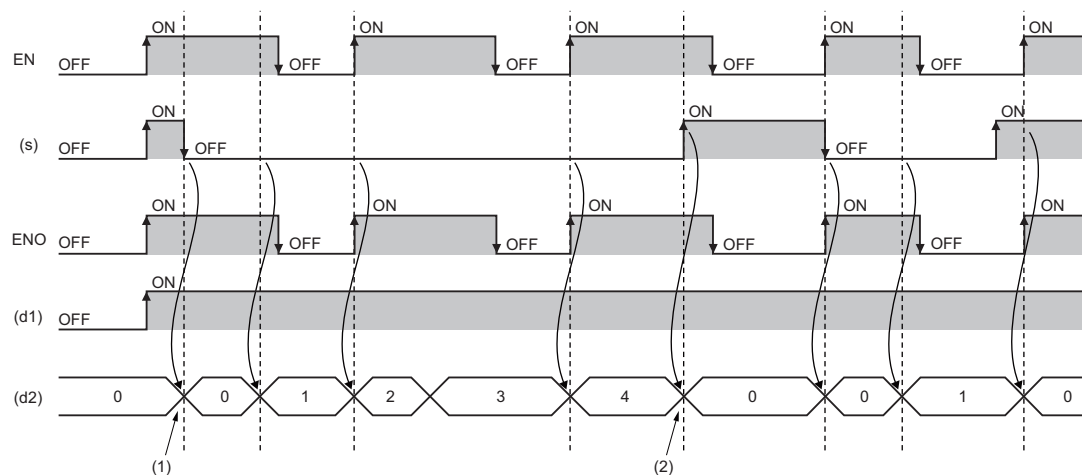
2. EN/ENO付きファンクションブロック

実行条件と演算結果は下記になります。

実行条件	演算結果	
EN	ENO	(d)
TRUE(演算実行)	TRUE(演算エラーなし)	演算出力値
	FALSE(演算エラーあり)	前回出力値
FALSE(演算停止)	FALSE	前回出力値

・ タイミングチャート

n=T#5s(5秒)の場合



(1) EN=ONかつ(s)のON→OFFで(d2)の時間計測を開始します。

(2) EN=ONかつ(s)のOFF→ONで(d2)を初期化します。

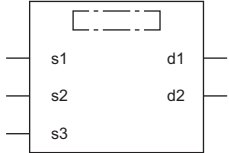
エラー

エラーコード (SD0)	内容
3281H	出力時間設定値が有効な範囲を超えているとき。

38.8 タイマファンクションブロック

TIMER_□_M

実行条件が成立すると、設定された時間までタイマカウントを実行します。

ラダー， FBD/LD	ST
 (□には、TIMER_10_FB_M, TIMER_100_FB_M, TIMER_HIGH_FB_M, TIMER_LOW_FB_M, TIMER_CONT_FB_M, TIMER_CONTHFB_Mが入ります。)	インスタンス名(Coil:=s1,Preset:=s2,ValueIn:=s3,ValueOut:=d1,Status:=d2);

設定データ

■内容，種別，データ型

引数	内容	種別	データ型
s1(Coil)	実行条件(TRUE: 実行, FALSE: 停止)	入力変数	BOOL
s2(Preset)	タイマ設定値	入力変数	INT
s3(ValueIn)	タイマ初期値	入力変数	INT
d1(ValueOut)	タイマ現在値	出力変数	ANY16
d2(Status)	出力	出力変数	BOOL

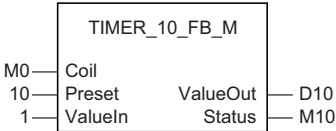
機能

■TIMER_10_FB_M

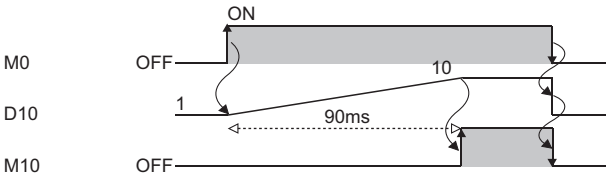
- (s1)の実行条件がONすると、現在値の計測を開始します。(s3)×10msから計測を開始し、(s2)×10msまで計測値が到達すると(d2)がONします。現在の計測値は(d1)に出力されます。
- (s1)の実行条件がOFFすると、現在値は(s3)の値となり、(d2)もOFFします。
- 高速タイマの計測単位(タイマ時限設定)をエンジニアリングツールのデフォルト値から変更した場合、変換/全変換時にワーニングが発生します。
- (s2)には0~32767の値が指定可能です。
- (s3)には-32768~32767の値が指定可能です。ただし、負の値を指定した場合、0が初期値となります。

例

[回路例]



[タイミングチャート]

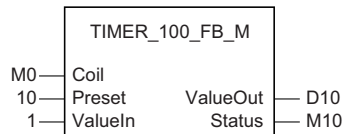


■TIMER_100_FB_M

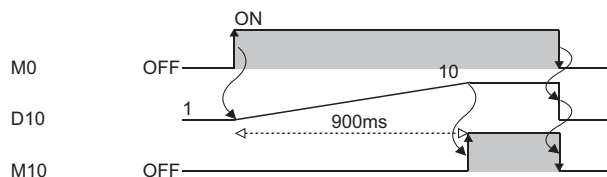
- (s1)の実行条件がONすると、現在値の計測を開始します。(s3)×100msから計測を開始し、(s2)×100msまで計測値が到達すると(d2)がONします。現在の計測値は(d1)に出力されます。
- (s1)の実行条件がOFFすると、現在値は(s3)の値となり、(d2)もOFFします。
- 低速タイマの計測単位(タイマ時限設定)をエンジニアリングツールのデフォルト値から変更した場合、変換/全変換時にワーニングが発生します。
- (s2)には0~32767の値が指定可能です。
- (s3)には-32768~32767の値が指定可能です。ただし、負の値を指定した場合、0が初期値となります。

例

[回路例]



[タイミングチャート]

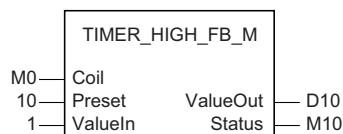


■TIMER_HIGH_FB_M

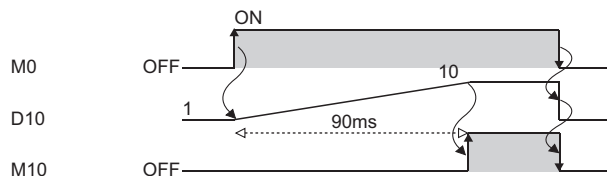
- 計測単位が0.1~100msの高速タイマです。(s1)の実行条件がONすると、現在値の計測を開始します。(s3)×0.1~100ms(可変。パラメータで設定)から計測を開始し、(s2)×0.1~100msまで計測値が到達すると(d2)がONします。現在の計測値は(d1)に出力されます。
- (s1)の実行条件がOFFすると、現在値は(s3)の値となり、(d2)もOFFします。
- 高速タイマの計測単位(タイマ時限設定)は、デフォルト値が10msです。計測単位は0.01ms~100msの範囲で変更できます。
- (s2)には0~32767の値が指定可能です。
- (s3)には-32768~32767の値が指定可能です。ただし、負の値を指定した場合、0が初期値となります。

例

[回路例]



[タイミングチャート]

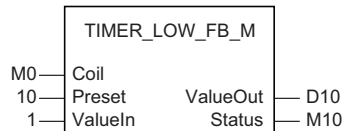


■TIMER_LOW_FB_M

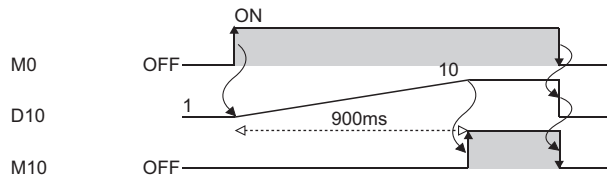
- 計測単位が1~1000msの低速タイマです。(s1)の実行条件がONすると、現在値の計測を開始します。(s3)×1~1000ms(可変。パラメータで設定)から計測を開始し、(s2)×1~1000msまで計測値が到達すると(d2)がONします。現在の計測値は(d1)に出力されます。
- (s1)の実行条件がOFFすると、現在値は(s3)の値となり、(d2)もOFFします。
- 低速タイマの計測単位(タイマ時限設定)は、デフォルト値が100msです。計測単位は1~1000msで、1ms単位で変更できます。
- (s2)には0~32767の値が指定可能です。
- (s3)には-32768~32767の値が指定可能です。ただし、負の値を指定した場合、0が初期値となります。

例

[回路例]



[タイミングチャート]



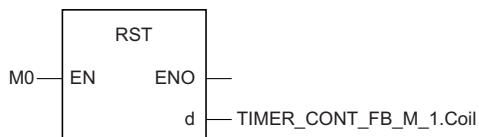
■TIMER_CONT_FB_M/TIMER_CONTHFB_M

- 変数がONしている時間を計測する積算タイマです。(s1)の実行条件がONすると、現在値の計測を開始します。積算タイマには低速積算タイマ(TIMER_CONT_FB_M)と高速積算タイマ(TIMER_CONTHFB_M)の2種類があります。
- (s3)×1~1000ms(高速積算タイマ時は0.1~100ms)(可変。パラメータで設定)から計測を開始し、(s2)×1~1000ms(高速積算タイマ時は0.1~100ms)まで計測値が到達すると(d2)がONします。現在の計測値は(d1)に出力されます。
- (s1)の実行条件がOFFしても、(d1)、(d2)のON/OFF状態を保持します。再度(s1)の実行条件がONすると、保持していた計測値から計測を再開します。
- 積算タイマの計測単位(時限)は、低速タイマ(TIMER_LOW_FB_M)、高速タイマ(TIMER_HIGH_FB_M)と同一です。
 - 低速積算タイマ:低速タイマ
 - 高速積算タイマ:高速タイマ
- (s2)には0~32767の値が指定可能です。
- (s3)には-32768~32767の値が指定可能です。ただし、負の値を指定した場合、0が初期値となります。
- 積算タイマの(d1)をリセットしたい場合は、FBの(s1)を直接リセットしてください。

例

ラベル名:TIMER_CONT_FB_M_1の場合

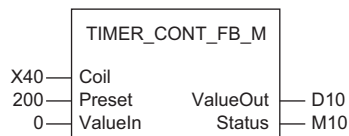
[ラダープログラム]



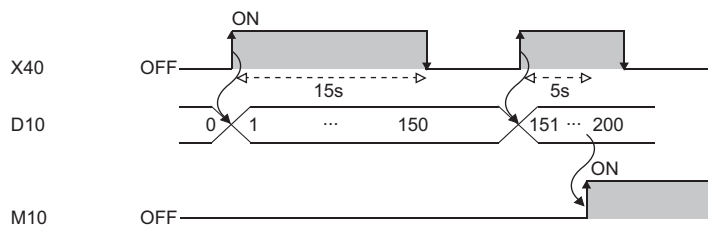
[STプログラム]

RST(M0,TIMER_CONT_FB_M_1.Coil)

[回路例]



[タイミングチャート]



エラー

演算エラーはありません。

MEMO

第8部

モーション制御ファンク
ションブロック

この部は下記の章構成となります。

39 モーション制御FBの概要

40 モーション制御ファンクションブロックの変数一覧

41 管理系のファンクションブロック

42 動作系のファンクションブロック

43 一般ファンクションブロック

39 モーション制御FBの概要

モーション制御FBの使用方法，仕様について説明します。
モーション制御FBには，PLCopen®で定められたFBを含みます。入出力信号の基本的な仕様は，PLCopen®のモーション制御FBに準じます。
ファンクションブロックの共通仕様は下記を参照してください。
📖 65ページ ファンクションブロック(FB)

39.1 モーション制御FB

モーション制御FBについて説明します。

モーション制御FBの使用方法

モーション制御FBの使用手順を下記に示します。

1. エンジニアリングツールにてモーション制御FBのインスタンス(グローバルラベル，またはローカルラベル)を作成します。
2. 作成したモーション制御FBのインスタンスに対し，入出力引数を設定するプログラムを作成します。
3. モーション制御FBの実行指令(Execute)，または有効(Enable)をTRUEにすると，制御を実行します。

Point🔍

- 1つのモーション制御FBインスタンスを実行中(バッファリング中を含む)に，軸入力(Axis_REF入力，またはAxis_GROUP_REF入力)を変更して複数の軸で使い回すことはできません。モーション制御FBの軸入力は非実行時のみ変更できます。実行中に軸入力を変更すると警告が発生し，変更は無視します。そのため，同時に制御する軸の数だけ，モーション制御FBのインスタンスを作成する必要があります。
- 1つのモーション制御FBインスタンスを複数プログラム(割込みプログラム含む並列に実行されるもの)から同時に呼出さないでください。同時に呼出した場合の動作は保証できません。

システム状態ごとのモーション制御FBの動作

モーション制御FBはRUN中のみ実行できます。

○: 可， ×: 不可

システムの状態	動作可否
STOP中	×
RUN中	○
中度異常中	×
重度異常中	×

モーション制御FBの種類

モーション制御FBは動作内容や実行方法により分類があります。

管理系FB/動作系FB/一般FB

モーション制御FBは動作内容によって、下記の種類があります。

種類	動作内容
管理系FB	軸、または軸グループを引数にとり、実行によって軸状態、または軸グループ状態が変化しないモーション制御FBです。(一部例外あり) - 基本的に同一の軸、または軸グループに対し複数のインスタンスを同時に実行できます。 - 同一の軸または軸グループに対して複数のインスタンスを同時に実行した場合、モーション制御FBの実行優先度が高い方から優先的に制御します。
動作系FB	軸、または軸グループを引数にとり、実行によって軸状態、または軸グループ状態が変化するモーション制御FBです。 - 基本的に同一の軸、または軸グループに対し1つのみ実行できます。ただし、モーション制御FBによっては、並列に実行できるものもあります。 - 動作系FBを実行中に管理系FBを実行しても、軸状態、または軸グループ状態は基本的に変化しません。ただし、モーション制御FBによっては、特定の状態遷移を引き起こすものがあります。 - 同一の軸または軸グループに対して複数のインスタンスを同時に実行した場合、モーション制御FBの実行優先度が高い方から優先的に制御します。
一般FB	軸、または軸グループを引数にとらないモーション制御FBです。 - 複数のインスタンスを同時に実行できます。軸と関連しないため、動作系FBや管理系FBとも相互に影響しません。

モーション制御FBの実行優先度について、下記に示します。

一: 優先度の割当てなし

優先度	モーション制御FB	
高 ↑ ↓ 低	1	• MC_Stop(強制停止) • MC_GroupStop(グループ強制停止) • MCv_MoveWait(軸制御開始ウェイト) • MCv_GroupMoveWait(軸グループ制御開始ウェイト)
	2	• MC_MoveAbsolute(絶対値位置決め) • MC_MoveRelative(相対値位置決め) • MCv_Jog(JOG運転) • MC_MoveVelocity(速度制御) • MC_TorqueControl(トルク制御) • MCv_SpeedControl(速度制御(位置ループを含む)) • MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御) • MCv_MoveLinearInterpolateRelative(相対値直線補間制御) • MCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御) • MCv_MoveCircularInterpolateRelative(相対値円弧補間制御) • MC_SetPosition(現在位置変更) • MCv_SetTorqueLimit(トルク制限値) • MC_CamIn(カム動作開始) • MC_GearIn(ギア動作開始) • MC_CombineAxes(加減算位置決め) • MCv_BacklashCompensationFilter(バックラッシュ補正フィルタ) • MCv_SmoothingFilter(スムージングフィルタ) • MCv_DirectionFilter(移動方向制限フィルタ) • MCv_SpeedLimitFilter(速度制限フィルタ) • MCv_AdvancedSync(アドバンス同期制御) • MCv_MovePositioningData(多軸位置決めデータ運転) • MCv_PositionControl(絶対位置追従制御) • MCv_CyclicPosition(モーションサイクリック位置制御) • MCv_CyclicVelocity(モーションサイクリック速度制御) • MCv_CyclicTorque(モーションサイクリックトルク制御)
	3	• MC_TouchProbe(タッチプローブ有効) • MCv_ChangeCycle(1サイクル現在値変更)
	—	• MC_GroupEnable(軸グループ有効) • MC_GroupDisable(軸グループ無効) • MC_Power(運転可) • MC_SetOverride(オーバーライド値設定) • MC_ReadParameter(/パラメータ読出) • MC_WriteParameter(/パラメータ書込) • MC_Reset(軸エラーリセット) • MC_GroupReset(軸グループエラーリセット) • MC_AbortTrigger(タッチプローブ無効) • MC_CamTableSelect(カムテーブル選択) • MCv_ReadProfileData(プロファイル読出) • MCv_WriteProfileData(プロファイル書込) • MCv_AllPower(全軸運転可) • MC_GroupSetOverride(軸グループオーバーライド値設定) • MCv_MotionErrorReset(モーションエラーリセット) • MCv_SyncOperationCycles(制御演算周期同期) • MCv_AdvPositionPerCycleCalc(アドバンス同期制御1サイクル現在位置計算) • MCv_AdvCamSetPositionCalc(アドバンス同期制御カム指令現在位置計算) • MC_DigitalCamSwitch(デジタルカムスイッチ出力)

実行指令(Execute)型/有効(Enable)型

モーション制御FBは、実行指令(Execute)で実行するものと、有効(Enable)で実行するものがあります。

種類	実行指令(Execute)型	有効(Enable)型	その他の型
管理系FB	• MC_GroupEnable(軸グループ有効) • MC_GroupDisable(軸グループ無効) • MC_SetPosition(現在位置変更) • MCv_SetTorqueLimit(トルク制限値) • MC_WriteParameter(パラメータ書込) • MC_Reset(軸エラーリセット) • MC_GroupReset(軸グループエラーリセット) • MC_TouchProbe(タッチプローブ有効) • MC_AbortTrigger(タッチプローブ無効) • MC_CamTableSelect(カムテーブル選択) • MCv_ChangeCycle(1サイクル現在値変更) • MCv_MotionErrorReset(モーションエラーリセット) • MCv_AdvPositionPerCycleCalc(アドバンスト同期制御1サイクル現在位置計算) • MCv_AdvCamSetPositionCalc(アドバンスト同期制御カム指令現在位置計算)	• MC_Power(運転可) • MC_SetOverride(オーバライド値設定) • MC_ReadParameter(パラメータ読出) • MCv_AllPower(全軸運転可) • MC_GroupSetOverride(軸グループオーバライド値設定) • MCv_SyncOperationCycles(制御演算周期同期) • MC_DigitalCamSwitch(デジタルカムスイッチ出力)	—
動作系FB	• MC_Home(原点復帰) • MC_Stop(強制停止) • MC_GroupStop(グループ強制停止) • MC_MoveAbsolute(絶対値位置決め) • MC_MoveRelative(相対値位置決め) • MC_MoveVelocity(速度制御) • MC_TorqueControl(トルク制御) • MCv_SpeedControl(速度制御(位置ループを含む)) • MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御) • MCv_MoveLinearInterpolateRelative(相対値直線補間制御) • MCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御) • MCv_MoveCircularInterpolateRelative(相対値円弧補間制御) • MC_CamIn(カム動作開始) • MC_GearIn(ギア動作開始) • MC_CombineAxes(加減算位置決め) • MCv_MoveWait(軸制御開始ウェイト) • MCv_GroupMoveWait(軸グループ制御開始ウェイト) • MCv_MovePositioningData(多軸位置決めデータ運転) • MCv_PositionControl(絶対位置追従制御) • MCv_CyclicPosition(モーションサイクリック位置制御) • MCv_CyclicVelocity(モーションサイクリック速度制御) • MCv_CyclicTorque(モーションサイクリックトルク制御)	• MCv_BacklashCompensationFilter(バックラッシュ補正フィルタ) • MCv_SmoothingFilter(スムージングフィルタ) • MCv_DirectionFilter(移動方向制限フィルタ) • MCv_SpeedLimitFilter(速度制限フィルタ) • MCv_AdvancedSync(アドバンスト同期制御)	• MCv_Jog(JOG運転)
一般FB	• MCv_ReadProfileData(プロファイル読出) • MCv_WriteProfileData(プロファイル書込)	—	—

下記に実行指令(Execute)、および有効(Enable)によるモーション制御FBの基本動作を示します。ただし、モーション制御FBによっては仕様が異なる場合があります。

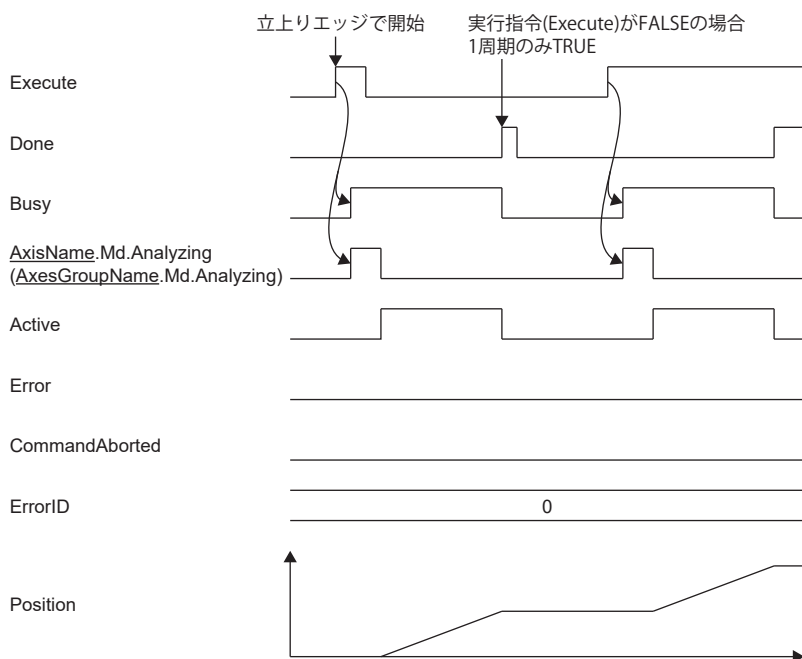
■実行指令(Execute)型によるモーション制御FBの基本動作

- 実行指令(Execute)型のモーション制御FBは、実行指令(Execute)の立上りエッジで入力パラメータを読み込み、動作を開始します。いったん動作を開始すると実行指令(Execute)をFALSEにしても完了するまで動作は継続します。
- 動作を開始すると、実行中(Busy)、実行完了(Done)、エラー (Error)、キャンセル受付(CommandAborted)の出力変数のうち1つのみTRUEになります。
- 実行完了(Done)、エラー (Error)、エラーコード(ErrorID)、キャンセル受付(CommandAborted)は、実行指令(Execute)の立下りエッジでリセットします。実行中(Busy)、制御中(Active)は、影響を受けません。
- 動作中に入力パラメータを変更した場合、実行指令(Execute)の再立上げ(再起動)、または連続更新(ContinuousUpdate)を使用した連続更新によって変更が反映します。
- 実行中(Busy)の立上りから解析中(AxisName.Md.Analyzing/AxesGroupName.Md.Analyzing)がTRUEになり、動作を開始すると解析中(AxisName.Md.Analyzing/AxesGroupName.Md.Analyzing)がFALSEになります。
- 実行指令(Execute)をパルスで使用する場合は、実行完了(Done)は1周期のみTRUEになります。

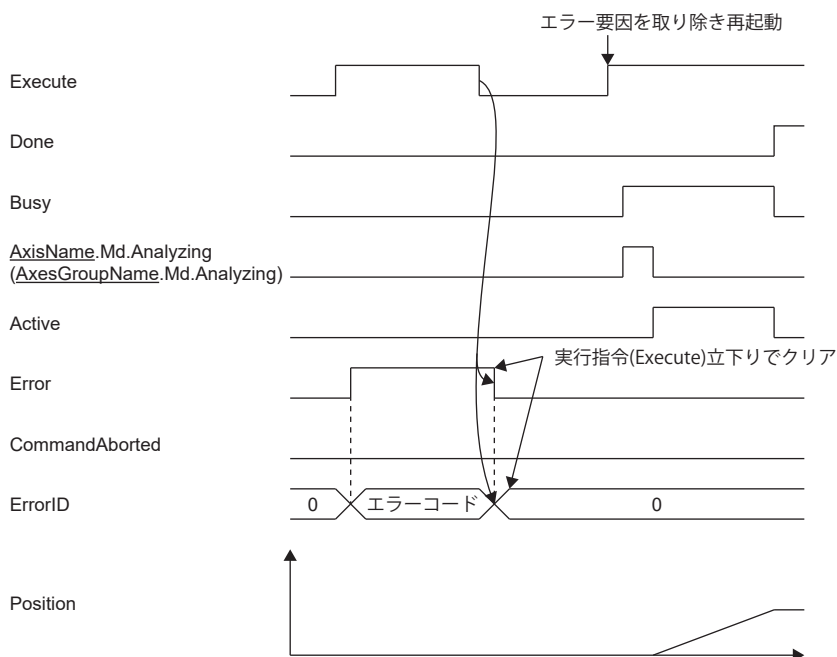
例

実行指令(Execute)型によるモーション制御FBのタイミングチャートを下記に示します。

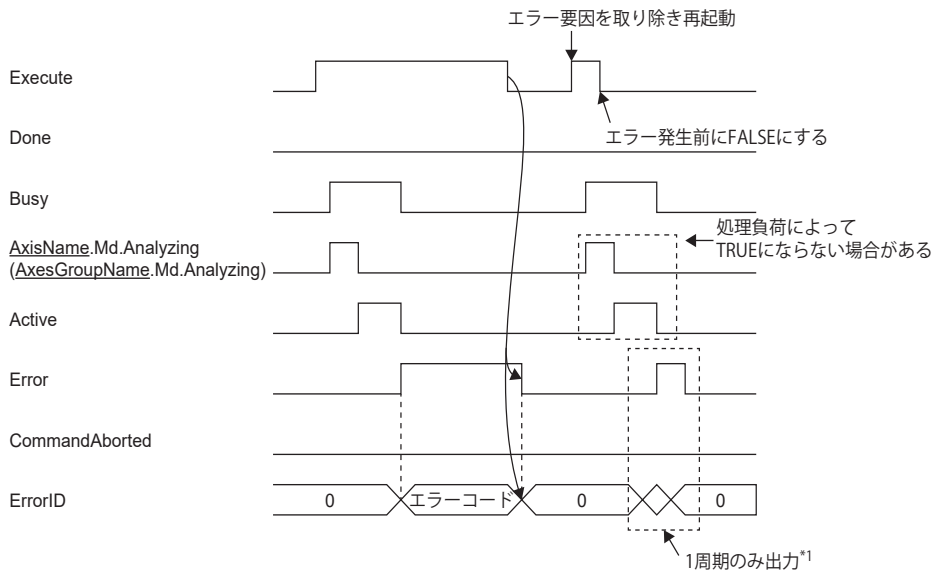
- 正常時



- 入出力変数異常時



・入力変数異常時



*1 モーション制御FBの終了条件(実行指令(Execute)がFALSE)が成立するため上記の動作となります。軸に関係のないモーション制御FB、または減速停止しないモーション制御FBの場合は、1周期のみエラー (Error)がTRUEになり、エラーコード(ErrorID)を出力します。減速停止を行う必要のあるモーション制御FBの場合は、軸が減速停止するまでエラー (Error)がTRUEになり、エラーコード(ErrorID)を維持し、軸の停止完了時にエラー (Error)がFALSEになり、エラーコード(ErrorID)をクリアします。

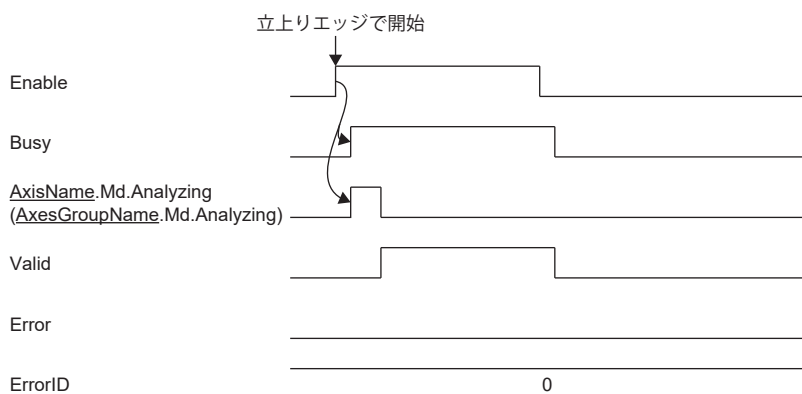
■有効(Enable)型によるモーション制御FBの基本動作

- 有効(Enable)型のモーション制御FBは、有効(Enable)がTRUEの間、繰り返し実行を続けます。
- 出力値有効(Valid)は、出力が有効な値であることを示します。出力値有効(Valid)がFALSEになった以降、各出力は変化しません。
- 出力値有効(Valid)/有効中(Enabled)/実行中(Busy)、エラー (Error)、実行中断(CommandAborted)の出力変数のうち1つのみTRUEになります。
- 実行中(Busy)の立上りから解析中(AxisName.Md.Analyzing/AxesGroupName.Md.Analyzing)がTRUEになり、動作を開始すると解析中(AxisName.Md.Analyzing/AxesGroupName.Md.Analyzing)がFALSEになります。
- 上記にない出力変数は、有効(Enable)の立下りでリセットします。(リセットのタイミングは出力値有効(Valid)/有効中(Enabled)/実行中(Busy)、エラー (Error)、実行中断(CommandAborted)と同じです。)

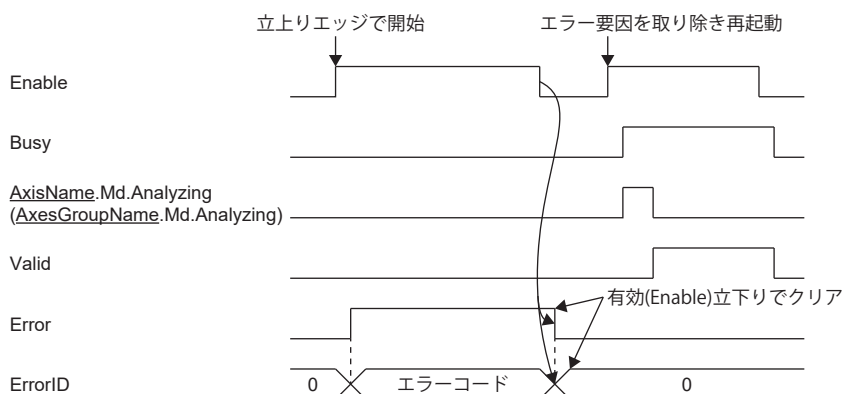
例

有効(Enable)型によるモーション制御FBのタイミングチャートを下記に示します。

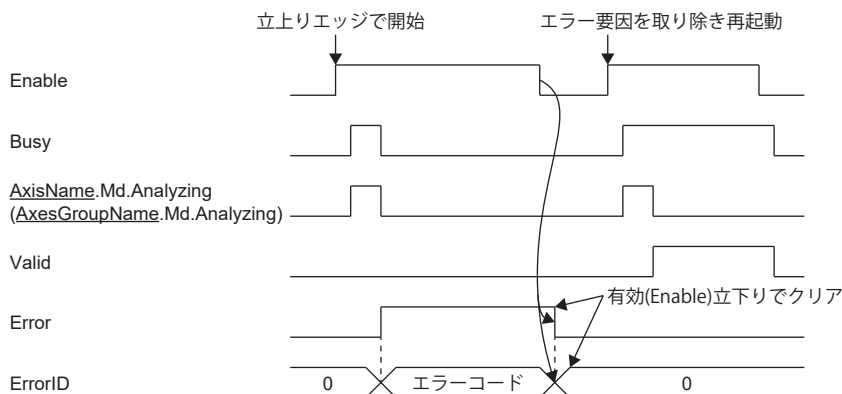
- 正常時



- 入出力変数異常時



- 入力変数異常時



モーション制御の種類

動作系モーション制御FBにより実行可能な軸、軸グループの制御には下記の種類があります。

大分類	中分類	小分類	内容
軸制御	単軸制御	位置決め制御	軸状態(AxisName.Md.AxisStatus)を「5: 位置決め運転中(DiscreteMotion)」にして、目標位置まで移動する制御です。
		連続制御	軸状態(AxisName.Md.AxisStatus)を「6: 連続動作運転中(ContinuousMotion)」にして、連続制御を行う制御です。
		同期制御	主軸(Master)と従軸(Slave)を入出力変数に持ち、従軸(Slave)の軸状態(AxisName.Md.AxisStatus)を「7: 同期運転中(SynchronizedMotion)」にして、同期制御を行う制御です。
		手動制御	有効(Enable)型で実行完了(Done)出力変数を持つ位置決め制御FB、連続制御FB、同期制御FBです。
		原点復帰制御	軸状態(AxisName.Md.AxisStatus)を「3: 原点復帰中(Homing)」にして、連続制御を行う制御です。
軸グループ制御	多軸制御	位置決め制御	軸グループ状態(AxesGroupName.Md.GroupStatus)を「5: 動作中(GroupMoving)」にして、目標位置まで移動する制御です。

エラー処理

モーション制御FBの実行時にエラーが発生するとエラー (Error)がTRUEになり、エラーコードをエラーコード(ErrorID)に出力します。このとき、軸の場合は、軸状態(AxisName.Md.AxisStatus)が「1: エラー停止中(ErrorStop)」, 軸グループの場合は、軸グループ状態(AxesGroupName.Md.GroupStatus)が「1: エラー停止中(GroupErrorStop)」に遷移します。動作可能な軸の状態が「1: エラー停止中(ErrorStop)」に遷移したとき、すべてのバッファリングFBは中断します。中断したモーション制御FBのエラー (Error)はTRUEになります。

その後、軸、または軸グループを始動するには、エラーリセットを実行する必要があります。

軸、または軸グループのエラーの確認方法、およびエラーのリセット方法については下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

Point

エラーコード(ErrorID)には、下記のいずれかと同じ値(制御により異なる)を出力します。ただし、イベントコードは出力しません。

- 軸エラーコード(AxisName.Md.ErrorID)
- 軸グループエラーコード(AxesGroupName.Md.ErrorID)
- モーション部最新システムエラーコード(MotionSystem.Md.ErrorID)

制御に使用する単位

モーションシステム内で取り扱う位置、速度、加減速度、ジャークの単位は、使用する軸の単位系に従います。

モーションシステム内で取り扱う位置や速度などの制御値には、下記の種類があります。

単位系、および位置、速度に関する制御値の詳細は、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

種類	内容
指定値	モーション制御FBへの入力に基づいた値(目標値)です。 (指定位置、指定速度など)
指令現在値	モーション演算により生成される現在の制御値です。 (指令現在位置、指令現在速度など)
フィードバック値	実軸におけるデバイス機器からのフィードバック値を軸の単位系に変換したものです。 (フィードバック位置、フィードバック速度など)

モーション制御FBの入出力変数

モーション制御FBの入出力変数について説明します。
モーション制御FBは、入出力変数、入力変数、出力変数を定義する必要があります。

入出力変数

軸情報(Axis)、軸グループ情報(AxesGroup)など、制御するドライバの軸、軸グループを設定します。
軸、軸グループは、エンジニアリングツールの軸設定、軸グループ設定にて軸を生成、初期化します。
設定した軸、軸グループは、グローバルラベルデータに軸変数、軸グループ変数として割り付けられます。
軸、軸グループの詳細については、下記を参照してください。
 使用するコントローラのユーザーズマニュアル

入力変数

目標位置や指令速度などの動作条件を設定します。

出力変数

モーション制御FBの状態、ドライバの状態、エラーの有無などを出力します。

クラス

入出力変数、入力変数、出力変数のクラスを下記に示します。

変数	クラス
入出力変数	VAR_IN_OUT
入力変数	VAR_INPUT
出力変数	VAR_OUTPUT

データ型

変数には、ビット長、処理方法、値の範囲などによって、変数の型を分類しています。
データ型については、下記を参照してください。
 28ページ データの指定方法

入力引数の省略

モーション制御FBの入力を省略した場合、モーション制御FBごとに定義した初期値を適用します。初期値の詳細は、使用するモーション制御FBを参照してください。
多重起動するモーション制御FBで速度などの入力を省略した場合、直前に制御していたモーション制御FBの入力値を引き継ぎます。

入出力のリフレッシュタイミング

モーション制御FBの引数は、モーション制御FBの呼出しタイミングでリフレッシュします。演算周期に同期してモーション制御FBの入出力を制御するには、演算周期と同じ周期の定周期プログラムからモーション制御FBを呼出してください。

モーション制御FBの注意事項

- モーション制御FBの入出力引数のリフレッシュは、モーション制御FBの呼出しタイミングで行いますが、実際に制御を実行する周期はモーション制御FBにより異なります。例えば動作系FBは、モーション制御FB呼出しタスク(スキャン/定周期/イベント)によらず、演算周期で制御を行います。詳細は、使用するモーション制御FBを参照してください。
- モーション制御FBの出力引数をリフレッシュするタイミングと、軸変数などの関連するラベルのリフレッシュタイミングは異なるため、各機能に記載されているタイミングチャートと実際の信号の変化は一致しない場合があります。各信号のリフレッシュ周期を考慮してプログラムを作成してください。
- モーション制御FBの動作中(実行中(Busy)がTRUE状態)は、モーション制御FBを呼出してください。プログラム内に呼び出さない場合、制御を継続しますが入力変数の値を変更しても制御に反映されなくなり、実行完了やエラー等の出力変数の更新ができなくなります。
- 初期実行タイププログラムでは、モーション制御FBの動作中にFB呼出しが停止する可能性があるため、モーション制御FBを使用しないでください。
- FOR~NEXT命令間や制御構文内(IF文、FOR文、CASE文など)と組合せて使用する場合、モーション制御FBの実行完了までモーション制御FBが呼び出されるように実行中(Busy)や実行完了(Done)などを確認するようにしてください。
- モーション制御FBは実行指令(Execute)、または有効(Enable)をTRUEにしたときに入力値を取り込みます。そのため、入力値は実行指令(Execute)、または有効(Enable)をTRUEにする前に設定してください。また、モーション制御FBの再起動や連続更新で複数の入力引数を変更する場合は、各引数の取込みタイミングを一致させるために、モーション制御FB呼出しタスクと同じタスクで変更を行ってください。
- モーション制御FBの実行指令(Execute)をTRUEにした後、実行指令(Execute)をFALSEにする場合は、実行中(Busy)がTRUE状態となってからFALSEにしてください。
- モーション制御では、軸の現在位置やモーション制御FBの目標位置などで、実数データ(浮動小数点データ)を使用しています。そのため、演算誤差を含む場合があります。例えば、MC_MoveRelative(相対値位置決め)のような相対位置指定の位置決めを繰り返し実行する場合、指令現在位置と設定した移動量との間に演算誤差が累積することがあります。演算誤差の累積が問題となる場合は、MC_MoveAbsolute(絶対値位置決め)のような絶対位置指定の位置決めを実行してください。また、モーション制御に関連する実数型のパラメータに小数点以下が含まれないように設定することで、機能内部で実数データ処理により発生する演算誤差を抑えることができます。モーション制御FBで使用する実数データ(浮動小数点データ)の詳細については、下記を参照してください。

☞ 38ページ 実数データ(浮動小数点データ)

- 1つのモーション制御FBインスタンスを複数プログラム(割込みプログラム含む並列に実行されるもの)から同時に呼出さないでください。同時に呼出した場合の動作は保証できません。

39.2 モーション制御FBの構成

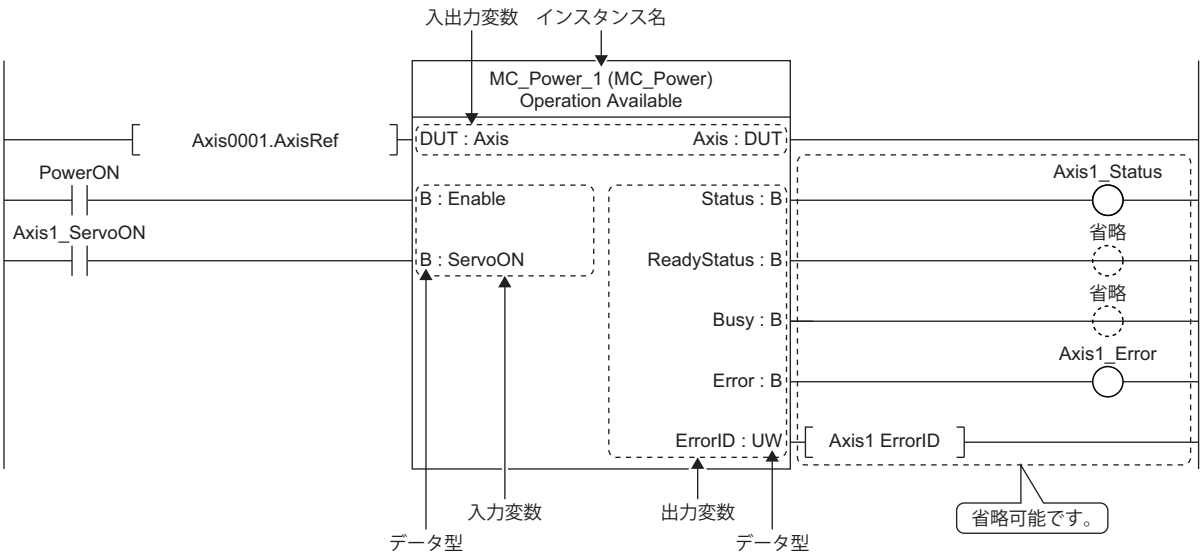
モーション制御FBのプログラム言語ごとの記述方法を説明します。

ラダー

ラダー言語では、下記のようにモーション制御FBを配置します。

例

MC_Power(運転可)の場合



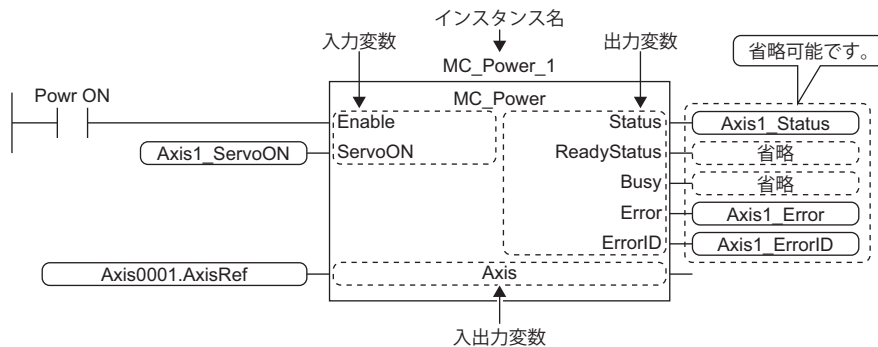
名称	内容
インスタンス名	FBごとに付けるインスタンス名です。 インスタンス名は変更できます。
入出力変数	軸情報(Axis), 軸グループ情報(AxesGroup)など, 制御するドライバなどの軸変数名を設定します。
入力変数	目標位置や指令速度などの動作条件を設定します。 設定を省略できます。省略した場合, 入力変数は初期値となります。
出力変数	FBの状態やドライバの状態を出力します。
データ型	入出力変数, 入力変数, 出力変数のデータ型を示します。 入出力変数, 入力変数, 出力変数のデータ型については, 下記を参照してください。 28ページ データの指定方法

FBD/LD(ファンクションブロックダイアグラム/ラダー)

FBD/LD言語では、下記のようにモーション制御FBを配置します。

例

MC_Power(運転可)の場合



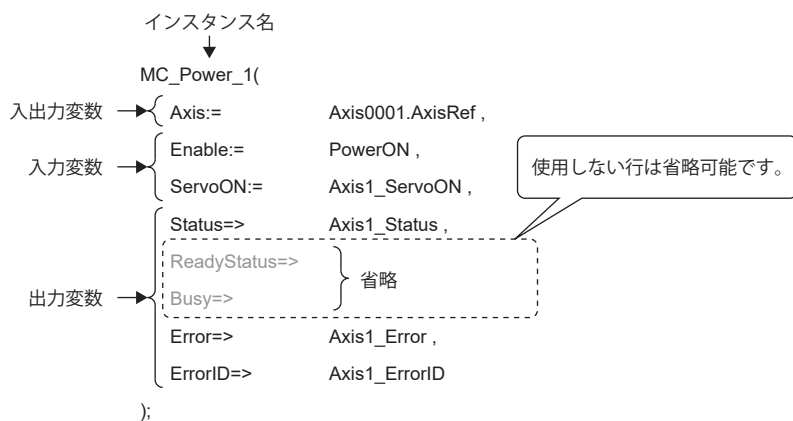
名称	内容
インスタンス名	FBごとに付けるインスタンス名です。 インスタンス名は変更できます。
入出力変数	軸情報(Axis), 軸グループ情報(AxesGroup)など, 制御するドライバなどの軸変数名を設定します。
入力変数	目標位置や指令速度などの動作条件を設定します。 設定を省略できます。省略した場合, 入力変数は初期値となります。
出力変数	FBの状態やドライバの状態を出力します。

ST(ストラクチャードテキスト)

ST言語では、下記のようにモーション制御FBを配置します。

例

MC_Power(運転可)の場合



名称	内容
インスタンス名	FBごとに付けるインスタンス名です。 インスタンス名は変更できます。
入出力変数	軸情報(Axis), 軸グループ情報(AxesGroup)など, 制御するドライバなどの軸変数名を設定します。
入力変数	目標位置や指令速度などの動作条件を設定します。 設定を省略できます。省略した場合, 入力変数は初期値となります。
出力変数	FBの状態やドライバの状態を出力します。

40 モーション制御ファンクションブロックの変数一覧

モーション制御FBに使用する変数の一覧について説明します。

40.1 変数一覧

モーション制御で扱う変数には下記の種類があります。

変数		内容	参照
軸変数		軸種別などのパラメータデータや、軸の現在位置などのモニタデータからなる軸に関する変数です。	☞ 1378ページ 軸変数
軸グループ変数		構成軸などのパラメータデータや、軸グループ状態などのモニタデータからなる軸グループに関する変数です。	☞ 1402ページ 軸グループ変数
システム変数		アドオンに関するパラメータデータや、演算周期などのモニタデータからなるシステムに関する変数です。	☞ 1408ページ システム変数
アドバンスト同期制御変数	入力軸変数	アドバンスト同期制御の入力軸のパラメータデータや、軸の現在位置などのモニタデータからなる入力軸に関する変数です。	☞ 1412ページ 入力軸変数
	出力軸変数	アドバンスト同期制御の出力軸のパラメータデータや、軸の現在位置などのモニタデータからなる出力軸に関する変数です。	☞ 1414ページ 出力軸変数
	システム変数	アドバンスト同期制御のアドオンに関するパラメータデータや、モニタデータからなるシステムに関する変数です。	☞ 1419ページ システム変数(アドバンスト同期制御)
	その他構造体の変数	アドバンスト同期制御で使用する上記以外に分類される構造体の変数です。	☞ 1437ページ その他構造体の変数(アドバンスト同期制御)
その他構造体の変数		上記以外に分類される構造体の変数です。	☞ 1419ページ その他構造体の変数

Point

本マニュアルに記載してある変数のアンダーライン付きの変数部分は、ユーザにて変数名を定義することを示しています。

AxisName.AxisRef.AxisNo

↓ユーザにて定義する変数

変数一覧表の見方

一覧表の各項目の見方を下記に示します。

■取込

項目	内容
演算周期	各演算周期を実行するタイミング
ノーマル	モーションシステムのノーマルタスクを実行するタイミング
サービス	モーションサービス処理制御を実行するタイミング
即時	イベントが発生したとき
始動時	FBが始動したとき
停止時	FBが停止したとき
システム起動時	モーションシステム起動時
機器接続時	デバイス機器接続時
レディ ON	"シーケンサレディ (MotionSystem.Cd.SequenceReady)"がONしたとき
システム基本周期	システム基本周期処理を実行するタイミング
演算周期(GroupEnable中)	各演算周期を実行するタイミング(GroupEnable中)
軸グループ有効時	軸グループが有効になるタイミング
演算周期(FB起動中)	演算周期処理を実行するタイミング(FB起動中)
カムデータ0点目通過時	1サイクル現在位置が現在実行中のカムデータの0点目を通過したとき<例> カム分解能512の場合の0点目の位置を通過したタイミング

■属性

表示	内容
LIST_WRITE_BACK	ラベルリスト(初期値書戻し可)
LIST_READ_ONLY	ラベルリスト(初期値書戻し不可)

軸変数

軸変数は、軸種別に応じて下記のデータ型となります。

軸種別	データ型
実ドライブ軸	AXIS_REAL
実エンコーダ軸	AXIS_ENCODER
仮想ドライブ軸	AXIS_VIRTUAL
仮想エンコーダ軸	AXIS_VIRTUAL_ENCODER
仮想連結軸	AXIS_VIRTUAL_LINK

Point

使用する軸種別により、設定可能な軸変数が異なります。(1396ページ 軸変数で各軸種別ごとに設定可能な変数)

各軸のデータ型は、下記のメンバを所有しています。

メンバ名	データ型*1*2	内容	参照
AxisRef	AXIS_REF	モーション制御用ファンクションブロックの入力/出力用のデータ構造です。 軸種別によらず固定の型となります。	1378ページ AxisName.AxisRef .(軸情報)
PrConst	AXIS_□_PRM_CONST	軸のパラメータデータ(定数)を格納します。 軸変数初期化時に設定値を展開します。 軸変数初期化後に制御への再取込みを実施しません。	1379ページ AxisName.PrConst .(軸パラメータ定数)
Pr	AXIS_□_PRM	軸のパラメータデータを格納します。 軸変数初期化時に初期値を展開します。 軸変数初期化後も制御への再取込みを実施します。 パラメータによって制御への取込みタイミングが異なります。	1383ページ AxisName.Pr .(軸パラメータ)
Md	AXIS_□_MONI	軸のモニタデータを格納します。 モニタデータごとに定められた周期でリフレッシュを実施します。	1387ページ AxisName.Md .(軸モニタデータ)
Cd	AXIS_□_CMD	軸制御用指令データを格納します。 制御演算周期ごとに最新の値を取得し制御に使用します。	1394ページ AxisName.Cd .(軸制御データ)

- *1 □: 各軸種別のデータ型
*2 軸種別によりデータ型のメンバが異なります。

AxisName.AxisRef.(軸情報)					
変数名	名称	取込	データ型	属性	内容
AxisNo	軸No.	—	WORD(UINT)	LIST_WRITE _BACK	軸No.を設定します。 • 0: 未設定 • 1~10000: 設定軸No.
StartIO	入出力No.	—	WORD(HEX)	LIST_WRITE _BACK	入出力No.を設定します。 デフォルトでは3E0Hが自動で入力されます。 • 000H~0FFH: 先頭入出力番号(16進数4桁で表したときの上3桁)

AxisName.PrConst.(軸パラメータ定数)

変数名	名称	取込	データ型	属性	内容
AddressOfStation	局アドレス設定	システム 起動時	WSTRING(63)	LIST_WRITE _BACK	デバイス局のネットワークアドレス(IPアドレス)を文字列で設定します。 多軸ドライブユニットのように、1つの局に複数の論理軸が含まれる場合、論理軸を識別するためのマルチドロップ番号を設定します。 ■単軸機器の場合 192.168.3.1 └── IPアドレス ■多軸機器の場合 <例>MR-J5W-GのC軸の場合 192.168.3.1#2 └── マルチドロップ番号 └── IPアドレス *: マルチドロップ番号は、「# + 番号(10進数)」で設定します。 • #0: A軸 • #1: B軸 • #2: C軸 *: マルチドロップ番号は省略できます。省略した場合、「#0」とみなします。
AxisType	軸種別設定	システム 起動時	INT (MC_AXIS_TYPE) 1451ページ MC_AXIS_TYPE	LIST_WRITE _BACK	軸種別を設定します。 • 0: 実ドライブ軸(DriveAxis) • 2: 実エンコーダ軸(EncoderAxis) • 3: 仮想ドライブ軸(VirtualDriveAxis) • 4: 仮想エンコーダ軸(VirtualEncoderAxis) • 5: 仮想連結軸(VirtualLinkAxis)
Encoder_AxisType	実エンコーダ軸タイプ設定	システム 起動時	INT (MC_ENCODER_AXIS_TYPE) 1451ページ MC_ENCODER_AXIS_TYPE	LIST_WRITE _BACK	実エンコーダ軸のタイプを設定します。 • 1: ドライブユニット経由(Drive)
Encoder_CounterDisableSignal	カウンタ無効化信号	システム 起動時	SIGNAL_SELECT 1424ページ SIGNAL_SELECT(信号選択)	LIST_WRITE _BACK	カウンタディセーブルに切換えるための信号を設定します。 カウンタ無効化信号(Encoder_CounterDisableSignal)固有の設定、および動作を下記に示します。 範囲外の値を設定した場合、パラメータ範囲外(軸)(エラーコード: 1A60H)となります。 ■対象(Target) • データ種別には、[VAR], [DEV], [CONST]のみ設定できます。 • [VAR], [DEV]指定時に参照しているデバイス局のネットワーク同期通信設定は、「同期する」のみ指定できます。 ■信号検出方法(Detection) 下記レベル検出のみ設定を許可します。 • 0: TRUE時検出(HighLevel) • 1: FALSE時検出(LowLevel) ■補正時間(CompensationTime) 「0.0[s]」のみ設定できます。 ■フィルタ時間(FilterTime) 設定範囲は「0.0~5.0[s]」です。
Encoder_RingCounterLowerValue	エンコーダリングカウンタ下限値	システム 起動時	DINT	LIST_WRITE _BACK	エンコーダリングカウンタの下限値を設定します。 ■PosActualValueが1ワード • -32768~32767 1ワードの範囲外の値を設定した場合、パラメータ範囲外(軸)(エラーコード: 1A60H)となります。 ■PosActualValueが2ワード、または省略 • -2147483648~2147483647

変数名	名称	取込	データ型	属性	内容
Encoder_RingCout_UpperValue	エンコーダリングカウンタ上限値	システム起動時	DINT	LIST_WRITE_BACK	エンコーダリングカウンタの上限値を設定します。 ■PosActualValueが1ワード ・-32768~32767 *:1ワードの範囲外の値を設定した場合、パラメータ範囲外(軸)(エラーコード: 1A60H)となります。 ■PosActualValueが2ワード、または省略 ・-2147483648~2147483647
HwStrokeLimit_FlsSignal	上限リミット信号	システム起動時	SIGNAL_SELECT  1424ページ SIGNAL_SELECT(信号選択)	LIST_WRITE_BACK	上限リミット信号(FLS)を使用する信号を設定します。外部信号として使用している、コントローラ管理のデバイス局が切断された場合、軸を停止します。上限リミット信号(HwStrokeLimit_FlsSignal)固有の設定、および動作を下記に示します。 ■入出力番号(StartIO) 入力値を無視します。 ■対象(Target) 設定がないと、信号無効と判断して常に信号未検出状態にします。 データ種別には、[VAR]、[DEV]、[CONST]のみ設定できません。 使用できない値が設定されている場合、パラメータ範囲外(軸)(エラーコード: 1A60H)となります。要因は下記の詳細コードで確認できます。 ・不正な文字列を設定した(詳細コード: 0001H) ・使用できないデータ種別を設定した(詳細コード: 0002H) ・使用できない型を指定した(詳細コード: 0003H) ・ラベル指定時に数値型以外の型を指定した(詳細コード: 0004H) ・存在しない信号を設定した(詳細コード: 0005H) ・外部機器からのアクセスが無効なグローバルラベルを設定した(詳細コード: 0005H) ・文字数の上限をオーバーした(詳細コード: 0007H) ・参照しているデバイス局のネットワーク同期通信設定に“同期しない”を設定した(詳細コード: 8002H) ・未対応のデバイスを設定した(詳細コード: 8002H) ・範囲外のデバイス番号を設定した(詳細コード: 8002H) ■信号検出方法(Detection) 下記レベル検出のみ設定を許可します。 ・0: TRUE時検出(HighLevel) ・1: FALSE時検出(LowLevel) 範囲外の値を設定した場合、パラメータ範囲外(軸)(エラーコード: 1A60H(詳細コード: 0067H))となります。 ■補正時間(CompensationTime) 入力値を無視します。 ■フィルタ時間(FilterTime) 設定範囲は「0.0~5.0[s]」です。 範囲外の値を設定した場合、各軸信号のフィルタ時間設定範囲外警告(イベントコード: 0D24H)となり、フィルタ時間は「0.0」で動作します。

変数名	名称	取込	データ型	属性	内容
HwStrokeLimit_RlsSignal	下限リミット信号	システム 起動時	SIGNAL_SELECT 1424ページ SIGNAL_SELECT(信号選択)	LIST_WRITE _BACK	下限リミット信号(RLS)を使用する信号を設定します。外部信号として使用している、コントローラ管理のデバイス局が切断された場合、軸を停止します。下限リミット信号(HwStrokeLimit_RlsSignal)固有の設定、および動作を下記に示します。 ■入出力番号(StartIO) 入力値を無視します。 ■対象(Target) 設定がないと、信号無効と判断して常に信号未検出状態にします。 データ種別には、[VAR], [DEV], [CONST]のみ設定できます。 使用できない値が設定されている場合、パラメータ範囲外(軸)(エラーコード: 1A60H)となります。要因は下記の詳細コードで確認できます。 - 不正な文字列を設定した(詳細コード: 0001H) - 使用できないデータ種別を設定した(詳細コード: 0002H) - 使用できない型を指定した(詳細コード: 0003H) - ラベル指定時に数値型以外の型を指定した(詳細コード: 0004H) - 存在しない信号を設定した(詳細コード: 0005H) - 外部機器からのアクセスが無効なグローバルラベルを設定した(詳細コード: 0005H) - 文字数の上限をオーバーした(詳細コード: 0007H) - 参照しているデバイス局のネットワーク同期通信設定に"同期しない"を設定した(詳細コード: 8002H) - 未対応のデバイスを設定した(詳細コード: 8002H) - 範囲外のデバイス番号を設定した(詳細コード: 8002H) ■信号検出方法(Detection) 下記レベル検出のみ設定を許可します。 0: TRUE時検出(HighLevel) 1: FALSE時検出(LowLevel) 範囲外の値を設定した場合、パラメータ範囲外(軸)(エラーコード: 1A60H(詳細コード: 0067H))となります。 ■補正時間(CompensationTime) 入力値を無視します。 ■フィルタ時間(FilterTime) 設定範囲は「0.0~5.0[s]」です。 範囲外の値を設定した場合、各軸信号のフィルタ時間設定範囲外警告(イベントコード: 0D24H)となり、フィルタ時間は「0.0」で動作します。
OperationCycle	制御周期設定	システム 起動時	INT	LIST_WRITE _BACK	■実軸 軸に割り当てたデバイス局の通信周期と同じ演算周期を自動で選択します。 該当する演算周期がない場合は"周期割り当て不正(軸)(エラーコード: 1B2BH)"となります。 0: 自動設定 ■仮想軸、または局アドレスが未設定の実軸 0: 第1演算周期で動作 1: 第1演算周期で動作 2: 第2演算周期で動作 3: 第3演算周期で動作
PosRestoration_AbsPosBase	絶対位置基準設定	システム 起動時	INT (MC_POS_SOURCE) 1452ページ MC_POS_SOURCE	LIST_WRITE _BACK	絶対位置システムを使用する場合、現在位置復元時に基準となる現在位置を設定します。 3: 送り機械位置(FeedMachinePosition)
PosRestoration_AbsPosEnable	絶対位置管理設定	システム 起動時	INT (MC_ABS_SYSTEM) 1451ページ MC_ABS_SYSTEM	LIST_WRITE _BACK	絶対位置管理を設定します。 0: 絶対位置システムを使用しない(ABSDisabled) 1: 絶対位置システムを使用する(Enabled) -1: 自動設定(接続機器から取得)(Auto)
PosRestoration_RestorationBase	現在位置復元基準位置設定	システム 起動時	INT (MC_POS_RESTORATION_BASE) 1453ページ MC_POS_RESTORATION_BASE	LIST_WRITE _BACK	絶対位置システムを使用する場合、現在位置復元の基準位置を設定します。 0: バックアップ位置 1: 原点 -1: 自動判定

変数名	名称	取込	データ型	属性	内容
RingCount_Enable	リングカウンタ有効選択	システム起動時	BOOL	LIST_WRITE_BACK	リングカウンタ有効/無効を設定します。 • FALSE: 無効 • TRUE: 有効
RingCount_LowerValue	リングカウンタ下限値	システム起動時	LREAL	LIST_WRITE_BACK	リングカウンタ下限値を設定します。 • -10000000000.0~10000000000.0 リングカウンタ無効時は取込みません
RingCount_UpperValue	リングカウンタ上限値	システム起動時	LREAL	LIST_WRITE_BACK	リングカウンタ上限値を設定します。 • -10000000000.0~10000000000.0 リングカウンタ無効時は取込みません
SlaveEmulate_Enable	軸エミュレート有効	システム起動時	BOOL	LIST_WRITE_BACK	エミュレート軸として使用するかどうかを設定します。 • FALSE: 無効 • TRUE: 有効
SlaveObject	オブジェクトデータ	システム起動時	SLAVE_OBJECT_REAL 1434ページ SLAVE_OBJECT_REAL	LIST_WRITE_BACK	実ドライブ軸にオブジェクトデータを設定します。
			SLAVE_OBJECT_VIRTUAL_ENCODER 1436ページ SLAVE_OBJECT_VIRTUAL_ENCODER		仮想エンコーダ軸にオブジェクトデータを設定します。
TorqueLimit_Max	トルク制限最大値	システム起動時	LREAL	LIST_WRITE_BACK	トルク制限値として指定可能な最大値を設定します。(正方向/負方向共通です。) • 0.0~1000.0[%]
TorqueLimit_NegativeInitial	負方向トルク制限初期値	システム起動時	LREAL	LIST_WRITE_BACK	負方向トルク制限値(AxisName.Cd.TorqueLimit_Negative)の初期値を設定します。 • 0.0~1000.0[%]
TorqueLimit_PositiveInitial	正方向トルク制限初期値	システム起動時	LREAL	LIST_WRITE_BACK	正方向トルク制限値(AxisName.Cd.TorqueLimit_Positive)の初期値を設定します。 • 0.0~1000.0[%]

AxisName.Pr.(軸パラメータ)

変数名	名称	取込	データ型	属性	内容
AccelerationLimit	加速度制限値	レディ ON	LREAL	LIST_WRITE _BACK	加速度制限値を設定します。 0.0000, 0.0001~2147483647.0[U/s²] 設定によって下記ようになります。 - 加減速方式が「1: 加減速時間一定方式(mcFixedTime)」の場合は、加速度の制限を行いません。 「0.0000」の場合は、加速度の制限を行いません。 0.0001未満の正数を設定した場合、「0.0000」として取込みます。
AccelerationZeroBehavior	始動時加減速度0指定時動作選択	レディ ON	INT (MC_ACC_ZERO_MODE) 1452ページ MC_ACC_ZERO_MODE	LIST_WRITE _BACK	始動時に、加速度、減速度、または加減速時間に「0.0」を設定した場合の動作を設定します。 -1: エラー (始動しない)(ACCErrror) 1: 最大加減速(MaximumAcceleration)
CmdInPos_Width	指令インポジション幅	レディ ON	LREAL	LIST_WRITE _BACK	指令インポジション幅を設定します。 0.0: 機能無効 0.0000000001~10000000000.0: 機能有効 0.0000000001未満の正数を設定した場合、「0.0」として取込みます。
DecelerationLimit	減速度制限値	レディ ON	LREAL	LIST_WRITE _BACK	減速度制限値を設定します。 0.0000, 0.0001~2147483647.0[U/s²] 設定によって下記ようになります。 - 加減速方式が「1: 加減速時間一定方式(mcFixedTime)」の場合は、加速度の制限を行いません。 「0.0000」の場合は、減速度の制限を行いません。 0.0001未満の正数を設定した場合、「0.0000」として取込みます。
Drive_UnitConvRatioNum	ドライバ単位変換分子	レディ ON	DWORD(UDINT)	LIST_WRITE _BACK	モーションシステムの指令単位をドライバの指令単位に変換するための分子を設定します。 1~2147483647
Drive_UnitConvRatioDen	ドライバ単位変換分母	レディ ON	DWORD(UDINT)	LIST_WRITE _BACK	モーションシステムの指令単位をドライバの指令単位に変換するための分母を設定します。 1~2147483647

変数名	名称	取込	データ型	属性	内容
ForcedStop_Signal	緊急停止信号	レディ ON	SIGNAL_SELECT 1424ページ SIGNAL_SELECT(信号選択)	LIST_WRITE _BACK	緊急停止を使用する信号を設定します。 外部信号として使用している、コントローラ管理のデバイス局が切断された場合、軸を停止します。 緊急停止信号(ForcedStop_Signal)固有の設定、および動作を下記に示します。 ■入出力番号(StartIO) 入力値を無視します。 ■対象(Target) 設定がないと、信号無効と判断して常に信号未検出状態にします。 データ種別は、[VAR]、[DEV]、[CONST]のみ指定できます。 使用できない値が設定されている場合、パラメータ範囲外(軸)(エラーコード: 1A60H)となります。要因は下記の詳細コードで確認できます。 - 不正な文字列を設定した(詳細コード: 0001H) - 使用できないデータ種別を設定した(詳細コード: 0002H) - 使用できない型を指定した(詳細コード: 0003H) - ラベル指定時に数値型以外の型を指定した(詳細コード: 0004H) - 存在しない信号を設定した(詳細コード: 0005H) - 外部機器からのアクセスが無効なグローバルラベルを設定した(詳細コード: 0005H) - 文字数の上限をオーバーした(詳細コード: 0007H) - 参照しているデバイス局のネットワーク同期通信設定に"同期しない"を設定した(詳細コード: 8002H) - 未対応のデバイスを設定した(詳細コード: 8002H) - 範囲外のデバイス番号を設定した(詳細コード: 8002H) ■信号検出方法(Detection) 下記レベル検出のみ設定を許可します。 0: TRUE時検出(HighLevel) 1: FALSE時検出(LowLevel) エッジ検出を設定した場合、パラメータ範囲外(軸)(エラーコード: 1A60H)となります。 ■補正時間(CompensationTime) 入力値を無視します。 ■フィルタ時間(FilterTime) 設定範囲は「0.0~5.0[s]」です。 範囲外の値を設定した場合、各軸信号のフィルタ時間設定範囲外警告(イベントコード: 0D24H)となり、フィルタ時間は「0.0」で動作します。
Homing_Required	原点復帰要否設定	レディ ON	BOOL	LIST_WRITE _BACK	原点復帰の要否を設定します。 原点復帰が不要な場合は、FALSEに設定します。 FALSEに設定すると、現在位置復元時に原点復帰要求がTRUEとなりません。 - FALSE: 原点復帰不要 - TRUE: 原点復帰必要
JerkLimit	ジャーク制限値	レディ ON	LREAL	LIST_WRITE _BACK	ジャーク制限値を設定します。 0.0000, 0.0001~2147483647.0[U/s³] 設定によって下記ようになります。 「0.0000」の場合は、ジャーク制限を行いません。 0.0001未満の正数を設定した場合、「0.0000」として取込みます。
OverrunOperation	オーバラン時動作設定	レディ ON	INT (MC_DECEL_STOP_MODE) 1451ページ MC_DECEL_STOP_MODE	LIST_WRITE _BACK	位置決め動作中に目標位置をオーバーした場合の動作を設定します。 1: 即停止(ImmediateStop) 2: 現在の加減速度を続行(KeepCurrentAcc)
StartableAtUnhomed	原点復帰未完時始動許可	レディ ON	BOOL	LIST_WRITE _BACK	原点復帰未完時に軸の始動を許可するか、許可しないかを設定します。 - FALSE: 許可しない - TRUE: 許可する

変数名	名称	取込	データ型	属性	内容
StopMode_Deceleration	停止時減速度	レディ ON	LREAL	LIST_WRITE _BACK	停止要因発生による減速停止時の減速度/減速時間を設定します。 ■加速度/減速度を指定する加減速方式(加減速度指定方式)の場合 ・0.0000, 0.0001~2147483647.0[U/s ²] ■加減速時間を指定する加減速方式(加減速時間一定方式)の場合 ・0.000000, 0.000001~8400.0[s] 設定によって下記ようになります。 ・始動時加減速度0指定時動作選択 (AccelerationZeroBehavior)に関係なく、「0.0」の場合は即停止します。 ・停止要因発生時に上限値で範囲外となった場合は上限値でクランプし、下限値で範囲外となった場合は「0(即停止)」として扱います。
StopMode_DecelerationCurve	減速停止時停止処理選択	レディ ON	INT (MC_STOP_CURVE_MODE) 📖 1451ページ MC_STOP_CURVE_MODE	LIST_WRITE _BACK	減速中(停止要因、自動減速含む)に、停止要因が発生した場合の動作を設定します。 ・1: 減速カーブ再作成(OverrideCurve)
StopMode_General	停止要因発生時停止選択	レディ ON	INT (MC_DECEL_STOP_MODE) 📖 1451ページ MC_DECEL_STOP_MODE	LIST_WRITE _BACK	停止要因が発生した場合の動作を設定します。 ・1: 即停止(ImmediateStop) ・2: 現在の加減速度を続行(KeepCurrentAcc) ・3: 加減速度を代替(AlternativeAcc) 減速度指定のないFB実行時は、即停止します。
StopMode_HwStrokeLimit	ハードウェアストロークリミットエラー発生時停止選択	レディ ON	INT (MC_DECEL_STOP_MODE) 📖 1451ページ MC_DECEL_STOP_MODE	LIST_WRITE _BACK	ハードウェアストロークリミットのエラーが発生した場合の動作を設定します。 ・1: 即停止(ImmediateStop) ・2: 現在の加減速度を続行(KeepCurrentAcc) ・3: 加減速度を代替(AlternativeAcc) 減速度指定のないFB実行時は、即停止します。
StopMode_ServoOff	運転中サーボOFF指令時処理選択	レディ ON	INT (MC_DECEL_STOP_MODE) 📖 1451ページ MC_DECEL_STOP_MODE	LIST_WRITE _BACK	運転中にサーボOFFとなった場合の動作を設定します。 ・0: 無視(Ignore) ・4: 即停止後サーボOFF(ServoOffAfterImmediateStop) ・5: 減速停止後サーボOFF(ServoOffAfterDecelStop)
StopMode_SwStrokeLimit	ソフトウェアストロークリミットエラー発生時停止選択	レディ ON	INT (MC_DECEL_STOP_MODE) 📖 1451ページ MC_DECEL_STOP_MODE	LIST_WRITE _BACK	ソフトウェアストロークリミットのエラーが発生した場合の動作を設定します。 ・1: 即停止(ImmediateStop) ・2: 現在の加減速度を続行(KeepCurrentAcc) ・3: 加減速度を代替(AlternativeAcc) 減速度指定のないFB実行時は、即停止します。
StopOption_DriverTargetIgnored	ドライバ指令破棄検出設定	レディ ON	BOOL	LIST_WRITE _BACK	軸動作中にドライバユニットの指令破棄状態を検出してエラーで停止するかを設定します。 ・FALSE: 検出無効 ・TRUE: 検出有効

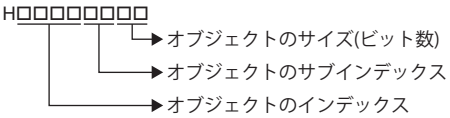
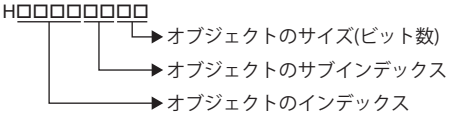
変数名	名称	取込	データ型	属性	内容
StopSignal	停止信号	レディ ON	SIGNAL_SELECT 1424ページ SIGNAL_SELECT(信号選択)	LIST_WRITE _BACK	外部入力信号の停止信号(STOP)を使用する信号を設定します。 外部信号として使用している、コントローラ管理のデバイス局が切断された場合、軸を停止します。 停止信号(StopSignal)固有の設定、および動作を下記に示します。 ■入出力番号(StartIO) 入力値を無視します。 ■対象(Target) 設定がないと、信号無効と判断して常に信号非検出状態にします。 データ種別には、[VAR], [DEV], [CONST]のみ設定できます。 使用できない値が設定されている場合、パラメータ範囲外(軸)(エラーコード: 1A60H)となります。要因は下記の詳細コードで確認できます。 - 不正な文字列を設定した(詳細コード: 0001H) - 使用できないデータ種別を設定した(詳細コード: 0002H) - 使用できない型を指定した(詳細コード: 0003H) - ラベル指定時に数値型以外の型を指定した(詳細コード: 0004H) - 存在しない信号を設定した(詳細コード: 0005H) - 外部機器からのアクセスが無効なグローバルラベルを設定した(詳細コード: 0005H) - 文字数の上限をオーバーした(詳細コード: 0007H) - 参照しているデバイス局のネットワーク同期通信設定に"同期しない"を設定した(詳細コード: 8002H) - 未対応のデバイスを設定した(詳細コード: 8002H) - 範囲外のデバイス番号を設定した(詳細コード: 8002H) ■信号検出方法(Detection) 下記のレベル検出のみ設定を許可します。 0: TRUE時検出(HighLevel) 1: FALSE時検出(LowLevel) エッジ検出を設定した場合、パラメータ範囲外(軸)(エラーコード: 1A60H)(詳細コード: 8003H)となります。 ■補正時間(CompensationTime) 入力値を無視します。 ■フィルタ時間(FilterTime) 設定範囲は「0.0~5.0[s]」です。 範囲外の値を設定した場合、各軸信号のフィルタ時間設定範囲外警告(イベントコード: 0D24H)となり、ファイル時間は「0.0」で動作します。
SwStrokeLimit_Lower	ソフトウェアストロークリミット下限値	レディ ON	LREAL	LIST_WRITE _BACK	ソフトウェアストロークリミット下限値を設定します。 -10000000000.0~10000000000.0
SwStrokeLimit_Target	ソフトウェアストロークリミット対象	レディ ON	INT (MC_POS_SOURCE) 1452ページ MC_POS_SOURCE	LIST_WRITE _BACK	ソフトウェアストロークリミットの対象を設定します。 -1: 無効(Invalid) 1: 指令現在位置(SetPosition) 3: 送り機械位置(FeedMachinePosition)
SwStrokeLimit_Upper	ソフトウェアストロークリミット上限値	レディ ON	LREAL	LIST_WRITE _BACK	ソフトウェアストロークリミット上限値を設定します。 -10000000000.0~10000000000.0
Unit_Position	位置指令単位	レディ ON	DWORD(HEX)	LIST_WRITE _BACK	モーション制御で使用する位置指令単位を設定します。 00000000H: pulse 00010000H: m - FD010000H: mm - FA010000H: μm - F7010000H: nm 00410000H: degree - FD410000H: $\times 10^{-3}$ degree - FA410000H: $\times 10^{-6}$ degree 00B40000H: Revolution 00C00000H: inch 00FF0000H: 任意単位
Unit_PositionString	位置指令単位文字列	レディ ON	WSTRING(31)	LIST_WRITE _BACK	モーション制御で使用する指令単位を文字列で設定します。 位置指令単位(Unit_Position)が「00FF0000H: 任意単位」の場合に設定します。

変数名	名称	取込	データ型	属性	内容
Unit_Velocity	速度指令単位	レディ ON	DWORD(HEX)	LIST_WRITE _BACK	モーション制御で使用する速度指令単位を設定します。 • 00000300H: U/s • 00004700H: U/min • FD000300H: U/ms • FA000300H: U/μs • F7000300H: U/ns
VelocityBias	速度バイアス値	レディ ON	LREAL	LIST_WRITE _BACK	軸ごとの速度バイアス値を設定します。 • 0.0000, 0.0001~2500000000.0 「0.0000」の場合は、速度バイアスは無効となります。
VelocityLimit_Negative	負方向速度制限値	レディ ON	LREAL	LIST_WRITE _BACK	アドレス減少方向速度制限値を設定します。 • 0.0001~2500000000.0
VelocityLimit_OverOperation	速度制限値オーバー時動作設定	レディ ON	INT (MC_VELOCITY_LIMIT_MODE) 1450ページ MC_VELOCITY_LIMIT_MODE	LIST_WRITE _BACK	制御中の速度制限値オーバー時の動作を設定します。 • 0: 無視(Ignore) • 3: 即停止(ImmediateStop) 設定によって下記ようになります。 • 「0: 無視(Ignore)」の場合、速度制限値を超える指令速度を検出すると、速度制限値オーバー警告(イベントコード: 0D03H)となり、速度制限値を超える指令をそのまま出力します。 • 「3: 即停止(ImmediateStop)」の場合、速度制限値を超える指令速度を検出すると、制御中速度制限値オーバー(エラーコード: 1AA4H)となり、実行中の位置決め制御が停止します。
VelocityLimit_Positive	正方向速度制限値	レディ ON	LREAL	LIST_WRITE _BACK	アドレス増加方向速度制限値を設定します。 • 0.0001~2500000000.0

AxisName.Md.(軸モニタデータ)

変数名	名称	取込	データ型	属性	内容
AccelerationLimit	加速度制限値	レディ ON	LREAL	LIST_READ_ ONLY	加速度制限値を格納します。
AccelerationOverride	加速度オーバライド係数	演算周期	LREAL	LIST_READ_ ONLY	加速度オーバライド係数を格納します。
AccelerationZeroBehavior	始動時加減速度0指定時動作選択	レディ ON	INT (MC_ACC_ZERO_MODE) 1452ページ MC_ACC_ZERO_MODE	LIST_READ_ ONLY	加速度、減速度、または加減速時間に「0.0」を設定した場合の動作を表示します。 • -1: エラー (始動しない)(ACCError) • 1: 最大加減速(MaximumAcceleration)
ActualPosition	フィードバック位置	演算周期	LREAL	LIST_READ_ ONLY	実際の現在位置を格納します。 「フィードバック位置 = 現在位置 × ドライバ単位変換分母 / ドライバ単位変換分子」の値となります。 リングカウンタ範囲で丸めた値を格納します。
ActualVelocity	フィードバック速度	演算周期	LREAL	LIST_READ_ ONLY	フィードバック速度を格納します。 フィードバック位置の差分から算出した速度です。 正方向(アドレス増加方向)へ移動する場合は正の値、負方向(アドレス減少方向)へ移動する場合は負の値となります。
Analyzing	解析中	演算周期	BOOL	LIST_READ_ ONLY	位置決め解析中の状態を表示します。 • FALSE: 解析中でない • TRUE: 解析中
AutoDeceleration	自動減速中	演算周期	BOOL	LIST_READ_ ONLY	自動減速処理の状態を表示します。 自動減速処理を行っている間、TRUEを格納します。 多重起動を行っている場合は、最終位置決めポイント実行中に自動減速処理を行っている間、TRUEになります。 制御変更を行った場合は、FALSEになります。 軸グループ動作中、構成軸の自動減速中 (AxisName.Md.AutoDeceleration)には、FALSEを格納します。 • FALSE: 自動減速中でない • TRUE: 自動減速中
AxisName	軸名称	システム 起動時	WSTRING(127)	LIST_READ_ ONLY	軸名称を表示します。

変数名	名称	取込	データ型	属性	内容
AxisStatus	軸状態	演算周期	INT (MC_AXIS_STATUS) 1449ページ MC_AXIS_STATUS	LIST_READ_ONLY	軸の状態を表示します。 • -1: 軸変数未初期化/軸/パラメータ異常(Invalid) • 0: 軸無効(Disabled) • 1: エラー停止中(ErrorStop) • 2: 減速停止中(Stopping) • 3: 原点復帰中(Homing) • 4: 待機中(Standstill) • 5: 位置決め運転中(DiscreteMotion) • 6: 連続動作運転中(ContinuousMotion) • 7: 同期運転中(SynchronizedMotion)
BufferingFBs	バッファリングFB数	始動時/ 停止時	INT	LIST_READ_ONLY	バッファリングFBの数(0~2)を表示します。
CmdInPos	指令インポジション	演算周期	BOOL	LIST_READ_ONLY	目標位置までの残距離が指令インポジション幅 (<u>AxisName</u> .Md.CmdInPos_Width)以下になったときにON になる信号です。 • TRUE: 指令インポジション幅以下 • FALSE: 指令インポジション幅より大
CmdInPos_Width	指令インポジション 幅	レディ ON	LREAL	LIST_READ_ONLY	指令インポジション幅を表示します。 指令インポジション幅(<u>AxisName</u> .Pr.CmdInPos_Width)の 設定値を取り込んだ値となります。 指令インポジション幅(<u>AxisName</u> .Md.CmdInPos_Width)が 「0.0」の場合、該当軸の指令インポジション機能は無効と なります。
CommandedAcceleration	指定加速度	演算周期	LREAL	LIST_READ_ONLY	ユーザが設定した加速度を格納します。(単位: U/s ²) 加減速方式が「1: 加減速時間一定方式(mcFixedTime)」の 場合は、指定した加減速時間を格納します。(単位: s) 軸グループ動作中、構成軸の指定加速度 (<u>AxisName</u> .Md.CommandedAcceleration)には「0.0」を格 納します。
CommandedDeceleration	指定減速度	演算周期	LREAL	LIST_READ_ONLY	ユーザが設定した減速度を格納します。(単位: U/s ²) 加減速方式が「1: 加減速時間一定方式(mcFixedTime)」の 場合は、不正な値です。 軸グループ動作中、構成軸の指定減速度 (<u>AxisName</u> .Md.CommandedDeceleration)には「0.0」を格 納します。
CommandedJerk	指定ジャーク	演算周期	LREAL	LIST_READ_ONLY	ユーザが設定したジャークを格納します。(単位: U/s ³) 軸グループ動作中、構成軸の指定ジャーク (<u>AxisName</u> .Md.CommandedJerk)には「0.0」を格納しま す。
CommandedPosition	指定位置	演算周期	LREAL	LIST_READ_ONLY	制御中の位置決め系のモーション制御FBで設定した目標 位置(Position), 移動量(Distance)を格納します。 位置決めが完了すると「0.0」を格納します。
CommandedVelocity	指定速度	演算周期	LREAL	LIST_READ_ONLY	制御中の動作系のモーション制御FBで取り込んだ指定速 度を格納します。 制御が完了すると「0.0」を格納します。 軸グループ動作中、構成軸の指定速度 (<u>AxisName</u> .Md.CommandedVelocity)には「0.0」を格納し ます。
Cst_SetTorque	cst/ct時指令現在トル ク	演算周期	LREAL	LIST_READ_ONLY	サイクリックトルクモード時および押当て制御モード時 の指令トルクを格納します。
Cst_TargetTorque	cst/ct時目標トルク	演算周期	LREAL	LIST_READ_ONLY	サイクリックトルクモード時および押当て制御モード時 の目標トルクを格納します。
CumulativePosition	累積現在位置	演算周期	LREAL	LIST_READ_ONLY	現在位置のアドレスを格納します。 現在位置変更すると、アドレスが現在位置変更値に変わ ります。 「-10000000000.0 ≤ 累積現在位置 < 10000000000.0」のリン グアドレスとなります。
DecelerationLimit	減速度制限値	レディ ON	LREAL	LIST_READ_ONLY	減速度制限値を格納します。

変数名	名称	取込	データ型	属性	内容
Drive_RPDO[1..64]	RPDO(モーションシステム→デバイス機器)マッピング	機器接続時	DWORD(HEX)	LIST_READ_ONLY	ネットワークパラメータのRPDOマッピング設定のうち、第n番目に指定したオブジェクトインデックスを示します。  - オブジェクトのサイズは、1バイト、2バイト、4バイト、8バイトをビット数で表示します。 - 第n番目のオブジェクトサイズが1バイトの場合、第n+1番目にはパディングオブジェクトが入ります。
Drive_TPDO[1..64]	TPDO(デバイス機器→モーションシステム)マッピング	機器接続時	DWORD(HEX)	LIST_READ_ONLY	ネットワークパラメータのTPDOマッピング設定のうち、第n番目に指定したオブジェクトインデックスを示します。  - オブジェクトのサイズは、1バイト、2バイト、4バイト、8バイトをビット数で表示します。 - 第n番目のオブジェクトサイズが1バイトの場合、第n+1番目にはパディングオブジェクトが入ります。
Drive_UnitConvRatioNum	ドライバ単位変換分子	レディ ON	DWORD(UDINT)	LIST_READ_ONLY	ドライバの指令単位に変換するための分子を格納します。
Drive_UnitConvRatioDen	ドライバ単位変換分母	レディ ON	DWORD(UDINT)	LIST_READ_ONLY	ドライバの指令単位に変換するための分母を格納します。
Driver_Mode	ドライバ制御モード	演算周期	INT (MC_DRIVE_MODE) 1450ページ MC_DRIVE_MODE	LIST_READ_ONLY	ドライバ制御モードの状態を表示します。 6: 原点復帰(hm) 8: 位置制御(csp) 9: 速度制御(csv) 10: トルク制御(cst) -104: 押当て制御(ct)
Driver_ReadyOn	ドライバレディ ON状態	演算周期	BOOL	LIST_READ_ONLY	接続ドライバの状態を表示します。 ドライバ状態(AxisName.Md.Driver_State)が下記の場合、TRUEになります。 2: Fault Reaction Active 5: Switched On 6: Operation Enable 7: Quick Stop Active
Driver_ServoOn	ドライバサーボON状態	演算周期	BOOL	LIST_READ_ONLY	接続ドライバの状態を表示します。 ドライバ状態(AxisName.Md.Driver_State)が下記の場合、TRUEになります。 6: Operation Enable 7: Quick Stop Active
Driver_State	ドライバ状態	演算周期	INT (MC_DRIVE_STATE) 1451ページ MC_DRIVE_STATE	LIST_READ_ONLY	ドライバ状態を表示します。 0: Not Ready To Switch On 1: Fault 2: Fault Reaction Active 3: Switch On Disabled 4: Ready To Switch On 5: Switched On 6: Operation Enable 7: Quick Stop Active -1: Invalid
DriverError	ドライブユニットエラー検出	即時	BOOL	LIST_READ_ONLY	ドライブユニットエラーの状態を表示します。 - FALSE: ドライブユニットエラーなし - TRUE: ドライブユニットエラーあり
DriverErrorID	ドライブユニットエラーコード	即時	WORD(HEX)	LIST_READ_ONLY	ドライブユニットエラー発生時、オブジェクトデータの「Current alarm(2A41H)」の上位16ビットを表示します。
DriverErrorDetailID	ドライブユニットエラー詳細コード	即時	WORD(HEX)	LIST_READ_ONLY	ドライブユニットエラー発生時、オブジェクトデータの「Current alarm(2A41H)」の下位16ビットを格納します。
Encoder_Connected	接続状態	演算周期	BOOL	LIST_READ_ONLY	仮想エンコーダ軸の接続状態を表示します。 - FALSE: 未接続 - TRUE: 接続中

変数名	名称	取込	データ型	属性	内容
Encoder_CounterDisable	カウンタ無効中	演算周期	BOOL	LIST_READ_ONLY	エンコーダからの入力状態を表示します。 ・FALSE: カウンタディセーブル ・TRUE: カウンタディセーブル
Error	軸エラー検出	即時	BOOL	LIST_READ_ONLY	軸エラーの状態を表示します。 ・FALSE: 軸エラーなし ・TRUE: 軸エラーあり
ErrorID	軸エラーコード	即時	WORD(HEX)	LIST_READ_ONLY	発生した軸エラーのエラーコードを表示します。
FeedMachinePosition	送り機械位置	演算周期	LREAL	LIST_READ_ONLY	機械座標による現在位置のアドレスを格納します。 原点復帰完了位置からの累積値となります。 現在位置変更しても、アドレスが変わりません。 「-10000000000.0 ≤ 送り機械位置 < 10000000000.0」のリングアドレスとなります。
FollowupDisable	フォローアップ無効中	演算周期	BOOL	LIST_READ_ONLY	フォローアップの無効化状態を表示します。 無効化されている場合、フォローアップを行いません。 ・FALSE: フォローアップ有効 ・TRUE: フォローアップ無効
ForcedStop_Released	緊急停止解除中	演算周期	BOOL	LIST_READ_ONLY	緊急停止の解除状態を表示します。 ・FALSE: 緊急停止 ・TRUE: 緊急停止解除
ForcedStop_Signal	緊急停止信号	レディON	SIGNAL_SELECT 1424ページ SIGNAL_SELECT(信号選択)	LIST_READ_ONLY	緊急停止信号の入力状態を表示します。 緊急停止信号(ForcedStop_Signal)固有の出力を下記に示します。 ■入出力番号(StartIO) 常に「0」を表示します。 ■対象(Target) パラメータの取込み結果を表示します。 ■信号検出方法(Detection) パラメータの取込み結果を表示します。 ■補正時間(CompensationTime) 常に「0.0」を表示します。 ■フィルタ時間(FilterTime) パラメータの取込み結果を表示します。
Homing_Complete	原点復帰完了	即時	BOOL	LIST_READ_ONLY	原点復帰が正常に完了したか、完了していないかを表示します。 運転開始時、または原点復帰が必要なとき、FALSEになります。 ・FALSE: 原点復帰未完 ・TRUE: 原点復帰完了
Homing_Request	原点復帰要求	即時	BOOL	LIST_READ_ONLY	原点復帰が必要か、必要でないかを表示します。 原点復帰完了にてFALSEになります。 ・FALSE: 原点復帰要求なし ・TRUE: 原点復帰要求あり
Homing_Required	原点復帰要否設定	レディON	BOOL	LIST_READ_ONLY	軸の原点復帰が必要か、必要でないかを表示します。 FALSEの場合、現在位置復元時に原点復帰要求がTRUEになりません。 ・FALSE: 原点復帰不要 ・TRUE: 原点復帰必要
Homing_Status	原点復帰動作状態	演算周期	WORD(HEX)	LIST_READ_ONLY	ドライバの原点復帰状態を表示します。 ・0000H: 原点復帰始動中 ・0001H: 原点復帰始動待機状態、または原点復帰中断 ・0002H: 原点復帰完了(目標位置に未到達) ・0003H: 原点復帰正常完了 ・0004H: 原点復帰エラー発生(速度は0以外) ・0005H: 原点復帰エラー発生(速度は0) ・FFFFH: 原点復帰中以外のとき
HwStrokeLimit_FlsStatus	上限リミット信号状態	演算周期	BOOL	LIST_READ_ONLY	上限リミット信号(FLS)の検出状態を表示します。 ・FALSE: 上限リミット信号(FLS)検出中 ・TRUE: 上限リミット信号(FLS)非検出 信号の入力状態ではなく、信号検出方法、フィルタ時間を反映した検出状態を表示します。
HwStrokeLimit_Override	ハードウェアストロークリミットオーバーライド	始動時	STRING(15)	LIST_READ_ONLY	ハードウェアストロークリミットチェックの有効/無効切換え指令の受け付け状態を表示します。 ・DISABLE: チェック無効化 ・ONLY_INSIDE: 範囲内方向へのみチェック無効 ・上記以外: 無効化要求なし

変数名	名称	取込	データ型	属性	内容
HwStrokeLimit_RlsStatus	下限リミット信号状態	演算周期	BOOL	LIST_READ_ONLY	下限リミット信号(RLS)の検出状態を表示します。 • FALSE: 下限リミット信号(RLS)検出中 • TRUE: 下限リミット信号(RLS)非検出 信号の入力状態ではなく、信号検出方法、フィルタ時間を反映した検出状態を表示します。
InVelocity	目標速度到達	演算周期	BOOL	LIST_READ_ONLY	目標速度に到達したか、到達していないかを表示します。 軸グループ動作中、構成軸の目標速度到達(AxisName.Md.InVelocity)には、FALSEを表示します。 • FALSE: 未到達 • TRUE: 到達
Io_PosActualValue	オブジェクトデータ_PosActualValue	演算周期	DINT	LIST_READ_ONLY	オブジェクトデータPosActualValueの値を表示します。(デバイス機器から受信する位置フィードバック) 実エンコーダ軸、仮想エンコーダ軸の場合は、エンコーダから取得した入力パルスの値[エンコーダパルス単位]を表示します。
Io_PosEncoderResolution	オブジェクトデータ_PosEncoderResolution	演算周期	DWORD(UDINT)	LIST_READ_ONLY	オブジェクトデータPosEncoderResolutionの値を表示します。 実エンコーダ軸、仮想エンコーダ軸の場合は、同期エンコーダの分解能を表示します。
Io_Statusword	オブジェクトデータ_Statusword	演算周期	WORD(HEX)	LIST_READ_ONLY	オブジェクトデータStatuswordの値を表示します。 実エンコーダ軸の場合は、エンコーダの状態をモーションシステム側でエミュレートして表示します。
Io_TargetPos	オブジェクトデータ_TargetPos	演算周期	DINT	LIST_READ_ONLY	オブジェクトデータTargetPosの値を表示します。(デバイス機器に送信する位置指令) 仮想ドライブ軸の場合は、累積現在位置をドライバ単位の指令値に変換した値を表示します。
Io_TargetVelocity	オブジェクトデータ_TargetVelocity	演算周期	DINT	LIST_READ_ONLY	デバイス機器に送信する速度指令の値を表示します。
Io_TorqueActualValue	オブジェクトデータ_TorqueActualValue	演算周期	INT	LIST_READ_ONLY	デバイス機器から受信するトルクフィードバックの値を表示します。
Io_VelActualValue	オブジェクトデータ_VelActualValue	演算周期	DINT	LIST_READ_ONLY	デバイス機器から受信する速度フィードバックの値を表示します。
JerkLimit	ジャーク制限値	レディオン	LREAL	LIST_READ_ONLY	ジャーク制限値を格納します。
OperationCycle	制御周期	システム起動時	INT	LIST_READ_ONLY	軸の制御演算周期を格納します。 • 0: 軸未初期化/制御周期設定範囲外 • 1: 第1演算周期で動作 • 2: 第2演算周期で動作 • 3: 第3演算周期で動作
OverrunOperation	オーバラン時動作設定	レディオン	INT (MC_DECEL_STOP_MODE)  1451ページ MC_DECEL_STOP_MODE	LIST_READ_ONLY	オーバラン動作設定の入力状態を表示します。 • 1: 即停止(ImmediateStop) • 2: 現在の加減速度を続行(KeepCurrentAcc)
PosRestoration_Status	現在位置復元状態	サービス	INT (MC_POS_RESTORATION_STATUS)  1452ページ MC_POS_RESTORATION_STATUS	LIST_READ_ONLY	現在位置復元状態を表示します。 現在位置復元要求を実行する必要がある場合、デバイス局接続時に「1: 現在位置復元復元要求待ち(WaitingRequest)」になります。 デバイス機器切断時に「0: 未実施(NotExecute)」となります。 • 0: 現在位置復元未実施(NotExecute) • 1: 現在位置復元復元要求待ち(WaitingRequest) • 2: インクリメンタルシステムで復元完了(RestoredInIncSystem) • 3: 絶対位置システムで復元完了(原点復帰未完了)(RestoredInAbsSystemUnHomed) • 4: 絶対位置システムで復元完了(RestoredInAbsSystem)

変数名	名称	取込	データ型	属性	内容
ProfileID	実行プロファイルID 番号	演算周期	WORD(UINT)	LIST_READ_ ONLY	現在実行しているプロファイルIDが格納されます。 以下のタイミングで更新されます。 ■指定したCamTableID(カムテーブルID)を格納されます。 ・MC_CamInを実行し、従軸の状態が「7:同期運転中 (SynchronizedMotion)」のときタイミング ・CamTableID(カムテーブルID)を変更したとき ■0クリアされます。 ・Periodic((繰り返し動作)が「FALSE:単発動作」かつ Options bit 16(1サイクル運転後の動作指定)がOFF(終了)でEndOfProfileがTRUEになったとき ・停止要因が発生したとき
SetAcceleration	指令現在加速度	演算周期	LREAL	LIST_READ_ ONLY	指令出力加速度(指令現在速度の差分から算出した加減速度)を格納します。*1(単位: U/s ²) 符号にて加速中か減速中かを表します。 目標速度に到達すると「0.0」を格納します。 軸グループ動作中、構成軸の指令現在加速度 (SetAcceleration)を格納します。 ■移動方向が正方向の場合(アドレス増加方向) ・0.0: 停止中または定速中 ・プラス符号: 加速中 ・マイナス符号: 減速中 ■移動方向が負方向の場合(アドレス減少方向) ・0.0: 停止中または定速中 ・プラス符号: 減速中 ・マイナス符号: 加速中
SetPosition	指令現在位置	演算周期	LREAL	LIST_READ_ ONLY	現在位置のアドレスを格納します。*1 指令現在位置は、累積現在位置をリングカウンタ範囲で丸めた値となります。
SetVelocity	指令現在速度	演算周期	LREAL	LIST_READ_ ONLY	指令出力速度を格納します。*1 指令現在位置の差分から算出した速度です。 正方向(アドレス増加方向)へ移動する場合は正の値、負方向(アドレス減少方向)へ移動する場合は負の値となります。 軸グループ動作中、構成軸の指令現在速度を格納します。
SlaveEmulate_Enable	軸エミュレート中	サービス	BOOL	LIST_READ_ ONLY	ドライバエミュレート運転が有効か、無効かを表示します。 ・FALSE: 無効 ・TRUE: 有効
StartableAtUnhomed	原点復帰未完時始動許可	レディ ON	BOOL	LIST_READ_ ONLY	原点復帰未完時の始動状態を表示します。 ・FALSE: 許可しない ・TRUE: 許可する
StopMode_Deceleration	停止時減速度	レディ ON	LREAL	LIST_READ_ ONLY	停止時減速度の入力状態を表示します。
StopMode_DecelerationCurve	減速停止時停止処理選択	レディ ON	INT (MC_STOP_CURVE_MODE) 📖 1451ページ MC_STOP_CURVE_MODE	LIST_READ_ ONLY	減速停止時停止処理選択の入力状態を表示します。 ・1: 減速カーブ再作成(OverrideCurve)
StopMode_General	停止要因発生時停止選択	レディ ON	INT (MC_DECEL_STOP_MODE) 📖 1451ページ MC_DECEL_STOP_MODE	LIST_READ_ ONLY	停止要因発生時停止選択の入力状態を表示します。 ・1: 即停止(ImmediateStop) ・2: 現在の加減速度を続行(KeepCurrentAcc) ・3: 加減速度を代替(AlternativeAcc)
StopMode_HwStrokeLimit	ハードウェアストロークリミットエラー発生時停止選択	レディ ON	INT (MC_DECEL_STOP_MODE) 📖 1451ページ MC_DECEL_STOP_MODE	LIST_READ_ ONLY	ハードウェアストロークリミットエラー発生時停止選択の入力状態を表示します。 ・1: 即停止(ImmediateStop) ・2: 現在の加減速度を続行(KeepCurrentAcc) ・3: 加減速度を代替(AlternativeAcc)
StopMode_ServoOff	運転中サーボOFF指令時処理選択	レディ ON	INT (MC_DECEL_STOP_MODE) 📖 1451ページ MC_DECEL_STOP_MODE	LIST_READ_ ONLY	運転中サーボOFF指令時処理選択の入力状態を表示します。 ・0: 無視(Ignore) ・4: 即停止後サーボOFF(ServoOffAfterImmediateStop) ・5: 減速停止後サーボOFF(ServoOffAfterDecelStop)

変数名	名称	取込	データ型	属性	内容
StopMode_SwStrokeLimit	ソフトウェアストロークリミットエラー発生時停止選択	レディオン	INT (MC_DECEL_STOP_MODE) 1451ページ MC_DECEL_STOP_MODE	LIST_READ_ONLY	ソフトウェアストロークリミットエラー発生時停止選択の入力状態を表示します。 ・1: 即停止(ImmediateStop) ・2: 現在の加減速度を続行(KeepCurrentAcc) ・3: 加減速度を代替(AlternativeAcc)
StopOption_DriverTargetIgnored	ドライバ指令破棄検出設定	レディオン	BOOL	LIST_READ_ONLY	ドライバ指令破棄検出設定の状態を表示します。 ・FALSE: 検出無効 ・TRUE: 検出有効
StopSignal	停止信号	レディオン	SIGNAL_SELECT 1424ページ SIGNAL_SELECT(信号選択)	LIST_READ_ONLY	外部入力信号の停止信号(STOP)の入力状態を表示します。停止信号(StopSignal)固有の出力を下記に示します。 ■入出力番号(StartIO) 常に「0」を表示します。 ■対象(Target) パラメータの取込み結果を表示します。 ■信号検出方法(Detection) パラメータの取込み結果を表示します。 ■補正時間(CompensationTime) 常に「0.0」を表示します。 ■フィルタ時間(FilterTime) パラメータの取込み結果を表示します。
StopStatus	停止状態	演算周期	BOOL	LIST_READ_ONLY	外部入力信号の停止信号(STOP)の入力状態を表示します。 ・FALSE: 停止処理解除 ・TRUE: 停止処理中
SwStrokeLimit_Lower	ソフトウェアストロークリミット下限値	レディオン	LREAL	LIST_READ_ONLY	ソフトウェアストロークリミット下限値を表示します。
SwStrokeLimit_Override	ソフトウェアストロークリミットオーバーライド	始動時	STRING(15)	LIST_READ_ONLY	ソフトウェアストロークリミット有効/無効状態を表示します。 ・DISABLE: チェック無効 ・ONLY_INSIDE: 範囲内方向へのみチェック無効 ・上記以外: チェック有効(無効化要求なし)
SwStrokeLimit_Target	ソフトウェアストロークリミット対象	レディオン	INT (MC_POS_SOURCE) 1452ページ MC_POS_SOURCE	LIST_READ_ONLY	ソフトウェアストロークリミットの対象を表示します。 ・-1: 無効(Invalid) ・1: 指令現在位置(SetPosition) ・3: 送り機械位置(FeedMachinePosition)
SwStrokeLimit_Upper	ソフトウェアストロークリミット上限値	レディオン	LREAL	LIST_READ_ONLY	ソフトウェアストロークリミット上限値を表示します。
TargetAcceleration	目標加速度	演算周期	LREAL	LIST_READ_ONLY	加速中に最大となる加速度を格納します。(単位: U/s ²) 軸グループ動作中, 構成軸の目標加速度(AxisName.Md.TargetAcceleration)には「0.0」を格納します。
TargetDeceleration	目標減速度	演算周期	LREAL	LIST_READ_ONLY	減速中に最大となる減速度を格納します。(単位: U/s ²) 軸グループ動作中, 構成軸の目標減速度(AxisName.Md.TargetDeceleration)には「0.0」を格納します。
TargetVelocity	目標速度	演算周期	LREAL	LIST_READ_ONLY	オーバーライド, 速度制限値を考慮した実際の目標速度を格納します。 制御が完了すると「0.0」を格納します。 軸グループ動作中, 構成軸の目標速度(AxisName.Md.TargetVelocity)には「0.0」を格納します。 ■位置決め制御の場合 移動方向に関係なく「0.0」以上の値となります。 ■位置決め制御以外の場合 正方向(アドレス増加方向)へ移動する場合は正の値, 負方向(アドレス減少方向)へ移動する場合は負の値となります。
TorqueLimit_Negative	負方向トルク制限値	演算周期	LREAL	LIST_READ_ONLY	現在有効となっている負方向トルク制限値を表示します。
TorqueLimit_Positive	正方向トルク制限値	演算周期	LREAL	LIST_READ_ONLY	現在有効となっている正方向トルク制限値を表示します。

変数名	名称	取込	データ型	属性	内容
Unit_Position	位置指令単位	レディ ON	DWORD(HEX)	LIST_READ_ONLY	モーション制御で使用する位置指令単位を表示します。 • 00000000H: pulse • 00010000H: m • FD010000H: mm • FA010000H: μm • F7010000H: nm • 00410000H: degree • FD410000H: $\times 10^{-3}$degree • FA410000H: $\times 10^{-6}$degree • 00B40000H: Revolution • 00C00000H: inch • 00FF0000H: 任意単位
Unit_PositionDisplay	位置指令単位表示	レディ ON	WSTRING(31)	LIST_READ_ONLY	現在制御中の位置指令単位を文字列で表示します。
Unit_PositionString	位置指令単位文字列	レディ ON	WSTRING(31)	LIST_READ_ONLY	位置指令単位(AxisName.Pr.Unit_Position)で「00FF0000H: 任意単位」を設定した場合に設定した位置指令単位文字列(AxisName.Pr.Unit_PositionString)の制御中の位置指令単位を文字列で表示します。
Unit_Velocity	速度指令単位	レディ ON	DWORD(HEX)	LIST_READ_ONLY	モーション制御で使用する速度指令単位を表示します。 • 00000300H: U/s • 00004700H: U/min • FD000300H: U/ms • FA000300H: U/μs • F7000300H: U/ns
Unit_VelocityDisplay	速度指令単位表示	レディ ON	WSTRING(31)	LIST_READ_ONLY	現在制御中の速度指令単位を文字列で出力します。
UseInGroup	軸グループ使用中	演算周期	BOOL	LIST_READ_ONLY	指定した軸が軸グループで使用中か、未使用かを表示します。 • FALSE: 未使用 • TRUE: 使用中
VelocityBias	速度バイアス値	レディ ON	LREAL	LIST_READ_ONLY	レディ ON時に取り込んだ速度バイアス値(AxisName.Pr.VelocityBias)の設定値を格納します。
VelocityLimit_Negative	負方向速度制限値	レディ ON	LREAL	LIST_READ_ONLY	アドレス減少方向速度制限値を格納します。
VelocityLimit_Operation	速度制限値オーバー時動作設定	レディ ON	INT (MC_VELOCITY_LIMIT_MODE) 1450ページ MC_VELOCITY_LIMIT_MODE	LIST_READ_ONLY	制御中の速度制限値オーバー時の動作を表示します。 • 0: 無視(Ignore) • 3: 即停止(ImmediateStop)
VelocityLimit_Positive	正方向速度制限値	レディ ON	LREAL	LIST_READ_ONLY	アドレス増加方向速度制限値を格納します。
VelocityOverride	速度オーバライド係数	演算周期	LREAL	LIST_READ_ONLY	速度オーバライド係数を表示します。
Warning	軸警告検出	即時	BOOL	LIST_READ_ONLY	軸警告の状態を表示します。 • FALSE: 軸警告なし • TRUE: 軸警告あり
WarningID	軸警告コード	即時	WORD(HEX)	LIST_READ_ONLY	軸警告が発生したときのイベントコードを格納します。

*1 浮動小数点誤差が発生するため、格納する値には誤差を含みます。

AxisName.Cd.(軸制御データ)

変数名	名称	取込	データ型	属性	範囲
AccelerationOverride	加速度オーバライド係数	演算周期	LREAL	LIST_READ_ONLY	加速度オーバライド係数を設定します。 • 0.01~10.00
Encoder_Connect	接続指令	演算周期	BOOL	LIST_READ_ONLY	仮想エンコーダ軸入力の接続状態の有効/無効を切換えます。 • FALSE: 無効 • TRUE: 有効
Encoder_CounterDisable	カウンタ無効中	演算周期	BOOL	LIST_READ_ONLY	カウンタディセーブルに切換えます。 • FALSE: 無実行 • TRUE: カウンタディセーブル要求実行
Encoder_InputValue	エンコーダ入力値	演算周期	DINT	LIST_READ_ONLY	仮想エンコーダ軸の入力値として使用する値を逐次設定します。

変数名	名称	取込	データ型	属性	範囲
ErrorReset	軸エラーリセット	サービス	BOOL	LIST_READ_ONLY	軸のエラー、警告、ドライブユニットエラーをリセットします。 エラーリセットが完了するとFALSEになります。 • FALSE: 無実行 • TRUE: エラーリセット実行
FollowupDisable	フォローアップ無効中	演算周期	BOOL	LIST_READ_ONLY	フォローアップを無効化します。 • FALSE: 無実行 • TRUE: フォローアップ無効化要求実行
Homing_ClearRequest	原点復帰要求クリア	演算周期	BOOL	LIST_READ_ONLY	原点復帰要求をクリアします。 原点復帰要求(Homing_Request)をFALSEにした後、原点復帰要求クリア(Homing_ClearRequest)は、自動的にFALSEになります。 • FALSE: 無実行 • TRUE: 原点復帰要求クリア実行
HwStrokeLimit_Override	ハードウェアストロークリミットオーバーライド	始動時	STRING(15)	LIST_READ_ONLY	ハードウェアストロークリミットチェックの有効/無効を一時的に切換えます。 • DISABLE: チェック無効化 • ONLY_INSIDE: 範囲内方向へのみチェック無効 • 上記以外: 無効化要求なし
PosRestoration_Request_EraseAbsPos	絶対位置消失・復元要求	サービス	INT (MC_POS_RESTORATION_REQUEST)  1453ページ MC_POS_RESTORATION_REQUEST	LIST_READ_ONLY	絶対位置システムが有効の場合に、現在位置復元を実行します。 現在位置復元状態(AxisName.Md.PosRestoration_Status)が「0: 未実施(NotExecute)」, または「1: 復元要求待ち(WaitingRequest)」の場合のみ、本変数は有効です。復元完了後に本変数を設定しても、現在位置復元は実施しません。 ドライブユニットにMELSERVOを使用している場合、本変数は無効になります。コントローラが自動でドライブユニットの状態をチェックし、現在位置復元を実施します。 • 0: 無処理(Auto) • 1: 現在位置復元実行(RestorationRequest) • 2: 絶対位置消失状態で現在位置復元実行(RestorationRequest_EraseAbsPos) 設定によって下記ようになります。 • 現在位置復元状態(AxisName.Md.PosRestoration_Status)が「1: 復元要求待ち(WaitingRequest)」のとき、デバイス局の状態をチェックし、復元が可能のとき、「1: 現在位置復元実行(RestorationRequest)」を設定してください。また、絶対位置が消失している場合は「2: 絶対位置消失状態で現在位置復元実行(RestorationRequest_EraseAbsPos)」を設定してください。 • 「2: 絶対位置消失状態で現在位置復元実行(RestorationRequest_EraseAbsPos)」を設定した場合、絶対位置消失状態となり、現在位置復元の実施後、原点復帰要求(AxisName.Md.Homing_Request)がTRUEになります。
SwStrokeLimit_Override	ソフトウェアストロークリミットオーバーライド	始動時	STRING(15)	LIST_READ_ONLY	ソフトウェアストロークリミットのチェック有効/無効を一時的に切換えます。 • DISABLE: チェック無効 • ONLY_INSIDE: 範囲内方向へのみチェック無効 • 上記以外: チェック有効(無効化要求なし)
TorqueLimit_Negative	負方向トルク制限値	演算周期	LREAL	LIST_READ_ONLY	負方向トルク制限値を設定します。 • 0.0~1000.0[%]
TorqueLimit_Positive	正方向トルク制限値	演算周期	LREAL	LIST_READ_ONLY	正方向トルク制限値を設定します。 • 0.0~1000.0[%]
VelocityOverride	速度オーバーライド係数	演算周期	LREAL	LIST_READ_ONLY	速度オーバーライド係数を設定します。 • 0.00~10.00

軸変数で各軸種別ごとに設定可能な変数

軸変数で使用する変数は、各軸種別ごとに設定可能な変数が異なります。
各軸種別で設定可能な変数を下記に示します。

■AxisName.AxisRef.(軸情報)

○: 設定可, ×: 設定不可

変数名	名称	軸種別				
		実ドライブ軸	実エンコーダ軸	仮想ドライブ軸	仮想エンコーダ軸	仮想連結軸
AxisNo	軸No.	○	○	○	○	○
StartIO	入出力No.	○	○	○	○	○

■AxisName.PrConst.(軸パラメータ定数)

○: 設定可, ×: 設定不可

変数名	名称	軸種別				
		実ドライブ軸	実エンコーダ軸	仮想ドライブ軸	仮想エンコーダ軸	仮想連結軸
AddressOfStation	局アドレス設定	○	○	×	×	×
AxisType	軸種別設定	○	○	○	○	○
Encoder_AxisType	実エンコーダ軸タイプ設定	×	○	×	×	×
Encoder_CounterDisableSignal	カウンタ無効化信号	×	○	×	○	×
Encoder_RingCout_LowerValue	エンコーダリングカウンタ下限値	×	×	×	○	×
Encoder_RingCout_UpperValue	エンコーダリングカウンタ上限値	×	×	×	○	×
HwStrokeLimit_FlsSignal	上限リミット信号	○	×	○	×	×
HwStrokeLimit_RlsSignal	下限リミット信号	○	×	○	×	×
OperationCycle	制御周期設定	○	○	○	○	○
PosRestoration_AbsPosBase	絶対位置基準設定	○	×	×	×	×
PosRestoration_AbsPosEnable	絶対位置管理設定	○	○	○	○	○
PosRestoration_RestorationBase	現在位置復元基準位置設定	○	×	×	×	×
RingCount_Enable	リングカウンタ有効選択	○	○	○	○	○
RingCount_LowerValue	リングカウンタ下限値	○	○	○	○	○
RingCount_UpperValue	リングカウンタ上限値	○	○	○	○	○
SlaveEmulate_Enable	軸エミュレート有効	○	×	×	×	×
SlaveObject	オブジェクトデータ	○	×	×	○	×
TorqueLimit_Max	トルク制限最大値	○	×	×	×	×
TorqueLimit_NegativeInitial	負方向トルク制限初期値	○	×	×	×	×
TorqueLimit_PositiveInitial	正方向トルク制限初期値	○	×	×	×	×

■AxisName.Pr.(軸パラメータ)

○: 設定可, ×: 設定不可

変数名	名称	軸種別				
		実ドライブ軸	実エンコーダ軸	仮想ドライブ軸	仮想エンコーダ軸	仮想連結軸
AccelerationLimit	加速度制限値	○	×	○	×	×
AccelerationZeroBehavior	始動時加減速度0指定時動作選択	○	×	○	×	×
CmdInPos_Width	指令インポジション幅	○	×	○	×	×
DecelerationLimit	減速度制限値	○	×	○	×	×
Drive_UnitConvRatioNum	ドライブ単位変換分子	○	○	×	○	×
Drive_UnitConvRatioDen	ドライブ単位変換分母	○	○	×	○	×
ForcedStop_Signal	緊急停止信号	○	×	○	×	×
Homing_Required	原点復帰要否設定	○	○	○	○	○
JerkLimit	ジャーク制限値	○	×	○	×	×
OverrunOperation	オーバラン時動作設定	○	×	○	×	×
StartableAtUnhomed	原点復帰未完時始動許可	○	○	○	○	○
StopMode_Deceleration	停止時減速度	○	×	○	×	×
StopMode_DecelerationCurve	減速停止時停止処理選択	○	×	○	×	×
StopMode_General	停止要因発生時停止選択	○	×	○	×	×
StopMode_HwStrokeLimit	ハードウェアストロークリミットエラー発生時停止選択	○	×	○	×	×
StopMode_ServoOff	運転中サーボOFF指令時処理選択	○	×	○	×	×
StopMode_SwStrokeLimit	ソフトウェアストロークリミットエラー発生時停止選択	○	×	○	×	×
StopOption_DriverTargetIgnored	ドライブ指令破棄検出設定	○	×	×	×	×
StopSignal	停止信号	○	×	○	×	×
SwStrokeLimit_Lower	ソフトウェアストロークリミット下限値	○	×	○	×	×
SwStrokeLimit_Target	ソフトウェアストロークリミット対象	○	×	○	×	×
SwStrokeLimit_Upper	ソフトウェアストロークリミット上限値	○	×	○	×	×
Unit_Position	位置指令単位	○	○	○	○	○
Unit_PositionString	位置指令単位文字列	○	○	○	○	○
Unit_Velocity	速度指令単位	○	○	○	○	○
VelocityBias	速度バイアス値	○	×	○	×	○
VelocityLimit_Negative	負方向速度制限値	○	×	○	×	×
VelocityLimit_OverOperation	速度制限値オーバ時動作設定	○	×	○	×	×
VelocityLimit_Positive	正方向速度制限値	○	×	○	×	×

■AxisName.Md.(軸モニタデータ)

○: 設定可, ×: 設定不可

変数名	名称	軸種別				
		実ドライブ軸	実エンコーダ軸	仮想ドライブ軸	仮想エンコーダ軸	仮想連結軸
AccelerationLimit	加速度制限値	○	×	○	×	×
AccelerationOverride	加速度オーバーライド係数	○	×	○	×	×
AccelerationZeroBehavior	始動時加減速度0指定時動作選択	○	×	○	×	×
ActualPosition	フィードバック位置	○	○	×	×	×
ActualVelocity	フィードバック速度	○	○	×	×	×
Analyzing	解析中	○	○	○	○	○
AutoDeceleration	自動減速中	○	×	○	×	×
AxisName	軸名称	○	○	○	○	○
AxisStatus	軸状態	○	○	○	○	○
BufferingFBs	バッファリングFB数	○	×	○	×	×
CmdInPos	指令インポジション	○	×	○	×	×
CmdInPos_Width	指令インポジション幅	○	×	○	×	×
CommandedAcceleration	指定加速度	○	×	○	×	×
CommandedDeceleration	指定減速度	○	×	○	×	×
CommandedJerk	指定ジャーク	○	×	○	×	×
CommandedPosition	指定位置	○	×	○	×	×
CommandedVelocity	指定速度	○	×	○	×	×
Cst_SetTorque	cst/ct時指令現在トルク	○	×	×	×	×
Cst_TargetTorque	cst/ct時目標トルク	○	×	×	×	×
CumulativePosition	累積現在位置	○	○	○	○	○
DecelerationLimit	減速度制限値	○	×	○	×	×
Drive_RPDO[1..64]	RPDO(モーションシステム→デバイス機器)マッピング	○	×	×	×	×
Drive_TPDO[1..64]	TPDO(デバイス機器→モーションシステム)マッピング	○	×	×	×	×
Drive_UnitConvRatioNum	ドライブ単位変換分子	○	○	×	×	×
Drive_UnitConvRatioDen	ドライブ単位変換分母	○	○	×	×	×
Driver_Mode	ドライブ制御モード	○	×	×	×	×
Driver_ReadyOn	ドライブレディ ON 状態	○	×	×	×	×
Driver_ServoOn	ドライブサーボON 状態	○	×	×	×	×
Driver_State	ドライブ状態	○	×	×	×	×
DriverError	ドライブユニットエラー 検出	○	×	×	×	×
DriverErrorID	ドライブユニットエラーコード	○	×	×	×	×
DriverErrorDetailID	ドライブユニットエラー詳細コード	○	×	×	×	×
Encoder_Connected	接続状態	×	×	×	○	×
Encoder_CounterDisable	カウンタ無効中	×	○	×	○	×
Error	軸エラー 検出	○	○	○	○	○
ErrorID	軸エラーコード	○	○	○	○	○
FeedMachinePosition	送り機械位置	○	×	×	×	×
FollowupDisable	フォローアップ無効中	○	×	×	×	×

変数名	名称	軸種別				
		実ドライブ軸	実エンコーダ軸	仮想ドライブ軸	仮想エンコーダ軸	仮想連結軸
ForcedStop_Released	緊急停止解除中	○	×	○	×	×
ForcedStop_Signal	緊急停止信号	○	×	○	×	×
Homing_Complete	原点復帰完了	○	○	○	○	○
Homing_Request	原点復帰要求	○	○	○	○	○
Homing_Required	原点復帰要否設定	○	○	○	○	○
Homing_Status	原点復帰動作状態	○	×	×	×	×
HwStrokeLimit_FlsStatus	上限リミット信号状態	○	×	○	×	×
HwStrokeLimit_Override	ハードウェアストロークリミットオーバーライド	○	×	○	×	×
HwStrokeLimit_RlsStatus	下限リミット信号状態	○	×	○	×	×
InVelocity	目標速度到達	○	×	○	×	×
Io_PosActualValue	オブジェクトデータ_PosActualValue	○	○	×	○	×
Io_PosEncoderResolution	オブジェクトデータ_PosEncoderResolution	×	○	×	○	×
Io_Statusword	オブジェクトデータ_Statusword	○	○	×	×	×
Io_TargetPos	オブジェクトデータ_TargetPos	○	×	○	×	×
Io_TargetVelocity	オブジェクトデータ_TargetVelocity	○	×	×	×	×
Io_TorqueActualValue	オブジェクトデータ_TorqueActualValue	○	×	×	×	×
Io_VelActualValue	オブジェクトデータ_VelActualValue	○	×	×	×	×
JerkLimit	ジャーク制限値	○	×	○	×	×
OperationCycle	制御周期	○	○	○	○	○
OverrunOperation	オーバラン時動作設定	○	×	○	×	×
PosRestoration_Status	現在位置復元状態	○	○	○	○	○
ProfileID	実行プロファイルID番号	○	×	○	×	○
SetAcceleration	指令現在加速度	○	×	○	×	×
SetPosition	指令現在位置	○	○	○	○	○
SetVelocity	指令現在速度	○	×	○	○	×
SlaveEmulate_Enable	軸エミュレート中	○	×	×	×	×
StartableAtUnhomed	原点復帰未完時始動許可	○	○	○	○	○
StopMode_Deceleration	停止時減速度	○	×	○	×	×
StopMode_DecelerationCurve	減速停止時停止処理選択	○	×	○	×	×
StopMode_General	停止要因発生時停止選択	○	×	○	×	×
StopMode_HwStrokeLimit	ハードウェアストロークリミットエラー発生時停止選択	○	×	○	×	×
StopMode_ServoOff	運転中サーボOFF指令時処理選択	○	×	○	×	×
StopMode_SwStrokeLimit	ソフトウェアストロークリミットエラー発生時停止選択	○	×	○	×	×
StopOption_DriverTargetIgnored	ドライバ指令破棄検出設定	○	×	×	×	×
StopSignal	停止信号	○	×	○	×	×

変数名	名称	軸種別				
		実ドライブ軸	実エンコーダ軸	仮想ドライブ軸	仮想エンコーダ軸	仮想連結軸
StopStatus	停止状態	○	×	○	×	×
SwStrokeLimit_Lower	ソフトウェアストロークリミット下限値	○	×	○	×	×
SwStrokeLimit_Override	ソフトウェアストロークリミットオーバーライド	○	×	○	×	×
SwStrokeLimit_Target	ソフトウェアストロークリミット対象	○	×	○	×	×
SwStrokeLimit_Upper	ソフトウェアストロークリミット上限値	○	×	○	×	×
TargetAcceleration	目標加速度	○	×	○	×	×
TargetDeceleration	目標減速度	○	×	○	×	×
TargetVelocity	目標速度	○	×	○	×	×
TorqueLimit_Negative	負方向トルク制限値	○	×	×	×	×
TorqueLimit_Positive	正方向トルク制限値	○	×	×	×	×
Unit_Position	位置指令単位	○	○	○	○	○
Unit_PositionDisplay	位置指令単位表示	○	○	○	○	○
Unit_PositionString	位置指令単位文字列	○	○	○	○	○
Unit_Velocity	速度指令単位	○	○	○	○	○
Unit_VelocityDisplay	速度指令単位表示	○	○	○	○	○
UseInGroup	軸グループ使用中	○	○	○	○	○
VelocityBias	速度バイアス値	○	×	○	×	○
VelocityLimit_Negative	負方向速度制限値	○	×	○	×	×
VelocityLimit_OverOperation	速度制限値オーバー時動作設定	○	×	○	×	×
VelocityLimit_Positive	正方向速度制限値	○	×	○	×	×
VelocityOverride	速度オーバーライド係数	○	×	○	×	×
Warning	軸警告検出	○	○	○	○	○
WarningID	軸警告コード	○	○	○	○	○

■AxisName.Cd.(軸制御データ)

○: 設定可, ×: 設定不可

変数名	名称	軸種別				
		実ドライブ軸	実エンコーダ軸	仮想ドライブ軸	仮想エンコーダ軸	仮想連結軸
AccelerationOverride	加速度オーバーライド係数	○	×	○	×	×
Encoder_Connect	接続指令	×	×	×	○	×
Encoder_CounterDisable	カウンタ無効中	×	○	×	○	×
Encoder_InputValue	エンコーダ入力値	×	×	×	○	×
ErrorReset	軸エラーリセット	○	○	○	○	○
FollowupDisable	フォローアップ無効中	○	×	×	×	×
Homing_ClearRequest	原点復帰要求クリア	○	○	○	○	○
HwStrokeLimit_Override	ハードウェアストロークリミットオーバーライド	○	×	○	×	×
PosRestoration_Request_EraseAbsPos	絶対位置消失・復元要求	○	×	×	×	×
SwStrokeLimit_Override	ソフトウェアストロークリミットオーバーライド	○	×	○	×	×
TorqueLimit_Negative	負方向トルク制限値	○	×	×	×	×
TorqueLimit_Positive	正方向トルク制限値	○	×	×	×	×
VelocityOverride	速度オーバーライド係数	○	×	○	×	×

軸グループ変数

軸グループ変数のデータ型はAXES_GROUP型で表します。
各軸グループのデータ型は、下記のメンバを所有しています。

メンバ名	データ型	内容	参照
AxesGroupRef	AXES_GROUP_REF	モーション制御用ファンクションブロックの入力/出力用のデータ構造です。	1402ページ AxesGroupName.AxesGroupRef.(軸グループ情報)
PrConst	AXES_GROUP_PRM_CONST	軸グループのパラメータデータ(定数)を格納します。 軸グループ変数初期化時に設定値を展開します。 軸グループ変数初期化後に制御への再取込みを実施しません。	1402ページ AxesGroupName.PrConst.(軸グループパラメータ定数)
Pr	AXES_GROUP_PRM	軸グループのパラメータデータを格納します。 軸グループ変数生成時に初期値を展開します。 軸グループ変数初期化後も制御への再取込みを実施します。 パラメータによって制御への取込みタイミングが異なります。	1402ページ AxesGroupName.Pr.(軸グループパラメータ)
Md	AXES_GROUP_MONI	軸グループのモニタデータを格納します。 モニタデータごとに定められた周期でリフレッシュを実施します。	1404ページ AxesGroupName.Md.(軸グループモニタデータ)
Cd	AXES_GROUP_CMD	軸グループ制御用指令データを格納します。 制御演算周期ごとに最新の値を取得し制御に使用します。	1407ページ AxesGroupName.Cd.(軸グループ制御データ)

AxesGroupName.AxesGroupRef.(軸グループ情報)

変数名	名称	取込	データ型	属性	内容
GroupNo	軸グループNo.	—	WORD(UINT)	LIST_WRITE _BACK	軸グループNo.を設定します。 • 0: 未設定 • 1~10000: 設定軸グループNo.
StartIO	入出力No.	—	WORD(HEX)	LIST_WRITE _BACK	入出力No.を設定します。

AxesGroupName.PrConst.(軸グループパラメータ定数)

変数名	名称	取込	データ型	属性	内容
AxesGroupType	軸グループ種別設定	システム 起動時	INT (MC_AXES_GROUP_TYPE) 1452ページ MC_AXES_GROUP_TYPE	LIST_WRITE _BACK	軸グループ種別を設定します。 • 0: 標準(Standard)

AxesGroupName.Pr.(軸グループパラメータ)

変数名	名称	取込	データ型	属性	内容
AccelerationLimit	加速度制限値	レディ ON	LREAL	LIST_WRITE _BACK	加速度制限値を設定します。 • 0.0000, 0.0001~2147483647.0 設定によって下記ようになります。 • 加減速方式が「1: 加減速時間一定方式(mcFixedTime)」の場合は、指定した加減速時間から算出した加速度に対して制限を行います。 • 「0.0000」の場合は、加速度の制限を行いません。 • 0.0001未満の正数を設定した場合、「0.0000」として取込みます。
AccelerationZeroBehavior	始動時加減速度0指定時動作選択	レディ ON	INT (MC_ACC_ZERO_MODE) 1452ページ MC_ACC_ZERO_MODE	LIST_WRITE _BACK	始動時に、加速度、減速度、または加減速時間に「0.0」を設定した場合の動作を設定します。 • -1: エラー (始動しない)(ACCError) • 1: 最大加減速(MaximumAcceleration)
Axis[1..16]	構成軸	レディ ON	AXIS_REF 1378ページ AxisName.AxisRef.(軸情報)	LIST_WRITE _BACK	軸グループを構成する軸情報(AxisName.AxisRef)の軸No.(AxisNo)を設定します。

変数名	名称	取込	データ型	属性	内容
CmdInPos_Width	指令インポジション幅	レディ ON	LREAL	LIST_WRITE _BACK	軸グループの合成軸上の指令インポジション幅を設定します。 • 0.0: 機能無効 • 0.000000001~10000000000.0: 機能有効 0.000000001未満の正数を設定した場合、「0.0」として取込みます。
DecelerationLimit	減速度制限値	レディ ON	LREAL	LIST_WRITE _BACK	減速度制限値を設定します。 • 0.0000, 0.0001~2147483647.0 設定によって下記ようになります。 • 加減速方式が「1: 加減速時間一定方式(mcFixedTime)」の場合は、指定した加減速時間から算出した減速度に対して制限を行います。 • 「0.0000」の場合は、減速度の制限を行いません。 • 0.0001未満の正数を設定した場合、「0.0000」として取込みます。
JerkLimit	ジャーク制限値	レディ ON	LREAL	LIST_WRITE _BACK	ジャーク制限値を設定します。 • 0.0000, 0.0001~2147483647.0 設定によって下記ようになります。 • 「0.0000」の場合は、ジャーク制限を行いません。 • 0.0001未満の正数を設定した場合、「0.0000」として取込みます。
OverrunOperation	オーバラン時動作設定	レディ ON	INT (MC_DECEL_STOP_MODE) 📖 1451ページ MC_DECEL_STOP_MODE	LIST_WRITE _BACK	動作中に目標位置をオーバした場合の動作を設定します。 • 1: 即停止(ImmediateStop)
StopMode_Deceleration	停止時減速度	レディ ON	LREAL	LIST_WRITE _BACK	停止要因発生による減速停止時の減速度/減速時間を設定します。 ■加速度/減速度を指定する加減速方式(加減速度指定方式)の場合 • 0.0000, 0.0001~2147483647.0 ■加減速時間を指定する加減速方式(加減速時間一定方式)の場合 • 0.000000, 0.000001~8400.0 設定によって下記ようになります。 • 始動時加減速度0指定時動作選択 (AccelerationZeroBehavior)に関係なく、「0.0」の場合は即停止します。 • 停止要因発生時に上限値で範囲外となった場合は上限値でクランプし、下限値で範囲外となった場合は「0(即停止)」として扱います。
StopMode_DecelerationCurve	減速停止時停止処理選択	レディ ON	INT (MC_STOP_CURVE_MODE) 📖 1451ページ MC_STOP_CURVE_MODE	LIST_WRITE _BACK	減速中(停止要因、自動減速含む)に、停止要因が発生した場合の動作を設定します。 • 1: 減速カーブ再作成(OverrideCurve)
StopMode_ErrorInGroup	軸停止要因発生時構成軸動作選択	レディ ON	INT (MC_DECEL_STOP_MODE) 📖 1451ページ MC_DECEL_STOP_MODE	LIST_WRITE _BACK	軸グループで運転中に、構成軸でドライバサーボOFFとなり、即停止の軸エラーが発生した場合、軸エラーが発生していない軸の動作を設定します。 • 1: 即停止(ImmediateStop)
StopMode_General	停止要因発生時停止選択	レディ ON	INT (MC_DECEL_STOP_MODE) 📖 1451ページ MC_DECEL_STOP_MODE	LIST_WRITE _BACK	軸グループの停止要因が発生した場合の動作を設定します。 • 1: 即停止(ImmediateStop) • 2: 現在の加減速度を続行(KeepCurrentAcc) • 3: 加減速度を代替(AlternativeAcc) 減速度指定のないFB実行時は、即停止します。

変数名	名称	取込	データ型	属性	内容
Unit_Position	位置指令単位	軸グループ有効時	DWORD(HEX)	LIST_WRITE _BACK	モーション制御で使用する位置指令単位を設定します。 • 00000000H: pulse • 00010000H: m • FD010000H: mm • FA010000H: μm • F7010000H: nm • 00410000H: degree • FD410000H: $\times 10^{-3}$degree • FA410000H: $\times 10^{-6}$degree • 00B40000H: Revolution • 00C00000H: inch • 00FF0000H: 任意単位
Unit_PositionString	位置指令単位文字列	軸グループ有効時	WSTRING(31)	LIST_WRITE _BACK	モーション制御で使用する指令単位を文字列で設定します。 位置指令単位(Unit_Position)が「00FF0000H: 任意単位」の場合に設定します。
Unit_Velocity	速度指令単位	軸グループ有効時	DWORD(HEX)	LIST_WRITE _BACK	モーション制御で使用する速度指令単位を設定します。 • 00000300H: U/s • 00004700H: U/min • FD000300H: U/ms • FA000300H: U/μs • F7000300H: U/ns
VelocityBias	速度バイアス値	レディON	LREAL	LIST_WRITE _BACK	軸グループごとの速度バイアス値を設定します。 「0.0000」の場合は、速度バイアスは無効となります。 • 0.0000, 0.0001~2500000000.0
VelocityLimit	速度制限値	レディON	LREAL	LIST_WRITE _BACK	軸グループの速度制限値(制御時の最高速度)を設定します。 • 0.0001~2500000000.0

AxesGroupName.Md.(軸グループモニタデータ)

変数名	名称	取込	データ型	属性	内容
AccelerationLimit	加速度制限値	レディON	LREAL	LIST_READ _ONLY	加速度制限値を格納します。
AccelerationOverride	加速度オーバライド係数	演算周期	LREAL	LIST_READ _ONLY	加速度オーバライド係数を格納します。
AccelerationZeroBehavior	始動時加減速度0指定時動作選択	レディON	INT (MC_ACC_ZERO_MODE) 1452ページ MC_ACC_ZERO_MODE	LIST_READ _ONLY	加速度, 減速度, または加減速時間に「0.0」を設定した場合の動作を表示します。 • -1: エラー (始動しない)(ACCError) • 1: 最大加減速(MaximumAcceleration)
ActualVelocity	フィードバック速度	演算周期 (GroupEnable中)	LREAL	LIST_READ _ONLY	フィードバック速度を格納します。 構成軸のフィードバック速度の合成速度です。構成軸の移動方向に関係なく「0.0」以上の値となります。 軸グループが無効の場合は、「0.0」を格納します。
Analyzing	解析中	演算周期	BOOL	LIST_READ _ONLY	軸グループの位置決め解析中の状態を表示します。 • FALSE: 解析中でない • TRUE: 解析中
AutoDeceleration	自動減速中	演算周期	BOOL	LIST_READ _ONLY	自動減速処理の状態を表示します。 自動減速処理が行われている間, TRUEを格納します。 多重起動を行っている場合は, 最終位置決めポイント実行中に自動減速処理が行われている間, TRUEになります。 制御変更が行われた場合は, FALSEになります。 • FALSE: 自動減速中でない • TRUE: 自動減速中
Axis[1..16]	構成軸	即時	AXIS_REF 1378ページ AxisName.AxisRef.(軸情報)	LIST_READ _ONLY	軸グループを構成する軸情報(AxisName.AxisRef)の軸No.(AxisNo)を格納します。
BufferingFBs	バッファリングFB数	始動時/ 停止時	INT	LIST_READ _ONLY	バッファリングFBの数(0~2)を表示します。

変数名	名称	取込	データ型	属性	内容
CmdInPos	指令インポジション	演算周期	BOOL	LIST_READ_ONLY	目標位置までの合成軸上の残距離が指令インポジション幅(AxesGroupName.Pr.CmdInPos_Width)以下であることを表示します。 • FALSE: 指令インポジション幅以上 • TRUE: 指令インポジション幅以下
CmdInPos_Width	指令インポジション幅	レディON	LREAL	LIST_READ_ONLY	軸グループの合成軸上の指令インポジション幅を表示します。 指令インポジション幅(AxesGroupName.Pr.CmdInPos_Width)の設定値を取り込んだ値となります。 指令インポジション幅(AxesGroupName.Md.CmdInPos_Width)が「0.0」の場合、該当軸グループの指令インポジション機能は無効となります。
CommandedAcceleration	指定加速度	演算周期	LREAL	LIST_READ_ONLY	ユーザが設定した加速度を格納します。(単位: U/s ²) 加減速方式が「1: 加減速時間一定方式(mcFixedTime)」の場合は指定した加減速時間を格納します。(単位: s)
CommandedDeceleration	指定減速度	演算周期	LREAL	LIST_READ_ONLY	ユーザが設定した減速度を格納します。(単位: U/s ²)加減速方式が「1: 加減速時間一定方式(mcFixedTime)」の場合は不正な値です。
CommandedJerk	指定ジャーク	演算周期	LREAL	LIST_READ_ONLY	ユーザが設定したジャークを格納します。(単位: U/s ³)
CommandedVelocity	指定速度	演算周期(GroupEnable中)	LREAL	LIST_READ_ONLY	制御中の動作系のモーション制御FBで取り込んだ指定速度を格納します。 制御が完了すると「0.0」を格納します。 直線補間制御にて、速度モード(VelocityMode)に基準軸速度(ReferenceAxisSpeed)、長軸速度(LongAxisSpeed)を設定した場合は、該当軸の指定速度を格納します。
DecelerationLimit	減速度制限値	レディON	LREAL	LIST_READ_ONLY	減速度制限値を格納します。
Error	軸グループエラー検出	即時	BOOL	LIST_READ_ONLY	軸グループエラー発生の状態を表示します。 • FALSE: 軸グループエラーなし • TRUE: 軸グループエラーあり
ErrorID	軸グループエラーコード	即時	WORD(HEX)	LIST_READ_ONLY	発生した軸グループエラーのエラーコードを表示します。
GroupName	軸グループ名称	即時	WSTRING(127)	LIST_READ_ONLY	軸グループの名称を格納します。
GroupStatus	軸グループ状態	演算周期	INT(MC_AXES_GROUP_STATUS) 1449ページ MC_AXES_GROUP_STATUS	LIST_READ_ONLY	軸グループの状態を表示します。 • -1: 軸グループ変数未初期化/軸グループパラメータ異常(Invalid) • 0: 軸グループ無効(GroupDisabled) • 1: エラー停止中(GroupErrorStop) • 2: 減速停止中(GroupStopping) • 4: 待機中(GroupStandby) • 5: 動作中(GroupMoving)
InterpolationAxes	補間軸	演算周期	DWORD(HEX)	LIST_READ_ONLY	軸グループの構成軸で、多軸位置決め制御に使用している構成軸を表示します。
InVelocity	目標速度到達	演算周期	BOOL	LIST_READ_ONLY	目標速度に到達したか、到達していないかを表示します。 • FALSE: 未到達 • TRUE: 到達
JerkLimit	ジャーク制限値	レディON	LREAL	LIST_READ_ONLY	ジャーク制限値を格納します。
NumberOfAxes	構成軸数	即時	INT	LIST_READ_ONLY	軸グループの構成軸数を表示します。
OverrunOperation	オーバラン時動作設定	レディON	INT(MC_DECEL_STOP_MODE) 1451ページ MC_DECEL_STOP_MODE	LIST_READ_ONLY	オーバラン動作設定の入力状態を表示します。 • 1: 即停止(ImmediateStop)

変数名	名称	取込	データ型	属性	内容
SetAcceleration	指令現在加速度	演算周期	LREAL	LIST_READ_ONLY	指令出力加速度(指令現在速度の差分から算出した加減速度)を格納します。(単位: U/s ²) 正方向(アドレス増加方向)へ加速度が変化する場合に正の値、負方向(アドレス減少方向)へ加速度が変化する場合に負の値となります。 浮動小数点誤差が発生するため、格納する値には誤差を含みます。
SetVelocity	指令現在速度	演算周期 (GroupEnable中)	LREAL	LIST_READ_ONLY	軸グループ動作中、指令出力速度を格納します。 構成軸の移動方向に関係なく「0.0」以上の値となります。 制御が完了すると「0.0」を格納します。 直線補間制御にて、速度モード(VelocityMode)に基準軸速度(ReferenceAxisSpeed)、長軸速度(LongAxisSpeed)を設定した場合は、該当軸の指令現在速度の絶対値を格納します。
StopMode_Deceleration	停止時減速度	レディON	LREAL	LIST_READ_ONLY	停止時減速度の入力状態を表示します。
StopMode_DecelerationCurve	減速停止時停止処理選択	レディON	INT (MC_STOP_CURVE_MODE) 📖 1451ページ MC_STOP_CURVE_MODE	LIST_READ_ONLY	減速停止時停止処理選択の入力状態を表示します。 ・ 1: 減速カーブ再作成(OverrideCurve)
StopMode_ErrorInGroup	軸停止要因発生時構成軸動作選択	レディON	INT (MC_DECEL_STOP_MODE) 📖 1451ページ MC_DECEL_STOP_MODE	LIST_READ_ONLY	軸停止要因発生時構成軸動作選択の入力状態を表示します。 ・ 1: 即停止(ImmediateStop)
StopMode_General	停止要因発生時停止選択	レディON	INT (MC_DECEL_STOP_MODE) 📖 1451ページ MC_DECEL_STOP_MODE	LIST_READ_ONLY	停止要因発生時停止選択の入力状態を表示します。 ・ 1: 即停止(ImmediateStop) ・ 2: 現在の加減速度を続行(KeepCurrentAcc) ・ 3: 加減速度を代替(AlternativeAcc)
TargetAcceleration	目標加速度	演算周期	LREAL	LIST_READ_ONLY	加速中に最大となる加速度を格納します。(単位: U/s ²)
TargetDeceleration	目標減速度	演算周期	LREAL	LIST_READ_ONLY	減速中に最大となる減速度を格納します。(単位: U/s ²)
TargetVelocity	目標速度	演算周期 (GroupEnable中)	LREAL	LIST_READ_ONLY	オーバーライド、速度制限値を考慮した実際の目標速度を格納します。 構成軸の移動方向に関係なく「0.0」以上の値となります。 制御が完了すると「0.0」を格納します。 直線補間制御にて、速度モード(VelocityMode)に基準軸速度(ReferenceAxisSpeed)、長軸速度(LongAxisSpeed)を設定した場合は、該当軸の目標速度を格納します。
Unit_Position	位置指令単位	軸グループ有効時	DWORD(HEX)	LIST_READ_ONLY	モーション制御で使用する位置指令単位を表示します。 ・ 00000000H: pulse ・ 00010000H: m ・ FD010000H: mm ・ FA010000H: μm ・ F7010000H: nm ・ 00410000H: degree ・ FD410000H: ×10 ⁻³ degree ・ FA410000H: ×10 ⁻⁶ degree ・ 00B40000H: Revolution ・ 00C00000H: inch ・ 00FF0000H: 任意単位
Unit_PositionDisplay	位置指令単位表示	軸グループ有効時	WSTRING(31)	LIST_READ_ONLY	現在制御中の位置指令単位を文字列で出力します。
Unit_PositionString	位置指令単位文字列	軸グループ有効時	WSTRING(31)	LIST_READ_ONLY	位置指令単位(AxesGroupName.Pr.Unit_Position)で「00FF0000H: 任意単位」を設定した場合に設定した位置指令単位文字列(AxesGroupName.Pr.Unit_PositionString)の制御中の位置指令単位を文字列で表示します。

変数名	名称	取込	データ型	属性	内容
Unit_Velocity	速度指令単位	軸グループ有効時	DWORD(HEX)	LIST_READ_ONLY	モーション制御で使用する速度指令単位を表示します。 • 00000300H: U/s • 00004700H: U/min • FD000300H: U/ms • FA000300H: U/μs • F7000300H: U/ns
Unit_VelocityDisplay	速度指令単位表示	軸グループ有効時	WSTRING(31)	LIST_READ_ONLY	現在制御中の速度指令単位を文字列で出力します。
VelocityBias	速度バイアス値	レディ ON	LREAL	LIST_READ_ONLY	レディ ON時に取り込んだ速度バイアス値 (AxisGroupName.Pr.VelocityBias) の設定値を格納します。
VelocityLimit	速度制限値	レディ ON	LREAL	LIST_READ_ONLY	軸グループの速度制限値を格納します。
VelocityOverride	速度オーバーライド係数	演算周期	LREAL	LIST_READ_ONLY	速度オーバーライド係数を表示します。
Warning	軸グループ警告検出	即時	BOOL	LIST_READ_ONLY	軸グループ警告の状態を表示します。 • FALSE: 軸グループ警告なし • TRUE: 軸グループ警告あり
WarningID	軸グループ警告コード	即時	WORD(HEX)	LIST_READ_ONLY	発生した軸グループ警告のイベントコードを格納します。

AxesGroupName.Cd.(軸グループ制御データ)

変数名	名称	取込	データ型	属性	内容
AccelerationOverride	加加速度オーバーライド係数	演算周期	LREAL	LIST_READ_ONLY	加加速度オーバーライド係数を設定します。 加減速時間を指定する加減速方式の場合でも、加加速度/減速度に対するオーバーライド係数を設定してください。 • 0.01~10.00
ErrorReset	軸グループエラーリセット	サービス	BOOL	LIST_READ_ONLY	軸グループのエラー、警告をリセットします。 • FALSE: 無実行 • TRUE: エラーリセット実行
VelocityOverride	速度オーバーライド係数	演算周期	LREAL	LIST_READ_ONLY	速度オーバーライド係数を設定します。 速度オーバーライドに0を設定した場合、イベント履歴に"速度オーバーライド「0」"を登録します。 • 0.00~10.00

システム変数

システム変数のデータ型はMT_SYSTEM型で表します。
システムのデータ型は、下記のメンバを所有しています。

メンバ名	データ型	内容	参照
PrConst	SYS_CONST	システムのパラメータデータ(定数)を格納します。 システム変数初期化時に設定値を展開します。 システム変数初期化後に制御への再取込みを実施しません。	1408ページ MotionSystem.PrConst.(モーションシステムパラメータ定数)
Pr	SYS_PRM	システムのパラメータデータを格納します。 システム変数初期化時に初期値を展開します。 パラメータによって制御への取込みタイミングが異なります。	1409ページ MotionSystem.Pr.(モーションシステムパラメータ)
Md	SYS_MONI	システムのモニターデータを格納します。 モニターデータごとに定められた周期でリフレッシュを実施します。	1410ページ MotionSystem.Md.(モーションシステムモニターデータ)
Cd	SYS_CMD	システム用指令データを格納します。モーションサービス処理制御ごとに最新の値を取得し制御に使用します。	1411ページ MotionSystem.Cd.(モーションシステム制御データ)

MotionSystem.PrConst.(モーションシステムパラメータ定数)

変数名	名称	取込	データ型	属性	内容
ExcludeWarning	除外ワーニング	システム 起動時	STRING(255)	LIST_WRITE _BACK	検出しない警告を設定します。 ・"": 未設定 ・"0x1000, 0x1001": 0x1000, 0x1001の警告を検出しない ・"0x1000-0x1010": 0x1000~0x1010の警告を検出しない
OperationCycle[1]	演算周期設定	システム 起動時	CYCLE_PARAM 1425ページ CYCLE_PARAM	LIST_WRITE _BACK	第1演算周期の各設定値を設定します。 演算周期設定(OperationCycle[1])固有の設定、および動作を下記に示します。 ■周期設定(Cycle) ・0: ネットワークの基本周期に同期
OperationCycle[2]	演算周期設定	システム 起動時	CYCLE_PARAM 1425ページ CYCLE_PARAM	LIST_WRITE _BACK	第2演算周期の各設定値を設定します。 演算周期設定(OperationCycle[2])固有の設定、および動作を下記に示します。 ■周期設定(Cycle) ・0: ネットワークの中速周期に同期
OperationCycle[3]	演算周期設定	システム 起動時	CYCLE_PARAM 1425ページ CYCLE_PARAM	LIST_WRITE _BACK	第3演算周期の各設定値を設定します。 演算周期設定(OperationCycle[3])固有の設定、および動作を下記に示します。 ■周期設定(Cycle) ・0: ネットワークの低速周期に同期
ProfileRootDrive	演算プロファイル ルートドライブ	システム 起動時	WSTRING(7)	LIST_WRITE _BACK	演算プロファイルの保存先ドライブを設定します。 ■設定範囲 ・データメモリ(ドライブ4): /rom ・SDメモリカード(ドライブ2): /sd
SequenceReadyInterlock	シーケンサレディ連 動選択	システム 起動時	BOOL	LIST_WRITE _BACK	"シーケンサレディ (MotionSystem.Cd.SequenceReady)"をONにする方式を選択します。 (1411ページ MotionSystem.Cd.(モーションシステム制御データ)) ・TRUE: CPU機能部と連動する。 ・FALSE: CPU機能部と連動しない。

MotionSystem.Pr.(モーションシステムパラメータ)

変数名	名称	取込	データ型	属性	内容
ForcedStop_Signal	全軸緊急停止信号	レディ ON	SIGNAL_SELECT 1424ページ SIGNAL_SELECT(信号選択)	LIST_WRITE _BACK	全軸緊急停止を使用する信号を設定します。 外部信号として使用している、コントローラ管理のデバイス局が切断された場合、全軸を停止します。 全軸緊急停止信号(ForcedStop_Signal)固有の設定、および動作を下記に示します。 ■入出力番号(StartIO) 入力値を無視します。 ■対象(Target) 設定がないと、信号無効と判断して常に信号未検出状態にします。 データ種別には、[VAR]、[DEV]、[CONST]のみ設定できます。 使用できないデータ種別を設定した場合、パラメータ範囲外(システム)(エラーコード: 1A62H)となります。 ■信号検出方法(Detection) 下記レベル検出のみ設定を許可します。 0: TRUE時検出(HighLevel) 1: FALSE時検出(LowLevel) エッジ検出を設定した場合、パラメータ範囲外(システム)(エラーコード: 1A62H)となります。 ■補正時間(CompensationTime) 入力値を無視します。 ■フィルタ時間(FilterTime) 設定範囲は「0.0~5.0[s]」です。 範囲外の値を設定した場合、システム信号のフィルタ時間設定範囲外警告(イベントコード: 0F0FH)となり、フィルタ時間は「0.0」で動作します。
StopMode_All	全軸停止要因発生時 停止選択	レディ ON	INT (MC_DECEL_STOP_MODE) 1451ページ MC_DECEL_STOP_MODE	LIST_WRITE _BACK	全軸の停止要因が発生した場合に、即停止するか、減速停止するかを設定します。 1: 即停止(ImmediateStop) 2: 現在の加減速度を続行(KeepCurrentAcc) 3: 加減速度を代替(AlternativeAcc) 減速度指定のないFB実行時は、即停止します。
StopMode_AllDeceleration	全軸停止時減速度	レディ ON	LREAL	LIST_WRITE _BACK	全軸の停止要因が発生した場合の減速停止時の減速度を指定します。 0.0000が設定されている場合は即停止します。 加減速方式により設定範囲は異なります。 - 加減速度指定方式: 0.0000, 0.0001~2147483647.0 - 加減速時間一定方式: 0.0000, 0.000001~8400.0

MotionSystem.Md.(モーションシステムモニタデータ)

変数名	名称	取込	データ型	属性	内容
Error	モーション部システムエラー検出	即時	BOOL	LIST_READ_ONLY	エラーの状態を表示します。 ・ FALSE: エラーなし ・ TRUE: エラーあり
ErrorID	モーション部最新システムエラーコード	即時	WORD(HEX)	LIST_READ_ONLY	最新のエラーコードを表示します。
ForcedStop_Released	緊急停止解除中	演算周期	BOOL	LIST_READ_ONLY	緊急停止解除状態を表示します。 ・ FALSE: 全軸緊急停止 ・ TRUE: 全軸緊急停止解除
ForcedStop_Signal	全軸緊急停止信号	レディ ON	SIGNAL_SELECT 1424ページ SIGNAL_SELECT(信号選択)	LIST_READ_ONLY	全軸緊急停止信号の入力状態を表示します。 全軸緊急停止信号(ForcedStop_Signal)固有の出力を下記に示します。 ■入出力番号(StartIO) 常に「0」を表示します。 ■対象(Target) パラメータの取込み結果を表示します。 ■信号検出方法(Detection) パラメータの取込み結果を表示します。 ■補正時間(CompensationTime) 常に「0.0」を表示します。 ■フィルタ時間(FilterTime) パラメータの取込み結果を表示します。
MotionService	モーションサービス処理モニタ	サービス	CYCLE_MONI 1425ページ CYCLE_MONI	LIST_READ_ONLY	モーションサービス処理のモニタ値を格納します。
OperationCycle[1..3]	演算周期モニタ	即時	CYCLE_MONI 1425ページ CYCLE_MONI	LIST_READ_ONLY	第1演算周期と、第2演算周期、第3演算周期のモニタ値を格納します。
Ready	準備完了	サービス	BOOL	LIST_READ_ONLY	内蔵モーション準備完了(X20)のON/OFF状態を表示します。 ・ FALSE: OFF(準備未完了) ・ TRUE: ON(準備完了)
StopMode_All	全軸停止要因発生時停止選択	レディ ON	INT (MC_DECEL_STOP_MODE) 1451ページ MC_DECEL_STOP_MODE	LIST_READ_ONLY	全軸停止要因発生時停止選択の入力状態を表示します。 ・ 1: 即停止(ImmediateStop) ・ 2: 現在の加減速度を続行(KeepCurrentAcc) ・ 3: 加減速度を代替(AlternativeAcc)
StopMode_AllDeceleration	全軸停止時減速度	レディ ON	LREAL	LIST_READ_ONLY	全軸停止時減速度の入力状態を表示します。
Sync	同期用フラグ	サービス	BOOL	LIST_READ_ONLY	同期用フラグ(X21)のON/OFF状態を表示します。 ・ FALSE: OFF(ユニットアクセス不可) ・ TRUE: ON(ユニットアクセス許可)
Warning	モーション部システム警告検出	即時	BOOL	LIST_READ_ONLY	警告の状態を表示します。 ・ FALSE: 警告なし ・ TRUE: 警告あり
WarningID	モーション部最新システム警告コード	即時	WORD(HEX)	LIST_READ_ONLY	最新のイベントコードを格納します。

MotionSystem.Cd.(モーションシステム制御データ)

変数名	名称	取込	データ型	属性	内容
ErrorReset	システムエラーリセット	サービス	BOOL	LIST_READ_ONLY	すべてのエラー、警告をリセットします。 • FALSE: 無実行 • TRUE: システムエラーリセット実行
SequenceReady	シーケンサレディ	サービス	BOOL	LIST_READ_ONLY	CPU機能部が正常であることを確認し、モーション機能部に知らせる信号状態を表示します。 • FALSE: シーケンサレディ OFF • TRUE: シーケンサレディ ON ■“シーケンサレディ連動選択 (MotionSystem.PrConst.SequenceReadyInterlock)”が TRUEの場合 下記の条件がすべて成立した時、本変数はTRUEになります。 • 同期用フラグ(X21)がOFF→ON • CPU機能部がSTOP→PAUSE/RUN ■“シーケンサレディ連動選択 (MotionSystem.PrConst.SequenceReadyInterlock)”が FALSEの場合 下記の条件がすべて成立した時、本変数はTRUEになります。 • 同期用フラグ(X21)がOFF→ON • CPU機能部がSTOP→PAUSE/RUN • シーケンサレディ (Y20)がOFF→ON

入力軸変数

アドバンスト同期制御で使用する入力軸変数のデータ型はADV_INPUT型で表します。
各入力軸のデータ型は、下記のメンバを所有しています。

メンバ名	データ型	内容	参照
LabelID	WORD(UINT)	入力軸のラベルIDを設定します。	📖 1412ページ AdvInputName.(入力軸設定)
Axis	AXIS_REF	アドバンスト同期制御の入力軸とする軸を設定します。 複数の入力軸情報に同一の軸を指定することはできません。	
PrConst	ADV_INPUT_PRM_CONST	入力軸とする軸のパラメータデータ(定数)を格納します。 入力軸変数初期化時に設定値を展開します。 入力軸変数初期化後に制御への再取込みを実施しません。	📖 1413ページ AdvInputName.PrConst.(入力軸パラメータ定数)
Pr	ADV_INPUT_PRM	入力軸とする軸のパラメータデータを格納します。 入力軸変数初期化時に初期値を展開します。 入力軸変数初期化後も制御への再取込みを実施します。 パラメータによって制御への取込みタイミングが異なります。	📖 1413ページ AdvInputName.Pr.(入力軸パラメータ)
Md	ADV_INPUT_MONI	入力軸とする軸のモニタデータを格納します。 モニタデータごとに定められた周期でリフレッシュを実施します。	📖 1413ページ AdvInputName.Md.(入力軸モニタデータ)

AdvInputName.(入力軸設定)

変数名	名称	取込	データ型	属性	内容
LabelID	軸ラベルID	システム 起動時	WORD(UINT)	LIST_WRITE _BACK	入力軸のラベルIDを設定します。 ・ 1~256
Axis	軸情報	システム 起動時	AXIS_REF 📖 1378ページ AxisName.AxisRef.(軸情報)	LIST_WRITE _BACK	アドバンスト同期制御の入力軸とする軸情報 (AxisName.AxisRef)を設定します。 生成可能な入力軸は、最大256個です。異なる入力軸の軸情報に同一の軸を設定できません。 ローカルラベルで定義した軸の軸情報(AxisName.AxisRef)は、設定しないでください。設定した場合入力軸は生成されません。 出力軸に設定可能な軸種別は、実ドライブ軸、仮想ドライブ軸、仮想連結軸です。使用できない軸を設定した場合、アドバンスト同期制御入力軸不正(エラーコード: 1A0EH)となります。

AdvInputName.PrConst.(入力軸パラメータ定数)

変数名	名称	取込	データ型	属性	内容
SourceValue	データソース選択	システム 起動時	INT (MC_SOURCE) 1448ページ MC_SOURCE	LIST_WRITE _BACK	アドバンス同期制御で参照する入力軸のデータソースを設定します。 1: 指令現在値(mcSetValue) 2: フィードバック値(mcActualValue) 101: 最新指令現在値(mcLatestSetValue) 102: 最新フィードバック値(mcLatestActualValue)
SmoothingTimeConstant	スムージング時定数	システム 起動時	WORD(UINT)	LIST_WRITE _BACK	入力軸からの入力移動量をスムージング処理するときの平均化時間を設定します。 0~5000[ms]
PhaseCompensationTimeConstant	位相補正時定数	システム 起動時	WORD(UINT)	LIST_WRITE _BACK	位相補正時の位相補正量を一次遅れで反映するときの時定数を設定します。 設定した時定数で位相補正量の63%を反映します。 0~65535[ms]
DirectionRestriction	移動方向制限	システム 起動時	INT (MC_INPUT_DIRECTION) 1454ページ MC_INPUT_DIRECTION	LIST_WRITE _BACK	入力軸からの入力移動量を一方方向に制限するときを設定します。 0: 移動方向制限なし(NoDirectionRestriction) 1: 現在位置が増加方向のみ許可(mcPositiveDirection) 2: 現在位置が減少方向のみ許可(mcNegativeDirection)

AdvInputName.Pr.(入力軸パラメータ)

変数名	名称	取込	データ型	属性	内容
PhaseCompensationAdvanceTime	位相補正進め時間	演算周期	DINT	LIST_WRITE _BACK	入力軸の位相(入力応答)を進めたり遅らせたりするときを設定します。 -100000000~100000000[μs]

AdvInputName.Md.(入力軸モニタデータ)

変数名	名称	取込	データ型	属性	内容
ActiveStatus	有効状態	即時	INT	LIST_READ _ONLY	入力軸の状態が格納されます。 -1: 無効 0: 未接続 1: 入力軸有効
CumulativePosition	累積現在位置	演算周期	LREAL	LIST_READ _ONLY	入力軸の累積現在位置を入力軸の軸情報(AdvInputName.Axis)に設定した軸の位置単位が格納されます。 入力軸の現在位置は、スムージング処理、位相補正処理、移動方向制限処理後の値です。
SetVelocity	指令現在速度	演算周期	LREAL	LIST_READ _ONLY	入力軸の速度を入力軸の軸情報(AdvInputName.Axis)に設定した軸の速度単位が格納されます。 入力軸の速度は、スムージング処理、位相補正処理、移動方向制限処理後の値です。
PhaseCompensationAmount	位相補正量	演算周期	LREAL	LIST_READ _ONLY	入力軸の位相補正量を入力軸の軸情報(AdvInputName.Axis)に設定した軸の位置指令単位が格納されます。 入力軸の位相補正量は、スムージング処理、位相補正処理後の値です。
DirectionRestrictionAmount	移動方向制限量	演算周期	LREAL	LIST_READ _ONLY	入力軸の移動方向制限時、許可方向と逆の入力移動量の累積値を入力軸に設定した軸の位置単位が格納されます。

出力軸変数

アドバンスト同期制御で使用する出力軸変数のデータ型はADV_OUTPUT型で表します。
各出力軸のデータ型は、下記のメンバを所有しています。

メンバ名	データ型	内容	参照
LabelID	WORD(UINT)	出力軸のラベルIDを設定します。	📖 1414ページ
AxisRef	AXIS_REF	アドバンスト同期制御の出力軸とする軸を設定します。 複数の出力軸情報に同一の軸を指定することはできません。	AdvOutputName.(出力軸設定)
PrConst	ADV_OUTPUT_PRM_CONST	出力軸とする軸のパラメータデータ(定数)を格納します。 出力軸変数初期化時に設定値を展開します。 出力軸変数初期化後に制御への再取込みを実施しません。	📖 1415ページ AdvOutputName.PrConst.(出力軸パラメータ定数)
Pr	ADV_OUTPUT_PRM	出力軸とする軸のパラメータデータを格納します。 出力軸変数初期化時に初期値を展開します。 出力軸変数初期化後も制御への再取込みを実施します。 パラメータによって制御への取込みタイミングが異なります。	📖 1416ページ AdvOutputName.Pr.(出力軸パラメータ)
Md	ADV_OUTPUT_MONI	出力軸とする軸のモニタデータを格納します。 モニタデータごとに定められた周期でリフレッシュを実施します。	📖 1418ページ AdvOutputName.Md.(出力軸モニタデータ)
Cd	ADV_OUTPUT_CMD	出力軸制御用指令データを格納します。 制御演算周期ごとに最新の値を取得し制御に使用します。	📖 1418ページ AdvOutputName.Cd.(出力軸制御データ)

AdvOutputName.(出力軸設定)

変数名	名称	取込	データ型	属性	内容
LabelID	軸ラベルID	システム 起動時	WORD(UINT)	LIST_WRITE _BACK	出力軸のラベルIDを設定します。 ・ 1~256
Axis	軸情報	システム 起動時	AXIS_REF 📖 1378ページ AxisName.AxisRef.(軸情報)	LIST_WRITE _BACK	アドバンスト同期制御の出力軸とする軸情報 (AxisName.AxisRef)を設定します。 生成可能な出力軸は、最大256個です。異なる出力軸の軸 情報に同一の軸を設定できません。設定した場合、アド バンスト同期制御出力軸不正(エラーコード: 1A0FH)とな ります。 ローカルラベルで定義した軸の軸情報(AxisName.AxisRef) は、設定しないでください。設定しても出力軸は生成さ れません。

AdvOutputName.PrConst.(出力軸パラメータ定数)

変数名	名称	取込	データ型	属性	内容
SmoothingTimeConstant	出力軸スムージング時定数	システム起動時	WORD(UINT)	LIST_WRITE _BACK	出力軸からの出力移動量をスムージング処理するときの平均化時間[ms]を設定します。 • 0~5000[ms]
MasterClutchSmoothingTimeConstant	主軸クラッチスムージング時定数	システム起動時	WORD(UINT)	LIST_WRITE _BACK	主軸クラッチスムージング方式 (AdvOutputName.Pr.Clutch.MasterSmoothingMethod) で「1: 時定数方式(指数)(TimeConstantExponent)」, または「2: 時定数方式(直線)(TimeConstantLinear)」を設定した場合に, 時定数を設定します。 クラッチON/OFF共通の時定数を設定します。 • 0~5000[ms]
AuxClutchSmoothingTimeConstant	補助軸クラッチスムージング時定数	システム起動時	WORD(UINT)	LIST_WRITE _BACK	補助軸クラッチスムージング方式 (AdvOutputName.Pr.Clutch.AuxSmoothingMethod) で「1: 時定数方式(指数)(TimeConstantExponent)」, または「2: 時定数方式(直線)(TimeConstantLinear)」を設定した場合に, 時定数を設定します。 クラッチON/OFF共通の時定数を設定します。 • 0~5000[ms]
MasterSpeedChangeGearSmoothingTimeConstant	主軸変速機スムージング時定数	システム起動時	WORD(UINT)	LIST_WRITE _BACK	主軸変速処理時の速度変化をスムージング処理するときの平均化時間を設定します。なお, スムージング処理により設定時間分だけ入力値の伝達が遅れます。 設定値が「0」の場合, ダイレクトに速度が変化します。 • 0~5000[ms]
AuxSpeedChangeGearSmoothingTimeConstant	補助軸変速機スムージング時定数	システム起動時	WORD(UINT)	LIST_WRITE _BACK	補助軸変速処理時の速度変化をスムージング処理するときの平均化時間を設定します。なお, スムージング処理により設定時間分だけ入力値の伝達が遅れます。 設定値が「0」の場合, ダイレクトに速度が変化します。 • 0~5000[ms]
OutSpeedChangeGearSmoothingTimeConstant	出力軸変速機スムージング時定数	システム起動時	WORD(UINT)	LIST_WRITE _BACK	出力軸変速処理時の速度変化をスムージング処理するときの平均化時間を設定します。なお, スムージング処理により設定時間分だけ入力値の伝達が遅れます。 設定値が「0」の場合, ダイレクトに速度が変化します。 • 0~5000[ms]

AdvOutputName.Pr.(出力軸パラメータ)

変数名	名称	取込	データ型	属性	内容
SubAxis	サブ入力軸対象	始動時	AXIS_REF 1378ページ AxisName.AxisRef.(軸情報)	LIST_WRITE _BACK	サブ入力軸の対象軸を設定します。 「0」を設定した場合は無効となります。 入力設定に割付けていない軸を設定した場合は、アドバ ンスト同期制御軸未設定(エラーコード: 1B22H)となりま す。
AuxAxis	補助軸対象	始動時	AXIS_REF 1378ページ AxisName.AxisRef.(軸情報)	LIST_WRITE _BACK	補助軸の対象軸を設定します。 「0」を設定した場合は無効となります。 入力設定に割付けていない軸を設定した場合は、アドバ ンスト同期制御軸未設定(エラーコード: 1B22H)となりま す。
Gear	ギアパラメータ	始動時	ADV_GEAR_PARAM 1437ページ ADV_GEAR_PARAM (ギアパラメータ)	LIST_WRITE _BACK	ギア機能のパラメータを設定します。
MasterOnClutchSignal	主軸ONクラッチ信号 設定	始動時	SIGNAL_SELECT 1424ページ SIGNAL_SELECT(信 号選択)	LIST_WRITE _BACK	主軸ONクラッチ制御設定 (AdvOutputName.Pr.Clutch.MasterOnControl)が「15: 入 出力データ指定(ClutchSignal)」の場合にクラッチの外部信 号を設定します。 主軸ONクラッチ信号設定(MasterOnClutchSignal)固有の 設定、および動作を下記に示します。 ■対象(Target) データ種別には、[OBJ], [VAR], [DEV], [LINK]のみ設定可 能です。 ■信号検出方法(Detection) 下記レベル検出のみ設定を許可します。 ・2: FALSE→TRUE(立上り)時検出(RisingEdge) ・3: TRUE→FALSE(立下り)時検出(FallingEdge) ・4: 立上り/立下り時検出(BothEdges) ■補正時間(CompensationTime) 設定範囲は「-5.0~5.0[s]」です。 ■フィルタ時間(FilterTime) 設定範囲は「0.0」です。
MasterOffClutchSignal	主軸OFFクラッチ信号 設定	始動時	SIGNAL_SELECT 1424ページ SIGNAL_SELECT(信 号選択)	LIST_WRITE _BACK	主軸OFFクラッチ制御設定 (AdvOutputName.Pr.Clutch.MasterOffControl)が「15: 入 出力データ指定(ClutchSignal)」の場合にクラッチの外部 信号を設定します。 主軸OFFクラッチ信号設定(MasterOffClutchSignal)固有の 設定、および動作を下記に示します。 ■対象(Target) データ種別には、[OBJ], [VAR], [DEV], [LINK]のみ設定可 能です。 ■信号検出方法(Detection) 下記レベル検出のみ設定を許可します。 ・2: FALSE→TRUE(立上り)時検出(RisingEdge) ・3: TRUE→FALSE(立下り)時検出(FallingEdge) ・4: 立上り/立下り時検出(BothEdges) ■補正時間(CompensationTime) 設定範囲は「-5.0~5.0[s]」です。 ■フィルタ時間(FilterTime) 設定範囲は「0.0」です。

変数名	名称	取込	データ型	属性	内容
AuxOnClutchSignal	補助軸ONクラッチ信号設定	始動時	SIGNAL_SELECT 1424ページ SIGNAL_SELECT(信号選択)	LIST_WRITE _BACK	補助軸ONクラッチ制御設定 (AdvOutputName.Pr.Clutch.AuxOnControl)が「15: 入出力データ指定(ClutchSignal)」の場合にクラッチの外部信号を設定します。 補助軸ONクラッチ信号設定(AuxOnClutchSignal)固有の設定、および動作を下記に示します。 ■対象(Target) データ種別には、[OBJ], [VAR], [DEV], [LINK]のみ設定可能です。 ■信号検出方法(Detection) 下記レベル検出のみ設定を許可します。 2: FALSE→TRUE(立上り)時検出(RisingEdge) 3: TRUE→FALSE(立下り)時検出(FallingEdge) 4: 立上り/立下り時検出(BothEdges) ■補正時間(CompensationTime) 設定範囲は「-5.0~5.0[s]」です。 ■フィルタ時間(FilterTime) 設定範囲は「0.0」です。
AuxOffClutchSignal	補助軸OFFクラッチ信号設定	始動時	SIGNAL_SELECT 1424ページ SIGNAL_SELECT(信号選択)	LIST_WRITE _BACK	補助軸OFFクラッチ制御設定 (AdvOutputName.Pr.Clutch.AuxOffControl)が「15: 入出力データ指定(ClutchSignal)」の場合にクラッチの外部信号を設定します。 補助軸OFFクラッチ信号設定(AuxOffClutchSignal)固有の設定、および動作を下記に示します。 ■対象(Target) データ種別には、[OBJ], [VAR], [DEV], [LINK]のみ設定可能です。 ■信号検出方法(Detection) 下記レベル検出のみ設定を許可します。 2: FALSE→TRUE(立上り)時検出(RisingEdge) 3: TRUE→FALSE(立下り)時検出(FallingEdge) 4: 立上り/立下り時検出(BothEdges) ■補正時間(CompensationTime) 設定範囲は「-5.0~5.0[s]」です。 ■フィルタ時間(FilterTime) 設定範囲は「0.0」です。
Clutch	クラッチパラメータ	—	ADV_CLUTCH_PARAM 1437ページ ADV_CLUTCH_PARAM(クラッチパラメータ)	LIST_WRITE _BACK	クラッチ機能のパラメータを設定します。
SpeedChangeGear	変速機パラメータ	—	ADV_SPEEDCHANGE_GEAR_PARAM 1443ページ ADV_SPEEDCHANGE_GEAR_PARAM(変速機パラメータ)	LIST_WRITE _BACK	変速機機能のパラメータを設定します。
Cam	カムパラメータ	—	ADV_CAM_PARAM 1444ページ ADV_CAM_PARAM(カムパラメータ)	LIST_WRITE _BACK	カム機能のパラメータを設定します。
Restore	同期制御初期位置パラメータ	—	ADV_RESTORE_PARAM 1446ページ ADV_RESTORE_PARAM(同期制御初期位置パラメータ)	LIST_WRITE _BACK	同期制御初期位置のパラメータを設定します。

AdvOutputName.Md.(出力軸モニタデータ)

変数名	名称	取込	データ型	属性	内容
SyncStatus	同期状態	即時	INT	LIST_READ_ONLY	出力軸の同期状態が格納されます。 • -1: 無効 • 0: 未同期 • 1: 同期中
MasterAxisNo	主軸	始動時	WORD(UINT)	LIST_READ_ONLY	出力軸と同期中の入力軸の軸情報(AdvInputName.Axis)に設定した主軸の軸番号が格納されます。接続していない場合は、「0」が格納されます。
SubAxisNo	サブ入力軸	始動時	WORD(UINT)	LIST_READ_ONLY	出力軸と同期中の入力軸の軸情報(AdvInputName.Axis)に設定したサブ入力軸の軸番号が格納されます。接続していない場合は、「0」が格納されます。
AuxAxisNo	補助軸	始動時	WORD(UINT)	LIST_READ_ONLY	出力軸と同期中の入力軸の軸情報(AdvInputName.Axis)に設定した補助軸の軸番号が格納されます。接続していない場合は、「0」が格納されます。
Clutch	クラッチモニタ	—	ADV_CLUTCH_MONI 📖 1441ページ ADV_CLUTCH_MONI(クラッチモニタ)	LIST_READ_ONLY	クラッチ機能のモニタデータが格納されます。
Cam	カムモニタ	—	ADV_CAM_MONI 📖 1445ページ ADV_CAM_MONI(カムモニタ)	LIST_READ_ONLY	カム機能のモニタデータが格納されます。

AdvOutputName.Cd.(出力軸制御データ)

変数名	名称	取込	データ型	属性	内容
Clutch	クラッチ制御データ	—	ADV_CLUTCH_CMD 📖 1441ページ ADV_CLUTCH_CMD(クラッチ制御データ)	LIST_READ_ONLY	クラッチ機能の制御データを設定します。
SpeedChangeGear	変速機制御データ	—	ADV_SPEEDCHANGE_GEAR_CMD 📖 1443ページ ADV_SPEEDCHANGE_GEAR_CMD(変速機制御データ)	LIST_WRITE_BACK	変速機機能の制御データを設定します。
Cam	カム制御データ	—	ADV_CAM_CMD 📖 1446ページ ADV_CAM_CMD(カム制御データ)	LIST_READ_ONLY	カム機能の制御データを設定します。

システム変数(アドバンスト同期制御)

システム変数のデータ型はSYSTEM型で表します。

システムのデータ型は、下記のメンバを所有しています。

メンバ名	データ型	内容	参照
PrConst	SYS_CONST	システムのパラメータデータ(定数)を格納します。 システム変数初期化時に設定値を展開します。 システム変数初期化後に制御への再取込みを実施しません。	1419ページ AddonSystem.PrConst.(アドオンシステムパラメータ定数)
Md	SYS_MONI	システムのモニタデータを格納します。 モニタデータごとに定められた周期でリフレッシュを実施します。	1419ページ AddonSystem.Md.(アドオンシステムモニタデータ)

AddonSystem.PrConst.(アドオンシステムパラメータ定数)

変数名	名称	取込	データ型	属性	内容
MT_MotionControl_AdvancedSync	アドオンMotionControl_AdvancedSyncパラメータ	システム起動時	ADDON_PARAM 1425ページ ADDON_PARAM	LIST_WRITE _BACK	アドオンMotionControl_AdvancedSync/パラメータで使用しているメモリの最大使用量を設定します。

AddonSystem.Md.(アドオンシステムモニタデータ)

変数名	名称	取込	データ型	属性	内容
MT_MotionControl_AdvancedSync	アドオンMotionControl_AdvancedSyncモニタ	サービス	ADDON_MONI 1425ページ ADDON_MONI	LIST_READ _ONLY	アドオンMotionControl_AdvancedSync/パラメータで使用しているメモリの空き容量を格納します。

その他構造体の変数

MC_CAM_REF

変数名	名称	データ型	内容
ProfileData	プロファイル	PROFILE_DATA 1424ページ PROFILE_DATA	カムデータのプロファイルを設定します。
CamID	カムID	MC_CAM_ID 1419ページ MC_CAM_ID	カムデータのプロファイルIDを設定します。

MC_POSITIONING_DATA_REF

変数名	名称	データ型	内容
ProfileData	プロファイル	PROFILE_DATA 1424ページ PROFILE_DATA	位置決めデータのプロファイルを設定します。
PositioningData	位置決めデータ	MC_POSITIONING_DATA_ID 1419ページ MC_POSITIONING_DATA_ID	位置決めデータのID番号を設定します。

MC_CAM_ID

変数名	名称	データ型	内容
ProfileID	プロファイルID	PROFILE_ID 1424ページ PROFILE_ID	カムデータのプロファイルIDを設定します。

MC_POSITIONING_DATA_ID

変数名	名称	データ型	内容
ProfileData	プロファイルID	PROFILE_ID 1424ページ PROFILE_ID	位置決めデータの演算プロファイルIDを設定します。

MC_TRIGGER_REF

変数名	名称	データ型	内容
Signal	トリガ信号	SIGNAL_SELECT 1424ページ SIGNAL_SELECT(信号選択)	トリガ信号を設定します。

MC_INPUT_REF

変数名	名称	データ型	内容
Signal	入力信号	SIGNAL_SELECT 1424ページ SIGNAL_SELECT(信号選択)	入力する信号を設定します。

MC_OUTPUT_REF

変数名	名称	データ型	内容
Signal	出力信号	SIGNAL_SELECT 1424ページ SIGNAL_SELECT(信号選択)	出力する信号を設定します。

MC_CAMSWITCH_REF

変数名	名称	データ型	内容
ProfileData	プロファイル	PROFILE_DATA 1424ページ PROFILE_DATA	デジタルカムスイッチのプロファイルを設定します。
CamSwitchData	カムスイッチデータ	MC_CAMSWITCH_DATA_REF 1420ページ MC_CAMSWITCH_DATA_REF	デジタルカムスイッチのID番号を設定します。

MC_TRACK_REF

変数名	名称	データ型	内容
OnCompensation	ON時間補正	LREAL	出力信号がONになるタイミングを補正します。
OffCompensation	OFF時間補正	LREAL	出力信号がOFFになるタイミングを補正します。
Hysteresis	スイッチング禁止距離	LREAL	データソースがスイッチングを禁止する距離を設定します。 データソースがスイッチングポイント付近で増減している場合に、出力信号のチャタリングを防止します

MC_CAMSWITCH_DATA_REF

変数名	名称	データ型	内容
ProfileData	プロファイルID	PROFILE_ID 1424ページ PROFILE_ID	デジタルカムスイッチの演算プロファイルIDを指定します。

MC_CAMSWITCH_SOURCE_REF

変数名	名称	データ型	内容
Source	データソース	TARGET_REF 1421ページ TARGET_REF(入力信号)	データソースを指定します。
RingCount_LowerValue	データソースリングカウンタ下限值	LREAL	データソースのリングカウンタ下限值を設定します。
RingCount_UpperValue	データソースリングカウンタ上限値	LREAL	データソースのリングカウンタ上限値を設定します。
SpeedAverageNum	速度平均回数	WORD(UINT)	スイッチのON/OFFタイミングを算出するために、いくつ前の演算周期までの速度平均をとるか設定します。

TARGET_REF(入力信号)

変数名	名称	データ型	内容
StartIO	IO番号	WORD(HEX)	IO番号を設定します。
Target	対象	WSTRING(63)	制御に使用する信号を、下記の文字列書式で設定します。 大文字、小文字は区別しません。スペースは無視します。 ■[種別](型)データ名.ビット位置@対象修飾 ・[種別]^{*1} ・(型)^{*2} ・データ名^{*1} ・.ビット位置^{*3} ・@対象修飾^{*4} ■[種別](型)WSTRING型ラベル [^] ・[種別]^{*1} ・(型)^{*2} ・WSTRING型ラベル^{^*8}

*1 「[種別]」には、データ種別、「データ名」には、対象データ設定します。設定する種別により、データ名の設定方法が異なります。

種別	対象	データ名		設定例
		書式	内容	
[OBJ]	デバイス機器のCANopenオブジェクト(サイクリックデータにマッピング済みのもの)	[OBJ]0x□□□□**##	オブジェクトのインデックス、サブインデックス、サイズを設定します。 ・□□□□: インデックス ・**: サブインデックス ・##: サイズ(ビット数) *: (型)の設定をオブジェクトのサイズに合わせてください。 *: 一部のオブジェクトは外部信号高精度入力として使用できます。 *: サイズが「1バイト」のオブジェクトは使用できません。	[OBJ]0x607A0020 *: MR-J5(W)-GのTarget position(Obj.607AH: 00H) 4バイトオブジェクト(20H)の場合
[VAR]	モーションシステムのラベル	[VAR]ラベル名	モーションシステム内のラベルを設定します。 *: ローカルラベルを設定する場合、「@対象修飾」にPOU名を設定します。	[VAR]ADUnit10.OutputEnable
[AXIS]	軸データ	[AXIS]MC_SOURCE	MC_SOURCE列挙型の列挙子を設定します。 データの(型)には、「LREAL」のみ設定できます。 ・1: 指令現在値(mcSetValue) ・2: フィードバック値(mcActualValue) ・101: 最新指令現在値(mcLatestSetValue) ・102: 最新フィードバック値(mcLatestActualValue) *: 本データに対する書込みはできません。	[AXIS]mcSetValue
[DEV]	モーションシステム内のデバイス(自ユニットリンクデバイスを含む)	[DEV]デバイス名	自ユニット内のデバイスを設定します。設定できるデバイスについては、下記を参照してください。  1423ページ [DEV]種別の対応デバイス一覧	[DEV]X40 [DEV]U0\G11500000.1
[CONST]	定数	[CONST]定数 [CONST]0x定数	浮動小数(E形式も使用可能)、10進整数、16進整数で設定します。 *: 本データに対する書込みはできません。	[CONST]1000 [CONST]0x100

*2 データの型を明示的に設定する場合に記述します。設定可能なデータ型を下記に示します。

種別で設定したデータ自身の型がBOOL型以外のデータに対し「(BOOL)」を設定した場合、指定するビットNo.に0を指定したものとして取り扱います。

型
(BOOL)
(INT)
(DINT)
(WORD)
(DWORD)
(REAL)
(LREAL)

*3 [種別]が下記のデータに対して、ビット位置を付加すると、BOOL型のデータとして扱います。
(型)とビット位置を同時に設定した場合は、(型)の設定を無視し、BOOL型のデータとなります。

種別	ビット位置設定範囲
[OBJ]	0~F
[VAR](WORD型, DWORD型, INT型, DINT型のみ)	
[DEV]	

*4 データを特定するための補助情報を設定します。設定する[種別]により、内容が異なります。
信号を使用する機能によっては、必ず設定する必要があります。対象修飾を必要としないデータ種別や機能で対象修飾を設定した場合は無視します。

種別	対象修飾		設定例
	書式	内容	
[OBJ]	@局アドレス	局アドレス(デバイス機器を識別するためのIPアドレス)を設定します。 多軸ドライブユニットのように、1つの局に複数の論理軸が含まれる場合、論理軸を識別するためのマルチドロップ番号を設定します。 @192.168.3.10#2 └─ マルチドロップ番号 └─ IPアドレス *: マルチドロップ番号は、「#+番号(10進数)」で設定します。 • #0: A軸 • #1: B軸 • #2: C軸 *: マルチドロップ番号は省略できます。省略した場合、「#0」とみなします。	@192.168.3.10 @192.168.3.10#2
[VAR]	@POU名	ローカルラベルを設定する場合にPOU名を設定します。 グローバルラベルとローカルラベルに同一ラベル名を使用する設定に関わらず、@POUがない場合は、グローバルラベルとして扱います。	
[AXIS]	@Position@CumulativePos	対象データ(位置)を設定します。 「データ名」に設定したMC_SOURCE型の列挙子により@Position, または@CumulativePosを設定します。*5	@Position @CumulativePos
[DEV]	—	文字修飾は設定できません。設定しても無視します。	
[CONST]	@+加算値 @-減算値	値を参照するたびに、読み出し値を変更するときに使用します。 対象修飾を設定すると周期ごとに値の変更ができます。 データ型がREAL, またはLREALの場合、本設定は無効となります。	[CONST](INT)0@-10*6 [CONST](BOOL)0x00.3@+1*7

*5 設定可能な列挙子を下記に示します。

○: 設定可能, ×: 設定不可

MC_SOURCE型の列挙子	内容	修飾	
		@Position	@CumulativePos
1: 指令現在値(mcSetValue)	前回の演算周期の指令値	○	○
2: フィードバック値(mcActualValue)	前回の演算周期のフィードバック値	○	×
101: 最新指令現在値(mcLatestSetValue)	今回の演算周期の指令値	○	○
102: 最新フィードバック値(mcLatestActualValue)	今回の演算周期のフィードバック	○	×

*6 参照値は、「0, -10, -20, -30, …」となります。

*7 参照値は、8周期(2³)ごとにTRUEとFALSEが切り替わります。(1バイトの値を毎周期「+1」してビット3の状態を取出します。)
[CONST]のビット指定は、「0~7」が使用できます。

*8 63文字を超える書式を設定する場合には、WSTRING型ラベルを使用し、下記の方法にて設定してください。

(1) モーションシステム内にWSTRING型グローバルラベルを宣言し、データ名.ビット位置@対象修飾の部分を格納します。

(2) TARGET_REFのWSTRING型ラベルとして、(1)のラベル名を設定します。(末尾に参照を示す「^」を付加します。)

■[DEV]種別の対応デバイス一覧

[DEV]種別の対応デバイス一覧を下記に示します。

R: 読出しのみ可能, RW: 読出しおよび書込みが可能, —: 未対応

デバイス種別	デバイス名	記号	対応
ユーザデバイス	入力	X	RW ^{*1*2}
	ダイレクトアクセス入力	DX	—
	出力	Y	RW ^{*1*2}
	ダイレクトアクセス出力	DY	—
	内部リレー	M	RW ^{*2}
	ラッチリレー	L	RW ^{*2}
	リンクリレー	B	RW ^{*2}
	リンク特殊リレー	SB	RW ^{*2}
	アナンシェータ	F	R
	エッジリレー	V	—
	ステップリレー (SFC対応機種のみ)	S	—
	タイマ	T	R ^{*3}
	積算タイマ	ST	R ^{*3}
	ロングタイマ	LT	R ^{*3}
	ロング積算タイマ	LST	R ^{*3}
	カウンタ	C	R ^{*3}
	ロングカウンタ	LC	R ^{*3}
	データレジスタ	D	RW ^{*2}
	リンクレジスタ	W	RW ^{*2}
	リンク特殊レジスタ	SW	RW ^{*2}
システムデバイス	特殊リレー	SM	RW
	特殊レジスタ	SD	RW
	ファンクション入力	FX	—
	ファンクション出力	FY	—
	ファンクションレジスタ	FD	—
ファイルレジスタ	ファイルレジスタ	R	—
		ZR	—
インデックスレジスタ	インデックスレジスタ	Z	—
	(32ビットインデックス修飾)	(ZZ)	—
	ロングインデックスレジスタ	LZ	—
ポインタ	ポインタ	P	—
	割込みポインタ	I	—
ユニットアクセスデバイス	ユニットアクセスデバイス	Un ¥ G	RW ^{*4}
CPUバッファメモリアクセスデバイス	CPUバッファメモリアクセスデバイス	U3E0 ¥ G	RW ^{*5}
リフレッシュデータレジスタ	リフレッシュデータレジスタ	RD	—
ネスティング	ネスティング	N	—
その他	ネットワークNo.指定デバイス	J	—
	インテリユニットNo.指定デバイス	U	—
	SFCブロックデバイス(SFC対応機種のみ)	BL	—
	SFC移行デバイス(SFC対応機種のみ)	TR	—

- *1 未割付けのデバイスのみ使用できます。
- *2 リフレッシュ設定でリフレッシュ先に設定している場合、デバイス局が接続されているか監視します。デバイス局が切断した場合の動作は各機能仕様書を参照してください。
"リフレッシュ設定"でリフレッシュ先に設定している場合、"ネットワーク同期通信設定"に"同期する"を設定している機器の信号のみ使用できます。
- *3 タイマ、積算タイマ、ロングタイマ、ロング積算タイマ、カウンタ、ロングカウンタは、接点(S)、コイル(C)、現在値(N)をもちます。データ名に指定する場合は、接点:「デバイス記号S」、コイル:「デバイス記号C」、現在値:「デバイス記号N」のように指定します。
データ名に積算タイマST10の接点を指定する場合: [DEV]STS10
データ名にロングカウンタLC10の現在値を指定する場合: [DEV]LCN10
タイマ、積算タイマ、ロングタイマ、ロング積算タイマ、カウンタ、ロングカウンタを使用する場合、型の指定をデバイスのサイズに合わせてください。
- *4 コントローラのモーション機能部またはCC-Link IE TSN機能部のバッファメモリを指定したときのみ対応です。
- *5 コントローラのCPUバッファメモリを指定したときのみ対応です。

SIGNAL_SELECT(信号選択)

変数名	名称	データ型	内容
Source	信号	TARGET_REF 1421ページ TARGET_REF(入力信号)	制御に使用するBOOL型の信号を設定します。 Source(信号)には、下記のデータ種別のみ設定できます。 • [OBJ](サイクリックデータにマッピング済のもの) • [VAR] • [DEV] • [CONST]
Detection	信号検出方法	INT (MC_SIGNAL_LOGIC) 1452ページ MC_SIGNAL_LOGIC	信号の論理を設定します。 • 0: TRUE時検出(HighLevel) • 1: FALSE時検出(LowLevel) • 2: FALSE→TRUE(立上り)時検出(RisingEdge) • 3: TRUE→FALSE(立下り)時検出(FallingEdge) • 4: 立上り/立下り時検出(BothEdges)
CompensationTime	補正時間	LREAL	入出力信号の補正時間を秒単位で設定します。 *: 各機能ごとに設定範囲が異なります。
FilterTime	フィルタ時間	LREAL	入出力信号のチャタリングを除去するためのフィルタ時間を秒単位で設定します。 *: 各機能ごとに設定範囲が異なります。

PROFILE_DATA

変数名	名称	データ型	内容
Location	演算プロファイル格納場所	FILE_LOCATION 1424ページ FILE_LOCATION	演算プロファイルのファイル名称、格納場所を設定します。
ID	プロファイルID	PROFILE_ID 1424ページ PROFILE_ID	演算プロファイルのプロファイルIDの番号を設定します。

FILE_LOCATION

変数名	名称	データ型	内容
FileName	ファイル名称	WSTRING(63)	ファイル名称を設定します。 (拡張子を含む63文字設定できます。)*1
Path	フォルダ指定	WSTRING(63)	ファイルを格納しているフォルダパスを設定します。 (63文字まで設定できます。)*1

*1 ファイル名称(FileName)とフォルダ指定(Path)の文字数は、合計127文字まで(null文字含む)としてください。

PROFILE_ID

変数名	名称	データ型	内容
Number	プロファイルID番号	WORD(UINT)	プロファイルID番号を設定します。 • 0~60000
StartIO	入出力番号	WORD	入出力No.を設定します。

CYCLE_PARAM

変数名	名称	データ型	内容
NumOfCycleOverWngDetectTimes	周期オーバー警告検出回数	WORD(UINT)	各種周期の周期オーバー警告を検出するまでの周期数を設定できます。0を設定した場合は警告を検出しません。 範囲外の値を設定した場合はデフォルトの値で動作します。 ・0~5回
NumOfCycleOverErrDetectTimes	周期オーバーエラー検出回数	WORD(UINT)	各種周期の周期オーバーエラーを検出するまでの周期数を設定できます。範囲外の値を設定した場合はデフォルトの値で動作します。 ・1~5回
CycleOverErrorType	周期オーバーエラー選択	INT (MC_ERROR_CLASS) 1452ページ MC_ERROR_CLASS	各周期のオーバーエラーの割り当てを設定します。 ・2: 軽度異常(MinorError) ・3: 中度異常(ModerateError)

CYCLE_MONI

変数名	名称	データ型	内容
ProcessingTime	処理時間	DWORD(UDINT)	各周期の処理時間[ns]を格納します。
MaximumProcessingTime	最大処理時間	DWORD(UDINT)	各周期の最大処理時間[ns]を格納します。
Cycle	設定周期	DWORD(UDINT)	各周期の周期設定[ns]を格納します。
CycleOver	周期オーバー	BOOL	各周期の周期オーバーを検出するとTRUEになります。

ADDON_PARAM

変数名	名称	データ型	内容
RamSizeMax	RAM最大サイズ	DWORD(UDINT)	アドオンで使用するシステムメモリ(RAM)の最大使用量[kバイト]を設定します。
BackupRamSizeMax	バックアップRAM最大サイズ	DWORD(UDINT)	アドオンで使用しているシステムメモリ(バックアップRAM)の最大使用量[kバイト]を設定します。

ADDON_MONI

変数名	名称	データ型	内容
RamUsage	RAM使用量	DWORD(UDINT)	アドオンで使用しているシステムメモリ(RAM)の現在の使用量[kバイト]を格納します。
RamMaxUsage	RAM使用量最大	DWORD(UDINT)	アドオンで使用しているシステムメモリ(RAM)の最大使用量[kバイト]を格納します。
BackupRamUsage	バックアップRAM使用量	DWORD(UDINT)	アドオンで使用しているシステムメモリ(バックアップRAM)の現在の使用量[kバイト]を格納します。
BackupRamMaxUsage	バックアップRAM使用量最大	DWORD(UDINT)	アドオンで使用しているシステムメモリ(バックアップRAM)の最大使用量[kバイト]を格納します。
Version	バージョン	STRING(15)	アドオンのバージョン情報を格納します。

INSTANCE_ID

変数名	名称	データ型	内容
StartIO	IO番号	WORD(HEX)	IO番号を設定します。
Number	インスタンスID番号	WORD(UINT)	インスタンスIDの番号を設定します。

PROFILE_CAM_DATA

変数名	名称 (デバイスオフセット)	データ型	内容
Interpolate	補間方法指定 (+0)	INT	カムデータの補間方法指定を設定します。 • 0: 直線補間 • 1: 区間ごとに指定 • 2: スプライン補間
Resolution	分解能 (+2)	DWORD(UDINT)	1サイクル分のカムの分解能を設定します。 • 8~65535
InputUnitString	入力単位文字列 (+4)	WSTRING(31)	入力データの単位を文字列で設定します。 省略した場合は、「pulse」として扱います。
OutputUnitString	出力単位文字列 (+36)	WSTRING(31)	出力データの単位を文字列で設定します。 省略した場合は、「pulse」として扱います。
StartPoint	開始点* ¹ (+68)	LREAL	開始点を設定します。 「0.0」を設定してください。 *: 「0.0」以外を設定した場合、演算プロファイル不正(エラーコード: 1B8FH)となります。
StartStroke	初期ストローク量* ¹ (+72)	LREAL	開始点に対するストローク量を設定します。 (-ストローク量(Stroke) ≤ 初期ストローク量(StartStroke) ≤ ストローク量(Stroke)) • -100.00000000~100.00000000
StartVelocity	初期速度* ² (+76)	LREAL	開始点に対する速度を設定します。 • -2500000000.0 < 初期速度(StartVelocity) ≤ 2500000000.0
StartAcceleration	初期加速度* ² (+80)	LREAL	開始点に対する加速度を設定します。 • -2147483647.0 < 初期加速度(StartAcceleration) ≤ 2147483647.0
CycleLength	1サイクル長* ¹ (+84)	LREAL	1サイクルに必要な入力量を設定します。 (0.000000000000001 ≤ 1サイクル長(CycleLength) ≤ 10000000000.0) • 0.000000000000001~10000000000.0
CycleMin	1サイクル最小値 (+88)	LREAL	1サイクルの最小値を設定します。 「0.0」を設定してください。 *: 「0.0」以外を設定した場合、演算プロファイル不正(エラーコード: 1B8FH)となります。
CycleMax	1サイクル最大値 (+92)	LREAL	1サイクルの最大値を設定します。 (1サイクル最小値(CycleMin) < 1サイクル最大値(CycleMax) ≤ 10000000000.0) • -10000000000.0~10000000000.0
CycleTime	1サイクル時間* ² (+96)	LREAL	1サイクルの時間を設定します。 (0.001 < 1サイクル時間(CycleTime) ≤ 100000.0) • 0.001~100000.000
Stroke	ストローク量 (+100)	LREAL	ストローク量を設定します。 (0.000000000000001 ≤ ストローク量(Stroke) ≤ 10000000000.0) • 0.000000000000001~10000000000.0
NumberOfSections	区間数* ¹ (+104)	DWORD(UDINT)	カムデータの区間数を設定します。 ■ 「1: 区間ごとに指定」を設定した場合 • 1~360 ■ 「2: スプライン補間」を設定した場合 • 3~360 *: 「0: 直線補間」を設定した場合は使用しません。設定しても無視します。
Options	オプション (+106)	DWORD(HEX)	オプションをビット指定で設定します。 ■ 繰り返し動作(ビット0) • 0: 単発動作 • 1: 繰り返し動作 ■ 主軸絶対位置(ビット1) • 0: 相対位置 • 1: 設定不可*³ ■ 従軸絶対位置(ビット2) • 0: 相対位置 • 1: 絶対位置 *: ビット3~31には「0」を設定してください。

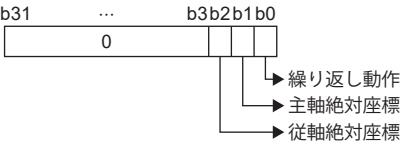
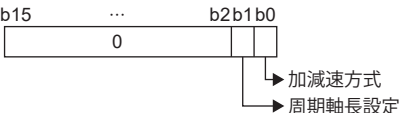
- *1 補間方法指定(Interpolate)で「0: 直線補間」を設定した場合は使用しません。設定しても無視します。
 *2 カム曲線が「12: 5次(調整)(FifthCurve_SpeedDesignation)」の場合のみ使用します。他曲線の場合は無視します。
 *3 「1」を設定すると、MasterAbsolute範囲外(エラーコード: 1B94H)となります。

PROFILE_CAM_ELEMENT

変数名	名称 (デバイスオフセット)	データ型	内容
CurveType	カム曲線種別 (+0)	INT (MC_CAM_CURVE_ TYPE) 1450ページ MC_CAM_CURVE_ TYPE	カム曲線を設定します。 • 0: 等速度(ConstantSpeed) • 1: 等加速度(ConstantAcceleration) • 2: 変形台形(DistortedTrapezoid) • 3: 変形正弦(DistortedSine) • 4: 変形等速度(DistortedConstantSpeed) • 5: サイクロイド(Cycloid) • 6: 5次(FifthCurve) • 7: トラペクロイド(Trapezoid) • 8: 逆トラペクロイド(ReverseTrapezoid) • 9: 複弦(DoubleHypotenuse) • 10: 逆複弦(ReverseDoubleHypotenuse) • 11: 単弦(SingleHypotenuse) • 12: 5次(調整)(FifthCurve_SpeedDesignation)
EndPoint	終了点*1 (+4)	LREAL	1サイクル長に対する位置(1サイクル現在値)を設定します。 • $0.0 < \text{終了点(EndPoint)} \leq 1 \text{サイクル長(CycleLength)}$
Stroke	ストローク (+8)	LREAL	ストローク位置を設定します。 • ストローク(Stroke)の絶対値 $\leq$ ストローク量(Stroke)
RangeP1	曲線適用範囲(P1) (+12)	LREAL	カム曲線の曲線適用範囲(始点: P1, 終点: P2)を設定します。 「 $P1 < P2$ 」となるように設定します。ただし、「 $P1 = P2 = 0$ 」のときは、「 $P1 = 0$ 」, 「 $P2 = 1$ 」を適用します。 • $0.0 \sim 1.0$
RangeP2	曲線適用範囲(P2) (+16)	LREAL	
RangeL1	加減速範囲補正(範囲L1) (+20)	LREAL	カム曲線で加減速する範囲(L1, L2)を設定します。 カム曲線により設定できる範囲が異なります。 「 $L1 = L2 = 0.0000$ 」のときは、カム曲線ごとのデフォルト値を適用します。 L1, L2を使用しない曲線では、設定値を無視します。 • $0.0001 < \text{範囲(L1, L2)} < 1.0000$
RangeL2	加減速範囲補正(範囲L2) (+24)	LREAL	
EndVelocity	終点速度*2 (+28)	LREAL	カム曲線の終点速度を設定します。 • $-2500000000.0 < \text{終点速度(EndVelocity)} < 2500000000.0$
EndAcceleration	終点加速度*2 (+32)	LREAL	カム曲線の終点加速度を設定します。 • $-2147483647.0 < \text{終点加速度(EndAcceleration)} < 2147483647.0$

- *1 最終区間に到達する前に終了点が1サイクル長を超えた場合、その時点の区間番号を最終区間番号とし、終了点を1サイクル長で上書きします。
 *2 演算プロファイルの補間方法指定により、読出し、または書込みに使用する設定項目が異なります。

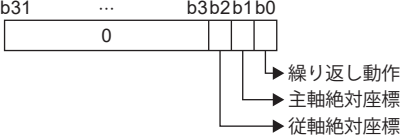
PROFILE_ROTARY_CUTTER

変数名	名称 (デバイスオフセット)	データ型	内容
Resolution	分解能 (+0)	DWORD(UDINT)	生成するカムの分解能を設定します。 • 8~32768
InputUnitString	入力単位文字列 (+2)	WSTRING(31)	入力単位文字列を設定します。
Options	オプション (+34)	DWORD(HEX)	オプションをビット指定で設定します。  ■繰り返し動作(ビット0) • 0: 単発動作 • 1: 繰り返し動作 ■主軸絶対位置(ビット1) • 0: 相対位置 • 1: 設定不可^{*1} ■従軸絶対位置(ビット2) • 0: 相対位置 • 1: 絶対位置 *: ビット3~31には「0」を設定してください。
AutoGenerationOptions	自動生成オプション (+36)	WORD(HEX)	自動生成オプションをビット指定で設定します。  ■加減速方式(ビット0) • 0: 台形加減速 • 1: S字加減速 ■同期軸長設定(ビット1) • 0: 直径 • 1: 周長 *: ビット2~15には「0」を設定してください。
NumberOfCutter	カッター数 (+37)	WORD(UINT)	カッター数を設定します。 • 設定範囲: 1~256
SyncSectionAccRatio	同期区間増速率 (+40)	LREAL	同期区間の同期速度を微調整するときに設定します。 「同期区間速度 = 同期速度 × (100% + 増速率)」となります。 • -50.0~50.0
SheetLength	シート長 (+44)	LREAL	シート長を設定します。 (0.0 < シート長(SheetLength) ≤ 10000000000.0) • 0.00000000000001~10000000000.0
SheetSyncWidth	シート同期幅 (+48)	LREAL	シート同期幅(シール幅)を設定します。 シート同期幅前後に退避動作の同期速度区間が必要な場合、退避幅分を足してください。 (0.0 < シート同期幅(SheetSyncWidth) < シート長(SheetLength)) (0.0 < シート同期幅(SheetSyncWidth) < 周長 / カッター数(NumberOfCutter)) • 0.00000000000001~10000000000.0
SyncAxisLength	同期軸長 (+52)	LREAL	ロータリーカッター軸長を設定します。 ■自動生成オプションが「0: 直径」の場合 「周長 = 設定値 × π」として計算します。 ■自動生成オプションが「1: 周長」の場合 「周長 = 設定値」として計算します。 (0.0 < 周長 < 10000000000.0) • 0.00000000000001~10000000000.0
SyncPositionAdjustment	同期位置調整 (+56)	LREAL	同期区間の位置調整を設定します。 負の値: 同期区間をシート先頭側に調整 0: シート中央が同期区間 正の値: 同期区間をシート終端側に調整 (同期位置調整(SyncPositionAdjustment)の絶対値 < シート長(SheetLength) / 2) • -10000000000.0~10000000000.0

変数名	名称 (デバイスオフセット)	データ型	内容
AccDecWidth	加減速幅 (+60)	LREAL	加減速領域のシート幅(片側分)を設定します。 負の値を設定した場合、加減速幅が最大になるように計算します。 (2 × 加減速幅(AccDecWidth) ≤ シート長(SheetLength) - シート同期幅(SheetSyncWidth)) • -10000000000.0~10000000000.0
AsyncSpd	非同期速度結果 (+64)	LREAL	自動生成が正常完了時、非同期速度が同期速度の比率として格納します。

*1 「1」を設定すると、MasterAbsolute範囲外(エラーコード: 1B94H)となります。

PROFILE_CAMSWITCH_DATA

変数名	名称 (デバイスオフセット)	データ型	内容
InputUnitString	入力単位文字列 (+0)	WSTRING(31)	入力データの単位を文字列で設定します。
NumberOfSwitches	スイッチ数 (+32)	DWORD(UDINT)	スイッチ総数を指定します。 • 1~10000
Options	オプション (+34)	DWORD(HEX)	オプションをビット指定で設定します。 ビット3~31には「0」を設定してください。  ■繰り返し動作(ビット0)*1 • 0: 単発動作 • 1: 繰り返し動作 ■主軸絶対座標(ビット1)*1 • 0: 相対座標 • 1: 絶対座標 ■従軸絶対位置(ビット2)*1 • 0: 相対座標 • 1: 絶対座標
ReferenceSpeed	基準速度 (+36)	LREAL	本設定は使用しません。0.0を設定してください。
ReferenceSpeedUnitString	基準速度単位文字列 (+40)	WSTRING(31)	本設定は使用しません。空白を設定してください。
RingCount_LowerValue	リングカウンタ下限値 (+72)	LREAL	リングカウンタの下限値を設定します。
RingCount_UpperValue	リングカウンタ上限値 (+76)	LREAL	リングカウンタの上限値を設定します。

*1 「1」を指定してください。

PROFILE_CAMSWITCH_ELEMENT

変数名	名称 (デバイスオフセット)	データ型	内容
FirstOnPosition	ON開始位置 (+0)	LREAL	スイッチがONする下限位置, またはスイッチがONする開始位置を設定します。 • -10000000000.0~10000000000.0
LastOnPosition	ON終了位置 (+4)	LREAL	スイッチがONする上限位置を設定します。 • -10000000000.0~10000000000.0
Direction	方向選択 (+8)	INT	スイッチがONとなる入力の方角を設定します。 • 0: 両方向 • 1: 正方向 • 2: 負方向
CamSwitchMode	スイッチモード選択 (+9)	INT	ON条件の指定方法を設定します。 • 0: スイッチ無効 • 1: 位置基準
Duration	ON時間 (+12)	LREAL	本設定は使用しません。設定値は無視されます。

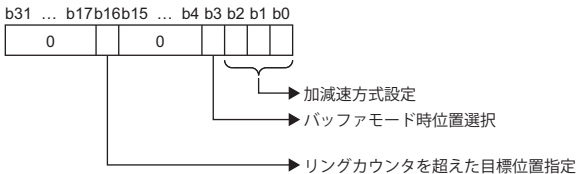
PROFILE_POSITIONING_DATA

変数名	名称 (デバイスオフセット)	データ型	内容
NumberOfDataSettings	位置決めデータ設定数 (+0)	WORD(UINT)	位置決めデータの数を設定します。 位置決めデータ設定数(NumberOfDataSettings)が位置決めデータ総数(TotalNumberOfData)以下となるように設定します。 • 0~5000 範囲外の値を設定, または位置決めデータ総数を超えた値を設定した場合, 展開時に演算プロファイル不正(エラーコード: 1B8FH)となります。
TotalNumberOfData	位置決めデータ総数 (+1)	WORD(UINT)	位置決めデータの総数を設定します。 展開や書き込みで設定する位置決めデータNo.の最大値を設定します。 • 1~5000 範囲外の値を設定した場合, 展開時に演算プロファイル不正(エラーコード: 1B8FH)となります。
Options	オプション (+2)	DWORD(HEX)	オプションを設定します。 「00000000H」を設定してください。

PROFILE_POSITIONING_DATA_ELEMENT

変数名	名称 (デバイスオフセット)	データ型	内容
DataNo	位置決めデータNo. (+0)	WORD(UINT)	位置決めデータのインデックス番号を設定します。 位置決めデータNo.(DataNo)の順に位置決めデータを実行します。位置決めデータNo.(DataNo)が昇順となるよう位置決めデータを設定します。 • 1~位置決めデータ総数(5000)
OperationPattern	運転パターン (+1)	INT (MC_OPERATION_P ATTERN) 1453ページ MC_OPERATION_P ATTERN	位置決めデータの運転パターンを設定します。 位置決めデータNo.を続けて実行するか, 終了するかを設定します。 • 0: 位置決め終了(PositioningComplete) • 1: 連続位置決め(ContinuousPositioning) • 2: 連続軌跡(BlendingLow)(ContinuousBlendingLow) • 3: 連続軌跡(BlendingPrevious)(ContinuousBlendingPrevious) • 4: 連続軌跡(BlendingNext)(ContinuousBlendingNext) • 5: 連続軌跡(BlendingHigh)(ContinuousBlendingHigh)
ControlMethod	制御方式 (+2)	INT (MC_CONTROL_ME THOD) 1453ページ MC_CONTROL_MET HOD	位置決めデータNo.の制御方式を設定します。 • 0080H: NOP(NOP) • 0082H: JUMP(JUMP) • 0083H: LOOP(LOOP) • 0084H: LEND(LEND) • 0101H: 絶対値直線補間(LinearAbsolute) • 0102H: 相対値直線補間(LinearRelative) • 0103H: 絶対値円弧補間(CircularAbsolute) • 0104H: 相対値円弧補間(CircularRelative) • 0105H: 絶対値ヘリカル補間(HelicalAbsolute) • 0106H: 相対値ヘリカル補間(HelicalRelative)

変数名	名称 (デバイスオフセット)	データ型	内容
InterpolationAxis1	補間軸1 (+3)	WORD(UINT)	軸グループの構成軸の中より、補間制御を行う補間軸を設定します。 制御方式(ControlMethod)の設定により、直線補間の直線補間軸、または円弧補間の円弧補間軸を設定します 補間軸として指定する構成軸は、位置決めデータごとに異なる構成軸と補間軸数を設定できます。ただし、連続軌跡データと次の位置決めデータの間で、指定した構成軸や補間軸数が異なる場合、位置決めデータ切換え時に各軸速度が急変することがありますので注意してください。 ・ 1~16
InterpolationAxis2	補間軸2 (+4)	WORD(UINT)	
InterpolationAxis3	補間軸3 (+5)	WORD(UINT)	
InterpolationAxis4	補間軸4 (+6)	WORD(UINT)	
CircMode	円弧補間モード (+7)	INT (MC_CIRC_MODE) 📖 1449ページ MC_CIRC_MODE	円弧補間制御の指定方法を設定します。 ・ 0: 境界点指定(mcBorder) ・ 1: 中心点指定(mcCenter) ・ 2: 半径指定(mcRadius)
CircPathChoice	経路選択 (+8)	INT (MC_CIRC_PATHCH OICE) 📖 1449ページ MC_CIRC_PATHCH OICE	円弧補間制御の回転方向を設定します。 ・ 0: CW(mcCW) ・ 1: CCW(mcCCW) ・ 2: 近回り(mcShortWay) ・ 3: 遠回り(mcLongWay) ・ 4: CW遠回り(mcCWLongWay) ・ 5: CCW遠回り(mcCCWLongWay)
Reserve1[0..2]	予約1 (+9)	WORD(UINT)	使用しません。(オフセット調整用)
Position1	目標位置/移動量/終点1 (+12)	LREAL	制御方式(ControlMethod)の設定により下記となります。 <直線補間> 目標位置/移動量/終点1~4(Position1~Position4)は、補間軸1~4(InterpolationAxis1~InterpolationAxis4)で設定した構成軸の目標位置/移動量を設定します。 ・ -10000000000.0~10000000000.0 「0」、または省略した補間軸1~4(InterpolationAxis1~InterpolationAxis4)に対応する、目標位置/移動量/終点1~4(Position1~Position4)は省略できます。 ■「0101H: 絶対値直線補間(LinearAbsolute)」の場合 絶対位置の目標位置を設定します。 ■「0102H: 相対値直線補間(LinearRelative)」の場合 位置決めデータ実行時の現在位置からの移動量を設定します。
Position2	目標位置/移動量/終点2 (+16)	LREAL	
Position3	目標位置/移動量/終点3 (+20)	LREAL	
Position4	目標位置/移動量/終点4 (+24)	LREAL	
Direction1	方向選択1 (+28)	INT (MC_DIRECTION) 📖 1448ページ MC_DIRECTION	直線補間制御にて現在位置から目標位置へ移動する方向を設定します。 制御方式(ControlMethod)で、「0101H: 絶対値直線補間(LinearAbsolute)」を設定した場合に設定します。 方向選択1~4(Direction1~Direction4)は、補間軸1~4(InterpolationAxis1~InterpolationAxis4)で設定した構成軸の方向を設定します。 ・ 1: 正方向(mcPositiveDirection) ・ 2: 負方向(mcNegativeDirection) ・ 3: 最短経路(mcShortestWay) 設定によって下記ようになります。 ・ ソフトウェアストロークリミット有効の補間軸は、設定を無視します。 ・ 「0」、または省略した補間軸1~4(InterpolationAxis1~InterpolationAxis4)に対応する、方向選択1~4(Direction1~Direction4)の配列要素は省略できます。 す。
Direction2	方向選択2 (+29)	INT (MC_DIRECTION) 📖 1448ページ MC_DIRECTION	
Direction3	方向選択3 (+30)	INT (MC_DIRECTION) 📖 1448ページ MC_DIRECTION	
Direction4	方向選択4 (+31)	INT (MC_DIRECTION) 📖 1448ページ MC_DIRECTION	

変数名	名称 (デバイスオフセット)	データ型	内容
CircAuxPoint1	補助点1 (+32)	LREAL	円弧補間制御を行う補助点(境界点, 中心点, 半径)の位置を設定します。 円弧補間モード(CircMode)の設定により設定範囲は, 下記となります。
CircAuxPoint2	補助点2 (+36)	LREAL	■「0: 境界点指定(mcBorder)」, 「1: 中心点指定(mcCenter)」の場合 制御方式(ControlMethod)により下記となります。
CircAuxPoint3	補助点3 (+40)	LREAL	・「0103H: 絶対値円弧補間(CircularAbsolute)」: -10000000000.0 ≤ 設定値 < 10000000000.0 リングカウンタ有効の場合, リングカウンタ範囲となります。
CircAuxPoint4	補助点4 (+44)	LREAL	・「0104H: 相対値円弧補間(CircularRelative)」: -10000000000.0 ~ 10000000000.0 ■「2: 半径指定(mcRadius)」の場合 ・0.000001 ~ 2147483647.0
VelocityMode	速度モード (+48)	INT (MC_INTERPOLATE_ SPEED_MODE) 1450ページ MC_INTERPOLATE_ SPEED_MODE	直線補間制御の速度モードを設定します。 ・0: 合成速度(VectorSpeed) ・1: 長軸速度(LongAxisSpeed) ・2: 基準軸速度(ReferenceAxisSpeed)
Reserve2[0..2]	予約2 (+49)	WORD(UINT)	使用しません。(オフセット調整用)
Velocity	速度 (+52)	LREAL	多軸補間の位置決め速度を設定します。 ・0.0000, 0.0001 ~ 2500000000.0
Acceleration	加速度 (+56)	LREAL	多軸補間の加速度を設定します。 位置決めデータオプション(DataOptions)の加減速方式設定(ビット0~2)の設定により設定範囲は下記となります。 ■「0: 加減速度指定方式(mcAccDec)」の場合 ・0.0000, 0.0001 ~ 2147483647.0[U/s ²] ■「1: 加減速時間一定方式(mcFixedTime)」の場合 ・0.000000, 0.000001 ~ 8400.0[s]
Deceleration	減速度 (+60)	LREAL	多軸補間の減速度を設定します。 ■「0: 加減速度指定方式(mcAccDec)」の場合 ・0.0000, 0.0001 ~ 2147483647.0[U/s ²] ■「1: 加減速時間一定方式(mcFixedTime)」の場合 ・減速度(Deceleration)を使用しません
Jerk	ジャーク (+64)	LREAL	多軸位置決め制御のジャークを設定します。 ■「0: 加減速度指定方式(mcAccDec)」の場合 ・0.0000, 0.0001 ~ 2147483647.0[U/s ³] ■「1: 加減速時間一定方式(mcFixedTime)」の場合 ・ジャーク(Jerk)を使用しません
CircErrorTolerance	円弧補間誤差許容値 (+68)	LREAL	中心指定の円弧補間を行う場合, 円弧補間誤差許容値を設定します。 ・0.000001 ~ 100000.0
DataOptions	位置決めデータオプション (+72)	DWORD(HEX)	オプションをビット指定で設定します。  ■加減速方式設定(ビット0~2) ・0: 加減速度指定方式(mcAccDec) ・1: 加減速時間一定方式(mcFixedTime) ■バッファモード時位置選択(ビット3) 制御方式(ControlMethod)で「0102H: 相対値直線補間(LinearRelative)」, 「0104H: 相対値円弧補間(CircularRelative)」を設定した場合に設定します。 ・0: 指令現在位置 ・1: フィードバック位置 「0101H: 絶対値直線補間(LinearAbsolute)」, 「0103H: 絶対値円弧補間(CircularAbsolute)」の場合は, 「0」を設定してください。 ■リングカウンタを超えた目標位置指定(ビット16) 制御方式(ControlMethod)で「0101H: 絶対値直線補間(LinearAbsolute)」を設定した場合に設定します。 ・0: 許可しない ・1: 許可する 「0102H: 相対値直線補間(LinearRelative)」, 「0103H: 絶対値円弧補間(CircularAbsolute)」, 「0104H: 相対値円弧補間(CircularRelative)」の場合は, 「0」を設定してください。

変数名	名称 (デバイスオフセット)	データ型	内容
Reserve3[0..1]	予約3 (+74)	WORD(UINT)	使用しません。(オフセット調整用)
DwellTime	ドウェルタイム (+76)	LREAL	ドウェルタイムを設定します。 • 0.0[s]: ドウェル機能無効 • 0.000001~8400.0[s]: ドウェル機能有効 0.000001未満の正数を設定した場合、「0.0[s]」として取込みます。
Mcode	Mコード (+80)	WORD(UINT)	位置決めデータごとのMコードを設定します。 Mコードを出力しない場合は、「0(初期値)」を設定します。 • 1~65535
McodeOutput_Override	Mコード出力タイミング オーバーライド (+81)	INT (MC_MCODE_OUTP UT_OVERRIDE) 1453ページ MC_MCODE_OUTP UT_OVERRIDE	位置決めデータごとのMコード出力タイミングを設定します。 • 0: FBオプションを使用(UseFbOptions) • 1: WITHモード(WithMode) • 2: AFTERモード(AfterMode)
JumpDestinationDataNo	ジャンプ先位置決めデータNo. (+82)	WORD(UINT)	JUMPで指定するジャンプ先の位置決めデータNo.を設定します。 制御方式(ControlMethod)が「0082H: JUMP(JUMP)」以外の位置決めデータの位置決めデータNo.(DataNo)を設定します。 • 1~位置決めデータ総数(5000)
ConditionSignalNo	条件信号No. (+83)	WORD(UINT)	条件信号No.を設定します。 条件信号を使用しない場合は、「0(初期値)」を設定します。 • 1~10
LoopCount	LOOP~LENDループ回数 (+84)	WORD(UINT)	ループ制御の繰り返し回数を設定します。 • 1~65535
SkipSignalNo	スキップ信号No. (+85)	WORD(UINT)	スキップ信号No.を設定します。 • 1~10 スキップ信号を使用しない場合は、「0」を設定してください。
Pitch	ピッチ (+86)	WORD(UINT)	ピッチを設定します。 「0」を設定してください。
Reserve4	予約4 (+87)	WORD(UINT)	使用しません。(オフセット調整用)

PROFILE_POSITIONING_DATA_EXTENDED

変数名	名称 (デバイスオフセット)	データ型	内容
ConditionSignal	条件信号 (+0)	WSTRING(63)	条件信号をTARGET_REF型の文字列で設定します。詳細は、下記を参照してください。 1421ページ TARGET_REF(入力信号) 文字列の設定範囲は、下記となります。*1 ■データ種別 • [VAR] • [DEV] ■型 • PROFILE_SIGNAL_SELECT_ELEMENT(1434ページ PROFILE_SIGNAL_SELECT_ELEMENT) ""(設定なし)の場合、読出し/書き込みを行いません。
SkipSignal	スキップ信号 (+64)	WSTRING(63)	スキップ信号をTARGET_REF型の文字列で設定します。詳細は、下記を参照してください。 1421ページ TARGET_REF(入力信号) 文字列の設定範囲は、下記となります。*1 ■データ種別 • [VAR] • [DEV] ■型 • PROFILE_SIGNAL_SELECT_ELEMENT(1434ページ PROFILE_SIGNAL_SELECT_ELEMENT) ""(設定なし)の場合、読出し/書き込みを行いません。

変数名	名称 (デバイスオフセット)	データ型	内容
PositioningData	位置決めデータ (+128)	WSTRING(63)	位置決めデータをTARGET_REF型の文字列で設定します。詳細は、下記を参照してください。 1421ページ TARGET_REF(入力信号) 文字列の設定範囲は、下記となります。 ■データ種別 • [VAR] • [DEV] ■型 • POFILE_POSITIONING_DATA_ELEMENT(1430ページ PROFILE_POSITIONING_DATA_ELEMENT) ""(設定なし)の場合、読出し/書込みデータ型不正(エラーコード: 1BAAH)となります。

*1 指定した型が各構造体メンバの要求を満たさない場合は、読出し/書込みデータ型不正(エラーコード: 1BAAH)となります。また、指定したデータ種別が各構造体メンバの要求を満たさない場合は、読出し/書込みデータデータ種別不正(エラーコード: 1BA8H)となります。

PROFILE_SIGNAL_SELECT_ELEMENT

変数名	名称 (デバイスオフセット)	データ型	内容*1
StartIO	入出力番号 (+0)	WORD(HEX)	使用しません。 設定を無視します。
Reserve1	予約1 (+1)	WORD(UINT)	使用しません。(オフセット調整用)
Target	対象 (+2)	WSTRING(63)	使用する信号を文字列で設定します。詳細は、下記を参照してください。 1421ページ TARGET_REF(入力信号) 文字列の設定範囲は、下記となります。 • ""(長さ0の文字列)*2 ■データ種別 • [OBJ]*3 • [VAR] • [DEV] ■型 • BOOL
Detection	信号検出方法 (+66)	INT (MC_SIGNAL_LOGIC) 1452ページ MC_SIGNAL_LOGIC	信号の論理を指定します。 • 0: TRUE時検出(HighLevel) • 1: FALSE時検出(LowLevel)
Reserve2	予約2 (+67)	WORD(UINT)	使用しません。(オフセット調整用)
CompensationTime	補正時間 (+68)	LREAL	「0.0」を設定します。
FilterTime	フィルタ時間 (+72)	LREAL	入出力信号のチャタリングを除去するためのフィルタ時間を秒単位で設定します。 • 0.0~5.0[s]

*1 書込み時に設定値によるエラーは発生しません。書込んだ展開エリアを演算プロファイル制御FBが使用するとき、設定範囲外の場合にエラーとなります。

*2 対象(Target)が「""(長さ0の文字列)」の場合、該当のスキップ信号を設定なしとして扱います。

*3 [OBJ]を設定した場合、対象修飾を省略できません。省略して設定すると、演算プロファイル制御プロファイルID設定不正(エラーコード: 0D48H)となります。

SLAVE_OBJECT_REAL

変数名	名称	データ型	内容
ControlWord	ControlWord	WSTRING(63)	ドライバ機器に状態の切換えを要求します。
EncoderIncrements	EncoderIncrements	WSTRING(63)	ドライバ機器のエンコーダ分解能を取得します。
FollowingErrActualValue	FollowingErrActualValue	WSTRING(63)	ドライバ機器の溜りパルスです。
GearRatioMotorRevolutions	GearRatioMotorRevolutions	WSTRING(63)	ドライバ機器にサーボモータ軸回転数(分子)を設定します。
HomeOffset	HomeOffset	WSTRING(63)	ドライバ機器に機械座標系のゼロ位置と原点復帰位置の差を設定します。
MaxMotorSpeed	MaxMotorSpeed	WSTRING(63)	ドライバ機器からサーボモータの最大速度を取得します。

変数名	名称	データ型	内容
MaxTorque	MaxTorque	WSTRING(63)	ドライバ機器からサーボモータの最大トルクを取得します。
ModesOfOp	ModesOfOp	WSTRING(63)	ドライバ機器へ制御モードの切換えを要求します。
ModesOfOpDisp	ModesOfOpDisp	WSTRING(63)	ドライバ機器の制御モードを取得します。
NegativeTorqueLimitValue	NegativeTorqueLimitValue	WSTRING(63)	ドライバ機器に負方向トルク制限値を設定します。
Polarity	Polarity	WSTRING(63)	ドライバ機器の回転方向選択を設定します。
PosActualValue	PosActualValue	WSTRING(63)	ドライバ機器の現在位置です。
PosEncoderResolution	PosEncoderResolution	WSTRING(63)	Sub Indexの総数(=2)を返信します。
PosEncoderResolutionMotorRevolutions	PosEncoderResolutionMotorRevolutions	WSTRING(63)	ドライバ機器の回転数です。
PositiveTorqueLimitValue	PositiveTorqueLimitValue	WSTRING(63)	ドライバ機器に正方向トルク制限値を設定します。
ShaftRevolutions	ShaftRevolutions	WSTRING(63)	ドライバ機器の駆動軸回転数(分母)を設定します。
SiUnitPos	SiUnitPos	WSTRING(63)	ドライバ機器のSI単位位置を取得します。
SiUnitVel	SiUnitVel	WSTRING(63)	ドライバ機器のSI単位速度を取得します。
StatusWord	StatusWord	WSTRING(63)	ドライバ機器の状態を取得します。
SupportedDriveModes	SupportedDriveModes	WSTRING(63)	ドライバ機器の対応する制御モードを取得します。
TargetPos	TargetPos	WSTRING(63)	ドライバ機器へ出力する指令位置です。
TargetTorque	TargetTorque	WSTRING(63)	ドライバ機器へ出力する指令トルクです。
TargetVelocity	TargetVelocity	WSTRING(63)	ドライバ機器へ出力する指令速度です。
TargetActualValue	TorqueActualValue	WSTRING(63)	ドライバ機器の現在トルクです。
VelActualValue	VelActualValue	WSTRING(63)	ドライバ機器の現在速度です。
vControlDi1	ControlDi1	WSTRING(63)	ドライバ機器に入力デバイスの設定を行います。
vControlDi2	ControlDi2	WSTRING(63)	ドライバ機器に入力デバイスの設定を行います。
vControlDi3	ControlDi3	WSTRING(63)	ドライバ機器に入力デバイスの設定を行います。
vControlDi4	ControlDi4	WSTRING(63)	ドライバ機器に入力デバイスの設定を行います。
vControlDi5	ControlDi5	WSTRING(63)	ドライバ機器に入力デバイスの設定を行います。
vControlDi6	ControlDi6	WSTRING(63)	ドライバ機器に入力デバイスの設定を行います。
vControlDi7	ControlDi7	WSTRING(63)	ドライバ機器に入力デバイスの設定を行います。
vCurrentAlarm	CurrentAlarm	WSTRING(63)	ドライバ機器から発生中のアラームを取得します。
vEncoderStatus1	EncoderStatus1	WSTRING(63)	ドライバ機器からエンコーダ状態を取得します。
vHomeAbsCounter	HomeAbsCounter	WSTRING(63)	ドライバ機器から原点として保存されたエンコーダ多回転カウンタを取得します。
vHomeCycleCounter	HomeCycleCounter	WSTRING(63)	ドライバ機器から電源投入時のエンコーダ1回転内位置を取得します。
vInitialAbsCounter	InitialAbsCounter	WSTRING(63)	ドライバ機器から電源投入時のエンコーダ多回転カウンタを取得します。
vInitialCycleCounter	InitialCycleCounter	WSTRING(63)	ドライバ機器から電源投入時のエンコーダ1回転内位置を取得します。
vInitialPos	InitialPos	WSTRING(63)	ドライバ機器から電源投入時の現在位置を取得します。
vMaxAbsCounter	MaxAbsCounter	WSTRING(63)	ドライバ機器からエンコーダ多回転カウンタの最大値を取得します。
vStatusDo1	StatusDo1	WSTRING(63)	ドライバ機器の出力デバイス状態を取得します。
vStatusDo2	StatusDo2	WSTRING(63)	ドライバ機器の出力デバイス状態を取得します。
vStatusDo3	StatusDo3	WSTRING(63)	ドライバ機器の出力デバイス状態を取得します。
vStatusDo4	StatusDo4	WSTRING(63)	ドライバ機器の出力デバイス状態を取得します。
vStatusDo5	StatusDo5	WSTRING(63)	ドライバ機器の出力デバイス状態を取得します。
vStatusDo6	StatusDo6	WSTRING(63)	ドライバ機器の出力デバイス状態を取得します。
vStatusDo7	StatusDo7	WSTRING(63)	ドライバ機器の出力デバイス状態を取得します。
vSupportedControlDi1	SupportedControlDi1	WSTRING(63)	ドライバ機器がサポートしている入力デバイスを取得します。
vSupportedControlDi2	SupportedControlDi2	WSTRING(63)	ドライバ機器がサポートしている入力デバイスを取得します。
vSupportedControlDi3	SupportedControlDi3	WSTRING(63)	ドライバ機器がサポートしている入力デバイスを取得します。
vSupportedControlDi4	SupportedControlDi4	WSTRING(63)	ドライバ機器がサポートしている入力デバイスを取得します。
vSupportedControlDi5	SupportedControlDi5	WSTRING(63)	ドライバ機器がサポートしている入力デバイスを取得します。
vSupportedControlDi6	SupportedControlDi6	WSTRING(63)	ドライバ機器がサポートしている入力デバイスを取得します。
vSupportedControlDi7	SupportedControlDi7	WSTRING(63)	ドライバ機器がサポートしている入力デバイスを取得します。
vSupportedStatusDo1	SupportedStatusDo1	WSTRING(63)	ドライバ機器がサポートしている出力デバイスを取得します。
vSupportedStatusDo2	SupportedStatusDo2	WSTRING(63)	ドライバ機器がサポートしている出力デバイスを取得します。

変数名	名称	データ型	内容
vSupportedStatusDo3	SupportedStatusDo3	WSTRING(63)	ドライバ機器がサポートしている出力デバイスを取得します。
vSupportedStatusDo4	SupportedStatusDo4	WSTRING(63)	ドライバ機器がサポートしている出力デバイスを取得します。
vSupportedStatusDo5	SupportedStatusDo5	WSTRING(63)	ドライバ機器がサポートしている出力デバイスを取得します。
vSupportedStatusDo6	SupportedStatusDo6	WSTRING(63)	ドライバ機器がサポートしている出力デバイスを取得します。
vSupportedStatusDo7	SupportedStatusDo7	WSTRING(63)	ドライバ機器がサポートしている出力デバイスを取得します。
vSyncAbsCounter	SyncAbsCounter	WSTRING(63)	ドライバ機器からスケール計測エンコーダの多回転カウンタを取得します。
vSyncCycleCounter	SyncCycleCounter	WSTRING(63)	ドライバ機器からスケール計測エンコーダの1回転内位置を取得します。
vVelLimitValue	VelLimitValue	WSTRING(63)	ドライバ機器に速度制限値を設定します。
vWatchDogCounterDI	WatchDogCounterDI	WSTRING(63)	ドライバ機器にウォッチドッグカウンタ値を通知します。
vWatchDogCounterUI	WatchDogCounterUI	WSTRING(63)	ドライバ機器にウォッチドッグカウンタ値を取得します。
vEncoderStatus2	EncoderStatus2	WSTRING(63)	ドライバ機器からスケール計測エンコーダ状態を取得します。
vScaleCycleCounter	ScaleCycleCounter	WSTRING(63)	ドライバ機器からスケール計測エンコーダの1回転内位置を取得します。
vScaleAbsCounter	ScaleAbsCounter	WSTRING(63)	ドライバ機器からスケール計測エンコーダの多回転カウンタを取得します。
vScaleMeasurementEncoderResolution	ScaleMeasurementEncoderResolution	WSTRING(63)	ドライバ機器からスケール計測エンコーダの分解能を取得します。
vScaleMeasurementEncoderReceptionStatus	ScaleMeasurementEncoderReceptionStatus	WSTRING(63)	ドライバ機器からエンコーダ1回転内位置を取得します。

SLAVE_OBJECT_VIRTUAL_ENCODER

変数名	名称	データ型	内容
PosActualValue	PosActualValue	WSTRING(63)	エンコーダ入力値として使用するデータを文字列書式にて設定します。種別には、[VAR], [DEV], [CONST]が設定できます。 ■[VAR]を設定した場合 - データ型が(INT), (DINT), (WORD), (DWORD)となるデータを設定します。 [VAR]を使用する場合、ローカルラベルは設定できません。 ■[DEV], [CONST]を設定した場合 (INT), (DINT), (WORD), (DWORD)にてデータ型を設定します。 *: 設定を省略した場合は、エンコーダ入力値 (<u>AxisName.Cd.Encoder_InputValue</u>)を使用します。

その他構造体の変数(アドバンスト同期制御)

40

ADV_GEAR_PARAM(ギアパラメータ)

変数名	名称	取込	データ型	属性	内容
MasterCompositeGear	主軸合成ギア	演算周期 (FB起動中)	WORD(HEX)	LIST_WRITE _BACK	主軸とサブ入力軸の現在位置の合成方法を設定します。 • 0000H: 主軸の入力なし/サブ入力軸の入力なし • 0001H: 主軸の入力+/サブ入力軸の入力なし • 0002H: 主軸の入力-/サブ入力軸の入力なし • 0010H: 主軸の入力なし/サブ入力軸の入力+ • 0011H: 主軸の入力+/サブ入力軸の入力+ • 0012H: 主軸の入力-/サブ入力軸の入力+ • 0020H: 主軸の入力なし/サブ入力軸の入力- • 0021H: 主軸の入力+/サブ入力軸の入力- • 0022H: 主軸の入力-/サブ入力軸の入力-
MasterNumerator	主軸ギア分子	始動時	DINT	LIST_WRITE _BACK	主軸ギアの分子を設定します。 • -2147483647~2147483647
MasterDenominator	主軸ギア分母	始動時	DWORD(UDINT)	LIST_WRITE _BACK	主軸ギアの方母を設定します。 • 1~2147483647
AuxCompositeGear	補助軸合成ギア	演算周期 (FB起動中)	WORD(HEX)	LIST_WRITE _BACK	主軸とサブ入力軸合成後の現在位置と補助軸の現在位置の合成方法を設定します。 • 0000H: 主軸の入力なし/補助軸の入力なし • 0001H: 主軸の入力+/補助軸の入力なし • 0002H: 主軸の入力-/補助軸の入力なし • 0010H: 主軸の入力なし/補助軸の入力+ • 0011H: 主軸の入力+/補助軸の入力+ • 0012H: 主軸の入力-/補助軸の入力+ • 0020H: 主軸の入力なし/補助軸の入力- • 0021H: 主軸の入力+/補助軸の入力- • 0022H: 主軸の入力-/補助軸の入力-
AuxNumerator	補助軸ギア分子	始動時	DINT	LIST_WRITE _BACK	補助軸ギアの分子を設定します。 • -2147483647~2147483647
AuxDenominator	補助軸ギア分母	始動時	DWORD(UDINT)	LIST_WRITE _BACK	補助軸ギアの方母を設定します。 • 1~2147483647

ADV_CLUTCH_PARAM(クラッチパラメータ)

変数名	名称	取込	データ型	属性	内容
MasterOnControl	主軸ONクラッチ制御設定	演算周期 (FB起動中)	INT (MC_CLUTCH_METHOD) 1454ページ MC_CLUTCH_METHOD	LIST_WRITE _BACK	主軸クラッチのON制御方法を設定します。 • 0: 無効(NoClutch)(直結動作) • 1: クラッチ指令(ClutchCommand) • 2: クラッチ指令立上り(ClutchLeadingEdge) • 3: クラッチ指令立下り(ClutchTrailingEdge) • 4: アドレスモード(ClutchAddress) • 15: 出力データ指定(ClutchSignal) *: 同期制御中にも変更できますが、「0: 無効(NoClutch)」以外の設定から「0: 無効(NoClutch)」へ変更することはできません。
MasterOffControl	主軸OFFクラッチ制御設定	演算周期 (FB起動中)	INT (MC_CLUTCH_METHOD) 1454ページ MC_CLUTCH_METHOD	LIST_WRITE _BACK	主軸クラッチのOFF制御方法を設定します。 • 0: 無効(NoClutch)(OFF制御無効) • 1: クラッチ指令(ClutchCommand)(ワンショット動作) • 2: クラッチ指令立上り(ClutchLeadingEdge) • 3: クラッチ指令立下り(ClutchTrailingEdge) • 4: アドレスモード(ClutchAddress) • 15: 出力データ指定(ClutchSignal)

変数名	名称	取込	データ型	属性	内容
MasterReference	主軸クラッチ参照アドレス設定	始動時	INT (MC_CLUTCH_REFERENCE) 1454ページ MC_CLUTCH_REFERENCE	LIST_WRITE _BACK	主軸クラッチ制御時に参照するアドレスを設定します。 0: ギア前現在位置 (GearFrontPosition) 1: ギア後1サイクル現在位置 (GearPositionPerCycle) *: 参照アドレスにより、ギアとクラッチの処理順序が変わりますので注意してください。
MasterOnAddress	主軸クラッチONアドレス	演算周期 (FB起動中)	LREAL	LIST_WRITE _BACK	主軸ONクラッチ制御設定 (AdvOutputName.Pr.Clutch.MasterOnControl)に「4: アドレスモード (ClutchAddress)」を設定したとき、クラッチをONするアドレスを設定します。 主軸クラッチ参照アドレス設定 (AdvOutputName.Pr.Clutch.MasterReference)が「1: ギア後1サイクル現在位置 (GearPositionPerCycle)」の場合、設定した値は「0.0~(1サイクル長 - 0.00001)」の範囲内に変換して制御します。 • -10000000000.0~10000000000.0
MasterMovementAmountBeforeOn	主軸クラッチON前移動量	クラッチON条件成立時	LREAL	LIST_WRITE _BACK	主軸クラッチON制御でON条件が成立してから実際にクラッチがONするまでの参照アドレスの移動量を符号付きの値で設定します。 • -2147483648.0~2147483647.0
MasterOffAddress	主軸クラッチOFFアドレス	演算周期 (FB起動中)	LREAL	LIST_WRITE _BACK	主軸OFFクラッチ制御設定 (AdvOutputName.Pr.Clutch.MasterOffControl)に「4: アドレスモード (ClutchAddress)」を設定したとき、クラッチをOFFするアドレスを設定します。 主軸クラッチ参照アドレス設定 (AdvOutputName.Pr.Clutch.MasterReference)が「1: ギア後1サイクル現在位置 (GearPositionPerCycle)」の場合、設定した値は「0.0~(1サイクル長 - 0.00001)」の範囲内に変換して制御します。 • -10000000000.0~10000000000.0
MasterMovementAmountBeforeOff	主軸クラッチOFF前移動量	クラッチOFF条件成立時	LREAL	LIST_WRITE _BACK	主軸クラッチOFF制御でOFF条件が成立してから実際にクラッチがOFFするまでの参照アドレスの移動量を符号付きの値で設定します。 • -2147483648.0~2147483647.0
MasterSmoothingMethod	主軸クラッチスムージング方式	始動時	INT (MC_CLUTCH_SMOOTHING_METHOD) 1454ページ MC_CLUTCH_SMOOTHING_METHOD	LIST_WRITE _BACK	主軸クラッチON/OFF時のスムージング方式を設定します。 0: ダイレクト (ClutchSmoothingDisabled) 1: 時定数方式(指数) (TimeConstantExponent) 2: 時定数方式(直線) (TimeConstantLinear) 3: 滑り量方式(指数) (SlippageExponent) 4: 滑り量方式(直線) (SlippageLinear) 5: 滑り量方式(直線: 入力量追従) (SlippageLinearFollow)

変数名	名称	取込	データ型	属性	内容
MasterOnSlippageAmount	主軸クラッチ ON時滑り量	クラッチ ON開始時	LREAL	LIST_WRITE _BACK	主軸クラッチスムージング方式 (AdvOutputName.Pr.Clutch.MasterS moothingMethod)に「3: 滑り量方 式(指数)(SlippageExponent)」、 「4: 滑り量方式(直線)(SlippageLinear)」、 「5: 滑り量方式(直線: 入力量追従) (SlippageLinearFollow)」を設定し た場合、クラッチをONしたときの 滑り量を設定します。 • 0.0~2147483647.0 *: 設定値が負の値の場合、主軸ク ラッチON時滑り量 (AdvOutputName.Pr.Clutch.Mast erOnSlippageAmount)を「0.0(ダイレクト)」として制御しま す。
MasterOffSlippageAmount	主軸クラッチ OFF時滑り量	クラッチ OFF開始時	LREAL	LIST_WRITE _BACK	主軸クラッチスムージング方式 (AdvOutputName.Pr.Clutch.MasterS moothingMethod)に「3: 滑り量方 式(指数)(SlippageExponent)」、 「4: 滑り量方式(直線)(SlippageLinear)」、 「5: 滑り量方式(直線: 入力量追従) (SlippageLinearFollow)」を設定し た場合、クラッチをOFFしたときの 滑り量を設定します。 • 0.0~2147483647.0 *: 設定値が負の値の場合、主軸ク ラッチOFF時滑り量 (AdvOutputName.Pr.Clutch.Mast erOffSlippageAmount)を「0.0(ダイレクト)」として制御しま す。
AuxOnControl	補助軸ONク ラッチ制御設 定	演算周期 (FB起動中)	INT (MC_CLUTCH_METHOD) 1454ページ MC_CLUTCH_METHOD	LIST_WRITE _BACK	補助軸クラッチのON制御方法を設 定します。 • 0: 無効(NoClutch)(直結動作) • 1: クラッチ指令(ClutchCommand) • 2: クラッチ指令立上り (ClutchLeadingEdge) • 3: クラッチ指令立下り (ClutchTrailingEdge) • 4: アドレスモード(ClutchAddress) • 15: 入出力データ指定 (ClutchSignal) *: 同期制御中にも変更できますが、 「0: 無効(直結動作)(NoClutch)」 以外の設定から「0: 無効(直結動 作)(NoClutch)」へ変更すること はできません。
AuxOffControl	補助軸OFFク ラッチ制御設 定	演算周期 (FB起動中)	INT (MC_CLUTCH_METHOD) 1454ページ MC_CLUTCH_METHOD	LIST_WRITE _BACK	補助軸クラッチのOFF制御方法を設 定します。 • 0: 無効(NoClutch)(OFF制御無効) • 1: クラッチ指令 (ClutchCommand)(ワンショット 動作) • 2: クラッチ指令立上り (ClutchLeadingEdge) • 3: クラッチ指令立下り (ClutchTrailingEdge) • 4: アドレスモード(ClutchAddress) • 15: 入出力データ指定 (ClutchSignal)
AuxReference	補助軸クラッ チ参照アドレ ス設定	始動時	INT (MC_CLUTCH_REFERENCE) 1454ページ MC_CLUTCH_REFERENCE	LIST_WRITE _BACK	補助軸クラッチ制御時に参照するア ドレスを設定します。 • 0: ギア前現在位置 (GearFrontPosition) • 1: ギア後1サイクル現在位置 (GearPositionPerCycle) *: 参照アドレスにより、ギアとク ラッチの処理順序が変わります ので注意してください。

変数名	名称	取込	データ型	属性	内容
AuxOnAddress	補助軸クラッチONアドレス	演算周期 (FB起動中)	LREAL	LIST_WRITE _BACK	補助軸ONクラッチ制御設定 (AdvOutputName.Pr.Clutch.AuxOnControl)に「4: アドレスモード (ClutchAddress)」を設定したとき、 クラッチをONするアドレスを設定 します。 補助軸クラッチ参照アドレス設定 (AdvOutputName.Pr.Clutch.AuxReference)が「1: ギア後1サイクル現在 位置(GearPositionPerCycle)」の場 合、設定した値は「0.0~(1サイクル 長 - 0.00001)」の範囲内に変換して 制御します。 • -10000000000.0~10000000000.0
AuxMovementAmountBefore On	補助軸クラッチON前移動量	クラッチ ON条件成 立時	LREAL	LIST_WRITE _BACK	補助軸クラッチON制御でON条件が 成立してから実際にクラッチがON するまでの参照アドレスの移動量を 符号付きの値で設定します。 • -2147483648.0~2147483647.0
AuxOffAddress	補助軸クラッチOFFアドレス	演算周期 (FB起動中)	LREAL	LIST_WRITE _BACK	補助軸OFFクラッチ制御設定 (AdvOutputName.Pr.Clutch.AuxOffControl)に「4: アドレスモード (ClutchAddress)」を設定したとき、 クラッチをOFFするアドレスを設定 します。 補助軸クラッチ参照アドレス設定 (AdvOutputName.Pr.Clutch.AuxReference)が「1: ギア後1サイクル現在 位置(GearPositionPerCycle)」の場 合、設定した値は「0.0~(1サイクル 長 - 0.00001)」の範囲内に変換して 制御します。 • -10000000000.0~10000000000.0
AuxMovementAmountBefore Off	補助軸クラッチOFF前移動量	クラッチ OFF条件成 立時	LREAL	LIST_WRITE _BACK	補助軸クラッチOFF制御でOFF条件 が成立してから実際にクラッチが OFFするまでの参照アドレスの移動 量を符号付きの値で設定します。 • -2147483648.0~2147483647.0
AuxSmoothingMethod	補助軸クラッチスムージング方式	始動時	INT (MC_CLUTCH_SMOOTHING_METHOD) 1454ページ MC_CLUTCH_SMOOTHING_METHOD	LIST_WRITE _BACK	補助軸クラッチON/OFF時のスムー ジング方式を設定します。 • 0: ダイレクト (ClutchSmoothingDisabled) • 1: 時定数方式(指数 (TimeConstantExponent)) • 2: 時定数方式(直線 (TimeConstantLinear)) • 3: 滑り量方式(指数 (SlippageExponent)) • 4: 滑り量方式(直線 (SlippageLinear)) • 5: 滑り量方式(直線: 入力量追従 (SlippageLinearFollow))
AuxOnSlippageAmount	補助軸クラッチON時滑り量	クラッチ ON開始時	LREAL	LIST_WRITE _BACK	補助軸クラッチスムージング方式 (AdvOutputName.Pr.Clutch.AuxSmoothingMethod)に「3: 滑り量方式(指 数)(SlippageExponent)」、「4: 滑り量 方式(直線)(SlippageLinear)」、「5: 滑 り量方式(直線: 入力量追従 (SlippageLinearFollow))」を設定し た場合、クラッチをONしたときの 滑り量を設定します。 • 0.0~2147483647.0 *: 設定値が負の値の場合、補助軸 クラッチON時滑り量 (AdvOutputName.Pr.Clutch.AuxOnSlippageAmount)を「0.0(ダ イレクト)」として制御します。

変数名	名称	取込	データ型	属性	内容
AuxOffSlippageAmount	補助軸クラッチOFF時滑り量	クラッチOFF開始時	LREAL	LIST_WRITE _BACK	補助軸クラッチスムージング方式(AdvOutputName.Pr.Clutch.AuxSmoothingMethod)に「3: 滑り量方式(指数)(SlippageExponent)」、「4: 滑り量方式(直線)(SlippageLinear)」、「5: 滑り量方式(直線: 入力量追従)(SlippageLinearFollow)」を設定した場合、クラッチをOFFしたときの滑り量を設定します。 • 0.0~2147483647.0 *: 設定値が負の値の場合、補助軸クラッチOFF時滑り量(AdvOutputName.Pr.Clutch.AuxOffSlippageAmount)を「0.0(ダイレクト)」として制御します。

ADV_CLUTCH_MONI(クラッチモニタ)

変数名	名称	取込	データ型	属性	内容
MasterOnOffStatus	主軸クラッチON/OFFステータス	演算周期 (FB起動中)	BOOL	LIST_READ _ONLY	主軸クラッチのON/OFF状態が格納されます。 • FALSE: クラッチOFF状態 • TRUE: クラッチON状態
MasterSmoothingStatus	主軸クラッチスムージングステータス	演算周期 (FB起動中)	BOOL	LIST_READ _ONLY	主軸クラッチのスムージング状態が格納されます。 • FALSE: クラッチスムージングなし • TRUE: クラッチスムージング中
MasterCumulativeSlippage	主軸クラッチ滑り量累積値	演算周期 (FB起動中)	LREAL	LIST_READ _ONLY	滑り量方式のクラッチスムージング時の滑り量累積値を符号付きで格納されます。
AuxOnOffStatus	補助軸クラッチON/OFFステータス	演算周期 (FB起動中)	BOOL	LIST_READ _ONLY	補助軸クラッチのON/OFF状態が格納されます。 • FALSE: クラッチOFF状態 • TRUE: クラッチON状態
AuxSmoothingStatus	補助軸クラッチスムージングステータス	演算周期 (FB起動中)	BOOL	LIST_READ _ONLY	補助軸クラッチのスムージング状態が格納されます。 • FALSE: クラッチスムージングなし • TRUE: クラッチスムージング中
AuxCumulativeSlippage	補助軸クラッチ滑り量累積値	演算周期 (FB起動中)	LREAL	LIST_READ _ONLY	滑り量方式のクラッチスムージング時の滑り量累積値を符号付きで格納されます。

ADV_CLUTCH_CMD(クラッチ制御データ)

変数名	名称	取込	データ型	属性	内容
MasterCommand	主軸クラッチ指令	演算周期 (FB起動中)	BOOL	LIST_READ _ONLY	主軸クラッチ指令のON/OFFを設定します。 主軸ONクラッチ制御設定(AdvOutputName.Pr.Clutch.MasterOnControl)が「1: クラッチ指令(ClutchCommand)」、「2: クラッチ指令立上り(ClutchLeadingEdge)」、「3: クラッチ指令立下り(ClutchTrailingEdge)」のときに使用します。 • FALSE: クラッチ指令OFF • TRUE: クラッチ指令ON
MasterInvalidCommand	主軸クラッチ制御無効指令	演算周期 (FB起動中)	BOOL	LIST_READ _ONLY	主軸クラッチ制御の有効/無効を切り換えます。 クラッチON/OFF状態はクラッチ制御が無効になる前の状態を維持します。 クラッチON前移動中やクラッチOFF前移動中は、クラッチ制御が無効になりません。クラッチON前移動やクラッチOFF前移動が完了した後にクラッチ制御が無効になります。 • FALSE: クラッチ制御有効 • TRUE: クラッチ制御無効

変数名	名称	取込	データ型	属性	内容
MasterForcedOff	主軸クラッチ強制OFF指令	演算周期 (FB起動中)	BOOL	LIST_READ_ONLY	主軸クラッチを強制的にOFFします。 • FALSE: クラッチ通常制御 • TRUE: クラッチ強制OFF *: クラッチスムージング実行中で、即時にクラッチからの出力が「0」になります。 *: 滑り量方式のスムージングが行われていた場合、累積滑り量は0クリアします。 *: クラッチ強制OFF後、設定値を「FALSE」にするとクラッチOFF状態から通常のクラッチ制御を再開します。
AuxCommand	補助軸クラッチ指令	演算周期 (FB起動中)	BOOL	LIST_READ_ONLY	補助軸クラッチ指令のON/OFFを設定します。 補助軸ONクラッチ制御設定 (AdvOutputName.Pr.Clutch.AuxOnControl)が「1: クラッチ指令(ClutchCommand)」、「2: クラッチ指令立上り(ClutchLeadingEdge)」、「3: クラッチ指令立下り(ClutchTrailingEdge)」のときに使用します。 • FALSE: クラッチ指令OFF • TRUE: クラッチ指令ON
AuxInvalidCommand	補助軸クラッチ制御無効指令	演算周期 (FB起動中)	BOOL	LIST_READ_ONLY	補助軸クラッチ制御の有効/無効を切り換えます。 クラッチON/OFF状態はクラッチ制御が無効になる前の状態を維持します。 クラッチON前移動中やクラッチOFF前移動中は、クラッチ制御が無効になりません。クラッチON前移動やクラッチOFF前移動が完了した後にクラッチ制御が無効になります。 • FALSE: クラッチ制御有効 • TRUE: クラッチ制御無効
AuxForcedOff	補助軸クラッチ強制OFF指令	演算周期 (FB起動中)	BOOL	LIST_READ_ONLY	補助軸クラッチを強制的にOFFします。 • FALSE: クラッチ通常制御 • TRUE: クラッチ強制OFF *: クラッチスムージング実行中で、即時にクラッチからの出力が「0」になります。 *: 滑り量方式のスムージングが行われていた場合、累積滑り量は0クリアします。 *: クラッチ強制OFF後、設定値を「FALSE」にするとクラッチOFF状態から通常のクラッチ制御を再開します。
MasterClutchSmoothingTimeConstant	主軸クラッチスムージング時定数変更値	始動時	INT	LIST_READ_ONLY	主軸クラッチスムージング時定数 (AdvOutPutName.Cd.Clutch.MasterClutchSmoothingTimeConstant)を変更します。 • 0~5000[ms] *: 負の値を設定した場合、スムージングは行いません。 *: 「0」を設定した場合、主軸クラッチスムージング時定数 (AdvOutPutName.PrConst.MasterClutchSmoothingTimeConstant)の設定値でスムージングします。
AuxClutchSmoothingTimeConstant	補助軸クラッチスムージング時定数変更値	始動時	INT	LIST_READ_ONLY	補助軸クラッチスムージング時定数 (AdvOutPutName.Cd.Clutch.AuxClutchSmoothingTimeConstant)を変更します。 • 0~5000[ms] *: 負の値を設定した場合、スムージングは行いません。 *: 「0」を設定した場合、補助軸クラッチスムージング時定数 (AdvOutPutName.PrConst.AuxClutchSmoothingTimeConstant)の設定値でスムージングします。

ADV_SPEEDCHANGEGEAR_PARAM(変速機パラメータ)

変数名	名称	取込	データ型	属性	内容
MasterGearIn	主軸変速機有効設定	始動時	BOOL	LIST_WRITE _BACK	主軸の変速機を使用するか、使用しないかを設定します。 • FALSE: 変速機なし • TRUE: 変速機あり
MasterRatioNumerator	主軸変速比分子	演算周期 (FB起動中)	DINT	LIST_WRITE _BACK	主軸変速比の分子を設定します。 • -2147483647~2147483647
MasterRatioDenominator	主軸変速比分母	演算周期 (FB起動中)	DWORD(UDINT)	LIST_WRITE _BACK	主軸変速比の分母を設定します。 • 1~2147483647
AuxGearIn	補助軸変速機有効設定	始動時	BOOL	LIST_WRITE _BACK	補助軸の変速機を使用するか、使用しないかを設定します。 • FALSE: 変速機なし • TRUE: 変速機あり
AuxRatioNumerator	補助軸変速比分子	演算周期 (FB起動中)	DINT	LIST_WRITE _BACK	補助軸変速比の分子を設定します。 • -2147483647~2147483647
AuxRatioDenominator	補助軸変速比分母	演算周期 (FB起動中)	DWORD(UDINT)	LIST_WRITE _BACK	補助軸変速比の分母を設定します。 • 1~2147483647
OutGearIn	出力軸変速機有効設定	始動時	BOOL	LIST_WRITE _BACK	出力軸の変速機を使用するか、使用しないかを設定します。 • FALSE: 変速機なし • TRUE: 変速機あり
OutRatioNumerator	出力軸変速比分子	演算周期 (FB起動中)	DINT	LIST_WRITE _BACK	出力軸変速比の分子を設定します。 • -2147483647~2147483647
OutRatioDenominator	出力軸変速比分母	演算周期 (FB起動中)	DWORD(UDINT)	LIST_WRITE _BACK	出力軸変速比の分母を設定します。 • 1~2147483647

ADV_SPEEDCHANGEGEAR_CMD(変速機制御データ)

変数名	名称	取込	データ型	属性	内容
MasterSmoothingTimeConstant	主軸変速機スムージング時定数変更値	始動時	INT	LIST_WRITE _BACK	主軸変速機スムージング時定数を変更します。 • 0~5000[ms] *: 負の値を設定した場合、スムージングは行いません。 *: 「0」を設定した場合、主軸変速機スムージング時定数 (AdvOutPutName.PrConst.MasterSpeedChangeGearSmoothingTimeConstant)の設定値でスムージングします。
AuxSmoothingTimeConstant	補助軸変速機スムージング時定数変更値	始動時	INT	LIST_WRITE _BACK	補助軸の変速機スムージング時定数を変更します。 • 0~5000[ms] *: 負の値を設定した場合、スムージングは行いません。 *: 「0」を設定した場合、補助軸変速機スムージング時定数 (AdvOutPutName.PrConst.AuxSpeedChangeGearSmoothingTimeConstant)の設定値でスムージングします。
OutSmoothingTimeConstant	出力軸変速機スムージング時定数変更値	始動時	INT	LIST_WRITE _BACK	出力軸の変速機スムージング時定数を変更します。 • 0~5000[ms] *: 負の値を設定した場合、スムージングは行いません。 *: 「0」を設定した場合、出力軸変速機スムージング時定数 (AdvOutPutName.PrConst.OutSpeedChangeGearSmoothingTimeConstant)の設定値でスムージングします。

ADV_CAM_PARAM(カムパラメータ)

変数名	名称	取込	データ型	属性	内容
LengthPerCycle	1サイクル長	始動時/カムデータ 0点目通過時	LREAL	LIST_WRITE _BACK	1サイクル現在位置を生成するためのカム軸の1サイクルの長さを設定します。 MCv_AdvancedSync(アドバンス同期制御)の主軸(Master)に設定した入力軸の単位となります。 • 1.0~2147483647.0
StrokeAmount	カムストローク量	始動時/カムデータ 0点目通過時	LREAL	LIST_WRITE _BACK	演算プロファイルデータの補間方法指定(Interpolate)が「1: 区間ごとに指定」, または, 「2: スプライン補間」の場合, またはデータ形式の種別に「ロータリーカッター」を設定した場合, ストローク比100%に対応するカムストローク量を出力軸位置単位で設定します。 • -2147483648.0~2147483647.0
CamNo	カムNo.	始動時/カムデータ 0点目通過時	WORD(UINT)	LIST_WRITE _BACK	カム制御に使用するカムNo.を設定します。 • 0~60000
StartingPoint	カム開始点	始動時/カムデータ 0点目通過時	DWORD(UDINT)	LIST_WRITE _BACK	カムデータを開始する位置を分解能で設定します。 演算プロファイルデータの補間方法指定(Interpolate)が「1: 区間ごとに指定」, または, 「2: スプライン補間」の場合, またはデータ形式の種別に「ロータリーカッター」を設定した場合のみ有効です。 カム開始点 (AdvOutputName.Pr.Cam.StrokeAmount)の初期値は「0」です。(カムデータ0点目(ストローク比0%)からカム軸を制御します。)「0」以外を設定すると, ストローク比が0%以外から始まるカム制御を行います。 • 0~65535
LengthPerCycleChange	1サイクル長 変更設定	始動時	INT (MC_LENGTH_PER_CYCLE _CHANGE)  1454ページ MC_LENGTH_PER_CYCLE _CHANGE	LIST_WRITE _BACK	1サイクル長 (AdvOutputName.Pr.Cam.LengthPerCycle)を同期制御中に変更する場合に設定します。 • 0: 無効(LengthChangeInvalid) • 1: 有効(LengthChangeValid)
PhaseCompensationAdvancedTime	位相補正進 め時間	始動時/演算周期 (FB起動中)	DINT	LIST_WRITE _BACK	カム制御時, 1サイクル現在位置の位相を進めたり, 遅らせたりするときに設定します。 • -100000000~100000000[μs]
PhaseCompensationTimeConstant	位相補正時 定数	始動時	WORD(UINT)	LIST_WRITE _BACK	位相補正時の位相補正量を一次遅れで反映するときの時定数を設定します。 設定した時定数で位相補正量の63%を反映します。 • 0~65535[ms]

ADV_CAM_MONI(カムモニタ)

変数名	名称	取込	データ型	属性	内容
PhaseCompensationAmount	位相補正量	演算周期 (FB起動中)	LREAL	LIST_READ _ONLY	カム軸の位相補正量をMCv_AdvancedSync(アドバンス同期制御)の主軸(Master)に設定した入力軸の単位で格納します。 位相補正時定数 (AdvOutputName.Pr.Cam.PhaseCompensationTimeConstant)でスムージング処理された後の位相補正量を格納します。
MasterCompositeGearSetPosition	主軸合成ギア後現在位置	演算周期 (FB起動中)	LREAL	LIST_READ _ONLY	主軸合成ギアで主軸とサブ入力軸を合成した後の現在位置を累積値として格納します。 MCv_AdvancedSync(アドバンス同期制御)の主軸(Master)に設定した入力軸の単位で格納します。
MasterGearPositionPerCycle	主軸ギア後1サイクル現在位置	演算周期 (FB起動中)	LREAL	LIST_READ _ONLY	主軸ギア後の入力移動量が「0.0~(1サイクル長 - 0.00001)」の範囲で格納します。 MCv_AdvancedSync(アドバンス同期制御)の主軸(Master)に設定した入力軸の単位となります。
AuxGearPositionPerCycle	補助軸ギア後1サイクル現在位置	演算周期 (FB起動中)	LREAL	LIST_READ _ONLY	補助軸ギア後の入力移動量を「0.0~(1サイクル長 - 0.00001)」の範囲で格納します。 MCv_AdvancedSync(アドバンス同期制御)の主軸(Master)に設定した入力軸の単位となります。
PositionPerCycle	1サイクル現在位置	演算周期 (FB起動中)	LREAL	LIST_READ _ONLY	1サイクル現在位置を「0.0~(1サイクル長 - 0.00001)」の範囲で格納します。位相補正処理後の現在位置をモニタできます。 MCv_AdvancedSync(アドバンス同期制御)の主軸(Master)に設定した入力軸の単位となります。
ReferenceSetPosition	カム基準位置	演算周期 (FB起動中)	LREAL	LIST_READ _ONLY	カム動作のカム基準位置となる指令現在位置を格納します。 出力軸の軸情報(AdvOutputName.Axis)に設定した軸の位置指令単位となります。
SetPosition	カム指令現在位置	演算周期 (FB起動中)	LREAL	LIST_READ _ONLY	カム軸の指令現在位置を格納します。 出力軸の軸情報(AdvOutputName.Axis)に設定した軸の位置指令単位となります。
CamNo	実行プロファイルNo.	演算周期 (FB起動中)	WORD(UINT)	LIST_READ _ONLY	実行中のカムNo.を格納します。同期制御中にカムNo.(AdvOutputName.Pr.Cam.CamNo)を変更した場合、実際に制御中のカムNo.が切り変わったときに更新します。実行プロファイルID番号(AxisName.Md.ProfileID)にも同じ値が格納されます。
StrokeAmount	実行カムストローク量	演算周期 (FB起動中)	LREAL	LIST_READ _ONLY	実行中のカムストローク量を格納します。 同期制御中にカムストローク量(AdvOutputName.Pr.Cam.StrokeAmount)を変更した場合、実際に制御中のカムストローク量が切り変わったときに更新します。
LengthPerCycle	実行1サイクル長	演算周期 (FB起動中)	LREAL	LIST_READ _ONLY	実行中のカムの1サイクル長を格納します。同期制御中に1サイクル長(AdvOutputName.Pr.Cam.LengthPerCycle)を変更した場合、実際に制御中の1サイクル長が切り変わったときに更新します。
SyncControlChangeStatus	同期制御変更状態	演算周期 (FB起動中)	INT	LIST_READ _ONLY	同期制御変更要求(AdvOutputName.Cd.Cam.SyncControlChangeRequest)にて同期制御変更を行った状態を格納します。 ・ -1: 同期制御変更失敗 ・ 0: 同期制御変更要求なし ・ 1: 同期制御変更中 ・ 2: 同期制御変更完了
StartingPoint	実行カム開始点	演算周期 (FB起動中)	DWORD(UDINT)	LIST_READ _ONLY	実行中のカム開始点を格納します。同期制御中にカム開始点(AdvOutputName.Pr.Cam.StartingPoint)を変更した場合、カムが切り変わったときに更新します。

ADV_CAM_CMD(カム制御データ)

変数名	名称	取込	データ型	属性	内容
SyncControlChangeRequest	同期制御変更要求	演算周期 (FB起動中)	BOOL	LIST_READ _ONLY	同期制御変更要求を行います。 TRUE(同期制御変更要求)を設定すると同期制御 変更命令 (AdvOutputName.Cd.Cam.SyncControlChangeC ommand)を実行します。 ・ FALSE: 同期制御変更要求なし ・ TRUE: 同期制御変更要求
SyncControlChangeCommand	同期制御変更命令	演算周期 (FB起動中)	INT (MC_SYNC_CHANGE_ COMMAND) 📄 1454ページ MC_SYNC_CHANGE_ COMMAND	LIST_READ _ONLY	同期制御変更命令を設定します。 ・ 0: カム基準位置移動 (ReferenceSetPositionMovement) ・ 1: 1サイクル現在位置変更 (ChangeCurrentPositionPerCycle) ・ 2: 主軸ギア後1サイクル現在位置変更 (ChangeMasterGearPositionPerCycle) ・ 3: 補助軸ギア後1サイクル現在位置変更 (ChangeAuxGearPositionPerCycle) ・ 4: 1サイクル現在位置移動 (PositionPerCycleMovement)
SyncControlChange	同期制御変更値	演算周期 (FB起動中)	LREAL	LIST_READ _ONLY	同期制御変更処理の変更値を設定します。 ・ -2147483648.0~2147483647.0
SyncControlReflectionTime	同期制御変更反映 時間	演算周期 (FB起動中)	WORD(UINT)	LIST_READ _ONLY	同期制御変更処理の反映時間を設定します。 ・ 0~65535[ms]
OutSmoothingTimeConstant	出力軸スムージン グ時定数変更値	始動時	INT	LIST_WRI TE_BACK	出力軸スムージング時定数を変更します。 ・ 0~5000[ms] *: 負の値を設定した場合、スムージングは行 いません。 *: 「0」を設定した場合、出力軸スムージン グ時定数 (AdvOutputName.PrConst.SmoothingTime Constant)の設定値でスムージングします。

ADV_RESTORE_PARAM(同期制御初期位置パラメータ)

変数名	名称	取込	データ型	属性	内容
MasterGearPositionPerCycleMethod	主軸ギア後1サイ クル現在位置設定 方法	始動時	INT (MC_GEAR_RESTORE_ METHOD) 📄 1454ページ MC_GEAR_RESTORE_ METHOD	LIST_WRITE _BACK	MCv_AdvancedSync(アドバンス同期制 御)始動時に主軸ギア後1サイクル現在位置 (AdvOutputName.Md.Gear.MasterGearPosi tionPerCycle)の設定方法を設定します。 ・ 0: 前回値(PreviousPosition) ・ 1: 主軸ギア後1サイクル現在位置初期設 定値(InitialGearPositionPerCycle) ・ 2: 入力軸から計算 (CalculateFromInputAxis)
MasterGearInitialPositionPerCycle	主軸ギア後1サイ クル現在位置初期 設定値	始動時	LREAL	LIST_WRITE _BACK	主軸ギア後1サイクル現在位置設定方法 (AdvOutputName.Md.Gear.MasterGearPosi tionPerCycleMethod)に「1: 主軸ギア後1サ イクル現在位置初期設定値 (InitialGearPositionPerCycle)」を設定した 場合、主軸ギア後1サイクル現在位置の初 期設定値を設定します。 「0.0~(1サイクル長 - 0.00001)」の範囲で設 定します。 ・ 0.0~2147483647.0
AuxGearPositionPerCycleMethod	補助軸ギア後1サ イクル現在位置設 定方法	始動時	INT (MC_GEAR_RESTORE_ METHOD) 📄 1454ページ MC_GEAR_RESTORE_ METHOD	LIST_WRITE _BACK	MCv_AdvancedSync(アドバンス同期制 御)始動時に補助軸ギア後1サイクル現在位 置 (AdvOutputName.Md.Gear.AuxGearPositio nPerCycle)の設定方法を設定します。 ・ 0: 前回値(PreviousPosition) ・ 1: 補助軸ギア後1サイクル現在位置初期 設定値(InitialGearPositionPerCycle) ・ 2: 入力軸から計算 (CalculateFromInputAxis)

変数名	名称	取込	データ型	属性	内容
AuxGearInitialPositionPerCycle	補助軸ギア後1サイクル現在位置初期設定値	始動時	LREAL	LIST_WRITE _BACK	補助軸ギア後1サイクル現在位置設定方法 (AdvOutputName.Md.Gear.AuxGearPositionPerCycle)に「1: 補助軸ギア後1サイクル現在位置初期設定値 (InitialGearPositionPerCycle)」を設定した場合、補助軸ギア後1サイクル現在位置の初期設定値を設定します。 「0.0~(1サイクル長 - 0.00001)」の範囲で設定します。 • 0.0~2147483647.0
PositionRestorationObject	カム軸位置復元対象	始動時	INT (MC_CAM_RESTORE_METHOD) 📄 1455ページ MC_CAM_RESTORE_METHOD	LIST_WRITE _BACK	同期制御始動時に復元する対象を設定します。 • 0: 1サイクル現在位置復元 (RestorePositionPerCycle) • 1: カム基準位置復元 (RestoreReferenceSetPosition) • 2: カム指令現在位置復元 (RestoreCamSetPosition)
ReferenceSetPositionMethod	カム基準位置設定方法	始動時	INT (MC_CAM_REFERENCE_METHOD) 📄 1455ページ MC_CAM_REFERENCE_METHOD	LIST_WRITE _BACK	カム軸位置復元対象 (AdvOutputName.Pr.Restore.PositionRestorationObject)に「0: 1サイクル現在位置復元 (RestorePositionPerCycle)」、「2: カム指令現在位置復元 (RestoreCamSetPosition)」を設定した場合、復元に使用するカム基準位置の設定方法を設定します。 • 0: 前回値 (PreviousPosition) • 1: カム基準位置初期設定値 (InitialReferenceSetPosition) • 2: カム指令現在位置 (CamSetPosition)
InitialReferenceSetPosition	カム基準位置初期設定値	始動時	LREAL	LIST_WRITE _BACK	カム基準位置設定方法 (AdvOutputName.Pr.Restore.PositionRestorationMethod)に「1: カム基準位置初期設定値 (InitialReferenceSetPosition)」を設定した場合、カム基準位置の初期設定値を出力軸の軸情報 (AdvOutputName.Axis)に設定した軸の単位で設定します。 • -10000000000.0~10000000000.0
PositionPerCycleMethod	1サイクル現在位置設定方法	始動時	INT (MC_CAM_CYCLE_METHOD) 📄 1455ページ MC_CAM_CYCLE_METHOD	LIST_WRITE _BACK	カム軸位置復元対象 (AdvOutputName.Pr.Restore.PositionRestorationObject)に「1: カム基準位置復元 (RestorePositionPerCycle)」、「2: カム指令現在位置復元 (RestoreCamSetPosition)」を設定した場合、復元に使用する1サイクル現在位置の設定方法を設定します。 • 0: 前回値 (PreviousPosition) • 1: 1サイクル現在位置初期設定値 (InitialPositionPerCycle) • 2: 主軸ギア後1サイクル現在位置 (MasterGearPositionPerCycle) • 3: 補助軸ギア後1サイクル現在位置 (AuxGearPositionPerCycle)
InitialPositionPerCycle	1サイクル現在位置初期設定値	始動時	LREAL	LIST_WRITE _BACK	カム軸位置復元対象 (AdvOutputName.Pr.Restore.PositionRestorationObject)の設定により値を設定します。 MCv_AdvancedSync(アドバンス同期制御)の主軸(Master)に設定した入力軸の単位となります。 「0.0~(1サイクル長 - 0.00001)」の範囲で設定します。 • 0.0~2147483647.0
RestorationAllowablePosition	カム指令現在位置復元許容移動量設定値	始動時	DWORD(UDINT)	LIST_WRITE _BACK	カム軸位置復元対象 (AdvOutputName.Pr.Restore.PositionRestorationObject)に「2: カム指令現在位置復元」を設定した場合、復元したカム指令現在位置と指令現在位置の差の許容値を出力軸の位置指令値の単位で設定します。 設定値が大きいと同期始動時に急激な動作になることがあります。 • 0.0~2147483647.0[pulse]

40.2 列挙子一覧

各種パラメータやモニタデータ，モーション制御FBで使用する列挙子の一覧を下記に示します。

ENUM列挙子

各種パラメータやモニタデータ，モーション制御FBで使用する列挙型の定数は，実際にはINT型の値を使用します。エンジニアリングツールでは，「列挙型名__列挙子名」のINT型グローバルラベルが使用できます。

例

MC_BUFFER_MODE型の列挙子「mcBuffered」をエンジニアリングツールで使用する場合

- ・INT型グローバルラベルで使用する時：「MC_BUFFER_MODE__mcBuffered」を設定します。

MC_BUFFER_MODE

列挙子	設定値	内容
mcAborting	0	Aborting
mcBuffered	1	Buffered
mcBlendingLow	2	BlendingLow
mcBlendingPrevious	3	BlendingPrevious
mcBlendingNext	4	BlendingNext
mcBlendingHigh	5	BlendingHigh

MC_EXECUTION_MODE

列挙子	設定値	内容
mcImmediately	0	直ちに実行
mcQueued	1	完了待ちして実行
mcSpeculatively	3	投機的に実行

MC_COMBINE_MODE

列挙子	設定値	内容
mcAddAxes	0	2つの入力軸の位置を加算
mcSubAxes	1	2つの入力軸の位置を減算

MC_DIRECTION

列挙子	設定値	内容
mcPositiveDirection	1	正方向
mcNegativeDirection	2	負方向
mcShortestWay	3	最短経路
mcCurrentDirection	4	現在方向

MC_SOURCE

列挙子	設定値	内容
mcSetValue	1	指令現在値
mcActualValue	2	フィードバック値
mcLatestSetValue	101	最新指令現在値
mcLatestActualValue	102	最新フィードバック値

MC_CIRC_MODE

列挙子	設定値	内容
mcBorder	0	境界点指定
mcCenter	1	中心点指定
mcRadius	2	半径指定

MC_CIRC_PATHCHOICE

列挙子	設定値	内容
mcCW	0	CW
mcCCW	1	CCW
mcShortWay	2	近回り
mcLongWay	3	遠回り
mcCWLongWay	4	CW遠回り
mcCCWLongWay	5	CCW遠回り

MC_START_MODE

列挙子	設定値	内容
mcImmediate	0	即時
mcAbsolute	1	絶対

MC_AXIS_STATUS

列挙子	設定値	内容
Invalid	-1	軸変数未初期化/軸パラメータ異常
Disabled	0	軸無効
ErrorStop	1	エラー停止中
Stopping	2	減速停止中
Homing	3	原点復帰中
Standstill	4	待機中
DiscreteMotion	5	位置決め運転中
ContinuousMotion	6	連続動作運転中
SynchronizedMotion	7	同期運転中

MC_AXES_GROUP_STATUS

列挙子	設定値	内容
Invalid	-1	軸グループ変数未初期化/軸グループパラメータ異常
GroupDisabled	0	軸グループ無効
GroupErrorStop	1	エラー停止中
GroupStopping	2	減速停止中
GroupStandby	4	待機中
GroupMoving	5	動作中

MC_DRIVE_MODE

列挙子	設定値	内容
NoModeChange	0	変更しない
hm	6	hm(原点復帰)
csp	8	csp(位置制御)
csv	9	csv(速度制御)
cst	10	cst(トルク制御)
ct	-104	ct(押当て制御)

MC_VELOCITY_LIMIT_MODE

列挙子	設定値	内容
Ignore	0	無視
Truncate	2	切り捨て
ImmediateStop	3	即停止
ClampWithoutRamp	4	クランプ(減速時傾斜なし)

MC_INTERPOLATE_SPEED_MODE

列挙子	設定値	内容
VectorSpeed	0	合成速度
LongAxisSpeed	1	長軸速度
ReferenceAxisSpeed	2	基準軸速度

MC_RECORD_MODE

列挙子	設定値	内容
OneShot	0	単発モード
RecordCount	1	指定回数モード
RingBuffer	2	リングバッファモード

MC_CAM_CURVE_TYPE

列挙子	設定値	内容
ConstantSpeed	0	等速度
ConstantAcceleration	1	等加速度
DistortedTrapezoid	2	変形台形
DistortedSine	3	変形正弦
DistortedConstantSpeed	4	変形等速度
Cycloid	5	サイクロイド
FifthCurve	6	5次
Trapecloid	7	トラペクロイド
ReverseTrapecloid	8	逆トラペクロイド
DoubleHypotenuse	9	複弦
ReverseDoubleHypotenuse	10	逆複弦
SingleHypotenuse	11	単弦
FifthCurve_SpeedDesignation	12	5次(調整)

MC_AXIS_TYPE

列挙子	設定値	内容
DriveAxis	0	実ドライブ軸
EncoderAxis	2	実エンコーダ軸
VirtualDriveAxis	3	仮想ドライブ軸
VirtualEncoderAxis	4	仮想エンコーダ軸
VirtualLinkAxis	5	仮想連結軸

MC_ENCODER_AXIS_TYPE

列挙子	設定値	内容
Drive	1	ドライブユニット経由

MC_ABS_SYSTEM

列挙子	設定値	内容
ABSDisabled	0	絶対位置システムを使用しない
Enabled	1	絶対位置システムを使用する
Auto	-1	自動設定(接続機器から取得)

MC_DRIVE_STATE

列挙子	設定値	内容
NotReadyToSwitchOn	0	Not Ready To Switch On
Fault	1	Fault
FaultReactionActive	2	Fault Reaction Active
SwitchOnDisabled	3	Switch On Disabled
ReadyToSwitchOn	4	Ready To Switch On
SwitchedOn	5	Switched On
OperationEnable	6	Operation Enable
QuickStopActive	7	Quick Stop Active
Invalid	-1	Invalid

MC_DECEL_STOP_MODE

列挙子	設定値	内容
Ignore	0	無視
ImmediateStop	1	即停止
KeepCurrentAcc	2	現在の加減速度を続行
AlternativeAcc	3	加減速度を代替
ServoOffAfterImmediateStop	4	即停止後サーボOFF
ServoOffAfterDecelStop	5	減速停止後サーボOFF

MC_STOP_CURVE_MODE

列挙子	設定値	内容
OverrideCurve	1	減速カーブ再作成

MC_POS_SOURCE

列挙子	設定値	内容
Invalid	-1	無効
SetPosition	1	指令現在位置
FeedMachinePosition	3	送り機械位置

MC_ACC_ZERO_MODE

列挙子	設定値	内容
ACCError	-1	エラー (始動しない)
MaximumAcceleration	1	最大加減速

MC_SIGNAL_LOGIC

列挙子	設定値	内容
HighLevel	0	TRUE時検出
LowLevel	1	FALSE時検出
RisingEdge	2	FALSE→TRUE(立上り)時検出
FallingEdge	3	TRUE→FALSE(立下り)時検出
BothEdges	4	立上り／立下り時検出

MC_ERROR_CLASS

列挙子	設定値	内容
_None	0	なし
Warning	1	警告
MinorError	2	軽度異常
ModerateError	3	中度異常
MajorError	4	重度異常

MC_EXECUTION_STATE

列挙子	設定値	内容
Ready	0	実行要求待ち
Executing	1	実行中
Done	2	実行完了
Error_	-1	エラー発生

MC_POS_RESTORATION_STATUS

列挙子	設定値	内容
NotExecute	0	未実施
WaitingRequest	1	復元要求待ち
RestoredInIncSystem	2	インクリメンタルシステムで復元完了
RestoredInAbsSystemUnHomed	3	絶対位置システムで復元完了(原点復帰未完)
RestoredInAbsSystem	4	絶対位置システムで復元完了

MC_AXES_GROUP_TYPE

列挙子	設定値	内容
Standard	0	標準

MC_OPERATION_PATTERN

列挙子	設定値	内容
PositioningComplete	0	位置決め終了
ContinuousPositioning	1	連続位置決め
ContinuousBlendingLow	2	連続軌跡(BlendingLow)
ContinuousBlendingPrevious	3	連続軌跡(BlendingPrevious)
ContinuousBlendingNext	4	連続軌跡(BlendingNext)
ContinuousBlendingHigh	5	連続軌跡(BlendingHigh)

MC_CONTROL_METHOD

列挙子	設定値	内容
NOP	128	NOP
JUMP	130	JUMP
LOOP	131	LOOP
LEND	132	LEND
LinearAbsolute	257	絶対値直線補間
LinearRelative	258	相対値直線補間
CircularAbsolute	259	絶対値円弧補間
CircularRelative	260	相対値円弧補間
HelicalAbsolute	261	絶対値ヘリカル補間
HelicalRelative	262	相対値ヘリカル補間

MC_MCODE_OUTPUT_OVERRIDE

列挙子	設定値	内容
UseFbOptions	0	FBオプションを使用
WithMode	1	WITHモード
AfterMode	2	AFTERモード

MC_STEP_MODE

列挙子	設定値	内容
Ignored	0	無視
StepPerDec	1	減速単位ステップ
StepPerPositioningData	2	位置決めデータ単位ステップ

MC_POS_RESTORATION_BASE

列挙子	設定値	内容
BackupPosition	0	バックアップ位置
HomePosition	1	原点
Auto	-1	自動判定

MC_POS_RESTORATION_REQUEST

列挙子	設定値	内容
NoRequest	0	無処理
RestorationRequest	1	現在位置復元実行
RestorationRequest_EraseAbsPos	2	絶対位置消失状態で現在位置復元実行

MC_INPUT_DIRECTION

列挙子	設定値	内容
NoDirectionRestriction	0	移動方向制限なし
mcPositiveDirection	1	現在位置が増加方向のみ許可
mcNegativeDirection	2	現在位置が減少方向のみ許可

MC_CLUTCH_METHOD

列挙子	設定値	内容
NoClutch	0	無効
ClutchCommand	1	クラッチ指令
ClutchLeadingEdge	2	クラッチ指令立上り
ClutchTrailingEdge	3	クラッチ指令立下り
ClutchAddress	4	アドレスモード
ClutchSignal	15	入出力データ指定

MC_CLUTCH_REFERENCE

列挙子	設定値	内容
GearFrontPosition	0	ギア前現在位置
GearPositionPerCycle	1	ギア後1サイクル現在位置

MC_CLUTCH_SMOOTHING_METHOD

列挙子	設定値	内容
ClutchSmoothingDisabled	0	ダイレクト
TimeConstantExponent	1	時定数方式(指数)
TimeConstantLinear	2	時定数方式(直線)
SlippageExponent	3	滑り量方式(指数)
SlippageLinear	4	滑り量方式(直線)
SlippageLinearFollow	5	滑り量方式(直線: 入力量追従)

MC_LENGTH_PER_CYCLE_CHANGE

列挙子	設定値	内容
LengthChangeInvalid	0	無効
LengthChangeValid	1	有効

MC_SYNC_CHANGE_COMMAND

列挙子	設定値	内容
ReferenceSetPositionMovement	0	カム基準位置移動
ChangeCurrentPositionPerCycle	1	1サイクル現在位置変更
ChangeMasterGearPositionPerCycle	2	主軸ギア後1サイクル現在位置変更
ChangeAuxGearPositionPerCycle	3	補助軸ギア後1サイクル現在位置変更
PositionPerCycleMovement	4	1サイクル現在位置移動

MC_GEAR_RESTORE_METHOD

列挙子	設定値	内容
PreviousPosition	0	前回値
InitialGearPositionPerCycle	1	ギア後1サイクル現在位置初期設定値
CalculateFromInputAxis	2	入力軸から計算

MC_CAM_RESTORE_METHOD

列挙子	設定値	内容
RestorePositionPerCycle	0	1サイクル現在位置復元
RestoreReferenceSetPosition	1	カム基準位置復元
RestoreCamSetPosition	2	カム指令現在位置復元

MC_CAM_REFERENCE_METHOD

列挙子	設定値	内容
PreviousPosition	0	前回値
InitialReferenceSetPosition	1	カム基準位置初期設定値
CamSetPosition	2	指令現在位置

MC_CAM_CYCLE_METHOD

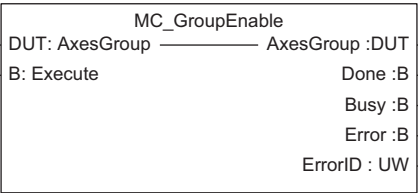
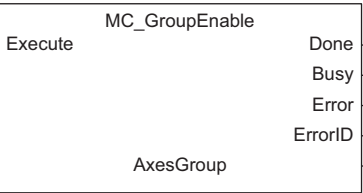
列挙子	設定値	内容
PreviousPosition	0	前回値
InitialPositionPerCycle	1	1サイクル現在位置初期設定値
MasterGearPositionPerCycle	2	主軸ギア後1サイクル現在位置
AuxGearPositionPerCycle	3	補助軸ギア後1サイクル現在位置

41 管理系のファンクションブロック

41.1 軸グループ有効

MC_GroupEnable

指定した軸グループの状態(AxesGroupName.Md.GroupStatus)を「0: 軸グループ無効(GroupDisabled)」から「4: 待機中(GroupStandby)」に遷移し、軸グループを有効にします。

ラダー	FBD/LD	ST
		<pre>MC_GroupEnable(AxesGroup:= ?AXES_GROUP_REF?, Execute:= ?BOOL?, Done=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
AxesGroup	軸グループ情報	AXES_GROUP_REF	起動時	—	省略不可	軸グループを設定します。 使用する変数(AxesGroupName.AxesGroupRef.)については、下記を参照してください。 📖 1402ページ AxesGroupName.AxesGroupRef.(軸グループ情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_GroupEnable(軸グループ有効)を実行します。

■出力変数

出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	「4: 待機中(GroupStandby)」状態に遷移したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MC_GroupEnable(軸グループ有効)を実行中は、TRUEになります。 軸グループ状態(AxesGroupName.Md.GroupStatus)が「4: 待機中(GroupStandby)」状態に遷移したときに、FALSEとなります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 📖 使用するコントローラのユーザーズマニュアル

機能

- AXES_GROUP_REF構造体の軸グループ状態(AxesGroupName.Md.GroupStatus)を「0: 軸グループ無効(GroupDisabled)」から「4: 待機中(GroupStandby)」に遷移します。
- 実行指令(Execute)がTRUEになるとMC_GroupEnable(軸グループ有効)を実行し、正常に処理を開始すると実行中(Busy)がTRUEになります。
- 処理が正常に完了すると、実行完了(Done)がTRUEになり、実行中(Busy)がFALSEになります。
- MC_GroupEnable(軸グループ有効)内で異常が発生した場合、エラー (Error)がTRUEになり、エラーコード(ErrorID)にエラーコードを格納します。エラーコードの詳細は、下記を参照してください。

使用するコントローラのユーザーズマニュアル

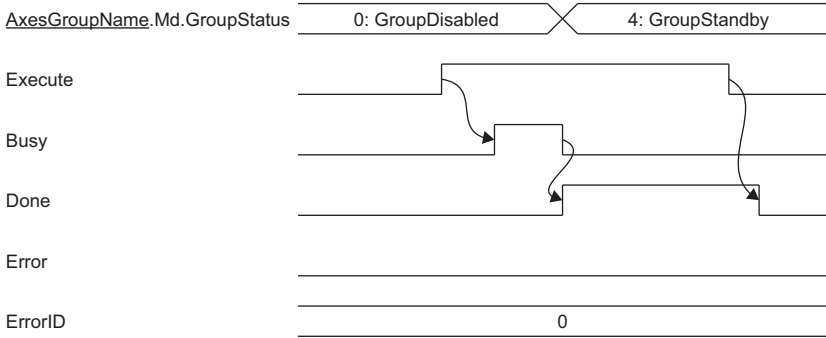
- 軸グループの状態遷移については、下記を参照してください。

使用するコントローラのユーザーズマニュアル

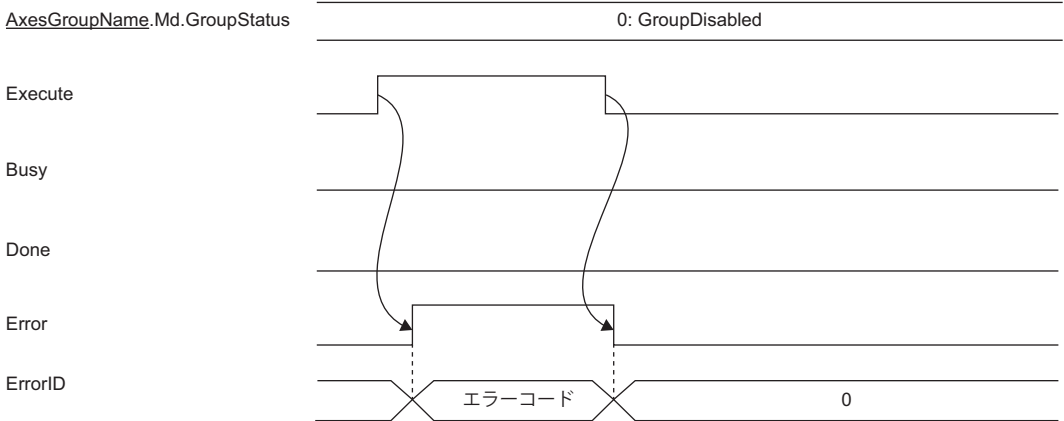
- MC_GroupEnable(軸グループ有効)は、すべての構成軸の軸状態(AxisName.Md.AxisStatus)が「4: 待機中(Standstill)」, または「0: 軸無効(Disabled)」状態の場合のみ実行できます。
- 軸グループ状態(AxesGroupName.Md.GroupStatus)が「4: 待機中(GroupStandby)」となったとき、構成軸の軸グループ使用中(AxisName.Md.UseInGroup)がTRUEになります。
- 軸グループ使用中(AxisName.Md.UseInGroup)がTRUEの構成軸を含む他の軸グループを設定してMC_GroupEnable(軸グループ有効)を実行した場合、軸グループ構成軸使用中(エラーコード: 1BB8H)となります。
- 構成軸のいずれかでエラーが発生した場合は、軸グループ状態(AxesGroupName.Md.GroupStatus)が「1: エラー停止中(GroupErrorStop)」になります。
- すでに軸グループ有効となっている軸グループにMC_GroupEnable(軸グループ有効)を実行した場合、実行完了(Done)がTRUEになり終了します。

■タイミングチャート

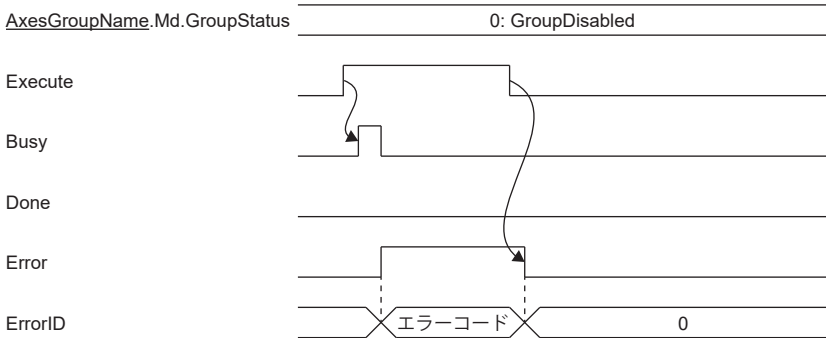
・ 正常完了の場合



・ 入出力変数異常の場合



・ 異常完了の場合



注意事項

- MC_GroupEnable(軸グループ有効)を実行する前に、AXES_GROUP_REF構造体の軸グループNo.(GroupNo)をあらかじめ設定しておく必要があります。
- 軸グループに設定したすべての構成軸の軸状態(AxisName.Md.AxisStatus)が「4: 待機中(Standstill)」, または「0: 軸無効(Disabled)」の場合のみ実行できます。
- 軸グループ変数初期化時にパラメータ異常が発生した場合、パラメータ範囲外(軸グループ)(エラーコード: 1A61H)となります。このとき、軸グループ状態(AxesGroupName.Md.GroupStatus)が「-1: 軸グループ変数未初期化/軸グループパラメータ異常(Invalid)」となります。「-1: 軸グループ変数未初期化/軸グループパラメータ異常(Invalid)」となった軸グループは、モニタデータなどのリフレッシュは実行されません。

プログラム例

軸グループ有効指令(bGroupEnable)をTRUEにし、軸グループ1(AxesGroup001)を有効にするプログラム例を下記に示します。

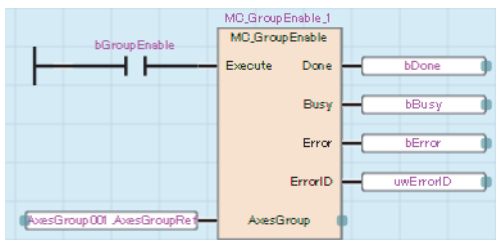
■軸グループ

軸グループNo.	ラベル名	データ型	コメント
1	AxesGroup001	AXES_GROUP_REF	軸グループ1

■使用するラベル

ラベル名	データ型	コメント
MC_GroupEnable_1	MC_GroupEnable	軸グループ有効FB
bGroupEnable	ビット	軸グループ有効指令
bDone	ビット	実行完了
bBusy	ビット	実行中
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム



■STプログラム

```
MC_GroupEnable_1(
  AxesGroup:= AxesGroup001.AxesGroupRef ,
  Execute:= bGroupEnable ,
  Done=> bDone ,
  Busy=> bBusy ,
  Error=> bError ,
  ErrorID=> uwErrorID
);
```

Point

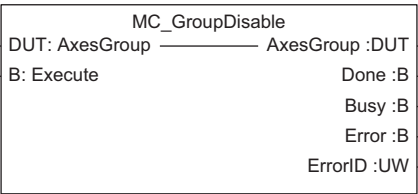
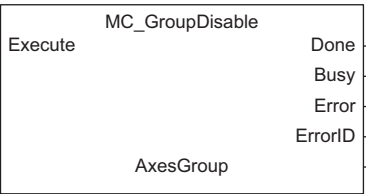
MC_GroupEnable(軸グループ有効)は、補間制御を行うモーション制御FBと組み合わせて使用します。補間制御とのプログラム例は、下記を参照してください。

- 1643ページ 絶対値直線補間制御
- 1658ページ 相対値直線補間制御
- 1671ページ 絶対値円弧補間制御
- 1688ページ 相対値円弧補間制御

41.2 軸グループ無効

MC_GroupDisable

指定した軸グループの状態(AxesGroupName.Md.GroupStatus)を「0: 軸グループ無効(GroupDisabled)」に遷移し、軸グループを無効にします。

ラダー	FBD/LD	ST
		<pre>MC_GroupDisable(AxesGroup:= ?AXES_GROUP_REF?, Execute:= ?BOOL?, Done=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
AxesGroup	軸グループ情報	AXES_GROUP_REF	起動時	—	省略不可	軸グループを設定します。 使用する変数(AxesGroupName.AxesGroupRef.)については、下記を参照してください。 1402ページ AxesGroupName.AxesGroupRef.(軸グループ情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_GroupDisable(軸グループ無効)を実行します。

■出力変数

出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	「0: 軸グループ無効(GroupDisabled)」状態に遷移したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MC_GroupDisable(軸グループ無効)を実行中は、TRUEになります。 軸グループ状態(AxesGroupName.Md.GroupStatus)が「0: 軸グループ無効(GroupDisabled)」状態に遷移したときに、FALSEとなります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 使用するコントローラのユーザーズマニュアル

機能

- AXES_GROUP_REF構造体の軸グループ状態(AxesGroupName.Md.GroupStatus)を「0: 軸グループ無効(GroupDisabled)」に遷移します。
- 実行指令(Execute)がTRUEでMC_GroupDisable(軸グループ無効)を実行し、正常に処理を開始すると実行中(Busy)がTRUEになります。
- 処理が正常に完了すると、実行完了(Done)がTRUEになり実行中(Busy)がFALSEになります。
- MC_GroupDisable(軸グループ無効)内で異常が発生した場合、エラー(Error)がTRUEになり、エラーコード(ErrorID)にエラーコードを格納します。エラーコードの詳細は、下記を参照してください。

使用するコントローラのユーザーズマニュアル

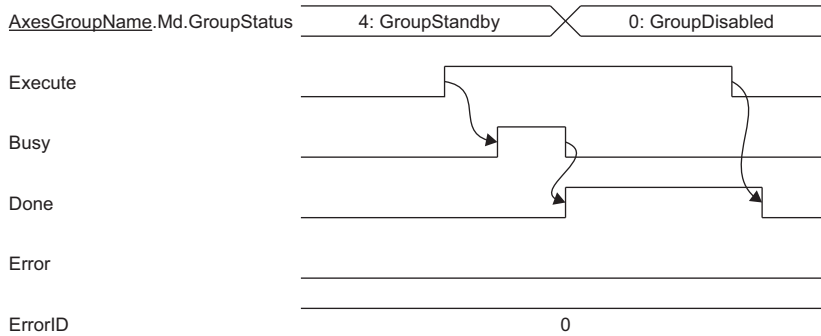
- 軸グループの状態遷移については、下記を参照してください。

使用するコントローラのユーザーズマニュアル

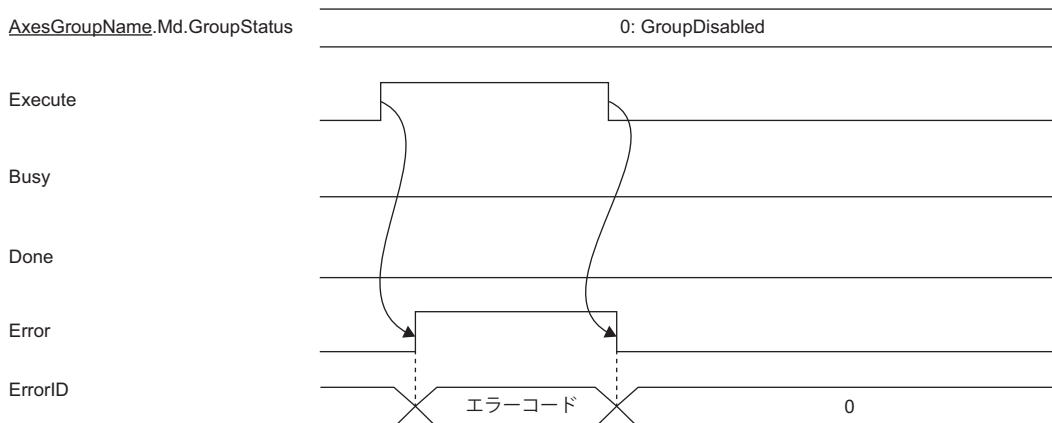
- MC_GroupDisable(軸グループ無効)は、軸グループ状態(AxesGroupName.Md.GroupStatus)が「4: 待機中(GroupStandby)」, または「1: エラー停止中(GroupErrorStop)」の場合のみ実行できます。軸グループ状態が「1: エラー停止中(GroupErrorStop)」で、軸グループが動作中にMC_GroupDisable(軸グループ無効)を実行した場合、停止完了後「0: 軸グループ無効(GroupDisabled)」へ遷移します。軸グループ状態(AxesGroupName.Md.GroupStatus)が「4: 待機中(GroupStandby)」, または「1: エラー停止中(GroupErrorStop)」以外でMC_GroupDisable(軸グループ無効)を実行した場合、軸グループ状態不正(軸グループ無効時)(エラーコード: 1AEFH)となり、実行中のFBは停止します。
- 軸グループ状態(AxesGroupName.Md.GroupStatus)が「0: 軸グループ無効(GroupDisabled)」になったとき、構成軸の軸グループ使用中(AxisName.Md.UseInGroup)がFALSEになります。
- 軸グループ無効となっている軸グループにMC_GroupDisable(軸グループ無効)を実行した場合、実行完了(Done)がTRUEになり終了します。

■タイミングチャート

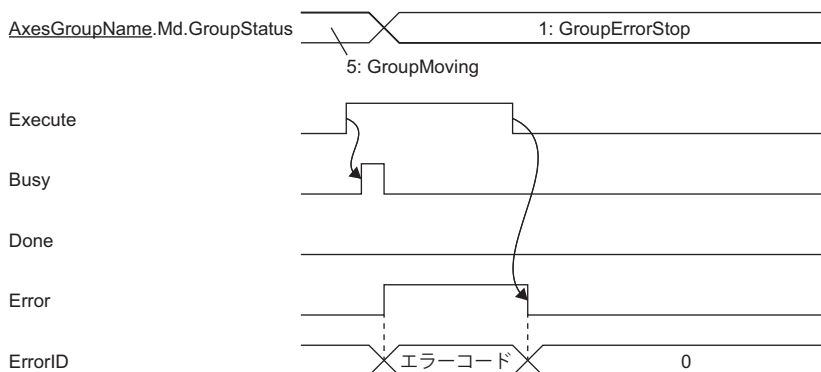
- 正常完了の場合



- 入出力変数異常の場合



- 異常完了の場合



注意事項

- MC_GroupDisable(軸グループ無効)を実行する前に、AXES_GROUP_REF構造体の軸グループNo.(GroupNo)をあらかじめ設定しておく必要があります。
- 軸グループ変数初期化時にパラメータ異常が発生した場合、パラメータ範囲外(軸グループ)(エラーコード: 1A61H)となります。このとき、軸グループ状態(AxesGroupName.Md.GroupStatus)が「-1: 軸グループ変数未初期化/軸グループパラメータ異常(Invalid)」となります。「-1: 軸グループ変数未初期化/軸グループパラメータ異常(Invalid)」となった軸グループは、モニターデータなどのリフレッシュは実行されません。

プログラム例

軸グループ無効指令(bGroupDisable)をTRUEにし、軸グループ1(AxesGroup001)を無効にするプログラム例を下記に示します。

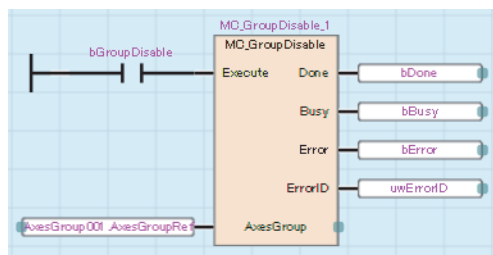
■軸グループ

軸グループNo.	ラベル名	データ型	コメント
1	AxesGroup001	AXES_GROUP_REF	軸グループ1

■使用するラベル

ラベル名	データ型	コメント
MC_GroupDisable_1	MC_GroupDisable	軸グループ無効FB
bGroupDisable	ビット	軸グループ無効指令
bDone	ビット	実行完了
bBusy	ビット	実行中
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム



■STプログラム

```
MC_GroupDisable_1(  
  AxesGroup:= AxesGroup001.AxesGroupRef ,  
  Execute:= bGroupDisable ,  
  Done=> bDone ,  
  Busy=> bBusy ,  
  Error=> bError ,  
  ErrorID=> uwErrorID  
);
```

Point

MC_GroupDisable(軸グループ無効)は、補間制御を行うモーション制御FBと組み合わせて使用します。補間制御とのプログラム例は、下記を参照してください。

- 1643ページ 絶対値直線補間制御
- 1658ページ 相対値直線補間制御
- 1671ページ 絶対値円弧補間制御
- 1688ページ 相対値円弧補間制御

41.3 運転可

41

MC_Power

指定した軸を運転可能状態に切り換えます。

ラダー	FBD/LD	ST
		<pre>MC_Power(Axis:= ?AXIS_REF?, Enable:= ?BOOL?, ServoON:= ?BOOL?, Status=> ?BOOL?, ReadyStatus=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	常時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 1378ページ AxisName.AxisRef.(軸情報) *: "シーケンサレディ (MotionSystem.Cd.SequenceReady)"をONに切換え後、初めてFBを呼び出したときに軸情報が確定します。

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Enable	有効	BOOL	常時	TRUE, FALSE	FALSE	TRUEにすると軸制御が有効となり、運転可能状態となります。 FALSEにすると軸制御が無効となり、運転可能状態を解除します。
ServoON	サーボON要求	BOOL	常時	TRUE, FALSE	FALSE	TRUEで軸のサーボONを要求します。

■出力変数

出力変数	名称	データ型	初期値	内容
Status	運転可	BOOL	FALSE	運転可能状態となったとき、TRUEになります。
ReadyStatus	レディ ON状態	BOOL	FALSE	ドライバレディがONすると、TRUEになります。
Busy	実行中	BOOL	FALSE	MC_Power(運転可)を実行したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 使用するコントローラのユーザーズマニュアル

機能

- 指定した軸の情報を初期化し、運転可能状態に切り換えます。
- 有効(Enable)とサーボON要求(ServoON)の入力をTRUEにすると、指定した軸を運転可能状態に切り換えます。
- 処理を開始すると実行中(Busy)がTRUEになります。
- ドライブユニットが運転可能状態になると運転可(Status)がTRUEになります。
- MC_Power(運転可)内で異常が発生した場合、エラー (Error)がTRUEになり、エラーコード(ErrorID)にエラーコードを格納します。エラーコードの詳細は、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

- AXIS_MONI構造体の軸状態(AxisName.Md.AxisStatus)は、「0: 軸無効(Disabled)」から「4: 待機中(Standstill)」の状態に移します。軸の状態遷移については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

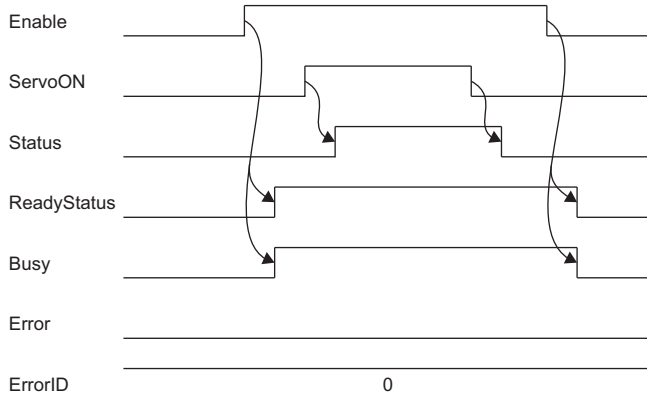
- 有効(Enable)とサーボON要求(ServoON)の入力により、軸のサーボON/OFF状態、ドライバ状態は下記のように切り換えます。

入力変数		出力変数		サーボON/OFF状態	ドライバ状態 (AxisName.Md.Driver_State)
有効(Enable)	サーボON要求 (ServoON)	レディ ON状態 (ReadyStatus)	運転可(Status)		
TRUE	TRUE	TRUE	TRUE	サーボON	6: Operation Enable (OperationEnable)
	FALSE	TRUE	FALSE	サーボOFF	5: Switched On (SwitchedOn)
FALSE	TRUE	FALSE	FALSE	サーボOFF	3: Switched On Disabled (SwitchOnDisabled)
	FALSE	FALSE	FALSE	サーボOFF	3: Switched On Disabled (SwitchOnDisabled)

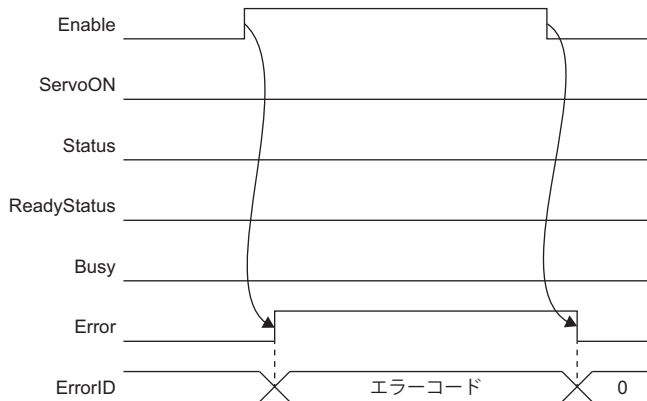
- サーボOFF中に実軸が外力で回った場合は、フォローアップ処理を行います。
- サーボON/OFFの制御は、制御モードに関係なく操作できます。サーボOFF時の制御モードは、ドライバの仕様に依存します。
- ドライブユニットエラー発生中、MC_Power(運転可)はドライバに送信しているため、再度有効(Enable)とサーボON要求(ServoON)をFALSEからTRUEにする必要はありません。

■タイミングチャート

・正常完了の場合



・異常完了の場合



注意事項

- MC_Power(運転可)を実行する前に、AXIS_REF構造体の軸No.(AxisNo)をあらかじめ設定しておく必要があります。
- MC_Power(運転可)は、"シーケンサレディ (MotionSystem.Cd.SequenceReady)"をONに切換え後、初めてFBを呼び出したときに軸情報が確定します。有効(Enable)がFALSEのときに軸情報を変更しても反映しません。
- 同じ軸に対して、MC_Power(運転可)を2つ以上配置しないでください。2つ以上配置した場合の動作は保証しません。
- 内蔵モーション準備完了(X20)がOFF中のときは、プログラム停止時のサーボON/OFF状態を維持します。

プログラム例

サーボON要求指令(bServoON)をTRUEにし，軸1(Axis0001)を運転可能状態とするプログラム例を下記に示します。

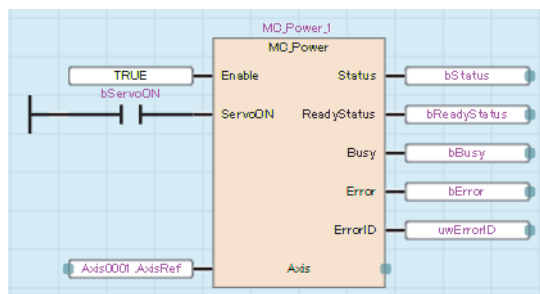
■軸

軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■使用するラベル

ラベル名	データ型	コメント
MC_Power_1	MC_Power	運転可FB
bServoON	ビット	サーボON要求指令
bStatus	ビット	運転可
bReadyStatus	ビット	レディ ON状態
bBusy	ビット	実行中
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム



■STプログラム

```

MC_Power_1(
  Axis:= Axis0001.AxisRef ,
  Enable:= TRUE ,
  ServoON:= bServoON ,
  Status=> bStatus ,
  ReadyStatus=> bReadyStatus ,
  Busy=> bBusy ,
  Error=> bError ,
  ErrorID=> uwErrorID
);

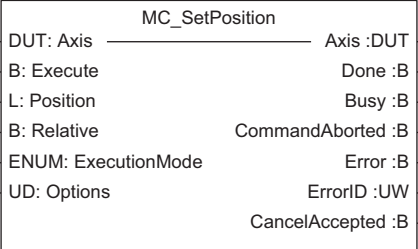
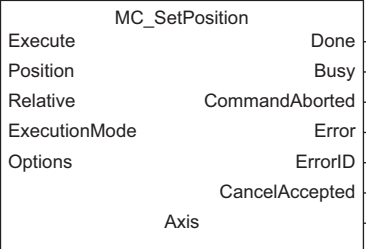
```


41.4 現在位置変更

41

MC_SetPosition

指定した軸の現在位置(指令位置, フィードバック位置)を変更します。

ラダー	FBD/LD	ST
		<pre>MC_SetPosition(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, Position:= ?LREAL?, Relative:= ?BOOL?, ExecutionMode:= ?INT?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?, CancelAccepted=> ?BOOL?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName, AxisRef.)については、下記を参照してください。 📖 1378ページ AxisName, AxisRef. (軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_SetPosition(現在位置変更)を実行します。
Position	目標位置	LREAL	起動時	-100000000000.0~100000000000.0	0.0	変更する目標位置の値を設定します。 相対位置選択(Relative)の設定により設定値が異なります。 詳細は、下記を参照してください。 📖 1469ページ 目標位置(Position)
Relative	相対位置選択	BOOL	起動時	TRUE, FALSE	FALSE	相対距離により現在位置を変更するか、絶対位置により現在位置を変更するかを設定します。 • FALSE: 絶対位置による現在位置変更 • TRUE: 相対距離による現在位置変更
ExecutionMode	起動モード	INT (MC_EXECUTION_MODE)	起動時	1, 3	3	MC_SetPosition(現在位置変更)を実行するための実行方法を設定します。 • 1: 完了待ちして実行(mcQueued)*1 • 3: 投機的に実行(mcSpeculatively)*2 詳細は、下記を参照してください。 📖 1469ページ 起動モード(ExecutionMode)
Options	オプション	DWORD(HEX)	起動時	00000000H~00010000H	00000000H	MC_SetPosition(現在位置変更)の機能オプションをビット指定で設定します。 詳細は、下記を参照してください。 📖 1471ページ オプション(Options)

*1 前のFBが動作完了してから実行します。

*2 指定軸で実行中のFBがない場合のみ実行できます。

■出力変数

出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	現在位置変更が完了したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MC_SetPosition(現在位置変更)を実行したときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	MC_SetPosition(現在位置変更)の実行が中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。エラーコードの詳細は、下記を参照してください。 📖使用するコントローラのユーザーズマニュアル
CancelAccepted	キャンセル受付	BOOL	FALSE	MC_SetPosition(現在位置変更)がキャンセルを受け取ったときにTRUEになります。

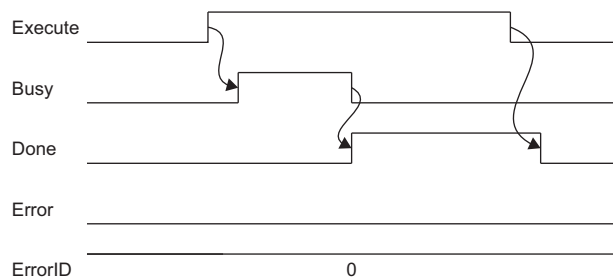
機能

- 指定した軸の現在位置を変更します。
- 相対位置選択(Relative)がTRUEの場合、現在位置に目標位置(相対距離)を加算した位置に変更します。
- 相対位置選択(Relative)がFALSEの場合、目標位置(絶対位置)に変更します。
- 実行指令(Execute)がTRUEになるとMC_SetPosition(現在位置変更)を実行し、正常に処理を開始すると実行中(Busy)がTRUEになります。
- 処理が完了し、現在位置を変更すると実行完了(Done)がTRUEになります。
- 起動モード(ExecutionMode)に「1: 完了待ちして実行(mcQueued)」を設定し、MC_SetPosition(現在位置変更)を始動後、処理が完了する前に軸状態(AxisName.Md.AxisStatus)が「0: 軸無効(Disabled)」, または「1: エラー停止中(ErrorStop)」となった場合、MC_SetPosition命令異常(エラーコード: 1B9CH)となります。
- MC_SetPosition(現在位置変更)内で異常が発生した場合、エラー (Error)がTRUEになり、エラーコード(ErrorID)にエラーコードを格納します。エラーコードの詳細は、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

■タイミングチャート

- 正常完了の場合



- 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■目標位置(Position)

目標位置(Position)で設定可能な値は、相対位置選択(Relative)がFALSEかTRUEかにより異なります。
目標位置(Position)は、下記範囲内で設定してください。
ただし、指定範囲内でも、現在位置変更後の指令現在位置がソフトウェアストロークリミット範囲外となる場合は、エラーとなり実行されません。

相対位置選択(Relative)	目標位置(Position)の設定範囲
FALSE(絶対位置による現在位置変更)	■リングカウンタ無効時 -100000000000.0 ≤ 設定値 < 100000000000.0 ■リングカウンタ有効時 リングカウンタ下限値 ≤ 設定値 < リングカウンタ上限値
TRUE(相対距離による現在位置変更)	■リングカウンタ無効時 -100000000000.0 ≤ 設定値 ≤ 100000000000.0 ■リングカウンタ有効時 -(リングカウンタ上限値 - リングカウンタ下限値) / 2 ≤ 設定値 ≤ (リングカウンタ上限値 - リングカウンタ下限値) / 2

■起動モード(ExecutionMode)

軸動作中にMC_SetPosition(現在位置変更)を実行した場合、変更するタイミングは設定した起動モード(ExecutionMode)により異なります。起動モード(ExecutionMode)のタイミングについて下記に示します。

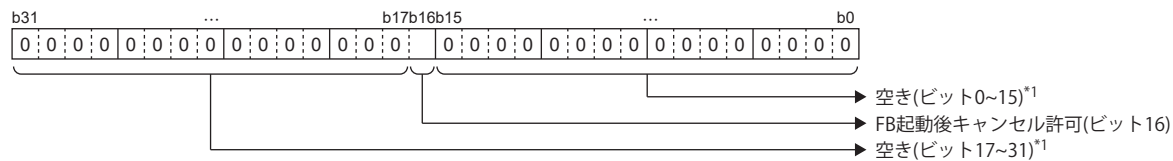
設定値	内容
1: 完了待ちして実行 (mcQueued)	実行中のFBが終了した後に実行します。 実行指令(Execute)の立上り検出により、実行中(Busy)がTRUEとなり、軸状態(AxisName.Md.AxisStatus)が「4: 待機中(Standstill)」となるまで待機します。 軸状態(AxisName.Md.AxisStatus)が「4: 待機中(Standstill)」になると、現在位置変更動作を開始します。 The diagram illustrates the timing of the position change operation. It shows the execution of MC_MoveAbsolute and MC_SetPosition. MC_MoveAbsolute starts at Position 100.0. When its Execute signal is triggered, it begins moving. The AxisName.Md.AxisStatus transitions from 5: DiscreteMotion to 4: Standstill. MC_SetPosition is called during the move. Its Execute signal is triggered, and it begins moving to the new position. The diagram shows the sequence of events from the start of the move to the completion of the position change.

設定値	内容
3: 投機的に実行 (mcSpeculatively)	実行指令(Execute)の立上り検出時に該当軸の軸状態(AxisName.Md.AxisStatus)が「4: 待機中(Standstill)」の場合に、現在位置変更が実行できます。 該当軸の軸状態(AxisName.Md.AxisStatus)が「4: 待機中(Standstill)」ではない場合、または実行中のFBがある場合は、MC_SetPosition命令異常(エラーコード: 1B9CH)となり、現在位置変更は実行しません。 AxisName.Md.AxisStatus 4: Standstill MC_SetPosition Position 190.0 ExecutionMode 3: mcSpeculatively Relative Execute Done Busy 現在位置変更後アドレス 指令現在位置

■オプション(Options)

MC_SetPosition(現在位置変更)で使用する機能オプションをビット指定で設定します。

ビット指定で設定する内容を下記に示します。



*1 空には「0」を設定してください。「0」以外を設定した場合、MC_SetPosition命令異常(エラーコード: 1B9CH)となります。

ビット	名称	内容
16	FB起動後キャンセル許可	MC_SetPosition(現在位置変更)起動後にキャンセルを許可するか、許可しないかを設定します。 ・ 0: 許可しない ・ 1: 許可する

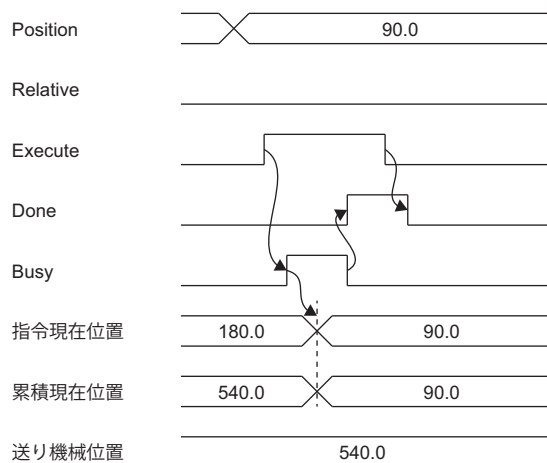
■動作概要

- 軸が停止状態でMC_SetPosition(現在位置変更)を実行した場合、実行指令(Execute)の立上り検出により、現在位置変更動作を開始します。現在位置変更では、指令現在位置、累積現在位置を目標位置(Position)で設定した位置に変更します。目標位置(Position)は、相対位置選択(Relative)がTRUEの場合は「相対距離による現在位置変更」、FALSEの場合は「絶対位置による現在位置変更」となります。

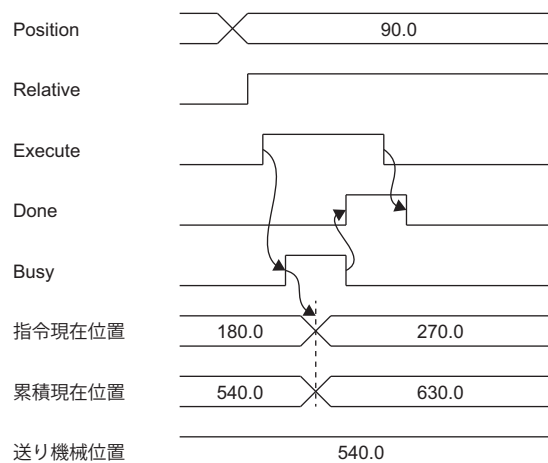
例

リングカウンタ下限値「0.0」、リングカウンタ上限値「360.0」、指令現在位置「180.0」(累積現在位置「540.0」)で停止時、「90.0」への現在位置変更を実行した場合

<相対位置選択(Relative)がFALSE(絶対位置)の場合>



<相対位置選択(Relative)がTRUE(相対距離)の場合>



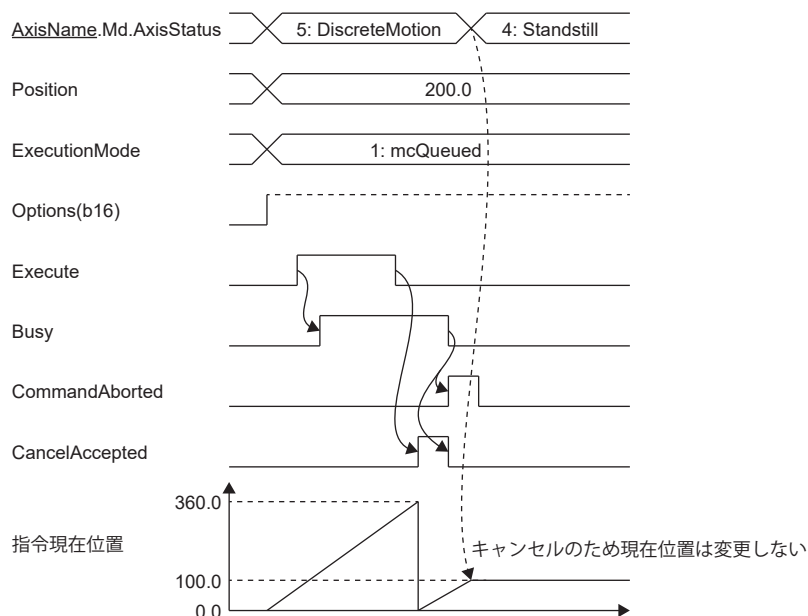
■キャンセル

MC_SetPosition(現在位置変更)実行後に待機状態となっている現在位置変更をキャンセルできます。

- キャンセルを有効にするには、「FB起動後キャンセル許可(オプション(Options): ビット16)」を「1: 許可する」に設定した状態でMC_SetPosition(現在位置変更)を実行します。
- 実行指令(Execute)の立下り検出により、キャンセルを開始します。
- キャンセルの受け付けは、出力ピンの実行中(Busy)がTRUEのときのみ行います。
- MC_SetPosition(現在位置変更)がキャンセルを受け取ると出力ピンのキャンセル受付(CancelAccepted)がTRUEになります。
- キャンセルが完了すると出力ピンの実行中断(CommandAborted)がTRUEになります。
- キャンセルした場合は、指令現在位置を変更しません。

例

起動モード(ExecutionMode)を「1: 完了待ちして実行(mcQueued)」で起動後、軸状態(AxisName.Md.AxisStatus)が「4: 待機中(Standstill)」への切り替わり待機中にキャンセルした場合



注意事項

- 軸動作中に現在位置を変更した場合、実行中のMC_SetPosition(現在位置変更)の目標位置は変化しませんが、現在位置を変更するため目標位値までの動作が変更されます。

プログラム例

現在値変更指令(bSetPosition)をTRUEにし、軸1(Axis0001)の現在値を「100.0」に変更するプログラム例を下記に示します。

■軸

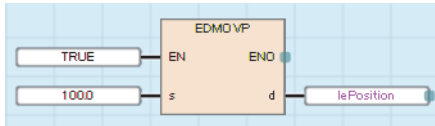
軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■使用するラベル

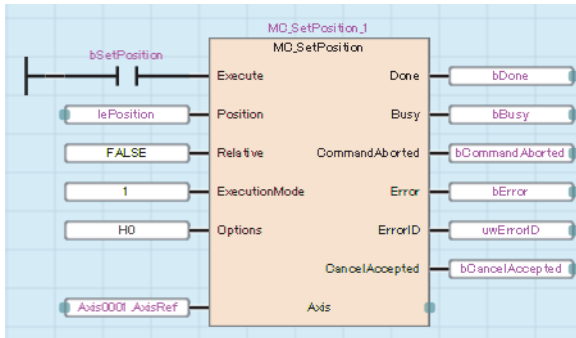
ラベル名	データ型	コメント
MC_SetPosition_1	MC_SetPosition	現在位置変更FB
bSetPosition	ビット	現在位置変更指令
lePosition	倍精度実数	目標位置
bDone	ビット	準備完了
bBusy	ビット	実行中
bCommandAborted	ビット	実行中断
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
bCancelAccepted	ビット	キャンセル受付

■FBD/LDプログラム

- 現在位置変更用データ設定



- 現在位置変更



■STプログラム

//-----現在位置変更用データ設定-----

lePosition:= 100.0;

//-----現在位置変更-----

```
MC_SetPosition_1(
  Axis:= Axis0001.AxisRef,
  Execute:= bSetPosition,
  Position:= lePosition,
  Relative:= FALSE,
  ExecutionMode:= MC_EXECUTION_MODE__mcQueued,
  Options:= H00000000,
  Done=> bDone,
  Busy=> bBusy,
  CommandAborted=> bCommandAborted,
  Error=> bError,
  ErrorID=> uwErrorID,
  CancelAccepted=> bCancelAccepted
);
```


41.5 トルク制限値

41

MCv_SetTorqueLimit

トルク制限値の変更を実行します。

ラダー	FBD/LD	ST
		<pre> MCv_SetTorqueLimit(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, PositiveValid:= ?BOOL?, PositiveValue:= ?LREAL?, NegativeValid:= ?BOOL?, NegativeValue:= ?LREAL?, ExecutionMode:= ?INT?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, CommandAborted=> ?BOOL?, CancelAccepted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?); </pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📖 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMCv_SetTorqueLimit(トルク制限値)を実行します。
PositiveValid	正方向トルク制限有効	BOOL	起動時/再起動可	TRUE, FALSE	FALSE	正方向のトルク制限設定値の変更を有効にするか、無効にするかを設定します。 • FALSE: 正方向のトルク制限を無効にします。 • TRUE: 正方向のトルク制限を有効にします。
PositiveValue	正方向トルク制限値	LREAL	起動時/再起動可	0.0~1000.0[%]	0.0	正方向へのトルク制限値の値を設定します。 正方向トルク制限有効(PositiveValid)がTRUEの場合に取り込みます。
NegativeValid	負方向トルク制限有効	BOOL	起動時/再起動可	TRUE, FALSE	FALSE	負方向のトルク制限設定値の変更を有効にするか、無効にするかを設定します。 • FALSE: 負方向のトルク制限を無効にします。 • TRUE: 負方向のトルク制限を有効にします。
NegativeValue	負方向トルク制限値	LREAL	起動時/再起動可	0.0~1000.0[%]	0.0	負方向へのトルク制限値の値を設定します。 負方向トルク制限有効(NegativeValid)がTRUEの場合に取り込みます。

入力変数	名称	データ型	取込	有効範囲	初期値	内容
ExecutionMode	起動モード	INT (MC_EXECUTION_MODE)	起動時	0, 3	0	MCv_SetTorqueLimit(トルク制限値)を実行するタイミングを設定します。 ・ 0: 直ちに実行(mclImmediately) ・ 3: 投機的に実行(mcSpeculatively) 詳細は、下記を参照してください。 📖 1478ページ 起動モード (ExecutionMode)
Options	オプション	DWORD(HEX)	起動時	00000000H~00010000H	00000000H	MCv_SetTorqueLimit(トルク制限値)の機能オプションをビット指定で設定します。 詳細は、下記を参照してください。 📖 1478ページ オプション(Options)

■出力変数

出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	トルク制限値変更が正常に完了したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MCv_SetTorqueLimit(トルク制限値)を実行したときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	MCv_SetTorqueLimit(トルク制限値)の実行が中断したときに、TRUEになります。
CancelAccepted	キャンセル受付	BOOL	FALSE	MCv_SetTorqueLimit(トルク制限値)がキャンセルを受け取ったときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 📖使用するコントローラのユーザーズマニュアル

機能

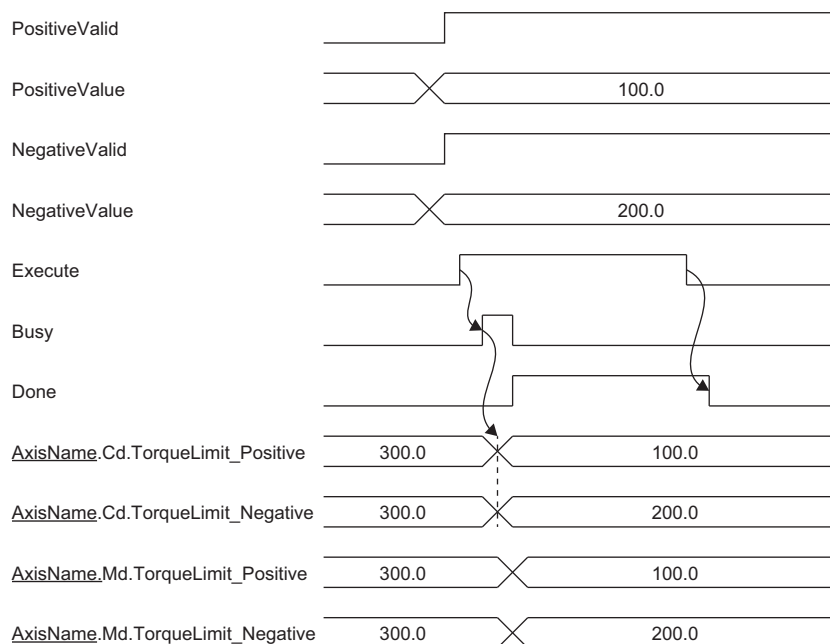
- 正方向トルク制限値, または負方向トルク制限値を変更します。
- 正方向トルク制限有効(PositiveValid)がTRUEの場合, 正方向トルク制限値を正方向トルク制限値(PositiveValue)の値に変更します。
- 負方向トルク制限有効(NegativeValid)がTRUEの場合, 負方向トルク制限値を負方向トルク制限値(NegativeValue)の値に変更します。
- 実行指令(Execute)がTRUEでMCv_SetTorqueLimit(トルク制限値)を実行し, 正常に処理を開始すると実行中(Busy)がTRUEになります。
- トルク制限値を変更するタイミングは, 起動モード(ExecutionMode)で設定します。
- 処理が完了し, トルク制限値を変更すると実行完了(Done)がTRUEになります。
- MCv_SetTorqueLimit(トルク制限値)内で異常が発生した場合, エラー (Error)がTRUEになり, エラーコード(ErrorID)にエラーコードを格納します。エラーコードの詳細は, 下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

- 正方向トルク制限有効(PositiveValid)と負方向トルク制限有効(NegativeValid)が共にFALSEの場合, トルク制限値の変更を行わず, 実行完了(Done)がTRUEになります。

■タイミングチャート

- 正常完了の場合



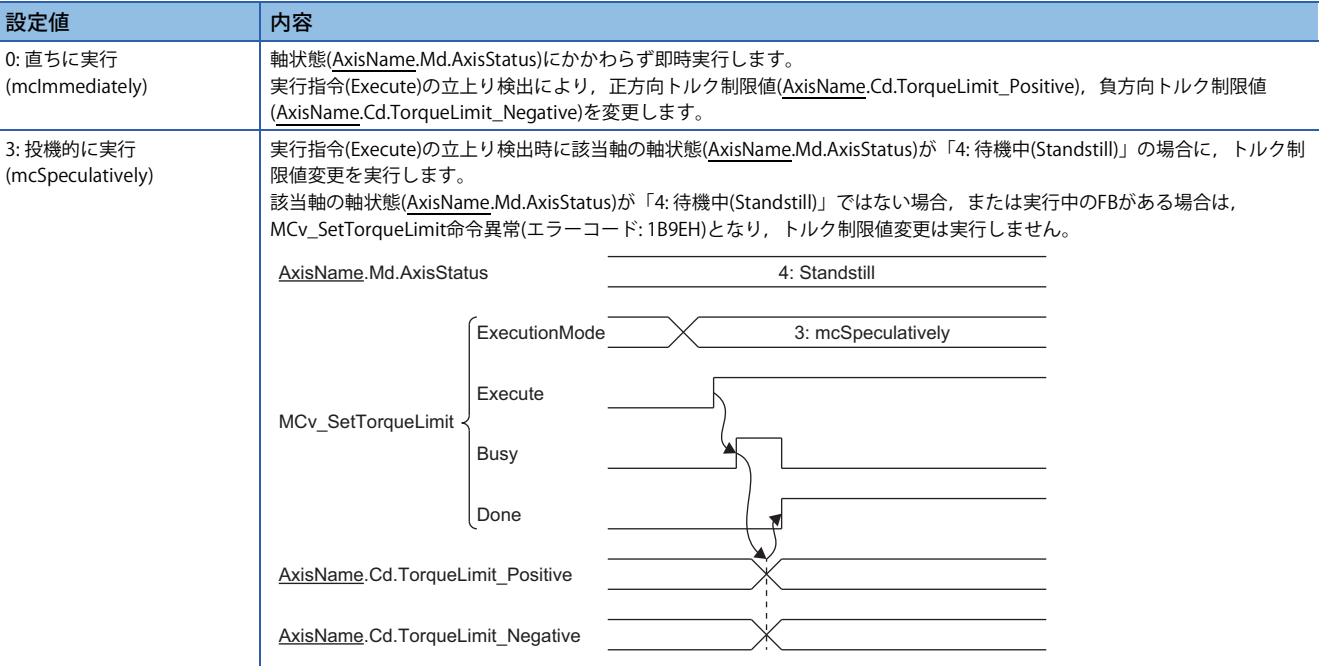
・ 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■起動モード(ExecutionMode)

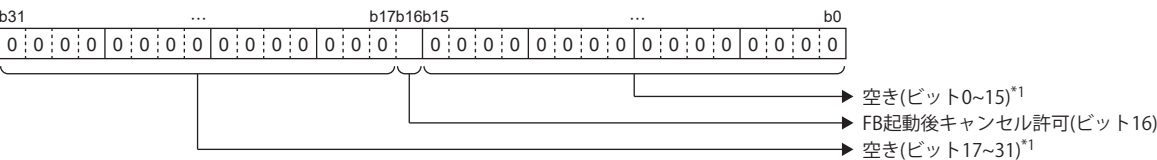
軸動作中にMCv_SetTorqueLimit(トルク制限値)を実行した場合、変更するタイミングは設定した起動モード(ExecutionMode)により異なります。起動モード(ExecutionMode)のタイミングについて下記に示します。



■オプション(Options)

MCv_SetTorqueLimit(トルク制限値)で使用するの機能オプションをビット指定で設定します。

ビット指定で設定する内容を下記に示します。

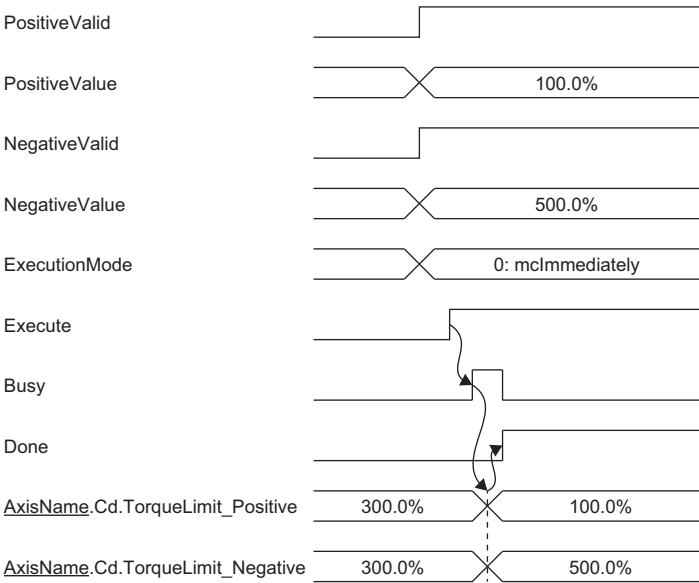


*1 空きには「0」を設定してください。「0」以外を設定した場合、MCv_SetTorqueLimit命令異常(エラーコード: 1B9EH)となります。

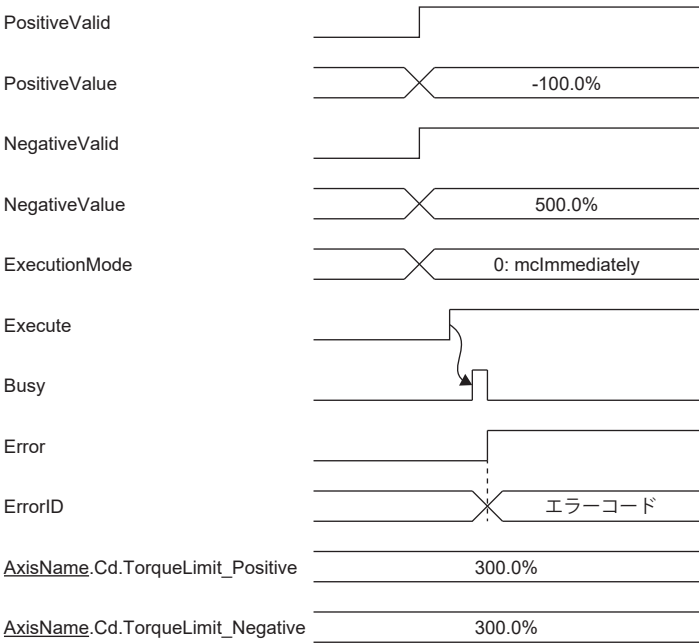
ビット	名称	内容
16	FB起動後キャンセル許可	MCv_SetTorqueLimit(トルク制限値)起動後にキャンセルを許可するか、許可しないかを設定します。 ・ 0: 許可しない ・ 1: 許可する

■動作概要

- MCv_SetTorqueLimit(トルク制限値)では、正方向トルク制限値(AxisName.Cd.TorqueLimit_Positive), 負方向トルク制限値(AxisName.Cd.TorqueLimit_Negative)を変更します。



- 正方向トルク制限値(PositiveValue), または負方向トルク制限値(NegativeValue)に範囲外の値を入力し, 実行指令(Execute)をTRUEにした場合, MCv_SetTorqueLimit(トルク制限値)がエラーとなり, 正方向トルク制限値(AxisName.Cd.TorqueLimit_Positive), 負方向トルク制限値(AxisName.Cd.TorqueLimit_Negative)は変更しません。



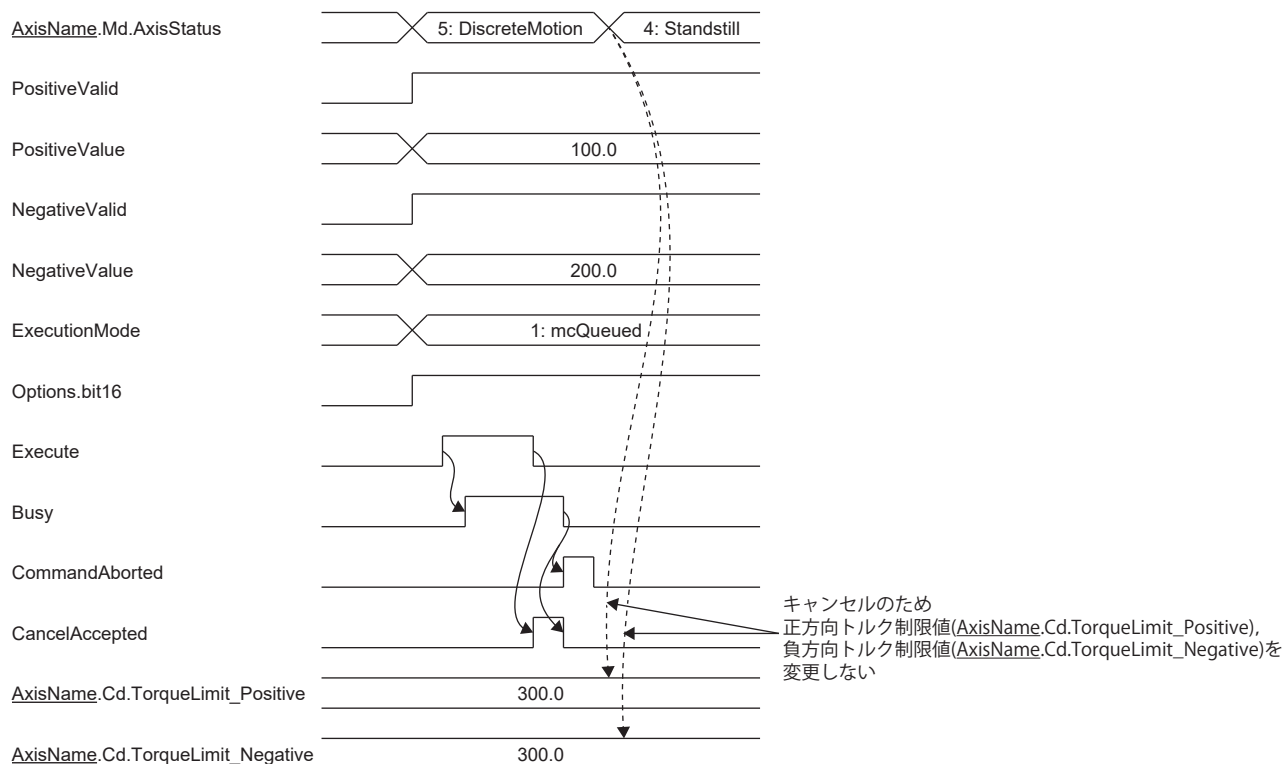
■キャンセル

MCv_SetTorqueLimit(トルク制限値)実行後に待機状態となっているトルク制限値変更をキャンセルできます。

- キャンセルを有効にするには、「FB起動後キャンセル許可(オプション(Options):ビット16)」を「1:許可する」に設定した状態で、MCv_SetTorqueLimit(トルク制限値)を実行します。
- 実行指令(Execute)の立下り検出により、キャンセルを開始します。
- キャンセルの受け付けは、出力ピンの実行中(Busy)がTRUEのときのみ行います。
- MCv_SetTorqueLimit(トルク制限値)がキャンセルを受け取ると出力ピンのキャンセル受付(CancelAccepted)がTRUEになります。
- キャンセルが完了すると出力ピンの実行中断(CommandAborted)がTRUEになります。
- キャンセルした場合は、正方向トルク制限値(AxisName.Cd.TorqueLimit_Positive)、負方向トルク制限値(AxisName.Cd.TorqueLimit_Negative)を変更しません。

例

起動モード(ExecutionMode)を「1:完了待ちして実行(mcQueued)」で起動後、軸状態(AxisName.Md.AxisStatus)が「4:待機中(Standstill)」への切り替わり待機中にキャンセルした場合



注意事項

- MCv_SetTorqueLimit(トルク制限値)では、正方向トルク制限値(AxisName.Cd.TorqueLimit_Positive)と負方向トルク制限値(AxisName.Cd.TorqueLimit_Negative)の変更を行います。

プログラム例

トルク制限値変更指令(bSetTorqueLimit)をTRUEにし、軸1(Axis0001)の正方向トルク制限値を「100.0」、負方向トルク制限値を「200.0」に変更するプログラム例を下記に示します。

■軸

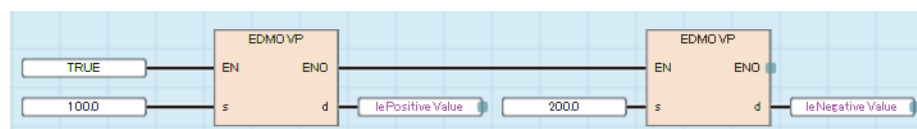
軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■使用するラベル

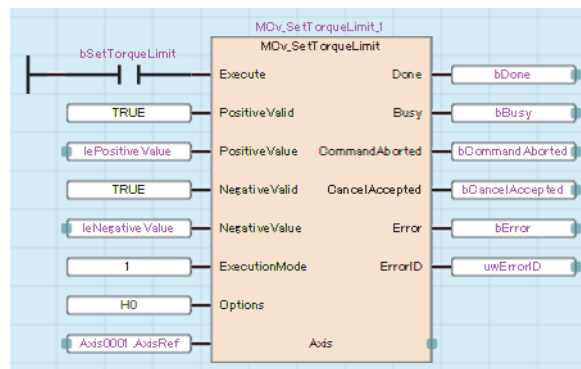
ラベル名	データ型	コメント
MCv_SetTorqueLimit_1	MCv_SetTorqueLimit	トルク制限値FB
bSetTorqueLimit	ビット	トルク制限値変更指令
lePositiveValue	倍精度実数	正方向トルク制限値
leNegativeValue	倍精度実数	負方向トルク制限値
bDone	ビット	準備完了
bBusy	ビット	実行中
bCommandAborted	ビット	実行中断
bCancelAccepted	ビット	キャンセル受付
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

- トルク制限値変更用データ設定



- トルク制限値変更



■STプログラム

//-----トルク制限値変更用データ設定-----

lePositiveValue:= 100.0;

leNegativeValue:= 200.0;

//-----トルク制限値変更-----

MCv_SetTorqueLimit_1(

Axis:= Axis0001.AxisRef ,

Execute:= bSetTorqueLimit ,

PositiveValid:= TRUE ,

PositiveValue:= lePositiveValue ,

NegativeValid:= TRUE ,

NegativeValue:= leNegativeValue ,

ExecutionMode:= MC_EXECUTION_MODE__mcQueued ,

Options:= H00000000 ,

Done=> bDone ,

Busy=> bBusy ,

CommandAborted=> bCommandAborted ,

CancelAccepted=> bCancelAccepted ,

Error=> bError ,

ErrorID=> uwErrorID

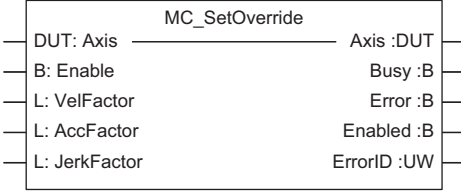
);

41.6 オーバライド値設定

41

MC_SetOverride

指定した軸の目標速度，目標加速度，目標減速度の変更を実行します。

ラダー	FBD/LD	ST
		<pre>MC_SetOverride(Axis:= ?AXIS_REF?, Enable:= ?BOOL?, VelFactor:= ?LREAL?, AccFactor:= ?LREAL?, JerkFactor:= ?LREAL?, Enabled=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Enable	有効	BOOL	常時	TRUE, FALSE	FALSE	TRUEでMC_SetOverride(オーバライド値設定)を実行します。
VelFactor	速度オーバライド係数	LREAL	常時	0.00~10.00	0.00	速度のオーバライド係数を設定します。 有効(Enable)がTRUEのときは、常時取り込みます。
AccFactor	加速度オーバライド係数	LREAL	常時	0.00, 0.01~10.00	0.00	加速度のオーバライド係数を設定します。 有効(Enable)がTRUEのときは、常時取り込みます。 「0.00」を設定した場合は、加速度オーバライド係数の変更を行わず、前回値のまま制御します。
JerkFactor	ジャークオーバライド係数	LREAL	常時	0.0	0.0	「0.0」を設定してください。 *: 「0.0」以外を設定した場合、ジャークオーバライド係数(JerkFactor)範囲外(エラーコード: 1BBEH)となります。

■出力変数

出力変数	名称	データ型	初期値	内容
Enabled	有効中	BOOL	FALSE	オーバライド値が正常に設定している場合に、TRUEになります。
Busy	実行中	BOOL	FALSE	MC_SetOverride(オーバライド値設定)を実行したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 使用するコントローラのユーザーズマニュアル

機能

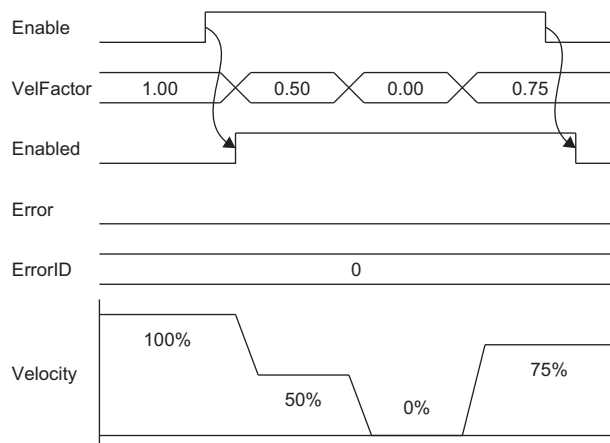
- ・指定した軸の目標速度，目標加速度，目標減速度を変更します。
- ・現在動作中の目標速度，目標加速度，目標減速度にオーバーライド係数を掛けた値に変更します。
- ・有効(Enable)がTRUEになるとMC_SetOverride(オーバーライド値設定)を実行します。オーバーライド係数が有効中のときは，有効中(Enabled)がTRUEになります。
- ・有効(Enable)がTRUEのときに，オーバーライド係数の値を変更すると，新たなオーバーライド係数を反映します。
- ・MC_SetOverride(オーバーライド値設定)内で異常が発生した場合，エラー (Error)がTRUEになり，エラーコード(ErrorID)にエラーコードを格納します。エラーコードの詳細は，下記を参照してください。

使用するコントローラのユーザーズマニュアル

- ・速度オーバーライド係数(VelFactor)に「0.00」の値を設定すると，軸状態(AxisName.Md.AxisStatus)が「4: 待機中 (Standstill)」に移行せずに軸を停止します。
- ・加速度オーバーライド係数(AccFactor)に「0.00」の値を設定すると，加速度オーバーライド係数を変更せず，前回の加速度オーバーライド係数を維持します。

■タイミングチャート

- ・正常完了の場合



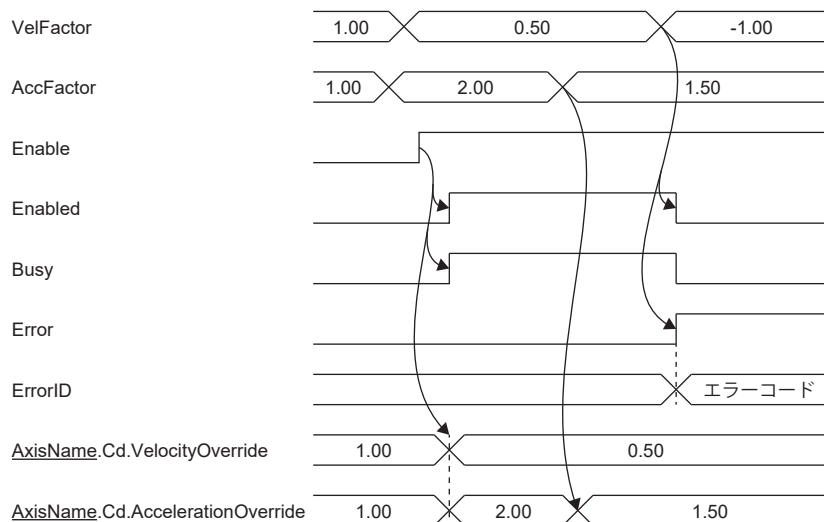
- ・異常完了の場合

異常完了時のタイミングチャートについては，下記を参照してください。

1370ページ 有効(Enable)型によるモーション制御FBの基本動作

■動作概要

- ・MC_SetOverride(オーバーライド値設定)では，速度オーバーライド係数(AxisName.Cd.VelocityOverride)，加速度オーバーライド係数(AxisName.Cd.AccelerationOverride)を変更します。



- ・オーバーライド係수에範圍外の値を設定した場合，MC_SetOverride(オーバーライド値設定)がエラーとなり以降の取込みを停止します。再度取込みを実行する場合は，有効(Enable)を再度立ち上げてください。

注意事項

- 同じ軸に対して、MC_SetOverride(オーバーライド値設定)を2つ以上配置しないでください。2つ以上配置した場合の動作は保証しません。
- 単軸制御では、軸制御データで設定したオーバーライド係数のみが影響します。
- MC_SetOverride(オーバーライド値設定)実行中に、速度オーバーライド係数(AxisName.Cd.VelocityOverride)、加速度オーバーライド係数(AxisName.Cd.AccelerationOverride)の直接操作を行わないでください。
- 速度オーバーライド後の速度が範囲外となる場合は、下記マニュアルの「速度範囲」を参照してください。

📖使用するコントローラのユーザーズマニュアル

- 加速度オーバーライド後の加速度、減速度、加速時間、減速時間が範囲外となる場合は、下記マニュアルの「加減速処理機能」を参照してください。

📖使用するコントローラのユーザーズマニュアル

プログラム例

オーバーライド値変更指令(bSetOverride)をTRUEにし、軸1(Axis0001)の目標速度、目標加速度、目標減速度に速度オーバーライド係数「1.0」、加減速度オーバーライド係数「2.0」を掛けた値に変更するプログラム例を下記に示します。

■軸

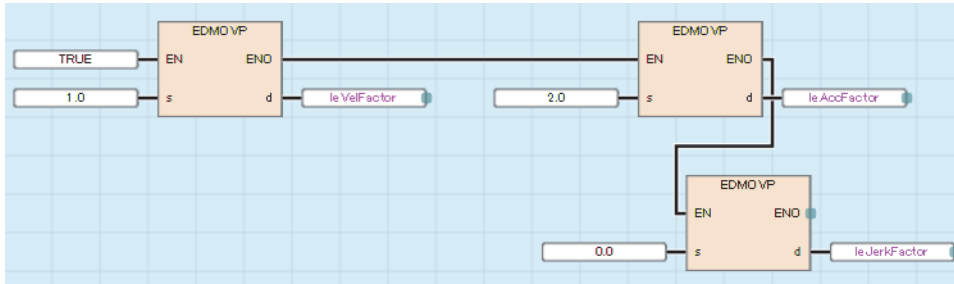
軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■使用するラベル

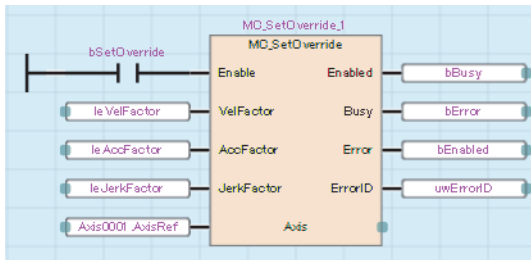
ラベル名	データ型	コメント
MC_SetOverride_1	MC_SetOverride	オーバーライド値設定FB
bSetOverride	ビット	オーバーライド値変更指令
leVelFactor	倍精度実数	速度オーバーライド係数
leAccFactor	倍精度実数	加減速度オーバーライド係数
leJerkFactor	倍精度実数	ジャークオーバーライド係数
bEnabled	ビット	有効中
bBusy	ビット	実行中
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

- ・ オーバライド値変更用データ設定



- ・ オーバライド値変更



■STプログラム

//-----オーバライド値変更用データ設定-----

leVelFactor:= 1.00;

leAccFactor:= 2.00;

leJerkFactor:= 0.0;

//-----オーバライド値変更-----

MC_SetOverride_1(

Axis:= Axis0001.AxisRef ,

Enable:= bSetOverride ,

VelFactor:= leVelFactor ,

AccFactor:= leAccFactor ,

JerkFactor:= leJerkFactor ,

Enabled=> bEnabled ,

Busy=> bBusy ,

Error=> bError ,

ErrorID=> uwErrorID

);

41.7 パラメータ読出

41

MC_ReadParameter

デバイス機器のオブジェクトの読出しを行います。

ラダー	FBD/LD	ST
		<pre> MC_ReadParameter(Axis:= ?AXIS_REF?, Enable:= ?BOOL?, ParameterNumber:= ?DWORD?, ReadCount:= ?WORD?, Options:= ?DWORD?, Valid=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?, SDOErrorID=> ?DWORD?, Value=> ?LREAL?, SDOStatus=> ?WORD?, ReadCounter=> ?WORD?); </pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📖 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Enable	有効	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_ReadParameter(/パラメータ読出)を実行します。
ParameterNumber	パラメータ番号	DWORD(HEX)	常時	00010000H~FFFFFFFFH	00000000H	デバイス機器のオブジェクトを設定します。 詳細は、下記を参照してください。 📖 1489ページ パラメータ番号 (ParameterNumber)
ReadCount	読出し回数	WORD(UNIT)	常時	0~65535	0	パラメータの読出し回数を設定します。 「0」を設定した場合、有効(Enable)がFALSEとなるまで連続読出しを行います。 再取込みは、パラメータ番号を変更したタイミングで行います。
Options	オプション	DWORD(HEX)	常時	00000000H~00010000H	00000000H	MC_ReadParameter(/パラメータ読出)の機能オプションをビット指定で設定します。 再取込みは、パラメータ番号を変更したタイミングで行います。 詳細は、下記を参照してください。 📖 1490ページ オプション(Options)

■出力変数

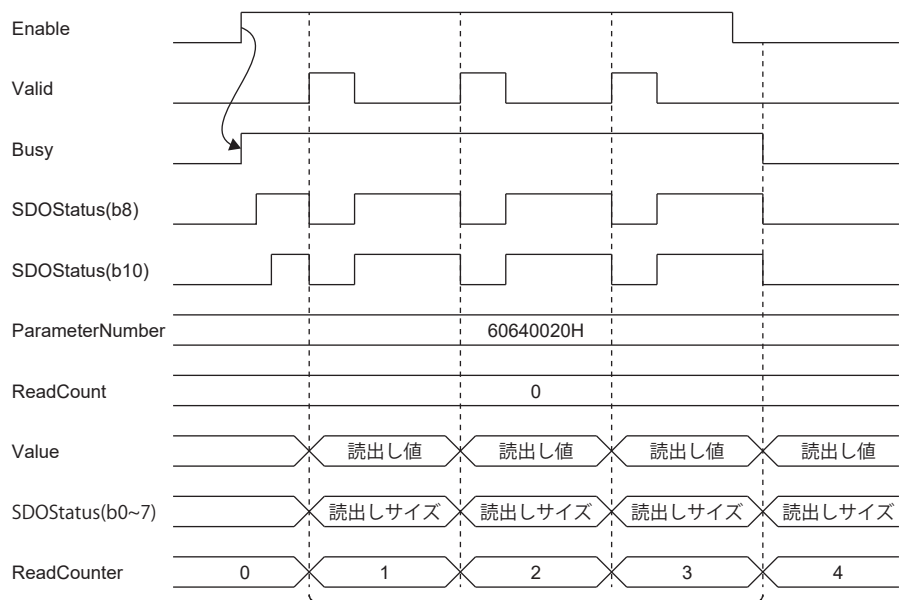
出力変数	名称	データ型	初期値	内容
Valid	出力値有効	BOOL	FALSE	出力値が有効なときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MC_ReadParameter(パラメータ読出)を実行したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。エラーコードの詳細は、下記を参照してください。 📖使用するコントローラのユーザーズマニュアル
SDOErrorID	トランジェントエラーコード	DWORD(HEX)	00000000H	SDO通信で異常が発生したときに、応答コード(SDO Abort Code)を返します。
Value	読出し値	LREAL	0.0	指定したパラメータの読出し値を出力します。対象となるパラメータが整数データの場合もLREAL型として格納します。
SDOStatus	SDO転送ステータス	WORD(UINT)	0	トランジェント要求の処理状態を格納します。詳細は、下記を参照してください。 📖1490ページ SDO転送ステータス(SDOStatus)
ReadCounter	読出し回数	WORD(UINT)	0	パラメータの読出し回数を格納します。読出し回数(ReadCount)に「0」を設定している場合、「0~65535」のリングカウンタとなります。

機能

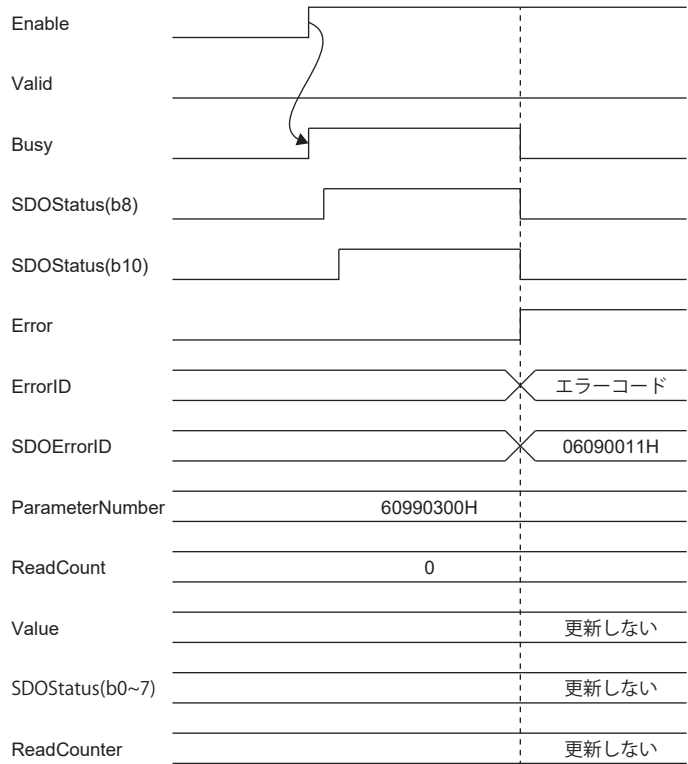
- パラメータ番号(ParameterNumber)に00010000H以降を設定することでオブジェクトデータを読み出します。この場合、トランジェント伝送機能を使用しオブジェクトデータを送受信します。
- デバイス機器からトランジェント要求に対するエラー (応答コード(SDO Abort Code))を受信した場合、エラー (Error)がTRUEになり、エラーコード(ErrorID)にSDO通信異常(エラーコード: 1A48H)、トランジェントエラーコード(SDOErrorID)にデバイス機器の応答コード(SDO Abort Code)を格納します。

■タイミングチャート

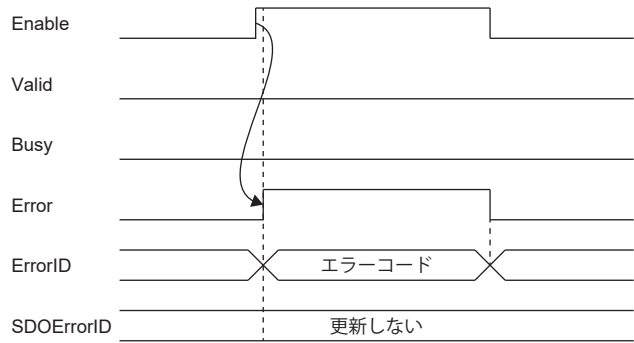
- 正常完了の場合



- ・ 異常完了の場合
 - ・ SDO通信エラー発生時



- ・ 入出力変数，入力変数異常

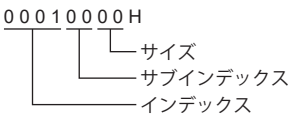


■パラメータ番号(ParameterNumber)

- ・ パラメータ読出し時に設定するパラメータ番号(ParameterNumber)は下記の区分となります。

パラメータ番号	内容
00010000H~FFFFFFFFH	デバイス機器のオブジェクト

- ・ デバイス機器のオブジェクトを設定するパラメータ番号(ParameterNumber)の内容を下記に示します。



- ・ デバイス機器のオブジェクト設定時のサイズは、1バイト(08H)/2バイト(10H)/4バイト(20H)/8バイト(40H)をビット数に置き換えて設定してください。「00H」を設定した場合、パラメータ読出しでは既定サイズでの読出しとなります。設定値以外の値を設定した場合、パラメータ番号範囲外(エラーコード: 1BC2H)となります。
- ・ パラメータ番号(ParameterNumber)で設定したサイズが、オブジェクトデータの既定サイズと異なる場合の動作は、ドライバの仕様によります。

例

オブジェクトインデックス「6099H」、サブインデックス「02H」のUNSIGNED32オブジェクト、サイズ「00H」を設定する場合
「60990200H」を設定します。

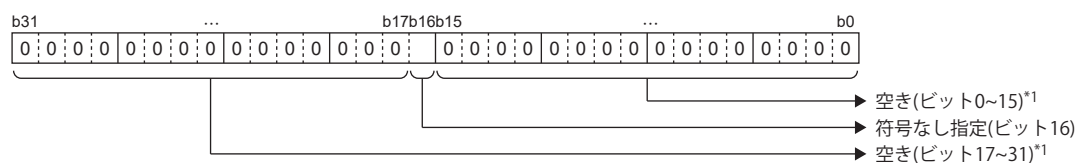
Point

各デバイス機器のオブジェクトについては、デバイス機器のマニュアルを参照してください。

■オプション(Options)

MC_ReadParameter(パラメータ読出)で使用する機能オプションをビット指定で設定します。

ビット指定で設定する内容を下記に示します。

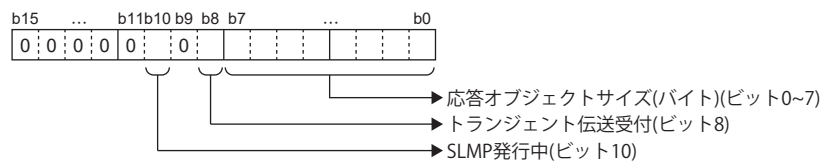


*1 空きには「0」を設定してください。「0」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

ビット	名称	内容
16	符号なし指定	整数データの読出し値を符号ありとして格納するか、符号なしとして格納するかを設定します。 • 0: 符号あり • 1: 符号なし

■SDO転送ステータス(SDOStatus)

トランジェント要求の処理状態を格納します。



ビット	名称	内容
0~7	応答オブジェクトサイズ(バイト)	処理完了時に、デバイス機器が応答したオブジェクトサイズを格納します。
8	トランジェント伝送受付	トランジェント伝送の状態を格納します。 • 0: トランジェント伝送未実行 • 1: トランジェント伝送受付中
10	SLMP発行中	SLMPの状態を格納します。 • 0: SLMP未実行 • 1: SLMP発行中

注意事項

- トランジェント伝送は同一軸に対して最大4つまで実行できます。すでに4つトランジェント伝送を実行している軸に対してオブジェクトデータを対象としたMC_ReadParameter(パラメータ読出)を実行すると、パラメータ読み書きFB実行不可エラー (エラーコード: 1BC1H)となり、後続のFBは実行しません。ただし、実行中のFBは継続します。
- ドライバ式原点復帰では、トランジェント伝送機能を用いてドライバの原点データが取得されます。そのため、MC_Home(原点復帰)実行時にオブジェクトデータを対象としたパラメータ読出しを実行している場合、MC_Home(原点復帰)でABS基準点リードエラー (エラーコード: 1AF9H)となる場合があります。
- 整数型、BOOL型のパラメータやオブジェクトデータを読出しする場合、MC_ReadParameter(パラメータ読出)で読出した値は、LREAL型、BOOL型に変換して出力されます。

例

MC_ReadParameter(パラメータ読出)で2バイトサイズ指定にてパラメータを読出し

パラメータ値	読出し値(Value)(データ型: LREAL)	
	符号なし指定(オプション(Options): ビット16) 「0: 符号あり」	符号なし指定(オプション(Options): ビット16) 「1: 符号なし」
10000	10000.0	10000.0
-20000	-20000.0	45536.0
65535	-1.0	65535.0
63356	0.0	0.0
131701	-1.0	65535.0

- MC_ReadParameter(パラメータ読出)により変更した値は保存しません。次回電源投入時に変更した値を使用する場合は、パラメータの保存を行ってください。
デバイス機器オブジェクトの場合、パラメータの保存方法はデバイス機器のマニュアルを参照してください。
- トランジェント伝送中のデバイス機器が解列した場合、解列したデバイス機器を対象としたトランジェント要求はすべてエラーとなり、SDO通信異常(エラーコード: 1A48H)が発生します。
- MC_ReadParameter(パラメータ読出)はドライバ状態(AxisName.Md.Driver_State)が「-1: 軸変数未初期化/軸パラメータ異常(Invalid)」または「0: Not Ready To Switch On(NotReadyToSwitchOn)」になっていないことを確認してから起動してください。
上記以外で実行した場合、パラメータ読み書きFB実行不可エラー (エラーコード: 1BC1H)が発生します。

プログラム例

パラメータ読み出し指令(bReadParameter)をTRUEにし、軸1(Axis0001)のインデックス「6099H」、サブインデックス「02H」、サイズ「00H」のオブジェクトの値を読み出すプログラム例を下記に示します。

■軸

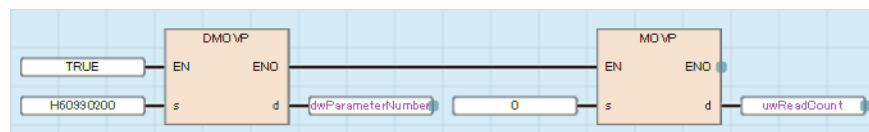
軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■使用するラベル

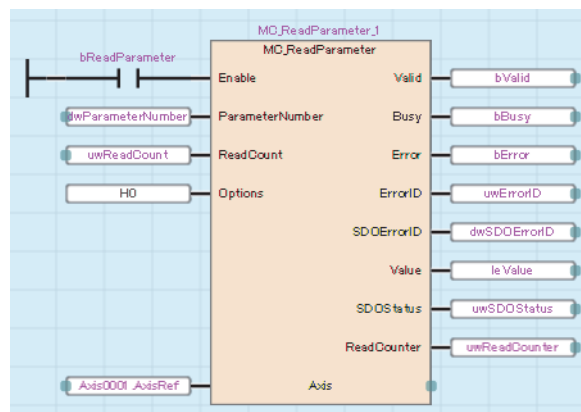
ラベル名	データ型	コメント
MC_ReadParameter_1	MC_ReadParameter	パラメータ読み出しFB
bReadParameter	ビット	パラメータ読み出し指令
dwParameterNumber	ダブルワード[符号なし]/ビット列[32ビット](16進数)	パラメータ番号
uwReadCount	ワード[符号なし]/ビット列[16ビット]	読み出し回数
bValid	ビット	出力値有効
bBusy	ビット	実行中
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
dwSDOErrorID	ダブルワード[符号なし]/ビット列[32ビット](16進数)	トランジェントエラーコード
leValue	倍精度実数	読み出し値
uwSDOStatus	ワード[符号なし]/ビット列[16ビット]	SDO転送ステータス
uwReadCounter	ワード[符号なし]/ビット列[16ビット]	読み出し回数

■FBD/LDプログラム

- パラメータ読み出しデータ設定



- パラメータ読み出し



■STプログラム

//-----パラメータ読出用データ設定-----

dwParameterNumber:= H60990200;

uwReadCount:= 0;

//-----パラメータ読出-----

MC_ReadParameter_1(

Axis:= Axis0001.AxisRef ,

Enable:= bReadParameter ,

ParameterNumber:= dwParameterNumber ,

ReadCount:= uwReadCount ,

Options:= H00000000 ,

Valid=> bValid ,

Busy=> bBusy ,

Error=> bError ,

ErrorID=> uwErrorID ,

SDOErrorID=> dwSDOErrorID ,

Value=> leValue ,

SDOStatus=> uwSDOStatus ,

ReadCounter=> uwReadCounter

);

41.8 パラメータ書込

MC_WriteParameter

デバイス機器のオブジェクトの書込みを行います。

ラダー	FBD/LD	ST
		<pre> MC_WriteParameter(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, ParameterNumber:= ?DWORD?, Value:= ?LREAL?, ExecutionMode:= ?INT?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?, SDOErrorID=> ?DWORD?, SDOStatus=> ?WORD?); </pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 ☞ 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_WriteParameter(/パラメータ書込)を実行します。
ParameterNumber	パラメータ番号	DWORD(HEX)	起動時	00010000H~FFFFFFFFH	00000000H	デバイス機器のオブジェクトを設定します。 詳細は、下記を参照してください。 ☞ 1496ページ パラメータ番号 (ParameterNumber)
Value	設定値	LREAL	起動時	—	0.0	設定したパラメータの設定値を設定します。
ExecutionMode	起動モード	INT (MC_EXECUTION_MODE)	起動時	0, 1	0	設定したパラメータの書込み方法を設定します。 ・0: 直ちに書込み(mclmmediately) ・1: 軸の停止を待ってから書込み (mcQueued) 詳細は、下記を参照してください。 ☞ 1497ページ 起動モード (ExecutionMode)
Options	オプション	DWORD(HEX)	起動時	00000000H~00010000H	00000000H	MC_WriteParameter(/パラメータ書込)の機能オプションをビット指定で設定します。 詳細は、下記を参照してください。 ☞ 1497ページ オプション(Options)

■出力変数

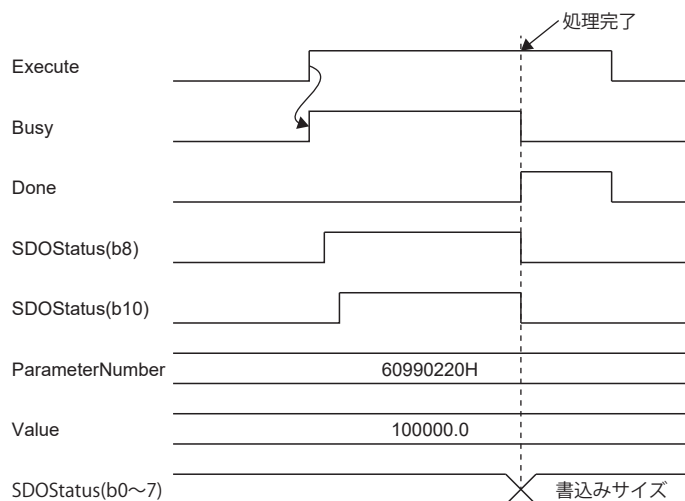
出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	パラメータの書き込みが完了したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MC_WriteParameter(パラメータ書き込み)を実行したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。エラーコードの詳細は、下記を参照してください。 📖使用するコントローラのユーザーズマニュアル
SDOErrorID	トランジェントエラーコード	DWORD(HEX)	0000000H	SDO通信で異常が発生したときに、応答コード(SDO Abort Code)を返します。
SDOStatus	SDO転送ステータス	WORD(UINT)	0	トランジェント要求の処理状態を格納します。詳細は、下記を参照してください。 📖 1497ページ SDO転送ステータス(SDOStatus)

機能

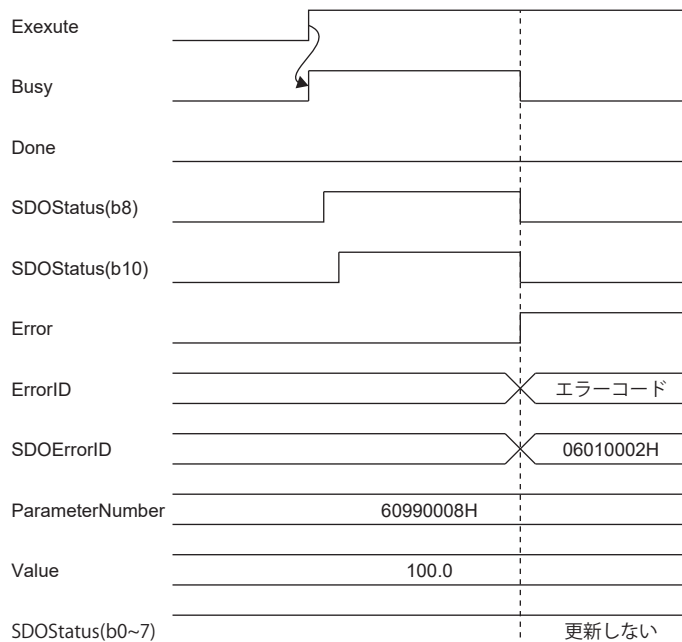
- パラメータ番号(ParameterNumber)に00010000H以降を設定することでオブジェクトデータを書き込みます。この場合、トランジェント伝送機能を使用しオブジェクトデータを送受信します。
- デバイス機器からトランジェント要求に対するエラー (応答コード(SDO Abort Code))を受信した場合、エラー (Error)がTRUEになり、エラーコード(ErrorID)にSDO通信異常(エラーコード: 1A48H)、トランジェントエラーコード(SDOErrorID)にデバイス機器の応答コード(SDO Abort Code)を格納します。

■タイミングチャート

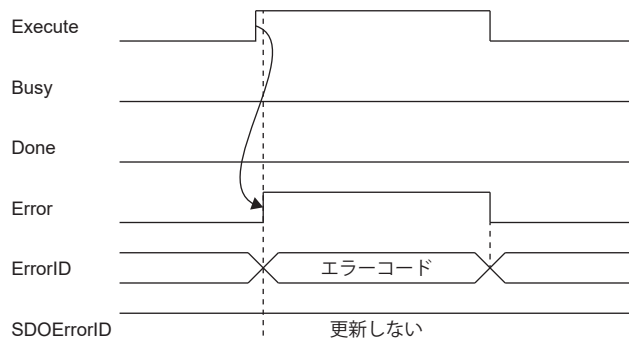
- 正常完了の場合



- ・異常完了の場合
 - ・SDO通信エラー発生時



- ・入出力変数，入力変数異常

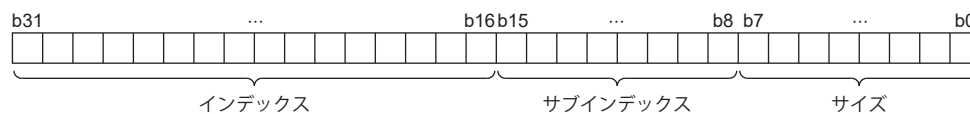


■パラメータ番号(ParameterNumber)

- ・パラメータ書き込み時に設定するパラメータ番号(ParameterNumber)は下記の区分となります。

パラメータ番号	内容
00010000H~FFFFFFFH	デバイス機器のオブジェクト

- ・デバイス機器のオブジェクトを設定するパラメータ番号(ParameterNumber)の内容を下記に示します。



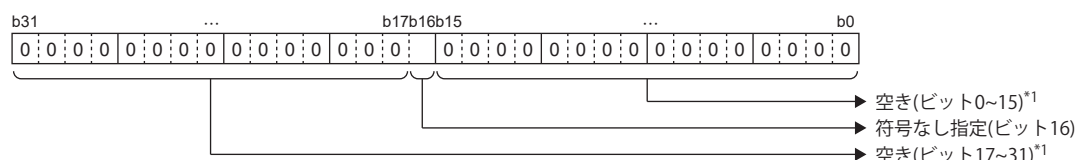
- ・デバイス機器のオブジェクト設定時のサイズは、1バイト(08H)/2バイト(10H)/4バイト(20H)/8バイト(40H)をビット数に置き換えて設定してください。「00H」を設定した場合、パラメータ書き込みでは4バイトでの書き込みとなります。設定値以外の値を設定した場合、パラメータ番号範囲外(エラーコード: 1BC2H)となります。
- ・パラメータ番号(ParameterNumber)で設定したサイズが、オブジェクトデータの既定サイズと異なる場合の動作は、ドライバの仕様によります。

例

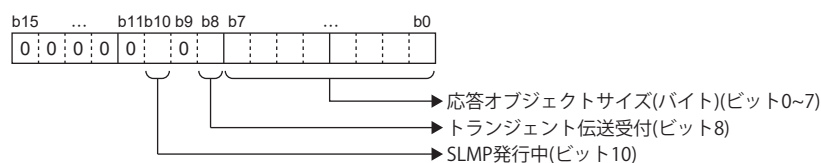
オブジェクトインデックス「6099H」、サブインデックス「02H」のUNSIGNED32オブジェクト、サイズ「00H」を設定する場合

「60990200H」を設定します。

設定値	内容
0: 直ちに書き込み(mcImmediately)	直ちに書き込みを実行します。 反映タイミングは各パラメータによるため、軸動作に影響を及ぼす可能性があります。
1: 軸の停止を待ってから書き込み(mcQueued)	軸の停止(軸状態(AxisName.Md.AxisStatus)が「0: 軸無効(Disabled)」, または「4: 待機中(Standstill)」)を待ってから書き込みを実行します。



ビット	名称	内容
16	符号なし指定	整数データの設定値を符号ありとして格納するか、符号なしとして格納するかを設定します。 ・0: 符号あり ・1: 符号なし



注意事項

例

設定値(Value) (データ型: LREAL)	書き込む値(2バイトデータ)	
	符号なし指定(オプション(Options): ビット16) 「0: 符号あり」	符号なし指定(オプション(Options): ビット16) 「1: 符号なし」
10000.5	10000	10000
-10000.4	-10000	55536
65535.0	-1	65535

設定値(Value) (データ型: LREAL)	書き込む値(2バイトデータ)	
	符号なし指定(オプション(Options): ビット16) 「0: 符号あり」	符号なし指定(オプション(Options): ビット16) 「1: 符号なし」
-65535.0	1	1
63356.0	0	0
131701.0	-1	65535

- MC_WriteParameter(パラメータ書込)により変更した値は保存しません。次回電源投入時に変更した値を使用する場合は、パラメータの保存を行ってください。
デバイス機器オブジェクトの場合、パラメータの保存方法はデバイス機器のマニュアルを参照してください。
- トランジェント伝送中のデバイス機器が解列した場合、解列したデバイス機器を対象としたトランジェント要求はすべてエラーとなり、SDO通信異常(エラーコード: 1A48H)が発生します。
- MC_WriteParameter(パラメータ書込)はドライバ状態(AxisName.Md.Driver_State)が「-1: 軸変数未初期化/軸パラメータ異常(Invalid)」または「0: Not Ready To Switch On(NotReadyToSwitchOn)」になっていないことを確認してから起動してください。
上記以外で実行した場合、パラメータ読み書きFB実行不可エラー (エラーコード: 1BC1H)が発生します。

プログラム例

パラメータ書込み指令(bWriteParameter)をTRUEにし、軸1(Axis0001)のインデックス「6099H」、サブインデックス「02H」、サイズ「00H」のオブジェクトに「100.0」を書き込むプログラム例を下記に示します。

■軸

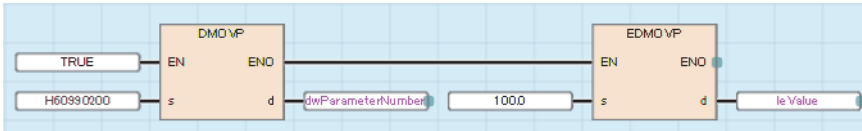
軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■使用するラベル

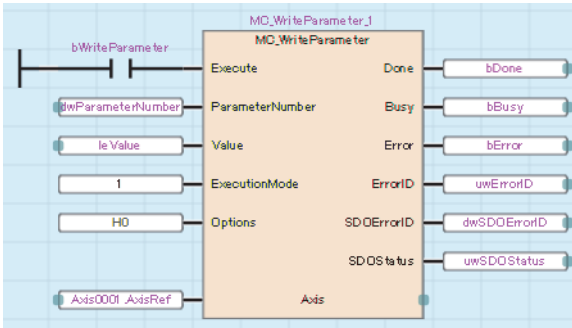
ラベル名	データ型	コメント
MC_WriteParameter_1	MC_WriteParameter	パラメータ書込FB
bWriteParameter	ビット	パラメータ書込み指令
dwParameterNumber	ダブルワード[符号なし]/ビット列[32ビット](16進数)	パラメータ番号
leValue	倍精度実数	設定値
bDone	ビット	実行完了
bBusy	ビット	実行中
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
dwSDOErrorID	ダブルワード[符号なし]/ビット列[32ビット](16進数)	トランジェントエラーコード
uwSDOStatus	ワード[符号なし]/ビット列[16ビット]	SDO転送ステータス

■FBD/LDプログラム

- パラメータ書込用データ設定



- パラメータ書込



■STプログラム

//-----パラメータ書込用データ設定-----

dwParameterNumber:= H60990200;

leValid:= 100.0;

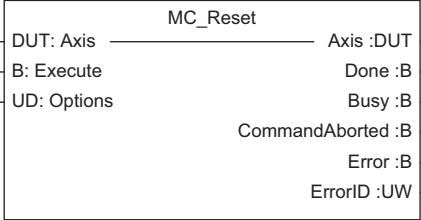
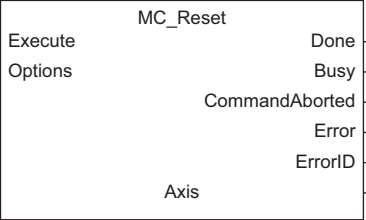
//-----パラメータ書込-----

```
MC_WriteParameter_1(
  Axis:= Axis001.AxisRef,
  Execute:= bWriteParameter,
  ParameterNumber:= dwParameterNumber,
  Value:= leValue,
  ExecutionMode:= MC_EXECUTION_MODE__mcQueued,
  Options:= H00000000,
  Done=> bDone,
  Busy=> bBusy,
  Error=> bError,
  ErrorID=> uwErrorID,
  SDOErrorID=> dwSDOErrorID,
  SDOStatus=> uwSDOStatus
);
```

41.9 軸エラーリセット

MC_Reset

軸のエラー，警告をリセットします。

ラダー	FBD/LD	ST
		<pre>MC_Reset(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については，下記を参照してください。 📖 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_Reset(軸エラーリセット)を実行します。
Options	オプション	DWORD(HEX)	起動時	00000000H	00000000H	「00000000H」を設定してください。 *: 「00000000H」以外を設定した場合，オプション範囲外(エラーコード:1BC3H)となります。

■出力変数

出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	リセットが完了したときに，TRUEになります。
Busy	実行中	BOOL	FALSE	MC_Reset(軸エラーリセット)を実行したときに，TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	タイムアウトにより，MC_Reset(軸エラーリセット)が中断するとTRUEになります。 実行指令(Execute)がFALSEになると，FALSEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに，TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに，エラーコードを返します。 エラーコードの詳細は，下記を参照してください。 📖使用するコントローラのユーザーズマニュアル

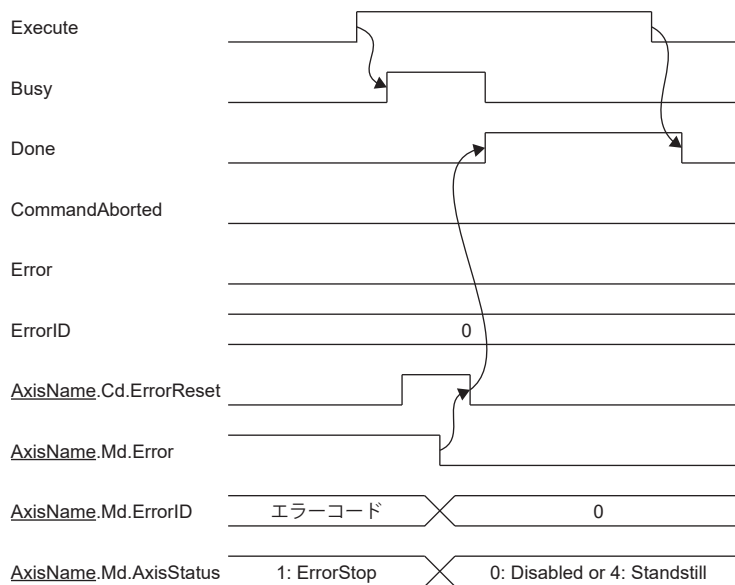
機能

- 実行指令(Execute)がTRUEでMC_Reset(軸エラーリセット)を実行し、処理を開始すると実行中(Busy)がTRUEになり、対象軸の軸エラーリセット(AxisName.Cd.ErrorReset)をTRUEにします。
- 軸のエラー、警告の解除が完了すると実行中(Busy)がFALSEになり、実行完了(Done)がTRUEになります。
- 軸状態(AxisName.Md.AxisStatus)が「1: エラー停止中(ErrorStop)」以外有的时候に実行すると、警告を解除後、実行中(Busy)がFALSEになり、実行完了(Done)がTRUEになります。
- 軸のエラー、警告の要因が残った状態で実行指令(Execute)をTRUEにしても、エラー、警告を解除しません。命令実行後から1秒以内にエラーが解除されない場合、実行中断(CommandAborted)がTRUEになります。いったん実行指令(Execute)をFALSEにし、エラー、警告の要因を取り除いた後、再度実行指令(Execute)をTRUEにしてください。
- MC_Reset(軸エラーリセット)内で異常が発生した場合、エラー (Error)がTRUEになり、エラーコード(ErrorID)にエラーコードを格納します。エラーコードの詳細は、下記を参照してください。

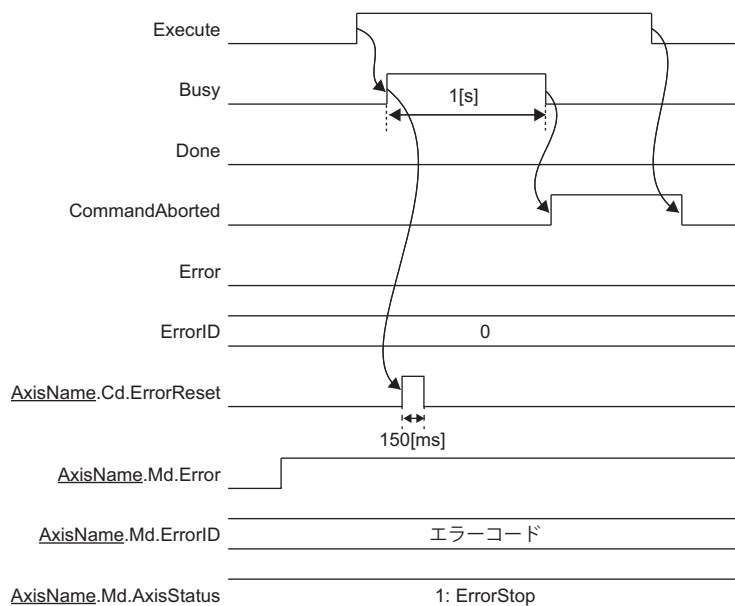
📖使用するコントローラのユーザーズマニュアル

■タイミングチャート

- 正常完了の場合



- タイムアウトの場合



注意事項

- MC_Reset(軸エラーリセット)を実行中に軸エラーリセット(AxisName.Cd.ErrorReset)の直接操作を行わないでください。

プログラム例

エラーリセット指令(bErrorReset)をTRUEにし、軸1(Axis0001)のエラー、警告をリセットするプログラム例を下記に示します。

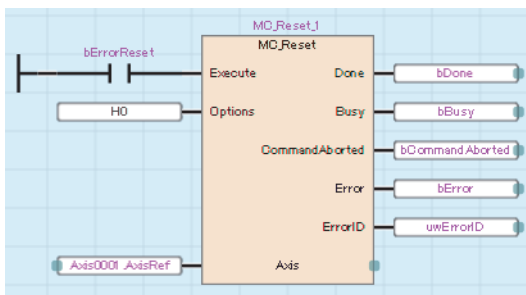
■軸

軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■使用するラベル

ラベル名	データ型	コメント
MC_Reset_1	MC_Reset	軸エラーリセットFB
bErrorReset	ビット	エラーリセット指令
bDone	ビット	実行完了
bBusy	ビット	実行中
bCommandAborted	ビット	実行中断
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム



■STプログラム

```
MC_Reset_1(  
  Axis:= Axis0001.AxisRef ,  
  Execute:= bErrorReset ,  
  Options:= H00000000 ,  
  Done=> bDone ,  
  Busy=> bBusy ,  
  CommandAborted=> bCommandAborted ,  
  Error=> bError ,  
  ErrorID=> uwErrorID  
);
```

41.10 軸グループエラーリセット

MC_GroupReset

軸グループ、および軸グループに属する各軸のエラー、警告をリセットします。

ラダー	FBD/LD	ST
		<pre>MC_GroupReset(AxesGroup:= ?AXES_GROUP_REF?, Execute:= ?BOOL?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
AxesGroup	軸グループ情報	AXES_GROUP_REF	起動時	—	省略不可	軸グループを設定します。 使用する変数(AxesGroupName.AxesGroupRef.)については、下記を参照してください。 1402ページ AxesGroupName.AxesGroupRef. (軸グループ情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_GroupReset(軸グループエラーリセット)を実行します。
Options	オプション	DWORD(HEX)	起動時	00000000H	00000000H	「00000000H」を設定してください。 *: 「00000000H」以外を設定した場合、オプション範囲外(エラーコード: 1BC3H)となります。

■出力変数

出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	軸グループの全軸のリセットが完了したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MC_GroupReset(軸グループエラーリセット)を実行したときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	タイムアウトにより、MC_GroupReset(軸グループエラーリセット)が中断するとTRUEになります。 実行指令(Execute)がFALSEになると、FALSEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 使用使用するコントローラのユーザーズマニュアル

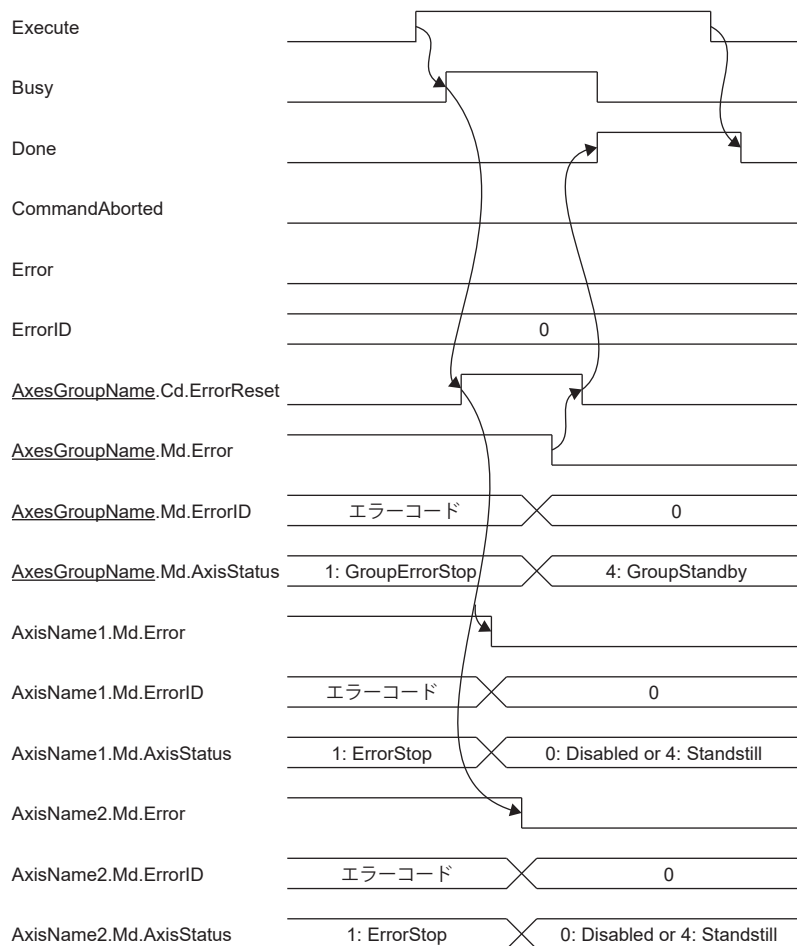
機能

- 実行指令(Execute)の立上り検知でMC_GroupReset(軸グループエラーリセット)を実行し、処理を開始すると実行中(Busy)がTRUEになり、対象軸グループの軸グループエラーリセット(AxesGroupName.Cd.ErrorReset)をTRUEにします。
- 軸グループのエラー、または警告、および単軸のエラー、または警告の解除が完了すると実行中(Busy)がFALSEになり、実行完了(Done)がTRUEとなります。
- 軸グループ状態(AxesGroupName.Md.GroupStatus)が「1: エラー停止中(GroupErrorStop)」以外のときに実行すると、警告を解除後、実行中(Busy)がFALSEになり、実行完了(Done)がTRUEになります。
- 軸のエラー、または警告の要因が残った状態で実行指令(Execute)をTRUEにしても、エラー、警告を解除しません。この場合、FB実行後から1秒以内にエラーを解除しないと、実行中断(CommandAborted)がTRUEとなり、対象軸グループの軸グループエラーリセット(AxesGroupName.Cd.ErrorReset)をFALSEにします。いったん実行指令(Execute)をFALSEにし、エラー、または警告の要因を取り除いた後、再度実行指令(Execute)をTRUEにしてください。
- MC_GroupReset(軸グループエラーリセット)内で異常が発生した場合、エラー(Error)がTRUEになり、エラーコード(ErrorID)にエラーコードを格納します。エラーコードの詳細は、下記を参照してください。

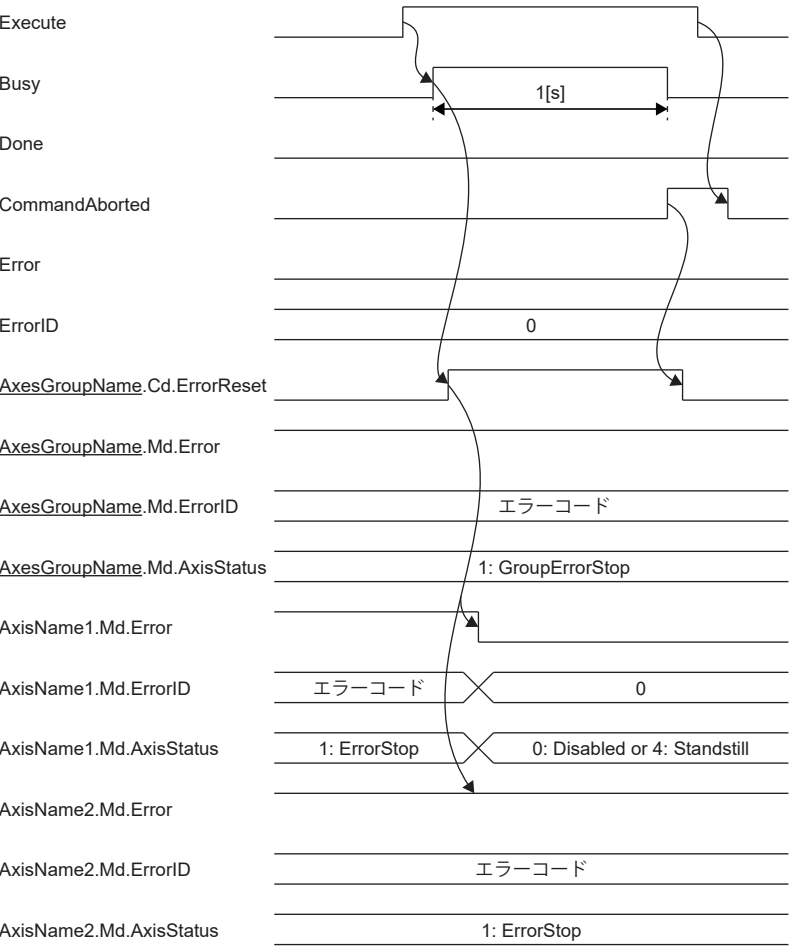
📖 使用するコントローラのユーザーズマニュアル

■ タイミングチャート

- 正常完了の場合



・ タイムアウトの場合



注意事項

- ・ MC_GroupReset(軸グループエラーリセット)を実行中に軸グループエラーリセット(AxesGroupName.Cd.ErrorReset)の直接操作を行わないでください。

プログラム例

軸グループエラーリセット指令(bGroupErrorReset)をTRUEにし，軸グループ1(AxesGroup001)のエラー，警告をリセットするプログラム例を下記に示します。

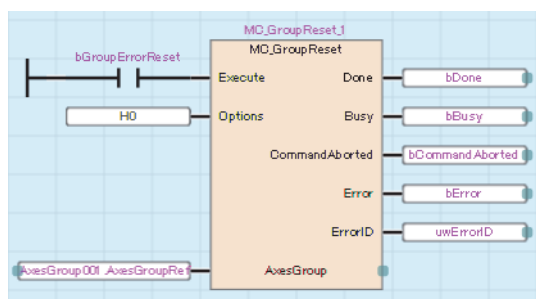
■軸グループ

軸グループNo.	ラベル名	データ型	コメント
1	AxesGroup001	AXES_GROUP_REF	軸グループ1

■使用するラベル

ラベル名	データ型	コメント
MC_GroupReset_1	MC_GroupReset	軸グループエラーリセットFB
bGroupErrorReset	ビット	軸グループエラーリセット指令
bDone	ビット	実行完了
bBusy	ビット	実行中
bCommandAborted	ビット	実行中断
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム



■STプログラム

```
MC_GroupReset_1(
  AxesGroup:= AxesGroup001.AxesGroupRef ,
  Execute:= bGroupErrorReset ,
  Options:= H00000000 ,
  Done=> bDone ,
  Busy=> bBusy ,
  CommandAborted=> bCommandAborted ,
  Error=> bError ,
  ErrorID=> uwErrorID
);
```


41.11 タッチプローブ有効

41

MC_TouchProbe

トリガイベント発生により任意のデータを記録します。

ラダー	FBD/LD	ST
		<pre>MC_TouchProbe(Axis:= ?AXIS_REF?, TriggerInput:= ?MC_TRIGGER_REF?, Execute:= ?BOOL?, ContinuousUpdate:= ?BOOL?, WindowOnly:= ?BOOL?, FirstPosition:= ?LREAL?, LastPosition:= ?LREAL?, ProbeData:= ?TARGET_REF?, CompensationTime:= ?LREAL?, RecordMode:= ?INT?, RecordCount:= ?WORD?, OutputBuffer:= ?TARGET_REF?, Done=> ?BOOL?, Busy=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?, PositionData=> ?LREAL?, RecordedPosition=> ?LREAL?, RecordedCounter=> ?WORD?, TouchProbeID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、 下記を参照してください。 ☞ 1378ページ AxisName.AxisRef.(軸情報)
TriggerInput	トリガ入力信号	MC_TRIGGER_REF	起動時	—	省略不可	トリガ入力信号を設定します。 詳細は、下記を参照してください。 ☞ 1512ページ トリガ入力信号(TriggerInput)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_TouchProbe(タッチプローブ有効)を実行します。
ContinuousUpdate	連続更新	BOOL	起動時	TRUE, FALSE	FALSE	連続して、ウィンドウ有効(WindowOnly)、下限位置(FirstPosition)、上限位置(LastPosition)の変更を有効にするか、無効にするかを設定します。 • FALSE: 無効 • TRUE: 有効
WindowOnly	ウィンドウ有効	BOOL	起動時/再起動可/連続更新可	TRUE, FALSE	FALSE	下限位置と上限位置で設定した有効範囲内でトリガイベントが発生したときのみ、検出するか、検出しないかを設定します。 • FALSE: 全範囲で検出 • TRUE: 設定した範囲でのみ検出 詳細は、下記を参照してください。 ☞ 1513ページ ウィンドウ有効(WindowOnly)/下限位置(FirstPosition)/上限位置(LastPosition)

入力変数	名称	データ型	取込	有効範囲	初期値	内容
FirstPosition	下限位置	LREAL	起動時/再起動可 /連続更新可	—	0.0	トリガイベントの有効範囲となる下限位置を設定します。 有効領域は下限位置を含みます。 詳細は、下記を参照してください。 ☞ 1513ページ ウィンドウ有効(WindowOnly)/ 下限位置(FirstPosition)/上限位置(LastPosition)
LastPosition	上限位置	LREAL	起動時/再起動可 /連続更新可	—	0.0	トリガイベントの有効範囲となる上限位置を設定します。 有効領域は上限位置を含みます。 詳細は、下記を参照してください。 ☞ 1513ページ ウィンドウ有効(WindowOnly)/ 下限位置(FirstPosition)/上限位置(LastPosition)
ProbeData	プローブデータ	TARGET_REF	起動時	—	—	プローブデータを設定します。 詳細は、下記を参照してください。 ☞ 1514ページ プローブデータ(ProbeData)
CompensationTime	補正時間	LREAL	起動時	-5.0~5.0[s]	0.0	タッチプローブ処理の遅れ時間を補正する値を設定します。 遅れを補正する場合は、正の値を設定します。 詳細は、下記を参照してください。 ☞ 1515ページ 補正時間(CompensationTime)
RecordMode	ラッチモード	INT (MC_RECORD_MODE)	起動時	0~2	0	データのラッチ方法を設定します。 ・0: 単発モード(OneShot) ・1: 指定回数モード(RecordCount) ・2: リングバッファモード(RingBuffer) 詳細は、下記を参照してください。 ☞ 1516ページ ラッチモード(RecordMode)
RecordCount	ラッチ回数	WORD(UINT)	起動時	1~65535	1	データのラッチ回数を設定します。 詳細は、下記を参照してください。 ☞ 1516ページ ラッチ回数(RecordCount)
OutputBuffer	ラッチデータ格納先	TARGET_REF	起動時	—	—	ラッチデータの格納先を設定します。 省略した場合は、ラッチ位置(RecordedPosition)にのみラッチデータを出力します。 詳細は、下記を参照してください。 ☞ 1517ページ ラッチデータ格納先 (OutputBuffer)

■出力変数

出力変数	名称	データ型	初期値	内容
Done	完了	BOOL	FALSE	制御が完了したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MC_TouchProbe(タッチプローブ有効)を実行したときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	エラーや多重起動などにより、MC_TouchProbe(タッチプローブ有効)の実行が中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 ☞ 使用するコントローラのユーザーズマニュアル
PositionData	現在値モニタデータ	LREAL	0.0	ラッチするデータの現在値を格納します。
RecordedPosition	ラッチ位置	LREAL	0.0	トリガイベント発生時の現在値を格納します。 複数回ラッチする場合は、最新データに上書きします。
RecordedCounter	ラッチ回数	WORD(UINT)	0	ラッチした回数を格納します。
TouchProbeID	プローブID	WORD(UINT)	0	MC_TouchProbe(タッチプローブ有効)固有のIDを格納します。

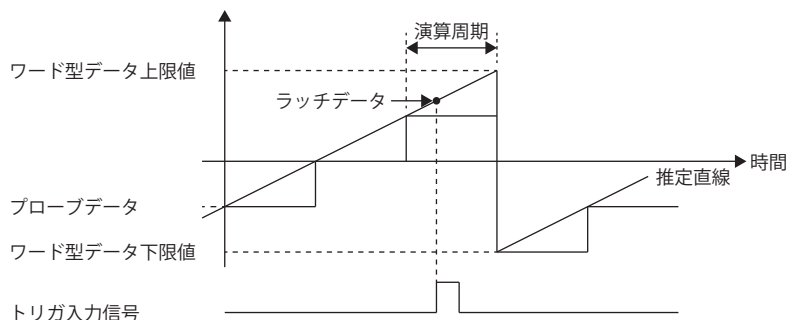
機能

- トリガ入力信号(TriggerInput)で指定した信号の立上り, 立下り, または両方向でラッチデータを推定計算します。
- ウィンドウ有効(WindowOnly)がTRUEの場合, トリガイベント発生時のラッチデータが下限位置(FirstPosition)から上限位置(LastPosition)の範囲内であることを確認します。範囲外の場合, データはラッチしません。
- ラッチモード(RecordMode)に従って, ラッチ位置(RecordedPosition), およびラッチデータ格納先(OutputBuffer)で設定した格納先にラッチデータを格納し, ラッチ回数(RecordedCounter)を更新します。
- 演算周期内のプローブデータを推定計算します。トリガ入力信号(TriggerInput)を入力したタイミングにおいて, 推定計算した値をラッチデータとします。下図のように値を計算します。

■プローブデータがWORD型の場合

- 外部信号高精度入力を使用する場合

トリガ入力タイミングはデバイス局が信号を検出した時刻となります。



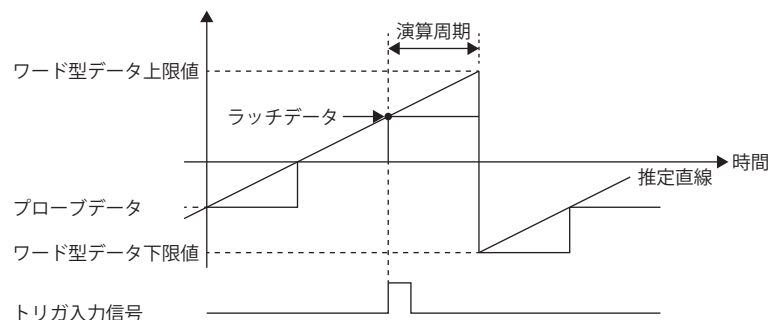
*: 外部信号選択の高精度入力の設定方法については, 下記を参照してください。

□使用するコントローラのユーザーズマニュアル

*: 信号検出時刻の精度については, 使用するデバイス機器のマニュアルを参照してください。

- 外部信号高精度入力を使用しない場合

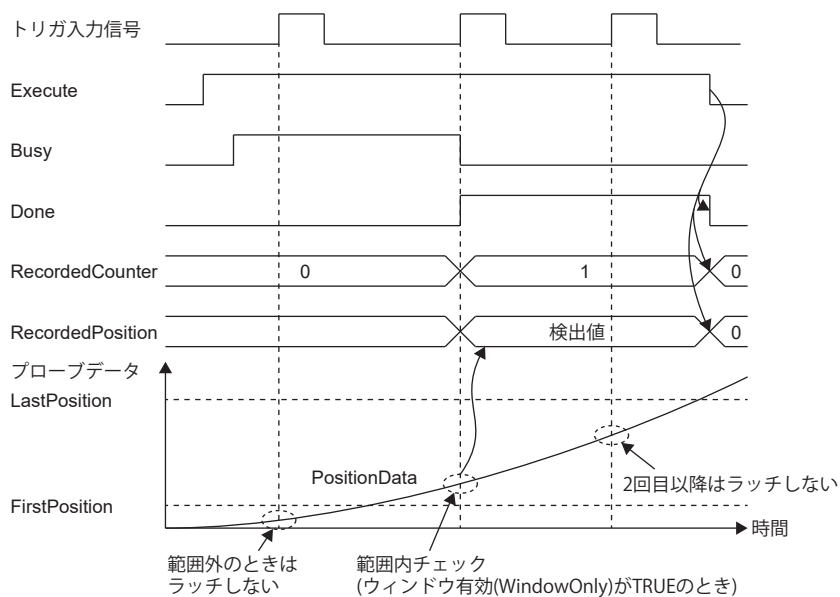
トリガ入力タイミングは演算周期となります。



■タイミングチャート

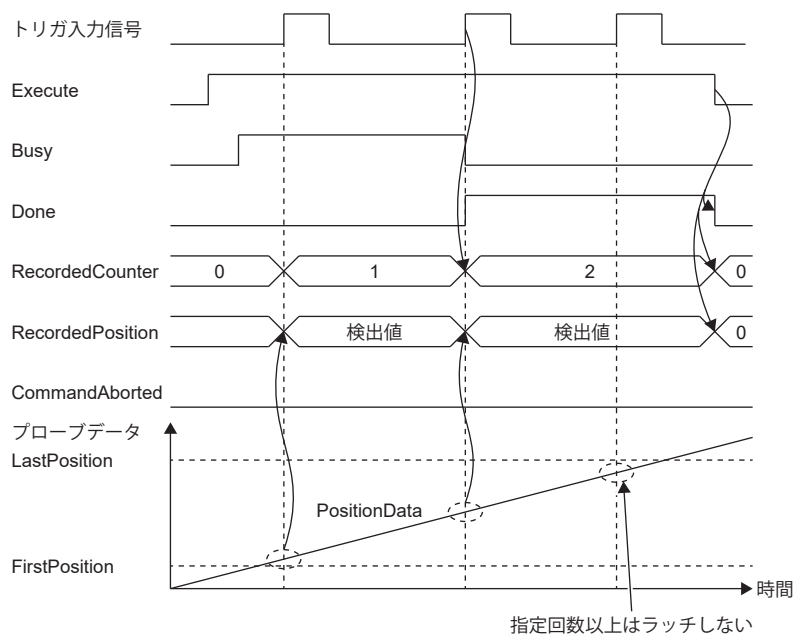
・正常完了の場合

・ラッチモード(RecordMode)が「0: 単発モード(OneShot)」の場合

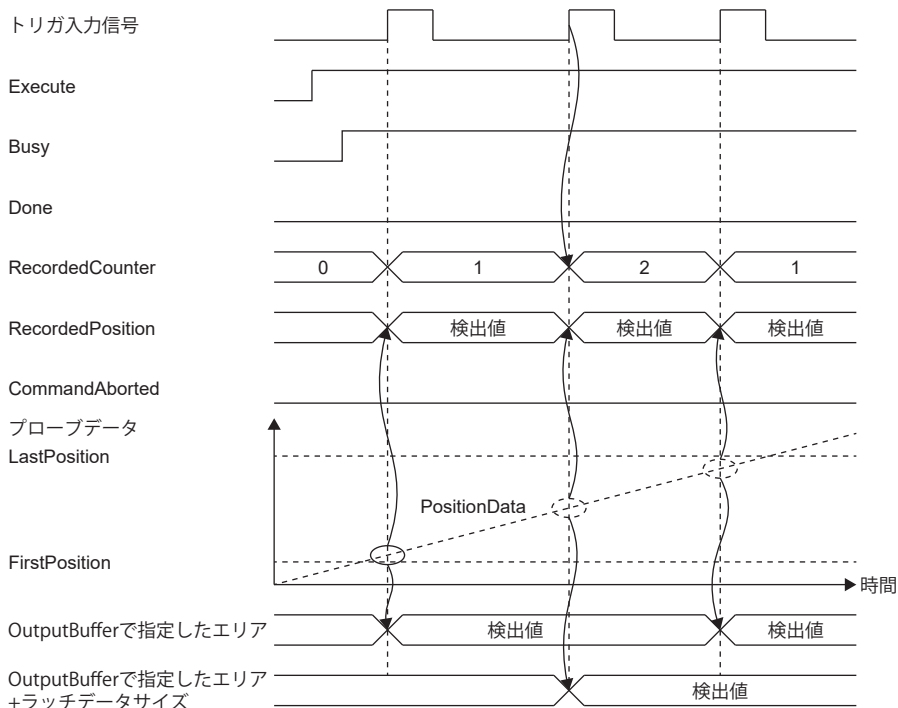


・ラッチモード(RecordMode)が「1: 指定回数モード(RecordCount)」の場合

ラッチ回数(RecordCount)が2回の場合



• ラッチモード(RecordMode)が「2: リングバッファモード(RingBuffer)」の場合
ラッチ回数(RecordCount)が2回の場合



• 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■トリガ入力信号(TriggerInput)

トリガ入力信号をMC_TRIGGER_REF構造体で設定します。MC_TRIGGER_REF構造体については、下記を参照してください。

📖 1420ページ MC_TRIGGER_REF

トリガ入力信号(TriggerInput)で設定するMC_TRIGGER_REF構造体のトリガ信号(Signal)の設定範囲を下記に示します。

トリガ信号(Signal)の型は、SIGNAL_SELECT構造体となります。SIGNAL_SELECT構造体については、下記を参照してください。

📖 1424ページ SIGNAL_SELECT(信号選択)

構造体	変数名	型	設定範囲
SIGNAL_SELECT(信号選択)	Source(信号)	TARGET_REF 📖 1421ページ TARGET_REF(入力信号)	■型 • BOOL ■データ種別 • [OBJ] ^{*1*2} • [LINK] • [VAR] • [DEV] • [CONST]
	Detection(信号検出方法)	INT (MC_SIGNAL_LOGIC) 📖 1452ページ MC_SIGNAL_LOGIC	• 2: FALSE→TRUE(立上り)時検出(RisingEdge) • 3: TRUE→FALSE(立下り)時検出(FallingEdge) • 4: 立上り/立下り時検出(BothEdges)
	CompensationTime(補正時間)	LREAL	-5.0~5.0[s]
	FilterTime(フィルタ時間)	LREAL	0.0~5.0[s]

*1 対象修飾を省略した場合、軸情報(Axis)の軸のオブジェクトを参照します。ただし、軸情報(Axis)の軸が局アドレスを持たない、または局アドレスが未設定の場合は、外部信号選択不正(エラーコード: 1BD6H)となります。対象修飾で局アドレスを指定した場合、軸情報(Axis)は無視します。

参照する局が、軸エミュレート機能有効かつ局アドレス設定済みの場合、エミュレートしているオブジェクトを参照します。

外部信号高精度入力を使用する場合は、Detection(信号検出方法)を4: 立上り/立下り時検出(BothEdges)に設定してください。

*2 外部信号高精度入力を使用できます。詳細は、下記を参照してください。

📖 使用するコントローラのユーザーズマニュアル

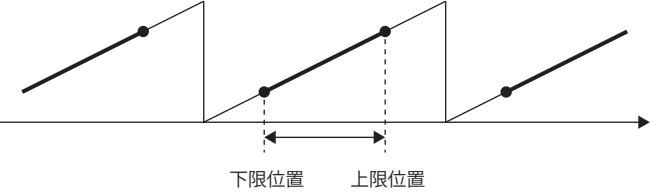
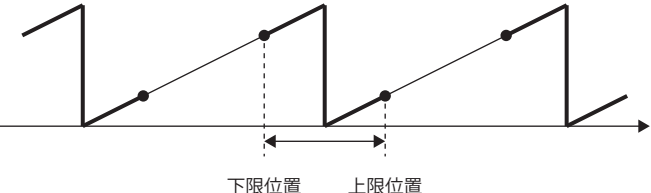
信号名(TriggerInput.Signal.Source.Target)に使用できない信号を使用した場合、外部信号選択文字列不正(エラーコード: 1BC5H)、外部信号選択不正(エラーコード: 1BD6Hまたは1BD7H)、アドオンSignalIO内部異常(エラーコード: 1BD8H)のいずれかを出力します。

Point

外部信号高精度入力を使用するときは、フィルタ時間(TriggerInput.Signal.FilterTime)に「0」を設定してください。「0」以外を設定した場合、信号の検出時刻を正しく取得できません。デバイス機器に入力フィルタが存在する場合は、デバイス機器のフィルタをご使用ください。デバイス機器で入力フィルタが存在しない場合は、タッチプローブ有効(MC_TouchProbe)のウィンドウ有効(WindowOnly)、下限位置(FirstPosition)、上限位置>LastPosition)を併用してノイズによるトリガ信号の誤検出を防いでください。

■ウィンドウ有効(WindowOnly)/下限位置(FirstPosition)/上限位置(LastPosition)

ウィンドウ有効(WindowOnly)が有効(TRUE)の場合、プローブデータが下限位置(FirstPosition)から上限位置(LastPosition)の範囲でのみトリガを検出します。無効(FALSE)の場合は全範囲に対してタッチプローブ処理を行います。

下限位置(FirstPosition)/上限位置(LastPosition)の関係	タッチプローブ処理
下限位置(FirstPosition) ≤ 上限位置(LastPosition)	プローブデータが下限位置(FirstPosition)以上かつ上限位置(LastPosition)以下のときに、タッチプローブ処理を行います。 
下限位置(FirstPosition) > 上限位置(LastPosition)	プローブデータが上限位置(LastPosition)以下、または下限位置(FirstPosition)以上のときに、タッチプローブ処理を行います。 

■プローブデータ(ProbeData)

ラッチするデータをTARGET_REF構造体で設定します。TARGET_REF構造体については、下記を参照してください。

☞ 1421ページ TARGET_REF(入力信号)

使用できるデータ種別を下記に示します。

データ種別	内容
[AXIS]	軸情報(Axis)の軸データを参照します。 対象修飾(@Position等)の設定が必要です。
[VAR]	軸情報(Axis)は無視します。
[DEV]	
[OBJ] (サイクリックデータにマッピング済みのもの)	- 対象修飾で局アドレスを設定した場合、軸情報(Axis)は無視します。 - 対象修飾を省略した場合、軸情報(Axis)の軸のオブジェクトを参照します。ただし、軸情報(Axis)の軸が局アドレスを持たない、または局アドレスが未設定の場合は、エラーとなります。 - 参照する局が、軸エミュレート機能が有効かつ局アドレスが設定済みの場合、エミュレートしているオブジェクトを参照します。

信号名(TARGET_REF.Target)に使用できない信号を使用した場合、外部信号選択文字列不正(エラーコード: 1BC5H), 外部信号選択不正(エラーコード: 1BD6Hまたは1BD7H), アドオンSignalIO内部異常(エラーコード: 1BD8H)のいずれかを出力します。

- データはリングカウンタとして更新します。リングカウンタの上限値、下限値はプローブデータ(ProbeData)のデータ種別、データ型により異なります。

データ種別	内容
[AXIS]	■対象修飾に「@Position」を設定した場合 - 軸のリングカウンタ上限値、リングカウンタ下限値となります。 ■対象修飾に「@CumulativePos」を設定した場合 - 軸の位置決め範囲上限値、位置決め範囲下限値となります。
[VAR]	データ型の最大値、最小値となります。
[DEV]	<例> [VAR]でワード型ラベルを設定した場合。
[OBJ]	- リングカウンタ上限値: 65535 - リングカウンタ下限値: 0

制約事項

1演算周期あたりの変化量が下記の式を満たすようにしてください。満たさないとプローブデータの実変化量と、ラッチした変化量が一致なくなることがあります。

$$1\text{演算周期の変化量} < \frac{|(\text{リングカウンタ最大値}) - (\text{リングカウンタ最小値}) + 1|}{2}$$

■補正時間(CompensationTime)

タッチプローブ処理の遅れ時間を補正します。トリガ入力信号の補正時間とは別にタッチプローブ処理固有の遅れなどを補正する場合に設定してください。遅れを補正する場合は正の値、進みを補正する場合は負の値を設定してください。

合計の補正時間 = 補正時間(CompensationTime) - 補正時間(TriggerInput.Signal.CompensationTime)

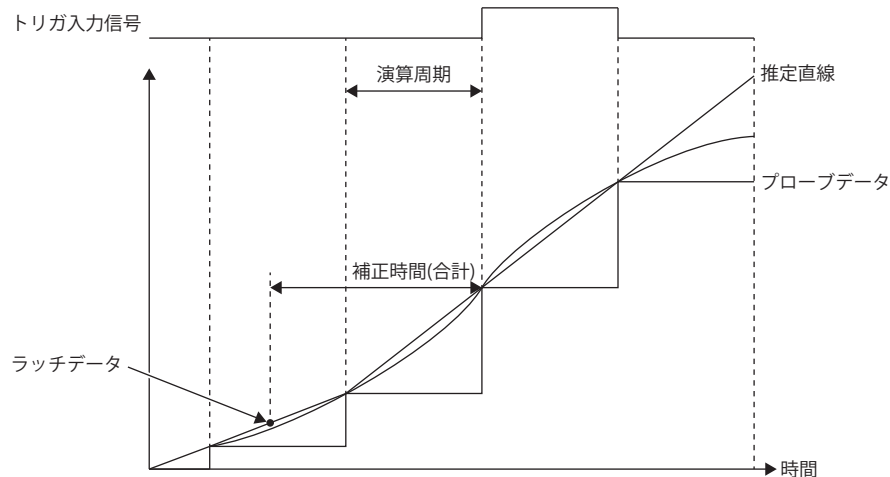
Point

遅れ時間を補正する際、ラッチデータの推定計算に過去値を使用しています。そのため、実行中(Busy)がTRUEのタイミングから補正時間が経過した時点で、精度が最大になります。また補正時間と演算周期により、下記のメモリを消費します。

・メモリ消費量 = 24バイト × (合計の補正時間 ÷ 演算周期)

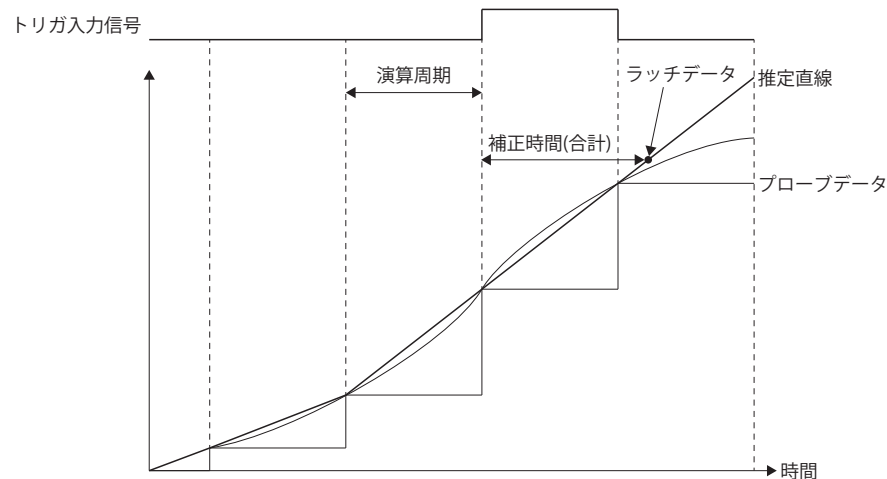
・遅れを補正する場合(合計の補正時間が正の値の場合)

ラッチしたタイミングから補正時間前の演算周期間のデータをもとに推定計算します。



・進みを補正する場合(合計の補正時間が負の値の場合)

ラッチしたタイミングの演算周期間のデータをもとに推定計算します。



■ラッチモード(RecordMode)

データのラッチ方法を設定します。ラッチ方法について下記に示します。

設定値	内容
0: 単発モード(OneShot)	1回のみラッチデータを格納します。
1: 指定回数モード(RecordCount)	- ラッチ回数(RecordCount)の設定値分、ラッチデータを格納します。 - ラッチデータ格納先(OutputBuffer)で指定した格納先へ、指定した格納先を先頭に、ラッチすることにラッチデータサイズ分シフトしたエリアにラッチデータを格納します。 - ラッチ回数(RecordCount)分のラッチが完了前にタッチプローブ機能を停止する場合は、MC_AbortTrigger(タッチプローブ無効)を実行してください。
2: リングバッファモード(RingBuffer)	- ラッチ回数(RecordCount)の設定値分のリングバッファにラッチデータを格納します。常時ラッチ動作を行います。 - ラッチデータ格納先(OutputBuffer)で指定した格納先へ、指定した格納先を先頭に、ラッチすることにラッチデータサイズ分シフトし、ラッチ回数(RecordCount)分のリングバッファにラッチデータを格納します。 - ラッチ回数(RecordCount)分のラッチが完了前にタッチプローブ機能を停止する場合は、MC_AbortTrigger(タッチプローブ無効)を実行してください。

- ラッチモード(RecordMode)の設定により、ラッチ回数(RecordedCounter)、およびラッチ位置(RecordedPosition)には下記の値を格納します。

ラッチモード(RecordMode)	ラッチ回数(RecordedCounter)	ラッチ位置(RecordedPosition)
0: 単発モード(OneShot)	ラッチ動作により「1」になります。	1回のみラッチデータを格納します。
1: 指定回数モード(RecordCount)	ラッチ動作のたびに「+1」します。	ラッチ動作のたびに最新のラッチデータを格納します。
2: リングバッファモード(RingBuffer)	ラッチ動作のたびに「+1」します。 ラッチ回数(RecordCount)以上になった場合、「0」に戻ります。	ラッチ動作のたびに最新のラッチデータを格納します。

■ラッチ回数(RecordCount)

ラッチモード(RecordMode)の設定により、ラッチ回数(RecordCount)には下記の値を設定します。

ラッチモード(RecordMode)	ラッチ回数(RecordCount)
0: 単発モード(OneShot)	無視します。
1: 指定回数モード(RecordCount)	データのラッチ回数を設定します。範囲外に設定した場合は、ラッチ回数不正(エラーコード: 1B8AH)となります。
2: リングバッファモード(RingBuffer)	リングバッファの回数を設定します。範囲外に設定した場合は、ラッチ回数不正(エラーコード: 1B8AH)となります。

■ラッチデータ格納先(OutputBuffer)

ラッチしたデータの格納先をTARGET_REF構造体で設定します。TARGET_REF構造体については、下記を参照してください。

📖 1421ページ TARGET_REF(入力信号)

- 使用できるデータ種別は、[VAR]と[DEV]です。
- ラッチしたデータはLREAL型として格納するため、LREAL型以外のデータをTARGET_REF構造体の対象(Target)に設定する場合は、明示的にデータの型に「(LREAL)」を設定してください。
- ラッチしたデータの格納先を下記に示します。

ラッチモード(RecordMode)	格納先
0: 単発モード(OneShot)	ラッチデータ格納先(OutputBuffer)で指定したエリア
1: 指定回数モード(RecordCount)	下記のエリア
2: リングバッファモード(RingBuffer)	ラッチデータ格納先(OutputBuffer)で設定したエリア + ラッチ回数(RecordedCounter) × ラッチデータサイズ

- ラッチデータ格納先(OutputBuffer)は省略できます。省略した場合は、ラッチ位置(RecordedPosition)にのみラッチデータを格納します。
- ラッチデータ格納先(OutputBuffer)をTARGET_REF構造体の対象(Target)に設定する例を下記に示します。

例

下記設定の場合

例	内容	設定
1	• ラッチモード(RecordMode)が「0: 単発モード(OneShot)」のとき。 • ラッチモード(RecordMode)が「1: 指定回数モード(RecordCount)」, または「2: リングバッファモード(RingBuffer)」でラッチ回数(RecordCount)が「1」のとき。	• [VAR]OutputBufferData1^{*1}
2	• ラッチモード(RecordMode)が「1: 指定回数モード(RecordCount)」, または「2: リングバッファモード(RingBuffer)」で、ラッチ回数(RecordCount)が「100」のとき。	• [VAR]OutputBufferData2^{*2} • [VAR](LREAL)OutputBufferData3^{*3}

*1 「OutputBufferData1」はLREAL型のインスタンス

*2 「OutputBufferData2」はLREAL[0..99]型のインスタンス

*3 「OutputBufferData3」はWORD[0..399]型のインスタンス

- ラッチデータ格納先(OutputBuffer)で設定した要素数がラッチ回数(RecordCount)より少ない場合、ラッチデータ格納先不足(エラーコード: 1BBFH)となります。また、ラッチ動作時にデバイスやラベルの範囲外にアクセスした場合もラッチデータ格納先不足(エラーコード: 1BBFH)となります。
- 各項目の設定値と動作の関係を下記に示します。

ラッチモード (RecordMode)	ラッチ回数 (RecordCount)	ラッチデータ格納先 (OutputBuffer)	検出動作	データ格納先
0: 単発モード(OneShot)	設定範囲内/設定範囲外	省略	単発	ラッチ位置(RecordedPosition)
		設定		ラッチ位置(RecordedPosition)/バッファ
1: 指定回数モード (RecordCount)	設定範囲外	省略/設定	エラー	—
	設定範囲内	省略	設定回数	ラッチ位置(RecordedPosition)
		設定		ラッチ位置(RecordedPosition)/バッファ
2: リングバッファモード (RingBuffer)	設定範囲外	省略/設定	エラー	—
	設定範囲内	省略	常時	ラッチ位置(RecordedPosition)
		設定		ラッチ位置(RecordedPosition)/バッファ

注意事項

- タッチプローブの制御(トリガ入力信号の検出, ラッチ)はFB呼出しタスク(スキャン/定周期/イベント)によらず, 下記の演算周期で行います。トリガ入力信号の検出周期はTriggerInputに設定したデータ種別によって異なります。

種別	周期
[VAR]	第1演算周期
[DEV]	
[CONST]	
[OBJ]	対象局の通信周期
[LINK]	
[VAR]ネットワークラベル指定時	

プローブデータ(ProbeData)に設定したデータは下記の周期でラッチします。

種別	周期
[VAR]	第1演算周期
[DEV]	
[OBJ]	
[LINK]	対象局の通信周期
[VAR]ネットワークラベル指定時	
[AXIS]	

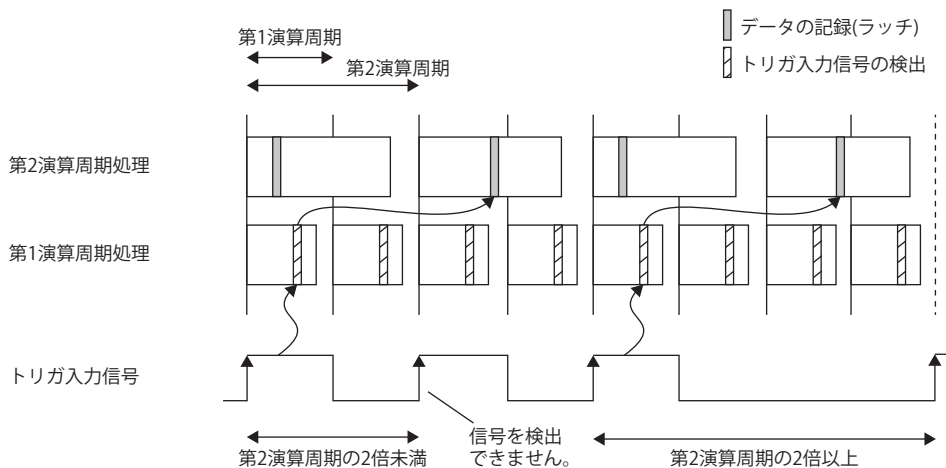
- 対象軸が未接続の場合やエラーとなった場合でも, MC_TouchProbe(タッチプローブ有効)は実行します。
- トリガ入力信号のTRUE, FALSEの各時間は下記より長くしてください。短い時間の場合, 正しく信号を検出できません。

外部信号高精度入力	トリガ入力信号のTRUE, FALSEの各時間
使用する	- デバイス局: デバイス局のフィルタ時間 - 入力ユニット: 入力ユニットの入力応答時間
使用しない	トリガ入力信号の検出周期

- トリガ入力信号の間隔は, トリガ入力信号の検出周期とラッチ周期のいずれか長い方の2倍以上の間隔としてください。それより短い間隔で入力した場合, 正しく信号を検出できません。ラッチするまでに複数トリガ入力信号を入力した場合, 最初に入力した信号を使用します。

例

トリガ入力信号の検出周期は第1演算周期, ラッチ周期は第2演算周期の場合



プログラム例

タッチプローブ起動(bTouchProbe)をTRUEにし、モーションシステム内のタッチプローブシグナル(G_bTouchProbeSignal)の立上りごとに軸1(Axis0001)のフィードバック位置を推定計算し、ラッチ位置(leRecordedPosition)に格納した後、タッチプローブ無効始動(bAbortTrigger)をTRUEにし、ラッチ動作を停止するプログラム例を下記に示します。

■軸

軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■入出力データ

データ名	データ種別	データ型	ソース種別	ソース	信号検出方法
SignalData0001	トリガ信号	MC_TRIGGER_REF	グローバルラベル	G_bTouchProbeSignal	FALSE→TRUE(立上り)時検出(RisingEdge)

■使用するグローバルラベル

ラベル名	データ型	コメント
G_bTouchProbeSignal	ビット	タッチプローブシグナル
G_leLatchPos_array	倍精度実数(0..1)	ラッチ位置格納用変数

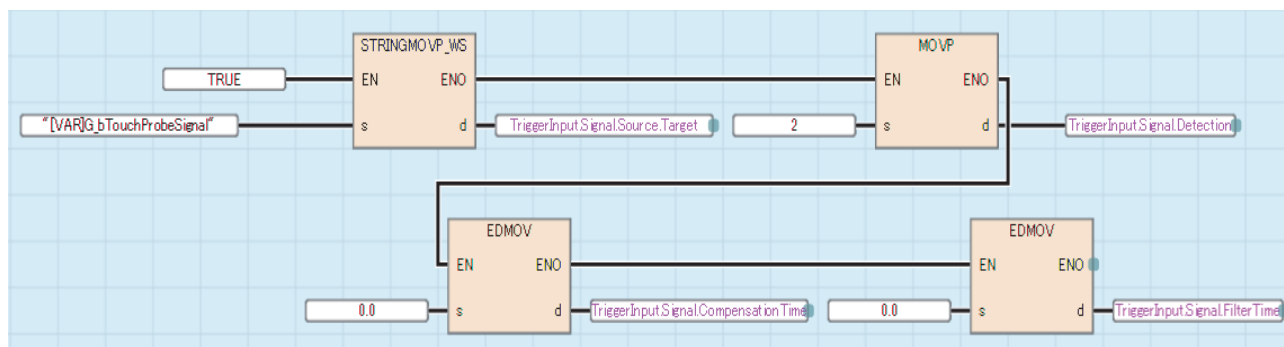
■使用するラベル

ラベル名	データ型	コメント
MC_TouchProbe_1	MC_TouchProbe	タッチプローブ有効FB
bTouchProbe	ビット	タッチプローブ起動
leFirstPosition	倍精度実数	下限位置
leLastPosition	倍精度実数	上限位置
leCompensationTime	倍精度実数	補正時間
ProbeData	TARGET_REF	プローブデータ
uwRecordCount	ワード[符号なし]/ビット列[16ビット]	ラッチ回数
OutputBuffer	TARGET_REF	ラッチデータ格納先
bTPDone	ビット	タッチプローブ有効完了
bTPBusy	ビット	タッチプローブ有効実行中
bTPCommandAborted	ビット	タッチプローブ有効実行中断
bTPError	ビット	タッチプローブ有効エラー
uwTPErrorID	ワード[符号なし]/ビット列[16ビット]	タッチプローブ有効エラーコード
lePositionData	倍精度実数	現在値モニタデータ
leRecordedPosition	倍精度実数	ラッチ位置
uwRecordedCounter	ワード[符号なし]/ビット列[16ビット]	ラッチ回数
uwTouchProbeID	ワード[符号なし]/ビット列[16ビット]	プローブID
MC_AbortTrigger_1	MC_AbortTrigger	タッチプローブ無効FB
bAbortTrigger	ビット	タッチプローブ無効始動
bATDone	ビット	タッチプローブ無効完了
bATBusy	ビット	タッチプローブ無効処理中
bATCommandAborted	ビット	タッチプローブ無効実行中断
bATError	ビット	タッチプローブ無効エラー
uwATErrorID	ワード[符号なし]/ビット列[16ビット]	タッチプローブ無効エラーコード
TriggerInput*1	MC_TRIGGER_REF	トリガ入力信号

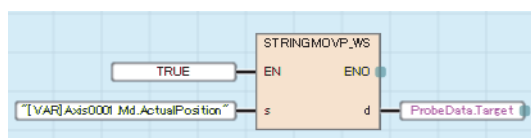
*1 FBD/LDのプログラムのみ使用(入出力データをFBD/LDのプログラムで使用するために必要)

■FBD/LDプログラム

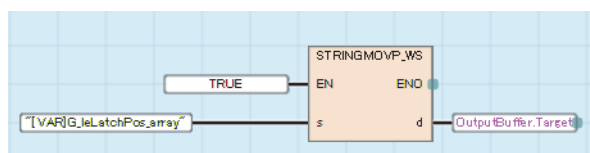
- トリガ入力信号(入出力データ)



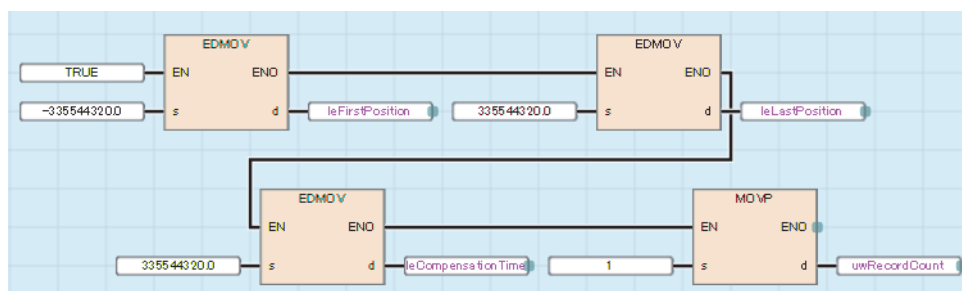
- ラッチ対象用データ



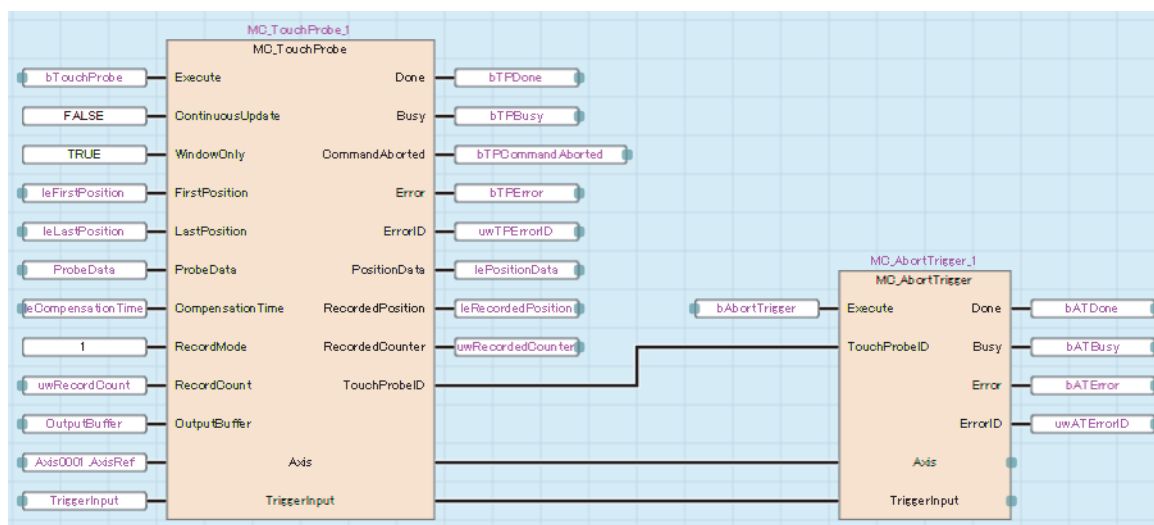
- タッチプローブ処理



- タッチプローブ用データ設定



- タッチプローブ有効/無効



■STプログラム

//-----ラッチ対象用データ-----

```
ProbeData.Target:= "[VAR]Axis0001.Md.ActualPosition";
```

//-----タッチプローブ処理-----

```
OutputBuffer.Target:= "[VAR]G_leLatchPos_array";
```

//-----タッチプローブ用データ設定-----

```
leFirstPosition:= -335544320.0;
```

```
leLastPosition:= 335544320.0;
```

```
leCompensationTime:= 0.0;
```

```
uwRecordCount:= 1;
```

//-----タッチプローブ有効-----

```
MC_TouchProbe_1(
```

```
  Axis:= Axis0001.AxisRef ,
```

```
  TriggerInput:= SignalData0001 ,
```

```
  Execute:= bTouchProbe ,
```

```
  ContinuousUpdate:= FALSE ,
```

```
  WindowOnly:= TRUE ,
```

```
  FirstPosition:= leFirstPosition ,
```

```
  LastPosition:= leLastPosition ,
```

```
  ProbeData:= ProbeData ,
```

```
  CompensationTime:= leCompensationTime ,
```

```
  RecordMode:= MC_RECORD_MODE__RecordCount ,
```

```
  RecordCount:= uwRecordCount ,
```

```
  OutputBuffer:= OutputBuffer ,
```

```
  Done=> bTPDone ,
```

```
  Busy=> bTPBusy ,
```

```
  CommandAborted=> bTPCommandAborted ,
```

```
  Error=> bTPError ,
```

```
  ErrorID=> uwTPErrorID ,
```

```
  PositionData=> lePositionData ,
```

```
  RecordedPosition=> leRecordedPosition ,
```

```
  RecordedCounter=> uwRecordedCounter ,
```

```
  TouchProbeID=> uwTouchProbeID
```

```
);
```

//-----タッチプローブ無効-----

```
MC_AbortTrigger_1(
```

```
  Axis:= Axis0001.AxisRef ,
```

```
  TriggerInput:= SignalData0001 ,
```

```
  Execute:= bAbortTrigger ,
```

```
  TouchProbeID:= MC_TouchProbe_1.TouchProbeID ,
```

```
  Done=> bATDone ,
```

```
  Busy=> bATBusy ,
```

```
  Error=> bAError ,
```

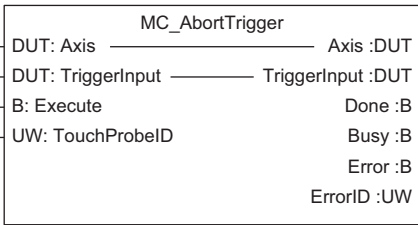
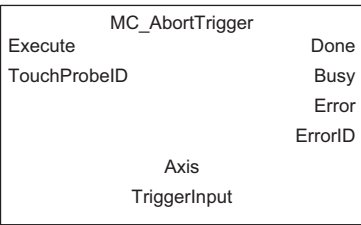
```
  ErrorID=> uwAErrorID
```

```
);
```

41.12 タッチプローブ無効

MC_AbortTrigger

実行中のラッチを無効にします。

ラダー	FBD/LD	ST
		<pre>MC_AbortTrigger(Axis:= ?AXIS_REF?, TriggerInput:= ?MC_TRIGGER_REF?, Execute:= ?BOOL?, TouchProbelID:= ?WORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 1378ページ AxisName.AxisRef.(軸情報)
TriggerInput	トリガ入力信号	MC_TRIGGER_REF	起動時	—	省略可	トリガ入力信号を設定します。 本FBでは設定できません。設定を行った場合、無視して動作します。

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	起動	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_AbortTrigger(タッチプローブ無効)を実行します。
TouchProbelID	タッチプローブID	WORD(UINT)	起動時	0~65535	0	無効にするタッチプローブの固有IDを設定します。 無効にするMC_TouchProbe(タッチプローブ有効)の実行指令(Execute)が立上り後のプローブID(TouchProbelID)に格納するタッチプローブIDを設定します。

■出力変数

出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	制御が完了したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MC_AbortTrigger(タッチプローブ無効)を実行したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 使用するコントローラのユーザーズマニュアル

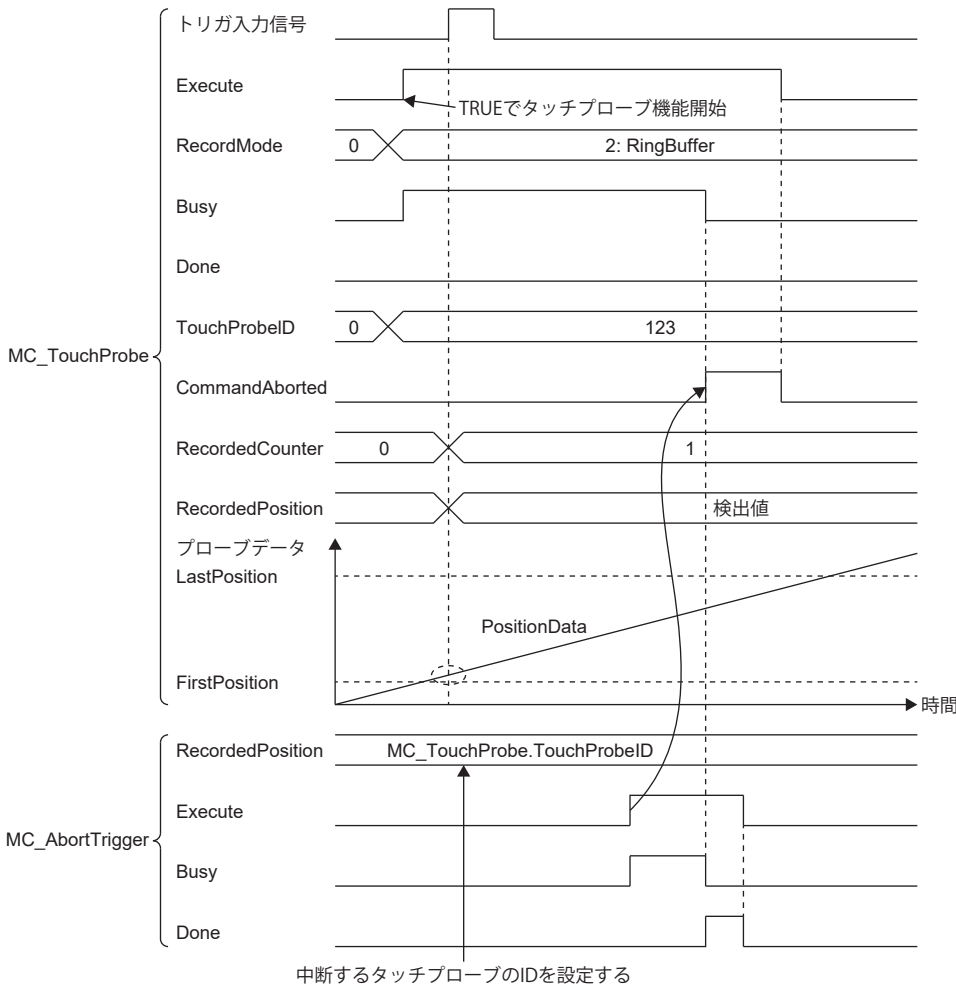
機能

- タッチプローブID(TouchProbelID)で設定したMC_TouchProbe(タッチプローブ有効)を無効にします。
- MC_AbortTrigger(タッチプローブ無効)実行時、タッチプローブID(TouchProbelID)で設定したMC_TouchProbe(タッチプローブ有効)が動作していない場合は、直ちに実行完了(Done)がTRUEになります。
- 軸情報(Axis)は無視するため、省略できます。

■タイミングチャート

・ 正常完了の場合

ラッチモード(RecordMode)が「2: リングバッファモード(RingBuffer)」の場合



・ 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

☞ 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■プログラム例

タッチプローブ起動(bTouchProbe)をTRUEにし、モーションシステム内のタッチプローブシグナル(G_bTouchProbeSignal)の立上りごとに軸1(Axis0001)のフィードバック位置を推定計算し、ラッチ位置(leRecordedPosition)に格納した後、タッチプローブ無効始動(bAbortTrigger)をTRUEにし、ラッチ動作を停止するプログラム例を下記に示します。

■軸

軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■入出力データ

データ名	データ種別	データ型	ソース種別	ソース	信号検出方法
SignalData0001	トリガ信号	MC_TRIGGER_REF	グローバルラベル	G_bTouchProbeSignal	FALSE→TRUE(立上り)時検出(RisingEdge)

■使用するグローバルラベル

ラベル名	データ型	コメント
G_bTouchProbeSignal	ビット	タッチプローブシグナル
G_leLatchPos_array	倍精度実数(0..1)	ラッチ位置格納用変数

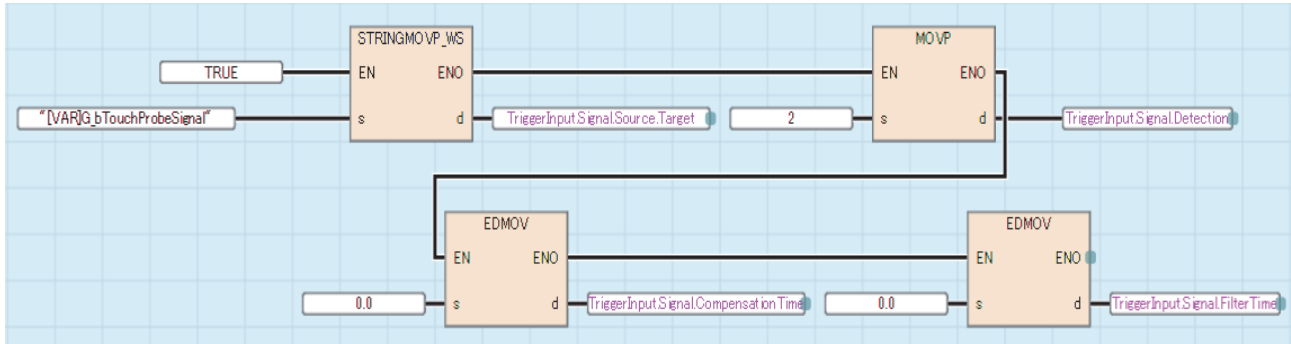
■使用するラベル

ラベル名	データ型	コメント
MC_TouchProbe_1	MC_TouchProbe	タッチプローブ有効FB
bTouchProbe	ビット	タッチプローブ起動
leFirstPosition	倍精度実数	下限位置
leLastPosition	倍精度実数	上限位置
leCompensationTime	倍精度実数	補正時間
ProbeData	TARGET_REF	プローブデータ
uwRecordCount	ワード[符号なし]/ビット列[16ビット]	ラッチ回数
OutputBuffer	TARGET_REF	ラッチデータ格納先
bTPDone	ビット	タッチプローブ有効完了
bTPBusy	ビット	タッチプローブ有効実行中
bTPCommandAborted	ビット	タッチプローブ有効実行中断
bTPError	ビット	タッチプローブ有効エラー
uwTPErrorID	ワード[符号なし]/ビット列[16ビット]	タッチプローブ有効エラーコード
lePositionData	倍精度実数	現在値モニタデータ
leRecordedPosition	倍精度実数	ラッチ位置
uwRecordedCounter	ワード[符号なし]/ビット列[16ビット]	ラッチ回数
uwTouchProbeID	ワード[符号なし]/ビット列[16ビット]	プローブID
MC_AbortTrigger_1	MC_AbortTrigger	タッチプローブ無効FB
bAbortTrigger	ビット	タッチプローブ無効始動
bATDone	ビット	タッチプローブ無効完了
bATBusy	ビット	タッチプローブ無効処理中
bATCommandAborted	ビット	タッチプローブ無効実行中断
bATError	ビット	タッチプローブ無効エラー
uwATErrorID	ワード[符号なし]/ビット列[16ビット]	タッチプローブ無効エラーコード
TriggerInput ^{*1}	MC_TRIGGER_REF	トリガ入力信号

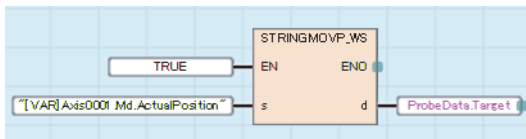
*1 FBD/LDのプログラムのみ使用(入出力データをFBD/LDのプログラムで使用するために必要)

■FBD/LDプログラム

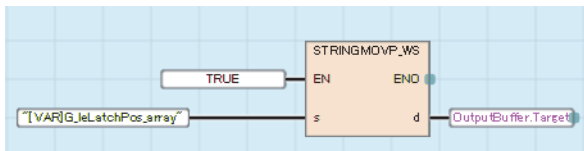
- トリガ入力信号(入出力データ)



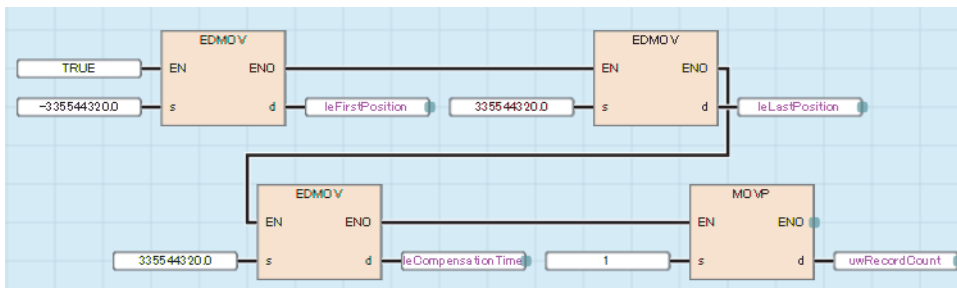
- ラッチ対象用データ



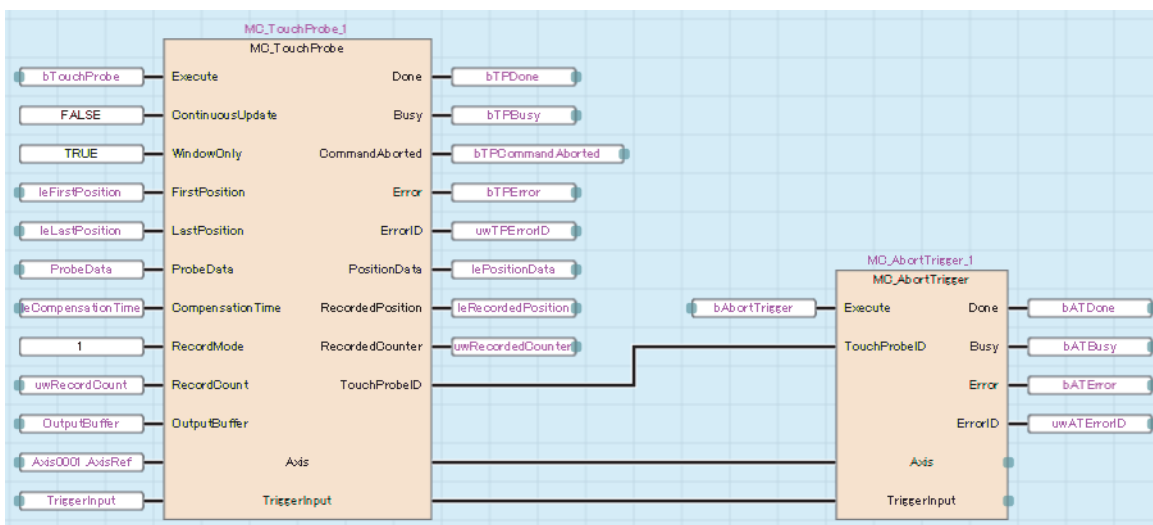
- タッチプローブ処理



- タッチプローブ用データ設定



- タッチプローブ有効/無効



■STプログラム

//-----ラッチ対象用データ-----

ProbeData.Target:= "[VAR]Axis0001.Md.ActualPosition";

//-----タッチプローブ処理-----

OutputBuffer.Target:= "[VAR]G_leLatchPos_array";

//-----タッチプローブ用データ設定-----

leFirstPosition:= -335544320.0;

leLastPosition:= 335544320.0;

leCompensationTime:= 0.0;

uwRecordCount:= 1;

//-----タッチプローブ有効-----

MC_TouchProbe_1(

Axis:= Axis0001.AxisRef ,

TriggerInput:= SignalData0001 ,

Execute:= bTouchProbe ,

ContinuousUpdate:= FALSE ,

WindowOnly:= TRUE ,

FirstPosition:= leFirstPosition ,

LastPosition:= leLastPosition ,

ProbeData:= ProbeData ,

CompensationTime:= leCompensationTime ,

RecordMode:= MC_RECORD_MODE__RecordCount ,

RecordCount:= uwRecordCount ,

OutputBuffer:= OutputBuffer ,

Done=> bTPDone ,

Busy=> bTPBusy ,

CommandAborted=> bTPCommandAborted ,

Error=> bTPError ,

ErrorID=> uwTPErrorID ,

PositionData=> lePositionData ,

RecordedPosition=> leRecordedPosition ,

RecordedCounter=> uwRecordedCounter ,

TouchProbeID=> uwTouchProbeID

);

//-----タッチプローブ無効-----

MC_AbortTrigger_1(

Axis:= Axis0001.AxisRef ,

TriggerInput:= SignalData0001 ,

Execute:= bAbortTrigger ,

TouchProbeID:= MC_TouchProbe_1.TouchProbeID ,

Done=> bATDone ,

Busy=> bATBusy ,

Error=> bAError ,

ErrorID=> uwAErrorID

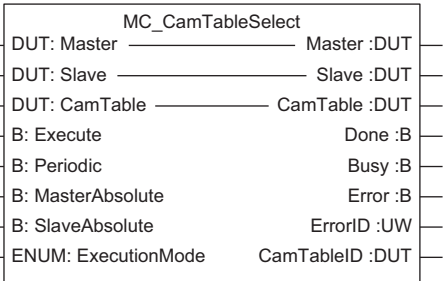
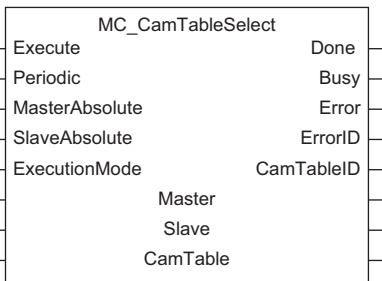
);

41.13 カムテーブル選択

41

MC_CamTableSelect

指定した演算プロファイル(カムデータ)を、展開エリアに格納します。

ラダー	FBD/LD	ST
		<pre>MC_CamTableSelect(Master:= ?AXIS_REF?, Slave:= ?AXIS_REF?, CamTable:= ?MC_CAM_REF?, Execute:= ?BOOL?, Periodic:= ?BOOL?, MasterAbsolute:= ?BOOL?, SlaveAbsolute:= ?BOOL?, ExecutionMode:= ?INT?, Done=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?, CamTableID=> ?MC_CAM_ID?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Master	主軸	AXIS_REF	起動時	—	省略可	設定値は無視します。
Slave	従軸	AXIS_REF	起動時	—	省略可	設定値は無視します。
CamTable	カムテーブル	MC_CAM_REF	起動時	—	省略不可	演算プロファイル(カムデータ)を設定します。 MC_CamTableSelect(カムテーブル選択)を実行時、 プロファイルID(MC_CAM_ID.ProfileID)は設定不要 です。 PROFILE_DATA.ID(MC_CAM_REF.ProfileData.ID)の 設定値を参照します。ただし、入出力 No.(CamTable.ProfileData.ID.StartIO)の設定は不要 です。 詳細は、下記を参照してください。 📖 1530ページ カムテーブル(CamTable)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	起動	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_CamTableSelect(カムテーブル選択)を実行します。
Periodic	繰り返し動作	BOOL	起動時	TRUE, FALSE	FALSE	演算プロファイル(カムデータ)を実行する動作を設定します。 • FALSE: 単発動作 • TRUE: 繰り返し動作 詳細は、下記を参照してください。 1530ページ 繰り返し動作(Periodic)
MasterAbsolute	主軸絶対座標	BOOL	起動時	FALSE	FALSE	主軸の座標を設定します。 • FALSE: 相対座標 • TRUE: 設定不可 詳細は、下記を参照してください。 1531ページ 主軸絶対座標(MasterAbsolute) *: 「TRUE」を設定すると、MasterAbsolute範囲外(エラーコード: 1B94H)となります。
SlaveAbsolute	従軸絶対座標	BOOL	起動時	TRUE, FALSE	FALSE	従軸の座標を設定します。 • FALSE: 相対座標 • TRUE: 絶対座標 詳細は、下記を参照してください。 1531ページ 従軸絶対座標(SlaveAbsolute)
ExecutionMode	起動モード	INT (MC_EXECUTION_MODE)	起動時	0, 1, 3	0	MC_CamTableSelect(カムテーブル選択)を実行するタイミングを設定します。 • 0: 直ちに実行(mcImmediately) • 1: 完了待ちして実行(mcQueued) • 3: 投機的に実行(mcSpeculatively) 詳細は、下記を参照してください。 1532ページ 起動モード(ExecutionMode)

■出力変数

出力変数	名称	データ型	初期値	内容
Done	完了	BOOL	FALSE	制御が完了したときに、TRUEになります。 動作完了時に起動(Execute)の状態により下記となります。 ■起動(Execute)がTRUEの場合 起動(Execute)をFALSEにするまでTRUEのままになります。 ■起動(Execute)がFALSEの場合 1周期だけTRUEになります。
Busy	実行中	BOOL	FALSE	MC_CamTableSelect(カムテーブル選択)を実行したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 使用するコントローラのユーザーズマニュアル
CamTableID	カムテーブルID	MC_CAM_ID	0	プロファイルIDを出力します。

機能

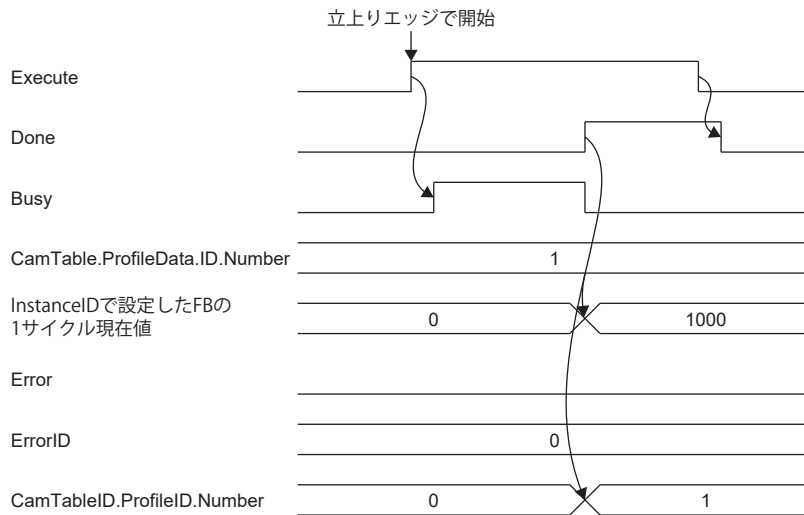
- MC_CamTableSelect(カムテーブル選択)を使用して、PROFILE_DATA構造体の演算プロファイル格納場所(Location)で設定する演算プロファイル(カムデータ)のファイルを展開エリアに展開します。
- 展開エリアに展開したデータには、PROFILE_ID構造体のプロファイルID(Number)を割り付けます。

Point

- 演算プロファイル(カムデータ)の作成についての詳細は、下記を参照してください。
 使用するコントローラのユーザーズマニュアル
- MC_CamTableSelect(カムテーブル選択)は、モーションサービス処理で行います。処理内容や演算プロファイルによっては、展開に時間がかかる場合があります。
- MC_CamTableSelect(カムテーブル選択)の入力変数として、PROFILE_DATA構造体のプロファイルID(ID)、繰り返し動作(Periodic)、主軸絶対座標(MasterAbsolute)、従軸絶対座標(SlaveAbsolute)がある場合、ファイルの設定値を無視し、入力変数の設定を参照して展開を行います。(ファイルの設定値を上書きすることはありません。)

■タイミングチャート

- 正常完了の場合



- 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■カムテーブル(CamTable)

演算プロファイルをMC_CAM_REF型のPROFILE_DATA構造体で設定します。PROFILE_DATA構造体については、下記を参照してください。

📖 1424ページ PROFILE_DATA

PROFILE_DATA構造体の演算プロファイル格納場所(Location)を参照して、ファイル名称(File Name)とフォルダ指定(Path)が示すファイルの演算プロファイルに対し展開を行います。

展開エリアには、PROFILE_DATA構造体のプロファイルID(ID)が示すPROFILE_ID構造体のプロファイルID番号(Number)を割り振ります。

プロファイルID番号(Number)に設定した値により下記の動作となります。

プロファイルID番号(Number)	動作
「0」を設定した場合	自動で空きIDを割り付け、プロファイルID番号(Number)にIDを格納します。 同一の演算プロファイルに対し、プロファイルID番号(Number)が「0」のMC_CamTableSelect(カムテーブル選択)を複数回実行すると、複数の異なる展開エリアに展開します。 空きIDがない場合、演算プロファイルID不足(エラーコード: 1B A0H)となり、展開処理は行いません。
「1~60000」を設定した場合	設定したIDを変えずに展開を行います。 すでに展開済みのIDを指定した場合は展開データを上書きします。

展開時のプロファイルID番号(Number)の値と展開先エリアの関係を下記に示します。

プロファイルID番号(Number)の値		展開先エリア
MC_CamTableSelect(カムテーブル選択) 実行前	MC_CamTableSelect(カムテーブル選択) 実行後	
0	1~60000を格納	新規
1~60000		すでに存在する場合は、展開したデータを上書きします。 存在しない場合は、新規に展開エリアを割り当てます。

- 機能固有型の場合、機能固有型のプロファイルID(MC_CAM_ID.ProfileIDなど)に、PROFILE_DATA.ID(MC_CAM_REF.ProfileData.IDなど)と同じ値を格納します。演算プロファイル制御FB(MC_CamInなど)の入力変数には、機能固有型のプロファイルIDを設定してください。

プロファイルID型の詳細については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

- 展開FB実行時、MC_CAM_REF.CamID.ProfileIDは設定不要です。MC_CAM_REF.ProfileData.IDの設定値を参照します。

展開FBの一覧については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

- 演算プロファイル格納場所(Location)で設定した演算プロファイルが存在しない場合、演算プロファイル無し(エラーコード: 1B 8EH)となり、展開処理を行いません。
- 演算プロファイル格納場所(Location)で設定した演算プロファイルが破損している、またはフォーマットに問題がある場合、演算プロファイル異常(エラーコード: 1B 91H)となり、展開処理を行いません。
- 演算プロファイル格納場所(Location)で指定した演算プロファイルの設定値が範囲外の場合、演算プロファイル不正(エラーコード: 1B 8FH)が発生し、展開処理は行われません。

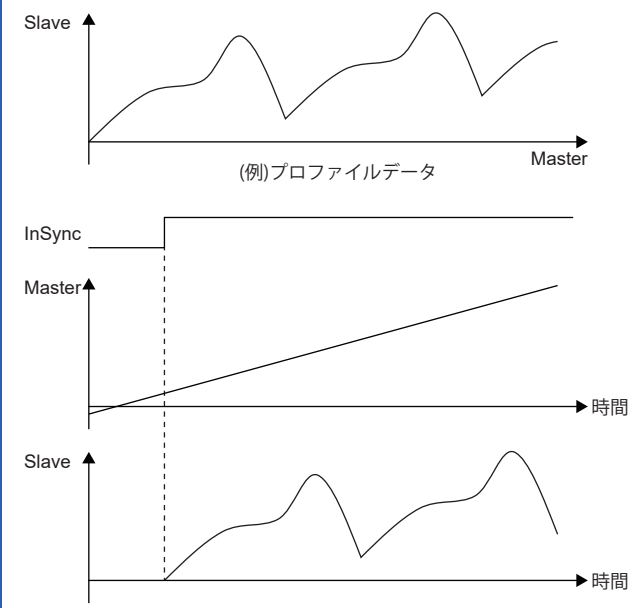
■繰り返し動作(Periodic)

繰り返し動作(Periodic)による、演算プロファイル制御実行時の動作を設定します。

設定値	内容
FALSE(単発動作)	演算プロファイルの終点まで実行すると、制御を終了します。
TRUE(繰り返し動作)	連続的に演算プロファイルの実行を繰り返します。

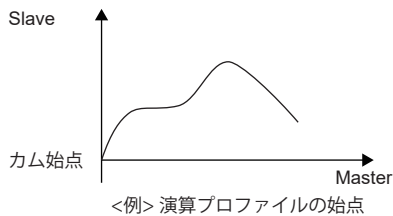
■主軸絶対座標(MasterAbsolute)

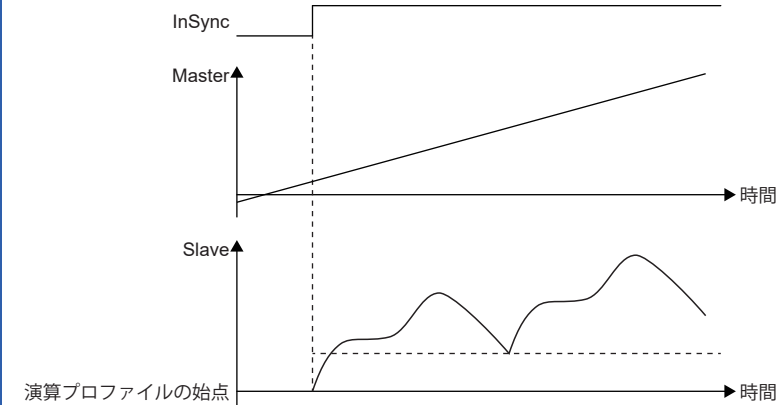
主軸絶対座標(MasterAbsolute)による、演算プロファイル制御実行時の動作を設定します。(入力単位が時間指定の場合は無視します。)

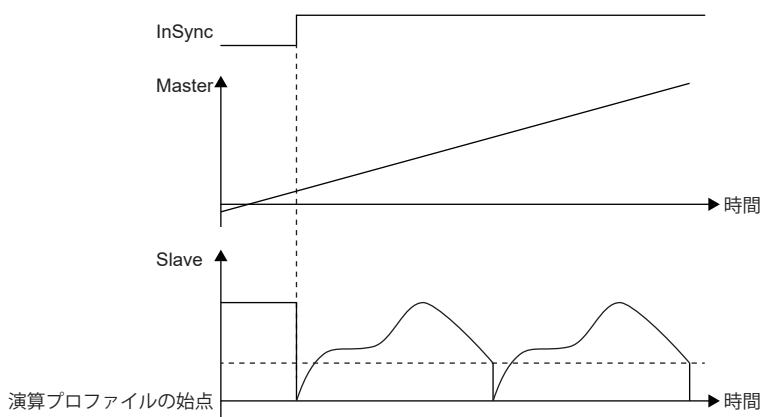
設定値	内容
FALSE(相対座標)	現在の主軸(Master)の値が演算プロファイルの始点となります。同期中(InSync)がTRUEになると、主軸(Master)の相対移動量に応じて従軸(Slave)を動作します。 
TRUE	TRUEは設定できません。 TRUEを設定するとMasterAbsolute範囲外(エラーコード: 1B94H)となります。

■従軸絶対座標(SlaveAbsolute)

従軸絶対座標(SlaveAbsolute)による、演算プロファイル制御実行時の動作を設定します。



設定値	内容
FALSE(相対座標)	同期中(InSync)がTRUEになると、現在の値を基準として従軸(Slave)を動作します。 

設定値	内容
TRUE(絶対座標)	同期中(InSync)がTRUEになった時点の従軸(Slave)が、演算プロファイルの1サイクル開始時に常に始点となるよう動作します。 カムデータの1サイクル長など演算プロファイルの主軸(Master)の値に制限があり、繰り返し動作(Periodic)が「TRUE: 繰り返し動作」で演算プロファイルの始点と終点で異なる場合は、次の1サイクル開始時点で最初の従軸(Slave)に戻るよう指令を1演算周期で出力します。  演算プロファイルの始点

■起動モード(ExecutionMode)

MC_CamTableSelect(カムテーブル選択)を実行するタイミングを設定します。

設定値	内容
0: 直ちに実行(mclmmediately)	展開エリアの内容を即座に書き込みます。 実行中の制御に影響を及ぼす可能性があります。 ただし、FBの実行中に展開を行う場合は、演算プロファイルの形式、分解能が一致していないと、演算プロファイル操作中(エラーコード: 1B90H)となります。
1: 完了待ちして実行(mcQueued)	実行中のFBの実行完了を待ち展開します。(Done = TRUE) 複数のFBが待ち状態となった場合、次に実行するFBは、優先度の高いタスクで実行しているFBとなります。 同じ優先度の場合は、起動順に実行します。
3: 投機的に実行(mcSpeculatively)	演算プロファイル操作中(エラーコード: 1B90H)となり、展開エリアは変更しません。

Point

実行中のFBがない場合は、起動モード(ExecutionMode)の設定によらず即座に実行します。

展開エリアにアクセスする各FB実行中は、各FBの実行中(Busy)がTRUEになります。必要に応じて起動モード(ExecutionMode)をインタロックに使用してください。

注意事項

- 制御中に展開エリアを書き換える場合、波形パターンによっては、ストローク値が急激に変化し、機械に衝撃を与える場合があります。変更前後の波形が不連続とならないよう、演算プロファイルを作成してください。

プログラム例

カムテーブル選択指令(bCamTableSelect)をTRUEにし、演算プロファイル(カムデータ1)を展開エリアに展開して、プロファイルIDを割り付けるプログラム例を下記に示します。

■軸

軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■演算プロファイル

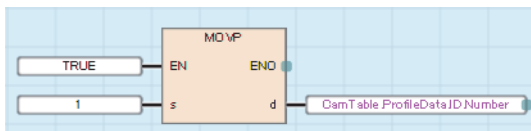
プロファイルID	ラベル名	データ型	コメント
1	ProfileData0001	MC_CAM_REF	カムデータ1

■使用するラベル

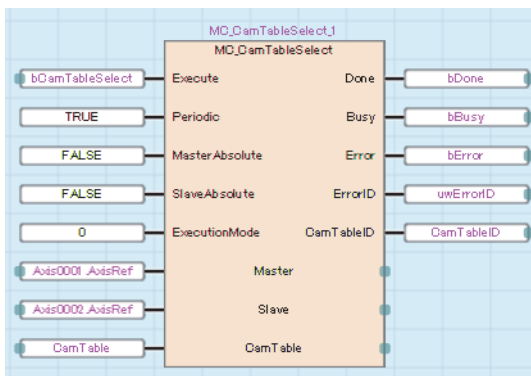
ラベル名	データ型	コメント
MC_CamTableSelect_1	MC_CamTableSelect	カムテーブル選択FB
CamTable	MC_CAM_REF	カムテーブル
bCamTableSelect	ビット	カムテーブル選択指令
bDone	ビット	実行完了
bBusy	ビット	実行中
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
CamTableID	MC_CAM_ID	カムテーブルID

■FBD/LDプログラム

- ・演算プロファイル(カムデータ)設定



- ・カムテーブル選択



■STプログラム

//-----演算プロファイル(カムデータ)設定-----

CamTable.ProfileData:= ProfileData0001.ProfileData;

//-----カムテーブル選択-----

```
MC_CamTableSelect_1(
  CamTable:= CamTable ,
  Execute:= bCamTableSelect ,
  Periodic:= TRUE ,
  MasterAbsolute:= FALSE ,
  SlaveAbsolute:= FALSE ,
  ExecutionMode:= MC_EXECUTION_MODE__mclmmediately ,
  Done=> bDone ,
  Busy=> bBusy ,
  Error=> bError ,
  ErrorID=> uwErrorID ,
  CamTableID=> CamTableID
);
```

41.14 1サイクル現在値変更

MCv_ChangeCycle

指定した演算プロファイル制御FBの1サイクル現在値を変更します。

ラダー	FBD/LD	ST
		<pre>MCv_ChangeCycle(Execute:= ?BOOL?, InstanceID:= ?INSTANCE_ID?, Cycle:= ?LREAL(0..2)?, Relative:= ?BOOL?, ExecutionMode:= ?INT?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?, CancelAccepted=> ?BOOL?);</pre>

設定データ

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	起動	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMCv_ChangeCycle(1サイクル現在値変更)を実行します。
InstanceID	インスタンスID	INSTANCE_ID	起動時	1~65535	0	1サイクル現在値を変更する演算プロファイル制御FBのインスタンスIDを設定します。 インスタンスIDは、"シーケンサレディ (MotionSystem.Cd.SequenceReady)"ON →OFFまで有効です。 MCv_ChangeCycle(1サイクル現在値変更)を使用する場合は、入出力番号(StartIO)を設定します。
Cycle	1サイクル現在値	LREAL[0..2]	起動時	—	0.0	変更する1サイクル現在値を設定します。 ・ 相対選択(Relative)がFALSEのとき、絶対位置を設定します。 ・ 相対選択(Relative)がTRUEのとき、相対距離を設定します。 詳細は、下記を参照してください。 ☞ 1536ページ 1サイクル現在値(Cycle)
Relative	相対選択	BOOL	起動時	TRUE, FALSE	FALSE	変更する1サイクル現在値の方法を設定します。 ・ FALSE: 絶対位置 ・ TRUE: 相対距離 詳細は、下記を参照してください。 ☞ 1537ページ 相対選択(Relative)
ExecutionMode	起動モード	INT (MC_EXECUTION_MODE)	起動時	0	0	MCv_ChangeCycle(1サイクル現在値変更)を実行するタイミングを設定します。 ・ 0: 直ちに実行(mclImmediately) 詳細は、下記を参照してください。 ☞ 1538ページ 起動モード (ExecutionMode)
Options	オプション	DWORD(HEX)	起動時	00000000H~00010000H	00000000H	MCv_ChangeCycle(1サイクル現在値変更)の機能オプションをビット指定で設定します。 詳細は、下記を参照してください。 ☞ 1539ページ オプション(Options)

■出力変数

出力変数	名称	データ型	初期値	内容
Done	完了	BOOL	FALSE	制御が完了したときに、TRUEになります。 動作完了時に起動(Execute)の状態により下記となります。 ■起動(Execute)がTRUEの場合 起動(Execute)をFALSEにするまでTRUEのままになります。 ■起動(Execute)がFALSEの場合 1周期だけTRUEになります。
Busy	実行中	BOOL	FALSE	MCv_ChangeCycle(1サイクル現在値変更)を実行したときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	エラーや多重起動などにより、MCv_ChangeCycle(1サイクル現在値変更)の実行が中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 □使用するコントローラのユーザーズマニュアル
CancelAccepted	キャンセル受付	BOOL	FALSE	MCv_ChangeCycle(1サイクル現在値変更)がキャンセルを受け取ったときにTRUEになります。

機能

- ・演算プロファイル制御FBの1サイクル現在値を制御実行時に設定した値に変更します。
- ・インスタンスID(InstanceID)に、演算プロファイル制御FB(MC_CamInなど)の公開変数のインスタンスID(InstanceID)を設定します。
- ・1サイクル現在値変更を行うと演算プロファイル制御FB(MC_CamInなど)の公開変数は下記のようにになります。

公開変数	更新値	備考
1サイクル現在値(InputPerCycle)	1サイクル現在値(Cycle)に設定した値	—
基準値(Reference)	1サイクル現在値(Cycle)に相当するストローク値	出力値(OutputData)を固定するために、基準値(Reference)を更新します。
出力値(OutputData)	更新しない	—

Point

MC_CamTableSelect(カムテーブル選択)の従軸絶対座標(SlaveAbsolute)がTRUE(絶対座標)の場合でも、1サイクル現在値変更を行った場合は、出力値(OutputData)を固定するために基準値(Reference)を更新します。

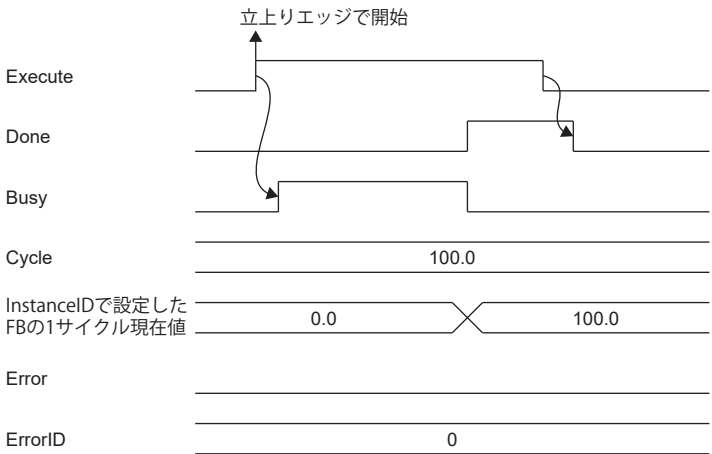
- ・1サイクル現在値変更には、下記の方法があります。

方法	説明
MCv_ChangeCycle(1サイクル現在値変更)	制御実行時に指定した値に変更します。

- ・変更する1サイクル現在値の指定には、相対選択(Relative)により「絶対指定」か「相対指定」を選択できます。
- ・MC_CamIn(カム動作開始)以外の演算プロファイル制御FBに1サイクル現在値変更を行うと、1サイクル現在値変更未対応(エラーコード: 1BAFH)が発生します。

■タイミングチャート

・ 正常完了の場合



・ 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■1サイクル現在値(Cycle)

変更する1サイクル現在値を設定します。

1サイクル現在値(Cycle)で設定可能な値は、相対選択(Relative)の設定により異なります。下記の設定範囲内で設定してください。範囲外の値を設定した場合は、MCv_ChangeCycle命令異常(エラーコード: 1BB9H(Cycleが範囲外(詳細コード: 1)))となります。

1サイクル現在値(Cycle)	相対選択(Relative)の設定値	1サイクル現在値(Cycle)の設定範囲
Cycle[0]	FALSE(絶対位置)	$0.0 \leq \text{設定値} < 1\text{サイクル現在値}$
	TRUE(相対距離)	$-(1\text{サイクル長})/2 \leq \text{設定値} \leq (1\text{サイクル長})/2$
Cycle[1], Cycle[2]	FALSE(絶対位置)	0.0
	TRUE(相対距離)	

■相対選択(Relative)

相対選択(Relative)の設定により、下記のように1サイクル現在値(Cycle)の値を変更します。

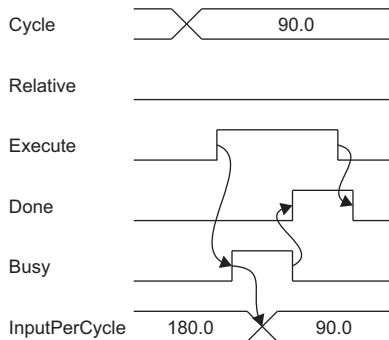
相対選択(Relative)の設定値	内容
FALSE(絶対位置)	1サイクル現在値(InputPerCycle) ^{*1} を、現在の1サイクル現在値(InputPerCycle) ^{*1} に1サイクル現在値(Cycle)を加算した値に変更します。
TRUE(相対距離)	1サイクル現在値(InputPerCycle) ^{*1} を、1サイクル現在値(Cycle)の値に変更します。

^{*1} 演算プロファイル制御FB(MC_CamInなど)の公開変数

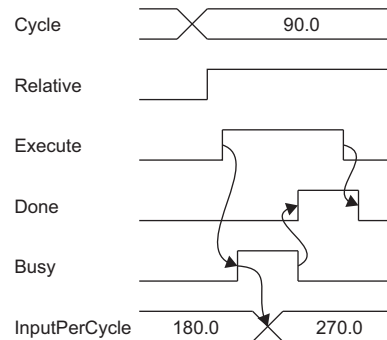
例

公開変数の1サイクル現在値(InputPerCycle)が「180.0」で停止時、1サイクル現在値(Cycle)に「90.0」の1サイクル現在値変更を設定した場合

・相対選択(Relative)がFALSE(絶対位置)のとき



・相対選択(Relative)がTRUE(相対距離)のとき



1サイクル現在値変更による1サイクル現在値の山越え可否を下記に示します。

○: 山越え可能, ×: 山越え不可能

相対選択(Relative)の設定値	1サイクル長と1サイクル現在値の移動量の関係	山越え可否
FALSE(絶対位置)	1サイクル長 > 1サイクル現在値の移動量	×
TRUE(相対距離)	1サイクル長 > 1サイクル現在値の移動量	○
	1サイクル長 ≤ 1サイクル現在値の移動量	○ ^{*1}

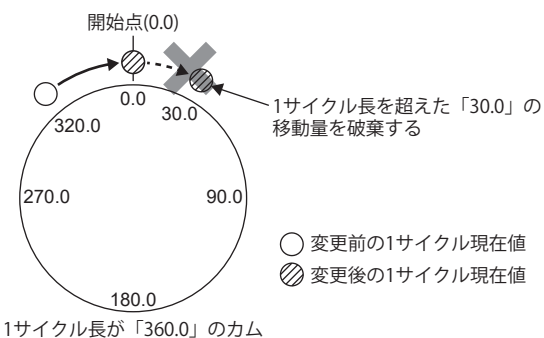
^{*1} MC_CamTableSelect(カムテーブル選択)で設定した繰り返し動作(Periodic)の設定値により、山越え時の動作が異なります。

繰り返し動作(Periodic)の設定値	内容
FALSE(単発動作)	1サイクル長を超える移動量は破棄します。
TRUE(繰り返し動作)	1サイクル長を超えて1サイクル現在値を変更します。

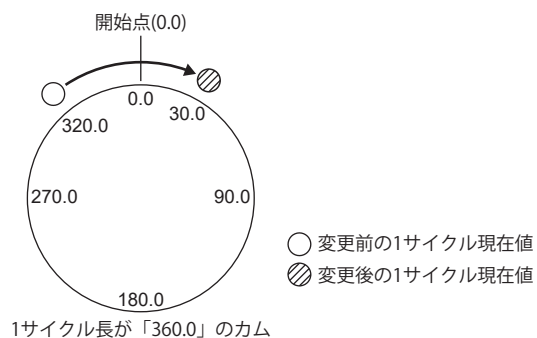
例

1サイクル現在値(Cycle)を「320.0」から「30.0」に変更した場合

・繰り返し動作(Periodic)がFALSE(単発動作)のとき



・繰り返し動作(Periodic)がTRUE(繰り返し動作)のとき



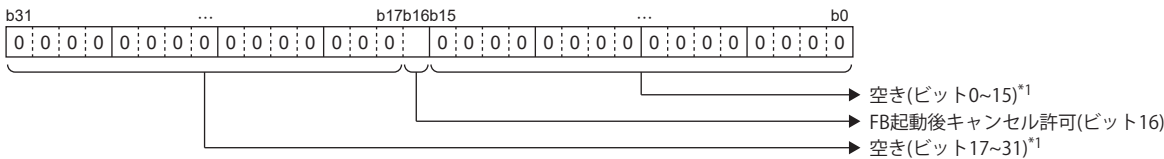
■起動モード(ExecutionMode)

MCv_ChangeCycle(1サイクル現在値変更)を実行した場合、1サイクル現在値を変更するタイミングは設定した起動モード(ExecutionMode)により異なります。起動モード(ExecutionMode)のタイミングについて下記に示します。

設定値	内容
0: 直ちに実行 (mclmmediately)	演算プロファイル制御FB(MC_CamInなどの)実行中(Busy)がTRUEであっても、起動(Execute)の立上り検出により即時実行します。 <例> MC_CamIn(カム動作開始)の実行中(Busy)がTRUEのとき、1サイクル現在値(Cycle)に「100.0」を設定し、1サイクル現在値変更を実行した場合(MC_CamIn(カム動作開始)の主軸(Master)の現在値は変化しないものとします。) MC_CamIn - Execute - Busy - Active - InSync - EndOfProfile - InputPerCycle: 0.0^{*1} → 100.0 - Reference: 0.0^{*1} → -100.0 - OutputData: 0.0^{*1} → (変化しない) - InstanceID: 0 → 1000 - 従軸の位置: 0.0^{*1} → (変化しない) - 従軸の状態 (AxisName.Md.AxisStatus): 4: Standstill → 7: SynchronizedMotion MCv_ChangeCycle - Execute - Done - Busy - CancelAccepted - CommandAborted - Cycle: 100.0 - InstanceID: 0 → 1000^{*3} - Error - ErrorID: 0 *1: 主軸が動かないため変化しません。 *2: 1サイクル現在値変更完了時に完了(Done)がTRUE, 実行中(Busy)がFALSEになります。 *3: 1サイクル現在値変更を実行するFBのインスタンスIDを手動で設定します。

■オプション(Options)

MCv_ChangeCycle(1サイクル現在値変更)で使用する機能オプションをビット指定で設定します。
ビット指定で設定する内容を下記に示します。



*1 空きには「0」を設定してください。「0」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

ビット	名称	内容
16	FB起動後キャンセル許可	MCv_ChangeCycle(1サイクル現在値変更)起動後に1サイクル現在値変更のキャンセルを許可するか、許可しないかを設定します。 • 0: 許可しない • 1: 許可する

- FB起動後キャンセル許可の設定により、FB実行後に待機状態となっている1サイクル現在値変更をキャンセルできます。
 - 起動(Execute)の立下り検出により、キャンセルを開始します。
 - キャンセルの受け付けは、実行中(Busy)がTRUEのときのみ行います。
 - FBがキャンセルを受け取ると、キャンセル受付(CancelAccepted)がTRUEになります。
 - キャンセルが完了すると、実行中断(CommandAborted)がTRUEになります。
 - キャンセルした場合は、1サイクル現在値(InputPerCycle)を変更しません。

プログラム例

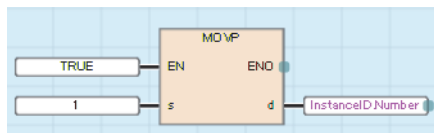
1サイクル現在値変更指令(bChangeCycle)をTRUEにし、インスタンスID(InstanceID)が「1」の1サイクル現在値を「90.0」に変更するプログラム例を下記に示します。

■使用するラベル

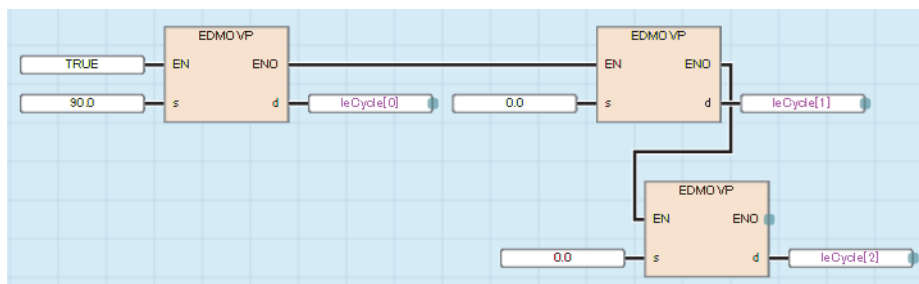
ラベル名	データ型	コメント
MCv_ChangeCycle_1	MCv_ChangeCycle	1サイクル現在値変更FB
bChangeCycle	ビット	1サイクル現在値変更指令
InstanceID	INSTANCE_ID	インスタンスID
leCycle	倍精度実数(0..2)	1サイクル現在値
bDone	ビット	実行完了
bBusy	ビット	実行中
bCommandAborted	ビット	実行中断
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
bCancelAccepted	ビット	キャンセル受付

■FBD/LDプログラム

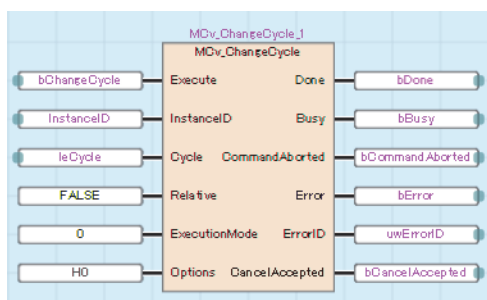
- ・演算プロファイル設定



- ・1サイクル現在値変更用データ設定



- ・1サイクル現在値変更



■STプログラム

//-----演算プロファイル設定-----

InstanceID.Number:= 1;

//-----1サイクル現在値変更用データ設定-----

leCycle[0]:= 90.0;

leCycle[1]:= 0.0;

leCycle[2]:= 0.0;

//-----1サイクル現在値変更-----

MCv_ChangeCycle_1(

Execute:= bChangeCycle ,

InstanceID:= InstanceID ,

Cycle:= leCycle ,

Relative:= FALSE ,

ExecutionMode:= MC_EXECUTION_MODE__mclImmediately ,

Options:= H00000000 ,

Done=> bDone ,

Busy=> bBusy ,

CommandAborted=> bCommandAborted ,

Error=> bError ,

ErrorID=> uwErrorID ,

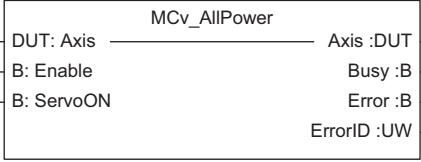
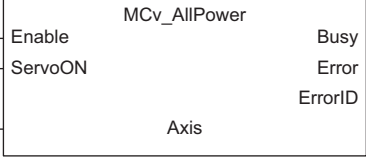
CancelAccepted=> bCancelAccepted

);

41.15 全軸運転可

MCv_AllPower

すべての軸を運転可能状態に切り換えます。

ラダー	FBD/LD	ST
		<pre>MCv_AllPower(Axis:= ?AXIS_REF?, Enable:= ?BOOL?, ServoON:= ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略可	MCv_AllPower(全軸運転可)を使用する場合は、入出力No.(StartIO)を設定します。ただし、設定した値は無視されます。また、軸No.(AxisNo)も無視されます。

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Enable	有効	BOOL	常時	TRUE, FALSE	FALSE	TRUEにすると軸制御が有効となり、運転可能状態となります。FALSEにすると軸制御が無効となり、運転可能状態を解除します。
ServoON	サーボON要求	BOOL	常時	TRUE, FALSE	FALSE	TRUEで軸のサーボONを要求します。

■出力変数

出力変数	名称	データ型	初期値	内容
Busy	実行中	BOOL	FALSE	MCv_AllPower(全軸運転可)を実行したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。エラーコードの詳細は、下記を参照してください。  使用するコントローラのユーザーズマニュアル

機能

- すべての軸の情報を初期化し、運転可能状態に切り換えます。
- 有効(Enable)とサーボON要求(ServoON)の入力をTRUEにすると、すべての軸を運転可能状態に切り換えます。
- 処理を開始すると実行中(Busy)がTRUEとなります。
- MCv_AllPower(全軸運転可)内で異常が発生した場合、エラー (Error)がTRUEになり、エラーコード(ErrorID)にエラーコードを格納します。エラーコードの詳細は、下記を参照してください。

使用するコントローラのユーザーズマニュアル

- 有効(Enable)とサーボON要求(ServoON)の入力により、全実軸のサーボON/OFF状態、ドライバ状態を下記のように切り換えます。

入力変数		サーボON/OFF状態	ドライバ状態 (AxisName.Md.Driver_State)
有効(Enable)	サーボON要求(ServoON)		
TRUE	TRUE	サーボON	6: Operation Enable (OperationEnable)
	FALSE	サーボOFF	5: Switched On (SwitchedOn)
FALSE	TRUE	サーボOFF	3: Switched On Disabled (SwitchOnDisabled)
	FALSE	サーボOFF	3: Switched On Disabled (SwitchOnDisabled)

- サーボOFF中に実軸が外力で回った場合は、フォローアップ処理を行います。
- サーボON/OFFの制御は、制御モードに関係なく操作できます。サーボOFF時の制御モードは、ドライバの仕様に依存します。
- ドライブユニットエラー発生中、MCv_AllPower(全軸運転可)はドライバに送信しているため、再度有効(Enable)とサーボON要求(ServoON)をFALSEからTRUEにする必要はありません。

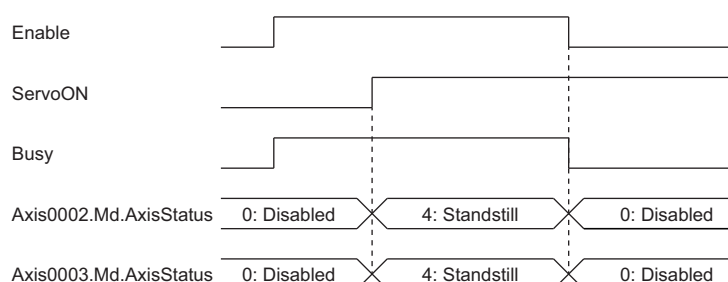
Point

MCv_AllPower(全軸運転可)使用时、個別にサーボOFFする場合は、MC_Power(運転可)を併用してください。MCv_AllPower(全軸運転可)とMC_Power(運転可)を併用した場合は、MC_Power(運転可)の指令を優先します。

■タイミングチャート

例

軸2(Axis0002)、軸3(Axis0003)を実ドライブ軸として設定した場合



注意事項

- MC_Power(運転可)を設定している軸の場合、運転化状態への切換えは、MC_Power(運転可)の指令を優先します。
- 内蔵モーション準備完了(X20)がOFF中のときは、プログラム停止時のサーボON/OFF状態を維持します。

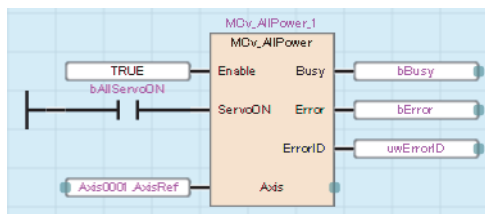
プログラム例

全軸サーボON/OFF(bAllServoON)をTRUEにし、すべての軸を運転可能状態とするプログラム例を下記に示します。

■使用するラベル

ラベル名	データ型	コメント
MCv_AllPower_1	MCv_AllPower	全軸サーボONFB
bAllServoON	ビット	全軸サーボON/OFF
bBusy	ビット	実行中
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム



■STプログラム

```
MCv_AllPower_1(
  Enable:= TRUE ,
  ServoON:= bAllServoON ,
  Busy=> bBusy ,
  Error=> bError ,
  ErrorID=> uwErrorID
);
```

41.16 軸グループオーバーライド値設定

MC_GroupSetOverride

指定した軸グループの目標速度，目標加速度，目標減速度の変更を実行します。

ラダー	FBD/LD	ST
		<pre>MC_GroupSetOverride(AxesGroup:= ?AXES_GROUP_REF?, Enable:= ?BOOL?, VelFactor:= ?LREAL?, AccFactor:= ?LREAL?, JerkFactor:= ?LREAL?, Enabled=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
AxesGroup	軸グループ情報	AXES_GROUP_REF	起動時	—	省略不可	軸グループを設定します。 MC_GroupSetOverride(軸グループオーバーライド値設定)を使用する場合は，出力No.(StartIO)を設定します。 使用する変数(AxesGroupName.AxesGroupRef.)については，下記を参照してください。 1402ページ AxesGroupName.AxesGroupRef.(軸グループ情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Enable	有効	BOOL	常時	TRUE, FALSE	FALSE	TRUEでMC_GroupSetOverride(軸グループオーバーライド値設定)を実行します。
VelFactor	速度オーバーライド係数	LREAL	常時	0.00~10.00	0.00	速度のオーバーライド係数を設定します。 有効(Enable)がTRUEのときは，常時取り込みます。
AccFactor	加速度オーバーライド係数	LREAL	常時	0.00, 0.01~10.00	0.00	加速度のオーバーライド係数を設定します。 有効(Enable)がTRUEのときは，常時取り込みます。 「0.00」を設定した場合は，加速度オーバーライド係数の変更を行わず，前回値のまま制御します。
JerkFactor	ジャークオーバーライド係数	LREAL	常時	0.0	0.0	「0.0」を設定してください。 *: 「0.0」以外を設定した場合，ジャークオーバーライド係数(JerkFactor)範囲外(エラーコード: 1BBEH)となります。

■出力変数

出力変数	名称	データ型	初期値	内容
Enabled	有効中	BOOL	FALSE	オーバーライド値が正常に設定している場合に，TRUEになります。
Busy	実行中	BOOL	FALSE	MC_GroupSetOverride(軸グループオーバーライド値設定)を実行したときに，TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに，TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに，エラーコードを返します。 エラーコードの詳細は，下記を参照してください。 使用するコントローラのユーザズマニュアル

機能

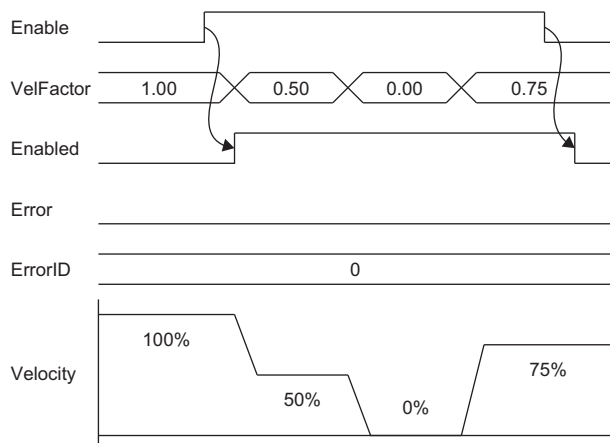
- 指定した軸グループの目標速度，目標加速度，目標減速度を変更します。
- 現在動作中の目標速度，目標加速度，目標減速度にオーバーライド係数を掛けた値に変更します。
- 有効(Enable)がTRUEになるとMC_GroupSetOverride(軸グループオーバーライド値設定)を実行します。オーバーライド係数が有効中のときは，有効中(Enabled)がTRUEになります。
- 有効(Enable)がTRUEのときに，オーバーライド係数の値を変更すると，新たなオーバーライド係数を反映します。
- MC_GroupSetOverride(軸グループオーバーライド値設定)内で異常が発生した場合，エラー (Error)がTRUEになり，エラーコード(ErrorID)にエラーコードを格納します。エラーコードの詳細は，下記を参照してください。

使用するコントローラのユーザーズマニュアル

- 速度オーバーライド係数(VelFactor)に「0.00」の値を設定すると，軸状態(AxisName.Md.AxisStatus)が「4: 待機中 (Standstill)」に移行せずに軸を停止します。
- 加速度オーバーライド係数(AccFactor)に「0.00」の値を設定すると，加速度オーバーライド係数を変更せず，前回の加速度オーバーライド係数を維持します。

■タイミングチャート

- 正常完了の場合



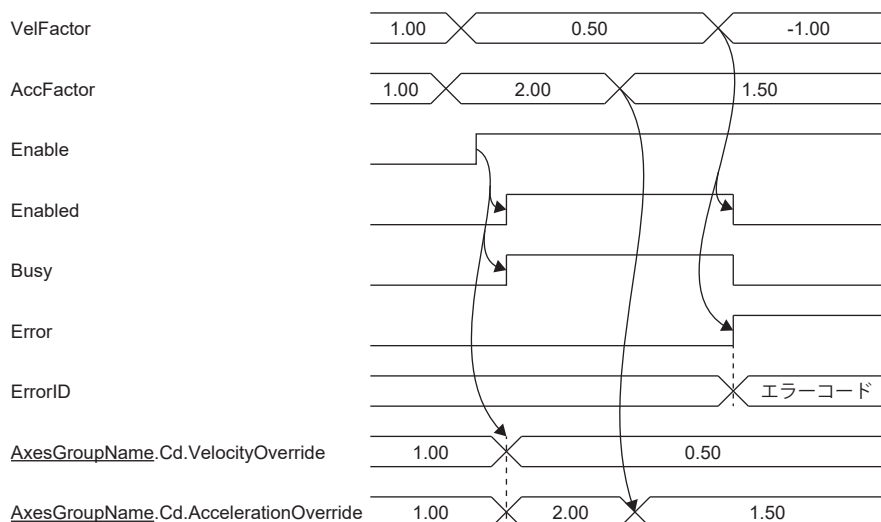
- 異常完了の場合

異常完了時のタイミングチャートについては，下記を参照してください。

1370ページ 有効(Enable)型によるモーション制御FBの基本動作

■動作概要

- MC_GroupSetOverride(軸グループオーバーライド値設定)では，速度オーバーライド係数(AxesGroupName.Cd.VelocityOverride)，加速度オーバーライド係数(AxesGroupName.Cd.AccelerationOverride)を変更します。



- オーバーライド係数に範囲外の値を設定した場合，MC_GroupSetOverride(軸グループオーバーライド値設定)がエラーとなり以降の取込みを停止します。再度取込みを実行する場合は，有効(Enable)を再度立ち上げてください。

注意事項

- 同じ軸グループに対して、MC_GroupSetOverride(軸グループオーバーライド値設定)を2つ以上配置しないでください。2つ以上配置した場合の動作は保証しません。
- 軸グループ制御では、軸グループ制御データで設定したオーバーライド係数のみが影響します。
- MC_GroupSetOverride(軸グループオーバーライド値設定)実行中に、速度オーバーライド係数(AxesGroupName.Cd.VelocityOverride)、加減速度オーバーライド係数(AxesGroupName.Cd.AccelerationOverride)の直接操作を行わないでください。
- 速度オーバーライド後の速度が範囲外となる場合は、下記マニュアルの「速度範囲」を参照してください。

使用するコントローラのユーザーズマニュアル

- 加減速度オーバーライド後の加速度、減速度、加速時間、減速時間が範囲外となる場合は、下記マニュアルの「加減速処理機能」を参照してください。

使用するコントローラのユーザーズマニュアル

プログラム例

軸グループオーバーライド値変更指令(bGroupSetOverride)をTRUEにし、軸グループ1(AxesGroup001)の目標速度、目標加速度、目標減速度に速度オーバーライド係数「1.0」、加減速度オーバーライド係数「2.0」を掛けた値に変更するプログラム例を下記に示します。

■軸グループ

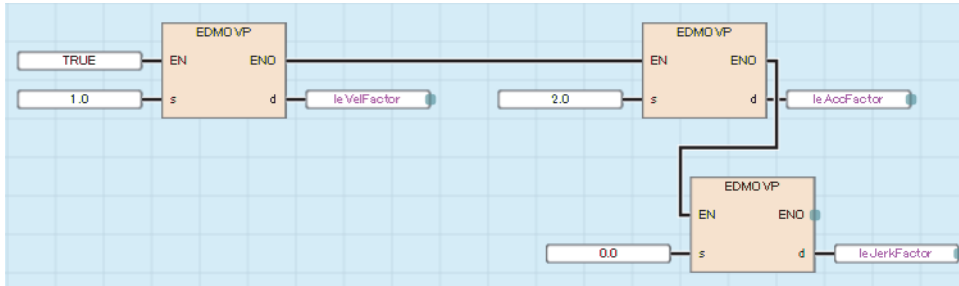
軸グループNo.	ラベル名	データ型	コメント
1	AxesGroup001	AXES_GROUP_REF	軸グループ1

■使用するラベル

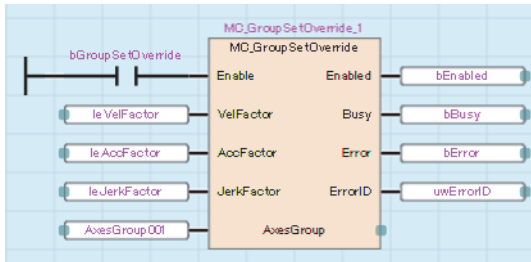
ラベル名	データ型	コメント
MC_GroupSetOverride_1	MC_GroupSetOverride	軸グループオーバーライド値設定FB
bGroupSetOverride	ビット	軸グループオーバーライド値変更指令
leVelFactor	倍精度実数	速度オーバーライド係数
leAccFactor	倍精度実数	加減速度オーバーライド係数
leJerkFactor	倍精度実数	ジャークオーバーライド係数
bEnabled	ビット	有効中
bBusy	ビット	実行中
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

- 軸グループオーバーライド値変更用データ設定



- 軸グループオーバーライド値変更



■STプログラム

//-----軸グループオーバーライド値変更用データ設定-----

leVelFactor:= 1.00;

leAccFactor:= 2.00;

leJerkFactor:= 0.0;

//-----軸グループオーバーライド値変更-----

MC_GroupSetOverride_1(

 AxesGroup:= AxesGroup001.AxesGroupRef ,

 Enable:= bGroupSetOverride ,

 VelFactor:= leVelFactor ,

 AccFactor:= leAccFactor ,

 JerkFactor:= leJerkFactor ,

 Enabled=> bEnabled ,

 Busy=> bBusy ,

 Error=> bError ,

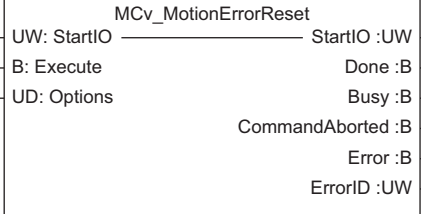
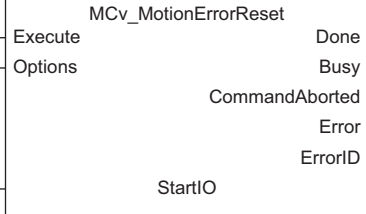
 ErrorID=> uwErrorID

);

41.17 モーションエラーリセット

MCv_MotionErrorReset

モーションシステムのすべてのエラー，警告をリセットします。

ラダー	FBD/LD	ST
		<pre>MCv_MotionErrorReset(StartIO:= ?WORD?, Execute:= ?BOOL?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
StartIO	入出力No.	WORD(HEX)	起動時	000H~0FEH	省略可	先頭入出力番号(16進数4桁で表したときの上3桁)を設定します。

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMCv_MotionErrorReset(モーションエラーリセット)を実行します。
Options	オプション	DWORD(HEX)	起動時	00000000H	00000000H	「00000000H」を設定してください。 *: 「00000000H」以外を設定した場合、オプション範囲外(エラーコード: 1BC3H)となります。

■出力変数

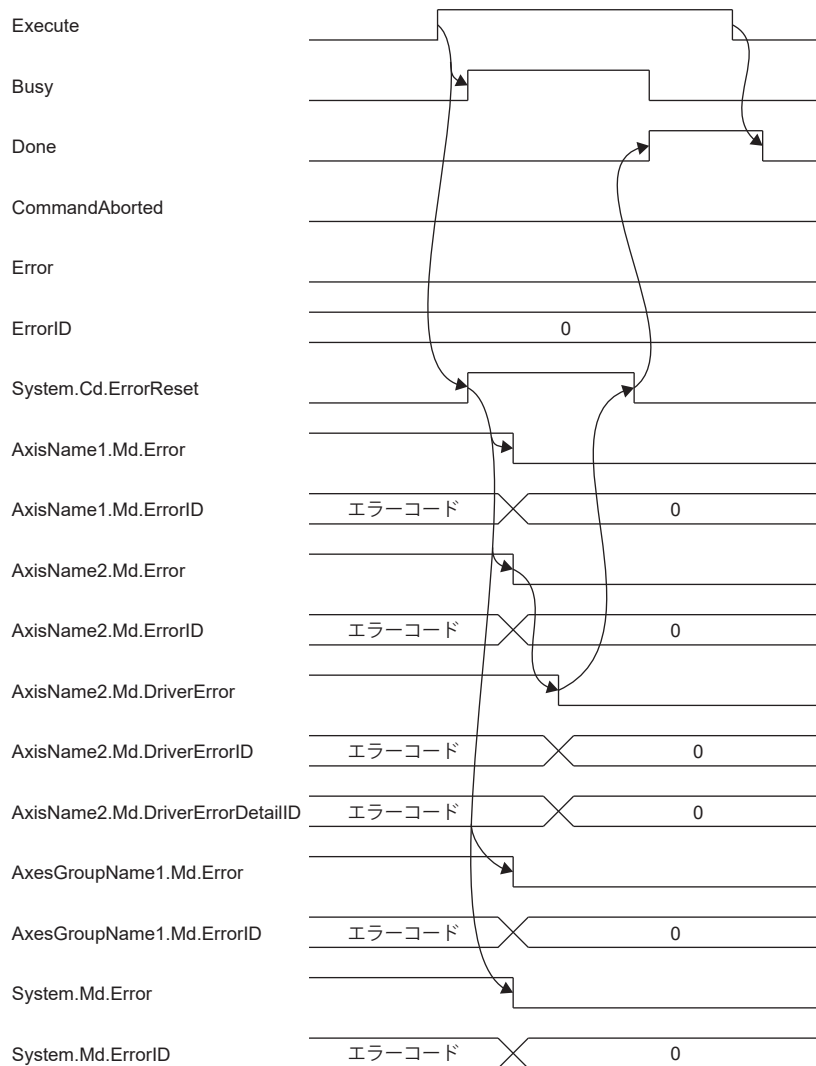
出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	リセットが完了したときに，TRUEになります。
Busy	実行中	BOOL	FALSE	MCv_MotionErrorReset(モーションエラーリセット)を実行したときに，TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	タイムアウトにより，MCv_MotionErrorReset(モーションエラーリセット)が中断するとTRUEになります。 実行指令(Execute)がFALSEになると，FALSEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに，TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに，エラーコードを返します。 エラーコードの詳細は，下記を参照してください。 □□使用するコントローラのユーザーズマニュアル

機能

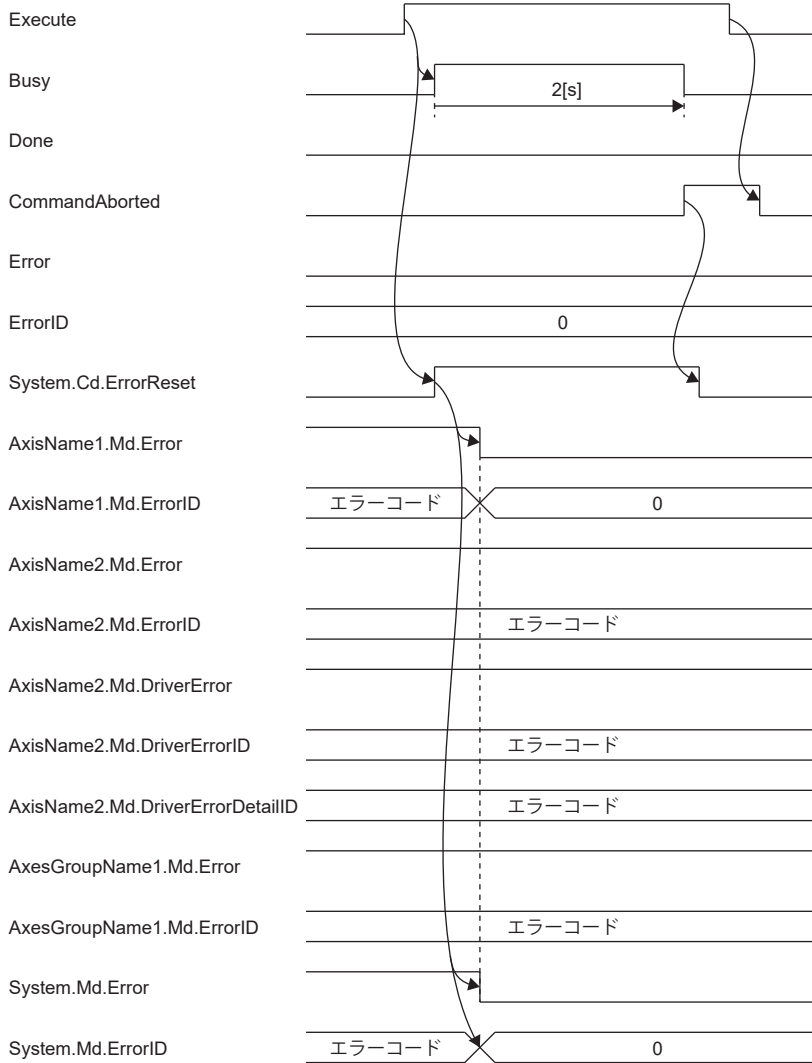
- 実行指令(Execute)がTRUEでMCv_MotionErrorReset(モーションエラーリセット)を実行し、処理を開始すると実行中(Busy)がTRUEになり、システムエラーリセット(MotionSystem.Cd.ErrorReset)をTRUEにします。
- モーションシステムのエラー、警告の解除が完了すると実行中(Busy)がFALSEになり、実行完了(Done)がTRUEになります。
- エラー、警告の要因が残った状態で実行指令(Execute)をTRUEにしても、エラー、警告を解除しません。この場合、FB実行後から2秒以内にエラーを解除しないと実行中断(CommandAborted)がTRUEとなり、システムエラーリセット(MotionSystem.Cd.ErrorReset)をFALSEにします。いったん実行指令(Execute)をFALSEにし、エラー、警告の要因を取り除いた後、再度実行指令(Execute)をTRUEにしてください。

■タイミングチャート

- 正常完了の場合



・ タイムアウトの場合



注意事項

- MCv_MotionErrorReset(モーションエラーリセット)では、特殊リレー (SM0)、特殊レジスタ(SD0)のクリアおよびユニットのエラー LEDの消灯は実施しません。これらの実施方法は使用するコントローラのユーザーズマニュアルを確認してください。
- MCv_MotionErrorReset(モーションエラーリセット)を実行中にシステムエラーリセット(MotionSystem.Cd.ErrorReset)の直接操作を行わないでください。

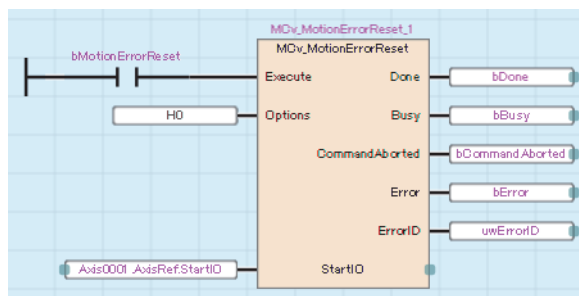
プログラム例

エラーリセット(bMotionErrorReset)をTRUEにし、モーションシステムのエラー、警告をすべてリセットするプログラム例を下記に示します。

■使用するラベル

ラベル名	データ型	コメント
MCv_MotionErrorReset_1	MCv_MotionErrorReset	モーションエラーリセットFB
bMotionErrorReset	ビット	エラーリセット
bDone	ビット	実行完了
bBusy	ビット	実行中
bCommandAborted	ビット	実行中断
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム



■STプログラム

```

MCv_MotionErrorReset_1(
  Execute:= bMotionErrorReset ,
  Options:= H00000000 ,
  Done=> bDone ,
  Busy=> bBusy ,
  CommandAborted=> bCommandAborted ,
  Error=> bError ,
  ErrorID=> uwErrorID
);

```

41.18 制御演算周期同期

41

MCv_SyncOperationCycles

軸の制御演算周期を設定した周期と同期します。

ラダー	FBD/LD	ST
		<pre> MCv_SyncOperationCycles(Axis:= ?AXIS_REF?, ValueSource:= ?TARGET_REF?, Enable:= ?BOOL?, OperationCycle:= ?INT?, Options:= ?DWORD?, Enabled=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?); </pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName,AxisRef.)については、下記を参照してください。 1378ページ AxisName,AxisRef.(軸情報)
ValueSource	データソース選択	SIGNAL_SELECT	起動時	—	省略可	入力は無視されます。

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Enable	有効	BOOL	常時	TRUE, FALSE	FALSE	TRUEになっている間、制御演算周期同期が有効になります。
OperationCycle	制御演算周期	INT	起動時	1~3	1	同期先の制御演算周期を設定します。 ・1: 第1演算周期 ・2: 第2演算周期 ・3: 第3演算周期
Options	オプション	DWORD(HEX)	起動時	00000000H	00000000H	「00000000H」を設定してください。 「00000000H」以外を設定した場合、とエラーになります。

■出力変数

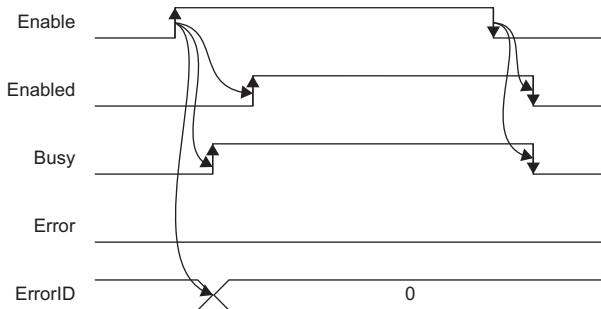
出力変数	名称	データ型	初期値	内容
Enabled	有効中	BOOL	FALSE	制御演算周期同期が正常に動作している場合に、TRUEになります。
Busy	実行中	BOOL	FALSE	MCv_SyncOperationCycles(制御演算周期同期)が実行中のときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 使用するコントローラのユーザーズマニュアル

機能

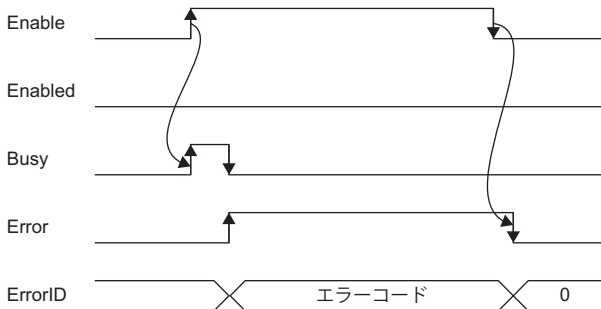
- MCv_SyncOperationCycles(制御演算周期同期)は、選択したデータ(軸またはデータソース)の制御演算周期を設定した値の周期と同期します。
- 単軸同期制御FBで、主軸と従軸の制御演算周期が異なるときに使用します。
- 有効(Enable)がTRUEになると制御演算周期同期が有効になります。制御演算周期同期が有効中のときは、有効中(Enabled)がTRUEになります。

■タイミングチャート

- 正常完了の場合



- 異常完了の場合



注意事項

- 本FBは選択したデータを指定した制御時演算周期に同期させますが、制御演算周期設定は変更しません。
- 入力と出力(同期制御では入力が主軸、出力が従軸)の制御演算周期が異なる場合、本FBを起動してから、同期制御を起動してください。同期せずに同期制御を起動した場合、演算周期不一致(エラーコード: 1B29H)が発生します。
- 本FBを実行すると演算時間が増加します。
- モーション制御FBの入力引数のリフレッシュは、FBの読出しタイミングで行われますが、本FBはFB呼出しタスク(スキャン/定周期/イベント)によらず、演算周期でリフレッシュします。

プログラム例

制御演算周期同期(MCv_SyncOperationCycles)を用いて制御周期の異なる軸を同期制御するプログラム例を下記に示します。

本プログラム例では、第1演算周期で制御するAxis0001を主軸、第2演算周期で制御するAxis0002を従軸としてMC_GearInを起動します。

MR-J5-GをサーボONするプログラムは省略しています。

■CC-Link IE TSN構成

局番	型名	IPアドレス	ネットワーク同期通信設定	通信周期設定
1	MR-J5-G	192.168.4.1	同期する	基本周期
2	MR-J5-G	192.168.4.2	同期する	中速

■軸

軸No.	ラベル名	局アドレス設定	軸種別	制御周期設定
1	Axis0001	192.168.4.1	実ドライブ軸	第1演算周期で動作
2	Axis0002	192.168.4.2	実ドライブ軸	第2演算周期で動作

■使用するラベル

ラベル名	データ型	コメント
bSyncStart	ビット	同期制御開始
bMCGearInExecute	ビット	ギア動作開始
MCv_SyncOperationCycles_1	MCv_SyncOperationCycles	制御演算周期同期FB
MC_GearIn_1	MC_GearIn	ギア動作開始FB

■STプログラム

```
SET(MCv_SyncOperationCycles_1.Enabled, bMCGearInExecute);
```

```
//-----制御演算周期同期-----
```

```
MCv_SyncOperationCycles_1(
  Axis:= Axis0001.AxisRef ,
  //ValueSource:= ?TARGET_REF? ,//省略
  Enable:= bSyncStart ,
  OperationCycle:= 2 ,//2:第2演算周期
  Options:= 0
);
```

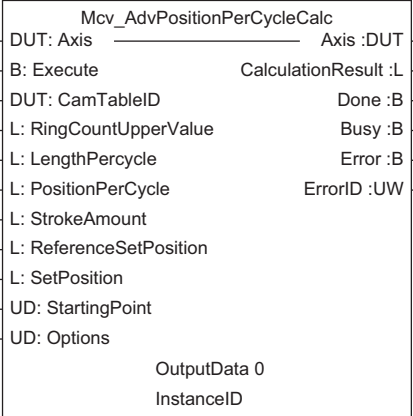
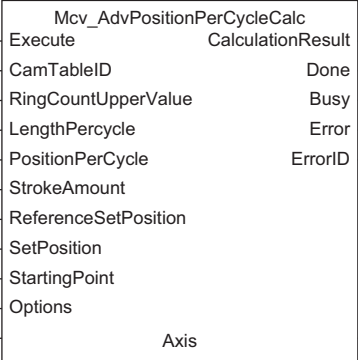
```
//-----ギア動作開始-----
```

```
MC_GearIn_1(
  Master:= Axis0001.AxisRef ,
  Slave:= Axis0002.AxisRef ,
  Execute:= bMCGearInExecute ,
  ContinuousUpdate:= TRUE ,
  RatioNumerator:= 1 ,
  RatioDenominator:= 1 ,
  MasterValueSource:= MC_SOURCE__mcSetValue ,
  Acceleration:= 1000.0 ,
  Deceleration:= 1000.0 ,
  Jerk:= 0.0 ,
  BufferMode:= MC_BUFFER_MODE__mcAborting ,
  Options:= 0
);
```

41.19 アドバンスト同期制御1サイクル現在位置計算

MCv_AdvPositionPerCycleCalc

指定したカムデータをもとに1サイクル現在位置を算出します。

ラダー	FBD/LD	ST
		<pre>MCv_AdvPositionPerCycleCalc(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, CamTableID:= ?MC_CAM_ID?, RingCountUpperValue:= ?LREAL?, LengthPerCycle:= ?LREAL?, PositionPerCycle:= ?LREAL?, StrokeAmount:= ?LREAL?, ReferenceSetPosition:= ?LREAL?, SetPosition:= ?LREAL?, StartingPoint:= ?DWORD?, Options:= ?DWORD?, CalculationResult=> ?LREAL?, Done=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略可	設定は無視されます。

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEで本FBを実行します。
CamTableID	カムテーブルID	MC_CAM_ID	起動時	0~6000	0	計算対象の演算カムテーブルIDを設定します。 PROFILE_ID構造体のプロファイルID番号(Number)に「0」を設定した場合は、直線カムとして計算します。
RingCountUpperValue	リングカウンタ上限	LREAL	起動時	0.0, 2.0~2147483647.0	0.0	カム位置計算でリングカウンタ上限を考慮する場合、計算対象の出力軸のリングカウンタ上限値を設定します。 「0.0」を設定した場合は、リングカウンタを「-100000000000.0~100000000000.0」として計算します。
LengthPerCycle	カム位置1サイクル長	LREAL	起動時	1.0~2147483647.0	1.0	カム計算に必要な1サイクル長を設定します。
PositionPerCycle	カム位置計算1サイクル現在位置	LREAL	起動時	0.0~2147483647.0	0.0	カム計算に必要な1サイクル現在位置を設定します。 カム位置1サイクル長(LengthPerCycle)以上の値を設定した場合、「0.0~(カム位置1サイクル長(LengthPerCycle)-0.00001)」の範囲内に変換して計算されます。
StrokeAmount	カム位置計算ストローク量	LREAL	起動時	-2147483648.0~2147483647.0	0.0	カム計算に必要なカムストローク量を設定します。
ReferenceSetPosition	カム位置計算カム基準位置	LREAL	起動時	-100000000000.0~100000000000.0	0.0	カム計算に必要なカム基準位置を設定します。
SetPosition	カム位置計算カム指令現在位置	LREAL	起動時	-100000000000.0~100000000000.0	0.0	カム計算に必要な出力軸の現在位置を設定します。

入力変数	名称	データ型	取込	有効範囲	初期値	内容
StartingPoint	カム位置計算 カム開始点	DWORD	起動時	0~65535	0	カム計算でカム開始点を考慮する場合に設定します。
Options	オプション	DWORD(HEX)	起動時	0固定	0	本FBでは設定できません。0に設定してください。0以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

■出力変数

出力変数	名称	データ型	初期値	内容
CalculationResult	計算結果	LREAL	0.0	1サイクル現在位置の計算結果を格納します。
Done	実行完了	BOOL	FALSE	計算が完了したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	本FBを実行したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。エラーコードの詳細は、下記を参照してください。 📖使用するコントローラのユーザーズマニュアル

■公開変数

公開変数	名称	データ型	初期値	内容
OutputData	出力データ	LREAL	0.0	- 直線カム、または演算プロファイルデータの補間方法指定(Interpolate)が「0: 直線補間」の場合、計算結果に対応するストローク量が格納されます。 - 演算プロファイルデータの補間方法指定が「1: 区間ごとに指定」、または「2: スプライン補間」の場合、またはデータ形式の種別に「ロータリーカッター」を設定した場合は、計算結果(1サイクル現在位置)に対応するカムデータのストローク比を%単位で格納します。 - 出力データ(OutputData)は、実行完了(Done)がTRUEになったときに値が格納されます。
InstanceID	インスタンスID	INSTANCE_ID	0	インスタンスIDです。 インスタンス生成時にシステムにより自動設定されます。

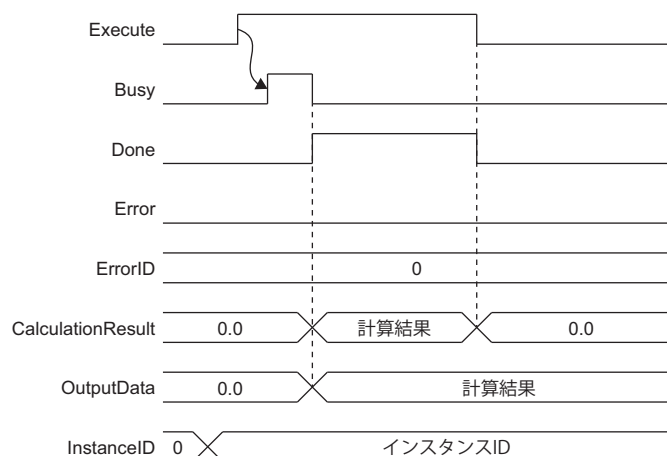
機能

- カムデータからカム位置計算カム基準位置(ReferenceSetPosition)とカム位置計算カム指令現在位置(SetPosition)を元に1サイクル現在位置を計算します。計算結果は、計算結果(CalculationResult)に出力します。
- 1サイクル現在位置計算についての詳細は、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

■タイミングチャート

- 正常完了の場合



- 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

注意事項

- カムデータは事前に展開設定で自動展開する、またはMC_CamTableSelect(カムテーブル選択)で展開エリアへ展開して使用してください。
- 1サイクル現在位置計算(MCv_AdvPositionPerCycleCalc)の計算結果は、MCv_AdvancedSync(アドバンスト同期制御)にのみ適用してください。MCv_AdvancedSync(アドバンスト同期制御)以外の同期制御FB(MC_CamInなど)に適用した場合、計算結果と実際の動作は一致しません。
- 座標データの1サイクル現在位置計算は、カムデータの最終点が1サイクル長によって下記の計算結果となる場合があります。座標データを使用する場合、カムデータの最終点を1サイクル長と一致させることを推奨します。

カムデータの最終点	内容
1サイクル長より大きい場合	1サイクル長の設定値によらずカムデータの全範囲のサーチを行うため、同期制御の軌跡と異なる計算結果となる場合があります。
1サイクル長より小さい場合	カム指令位置とカム基準位置の関係によって、1サイクル現在値が1サイクル長より大きくなる場合があります。

プログラム例

本FBのプログラム例については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

41.20 アドバンスト同期制御カム指令現在位置計算

41

MCv_AdvCamSetPositionCalc

指定したカムデータをもとにカム指令現在位置を算出します。

ラダー	FBD/LD	ST
<pre> MCv_AdvCamSetPositionCalc DUT: Axis Axis :DUT B: Execute CalculationResult :L DUT: CamTableID Done :B L: RingCountUpperValue Busy :B L: LengthPerCycle Error :B L: StrokeAmount ErrorID :UW L: ReferenceSetPosition L: PositionPerCycle UD: StartingPoint UD: Options InputData 0 InstanceID </pre>	<pre> MCv_AdvCamSetPositionCalc Execute CalculationResult CamTableID Done RingCountUpperValue Busy LengthPerCycle Error StrokeAmount ErrorID ReferenceSetPosition PositionPerCycle StartingPoint Options Axis </pre>	<pre> MCv_AdvCamSetPositionCalc(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, CamTableID:= ?MC_CAM_ID?, RingCountUpperValue:= ?LREAL?, LengthPerCycle:= ?LREAL?, StrokeAmount:= ?LREAL?, ReferenceSetPosition:= ?LREAL?, PositionPerCycle:= ?LREAL?, StartingPoint:= ?DWORD?, Options:= ?DWORD?, CalculationResult=> ?LREAL?, Done=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?); </pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略可	設定は無視されます。

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEで本FBを実行します。
CamTableID	カムテーブルID	MC_CAM_ID	起動時	0~6000	0	計算対象の演算カムテーブルIDを設定します。 PROFILE_ID構造体のプロファイルID番号(Number)に「0」を設定した場合は、直線カムとして計算します。
RingCountUpperValue	リングカウンタ上限	LREAL	起動時	0.0, 2.0~2147483647.0	0.0	カム位置計算でリングカウンタ上限を考慮する場合、計算対象の出力軸のリングカウンタ上限値を設定します。 「0.0」を設定した場合は、リングカウンタを「-10000000000.0~10000000000.0」として計算します。
LengthPerCycle	カム位置1サイクル長	LREAL	起動時	1.0~2147483647.0	1.0	カム計算に必要な1サイクル長を設定します。
StrokeAmount	カム位置計算ストローク量	LREAL	起動時	-2147483648.0~2147483647.0	0.0	カム計算に必要なカムストローク量を設定します。
ReferenceSetPosition	カム位置計算カム基準位置	LREAL	起動時	-10000000000.0~10000000000.0	0.0	カム計算に必要なカム基準位置を設定します。
PositionPerCycle	カム位置計算1サイクル現在位置	LREAL	起動時	0.0~2147483647.0	0.0	カム計算に必要な1サイクル現在位置を設定します。 カム位置1サイクル長(LengthPerCycle)以上の値を設定した場合、「0.0~(カム位置1サイクル長(LengthPerCycle)-0.00001)」の範囲内に変換して計算されます。
StartingPoint	カム位置計算カム開始点	DWORD	起動時	0~65535	0	カム計算でカム開始点を考慮する場合に設定します。

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Options	オプション	DWORD(HEX)	起動時	0固定	0	本FBでは設定できません。0に設定してください。0以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

■出力変数

出力変数	名称	データ型	初期値	内容
CalculationResult	計算結果	LREAL	0.0	カム指令現在位置の計算結果を格納します。
Done	実行完了	BOOL	FALSE	計算が完了したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	本FBを実行したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。エラーコードの詳細は、下記を参照してください。 📖使用するコントローラのユーザーズマニュアル

■公開変数

公開変数	名称	データ型	初期値	内容
InputData	入力データ	LREAL	0.0	- 直線カムの場合は、計算結果に対応する1サイクル現在位置を格納します。 - 演算プロファイルデータの補間方法指定(Interpolate)が「0:直線補間」の場合、計算結果(カム指令現在位置)に対応する1サイクル現在位置を格納します。 - 演算プロファイルデータの補間方法指定が「1:区間ごとに指定」、または「2:スプライン補間」の場合、またはデータ形式の種別に「ロータリーカッター」を設定した場合は、計算結果(カム指令現在位置)に対応するカムデータの分解能を格納します。 - 入力データ(InputData)は、実行完了(Done)がTRUEになったときに値を格納します。
InstanceID	インスタンスID	INSTANCE_ID	0	インスタンスIDです。インスタンス生成時にシステムにより自動設定します。

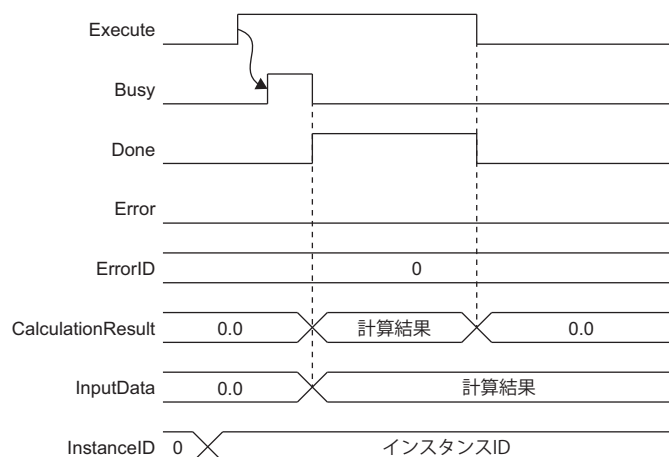
機能

- カム位置計算1サイクル現在位置(PositionPerCycle)とカム位置計算カム基準位置(ReferenceSetPosition)を元にカム指令現在位置を計算します。計算結果は、計算結果(CalculationResult)に出力します。
- カム指令現在位置計算についての詳細は、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

■タイミングチャート

- 正常完了の場合



- 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

注意事項

- カムデータは事前に展開設定で自動展開する，またはMC_CamTableSelect(カムテーブル選択)で展開エリアへ展開して使用してください。
- カム指令現在位置計算(MCv_AdvCamSetPositionCalc)の計算結果は，MCv_AdvancedSync(アドバンスト同期制御)にのみ適用してください。MCv_AdvancedSync(アドバンスト同期制御)以外の同期制御FB(MC_CamInなど)に適用した場合，計算結果と実際の動作は一致しません。

プログラム例

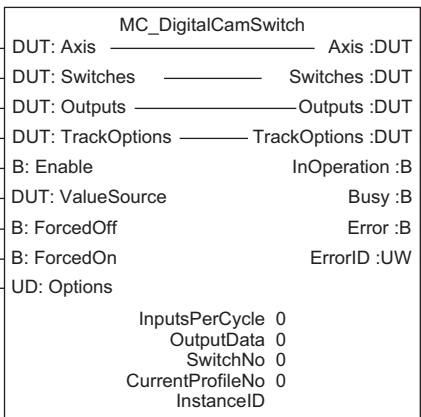
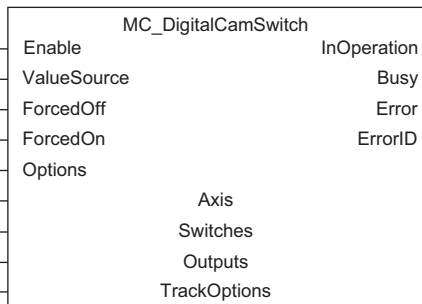
本FBのプログラム例については，下記を参照してください。

 使用するコントローラのユーザーズマニュアル

41.21 デジタルカムスイッチ出力

MC_DigitalCamSwitch

任意データの値に応じてON/OFF信号を出力します。

ラダー	FBD/LD	ST
		<pre>MC_DigitalCamSwitch(Axis:= ?AXIS_REF?, Switches:= ?MC_CAMSWITCH_DATA_REF?, Outputs:= ?MC_OUTPUT_REF?, TrackOptions:= ?MC_TRACK_REF? ,Enable:= ?BOOL?, ValueSource:= ?MC_CAMSWITCH_SOURCE_REF?, ForcedOff:= ?BOOL?, ForcedOn:= ?BOOL?, Options:= ?DWORD?, InOperation=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸	AXIS_REF	起動時	—	省略可	軸を設定します。 使用する変数(AxisName.AxisRef)については、下記を参照してください。 ☞ 1378ページ AxisName.AxisRef .(軸情報)
Switches	スイッチ	MC_CAMSWITCH_DATA_REF ☞ 1420ページ MC_CAMSWITCH_REF	常時	1~60000	省略不可	演算プロファイルを使用してスイッチ設定を設定します。 ☞ 1563ページ スイッチ(Switches)
Outputs	出力信号	MC_OUTPUT_REF	起動時	—	省略不可	出力信号を設定します。 詳細は、下記を参照してください。 ☞ 1564ページ 出力信号(Outputs)
TrackOptions	トラックオプション	MC_TRACK_REF	常時	—	—	トラックオプションを設定します。 詳細は、下記を参照してください。 ☞ 1565ページ トラックオプション(TrackOptions)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Enable	有効	BOOL	常時	TRUE, FALSE	FALSE	TRUEになっている間、本FBを実行します。
ValueSource	データソース選択	MC_CAMSWITCH_SOURCE_REF	起動時	—	—	データソースを設定します。 詳細は、下記を参照してください。 ☞ 1566ページ データソース選択(ValueSource)
ForcedOff	強制OFF	BOOL	常時	TRUE, FALSE	FALSE	TRUEになっている間、出力信号が強制的にOFFになります。 詳細は、下記を参照してください。 ☞ 1567ページ 強制OFF(ForcedOff)
ForcedOn	強制ON	BOOL	常時	TRUE, FALSE	FALSE	TRUEになっている間、出力信号が強制的にONになります。 詳細は、下記を参照してください。 ☞ 1567ページ 強制ON(ForcedOn)
Options	オプション	DWORD(HEX)	起動時	—	00000000H	機能オプションを設定します。 詳細は、下記を参照してください。 ☞ 1567ページ オプション(Options)

■出力変数

出力変数	名称	データ型	初期値	内容
InOperation	作動中	BOOL	FALSE	出力信号が有効なときに、TRUEになります。
Busy	実行中	BOOL	FALSE	本FBが実行中のときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 📖使用するコントローラのユーザーズマニュアル

■公開変数

公開変数	名称	データ型	初期値	内容
InputsPerCycle	1サイクル現在位置	LREAL	0	データソースの値を格納します。
OutputData	出力値	BOOL	0	出力信号への出力値を格納します。
SwitchNo	スイッチ番号	WORD(UINT)	0	ON中のスイッチ番号を格納します。
CurrentProfileNo	実行中プロファイルNo.	WORD(UINT)	0	実行中のプロファイルIDを格納します。
InstanceID	インスタンスID	INSTANCE_ID	0	インスタンスIDです。 インスタンス生成時にシステムにより自動設定します。

機能

データソース選択(ValueSource)に設定したデータソースがスイッチ設定の条件を満たすか判定し、出力信号を出力します。
デジタルカムスイッチ機能の詳細は下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

■タイミングチャート

本FBのタイミングチャートは下記を参照してください。

📖 1370ページ 有効(Enable)型によるモーション制御FBの基本動作

■スイッチ(Switches)

演算プロファイルIDをMC_CAMSWITCH_DATA_REF構造体で指定します。

設定方法は下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

実行中にスイッチ(Switches)を範囲外に変更した場合、または展開エリアがない場合は、プログラム警告(イベントコード: 00F19H)となり、変更前の値で動作を継続します。

■出力信号(Outputs)

出力信号をMC_OUTPUT_REF構造体で設定します。MC_OUTPUT_REF構造体については、下記を参照してください。

📖 1420ページ MC_OUTPUT_REF

出力信号(Outputs)で設定するMC_OUTPUT_REF構造体の出力信号(Outputs)の設定範囲を下記に示します。

出力信号(Outputs)の型は、SIGNAL_SELECT構造体となります。SIGNAL_SELECT構造体については、下記を参照してください。

📖 1424ページ SIGNAL_SELECT(信号選択)

構造体	変数名	型	設定範囲
SIGNAL_SELECT(信号選択)	Source(信号)	TARGET_REF 📖 1421ページ TARGET_REF(入力信号)	■型 ・ BOOL ■データ種別 ・ [LINK]*1
	Detection(信号検出方法)	INT (MC_SIGNAL_LOGIC) 📖 1452ページ MC_SIGNAL_LOGIC	ビット出力信号の信号論理は選択できません。
	CompensationTime(補正時間)*2	LREAL	-5.0~5.0[s]
	FilterTime(フィルタ時間)	LREAL	0.0[s]*3

*1 [LINK]を指定した場合、対応するネットワークラベルへの書き込みは無視されます。外部信号高精度出力を使用する場合にのみ、設定できます。詳細は、使用するコントローラのユーザーズマニュアルを確認してください。

*2 機器の応答時間による遅れなどを補正する場合に設定します。時間補正の詳細は、使用するコントローラのユーザーズマニュアルを確認してください。

*3 0.0以外を設定した場合、信号のフィルタ時間設定範囲外(エラーコード: 1BC8H)になります。チャタリングを除去する場合は、トラックオプション(TrackOptions)のHysteresis(スイッチング禁止距離)を設定してください。

複数のFBインスタンスで同一の出力信号を設定した場合、出力結果の論理和を出力します。

[OBJ]指定時に対象修飾を省略した場合は、軸(Axis)に指定した軸のオブジェクトを参照します。ただし、軸(Axis)に指定した軸が局アドレスを持たない、または局アドレスが未設定の場合、外部信号選択不正(エラーコード: 1B31H)を出力します。

[OBJ]指定時に対象修飾で局アドレスを指定した場合、軸(Axis)の設定は無視されます。

[OBJ]指定時に参照する局が、軸エミュレート機能有効かつ局アドレス設定済みの場合、エミュレートしているオブジェクトを参照します。

出力信号のリフレッシュは、使用するプログラム実行周期によらず、下記のタイミングで行います。

データ種別	リフレッシュを行うタイミング
[LINK]	対象局の通信周期

■トラックオプション(TrackOptions)

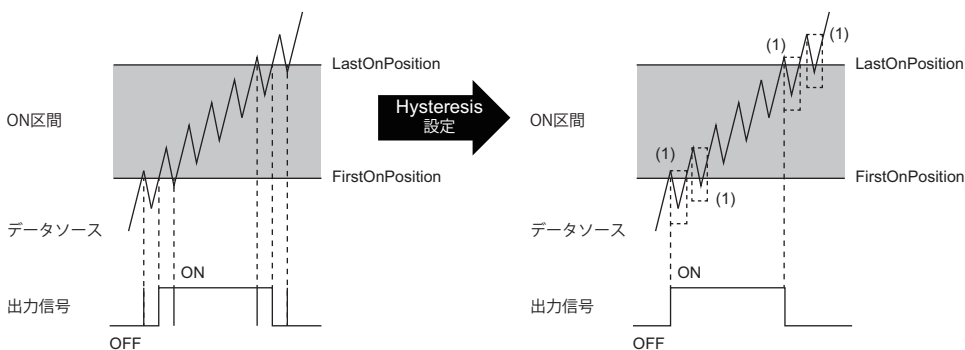
トラックオプションをMC_TRACK_REF構造体で設定します。MC_TRACK_REF構造体については、下記を参照してください。

📖 1420ページ MC_TRACK_REF

トラックオプション(TrackOptions)で設定するMC_TRACK_REF構造体のトラックオプション(TrackOptions)の設定範囲を下記に示します。

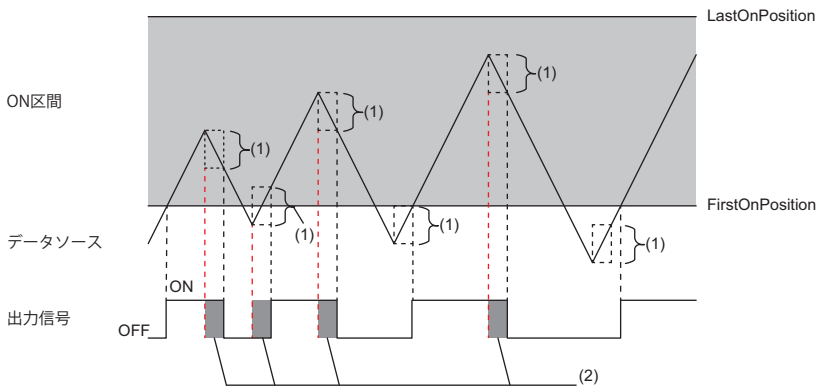
項目	設定範囲	内容
Hysteresis(スイッチング禁止距離)	0.0~10000000000.0	データソースの速度が不安定な場合に出力信号のチャタリングを防止するための不感帯を設定します。^{*1} データソースリングカウンタ範囲の半分未満を設定してください。データソースリングカウンタ範囲の半分以上を設定した場合、MC_DigitalCamSwitch命令異常(エラーコード: 1BCAH)になります。 実行中に設定値の範囲外に変更した場合、プログラム警告(イベントコード: 00F19H)になり、変更前の値で動作を継続します。 データソースの移動方向が反転した位置からHysteresis内にある間、Hysteresisを変更しても制御に反映しません。Hysteresis外となったタイミングで制御に反映します。

*1 詳細は下記になります。



(1) Hysteresis(スイッチング禁止距離)

- スイッチ設定のスイッチモード選択(CamSwitchMode)が「1: 位置基準」の場合、データソースの変化方向が反転した位置からHysteresis(スイッチング禁止距離)の間にあるときは、出力信号のON/OFFの切替えを行いません。スイッチ設定の方向選択(Direction)が「1: 正方向」の場合、下記のように動作します。



(1) Hysteresis(スイッチング禁止距離)

(2) 出力信号のON/OFFの切替えを行いません。

■データソース選択(ValueSource)

下記をMC_CAMSWITCH_SOURCE_REF構造体で指定します。

- データソース
- データソースの上限値および下限値

MC_CAMSWITCH_SOURCE_REF構造体については、下記を参照してください。

📄 1420ページ MC_CAMSWITCH_SOURCE_REF

変数名	型	設定範囲	内容
Source(データソース)	TARGET_REF 📄 1421ページ TARGET_REF(入力信号)	■型 • LREAL *1 ■データ種別 • [AXIS] • [VAR] • [DEV] • [CONST]	データ種別に[AXIS]を指定した場合は軸(Axis)で指定した軸データを参照します。対象の軸がエラーまたは切断となった場合でも、本FBは制御を続行します。 [AXIS]を指定し、本FBの実行中に対象の軸に対して現在位置変更または原点復帰を実施した場合、出力信号のON/OFFが意図しない結果となる場合があります。本FBの停止中または強制OFF、強制ON中に行ってください。 データ種別に[VAR], [DEV], [CONST]を指定した場合、軸(Axis)で指定した軸データは無視します。
RingCount_LowerValue(データソースリングカウンタ下限値)*2	LREAL	-10000000000.0 ~10000000000.0	データソースのリングカウンタ範囲を下記の式の範囲になるように設定します。 $\text{RingCount\_LowerValue} \leq \text{データソース} < \text{RingCount\_UpperValue}$ データ型に(LREAL)を指定する前のデータソース範囲に合わせて設定してください。 ■データソースにINT型データを使用する場合の設定例 RingCount_LowerValue: -32768.0 RingCount_UpperValue: 32768.0 ■0~99の範囲の整数値をとるユーザ定義データを使用する場合の設定例 RingCount_LowerValue: 0.0 RingCount_UpperValue: 100.0 RingCount_LowerValue(データソースリングカウンタ下限値)がRingCount_UpperValue(データソースリングカウンタ上限値)以上の場合、MC_DigitalCamSwitch命令異常(エラーコード: 1BCAH)になります。 RingCount_UpperValue(データソースリングカウンタ上限値)-RingCount_LowerValue(データソースリングカウンタ下限値)の絶対値が2未満の場合、MC_DigitalCamSwitch命令異常(エラーコード: 1BCAH)になります。 リフレッシュ周期あたりのデータソースの移動量がデータソースリングカウンタ範囲の半分未満になるようにしてください。リフレッシュ周期あたりのデータソースの移動量がデータソースリングカウンタ範囲の半分以上になる場合、方向の判定が正常に行えず、正しいタイミングで出力信号を出力できません。
RingCount_UpperValue(データソースリングカウンタ上限値)*2			

*1 ON区間判定はLREAL型として動作します。LREAL型以外のデータを指定する場合は、明示的にデータの型に(LREAL)を指定してください。データの型に(LREAL)を指定しない場合は、外部信号選択文字列不正(エラーコード: 1BC5H)となります。

*2 Source(データソース)に[Axis]を指定した場合は軸(Axis)で指定した軸データを参照するため、設定値は無視されます。

1つのデータソースに対して、複数のFBインスタンスを同時に実行できます。

出力信号のリフレッシュは、使用するプログラム実行周期によらず、下記のタイミングで行います。

データ種別	リフレッシュを行うタイミング
[VAR]	ネットワークラベル: 対象局の通信周期 ネットワークラベル以外: 第1演算周期
[DEV]	第1演算周期
[CONST]	
[AXIS]	対象局の通信周期

■強制OFF(ForcedOff)

実行中に出力信号を強制的にOFFにします。

強制OFF(ForcedOff)の状態	動作
FALSE	スイッチ設定に基づき、出力信号のON/OFFを行います。 □使用するコントローラのユーザーズマニュアル
TRUE	出力信号が強制的にOFFにします。

■強制ON(ForcedOn)

実行中に出力信号を強制的にONにします。

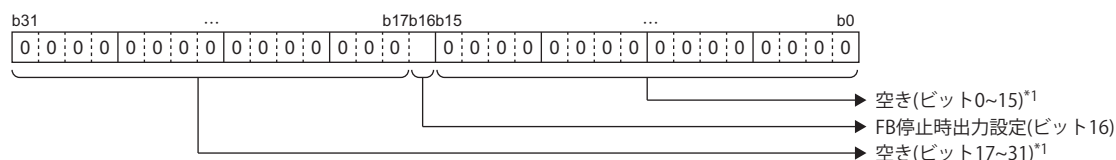
強制OFF(ForcedOff)がTRUEになっているときに、強制ON(ForcedOn)をTRUEにすると強制的にONになります。

強制ON(ForcedOn)の状態	動作
FALSE	スイッチ設定に基づき、出力信号のON/OFFを行います。 □使用するコントローラのユーザーズマニュアル
TRUE	出力信号が強制的にONにします。

■オプション(Options)

本FBで使用する機能オプションをビット指定で設定します。

ビット指定で設定する内容を下記に示します。



*1 空には「0」を設定してください。「0」以外を設定した場合、MC_DigitalCamSwitch命令異常(エラーコード: 1BCAH)となり始動しません。

ビット	名称	内容
16	FB停止時出力設定	有効(Enable)がTRUE→FALSEになった、"シーケンサレディ (MotionSystem.Cd.SequenceReady)"がON→OFFになった、またはエラー発生時の場合に、FB停止時に出力信号を保持する、またはOFFにするかを設定します。 0: 保持する 1: OFFにする 実行中に"シーケンサレディ (MotionSystem.Cd.SequenceReady)"が切り替わった場合の動作を下記に示します。 シーケンサレディ ON区間 出力信号 (FB停止時出力設定: 0の場合) 出力信号 (FB停止時出力設定: 1の場合) ■Outputs: [LINK]指定時 コントローラSTOP・エラー時の出力については、下記を参照してください。 □使用するコントローラのユーザーズマニュアル リンクデバイスRWwへの出力は保持となるため、リモートI/Oの出力HOLD/CLEAR設定機能が優先されません。詳細は下記を参照してください。 □使用するコントローラのユーザーズマニュアル

注意事項

出力信号(Outputs)に設定した信号と、出力値(OutputData)が異なる場合は、他機能が出力信号(Outputs)に設定した信号を操作していないか確認してください。

プログラム例

本FBのプログラム例については、下記を参照してください。

使用するコントローラのユーザーズマニュアル

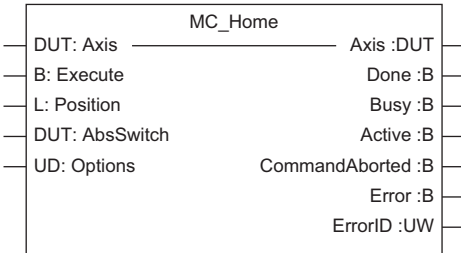
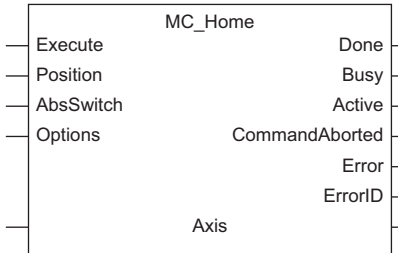
42 動作系のファンクションブロック

42

42.1 原点復帰

MC_Home

指定した軸の原点復帰を行います。

ラダー	FBD/LD	ST
		<pre>MC_Home(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, Position:= ?LREAL?, AbsSwitch:= ?MC_INPUT_REF?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 ☞ 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

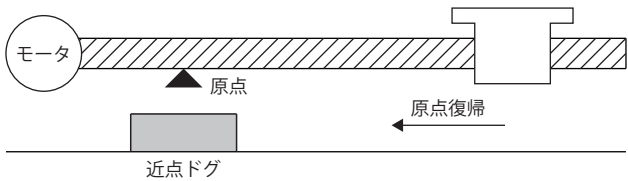
入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_Home(原点復帰)を実行します。
Position	目標位置	LREAL	起動時	-10000000000.0~10000000000.0	0.0	原点アドレスを設定します。 下記範囲内で設定してください。 • -10000000000.0 ≤ 設定値 < 10000000000.0 *: リングカウンタ有効の場合は、リングカウンタ範囲となります。
AbsSwitch	原点スイッチ	MC_INPUT_REF	起動時	—	—	ドライバ式原点復帰でデバイス機器に伝達する近点ドグ信号を設定します。 詳細は、下記を参照してください。 ☞ 1574ページ 原点スイッチ (AbsSwitch)
Options	オプション	DWORD(HEX)	起動時	00000000H~00010000H	00000000H	MC_Home(原点復帰)の機能オプションをビット指定で設定します。 詳細は、下記を参照してください。 ☞ 1575ページ オプション(Options)

■出力変数

出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	原点復帰が完了したときに、TRUEとなります。
Busy	実行中	BOOL	FALSE	MC_Home(原点復帰)を実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	MC_Home(原点復帰)が軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	MC_Home(原点復帰)の実行がエラーや多重起動などにより中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。  使用するコントローラのユーザーズマニュアル

機能

- 原点復帰制御では、機械原点の確立を行います。このとき、モーションシステムやドライバに記憶されているアドレス情報は使用しません。原点復帰後は、機械的に確立した位置を位置決め制御の起点である「原点」とします。



- 原点復帰方式には、「ドライバ式原点復帰」、「データセット式原点復帰」があります。
下記の条件により、原点復帰制御始動時の原点復帰方式となります。

原点復帰方式	原点復帰方式の条件
ドライバ式原点復帰	下記の条件をすべて満たす場合は、ドライバ式原点復帰となります。 • 軸種別が実ドライブ軸のとき。 • ドライバがHomingモードに対応しているとき。 • 「Home offset(607CH)」をオブジェクトデータに設定しているとき。
データセット式原点復帰	上記以外の場合は、データセット式原点復帰となります。

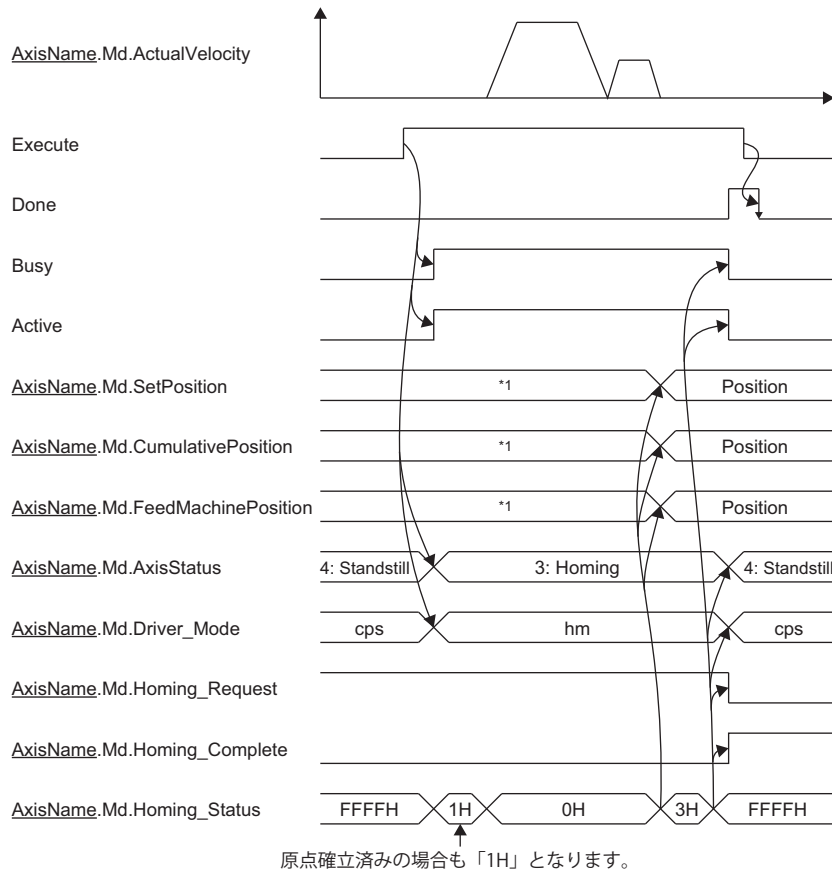
■ドライバ式原点復帰

ドライバをHomingモードに切り換え、ドライバ側で設定した位置決めパターンに基づいて原点復帰を行います。

原点復帰方式や各種パラメータを変更する場合は、MC_WriteParameter(パラメータ書込)でドライバの原点復帰データを変更してください。原点復帰の動作や設定可能なパラメータは、ドライバの仕様に依存します。ドライバについての詳細は、ドライバのマニュアルを参照してください。

• タイミングチャート

• 通常運転時



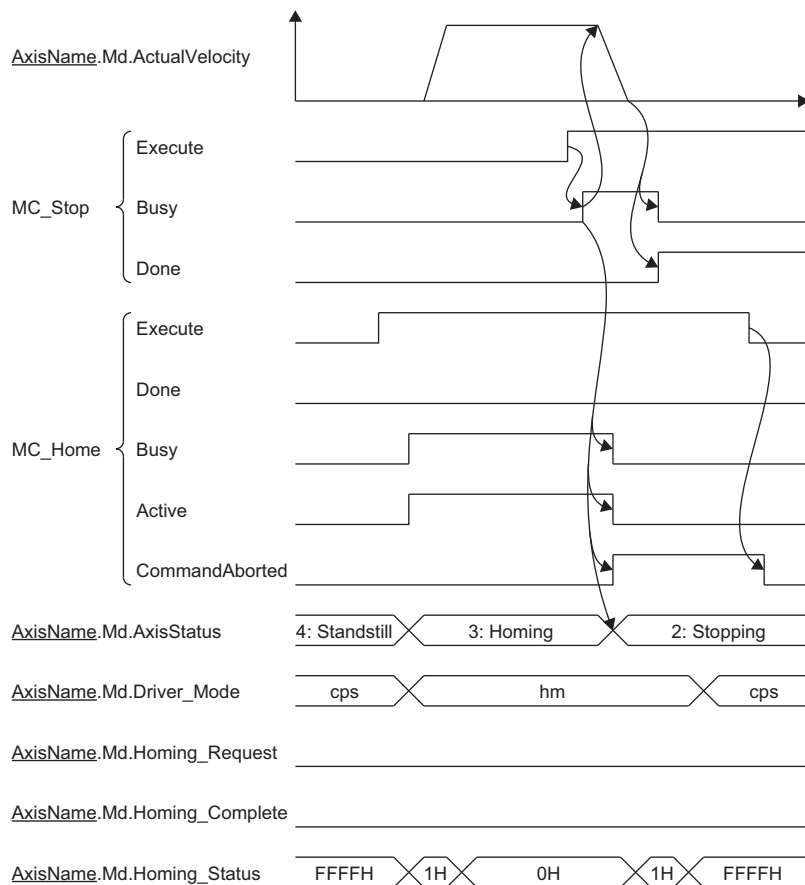
*1: フォローアップ有効/無効選択(オプション(Options): ビット16)が「0: フォローアップ無効」の場合、原点復帰中は更新しません。「1: フォローアップ有効」の場合、原点復帰中は現在位置の変動に合わせて更新します。

• 異常完了時

異常完了時のタイミングチャートについては、下記を参照してください。

1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

- MC_Stop(強制停止)実行時



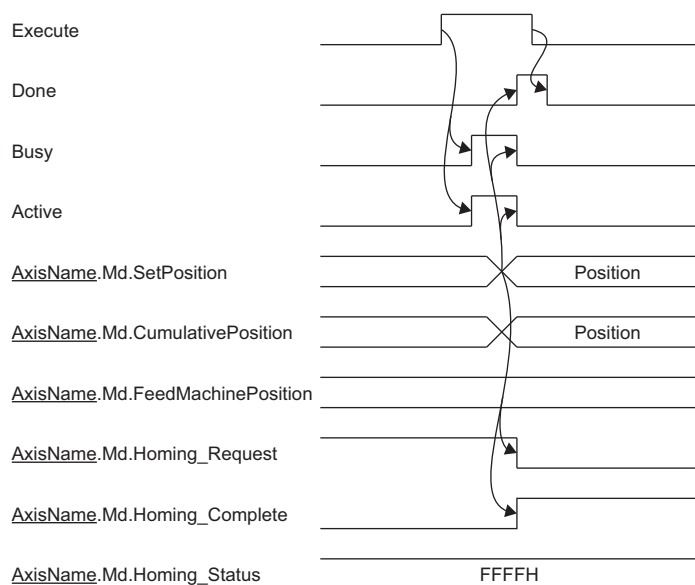
- 原点復帰時にMC_Stop(強制停止)をTRUEにした場合、ドライバに「HALT」信号を送信します。HALTに対応していないドライバを使用する場合、MC_Stop(強制停止)では停止しないため、緊急停止を使用してください。
- 原点復帰時の停止処理は、ドライバの仕様に従います。そのため、MC_Stop(強制停止)の減速度(Deceleration)とジャーク(Jerk)を無視します。

■データセット式原点復帰

仮想軸やデバイス機器側に原点情報を持たない実軸に対して原点復帰を行います。モーションシステム内部で完結し、外部信号などは使用しません。原点復帰を行ったときの目標位置(Position)を原点としてモーションシステムに登録し、指令現在位置(AxisName.Md.SetPosition)と累積現在位置(AxisName.Md.CumulativePosition)を目標位置(Position)に書き換えます。

- タイミングチャート

- 通常運転時



- 異常完了時

異常完了時のタイミングチャートについては、下記を参照してください。

1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■原点復帰要求について

- 原点復帰を実行する必要がある場合、モーションシステムは原点復帰要求(AxisName.Md.Homing_Request)をTRUEにします。原点復帰要求(AxisName.Md.Homing_Request)は、下記の場合にTRUEになります。

- システム電源を投入、またはリセットしたとき。^{*1}
- 軸種別を変更したとき。
- 原点復帰を始動したとき。(原点復帰が正常に完了しない限り、原点復帰要求はFALSEになりません。)
- 軸のドライバ単位変換(分子/分母)を変更したとき。
- 現在値復元を実行したとき。
- モーションシステム内の絶対位置データがメモリ異常などの要因により消失したとき。
- ドライバの電源を投入したとき。^{*1*2}
- ドライバ側で「絶対位置消失」を検出したとき。^{*2}
- オブジェクトデータの「Polarity(607EH)(b7: position polarity)」を変更したとき。^{*2}
- ドライバ側の電子ギアを変更したとき。^{*2}
- ドライバやモータエンコーダの変更を検出したとき。
- スケール計測エンコーダで絶対位置消失を検出したとき。^{*3}
- スケール計測エンコーダのエンコーダ分解能を変更したとき。^{*3}
- エンコーダリングカウンタ上限値/下限値を変更したとき。^{*4}

*1 絶対位置システムの場合、システム/ドライバ電源断、リセット時に原点復帰を完了している場合はTRUEになりません。

*2 実ドライブ軸のみ。

*3 実エンコーダ軸(ドライブユニット経由)のみ。

*4 仮想エンコーダ軸のみ。

Point

原点復帰要求(AxisName.Md.Homing_Request)がTRUEになると、イベント履歴に記録します。
イベント履歴で内容を確認できます。

- 原点復帰を行う必要のないシステムでは、原点復帰要求クリアを行ってください。原点復帰要求クリア(AxisName.Cd.Homing_ClearRequest)をTRUEにすることで、原点復帰要求(AxisName.Md.Homing_Request)は、FALSEになります。

Point

原点復帰要求(AxisName.Md.Homing_Request)をプログラム等で直接FALSEに書き換えた場合、モーションシステム内部の原点復帰要求はクリアしません。必ず原点復帰要求クリア(AxisName.Cd.Homing_ClearRequest)を使用して原点復帰要求をFALSEにしてください。

■原点スイッチ(AbsSwitch)

ドライバ式原点復帰でデバイス機器に伝達する近点ドグ信号を設定します。
原点スイッチ(AbsSwitch)は、MC_INPUT_REF構造体で設定します。MC_INPUT_REF構造体については、下記を参照してください。

📖 1420ページ MC_INPUT_REF

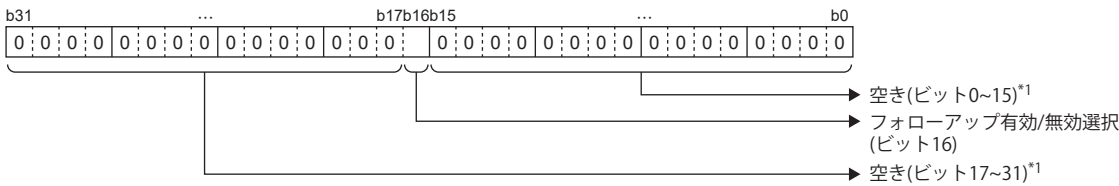
原点スイッチ(AbsSwitch)で設定する入力信号(AbsSwitch.Signal)の設定範囲を下記に示します。
トリガ信号(TriggerInput.Signal)の型は、SIGNAL_SELECT構造体となります。

構造体	変数名	型	設定範囲
SIGNAL_SELECT(信号選択)	Source(信号)	TARGET_REF 📖 1421ページ TARGET_REF(入力信号)	■型 • BOOL ■データ種別 • [OBJ] • [VAR] • [DEV] • [CONST]
	Detection(信号検出方法)	INT (MC_SIGNAL_LOGIC) 📖 1452ページ MC_SIGNAL_LOGIC	• 0: TRUE時検出(HighLevel) • 1: FALSE時検出(LowLevel)
	CompensationTime(補正時間)	LREAL	0.0[s]
	FilterTime(フィルタ時間)	LREAL	0.0~5.0[s]

- [OBJ]指定時に対象修飾を省略した場合は、軸(Axis)の軸のオブジェクトを参照します。ただし軸(Axis)の軸が局アドレスを持たない、または局アドレスが未設定の場合、外部信号選択不正(エラーコード: 1B31H)を出力します。
- [OBJ]指定時に参照する局が軸エミュレート機能有効かつ局アドレス設定済みの場合、エミュレートしているオブジェクトを参照します。
- 信号名(AbsSwitch.Signal.Source.Target)に使用できない信号を使用した場合、外部信号選択文字列不正(エラーコード: 1B05H, 外部信号選択不正(エラーコード: 1B31Hまたは1B32H), アドオンSignalIO内部異常(エラーコード: 1B33Hのいずれかを出力します。
- MELSERVO以外のドライブユニットを軸に指定している場合、原点スイッチで指定した信号をドグ信号としてドライブユニットに伝達できません。

■オプション(Options)

MC_Home(原点復帰)で使用する機能オプションをビット指定で設定します。
ビット指定で設定する内容を下記に示します。



*1 空きには「0」を設定してください。「0」以外を設定した場合、エラーになります。

ビット	名称	内容
16	フォローアップ有効/無効選択	ドライバ式原点復帰中に、指令現在位置のフォローアップを有効にするか、無効にするかを設定します。 • 0: フォローアップ無効 • 1: フォローアップ有効

注意事項

- 原点復帰中にMC_Stop(強制停止)以外のFBを起動できません。
- 原点復帰始動時の目標位置(Position)に対するソフトウェアストロークリミットチェックは、下記となります。

原点復帰方式	内容
ドライバ式原点復帰	ソフトウェアストロークリミットチェックを行いません。
データセット式原点復帰	ソフトウェアストロークリミットチェックを行います。

- 原点復帰始動時に始動位置のソフトウェアストロークリミットチェックは行いません。
- 原点復帰中は、ソフトウェアストロークリミットのチェックを行いません。
- 原点復帰完了時の移動方向は、正方向となります。
- 原点復帰は、速度バイアス機能非対応です。ドライバ式原点復帰時のモータ最小速度は、ドライバで設定を行ってください。

■ドライバ式原点復帰の注意事項

- サーボOFF中は、原点復帰を始動できません。そのため、ドライバの原点復帰方式(Homing method35, 37(データセット))をサーボOFF中に実行できません。
- モーションシステムに外部信号(軸変数で設定するハードウェアストロークリミット、および原点スイッチ(AbsSwitch)で設定する近点ドグ)の割付けをしている場合は、外部信号をドライバに伝達します。ただし近点ドグ信号を基準とする(Z相基準ではない)方式で位置精度が必要な場合は、ドライバ内蔵のDIを使用することを推奨します。
- 原点復帰が完了する直前に、フィードバック位置(AxisName.Md.ActualPosition)などのモニタデータが実際のモータ動作によらず急激に変動する場合があります。本現象が発生した場合でも原点復帰後のモニタデータには影響はありません。
- フォローアップ有効/無効選択(オプション(Options): ビット16)に「0: フォローアップ無効」を設定した場合、ドライバ式原点復帰中に、指令現在位置のフォローアップを行いません。このとき、原点復帰中の軸を主軸に設定して同期制御を実行しないでください。従軸にて制御中速度範囲オーバ(エラーコード: 1B1AH)が発生する場合があります。
- フォローアップ有効/無効選択(オプション(Options): ビット16)に「1: フォローアップ有効」を設定し、原点復帰中の軸を主軸に設定して同期制御を実行した場合、従軸は主軸データソース選択により下記の動作となります。

主軸データソース選択(MasterValueSource)	同期制御中の従軸動作
1: 指令現在(mcSetValue)101: 最新指令現在値(mcLatestSetValue)	原点復帰終了後も同期制御を継続します。
2: フィードバック値(mcActualValue) 102: 最新フィードバック値(mcLatestActualValue)	原点復帰完了まで主軸に同期しますが、終了時点で同期制御せずモータ急加速または過大な指令が入力されたことによりドライバ側でエラーまたはアラームが発生する可能性があります。詳細は各ドライバのマニュアルを参照してください。 例: MR-J5の場合は、サーボアラーム([AL.031.1(サーボモータ速度異常)], または[AL. 035.1(指令周波数異常)])が発生する可能性があります。

- 原点復帰動作状態(AxisName.Md.Homing_Status)が「0002H: 原点復帰完了(目標位置に未到達)」のときに停止要因が発生した場合、フォローアップした指令現在位置が大きく変動します。そのため、主軸データソース選択(MasterValueSource)にて、「1: 指令現在(mcSetValue)」, または「101: 最新指令現在値(mcLatestSetValue)」を設定した場合でも、ドライバ側でエラーまたはサーボアラームが発生する可能性があります。詳細は各ドライバのマニュアルを参照してください。
例: MR-J5の場合は、サーボアラーム([AL.031.1(サーボモータ速度異常)], または[AL. 035.1(指令周波数異常)])が発生する可能性があります。
- 下記のオブジェクトデータをマッピングしていない場合、バックアップの保証ができません。

Home cycle counter(2D3DH)

Home ABS counter(2D3EH)

- MR-J5(W)-G接続時、サーボアンプに入力したDOG信号を原点スイッチ信号として設定できます。設定方法の詳細は、下記を参照ください。

📖使用するコントローラのユーザーズマニュアル

- 原点復帰中は、ハードウェアストロークリミット信号の検出を行い、検出するとドライバに「HALT」信号を送信します。ドライバ側のリミットスイッチ信号を使用して停止を行う場合は、ハードウェアストロークリミットオーバーライド(AxisName.Cd.HwStrokeLimit_Override)を「DISABLE(チェック無効)」にして、ハードウェアストロークリミットチェックを一時的に無効にしてください。
- 原点復帰始動時、ハードウェアストロークリミットオーバーライド(AxisName.Cd.HwStrokeLimit_Override)に「ONLY_INSIDE(範囲内方向へのみチェック無効)」を設定している場合、始動不可(エラーコード: 1AFCH)となり、原点復帰始動を行いません。
- MELSERVO以外のドライブユニットで、原点復帰を実施する場合、「Home offset(607CH)」が0となるようにドライブユニットのパラメータを設定してください。0以外を設定した場合、原点復帰完了時に意図しない動作となる可能性があります。

プログラム例

原点復帰指令(bHomingCMD)をTRUEにし，原点スイッチ「[DEV](BOOL)D0」により，軸1(Axis0001)の原点復帰を行うプログラム例を下記に示します。

■軸

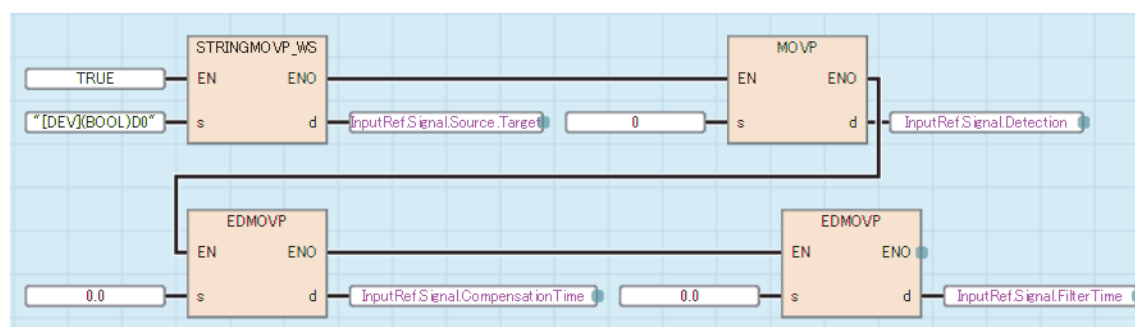
軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■使用するラベル

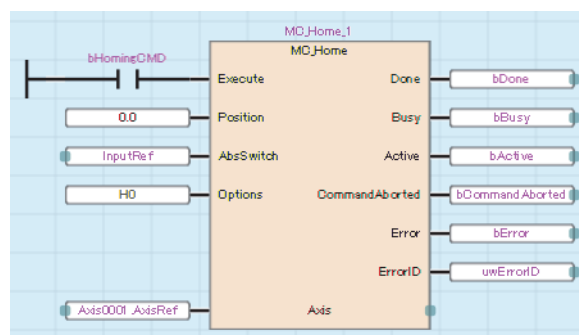
ラベル名	データ型	コメント
MC_Home_1	MC_Home	原点復帰FB
bHomingCMD	ビット	原点復帰指令
InputRef	MC_INPUT_REF	原点スイッチ
bDone	ビット	実行完了
bBusy	ビット	実行中
bActive	ビット	制御中
bCommandAborted	ビット	実行中断
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

- 原点スイッチデータ設定



- 原点復帰



■STプログラム

//-----原点スイッチデータ設定-----

InputRef.Signal.Source.Target:= "[DEV](BOOL)D0";

InputRef.Signal.Detection:= MC_SIGNAL_LOGIC__HighLevel;

InputRef.Signal.CompensationTime:= 0.0;

InputRef.Signal.FilterTime:= 0.0;

//-----原点復帰-----

MC_Home_1(

Axis:= Axis0001.AxisRef ,

Execute:= bHomingCMD ,

Position:= 0.0 ,

AbsSwitch:= InputRef ,

Options:= H00000000 ,

Done=> bDone ,

Busy=> bBusy ,

Active=> bActive ,

CommandAborted=> bCommandAborted ,

Error=> bError ,

ErrorID=> uwErrorID

);

42.2 強制停止

42

MC_Stop

指定した軸を減速停止します。

ラダー	FBD/LD	ST
		<pre>MC_Stop(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, Deceleration:= ?LREAL?, Jerk:= ?LREAL?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📖 1378ページ AxisName.AxisRef. (軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_Stop(強制停止)を実行します。
Deceleration	減速度	LREAL	起動時/ 再起動可	0.0000, 0.0001~2147483647.0	0.0	減速度を設定します。 加減速方式により設定内容が異なります。 詳細は、下記マニュアルの「加減速処理機能」を参照してください。 📖 使用するコントローラのユーザズマニュアル *: 加減速方式は、実行中の制御で指定した方式を引き継ぎます。
Jerk	ジャーク	LREAL	起動時/ 再起動可	0.0	0.0	「0.0」を設定してください。 *: 「0.0」以外を設定した場合、ジャーク範囲外(エラーコード: 1A8FH)となります。
Options	オプション	DWORD(HEX)	起動時	00000000H	00000000H	「00000000H」を設定してください。 *: 「00000000H」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

■出力変数

出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	速度0に到達したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MC_Stop(強制停止)を実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	MC_Stop(強制停止)が軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	MC_Stop(強制停止)の実行が中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 📖 使用するコントローラのユーザズマニュアル

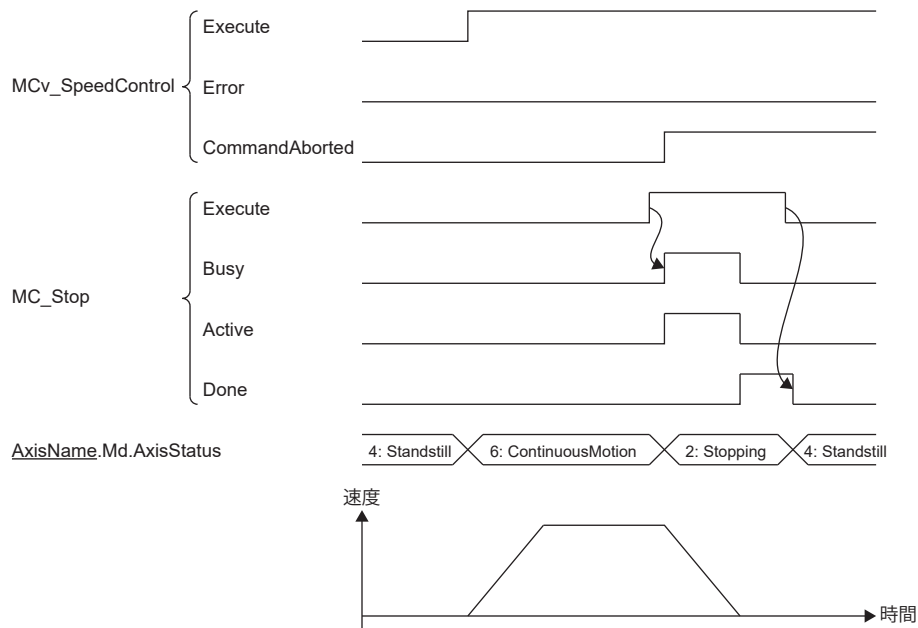
機能

- MC_Stop(強制停止)は、減速度(Deceleration)を設定し、現在制御中のFBを減速停止します。
- MC_Stop(強制停止)を実行すると現在実行中のFBは、実行中断(CommandAborted)がTRUEになり、軸状態(AxisName.Md.AxisStatus)は「2: 減速停止中(Stopping)」に遷移します。実行指令(Execute)がTRUEのとき、または速度が「0.0」に未達のときは「2: 減速停止中(Stopping)」を保持します。停止完了にて実行完了(Done)がTRUE、実行指令(Execute)がFALSEになると「4: 待機中(Standstill)」に遷移します。

■タイミングチャート

- 正常完了の場合

MCv_SpeedControl(速度制御(位置ループを含む))中にMC_Stop(強制停止)を実行した場合



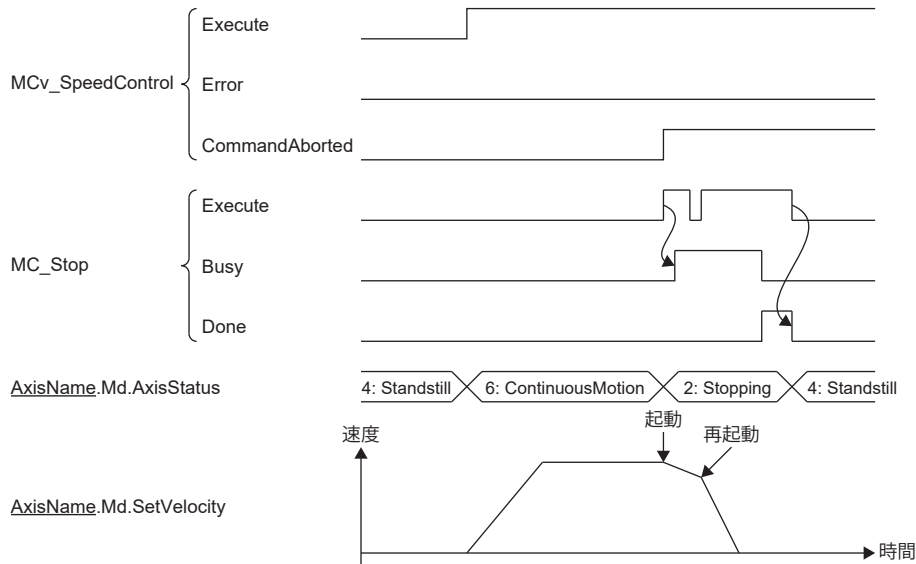
- 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

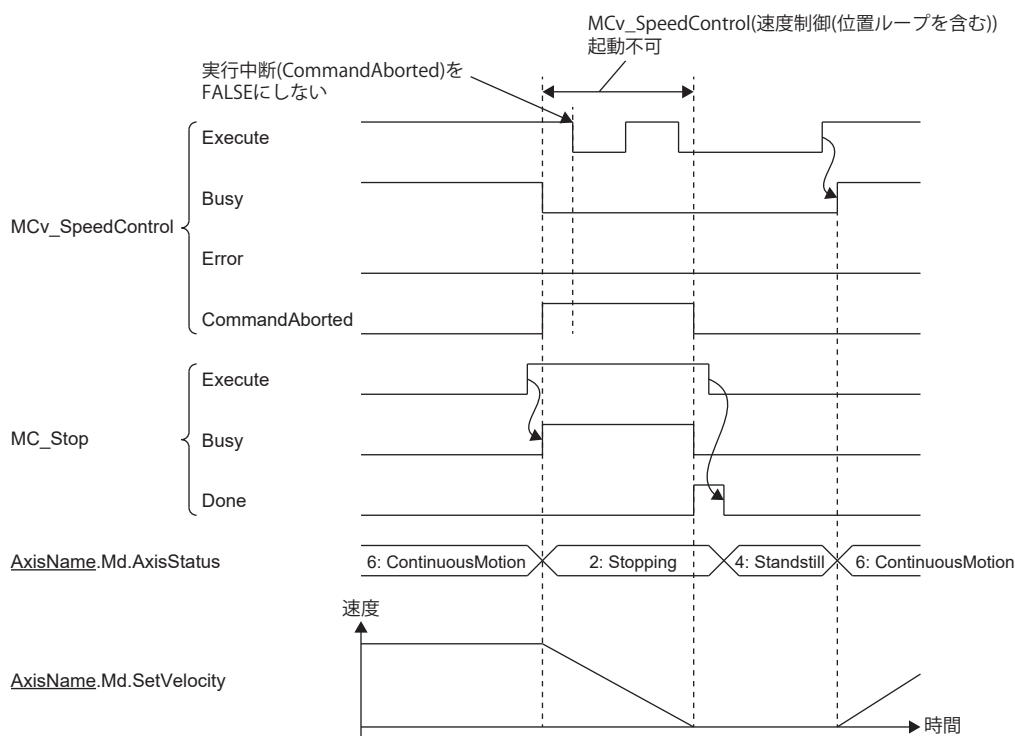
📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■動作概要

- MC_Stop(強制停止)による減速停止中と入力変数の実行指令(Execute)がTRUEのときは、MC_Stop(強制停止)以外の動作指令を受け付けません。
- 減速度(Deceleration)を「0.0」、または省略すると即停止します。
- 加減速方式およびジャークは、実行中の制御で設定した方式を引き継いで減速します。
- 単軸同期制御中にMC_GroupStop(グループ強制停止)を実行した場合、主軸への同期は中止します。
- 減速度(Deceleration)の設定値を変更してMC_Stop(強制停止)を再起動した場合、再起動した時点からMC_Stop(強制停止)に設定した減速度(Deceleration)を基に減速停止を行います。



- 軸状態(AxisName.Md.AxisStatus)が「2: 減速停止中(Stopping)」中にMC_Stop(強制停止)以外の軸を運転するFBを実行した場合、始動不可(エラーコード: 1AFCH)となり、MC_Stop(強制停止)のエラー (Error)がTRUEになり、軸状態(AxisName.Md.AxisStatus)が「1: エラー停止中(ErrorStop)」になります。
- 軸停止後、軸エラーリセット(AxisName.Cd.ErrorReset)を実行することで「4: 待機中(Standstill)」へ遷移します。
- MC_Stop(強制停止)により実行中断したFBの実行指令(Execute)をFALSEにしても、いったん停止動作が開始すると停止完了するまで実行中断(CommandAborted)はTRUEを継続します。停止完了後、実行中断したFBの実行指令(Execute)がFALSEの場合、実行中断(CommandAborted)がFALSEになります。



プログラム例

軸停止要求(bStop)をTRUEにし、減速度「100000.0」で軸1(Axis0001)を減速停止させるプログラム例を下記に示します。

■軸

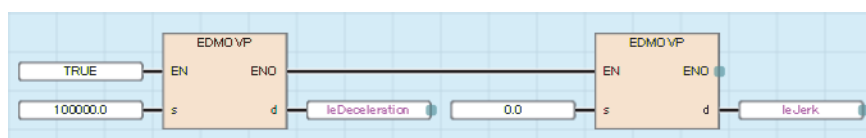
軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■使用するラベル

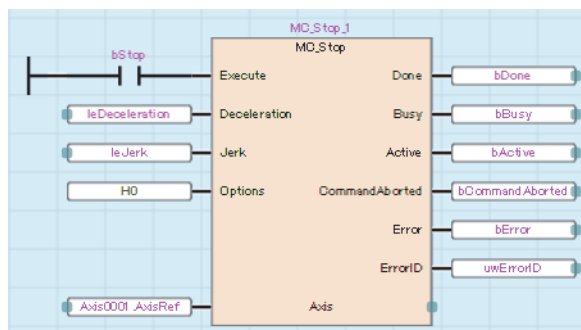
ラベル名	データ型	コメント
MC_Stop_1	MC_Stop	軸停止FB
bStop	ビット	軸停止要求
leDeceleration	倍精度実数	減速度
leJerk	倍精度実数	ジャーク
bDone	ビット	完了
bBusy	ビット	実行中
bActive	ビット	制御中
bCommandAborted	ビット	実行中断
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

- 軸停止用データ設定



- 軸停止



■STプログラム

//-----軸停止用データ設定-----

leDeceleration:= 100000.0;

leJerk:= 0.0;

//-----軸停止-----

MC_Stop_1(

Axis:= Axis0001.AxisRef ,

Execute:= bStop ,

Deceleration:= leDeceleration ,

Jerk:= leJerk ,

Options:= H00000000 ,

Done=> bDone ,

Busy=> bBusy ,

Active=> bActive ,

CommandAborted=> bCommandAborted ,

Error=> bError ,

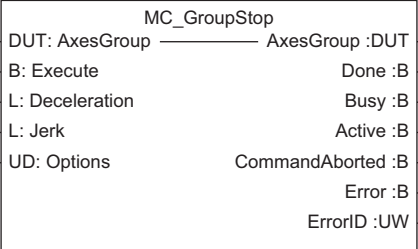
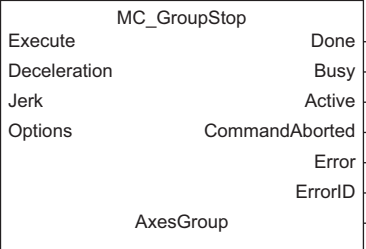
ErrorID=> uwErrorID

);

42.3 グループ強制停止

MC_GroupStop

指定した軸グループを減速停止します。

ラダー	FBD/LD	ST
		<pre>MC_GroupStop(AxesGroup:= ?AXES_GROUP_REF?, Execute:= ?BOOL?, Deceleration:= ?LREAL?, Jerk:= ?LREAL?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
AxesGroup	軸グループ情報	AXES_GROUP_REF	起動時	—	省略不可	軸グループを設定します。 使用する変数(AxesGroupName.AxesGroupRef.)については、下記を参照してください。 1402ページ AxesGroupName.AxesGroupRef. (軸グループ情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_GroupStop(グループ強制停止)を実行します。
Deceleration	減速度	LREAL	起動時/ 再起動可	0.0000, 0.0001~2147483647.0	0.0	減速度を設定します。 加減速方式により設定内容が異なります。 詳細は、下記マニュアルの「加減速処理機能」を参照してください。 □□使用するコントローラのユーザーズマニュアル *: 加減速方式は、実行中の制御で指定した方式を引き継ぎます。
Jerk	ジャーク	LREAL	起動時	0.0	0.0	「0.0」を設定してください。 *: 「0.0」以外を設定した場合、ジャーク範囲外(エラーコード: 1A8FH)となります。
Options	オプション	DWORD(HEX)	起動時	00000000H	00000000H	「00000000H」を設定してください。 *: 「00000000H」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

■出力変数

出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	速度0に到達したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	速度0への動作中のときに、TRUEになります。
Active	制御中	BOOL	FALSE	MC_GroupStop(グループ強制停止)が軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	MC_GroupStop(グループ強制停止)の実行が中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 □□使用するコントローラのユーザーズマニュアル

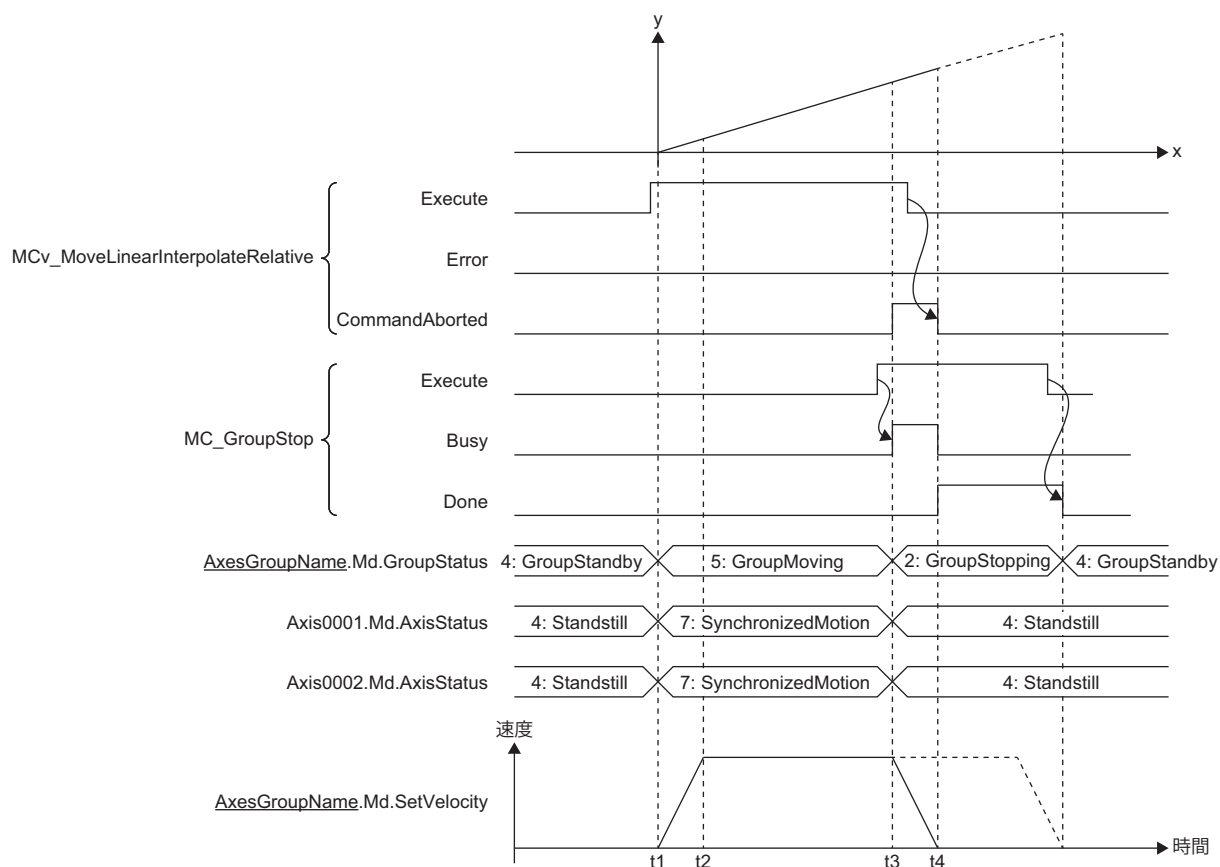
機能

- MC_GroupStop(グループ強制停止)は、減速度(Deceleration)を設定し、現在制御中のFBを減速停止します。軸グループは直前の動作の軌跡上を通り減速停止します。
- MC_GroupStop(グループ強制停止)を実行すると現在実行中のFBは、実行中断(CommandAborted)がTRUEになり、軸グループ状態(AxesGroupName.Md.GroupStatus)は、「2: 減速停止中(GroupStopping)」に遷移します。実行指令(Execute)がTRUEのとき、または速度が「0.0」に未達のときは「2: 減速停止中(GroupStopping)」を保持します。停止完了にて実行完了(Done)がTRUE、実行指令(Execute)がFALSEになると「4:待機中(GroupStandby)」に遷移します。

■タイミングチャート

- 正常完了の場合

MCv_MoveLinearInterpolateRelative中(相対値直線補間制御)にMC_GroupStop(グループ強制停止)を実行した場合



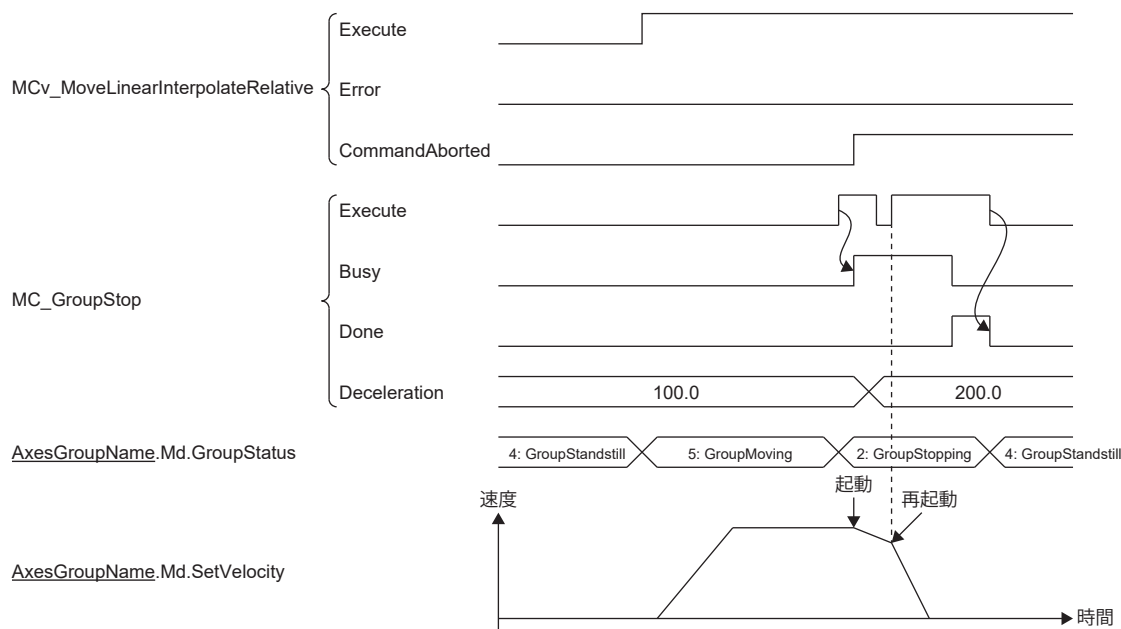
- 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

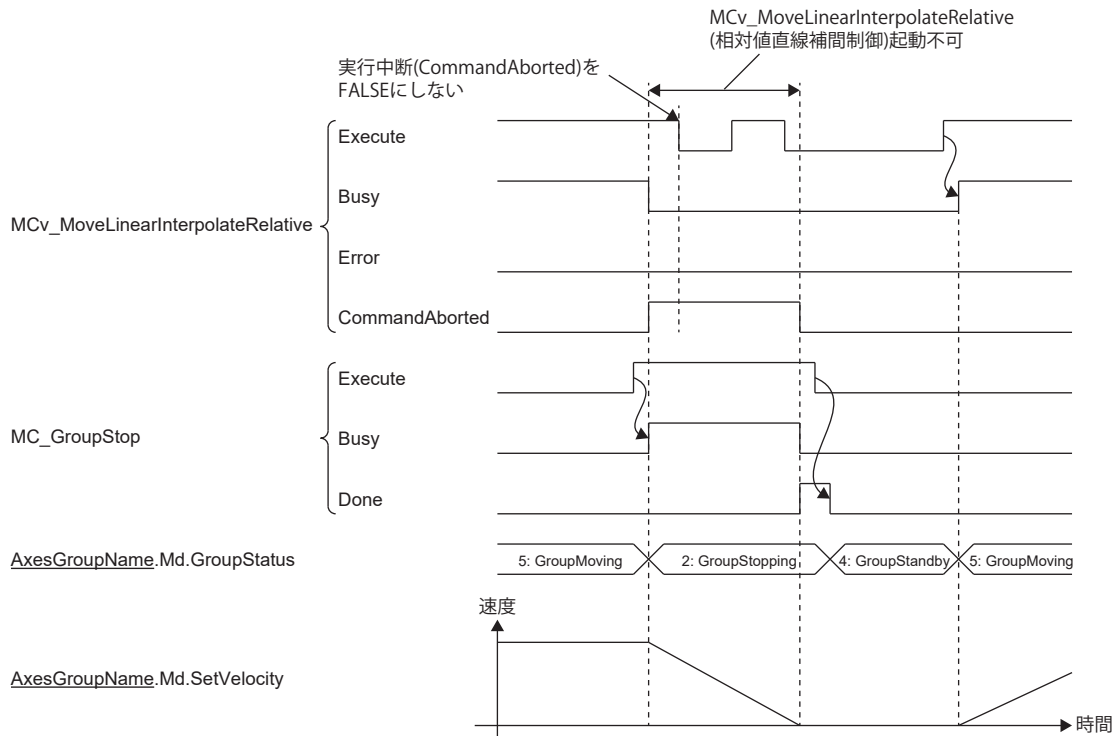
📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■動作概要

- MC_GroupStop(グループ強制停止)による減速停止中と入力変数の実行指令(Execute)がTRUEのときは、MC_GroupStop(グループ強制停止)以外の動作指令を受け付けません。
- 減速度(Deceleration)を「0.0」、または省略すると即停止します。
- 加減速方式およびジャークは、実行中の制御で指定した方式を引き継いで減速します。
- 減速度(Deceleration)の設定値を変更してMC_GroupStop(グループ強制停止)を再起動した場合、再起動した時点からMC_GroupStop(グループ強制停止)に設定した減速度(Deceleration)、制御中のFBで使用しているジャーク(Jerk)を基に減速停止を行います。



- 軸グループ状態(AxesGroupName.Md.GroupStatus)が「2: 減速停止中(GroupStopping)」中にMC_GroupStop(グループ強制停止)以外の軸を運転するFBを実行した場合、始動不可(エラーコード: 1AFCH)となり、MC_GroupStop(グループ強制停止)のエラー (Error)がTRUEになり、軸グループ状態(AxesGroupName.Md.GroupStatus)が「1: エラー停止中(GroupErrorStop)」になります。
- MC_GroupStop(グループ強制停止)により実行中断したFBの実行指令(Execute)をFALSEにしても、いったん停止動作が開始すると停止完了するまで実行中断(CommandAborted)はTRUEを継続します。停止完了後、実行中断したFBの実行指令(Execute)がFALSEの場合、実行中断(CommandAborted)がFALSEになります。



プログラム例

軸グループ停止要求(bGroupStop)をTRUEにし、減速度「100000.0」で軸グループ1(AxesGroup001)を減速停止させるプログラム例を下記に示します。

■軸グループ

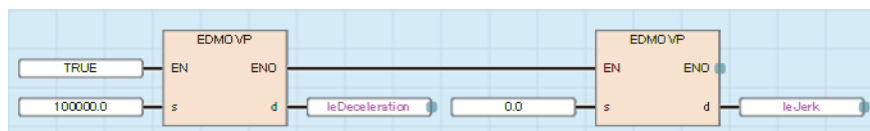
軸グループNo.	ラベル名	データ型	コメント
1	AxesGroup001	AXES_GROUP_REF	軸グループ1

■使用するラベル

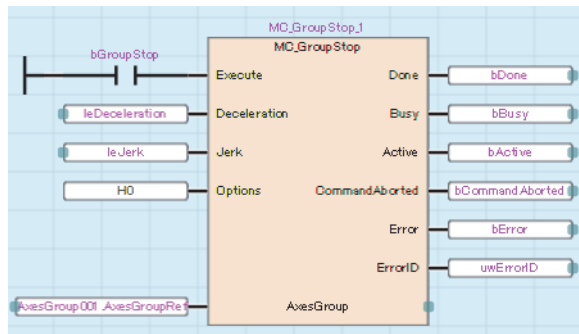
ラベル名	データ型	コメント
MC_GroupStop_1	MC_GroupStop	軸グループ停止FB
bGroupStop	ビット	軸グループ停止要求
leDeceleration	倍精度実数	減速度
leJerk	倍精度実数	ジャーク
bDone	ビット	完了
bBusy	ビット	実行中
bActive	ビット	制御中
bCommandAborted	ビット	実行中断
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

- 軸グループ停止用データ設定



- 軸グループ停止



■STプログラム

//-----軸グループ停止用データ設定-----

leDeceleration:= 100000.0;

leJerk:= 0.0;

//-----軸グループ停止-----

MC_GroupStop_1(

 AxesGroup:= AxesGroup001.AxesGroupRef ,

 Execute:= bGroupStop ,

 Deceleration:= leDeceleration ,

 Jerk:= leJerk ,

 Options:= H00000000 ,

 Done=> bDone ,

 Busy=> bBusy ,

 Active=> bActive ,

 CommandAborted=> bCommandAborted ,

 Error=> bError ,

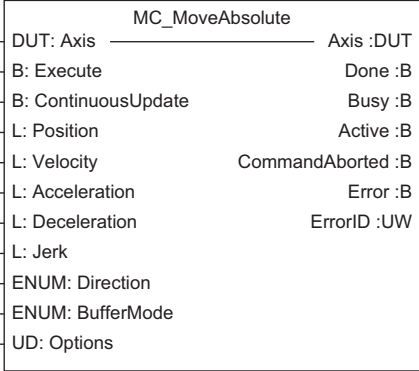
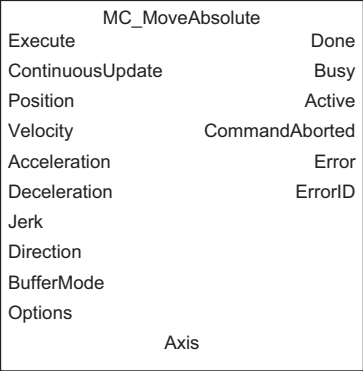
 ErrorID=> uwErrorID

);

42.4 絶対値位置決め

MC_MoveAbsolute

絶対位置の目標位置を設定し、位置決めを実行します。

ラダー	FBD/LD	ST
		<pre>MC_MoveAbsolute(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, ContinuousUpdate:= ?BOOL?, Position:= ?LREAL?, Velocity:= ?LREAL?, Acceleration:= ?LREAL?, Deceleration:= ?LREAL?, Jerk:= ?LREAL?, Direction:= ?INT?, BufferMode:= ?INT?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📖 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_MoveAbsolute(絶対値位置決め)を実行します。
Continuous Update	連続更新	BOOL	起動時	TRUE, FALSE	FALSE	連続して、目標位置(Position)、速度(Velocity)、加速度(Acceleration)、減速度(Deceleration)の変更を有効にするか、無効にするかを設定します。 • FALSE: 無効 • TRUE: 有効
Position	目標位置	LREAL	起動時/再起動可/連続更新可	-100000000000.0~10000000000.0	0.0	絶対位置の目標位置を設定します。 設定可能な範囲は、各設定により異なります。 詳細は、下記を参照してください。 📖 1592ページ 目標位置(Position)
Velocity	速度	LREAL	起動時/再起動可/連続更新可	0.0, 0.0001~2500000000.0	0.0	速度を設定します。 詳細は、下記を参照してください。 📖 1593ページ 速度(Velocity)
Acceleration	加速度	LREAL	起動時/再起動可/連続更新可	0.0000, 0.0001~2147483647.0	0.0	加速度を設定します。 詳細は、下記を参照してください。 📖 1593ページ 加速度(Acceleration)
Deceleration	減速度	LREAL	起動時/再起動可/連続更新可	0.0000, 0.0001~2147483647.0	0.0	減速度を設定します。 詳細は、下記を参照してください。 📖 1593ページ 減速度(Deceleration)
Jerk	ジャーク	LREAL	起動時	0.0000, 0.0001~2147483647.0	0.0	ジャークを設定します。 詳細は、下記を参照してください。 📖 1593ページ ジャーク(Jerk)

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Direction	方向選択	INT (MC_DIRECTION)	起動時	1~3	0	ソフトウェアストロークリミット無効時に、現在位置から目標位置へ移動する方向を設定します。 ・ 1: 正方向(mcPositiveDirection) ・ 2: 負方向(mcNegativeDirection) ・ 3: 最短経路(mcShortestWay) *: 設定を省略した場合、方向選択範囲外(エラーコード: 1AA5H)となります。 詳細は、下記を参照してください。 1594ページ 方向選択(Direction)
BufferMode	バッファモード	INT (MC_BUFFER_MODE)	起動時	0~5	0	バッファモードを設定します。 ・ 0: Aborting(mcAborting) ・ 1: Buffered(mcBuffered) ・ 2: BlendingLow(mcBlendingLow) ・ 3: BlendingPrevious(mcBlendingPrevious) ・ 4: BlendingNext(mcBlendingNext) ・ 5: BlendingHigh(mcBlendingHigh) 詳細は、下記を参照してください。 1597ページ バッファモード(BufferMode)
Options	オプション	DWORD(HEX)	起動時	00000000H~00010021H	00000000H	MC_MoveAbsolute(絶対値位置決め)の機能オプションを設定します。 詳細は、下記を参照してください。 1598ページ オプション(Options)

■出力変数

出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	目標位置へ到達したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MC_MoveAbsolute(絶対値位置決め)を実行中のときに、TRUEになります。 目標位置に到達した後、FALSEになります。
Active	制御中	BOOL	FALSE	MC_MoveAbsolute(絶対値位置決め)が軸を制御中のときに、TRUEになります。 目標位置に到達した後、FALSEになります。
CommandAborted	実行中断	BOOL	FALSE	MC_MoveAbsolute(絶対値位置決め)の実行が中断したときに、TRUEになります。 実行指令(Execute)がFALSEになると、FALSEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 使用するコントローラのユーザーズマニュアル

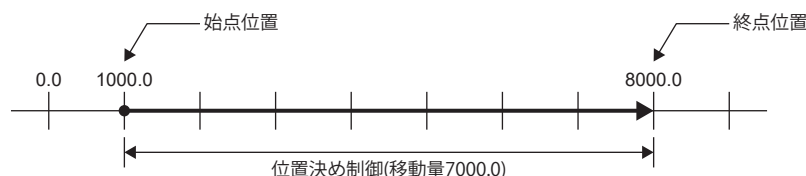
機能

- 目標位置(Position)、速度(Velocity)、加速度(Acceleration)、減速度(Deceleration)、ジャーク(Jerk)、方向選択(Direction)、バッファモード(BufferMode)、オプション(Options)を設定し、始動時の現在位置(始点位置)から、目標位置(Position)に設定した指定位置(終点位置)へ位置決めを行います。

例

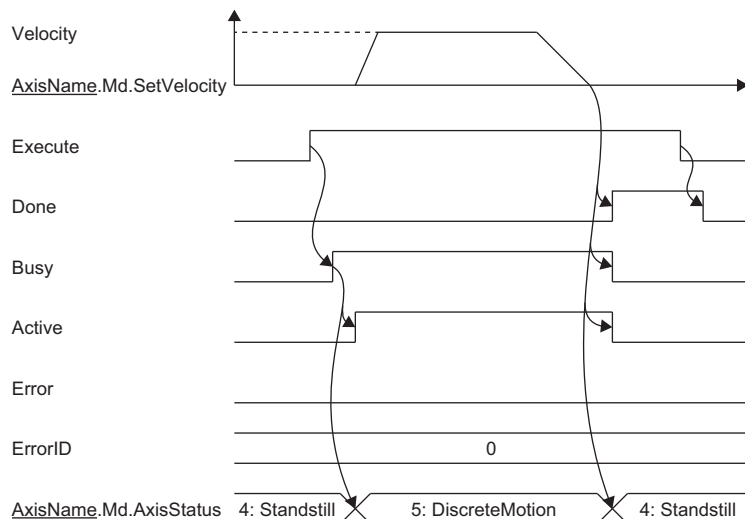
始点位置(現在の停止位置)が「1000.0」のとき、目標位置(Position)に「8000.0」を設定した場合

- 正方向に移動量が「7000.0(8000.0 - 1000.0)」の位置決めを行います。



■タイミングチャート

・正常完了の場合



・異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■目標位置(Position)

絶対位置の目標位置を設定します。有効範囲は、ソフトウェアストロークリミット有効/無効、方向選択(Direction)、リングカウンタを超えた目標位置指定(オプション(Options): ビット16)により異なります。

- ・ソフトウェアストロークリミット有効時(ソフトウェアストロークリミット対象(AxisName.Md.SwStrokeLimit_Target)が「-1: 無効(Invalid)」以外のとき)

方向選択(Direction)、リングカウンタを超えた目標位置指定(オプション(Options): ビット16)の設定に関係なく、有効範囲は下記となります。

有効範囲

リングカウンタ下限値 ≤ 目標位置 < リングカウンタ上限値

範囲外の値を設定した場合、目標位置範囲外(エラーコード: 1A85H)となり始動しません。また、有効範囲内であってもソフトウェアストロークリミットオーバとなる位置を目標位置として設定した場合、ソフトウェアストロークリミットオーバ(目標位置)(エラーコード: 1A80H)となり始動しません。

- ・ソフトウェアストロークリミット無効時(ソフトウェアストロークリミット対象(AxisName.Md.SwStrokeLimit_Target)が「-1: 無効(Invalid)」のとき)

「リングカウンタを超えた目標位置指定(オプション(Options): ビット16)」を「1: 許可する」に設定することで、リングカウンタ範囲1回転以上の位置決めができます。

範囲外の値を設定した場合、目標位置範囲外(エラーコード: 1A85H)となり始動しません。

方向選択(Direction)	リングカウンタを超えた目標位置指定(オプション(Options): ビット16)	有効範囲
1: 正方向(mcPositiveDirection)	0: 許可しない	リングカウンタ下限値 ≤ 目標位置 < リングカウンタ上限値
	1: 許可する	リングカウンタ下限値 ≤ 目標位置 < 位置決め範囲上限値
2: 負方向(mcNegativeDirection)	0: 許可しない	リングカウンタ下限値 ≤ 目標位置 < リングカウンタ上限値
	1: 許可する	位置決め範囲下限値 ≤ 目標位置 < リングカウンタ上限値
3: 最短経路(mcShortestWay)	0: 許可しない	リングカウンタ下限値 ≤ 目標位置 < リングカウンタ上限値
	1: 許可する	リングカウンタ下限値 ≤ 目標位置 < リングカウンタ上限値

- ・目標位置へ移動する方向については、下記を参照してください。

📖 1594ページ 方向選択(Direction)

■速度(Velocity)

MC_MoveAbsolute(絶対値位置決め)での指令速度を設定します。

設定範囲
0.0, 0.0001~2500000000.0 ^{*1*2}

- *1 浮動小数点演算を行うため、指令速度の下限値には下記の制約が生じます。
指令速度を演算周期換算した速度が「0.00001」未満となる場合、演算周期換算速度範囲外(エラーコード: 1B16H)(速度変更時は演算周期換算速度範囲外警告(イベントコード: 00D2FH))となります。浮動小数点演算の精度を向上させるためには位置指令単位(AxisName.Pr.Unit_Position)、または速度指令単位(AxisName.Pr.Unit_Velocity)を変更することで演算周期換算後の速度が「0.00001」未満とならないように設定してください。
- *2 多重起動を行うモーション制御FBにて指定速度が「0.0」の場合、直前のモーション制御FBの指定速度となります。

■加速度(Acceleration)

MC_MoveAbsolute(絶対値位置決め)での加速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の設定により設定範囲が異なります。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	0.000000 ^{*1} , 0.000001~8400.0[s]までの正数

- *1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更、減速度変更時は、変更を受け付けません。

■減速度(Deceleration)

MC_MoveAbsolute(絶対値位置決め)での減速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合、減速度(Deceleration)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

- *1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更、減速度変更時は、変更を受け付けません。

■ジャーク(Jerk)

MC_MoveAbsolute(絶対値位置決め)でのジャークを設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合、ジャーク(Jerk)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000, 0.0001~2147483647.0[U/s ³]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

■方向選択(Direction)

ソフトウェアストロークリミット有効時は、方向選択(Direction)の設定を無視します。ソフトウェアストロークリミット範囲外を跨がない方向での位置決め制御を行います。ただし、正方向、負方向ともにソフトウェアストロークリミット範囲外を跨がない場合、現在位置を基準にして、目標位置へ近い方向(移動量の絶対値が短い方)での位置決め制御を行います。正方向、負方向で距離が同じ場合は現在方向で動作します。

ソフトウェアストロークリミット無効時は、方向選択(Direction)の「1: 正方向(mcPositiveDirection)」、「2: 負方向(mcNegativeDirection)」、「3: 最短経路(mcShortestWay)」から選択でき、現在位置から目標位置へ移動する方向を設定します。

- 1: 正方向(mcPositiveDirection)

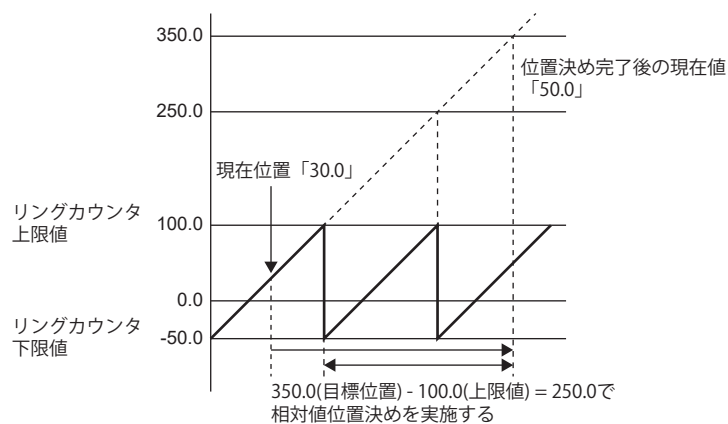
現在位置から目標位置に向かって正方向(アドレス増加)の位置決めを行います。

リングカウンタを超えた目標位置指定(オプション(Options): ビット16)	設定範囲	動作
■0: 許可しない 目標位置の設定範囲は、「リングカウンタ下限値 ≤ 目標位置 < リングカウンタ上限値」です。	現在値 ≤ 目標位置 < リングカウンタ上限値	リングカウンタ上限値 現在値「0.0」 リングカウンタ下限値
	リングカウンタ下限値 ≤ 目標位置 < 現在値	リングカウンタ上限値 現在値「0.0」 リングカウンタ下限値
■1: 許可する リングカウンタ上限値をオーバフローする位置決め運転が可能となります。この場合、リングカウンタ上限値を基準に、超えた分の移動量を相対移動量として位置決めを行います。 リングカウンタ下限値よりも小さな目標位置を設定した場合は、目標位置範囲外(エラーコード: 1A85H)となり始動しません。	現在値 ≤ 目標位置 < リングカウンタ上限値 リングカウンタ下限値 ≤ 目標位置 < 現在値	リングカウンタ上限値 現在値「0.0」 リングカウンタ下限値
	リングカウンタ上限値 ≤ 目標位置 < 位置決め範囲上限値	位置決め範囲上限値 リングカウンタ上限値「100.0」 現在値「0.0」 リングカウンタ上限値 リングカウンタ下限値

例

下記の値を設定した場合の動作(リングカウンタ上限値を超えた後の相対値位置決めを行います。)

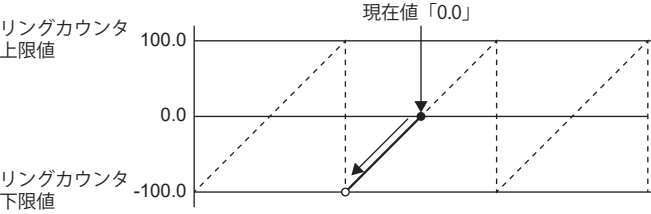
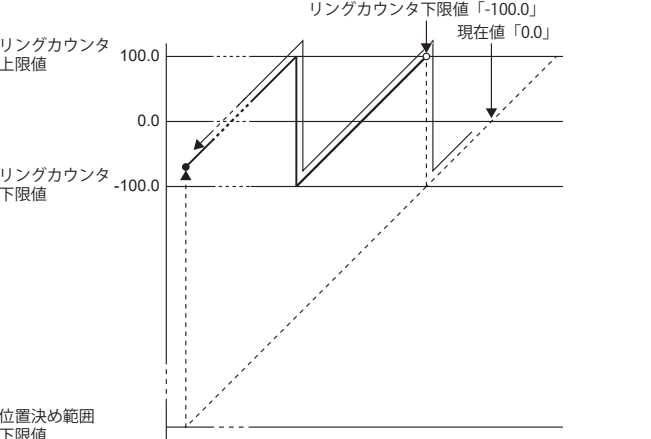
- リングカウンタ上限値: 100.0
- リングカウンタ下限値: -50.0
- 現在位置: 30.0
- 目標位置: 350.0



• 2: 負方向(mcNegativeDirection)

現在位置から負方向(アドレス減少)の目標位置に向かって位置決めを行います。

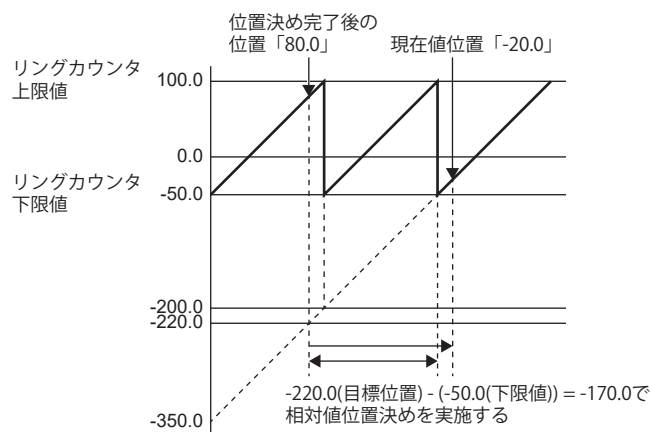
リングカウンタを超えた目標位置指定(オプション(Options): ビット16)	設定範囲	動作
■0: 許可しない 目標位置の設定範囲は、「リングカウンタ下限値 ≤ 目標位置 < リングカウンタ上限値」です。	リングカウンタ下限値 ≤ 目標位置 ≤ 現在値	現在値「0.0」 リングカウンタ上限値 リングカウンタ下限値
	現在値 < 目標位置 < リングカウンタ上限値	現在値「0.0」 リングカウンタ上限値 リングカウンタ下限値

リングカウンタを超えた目標位置指定(オプション)(Options): ビット16)	設定範囲	動作
■1: 許可する リングカウンタ下限値をアンダーフローする位置決め運転が可能となります。この場合、リングカウンタ下限値を基準に、超えた分の移動量を相対移動量として位置決めを行います。 リングカウンタ上限値以上の目標位置を設定した場合は、目標位置範囲外(エラーコード: 1A85H)となり始動しません。	- リングカウンタ下限値 ≤ 目標位置 ≤ 現在値 - 現在値 < 目標位置 < リングカウンタ上限値	 リングカウンタ上限値 リングカウンタ下限値 現在値「0.0」
	位置決め範囲下限値 ≤ 目標位置 < リングカウンタ下限値	 リングカウンタ上限値 リングカウンタ下限値 位置決め範囲下限値 リングカウンタ下限値「-100.0」 現在値「0.0」

例

下記の値を設定した場合の動作

- ・リングカウンタ上限値: 100.0
- ・リングカウンタ下限値: -50.0
- ・現在位置: -20.0
- ・目標位置: -220.0



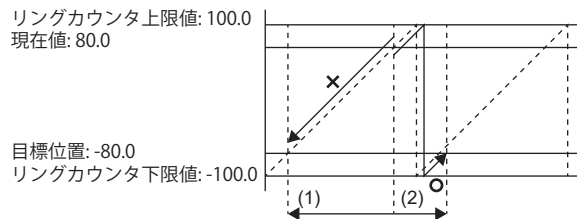
• 3: 最短経路(mcShortestWay)

現在位置を基準にして、目標位置へ近い方向(移動量の絶対値が短い方)での位置決め制御を行います。正方向、負方向で距離が同じ場合は現在方向で動作します。リングカウンタを超えた目標位置指定(オプション(Options): ビット16)の設定にかかわらず、目標位置の設定範囲は、「リングカウンタ下限値 ≤ 目標位置 < リングカウンタ上限値」です。範囲外の値を設定した場合は、目標位置範囲外(エラーコード: 1A85H)となり始動しません。

例

下記の値を設定した場合の動作

- リングカウンタ上限値: 100.0
- リングカウンタ下限値: -100.0
- 現在位置: 80.0
- 目標位置: -80.0



*1 (1)の場合は移動量が「160.0」、(2)の場合は移動量が「40.0」となるため、(2)の正方向で移動します。

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

MC_MoveAbsolute(絶対値位置決め)で設定可能なバッファモードを下記に示します。

設定値	内容
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。
1: Buffered(mcBuffered)	制御中のFBに次のFBをバッファリングします。 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFB完了にて、バッファリングFBを順次実行します。
2: BlendingLow(mcBlendingLow)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBとバッファリングFBの目標速度のうち低い方の速度を切換え速度とします。
3: BlendingPrevious(mcBlendingPrevious)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBの目標速度を切換え速度とします。
4: BlendingNext(mcBlendingNext)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、バッファリングFBの目標速度を切換え速度とします。
5: BlendingHigh(mcBlendingHigh)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBとバッファリングFBの目標速度のうち高い方の速度を切換え速度とします。

*1 制御中のFBとバッファリングFB間で停止を行いません。

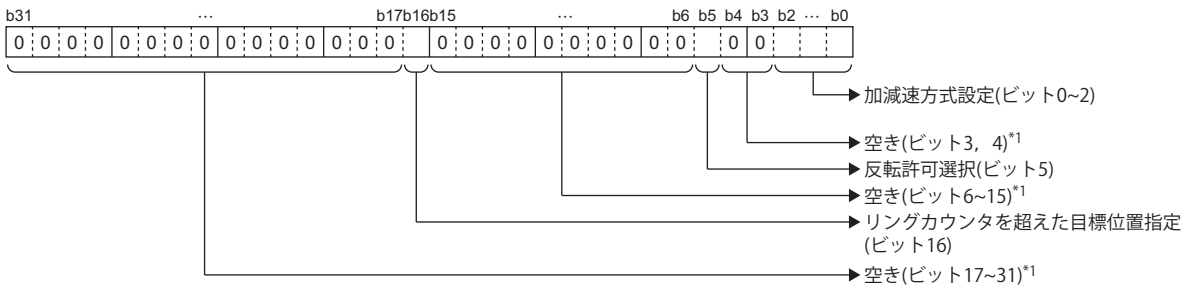
Point

多重起動(バッファモード)の詳細については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

■オプション(Options)

MC_MoveAbsolute(絶対値位置決め)で使用する機能オプションをビット指定で設定します。
ビット指定で設定する内容を下記に示します。



*1 空きには「0」を設定してください。「0」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

ビット	名称	内容
0~2	加減速方式設定	制御を行うための加減速方式を設定します。 • 0: 加減速度指定方式(mcAccDec) • 1: 加減速時間一定方式(mcFixedTime)
5	反転許可選択	実行中のFBとバッファリングするFBで動作方向が異なる場合に反転を許可するか、許可しないかを設定します。 • 0: 許可する • 1: 許可しない *: 軸動作中に後続のバッファリングFBに切り換わるような場合、または目標位置変更時に有効になります。
16	リングカウンタを超えた目標位置指定	ソフトウェアストロークリミット無効時に、リングカウンタ上限値、リングカウンタ下限値を超えた目標位置を許可するか、許可しないかを設定します。 • 0: 許可しない • 1: 許可する

• 加減速方式設定(ビット0~2)

設定値	内容
0: 加減速度指定方式(mcAccDec)	MC_MoveAbsolute(絶対値位置決め)で設定した加速度(Acceleration)、減速度(Deceleration)、ジャーク(Jerk)を用いて、加速/減速する方式です。
1: 加減速時間一定方式(mcFixedTime)	速度に関係なく、MC_MoveAbsolute(絶対値位置決め)で設定した加減速時間を用いて、加速/減速する方式です。加減速時間は、加速度(Acceleration)に設定し、減速度(Deceleration)、ジャーク(Jerk)は使用しません。

• 反転許可選択(ビット5)

設定値	内容
0: 許可する	反転を許可します。 一度減速停止を行い、減速停止完了後、変更した方向へ動作を開始します。
1: 許可しない	反転を許可しません。 切換え時にオーバランエラー (エラーコード: 1AE3H)となり減速停止します。

■必須オブジェクトデータ

MC_MoveAbsolute(絶対値位置決め)を使用する場合、下記のオブジェクトデータを軸に設定してください。

- Target position(607AH)

オブジェクトデータを設定していない場合、必須オブジェクトデータ未設定(エラーコード: 1AF7H)となり始動しません。
オブジェクトデータ設定についての詳細は、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

プログラム例

単軸絶対値位置決め始動(bMoveAbsolute)をTRUEにし、下記の設定により軸1(Axis0001)の絶対値位置決め制御を行うプログラム例を下記に示します。

・動作



・設定

項目	設定値
目標位置	80000.0
速度	50000.0
加速度	100000.0
減速度	100000.0
ジャーク	0.0

■軸

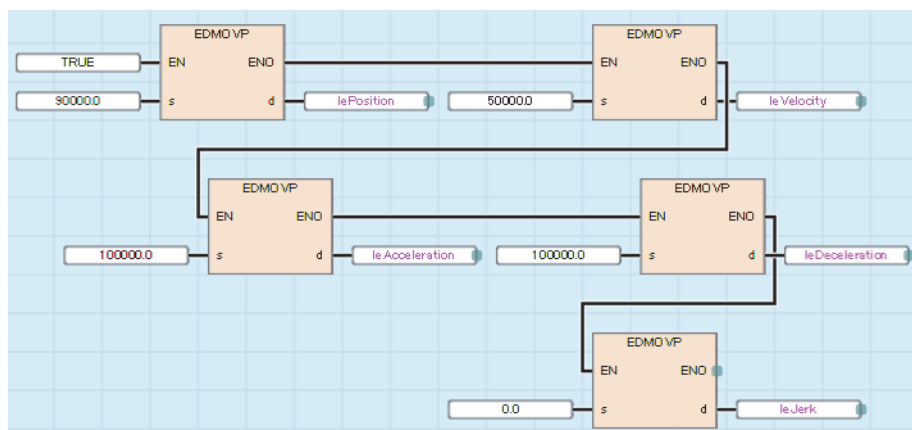
軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■使用するラベル

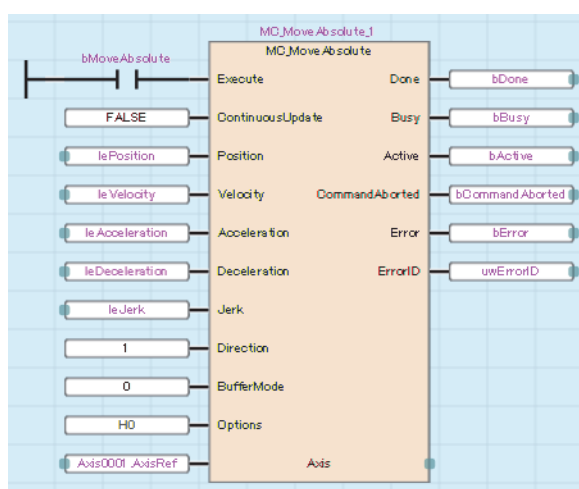
ラベル名	データ型	コメント
MC_MoveAbsolute_1	MC_MoveAbsolute	絶対値位置決めFB
bMoveAbsolute	ビット	単軸絶対値位置決め始動
lePosition	倍精度実数	目標位置
leVelocity	倍精度実数	速度
leAcceleration	倍精度実数	加速度
leDeceleration	倍精度実数	減速度
leJerk	倍精度実数	ジャーク
bDone	ビット	実行完了
bBusy	ビット	実行中
bActive	ビット	制御中
bCommandAborted	ビット	実行中断
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

- 単軸位置決め制御用データ設定



- 絶対値位置決め



■STプログラム

//-----単軸位置決め制御用データ設定-----

```
lePosition:= 80000.0;  
leVelocity:= 50000.0;  
leAcceleration:= 100000.0;  
leDeceleration:= 100000.0;  
leJerk:= 0.0;
```

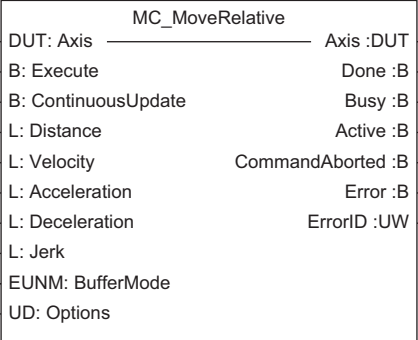
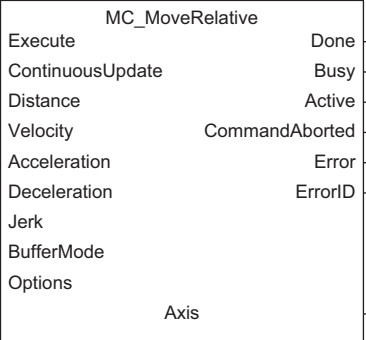
//-----絶対値位置決め-----

```
MC_MoveAbsolute_1(  
  Axis:= Axis0001.AxisRef ,  
  Execute:= bMoveAbsolute ,  
  ContinuousUpdate:= FALSE ,  
  Position:= lePosition ,  
  Velocity:= leVelocity ,  
  Acceleration:= leAcceleration ,  
  Deceleration:= leDeceleration ,  
  Jerk:= leJerk ,  
  Direction:= MC_DIRECTION__mcPositiveDirection ,  
  BufferMode:= MC_BUFFER_MODE__mcAborting ,  
  Options:= H00000000 ,  
  Done=> bDone ,  
  Busy=> bBusy ,  
  Active=> bActive ,  
  CommandAborted=> bCommandAborted ,  
  Error=> bError ,  
  ErrorID=> uwErrorID  
);
```

42.5 相対値位置決め

MC_MoveRelative

相対位置の移動量を設定し、位置決めを実行します。

ラダー	FBD/LD	ST
		<pre>MC_MoveRelative(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, ContinuousUpdate:= ?BOOL?, Distance:= ?LREAL?, Velocity:= ?LREAL?, Acceleration:= ?LREAL?, Deceleration:= ?LREAL?, Jerk:= ?LREAL?, BufferMode:= ?INT?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📖 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_MoveRelative(相対値位置決め)を実行します。
Continuous Update	連続更新	BOOL	起動時	TRUE, FALSE	FALSE	連続して、移動量(Distance)、速度(Velocity)、加速度(Acceleration)、減速度(Deceleration)の変更を有効にするか、無効にするかを設定します。 • FALSE: 無効 • TRUE: 有効
Distance	移動量	LREAL	起動時/再起動可 /連続更新可	- 100000000000.0~100000000000.0	0.0	始動時の現在位置から終点への相対位置を設定します。 詳細は、下記を参照してください。 📖 1604ページ 移動量(Distance)
Velocity	速度	LREAL	起動時/再起動可 /連続更新可	0.0, 0.0001~25000000000.0	0.0	速度を設定します。 詳細は、下記を参照してください。 📖 1604ページ 速度(Velocity)
Acceleration	加速度	LREAL	起動時/再起動可 /連続更新可	0.0000, 0.0001~2147483647.0	0.0	加速度を設定します。 詳細は、下記を参照してください。 📖 1604ページ 加速度(Acceleration)
Deceleration	減速度	LREAL	起動時/再起動可 /連続更新可	0.0000, 0.0001~2147483647.0	0.0	減速度を設定します。 詳細は、下記を参照してください。 📖 1605ページ 減速度(Deceleration)
Jerk	ジャーク	LREAL	起動時	0.0000, 0.0001~2147483647.0	0.0	ジャークを設定します。 詳細は、下記を参照してください。 📖 1605ページ ジャーク(Jerk)

入力変数	名称	データ型	取込	有効範囲	初期値	内容
BufferMode	バッファモード	INT (MC_BUFFER_MODE)	起動時	0~5	0	バッファモードを設定します。 • 0: Aborting(mcAborting) • 1: Buffered(mcBuffered) • 2: BlendingLow(mcBlendingLow) • 3: BlendingPrevious(mcBlendingPrevious) • 4: BlendingNext(mcBlendingNext) • 5: BlendingHigh(mcBlendingHigh) 詳細は、下記を参照してください。 1605ページ バッファモード (BufferMode)
Options	オプション	DWORD(H EX)	起動時	00000000H~00000021H	00000000H	MC_MoveRelative(相対値位置決め)の機能オプションを設定します。 詳細は、下記を参照してください。 1606ページ オプション(Options)

■出力変数

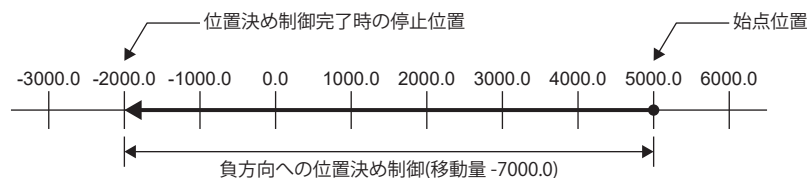
出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	相対位置へ到達したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MC_MoveRelative(相対値位置決め)を実行したときに、TRUEになります。 相対位置へ到達した後、FALSEになります。
Active	制御中	BOOL	FALSE	MC_MoveRelative(相対値位置決め)が軸を制御中のときに、TRUEになります。 相対位置へ到達した後、FALSEになります。
CommandAborted	実行中断	BOOL	FALSE	MC_MoveRelative(相対値位置決め)の実行が中断したときに、TRUEになります。 実行指令(Execute)がFALSEになると、FALSEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 使用するコントローラのユーザーズマニュアル

機能

- 移動量(Distance)、速度(Velocity)、加速度(Acceleration)、減速度(Deceleration)、ジャーク(Jerk)、バッファモード(BufferMode)を設定し、始動時の現在位置(始点位置)から移動量(Distance)に設定した移動量の位置決めを行います。運転中の方向は、移動量の符号により決まります。軸状態(AxisName.Md.AxisStatus)は、「5: 位置決め運転中(DiscreteMotion)」となります。

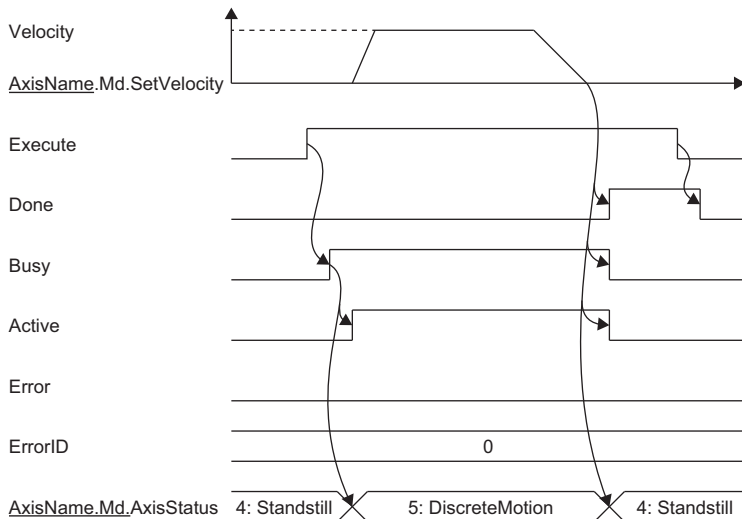
例

始点位置(現在の停止位置)が「5000.0」のとき、移動量に「-7000.0」を設定した場合
 ・「-2000.0」への位置決めを行います。



■タイミングチャート

・正常完了の場合



・異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■移動量(Distance)

始動時の現在位置から終点までの移動量を設定します。

移動方向は移動量の符号(+/-)により決まります。

移動量	内容
正(+)の場合	正方向(アドレス増加方向)に移動します。
負(-)の場合	負方向(アドレス減少方向)に移動します。

移動量が「0」の場合、軸は動作しませんが実行完了(Done)がTRUEになります。

移動量を範囲外に設定した場合、目標位置範囲外(エラーコード: 1A85H)となり始動しません。

ソフトウェアストロークリミット有効時、「現在値+移動量」がソフトウェアストロークリミット上限値、ソフトウェアストロークリミット下限値をオーバーフローする場合は、ソフトウェアストロークリミットオーバ(目標位置)(エラーコード: 1A80H)となり始動しません。

■速度(Velocity)

MC_MoveRelative(相対値位置決め)での指令速度を設定します。

設定範囲
0.0, 0.0001~2500000000.0 ^{*1} 2

- *1 浮動小数点演算を行うため、指令速度の下限値には下記の制約が生じます。
指令速度を演算周期換算した速度が「0.00001」未満となる場合、演算周期換算速度範囲外(エラーコード: 1B16H)(速度変更時は演算周期換算速度範囲外警告(イベントコード: 00D2FH))となります。浮動小数点演算の精度を向上させるためには位置指令単位(AxisName.Pr.Unit_Position)、または速度指令単位(AxisName.Pr.Unit_Velocity)を変更することで演算周期換算後の速度が「0.00001」未満とならないように設定してください。
- *2 多重起動を行うモーション制御FBにて指定速度が「0.0」の場合、直前のモーション制御FBの指定速度となります。

■加速度(Acceleration)

MC_MoveRelative(相対値位置決め)での加速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の設定により設定範囲が異なります。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	0.000000 ^{*1} , 0.000001~8400.0[s]までの正数

- *1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更、減速度変更時は、変更を受け付けません。

■減速度(Deceleration)

MC_MoveRelative(相対値位置決め)での減速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合、減速度(Deceleration)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

*1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更、減速度変更時は、変更を受け付けません。

■ジャーク(Jerk)

MC_MoveRelative(相対値位置決め)でのジャークを設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合、ジャーク(Jerk)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000, 0.0001~2147483647.0[U/s ³]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

MC_MoveRelative(相対値位置決め)で設定可能なバッファモードを下記に示します。

設定値	内容
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。
1: Buffered(mcBuffered)	制御中のFBに次のFBをバッファリングします。 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFB完了にて、バッファリングFBを順次実行します。
2: BlendingLow(mcBlendingLow)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBとバッファリングFBの目標速度のうち低い方の速度を切換え速度とします。
3: BlendingPrevious(mcBlendingPrevious)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBの目標速度を切換え速度とします。
4: BlendingNext(mcBlendingNext)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、バッファリングFBの目標速度を切換え速度とします。
5: BlendingHigh(mcBlendingHigh)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBとバッファリングFBの目標速度のうち高い方の速度を切換え速度とします。

*1 制御中のFBとバッファリングFB間で停止を行いません。

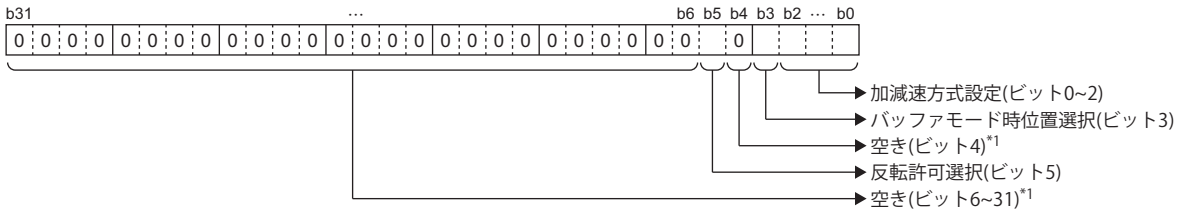
Point

多重起動(バッファモード)の詳細については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

■オプション(Options)

MC_MoveRelative(相対値位置決め)で使用する機能オプションをビット指定で設定します。
ビット指定で設定する内容を下記に示します。



*1 空きには「0」を設定してください。「0」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

ビット	名称	内容
0~2	加減速方式設定	制御を行うための加減速方式を設定します。 • 0: 加減速度指定方式(mcAccDec) • 1: 加減速時間一定方式(mcFixedTime)
3	バッファモード時位置選択	相対値位置決め制御を多重起動する場合の位置を設定します。 • 0: 指令現在位置 • 1: フィードバック位置 *: バッファモード(BufferMode)の「0: Aborting(mcAborting)」設定時に有効になります。
5	反転許可選択	実行中のFBとバッファリングするFBで動作方向が異なる場合に反転を許可するか、許可しないかを設定します。 • 0: 許可する • 1: 許可しない *: 軸動作中に後続のバッファリングFBに切り換わるような場合、または目標位置変更時に有効になります。

• 加減速方式設定(ビット0~2)

設定値	内容
0: 加減速度指定方式 (mcAccDec)	MC_MoveRelative(相対値位置決め)で設定した加速度(Acceleration), 減速度(Deceleration), ジャーク(Jerk)を用いて, 加速/減速する方式です。
1: 加減速時間一定方式 (mcFixedTime)	速度に関係なく, MC_MoveRelative(相対値位置決め)で設定した加減速時間を用いて, 加速/減速する方式です。加減速時間は, 加速度(Acceleration)に設定し, 減速度(Deceleration), ジャーク(Jerk)は使用しません。

• バッファモード時位置選択(ビット3)

設定値	内容
0: 指令現在位置	指令現在位置からの相対位置制御です。 <例> 移動量(Distance)が「5000.0」、オプション(Options)が「00000000H(ビット3が0: 指令現在位置)」の設定で多重起動した場合 指令現在位置 フィードバック位置 多重起動 位置決め完了 指令現在位置基準で 移動量(Distance)だけ移動
1: フィードバック位置	フィードバック位置からの相対位置制御です。 <例> 移動量(Distance)が「5000.0」、オプション(Options)が「00000008H(ビット3が1: フィードバック位置)」の設定で多重起動した場合 指令現在位置 フィードバック位置 多重起動 位置決め完了 フィードバック位置基準で 移動量(Distance)だけ移動

- ・反転許可選択(ビット5)

設定値	内容
0: 許可する	反転を許可します。 一度減速停止を行い、減速停止完了後、変更した方向へ動作を開始します。
1: 許可しない	反転を許可しません。 切換え時にオーバーランエラー (エラーコード: 1AE3H) となり減速停止します。

■必須オブジェクトデータ

MC_MoveRelative(相対値位置決め)を使用する場合、下記のオブジェクトデータを軸に設定してください。

- ・ Target position(607AH)

オブジェクトデータを設定していない場合、必須オブジェクトデータ未設定(エラーコード: 1AF7H)となり始動しません。

オブジェクトデータ設定に関する詳細は、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

注意事項

- ・再起動/連続更新しても、最初にMC_MoveRelative(相対値位置決め)を起動した現在位置が始動アドレスとなります。
- ・浮動小数点型で処理しているため、相対値位置決め制御を繰り返し実行すると演算誤差により設定した移動量に達しない場合があります。

プログラム例

単軸相対値位置決め始動(bMoveRelative)をTRUEにし、下記の設定により軸1(Axis0001)の相対値位置決め制御を行うプログラム例を下記に示します。

- ・動作



- ・設定

項目	設定値
目標位置	80000.0
速度	50000.0
加速度	100000.0
減速度	100000.0
ジャーク	0.0

■軸

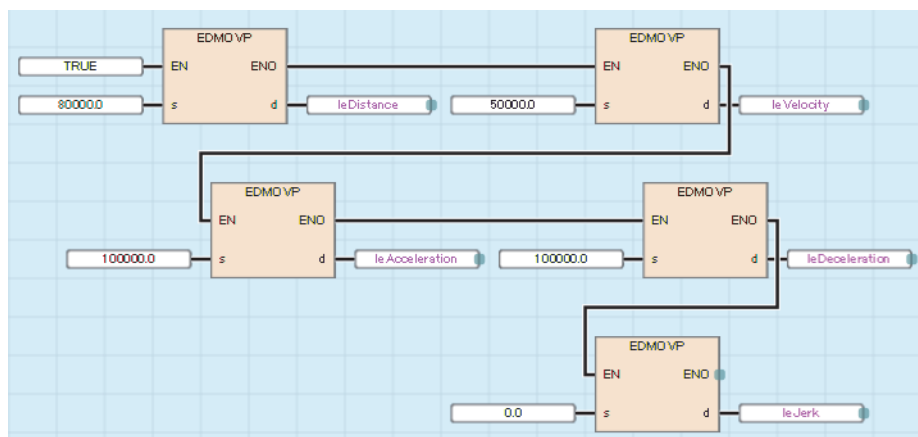
軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■使用するラベル

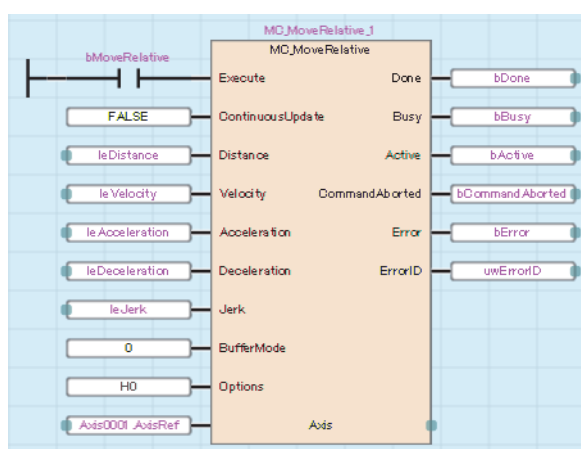
ラベル名	データ型	コメント
MC_MoveRelative_1	MC_MoveRelative	相対値位置決めFB
bMoveRelative	ビット	単軸相対値位置決め始動
leDistance	倍精度実数	移動量
leVelocity	倍精度実数	速度
leAcceleration	倍精度実数	加速度
leDeceleration	倍精度実数	減速度
leJerk	倍精度実数	ジャーク
bDone	ビット	実行完了
bBusy	ビット	実行中
bActive	ビット	制御中
bCommandAborted	ビット	実行中断
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

- 単軸位置決め制御用データ設定



- 相対値位置決め



■STプログラム

//-----単軸位置決め制御用データ設定-----

leDistance:= 80000.0;

leVelocity:= 50000.0;

leAcceleration:= 100000.0;

leDeceleration:= 100000.0;

leJerk:= 0.0;

//-----相対値位置決め-----

MC_MoveRelative_1(

Axis:= Axis0001.AxisRef ,

Execute:= bMoveRelative ,

ContinuousUpdate:= FALSE ,

Distance:= leDistance ,

Velocity:= leVelocity ,

Acceleration:= leAcceleration ,

Deceleration:= leDeceleration ,

Jerk:= leJerk ,

BufferMode:= MC_BUFFER_MODE__mcAborting ,

Options:= H00000000 ,

Done=> bDone ,

Busy=> bBusy ,

Active=> bActive ,

CommandAborted=> bCommandAborted ,

Error=> bError ,

ErrorID=> uwErrorID

);

42.6 JOG運転

MCv_Jog

指令速度に従いJOG運転を実行します。

ラダー	FBD/LD	ST
		<pre> MCv_Jog(Axis:= ?AXIS_REF?, JogForward:= ?BOOL?, JogBackward:= ?BOOL?, Velocity:= ?LREAL?, Acceleration:= ?LREAL?, Deceleration:= ?LREAL?, Jerk:= ?LREAL?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?); </pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 ☞ 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
JogForward	正転JOG指令	BOOL	常時	TRUE, FALSE	FALSE	TRUEで正方向へのMCv_Jog(JOG運転)を実行します。 詳細は、下記を参照してください。 ☞ 1612ページ 正転JOG指令(JogForward)/逆転JOG指令(JogBackward)
JogBackward	逆転JOG指令	BOOL	常時	TRUE, FALSE	FALSE	TRUEで逆方向へのMCv_Jog(JOG運転)を実行します。 詳細は、下記を参照してください。 ☞ 1612ページ 正転JOG指令(JogForward)/逆転JOG指令(JogBackward)
Velocity	速度	LREAL	起動時	0.0, 0.0001~2500000000.0	0.0	指令速度を設定します。 詳細は、下記を参照してください。 ☞ 1613ページ 速度(Velocity)
Acceleration	加速度	LREAL	起動時	0.0000, 0.0001~2147483647.0	0.0	加速度を設定します。 詳細は、下記を参照してください。 ☞ 1613ページ 加速度(Acceleration)
Deceleration	減速度	LREAL	起動時	0.0000, 0.0001~2147483647.0	0.0	減速度を設定します。 詳細は、下記を参照してください。 ☞ 1614ページ 減速度(Deceleration)
Jerk	ジャーク	LREAL	起動時	0.0000, 0.0001~2147483647.0	0.0	ジャークを設定します。 詳細は、下記を参照してください。 ☞ 1614ページ ジャーク(Jerk)
Options	オプション	DWORD(HEX)	起動時	00000000H~00000001H	00000000H	MCv_Jog(JOG運転)の機能オプションをビット指定で設定します。 詳細は、下記を参照してください。 ☞ 1614ページ オプション(Options)

■出力変数

出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	JOG指令のOFFにより、減速停止が完了したときに、1スキャンだけTRUEになります。
Busy	実行中	BOOL	FALSE	MCv_Jog(JOG運転)を実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	MCv_Jog(JOG運転)が軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	MCv_Jog(JOG運転)の実行が中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。エラーコードの詳細は、下記を参照してください。  使用するコントローラのユーザーズマニュアル

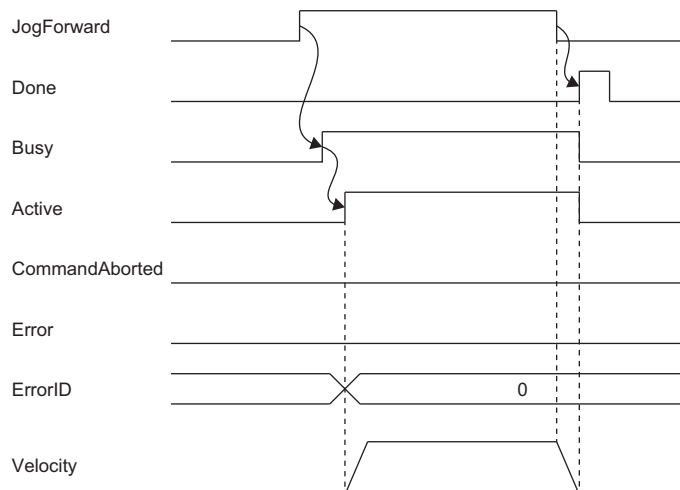
機能

- ・正転JOG指令(JogForward), または逆転JOG指令(JogBackward)をTRUEにすると、指定の方向へ対象軸が移動します。
- ・JOG運転中の軸状態(AxisName.Md.AxisStatus)は、「6: 連続動作運転中(ContinuousMotion)」になります。
- ・正転JOG指令(JogForward), または逆転JOG指令(JogBackward)をFALSEにすると、減速停止を行います。
- ・減速停止完了にて、軸状態(AxisName.Md.AxisStatus)が「4: 待機中(Standstill)」になります。
- ・正転JOG指令(JogForward), または逆転JOG指令(JogBackward)のFALSEによる減速中に、エラー (Error)がTRUEとなった場合、正転JOG指令(JogForward), または逆転JOG指令(JogBackward)をTRUEにするまで、エラー (Error)はTRUEとなります。
- ・JOG運転動作中に他の動作FBを起動した場合、起動した動作FBのバッファモード(BufferMode)の設定に従って動作を行います。
- ・他の動作FB中にJOG運転を始動した場合、始動要求を無視し、運転中始動警告(イベントコード: 00D01H)となります。軸状態(AxisName.Md.AxisStatus)が「4: 待機中(Standstill)」のときに始動してください。
- ・JOG運転中に速度を変更する場合は、オーバーライド機能による速度変更を使用してください。オーバーライド機能の詳細は、下記を参照してください。

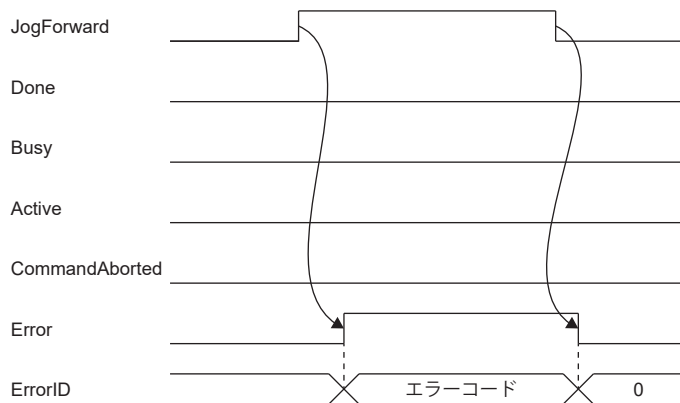
使用するコントローラのユーザーズマニュアル

■タイミングチャート

- ・正常完了の場合

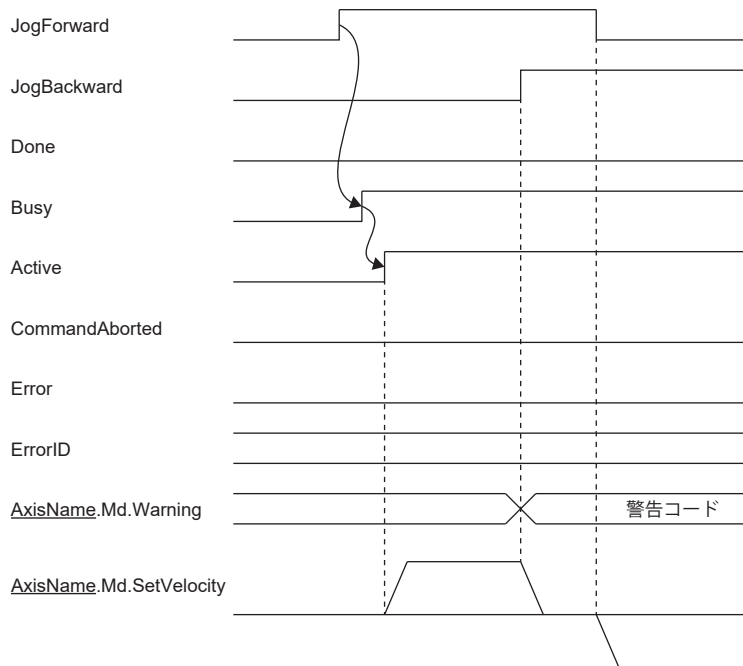


・異常完了の場合

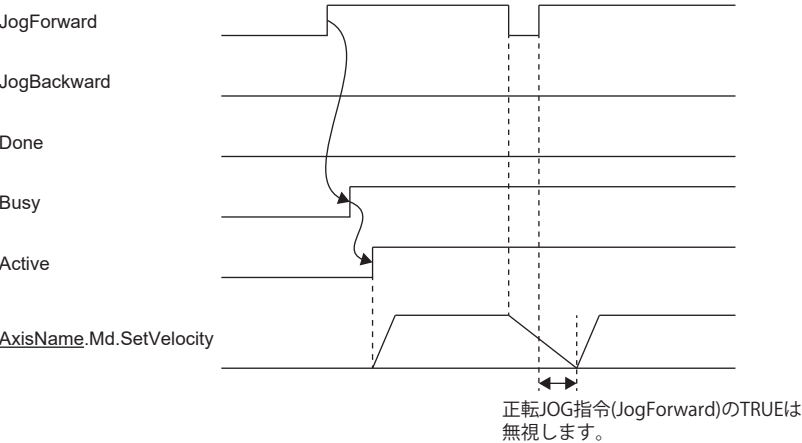


■正転JOG指令(JogForward)/逆転JOG指令(JogBackward)

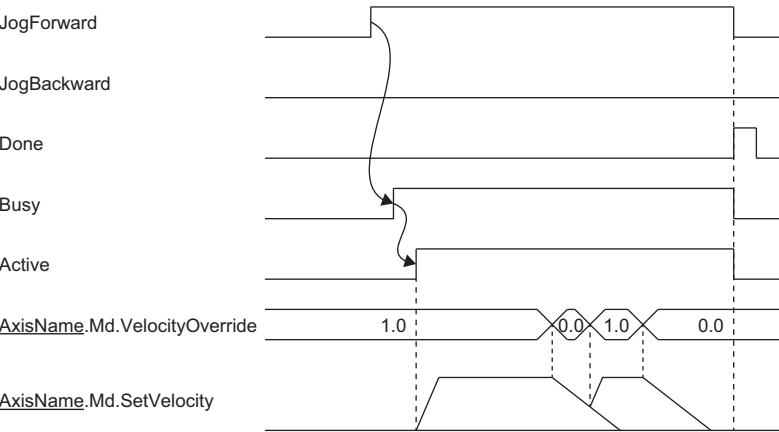
- 正転JOG指令(JogForward)をTRUEにすると正転JOG運転を、逆転JOG指令(JogBackward)をTRUEにすると逆転JOG運転を始動し、指定の方向へ対象軸が移動します。
- 正転JOG指令(JogForward)と逆転JOG指令(JogBackward)を同時にTRUEにした場合、両方向JOG指令入力警告(イベントコード: 00D02H)となり、実行中断(CommandAborted)がTRUEになり、JOG運転を行いません。その後、両方向のJOG指令をFALSEにし、実行中断(CommandAborted)をFALSEの状態となった後に、正転JOG指令(JogForward)、または逆転JOG指令(JogBackward)のどちらか一方をTRUEにすると、JOG運転を行います。
- JOG運転中に、MCv_Jog(JOG運転)のもう一方(正転JOG指令(JogForward)、または逆転JOG指令(JogBackward))のJOG指令がTRUEになった場合、両方向JOG指令入力警告(イベントコード: 00D02H)となり、TRUEを検出した時点から減速停止を行います。減速停止完了後、どちらかのJOG指令(正転JOG指令(JogForward)、または逆転JOG指令(JogBackward))のみをTRUEにすると、移動を再開します。JOG運転を行う場合は、正転JOG指令(JogForward)、または逆転JOG指令(JogBackward)のどちらか一方のみTRUEにしてください。



- 正転JOG指令(JogForward), または逆転JOG指令(JogBackward)のFALSEによる減速停止中に, JOG指令をTRUEにしても再加速しません。減速停止完了後に, 加速を行います。



- JOG運転による移動中に, 減速停止と加速を繰り返す動作を行う場合は, オーバライド機能による速度変更を使用してください。



■速度(Velocity)

MCv_Jog(JOG運転)での指令速度を設定します。
正転JOG指令(JogForward), または逆転JOG指令(JogBackward)のTRUEによる始動時に取込みを行います。JOG運転中の変更は受け付けません。

設定範囲
0.0, 0.0001~2500000000.0 ^{*1*2}

- *1 浮動小数点演算を行うため, 指令速度の下限值には下記の制約が生じます。
指令速度を演算周期換算した速度が「0.00001」未満となる場合, 演算周期換算速度範囲外(エラーコード: 1B16H)(速度変更時は演算周期換算速度範囲外警告(イベントコード: 00D2FH))となります。浮動小数点演算の精度を向上させるためには位置指令単位(AxisName.Pr.Unit_Position), または速度指令単位(AxisName.Pr.Unit_Velocity)を変更することで演算周期換算後の速度が「0.00001」未満とならないように設定してください。
- *2 多重起動を行うモーション制御FBにて指定速度が「0.0」の場合, 直前のモーション制御FBの指定速度となります。

■加速度(Acceleration)

MCv_Jog(JOG運転)での加速度を設定します。
正転JOG指令(JogForward), または逆転JOG指令(JogBackward)のTRUEによる始動時に取込みを行います。JOG運転中の変更は受け付けません。
加減速方式設定(オプション(Options): ビット0~2)の設定により設定範囲が異なります。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	0.000000 ^{*1} , 0.000001~8400.0[s]までの正数

- *1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって, 動作が変わります。加速度変更, 減速度変更時は, 変更を受け付けません。

■減速度(Deceleration)

MCv_Jog(JOG運転)での減速度を設定します。

正転JOG指令(JogForward), または逆転JOG指令(JogBackward)のTRUEによる始動時に取込みを行います。JOG運転中の変更は受け付けません。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合, 設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合, 減速度(Deceleration)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

*1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって, 動作が変わります。加速度変更, 減速度変更時は, 変更を受け付けません。

■ジャーク(Jerk)

MCv_Jog(JOG運転)でのジャークを設定します。

正転JOG指令(JogForward), または逆転JOG指令(JogBackward)のTRUEによる始動時に取込みを行います。JOG運転中の変更は受け付けません。

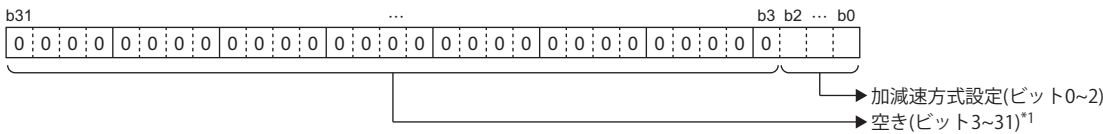
加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合, 設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合, ジャーク(Jerk)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000, 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

■オプション(Options)

MCv_Jog(JOG運転)で使用する機能オプションをビット指定で設定します。

ビット指定で設定する内容を下記に示します。



*1 空きには「0」を設定してください。「0」以外を設定した場合, Options範囲外(エラーコード: 1ABBH)となります。

ビット	名称	内容
0~2	加減速方式設定	制御を行うための加減速方式を設定します。 0: 加減速度指定方式(mcAccDec) 1: 加減速時間一定方式(mcFixedTime)

・加減速方式設定(ビット0~2)

設定値	内容
0: 加減速度指定方式(mcAccDec)	MCv_Jog(JOG運転)で設定した加速度(Acceleration), 減速度(Deceleration), ジャーク(Jerk)を用いて, 加速/減速する方式です。
1: 加減速時間一定方式(mcFixedTime)	速度に関係なく, MCv_Jog(JOG運転)で設定した加減速時間を用いて, 加速/減速する方式です。 加減速時間は, 加速度(Acceleration)に設定し, 減速度(Deceleration), ジャーク(Jerk)は使用しません。

■必須オブジェクトデータ

MCv_Jog(JOG運転)を使用する場合, 下記のオブジェクトデータを軸に設定してください。

・Target position(607AH)

オブジェクトデータを設定していない場合, 必須オブジェクトデータ未設定(エラーコード: 1AF7H)となり始動しません。

オブジェクトデータ設定に関する詳細は, 下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

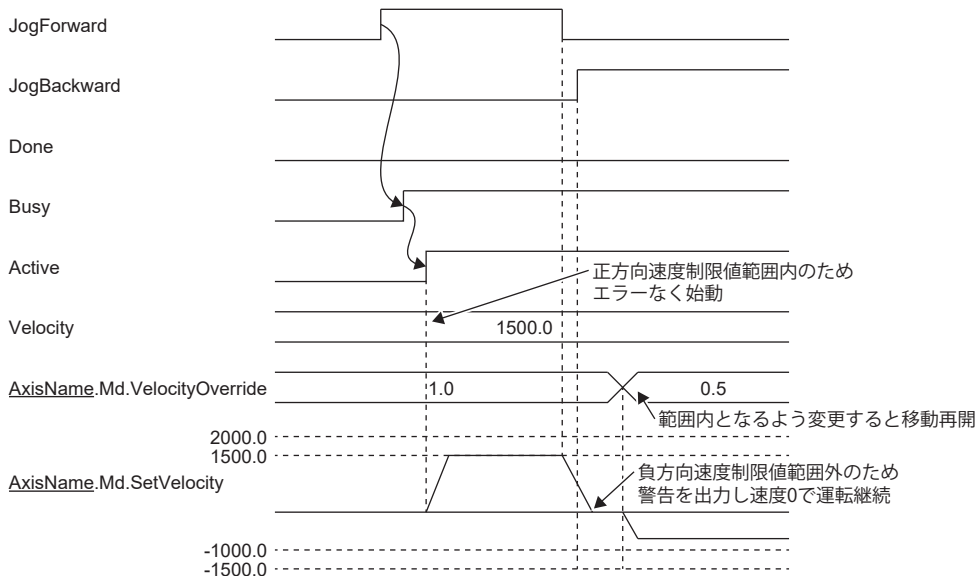
注意事項

- 安全のため、速度(Velocity)に設定するの値は、最初に小さめの値を設定して動きを確かめてから、徐々に値を大きくしてください。
- 上限リミット、下限リミットの近くでJOG運転を行う場合は、ハードウェアストロークリミット機能を使用してください。ハードウェアストロークリミット機能を使用しないと、ワークが移動範囲を超え、事故の原因となることがあります。
- ソフトウェアストロークリミット機能が有効で、ソフトウェアストロークリミットオーバーライド(AxisName.Cd.SwStrokeLimit_Override)がDISABLE(チェック無効)、およびONLY_INSIDE(範囲内方向へのみチェック無効)以外のとき、MCv_Jog(JOG運転)実行時にソフトウェアストロークリミットオーバーライド(AxisName.Cd.SwStrokeLimit_Override)をONLY_INSIDE(範囲内方向へのみチェック無効)に書き換え、JOG運転完了時に空白へ書き換えます。
- ハードウェアストロークリミット機能が有効で、ハードウェアストロークリミットオーバーライド(AxisName.Cd.HwStrokeLimit_Override)がDISABLE(チェック無効)、およびONLY_INSIDE(範囲内方向へのみチェック無効)以外のとき、MCv_Jog(JOG運転)実行時にハードウェアストロークリミットオーバーライド(AxisName.Cd.HwStrokeLimit_Override)をONLY_INSIDE(範囲内方向へのみチェック無効)に書き換え、JOG運転完了時に空白へ書き換えます。
- MCv_Jog(JOG運転)実行中は、ソフトウェアストロークリミットオーバーライド(AxisName.Cd.SwStrokeLimit_Override)、およびハードウェアストロークリミットオーバーライド(AxisName.Cd.HwStrokeLimit_Override)を変更しないでください。
- JOG運転中に多重起動した場合、運転中に次のFBの解析を行うため、多重起動時のソフトウェアストロークリミットオーバーライド(AxisName.Cd.SwStrokeLimit_Override)、およびハードウェアストロークリミットオーバーライド(AxisName.Cd.HwStrokeLimit_Override)の値が次のFBに引き継がれますのでご注意ください。
- JOG運転中に減速停止して始動時と逆方向に移動をするときに、指令速度が速度制限値を超える場合、方向反転時の速度制限値オーバー警告(イベントコード: 00D20H)となります。また、加速時間が8400秒を超える場合は、方向反転時の加速時間オーバー警告(イベントコード: 00D32H)となります。(速度0で運転を継続します。)要因を取り除くように制御変更を行うことで、移動を開始します。

例

下記の値を設定した場合

- 正方向速度制限値(AxisName.Pr.VelocityLimit_Positive): 2000.0
- 負方向速度制限値(AxisName.Pr.VelocityLimit_Negative): 1000.0



プログラム例

正転JOG運転を行う場合はJOG正転指令(bJogForward), 逆転JOG運転を行う場合はJOG逆転指令(bJogBackward)をTRUEにし, 下記の設定により軸1(Axis0001)のJOG運転を行うプログラム例を下記に示します。

・設定

項目	設定値
速度	1500.0
加速度	2000.0
減速度	2000.0
ジャーク	0.0

■軸

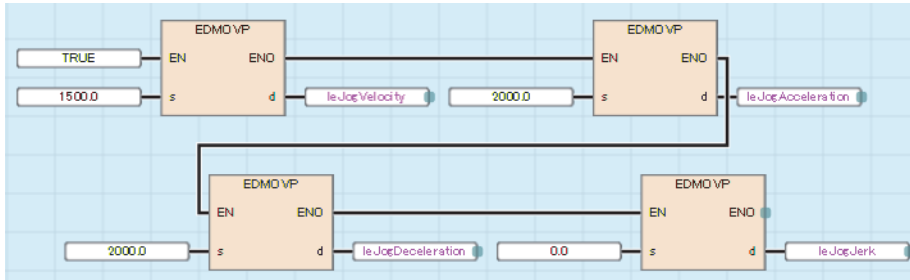
軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■使用するラベル

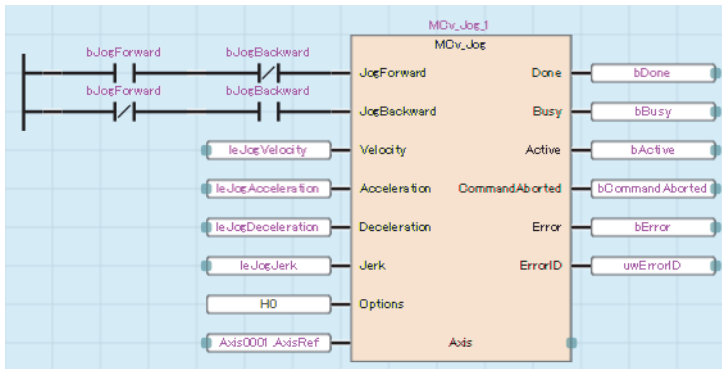
ラベル名	データ型	コメント
MCv_Jog_1	MCv_Jog	JOG運転FB
bJogForward	ビット	JOG正転指令
bJogBackward	ビット	JOG逆転指令
leJogVelocity	倍精度実数	JOG速度
leJogAcceleration	倍精度実数	JOG加速度
leJogDeceleration	倍精度実数	JOG減速度
leJogJerk	倍精度実数	JOGジャーク
bJogF	ビット	JOG正転
bJogB	ビット	JOG逆転
bDone	ビット	実行完了
bBusy	ビット	実行中
bActive	ビット	制御中
bCommandAborted	ビット	実行中断
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

• JOG運転用データ設定



• Axis0001 JOG運転



■STプログラム

//-----JOG運転用データ設定-----

leJogVelocity:= 1500.0;

leJogAcceleration:= 2000.0;

leJogDeceleration:= 2000.0;

leJogJerk:= 0.0;

//-----Axis0001 JOG運転-----

bJogForward:= (bJogForward=TRUE) & (bJogBackward=FALSE);

bJogBackward:= (bJogForward=FALSE) & (bJogBackward=TRUE);

MCv_Jog_1(

Axis:= Axis0001.AxisRef ,

JogForward:= bJogForward ,

JogBackward:= bJogBackward ,

Velocity:= leJogVelocity ,

Acceleration:= leJogAcceleration ,

Deceleration:= leJogDeceleration ,

Jerk:= leJogJerk ,

Options:= H00000000 ,

Done=> bDone ,

Busy=> bBusy ,

Active=> bActive ,

CommandAborted=> bCommandAborted ,

Error=> bError ,

ErrorID=> uwErrorID

);

42.7 速度制御

MC_MoveVelocity

ドライバをcsvに切り換え，指定した速度に従い速度制御を行います。

ラダー	FBD/LD	ST
MC_MoveVelocity DUT: Axis Axis :DUT B: Execute InVelocity :B B: ContinuousUpdate Busy :B L: Velocity Active :B L: Acceleration CommandAborted :B L: Deceleration Error :B L: Jerk ErrorID :UW ENUM: Direction ENUM: BufferMode UD: Options	MC_MoveVelocity Execute InVelocity ContinuousUpdate Busy Velocity Active Acceleration CommandAborted Deceleration Error Jerk ErrorID Direction BufferMode Options Axis	MC_MoveVelocity(Axis:= ?AXIS_REF? , Execute:= ?BOOL? , ContinuousUpdate:= ?BOOL? , Velocity:= ?LREAL? , Acceleration:= ?LREAL? , Deceleration:= ?LREAL? , Jerk:= ?LREAL? , Direction:= ?INT? , BufferMode:= ?INT? , Options:= ?DWORD? , InVelocity=> ?BOOL? , Busy=> ?BOOL? , Active=> ?BOOL? , CommandAborted=> ?BOOL? , Error=> ?BOOL? , ErrorID=> ?WORD?);

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については，下記を参照してください。 📄 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_MoveVelocity(速度制御)を実行します。
Continuous Update	連続更新	BOOL	起動時	TRUE, FALSE	FALSE	連続して、速度(Velocity)、加速度(Acceleration)、減速度(Deceleration)の変更を有効にするか、無効にするかを設定します。 ・ FALSE: 無効 ・ TRUE: 有効
Velocity	速度	LREAL	起動時/再起動可/連続更新可	0.0, ±0.0001~±2500000000.0	0.0	指令速度を設定します。 速度がマイナスの場合は、負方向へ移動します。「0.0」を設定した場合、軸は動作しませんが軸状態(AxisName.Md.AxisStatus)は「6: 連続動作運転中(ContinuousMotion)」になります。 詳細は、下記を参照してください。 📖 1621ページ 速度(Velocity)
Acceleration	加速度	LREAL	起動時/再起動可/連続更新可	0.0000, 0.0001~2147483647.0	0.0	加速度を設定します。 詳細は、下記を参照してください。 📖 1621ページ 加速度(Acceleration)
Deceleration	減速度	LREAL	起動時/再起動可/連続更新可	0.0000, 0.0001~2147483647.0	0.0	減速度を設定します。 詳細は、下記を参照してください。 📖 1621ページ 減速度(Deceleration)
Jerk	ジャーク	LREAL	起動時	0.0000, 0.0001~2147483647.0	0.0	ジャークを設定します。 詳細は、下記を参照してください。 📖 1621ページ ジャーク(Jerk)
Direction	方向選択	INT (MC_DIRECTION)	起動時	1, 2	0	方向選択を設定します。 ・ 1: 正方向(mcPositiveDirection) ・ 2: 負方向(mcNegativeDirection) *: 「2: 負方向(mcNegativeDirection)」を設定し、速度(Velocity)がマイナスの場合、モータの移動方向は正方向となります。 *: 設定を省略した場合、方向選択範囲外(エラーコード: 1AA5H)となります。
BufferMode	バッファモード	INT (MC_BUFFER_MODE)	起動時	0, 1	0	バッファモードを設定します。 ・ 0: Aborting(mcAborting) ・ 1: Buffered(mcBuffered) 詳細は、下記を参照してください。 📖 1621ページ バッファモード(BufferMode)
Options	オプション	DWORD(HEX)	起動時	00000000H~0002001H	00000000H	MC_MoveVelocity(速度制御)の機能オプションをビット指定で設定します。 詳細は、下記を参照してください。 📖 1622ページ オプション(Options)

■出力変数

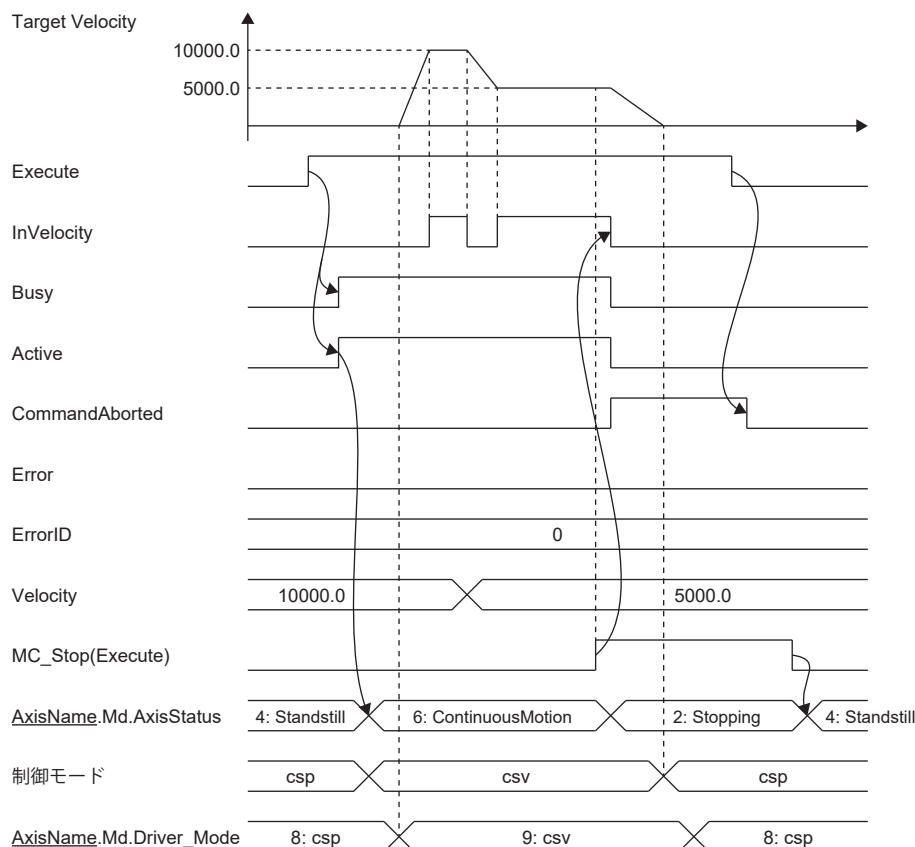
出力変数	名称	データ型	初期値	内容
InVelocity	目標速度到達	BOOL	FALSE	モーションシステムで計算した指令速度が目標速度に到達したときに、TRUEになります。 連続更新(ContinuousUpdate)が有効(TRUE)時の変更により目標速度を変更した場合、変更後の目標速度に到達するまでFALSEになります。
Busy	実行中	BOOL	FALSE	MC_MoveVelocity(速度制御)を実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	MC_MoveVelocity(速度制御)が軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	MC_MoveVelocity(速度制御)の実行が中断したときに、TRUEになります。 実行指令(Execute)がFALSEになると、FALSEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。  使用するコントローラのユーザーズマニュアル

機能

- MC_MoveVelocity(速度制御)は、ドライバの制御モードをcsvに切り換えて制御を行います。設定した加速度(Acceleration)、減速度(Deceleration)、ジャーク(Jerk)に従い指令速度を制御します。MC_MoveVelocity(速度制御)を終了するには、MC_Stop(強制停止)を起動してください。

■タイミングチャート

- 正常完了の場合



- 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■速度(Velocity)

MC_MoveVelocity(速度制御)での指令速度を設定します。

設定範囲
0.0, $\pm 0.0001 \sim \pm 2500000000.0^{*1*2}$
*1 浮動小数点演算を行うため、指令速度の下限値には下記の制約が生じます。 指令速度を演算周期換算した速度が「0.00001」未満となる場合、演算周期換算速度範囲外(エラーコード: 1B16H)(速度変更時は演算周期換算速度範囲外警告(イベントコード: 00D2FH))となります。浮動小数点演算の精度を向上させるためには位置指令単位(AxisName.Pr.Unit_Position)、または速度指令単位(AxisName.Pr.Unit_Velocity)を変更することで演算周期換算後の速度が「0.00001」未満とならないように設定してください。 *2 多重起動を行うモーション制御FBにて指定速度が「0.0」の場合、直前のモーション制御FBの指定速度となります。

■加速度(Acceleration)

MC_MoveVelocity(速度制御)での加速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の設定により設定範囲が異なります。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	0.000000 ^{*1} , 0.000001~8400.0[s]までの正数

*1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更、減速度変更時は、変更を受け付けません。

■減速度(Deceleration)

MC_MoveVelocity(速度制御)での減速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合、減速度(Deceleration)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

*1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更、減速度変更時は、変更を受け付けません。

■ジャーク(Jerk)

MC_MoveVelocity(速度制御)でのジャークを設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合、ジャーク(Jerk)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000, 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

MC_MoveVelocity(速度制御)で設定可能なバッファモードを下記に示します。

設定値	内容
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。
1: Buffered(mcBuffered)	制御中のFBに次のFBをバッファリングします。 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFB完了にて、バッファリングFBを順次実行します。

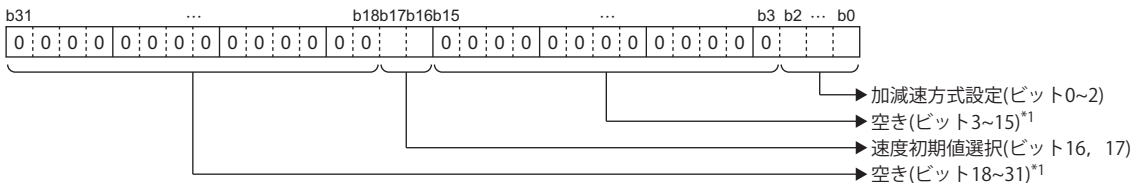
Point

多重起動(バッファモード)の詳細については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

■オプション(Options)

MC_MoveVelocity(速度制御)で使用する機能オプションをビット指定で設定します。
ビット指定で設定する内容を下記に示します。



*1 空きには「0」を設定してください。「0」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

ビット	名称	内容
0~2	加減速方式設定	制御を行うための加減速方式を設定します。 • 0: 加減速度指定方式(mcAccDec) • 1: 加減速時間一定方式(mcFixedTime)
16, 17	速度初期値選択	csp(サイクリック位置モード)からcsv(サイクリック速度モード)への制御モードを切り換えたときの速度初期値を設定します。 • 0: 指令速度 • 1: フィードバック速度 • 2: 自動選択

・加減速方式設定(ビット0~2)

設定値	内容
0: 加減速度指定方式(mcAccDec)	MC_MoveAbsolute(絶対値位置決め)で設定した加速度(Acceleration), 減速度(Deceleration), ジャーク(Jerk)を用いて, 加速/減速する方式です。
1: 加減速時間一定方式(mcFixedTime)	速度に関係なく, MC_MoveAbsolute(絶対値位置決め)で設定した加減速時間を用いて, 加速/減速する方式です。加減速時間は, 加速度(Acceleration)に設定し, 減速度(Deceleration), ジャーク(Jerk)は使用しません。

・速度初期値選択(ビット16, 17)

設定値	内容
0: 指令速度	切換え直後にドライブユニットへ指令する速度は, 指令中の速度となります。
1: フィードバック速度	切換え時にドライブユニットより受信したモータ回転数となります。 ^{*1}
2: 自動選択	切換え直後にドライブユニットへ指令する速度は, 「0: 指令速度」と「1: フィードバック速度」で低い方の速度となります。 ^{*2}

*1 オブジェクトデータの「Velocity actual value(606CH)」をマッピングしていない場合, 切換え時にドライブユニットより受信したモータ回転数となりません。(速度初期値は「0」となります。)

*2 オブジェクトデータの「Velocity actual value(606CH)」をマッピングしていない場合, 切換え直後にドライブユニットへ指令する速度は「0: 指令速度」となります。

■必須オブジェクトデータ

MC_MoveVelocity(速度制御)を使用する場合, 下記のオブジェクトデータを軸に設定してください。

- Target velocity(60FFH)

注意事項

- 速度オーバーライド係数(AxisName.Cd.VelocityOverride), 加速度オーバーライド係数(AxisName.Cd.AccelerationOverride)が有効です。
- フォローアップにより指令現在位置, 送り機械位置を更新します。
- 制御モードが切り換わるまでの時間はドライバの仕様によります。
- 制御モード切換え中に停止要因が発生した場合, 即停止します。
- 制御モード切換え中に位置決め制御用のFBを起動しないでください。ドライバ制御モード(AxisName.Md.Driver_Mode)が「9: 速度制御(csv)」に切り換わったことを確認した後, 位置決め制御用のFBを起動してください。
- MR-J5(W)-Gを使用して, モータの停止を待たずにcspからcsvに切替える場合, またはcsvからcspに切替える場合, 下記に注意してください。
 - サーボパラメータ(拡張設定)の「制御切換え時ZSP無効選択(PC76.1)」を「1: 無効(ZSP範囲に関わらず制御切換えを行います)」に設定し, 零速度状態の監視を無効に設定してください。ただし, 制御モード切換え時に振動や衝撃が生じることがあります。
 - サーボパラメータ(基本設定)の「電子ギア分子(PA06)」, 「電子ギア分母(PA07)」の設定値については, 下記を参照してください。

□使用するコントローラのユーザーズマニュアル

プログラム例

速度制御指令(bMoveVelocity)をTRUEにし、速度制御へ切り換え、下記の設定により軸1(Axis0001)の速度制御を行うプログラム例を下記に示します。

・設定

項目	設定値
速度	100000.0
加速度	50000.0
減速度	50000.0
ジャーク	0.0

■軸

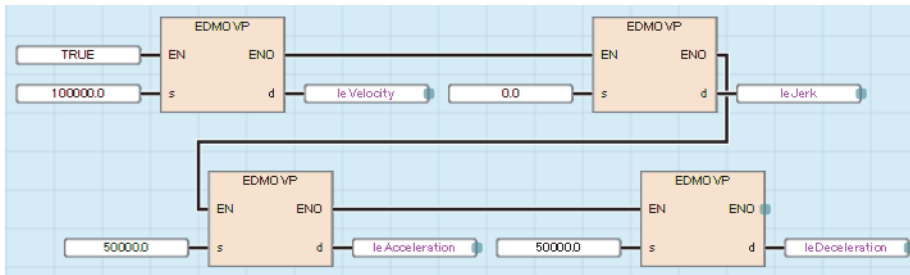
軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■使用するラベル

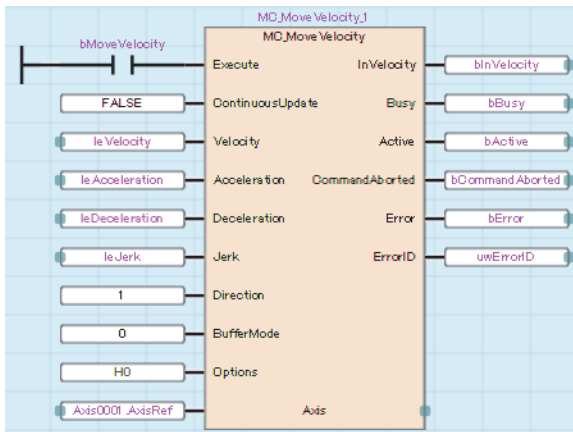
ラベル名	データ型	コメント
MC_MoveVelocity_1	MC_MoveVelocity	速度制御FB
bMoveVelocity	ビット	速度制御指令
leVelocity	倍精度実数	指令速度
leAcceleration	倍精度実数	加速度
leDeceleration	倍精度実数	減速度
leJerk	倍精度実数	ジャーク
bInVelocity	ビット	目標速度到達
bBusy	ビット	実行中
bActive	ビット	制御中
bCommandAborted	ビット	実行中断
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

• 速度制御用データ設定



• 速度制御



■STプログラム

//-----速度制御用データ設定-----

leVelocity:= 100000.0;

leAcceleration:= 50000.0;

leDeceleration:= 50000.0;

leJerk:= 0.0;

//-----速度制御-----

```
MC_MoveVelocity_1(
  Axis:= Axis0001.AxisRef ,
  Execute:= bMoveVelocity ,
  ContinuousUpdate:= FALSE ,
  Velocity:= leVelocity ,
  Acceleration:= leAcceleration ,
  Deceleration:= leDeceleration ,
  Jerk:= leJerk ,
  Direction:= MC_DIRECTION__mcPositiveDirection ,
  BufferMode:= MC_BUFFER_MODE__mcAborting ,
  Options:= H00000000 ,
  InVelocity=> bInVelocity ,
  Busy=> bBusy ,
  Active=> bActive ,
  CommandAborted=> bCommandAborted ,
  Error=> bError ,
  ErrorID=> uwErrorID
);
```

42.8 トルク制御

42

MC_TorqueControl

ドライバをcstに切り換え、指定した目標トルクに従いトルク制御を行います。

ラダー	FBD/LD	ST
		<pre> MC_TorqueControl(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, ContinuousUpdate:= ?BOOL?, Torque:= ?LREAL?, TorquePositiveRamp:= ?LREAL?, TorqueNegativeRamp:= ?LREAL?, LimitVelocity:= ?LREAL?, Acceleration:= ?LREAL?, Deceleration:= ?LREAL?, Jerk:= ?LREAL?, Direction:= ?INT?, BufferMode:= ?INT?, Options:= ?DWORD?, InTorque=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?); </pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、 下記を参照してください。 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_TorqueControl(トルク制御)を実行します。
ContinuousUpdate	連続更新	BOOL	起動時	TRUE, FALSE	FALSE	連続して、目標トルク(Torque)、トルク正方向ランプ(TorquePositiveRamp)、トルク負方向ランプ(TorqueNegativeRamp)、制限速度(LimitVelocity)、加速度(Acceleration)、減速度(Deceleration)の変更を有効にするか、無効にするかを設定します。 ・FALSE: 無効 ・TRUE: 有効
Torque	目標トルク	LREAL	起動時/再起動可/連続更新可	-1000.0~1000.0[%]	0.0	指令トルクを設定します。 使用するサーボモータの定格トルクに対する比率を%単位で設定します。 小数点以下の有効桁数は、オブジェクトデータマップに割り当てたIDにより異なります。 有効桁数以下の値は切り捨てます。 指令トルクを正方向(アドレス増加方向)に出力する場合は正の値、負方向(アドレス減少方向)に出力する場合は負の値を設定してください。

入力変数	名称	データ型	取込	有効範囲	初期値	内容
TorquePositiveRamp	トルク正方向ランプ	LREAL	起動時/再起動可 /連続更新可	■「0: ランプ方式」の場合 0.0~1000.0[%/s] ■「1: 時定数方式」, 「2: 時間一定方式」の場合 0.0~8400.0[s]	0.0	正方向の指令トルクを設定します。 詳細は下記を参照してください。 ☞ 1628ページ トルク正方向ランプ (TorquePositiveRamp)/トルク負方向ランプ (TorqueNegativeRamp)
TorqueNegativeRamp	トルク負方向ランプ	LREAL	起動時/再起動可 /連続更新可		0.0	負方向の指令トルクを設定します。 ☞ 1628ページ トルク正方向ランプ (TorquePositiveRamp)/トルク負方向ランプ (TorqueNegativeRamp)
LimitVelocity	制限速度	LREAL	起動時/再起動可 /連続更新可	0.0001~2500000000.0	0.0	サイクリックトルクモード時の速度制限値を設定します。 *: 速度制限値の値についてはドライバのマ ニュアルを参照してください。 ■MR-J5(W)-Gを使用する場合 ・速度制限値はオブジェクトデータの「Velocity limit value(2D20H)」を使用します。 ・オブジェクトデータの「Velocity limit value(2D20H)」をマッピングしていない場 合、コントローラは速度指令は有効になりま せん。(サーボパラメータの「速度制限 (PT67)」の設定値が有効になります。) *: オブジェクトデータの詳細は、下記を参照 してください。 ☞ 使用するコントローラのユーザーズマ ニュアル
Acceleration	加速度	LREAL	起動時/再起動可 /連続更新可	0.0000, 0.0001~2147483647.0	0.0	制限速度に達するまでの加速度/加減速時間を設 定します。 詳細は、下記を参照してください。 ☞ 1629ページ 加速度(Acceleration)
Deceleration	減速度	LREAL	起動時/再起動可 /連続更新可	0.0000, 0.0001~2147483647.0	0.0	制限速度に達するまでの減速度を設定します。 詳細は、下記を参照してください。 ☞ 1629ページ 減速度(Deceleration)
Jerk	ジャーク	LREAL	起動時	0.0000, 0.0001~2147483647.0	0.0	制限速度に達するまでのジャークを設定します。 詳細は、下記を参照してください。 ☞ 1629ページ ジャーク(Jerk) *: 「0.0」, または設定を省略した場合、ジャ ークを適用しません。
Direction	方向選択	INT (MC_DIRECTION)	起動時	0	0	「0」を設定してください。 *: 「0」以外を設定した場合、方向選択範囲外(エラーコード: 1AA5H)となります。
BufferMode	バッファ モード	INT (MC_BUFFER _MODE)	起動時	0, 1	0	バッファモードを設定します。 ・0: Aborting(mcAborting) ・1: Buffered(mcBuffered) 詳細は下記を参照してください。 ☞ 1629ページ バッファモード(BufferMode)
Options	オプション	DWORD(HEX)	起動時	00000000H~00060001H	00000000H	MC_TorqueControl(トルク制御)の機能オプシ ョンをビット指定で設定します。 詳細は下記を参照してください。 ☞ 1630ページ オプション(Options)

■出力変数

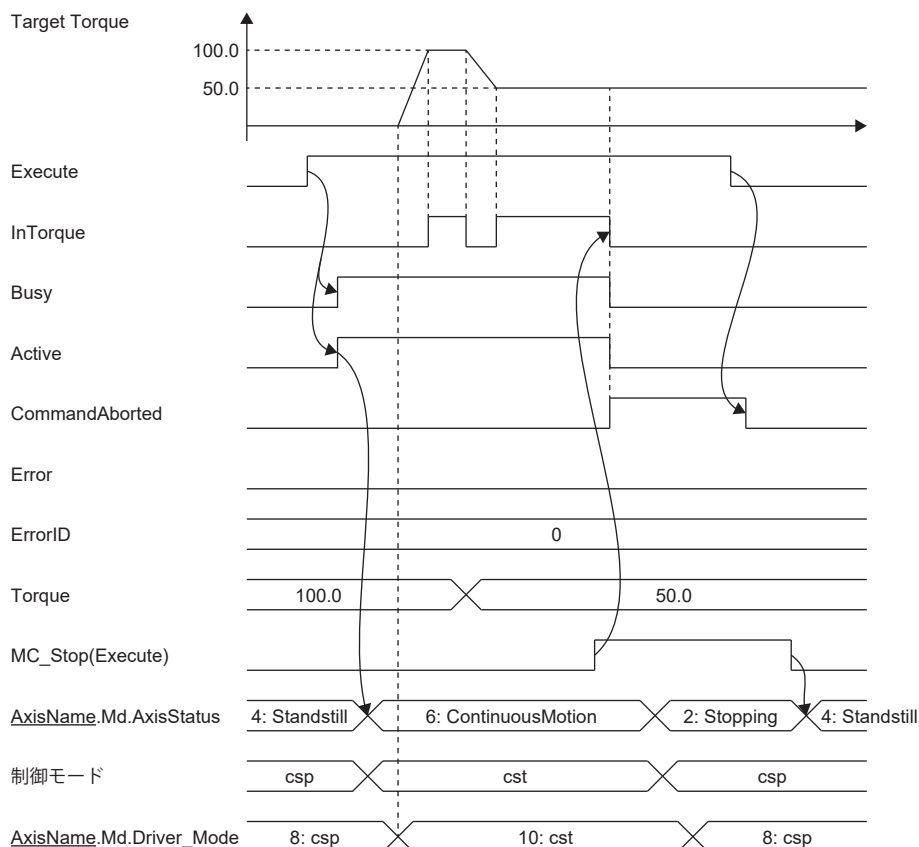
出力変数	名称	データ型	初期値	内容
InTorque	目標トルク到達	BOOL	FALSE	指令トルクが目標トルクに到達したときに、TRUEになります。 連続更新(ContinuousUpdate)がTRUE時の変更により目標トルクを変更した場合、変更後の目標トルクに到達するまでFALSEになります。
Busy	実行中	BOOL	FALSE	MC_TorqueControl(トルク制御)を実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	MC_TorqueControl(トルク制御)が軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	MC_TorqueControl(トルク制御)の実行が中断したときに、TRUEになります。 実行指令(Execute)がFALSEになると、FALSEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。  使用するコントローラのユーザーズマニュアル

機能

- MC_TorqueControl(トルク制御)は、ドライバの制御モードをcst(サイクリックトルクモード)に切り換え、制御を行います。設定したトルク正方向ランプ(TorquePositiveRamp)、トルク負方向ランプ(TorqueNegativeRamp)に従い指令トルクを制御します。MC_TorqueControl(トルク制御)を終了するには、MC_Stop(強制停止)を起動してください。

■タイミングチャート

- 正常完了の場合



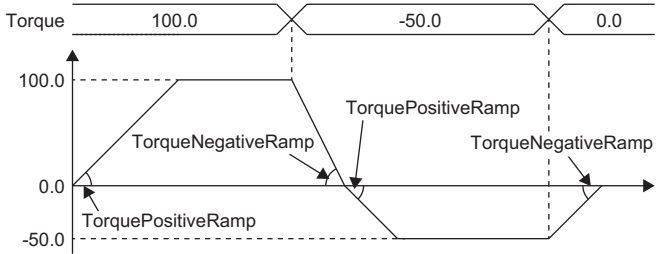
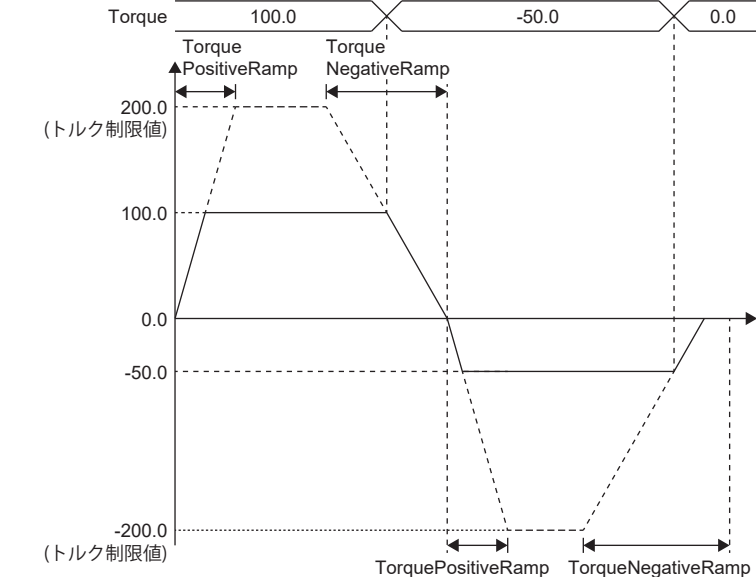
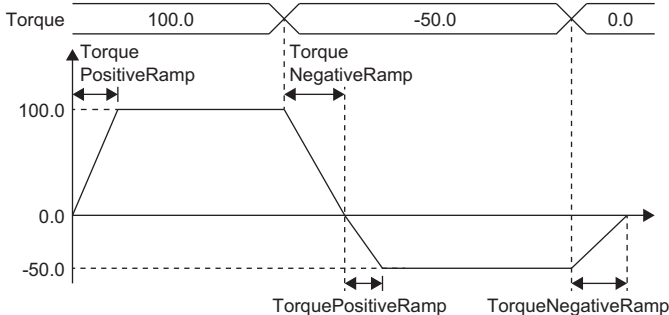
- 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■トルク正方向ランプ(TorquePositiveRamp)/トルク負方向ランプ(TorqueNegativeRamp)

指定する指令トルクを設定します。
現在の指令トルクから目標トルクに到達するまでの設定値を設定します。
目標トルクに到達するまでの設定値は、トルクランプ機能選択(オプション(Option): ビット16, 17)の設定により異なります。

トルクランプ機能選択(オプション(Option): ビット16, 17)の設定値	内容	設定範囲
0: ランプ方式	現在の指令トルクから目標トルクに到達するまでの傾斜を設定します。 	0.0~1000.0[%/s]
1: 時定数方式	指令トルクが「0」から正方向トルク制限値、負方向トルク制限値に到達するまでの時間を設定します。 	0.0~8400.0[s]
2: 時間一定方式	現在の指令トルクから目標トルクに到達するまでの時間を設定します。 	0.0~8400.0[s]

目標トルクの変更により出力トルクの変化する場合、トルク負方向ランプ(TorqueNegativeRamp)の設定値に従い指令トルクが「0.0」となり、その後、トルク正方向ランプ(TorquePositiveRamp)の設定値に従い目標トルクとなります。
トルク正方向ランプ(TorquePositiveRamp)、トルク負方向ランプ(TorqueNegativeRamp)に「0.0」を設定した場合、1演算周期で目標トルクに到達します。

■加速度(Acceleration)

MC_TorqueControl(トルク制御)での加速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の設定により設定範囲が異なります。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	0.000000 ^{*1} , 0.000001~8400.0[s]までの正数

*1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更、減速度変更時は、変更を受け付けません。

■減速度(Deceleration)

MC_TorqueControl(トルク制御)での減速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合、減速度(Deceleration)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

*1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更、減速度変更時は、変更を受け付けません。

■ジャーク(Jerk)

MC_TorqueControl(トルク制御)でのジャークを設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合、ジャーク(Jerk)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000, 0.0001~2147483647.0[U/s ³]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

MC_TorqueControl(トルク制御)で設定可能なバッファモードを下記に示します。

設定値	内容
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。 即時トルク制御に切り換わります。
1: Buffered(mcBuffered)	制御中のFBに次のFBをバッファリングします。 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFB完了にて、バッファリングFBを順次実行します。 前のFBが完了した後、トルク制御に切り換わります。 前のFBがMC_TorqueControl(トルク制御)の場合、目標トルク到達(InTorque)がTRUEのときに切り換わります。

Point

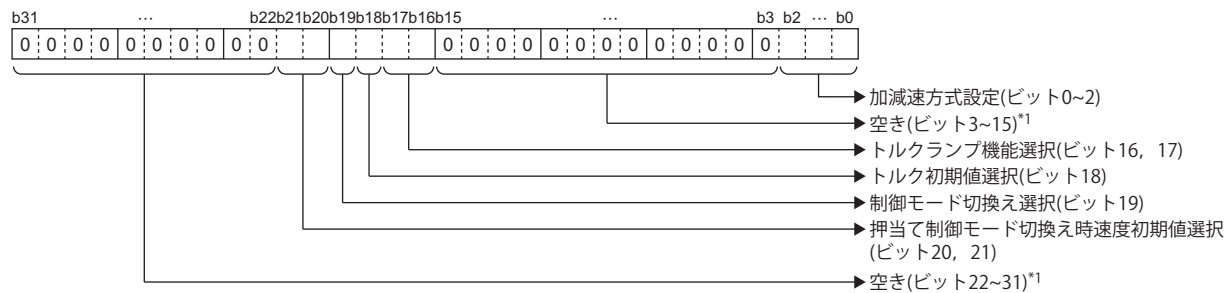
多重起動(バッファモード)の詳細については、下記を参照してください。

 使用するコントローラのユーザズマニュアル

■オプション(Options)

MC_TorqueControl(トルク制御)で使用する機能オプションをビット指定で設定します。

ビット指定で設定する内容を下記に示します。



*1 空きには「0」を設定してください。「0」以外を設定した場合、Options範囲外(エラーコード: 1ABH)となります。

ビット	名称	内容
0~2	加減速方式設定	制御を行うための加減速方式を設定します。 0: 加減速度指定方式(mcAccDec) 1: 加減速時間一定方式(mcFixedTime)
16, 17	トルクランプ機能選択	指令トルクから目標トルクに到達するまでの方式を設定します。 0: ランプ方式 1: 時定数方式 2: 時間一定方式 *: 詳細は下記を参照してください。 1628ページ トルク正方向ランプ(TorquePositiveRamp)/トルク負方向ランプ(TorqueNegativeRamp)
18	トルク初期値選択	cst(サイクリックトルクモード)へ切り換えたときのトルク初期値を設定します。 0: 目標トルク 1: フィードバックトルク
19	制御モード切換え選択*2	切換える制御モードを設定します。 0: cst(サイクリックトルクモード) 1: ct(押当て制御モード)
20, 21	押当て制御モード切換え時速度初期値選択*3	csp(サイクリック位置モード)からct(押当て制御モード)へ制御モードを切り換えたときの速度初期値を設定します。 0: 指令速度 1: フィードバック速度 2: 自動選択

*2 オブジェクトデータの「Supported drive modes(6502Hh)(ビット20)」がTRUEの場合に押当て制御へ切り換えられます。FALSEの状態
 で「1: ct(押当て制御モード)」を設定している場合、ドライバ制御モード未対応(エラーコード: 1B1BH)となります。
 押当て制御モードについての詳細は、下記を参照してください。

使用するコントローラのユーザーズマニュアル

*3 制御モード切換え選択(Options: ビット19)に「1: ct(押当て制御モード)」を設定している場合のみ有効となります。

・加減速方式設定(ビット0~2)

設定値	内容
0: 加減速度指定方式(mcAccDec)	MC_MoveAbsolute(絶対値位置決め)で設定した加速度(Acceleration), 減速度(Deceleration), ジャーク(Jerk)を用いて、加速/減速する方式です。
1: 加減速時間一定方式(mcFixedTime)	速度に関係なく、MC_MoveAbsolute(絶対値位置決め)で設定した加減速時間を用いて、加速/減速する方式です。加減速時間は、加速度(Acceleration)に設定し、減速度(Deceleration), ジャーク(Jerk)は使用しません。

・トルク初期値選択(ビット18)

設定値	内容
0: 目標トルク	制御モード切換え直後、トルク正方向ランプ(TorquePositiveRamp), トルク負方向ランプ(TorqueNegativeRamp)の値にかかわらず、始動時の目標トルク(Torque)の値がそのまま指令トルクとなります。
1: フィードバックトルク	切換え時のTorque actual valueの値が指令トルクとなります。

Point

通常は「0: 目標トルク」を設定してください。モータへの指令完了直後、サーボモータの停止を待たずに制御モードを移行する場合のみ「1: フィードバックトルク」を設定してください。

- ・ 押当て制御モード切換え時速度初期値選択(ビット20, 21)

設定値	内容
0: 指令速度	切換え直後にドライブユニットへ指令する速度は、指令中の速度となります。
1: フィードバック速度	切換え時にドライブユニットより受信したモータ回転数となります。 ^{*1}
2: 自動選択	切換え直後にドライブユニットへ指令する速度は、「0: 指令速度」と「1: フィードバック速度」で低い方の速度となります。 ^{*2}

*1 オブジェクトデータの「Velocity actual value(606CH)」をマッピングしていない場合、切換え時にドライブユニットより受信したモータ回転数となりません。(速度初期値は「0」となります。)

*2 オブジェクトデータの「Velocity actual value(606CH)」をマッピングしていない場合、切換え直後にドライブユニットへ指令する速度は、「0: 指令速度」となります。

■必須オブジェクトデータ

MC_TorqueControl(トルク制御)を使用する場合、下記のオブジェクトデータを軸に設定してください。

- ・ Target torque(6071H)

注意事項

- ・ 速度オーバーライド係数(AxisName.Cd.VelocityOverride), 加速度オーバーライド係数(AxisName.Cd.AccelerationOverride)が有効です。
- ・ フォローアップにより指令現在位置, 送り機械位置を更新します。
- ・ 制御モードが切り換わるまでの時間はドライバの仕様によります。
- ・ 制御モード切換え中に停止要因が発生した場合, 即停止します。
- ・ 制御モード切換え中に位置決め制御用のFBを起動しないでください。ドライバ制御モード(AxisName.Md.Driver_Mode)が「10: トルク制御(cst)」に切り換わったことを確認した後, 位置決め制御用のFBを起動してください。
- ・ MR-J5(W)-Gを使用して, モータの停止を待たずにcspからcstに切換える場合, またはcstからcspに切換える場合, 下記に注意してください。
 - ・ サーボパラメータ(拡張設定)の「制御切換え時ZSP無効選択(PC76.1)」を「1: 無効(ZSP範囲に関わらず制御切換えを行います)」に設定し, 零速度状態の監視を無効に設定してください。ただし, 制御モード切換え時に振動や衝撃が生じることがあります。
 - ・ サーボパラメータ(基本設定)の「電子ギア分子(PA06)」, 「電子ギア分母(PA07)」の設定値については, 下記を参照してください。
- 使用するコントローラのユーザーズマニュアル
- ・ Velocity limit valueがマッピングされていない場合, 停止要因発生時に指令トルクを0に変更します。

■cst(サイクリックトルクモード)中のトルク制限変更について

- ・ 再起動または連続更新で, 目標トルクをトルク制限値(正方向トルク制限値(AxisName.Md.TorqueLimit_Positive), 負方向トルク制限値(AxisName.Md.TorqueLimit_Negative))よりも大きな値に変更した場合, トルク制限値オーバ警告(イベントコード: 00D12H)となり, 変更前の値で動作します。
- ・ トルク制御中に, 目標トルクよりも小さい正方向トルク制限値(AxisName.Cd.TorqueLimit_Positive), 負方向トルク制限値(AxisName.Cd.TorqueLimit_Negative)に変更した場合, 目標トルクをトルク制限値に変更します。このとき, 指令トルクは1周期で変更します。

プログラム例

トルク制御指令(bTorqueControl)をTRUEにし、トルク制御へ切り換え、下記の設定により軸1(Axis0001)のトルク制御を行うプログラム例を下記に示します。

・設定

項目	設定値
速度	100000.0
加速度	50000.0
減速度	50000.0
ジャーク	0.0

■軸

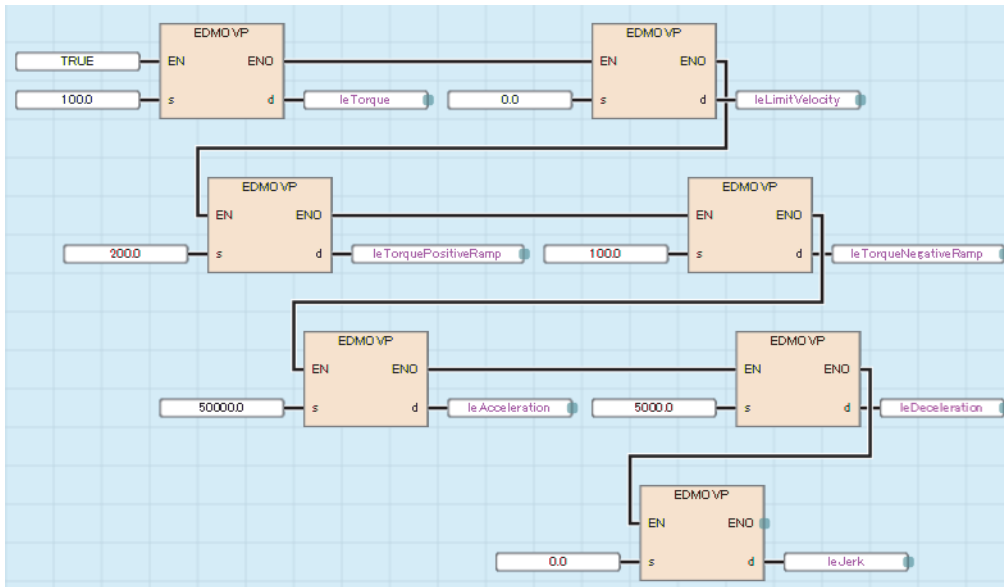
軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■使用するラベル

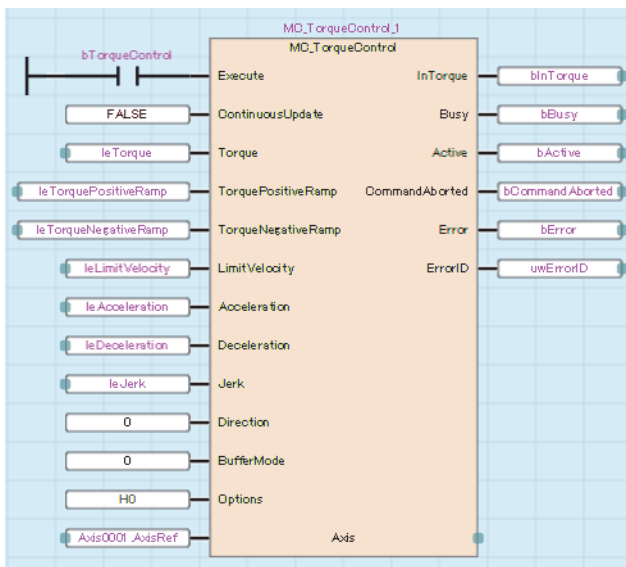
ラベル名	データ型	コメント
MC_TorqueControl_1	MC_TorqueControl	トルク制御FB
bTorqueControl	ビット	トルク制御指令
leTorque	倍精度実数	目標トルク
leTorquePositiveRamp	倍精度実数	トルク正方向ランプ
leTorqueNegativeRamp	倍精度実数	トルク負方向ランプ
leLimitVelocity	倍精度実数	制限速度
leAcceleration	倍精度実数	加速度
leDeceleration	倍精度実数	減速度
leJerk	倍精度実数	ジャーク
bInTorque	ビット	目標トルク到達
bBusy	ビット	実行中
bActive	ビット	制御中
bCommandAborted	ビット	実行中断
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

・トルク制御用データ設定



・トルク制御



■STプログラム

//-----トルク制御用データ設定-----

```
leTorque:= 100.0;  
leTorquePositiveRamp:= 200.0;  
leTorqueNegativeRamp:= 100.0;  
leLimitVelocity:= 100000.0;  
leAcceleration:= 50000.0;  
leDeceleration:= 50000.0;  
leJerk:= 0.0;
```

//-----トルク制御-----

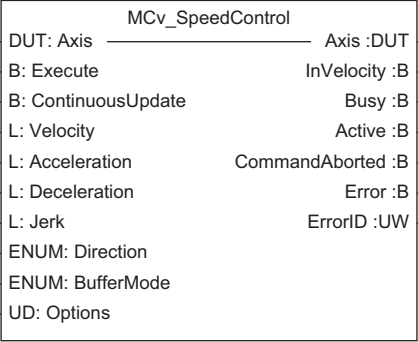
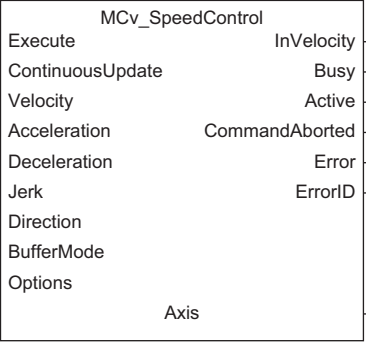
```
MC_TorqueControl_1(  
  Axis:= Axis0001.AxisRef ,  
  Execute:= bTorqueControl ,  
  ContinuousUpdate:= FALSE ,  
  Torque:= leTorque ,  
  TorquePositiveRamp:= leTorquePositiveRamp ,  
  TorqueNegativeRamp:= leTorqueNegativeRamp ,  
  LimitVelocity:= leLimitVelocity ,  
  Acceleration:= leAcceleration ,  
  Deceleration:= leDeceleration ,  
  Jerk:= leJerk ,  
  Direction:= 0 ,  
  BufferMode:= MC_BUFFER_MODE__mcAborting ,  
  Options:= H00000000 ,  
  InTorque=> bInTorque ,  
  Busy=> bBusy ,  
  Active=> bActive ,  
  CommandAborted=> bCommandAborted ,  
  Error=> bError ,  
  ErrorID=> uwErrorID  
);
```


42.9 速度制御(位置ループを含む)

42

MCv_SpeedControl

位置ループを含む速度制御を実行します。

ラダー	FBD/LD	ST
		<pre>MCv_SpeedControl(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, ContinuousUpdate:= ?BOOL?, Velocity:= ?LREAL?, Acceleration:= ?LREAL?, Deceleration:= ?LREAL?, Jerk:= ?LREAL?, Direction:= ?INT?, BufferMode:= ?INT?, Options:= ?DWORD?, InVelocity=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMCv_SpeedControl(速度制御(位置ループを含む))を実行します。
Continuous Update	連続更新	BOOL	起動時	TRUE, FALSE	FALSE	連続して、速度(Velocity)、加速度(Acceleration)、減速度(Deceleration)の変更を有効にするか、無効にするかを設定します。 • FALSE: 無効 • TRUE: 有効
Velocity	速度	LREAL	起動時/再起動可 /連続更新可	0.0, ±0.0001~±2500000000.0	0.0	指令速度を設定します。 速度がマイナスの場合は、逆転方向へ移動します。 設定を省略した場合、軸は動作しませんが軸状態(AxisName.Md.AxisStatus)は「6: 連続動作運転中(ContinuousMotion)」となります。 詳細は、下記を参照してください。 1637ページ 速度(Velocity)
Acceleration	加速度	LREAL	起動時/再起動可 /連続更新可	0.0000, 0.0001~2147483647.0	0.0	加速度を設定します。 詳細は、下記を参照してください。 1637ページ 加速度(Acceleration)
Deceleration	減速度	LREAL	起動時/再起動可 /連続更新可	0.0000, 0.0001~2147483647.0	0.0	減速度を設定します。 詳細は、下記を参照してください。 1638ページ 減速度(Deceleration)
Jerk	ジャーク	LREAL	起動時	0.0000, 0.0001~2147483647.0	0.0	ジャークを設定します。 詳細は、下記を参照してください。 1638ページ ジャーク(Jerk)

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Direction	方向選択	INT (MC_DIRECTION)	起動時	1, 2	0	方向選択を設定します。 • 1: 正方向(mcPositiveDirection) • 2: 負方向(mcNegativeDirection) *: 「2: 負方向(mcNegativeDirection)」を設定し、速度(Velocity)がマイナスの場合、モータの移動方向は正方向となります。 *: 設定を省略した場合、方向選択範囲外(エラーコード: 1AA5H)となります。
BufferMode	バッファモード	INT (MC_BUFFER_MODE)	起動時	0~5	0	バッファモードを設定します。 • 0: Aborting(mcAborting) • 1: Buffered(mcBuffered) • 2: BlendingLow(mcBlendingLow) • 3: BlendingPrevious(mcBlendingPrevious) • 4: BlendingNext(mcBlendingNext) • 5: BlendingHigh(mcBlendingHigh) 詳細は、下記を参照してください。 📖 1638ページ バッファモード(BufferMode)
Options	オプション	DWORD(HEX)	起動時	00000000H~00000001H	00000000H	MCv_SpeedControl(速度制御(位置ループを含む))の機能オプションをビット指定で設定します。 詳細は、下記を参照してください。 📖 1639ページ オプション(Options)

■出力変数

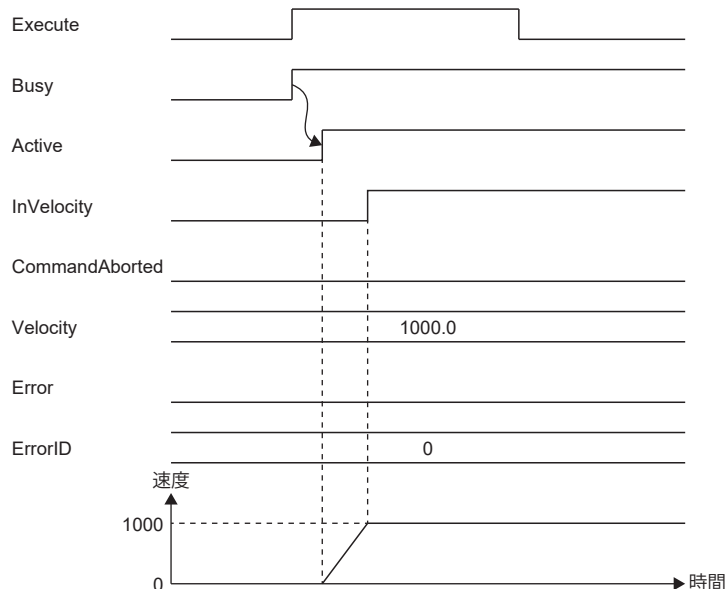
出力変数	名称	データ型	初期値	内容
InVelocity	目標速度到達	BOOL	FALSE	目標速度に指令速度が到達したときに、TRUEになります。 連続更新(ContinuousUpdate)が有効(TRUE)時の変更により目標速度を変更した場合、変更後の目標速度に到達するまでFALSEになります。
Busy	実行中	BOOL	FALSE	MCv_SpeedControl(速度制御(位置ループを含む))を実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	MCv_SpeedControl(速度制御(位置ループを含む))が軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	MCv_SpeedControl(速度制御(位置ループを含む))の実行が中断したときに、TRUEになります。 実行指令(Execute)がFALSEになると、FALSEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 📖 使用するコントローラのユーザーズマニュアル

機能

- ドライバ側の制御モードをcspとし、設定した軸を設定した速度により速度制御を実行します。軸を停止するには、MC_Stop(強制停止)を使用するか、ほかの動作系FBを起動します。ドライバ側の制御モードをcsvとして速度制御を行う場合は、MC_MoveVelocity(速度制御)を使用してください。

■タイミングチャート

- 正常完了の場合



- 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

☞ 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■速度(Velocity)

MCv_SpeedControl(速度制御(位置ループを含む))での指令速度を設定します。

設定範囲

0.0, $\pm 0.0001 \sim \pm 2500000000.0^{*1*2}$

- *1 浮動小数点演算を行うため、指令速度の下限值には下記の制約が生じます。
指令速度を演算周期換算した速度が「0.00001」未満となる場合、演算周期換算速度範囲外(エラーコード: 1B16H)(速度変更時は演算周期換算速度範囲外警告(イベントコード: 00D2FH))となります。浮動小数点演算の精度を向上させるためには位置指令単位(AxisName.Pr.Unit_Position), または速度指令単位(AxisName.Pr.Unit_Velocity)を変更することで演算周期換算後の速度が「0.00001」未満とならないように設定してください。
- *2 多重起動を行うモーション制御FBにて指定速度が「0.0」の場合、直前のモーション制御FBの指定速度となります。

■加速度(Acceleration)

MCv_SpeedControl(速度制御(位置ループを含む))での加速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の設定により設定範囲が異なります。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000^{*1} , $0.0001 \sim 2147483647.0[U/s^2]$ までの正数
1: 加減速時間一定方式(mcFixedTime)	0.000000^{*1} , $0.000001 \sim 8400.0[s]$ までの正数

- *1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更, 減速度変更時は, 変更を受け付けません。

■減速度(Deceleration)

MCv_SpeedControl(速度制御(位置ループを含む))での減速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合、減速度(Deceleration)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

*1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更、減速度変更時は、変更を受け付けません。

■ジャーク(Jerk)

MCv_SpeedControl(速度制御(位置ループを含む))でのジャークを設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合、ジャーク(Jerk)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000, 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

MCv_SpeedControl(速度制御(位置ループを含む))で設定可能なバッファモードを下記に示します。

設定値	内容
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。
1: Buffered(mcBuffered)	制御中のFBに次のFBをバッファリングします。 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFB完了にて、バッファリングFBを順次実行します。
2: BlendingLow(mcBlendingLow)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBとバッファリングFBの目標速度のうち低い方の速度を切換え速度とします。
3: BlendingPrevious(mcBlendingPrevious)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBの目標速度を切換え速度とします。
4: BlendingNext(mcBlendingNext)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、バッファリングFBの目標速度を切換え速度とします。
5: BlendingHigh(mcBlendingHigh)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBとバッファリングFBの目標速度のうち高い方の速度を切換え速度とします。

*1 制御中のFBとバッファリングFB間で停止を行いません。

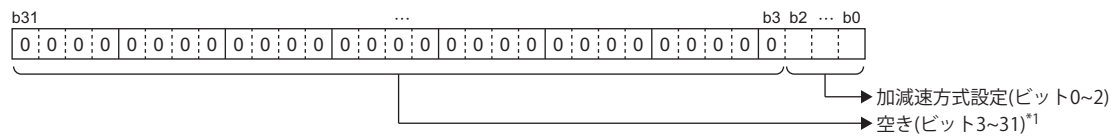
Point

多重起動(バッファモード)の詳細については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

■オプション(Options)

MCv_SpeedControl(速度制御(位置ループを含む))で使用する機能オプションをビット指定で設定します。
ビット指定で設定する内容を下記に示します。



*1 空には「0」を設定してください。「0」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

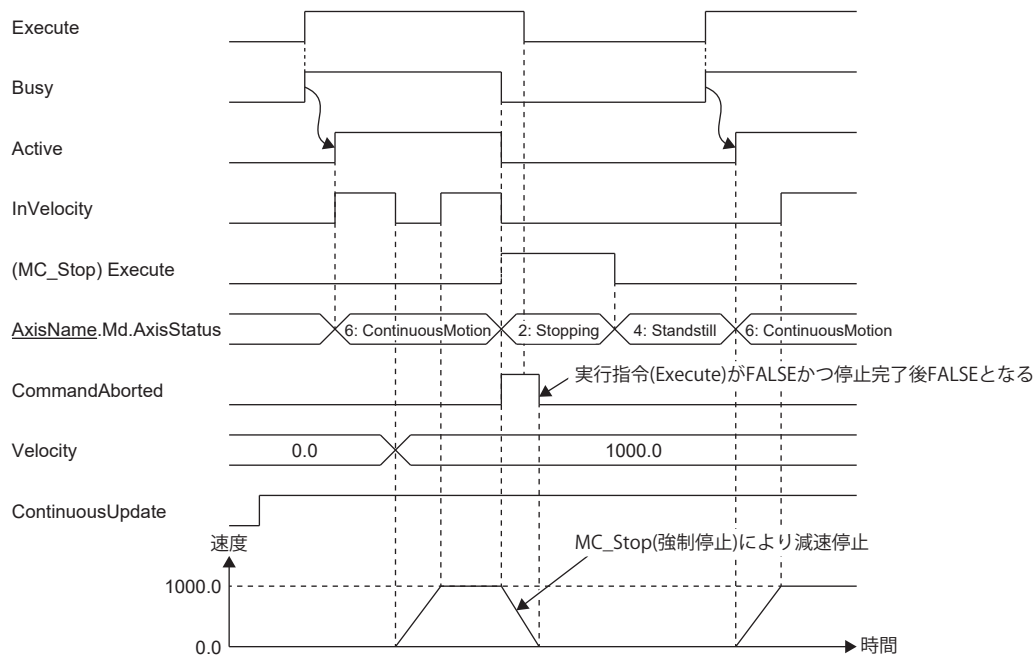
ビット	名称	内容
0~2	加減速方式設定	制御を行うための加減速方式を設定します。 0: 加減速度指定方式(mcAccDec) 1: 加減速時間一定方式(mcFixedTime)

・加減速方式設定(ビット0~2)

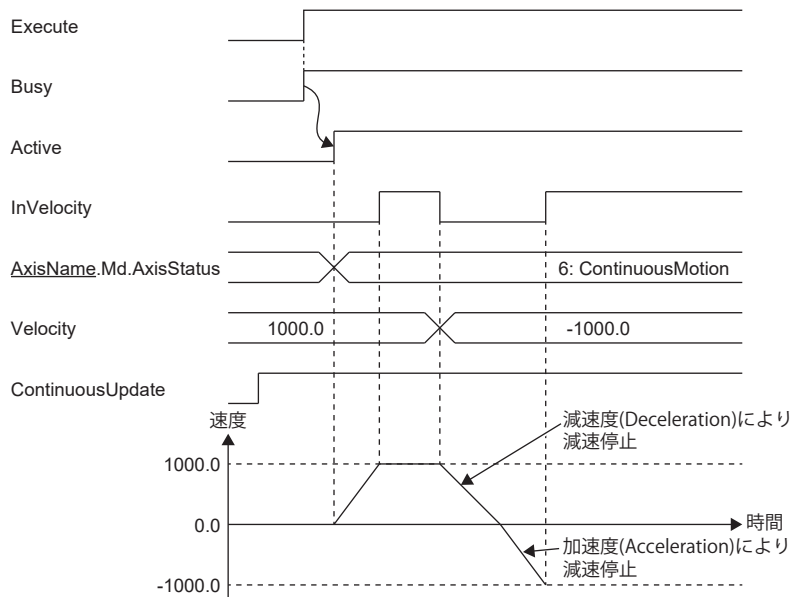
設定値	内容
0: 加減速度指定方式(mcAccDec)	MCv_SpeedControl(速度制御(位置ループを含む))で設定した加速度(Acceleration), 減速度(Deceleration), ジャーク(Jerk)を用いて, 加速/減速する方式です。
1: 加減速時間一定方式(mcFixedTime)	速度に関係なく, MCv_SpeedControl(速度制御(位置ループを含む))で設定した加減速時間を用いて, 加速/減速する方式です。 加減速時間は, 加速度(Acceleration)に設定し, 減速度(Deceleration), ジャーク(Jerk)は使用しません。

■動作概要

- ・始動時および停止要因発生時の動作は下記となります。



- 連続更新(ContinuousUpdate)がTRUEのときに、速度(Velocity)の符号が反転し、運転の向きが変化する場合は、いったん減速停止後に目標速度へ向けて加速します。



■必須オブジェクトデータ

単軸のMCv_SpeedControl(速度制御(位置ループを含む))を使用する場合、下記のオブジェクトデータを軸に設定してください。

- Target position(607AH)

オブジェクトデータを設定していない場合、必須オブジェクトデータ未設定(エラーコード: 1AF7H)となり始動しません。オブジェクトデータ設定についての詳細は、下記を参照してください。

使用するコントローラのユーザーズマニュアル

プログラム例

速度制御指令(bSpeedControl)をTRUEにし、位置ループを含む速度制御へ切り換え、下記の設定により軸1(Axis0001)の位置ループを含む速度制御を行うプログラム例を下記に示します。

・設定

項目	設定値
速度	100000.0
加速度	50000.0
減速度	50000.0
ジャーク	0.0

■軸

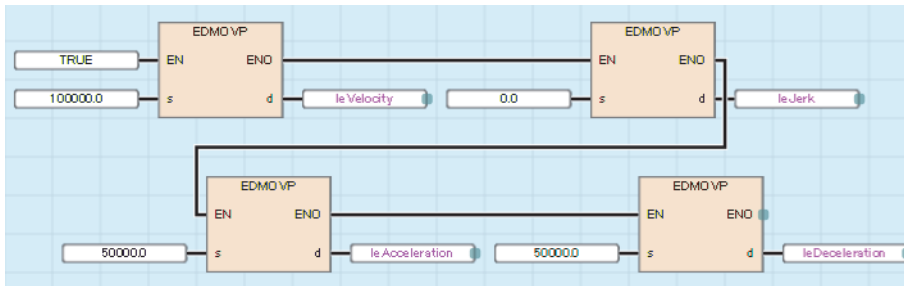
軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1

■使用するラベル

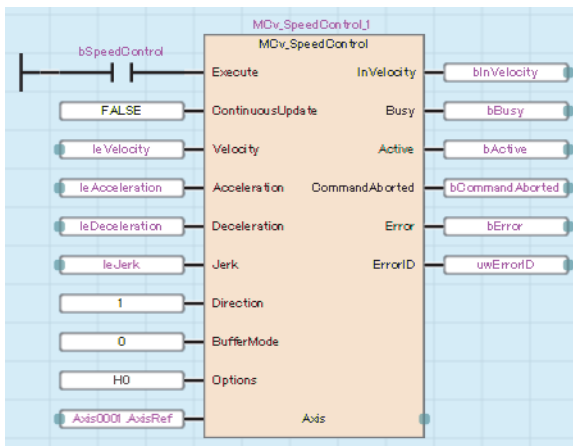
ラベル名	データ型	コメント
MCv_SpeedControl_1	MCv_SpeedControl	速度制御FB
bSpeedControl	ビット	速度制御指令
leVelocity	倍精度実数	指令速度
leAcceleration	倍精度実数	加速度
leDeceleration	倍精度実数	減速度
leJerk	倍精度実数	ジャーク
bInVelocity	ビット	目標速度到達
bBusy	ビット	実行中
bActive	ビット	制御中
bCommandAborted	ビット	実行中断
bError	ビット	エラー
uwErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

- 速度制御用データ設定



- 速度制御(位置ループを含む)



■STプログラム

//-----速度制御用データ設定-----

leVelocity:= 100000.0;

leAcceleration:= 50000.0;

leDeceleration:= 50000.0;

leJerk:= 0.0;

//-----速度制御(位置ループを含む)-----

```
MCv_SpeedControl_1(
    Axis:= Axis0001.AxisRef ,
    Execute:= bSpeedControl ,
    ContinuousUpdate:= FALSE ,
    Velocity:= leVelocity ,
    Acceleration:= leAcceleration ,
    Deceleration:= leDeceleration ,
    Jerk:= leJerk ,
    Direction:= MC_DIRECTION__mcPositiveDirection ,
    BufferMode:= MC_BUFFER_MODE__mcAborting ,
    Options:= H00000000 ,
    InVelocity=> bInVelocity ,
    Busy=> bBusy ,
    Active=> bActive ,
    CommandAborted=> bCommandAborted ,
    Error=> bError ,
    ErrorID=> uwErrorID
);
```


42.10 絶対値直線補間制御

42

MCv_MoveLinearInterpolateAbsolute

設定した軸グループの絶対位置による目標位置を設定し、直線補間制御による位置決めを実行します。

ラダー	FBD/LD	ST
		<pre> MCv_MoveLinearInterpolateAbsolute(AxesGroup:= ?AXES_GROUP_REF?, Execute:= ?BOOL?, ContinuousUpdate:= ?BOOL?, LinearAxes:= ?INT(0..15)?, Position:= ?REAL(0..15)?, Velocity:= ?REAL?, Acceleration:= ?REAL?, Deceleration:= ?REAL?, Jerk:= ?REAL?, VelocityMode:= ?INT?, Direction:= ?INT(0..15)?, BufferMode:= ?INT?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?); </pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
AxesGroup	軸グループ情報	AXES_GROUP_REF	起動時	—	省略不可	軸グループを設定します。 使用する変数 (AxesGroupName.AxesGroupRef.)については、下記を参照してください。 1402ページ AxesGroupName.AxesGroupRef.(軸グループ情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	起動	BOOL	起動時	TRUE, FALSE	FALSE	TRUEで MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)を実行します。
ContinuousUpdate	連続更新	BOOL	起動時	TRUE, FALSE	FALSE	連続して、速度 (Velocity)、加速度 (Acceleration)、減速度 (Deceleration)の変更を有効にするか、無効にするかを設定します。 連続更新により、実行中のインスタンスに対し、動作を中断することなく入力変数の再取込みを行います。 • FALSE: 無効 • TRUE: 有効

入力変数	名称	データ型	取込	有効範囲	初期値	内容
LinearAxes	直線補間軸	INT[0..15]	起動時	1~16	0	直線補間制御に使用する軸を構成軸から設定します。 構成軸のインデックス番号(1~16)を配列で設定します。 *: 速度モード (VelocityMode)に「2: 基準軸速度 (ReferenceAxisSpeed)」を設定した場合、配列の第1要素を基準軸とみなします。 詳細は、下記を参照してください。 📖 1647ページ 直線補間軸(LinearAxes)
Position	目標位置	LREAL[0..15]	起動時	-10000000000.0~10000000000.0	0.0	直線補間の目標位置を設定します。 1次元の配列データです。 構成軸1~16の絶対位置として扱います。 詳細は、下記を参照してください。 📖 1647ページ 目標位置(Position)
Velocity	速度	LREAL	起動時/再起動可/連続更新可	0.0, 0.0001~2500000000.0	0.0	速度指令値を設定します。 詳細は、下記を参照してください。 📖 1648ページ 速度(Velocity)
Acceleration	加速度	LREAL	起動時/再起動可/連続更新可	0.0000, 0.0001~2147483647.0	0.0	加速度を設定します。 詳細は、下記を参照してください。 📖 1648ページ 加速度(Acceleration)
Deceleration	減速度	LREAL	起動時/再起動可/連続更新可	0.0000, 0.0001~2147483647.0	0.0	減速度を設定します。 詳細は、下記を参照してください。 📖 1648ページ 減速度(Deceleration)
Jerk	ジャーク	LREAL	起動時	0.0000, 0.0001~2147483647.0	0.0	ジャークを設定します。 詳細は、下記を参照してください。 📖 1648ページ ジャーク(Jerk)
VelocityMode	速度モード	INT (MC_INTERPOLATE_SPEED_MODE)	起動時	0~2	0	補間制御の速度モードを設定します。 • 0: 合成速度 (VectorSpeed) • 1: 長軸速度 (LongAxisSpeed) • 2: 基準軸速度 (ReferenceAxisSpeed) 詳細は、下記を参照してください。 📖 1649ページ 速度モード(VelocityMode)

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Direction	方向選択	INT (MC_DIRECTION[0..15])	起動時	1~3	0	方向選択を設定します。 1次元の配列データです。 構成軸1~16の方向選択として扱います。 • 1: 正方向 (mcPositiveDirection) • 2: 負方向 (mcNegativeDirection) • 3: 最短経路 (mcShortestWay) *: 設定を省略した場合、 方向選択範囲外(エラーコード: 1AA5H)となります。 詳細は、下記を参照してください。 1651ページ 方向選択 (Direction)
BufferMode	バッファモード	INT (MC_BUFFER_MODE)	起動時	0~5	0	バッファモードを設定します。 • 0: Aborting(mcAborting) • 1: Buffered(mcBuffered) • 2: BlendingLow(mcBlendingLow) • 3: BlendingPrevious(mcBlendingPrevious) • 4: BlendingNext(mcBlendingNext) • 5: BlendingHigh(mcBlendingHigh) 詳細は、下記を参照してください。 1651ページ バッファモード(BufferMode)
Options	オプション	DWORD(HEX)	起動時	00000000H~00010001H	00000000H	MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)の機能オプションをビット指定で設定します。 詳細は、下記を参照してください。 1652ページ オプション(Options)

■出力変数

出力変数	名称	データ型	初期値	内容
Done	完了	BOOL	FALSE	制御が完了したときに、TRUEになります。 動作完了時に起動(Execute)の状態により下記となります。 ■起動(Execute)がTRUEの場合 起動(Execute)をFALSEにするまでTRUEのままになります。 ■起動(Execute)がFALSEの場合 1周期だけTRUEになります。
Busy	実行中	BOOL	FALSE	MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)を実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)が軸を制御中のときに、TRUEになります。 同一軸グループに対して複数のMCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)を実行した場合、制御中(Active)がTRUEになるのは、1つのMCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)のみです。
CommandAborted	実行中断	BOOL	FALSE	エラーや多重起動などにより、MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)の実行が中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 📖使用するコントローラのユーザーズマニュアル

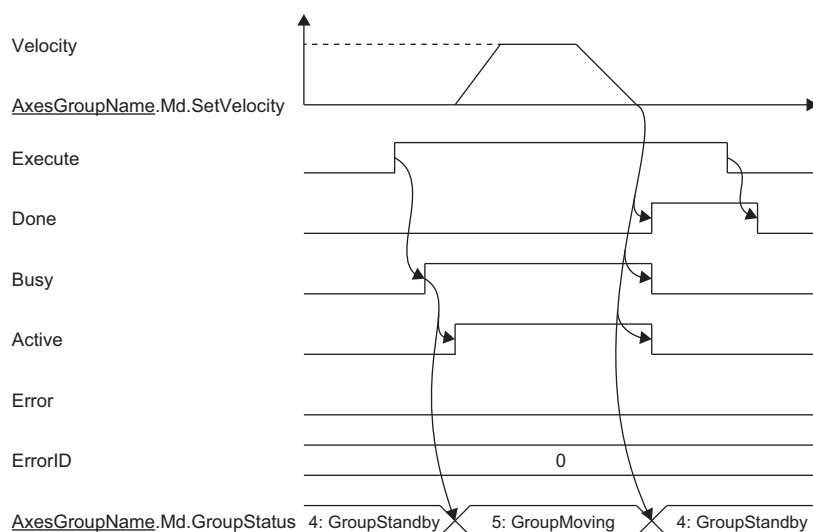
機能

- 直線補間制御は、軸グループを指定して、始点(移動開始点)から終点までの軌跡が直線となるように補間制御を行います。直線補間制御では、最大4軸を用いた補間制御を行います。
- MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)では、絶対位置の目標位置を指定して絶対値直線補間制御を行います。
- 1軸~3軸の絶対値直線補間制御の動作については、下記マニュアルの「直線補間制御」を参照してください。

📖使用するコントローラのユーザーズマニュアル

■タイミングチャート

- 正常完了の場合



- 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■直線補間軸(LinearAxes)

直線補間制御では、軸グループに設定した構成軸から、任意の軸を使用して直線補間を行います。

軸グループの構成軸の中より、直線補間を行う構成軸を直線補間軸(LinearAxes)により設定します。直線補間軸(LinearAxes)は、16個の配列要素を持ちます。配列には、補間制御で使用する構成軸のインデックス番号(1~16: 構成軸1~16)を補間制御で使用する構成軸のみを前詰めで設定し、残りの配列には「0」を設定します。「0」を設定した配列は省略することもできます。

- 直線補間軸の設定数が直線補間制御の最大補間軸数である4軸以下となるように設定してください。
- 補間制御で使用する構成軸の数は、構成軸に登録した軸の数以下となるように設定してください。
- 速度モード(VelocityMode)が「2: 基準軸速度(ReferenceAxisSpeed)」の場合、直線補間軸(LinearAxes)の第1要素の構成軸が基準軸となります。
- 下記を設定した場合、直線補間軸設定不正(エラーコード: 1B08H)となり始動しません。
 - 軸が設定していない構成軸を設定した場合
 - 直線補間軸(LinearAxes)の第1要素が「0」の場合
 - 直線補間軸(LinearAxes)に同じインデックス番号を重複して設定した場合
 - 直線補間軸(LinearAxes)の設定軸数が最大補間軸数を越えた場合

例

直線補間軸に構成軸2, 3, 4を設定する場合

```
LinearAxes[0]:= 2;
LinearAxes[1]:= 3;
LinearAxes[2]:= 4;
LinearAxes[3]:= 0;*1
:
LinearAxes[15]:= 0;*1
```

*1 「LinearAxes[3]:= 0;~LinearAxes[15]:= 0;」は省略できます。

■目標位置(Position)

直線補間の目標位置を設定します。

目標位置(Position)は16個の配列要素を持ちます。

- 設定可能な有効範囲は、ソフトウェアストロークリミット有効/無効、方向選択(Direction)、リングカウンタを超えた目標位置指定(オプション(Options): ビット16)により異なります。詳細は下記を参照してください。

1592ページ 目標位置(Position)

- Position[0..15]は、構成軸1~16の目標位置を示します。直線補間軸(LinearAxes)で設定した構成軸に対して、位置を設定します。
- 直線補間軸に設定していない構成軸の目標位置(Position)は無視します。
- 直線補間軸に設定したすべての構成軸より後方の目標位置(Position)は省略できます。

例

直線補間軸に構成軸2, 3, 4を設定し、構成軸2, 3, 4の目標位置に2000.0, 3000.0, 4000.0を設定する場合

```
LinearAxes[0]:= 2;
LinearAxes[1]:= 3;
LinearAxes[2]:= 4;
LinearAxes[3]:= 0;*1
:
LinearAxes[15]:= 0;*1
Position[0]:= 0.0;
Position[1]:= 2000.0;
Position[2]:= 3000.0;
Position[3]:= 4000.0;
Position[4]:= 0.0;*1
:
Position[15]:= 0.0;*1
```

*1 「LinearAxes[3]:= 0;~LinearAxes[15]:= 0;」と「Position[4]:= 0.0;~Position[15]:= 0.0;」は省略できます。

■速度(Velocity)

MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)での経路の最大速度を設定します。

設定範囲
0.0, 0.0001~2500000000.0 ^{*1*2}

- *1 浮動小数点演算を行うため、指令速度の下限値には下記の制約が生じます。
指令速度を演算周期換算した速度が「0.00001」未満となる場合、演算周期換算速度範囲外(エラーコード: 1B16H)(速度変更時は演算周期換算速度範囲外警告(イベントコード: 00D2FH))となります。浮動小数点演算の精度を向上させるためには位置指令単位(AxisName.Pr.Unit_Position)、または速度指令単位(AxisName.Pr.Unit_Velocity)を変更することで演算周期換算後の速度が「0.00001」未満とならないように設定してください。
- *2 多重起動を行うモーション制御FBにて指定速度が「0.0」の場合、直前のモーション制御FBの指定速度となります。

■加速度(Acceleration)

MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)での加速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の設定により設定範囲が異なります。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	0.000000 ^{*1} , 0.000001~8400.0[s]までの正数

- *1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更、減速度変更時は、変更を受け付けません。

■減速度(Deceleration)

MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)での減速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合、減速度(Deceleration)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

- *1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更、減速度変更時は、変更を受け付けません。

■ジャーク(Jerk)

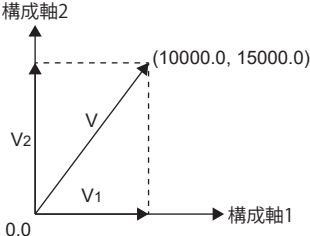
MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)でのジャークを設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合、ジャーク(Jerk)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000, 0.0001~2147483647.0[U/s ³]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

■速度モード(VelocityMode)

直線補間制御における、速度モードを設定します。速度モード(VelocityMode)に「2: 基準軸速度(ReferenceAxisSpeed)」を設定した場合、基準軸は直線補間軸(LinearAxes)の第1要素で設定した構成軸となります。

設定値	内容
0: 合成速度(VectorSpeed)	速度(Velocity)は、合成速度として設定します。 各軸の位置決め速度(V_n)は、設定した制御対象の位置決め速度(V)に基づいて、モーションシステムが各軸の移動量(D_n)から算出します。 <例> 2軸の直線補間制御の場合  - 構成軸1の移動量(D_1): 10000.0[pulse] - 構成軸2の移動量(D_2): 15000.0[pulse] - 軸グループの速度単位: [s] - 合成速度(V): 7000.0[pulse/s] 上記の場合、各軸の位置決め速度は下記の計算式によって、モーションシステムが算出します。 - 構成軸1の位置決め速度: $V_1 = V \times D_1 / \sqrt{D_1^2 + D_2^2}$ - 構成軸2の位置決め速度: $V_2 = V \times D_2 / \sqrt{D_1^2 + D_2^2}$ ■ポイント - 合成速度の場合、設定した速度制限値は、合成速度としての速度(Velocity)に対して有効です。
1: 長軸速度(LongAxisSpeed)	速度(Velocity)は、長軸の速度として設定します。 各補間軸に設定した指定位置のうち、最も移動量が大きい補間軸の位置決め速度(長軸速度: V)に基づいて制御します。 その他の補間軸の位置決め速度(V_n)は、モーションシステムが各補間軸の移動量(D_n)から算出します。 <例> 4軸の直線補間制御の場合 - 構成軸1の移動量(D_1): 10000.0[pulse] - 構成軸2の移動量(D_2): 15000.0[pulse] - 構成軸3の移動量(D_3): 5000.0[pulse] - 構成軸4の移動量(D_4): 20000.0[pulse] - 構成軸4の速度単位: [s] - 長軸速度(V): 7000.0[pulse/s] 上記の場合、長軸は移動量が最も大きい構成軸4であり、構成軸4を長軸速度で制御します。 他構成軸の位置決め速度は、下記の計算式によってモーションシステムが算出します。 - 構成軸1の位置決め速度: $V_1 = D_1 / D_4 \times V$ - 構成軸2の位置決め速度: $V_2 = D_2 / D_4 \times V$ - 構成軸3の位置決め速度: $V_3 = D_3 / D_4 \times V$ ■ポイント - 長軸速度の場合、設定した速度制限値は、長軸速度としての速度(Velocity)に対して有効です。 - 長軸速度指定時の合成速度は、速度制限値よりも大きくなる場合がありますので注意してください。2軸の直線補間で下記のような値を設定している場合は、合成速度が速度制限値を超えます。 <例> 設定項目と設定値が下記の場合 - 直線補間軸: 構成軸1, 構成軸2 - 構成軸1の移動量: 100[pulse] - 構成軸2の移動量: 200[pulse] - 長軸速度: 50[pulse/s] - 構成軸2の速度制限値: 55[pulse/s] 上記の場合、基準軸は移動量が最も大きい構成軸2であり、構成軸2で設定した速度制限値で制御します。また、各軸の位置決め速度および合成速度は下記となります。 - 構成軸1の位置決め速度: $100 / 200 \times 50 = 25$[pulse/s] - 構成軸2の位置決め速度: 50[pulse/s] - 合成速度: $\sqrt{(25^2 + 50^2)} = 55.9$[pulse/s] 合成速度は、構成軸2の速度制限値「55」を超えた値となります。

設定値	内容
2: 基準軸速度(ReferenceAxisSpeed)	速度(Velocity)は、基準軸の速度として設定します。 設定した基準軸の位置決め速度(基準軸速度: V)に基づいて、その他の補間軸の位置決め速度(Vn)を、モーションシステムが、各補間軸の移動量(Dn)から算出して制御します。 基準軸に設定した構成軸の移動量が「0.0」の場合、基準軸移動量0(エラーコード: 1AFAH)となります。 <例> 4軸の直線補間制御の場合 - 構成軸1の移動量(D₁): 10000.0[pulse] - 構成軸2の移動量(D₂): 15000.0[pulse] - 構成軸3の移動量(D₃): 5000.0[pulse] - 構成軸4の移動量(D₄): 20000.0[pulse] - 基準軸速度(V): 7000.0[pulse/s] 上記の場合、基準軸は構成軸4であり、構成軸4で設定した位置決め速度で制御します。 他軸の位置決め速度は、下記の計算式によってモーションシステムが算出します。 - 構成軸1の位置決め速度: $V_1 = D_1 / D_4 \times V$ - 構成軸2の位置決め速度: $V_2 = D_2 / D_4 \times V$ - 構成軸3の位置決め速度: $V_3 = D_3 / D_4 \times V$ ■ポイント - 基準軸速度の場合、設定した速度制限値は、基準軸速度としての速度(Velocity)に対して有効です。 - 基準軸より移動量大きい軸の位置決め速度は、設定した基準軸速度よりも大きくなりますので注意してください。2軸の直線補間で下記のような値を設定している場合は、構成軸2の位置決め速度と合成速度が速度制限値を超えます。 <例> 設定項目と設定値が下記の場合 - 直線補間軸: 構成軸1, 構成軸2 - 構成軸1の移動量: 100[pulse] - 構成軸2の移動量: 200[pulse] - 基準軸速度: 50[pulse/s] - 構成軸1の速度制限値: 55[pulse/s] 上記の場合、基準軸は構成軸1であり、構成軸1で設定した速度制限値で制御されます。また、各補間軸の位置決め速度および合成速度は下記となります。 - 構成軸1の位置決め速度: 50[pulse/s] - 構成軸2の位置決め速度: $200 / 100 \times 50 = 100$[pulse/s] - 合成速度: $\sqrt{(50^2 + 100^2)} = 111.8$[pulse/s] 構成軸2の位置決め速度と合成速度は、構成軸1の速度制限値「55」を超えた値となります。

制約事項

- 速度モード(VelocityMode)に「0: 合成速度(VectorSpeed)」を設定したとき、各軸の移動量が「4,294,967,296.0(=2³²)」を超える場合は、直線移動量範囲外(エラーコード: 1B10H)となり始動しません。
- バッファモード(BufferMode)を「0: Aborting(mcAborting)」に設定して、MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)を多重起動する場合、速度モード(VelocityMode)には「0: 合成速度(VectorSpeed)」を設定してください。「0: 合成速度(VectorSpeed)」以外を設定した場合は、速度モード範囲外(エラーコード: 1AC9H)となります。

■方向選択(Direction)

正方向，負方向，最短経路から選択でき，現在位置から目標位置へ移動する方向を設定します。方向選択(Direction)は16個の配列要素を持ちます。

- Direction[0..15]は，構成軸1~16の方向選択を示します。直線補間軸(LinearAxes)で設定した構成軸に対して，値を設定します。
- 直線補間軸に設定していない構成軸の方向選択(Direction)は無視します。
- 直線補間軸に設定したすべての構成軸より後方の方向選択(Direction)は省略できます。
- 設定値以外を設定した場合，方向選択範囲外(エラーコード: 1AA5H)となり始動しません。
- ソフトウェアストロークリミットが有効の補間軸は，方向選択(Direction)の設定を無視します。方向選択の動作詳細は，下記を参照してください。

📖 1594ページ 方向選択(Direction)

設定値	内容
1: 正方向(mcPositiveDirection)	現在位置から正方向(アドレス増加)の目標位置に向かって位置決めを行います。
2: 負方向(mcNegativeDirection)	現在位置から負方向(アドレス減少)の目標位置に向かって位置決めを行います。
3: 最短経路(mcShortestWay)	現在位置と目標位置の距離が近い方向に向かって位置決めを行います。

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)で設定可能なバッファモードを下記に示します。

設定値	内容
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。
1: Buffered(mcBuffered)	制御中のFBに次のFBをバッファリングします。 制御中のFBをすでにバッファリングしている場合，直前のFBの次にバッファリングします。(最大2つ) 制御中のFB完了にて，バッファリングFBを順次実行します。
2: BlendingLow(mcBlendingLow)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合，直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち，バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき，制御中のFBとバッファリングFBの目標速度のうち低い方の速度を切換え速度とします。
3: BlendingPrevious(mcBlendingPrevious)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合，直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち，バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき，制御中のFBの目標速度を切換え速度とします。
4: BlendingNext(mcBlendingNext)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合，直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち，バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき，バッファリングFBの目標速度を切換え速度とします。
5: BlendingHigh(mcBlendingHigh)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合，直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち，バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき，制御中のFBとバッファリングFBの目標速度のうち高い方の速度を切換え速度とします。

*1 制御中のFBとバッファリングFB間で停止を行いません。

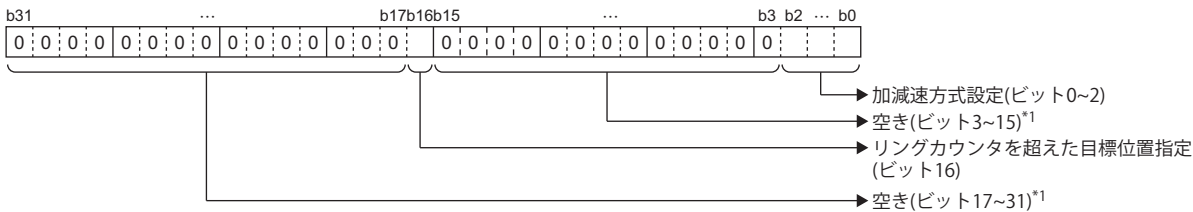
Point

多重起動(バッファモード)の詳細については，下記を参照してください。

📖 使用するコントローラのユーザーズマニュアル

■オプション(Options)

MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)で使用する機能オプションをビット指定で設定します。
ビット指定で設定する内容を下記に示します。



*1 空きには「0」を設定してください。「0」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

ビット	名称	内容
0~2	加減速方式設定	制御を行うための加減速方式を設定します。 • 0: 加減速度指定方式(mcAccDec) • 1: 加減速時間一定方式(mcFixedTime)
16	リングカウンタを超えた目標位置指定	ソフトウェアストロークリミット無効時に、リングカウンタ上限値、リングカウンタ下限値を超えた目標位置を許可するか、許可しないかを設定します。 • 0: 許可しない • 1: 許可する

・加減速方式設定(ビット0~2)

設定値	内容
0: 加減速度指定方式(mcAccDec)	MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)で設定した加速度(Acceleration), 減速度(Deceleration), ジャーク(Jerk)を用いて, 加速/減速する方式です。
1: 加減速時間一定方式(mcFixedTime)	速度に関係なく, MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)で設定した加減速時間を用いて, 加速/減速する方式です。 加減速時間は, 加速度(Acceleration)に設定し, 減速度(Deceleration), ジャーク(Jerk)は使用しません。

■必須オブジェクトデータ

MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)を使用する場合, 指定した軸グループの構成軸すべてに下記のオブジェクトデータを設定してください。

・ Target position(607AH)

オブジェクトデータを設定していない構成軸がある場合, 必須オブジェクトデータ未設定(エラーコード: 1AF7H)となり始動しません。

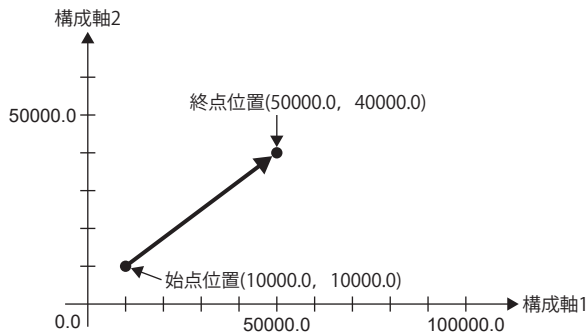
オブジェクトデータ設定に関する詳細は, 下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

プログラム例

直線補間制御始動(bLinearInterpolateCMD)をTRUEにし、軸グループ1(AxesGroup001)を有効にした後、下記の設定により軸グループ1(AxesGroup001)の2軸直線補間の絶対値位置決め制御を行うプログラム例を下記に示します。

・動作



・軸グループ

項目	設定値
構成軸[1]	Axis0001
構成軸[2]	Axis0002

・設定

項目	設定値	
	構成軸1	構成軸2
目標位置	50000.0	40000.0
速度	50000.0	
加速度	50000.0	
減速度	50000.0	
ジャーク	0.0	
方向選択	正方向	正方向

■軸

軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1
2	Axis0002	AXIS_REF	軸2

■軸グループ

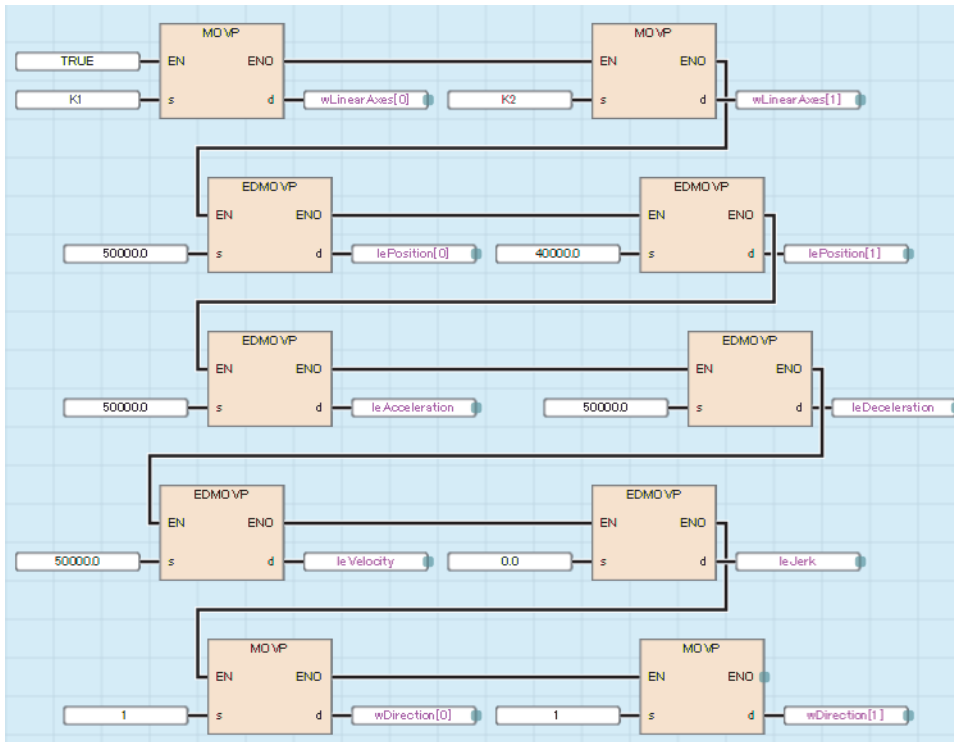
軸グループNo.	ラベル名	データ型	コメント
1	AxesGroup001	AXES_GROUP_REF	軸グループ1

■使用するラベル

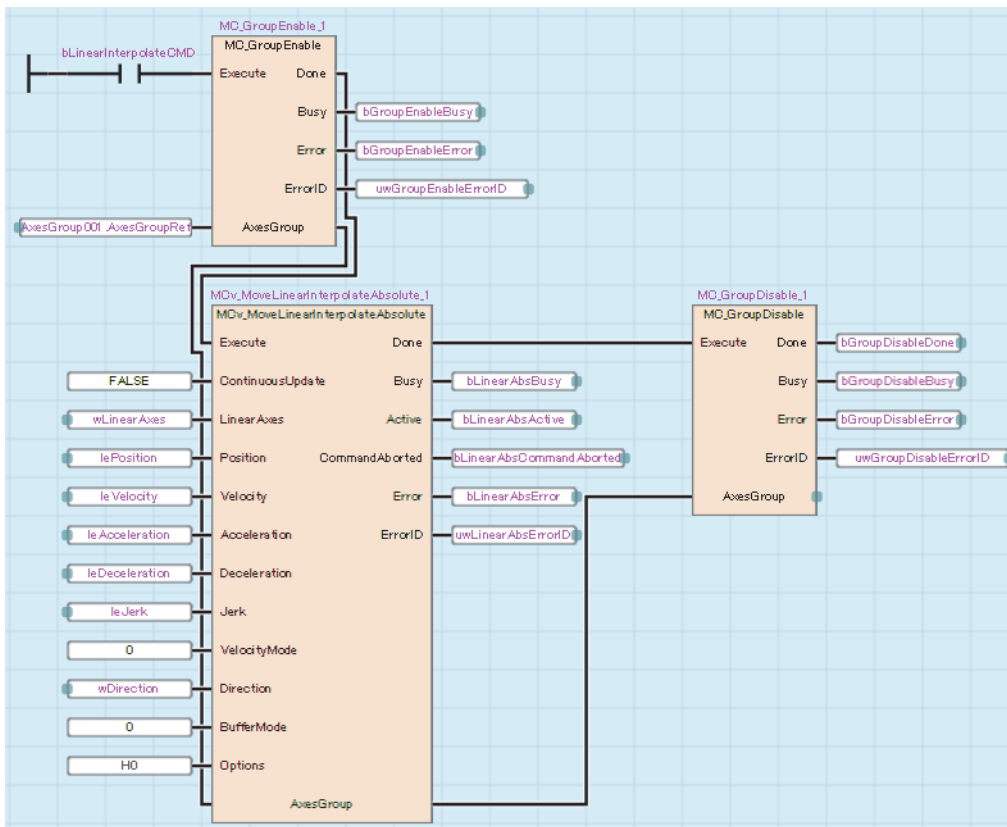
ラベル名	データ型	コメント
MCv_MoveLinearInterpolateAbsolute_1	MCv_MoveLinearInterpolateAbsolute	絶対値直線補間制御FB
wLinearAxes	ワード[符号付き](0..15)	直線補間軸
lePosition	倍精度実数(0..15)	目標位置
leVelocity	倍精度実数	速度
leAcceleration	倍精度実数	加速度
leDeceleration	倍精度実数	減速度
leJerk	倍精度実数	ジャーク
wDirection	ワード[符号付き](0..15)	方向選択
bLinearAbsDone	ビット	実行完了
bLinearAbsBusy	ビット	実行中
bLinearAbsActive	ビット	制御中
bLinearAbsCommandAborted	ビット	実行中断
bLinearAbsError	ビット	エラー
uwLinearAbsErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
MC_GroupEnable_1	MC_GroupEnable	軸グループ有効FB
bLinearInterpolateCMD	ビット	直線補間制御始動
bGroupEnableDone	ビット	軸グループ有効完了
bGroupEnableBusy	ビット	実行中
bGroupEnableError	ビット	エラー
uwGroupEnableErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
MC_GroupDisable_1	MC_GroupDisable	軸グループ無効FB
bGroupDisableDone	ビット	軸グループ無効完了
bGroupDisableBusy	ビット	実行中
bGroupDisableError	ビット	エラー
uwGroupDisableErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

・2軸直線補間制御用データ設定



・軸グループ有効/2軸直線補間制御/軸グループ無効



■STプログラム

//-----2軸直線補間制御用データ設定-----

```
wLinearAxes[0]:= 1;  
wLinearAxes[1]:= 2;  
lePosition[0]:= 50000.0;  
lePosition[1]:= 40000.0;  
leVelocity:= 50000.0;  
leAcceleration:= 50000.0;  
leDeceleration:= 50000.0;  
leJerk:= 0.0;  
wDirection[0]:= MC_DIRECTION__mcPositiveDirection;  
wDirection[1]:= MC_DIRECTION__mcPositiveDirection;
```

//-----軸グループ有効-----

```
MC_GroupEnable_1(  
  AxesGroup:= AxesGroup001.AxesGroupRef ,  
  Execute:= bLinearInterpolateCMD ,  
  Done=> bGroupEnableDone ,  
  Busy=> bGroupEnableBusy ,  
  Error=> bGroupEnableError ,  
  ErrorID=> uwGroupEnableErrorID  
);
```

//-----2軸直線補間制御-----

```
MCv_MoveLinearInterpolateAbsolute_1(  
  AxesGroup:= AxesGroup001.AxesGroupRef ,  
  Execute:= bGroupEnableDone ,  
  ContinuousUpdate:= FALSE ,  
  LinearAxes:= wLinearAxes ,  
  Position:= lePosition ,  
  Velocity:= leVelocity ,  
  Acceleration:= leAcceleration ,  
  Deceleration:= leDeceleration ,  
  Jerk:= leJerk ,  
  VelocityMode:= MC_INTERPOLATE_SPEED_MODE__VectorSpeed ,  
  Direction:= wDirection ,  
  BufferMode:= MC_BUFFER_MODE__mcAborting ,  
  Options:= H00000000 ,  
  Done=> bLinearAbsDone ,  
  Busy=> bLinearAbsBusy ,  
  Active=> bLinearAbsActive ,  
  CommandAborted=> bLinearAbsCommandAborted ,  
  Error=> bLinearAbsError ,  
  ErrorID=> uwLinearAbsErrorID  
);
```

```
//-----軸グループ無効-----  
MC_GroupDisable_1(  
  AxesGroup:= AxesGroup001.AxesGroupRef ,  
  Execute:= bLinearAbsDone ,  
  Done=> bGroupDisableDone ,  
  Busy=> bGroupDisableBusy ,  
  Error=> bGroupDisableError ,  
  ErrorID=> uwGroupDisableErrorID  
);
```

42.11 相対値直線補間制御

MCv_MoveLinearInterpolateRelative

設定した軸グループの相対位置による移動量を設定し、直線補間制御による位置決めを実行します。

ラダー	FBD/LD	ST
		<pre>MCv_MoveLinearInterpolateRelative(AxesGroup:= ?AXES_GROUP_REF?, Execute:= ?BOOL?, ContinuousUpdate:= ?BOOL?, LinearAxes:= ?INT(0..15)?, Distance:= ?LREAL(0..15)?, Velocity:= ?LREAL?, Acceleration:= ?LREAL?, Deceleration:= ?LREAL?, Jerk:= ?LREAL?, VelocityMode:= ?INT?, BufferMode:= ?INT?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
AxesGroup	軸グループ情報	AXES_GROUP_REF	起動時	—	省略不可	軸グループを設定します。 使用する変数 (AxesGroupName.AxesGroupRef.) については、下記を参照してください。 📖 1402ページ AxesGroupName.AxesGroupRef.(軸グループ情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	起動	BOOL	起動時	TRUE, FALSE	FALSE	TRUEで MCv_MoveLinearInterpolateRelative(相対値直線補間制御)を実行します。
ContinuousUpdate	連続更新	BOOL	起動時	TRUE, FALSE	FALSE	連続して、速度(Velocity)、加速度(Acceleration)、減速度(Deceleration)の変更を有効にするか、無効にするかを設定します。 連続更新により、実行中のインスタンスに対し、動作を中断することなく入力変数の再取込みを行います。 • FALSE: 無効 • TRUE: 有効
LinearAxes	直線補間軸	INT[0..15]	起動時	1~16	0	直線補間制御に使用する軸を構成軸から指定します。 構成軸のインデックス番号(1~16)を配列で指定します。 *: 速度モード(VelocityMode)に「2: 基準軸速度 (ReferenceAxisSpeed)」を設定した場合、配列の第1要素を基準軸とみなします。 詳細は、下記を参照してください。 📖 1661ページ 直線補間軸 (LinearAxes)

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Distance	移動量	LREAL[0..15]	起動時	-10000000000.0~10000000000.0	0.0	始動時の現在位置から終点までの移動量を設定します。 1次元の配列データです。構成軸1~16の相対位置として扱います。詳細は、下記を参照してください。 ☞ 1661ページ 移動量(Distance)
Velocity	速度	LREAL	起動時/再起動可 /連続更新可	0.0, 0.0001~2500000000.0	0.0	速度指令値を設定します。 詳細は、下記を参照してください。 ☞ 1662ページ 速度(Velocity)
Acceleration	加速度	LREAL	起動時/再起動可 /連続更新可	0.0000, 0.0001~2147483647.0	0.0	加速度を設定します。 詳細は、下記を参照してください。 ☞ 1662ページ 加速度(Acceleration)
Deceleration	減速度	LREAL	起動時/再起動可 /連続更新可	0.0000, 0.0001~2147483647.0	0.0	減速度を設定します。 詳細は、下記を参照してください。 ☞ 1662ページ 減速度(Deceleration)
Jerk	ジャーク	LREAL	起動時	0.0000, 0.0001~2147483647.0	0.0	ジャークを設定します。 詳細は、下記を参照してください。 ☞ 1662ページ ジャーク(Jerk)
VelocityMode	速度モード	INT (MC_INTERPOLATE_SPEED_MODE)	起動時	0~2	0	補間制御の速度モードを設定します。 • 0: 合成速度(VectorSpeed) • 1: 長軸速度(LongAxisSpeed) • 2: 基準軸速度(ReferenceAxisSpeed) 詳細は、下記を参照してください。 ☞ 1663ページ 速度モード(VelocityMode)
BufferMode	バッファモード	INT (MC_BUFFER_MODE)	起動時	0~5	0	バッファモードを設定します。 • 0: Aborting(mcAborting) • 1: Buffered(mcBuffered) • 2: BlendingLow(mcBlendingLow) • 3: BlendingPrevious(mcBlendingPrevious) • 4: BlendingNext(mcBlendingNext) • 5: BlendingHigh(mcBlendingHigh) 詳細は、下記を参照してください。 ☞ 1665ページ バッファモード(BufferMode)
Options	オプション	DWORD(HEX)	起動時	00000000H~00000005H	00000000H	MCv_MoveLinearInterpolateRelative(相対値直線補間制御)の機能オプションをビット指定で設定します。 詳細は、下記を参照してください。 ☞ 1666ページ オプション(Options)

■出力変数

出力変数	名称	データ型	初期値	内容
Done	完了	BOOL	FALSE	制御が完了したときに、TRUEになります。 動作完了時に起動(Execute)の状態により下記となります。 ■起動(Execute)がTRUEの場合 起動(Execute)をFALSEにするまでTRUEのままになります。 ■起動(Execute)がFALSEの場合 1周期だけTRUEになります。
Busy	実行中	BOOL	FALSE	MCv_MoveLinearInterpolateRelative(相対値直線補間制御)を実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	MCv_MoveLinearInterpolateRelative(相対値直線補間制御)が軸を制御中のときに、TRUEになります。 同一軸グループに対して複数のMCv_MoveLinearInterpolateRelative(相対値直線補間制御)を実行した場合、制御中(Active)がTRUEになるのは、1つのMCv_MoveLinearInterpolateRelative(相対値直線補間制御)のみです。
CommandAborted	実行中断	BOOL	FALSE	エラーや多重起動などにより、MCv_MoveLinearInterpolateRelative(相対値直線補間制御)の実行が中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 📖使用するコントローラのユーザーズマニュアル

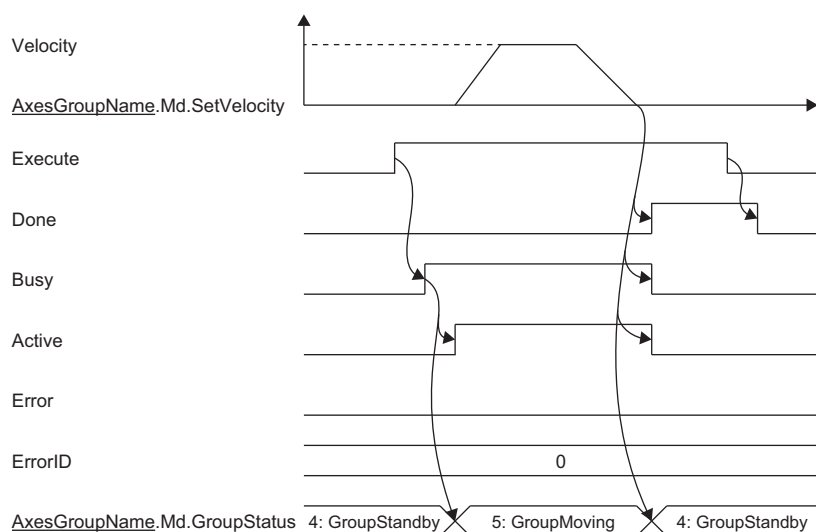
機能

- 直線補間制御は、軸グループを指定して、始点(移動開始点)から終点までの軌跡が直線となるように補間制御を行います。直線補間制御では、最大4軸を用いた補間制御を行います。
- MCv_MoveLinearInterpolateRelative(相対値直線補間制御)では、現在位置からの相対移動量を指定して直線補間制御を行います。
- 1軸~3軸相対値直線補間制御の動作については、下記マニュアルの「直線補間制御」を参照してください。

📖使用するコントローラのユーザーズマニュアル

■タイミングチャート

- 正常完了の場合



- 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■直線補間軸(LinearAxes)

直線補間制御では、軸グループに設定した構成軸から、任意の軸を使用して直線補間を行います。

軸グループの構成軸の中より、直線補間を行う構成軸を直線補間軸(LinearAxes)により設定します。直線補間軸(LinearAxes)は、16個の配列要素を持ちます。配列には、補間制御で使用する構成軸のインデックス番号(1~16: 構成軸1~16)を補間制御で使用する構成軸のみを前詰めで設定し、残りの配列には「0」を設定します。「0」を設定した配列は省略することもできます。

- 直線補間軸の設定数が直線補間制御の最大補間軸数である4軸以下となるように設定してください。
- 補間制御で使用する構成軸の数は、構成軸に登録した軸の数以下となるように設定してください。
- 速度モード(VelocityMode)が「2: 基準軸速度(ReferenceAxisSpeed)」の場合、直線補間軸(LinearAxes)の第1要素の構成軸が基準軸となります。
- 下記を設定した場合、直線補間軸設定不正(エラーコード: 1B0BH)となり始動しません。
 - 軸が設定していない構成軸を設定した場合
 - 直線補間軸(LinearAxes)の第1要素が「0」の場合
 - 直線補間軸(LinearAxes)に同じインデックス番号を重複して設定した場合
 - 直線補間軸(LinearAxes)の設定軸数が最大補間軸数を超えた場合

例

直線補間軸に構成軸2, 3, 4を設定する場合

```
LinearAxes[0]:= 2;
LinearAxes[1]:= 3;
LinearAxes[2]:= 4;
LinearAxes[3]:= 0;*1
:
LinearAxes[15]:= 0;*1
```

*1 「LinearAxes[3]:= 0;~LinearAxes[15]:= 0;」は省略できます。

■移動量(Distance)

始動時の現在位置から終点までの移動量を設定します。移動量(Distance)は16個の配列要素を持ちます。

- 移動量(Distance)の入力範囲は、-100000000000.0~100000000000.0です。
- Distance[0..15]は、構成軸1~16の移動量を示します。直線補間軸(LinearAxes)で設定した構成軸に対して、位置を設定します。
- 直線補間軸に設定していない構成軸の移動量(Distance)は無視します。
- 直線補間軸に設定したすべての構成軸より後方の移動量(Distance)は省略できます。

例

直線補間軸に構成軸2, 3, 4を設定し、構成軸2, 3, 4の目標位置に2000.0, 3000.0, 4000.0を設定する場合

```
LinearAxes[0]:= 2;
LinearAxes[1]:= 3;
LinearAxes[2]:= 4;
LinearAxes[3]:= 0;*1
:
LinearAxes[15]:= 0;*1
Distance[0]:= 0.0;
Distance[1]:= 2000.0;
Distance[2]:= 3000.0;
Distance[3]:= 4000.0;
Distance[4]:= 0.0;*1
:
Distance[15]:= 0.0;*1
```

*1 「LinearAxes[3]:= 0;~LinearAxes[15]:= 0;」と「Distance[4]:= 0.0;~Distance[15]:= 0.0;」は省略できます。

■速度(Velocity)

MCv_MoveLinearInterpolateRelative(相対値直線補間制御)での経路の最大速度を設定します。

設定範囲
0.0, 0.0001~2500000000.0*1*2

- *1 浮動小数点演算を行うため、指令速度の下限値には下記の制約が生じます。
指令速度を演算周期換算した速度が「0.00001」未満となる場合、演算周期換算速度範囲外(エラーコード: 1B16H)(速度変更時は演算周期換算速度範囲外警告(イベントコード: 00D2FH))となります。浮動小数点演算の精度を向上させるためには位置指令単位(AxisName.Pr.Unit_Position)、または速度指令単位(AxisName.Pr.Unit_Velocity)を変更することで演算周期換算後の速度が「0.00001」未満とならないように設定してください。
- *2 多重起動を行うモーション制御FBにて指定速度が「0.0」の場合、直前のモーション制御FBの指定速度となります。

■加速度(Acceleration)

MCv_MoveLinearInterpolateRelative(相対値直線補間制御)での加速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の設定により設定範囲が異なります。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000*1, 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	0.000000*1, 0.000001~8400.0[s]までの正数

- *1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更、減速度変更時は、変更を受け付けません。

■減速度(Deceleration)

MCv_MoveLinearInterpolateRelative(相対値直線補間制御)での減速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合、減速度(Deceleration)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000*1, 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

- *1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更、減速度変更時は、変更を受け付けません。

■ジャーク(Jerk)

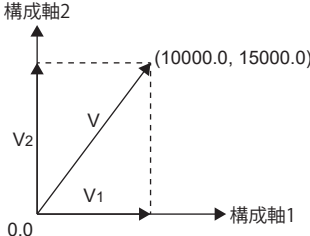
MCv_MoveLinearInterpolateRelative(相対値直線補間制御)でのジャークを設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合、ジャーク(Jerk)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000, 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

■速度モード(VelocityMode)

直線補間制御における、速度モードを設定します。速度モード(VelocityMode)に「2: 基準軸速度(ReferenceAxisSpeed)」を設定した場合、基準軸は直線補間軸(LinearAxes)の第1要素で設定した構成軸となります。

設定値	内容
0: 合成速度(VectorSpeed)	速度(Velocity)は、合成速度として設定します。 各軸の位置決め速度(V_n)は、設定した制御対象の位置決め速度(V)に基づいて、モーションシステムが各軸の移動量(D_n)から算出します。 <例> 2軸の直線補間制御の場合  - 構成軸1の移動量(D_1): 10000.0[pulse] - 構成軸2の移動量(D_2): 15000.0[pulse] - 軸グループの速度単位: [s] - 合成速度(V): 7000.0[pulse/s] 上記の場合、各軸の位置決め速度は下記の計算式によって、モーションシステムが算出します。 - 構成軸1の位置決め速度: $V_1 = V \times D_1 / \sqrt{D_1^2 + D_2^2}$ - 構成軸2の位置決め速度: $V_2 = V \times D_2 / \sqrt{D_1^2 + D_2^2}$ ■ポイント - 合成速度の場合、設定した速度制限値は、合成速度としての速度(Velocity)に対して有効です。
1: 長軸速度(LongAxisSpeed)	速度(Velocity)は、長軸の速度として設定します。 各補間軸に設定した指定位置のうち、最も移動量大きい補間軸の位置決め速度(長軸速度: V)に基づいて制御します。その他の補間軸の位置決め速度(V_n)は、モーションシステムが各補間軸の移動量(D_n)から算出します。 <例> 4軸の直線補間制御の場合 - 構成軸1の移動量(D_1): 10000.0[pulse] - 構成軸2の移動量(D_2): 15000.0[pulse] - 構成軸3の移動量(D_3): 5000.0[pulse] - 構成軸4の移動量(D_4): 20000.0[pulse] - 構成軸4の速度単位: [s] - 長軸速度(V): 7000.0[pulse/s] 上記の場合、長軸は移動量が最も大きい構成軸4であり、構成軸4を長軸速度で制御します。他構成軸の位置決め速度は、下記の計算式によってモーションシステムが算出します。 - 構成軸1の位置決め速度: $V_1 = D_1 / D_4 \times V$ - 構成軸2の位置決め速度: $V_2 = D_2 / D_4 \times V$ - 構成軸3の位置決め速度: $V_3 = D_3 / D_4 \times V$ ■ポイント - 長軸速度の場合、設定した速度制限値は、長軸速度としての速度(Velocity)に対して有効です。 - 長軸速度指定時の合成速度は、速度制限値よりも大きくなることがありますので注意してください。2軸の直線補間で下記のような値を設定している場合は、合成速度が速度制限値を超えます。 <例> 設定項目と設定値が下記の場合 - 直線補間軸: 構成軸1, 構成軸2 - 構成軸1の移動量: 100[pulse] - 構成軸2の移動量: 200[pulse] - 長軸速度: 50[pulse/s] - 構成軸2の速度制限値: 55[pulse/s] 上記の場合、基準軸は移動量が最も大きい構成軸2であり、構成軸2で設定した速度制限値で制御します。また、各軸の位置決め速度および合成速度は下記となります。 - 構成軸1の位置決め速度: $100 / 200 \times 50 = 25$[pulse/s] - 構成軸2の位置決め速度: 50[pulse/s] - 合成速度: $\sqrt{25^2 + 50^2} = 55.9$[pulse/s] 合成速度は、構成軸2の速度制限値「55」を超えた値となります。

設定値	内容
2: 基準軸速度(ReferenceAxisSpeed)	速度(Velocity)は、基準軸の速度として設定します。 設定した基準軸の位置決め速度(基準軸速度: V)に基づいて、その他の補間軸の位置決め速度(Vn)を、モーションシステムが、各補間軸の移動量(Dn)から算出して制御します。 基準軸に設定した構成軸の移動量が「0.0」の場合、基準軸移動量0(エラーコード: 1AFAH)となります。 <例> 4軸の直線補間制御の場合 - 構成軸1の移動量(D₁): 10000.0[pulse] - 構成軸2の移動量(D₂): 15000.0[pulse] - 構成軸3の移動量(D₃): 5000.0[pulse] - 構成軸4の移動量(D₄): 20000.0[pulse] - 基準軸速度(V): 7000.0[pulse/s] 上記の場合、基準軸は構成軸4であり、構成軸4で設定した位置決め速度で制御します。 他軸の位置決め速度は、下記の計算式によってモーションシステムが算出します。 - 構成軸1の位置決め速度: $V_1 = D_1 / D_4 \times V$ - 構成軸2の位置決め速度: $V_2 = D_2 / D_4 \times V$ - 構成軸3の位置決め速度: $V_3 = D_3 / D_4 \times V$ ■ポイント - 基準軸速度の場合、設定した速度制限値は、基準軸速度としての速度(Velocity)に対して有効です。 - 基準軸より移動量大きい軸の位置決め速度は、設定した基準軸速度よりも大きくなりますので注意してください。2軸の直線補間で下記のような値を設定している場合は、構成軸2の位置決め速度と合成速度が速度制限値を超えます。 <例> 設定項目と設定値が下記の場合 - 直線補間軸: 構成軸1, 構成軸2 - 構成軸1の移動量: 100[pulse] - 構成軸2の移動量: 200[pulse] - 基準軸速度: 50[pulse/s] - 構成軸1の速度制限値: 55[pulse/s] 上記の場合、基準軸は構成軸1であり、構成軸1で設定した速度制限値で制御されます。また、各補間軸の位置決め速度および合成速度は下記となります。 - 構成軸1の位置決め速度: 50[pulse/s] - 構成軸2の位置決め速度: $200 / 100 \times 50 = 100[pulse/s]$ - 合成速度: $\sqrt{(50^2 + 100^2)} = 111.8[pulse/s]$ 構成軸2の位置決め速度と合成速度は、構成軸1の速度制限値「55」を超えた値となります。

制約事項

- 速度モード(VelocityMode)に「0: 合成速度(VectorSpeed)」を設定したとき、各軸の移動量が「4,294,967,296.0(=2³²)」を超える場合は、直線移動量範囲外(エラーコード: 1B10H)となり始動しません。
- バッファモード(BufferMode)を「0: Aborting(mcAborting)」に設定して、MCv_MoveLinearInterpolateAbsolute(絶対値直線補間制御)を多重起動する場合、速度モード(VelocityMode)には「0: 合成速度(VectorSpeed)」を設定してください。「0: 合成速度(VectorSpeed)」以外を設定した場合は、速度モード範囲外(エラーコード: 1AC9H)となります。

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

MCv_MoveLinearInterpolateRelative(相対値直線補間制御)で設定可能なバッファモードを下記に示します。

設定値	内容
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。
1: Buffered(mcBuffered)	制御中のFBに次のFBをバッファリングします。 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFB完了にて、バッファリングFBを順次実行します。
2: BlendingLow(mcBlendingLow)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBとバッファリングFBの目標速度のうち低い方の速度を切換え速度とします。
3: BlendingPrevious(mcBlendingPrevious)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBの目標速度を切換え速度とします。
4: BlendingNext(mcBlendingNext)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、バッファリングFBの目標速度を切換え速度とします。
5: BlendingHigh(mcBlendingHigh)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBとバッファリングFBの目標速度のうち高い方の速度を切換え速度とします。

*1 制御中のFBとバッファリングFB間で停止を行いません。

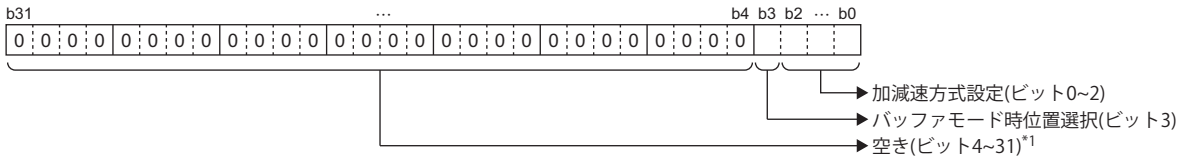
Point

多重起動(バッファモード)の詳細については、下記を参照してください。

 使用するコントローラのユーザーズマニュアル

■オプション(Options)

MCv_MoveLinearInterpolateRelative(相対値直線補間制御)で使用する機能オプションをビット指定で設定します。
ビット指定で設定する内容を下記に示します。



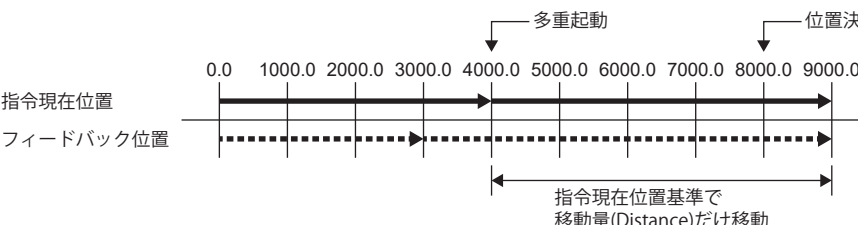
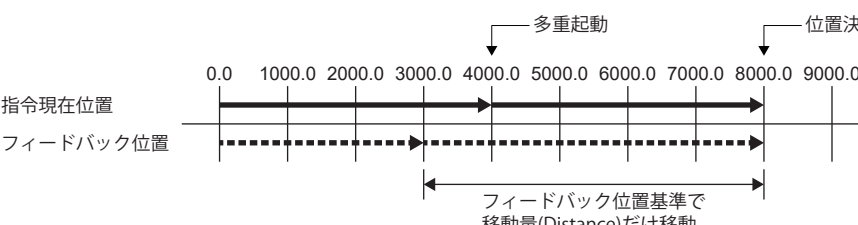
*1 空きには「0」を設定してください。「0」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

ビット	名称	内容
0~2	加減速方式設定	制御を行うための加減速方式を設定します。 0: 加減速度指定方式(mcAccDec) 1: 加減速時間一定方式(mcFixedTime)
3	バッファモード時位置選択	相対値位置決め制御を多重起動する場合の位置を設定します。 0: 指令現在位置 1: フィードバック位置 *: バッファモード(BufferMode)の「0: Aborting(mcAborting)」設定時に有効になります。

• 加減速方式設定(ビット0~2)

設定値	内容
0: 加減速度指定方式 (mcAccDec)	MCv_MoveLinearInterpolateRelative(相対値直線補間制御)で設定した加速度(Acceleration), 減速度(Deceleration), ジャーク(Jerk)を用いて, 加速/減速する方式です。
1: 加減速時間一定方式 (mcFixedTime)	速度に関係なく, MCv_MoveLinearInterpolateRelative(相対値直線補間制御)で設定した加減速時間を用いて, 加速/減速する方式です。 加減速時間は, 加速度(Acceleration)に設定し, 減速度(Deceleration), ジャーク(Jerk)は使用しません。

• バッファモード時位置選択(ビット3)

設定値	内容
0: 指令現在位置	指令現在位置からの相対位置制御です。 <例> 移動量(Distance)が「5000.0」、オプション(Options)が「00000000H(ビット3が0: 指令現在位置)」の設定で多重起動した場合 
1: フィードバック位置	フィードバック位置からの相対位置制御です。 <例> 移動量(Distance)が「5000.0」、オプション(Options)が「00000008H(ビット3が1: フィードバック位置)」の設定で多重起動した場合 

■必須オブジェクトデータ

MCv_MoveLinearInterpolateRelative(相対値直線補間制御)を使用する場合, 指定した軸グループの構成軸すべてに下記のオブジェクトデータを設定してください。

• Target position(607AH)

オブジェクトデータを設定していない構成軸がある場合, 必須オブジェクトデータ未設定(エラーコード: 1AF7H)となり始動しません。

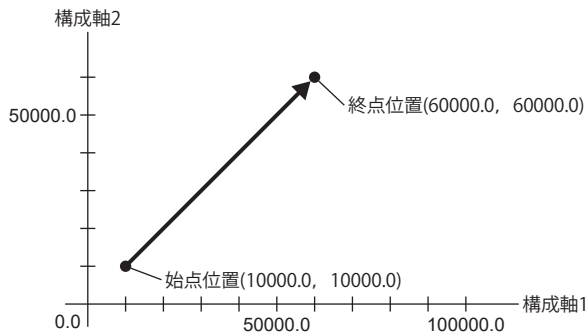
オブジェクトデータ設定に関する詳細は, 下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

プログラム例

直線補間制御始動(bLinearInterpolateCMD)をTRUEにし、軸グループ1(AxesGroup001)を有効にした後、下記の設定により軸グループ1(AxesGroup001)の2軸直線補間の相対値位置決め制御を行うプログラム例を下記に示します。

・動作



・軸グループ

項目	設定値
構成軸[1]	Axis0001
構成軸[2]	Axis0002

・設定

項目	設定値	
	構成軸1	構成軸2
目標位置	50000.0	50000.0
速度	50000.0	
加速度	50000.0	
減速度	50000.0	
ジャーク	0.0	

■軸

軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1
2	Axis0002	AXIS_REF	軸2

■軸グループ

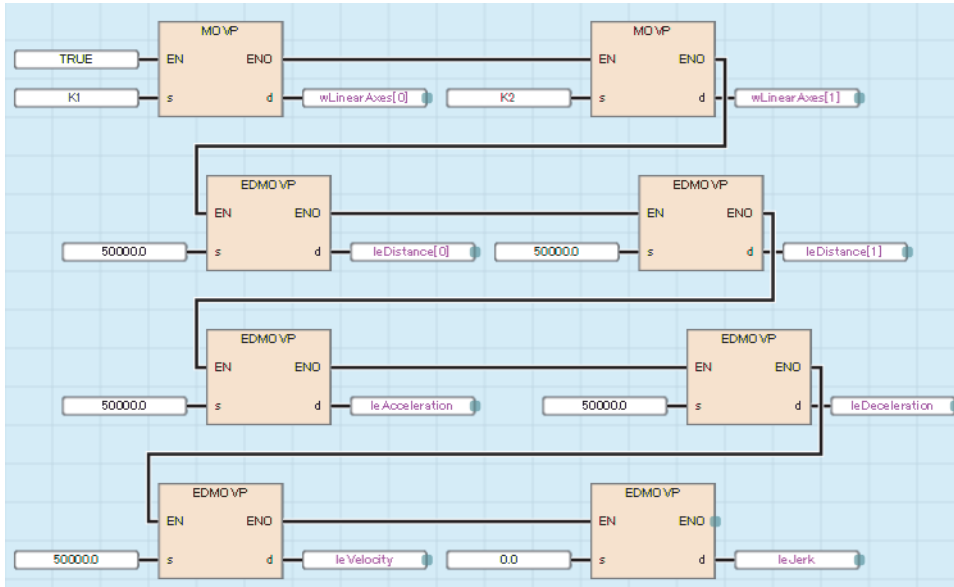
軸グループNo.	ラベル名	データ型	コメント
1	AxesGroup001	AXES_GROUP_REF	軸グループ1

■使用するラベル

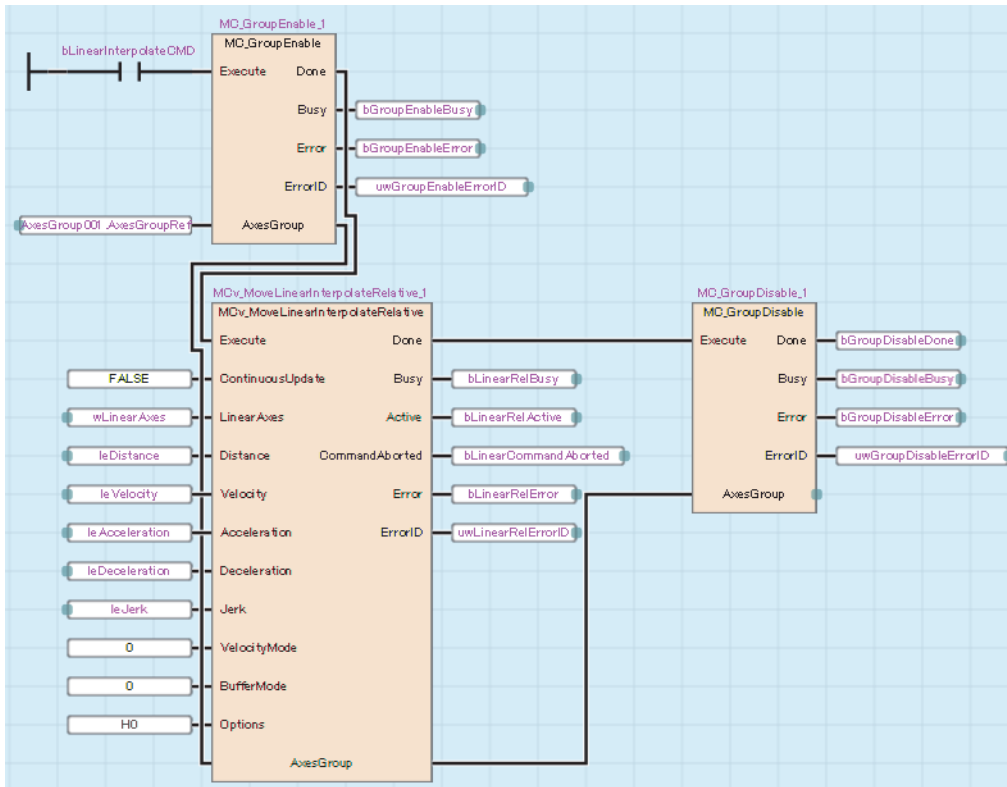
ラベル名	データ型	コメント
MCv_MoveLinearInterpolateRelative_1	MCv_MoveLinearInterpolateRelative	相対値直線補間制御FB
wLinearAxes	ワード[符号付き](0..15)	直線補間軸
leDistance	倍精度実数(0..15)	移動量
leVelocity	倍精度実数	速度
leAcceleration	倍精度実数	加速度
leDeceleration	倍精度実数	減速度
leJerk	倍精度実数	ジャーク
bLinearRelDone	ビット	実行完了
bLinearRelBusy	ビット	実行中
bLinearRelActive	ビット	制御中
bLinearCommandAborted	ビット	実行中断
bLinearRelError	ビット	エラー
uwLinearRelErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
MC_GroupEnable_1	MC_GroupEnable	軸グループ有効FB
bLinearInterpolateCMD	ビット	補間制御制御始動
bGroupEnableDone	ビット	軸グループ有効完了
bGroupEnableBusy	ビット	実行中
bGroupEnableError	ビット	エラー
uwGroupEnableErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
MC_GroupDisable_1	MC_GroupDisable	軸グループ無効FB
bGroupDisableDone	ビット	軸グループ無効完了
bGroupDisableBusy	ビット	実行中
bGroupDisableError	ビット	エラー
uwGroupDisableErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

・2軸直線補間制御用データ設定



・軸グループ有効/2軸直線補間制御/軸グループ無効



■STプログラム

//-----2軸直線補間制御用データ設定-----

```
wLinearAxes[0]:= 1;  
wLinearAxes[1]:= 2;  
leDistance[0]:= 50000.0;  
leDistance[1]:= 50000.0;  
leVelocity:= 50000.0;  
leAcceleration:= 50000.0;  
leDeceleration:= 50000.0;  
leJerk:= 0.0;
```

//-----軸グループ有効-----

```
MC_GroupEnable_1(  
  AxesGroup:= AxesGroup001.AxesGroupRef ,  
  Execute:= bLinearInterpolateCMD ,  
  Done=> bGroupEnableDone ,  
  Busy=> bGroupEnableBusy ,  
  Error=> bGroupEnableError ,  
  ErrorID=> uwGroupEnableErrorID  
);
```

//-----2軸直線補間制御-----

```
MCv_MoveLinearInterpolateRelative_1(  
  AxesGroup:= AxesGroup001.AxesGroupRef ,  
  Execute:= bGroupEnableDone ,  
  ContinuousUpdate:= FALSE ,  
  LinearAxes:= wLinearAxes ,  
  Distance:= leDistance ,  
  Velocity:= leVelocity ,  
  Acceleration:= leAcceleration ,  
  Deceleration:= leDeceleration ,  
  Jerk:= leJerk ,  
  VelocityMode:= MC_INTERPOLATE_SPEED_MODE__VectorSpeed ,  
  BufferMode:= MC_BUFFER_MODE__mcAborting ,  
  Options:= H00000000 ,  
  Done=> bLinearRelDone ,  
  Busy=> bLinearRelBusy ,  
  Active=> bLinearRelActive ,  
  CommandAborted=> bLinearRelCommandAborted ,  
  Error=> bLinearRelError ,  
  ErrorID=> uwLinearRelErrorID  
);
```

//-----軸グループ無効-----

```
MC_GroupDisable_1(  
  AxesGroup:= AxesGroup001.AxesGroupRef ,  
  Execute:= bLinearRelDone ,  
  Done=> bGroupDisableDone ,  
  Busy=> bGroupDisableBusy ,  
  Error=> bGroupDisableError ,  
  ErrorID=> uwGroupDisableErrorID  
);
```

42.12 絶対値円弧補間制御

42

MCv_MoveCircularInterpolateAbsolute

設定した軸グループの構成軸を用いて、絶対位置の終点、および補助点を設定し、2軸の円弧補間による位置決めを実行します。

ラダー	FBD/LD	ST
MCv_MoveCircularInterpolateAbsolute DUT: AxesGroup ——— AxesGroup :DUT B: Execute Done :B B: ContinuousUpdate Busy :B W: CircAxes Active :B ENUM: CircMode CommandAborted :B L: AuxPoint Error :B L: EndPoint ErrorID :UW ENUM: PathChoice L: Velocity L: Acceleration L: Deceleration L: Jerk L: CircularErrorTolerance ENUM: BufferMode UD: Options	MCv_MoveCircularInterpolateAbsolute Execute Done ContinuousUpdate Busy CircAxes Active CircMode CommandAborted AuxPoint Error EndPoint ErrorID PathChoice Velocity Acceleration Deceleration Jerk CircularErrorTolerance BufferMode Options AxesGroup	MCv_MoveCircularInterpolateAbsolute(AxesGroup:= ?AXES_GROUP_REF?, Execute:= ?BOOL?, ContinuousUpdate:= ?BOOL?, CircAxes:= ?INT(0..1)?, CircMode:= ?INT?, AuxPoint:= ?LREAL(0..15)?, EndPoint:= ?LREAL(0..15)?, PathChoice:= ?INT?, Velocity:= ?LREAL?, Acceleration:= ?LREAL?, Deceleration:= ?LREAL?, Jerk:= ?LREAL?, CircularErrorTolerance:= ?LREAL?, BufferMode:= ?INT?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
AxesGroup	軸グループ情報	AXES_GROUP_REF	起動時	—	省略不可	軸グループを設定します。 使用する変数 (AxesGroupName.AxesGroup Ref.)については、下記を参照 してください。 1402ページ AxesGroupName.AxesGroup Ref.(軸グループ情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	起動	BOOL	起動時	TRUE, FALSE	FALSE	TRUEで MCv_MoveCircularInterpolat eAbsolute(絶対値円弧補間制 御)を実行します。
Continuous Update	連続更新	BOOL	起動時	TRUE, FALSE	FALSE	連続して、速度(Velocity), 加速度(Acceleration), 減速 度(Deceleration)の変更を有 効にするか、無効にするかを 設定します。 連続更新により、実行中のイ ンスタンスに対し、動作を中 断することなく入力変数の再 取込みを行います。 • FALSE: 無効 • TRUE: 有効

入力変数	名称	データ型	取込	有効範囲	初期値	内容
CircAxes	円弧補間軸	INT[0..1]	起動時	1~16	0	円弧補間制御に使用する軸を構成軸から設定します。 構成軸のインデックス番号(1~16)を配列で設定します。 配列の第1要素が基準軸となります。 詳細は、下記を参照してください。 📖 1676ページ 円弧補間軸 (CircAxes)
CircMode	円弧補間モード	INT (MC_CIRC_MODE)	起動時	0~2	0	円弧補間の指定方法を設定します。 ・ 0: 境界点指定(mcBorder) ・ 1: 中心点指定(mcCenter) ・ 2: 半径指定(mcRadius) 詳細は、下記を参照してください。 📖 1676ページ 円弧補間モード (CircMode)
AuxPoint	補助点	LREAL[0..15]	起動時	■ 「0: 境界点指定(mcBorder)」, 「1: 中心点指定(mcCenter)」の場合: -100000000000.0~100000000000.0 ■ 「2: 半径指定(mcRadius)」の場合: 0.000001~2147483647.0	0.0	軸の単位に応じた補助点(境界点, 中心点, 半径)の絶対位置を設定します。 1次元の配列データです。円弧補間モード(CircMode)により下記となります。 ■ 「0: 境界点指定(mcBorder)」, 「1: 中心点指定(mcCenter)」の場合: 構成軸1~16の絶対位置として扱います。境界点, 中心点の絶対位置を設定します。 ■ 「2: 半径指定(mcRadius)」の場合: 第1要素を半径として扱い, 第2要素以降を無視します。 詳細は、下記を参照してください。 📖 1677ページ 補助点 (AuxPoint)
EndPoint	終点	LREAL[0..15]	起動時	-100000000000.0~100000000000.0	0.0	終点の位置を設定します。 1次元の配列データです。構成軸1~16の絶対位置として扱います。 詳細は、下記を参照してください。 📖 1677ページ 終点 (EndPoint)

入力変数	名称	データ型	取込	有効範囲	初期値	内容
PathChoice	経路選択	INT (MC_CIRC_PATHCHOICE)	起動時	0~5	0	円弧補間の回転方向を設定します。 • 0: CW(mcCW) • 1: CCW(mcCCW) • 2: 近回り(mcShortWay) • 3: 遠回り(mcLongWay) • 4: CW遠回り (mcCWLongWay) • 5: CCW遠回り (mcCCWLongWay) *: 円弧補間モード (CircMode)に「0: 境界点指定(mcBorder)」を設定した場合、入力を無視します。 *: 円弧補間モード (CircMode)に「2: 半径指定(mcRadius)」を設定した場合、「0: CW(mcCW)」はCW近回り、「1: CCW(mcCCW)」はCCW近回りを意味します。 詳細は、下記を参照してください。  1678ページ 経路選択 (PathChoice)
Velocity	速度	LREAL	起動時/再起動可/連続更新可	0.0, 0.0001~2500000000.0	0.0	速度指令値を設定します。 詳細は、下記を参照してください。  1680ページ 速度 (Velocity)
Acceleration	加速度	LREAL	起動時/再起動可/連続更新可	0.0000, 0.0001~2147483647.0	0.0	加速度を設定します。 詳細は、下記を参照してください。  1680ページ 加速度 (Acceleration)
Deceleration	減速度	LREAL	起動時/再起動可/連続更新可	0.0000, 0.0001~2147483647.0	0.0	減速度を設定します。 詳細は、下記を参照してください。  1680ページ 減速度 (Deceleration)
Jerk	ジャーク	LREAL	起動時	0.0000, 0.0001~2147483647.0	0.0	ジャークを設定します。 詳細は、下記を参照してください。  1680ページ ジャーク (Jerk)
CircularError Tolerance	円弧補間誤差許容値	LREAL	起動時	0.000001~100000.0	100.0	円弧補間誤差の許容範囲を設定します。 詳細は、下記を参照してください。  1680ページ 円弧補間誤差許容値 (CircularErrorTolerance)

入力変数	名称	データ型	取込	有効範囲	初期値	内容
BufferMode	バッファモード	INT (MC_BUFFER_MODE)	起動時	0~5	0	バッファモードを設定します。 • 0: Aborting(mcAborting) • 1: Buffered(mcBuffered) • 2: BlendingLow(mcBlendingLow) • 3: BlendingPrevious(mcBlendingPrevious) • 4: BlendingNext(mcBlendingNext) • 5: BlendingHigh(mcBlendingHigh) 詳細は、下記を参照してください。  1681ページ バッファモード(BufferMode)
Options	オプション	DWORD(HEX)	起動時	00000000H~00000001H	00000000H	MCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御)の機能オプションをビット指定で設定します。 詳細は、下記を参照してください。  1681ページ オプション(Options)

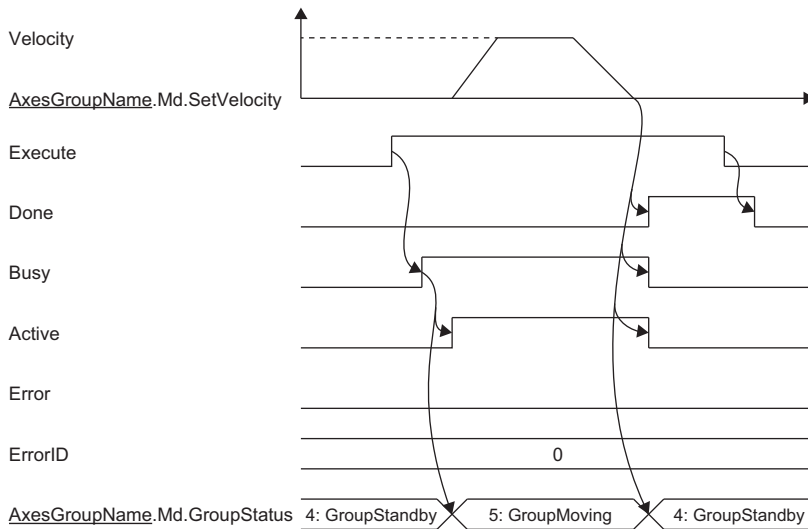
■出力変数

出力変数	名称	データ型	初期値	内容
Done	完了	BOOL	FALSE	制御が完了したときに、TRUEになります。 動作完了時に起動(Execute)の状態により下記となります。 ■起動(Execute)がTRUEの場合 起動(Execute)をFALSEにするまでTRUEのままになります。 ■起動(Execute)がFALSEの場合 1周期だけTRUEになります。
Busy	実行中	BOOL	FALSE	MCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御)を実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	MCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御)が軸を制御中のときに、TRUEになります。 同一軸グループに対して複数のMCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御)を実行した場合、制御中(Active)がTRUEになるのは1つのMCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御)のみです。
CommandAborted	実行中断	BOOL	FALSE	エラーや多重起動などにより、MCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御)の実行が中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。  使用するコントローラのユーザーズマニュアル

- 円弧補間制御は、軸グループを指定し、機械的に直交に配置した直線軸に対して、始点(移動開始点)から終点までの軌跡が円弧となるように補間制御を行います。
- 円弧補間制御では、軸グループに設定した構成軸より、任意の2軸を用いた補間制御を行います。
- 円弧補間制御では、境界点指定、中心点指定、半径指定の円弧補間方式を設定できます。
- MCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御)では、絶対位置の終点および補助点を指定して2軸の円弧補間制御を行います。

■タイミングチャート

・正常完了の場合



・異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■円弧補間軸(CircAxes)

円弧補間制御では、軸グループに設定した構成軸から、任意の2軸を使用して円弧補間を行います。

軸グループの構成軸の中より、円弧補間を行う軸を円弧補間軸(CircAxes)により設定します。円弧補間軸(CircAxes)は、2個の配列要素を持ちます。配列には、補間制御で使用する構成軸のインデックス番号(1~16: 構成軸1~16)を設定します。配列の第1要素が基準軸となります。

軸を設定していない構成軸を指定した場合、円弧補間軸未設定(エラーコード: 1ACAH)となります。

例

円弧補間軸に構成軸2, 3を設定する場合

```
CircAxes[0]:= 2;
```

```
CircAxes[1]:= 3;
```

■円弧補間モード(CircMode)

円弧補間制御を行うための円弧補間方式(境界点指定, 中心点指定, 半径指定)を設定します。

円弧補間モード(CircMode)により、補助点(AuxPoint)の設定内容が異なります。

設定値	補助点(AuxPoint)
0: 境界点指定(mcBorder)	補助点(AuxPoint)は、始点と終点を結ぶ円弧上の境界点を設定します。 円弧補間軸(CircAxes)で設定した2軸の構成軸を用いて、設定した境界点を通る円弧の軌跡で位置決め制御を行います。
1: 中心点指定(mcCenter)	補助点(AuxPoint)は、円弧の中心点を設定します。 円弧補間軸(CircAxes)で設定した2軸の構成軸を用いて、設定した中心点を中心とした円弧の軌跡で位置決め制御を行います。
2: 半径指定(mcRadius)	補助点(AuxPoint)は、円弧の半径を設定します。 円弧補間軸(CircAxes)で設定した2軸の構成軸を用いて、設定した半径を持つ円弧の軌跡で位置決め制御を行います。

設定した円弧補間モード(CircMode)による円弧補間制御の動作については、下記を参照してください。

📖 使用するコントローラのユーザーズマニュアル

■補助点(AuxPoint)

円弧補間を行う補助点(境界点, 中心点, 半径)の位置を設定します。

補助点(AuxPoint)は, 16個の配列要素を持ちます。補助点(AuxPoint)は, 円弧補間モード(CircMode)により設定内容が異なります。

円弧補間モード(CircMode)	設定範囲	補助点(AuxPoint)の設定
0: 境界点指定(mcBorder)	-100000000000.0 ≤ 設定値 < 100000000000.0 ^{*1}	補助点(AuxPoint)は, 始点と終点を結ぶ円弧上の境界点を設定します。 AuxPoint[0..15]は, 構成軸1~16の補助点位置を示します。 • 円弧補間軸に設定していない構成軸の補助点(AuxPoint)は無視します。 • 円弧補間軸に設定したすべての構成軸より後方の補助点(AuxPoint)は省略できます。
1: 中心点指定(mcCenter)		補助点(AuxPoint)は, 円弧の中心点を設定します。 AuxPoint[0..15]は, 構成軸1~16の補助点位置を示します。 • 円弧補間軸に設定していない構成軸の補助点(AuxPoint)は無視します。 • 円弧補間軸に設定したすべての構成軸より後方の補助点(AuxPoint)は省略できます。
2: 半径指定(mcRadius)	0.000001~2147483647.0	補助点(AuxPoint)は, 円弧の半径を設定します。 AuxPoint[0]は円弧の半径を示します。 • AuxPoint[1]以降の入力は無視するため省略できます。

*1 リングカウンタ有効の場合は, リングカウンタ範囲となります。

例

円弧補間軸に構成軸2, 3を設定し, 構成軸2, 3の補助点の絶対位置に2000.0, 3000.0を設定する場合

```
CircAxes[0]:= 2;
CircAxes[1]:= 3;
AuxPoint[0]:= 0.0;
AuxPoint[1]:= 2000.0;
AuxPoint[2]:= 3000.0;
AuxPoint[3]:= 0.0;*2
⋮
AuxPoint[15]:= 0.0;*2
```

*2 「AuxPoint[3]:= 0.0;~AuxPoint[15]:= 0.0;」は省略できます。

■終点(EndPoint)

絶対位置によるの終点の位置を設定します。

終点(EndPoint)は16個の配列要素を持ちます。EndPoint[0..15]は構成軸1~16の終点位置を示します。

- 円弧補間軸に設定していない構成軸の終点(EndPoint)は無視します。
- 円弧補間軸に設定したすべての構成軸より後方の終点(EndPoint)は省略できます。

終点(EndPoint)は下記範囲内で設定してください。

設定範囲
-100000000000.0 ≤ 設定値 < 100000000000.0 ^{*1}

*1 リングカウンタ有効の場合は, リングカウンタ範囲となります。

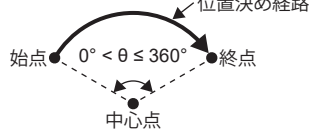
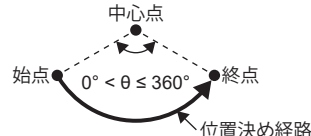
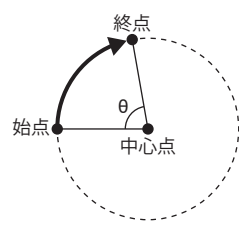
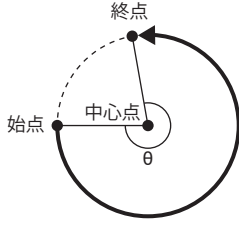
■経路選択(PathChoice)

円弧補間における、回転方向を設定します。回転方向は円弧補間軸(CircAxes)に設定した配列の第1要素を基準軸として考えます。

経路選択(PathChoice)は、円弧補間モード(CircMode)で「1: 中心点指定(mcCenter)」, または「2: 半径指定(mcRadius)」を設定した場合に設定します。「0: 境界点指定(mcBorder)」を設定した場合は、入力を見捨てします。

設定した円弧補間モード(CircMode)による経路選択(PathChoice)の設定値、設定内容、制御可能な円弧の中心角、経路を下記に示します。

・「1: 中心点指定(mcCenter)」の経路

設定値	設定内容	制御可能な円弧の中心角	経路
0: CW(mcCW)	CW	$0^\circ < \theta \leq 360^\circ$	
1: CCW(mcCCW)	CCW	$0^\circ < \theta \leq 360^\circ$	
2: 近回り (mcShortWay)	近回り ・ 始点、中心点、終点が一直線上にある場合、CW となります。(CW方向に半円を描きます。) ・ 始点=終点の場合、経路選択設定不正(エラーコード: 1B06H)となります。	$0^\circ < \theta \leq 180^\circ$	
3: 遠回り (mcLongWay)	遠回り ・ 始点、中心点、終点が一直線上にある場合、CW となります。(CW方向に半円を描きます。) ・ 始点=終点の場合、CWとなります。(CW方向に真 円を描きます。)	$180^\circ \leq \theta \leq 360^\circ$	

・「2: 半径指定(mcRadius)」の経路

設定値	設定内容	制御可能な円弧の中心角	経路
0: CW(mcCW)	CW(CW近回り) ・始点, 中心点, 終点が一直線上にある場合, 経路選択設定不正(エラーコード: 1B06H)となります。	$0^\circ < \theta < 180^\circ$	
1: CCW(mcCCW)	CCW(CCW近回り) ・始点, 中心点, 終点が一直線上にある場合, 経路選択設定不正(エラーコード: 1B06H)となります。	$0^\circ < \theta < 180^\circ$	
4: CW遠回り (mcCWLongWay)	CW遠回り ・始点, 中心点, 終点が一直線上にある場合, 円弧は半円を描きます。	$180^\circ \leq \theta < 360^\circ$	
5: CCW遠回り (mcCCWLongWay)	CCW遠回り ・始点, 中心点, 終点が一直線上にある場合, 円弧は半円を描きます。	$180^\circ \leq \theta < 360^\circ$	

■速度(Velocity)

MCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御)での経路の最大速度を設定します。

設定範囲
0.0, 0.0001~2500000000.0 ^{*1*2}

- *1 浮動小数点演算を行うため、指令速度の下限値には下記の制約が生じます。
指令速度を演算周期換算した速度が「0.00001」未満となる場合、演算周期換算速度範囲外(エラーコード: 1B16H)(速度変更時は演算周期換算速度範囲外警告(イベントコード: 00D2FH))となります。浮動小数点演算の精度を向上させるためには位置指令単位(AxisName.Pr.Unit_Position), または速度指令単位(AxisName.Pr.Unit_Velocity)を変更することで演算周期換算後の速度が「0.00001」未満とならないように設定してください。
- *2 多重起動を行うモーション制御FBにて指定速度が「0.0」の場合、直前のモーション制御FBの指定速度となります。

■加速度(Acceleration)

MCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御)での加速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の設定により設定範囲が異なります。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	0.000000 ^{*1} , 0.000001~8400.0[s]までの正数

- *1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更, 減速度変更時は, 変更を受け付けません。

■減速度(Deceleration)

MCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御)での減速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合, 設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合, 減速度(Deceleration)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

- *1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更, 減速度変更時は, 変更を受け付けません。

■ジャーク(Jerk)

MCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御)でのジャークを設定します。

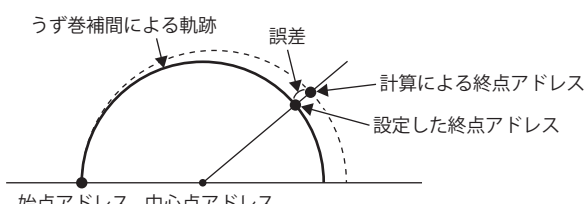
加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合, 設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合, ジャーク(Jerk)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000, 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

■円弧補間誤差許容値(CircularErrorTolerance)

中心点指定の円弧補間制御では, 始点位置と中心点位置から算出した円弧の軌跡と, 終点(EndPoint)に設定した終点位置がずれる場合があります。

円弧補間誤差許容値(CircularErrorTolerance)は, 算出した円弧の軌跡と終点位置の誤差の許容範囲を設定します。

許容範囲	内容
算出した誤差 ≤ 円弧補間誤差許容値(CircularErrorTolerance)	うず巻補間により誤差補正を行いながら, 設定している終点アドレスに円弧補間を行います。  うず巻補間による軌跡 誤差 計算による終点アドレス 設定した終点アドレス 始点アドレス 中心点アドレス
算出した誤差 > 円弧補間誤差許容値(CircularErrorTolerance)	位置決め始動時に円弧補間誤差許容値オーバー(エラーコード: 1AD9H)となり始動しません。位置決め制御中の場合は, エラー検出時に即停止します。

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

MCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御)設定可能なバッファモードを下記に示します。

設定値	内容
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。
1: Buffered(mcBuffered)	制御中のFBに次のFBをバッファリングします。 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFB完了にて、バッファリングFBを順次実行します。
2: BlendingLow(mcBlendingLow)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBとバッファリングFBの目標速度のうち低い方の速度を切換え速度とします。
3: BlendingPrevious(mcBlendingPrevious)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBの目標速度を切換え速度とします。
4: BlendingNext(mcBlendingNext)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、バッファリングFBの目標速度を切換え速度とします。
5: BlendingHigh(mcBlendingHigh)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBとバッファリングFBの目標速度のうち高い方の速度を切換え速度とします。

*1 制御中のFBとバッファリングFB間で停止を行いません。

Point

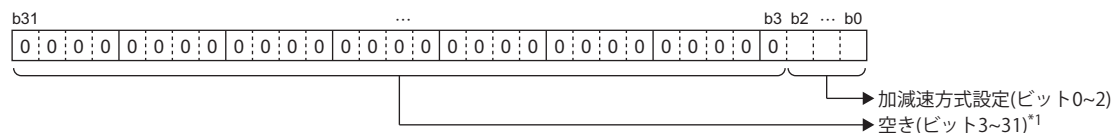
多重起動(バッファモード)の詳細については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

■オプション(Options)

MCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御)で使用する機能オプションをビット指定で設定します。

ビット指定で設定する内容を下記に示します。



*1 空きには「0」を設定してください。「0」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

ビット	名称	内容
0~2	加減速方式設定	制御を行うための加減速方式を設定します。 • 0: 加減速度指定方式(mcAccDec) • 1: 加減速時間一定方式(mcFixedTime)

・加減速方式設定(ビット0~2)

設定値	内容
0: 加減速度指定方式(mcAccDec)	MCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御)で設定した加速度(Acceleration), 減速度(Deceleration), ジャーク(Jerk)を用いて、加速/減速する方式です。
1: 加減速時間一定方式(mcFixedTime)	速度に関係なく、MCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御)で設定した加減速時間を用いて、加速/減速する方式です。 加減速時間は、加速度(Acceleration)に設定し、減速度(Deceleration), ジャーク(Jerk)は使用しません。

■必須オブジェクトデータ

MCv_MoveCircularInterpolateAbsolute(絶対値円弧補間制御)を使用する場合、指定した軸グループの構成軸すべてに下記のオブジェクトデータを設定してください。

- Target position(607AH)

オブジェクトデータを設定していない構成軸がある場合、必須オブジェクトデータ未設定(エラーコード: 1AF7H)となり始動しません。

オブジェクトデータ設定に関する詳細は、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

注意事項

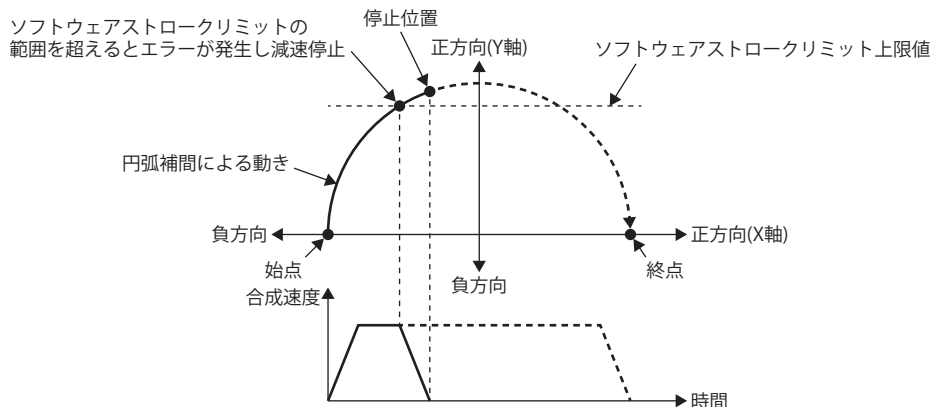
- ストロークリミットを無効に設定した軸を含む円弧補間は行えません。円弧補間時ソフトウェアストロークリミット無効(エラーコード: 1ADAH)となり始動しません。
- 補間動作中に補間経路がストロークリミット範囲を超えた場合は、ソフトウェアストロークリミットオーバ(正方向)(エラーコード: 1A83H), またはソフトウェアストロークリミットオーバ(負方向)(エラーコード: 1A84H)となり運転を停止します。

例

Y軸の正方向でソフトウェアストロークリミットの上限を超えた場合

即停止する場合は、エラー発生と同時に停止します。

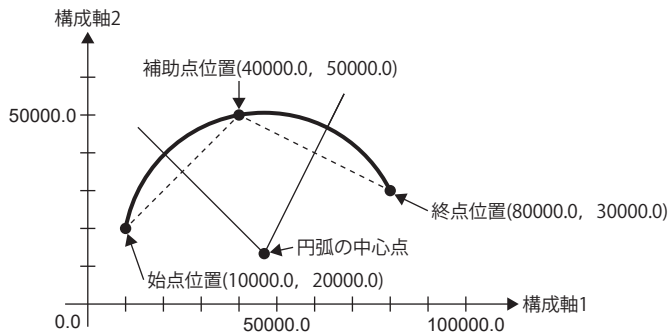
減速停止する場合は、下図のように円弧の軌跡に沿って減速し、停止します。



プログラム例

円弧補間制御始動(bCircularInterpolateCMD)をTRUEにし、軸グループ1(AxesGroup001)を有効にした後、下記の設定により軸グループ1(AxesGroup001)の2軸円弧補間(補助点指定)の絶対値位置決め制御を行うプログラム例を下記に示します。

・動作



・軸グループ

項目	設定値
構成軸[1]	Axis0001
構成軸[2]	Axis0002

・設定

項目	設定値	
	構成軸1	構成軸2
補助点	40000.0	50000.0
終点	80000.0	30000.0
速度	50000.0	
加速度	50000.0	
減速度	50000.0	
ジャーク	0.0	
円弧補間誤差許容値	100.0	

■軸

軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1
2	Axis0002	AXIS_REF	軸2

■軸グループ

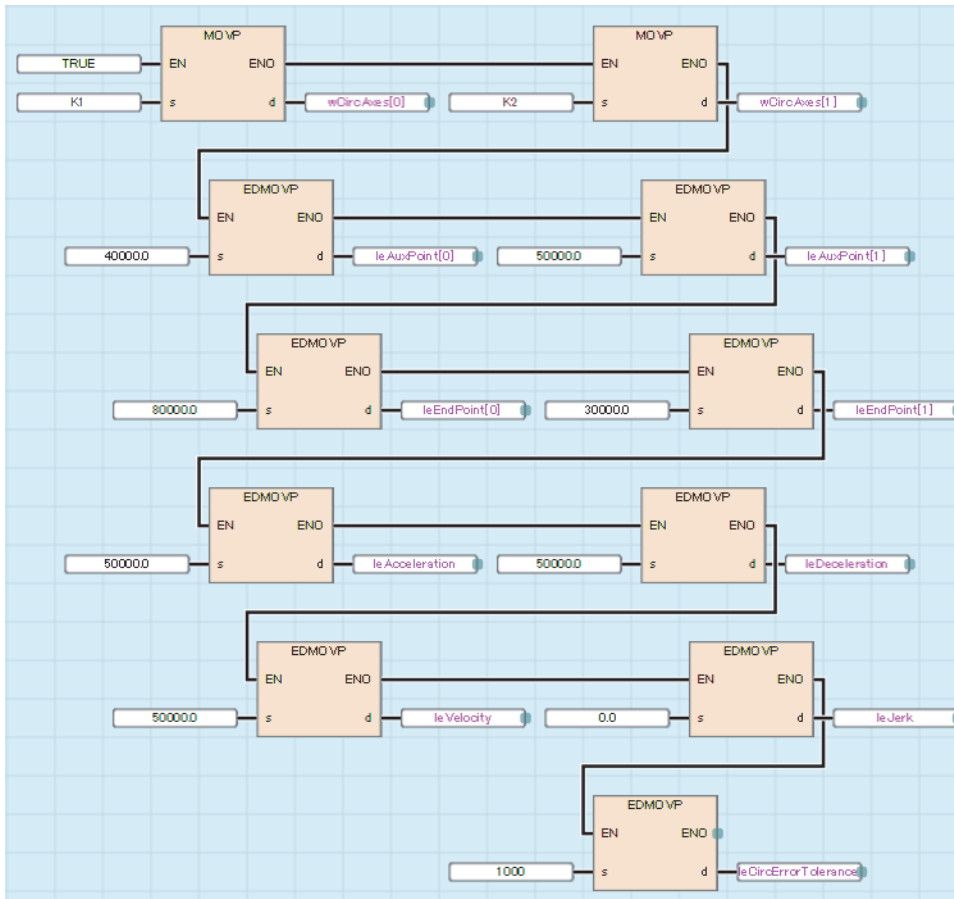
軸グループNo.	ラベル名	データ型	コメント
1	AxesGroup001	AXES_GROUP_REF	軸グループ1

■使用するラベル

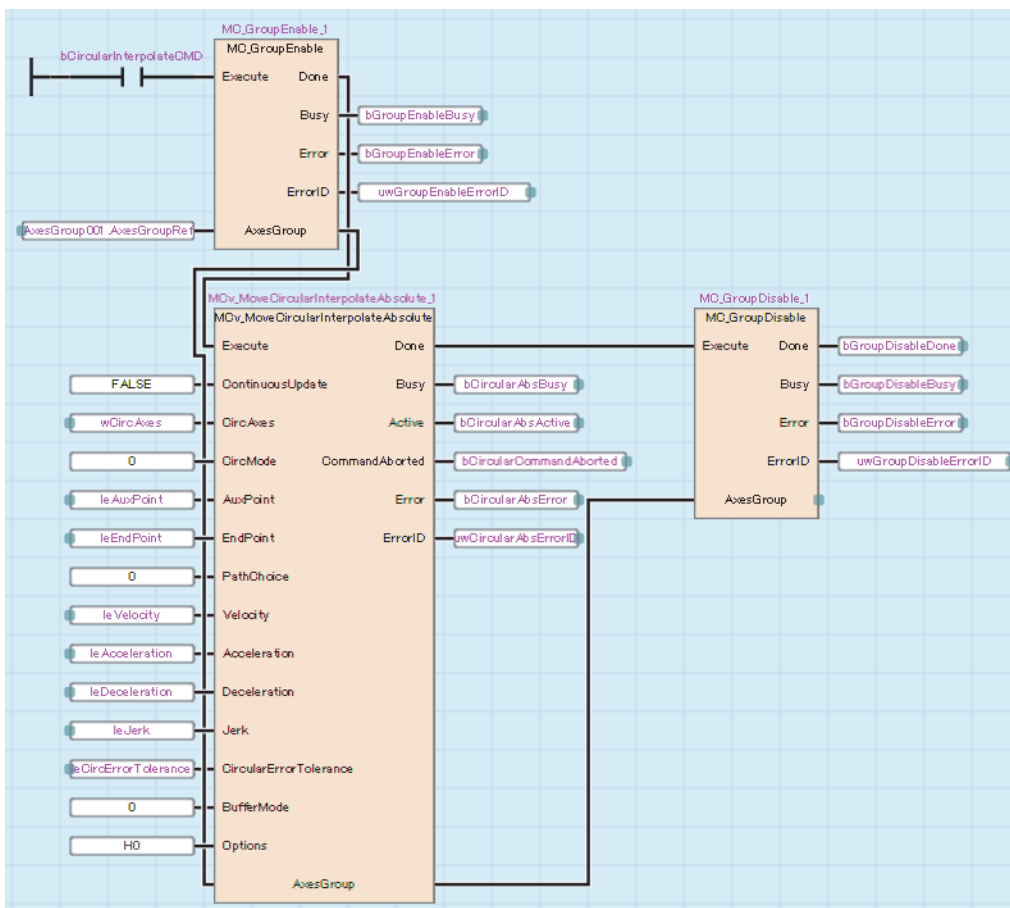
ラベル名	データ型	コメント
MCv_MoveCircularInterpolateAbsolute_1	MCv_MoveCircularInterpolateAbsolute	絶対値円弧補間制御FB
wCircAxes	ワード[符号付き](0..1)	円弧補間軸
leAuxPoint	倍精度実数(0..15)	補助点
leEndPoint	倍精度実数(0..15)	終点
leVelocity	倍精度実数	速度
leAcceleration	倍精度実数	加速度
leDeceleration	倍精度実数	減速度
leJerk	倍精度実数	ジャーク
leCircErrorTolerance	倍精度実数	円弧補間誤差許容値
bCircularAbsDone	ビット	実行完了
bCircularAbsBusy	ビット	実行中
bCircularAbsActive	ビット	制御中
bCircularCommandAborted	ビット	実行中断
bCircularAbsError	ビット	エラー
uwCircularAbsErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
MC_GroupEnable_1	MC_GroupEnable	軸グループ有効FB
bCircularInterpolateCMD	ビット	円弧補間制御始動
bGroupEnableDone	ビット	軸グループ有効完了
bGroupEnableBusy	ビット	実行中
bGroupEnableError	ビット	エラー
uwGroupEnableErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
MC_GroupDisable_1	MC_GroupDisable	軸グループ無効FB
bGroupDisableDone	ビット	軸グループ無効完了
bGroupDisableBusy	ビット	実行中
bGroupDisableError	ビット	エラー
uwGroupDisableErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

・2軸円弧補間制御用データ設定



・軸グループ有効/円弧補間制御(境界点指定)/軸グループ無効



■STプログラム

//-----2軸円弧補間制御用データ設定-----

```
wCircAxes[0]:= 1;  
wCircAxes[1]:= 2;  
leAuxPoint[0]:= 40000.0;  
leAuxPoint[1]:= 50000.0;  
leEndPoint[0]:= 80000.0;  
leEndPoint[1]:= 30000.0;  
leVelocity:= 50000.0;  
leAcceleration:= 50000.0;  
leDeceleration:= 50000.0;  
leJerk:= 0.0;  
leCircularErrorTolerance:= 100.0;
```

//-----軸グループ有効-----

```
MC_GroupEnable_1(  
  AxesGroup:= AxesGroup001.AxesGroupRef ,  
  Execute:= bCircularInterpolateCMD ,  
  Done=> bGroupEnableDone ,  
  Busy=> bGroupEnableBusy ,  
  Error=> bGroupEnableError ,  
  ErrorID=> uwGroupEnableErrorID  
);
```

//-----円弧補間制御(境界点指定)-----

```
MCv_MoveCircularInterpolateAbsolute_1(  
  AxesGroup:= AxesGroup001.AxesGroupRef ,  
  Execute:= bGroupEnableDone ,  
  ContinuousUpdate:= FALSE ,  
  CircAxes:= wCircAxes ,  
  CircMode:= MC_CIRC_MODE__mcBorder ,  
  AuxPoint:= leAuxPoint ,  
  EndPoint:= leEndPoint ,  
  PathChoice:= MC_CIRC_PATHCHOICE__mcCW ,  
  Velocity:= leVelocity ,  
  Acceleration:= leAcceleration ,  
  Deceleration:= leDeceleration ,  
  Jerk:= leJerk ,  
  CircularErrorTolerance:= leCircularErrorTolerance ,  
  BufferMode:= MC_BUFFER_MODE__mcAborting ,  
  Options:= H00000000 ,  
  Done=> bCircularAbsDone ,  
  Busy=> bCircularAbsBusy ,  
  Active=> bCircularAbsActive ,  
  CommandAborted=> bCircularAbsCommandAborted ,  
  Error=> bCircuAbsError ,  
  ErrorID=> uwCircularAbsErrorID  
);
```

```
//-----軸グループ無効-----  
MC_GroupDisable_1(  
  AxesGroup:= AxesGroup001.AxesGroupRef ,  
  Execute:= bCircularAbsDone ,  
  Done=> bGroupDisableDone ,  
  Busy=> bGroupDisableBusy ,  
  Error=> bGroupDisableError ,  
  ErrorID=> uwGroupDisableErrorID  
);
```

42.13 相対値円弧補間制御

MCv_MoveCircularInterpolateRelative

設定した軸グループの構成軸を用いて、始動時の現在位置から終点、および補助点へ相対位置を設定し、2軸の円弧補間による位置決めを実行します。

ラダー	FBD/LD	ST
		<pre> MCv_MoveCircularInterpolateRelative(AxesGroup:= ?AXES_GROUP_REF?, Execute:= ?BOOL?, ContinuousUpdate:= ?BOOL?, CircAxes:= ?INT(0..1)?, CircMode:= ?INT?, AuxPoint:= ?LREAL(0..15)?, EndPoint:= ?LREAL(0..15)?, PathChoice:= ?INT?, Velocity:= ?LREAL?, Acceleration:= ?LREAL?, Deceleration:= ?LREAL?, Jerk:= ?LREAL?, CircularErrorTolerance:= ?LREAL?, BufferMode:= ?INT?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?); </pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
AxesGroup	軸グループ情報	AXES_GROUP_REF	起動時	—	省略不可	軸グループを設定します。 使用する変数 (AxesGroupName.AxesGroupRef.) については、下記を参照してください。 1402ページ AxesGroupName.AxesGroupRef.(軸グループ情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	起動	BOOL	起動時	TRUE, FALSE	FALSE	TRUEで MCv_MoveCircularInterpolateRelative(相対値円弧補間制御)を実行します。
ContinuousUpdate	連続更新	BOOL	起動時	TRUE, FALSE	FALSE	連続して、速度(Velocity)、加速度(Acceleration)、減速度(Deceleration)の変更を有効にするか、無効にするかを設定します。 連続更新により、実行中のインスタンスに対し、動作を中断することなく入力変数の再取込みを行います。 • FALSE: 無効 • TRUE: 有効

入力変数	名称	データ型	取込	有効範囲	初期値	内容
CircAxes	円弧補間軸	INT[0..1]	起動時	1~16	0	円弧補間制御に使用する軸を構成軸から設定します。 構成軸のインデックス番号(1~16)を配列で設定します。配列の第1要素が基準軸となります。 詳細は、下記を参照してください。 ☞ 1691ページ 円弧補間軸 (CircAxes)
CircMode	円弧補間モード	INT (MC_CIRC_MODE)	起動時	0~2	0	円弧補間の指定方法を設定します。 ・ 0: 境界点指定(mcBorder) ・ 1: 中心点指定(mcCenter) ・ 2: 半径指定(mcRadius) 詳細は、下記を参照してください。 ☞ 1691ページ 円弧補間モード (CircMode)
AuxPoint	補助点	LREAL[0..15]	起動時	■ 「0: 境界点指定(mcBorder)」, 「1: 中心点指定(mcCenter)」の場合: - 10000000000.0~10000000000.0 ■ 「2: 半径指定(mcRadius)」の場合: 0.000001~2147483647.0	0.0	始動時の現在位置から補助点(境界点, 中心点, 半径)の相対位置を設定します。 1次元の配列データです。円弧補間モード(CircMode)により下記となります。 ■ 「0: 境界点指定(mcBorder)」, 「1: 中心点指定(mcCenter)」の場合: 構成軸1~16の相対位置として扱います。境界点, 中心点の絶対位置を設定します。 ■ 「2: 半径指定(mcRadius)」の場合: 第1要素を半径として扱い, 第2要素以降を無視します。 詳細は、下記を参照してください。 ☞ 1692ページ 補助点(AuxPoint)
EndPoint	終点	LREAL[0..15]	起動時	- 10000000000.0~10000000000.0	0.0	始動時の現在位置から終点までの移動量を設定します。 1次元の配列データです。構成軸1~16の絶対位置として扱います。 詳細は、下記を参照してください。 ☞ 1692ページ 終点(EndPoint)
PathChoice	経路選択	INT (MC_CIRC_PATHCHOICE)	起動時	0~5	0	円弧補間の回転方向を設定します。 ・ 0: CW(mcCW) ・ 1: CCW(mcCCW) ・ 2: 近回り(mcShortWay) ・ 3: 遠回り(mcLongWay) ・ 4: CW遠回り(mcCWLongWay) ・ 5: CCW遠回り(mcCCWLongWay) *: 円弧補間モード(CircMode)に「0: 境界点指定(mcBorder)」を設定した場合, 入力が無視します。 *: 円弧補間モード(CircMode)に「2: 半径指定(mcRadius)」を設定した場合, 「0: CW(mcCW)」はCW近回り, 「1: CCW(mcCCW)」はCCW近回りを意味します。 詳細は、下記を参照してください。 ☞ 1693ページ 経路選択 (PathChoice)
Velocity	速度	LREAL	起動時/再起動可/連続更新可	0.0, 0.0001~2500000000.0	0.0	速度指令値を設定します。 詳細は、下記を参照してください。 ☞ 1695ページ 速度(Velocity)
Acceleration	加速度	LREAL	起動時/再起動可/連続更新可	0.0000, 0.0001~2147483647.0	0.0	加速度を設定します。 詳細は、下記を参照してください。 ☞ 1695ページ 加速度 (Acceleration)
Deceleration	減速度	LREAL	起動時/再起動可/連続更新可	0.0000, 0.0001~2147483647.0	0.0	減速度を設定します。 詳細は、下記を参照してください。 ☞ 1695ページ 減速度 (Deceleration)

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Jerk	ジャーク	LREAL	起動時	0.0000, 0.0001~2147483647.0	0.0	ジャークを設定します。 詳細は、下記を参照してください。 1695ページ ジャーク(Jerk)
CircularError Tolerance	円弧補間誤差許容値	LREAL	起動時	0.000001~100000.0	100.0	円弧補間誤差の許容範囲を設定します。 詳細は、下記を参照してください。 1695ページ 円弧補間誤差許容値(CircularErrorTolerance)
BufferMode	バッファモード	INT (MC_BUFFER_MODE)	起動時	0~5	0	バッファモードを設定します。 0: Aborting(mcAborting) 1: Buffered(mcBuffered) 2: BlendingLow(mcBlendingLow) 3: BlendingPrevious(mcBlendingPrevious) 4: BlendingNext(mcBlendingNext) 5: BlendingHigh(mcBlendingHigh) 詳細は、下記を参照してください。 1696ページ バッファモード(BufferMode)
Options	オプション	DWORD(HEX)	起動時	00000000H~00000005H	00000000H	MCv_MoveCircularInterpolateRelative(相対値円弧補間制御)の機能オプションをビット指定で設定します。 詳細は、下記を参照してください。 1696ページ オプション(Options)

■出力変数

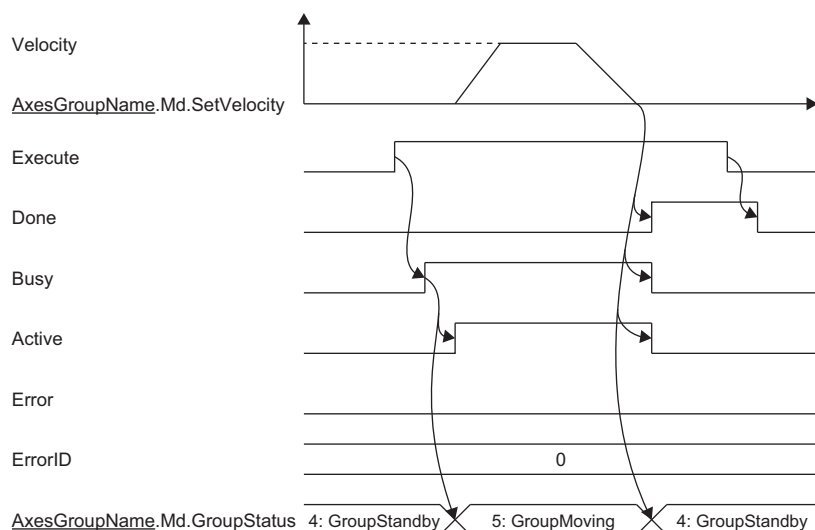
出力変数	名称	データ型	初期値	内容
Done	完了	BOOL	FALSE	制御が完了したときに、TRUEになります。 動作完了時に起動(Execute)の状態により下記となります。 ■起動(Execute)がTRUEの場合 起動(Execute)をFALSEにするまでTRUEのままになります。 ■起動(Execute)がFALSEの場合 1周期だけTRUEになります。
Busy	実行中	BOOL	FALSE	MCv_MoveCircularInterpolateRelative(相対値円弧補間制御)を実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	MCv_MoveCircularInterpolateRelative(相対値円弧補間制御)が軸を制御中のときに、TRUEになります。 同一軸グループに対して複数のMCv_MoveCircularInterpolateRelative(相対値円弧補間制御)を実行した場合、制御中(Active)がTRUEになるのは1つのMCv_MoveCircularInterpolateRelative(相対値円弧補間制御)のみです。
CommandAborted	実行中断	BOOL	FALSE	エラーや多重起動などにより、MCv_MoveCircularInterpolateRelative(相対値円弧補間制御)の実行が中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 1696ページ 使用するコントローラのユーザーズマニュアル

機能

- ・円弧補間制御は、軸グループを指定し、機械的に直交に配置した直線軸に対して、始点(移動開始点)から終点までの軌跡が円弧となるように補間制御を行います。
- ・円弧補間制御では、軸グループに設定した構成軸より、任意の2軸を用いた補間制御を行います。
- ・円弧補間制御では、境界点指定、中心点指定、半径指定の円弧補間方式を設定できます。
- ・MCv_MoveCircularInterpolateRelative(相対値円弧補間制御)では、現在位置から終点、および補助点の移動量を指定して2軸の円弧補間制御を行います。

■タイミングチャート

- ・正常完了の場合



- ・異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■円弧補間軸(CircAxes)

円弧補間制御では、軸グループに設定した構成軸から、任意の2軸を使用して円弧補間を行います。

軸グループの構成軸の中より、円弧補間を行う軸を円弧補間軸(CircAxes)により設定します。円弧補間軸(CircAxes)は、2個の配列要素を持ちます。配列には、補間制御で使用する構成軸のインデックス番号(1~16: 構成軸1~16)を設定します。配列の第1要素が基準軸となります。

軸を設定していない構成軸を設定した場合、円弧補間軸未設定(エラーコード: 1ACAH)となります。

例

円弧補間軸に構成軸2, 3を設定する場合

CircAxes[0]:= 2;

CircAxes[1]:= 3;

■円弧補間モード(CircMode)

円弧補間制御を行うための円弧補間方式(境界点指定, 中心点指定, 半径指定)を設定します。

円弧補間モード(CircMode)により、補助点(AuxPoint)の設定内容が異なります。

設定値	補助点(AuxPoint)
0: 境界点指定(mcBorder)	補助点(AuxPoint)は、始点と終点を結ぶ円弧上の境界点を設定します。 円弧補間軸(CircAxes)で設定した2軸の構成軸を用いて、設定した境界点を通る円弧の軌跡で位置決め制御を行います。
1: 中心点指定(mcCenter)	補助点(AuxPoint)は、円弧の中心点を設定します。 円弧補間軸(CircAxes)で設定した2軸の構成軸を用いて、設定した中心点を中心とした円弧の軌跡で位置決め制御を行います。
2: 半径指定(mcRadius)	補助点(AuxPoint)は、円弧の半径を設定します。 円弧補間軸(CircAxes)で設定した2軸の構成軸を用いて、設定した半径を持つ円弧の軌跡で位置決め制御を行います。

設定した円弧補間モード(CircMode)による円弧補間制御の動作については、下記を参照してください。

📖 使用するコントローラのユーザーズマニュアル

■補助点(AuxPoint)

円弧補間を行う補助点(境界点, 中心点, 半径)の位置を設定します。

補助点(AuxPoint)は, 16個の配列要素を持ちます。補助点(AuxPoint)は, 円弧補間モード(CircMode)により設定内容が異なります。

円弧補間モード(CircMode)	設定範囲	補助点(AuxPoint)の設定
0: 境界点指定(mcBorder)	-10000000000.0~10000000000.0	補助点(AuxPoint)は, 始点と終点を結ぶ円弧上の境界点を設定します。 AuxPoint[0..15]は, 構成軸1~16の補助点位置を示します。 - 円弧補間軸に設定していない構成軸の補助点(AuxPoint)は無視します。 - 円弧補間軸に設定したすべての構成軸より後方の補助点(AuxPoint)は省略できます。
1: 中心点指定(mcCenter)		補助点(AuxPoint)は, 円弧の中心点を設定します。 AuxPoint[0..15]は, 構成軸1~16の補助点位置を示します。 - 円弧補間軸に設定していない構成軸の補助点(AuxPoint)は無視します。 - 円弧補間軸に設定したすべての構成軸より後方の補助点(AuxPoint)は省略できます。
2: 半径指定(mcRadius)	0.000001~2147483647.0	補助点(AuxPoint)は, 円弧の半径を設定します。 AuxPoint[0]は円弧の半径を示します。 AuxPoint[1]以降の入力は無視するため省略できます。

例

円弧補間軸に構成軸2, 3を設定し, 構成軸2, 3の補助点の相対位置に2000.0, 3000.0を設定する場合

```
CircAxes[0]:= 2;
```

```
CircAxes[1]:= 3;
```

```
AuxPoint[0]:= 0.0;
```

```
AuxPoint[1]:= 2000.0;
```

```
AuxPoint[2]:= 3000.0;
```

```
AuxPoint[3]:= 0.0; *1
```

```
⋮
```

```
AuxPoint[15]:= 0.0; *1
```

*1 「AuxPoint[3]:= 0.0;~AuxPoint[15]:= 0.0;」は省略できます。

■終点(EndPoint)

始動時の現在位置から終点までの移動量を設定します。

終点(EndPoint)は16個の配列要素を持ちます。EndPoint[0..15]は構成軸1~16の終点位置を示します。

- 円弧補間軸に設定していない構成軸の終点(EndPoint)は無視します。
- 円弧補間軸に設定したすべての構成軸より後方の終点(EndPoint)は省略できます。

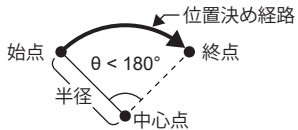
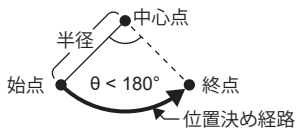
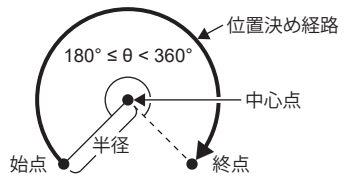
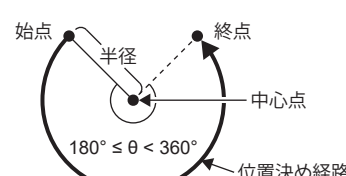
■経路選択(PathChoice)

円弧補間における、回転方向を設定します。回転方向はCircAxes(円弧補間軸)に設定した配列の第1要素を基準軸として考えます。経路選択(PathChoice)は、円弧補間モード(CircMode)で「1: 中心点指定(mcCenter)」, または「2: 半径指定(mcRadius)」を設定した場合に設定します。「0: 境界点指定(mcBorder)」を設定した場合は、入力を見捨てします。設定した円弧補間モード(CircMode)による経路選択(PathChoice)の設定値、設定内容、制御可能な円弧の中心角、経路を下記に示します。

・「1: 中心点指定(mcCenter)」の経路

設定値	設定内容	制御可能な円弧の中心角	経路
0: CW(mcCW)	CW	$0^\circ < \theta \leq 360^\circ$	
1: CCW(mcCCW)	CCW	$0^\circ < \theta \leq 360^\circ$	
2: 近回り (mcShortWay)	近回り ・ 始点, 中心点, 終点が一直線上にある場合, CW となります。(CW方向に半円を描きます。) ・ 始点 = 終点の場合, 経路選択設定不正(エラー コード: 1B06H)となります。	$0^\circ < \theta \leq 180^\circ$	
3: 遠回り (mcLongWay)	遠回り ・ 始点, 中心点, 終点が一直線上にある場合, CW となります。(CW方向に半円を描きます。) ・ 始点 = 終点の場合, CWとなります。(CW方向に 真円を描きます。)	$180^\circ \leq \theta \leq 360^\circ$	

• 「2: 半径指定(mcRadius)」の経路

設定値	設定内容	制御可能な円弧の中心角	経路
0: CW(mcCW)	CW ・ 始点, 中心点, 終点が一直線上にある場合, 経路選択設定不正(エラーコード: 1B06H)となります。	$0^\circ < \theta < 180^\circ$	
1: CCW(mcCCW)	CCW ・ 始点, 中心点, 終点が一直線上にある場合, 経路選択設定不正(エラーコード: 1B06H)となります。	$0^\circ < \theta < 180^\circ$	
4: CW遠回り (mcCWLongWay)	CW遠回り ・ 始点, 中心点, 終点が一直線上にある場合, 円弧は半円を描きます。	$180^\circ \leq \theta < 360^\circ$	
5: CCW遠回り (mcCCWLongWay)	CCW遠回り ・ 始点, 中心点, 終点が一直線上にある場合, 円弧は半円を描きます。	$180^\circ \leq \theta < 360^\circ$	

■速度(Velocity)

MCv_MoveCircularInterpolateRelative(相対値円弧補間制御)での経路の最大速度を設定します。

設定範囲
0.0, 0.0001~2500000000.0 ^{*1*2}

- *1 浮動小数点演算を行うため、指令速度の下限値には下記の制約が生じます。
指令速度を演算周期換算した速度が「0.00001」未満となる場合、演算周期換算速度範囲外(エラーコード: 1B16H)(速度変更時は演算周期換算速度範囲外警告(イベントコード: 00D2FH))となります。浮動小数点演算の精度を向上させるためには位置指令単位(AxisName.Pr.Unit_Position)、または速度指令単位(AxisName.Pr.Unit_Velocity)を変更することで演算周期換算後の速度が「0.00001」未満とならないように設定してください。
- *2 多重起動を行うモーション制御FBにて指定速度が「0.0」の場合、直前のモーション制御FBの指定速度となります。

■加速度(Acceleration)

MCv_MoveCircularInterpolateRelative(相対値円弧補間制御)での加速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の設定により設定範囲が異なります。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	0.000000 ^{*1} , 0.000001~8400.0[s]までの正数

- *1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更、減速度変更時は、変更を受け付けません。

■減速度(Deceleration)

MCv_MoveCircularInterpolateRelative(相対値円弧補間制御)での減速度を設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合、減速度(Deceleration)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

- *1 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更、減速度変更時は、変更を受け付けません。

■ジャーク(Jerk)

MCv_MoveCircularInterpolateRelative(相対値円弧補間制御)でのジャークを設定します。

加減速方式設定(オプション(Options): ビット0~2)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。「1: 加減速時間一定方式(mcFixedTime)」を設定した場合、ジャーク(Jerk)は使用しません。

加減速方式設定(オプション(Options): ビット0~2)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000, 0.0001~2147483647.0[U/s ²]までの正数
1: 加減速時間一定方式(mcFixedTime)	使用しません

■円弧補間誤差許容値(CircularErrorTolerance)

中心点指定の円弧補間制御では、始点位置と中心点位置から算出した円弧の軌跡と、終点(EndPoint)に設定した終点位置がずれる場合があります。

円弧補間誤差許容値(CircularErrorTolerance)は、算出した円弧の軌跡と終点位置の誤差の許容範囲を設定します。

許容範囲	内容
算出した誤差 ≤ 円弧補間誤差許容値(CircularErrorTolerance)	うず巻補間により誤差補正を行いながら、設定している終点アドレスに円弧補間を行います。
算出した誤差 > 円弧補間誤差許容値(CircularErrorTolerance)	位置決め始動時に円弧補間誤差許容値オーバー(エラーコード: 1AD9H)となり始動しません。位置決め制御中の場合は、エラー検出時に即停止します。

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

MCv_MoveCircularInterpolateRelative(相対値円弧補間制御)で設定可能なバッファモードを下記に示します。

設定値	内容
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。
1: Buffered(mcBuffered)	制御中のFBに次のFBをバッファリングします。 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFB完了にて、バッファリングFBを順次実行します。
2: BlendingLow(mcBlendingLow)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBとバッファリングFBの目標速度のうち低い方の速度を切換え速度とします。
3: BlendingPrevious(mcBlendingPrevious)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBの目標速度を切換え速度とします。
4: BlendingNext(mcBlendingNext)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、バッファリングFBの目標速度を切換え速度とします。
5: BlendingHigh(mcBlendingHigh)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBとバッファリングFBの目標速度のうち高い方の速度を切換え速度とします。

^{*1} 制御中のFBとバッファリングFB間で停止を行いません。

Point

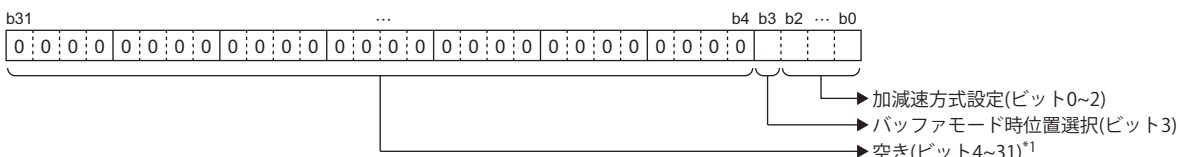
多重起動(バッファモード)の詳細については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

■オプション(Options)

MCv_MoveCircularInterpolateRelative(相対値円弧補間制御)で使用する機能オプションをビット指定で設定します。

ビット指定で設定する内容を下記に示します。

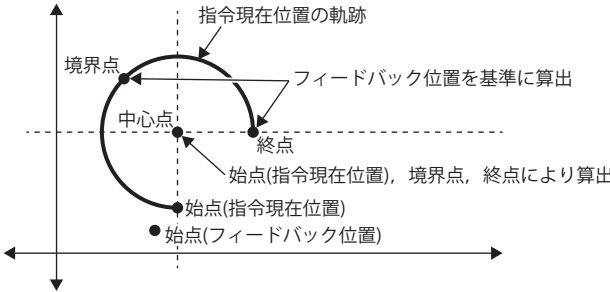
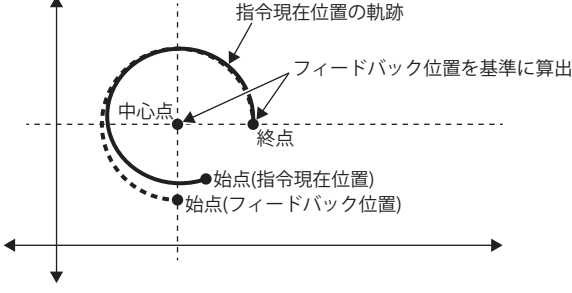
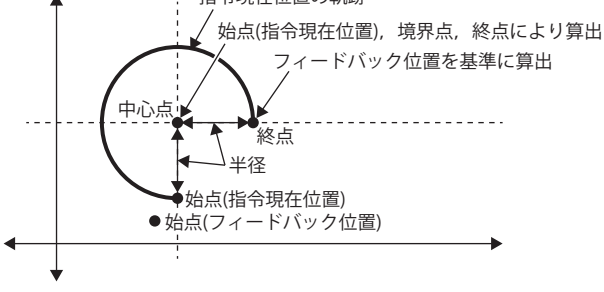


^{*1} 空きには「0」を設定してください。「0」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

ビット	名称	内容
0~2	加減速方式設定	制御を行うための加減速方式を設定します。 • 0: 加減速度指定方式(mcAccDec) • 1: 加減速時間一定方式(mcFixedTime)
3	バッファモード時位置選択	相対値位置決め制御を多重起動する場合の位置を設定します。 • 0: 指令現在位置 • 1: フィードバック位置 *: バッファモード(BufferMode)の「0: Aborting(mcAborting)」設定時に有効になります。

・加減速方式設定(ビット0~2)

設定値	内容
0: 加減速度指定方式(mcAccDec)	MCv_MoveCircularInterpolateRelative(相対値円弧補間制御)で設定した加速度(Acceleration), 減速度(Deceleration), ジャーク(Jerk)を用いて、加速/減速する方式です。
1: 加減速時間一定方式(mcFixedTime)	速度に関係なく、MCv_MoveCircularInterpolateRelative(相対値円弧補間制御)で設定した加減速時間を用いて、加速/減速する方式です。 加減速時間は、加速度(Acceleration)に設定し、減速度(Deceleration), ジャーク(Jerk)は使用しません。

設定値	内容
0: 指令現在位置	指令現在位置からの相対位置制御です。
1: フィードバック位置	フィードバック位置からの相対位置制御です。 補助点(境界点, 中心点, 半径), 終点はフィードバック位置を基準に算出し, 始点は指令現在位置を使用します。 円弧補間モード(CircMode)ごとの動作を下記に示します。 ■「0: 境界点指定(mcBorder)」の場合 境界点, 終点はフィードバック位置を基準に算出します。中心点は始点(指令現在位置), 境界点, 終点によって決まります。  ■「1: 中心点指定(mcCenter)」の場合 中心点, 終点はフィードバック位置を基準に算出します。円弧の軌跡は楕円の円弧となる場合があります。  ■「2: 半径指定(mcRadius)」の場合 終点はフィードバック位置を基準に算出します。中心点は始点(指令現在位置), 半径, 終点によって決まります。 

■必須オブジェクトデータ

MCv_MoveCircularInterpolateRelative(相対値円弧補間制御)を使用する場合、指定した軸グループの構成軸すべてに下記のオブジェクトデータを設定してください。

- Target position(607AH)

オブジェクトデータを設定していない構成軸がある場合、必須オブジェクトデータ未設定(エラーコード: 1AF7H)となり始動しません。

オブジェクトデータ設定に関する詳細は、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

注意事項

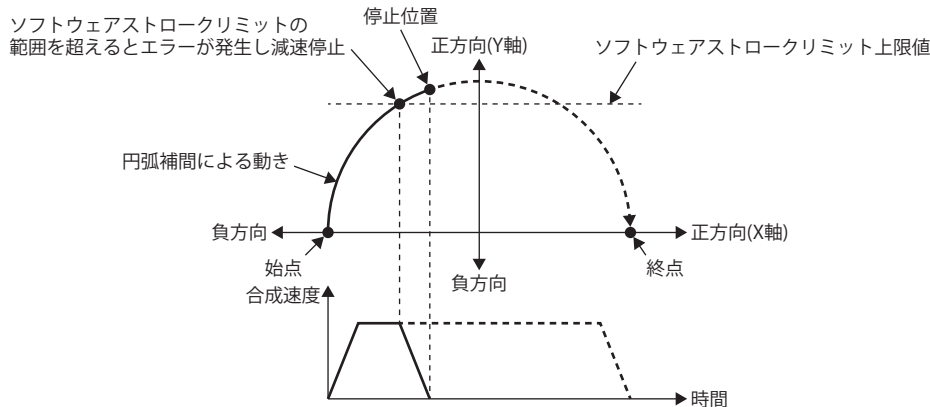
- ストロークリミットを無効に設定した軸を含む円弧補間は行えません。円弧補間時ソフトウェアストロークリミット無効(エラーコード: 1ADAH)となり始動しません。
- 補間動作中に補間経路がストロークリミット範囲を超えた場合は、ソフトウェアストロークリミットオーバ(正方向)(エラーコード: 1A03H), またはソフトウェアストロークリミットオーバ(負方向)(エラーコード: 1A84H)となり運転を停止します。

例

Y軸の正方向でソフトウェアストロークリミットの上限を超えた場合

即停止する場合は、エラー発生と同時に停止します。

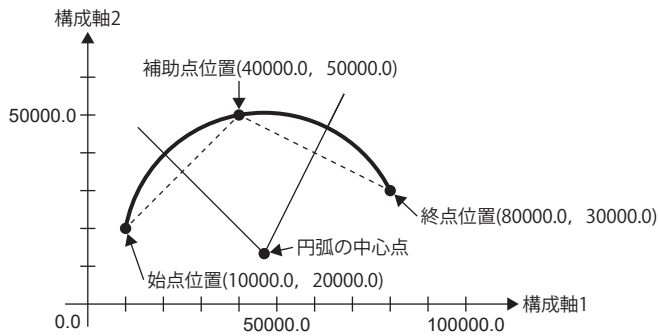
減速停止する場合は、下図のように円弧の軌跡に沿って減速し、停止します。



プログラム例

円弧補間制御始動(bCircularInterpolateCMD)をTRUEにし、軸グループ1(AxesGroup001)を有効にした後、下記の設定により軸グループ1(AxesGroup001)の2軸円弧補間(補助点指定)の相対値位置決め制御を行うプログラム例を下記に示します。

・動作



・軸グループ

項目	設定値
構成軸[1]	Axis0001
構成軸[2]	Axis0002

・設定

項目	設定値	
	構成軸1	構成軸2
補助点	30000.0	30000.0
終点	70000.0	10000.0
速度	50000.0	
加速度	50000.0	
減速度	50000.0	
ジャーク	0.0	
円弧補間誤差許容値	100.0	

■軸

軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1
2	Axis0002	AXIS_REF	軸2

■軸グループ

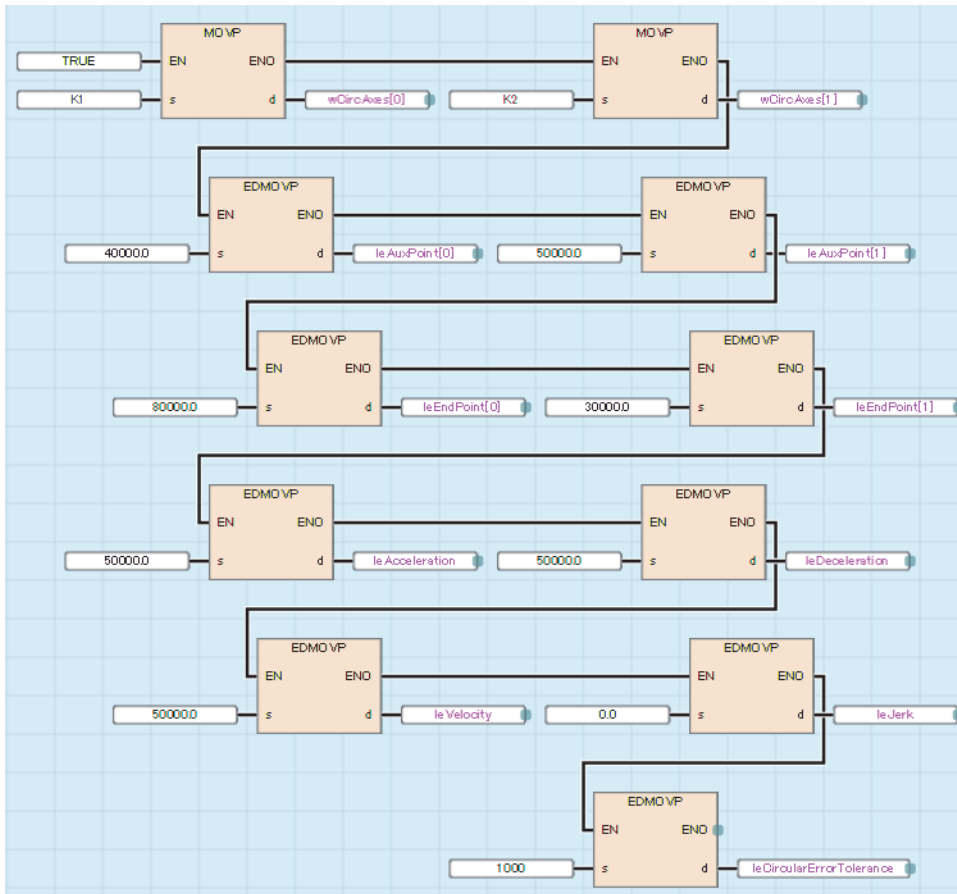
軸グループNo.	ラベル名	データ型	コメント
1	AxesGroup001	AXES_GROUP_REF	軸グループ1

■使用するラベル

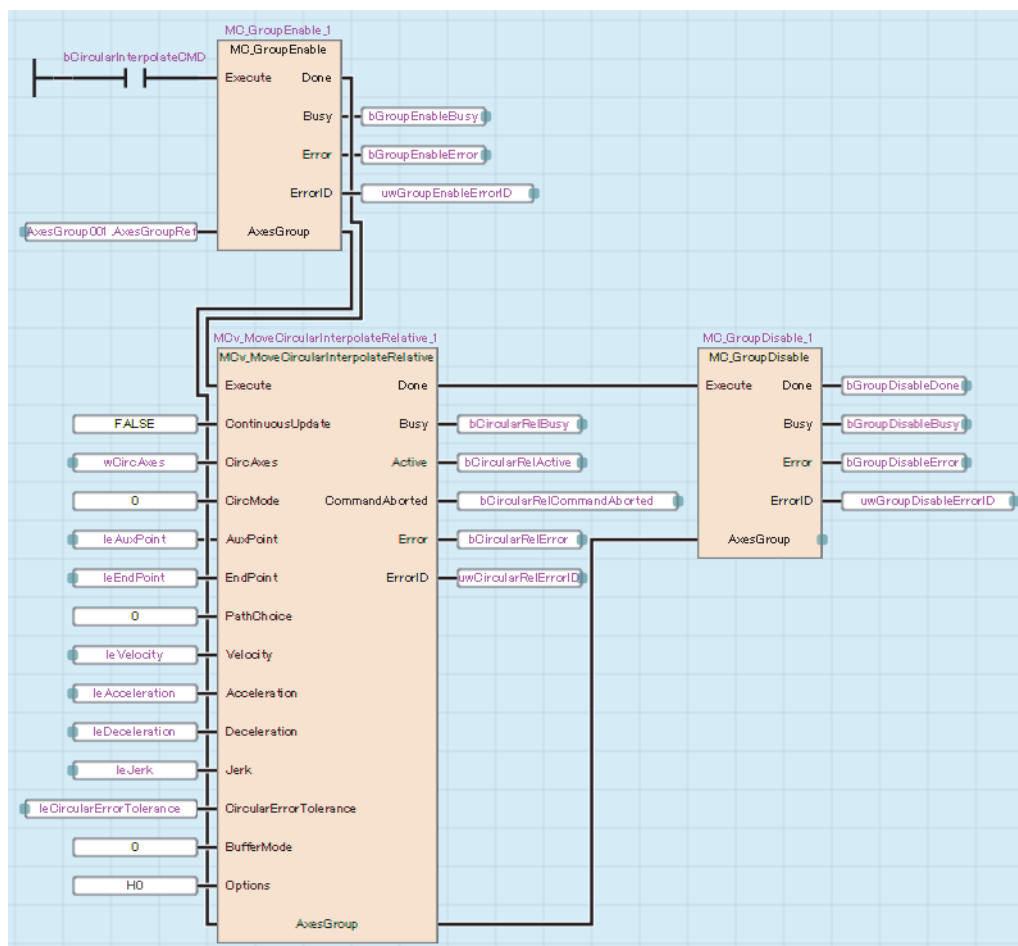
ラベル名	データ型	コメント
MCv_MoveCircularInterpolateRelative_1	MCv_MoveCircularInterpolateRelative	相対値円弧補間制御FB
wCircAxes	ワード[符号付き](0..1)	円弧補間軸
leAuxPoint	倍精度実数(0..15)	補助点
leEndPoint	倍精度実数(0..15)	終点
leVelocity	倍精度実数	速度
leAcceleration	倍精度実数	加速度
leDeceleration	倍精度実数	減速度
leJerk	倍精度実数	ジャーク
leCircularErrorTolerance	倍精度実数	円弧補間誤差許容値
bCircularRelDone	ビット	実行完了
bCircularRelBusy	ビット	実行中
bCircularRelActive	ビット	制御中
bCircularRelCommandAborted	ビット	実行中断
bCircularRelError	ビット	エラー
uwCircularRelErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
MC_GroupEnable_1	MC_GroupEnable	軸グループ有効FB
bCircularInterpolateCMD	ビット	円弧補間制御始動
bGroupEnableDone	ビット	軸グループ有効完了
bGroupEnableBusy	ビット	実行中
bGroupEnableError	ビット	エラー
uwGroupEnableErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
MC_GroupDisable_1	MC_GroupDisable	軸グループ無効FB
bGroupDisableDone	ビット	軸グループ無効完了
bGroupDisableBusy	ビット	実行中
bGroupDisableError	ビット	エラー
uwGroupDisableErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

・2軸円弧補間制御用データ設定



• 軸グループ有効/円弧補間制御(境界点指定)/軸グループ無効



■STプログラム

//-----2軸円弧補間制御用データ設定-----

```
wCircAxes[0]:= 1;
wCircAxes[1]:= 2;
leAuxPoint[0]:= 30000.0;
leAuxPoint[1]:= 30000.0;
leEndPoint[0]:= 70000.0;
leEndPoint[1]:= 10000.0;
leVelocity:= 50000.0;
leAcceleration:= 50000.0;
leDeceleration:= 50000.0;
leJerk:= 0.0;
leCircularErrorTolerance:= 100.0;
```

//-----軸グループ有効-----

```
MC_GroupEnable_1(
  AxesGroup:= AxesGroup001.AxesGroupRef ,
  Execute:= bCircularInterpolateCMD ,
  Done=> bGroupEnableDone ,
  Busy=> bGroupEnableBusy ,
  Error=> bGroupEnableError ,
  ErrorID=> uwGroupEnableErrorID
);
```

//-----円弧補間制御(境界点指定)-----

```
MCv_MoveCircularInterpolateRelative_1(
  AxesGroup:= AxesGroup001.AxesGroupRef ,
  Execute:= bGroupEnableDone ,
  ContinuousUpdate:= FALSE ,
  CircAxes:= wCircAxes ,
  CircMode:= MC_CIRC_MODE__mcBorder ,
  AuxPoint:= leAuxPoint ,
  EndPoint:= leEndPoint ,
  PathChoice:= MC_CIRC_PATHCHOICE__mcCW ,
  Velocity:= leVelocity ,
  Acceleration:= leAcceleration ,
  Deceleration:= leDeceleration ,
  Jerk:= leJerk ,
  CircularErrorTolerance:= leCircularErrorTolerance ,
  BufferMode:= MC_BUFFER_MODE__mcAborting ,
  Options:= H00000000 ,
  Done=> bCircularRelDone ,
  Busy=> bCircularRelBusy ,
  Active=> bCircularRelActive ,
  CommandAborted=> bCircularRelCommandAborted ,
  Error=> bCircularRelError ,
  ErrorID=> uwCircularRelErrorID
);
```

```
//-----軸グループ無効-----  
MC_GroupDisable_1(  
  AxesGroup:= AxesGroup001.AxesGroupRef ,  
  Execute:= bCircularRelDone ,  
  Done=> bGroupDisableDone ,  
  Busy=> bGroupDisableBusy ,  
  Error=> bGroupDisableError ,  
  ErrorID=> uwGroupDisableErrorID  
);
```

42.14 カム動作開始

MC_CamIn

指定したカムデータに従ってカム動作を開始します。

ラダー	FBD/LD	ST
MC_CamIn DUT: MasterMaster :DUT DUT: SlaveSlave :DUT B: ExecuteInSync :B B: ContinuousUpdateBusy :B L: MasterOffsetActive :B L: SlaveOffsetCommandAborted :B L: MasterScalingError :B L: SlaveScalingErrorID :UW L: MasterStartDistanceEndOfProfile :B L: MasterSyncPosition ENUM: StartMode ENUM: MasterValueSource DUT: CamTableID ENUM: BufferMode UD: Options InputPerCycle 0 Reference 0 OutputData 0 InstanceID 0	MC_CamIn ExecuteInSync ContinuousUpdateBusy MasterOffsetActive SlaveOffsetCommandAborted MasterScalingError SlaveScalingErrorID MasterStartDistanceEndOfProfile MasterSyncPosition StartMode MasterValueSource CamTableID BufferMode Options Master Slave	MC_CamIn(Master:= ?AXIS_REF? , Slave:= ?AXIS_REF? , Execute:= ?BOOL? , ContinuousUpdate:= ?BOOL? , MasterOffset:= ?LREAL? , SlaveOffset:= ?LREAL? , MasterScaling:= ?LREAL? , SlaveScaling:= ?LREAL? , MasterStartDistance:= ?LREAL? , MasterSyncPosition:= ?LREAL? , StartMode:= ?INT? , MasterValueSource:= ?INT? , CamTableID:= ?MC_CAM_ID? , BufferMode:= ?INT? , Options:= ?DWORD? , InSync=> ?BOOL? , Busy=> ?BOOL? , Active=> ?BOOL? , CommandAborted=> ?BOOL? , Error=> ?BOOL? , ErrorID=> ?WORD? , EndOfProfile=> ?BOOL?);

MC_CamIn

ExecuteInSync

ContinuousUpdateBusy

MasterOffsetActive

SlaveOffsetCommandAborted

MasterScalingError

SlaveScalingErrorID

MasterStartDistanceEndOfProfile

MasterSyncPosition

StartMode

MasterValueSource

CamTableID

BufferMode

Options

Master

Slave

MC_CamIn(
Master:= ?AXIS_REF? ,
Slave:= ?AXIS_REF? ,
Execute:= ?BOOL? ,
ContinuousUpdate:= ?BOOL? ,
MasterOffset:= ?LREAL? ,
SlaveOffset:= ?LREAL? ,
MasterScaling:= ?LREAL? ,
SlaveScaling:= ?LREAL? ,
MasterStartDistance:= ?LREAL? ,
MasterSyncPosition:= ?LREAL? ,
StartMode:= ?INT? ,
MasterValueSource:= ?INT? ,
CamTableID:= ?MC_CAM_ID? ,
BufferMode:= ?INT? ,
Options:= ?DWORD? ,
InSync=> ?BOOL? ,
Busy=> ?BOOL? ,
Active=> ?BOOL? ,
CommandAborted=> ?BOOL? ,
Error=> ?BOOL? ,
ErrorID=> ?WORD? ,
EndOfProfile=> ?BOOL?)

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Master	主軸	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📄 1378ページ AxisName.AxisRef.(軸情報)
Slave	従軸	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📄 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_CamIn(カム動作開始)を実行します。
ContinuousUpdate	連続更新	BOOL	起動時	TRUE, FALSE	FALSE	連続して、主軸オフセット(MasterOffset)、従軸オフセット(SlaveOffset)、主軸係数(MasterScaling)、従軸係数(SlaveScaling)、カムテーブルID(CamTableID)の変更を有効にするか、無効にするかを設定します。 ・ FALSE: 無効 ・ TRUE: 有効
MasterOffset	主軸オフセット	LREAL	起動時/再起動可/連続更新可	-10000000000.0~10000000000.0	0.0	主軸(Master)の位相をオフセット量分ずらしします。 詳細は下記を参照してください。 ☞ 1719ページ 主軸オフセット(MasterOffset)
SlaveOffset	従軸オフセット	LREAL	起動時/再起動可/連続更新可	-10000000000.0~10000000000.0	0.0	従軸(Slave)の変位をオフセット量分ずらしします。 詳細は下記を参照してください。 ☞ 1720ページ 従軸オフセット(SlaveOffset)
MasterScaling	主軸係数	LREAL	起動時/再起動可/連続更新可	0.01~10.0	1.0	カムテーブルの1サイクル長を拡大、縮小します。 詳細は下記を参照してください。 ☞ 1721ページ 主軸係数(MasterScaling)
SlaveScaling	従軸係数	LREAL	起動時/再起動可/連続更新可	0.01~10.0	1.0	カムテーブルのストローク量を拡大、縮小します。 詳細は下記を参照してください。 ☞ 1721ページ 従軸係数(SlaveScaling)
MasterStartDistance	主軸追従距離	LREAL	起動時	-10000000000.0~10000000000.0	0.0	設定値は無視します。
MasterSyncPosition	主軸同期開始位置	LREAL	起動時	-10000000000.0~10000000000.0	0.0	1サイクル現在値(InputPerCycle)が同期を開始する主軸(Master)の位置を設定します。 *: 主軸同期開始位置の通過 チェック対象指定(オプション(Options): ビット21)が「1: 主軸の指令現在位置、またはフィードバック位置」の場合、有効範囲はリングカウンタ範囲内になります。
StartMode	開始モード	INT (MC_START_MODE)	起動時	0, 1	0	カム動作を開始するタイミングを設定します。 ・ 0: 即時(mcImmediate) ・ 1: 絶対(mcAbsolute) 詳細は下記を参照してください。 ☞ 1722ページ 開始モード(StartMode)
MasterValueSource	主軸データソース選択	INT (MC_SOURCE)	起動時	1, 2, 101, 102	1	主軸(Master)のデータソースを設定します。 ・ 1: 指令現在(mcSetValue) ・ 2: フィードバック値(mcActualValue) ・ 101: 最新指令現在値(mcLatestSetValue) ・ 102: 最新フィードバック値(mcLatestActualValue) 詳細は下記を参照してください。 ☞ 1724ページ 主軸データソース選択(MasterValueSource)

入力変数	名称	データ型	取込	有効範囲	初期値	内容
CamTableID	カムテーブルID	MC_CAM_ID	起動時/再起動可 /連続更新可	1~60000	0	カムのIDを設定します。 カムのIDは、 MC_CamTableSelect(カムテーブル選択)で事前に展開エリアへ展開して使用します。 MC_CAM_REF.CamIDまたは MC_CAM_REF.ProfileData.IDを設定してください。  1527ページ カムテーブル選択
BufferMode	バッファモード	INT (MC_BUFFER_MODE)	起動時	0, 1	0	バッファモードを設定します。 • 0: Aborting(mcAborting) • 1: Buffered(mcBuffered) 詳細は下記を参照してください。  1725ページ バッファモード (BufferMode)
Options	オプション	DWORD(HEX)	起動時	00000000H~00210000H	00000000H	MC_CamIn(カム動作開始)の機能オプションをビット指定で設定します。 詳細は、下記を参照してください。  1726ページ オプション (Options)

■出力変数

出力変数	名称	データ型	初期値	内容
InSync	同期中	BOOL	FALSE	出力値(OutputData)が同期を開始したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MC_CamIn(カム動作開始)を実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	1サイクル現在値(InputPerCycle)が同期を開始したときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	MC_CamIn(カム動作開始)の実行が中断したときに、TRUEになります。 実行指令(Execute)がFALSEになると、FALSEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。  使用するコントローラのユーザーズマニュアル
EndOfProfile	カムサイクル完了	BOOL	FALSE	制御中(Active)がTRUEになった後、1サイクル長分移動するたびに、FBを呼び出すPOU(プログラム部品)の実行周期の1周期のみ、TRUEになります。 詳細は、下記を参照してください。  1726ページ カムサイクル完了(EndOfProfile)

■公開変数

公開変数	名称	データ型	初期値	内容
InputPerCycle	1サイクル現在値	LREAL	0.0	1サイクル現在値を格納します。 詳細は、下記を参照してください。  1726ページ 1サイクル現在値(InputPerCycle)
Reference	基準値	LREAL	0.0	基準値を格納します。 詳細は、下記を参照してください。  1727ページ 基準値(Reference)
OutputData	出力値	LREAL	0.0	出力値を格納します。 詳細は、下記を参照してください。  1728ページ 出力値(OutputData)
InstanceID	インスタンスID	WORD(UINT)	0	インスタンスIDです。 インスタンス生成時にシステムにより自動設定します。 FBの入力等で使用します。

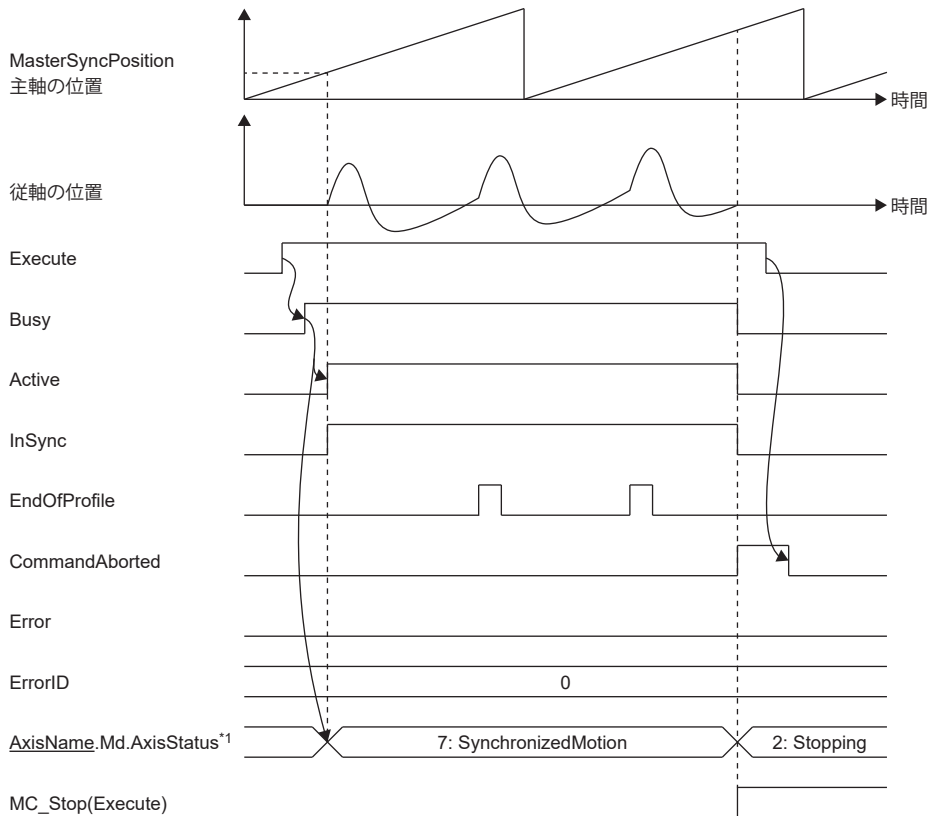
機能

- MC_CamIn(カム動作開始)は、主軸オフセット(MasterOffset)、従軸オフセット(SlaveOffset)、主軸係数(MasterScaling)、従軸係数(SlaveScaling)、主軸追従距離(MasterStartDistance)、主軸同期開始位置(MasterSyncPosition)、開始モード(StartMode)、主軸データソース選択(MasterValueSource)、カムテーブルID(CamTableID)、バッファモード(BufferMode)を設定し、カム動作を実行します。
- 動作を終了する場合は、MC_Stop(強制停止)にて行います。

■ タイミングチャート

・ 正常完了の場合

開始モード(StartMode)が「1: 絶対(mcAbsolute)」, MC_CamTableSelect(カムテーブル選択)で設定した主軸絶対座標(MasterAbsolute)が「FALSE(相対座標)」の場合



*1 従軸(Slave)の軸状態です。

・ 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■カム動作の開始

開始モード(StartMode)の設定により、カム動作の同期タイミングとカム制御データの反映タイミングを設定できます。

・カム動作の同期タイミング

開始モード(StartMode)と関連パラメータ(主軸同期開始位置(MasterSyncPosition)と主軸追従距離(MasterStartDistance))により、1サイクル現在値(InputPerCycle)、基準値(Reference)、出力値(OutputData)の同期タイミング(更新を開始するタイミング)は下記ようになります。

開始モード(StartMode)*1	同期を開始するタイミング		
	1サイクル現在値(InputPerCycle)	基準値(Reference)	出力値(OutputData)
0: 即時(mclmmediate)	MC_CamIn(カム動作開始)を実行したとき。 主軸同期開始位置(MasterSyncPosition)と主軸追従距離(MasterStartDistance)は無視します。		
1: 絶対(mcAbsolute)	主軸の位置*2が主軸同期開始位置(MasterSyncPosition)を通過*3したとき。 主軸追従距離(MasterStartDistance)は無視します。		

*1 開始モード(StartMode)については、下記を参照してください。

📖 1722ページ 開始モード(StartMode)

*2 主軸同期開始位置の通過チェック対象指定(オプション(Options): ビット21)により使用する位置データを設定します。詳細は、下記を参照してください。

📖 1726ページ オプション(Options)

*3 主軸の位置が、主軸同期開始位置(MasterSyncPosition)(または、主軸同期開始位置(MasterSyncPosition)+主軸追従距離(MasterStartDistance))を跨いだ状態です。同値になった場合は未通過とし、同値から移動した場合に通過とします。

MC_CamIn(カム動作開始)実行後、1サイクル現在値(InputPerCycle)が同期すると、制御中(Active)がTRUEとなり、出力値(OutputData)が同期すると同期中(InSync)がTRUEとなります。

・FBの再起動/連続更新による制御変更のタイミング

FBの再起動/連続更新により変更したパラメータを制御に反映するタイミングは、開始モード(StartMode)により下記ようになります。

開始モード(StartMode)	反映タイミング
0: 即時(mclmmediate)	即時
1: 絶対(mcAbsolute)	1サイクル現在値(InputPerCycle)がカムテーブルの1点目を通過したとき。

変更値が範囲外になった場合、該当するエラーコードやイベントコードを出力し、変更前の値を維持して制御を続けます。

パラメータ	エラー
MasterOffset(主軸オフセット)	主軸オフセット範囲外(エラーコード: 1AAFH)
SlaveOffset(従軸オフセット)	従軸オフセット範囲外(エラーコード: 1AB0H)
MasterScaling(主軸係数)	主軸係数範囲外(エラーコード: 1AB1H)
SlaveScaling(従軸係数)	従軸係数範囲外(エラーコード: 1AB2H)
CamTableID(カムテーブルID)	・ CamTableID(カムテーブルID)の数値が範囲外の場合 カムテーブルID範囲外(エラーコード: 1AAEH) ・ CamTableID(カムテーブルID)の展開エリアがない場合 カムテーブル無し警告(イベントコード: 00D44H)

■カムテーブルの情報

カムテーブルの情報として、MC_CamTableSelect(カムテーブル選択)で設定した繰り返し動作、主軸絶対座標、従軸絶対座標による、MC_CamIn(カム動作開始)実行時のカム動作を下記に示します。

○: 制御に使用する, ×: 制御に使用しない

名称	変数名	補間方法指定(Interpolate)			備考
		0: 直線補間	1: 区間ごとに指定	2: スプライン補間	
繰り返し動作	Periodic	○	○	○	エンジニアリングツールで演算プロファイルの展開設定、またはMC_CamTableSelect(カムテーブル選択)の入力変数にて設定する項目です。
主軸絶対座標	MasterAbsolute	○	○	○	
従軸絶対座標	SlaveAbsolute	○	○	○	
補間方法指定	Interpolate	○	○	○	カムデータ形式/ロータリカッター形式の演算プロファイルから展開エリアに展開したデータの制御に関する項目です。
1サイクル長	CycleLength	○	○	○	
ストローク量	Stroke	○	○	○	
開始点	StartPoint	×	×	×	
初期ストローク量	StartStroke	×	○	○	

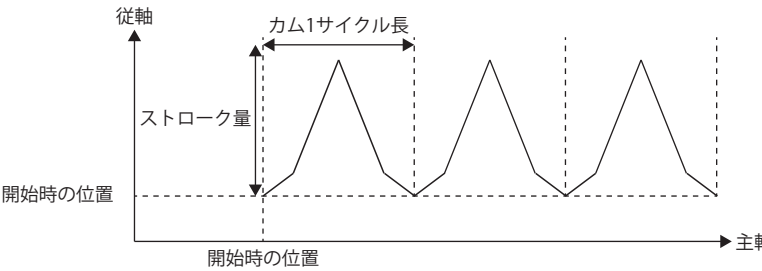
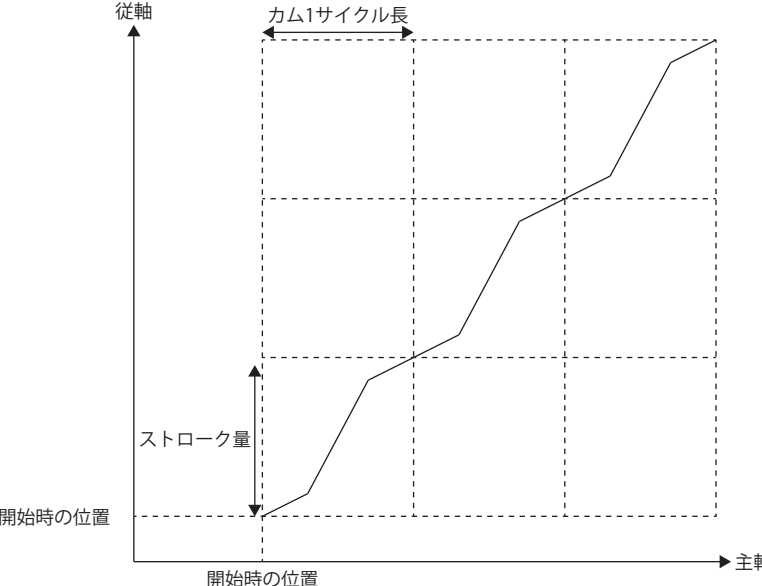
・繰り返し動作(Periodic)

繰り返し動作(Periodic)の設定によるMC_CamIn(カム動作開始)実行時の動作を下記に示します。

繰り返し動作(Periodic)	内容
FALSE(単発動作)	制御中(Active)がTRUEになってから、1サイクルだけ運転を行います。その後は、1サイクル運転後の動作指定(オプション(Options): ビット16)で設定した動作となります。 ■1サイクル運転後の動作指定(オプション(Options): ビット16)が「0: 終了」の場合 1サイクル終了後に同期状態が解除され、その後、実行指令(Execute)を再立ち上げると、基準値(Reference)、出力値(OutputData)は初期化されます。 ・従軸絶対座標(SlaveAbsolute)がFALSE(相対座標)の場合

繰り返し動作(Periodic)	内容
FALSE(単発動作)	• 従軸絶対座標(SlaveAbsolute)がTRUE(絶対座標)の場合 主軸の累積現在位置 InputPerCycle Reference OutputData 従軸の累積現在位置 従軸の状態 (AxisName.Md.AxisStatus) 従軸のプロファイルID Execute Busy Active InSync EndOfProfile InstanceID 時間 時間 時間 時間 時間 4: Standstill 7: SynchronizedMotion 4: Standstill 7: SynchronizedMotion 4: Standstill 0 1 0 1 0 時間 時間 時間 時間 時間 0 1000 0.0 Referenceは更新されず、初期値のままになります。 Reference、初期ストローク量(StartStroke)から初期値を算出します。

繰り返し動作(Periodic)	内容
FALSE(単発動作)	■1サイクル運転後の動作指定(オプション(Options): ビット16)が「1: 再起動まち」の場合 1サイクル終了後も同期状態を維持し、実行指令(Execute)を再立ち上げると、基準値(Reference), 出力値(OutputData)の値も維持します。 同期状態の解除は従軸へ停止要因を入力します。 1サイクル終了後の再起動中に、連続更新可の入力ラベルの値を変更した場合も、実行指令(Execute)を再立ち上げるまで入力取込みを行いません。 実行指令(Execute)を再立ち上げると、開始モード(StartMode)の設定に関わらず、即時、従軸が動き出します。 1サイクル終了後の再起動待ち中に、1サイクル現在値変更を実行すると、次のサイクルに対して有効となります。

繰り返し動作(Periodic)	内容
TRUE(繰り返し動作)	連続的にカムテーブルの実行を繰り返します。カム動作(往復カム, または送りカム)により下記の動作となります。 ■往復カム 繰り返しによるカムテーブル始点の従軸(Slave)の位置が毎回同じ場合は, 往復カムとして動作します。  ■送りカム 繰り返しによるカムテーブル始点の従軸(Slave)の位置が始点と終点で指令現在位置が異なる場合は, 送りカムとして動作します。 

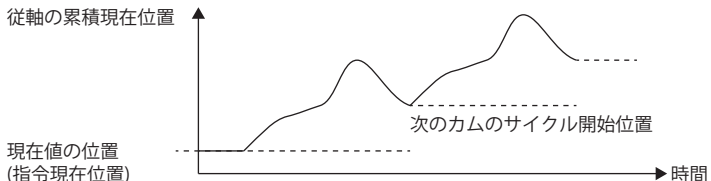
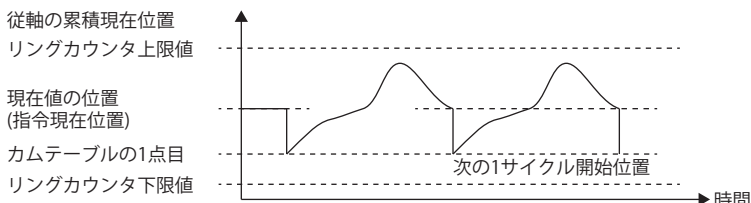
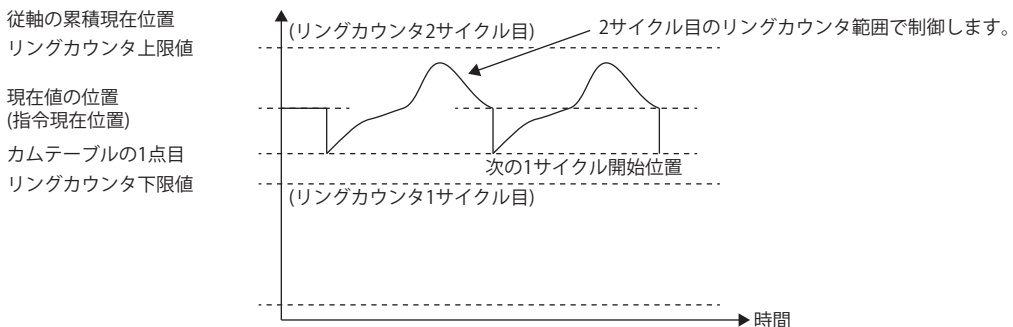
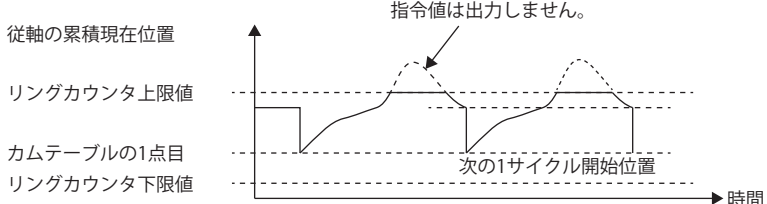
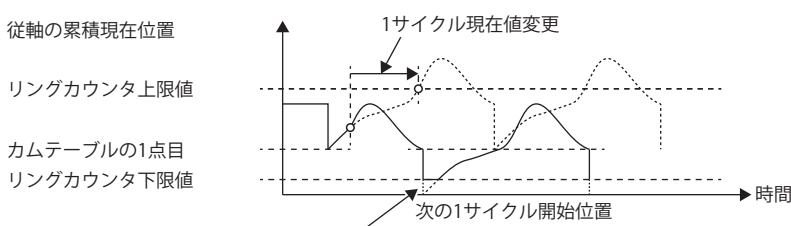
・主軸絶対座標(MasterAbsolute)

主軸絶対座標(MasterAbsolute)の設定によるカム動作を下記に示します。

主軸絶対座標 (MasterAbsolute)	内容
FALSE(相対座標)	主軸同期開始位置(MasterSyncPosition)がカムデータの始点となります。同期中(InSync)がTRUEになると、主軸(Master)の相対移動量に応じてカム動作を実行します。 カムテーブルと主軸(Master)のリングカウンタが合わなくても、連続的にカム動作を実行します。 <例> 下記設定の場合 - 主軸: 現在値のリングカウンタ: 0.0000~360.0000[degree] - 主軸同期開始位置(MasterSyncPosition): 180.0000[degree] - カム: カム1サイクル長: 540.0000[degree] [主軸] [従軸] 主軸が「180.0000」の位置を通過時に従軸が動作開始。 主軸の移動量に応じて動作する。 送りカム InSync

・ 従軸絶対座標(SlaveAbsolute)

従軸絶対座標(SlaveAbsolute)の設定によるカム動作を下記に示します。

従軸絶対座標 (SlaveAbsolute)	内容
FALSE(相対座標)	同期中(InSync)がTRUEになると、従軸は現在の位置(指令現在位置)からカムテーブルの動作を実行します。カムテーブルの繰り返し動作(Periodic)が「TRUE: 繰り返し動作」の場合、カムテーブルの1サイクルを完了すると1サイクル完了時のストローク位置(指令現在位置)から次のサイクルを開始します。  従軸の累積現在位置 現在値の位置 (指令現在位置) 次のカムのサイクル開始位置 時間
TRUE(絶対座標)	同期中(InSync)がTRUEになった時点の従軸(Slave)の現在値を含むリングカウンタ範囲でカム動作を実行します。同期中(InSync)およびカムサイクル完了(EndOfProfile)がTRUEになったときに、従軸(Slave)の指令現在位置がカムテーブルの始点に戻るよう、指令を1演算周期で出力します。このときの指令が大きい場合、ドライブユニットへの位置指令や速度指令が大きくなり、MR-J5(W)-Gのサーボエラー ([AL. 035(指令周波数異常)])の発生要因となります。従軸(Slave)の指令現在位置がリングカウンタ範囲外となる値は出力しません。 ■ストローク範囲が従軸のリングカウンタ範囲内の場合  従軸の累積現在位置 リングカウンタ上限値 現在値の位置 (指令現在位置) カムテーブルの1点目 リングカウンタ下限値 次の1サイクル開始位置 時間 ■ストローク範囲が従軸のリングカウンタ範囲内(2サイクル目でカムを起動した)の場合  従軸の累積現在位置 リングカウンタ上限値 現在値の位置 (指令現在位置) カムテーブルの1点目 リングカウンタ下限値 次の1サイクル開始位置 時間 ■ストローク範囲が従軸のリングカウンタ範囲外の場合  従軸の累積現在位置 リングカウンタ上限値 カムテーブルの1点目 リングカウンタ下限値 次の1サイクル開始位置 時間 ■1サイクル現在値変更を実行した場合 ・ 従軸(Slave)が動作しないように基準値(Reference)を補正するため、リングカウンタ範囲外となる範囲が変化します。従軸(Slave)を動作させて1サイクル現在値を変更する場合は、主軸オフセット(MasterOffset)を変更してください。  従軸の累積現在位置 リングカウンタ上限値 カムテーブルの1点目 リングカウンタ下限値 次の1サイクル開始位置 時間 リングカウンタ範囲外となる範囲が変化します。 リングカウンタ範囲外となる指令値は出力しません。

• 1サイクル長(CycleLength)

1サイクル長には1サイクルに必要な入力量を設定します。詳細は、下記を参照してください。

📖 1726ページ 1サイクル現在値(InputPerCycle)

• ストローク量(Stroke)

ストローク量にはストローク比100%に対応するストローク量を設定します。詳細は、下記を参照してください。

📖 1728ページ 出力値(OutputData)

• 開始点(StartPoint)と初期ストローク量(StartStroke)

補間方法指定(Interpolate)により、カム動作に使用する開始点(StartPoint)と初期ストローク量(StartStroke)が異なります。

補間方法指定(Interpolate)	カム動作の開始点	カム動作の初期ストローク量
0: 直線補間	カムテーブルの1点目の入力値	カムテーブルの1点目の出力値
1: 区間ごとに指定	カムテーブルの1点目の開始点(StartPoint)	カムテーブルの初期ストローク量(StartStroke)
2: スプライン補間		

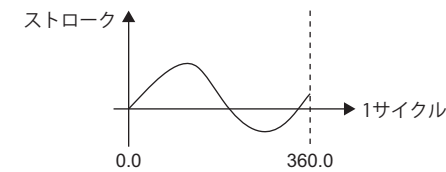
■カム動作の開始点

1サイクル現在値(InputPerCycle)が「0.0」となるよう動作します。

• カムテーブル

■カム(1)(カム動作の開始点が「0.0」)

- 補間方法: 区間ごとに指定
- 1サイクル長: 360.0
- 開始点: 0.0
- 初期ストローク量: 0.0



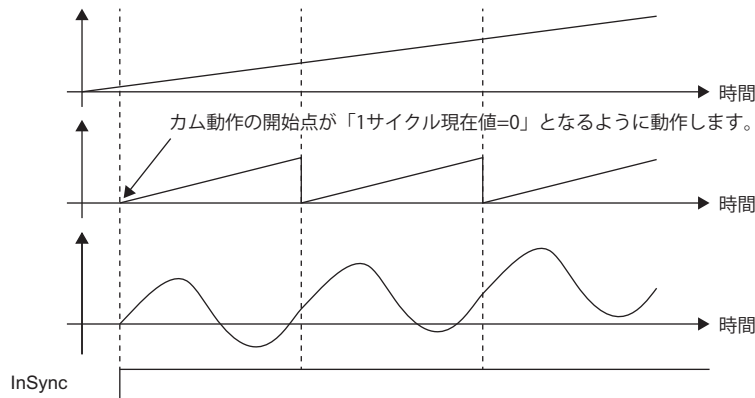
• カム動作

下記の動作となります。

主軸の指令現在位置

1サイクル現在値

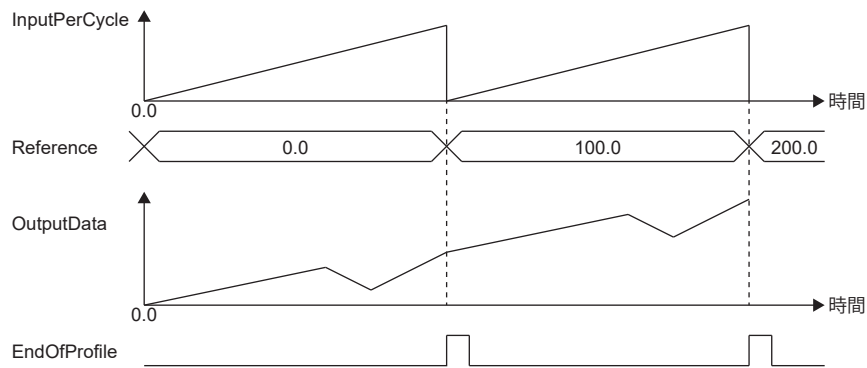
従軸の指令現在位置



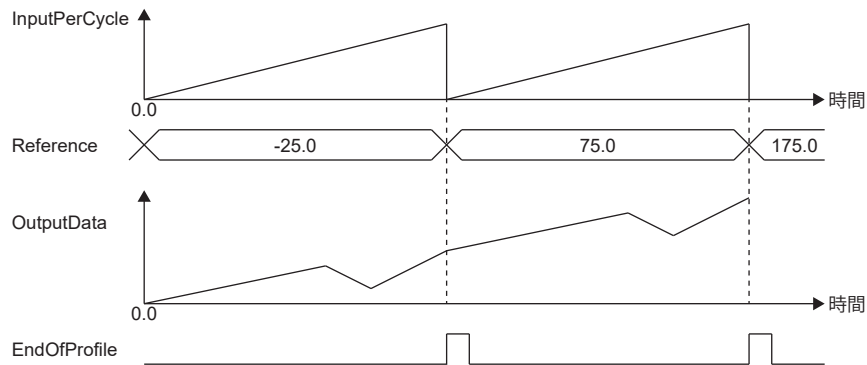
■カム動作の初期ストローク量

同期開始時の出力値(OutputData)となるよう動作します。カム動作の初期ストローク量により従軸(Slave)が動作しないようにするため、基準値(Reference)から初期ストローク量を減算します。

- ・カム動作の初期ストローク量(StartStroke)が「0.0」の場合



- ・カム動作の初期ストローク量(StartStroke)が「25.0」の場合

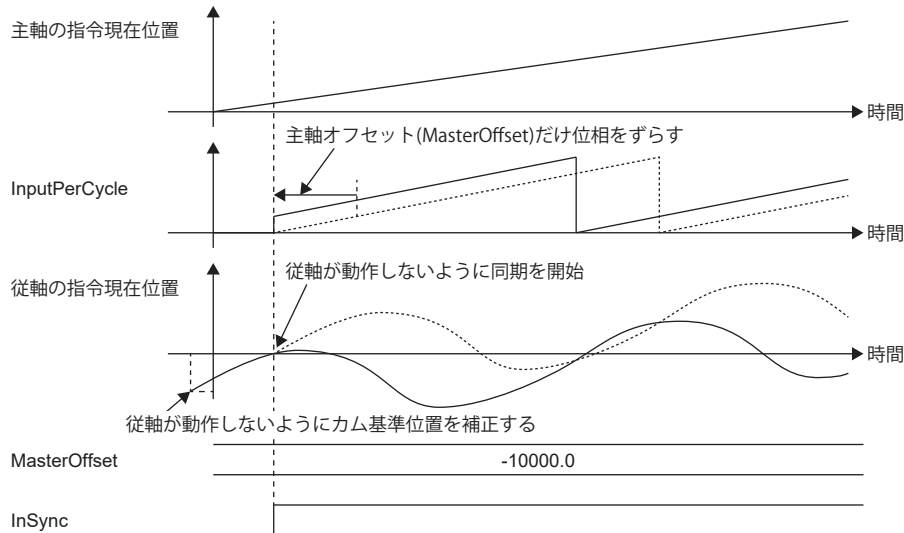


■主軸オフセット(MasterOffset)

主軸オフセット(MasterOffset)を設定することで、主軸(Master)の位相をオフセット量分ずらしします。(主軸追従距離(MasterStartDistance)と主軸同期開始位置(MasterSyncPosition)には影響しません。)

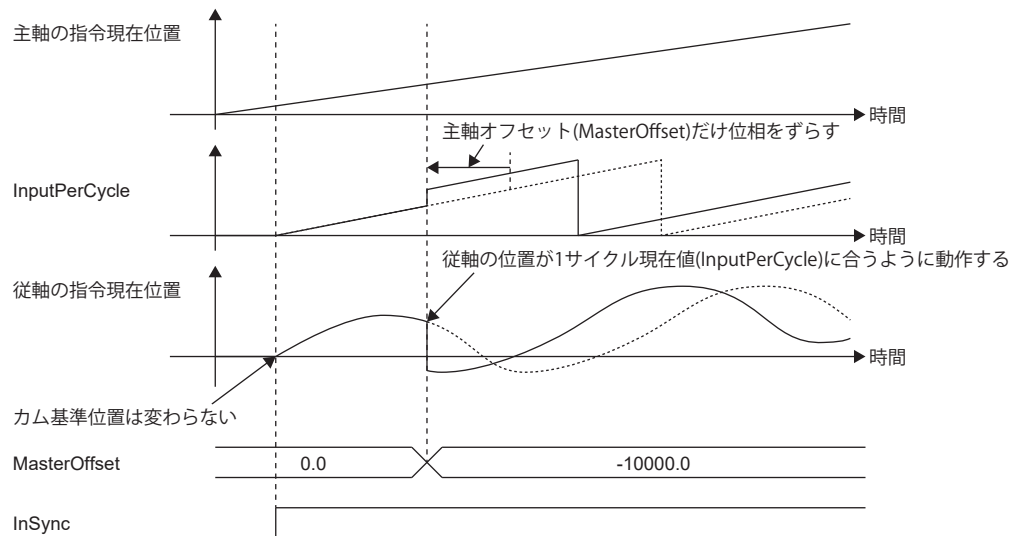
動作開始時に「0.0」以外を設定した場合、同期中(InSync)の立上り時に、主軸(Master)の位置に対してオフセット量分加算したカム1サイクル位置となります。このとき、従軸(Slave)が動作しないようにするため、基準値(Reference)を補正します。

- ・ 開始モード(StartMode)が「0: 即時(mclImmediate)」の場合



同期中(InSync)がTRUEのときに変更した場合は、従軸(Slave)の位置が主軸オフセット(MasterOffset)を加算した1サイクル現在値に合うように動作します。

- ・ 開始モード(StartMode)が「0: 即時(mclImmediate)」の場合

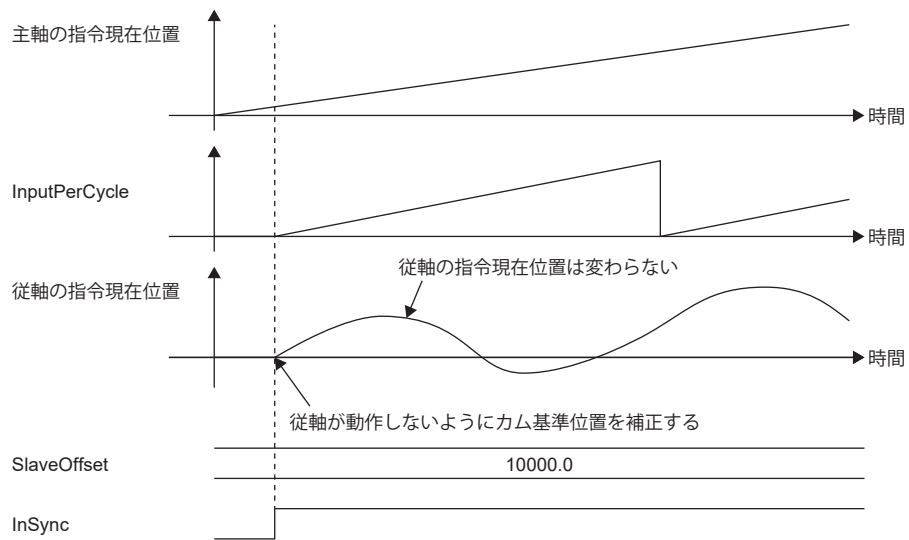


■従軸オフセット(SlaveOffset)

従軸(Slave)の変位をオフセット量分ずらしします。

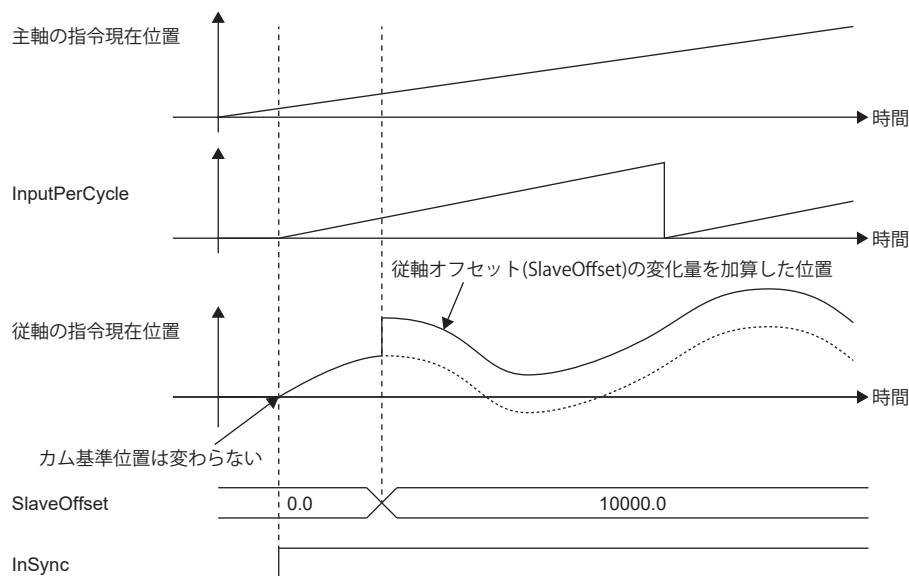
同期中(InSync)の立上り前に「0.0」以外を設定した場合、同期中(InSync)の立上り時に従軸(Slave)が動作しないようにするため、基準値(Reference)を補正します。

- ・開始モード(StartMode)が「0: 即時(mclmmediate)」の場合



同期中(InSync)がTRUEのときに変更した場合は、従軸(Slave)の位置が、オフセット変化量を加算した位置に合うよう動作します。

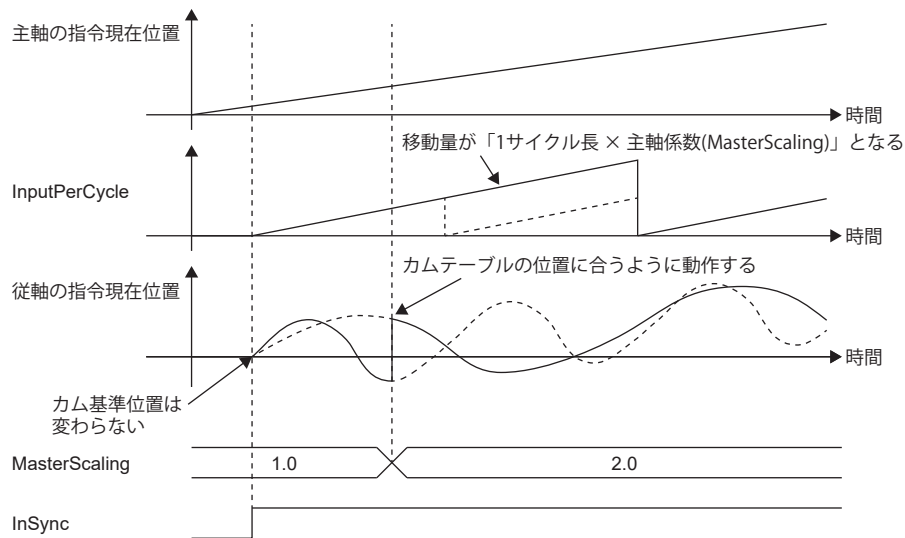
- ・開始モード(StartMode)が「0: 即時(mclmmediate)」の場合



■主軸係数(MasterScaling)

主軸係数(MasterScaling)は、カムテーブルの1サイクル長を拡大、縮小します。

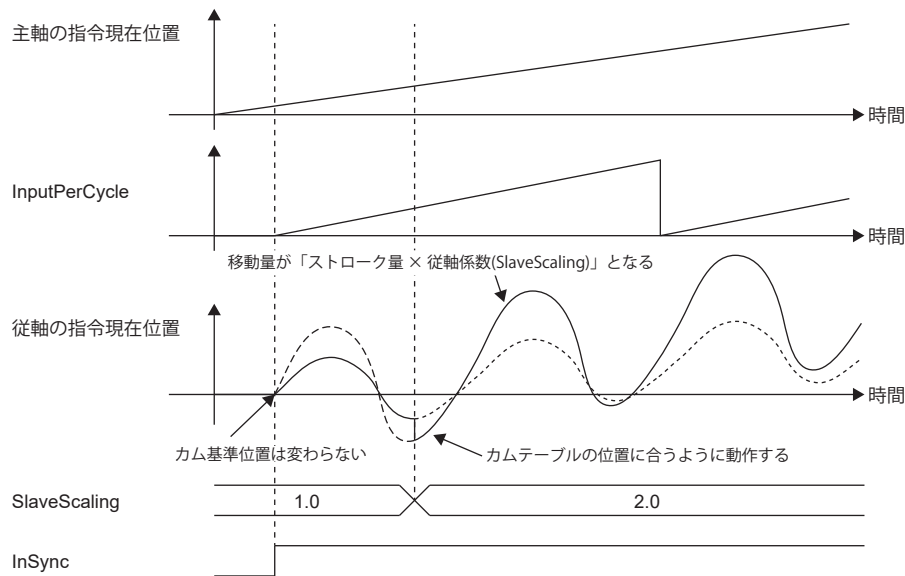
同期中に変更した場合、従軸(Slave)の位置がカムテーブルの位置に合うように移動量分の指令を出力します。



■従軸係数(SlaveScaling)

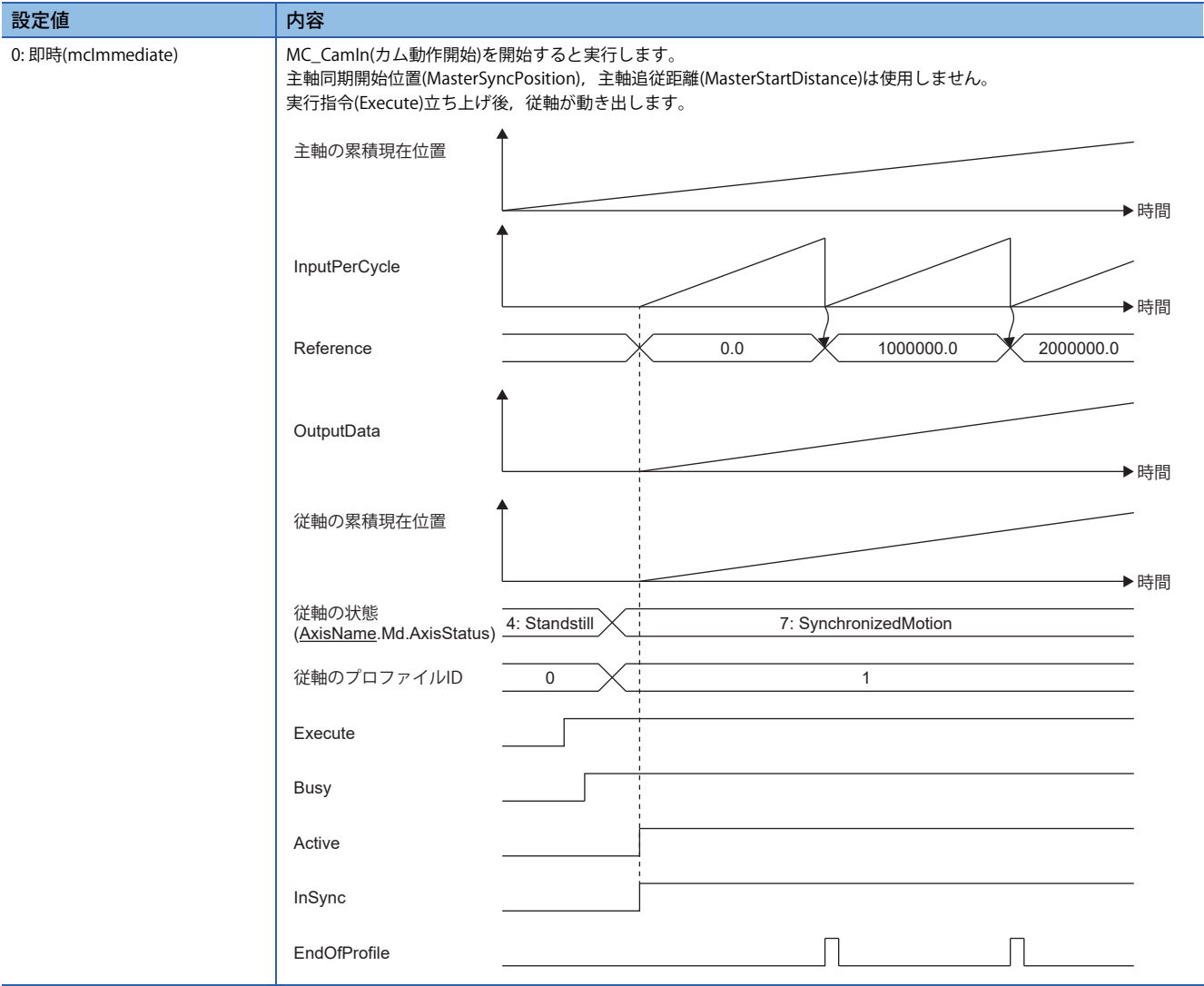
従軸係数(SlaveScaling)は、カムテーブルのストローク量を拡大、縮小します。

同期中に変更した場合、従軸(Slave)の位置がカムテーブルの位置に合うように移動量分の指令を出力します。



■開始モード(StartMode)

カム動作の同期タイミングとカム制御データの反映タイミングを設定します。
設定値範囲以外を設定した場合は、開始モード範囲外(エラーコード: 1AB5H)となります。
MC_CamIn(カム動作開始)実行後、1サイクル現在値(InputPerCycle)が同期すると制御中(Active)がTRUEになり、出力値(OutputData)が同期すると同期中(InSync)がTRUEになります。



設定値	内容
1: 絶対(mcAbsolute)	主軸の位置が主軸同期開始位置(MasterSyncPosition)を通過すると実行します。 主軸追従距離(MasterStartDistance)は使用しません。 実行指令(Execute)立ち上げ後、主軸の累積現在位置が主軸同期開始位置(MasterSyncPosition)を通過した後に従軸が動き出します。 主軸の累積現在位置が主軸同期開始位置(MasterSyncPosition)を通過したかのチェックは、従軸の状態が「7: 同期運転中(SynchronizedMotion)」になったタイミングから開始します。 主軸の累積現在位置が主軸同期開始位置(MasterSyncPosition)を通過 主軸の累積現在位置 MasterSyncPosition InputPerCycle Reference 0.0 1000000.0 OutputData 従軸の累積現在位置 従軸の状態 (AxisName.Md.AxisStatus) 4: Standstill 7: SynchronizedMotion 従軸のプロファイルID 0 1 Execute Busy Active InSync EndOfProfile

■主軸データソース選択(MasterValueSource)

従軸(Slave)が単軸同期制御を実行する主軸(Master)の位置の種別を設定します。

主軸(Master)と従軸(Slave)の演算周期が異なる場合やFBの実行順により、動作が変わる場合があります。

実軸以外の軸種別を主軸(Master)に設定して主軸データソース選択(MasterValueSource)にフィードバック値を設定した場合は、指令現在値と同じ値で動作します。

設定値	内容
1: 指令現在値(mcSetValue)	前回の演算周期における主軸の指令位置を使用します。
2: フィードバック値(mcActualValue)	前回の演算周期における主軸のフィードバック位置を使用します。
101: 最新指令現在値(mcLatestSetValue)	今回の演算周期における主軸の指令位置を使用します。
102: 最新フィードバック値(mcLatestActualValue)	今回の演算周期における主軸のフィードバック位置を使用します。

Point

主軸データソース選択(MasterValueSource)に「1: 指令現在値(mcSetValue)」, 「101: 最新指令現在値(mcLatestSetValue)」を設定し、主軸がサーボアラームや緊急停止によってサーボOFFとなった場合、値の変化量が大きくなる場合があります。主軸データソース選択(MasterValueSource)に「2: フィードバック値(mcActualValue)」, 「102: 最新フィードバック値(mcLatestActualValue)」を設定することで防ぐことができます。

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

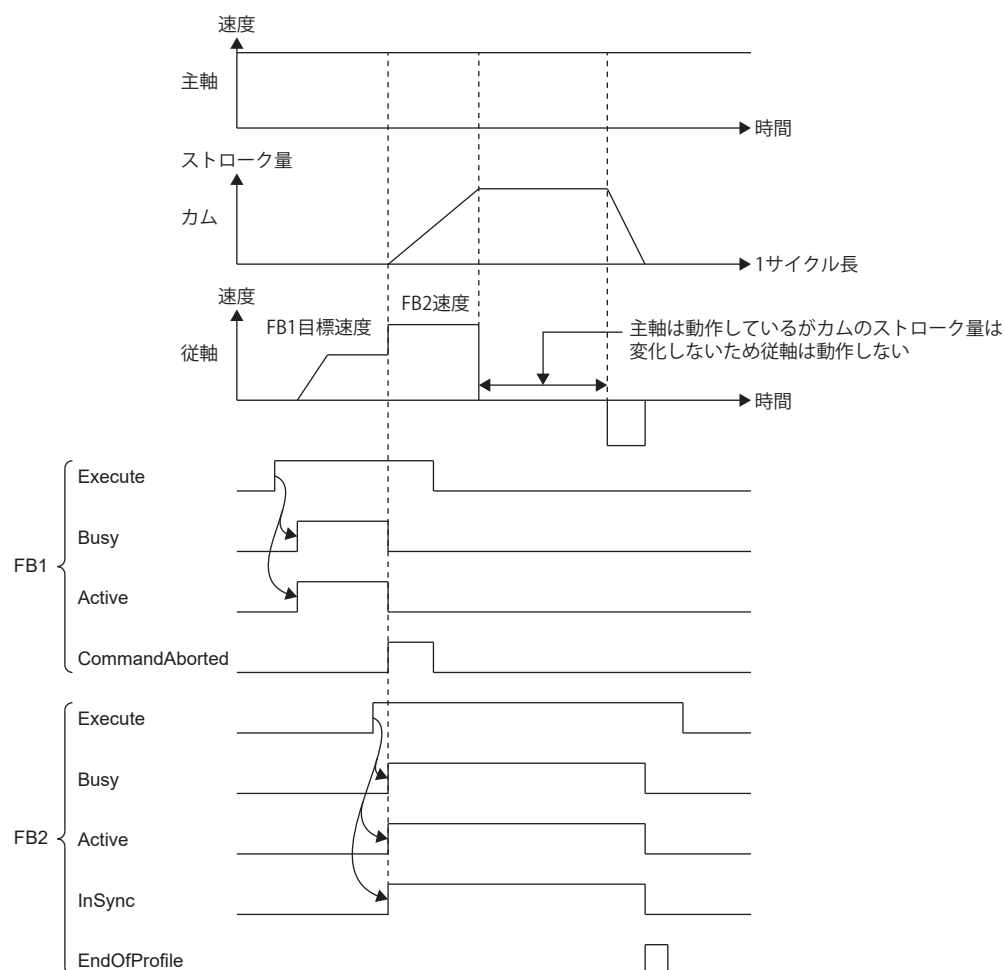
MC_CamIn(カム動作開始)では、下記のバッファモードが設定できます。

MC_CamIn(カム動作開始)の速度は、同期しているFB(主軸)に追従しているため、FB切換え時に主軸速度と設定したカムデータに基づき速度が即時変わります。各モードの切換え条件を下記に示します。

設定値	内容	切換え条件
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。	常時
1: Buffered(mcBuffered)	制御中のFBに次のFBをバッファリングします。 制御中のFBがすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFB完了にて、バッファリングFBを順次実行します。	カムサイクル完了(EndOfProfile)がTRUEのとき

例

実行中のFB1に連結起動FB2(従軸(Slave): MC_CamIn(カム動作開始))を開始モード(StartMode)の「0: 即時(mcImmediate)」を設定し、バッファモード(BufferMode)を「0: 常時(Aborting)」とした場合



FB2の速度が発生するのは、同期中(InSync)がTRUEになってからとなります。

開始モード(StartMode)が「0: 即時(mcImmediate)」以外の場合、FB起動後すぐに同期中(InSync)がTRUEにならないため、実行中(Busy)、制御中(Active)、および同期中(InSync)がTRUEになるまでの間は速度0となります。

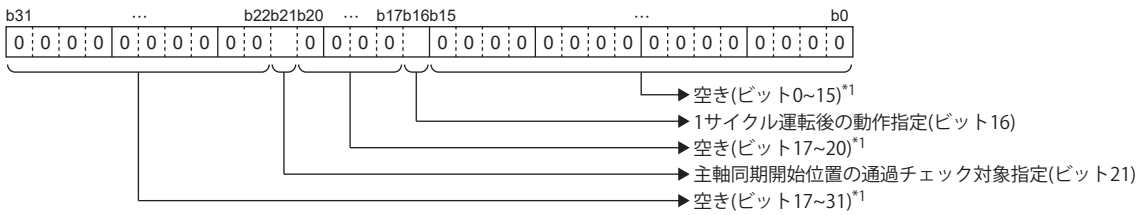
Point

多重起動(バッファモード)の詳細については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

■オプション(Options)

MC_CamIn(カム動作開始)で使用する機能オプションをビット指定で設定します。
ビット指定で設定する内容を下記に示します。

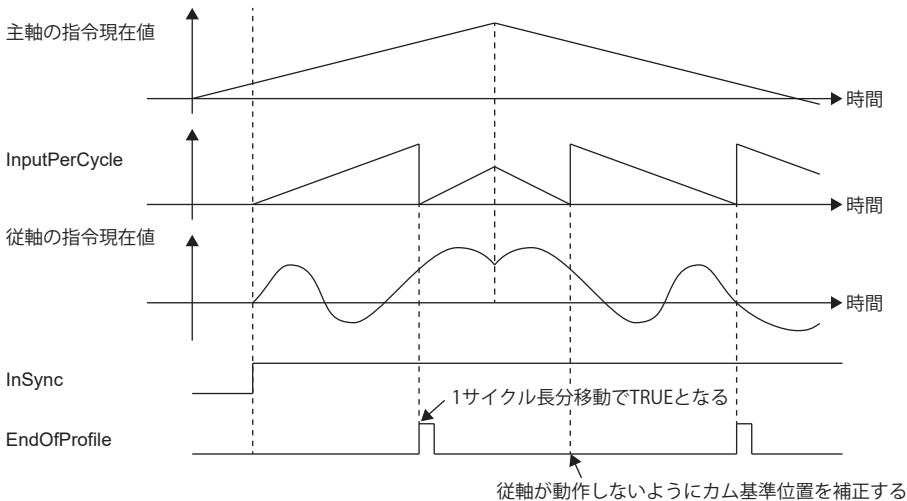


*1 空きには「0」を設定してください。「0」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

ビット	名称	内容
16	1サイクル運転後の動作指定	MC_CamTableSelect(カムテーブル選択)の繰り返し動作(Periodic)に「FALSE: 単発動作」を設定したときの1サイクル運転後の動作を設定します。 • 0: 終了 • 1: 再起動待ち
21	主軸同期開始位置の通過チェック対象指定	主軸同期開始位置(MasterSyncPosition)の通過チェックの対象を設定します。 • 0: 主軸の累積現在位置、またはフィードバック位置を累積位置換算したもの • 1: 主軸の指令現在位置、またはフィードバック位置

■カムサイクル完了(EndOfProfile)

制御中(Active)がTRUEになった後、1サイクル長分移動するたびに、FBを呼び出すPOU(プログラム部品)の実行周期の1周期のみ、TRUEになります。



■1サイクル現在値(InputPerCycle)

初期値は、0.0です。

制御中(Active)がTRUEになった後、主軸(Master)の移動量を下記のとおり反映します。

- 1サイクル現在値 = (主軸の累積移動量 - 主軸オフセット + 1サイクル現在値変更値) MOD 1サイクル長
- モニタ値の範囲は、カムテーブルに設定した1サイクル長の下記の範囲となります。
- $0 \leq \text{MC\_CamIn(カム動作開始)の1サイクル現在値(InputPerCycle)} < (1\text{サイクル長} \times \text{主軸係数})$

■基準値(Reference)

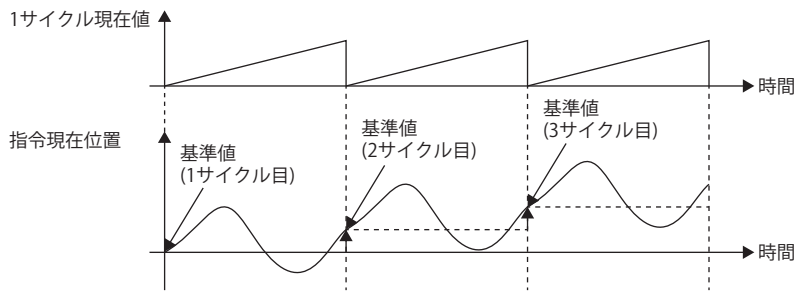
MC_CamIn(カム動作開始)実行時は、同期中(InSync)がTRUEになる指令現在位置を基準に、開始するカム位置(1サイクル現在値)から算出します。

- カムテーブル中の初期ストローク量分減算するため、初期値の計算式は下記となります。

従軸絶対座標 (SlaveAbsolute)	計算式
FALSE(相対座標) TRUE(絶対座標)	基準値 = 同期開始時の従軸位置 - (同期中(InSync)がTRUEになったときのカム位置(1サイクル現在値)に基づくストローク値 + 初期ストローク量) × 従軸係数
	基準値 = 同期中(InSync)がTRUEになったときの指令現在位置=0に相当する累積現在位置

- 往復カム動作と送りカム動作の場合の基準値は下記となります。

動作	内容
往復カム動作、または従軸絶対座標(SlaveAbsolute)が「TRUE(絶対座標)」のとき	基準値を更新しません。
送りカム動作 かつ 従軸絶対座標(SlaveAbsolute)が「FALSE(相対座標)」のとき	基準値は下記のように算出します。 基準値 = (元の基準値 + (最終点目のストローク値 - 1点目のストローク値) × 従軸係数)



- 基準値は下記のタイミングで更新します。

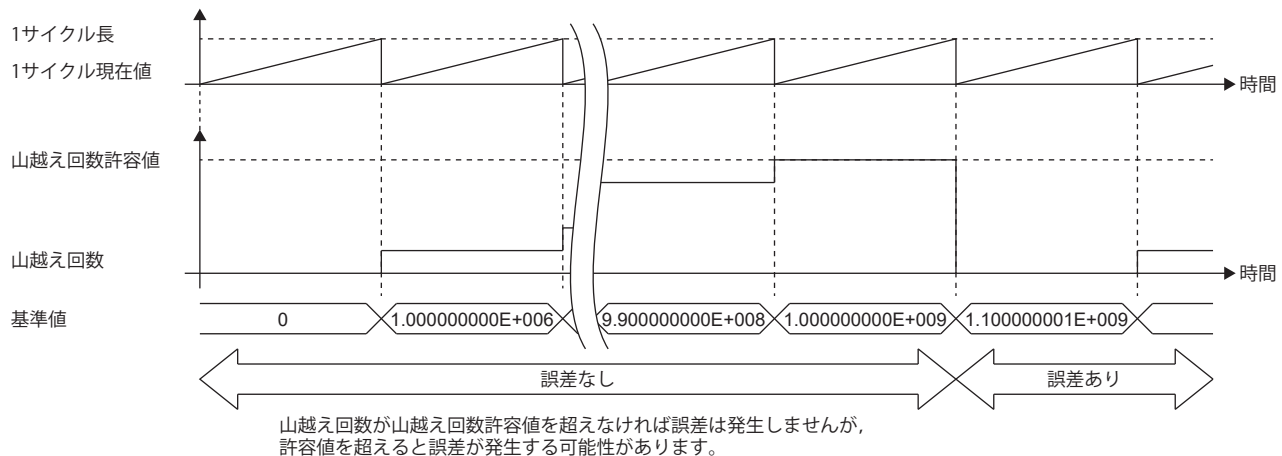
更新タイミング	基準値の算出式
1サイクル現在値(InputPerCycle)がアドレス増加方向にカムテーブルの最終点目を通過したとき	元の基準値 + ((最終点目のストローク値 - 1点目のストローク値) × 従軸係数)
1サイクル現在値(InputPerCycle)がアドレス減少方向にカムテーブルの1点目を通過したとき	元の基準値 - ((最終点目のストローク値 - 1点目のストローク値) × 従軸係数)
1サイクル現在値を変更したとき (MCv_ChangeCycle(1サイクル現在値変)の完了(Done)がTRUEのとき)	元の基準値 - (変更後の1サイクル現在値に相当するストローク値 × 従軸係数)

<基準値の誤差について>

下記いずれかの動作を行うと基準値に誤差が発生する可能性があります。

- 送りカムかつ同期中(InSync)がTRUEになった後、カムサイクル完了(EndOfProfile)のTRUEの出力回数が山越え回数許容値を超えたとき。^{*1}
- 送りカムかつ下記の制御パラメータに小数を含んだとき。
 - (最終点目のストローク値 - 1点目のストローク値)
 - 従軸係数
- 1サイクル現在値変更を相対指定で繰り返したとき。

^{*1}: 山越え回数が山越え回数許容値を越えなければ誤差は発生しませんが、山越え回数許容値を越えると誤差が発生する可能性があります。(演算誤差の有無に関わらず動作を継続します。)



基準値の誤差が大きくなった場合、同期状態を解除するか、絶対位置指定の現在値変更を実行してリセットしてください。

山越え回数許容値は、下記の計算式から計算した可変の値となります。

山越え回数許容値 = 山越え回数限界値 / ((最終点目の出力値 - 1点目の出力値) / 従軸係数)

^{*}: 山越え回数限界値は、位置決め範囲上限値である「10000000000」となります。

(例)

1点目の出力値を「0」、最終点目の出力値を「1000000」、出力係数を「10」とした場合
山越え回数許容値は「100000」となります。

モニタ値の範囲は、下記になります。

- 従軸のリングカウンタ下限値 ≤ 基準値 < 従軸のリングカウンタ上限値

■出力値(OutputData)

初期値はカムテーブル中の初期ストローク量となります。

同期中(InSync)がTRUEのとき、下記のように算出します。

- 出力値 = 基準値 + ((ストローク量 × 1サイクル現在値に対応するストローク比) × 従軸係数) + 従軸オフセット

または

- 出力値 = 基準値 + ((1サイクル現在値に対応する出力値) × 従軸係数) + 従軸オフセット

モニタ値の範囲は、従軸のリングカウンタ下限値 ≤ 基準値 < 従軸のリングカウンタ上限値となります。

注意事項

- 主軸オフセット(MasterOffset)、従軸オフセット(SlaveOffset)、主軸係数(MasterScaling)、従軸係数(SlaveScaling)、カムテーブルID(CamTableID)を変更すると、制御開始時や制御変更時に従軸が急激に移動し、機械に衝撃を与えることがあります。設定値や変更タイミングは十分検討してください。
- カム制御中の軸をエンジニアリングツールでモニタするときの位置、速度などの単位は、軸の単位設定に準じます。演算プロファイルで設定した1サイクル長やストローク量の単位は使用しません。

プログラム例

カム動作開始指令(bCamInStart)をTRUEにし、JOG運転で始動させる軸2(Axis0002)を主軸に、下記の設定により演算プロファイル(カムデータ1)に従って従軸(Axis0001)のカム動作を行うプログラム例を下記に示します。

・設定

項目	設定値
JOG運転用データ設定	速度
	1000.0
	加速度
	5000.0
	減速度
	5000.0
	ジャーク
	0.0

■軸

軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1
2	Axis0002	AXIS_REF	軸2

■演算プロファイル

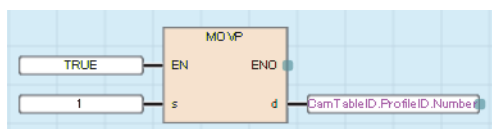
プロファイルID	ラベル名	データ型	コメント
1	ProfileData0001	MC_CAM_REF	カムデータ1

■使用するラベル

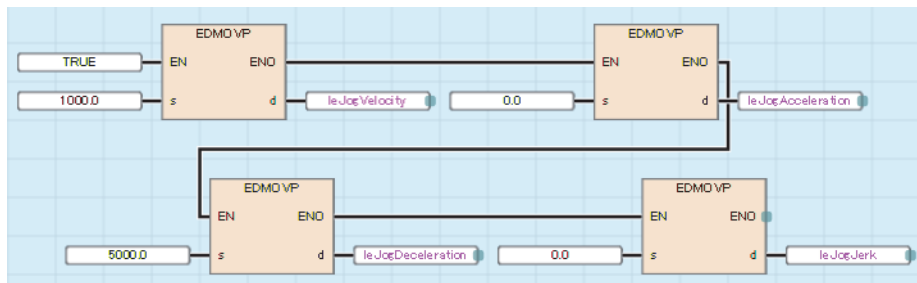
ラベル名	データ型	コメント
MC_CamIn_1	MC_CamIn	カム動作開始FB
bCamInStart	ビット	カム動作開始指令
CamTableID	MC_CAM_ID	カムテーブルID
bInSync	ビット	同期中
bCamBusy	ビット	実行中
bCamActive	ビット	制御中
bCamCommandAborted	ビット	実行中断
bCamError	ビット	エラー
uwCamErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
bEndOfProfile	ビット	カムサイクル完了
MCv_Jog_1	MCv_Jog	JOG運転FB
leJogVelocity	倍精度実数	JOG速度
leJogAcceleration	倍精度実数	JOG加速度
leJogDeceleration	倍精度実数	JOG減速度
leJogJerk	倍精度実数	JOGジャーク
bJogDone	ビット	実行完了
bJogBusy	ビット	実行中
bJogActive	ビット	制御中
bJogCommandAborted	ビット	実行中断
bJogError	ビット	エラー
uwJogErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

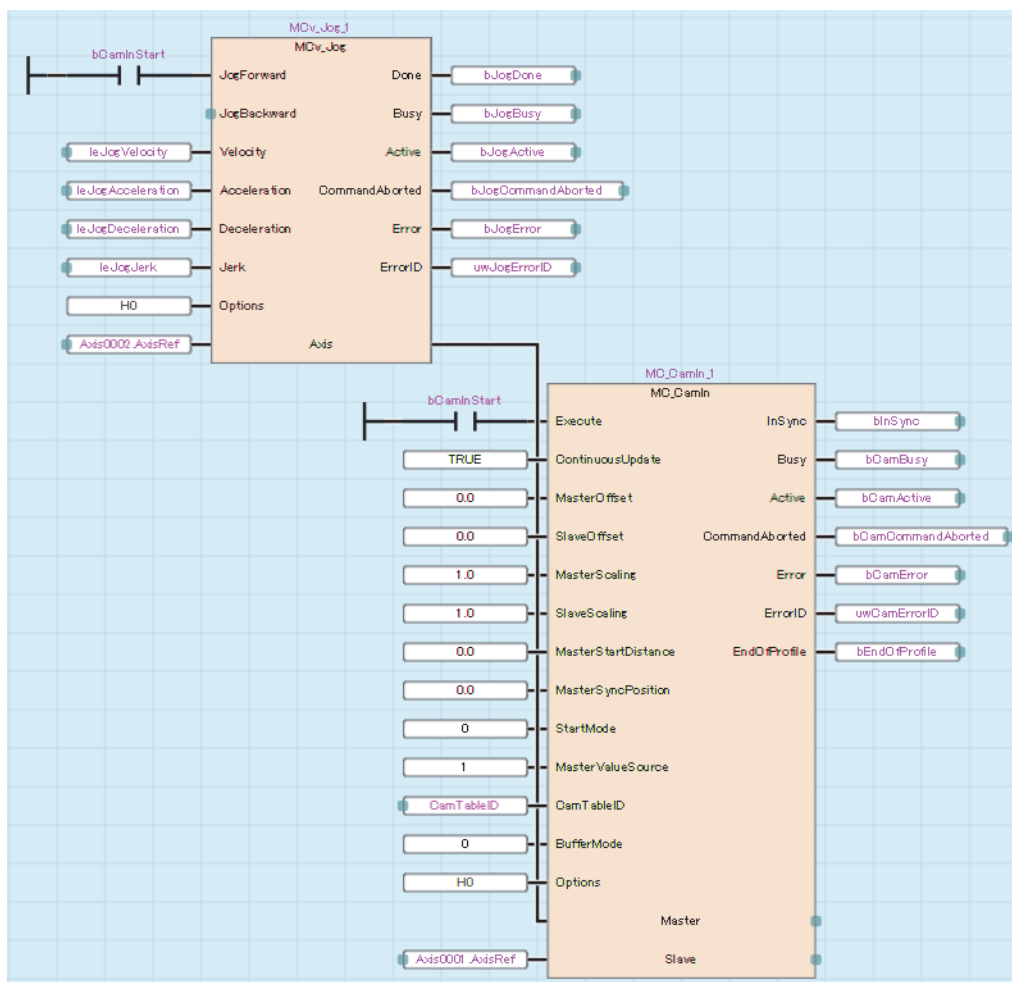
- 演算プロファイル(カムデータ)設定



- JOG運転用データ設定



- JOG運転/カム動作開始



■STプログラム

//-----演算プロファイル(カムデータ)設定-----

CamTableID.ProfileID.Number:= 1;

//-----JOG運転用データ設定-----

leJogVelocity:= 1000.0;

leJogAcceleration:= 5000.0;

leJogDeceleration:= 5000.0;

leJogJerk:= 0.0;

//-----JOG運転-----

MCv_Jog_1(

Axis:= Axis0002.AxisRef ,

JogForward:= bCamInStart ,

Velocity:= leJogVelocity ,

Acceleration:= leJogAcceleration ,

Deceleration:= leJogDeceleration ,

Jerk:= leJogJerk ,

Options:= H00000000 ,

Done=> bJogDone ,

Busy=> bJogBusy ,

Active=> BJogActive ,

CommandAborted=> bJogCommandAborted ,

Error=> bJogError ,

ErrorID=> uwJogErrorID

);

//-----カム動作開始-----

MC_CamIn_1(

Master:= Axis0002.AxisRef ,

Slave:= Axis0001.AxisRef ,

Execute:= bCamInStart ,

ContinuousUpdate:= TRUE ,

MasterOffset:= 0.0 ,

SlaveOffset:= 0.0 ,

MasterScaling:= 1.0 ,

SlaveScaling:= 1.0 ,

MasterStartDistance:= 0.0 ,

MasterSyncPosition:= 0.0 ,

StartMode:= MC_START_MODE__mcImmediate ,

MasterValueSource:= MC_SOURCE__mcSetValue ,

CamTableID:= CamTableID ,

BufferMode:= MC_BUFFER_MODE__mcAborting ,

Options:= H00000000 ,

InSync=> blnSync ,

Busy=> bCamBusy ,

Active=> bCamActive ,

CommandAborted=> bCamCommandAborted ,

Error=> bCamError ,

ErrorID=> uwCamErrorID ,

EndOfProfile=> bEndOfProfile

);

42.15 ギア動作開始

MC_GearIn

指定したギア比に従いギア動作を開始します。

ラダー	FBD/LD	ST
MC_GearIn DUT: MasterMaster :DUT DUT: SlaveSlave :DUT B: ExecuteInGear :B B: ContinuousUpdateBusy :B D: RatioNumeratorActive :B UD: RatioDenominatorCommandAborted :B ENUM: MasterValueSourceError :B L: AccelerationErrorID :UW L: Deceleration L: Jerk ENUM: BufferMode UD: Options	MC_GearIn ExecuteInGear ContinuousUpdateBusy RatioNumeratorActive RatioDenominatorCommandAborted MasterValueSourceError AccelerationErrorID Deceleration Jerk BufferMode Options Master Slave	<pre>MC_GearIn(Master:= ?AXIS_REF? , Slave:= ?AXIS_REF? , Execute:= ?BOOL? , ContinuousUpdate:= ?BOOL? , RatioNumerator:= ?DINT? , RatioDenominator:= ?DWORD? , MasterValueSource:= ?INT? , Acceleration:= ?LREAL? , Deceleration:= ?LREAL? , Jerk:= ?LREAL? , BufferMode:= ?INT? , Options:= ?DWORD? , InGear=> ?BOOL? , Busy=> ?BOOL? , Active=> ?BOOL? , CommandAborted=> ?BOOL? , Error=> ?BOOL? , ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Master	主軸	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📄 1378ページ AxisName.AxisRef.(軸情報)
Slave	従軸	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📄 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

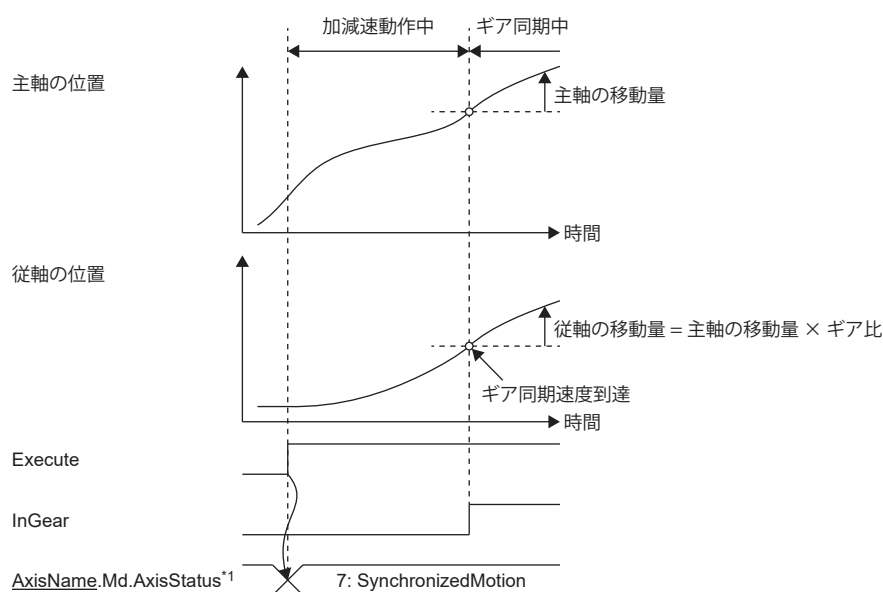
入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_GearIn(ギア動作開始)を実行します。
Continuous Update	連続更新	BOOL	起動時	TRUE, FALSE	FALSE	連続して、ギア比分子(RatioNumerator)、ギア比分母(RatioDenominator)、加速度(Acceleration)、減速度(Deceleration)の変更を有効にするか、無効にするかを設定します。 ・FALSE: 無効 ・TRUE: 有効
RatioNumerator	ギア比分子	DINT	起動時/再起動可/連続更新可	-2147483648~2147483647	1	主軸(Master)の分子の値を設定します。詳細は、下記を参照してください。 1736ページ ギア比分子(RatioNumerator)/ギア比分母(RatioDenominator)
RatioDenominator	ギア比分母	DWORD(UDINT)	起動時/再起動可/連続更新可	1~2147483647	1	主軸(Master)の分母の値を設定します。詳細は、下記を参照してください。 1736ページ ギア比分子(RatioNumerator)/ギア比分母(RatioDenominator)
MasterValue Source	主軸データソース選択	INT (MC_SOURCE)	起動時	1, 2, 101, 102	1	主軸(Master)のデータソースを設定します。 ・1: 指令現在(mcSetValue) ・2: フィードバック値(mcActualValue) ・101: 最新指令現在値(mcLatestSetValue) ・102: 最新フィードバック値(mcLatestActualValue) 詳細は、下記を参照してください。 1738ページ 主軸データソース選択(MasterValueSource)
Acceleration	加速度	LREAL	起動時/再起動可/連続更新可	0.0000, 0.0001~2147483647.0	0.0	加速動作の加速度を設定します。ギア同期速度に到達するとギア比到達(InGear)がTRUEになり、従軸(Slave)は主軸(Master)の速度にギア比換算した速度で制御します。
Deceleration	減速度	LREAL	起動時/再起動可/連続更新可	0.0000, 0.0001~2147483647.0	0.0	減速動作の減速度を設定します。ギア同期速度に到達するとギア比到達(InGear)がTRUEになり、従軸(Slave)は主軸(Master)の速度にギア比換算した速度で制御します。
Jerk	ジャーク	LREAL	起動時	0.0000, 0.0001~2147483647.0	0.0	加減速動作開始時のジャークを設定します。
BufferMode	バッファモード	INT (MC_BUFFER_MODE)	起動時	0, 1	0	バッファモードを設定します。 ・0: Aborting(mcAborting) ・1: Buffered(mcBuffered) 詳細は、下記を参照してください。 1738ページ バッファモード(BufferMode)
Options	オプション	DWORD(HEX)	起動時	00000000H	00000000H	「00000000H」を設定してください。 *: 「00000000H」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

■出力変数

出力変数	名称	データ型	初期値	内容
InGear	ギア比到達	BOOL	FALSE	ギア同期速度へ到達したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MC_GearIn(ギア動作開始)を実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	従軸(Slave)を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	MC_GearIn(ギア動作開始)の実行が中断したときに、TRUEになります。 実行指令(Execute)がFALSEになると、FALSEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 📖使用するコントローラのユーザーズマニュアル

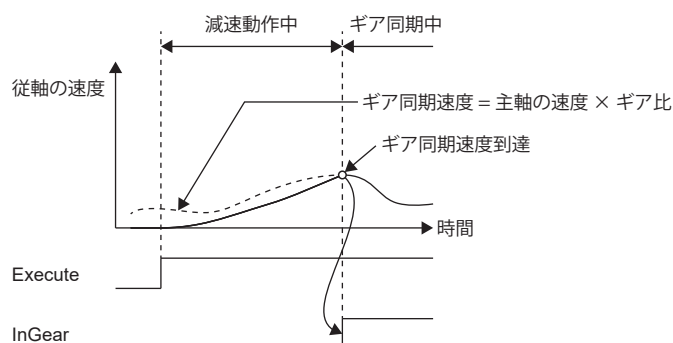
機能

- MC_GearIn(ギア動作開始)は、ギア比分子(RatioNumerator)、ギア比分母(RatioDenominator)、主軸データソース選択(MasterValueSource)、加速度(Acceleration)、減速度(Deceleration)、ジャーク(Jerk)、バッファモード(BufferMode)を設定し、ギア動作を開始します。
- 動作を終了する場合は、MC_Stop(強制停止)にて行います。
- FB実行後、従軸(Slave)は主軸(Master)の速度をギア比換算した値をギア同期速度として、ギア同期速度に到達するまで加減速動作を行います。ギア同期速度到達後、ギア比到達(InGear)がTRUEになり、従軸(Slave)は主軸(Master)の速度にギア比換算した速度で制御します。



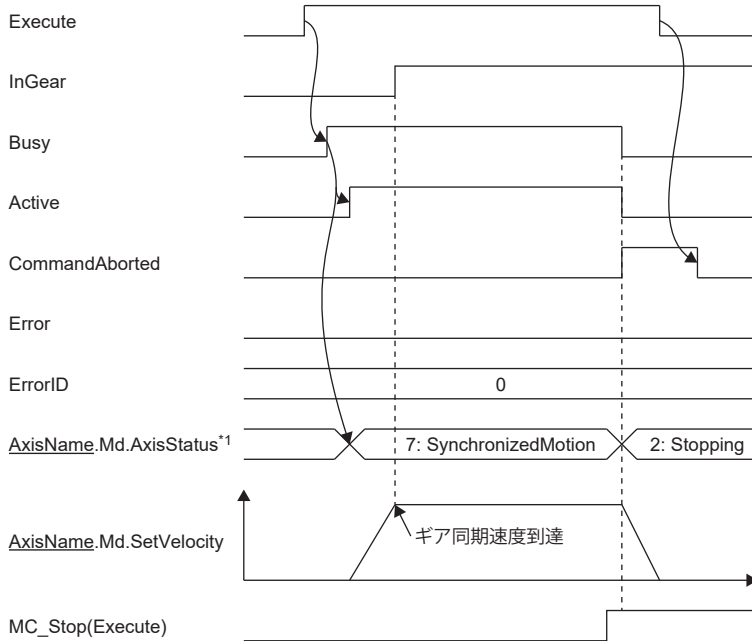
*1 従軸(Slave)の軸状態です。

- 加減速動作中に主軸(Master)の速度が変動した場合、ギア同期速度も更新します。



■タイミングチャート

・正常完了の場合



*1 従軸(Slave)の軸状態です。

・異常完了の場合

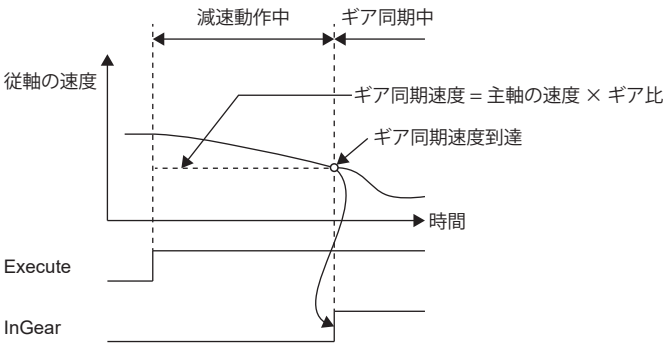
異常完了時のタイミングチャートについては、下記を参照してください。

1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■ギア同期速度に到達するまでの加減速動作

- MC_GearIn(ギア動作開始)の加減速方式は、加減速度指定方式です。
- 動作開始時の従軸(Slave)の速度とギア同期速度によって、下記4パターンの加減速動作を行います。加減速動作中にギア同期速度を更新しても、加減速動作のパターンは変更しません。

従軸(Slave)の速度と目標速度	加減速動作
従軸(Slave)の速度 < 0 < ギア同期速度 または 従軸(Slave)の速度 > 0 > ギア同期速度	速度0まで減速動作を行い、速度0からギア同期速度まで加速動作を行います。
従軸(Slave)の速度(絶対値) < ギア同期速度(絶対値)	ギア同期速度まで加速動作を行います。

従軸(Slave)の速度と目標速度	加減速動作
従軸(Slave)の速度(絶対値) > ギア同期速度(絶対値)	ギア同期速度まで減速動作を行います。 
従軸(Slave)の速度 = ギア同期速度	加減速動作をしません。 動作開始後、即座にギア比到達(InGear)がTRUEになり、従軸(Slave)は主軸(Master)の速度にギア比換算した速度で制御します。

■ギア比分子(RatioNumerator)/ギア比分母(RatioDenominator)

ギア比分子(RatioNumerator)とギア比分母(RatioDenominator)は、組み合わせて設定します。

ギア比分子(RatioNumerator)は、主軸(Master)の速度を変換して伝達するときの分子の値を設定します。

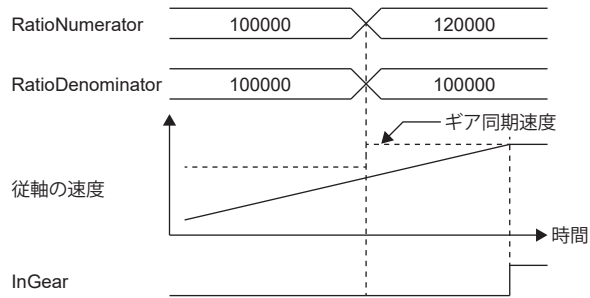
ギア比分母(RatioDenominator)は、主軸(Master)の移動量を変換して合成するときの分母の値を設定します。

下記のように速度を変換します。

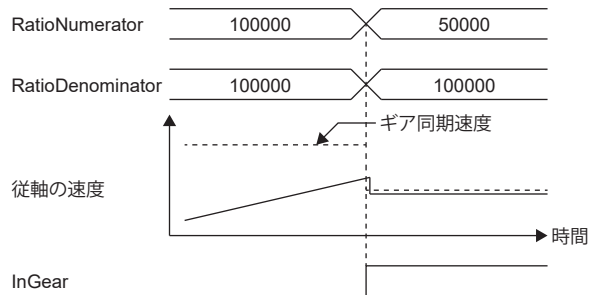
$$\text{変換後の主軸(Master)速度} = \text{変換前の主軸(Master)速度} \times \frac{\text{ギア比分子(RatioNumerator)}}{\text{ギア比分母(RatioDenominator)}}$$

- ギア比分母(RatioDenominator)には正の値を設定してください。
- ギア比分子(RatioNumerator)の設定値に負の値を設定することで、速度を逆転して伝達できます。加減速動作中にギア比を変更した場合、ギア同期速度が変化します。そのため、ギア同期速度の変化によっては、従軸(Slave)の速度がダイレクトに変化してギア動作を開始する場合があります。
- ギア比分子(RatioNumerator)の設定値に「0」を設定することで、変換後の主軸(Master)の速度を「0」にできます。
- 加減速動作中にギア比を変更した場合、目標速度が変化します。そのため、目標速度の変化によっては、従軸(Slave)の速度がダイレクトに変化してギア動作を開始する場合があります。
 - 加減速動作中でのギア比変更

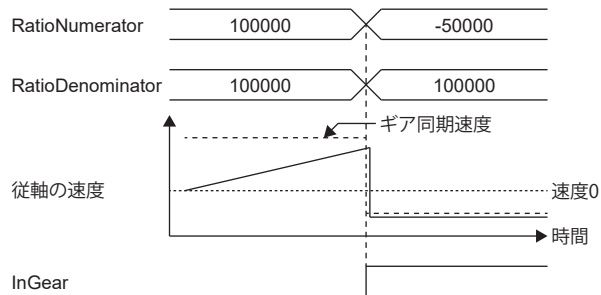
- ギア比増加(ギア同期速度を従軸(Slave)の速度より高い値にする)



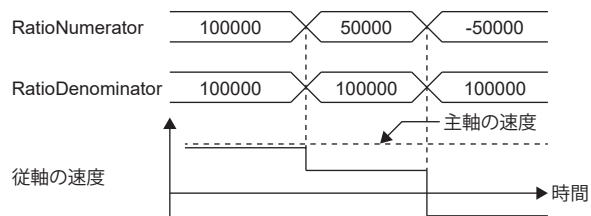
- ギア比減少(ギア同期速度を従軸(Slave)の速度より低い値にする)



- ギア比反転(ギア同期速度を「0」より低い値にする)



- ギア同期中にギア比を変更した場合、従軸(Slave)の速度がダイレクトに変化します。
- ギア同期中でのギア比変更



■主軸データソース選択(MasterValueSource)

従軸(Slave)が単軸同期制御を実行する主軸(Master)の位置の種別を設定します。

主軸(Master)と従軸(Slave)の演算周期が異なる場合やFBの実行順により、動作が変わる場合があります。

実軸以外の軸種別を主軸(Master)に設定して主軸データソース選択(MasterValueSource)にフィードバック値を設定した場合は、指令現在値と同じ値で動作します。

設定値	内容
1: 指令現在値(mcSetValue)	前回の演算周期における主軸の指令位置を使用します。
2: フィードバック値(mcActualValue)	前回の演算周期における主軸のフィードバック位置を使用します。
101: 最新指令現在値(mcLatestSetValue)	今回の演算周期における主軸の指令位置を使用します。
102: 最新フィードバック値(mcLatestActualValue)	今回の演算周期における主軸のフィードバック位置を使用します。

Point

主軸データソース選択(MasterValueSource)に「1: 指令現在値(mcSetValue)」、「101: 最新指令現在値(mcLatestSetValue)」を設定し、主軸がサーボアラームや緊急停止によってサーボOFFとなった場合、値の変化量が大きくなる場合があります。主軸データソース選択(MasterValueSource)に「2: フィードバック値(mcActualValue)」、「102: 最新フィードバック値(mcLatestActualValue)」を設定することで防ぐことができます。

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

MC_GearIn(ギア動作開始)では、下記のバッファモードが設定できます。

設定値	内容
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。
1: Buffered(mcBuffered)	制御中のFBに次のFBをバッファリングします。 制御中のFBがすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFB完了にて、バッファリングFBを順次実行します。

Point

多重起動(バッファモード)の詳細については、下記を参照してください。

 使用するコントローラのユーザーズマニュアル

注意事項

- ギア比を変更した場合、従軸の速度がダイレクトに変化します。速度変化をスムージング処理したい場合は、MCv_SmoothingFilter(スムージングフィルタ)と合わせて使用してください。
- 加減速動作中に変換後の主軸(Master)の速度が加減速時間上限値を超える場合、加速時間制限オーバ警告(イベントコード: 00D04H)、または減速時間制限オーバ警告(イベントコード: 00D05H)となり、加減速動作を停止し、警告検出時の速度で動作を継続します。上記警告が発生した場合は、変換後の主軸(Master)の速度、加速度(Acceleration)、または減速度(Deceleration)を調整して加減速時間上限値を超えないように設定してください。

プログラム例

ギア動作開始指令(bGearInStart)をTRUEにし、JOG運転で始動させる軸1(Axis0001)を主軸に、下記の設定により指定したギア比に従って従軸(Axis0002)のギア動作を行うプログラム例を下記に示します。

・設定

項目	設定値
ギア動作データ設定	加速度
	減速度
	ジャーク
JOG運転用データ設定	速度
	加速度
	減速度
	ジャーク

■軸

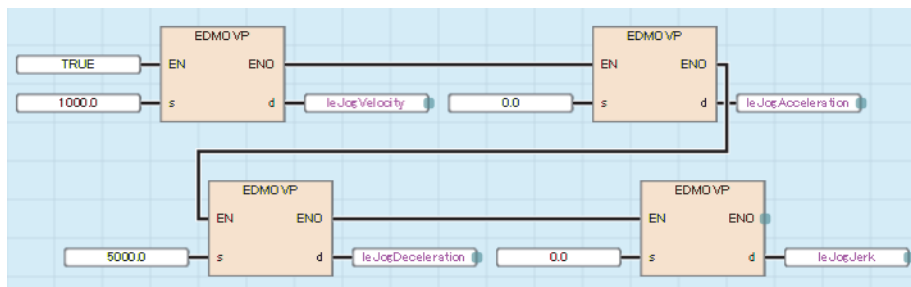
軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1
2	Axis0002	AXIS_REF	軸2

■使用するラベル

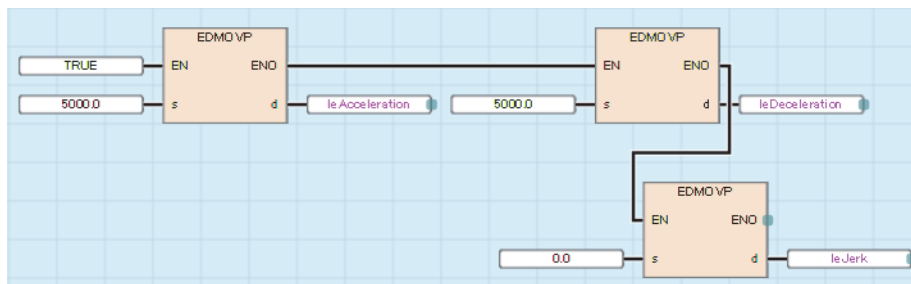
ラベル名	データ型	コメント
MC_GearIn_1	MC_GearIn	ギア動作開始FB
bGearInStart	ビット	ギア動作開始指令
leAcceleration	倍精度実数	加速度
leDeceleration	倍精度実数	減速度
leJerk	倍精度実数	ジャーク
bInGear	ビット	同期中
bGearBusy	ビット	実行中
bGearActive	ビット	制御中
bGearCommandAborted	ビット	実行中断
bGearError	ビット	エラー
uwGearErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
MCv_Jog_1	MCv_Jog	JOG運転FB
leJogVelocity	倍精度実数	JOG速度
leJogAcceleration	倍精度実数	JOG加速度
leJogDeceleration	倍精度実数	JOG減速度
leJogJerk	倍精度実数	JOGジャーク
bJogDone	ビット	実行完了
bJogActive	ビット	制御中
bJogCommandAborted	ビット	実行中断
bJogError	ビット	エラー
uwJogErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

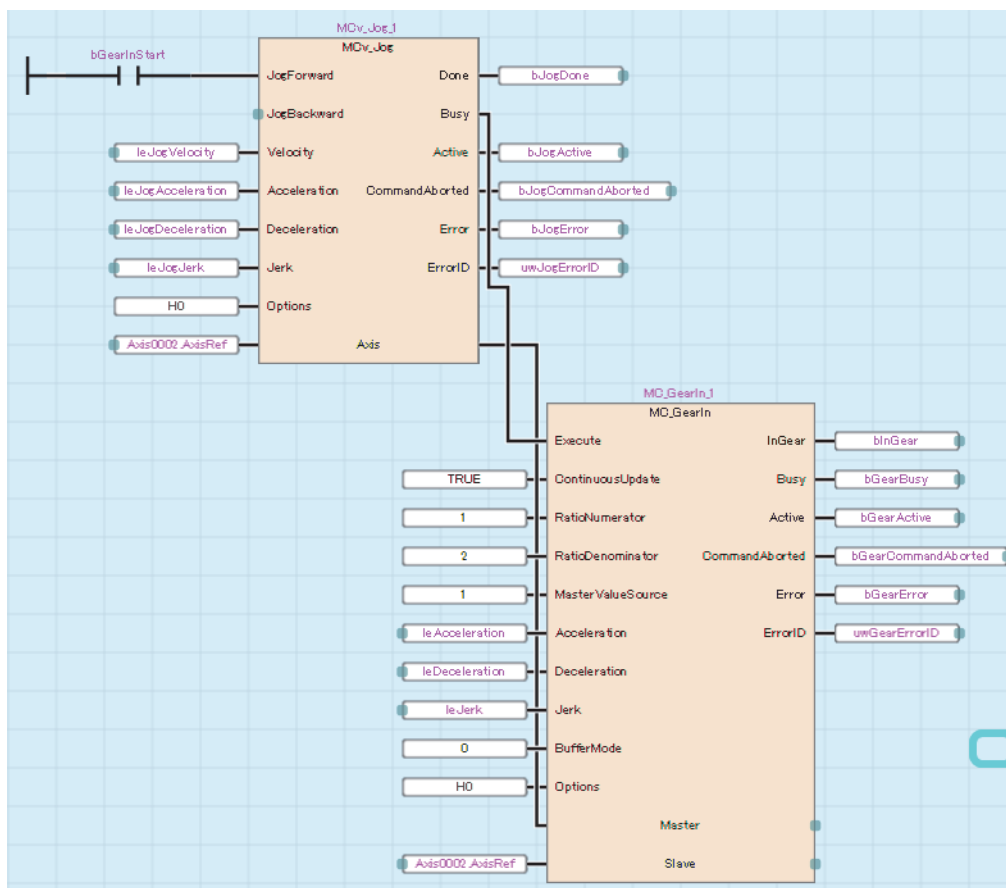
• JOG運転用データ設定



• ギア動作データ設定



• JOG運転/ギア動作開始



■STプログラム

//-----JOG運転用データ設定-----

```
leJogVelocity:= 1000.0;
leJogAcceleration:= 5000.0;
leJogDeceleration:= 5000.0;
leJogJerk:= 0.0;
```

//-----ギア動作用データ設定-----

```
leAcceleration:= 5000.0;
leDeceleration:= 5000.0;
leJerk:= 0.0;
```

//-----JOG運転-----

```
MCv_Jog_1(
  Axis:= Axis0001.AxisRef ,
  JogForward:= bGearInStart ,
  Velocity:= leJogVelocity ,
  Acceleration:= leJogAcceleration ,
  Deceleration:= leJogDeceleration ,
  Jerk:= leJogJerk ,
  Options:= H00000000 ,
  Done=> bJogDone ,
  Busy=> bJogBusy ,
  Active=> BJogActive ,
  CommandAborted=> bJogCommandAborted ,
  Error=> bJogError ,
  ErrorID=> uwJogErrorID
);
```

//-----ギア動作開始-----

```
MC_GearIn_1(
  Master:= Axis0001.AxisRef ,
  Slave:= Axis0002.AxisRef ,
  Execute:= bJogBusy ,
  ContinuousUpdate:= TRUE ,
  RatioNumerator:= 1 ,
  RatioDenominator:= 2 ,
  MasterValueSource:= MC_SOURCE__mcSetValue ,
  Acceleration:= leAcceleration ,
  Deceleration:= leDeceleration ,
  Jerk:= leJerk ,
  BufferMode:= MC_BUFFER_MODE__mcAborting ,
  Options:= H00000000 ,
  InGear=> bInGear ,
  Busy=> bGearBusy ,
  Active=> bGearActive ,
  CommandAborted=> bGearCommandAborted ,
  Error=> bGearError ,
  ErrorID=> uwGearErrorID
);
```

42.16 加減算位置決め

MC_CombineAxes

指定した主軸2軸の移動量を加減算した値を指令位置とし、位置決めを行います。

ラダー	FBD/LD	ST
		<pre>MC_CombineAxes(Master1:= ?AXIS_REF?, Master2:= ?AXIS_REF?, Slave:= ?AXIS_REF?, Execute:= ?BOOL?, ContinuousUpdate:= ?BOOL?, CombineMode:= ?INT?, GearRatioNumeratorM1:= ?DINT?, GearRatioDenominatorM1:= ?DWORD?, GearRatioNumeratorM2:= ?DINT?, GearRatioDenominatorM2:= ?DWORD?, MasterValueSourceM1:= ?INT?, MasterValueSourceM2:= ?INT?, BufferMode:= ?INT?, Options:= ?DWORD?, InSync=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Master1	主軸1	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📖 1378ページ AxisName.AxisRef.(軸情報)
Master2	主軸2	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📖 1378ページ AxisName.AxisRef.(軸情報)
Slave	従軸	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📖 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMC_CombineAxes(加減算位置決め)を実行します。
Continuous Update	連続更新	BOOL	起動時	TRUE, FALSE	FALSE	連続して、加減算方法選択 (CombineMode)、主軸1ギア比分子 (GearRatioNumeratorM1)、主軸1ギア比分母 (GearRatioDenominatorM1)、主軸2ギア比分子 (GearRatioNumeratorM2)、主軸2ギア比分母 (GearRatioDenominatorM2)の変更を有効にするか、無効にするかを設定します。 • FALSE: 無効 • TRUE: 有効

入力変数	名称	データ型	取込	有効範囲	初期値	内容
CombineMode	加減算方法選択	INT (MC_COMBINE_MODE)	起動時/再起動可/ 連続更新可	0, 1	1	主軸1(Master1)と主軸2(Master2)の移動量の合成方法を設定します。 0: 2つの入力軸の位置を加算 (mcAddAxes) 1: 2つの入力軸の位置を減算 (mcSubAxes) 詳細は、下記を参照してください。 1745ページ 加減算方法選択 (CombineMode)
GearRatioNumeratorM1	主軸1ギア比分子	DINT	起動時/再起動可/ 連続更新可	-2147483648~2147483647	1	主軸1(Master1)の分子の値を設定します。 詳細は、下記を参照してください。 1745ページ 主軸1ギア比分子 (GearRatioNumeratorM1)/主軸1ギア比分母 (GearRatioDenominatorM1)
GearRatioDenominatorM1	主軸1ギア比分母	DWORD(UDINT)	起動時/再起動可/ 連続更新可	1~2147483647	1	主軸1(Master1)の分母の値を設定します。 詳細は、下記を参照してください。 1745ページ 主軸1ギア比分子 (GearRatioNumeratorM1)/主軸1ギア比分母 (GearRatioDenominatorM1)
GearRatioNumeratorM2	主軸2ギア比分子	DINT	起動時/再起動可/ 連続更新可	-2147483648~2147483647	1	主軸2(Master2)の分子の値を設定します。 詳細は、下記を参照してください。 1746ページ 主軸2ギア比分子 (GearRatioNumeratorM2)/主軸2ギア比分母 (GearRatioDenominatorM2)
GearRatioDenominatorM2	主軸2ギア比分母	DWORD(UDINT)	起動時/再起動可/ 連続更新可	1~2147483647	1	主軸2(Master2)の分母の値を設定します。 詳細は、下記を参照してください。 1746ページ 主軸2ギア比分子 (GearRatioNumeratorM2)/主軸2ギア比分母 (GearRatioDenominatorM2)
MasterValueSourceM1	主軸1データソース選択	INT (MC_SOURCE)	起動時	1, 2, 101, 102	1	主軸1(Master1)のデータソースを設定します。 1: 指令現在(mcSetValue) 2: フィードバック値 (mcActualValue) 101: 最新指令現在値 (mcLatestSetValue) 102: 最新フィードバック値 (mcLatestActualValue) 詳細は、下記を参照してください。 1746ページ 主軸1データソース選択 (MasterValueSourceM1)/主軸2データソース選択 (MasterValueSourceM2)
MasterValueSourceM2	主軸2データソース選択	INT (MC_SOURCE)	起動時	1, 2, 101, 102	1	主軸2(Master2)のデータソースを設定します。 1: 指令現在(mcSetValue) 2: フィードバック値 (mcActualValue) 101: 最新指令現在値 (mcLatestSetValue) 102: 最新フィードバック値 (mcLatestActualValue) 詳細は、下記を参照してください。 1746ページ 主軸1データソース選択 (MasterValueSourceM1)/主軸2データソース選択 (MasterValueSourceM2)
BufferMode	バッファモード	INT (MC_BUFFER_MODE)	起動時	0, 1	0	バッファモードを設定します。 0: Aborting(mcAborting) 1: Buffered(mcBuffered) 詳細は、下記を参照してください。 1747ページ バッファモード (BufferMode)

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Options	オプション	DWORD(HEX)	起動時	00000000H	00000000H	「00000000H」を設定してください。 *: 「00000000H」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

■出力変数

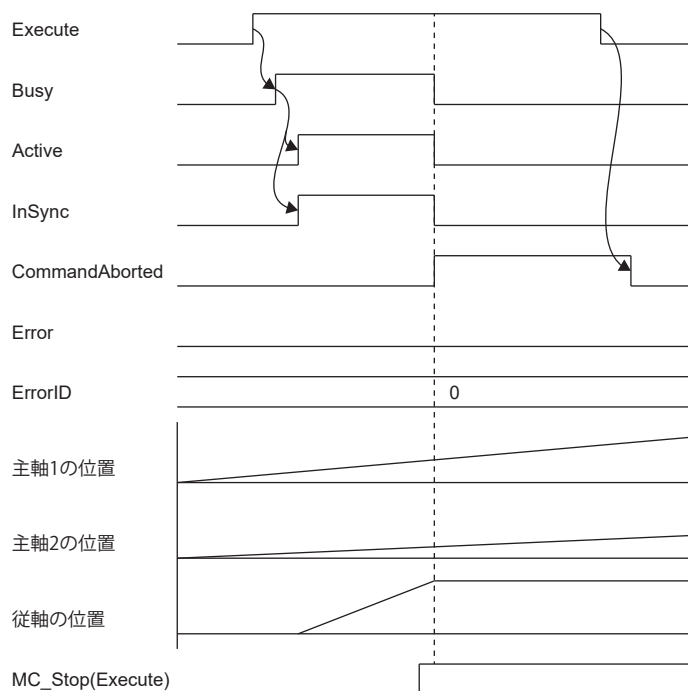
出力変数	名称	データ型	初期値	内容
InSync	同期中	BOOL	FALSE	従軸(Slave)が同期を開始したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MC_CombineAxes(加減算位置決め)を実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	従軸(Slave)を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	MC_CombineAxes(加減算位置決め)の実行が中断したときに、TRUEになります。 実行指令(Execute)がFALSEになると、FALSEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。  使用するコントローラのユーザーズマニュアル

機能

- MC_CombineAxes(加減算位置決め)では、加減算方式選択(CombineMode)、主軸1ギア比分子(GearRatioNumeratorM1)、主軸1ギア比分母(GearRatioDenominatorM1)、主軸2ギア比分子(GearRatioNumeratorM2)、主軸2ギア比分母(GearRatioDenominatorM2)、主軸1データソース選択(MasterValueSourceM1)、主軸2データソース選択(MasterValueSourceM2)、バッファモード(BufferMode)を設定し、加減算位置決めを行います。
- 動作を終了する場合は、MC_Stop(強制停止)にて行います。
- 主軸1(Master1)と主軸2(Master2)の移動量を合成します。また、各主軸にギア比を設定できます。移動量の合成は、主軸1(Master1)と主軸2(Master2)の単位に関係なく、数値を加減算します。

■タイミングチャート

- 正常完了の場合



- 異常完了の場合

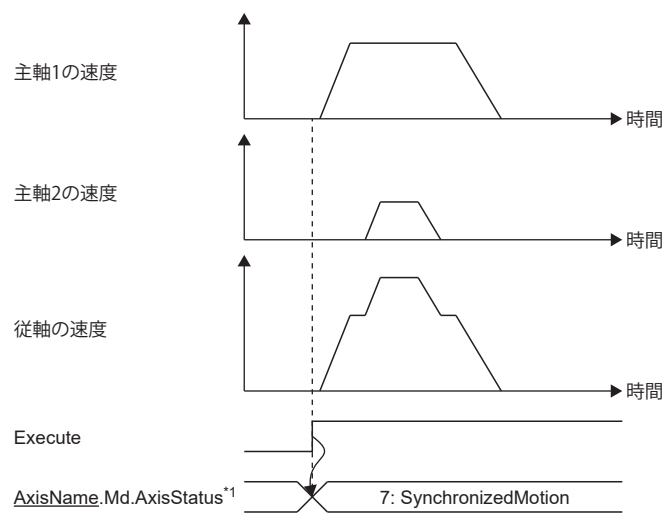
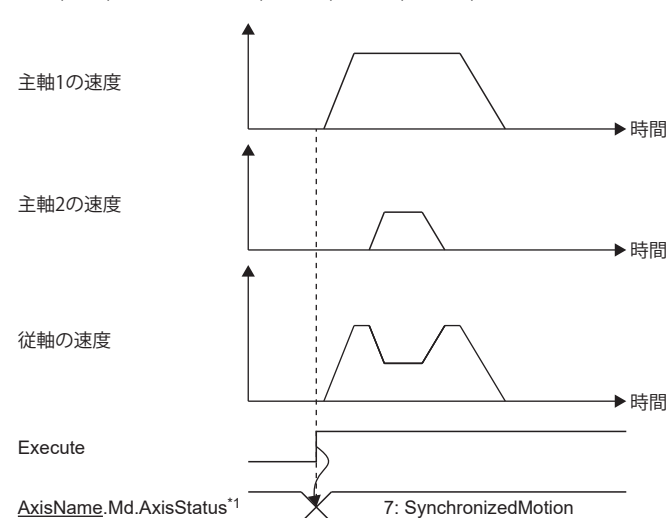
異常完了時のタイミングチャートについては、下記を参照してください。

 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■加減算方法選択(CombineMode)

主軸1(Master1)と主軸2(Master2)の移動量の合成方法を設定します。

移動量の合成は、主軸1(Master1)と主軸2(Master2)の単位に関係なく、数値を加減算します。

設定値	内容
0: 2つの入力軸の位置を加算(mcAddAxes)	主軸1(Master1)と主軸2(Master2)の移動量を加算して出力します。 従軸(Slave)の移動量 = 主軸1(Master1) + 主軸2(Master2)の移動量  The graphs show Master1 and Master2 velocities as trapezoidal pulses. The Slave velocity is the sum of both. The timeline shows the Execute signal and the AxisName.Md.AxisStatus signal, which transitions to '7: SynchronizedMotion' when both masters are active.
1: 2つの入力軸の位置を減算(mcSubAxes)	主軸1(Master1)から主軸2(Master2)の移動量を減算して出力します。 従軸(Slave)の移動量 = 主軸1(Master1) - 主軸2(Master2)の移動量  The graphs show Master1 and Master2 velocities as trapezoidal pulses. The Slave velocity is the difference between Master1 and Master2. The timeline shows the Execute signal and the AxisName.Md.AxisStatus signal, which transitions to '7: SynchronizedMotion' when both masters are active.

*1 従軸(Slave)の軸状態です。

■主軸1ギア比分子(GearRatioNumeratorM1)/主軸1ギア比分母(GearRatioDenominatorM1)

主軸1ギア比分子(GearRatioNumeratorM1)と主軸1ギア比分母(GearRatioDenominatorM1)は、組み合わせて設定します。

主軸1ギア比分子(GearRatioNumeratorM1)は、主軸1(Master1)の移動量を変換して合成するときの分子の値を設定します。
主軸1ギア比分母(GearRatioDenominatorM1)は、主軸1(Master1)の移動量を変換して合成するときの分母の値を設定します。

下記のように移動量を変換します。

$$\text{変換後の主軸1(Master1)移動量} = \text{変換前の主軸1(Master1)移動量} \times \frac{\text{主軸1ギア比分子(GearRatioNumeratorM1)}}{\text{主軸1ギア比分母(GearRatioDenominatorM1)}}$$

- 主軸1ギア比分母(GearRatioDenominatorM1)には正の値を設定してください。
- 主軸1ギア比分子(GearRatioNumeratorM1)の設定値に負の値を設定することで、移動量を逆転して合成できます。
- 主軸1ギア比分子(GearRatioNumeratorM1)の設定値に「0」を設定することで、変換後の主軸1(Master1)の移動量を「0」にして合成できます。

■主軸2ギア比分子(GearRatioNumeratorM2)/主軸2ギア比分母(GearRatioDenominatorM2)

主軸2ギア比分子(GearRatioNumeratorM2)と主軸2ギア比分母(GearRatioDenominatorM2)は、組み合わせて設定します。主軸2ギア比分子(GearRatioNumeratorM2)は、主軸2(Master2)の移動量を変換して合成するときの分子の値を設定します。主軸2ギア比分母(GearRatioDenominatorM2)は、主軸2(Master2)の移動量を変換して合成するときの分母の値を設定します。

下記のように移動量を変換します。

$$\text{変換後の主軸2(Master2)移動量} = \text{変換前の主軸2(Master2)移動量} \times \frac{\text{主軸2ギア比分子(GearRatioNumeratorM2)}}{\text{主軸2ギア比分母(GearRatioDenominatorM2)}}$$

- ・主軸2ギア比分母(GearRatioDenominatorM2)には正の値を設定してください。
- ・主軸2ギア比分子(GearRatioNumeratorM2)の設定値に負の値を設定することで、移動量を逆転して合成できます。
- ・主軸2ギア比分子(GearRatioNumeratorM2)の設定値に「0」を設定することで、変換後の主軸2(Master2)の移動量を「0」にして合成できます。

■主軸1データソース選択(MasterValueSourceM1)/主軸2データソース選択(MasterValueSourceM2)

従軸(Slave)が単軸同期制御を実行する主軸1(Master1)、または主軸2(Master2)の位置の種別を設定します。

主軸1(Master1)、または主軸2(Master2)と従軸(Slave)の演算周期が異なる場合やFBの実行順により、動作が変わる場合があります。

実軸以外の軸種別を主軸1(Master1)、または主軸2(Master2)に設定して主軸1データソース選択(MasterValueSourceM1)、または主軸2データソース選択(MasterValueSourceM2)にフィードバック値を設定した場合は、指令現在値と同じ値で動作します。

設定値	内容
1: 指令現在値(mcSetValue)	前回の演算周期における主軸の指令位置を使用します。
2: フィードバック値(mcActualValue)	前回の演算周期における主軸のフィードバック位置を使用します。
101: 最新指令現在値(mcLatestSetValue)	今回の演算周期における主軸の指令位置を使用します。
102: 最新フィードバック値(mcLatestActualValue)	今回の演算周期における主軸のフィードバック位置を使用します。

Point

主軸1データソース選択(MasterValueSourceM1)、または主軸2データソース選択(MasterValueSourceM2)に「1: 指令現在値(mcSetValue)」、「101: 最新指令現在値(mcLatestSetValue)」を設定し、主軸がサーボアラームや緊急停止によってサーボOFFとなった場合、値の変化量が大きくなる場合があります。主軸1データソース選択(MasterValueSourceM1)、または主軸2データソース選択(MasterValueSourceM2)に「2: フィードバック値(mcActualValue)」、「102: 最新フィードバック値(mcLatestActualValue)」を設定することで防ぐことができます。

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

MC_CombineAxes(加減算位置決め)では、下記のバッファモードが設定できます。

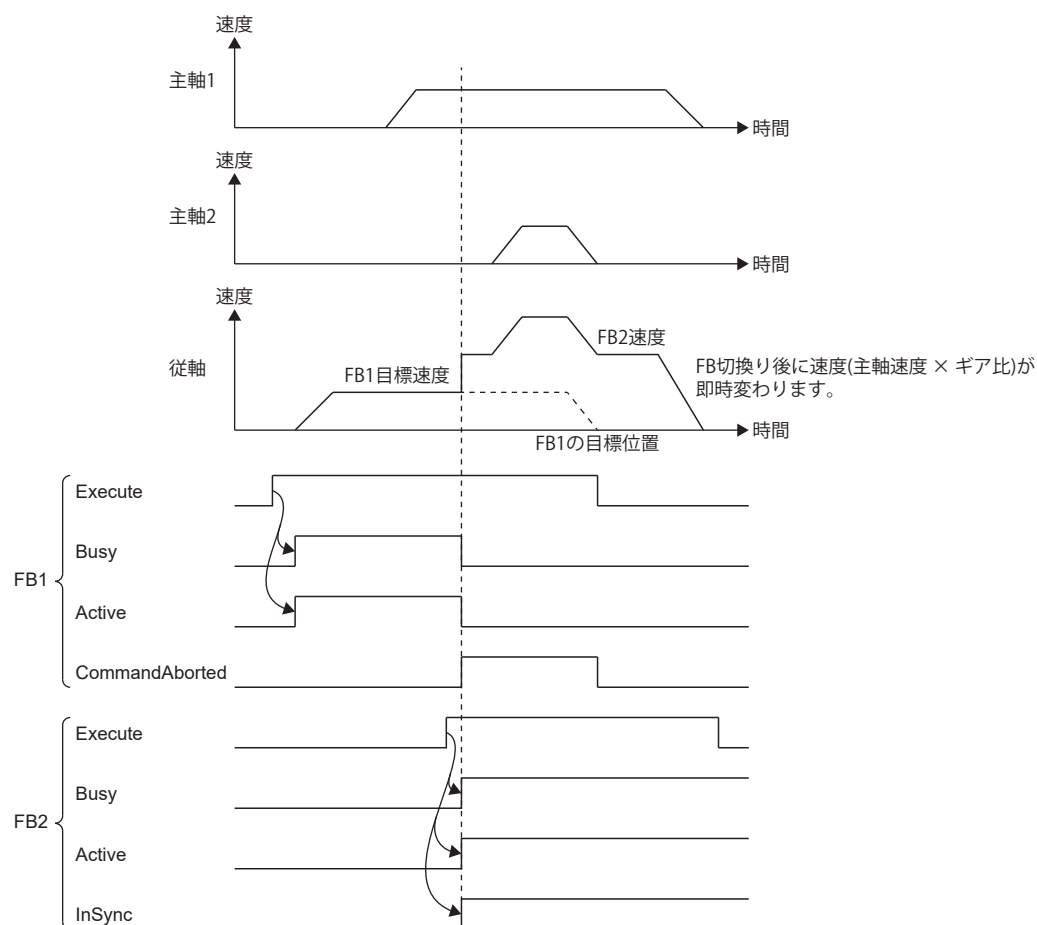
MC_CombineAxes(加減算位置決め)の速度は、同期しているFB(主軸)に追従しているため、FB切換え時に主軸速度と設定したギアデータに基づき速度が即時変わります。

設定値	内容
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。
1: Buffered(mcBuffered)	制御中のFBに次のFBをバッファリングします。 制御中のFBがすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ)制御中のFB完了にて、バッファリングFBを順次実行します。

例

実行中のFB1に連結起動FB2(従軸(Slave): MC_CombineAxes(加減算位置決め))にバッファモード(BufferMode)を「0: 常時(Aborting)」とした場合

- ・加減算方法選択(CombineMode)は、「0: 2つの入力軸の位置を加算(mcAddAxes)」のとき



FB2の速度が発生するのは、同期中(InSync)がTRUEになってからとなります。

Point

多重起動(バッファモード)の詳細については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

プログラム例

加減算位置決め開始指令(bCombineAxesStart)をTRUEにし、下記の設定によりJOG運転1FBの軸2(Axis0002)とJOG運転1FBの軸3(Axis0003)の主軸2軸の移動量を加減算した値を指令値とし、従軸(Axis0001)の動作を行うプログラム例を下記に示します。

- ・ 設定

項目		設定値	
		JOG運転1用データ	JOG運転2用データ
JOG運転用データ設定	速度	1000.0	500.0
	加速度	5000.0	5000.0
	減速度	5000.0	5000.0
	ジャーク	0.0	0.0

■軸

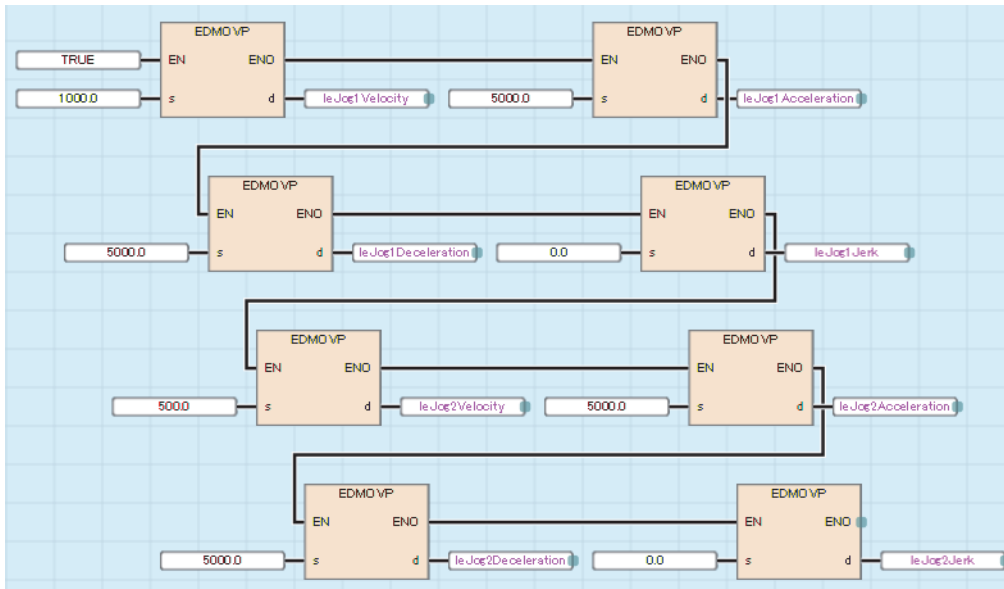
軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1
2	Axis0002	AXIS_REF	軸2
3	Axis0003	AXIS_REF	軸3

■使用するラベル

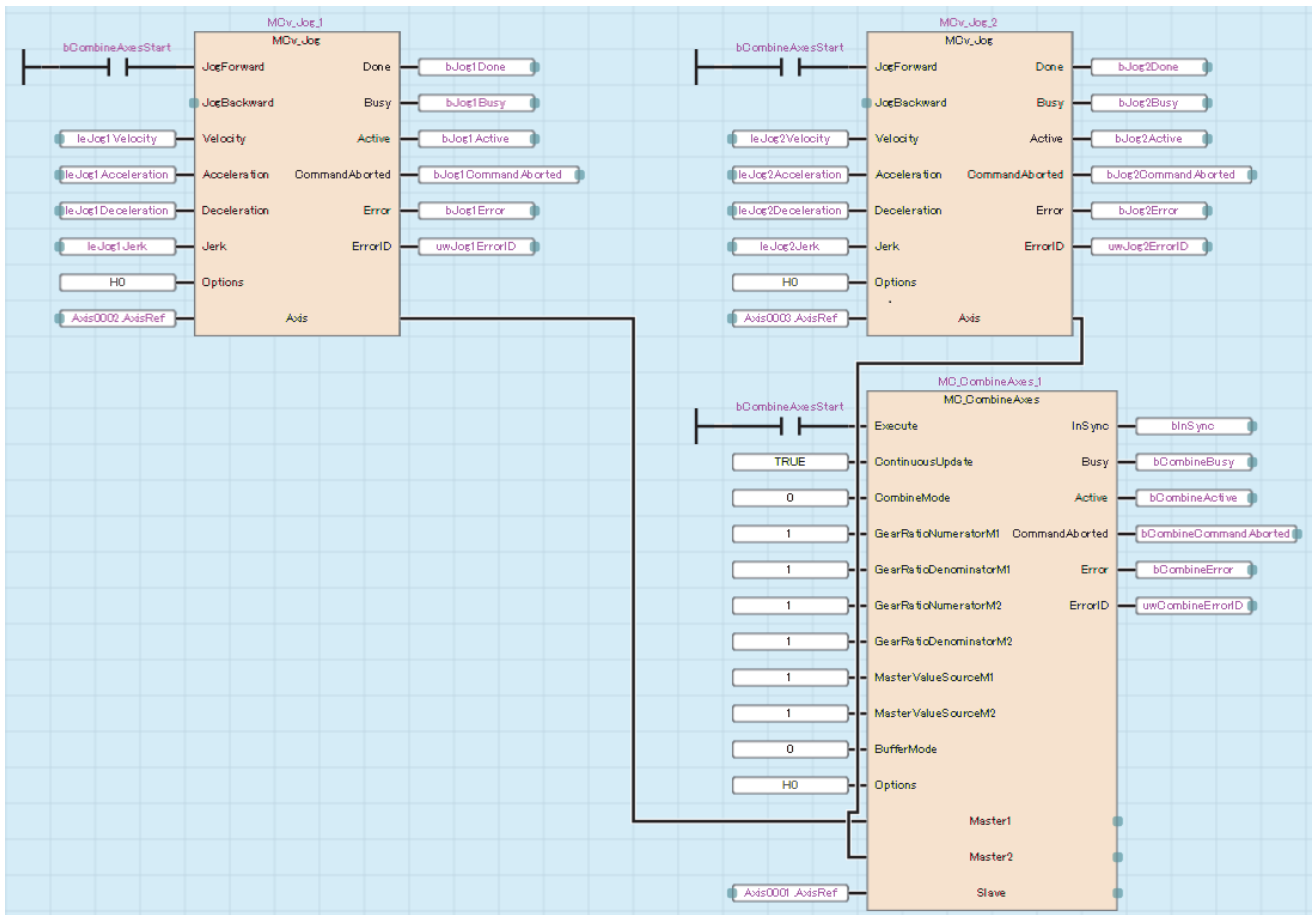
ラベル名	データ型	コメント
MC_CombineAxes_1	MC_CombineAxes	加減算位置決めFB
bCombineAxesStart	ビット	加減算位置決め開始指令
bInSync	ビット	同期中
bCombineBusy	ビット	実行中
bCombineActive	ビット	制御中
bCombineCommandAborted	ビット	実行中断
bCombineError	ビット	エラー
uwCombineErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
MCv_Jog_1	MCv_Jog	JOG運転1FB
leJog1Velocity	倍精度実数	JOG速度
leJog1Acceleration	倍精度実数	JOG加速度
leJog1Deceleration	倍精度実数	JOG減速度
leJog1Jerk	倍精度実数	JOGジャーク
bJog1Done	ビット	実行完了
bJog1Busy	ビット	実行中
bJog1Active	ビット	制御中
bJog1CommandAborted	ビット	実行中断
bJog1Error	ビット	エラー
uwJog1ErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
MCv_Jog_2	MCv_Jog	JOG運転2FB
leJog2Velocity	倍精度実数	JOG速度
leJog2Acceleration	倍精度実数	JOG加速度
leJog2Deceleration	倍精度実数	JOG減速度
leJog2Jerk	倍精度実数	JOGジャーク
bJog2Done	ビット	実行完了
bJog2Busy	ビット	実行中
bJog2Active	ビット	制御中
bJog2CommandAborted	ビット	実行中断
bJog2Error	ビット	エラー
uwJog2ErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード

■FBD/LDプログラム

・ JOG運転用データ設定



・ JOG運転1/JOG運転2/加減算位置決め



■STプログラム

//-----JOG運転用データ設定-----

```
leJog1Velocity:= 1000.0;  
leJog1Acceleration:= 5000.0;  
leJog1Deceleration:= 5000.0;  
leJog1Jerk:= 0.0;  
leJog2Velocity:= 500.0;  
leJog2Acceleration:= 5000.0;  
leJog2Deceleration:= 5000.0;  
leJog2Jerk:= 0.0;
```

//-----JOG運転1-----

```
MCv_Jog_1(  
  Axis:= Axis0002.AxisRef ,  
  JogForward:= bCombineAxesStart ,  
  Velocity:= leJog1Velocity ,  
  Acceleration:= leJog1Acceleration ,  
  Deceleration:= leJog1Deceleration ,  
  Jerk:= leJog1Jerk ,  
  Options:= H00000000 ,  
  Done=> bJog1Done ,  
  Busy=> bJog1Busy ,  
  Active=> bJog1Active ,  
  CommandAborted=> bJog1CommandAborted ,  
  Error=> bJog1Error ,  
  ErrorID=> uwJog1ErrorID  
);
```

//-----JOG運転2-----

```
MCv_Jog_2(  
  Axis:= Axis0003.AxisRef ,  
  JogForward:= bCombineAxesStart ,  
  Velocity:= leJog2Velocity ,  
  Acceleration:= leJog2Acceleration ,  
  Deceleration:= leJog2Deceleration ,  
  Jerk:= leJog2Jerk ,  
  Options:= H00000000 ,  
  Done=> bJog2Done ,  
  Busy=> bJog2Busy ,  
  Active=> bJog2Active ,  
  CommandAborted=> bJog2CommandAborted ,  
  Error=> bJog2Error ,  
  ErrorID=> uwJog2ErrorID  
);
```

```
//-----加減算位置決め-----  
MC_CombineAxes_1(  
    Master1:= Axis0002.AxisRef ,  
    Master2:= Axis0003.AxisRef ,  
    Slave:= Axis0001.AxisRef ,  
    Execute:= bCombineAxesStart ,  
    ContinuousUpdate:= TRUE ,  
    CombineMode:= MC_COMBINE_MODE__mcAddAxes ,  
    GearRatioNumeratorM1:= 1 ,  
    GearRatioDenominatorM1:= 1 ,  
    GearRatioNumeratorM2:= 1 ,  
    GearRatioDenominatorM2:= 1 ,  
    MasterValueSourceM1:= MC_SOURCE__mcSetValue ,  
    MasterValueSourceM2:= MC_SOURCE__mcSetValue ,  
    BufferMode:= MC_BUFFER_MODE__mcAborting ,  
    Options:= H00000000 ,  
    InSync=> bInSync ,  
    Busy=> bCombineBusy ,  
    Active=> bCombineActive ,  
    CommandAborted=> bCombineCommandAborted ,  
    Error=> bCombineError ,  
    ErrorID=> uwCombineErrorID  
);
```

42.17 バックラッシュ補正フィルタ

MCv_BacklashCompensationFilter

移動方向に従い、機械系のバックラッシュ量を補正するフィルタ処理を行います。

ラダー	FBD/LD	ST
MCv_BacklashCompensationFilter DUT: Master Master :DUT DUT: Slave Slave :DUT B: Enable Valid :B B: FilterDisable Busy :B L: BacklashAmount Active :B ENUM: BacklashDirection CommandAborted :B ENUM: MasterValueSource Error :B UD: Options ErrorID :UW FilterPool :L	MCv_BacklashCompensationFilter Enable Valid FilterDisable Busy BacklashAmount Active BacklashDirection CommandAborted MasterValueSource Error Options ErrorID FilterPool Master Slave	MCv_BacklashCompensationFilter(Master:= ?AXIS_REF?, Slave:= ?AXIS_REF?, Enable:= ?BOOL?, FilterDisable:= ?BOOL?, BacklashAmount:= ?LREAL?, BacklashDirection:= ?INT?, MasterValueSource:= ?INT?, Options:= ?DWORD?, Valid=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?, FilterPool=> ?LREAL?);

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Master	主軸	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📄 1378ページ AxisName.AxisRef. (軸情報)
Slave	従軸	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📄 1378ページ AxisName.AxisRef. (軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Enable	有効	BOOL	常時	TRUE, FALSE	FALSE	TRUEにすると、MCv_BacklashCompensationFilter(バックラッシュ補正フィルタ)を実行します。
FilterDisable	フィルタ無効	BOOL	常時	TRUE, FALSE	FALSE	フィルタ処理の方法を設定します。 • FALSE: フィルタ設定に従って動作します。 • TRUE: フィルタは無効となり、主軸 (Master)の入力をそのまま従軸 (Slave)に伝達します。 *: TRUEにした場合、フィルタ累積値 (FilterPool)は、「0」になり、破棄されたフィルタ累積値(PurgedFilterPool)へ、破棄したフィルタ累積値(FilterPool)を格納します。
BacklashAmount	バックラッシュ量	LREAL	起動時	0.0~2500000000.0[U]	0.0	機械系のバックラッシュ (ガタ)量を設定します。
BacklashDirection	バックラッシュ補正方向	INT (MC_DIRECTION)	起動時	1, 2	0	バックラッシュ補正する方向を設定します。 • 1: 正方向(mcPositiveDirection) • 2: 負方向(mcNegativeDirection) *: 設定を省略した場合、バックラッシュ補正方向範囲外(エラーコード: 1BC0H)となります。 詳細は、下記を参照してください。 1757ページ バックラッシュ補正方向 (BacklashDirection)
MasterValueSource	主軸データソース選択	INT (MC_SOURCE)	起動時	1, 101	1	主軸(Master)のデータソースを設定します。 • 1: 指令現在値(mcSetValue) • 101: 最新指令現在値(mcLatestSetValue) 詳細は、下記を参照してください。 1757ページ 主軸データソース選択 (MasterValueSource)
Options	オプション	DWORD(HEX)	起動時	00000000H	00000000H	機能オプションを設定します。 詳細は、下記を参照してください。 1758ページ オプション(Options)

■出力変数

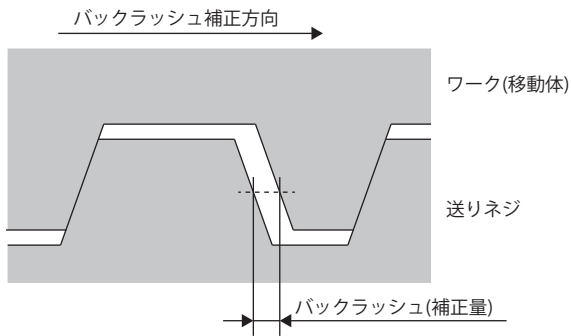
出力変数	名称	データ型	初期値	内容
Valid	出力値有効	BOOL	FALSE	出力値が有効なときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MCv_BacklashCompensationFilter(バックラッシュ補正フィルタ)を実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	MCv_BacklashCompensationFilter(バックラッシュ補正フィルタ)が軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	MCv_BacklashCompensationFilter(バックラッシュ補正フィルタ)の実行が中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 1757ページ 使用するコントローラのユーザーズマニュアル
FilterPool	フィルタ累積値	LREAL	0.0	フィルタ処理にて累積した従軸の移動量を出力します。 バックラッシュフィルタの場合、フィルタ累積値 (FilterPool)の正負を反転した値が従軸に伝達した補正量となります。

■公開変数

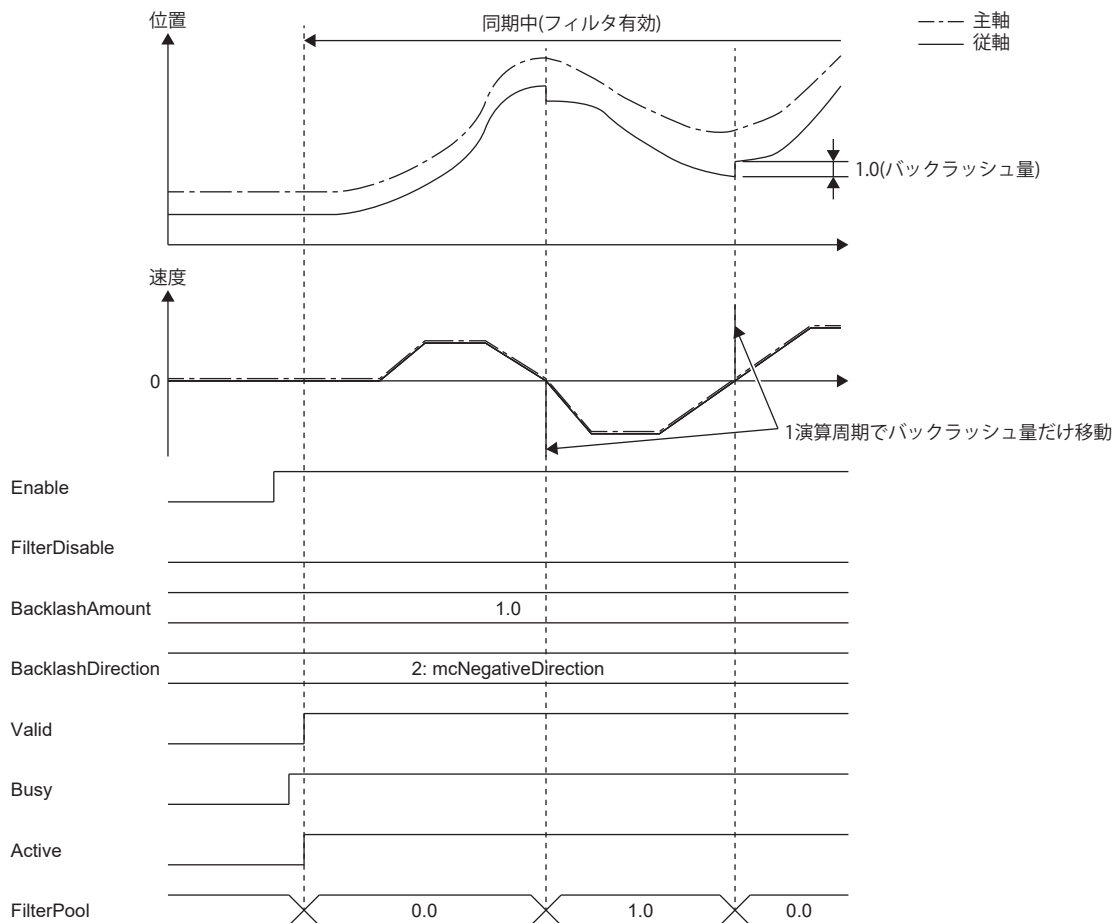
公開変数	名称	データ型	初期値	内容
PurgedFilterPool	破棄されたフィルタ累積値	LREAL	0.0	フィルタ無効(FilterDisable)をTRUEにすることで破棄したフィルタ累積値(FilterPool)を格納します。 破棄されたフィルタ累積値(PurgedFilterPool)は、累積しません。 FB始動時に0クリアします。

機能

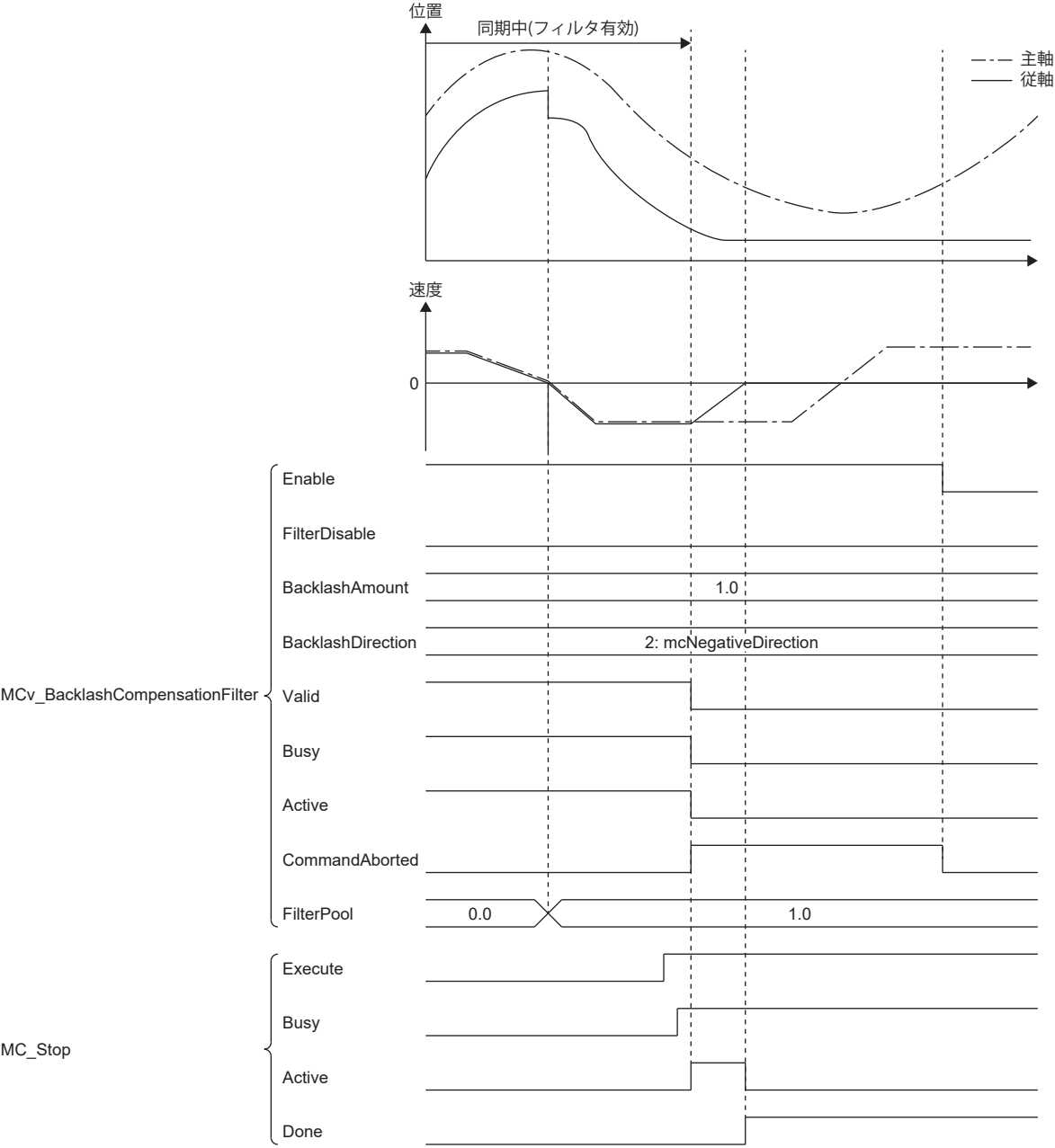
- バックラッシュ補正フィルタは、機械系のバックラッシュ (ガタ) 量の補正を行うフィルタです。



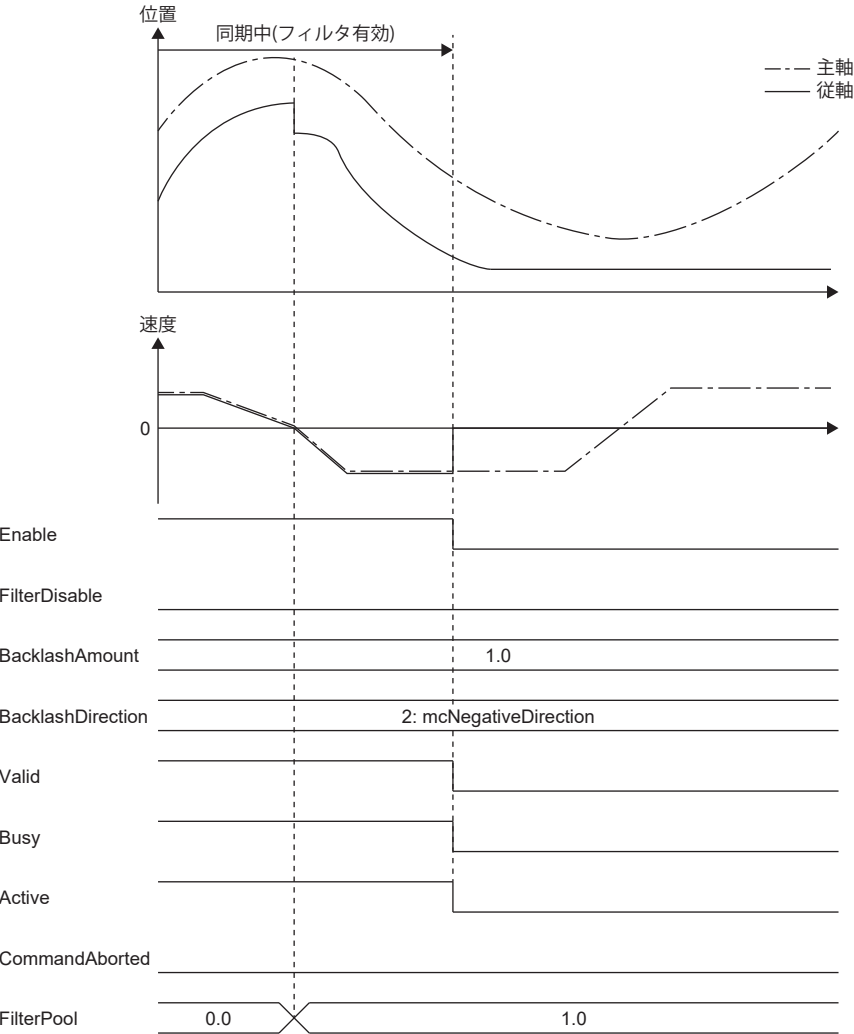
- 主軸(Master)の移動方向がバックラッシュ補正方向(BacklashDirection)と同じ場合、主軸(Master)の現在位置をバックラッシュ補正方向(BacklashDirection)の方向にバックラッシュ量(BacklashAmount)分シフトした位置に対して従軸(Slave)は同期します。主軸(Master)の移動方向がバックラッシュ補正方向(BacklashDirection)と逆の場合、主軸(Master)の現在位置に対して従軸(Slave)は同期します。
主軸(Master)の移動方向が変わるごとにバックラッシュ量(BacklashAmount)だけ余分に従軸(Slave)の移動量が発生することで、バックラッシュ補正を行います。



• バックラッシュ補正フィルタ固有の停止動作について示します。
従軸(Slave)に停止要因が発生すると従軸(Slave)は同期位置から停止動作を開始します。補正による移動量(フィルタ累積値(FilterPool)の値)は元に戻しません。

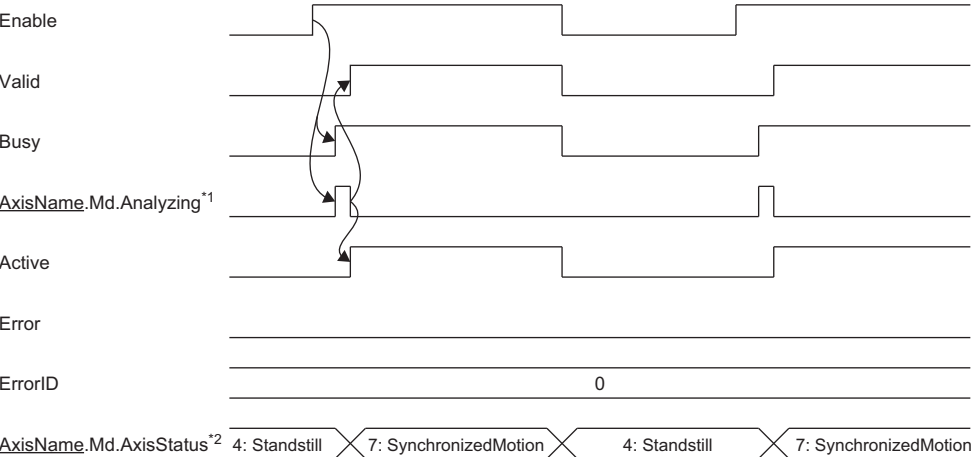


- ・フィルタ無効時(有効(Enable)立下り時)の動作
- 有効(Enable)を立ち下げると従軸(Slave)は同期位置で即停止します。補正による移動量(フィルタ累積値(FilterPool)の値)は元に戻しません。



■タイミングチャート

- ・正常完了の場合



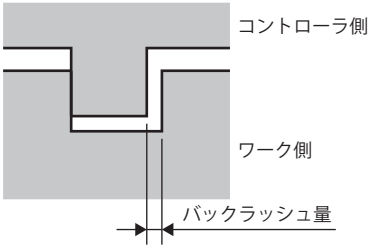
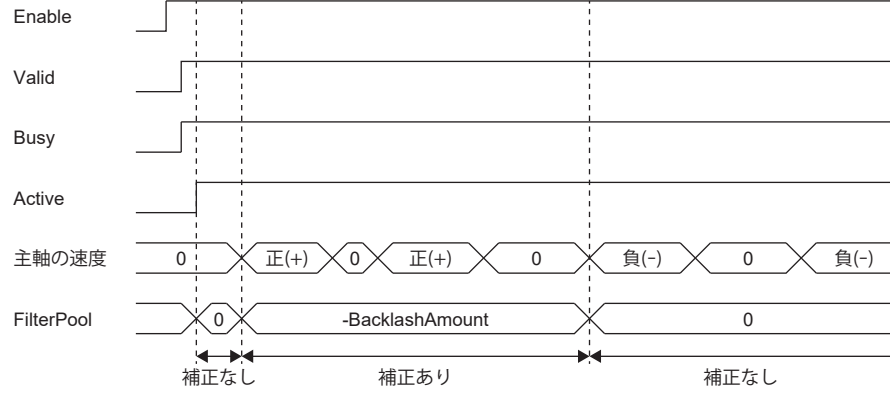
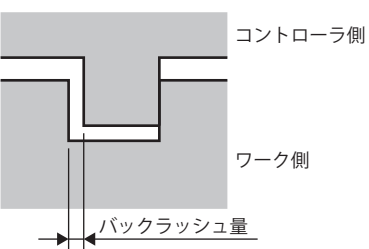
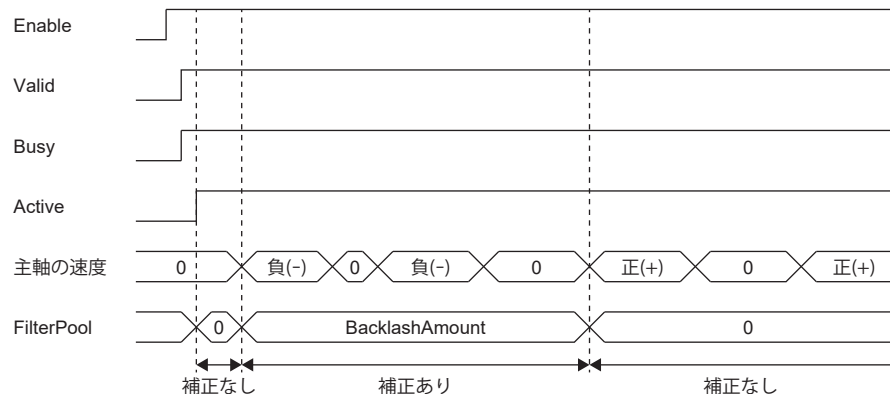
- *1 従軸(Slave)の解析中です。
 - *2 従軸(Slave)の軸状態です。
 - ・異常完了の場合
- 異常完了時のタイミングチャートについては、下記を参照してください。
- 📖 1370ページ 有効(Enable)型によるモーション制御FBの基本動作

■バックラッシュ補正方向(BacklashDirection)

バックラッシュ補正する方向を設定します。

主軸(Master)の移動方向がバックラッシュ補正方向(BacklashDirection)の場合、従軸(Slave)に補正量を出力します。

バックラッシュ補正フィルタ有効時(フィルタ無効(FilterDisable)をFALSEで起動したとき、または実行中にフィルタ無効(FilterDisable)を下下げたとき)に主軸(Master)の指令現在位置のアドレス増加方向に対して、機械系のバックラッシュ状態が下記の関係にあるとして補正を行います。

設定値	内容
1: 正方向(mcPositiveDirection) 指令現在位置の アドレス増加方向 	主軸(Master)の移動方向が正方向のときに、従軸(Slave)は主軸(Master)の現在位置に対し、正方向にバックラッシュ量(BacklashAmount)分シフトした位置に対して同期します。 
2: 負方向(mcNegativeDirection) 指令現在位置の アドレス増加方向 	主軸(Master)の移動方向が負方向のときに、従軸(Slave)は主軸(Master)の現在位置に対し、負方向にバックラッシュ量(BacklashAmount)分シフトした位置に対して同期します。 

■主軸データソース選択(MasterValueSource)

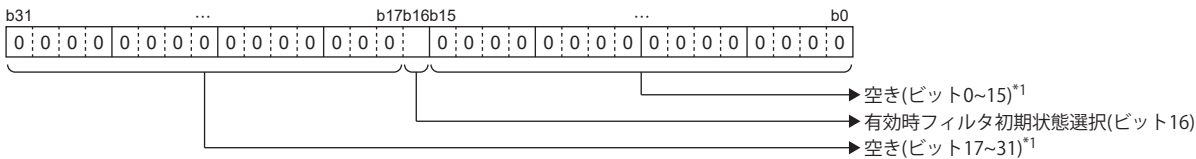
従軸(Slave)が単軸同期制御を実行する主軸(Master)の位置の種別を設定します。

主軸(Master)と従軸(Slave)の演算周期が異なる場合やFBの実行順により、動作が変わる場合があります。

設定値	内容
1: 指令現在値(mcSetValue)	前回の演算周期における主軸の指令位置を使用します。
101: 最新指令現在値(mcLatestSetValue)	今回の演算周期における主軸の指令位置を使用します。

■オプション(Options)

MCv_BacklashCompensationFilter(バックラッシュ補正フィルタ)で使用する機能オプションをビット指定で設定します。ビット指定で設定する内容を下記に示します。



*1 空きには「0」を設定してください。「0」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

ビット	名称	内容
16	有効時フィルタ初期状態選択	バックラッシュ補正フィルタ有効時(フィルタ無効(FilterDisable)をFALSEで起動したとき、または実行中にフィルタ無効(FilterDisable)を下げたとき)の初期状態を設定します。 • 0: 補正初回 *: 「0」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となり始動しません。

• 有効時フィルタ初期状態選択(ビット16)

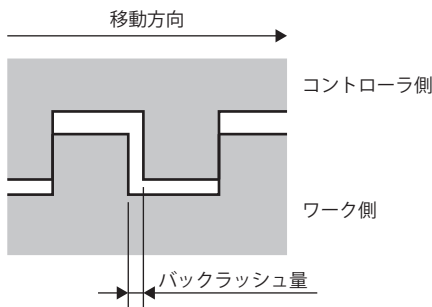
設定値	内容
0: 補正初回	フィルタ有効時の主軸(Master)の現在方向に関わらず、主軸(Master)と従軸(Slave)の現在位置がバックラッシュ補正を加算していない位置関係であるとしてフィルタ動作を開始します。

■バックラッシュ補正フィルタを有効にする手順

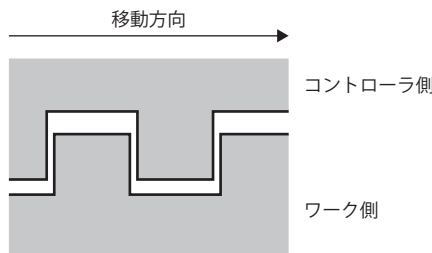
バックラッシュ補正フィルタを有効にする手順の使用例を示します。原点復帰制御により機械位置が確立した後や、バックラッシュ補正フィルタを途中で停止して再開する場合に行ってください。

1. 実ドライブ軸を原点復帰制御やJOG運転などで移動して、下記の正しい例のように一方向に機械系のバックラッシュが生じた状態としてください。

■正しい例



■誤りの例



2. 現在位置変更等により主軸(Master)の指令現在位置を実ドライブ軸の指令現在位置に合わせます。

3. 手順1.で実ドライブ軸を移動した方向の逆方向をバックラッシュ補正方向(BacklashDirection)に設定し、バックラッシュ補正フィルタを起動します。後段にファンクションブロックがある場合はそれらも起動します。

注意事項

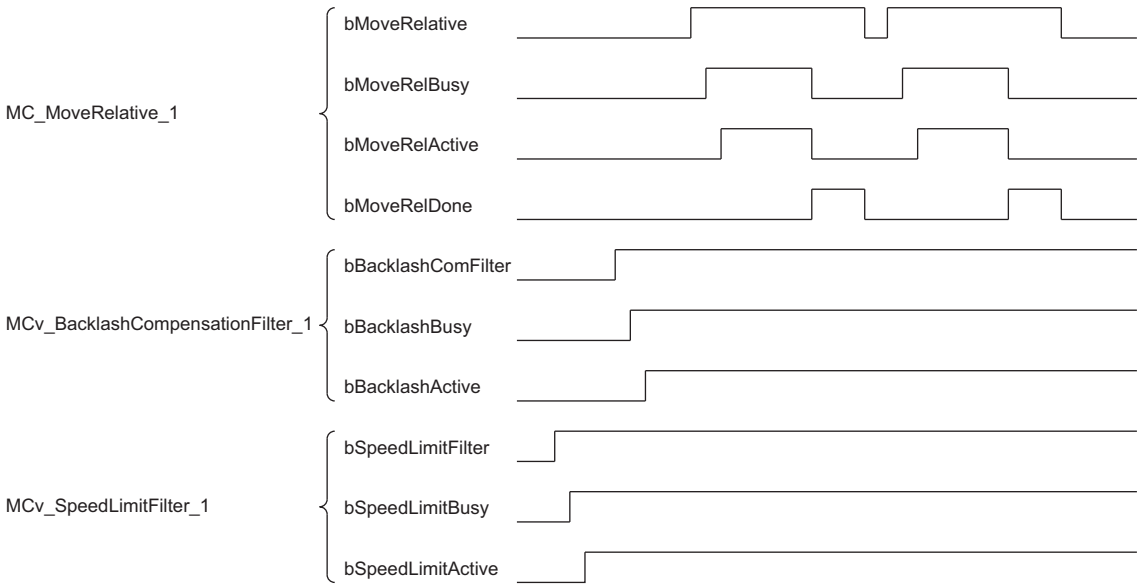
- バックラッシュ補正は、指令方向とバックラッシュの状態が、外力などで変わらないことを前提とした機能です。下記の場合に使用すると正しい補正が行われません。
 - 上下軸など常に一定方向に外力がかかっている機械
 - バックラッシュ量が機械位置により変化する機構(ラック&ピニオン機構等)
- 主軸(Master)の移動方向が変わる時点で従軸(Slave)は、バックラッシュ量(BacklashAmount)だけ1演算周期で移動します。値によっては、速度制限値やドライブユニットが許容できる指令周波数を超える場合があるため、バックラッシュ補正フィルタの後段に速度制限フィルタやスムージングフィルタを接続してください。
- 従軸(Slave)の現在位置(指令現在位置、送り機械位置等)は、バックラッシュ量(BacklashAmount)の補正を加算した位置となります。主軸(Master)の現在位置には、バックラッシュ量(BacklashAmount)の補正を加算しません。
- 指令フィルタの動作中にフィルタ無効(FilterDisable)の切り換えを行うと、従軸(Slave)の速度が急激に変化することがあります。
- 指令フィルタを有効にした後、従軸(Slave)に停止要因が発生して従軸(Slave)や後段の同期制御ファンクションブロックに主軸(Master)の指令を伝達しなくなった場合、同期位置関係が崩れます。指令フィルタを再度有効にする前に、必要に応じて同期位置合わせを行ってください。
- 演算周期混在機能使用時、主軸と従軸の制御演算周期設定が異なる場合は制約が生じます。詳細は、各機能および下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

プログラム例

バックラッシュ補正フィルタの後段に速度制限フィルタを接続してフィルタ処理を行うプログラム例を下記に示します。

• タイミングチャート



• 設定

項目		設定値
バックラッシュ量		1.0
単軸位置決め用データ設定	移動量	1000.0
	速度	20.0
	加速度	300.0
	減速度	300.0
	ジャーク	0.0
正方向/負方向制限値	正方向制限値	100.0
	負方向制限値	100.0

■軸

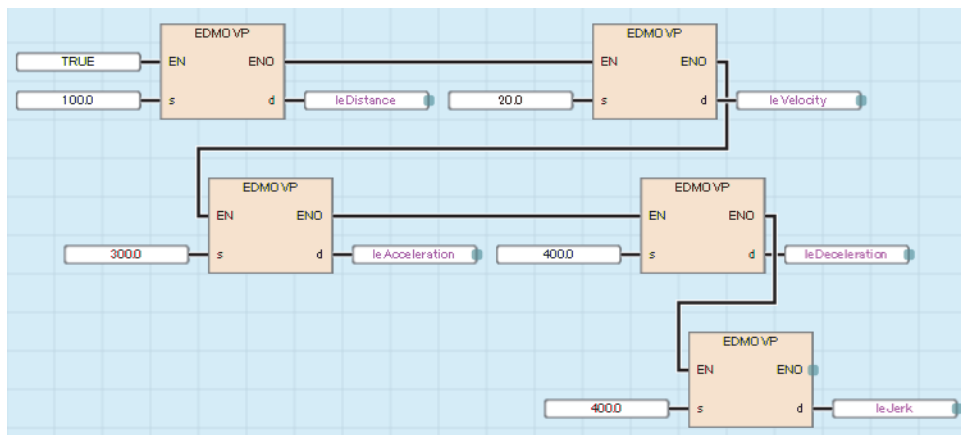
軸No.	ラベル名	データ型	コメント
1	VirtualAxis0001	AXIS_REF	軸1(仮想ドライブ軸)
2	VirtualAxis0002	AXIS_REF	軸2(仮想ドライブ軸)
3	Axis0003	AXIS_REF	軸3(実ドライブ軸)

■使用するラベル

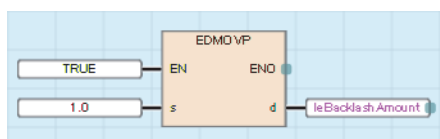
ラベル名	データ型	コメント
MC_MoveRelative_1	MC_MoveRelative	相対値位置決めFB
bMoveRelative	ビット	単軸相対値位置決め始動
leDistance	倍精度実数	移動量
leVelocity	倍精度実数	速度
leAcceleration	倍精度実数	加速度
leDeceleration	倍精度実数	減速度
leJerk	倍精度実数	ジャーク
bMoveRelDone	ビット	実行完了
bMoveRelBusy	ビット	実行中
bMoveRelActive	ビット	制御中
bMoveRelCommandAborted	ビット	実行中断
bMoveRelError	ビット	エラー
uwMoveRelErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
MCv_BacklashCompensationFilter_1	MCv_BacklashCompensationFilter	バックラッシュ補正フィルタFB
bBacklashComFilter	ビット	バックラッシュ補正フィルタ指令
leBacklashAmount	倍精度実数	バックラッシュ量
bBacklashValid	ビット	出力値有効
bBacklashBusy	ビット	実行中
bBacklashActive	ビット	制御中
bBacklashCommandAborted	ビット	実行中断
bBacklashError	ビット	エラー
uwBacklashErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
leBacklashFilterPool	倍精度実数	フィルタ累積値
MCv_SpeedLimitFilter_1	MCv_SpeedLimitFilter	速度制限フィルタ
bSpeedLimitFilter	ビット	速度制限フィルタ指令
lePositiveLimit	倍精度実数	正方向制限値
leNegativeLimit	倍精度実数	負方向制限値
bSpeedLimitValid	ビット	出力値有効
bSpeedLimitBusy	ビット	実行中
bSpeedLimitActive	ビット	制御中
bSpeedLimitCommandAborted	ビット	実行中断
bSpeedLimitError	ビット	エラー
uwSpeedLimitErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
leFilterPool	倍精度実数	フィルタ累積値

■FBD/LDプログラム

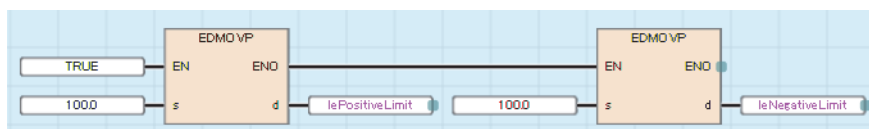
- 単軸位置決め用データ設定



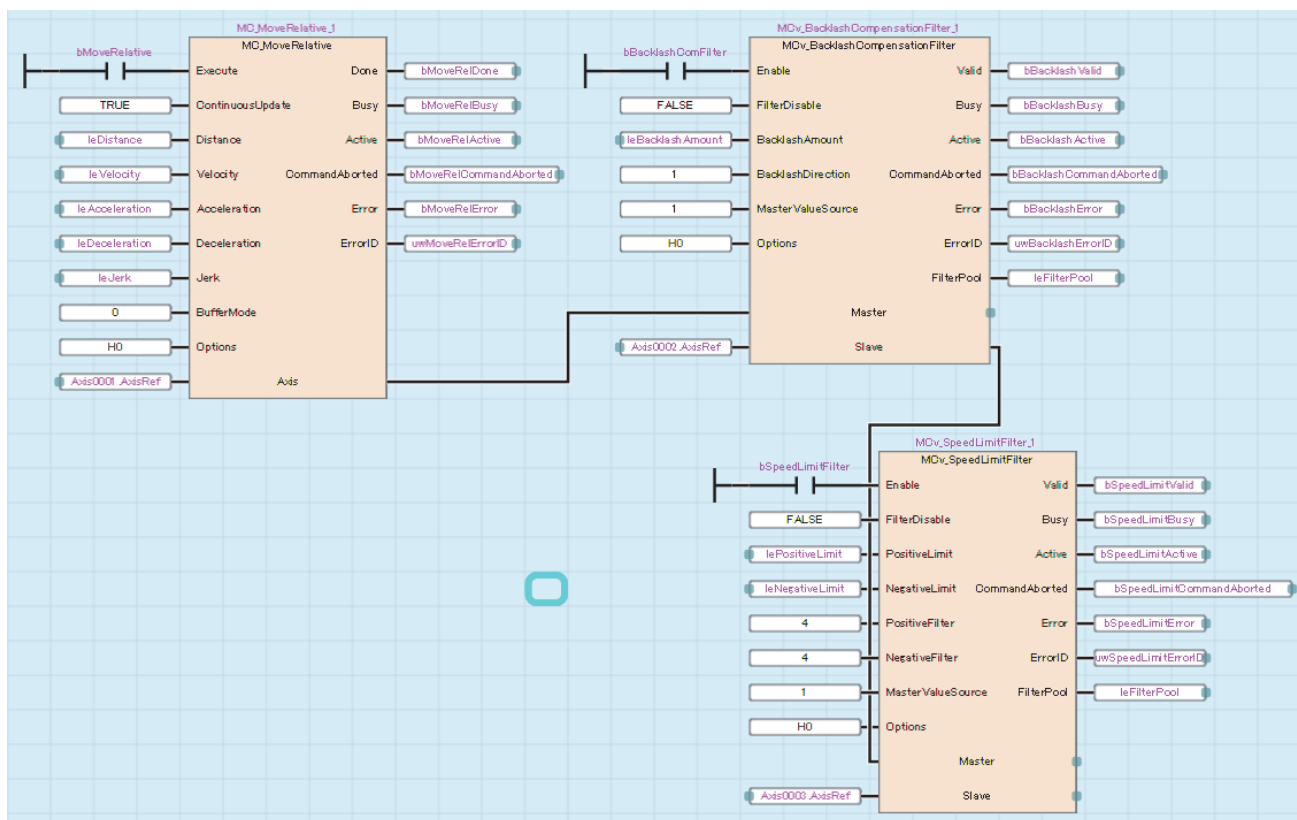
- バックラッシュ量



- 正方向/負方向制限値



- 相対値位置決め/バックラッシュ補正フィルタ/速度制限フィルタ



■STプログラム

//-----単軸位置決め用データ設定-----

```
leDistance:= 100.0;
leVelocity:= 20.0;
leAcceleration:= 300.0;
leDeceleration:= 400.0;
leJerk:= 0.0;
```

//-----相対値位置決め-----

```
MC_MoveRelative_1(
  Axis:= VirtualAxis0001.AxisRef ,
  Execute:= bMoveRelative ,
  ContinuousUpdate:= TRUE ,
  Distance:= leDistance ,
  Velocity:= leVelocity ,
  Acceleration:= leAcceleration ,
  Deceleration:= leDeceleration ,
  Jerk:= leJerk ,
  BufferMode:= MC_BUFFER_MODE__mcAborting ,
  Options:= H00000000 ,
  Done=> bMoveRelDone ,
  Busy=> bMoveRelBusy ,
  Active=> bMoveRelActive ,
  CommandAborted=> bMoveRelCommandAborted ,
  Error=> bMoveRelError ,
  ErrorID=> uwMoveRelErrorID
);
```

//-----バックラッシュ量-----

```
leBacklashAmount:= 1.0;
```

//-----バックラッシュ補正フィルタ-----

```
MCv_BacklashCompensationFilter_1(
  Master:= MC_MoveRelative_1.Axis ,
  Slave:= VirtualAxis0002.AxisRef ,
  Enable:= bBacklashComFilter ,
  FilterDisable:= FALSE ,
  BacklashAmount:= leBacklashAmount ,
  BacklashDirection:= MC_DIRECTION__mcPositiveDirection ,
  MasterValueSource:= MC_SOURCE__mcSetValue ,
  Options:= H00000000 ,
  Valid=> bBacklashValid ,
  Busy=> bBacklashBusy ,
  Active=> bBacklashActive ,
  CommandAborted=> bBacklashCommandAborted ,
  Error=> bBacklashError ,
  ErrorID=> uwBacklashErrorID ,
  FilterPool=> leBacklashFilterPool
);
```

```

//-----正方向/負方向制限値-----
lePositiveLimit:= 100.0;
leNegativeLimit:= 100.0;

//-----速度制限フィルタ-----
MCv_SpeedLimitFilter_1(
  Master:= MCv_BacklashCompensationFilter_1.Slave ,
  Slave:= Axis0003.AxisRef ,
  Enable:= bSpeedLimitFilter ,
  FilterDisable:= FALSE ,
  PositiveLimit:= lePositiveLimit ,
  NegativeLimit:= leNegativeLimit ,
  PositiveFilter:= MC_VELOCITY_LIMIT_MODE__ClampWithoutRamp ,
  NegativeFilter:= MC_VELOCITY_LIMIT_MODE__ClampWithoutRamp ,
  MasterValueSource:= MC_SOURCE__mcSetValue ,
  Options:= H00000000 ,
  Valid=> bSpeedLimitValid ,
  Busy=> bSpeedLimitBusy ,
  Active=> bSpeedLimitActive ,
  CommandAborted=> bSpeedLimitCommandAborted ,
  Error=> bSpeedLimitError ,
  ErrorID=> uwSpeedLimitErrorID ,
  FilterPool=> leFilterPool
);

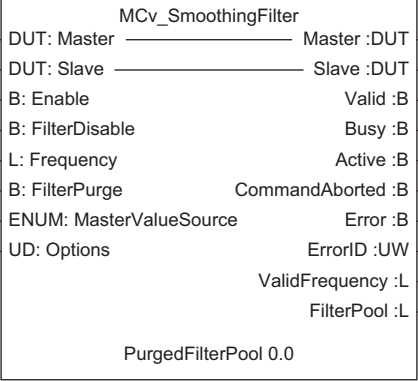
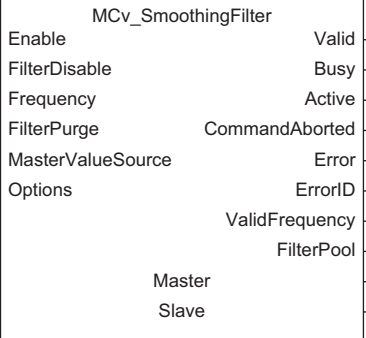
```

42.18 スムージングフィルタ

42

MCv_SmoothingFilter

指定した周波数に従い、フィルタ処理を行います。

ラダー	FBD/LD	ST
		<pre>MCv_SmoothingFilter(Master:= ?AXIS_REF?, Slave:= ?AXIS_REF?, Enable:= ?BOOL?, FilterDisable:= ?BOOL?, Frequency:= ?LREAL?, FilterPurge:= ?BOOL?, MasterValueSource:= ?INT?, Options:= ?DWORD?, Valid=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?, ValidFrequency=> ?LREAL?, FilterPool=> ?LREAL?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Master	主軸	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、 下記を参照してください。 📖 1378ページ AxisName.AxisRef.(軸情報)
Slave	従軸	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、 下記を参照してください。 📖 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Enable	有効	BOOL	常時	TRUE, FALSE	FALSE	TRUEにすると、MCv_SmoothingFilter(スムージングフィルタ)を実行します。
FilterDisable	フィルタ無効	BOOL	常時	TRUE, FALSE	FALSE	フィルタ処理の方法を設定します。 ・FALSE: フィルタ設定に従って動作します。 ・TRUE: フィルタは無効となり、主軸(Master)の入力をそのまま従軸(Slave)に伝達します。 詳細は、下記を参照してください。 📖 1768ページ フィルタ無効(FilterDisable)
Frequency	周波数	LREAL	常時	0.20~250.00[Hz]	10.00	振動を抑制する周波数を設定します。 詳細は、下記を参照してください。 📖 1768ページ 周波数(Frequency)
FilterPurge	フィルタパージ	BOOL	起動時	TRUE, FALSE	FALSE	フィルタ累積値がある状態での終了動作を設定します。 ・FALSE: 従軸を即時停止します。 ・TRUE: 「0」になるまで動作した後、停止します。 詳細は、下記を参照してください。 📖 1769ページ フィルタパージ(FilterPurge)

入力変数	名称	データ型	取込	有効範囲	初期値	内容
MasterValueSource	主軸データソース 選択	INT (MC_SOURCE)	起動時	1, 2, 101, 102	1	主軸(Master)のデータソースを設定します。 1: 指令現在値(mcSetValue) 2: フィードバック値(mcActualValue) 101: 最新指令現在値(mcLatestSetValue) 102: 最新フィードバック値 (mcLatestActualValue) 詳細は、下記を参照してください。  1770ページ 主軸データソース選択 (MasterValueSource)
Options	オプション	DWORD(HEX)	起動時	00000000H	00000000H	「00000000H」を設定してください。 *: 「00000000H」以外を設定した場合、 Options範囲外(エラーコード: 1ABBH)となり ます。

■出力変数

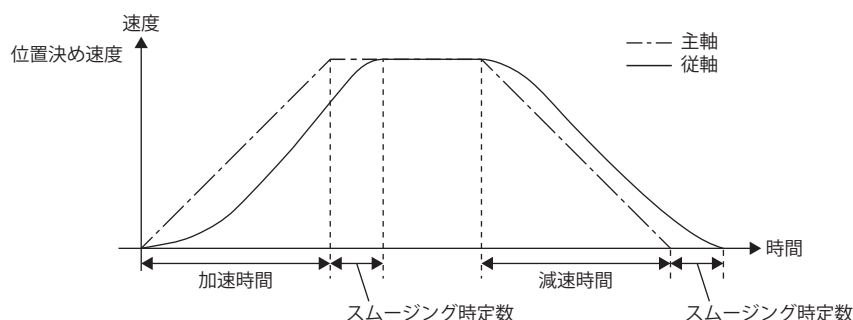
出力変数	名称	データ型	初期値	内容
Valid	出力値有効	BOOL	FALSE	出力値が有効なときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MCv_SmoothingFilter(スムージングフィルタ)を実 行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	MCv_SmoothingFilter(スムージングフィルタ)が軸 を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	MCv_SmoothingFilter(スムージングフィルタ)の実 行が中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。  使用するコントローラのユーザーズマニュアル
ValidFrequency	有効周波数	LREAL	0.00	有効となっている周波数を出力します。
FilterPool	フィルタ累積値	LREAL	0.0	フィルタ処理にて累積した従軸の移動量を出力し ます。 フィルタ累積値(FilterPool)は、有効(Enable)を TRUEにしたときにクリアします。 フィルタ累積値(FilterPool)の値が位置決め範囲を 超えた場合、フィルタ累積値オーバ警告(イベント コード: 00D13H)となります。 警告検出時、フィルタ累積値(FilterPool)の誤差が 大きくなる場合があります。

■公開変数

公開変数	名称	データ型	初期値	内容
PurgedFilterPool	破棄されたフィルタ累積値	LREAL	0.0	フィルタ無効(FilterDisable)をTRUEにすることで破 棄したフィルタ累積値(FilterPool)を格納します。 破棄されたフィルタ累積値(PurgedFilterPool)は、 累積しません。 FB始動時に0クリアします。

機能

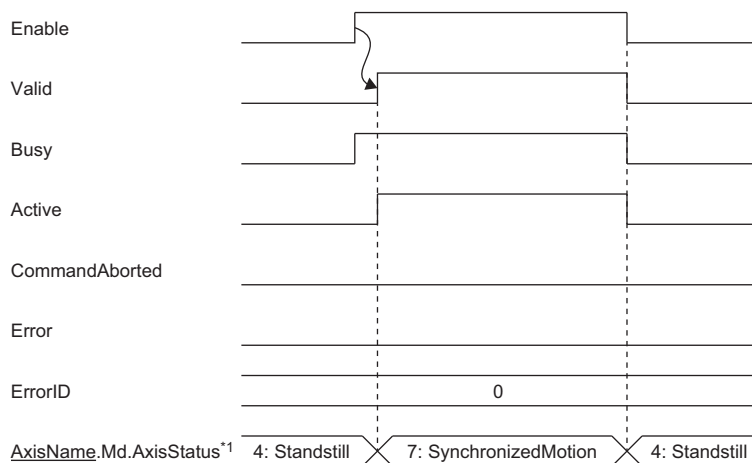
- スムージングフィルタは、設定した周波数(Frequency)より高い周波数を除去でき、設定値より高い波形全体が滑らかな加減速波形となるフィルタです。
- スムージングフィルタは、主軸データソース選択(MasterValueSource)で設定した主軸(Master)の値に対して、フィルタ処理を実行した結果を従軸(Slave)に出力します。
- スムージングフィルタ実行中の従軸(Slave)の軸状態(AxisName.Md.AxisStatus)は、「7: 同期運転中(SynchronizedMotion)」になります。
- スムージングの時定数は、「1/周波数(Frequency)[s]」となり、加速時間と減速時間はスムージング時定数の分だけ長くなります。周波数(Frequency)が10[Hz]の場合、スムージング時定数が「0.1[s] = 100[ms]」となります。



- 従軸(Slave)が他の動作FB中に、スムージングフィルタを始動した場合、始動要求を無視し、始動不可(エラーコード: 1AFCH)となります。従軸(Slave)の軸状態(AxisName.Md.AxisStatus)が「4: 待機中(Standstill)」のときに始動してください。
- スムージングフィルタ実行中に、多重起動可能なファンクションブロックは、MC_Stop(強制停止)のみです。

■タイミングチャート

- 正常完了の場合



*1 従軸(Slave)の軸状態です。

- 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

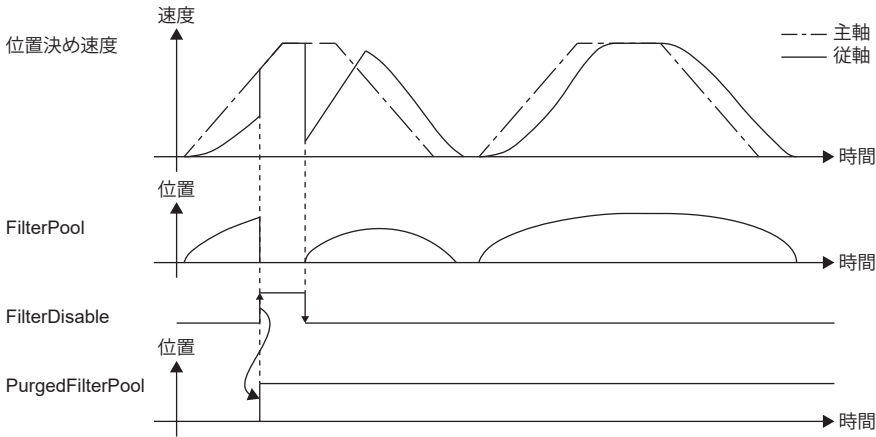
☞ 1370ページ 有効(Enable)型によるモーション制御FBの基本動作

■フィルタ無効(FilterDisable)

フィルタ処理の方法を設定します。

設定値	内容
FALSE	フィルタ設定に従って動作します。
TRUE	フィルタは無効となり、主軸(Master)の入力をそのまま従軸(Slave)に伝達します。

- ・フィルタ無効(FilterDisable)をTRUEにした場合、フィルタ累積値(FilterPool)は、「0.0」になり、破棄されたフィルタ累積値(PurgedFilterPool)へ、破棄したフィルタ累積値(FilterPool)を格納します。



■周波数(Frequency)

振動を抑制する周波数を設定します。

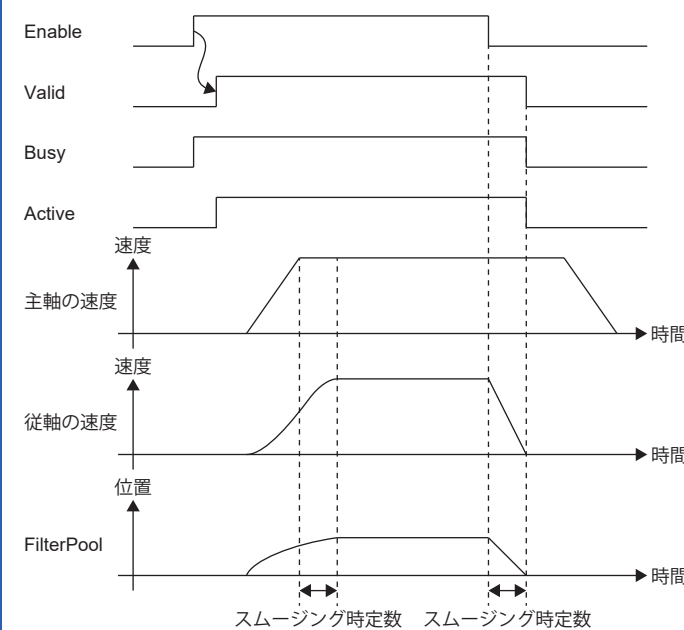
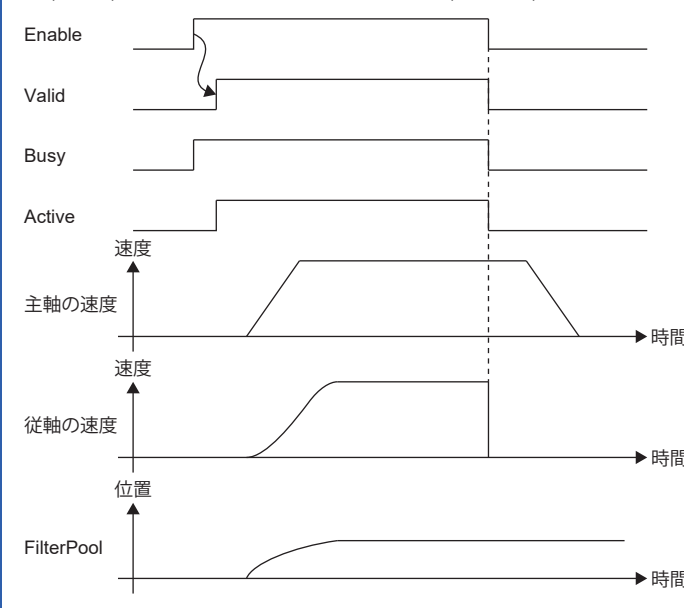
周波数(Frequency)の設定可能な範囲を下記に示します。

1/従軸演算周期[s]	設定範囲
250.00[Hz]以上	0.20~250.00[Hz]
250.00[Hz]未満	0.20~(1/従軸の演算周期[s])[Hz]

- ・周波数(Frequency)に範囲外の値を設定した場合、周波数指定範囲外(エラーコード: 1AE6H)となります。
- ・有効となっている周波数は、有効周波数(ValidFrequency)にて確認できます。
- ・周波数(Frequency)を変更する場合は、フィルタ累積値(FilterPool)が「0.0」のときに変更してください。「0.0」以外の状態で変更しても、「0.0」になるまで反映しません。
- ・周波数(Frequency)は、有効周波数(ValidFrequency)より低い周波数へ変更できません。低い周波数を設定した場合、周波数指定範囲外(エラーコード: 1AE6H)となります。
- ・主軸が正転/逆転を繰り返す運転パターンで動作している最中に、フィルタ累積値(FilterPool)が「0.0」になることがあります。フィルタ動作が安定していない状態で周波数(Frequency)の値を変更するとフィルタ動作を途中で打ち切るため、主軸と従軸のずれの原因となります。周波数(Frequency)を変更する場合は、主軸の停止確認後、フィルタ時定数分待ってから変更してください。

■フィルタパージ(FilterPurge)

フィルタ累積値(FilterPool)が「0.0」以外の状態で、有効(Enable)をFALSEにした場合の動作を設定します。

設定値	内容
TRUE	有効(Enable)がFALSEになった場合、フィルタ累積値(FilterPool)が「0.0」になるまで動作した後、停止します。 
FALSE	有効(Enable)がFALSEになった場合、フィルタ累積値(FilterPool)の状態にかかわらず、従軸を即時停止します。 

■主軸データソース選択(MasterValueSource)

従軸(Slave)が単軸同期制御を実行する主軸(Master)の位置の種別を設定します。

主軸(Master)と従軸(Slave)の演算周期が異なる場合やFBの実行順により、動作が変わる場合があります。

実軸以外の軸種別を主軸(Master)に設定して主軸データソース選択(MasterValueSource)にフィードバック値を設定した場合は、指令現在値と同じ値で動作します。

設定値	内容
1: 指令現在値(mcSetValue)	前回の演算周期における主軸の指令位置を使用します。
2: フィードバック値(mcActualValue)	前回の演算周期における主軸のフィードバック位置を使用します。
101: 最新指令現在値(mcLatestSetValue)	今回の演算周期における主軸の指令位置を使用します。
102: 最新フィードバック値(mcLatestActualValue)	今回の演算周期における主軸のフィードバック位置を使用します。

Point

主軸データソース選択(MasterValueSource)に「1: 指令現在値(mcSetValue)」、「101: 最新指令現在値(mcLatestSetValue)」を設定し、主軸がサーボアラームや緊急停止によってサーボOFFとなった場合、値の変化量が大きくなる場合があります。主軸データソース選択(MasterValueSource)に「2: フィードバック値(mcActualValue)」、「102: 最新フィードバック値(mcLatestActualValue)」を設定することで防ぐことができます。

注意事項

- ・スムージングフィルタは、フィルタ段数が大きくなるほど演算負荷が増加します。フィルタ段数が5000のときの演算負荷の目安は下記となります。

使用するユニット	演算負荷
MXR300-16 MXR300-32 MXR300-64	170μs
MXR500-128 MXR500-256	130μs

- ・フィルタ段数は、下記の式で算出します。小数点以下は四捨五入します。ただし四捨五入した結果が「0」となる場合は、1段として動作します。

フィルタ段数 = (1 / 周波数[Hz]) / 演算周期[s]

- ・スムージングフィルタは、始動時にフィルタ処理で使用するメモリを確保します。メモリ使用量は、下記の計算式で求めることができます。メモリ不足が発生する場合は、アドオンMotionControl_AxisFilterパラメータ(Addon_MotionControl_AxisFilter)のRAM最大サイズ(AddonSystem.PrConst.MT_MotionControl_AxisFilter.RamSizeMax)を変更してください。

メモリ使用量 = フィルタ段数 × 8[バイト]

- ・主軸や従軸の移動量が大きく、下記に該当すると演算誤差が大きくなる場合があります。主軸や従軸の単位設定を見直し、移動範囲が小さくなるようにしてください。誤差が大きくなった場合、同期状態を解除するか絶対位置指定の現在値変更を実行してください。
- ・制御中のフィルタ段数分の移動量の総和(下記計算式)が倍精度浮動小数で表現できる有効桁数(約15桁)を超えたとき。

フィルタ段数 - 1

$$\sum_{i=0} (i \text{ 演算周期前の主軸の位置} - \text{制御開始時の主軸の位置})$$

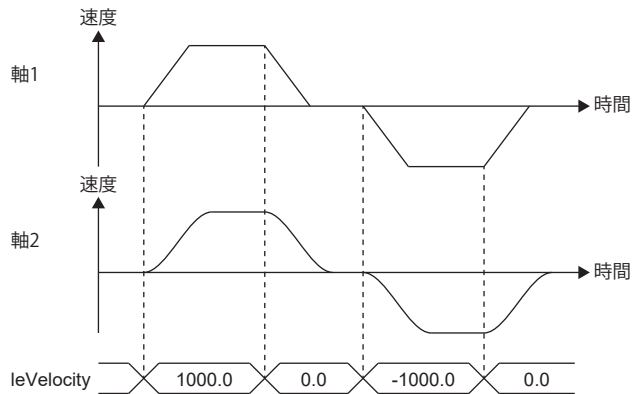
- ・指令フィルタの動作中にフィルタ無効(FilterDisable)の切換えを行うと、従軸(Slave)の速度が急激に変化する場合があります。
- ・指令フィルタを有効にした後、従軸(Slave)に停止要因が発生して従軸(Slave)や後段の同期制御ファンクションブロックに主軸(Master)の指令を伝達しなくなった場合、同期位置関係が崩れます。指令フィルタを再度有効にする前に、必要に応じて同期位置合わせを行ってください。
- ・演算周期混在機能使用時、主軸と従軸の制御演算周期設定が異なる場合は制約が生じます。詳細は、各機能および下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

プログラム例

台形加減速を滑らかな加減速波形にするフィルタ処理を行うプログラム例を下記に示します。

・動作



・設定

項目	設定値
周波数	10.00
速度制御用データ設定	速度
	5000.0
	加速度
	100.0
	減速度
	100.0
	ジャーク
	0.0

■軸

軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1
2	Axis0002	AXIS_REF	軸2

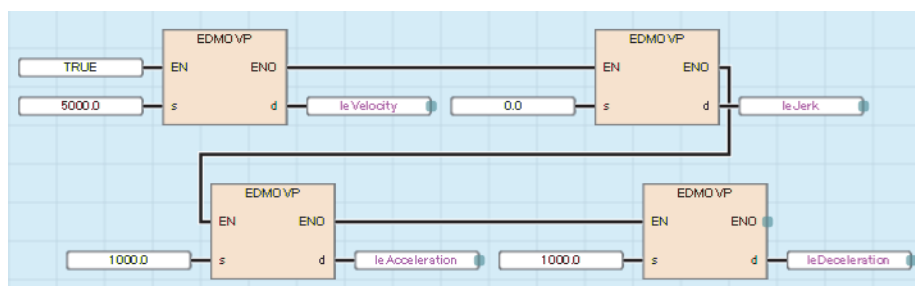
■使用するラベル

ラベル名	データ型	コメント
MCv_SpeedControl_1	MCv_SpeedControl	速度制御制御(ループを含む)FB
bSpeedControl	ビット	速度制御指令
leVelocity	倍精度実数	指令速度
leAcceleration	倍精度実数	加速度
leDeceleration	倍精度実数	減速度
leJerk	倍精度実数	ジャーク
bSpeedInVelocity	ビット	目標速度到達
bSpeedBusy	ビット	実行中
bSpeedActive	ビット	制御中
bSpeedCommandAborted	ビット	実行中断
bSpeedError	ビット	エラー
uwSpeedErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
MCv_SmoothingFilter_1	MCv_SmoothingFilter	スムージングフィルタFB
bSmoothingFilter	ビット	スムージングフィルタ指令
leFrequency	倍精度実数	周波数
bSmoothingValid	ビット	出力値有効
bSmoothingInVelocity	ビット	目標速度到達
bSmoothingBusy	ビット	実行中
bSmoothingActive	ビット	制御中
bSmoothingCommandAborted	ビット	実行中断
bSmoothingError	ビット	エラー
uwSmoothingErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
leSmoothingValidFrequency	倍精度実数	有効周波数

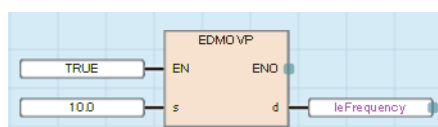
ラベル名	データ型	コメント
leSmoothingFilterPool	倍精度実数	フィルタ累積値

■FBD/LDプログラム

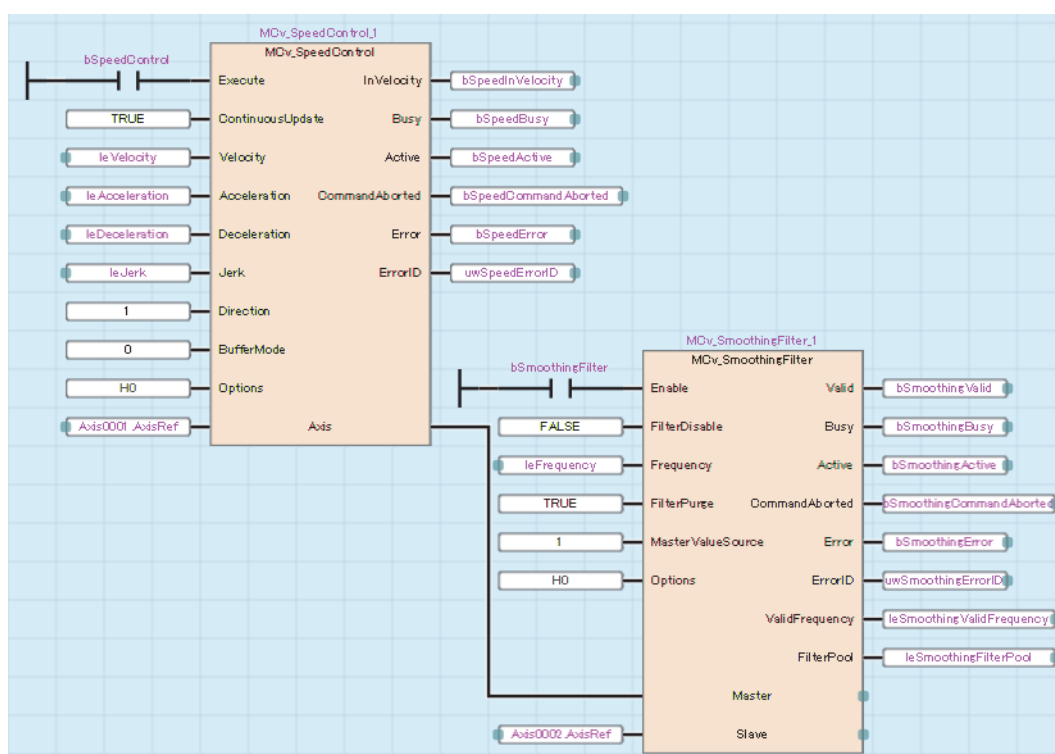
- 速度制御用データ設定



- 周波数



- 速度制御/スムージングフィルタ



■STプログラム

//-----速度制御用データ設定-----

```
leVelocity:= 5000.0;
leAcceleration:= 1000.0;
leDeceleration:= 1000.0;
leJerk:= 0.0;
```

//-----速度制御-----

```
MCv_SpeedControl_1(
  Axis:= Axis0001.AxisRef ,
  Execute:= bSpeedControl ,
  ContinuousUpdate:= TRUE ,
  Velocity:= leVelocity ,
  Acceleration:= leAcceleration ,
  Deceleration:= leDeceleration ,
  Jerk:= leJerk ,
  Direction:= MC_DIRECTION__mcPositiveDirection ,
  BufferMode:= MC_BUFFER_MODE__mcAborting ,
  Options:= H00000000 ,
  InVelocity=> bSpeedInVelocity ,
  Busy=> bSpeedBusy ,
  Active=> bSpeedActive ,
  CommandAborted=> bSpeedCommandAborted ,
  Error=> bSpeedError ,
  ErrorID=> uwSpeedErrorID
);
```

//-----周波数-----

```
leFrequency:= 10.00;
```

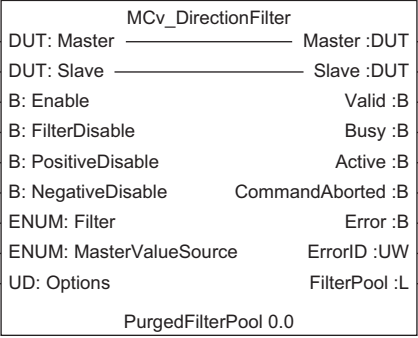
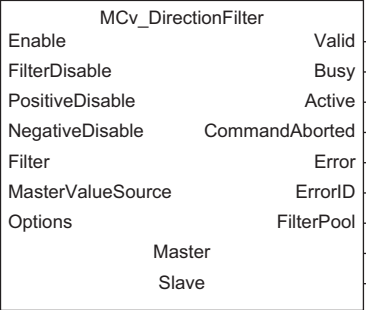
//-----スムージングフィルタ-----

```
MCv_SmoothingFilter_1(
  Master:= MCv_SpeedControl_1.Axis ,
  Slave:= Axis0002.AxisRef ,
  Enable:= bSmoothingFilter ,
  FilterDisable:= FALSE ,
  Frequency:= leFrequency ,
  FilterPurge:= TRUE ,
  MasterValueSource:= MC_SOURCE__mcSetValue ,
  Options:= H00000000 ,
  Valid=> bSmoothingValid ,
  Busy=> bSmoothingBusy ,
  Active=> bSmoothingActive ,
  CommandAborted=> bSmoothingCommandAborted ,
  Error=> bSmoothingError ,
  ErrorID=> uwSmoothingErrorID ,
  ValidFrequency=> leSmoothingValidFrequency ,
  FilterPool=> leSmoothingFilterPool
);
```

42.19 移動方向制限フィルタ

MCv_DirectionFilter

設定した移動方向に対する移動を制限するフィルタ処理を行います。

ラダー	FBD/LD	ST
		<pre>MCv_DirectionFilter(Master:= ?AXIS_REF?, Slave:= ?AXIS_REF?, Enable:= ?BOOL?, FilterDisable:= ?BOOL?, PositiveDisable:= ?BOOL?, NegativeDisable:= ?BOOL?, Filter:= ?INT?, MasterValueSource:= ?INT?, Options:= ?DWORD?, Valid=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?, FilterPool=> ?LREAL?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Master	主軸	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、 下記を参照してください。 1378ページ AxisName.AxisRef.(軸情報)
Slave	従軸	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、 下記を参照してください。 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Enable	有効	BOOL	常時	TRUE, FALSE	FALSE	TRUEにすると、MCv_DirectionFilter(移動方向制限フィルタ)を実行します。
FilterDisable	フィルタ無効	BOOL	常時	TRUE, FALSE	FALSE	フィルタ処理の方法を設定します。 • FALSE: フィルタ設定に従って動作します。 • TRUE: フィルタは無効となり、主軸(Master)の入力をそのまま従軸(Slave)に伝達します。 詳細は、下記を参照してください。 1777ページ フィルタ無効(FilterDisable)
PositiveDisable	正方向制限	BOOL	起動時	TRUE, FALSE	FALSE	TRUEにすると、正方向への軸動作を制限します。 主軸(Master)の正転動作時に従軸(Slave)は、動作しません。 正方向制限(PositiveDisable)と負方向制限(NegativeDisable)の両方をTRUEにすると、主軸(Master)を動作させても従軸(Slave)は動作しなくなります。 • FALSE: 制限なし • TRUE: 制限あり

入力変数	名称	データ型	取込	有効範囲	初期値	内容
NegativeDisable	負方向制限	BOOL	起動時	TRUE, FALSE	FALSE	TRUEにすると、負方向への軸動作を制限します。 主軸(Master)の逆転動作時に従軸(Slave)は、動作しません。 正方向制限(PositiveDisable)と負方向制限(NegativeDisable)の両方をTRUEにすると、主軸(Master)を動作しても従軸(Slave)は動作しなくなります。 • FALSE: 制限なし • TRUE: 制限あり
Filter	フィルタ動作	INT (MC_VELOCITY_LIMIT_MODE)	起動時	0, 2, 4	2	フィルタ動作を設定します。 • 0: 無視(Ignore) • 2: 切り捨て(Truncate) • 4: クランプ(減速時傾斜なし)(ClampWithoutRamp) 詳細は、下記を参照してください。 📖 1777ページ フィルタ動作(Filter)
MasterValueSource	主軸データソース選択	INT (MC_SOURCE)	起動時	1, 2, 101, 102	1	主軸(Master)のデータソースを設定します。 • 1: 指令現在値(mcSetValue) • 2: フィードバック値(mcActualValue) • 101: 最新指令現在値(mcLatestSetValue) • 102: 最新フィードバック値(mcLatestActualValue) 詳細は、下記を参照してください。 📖 1778ページ 主軸データソース選択(MasterValueSource)
Options	オプション	DWORD(HEX)	起動時	00000000H	00000000H	「00000000H」を設定してください。 *: 「00000000H」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

■出力変数

出力変数	名称	データ型	初期値	内容
Valid	出力値有効	BOOL	FALSE	出力値が有効なときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MCv_DirectionFilter(移動方向制限フィルタ)を実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	MCv_DirectionFilter(移動方向制限フィルタ)が軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	MCv_DirectionFilter(移動方向制限フィルタ)の実行が中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 📖 使用するコントローラのユーザーズマニュアル
FilterPool	フィルタ累積値	LREAL	0.0	フィルタ処理にて累積した従軸の移動量を出力します。 フィルタ累積値(FilterPool)が「0.0」以外の状態で、有効(Enable)をFALSEにする場合、フィルタ累積値(FilterPool)はFALSEセット時の状態を保持します。 フィルタ累積値(FilterPool)は、次回有効(Enable)をTRUEにしたときにクリアします。 フィルタ累積値(FilterPool)の値が位置決め範囲を超えた場合、フィルタ累積値オーバー警告(イベントコード: 00D13H)となります。 警告検出時、フィルタ累積値(FilterPool)の誤差が大きくなる場合があります。

■公開変数

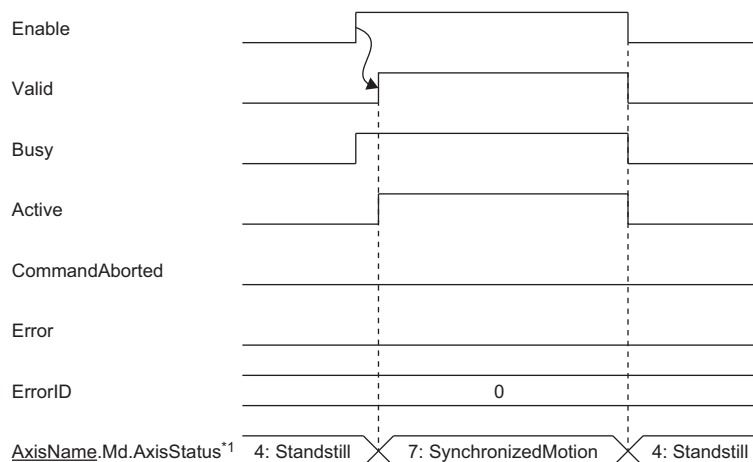
公開変数	名称	データ型	初期値	内容
PurgedFilterPool	破棄されたフィルタ累積値	LREAL	0.0	フィルタ無効(FilterDisable)をTRUEにすることで破棄されたフィルタ累積値(FilterPool)を格納します。 破棄されたフィルタ累積値(PurgedFilterPool)は、累積しません。 FB起動時に0クリアします。

機能

- 移動方向制限フィルタは、主軸の移動に対して、従軸の移動を一方向に制限するフィルタです。
- 移動方向制限フィルタは、主軸データソース選択(MasterValueSource)で設定した主軸(Master)の値に対して、フィルタ処理を実行した結果を従軸(Slave)に出力します。
- 移動方向制限フィルタ実行中の従軸(Slave)の軸状態(AxisName.Md.AxisStatus)は、「7: 同期運転中(SynchronizedMotion)」になります。
- 従軸が他の動作FB中に、移動方向制限フィルタを始動した場合、始動要求を無視し、始動不可(エラーコード: 1AFCH)となります。従軸(Slave)の軸状態(AxisName.Md.AxisStatus)が「4: 待機中(Standstill)」のときに始動してください。
- 移動方向制限フィルタ実行中に、多重起動可能なファンクションブロックはMC_Stop(強制停止)のみです。
- 有効(Enable)をFALSEにする場合は、従軸が停止している状態で行ってください。従軸動作中に有効(Enable)をFALSEにすると、従軸は即停止します。

■タイミングチャート

- 正常完了の場合



*1 従軸(Slave)の軸状態です。

- 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

📖 1370ページ 有効(Enable)型によるモーション制御FBの基本動作

■フィルタ無効(FilterDisable)

フィルタ処理の方法を設定します。

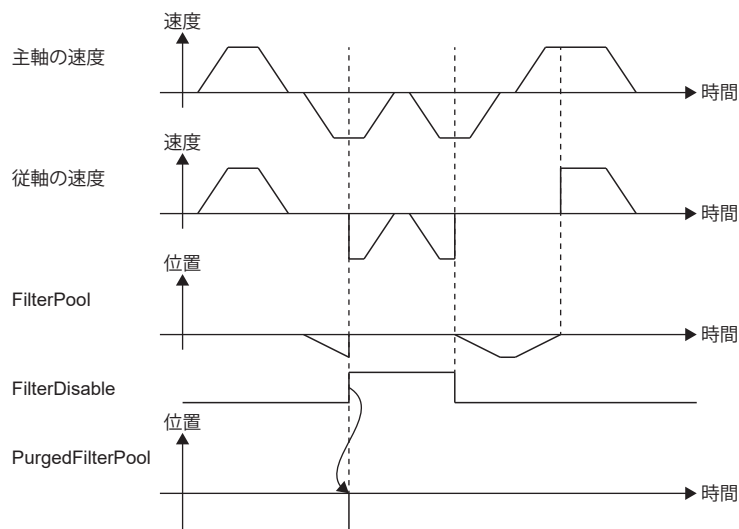
設定値	内容
FALSE	フィルタ設定に従って動作します。
TRUE	フィルタは無効となり、主軸(Master)の入力をそのまま従軸(Slave)に伝達します。

- ・フィルタ無効(FilterDisable)をTRUEにした場合、フィルタ累積値(FilterPool)は、「0.0」になり、破棄されたフィルタ累積値(PurgedFilterPool)へ、破棄したフィルタ累積値(FilterPool)を格納します。

例

下記を設定した場合

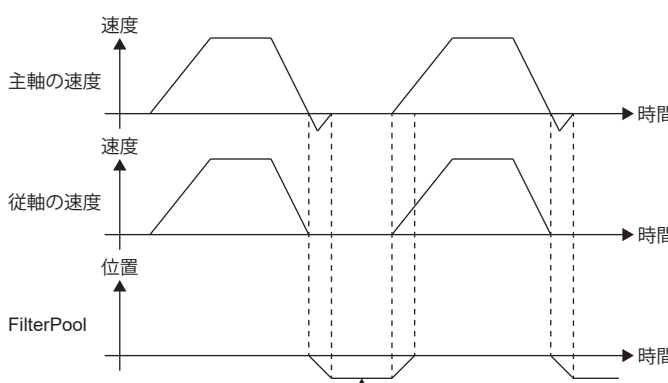
入力ピン	設定値
フィルタ動作(Filter)	4: クランプ(減速時傾斜なし)(ClampWithoutRamp)
正方向制限(PositiveDisable)	FALSE(制限なし)
負方向制限(NegativeDisable)	TRUE(制限あり)



■フィルタ動作(Filter)

- ・フィルタ動作(Filter)の設定によって、制限時の従軸の動作を設定します。

設定値	内容
0: 無視(Ignore)	・移動方向制限を行いません。 ・フィルタ累積値(FilterPool)を更新しません。
2: 切り捨て(Truncate)	・制限した方向への移動量を無視し、フィルタ累積値(FilterPool)に加算しません。 ・制限した方向への移動量が切り捨てられるため、制限した方向への移動量分主軸と従軸の位置関係がずれます。 <例> 正方向制限(PositiveDisable)がFALSE、負方向制限(NegativeDisable)がTRUEの場合

設定値	内容
4: クランプ(減速時傾斜なし) (ClampWithoutRamp)	- 制限した方向への移動量をフィルタ累積値(FilterPool)に加算します。 - フィルタ累積値(FilterPool)は、反転方向への移動に反映します。そのため、制限方向への動作を繰り返しても、フィルタ累積値(FilterPool)が「0.0」の状態の主軸と従軸の位置関係がずれることはありません。 <例> 正方向制限(PositiveDisable)がFALSE(制限なし)、負方向制限(NegativeDisable)がTRUE(制限あり)の場合  フィルタ累積値として累積し、許可方向への入力移動量になると反映する

・フィルタ動作(Filter)に設定範囲外の値を設定した場合、フィルタ動作指定範囲外(エラーコード: 1AE7H)となります。

■主軸データソース選択(MasterValueSource)

従軸(Slave)が単軸同期制御を実行する主軸(Master)の位置の種別を設定します。

主軸(Master)と従軸(Slave)の演算周期が異なる場合やFBの実行順により、動作が変わる場合があります。

実軸以外の軸種別を主軸(Master)に設定して主軸データソース選択(MasterValueSource)にフィードバック値を設定した場合は、指令現在値と同じ値で動作します。

設定値	内容
1: 指令現在値(mcSetValue)	前回の演算周期における主軸の指令位置を使用します。
2: フィードバック値(mcActualValue)	前回の演算周期における主軸のフィードバック位置を使用します。
101: 最新指令現在値(mcLatestSetValue)	今回の演算周期における主軸の指令位置を使用します。
102: 最新フィードバック値(mcLatestActualValue)	今回の演算周期における主軸のフィードバック位置を使用します。

Point

主軸データソース選択(MasterValueSource)に「1: 指令現在値(mcSetValue)」、「101: 最新指令現在値(mcLatestSetValue)」を設定し、主軸がサーボアラームや緊急停止によってサーボOFFとなった場合、値の変化量が大きくなる場合があります。主軸データソース選択(MasterValueSource)に「2: フィードバック値(mcActualValue)」、「102: 最新フィードバック値(mcLatestActualValue)」を設定することで防ぐことができます。

注意事項

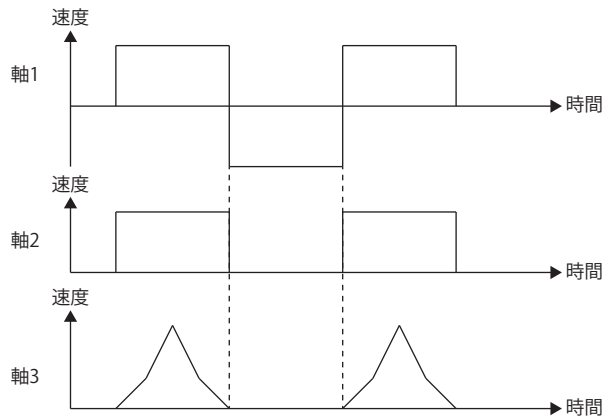
- 指令フィルタの動作中にフィルタ無効(FilterDisable)の切り換えを行うと、従軸(Slave)の速度が急激に変化することがあります。
- 指令フィルタを有効にした後、従軸(Slave)に停止要因が発生して従軸(Slave)や後段の同期制御ファンクションブロックに主軸(Master)の指令を伝達しなくなった場合、同期位置関係が崩れます。指令フィルタを再度有効にする前に、必要に応じて同期位置合わせを行ってください。
- 演算周期混在機能使用時、主軸と従軸の制御演算周期設定が異なる場合は制約が生じます。詳細は、各機能および下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

プログラム例

カム動作の方向を制限するフィルタ処理を行うプログラム例を下記に示します。

・動作



・設定

項目	設定値
カムID設定	1

■軸

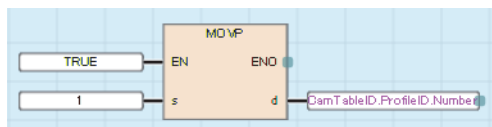
軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REF	軸1
2	Axis0002	AXIS_REF	軸2
3	Axis0003	AXIS_REF	軸3

■使用するラベル

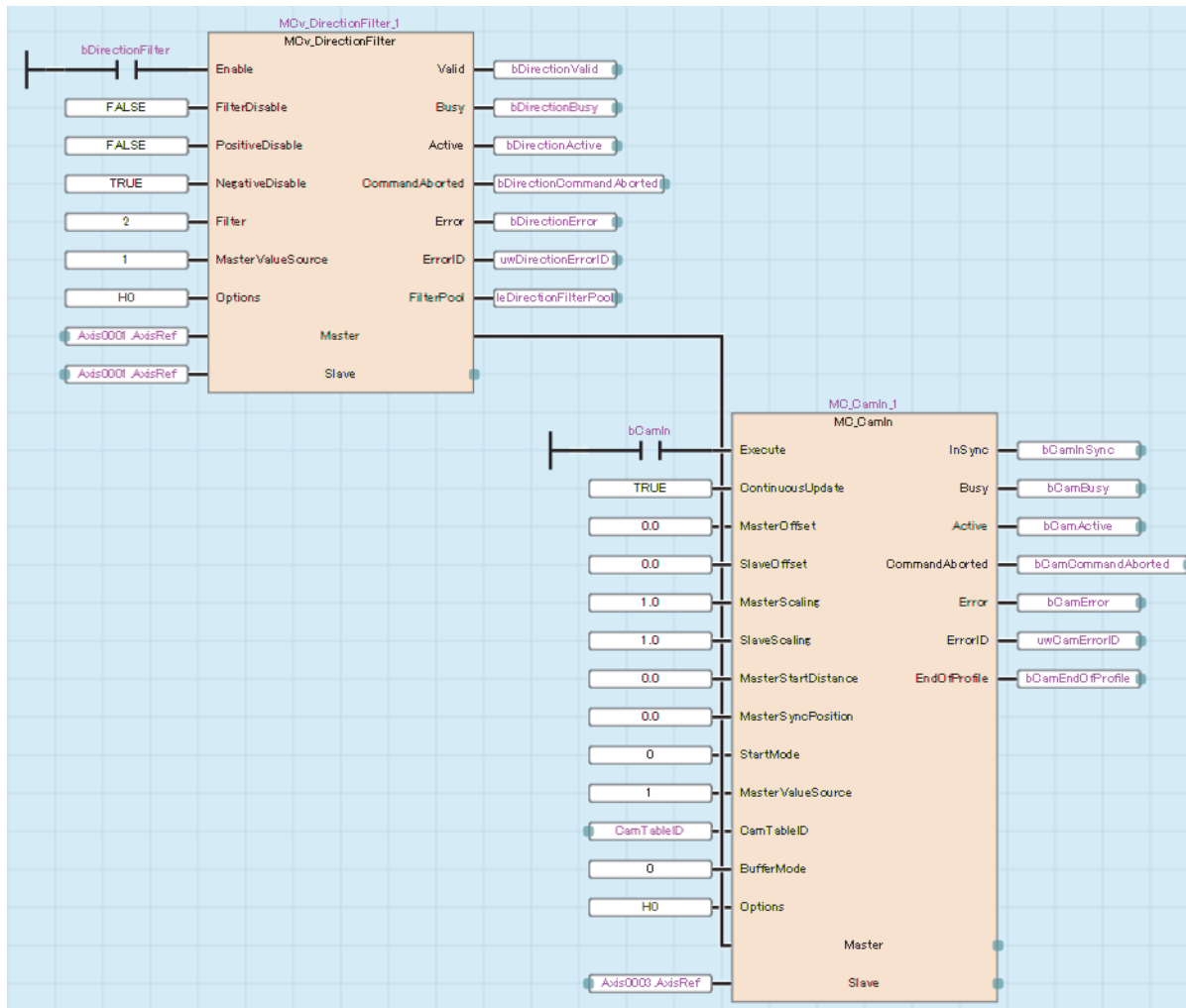
ラベル名	データ型	コメント
MCv_DirectionFilter_1	MCv_DirectionFilter	移動方向制限フィルタFB
bDirectionFilter	ビット	移動方向制限フィルタ指令
bDirectionValid	ビット	出力値有効
bDirectionBusy	ビット	実行中
bDirectionActive	ビット	制御中
bDirectionCommandAborted	ビット	実行中断
bDirectionError	ビット	エラー
uwDirectionErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
leDirectionFilterPool	倍精度実数	破棄されたフィルタ累積値
MC_CamIn_1	MC_CamIn	カム動作開始FB
bCamIn	ビット	カム動作指令
CamTableID	MC_CAM_ID	カムID
bCamInSync	ビット	同期中
bCamBusy	ビット	実行中
bCamActive	ビット	制御中
bCamCommandAborted	ビット	実行中断
bCamError	ビット	エラー
uwCamErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
bCamEndOfProfile	ビット	カムサイクル完了

■FBD/LDプログラム

- カムID設定



- 移動方向制限フィルタ/カム動作



■STプログラム

//-----移動方向制限フィルタ-----

```
MCv_DirectionFilter_1(
  Master:= Axis0001.AxisRef ,
  Slave:= Axis0002.AxisRef ,
  Enable:= bDirectionFilter ,
  FilterDisable:= FALSE ,
  PositiveDisable:= FALSE ,
  NegativeDisable:= TRUE ,
  Filter:= MC_VELOCITY_LIMIT_MODE__Truncate ,
  MasterValueSource:= MC_SOURCE__mcSetValue ,
  Options:= H00000000 ,
  Valid=> bDirectionValid ,
  Busy=> bDirectionBusy ,
  Active=> bDirectionActive ,
  CommandAborted=> bDirectionCommandAborted ,
  Error=> bDirectionError ,
  ErrorID=> uwDirectionErrorID ,
  FilterPool=> leDirectionFilterPool
);
```

//カムID設定

```
CamTableID.ProfileID.Number:= 1;
```

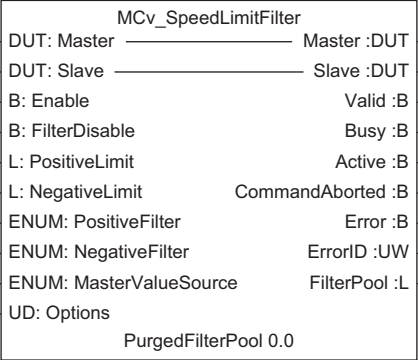
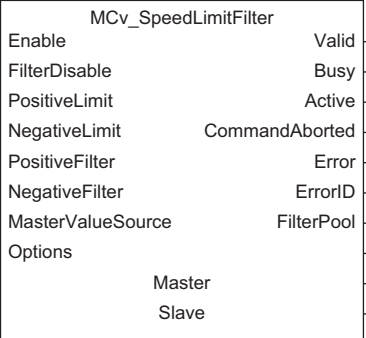
//カム動作

```
MC_CamIn_1(
  Master:= MCv_DirectionFilter_1.Slave ,
  Slave:= Axis0003.AxisRef ,
  Execute:= bCamIn ,
  ContinuousUpdate:= TRUE ,
  MasterOffset:= 0.0 ,
  SlaveOffset:= 0.0 ,
  MasterScaling:= 1.0 ,
  SlaveScaling:= 1.0 ,
  MasterStartDistance:= 0.0 ,
  MasterSyncPosition:= 0.0 ,
  StartMode:= MC_START_MODE__mclImmediate ,
  MasterValueSource:= MC_SOURCE__mcSetValue ,
  CamTableID:= CamTableID ,
  BufferMode:= MC_BUFFER_MODE__mcAborting ,
  Options:= H00000000 ,
  InSync=> bCamInSync ,
  Busy=> bCamBusy ,
  Active=> bCamActive ,
  CommandAborted=> bCamCommandAborted ,
  Error=> bCamError ,
  ErrorID=> uwCamErrorID ,
  EndOfProfile=> bCamEndOfProfile
);
```

42.20 速度制限フィルタ

MCv_SpeedLimitFilter

設定した制限値の速度に制限するフィルタ処理を行います。

ラダー	FBD/LD	ST
		<pre>MCv_SpeedLimitFilter(Master:= ?AXIS_REF?, Slave:= ?AXIS_REF?, Enable:= ?BOOL?, FilterDisable:= ?BOOL?, PositiveLimit:= ?LREAL?, NegativeLimit:= ?LREAL?, PositiveFilter:= ?INT?, NegativeFilter:= ?INT?, MasterValueSource:= ?INT?, Options:= ?DWORD?, Valid=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?, FilterPool=> ?LREAL?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Master	主軸	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📖 1378ページ AxisName.AxisRef.(軸情報)
Slave	従軸	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📖 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Enable	有効	BOOL	常時	TRUE, FALSE	FALSE	TRUEにすると、MCv_SpeedLimitFilter(速度制限フィルタ)を実行します。
FilterDisable	フィルタ無効	BOOL	常時	TRUE, FALSE	FALSE	フィルタ処理の方法を設定します。 ・FALSE: フィルタ設定に従って動作します。 ・TRUE: フィルタは無効となり、主軸(Master)の入力をそのまま従軸(Slave)に伝達します。 詳細は、下記を参照してください。 📖 1785ページ フィルタ無効(FilterDisable)
PositiveLimit	正方向制限値	LREAL	起動時	0.0, 0.0001~2500000000.0	0.0	正方向の速度制限値を設定します。 主軸(Master)の正方向入力速度に対して、従軸(Slave)に出力する速度の上限値を設定します。 「0.0」を設定した場合、正方向へ従軸は動作しません。 範囲外の値を設定した場合は、方向制限値指定範囲外(エラーコード: 1AE8H)となります。
NegativeLimit	負方向制限値	LREAL	起動時	0.0, 0.0001~2500000000.0	0.0	負方向の速度制限値を設定します。 主軸(Master)の負方向入力速度に対して、従軸(Slave)に出力する速度の上限値を設定します。 「0.0」を設定した場合、負方向へ従軸は動作しません。 範囲外の値を設定した場合は、方向制限値指定範囲外(エラーコード: 1AE8H)となります。

入力変数	名称	データ型	取込	有効範囲	初期値	内容
PositiveFilter	正方向フィルタ動作	INT (MC_VELOCITY_LIMIT_MODE)	起動時	0, 2, 4	2	正方向制限値を超えた場合のフィルタ動作を設定します。 • 0: 無視(Ignore) • 2: 切り捨て(Truncate) • 4: クランプ(減速時傾斜なし) (ClampWithoutRamp) 詳細は、下記を参照してください。 1785ページ 正方向フィルタ動作 (PositiveFilter)/負方向フィルタ動作(NegativeFilter)
NegativeFilter	負方向フィルタ動作	INT (MC_VELOCITY_LIMIT_MODE)	起動時	0, 2, 4	2	負方向制限値を超えた場合のフィルタ動作を設定します。 • 0: 無視(Ignore) • 2: 切り捨て(Truncate) • 4: クランプ(減速時傾斜なし) (ClampWithoutRamp) 詳細は、下記を参照してください。 1785ページ 正方向フィルタ動作 (PositiveFilter)/負方向フィルタ動作(NegativeFilter)
MasterValueSource	主軸データソース選択	INT (MC_SOURCE)	起動時	1, 2, 101, 102	1	主軸(Master)のデータソースを設定します。 • 1: 指令現在値(mcSetValue) • 2: フィードバック値(mcActualValue) • 101: 最新指令現在値(mcLatestSetValue) • 102: 最新フィードバック値 (mcLatestActualValue) 詳細は、下記を参照してください。 1786ページ 主軸データソース選択 (MasterValueSource)
Options	オプション	DWORD(HEX)	起動時	00000000H	00000000H	「00000000H」を設定してください。 *: 「00000000H」以外を設定した場合、Options 範囲外(エラーコード: 1ABBH)となります。

■出力変数

出力変数	名称	データ型	初期値	内容
Valid	出力値有効	BOOL	FALSE	出力値が有効なときに、TRUEになります。
Busy	実行中	BOOL	FALSE	MCv_SpeedLimitFilter(速度制限フィルタ)を実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	FBが軸を制御中であることを示します。 MCv_SpeedLimitFilter(速度制限フィルタ)が軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	MCv_SpeedLimitFilter(速度制限フィルタ)の実行が中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 使用コントローラのユーザーズマニュアル
FilterPool	フィルタ累積値	LREAL	0.0	フィルタ処理にて累積した従軸の移動量を出力します。 フィルタ累積値(FilterPool)が「0.0」以外の状態で、有効(Enable)をFALSEにする場合、フィルタ累積値(FilterPool)はFALSEセット時の状態を保持します。 フィルタ累積値(FilterPool)は、次回有効(Enable)をTRUEにしたときにクリアします。 フィルタ累積値(FilterPool)の値が位置決め範囲を超えた場合、フィルタ累積値オーバ警告(イベントコード: 00D13H)となります。 警告検出時、フィルタ累積値(FilterPool)の誤差が大きくなる場合があります。

■公開変数

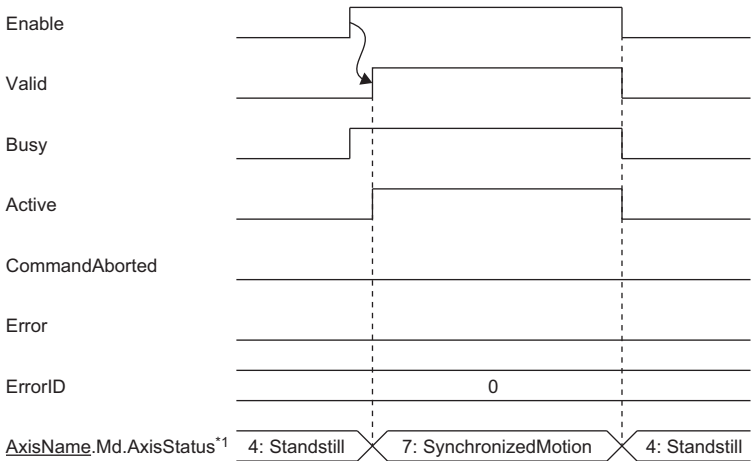
公開変数	名称	データ型	初期値	内容
PurgedFilterPool	破棄されたフィルタ累積値	LREAL	0.0	フィルタ無効(FilterDisable)をTRUEにすることで破棄したフィルタ累積値(FilterPool)を格納します。破棄されたフィルタ累積値(PurgedFilterPool)は、累積しません。FB始動時に0クリアします。

機能

- 速度制限フィルタは、主軸の入力速度に対して、特定の制限値を設け、設定した制限値の速度を従軸に出力するフィルタです。
- 速度制限フィルタは、主軸データソース選択(MasterValueSource)で設定した主軸(Master)の値に対して、フィルタ処理を実行した結果を従軸(Slave)に出力します。
- 速度制限フィルタ実行中の従軸(Slave)の軸状態(AxisName.Md.AxisStatus)は、「7: 同期運転中(SynchronizedMotion)」になります。
- 従軸が他の動作FB中に、速度制限フィルタを始動した場合、始動要求を無視し、始動不可(エラーコード: 1AFCH)となります。従軸(Slave)の軸状態(AxisName.Md.AxisStatus)が「4: 待機中(Standstill)」のときに始動してください。
- 速度制限フィルタ実行中に、多重起動可能なファンクションブロックはMC_Stop(強制停止)のみです。
- 有効(Enable)をFALSEにする場合は、従軸が停止している状態で行ってください。従軸動作中に有効(Enable)をFALSEにすると、従軸は即停止します。

■タイミングチャート

- 正常完了の場合



*1 従軸(Slave)の軸状態です。

- 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

1370ページ 有効(Enable)型によるモーション制御FBの基本動作

■フィルタ無効(FilterDisable)

フィルタ処理の方法を設定します。

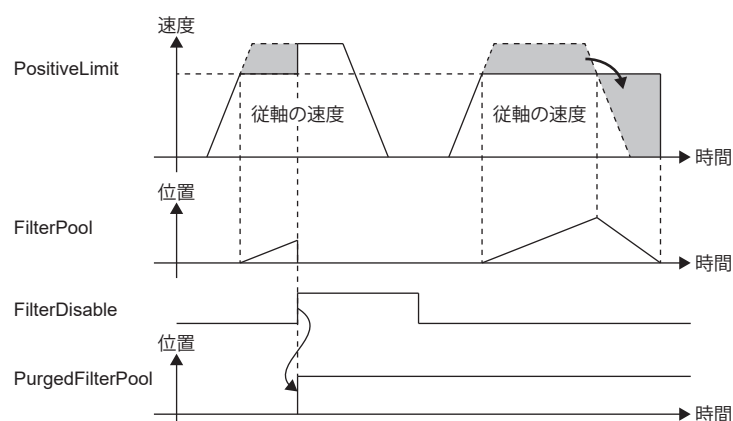
設定値	内容
FALSE	フィルタ設定に従って動作します。
TRUE	フィルタは無効となり、主軸(Master)の入力をそのまま従軸(Slave)に伝達します。

- フィルタ無効(FilterDisable)をTRUEにした場合、フィルタ累積値(FilterPool)は、「0.0」になり、破棄されたフィルタ累積値(PurgedFilterPool)へ、破棄したフィルタ累積値(FilterPool)を格納します。

例

下記を設定した場合

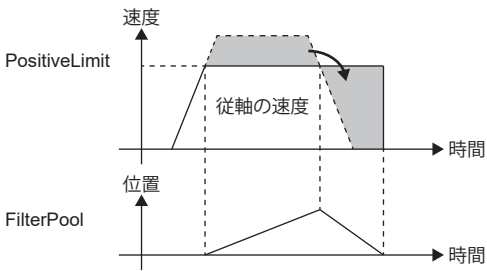
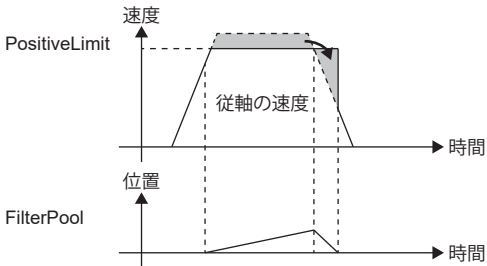
入力ピン	設定値
正方向フィルタ動作(PositiveFilter)	4: クランプ(減速時傾斜なし)(ClampWithoutRamp)



■正方向フィルタ動作(PositiveFilter)/負方向フィルタ動作(NegativeFilter)

- 正方向フィルタ動作(PositiveFilter)と負方向フィルタ動作(NegativeFilter)の設定によって、制限時の従軸の動作を設定します。

設定値	内容
0: 無視(Ignore)	- 速度制限を行いません。 - フィルタ累積値(FilterPool)を更新しません。
2: 切り捨て(Truncate)	- 制限値を超えた分は、従軸(Slave)に出力しません。 - 制限した移動量が切り捨てられるため、切り捨てた分、主軸と従軸の位置関係がずれます。

設定値	内容
4: クランプ(減速時傾斜なし) (ClampWithoutRamp)	- 制限値を超えた分は、フィルタ累積値(FilterPool)として累積し、累積分は遅延して出力します。フィルタ累積値(FilterPool)の出力は減速動作をしません。 - 制限値に「0.0」を設定した場合、該当方向への主軸の入力は、すべてフィルタ累積値(FilterPool)として累積し、従軸へ遅延して出力を行いません。累積分は逆方向の主軸の入力で減算し、フィルタ累積値(FilterPool)が「0.0」になるまで従軸は動作しません。 ■フィルタ累積値が減速距離より大きい場合  ■フィルタ累積値が減速距離より小さい場合 

・フィルタ動作(Filter)に設定範囲外の値を設定した場合、フィルタ動作指定範囲外(エラーコード: 1AE7H)となります。

■主軸データソース選択(MasterValueSource)

従軸(Slave)が単軸同期制御を実行する主軸(Master)の位置の種別を設定します。

主軸(Master)と従軸(Slave)の演算周期が異なる場合やFBの実行順により、動作が変わる場合があります。

実軸以外の軸種別を主軸(Master)に設定して主軸データソース選択(MasterValueSource)にフィードバック値を設定した場合は、指令現在値と同じ値で動作します。

設定値	内容
1: 指令現在値(mcSetValue)	前回の演算周期における主軸の指令位置を使用します。
2: フィードバック値(mcActualValue)	前回の演算周期における主軸のフィードバック位置を使用します。
101: 最新指令現在値(mcLatestSetValue)	今回の演算周期における主軸の指令位置を使用します。
102: 最新フィードバック値(mcLatestActualValue)	今回の演算周期における主軸のフィードバック位置を使用します。

Point

主軸データソース選択(MasterValueSource)に「1: 指令現在値(mcSetValue)」、「101: 最新指令現在値(mcLatestSetValue)」を設定し、主軸がサーボアラームや緊急停止によってサーボOFFとなった場合、値の変化量が大きくなる場合があります。主軸データソース選択(MasterValueSource)に「2: フィードバック値(mcActualValue)」、「102: 最新フィードバック値(mcLatestActualValue)」を設定することで防ぐことができます。

注意事項

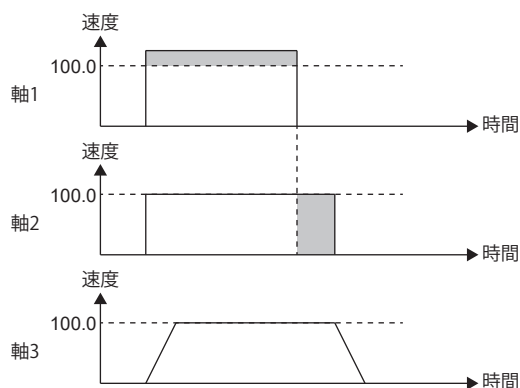
- 指令フィルタの動作中にフィルタ無効(FilterDisable)の切換えを行うと、従軸(Slave)の速度が急激に変化することがあります。
- 指令フィルタを有効にした後、従軸(Slave)に停止要因が発生して従軸(Slave)や後段の同期制御ファンクションブロックに主軸(Master)の指令を伝達しなくなった場合、同期位置関係が崩れます。指令フィルタを再度有効にする前に、必要に応じて同期位置合わせを行ってください。
- 演算周期混在機能使用時、主軸と従軸の制御演算周期設定が異なる場合は制約が生じます。詳細は、各機能および下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

プログラム例

同期エンコーダの速度を制限するフィルタ処理を行うプログラム例を下記に示します。

・動作



・設定

項目		設定値
正方向/負方向制限値	正方向制限値	100.0
	負方向制限値	0.0
バックラッシュ量		1.0

■軸

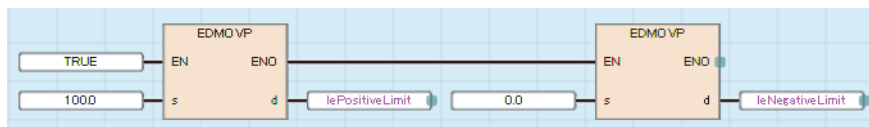
軸No.	ラベル名	データ型	コメント
1	EncoderAxis0001	AXIS_REF	軸1
2	Axis0002	AXIS_REF	軸2
3	Axis0003	AXIS_REF	軸3

■使用するラベル

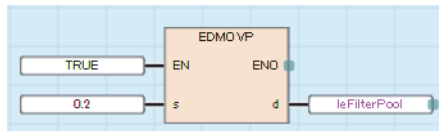
ラベル名	データ型	コメント
MCv_SpeedLimitFilter_1	MCv_SpeedLimitFilter	速度制限フィルタFB
bSpeedLimitFilter	ビット	速度制限フィルタ指令
lePositiveLimit	倍精度実数	正方向制限値
leNegativeLimit	倍精度実数	負方向制限値
bSpeedLimitValid	ビット	出力値有効
bSpeedLimitBusy	ビット	実行中
bSpeedLimitActive	ビット	制御中
bSpeedLimitCommandAborted	ビット	実行中断
bSpeedLimitError	ビット	エラー
uwSpeedLimitErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
leFilterPool	倍精度実数	フィルタ累積値
MCv_SmoothingFilter_1	MCv_SmoothingFilter	スムージングフィルタFB
bSmoothingFilter	ビット	スムージングフィルタ指令
leFrequency	倍精度実数	周波数
bSmoothingValid	ビット	出力値有効
bSmoothingInVelocity	ビット	目標速度到達
bSmoothingBusy	ビット	実行中
bSmoothingActive	ビット	制御中
bSmoothingCommandAborted	ビット	実行中断
bSmoothingError	ビット	エラー
uwSmoothingErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
leSmoothingValidFrequency	倍精度実数	有効周波数
leSmoothingFilterPool	倍精度実数	フィルタ累積値

■FBD/LDプログラム

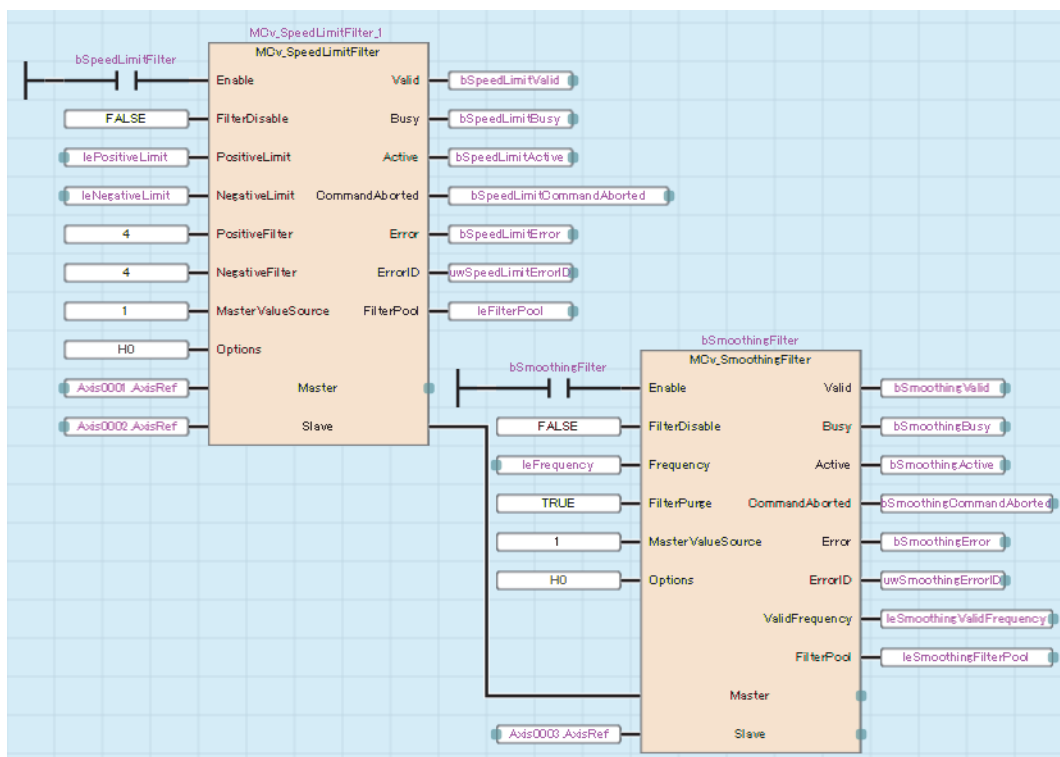
- 正方向/負方向制限値



- 周波数



- 速度制限フィルタ/スムージングフィルタ



■STプログラム

//-----正方向/負方向制限値-----

lePositiveLimit:= 100.0;

leNegativeLimit:= 0.0;

//-----速度制限フィルタ-----

```
MCv_SpeedLimitFilter_1(
  Master:= EncoderAxis0001.AxisRef ,
  Slave:= Axis0002.AxisRef ,
  Enable:= bSpeedLimitFilter ,
  FilterDisable:= FALSE ,
  PositiveLimit:= lePositiveLimit ,
  NegativeLimit:= leNegativeLimit ,
  PositiveFilter:= MC_VELOCITY_LIMIT_MODE__ClampWithoutRamp ,
  NegativeFilter:= MC_VELOCITY_LIMIT_MODE__ClampWithoutRamp ,
  MasterValueSource:= MC_SOURCE__mcSetValue ,
  Options:= H00000000 ,
  Valid=> bSpeedLimitValid ,
  Busy=> bSpeedLimitBusy ,
  Active=> bSpeedLimitActive ,
  CommandAborted=> bSpeedLimitCommandAborted ,
  Error=> bSpeedLimitError ,
  ErrorID=> uwSpeedLimitErrorID ,
  FilterPool=> leFilterPool
);
```

//-----周波数-----

leFrequency:= 0.20;

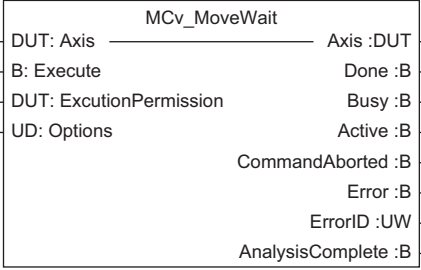
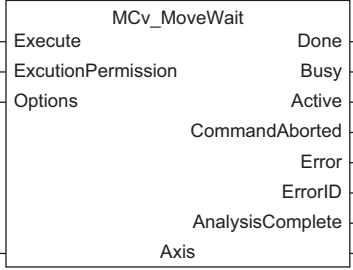
//-----スムージングフィルタ-----

```
MCv_SmoothingFilter_1(
  Master:= MCv_SpeedControl_1.Slave ,
  Slave:= Axis0003.AxisRef ,
  Enable:= bSmoothingFilter ,
  FilterDisable:= FALSE ,
  Frequency:= leFrequency ,
  FilterPurge:= TRUE ,
  MasterValueSource:= MC_SOURCE__mcSetValue ,
  Options:= H00000000 ,
  Valid=> bSmoothingValid ,
  Busy=> bSmoothingBusy ,
  Active=> bSmoothingActive ,
  CommandAborted=> bSmoothingCommandAborted ,
  Error=> bSmoothingError ,
  ErrorID=> uwSmoothingErrorID ,
  ValidFrequency=> leSmoothingValidFrequency ,
  FilterPool=> leSmoothingFilterPool
);
```

42.21 軸制御開始ウェイト

MCv_MoveWait

指定した軸の多重起動した動作系FBの動作を、実行許可信号の入力により即時実行します。

ラダー	FBD/LD	ST
		<pre>MCv_MoveWait(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, ExecutionPermission:= ?MC_INPUT_REF?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?, AnalysisComplete=> ?BOOL?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📖 1378ページ AxisName.AxisRef. (軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMCv_MoveWait(軸制御開始ウェイト)を実行します。
ExecutionPermission	実行許可信号	MC_INPUT_REF	起動時	—	省略不可	軸制御開始ウェイトの解除に使用する信号を設定します。
Options	オプション	DWORD(HEX)	起動時	00000000H	00000000H	「00000000H」を設定してください。 「00000000H」以外を設定した場合、とエラーになります。

■出力変数

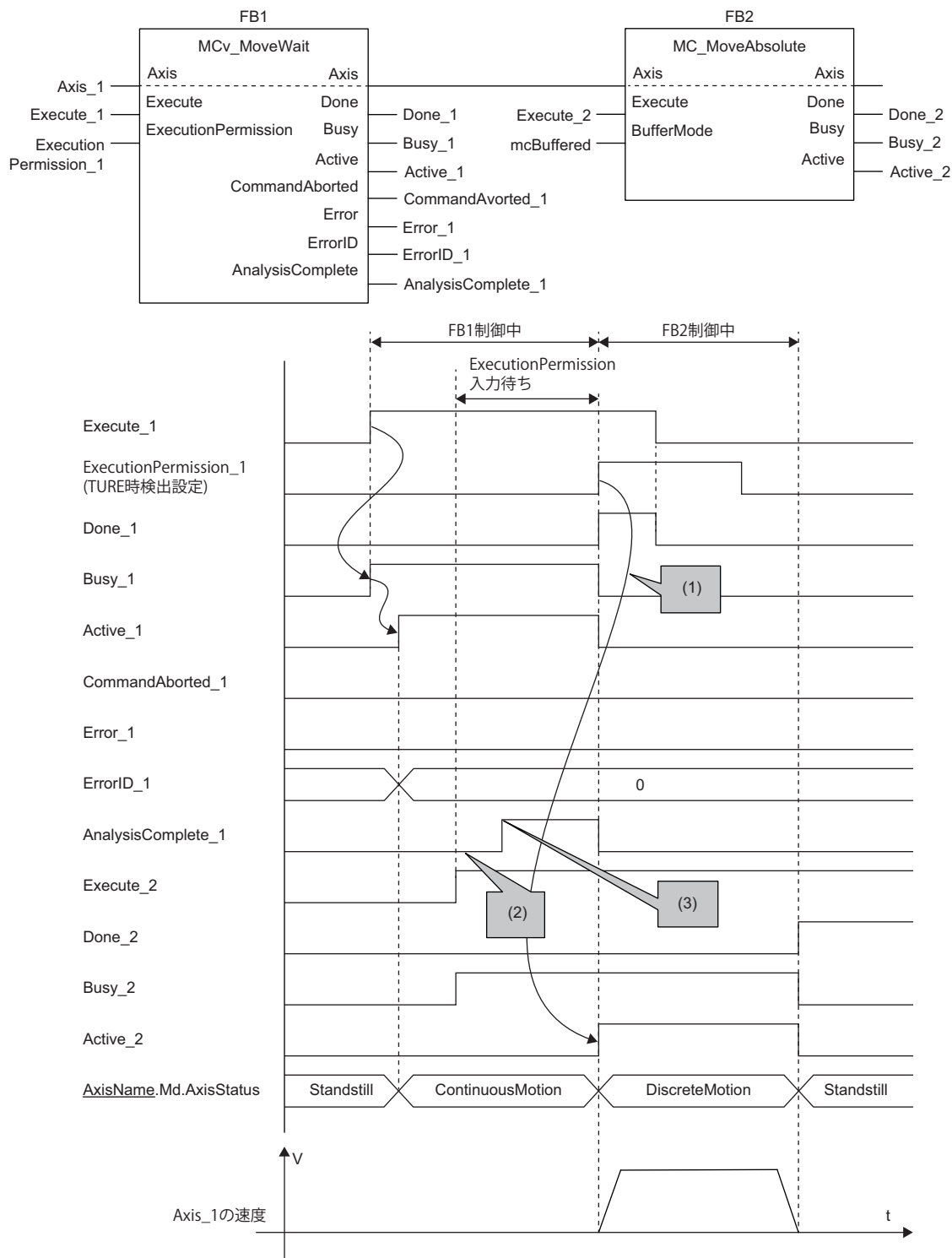
出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	ExecutionPermission(実行許可信号)で設定した信号の入力検出、かつAnalysisComplete(解析完了)がTRUEになったときに、TRUEとなり、MCv_MoveWait(軸制御開始ウェイト)の制御が完了したことを示します。
Busy	実行中	BOOL	FALSE	MCv_MoveWait(軸制御開始ウェイト)が実行中のときに、TRUEになります。
Active	制御中	BOOL	FALSE	MCv_MoveWait(軸制御開始ウェイト)が軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	他のFBにより実行中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 📖 使用するコントローラのユーザーズマニュアル
AnalysisComplete	解析完了	BOOL	FALSE	多重起動した動作系FBの解析が完了ときに、TRUEになります。

機能

- 多重起動した単軸制御の動作系FBの軸動作をExecutionPermission(実行許可信号)の入力タイミングで即時開始します。
- MCv_MoveWait(軸制御開始ウェイト)起動中に他の動作系FBの多重起動すると、多重起動したFBは解析完了まで実行され、ExecutionPermission(実行許可信号)が入力されるまで待機します。多重起動したFBの解析完了のタイミングはAnalysisComplete(解析完了)で確認できます。
- AnalysisComplete(解析完了)がTRUEになった後にExecutionPermission(実行許可信号)を実行してください。
- ExecutionPermission(実行許可信号)が入力されるまでの間、軸状態(AxisName.Md.AxisStatus)は「6: 連続動作運転中(ContinuousMotion)」になります。
- ExecutionPermission(実行許可信号)が入力されると、信号入力を検出した1演算周期後に多重起動したFBを実行します。このとき、軸状態(AxisName.Md.AxisStatus)は多重起動したFBに応じた状態に遷移します。
- AnalysisComplete(解析完了)がTRUEになった後に、ExecutionPermission(実行許可信号)を入力してください。

■ タイミングチャート

・ 正常完了の場合



(1): ExecutionPermission入力を検出した1演算周期後にFB2が起動します。

(2): FB2の解析中

(3): FB2の解析完了

・ 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■ExecutionPermission(実行許可信号)

MCv_MoveWait(軸制御開始ウェイト)の解除に使用する信号を指定します。

ExecutionPermission(実行許可信号)は、MC_INPUT_REF構造体で設定します。MC_INPUT_REF構造体については、下記を参照してください。

1420ページ MC_INPUT_REF

ExecutionPermission(実行許可信号)で設定する入力信号(ExecutionPermission.Signal)の設定範囲を下記に示します。

入力信号(ExecutionPermission.Signal)の型は、SIGNAL_SELECT構造体となります。

構造体	変数名	型	設定範囲
SIGNAL_SELECT(信号選択)	Source.Target(信号)*1	TARGET_REF 1421ページ TARGET_REF(入力信号)	■型 • BOOL ■データ種別 • [OBJ] • [VAR] • [DEV] • [CONST]
	Detection(信号検出方法)	INT (MC_SIGNAL_LOGIC) 1452ページ MC_SIGNAL_LOGIC	• 0: TRUE時検出(HighLevel) • 1: FALSE時検出(LowLevel)
	CompensationTime(補正時間)*2	LREAL	-5.0~0.0[s]
	FilterTime(フィルタ時間)	LREAL	0.0~5.0[s]

*1 外部信号高精度入力の使用はできません。外部信号高精度入力用の信号を指定した場合、通常の信号と同じ精度で動作します。

*2 補正時間(ExecutionPermission.Signal.CompensationTime)に1演算周期より小さい値を指定した場合、0.0(補正なし)として動作します。

[OBJ]指定時に対象修飾を省略した場合は、軸(Axis)の軸のオブジェクトを参照します。ただし軸(Axis)の軸が局アドレスを持たない、または局アドレスが未設定の場合、外部信号選択不正(エラーコード: 1B31H)を出力します。

[OBJ]指定時に参照する局が軸エミュレート機能が有効かつ局アドレスが設定済みの場合、エミュレートしているオブジェクトを参照します。

ExecutionPermission(実行許可信号)の進めを補正する場合、または多重起動した動作系FBの実行タイミングを調整したい場合は、補正時間(ExecutionPermission.Signal.CompensationTime)を設定してください。

ExecutionPermission(実行許可信号)のチャタリングによる信号誤検知を防止したい場合は、フィルタ時間(ExecutionPermission.Signal.FilterTime)を設定してください。

信号名(ExecutionPermission.Signal.Source.Target)に使用できない信号を使用した場合、外部信号選択文字列不正(エラーコード: 1B05H)、外部信号選択不正(エラーコード: 1B31Hまたは1B32H)、アドオンSignalIO内部異常(エラーコード: 1B33H)のいずれかを出力します。

注意事項

- ExecutionPermission(実行許可信号)の入力は、AnalysisComplete(解析完了)がTRUEとなった後に行ってください。多重起動した動作系FBが存在しない状態、または解析が完了していない状態(AnalysisComplete(解析完了)がFALSEの状態)で、ExecutionPermission(実行許可信号)の信号入力を検出しても、Done(実行完了)はTRUEなりません。
- MCv_MoveWait(軸制御開始ウェイト)のActive(実行中)がTRUEとなってから動作系FBを多重起動してください。MCv_MoveWait(軸制御開始ウェイト)の実行時にエラーが発生した場合、ExecutionPermission(実行許可信号)の入力を待たずに多重起動した動作系FBが実行される可能性があります。
- 演算周期混在機能使用時、MCv_MoveWait(軸制御開始ウェイト)の対象の軸の演算周期とExecutionPermission(実行許可信号)のリフレッシュ周期を同一にしてください。リフレッシュ周期が異なる場合、位置決めの開始タイミングがばらつく可能性があります。
- 異なる演算周期の軸を同時に始動する場合、軸の演算周期は同一に設定してください。軸の始動タイミングが1演算周期分ずれる可能性があります。異なる演算周期の軸を同時に始動させたい場合は、信号名(ExecutionPermission.Signal.Source.Target)と信号論理(ExecutionPermission.Signal.Detection)を同一に設定した複数のExecutionPermission(実行許可信号)を使用して、補正時間(ExecutionPermission.Signal.CompensationTime)を調整してください。
- MCv_MoveWait(軸制御開始ウェイト)実行時は、軸状態(AxisName.Md.AxisStatus)が「4: 待機中(Standstill)」であることを確認してください。始動できない軸状態(AxisName.Md.AxisStatus)となっている場合、始動不可(エラーコード: 1AFCH)が発生して、運転中の制御が停止します。
- 他の動作系FB実行中にMCv_MoveWait(軸制御開始ウェイト)を始動した場合、始動要求は無視され、運転中始動警告(イベントコード: 00D01H)が発生します。
- 多重起動する動作系FBのバッファモード(BufferMode)に、Bufferedを設定してください。Abortingを設定して多重起動した場合、MCv_MoveWait(軸制御開始ウェイト)を中断し、ExecutionPermission(実行許可信号)の入力状態は無視され、多重起動した動作系FBを即時実行します。Blendingを設定した多重起動した場合、BufferMode指定不可(エラーコード: 1AE5H)が発生します。

プログラム例

制御開始(bExecute)をTRUEにすることでMC_MoveRelativeとMCv_MoveLinearInterpolateRelativeを解析し待機させ、実行許可信号(bExecutionPermission)をTRUEにすることで待機中のFBを同時に始動するプログラム例を下記に示します。

■軸

軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REAL	軸1
2	Axis0002	AXIS_REAL	軸2
3	Axis0003	AXIS_REAL	軸3

■軸グループ

軸グループNo.	ラベル名	データ型	コメント
1	AxisGroup0001	AXIS_GROUP	軸グループ1

項目	設定値
構成軸[1]	Axis0001
構成軸[2]	Axis0002

■入出力データ

ラベル名	データ種別	データ型	ソース種別	ソース	信号検出方法	補正時間	フィルタ時間
SignalData0001	入力信号	MC_INPUT_REF	ラベル	bExecutionPermission	HighLevel	0.0	0.0

■使用するラベル

ラベル名	データ型	コメント
bExecute	ビット	制御開始
bExecutionPermission	ビット	実行許可信号
MCv_MoveWait_1	MCv_MoveWait	軸制御開始ウェイトFB
MC_MoveRelative_1	MC_MoveRelative	相対値位置決めFB
MCv_GroupMoveWait_1	MCv_GroupMoveWait	軸グループ制御開始ウェイトFB
MCv_MoveLinearInterpolateRelative_1	MCv_MoveLinearInterpolateRelative	相対値直線補間制御FB
wMoveLinearInterpolateRelativeLinearAxes	ワード[符号付き](0..15)	直線補間軸
leMoveLinearInterpolateRelativePosition	倍精度実数(0..15)	目標位置

■STプログラム

```
SET( MCv_MoveWait_1.Active , bMoveRelativeExecute );
SET( MCv_GroupMoveWait_1.Active , bMoveLinearInterpolateRelativeExecute );
RST( NOT bExecute , bMoveRelativeExecute );
RST( NOT bExecute , bMoveLinearInterpolateRelativeExecute );
//-----軸制御開始ウェイト-----
MCv_MoveWait_1(
    Axis:= Axis0001.AxisRef ,
    Execute:= bExecute ,
    ExecutionPermission:= SignalData0001 ,
    Options:= H00000000
);
//-----相対値位置決め-----
MC_MoveRelative_1(
    Axis:= Axis0001.AxisRef ,
    Execute:= bMoveRelativeExecute ,
    ContinuousUpdate:= FALSE ,
    Distance:= 10000.0 ,
    Velocity:= 1000.0 ,
    Acceleration:= 1000.0 ,
    Deceleration:= 1000.0 ,
    Jerk:= 0.0 ,
    BufferMode:= MC_BUFFER_MODE__mcBuffered ,
    Options:= H00000000
);
//-----軸グループ制御開始ウェイト-----
MCv_GroupMoveWait_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bExecute ,
    ExecutionPermission:= SignalData0001 ,
    Options:= H00000000
);
//-----相対値直線補間制御用データ設定-----
wMoveLinearInterpolateRelativeLinearAxes[0] := 1;
wMoveLinearInterpolateRelativeLinearAxes[1] := 2;
leMoveLinearInterpolateRelativeDistance[0] := 10000.0;
leMoveLinearInterpolateRelativeDistance[1] := 10000.0;
//-----相対値直線補間制御-----
MCv_MoveLinearInterpolateRelative_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bMoveLinearInterpolateRelativeExecute ,
    ContinuousUpdate:= FALSE ,
    LinearAxes:= wMoveLinearInterpolateRelativeLinearAxes ,
    Distance:= leMoveLinearInterpolateRelativeDistance ,
    Velocity:= 1000.0 ,
    Acceleration:= 1000.0 ,
    Deceleration:= 1000.0 ,
    Jerk:= 0.0 ,
    VelocityMode:= MC_INTERPOLATE_SPEED_MODE__VectorSpeed ,
    BufferMode:= MC_BUFFER_MODE__mcBuffered ,
    Options:= H00000000
);
```

42.22 軸グループ制御開始ウェイト

42

MCv_GroupMoveWait

指定した軸グループの多重起動した動作系FBの動作を、実行許可信号の入力により即時実行します。

ラダー	FBD/LD	ST
		<pre>MCv_GroupMoveWait(AxesGroup:= ?AXES_GROUP_REF?, Execute:= ?BOOL?, ExecutionPermission:= ?MC_INPUT_REF?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?, AnalysisComplete=> ?BOOL?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
AxesGroup	軸グループ情報	AXES_GROUP_REF	起動時	—	省略不可	軸グループを設定します。 使用する変数 (AxesGroupName,AxesGroupRef.)については、下記を参照してください。 1402ページ AxesGroupName,AxesGroupRef.(軸グループ情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMCv_GroupMoveWait(軸グループ制御開始ウェイト)を実行します。
ExecutionPermission	実行許可信号	MC_INPUT_REF	起動時	—	省略不可	軸グループ制御開始ウェイトの解除に使用する信号を設定します。
Options	オプション	DWORD(HEX)	起動時	00000000H	00000000H	「00000000H」を設定してください。 「00000000H」以外を設定した場合、とエラーになります。

■出力変数

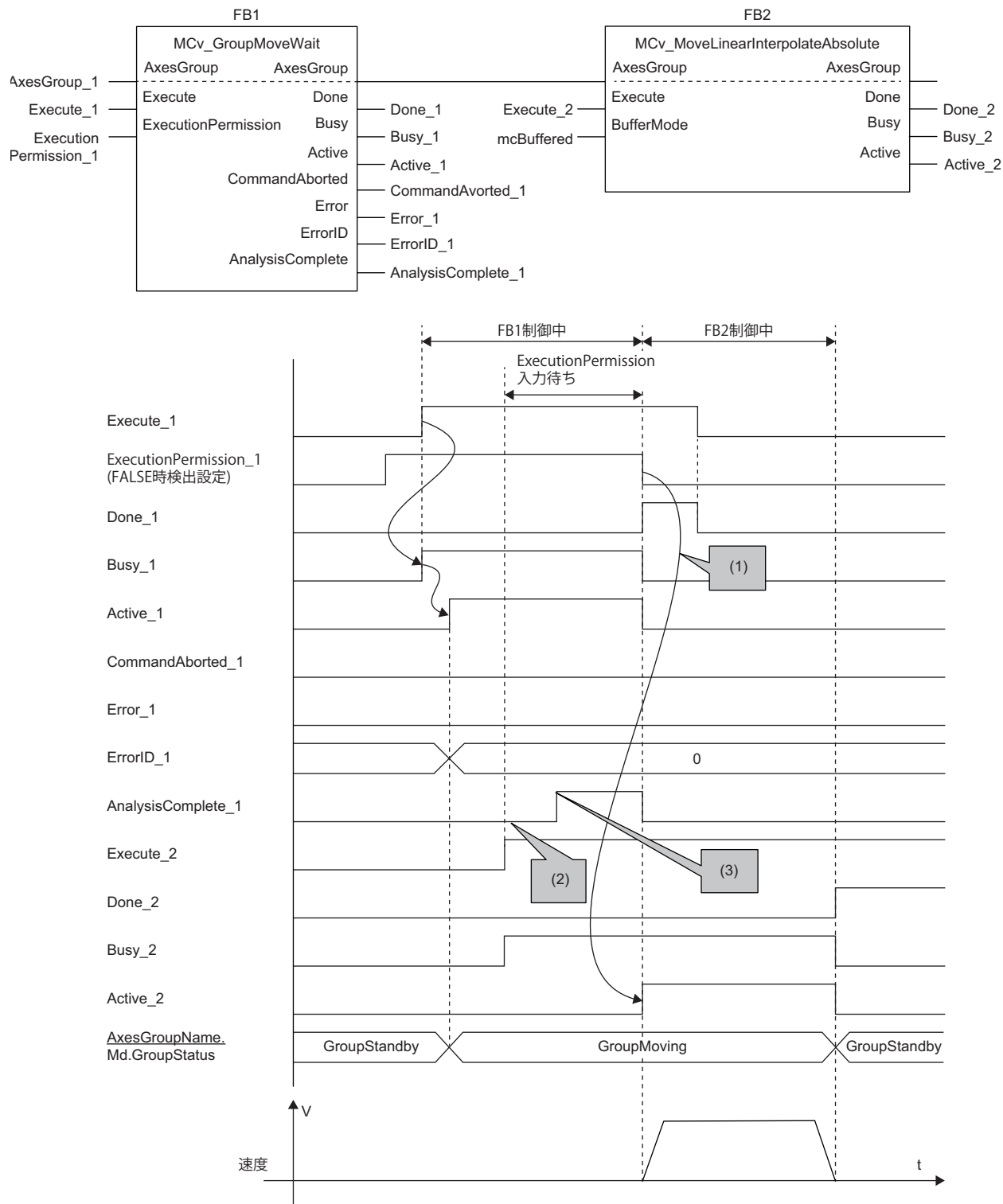
出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	ExecutionPermission(実行許可信号)で設定した信号の入力検出、かつAnalysisComplete(解析完了)がTRUEになったときに、TRUEとなり、MCv_GroupMoveWait(軸グループ制御開始ウェイト)の制御が完了したことを示します。
Busy	実行中	BOOL	FALSE	MCv_GroupMoveWait(軸グループ制御開始ウェイト)が実行中のときに、TRUEになります。
Active	制御中	BOOL	FALSE	MCv_GroupMoveWait(軸グループ制御開始ウェイト)が軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	他のFBにより実行中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 使用するコントローラのユーザーズマニュアル
AnalysisComplete	解析完了	BOOL	FALSE	多重起動した動作系FBの解析が完了ときに、TRUEになります。

機能

- 多重起動した多軸制御の動作系FBの軸グループ動作をExecutionPermission(実行許可信号)の入力タイミングで即時実行します。
- MCv_GroupMoveWait(軸グループ制御開始ウェイト)起動中に他の動作系FBの多重起動すると、多重起動したFBは解析完了まで実行され、ExecutionPermission(実行許可信号)が入力されるまで待機します。多重起動したFBの解析完了のタイミングはAnalysisComplete(解析完了)で確認できます。
- AnalysisComplete(解析完了)がTRUEになった後にExecutionPermission(実行許可信号)を実行してください。
- ExecutionPermission(実行許可信号)が入力されるまでの間、軸グループ状態(AxisGroupName.Md.GroupStatus)は「5: 動作中(GroupMoving)」になります。
- ExecutionPermission(実行許可信号)が入力されると、信号入力を検出した1演算周期後に多重起動したFBを実行します。
- AnalysisComplete(解析完了)がTRUEになった後に、ExecutionPermission(実行許可信号)を入力してください。

■ タイミングチャート

・ 正常完了の場合



(1): ExecutionPermission入力を検出した1演算周期後にFB2が起動します。

(2): FB2の解析中

(3): FB2の解析完了

・ 異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■ExecutionPermission(実行許可信号)

MCv_GroupMoveWait(軸グループ制御開始ウェイト)の解除に使用する信号を指定します。
ExecutionPermission(実行許可信号)は、MC_INPUT_REF構造体で設定します。MC_INPUT_REF構造体については、下記を参照してください。

1420ページ MC_INPUT_REF

ExecutionPermission(実行許可信号)で設定する入力信号(ExecutionPermission.Signal)の設定範囲を下記に示します。
入力信号(ExecutionPermission.Signal)の型は、SIGNAL_SELECT構造体となります。

構造体	変数名	型	設定範囲
SIGNAL_SELECT(信号選択)	Source.Target(信号)*1	TARGET_REF 1421ページ TARGET_REF(入力信号)	■型 ・ BOOL ■データ種別 ・ [OBJ] ・ [VAR] ・ [DEV] ・ [CONST]
	Detection(信号検出方法)	INT (MC_SIGNAL_LOGIC) 1452ページ MC_SIGNAL_LOGIC	・ 0: TRUE時検出(HighLevel) ・ 1: FALSE時検出(LowLevel)
	CompensationTime(補正時間)*2	LREAL	-5.0~0.0[s]
	FilterTime(フィルタ時間)	LREAL	0.0~5.0[s]

*1 外部信号高精度入力の使用はできません。外部信号高精度入力用の信号を指定した場合、通常の信号と同じ精度で動作します。
*2 補正時間(ExecutionPermission.Signal.CompensationTime)に1演算周期より小さい値を指定した場合、0.0(補正なし)として動作します。

[OBJ]指定時に対象修飾を省略した場合は外部信号選択不正(エラーコード: 1B31H)を出力します。
[OBJ]指定時に参照する局が軸エミュレート機能が有効かつ局アドレスが設定済みの場合、エミュレートしているオブジェクトを参照します。
ExecutionPermission(実行許可信号)の進めを補正する場合、多重起動した動作系FBの実行タイミングを調整したい場合は、補正時間(ExecutionPermission.Signal.CompensationTime)を設定してください。
ExecutionPermission(実行許可信号)のチャタリングによる信号誤検知を防止したい場合は、フィルタ時間(ExecutionPermission.Signal.FilterTime)を設定してください。
信号名(ExecutionPermission.Signal.Source.Target)に使用できない信号を使用した場合、外部信号選択文字列不正(エラーコード:1B05H)、外部信号選択不正(エラーコード: 1B31Hまたは1B32H)、アドオンSignalIO内部異常(エラーコード: 1B33H)のいずれかを出力します。

- ExecutionPermission(実行許可信号)の入力は、AnalysisComplete(解析完了)がTRUEとなった後に行ってください。多重起動した動作系FBが存在しない状態、または解析が完了していない状態(AnalysisComplete(解析完了)がFALSEの状態)で、ExecutionPermission(実行許可信号)の信号入力を検出しても、Done(実行完了)はTRUEなりません。
- MCv_GroupMoveWait(軸グループ制御開始ウェイト)のActive(実行中)がTRUEとなってから動作系FBを多重起動してください。MCv_GroupMoveWait(軸グループ制御開始ウェイト)の実行時にエラーが発生した場合、ExecutionPermission(実行許可信号)の入力を待たずに多重起動した動作系FBが実行される可能性があります。
- 演算周期混在機能使用時、MCv_GroupMoveWait(軸グループ制御開始ウェイト)の対象の軸の演算周期とExecutionPermission(実行許可信号)のリフレッシュ周期を同一にしてください。リフレッシュ周期が異なる場合、位置決め開始タイミングがばらつく可能性があります。
- 異なる演算周期の軸を同時に始動する場合、軸の演算周期は同一に設定してください。軸の始動タイミングが1演算周期分ずれる可能性があります。異なる演算周期の軸を同時に始動させたい場合は、信号名(ExecutionPermission.Signal.Source.Target)と信号論理(ExecutionPermission.Signal.Detection)を同一に設定した複数のExecutionPermission(実行許可信号)を使用して、補正時間(ExecutionPermission.Signal.CompensationTime)を調整してください。
- MCv_GroupMoveWait(軸グループ制御開始ウェイト)実行時は、軸グループ状態(AxisGroupName.Md.GroupStatus)が「4: 待機中(GroupStandby)」であることを確認してください。始動できない軸グループ状態(AxisGroupName.Md.GroupStatus)となっている場合、始動不可(エラーコード: 1AFCH)が発生して、運転中の制御が停止します。
- 他の動作系FB実行中にMCv_GroupMoveWait(軸グループ制御開始ウェイト)を始動した場合、始動要求は無視され、運転中始動警告(イベントコード: 00D01H)が発生します。
- 多重起動する動作系FBのバッファモード(BufferMode)に、Bufferedを設定してください。Abortingを設定して多重起動した場合、MCv_GroupMoveWait(軸グループ制御開始ウェイト)を中断し、ExecutionPermission(実行許可信号)の入力状態は無視され、多重起動した動作系FBを即時実行します。Blendingを設定した多重起動した場合、BufferMode指定不可(エラーコード: 1AE5H)が発生します。

プログラム例

制御開始(bExecute)をTRUEにすることでMC_MoveRelativeとMCv_MoveLinearInterpolateRelativeを解析し待機させ、実行許可信号(bExecutionPermission)をTRUEにすることで待機中のFBを同時に始動するプログラム例を下記に示します。

■軸

軸No.	ラベル名	データ型	コメント
1	Axis0001	AXIS_REAL	軸1
2	Axis0002	AXIS_REAL	軸2
3	Axis0003	AXIS_REAL	軸3

■軸グループ

軸グループNo.	ラベル名	データ型	コメント
1	AxisGroup0001	AXIS_GROUP	軸グループ1

項目	設定値
構成軸[1]	Axis0001
構成軸[2]	Axis0002

■入出力データ

ラベル名	データ種別	データ型	ソース種別	ソース	信号検出方法	補正時間	フィルタ時間
SignalData0001	入力信号	MC_INPUT_REF	ラベル	bExecutionPermission	HighLevel	0.0	0.0

■使用するラベル

ラベル名	データ型	コメント
bExecute	ビット	制御開始
bExecutionPermission	ビット	実行許可信号
MCv_MoveWait_1	MCv_MoveWait	軸制御開始ウェイトFB
MC_MoveRelative_1	MC_MoveRelative	相対値位置決めFB
MCv_GroupMoveWait_1	MCv_GroupMoveWait	軸グループ制御開始ウェイトFB
MCv_MoveLinearInterpolateRelative_1	MCv_MoveLinearInterpolateRelative	相対値直線補間制御FB
wMoveLinearInterpolateRelativeLinearAxes	ワード[符号付き](0..15)	直線補間軸
leMoveLinearInterpolateRelativePosition	倍精度実数(0..15)	目標位置

■STプログラム

```

SET( MCv_MoveWait_1.Active , bMoveRelativeExecute );
SET( MCv_GroupMoveWait_1.Active , bMoveLinearInterpolateRelativeExecute );
RST( NOT bExecute , bMoveRelativeExecute );
RST( NOT bExecute , bMoveLinearInterpolateRelativeExecute );
//-----軸制御開始ウェイト-----
MCv_MoveWait_1(
    Axis:= Axis0001.AxisRef ,
    Execute:= bExecute ,
    ExecutionPermission:= SignalData0001 ,
    Options:= H00000000
);
//-----相対値位置決め-----
MC_MoveRelative_1(
    Axis:= Axis0001.AxisRef ,
    Execute:= bMoveRelativeExecute ,
    ContinuousUpdate:= FALSE ,
    Distance:= 10000.0 ,
    Velocity:= 1000.0 ,
    Acceleration:= 1000.0 ,
    Deceleration:= 1000.0 ,
    Jerk:= 0.0 ,
    BufferMode:= MC_BUFFER_MODE__mcBuffered ,
    Options:= H00000000
);
//-----軸グループ制御開始ウェイト-----
MCv_GroupMoveWait_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bExecute ,
    ExecutionPermission:= SignalData0001 ,
    Options:= H00000000
);
//-----相対値直線補間制御用データ設定-----
wMoveLinearInterpolateRelativeLinearAxes[0] := 1;
wMoveLinearInterpolateRelativeLinearAxes[1] := 2;
leMoveLinearInterpolateRelativeDistance[0] := 10000.0;
leMoveLinearInterpolateRelativeDistance[1] := 10000.0;
//-----相対値直線補間制御-----
MCv_MoveLinearInterpolateRelative_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bMoveLinearInterpolateRelativeExecute ,
    ContinuousUpdate:= FALSE ,
    LinearAxes:= wMoveLinearInterpolateRelativeLinearAxes ,
    Distance:= leMoveLinearInterpolateRelativeDistance ,
    Velocity:= 1000.0 ,
    Acceleration:= 1000.0 ,
    Deceleration:= 1000.0 ,
    Jerk:= 0.0 ,
    VelocityMode:= MC_INTERPOLATE_SPEED_MODE__VectorSpeed ,
    BufferMode:= MC_BUFFER_MODE__mcBuffered ,
    Options:= H00000000
);

```

42.23 アドバンスト同期制御

MCv_AdvancedSync

指定したアドバンスト同期制御設定に従って同期制御を開始します。

ラダー	FBD/LD	ST
		<pre> MCv_AdvancedSync(Master:= ?AXIS_REF?, Slave:= ?AXIS_REF?, Enable:= ?BOOL?, Acceleration:= ?LREAL?, Deceleration:= ?LREAL?, Jerk:= ?LREAL?, Options:= ?DWORD?, Done=> ?BOOL?, InSync=> ?BOOL?, CycleZeroPoint=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?); </pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Master	主軸	AXIS_REF	起動時	—	省略不可	入力軸を設定します。 省略した場合は、主軸番号範囲外(エラーコード: 1B0BH)となります。 対応する入力設定がない場合、始動時にアドバンスト同期制御軸未設定(エラーコード: 1B22H)となります。
Slave	従軸	AXIS_REF	起動時	—	省略不可	出力軸を設定します。 省略した場合は、従軸番号範囲外(エラーコード: 1B0CH)となります。 対応する出力設定がない場合、始動時にアドバンスト同期制御軸未設定(エラーコード: 1B22H)となります。

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Enable	有効	BOOL	常時	TRUE, FALSE	FALSE	TRUEにすると、本FBを実行します。
Acceleration	加速度	LREAL	起動時	0.0	0.0	「0.0」を設定してください。 「0.0」以外を設定した場合、加速度範囲外(エラーコード: 1A8DH)となります。
Deceleration	減速度	LREAL	起動時	0.0000, 0.0001~2147483647.0	0.0	減速度を設定します。 詳細は、下記を参照してください。 ☞ 1808ページ 減速度(Deceleration)
Jerk	ジャーク	LREAL	起動時	0.0	0.0	「0.0」を設定してください。 「0.0」以外を設定した場合、ジャーク範囲外(エラーコード: 1A8FH)となります。
Options	オプション	DWORD(HEX)	起動時	下記を参照してください。 ☞ 1808ページ オプション(Options)	0000000H	本FBの機能オプションをビット指定で設定します。 詳細は、下記を参照してください。 ☞ 1808ページ オプション(Options)

■出力変数

出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	下記のときにTRUEになります。 ・本FBの次に実行するモーション制御FBで設定する多重起動(バッファモード)の「1: Buffered」による次のモーション制御FBに移行した後。 ・同期制御解析モードにて解析が完了したとき。
InSync	同期中	BOOL	FALSE	出力軸が同期を開始したときに、TRUEになります。
CycleZeroPoint	カムサイクル完了	BOOL	FALSE	制御中(Active)がTRUEとなった後、直線カム、またはカム開始点が「0」のカムを使用している場合、1サイクル現在位置が山越えするたびにTRUEになります。 カム開始点が「0」以外のカムを使用している場合、1サイクル現在位置がカムデータ0点目の位置を通過時にTRUEになります。
Busy	実行中	BOOL	FALSE	本FBを実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	本FBが軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	本FBの実行が中断したときに、TRUEになります。 有効(Enable)がFALSEになると、FALSEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。  使用するコントローラのユーザーズマニュアル

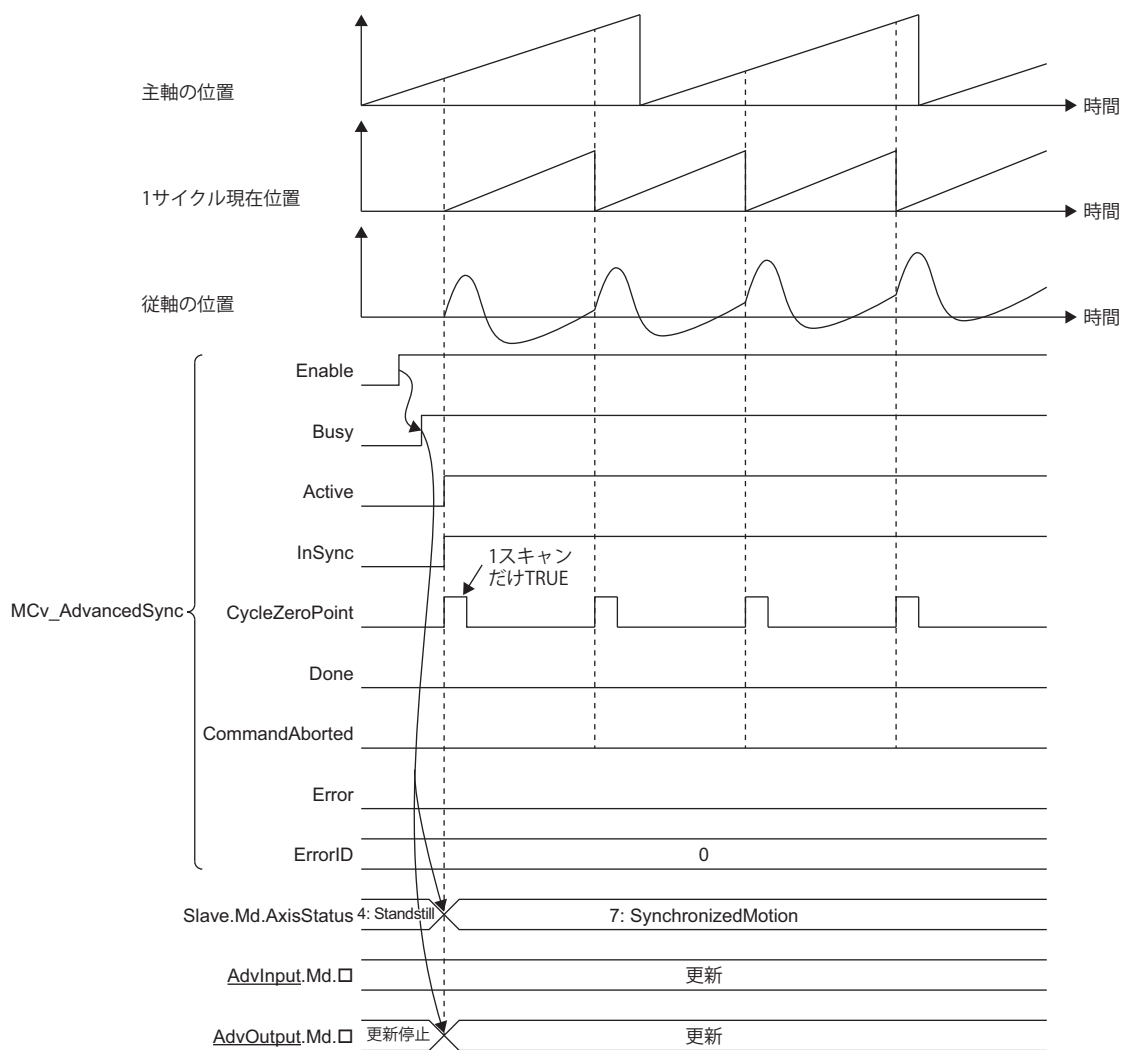
機能

- ・本FBを実行すると指定したアドバンス同期制御設定に従ってアドバンス同期制御を開始します。
- ・従軸の軸状態(AxisName.Md.AxisStatus)が「4: 待機中(Standstill)」の場合のみ同期制御を開始できます。開始するとに従軸の軸状態(AxisName.Md.AxisStatus)が「7: 同期運転中(SynchronizedMotion)」となります。同期制御開始時、「4: 待機中(Standstill)」以外の場合、従軸の軸状態(AxisName.Md.AxisStatus)は「1: エラー停止中(ErrorStop)」となり、始動不可(エラーコード: 1AFCH)となります。
- ・本FBの有効(Enable)をFALSEにした場合、または出力軸の軸情報に設定した軸に対して停止要因が発生した場合に、停止要因とその停止処理に従い従軸(Slave)は即停止/減速停止となります。
- ・アドバンス同期制御についての詳細は、下記を参照してください。

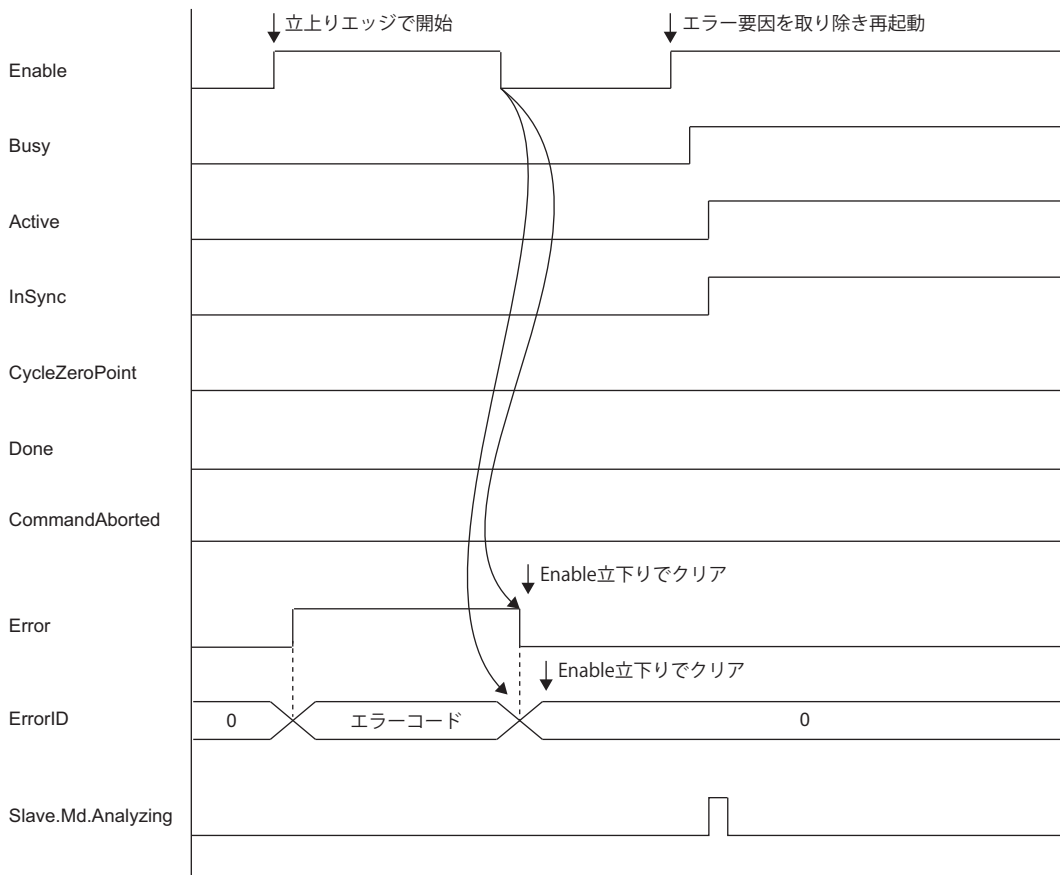
使用するコントローラのユーザーズマニュアル

■タイミングチャート

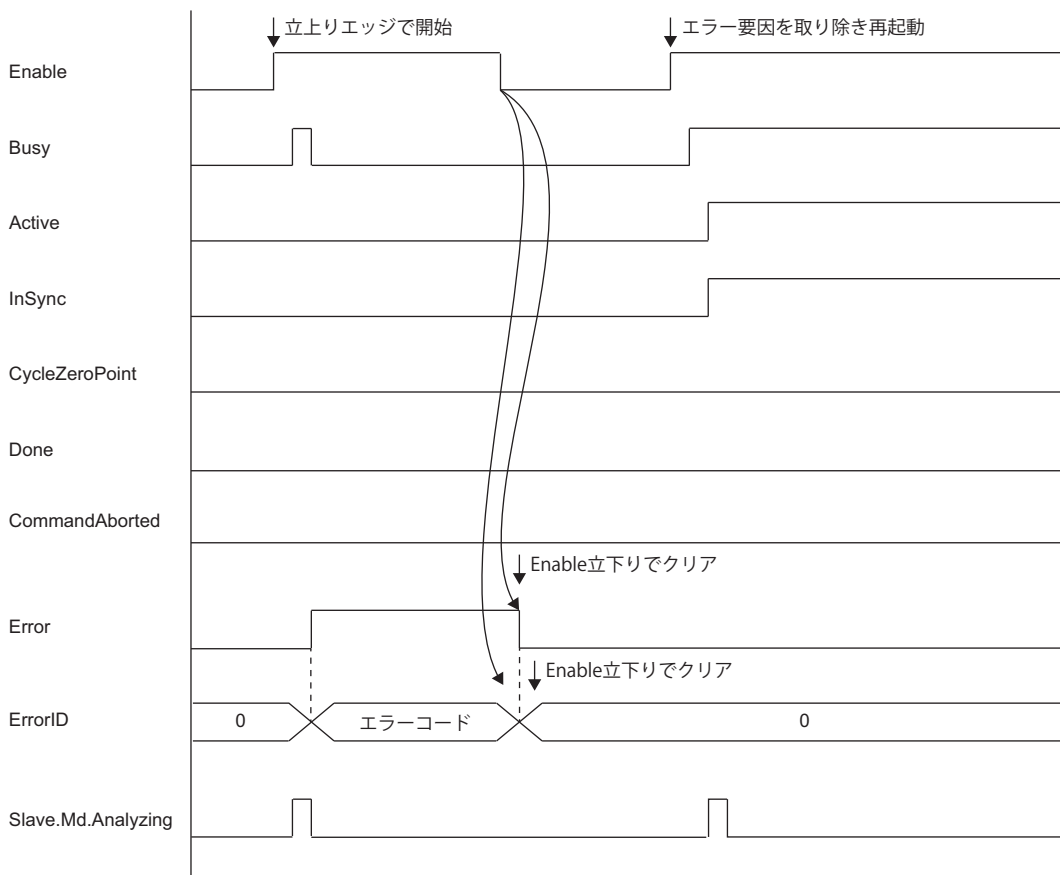
・正常完了の場合



・ 入出力変数異常の場合



・ 入力変数異常の場合



■減速度(Deceleration)

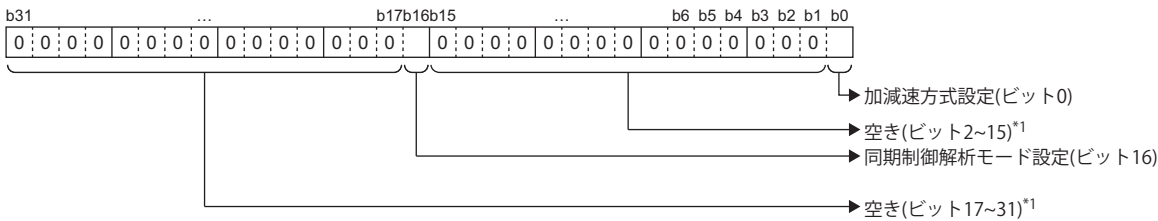
減速度を設定します。
加減速方式設定(オプション(Options): ビット0~1)の「0: 加減速度指定方式(mcAccDec)」を設定した場合、設定範囲を設定します。

加減速方式設定(オプション(Options): ビット0~1)	設定範囲
0: 加減速度指定方式(mcAccDec)	0.0000 ^{*1} , 0.0001~2147483647.0[U/s ²]までの正数

^{*1} 始動時は始動時加減速度0指定時動作選択(AxisName.Pr.AccelerationZeroBehavior)によって、動作が変わります。加速度変更, 減速度変更時は, 変更を受け付けません。

■オプション(Options)

本FBで使用する機能オプションをビット指定で設定します。
ビット指定で設定する内容を下記に示します。



^{*1} 空きには「0」を設定してください。

ビット	名称	内容
0	加減速方式設定	制御を行うための加減速方式を設定します。 • 0: 加減速度指定方式(mcAccDec)
16	同期制御解析モード設定	同期制御解析モードを設定します。 • 0: 無効 • 1: 有効

・ 加減速方式設定(ビット0~1)

設定値	内容
0: 加減速度指定方式(mcAccDec)	本FBで設定した減速度(Deceleration)を用いて, 加速/減速する方式です。

注意事項

- 本FBを実行中に主軸(Master), 補助軸(AdvOutputName.Pr.AuxAxis), サブ入力軸(AdvOutputName.Pr.SubAxis)の入力軸情報に設定した軸に停止要因が発生しても, 従軸(Slave)は入力軸から伝達された指令で制御を継続します。従軸(Slave)の軸状態は変化しません。また, 出力軸の軸情報(AdvOutputName.Axis)に設定した軸に停止要因が発生しても, 入力軸の軸情報(AdvInputName.Axis)に設定した軸には影響しません。
- 本FBの起動時に下記の場合, 主軸従軸循環参照(エラーコード: 1AADH)になります。

エラーの発生条件

軸を複数連結するとき, 主軸(Master), 補助軸(AdvOutputName.Pr.AuxAxis), サブ入力軸(AdvOutputName.Pr.SubAxis)を後段の単軸同期制御FBの従軸(Slave)に設定した場合

単軸同期制御FBを257個以上連結した場合

- 主軸(Master), 補助軸(AdvOutputName.Pr.AuxAxis), サブ入力軸(AdvOutputName.Pr.SubAxis)がサーボアラームや緊急停止によってサーボOFFとなった場合, 従軸(Slave)が予期せぬ動作となる恐れがあります。そのため, MC_Stop(強制停止)を使用して本FBの動作を停止することを推奨します。
- 本FB始動の解析中(従軸(Slave)の軸状態(AxisName.Md.AxisStatus)が「7: 同期運転中(SynchronizedMotion)」になるまで)に入力軸が動作した場合, 解析中の入力軸の移動量や現在値変更等による変化量が同期制御開始直後に反映されます。入力軸の移動量や変化量によっては, 出力軸が急加速する場合がありますので, 同期制御中になったことを確認後, 入力軸の動作を開始してください。
- 同期制御中に設定値の取込みを実施するADV_INPUTおよびADV_OUTPUTのメンバ変数の設定値を本FB実行中に範囲外へ変更した場合, パラメータ範囲外(アドバンス同期制御入力軸)またはパラメータ範囲外(アドバンス同期制御出力軸)となり, 変更前の設定値を維持して制御を続けます。
- 本FB起動時にADV_INPUTおよびADV_OUTPUTのメンバ変数の設定値が範囲外の場合, パラメータ範囲外(アドバンス同期制御入力軸)またはパラメータ範囲外(アドバンス同期制御出力軸)となり, 始動しません。
- 本FBの起動時に下記の場合, 演算周期不一致(エラーコード: 1B29H)になります。MCv_SyncOperationCycles(制御演算周期同期)を使用した場合も下記条件に当てはまる場合は演算周期不一致(エラーコード: 1B29H)になります。

エラーの発生条件

主軸(Master), 補助軸(AdvOutputName.Pr.AuxAxis), サブ入力軸(AdvOutputName.Pr.SubAxis)と従軸(Slave)の制御演算周期が異なる場合

出力軸の制御演算周期とクラッチ信号のリフレッシュ周期が異なる場合

プログラム例

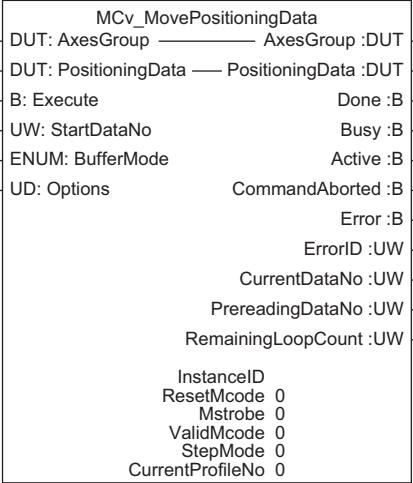
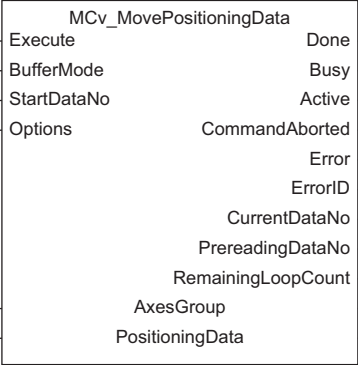
本FBのプログラム例については, 下記マニュアルを参照してください。

📖使用するコントローラのユーザーズマニュアル

42.24 多軸位置決めデータ運転

MCv_MovePositioningData

指定した位置決めデータに従い、軸グループを動作します。

ラダー	FBD/LD	ST
		<pre>MCv_MovePositioningData(AxesGroup:= ?AXES_GROUP_REF?, PositioningData:= ?MC_POSITIONING_DATA_ID?, Execute:= ?BOOL?, StartDataNo:= ?WORD?, BufferMode:= ?INT?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?, CurrentDataNo=> ?WORD?, PrereadingDataNo=> ?WORD?, RemainingLoopCount=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
AxesGroup	軸グループ情報	AXES_GROUP_REF	起動時	—	省略不可	軸グループを設定します。
PositioningData	位置決めデータ	MC_POSITIONING_DATA_ID	起動時	—	省略不可	位置決めデータの演算プロファイルIDを設定します。 演算プロファイルIDは、MC_POSITIONING_DATA_ID構造体で設定します。MC_POSITIONING_DATA_ID構造体については、下記を参照してください。 ☞ 1419ページ MC_POSITIONING_DATA_ID

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEで本FBを実行します。
StartDataNo	開始位置決めデータNo.	WORD(UINT)	起動時	0~5000	0	開始する位置決めデータNo.を設定します。「0」を設定して本FBを実行した場合、位置決めデータNo.1から開始します。
BufferMode	バッファモード	INT (MC_BUFFER_MODE)	起動時	0, 1	0	バッファモードを設定します。 ・ 0: Aborting(mcAborting) ・ 1: Buffered(mcBuffered) 詳細は、下記を参照してください。 ☞ 1813ページ バッファモード (BufferMode)
Options	オプション	DWORD(HEX)	起動時	下記を参照してください。 ☞ 1813ページ オプション (Options)	0000000H	本FBの機能オプションをビット指定で設定します。 詳細は、下記を参照してください。 ☞ 1813ページ オプション(Options)

■出力変数

出力変数	名称	データ型	初期値	内容
Done	実行完了	BOOL	FALSE	目標位置に到達したときに、TRUEになります。
Busy	実行中	BOOL	FALSE	本FBを実行したときに、TRUEになります。
Active	制御中	BOOL	FALSE	本FBが軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	本FBの実行が中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。  使用するコントローラのユーザーズマニュアル
CurrentDataNo	実行中位置決めデータNo.	WORD(UINT)	0	実行中の位置決めデータNo.を示します。 位置決めデータNo.切り換え時に更新します。
PrereadingDataNo	先読み位置決めデータNo.	WORD(UINT)	0	先読みした位置決めデータNo.の終端を格納します。 実行中位置決めデータNo.(CurrentDataNo)から先読み位置決めデータNo.(PrereadingDataNo)までの位置決めデータNo.を先読みします。
RemainingLoopCount	残ループ回数	WORD(UINT)	0	制御方式がLOOPの位置決めデータを実行時に、残りの繰り返し回数を格納します。 繰り返しループの先頭で、格納値を「1」減らします。 格納値が「0」の場合、制御方式がLENDの位置決めデータでループしません。

■公開変数

公開変数	名称	データ型	初期値	内容
InstanceID	インスタンスID	INSTANCE_ID	0	インスタンスIDです。 インスタンス生成時にシステムにより自動設定されます。
ResetMcode	Mコードリセット	BOOL	FALSE	TRUEでMコードのリセットを行います。 リセットによりMストロブ(Mstrobe)がFALSEとなります。
Mstrobe	Mストロブ	BOOL	FALSE	Mコード出力が有効中のときに、TRUEになります。 Mコード出力タイミング(オプション(Options): ビット16)でTRUEとなります。 Mコードリセット(ResetMcode)のTRUEによりFALSEになります。
ValidMcode	有効Mコード	WORD(UINT)	0	有効な(最新で出力した)Mコードを格納されます。 位置決めデータのMコード出力が有効な場合、Mコード出力タイミング(オプション(Options): ビット16)で位置決めデータNo.のMコードに更新します。 *: Mコードリセット(ResetMcode)のTRUEではリセットしません。
StepMode	ステップモード	INT (MC_STEP_MODE)	0	ステップモードを設定します。 0: 無視(Ignored) 1: 減速単位ステップ(StepPerDec) 2: 位置決めデータ単位ステップ(StepPerPositioningData) 詳細は、下記を参照してください。  1813ページ ステップモード(StepMode)
CurrentProfileNo	実行中プロファイルNo.	WORD(UINT)	0	実行中のプロファイルID番号が格納されます。 詳細は、下記を参照してください。  1814ページ 実行中プロファイルNo.(CurrentProfileNo)

- ・公開変数の取込/更新を行う周期を下記に示します。

公開変数	取込/更新周期
InstanceID	FBを呼び出すPOU(プログラム部品)の実行周期
ResetMcode	軸グループの構成軸の制御周期 ^{*1}
Mstrobe	
ValidMcode	
StepMode	FBを呼び出すPOU(プログラム部品)の実行周期
CurrentProfileNo	

*1 出力変数と更新タイミングが合わない場合があります。更新タイミングを合わせる場合は、同じ周期のプログラムからFBを呼び出して下さい。

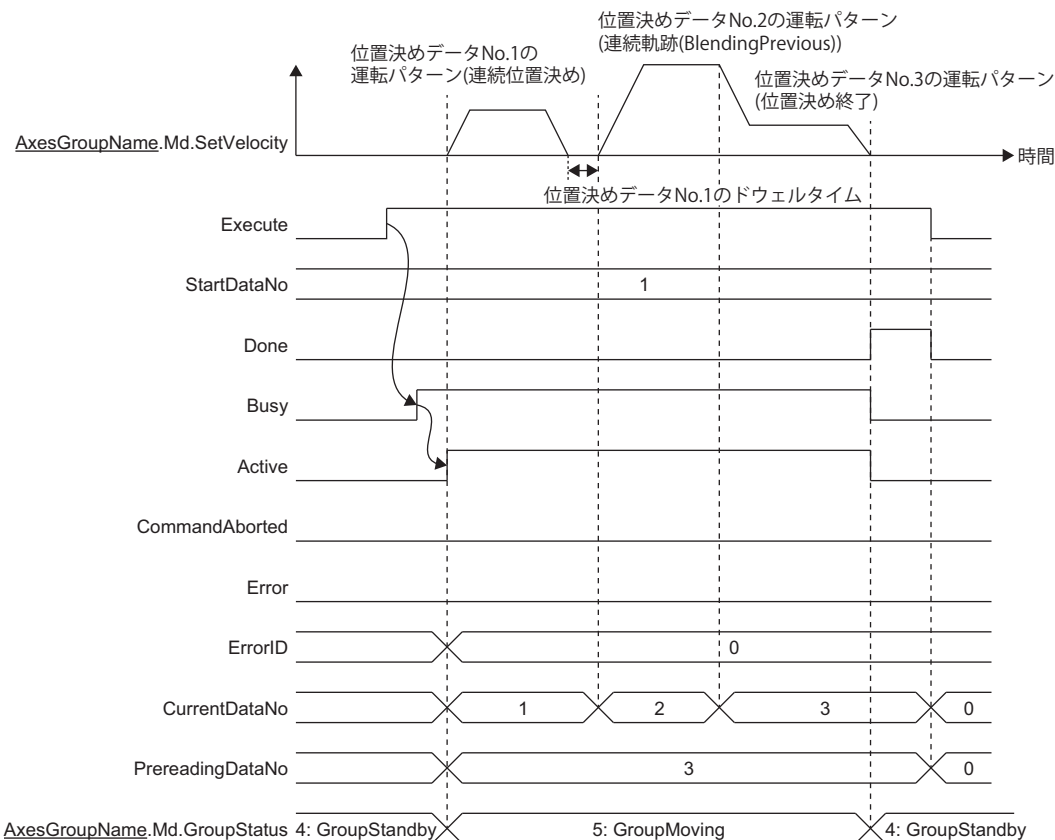
機能

- ・軸グループに対して、演算プロファイル(多軸位置決めデータ形式)に指定した連続的な位置決め動作を行います。
- ・多軸位置決めデータ運転は、開始位置決めデータNo.(StartDataNo)から位置決めデータを昇順に読み、多軸位置決め動作を内部的に連続して実行します。
- ・多軸位置決めデータ運転についての詳細は、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

■タイミングチャート

- ・正常完了の場合



- ・異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

本FBで設定可能なバッファモードを下記に示します。

設定値	内容
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。
1: Buffered(mcBuffered)	制御中のFBに次のFBをバッファリングします。 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFB完了にて、バッファリングFBを順次実行します。

Point

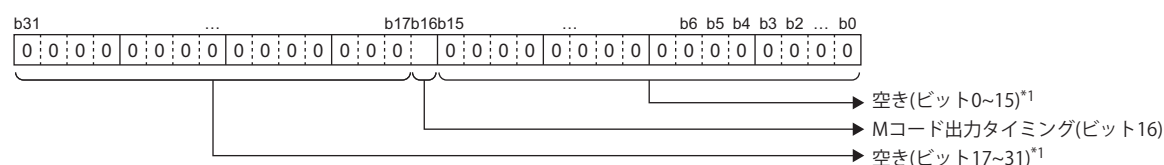
多重起動(バッファモード)の詳細については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

■オプション(Options)

本FBで使用する機能オプションをビット指定で設定します。

ビット指定で設定する内容を下記に示します。



*1 空きには「0」を設定してください。「0」以外を設定した場合、Options範囲外(エラーコード: 1ABB)となります。

ビット	名称	内容
16	Mコード出力タイミング	Mコードを出力するタイミングを設定します。 • 0: WITHモード • 1: AFTERモード

• Mコード出力タイミング(ビット16)

設定値	内容
0: WITHモード*2	位置決めデータの位置決め開始時にMストロブ(Mstrobe)をTRUEにし、位置決めデータのMコード(Mcode)を有効Mコード(ValidMcode)に格納します。
1: AFTERモード*2	目標位置到達時(ドウェルタイム設定時は設定時間経過後)にMストロブ(Mstrobe)をTRUEにし、Mコード(Mcode)に位置決めデータのMコードを格納します。

*2 各モードの詳細は、下記マニュアルを参照してください。

📖使用するコントローラのユーザーズマニュアル

■ステップモード(StepMode)

ステップモードを設定します。

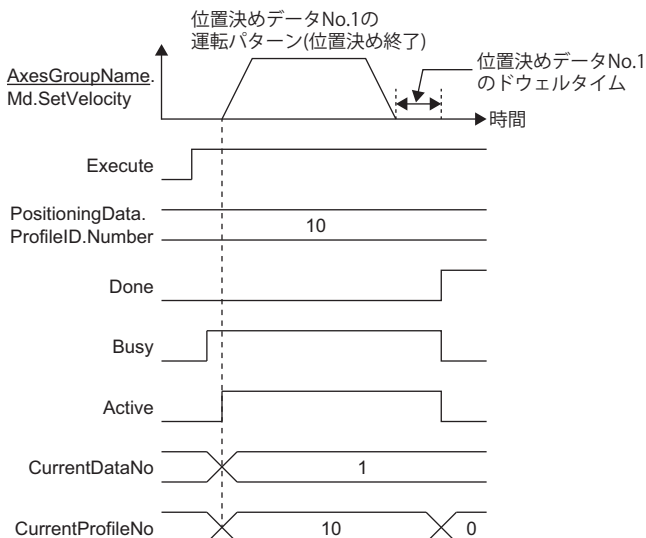
設定値	内容
0: 無視(Ignored)	ステップ機能を無効にします。 ステップモードを無視します。
1: 減速単位ステップ(StepPerDec)	ステップ機能を有効にします。 減速単位でステップ運転を行います。
2: 位置決めデータ単位ステップ(StepPerPositioningData)	ステップ機能を有効にします。 位置決めデータ単位でステップ運転を行います。

■実行中プロファイルNo.(CurrentProfileNo)

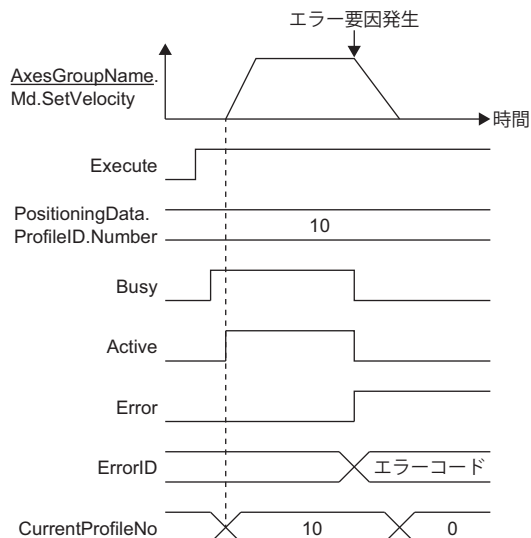
実行中のプロファイルID番号を格納します。

- ・プロファイルID番号に紐づく展開エリアの演算プロファイルを使用中であることを示します。
- ・動作の開始時にプロファイルID番号を格納し、動作完了にて0クリアします。

■正常時



■動作開始後にエラーが発生した場合



- ・実行中プロファイルNo.(CurrentProfileNo)の格納値は、軸グループ構成軸の実行プロファイルID番号 (AxisName.Md.ProfileID)には格納しません。

■必須オブジェクトデータ

本FBを使用する場合、指定した軸グループの構成軸すべてに下記のオブジェクトデータに割り付けたオブジェクトをPDOにマッピングしてください。

- ・ Target position(607AH)

オブジェクトをPDOにマッピングしていない構成軸がある場合、必須オブジェクトデータ未設定(エラーコード: 1AF7H)となり始動しません。

オブジェクトデータ設定に関する詳細は、下記のマニュアルを参照してください。

📖使用するコントローラのユーザズマニュアル

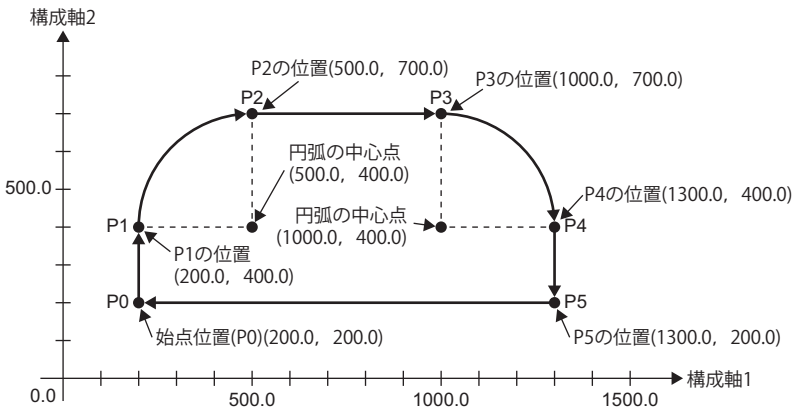
注意事項

- ・本FBは、先読みや外部信号設定などの演算プロファイル解析により、始動や多重起動(Aborting)の動作が遅延する場合があります。

プログラム例

軸グループ1(AxesGroup001)の位置決め制御を行う例を示します。

・動作



位置決めデータNo.	軌跡	位置決め動作
1	P0→P1	始点位置(P0)から位置(P1)へ絶対値直線補間(LinearAbsolute)による位置決め
2	P1→P2	位置(P1)から位置(P2)へ絶対値円弧補間(CircularAbsolute)による位置決め
3	P2→P3	位置(P2)から位置(P3)へ絶対値直線補間(LinearAbsolute)による位置決め
4	P3→P4	位置(P3)から位置(P4)へ相対値円弧補間(CircularRelative)による位置決め
5	P4→P5	位置(P4)から位置(P5)へ相対値直線補間(LinearRelative)による位置決め
6	P5→P0	位置(P5)から終点位置(P0)へ絶対値直線補間(LinearAbsolute)による位置決め

・軸グループ

項目	設定値
構成軸[1]	Axis0001
構成軸[2]	Axis0002

■軸

軸No.	ラベル名	データ型	コメント
1	Axis0001*1	AXIS_REF	軸1
2	Axis0002*1	AXIS_REF	軸2

*1 円弧補間制御を行うために軸パラメータのソフトウェアストロークリミット対象(AxisName.Pr.SwStrokeLimit_Target)を「有効(「-1: 無効(Invalid)」以外)」にします。(プログラム例では、「1: 指令現在位置(SetPosition)」を設定しています。)

■軸グループ

軸グループNo.	ラベル名	データ型	コメント
1	AxesGroup001	AXES_GROUP_REF	軸グループ1

■演算プロファイル

プロファイルID	ラベル名	データ型	コメント
1	PositioningData0001	MC_POSITIONING_DATA_REF	多軸位置決めデータ1

・演算プロファイル(PositioningData0001(多軸位置決めデータ))の設定

項目		位置決めデータNo.					
		1	2	3	4	5	6
運転パターン		連続軌跡 (BlendingPrevious)	連続軌跡 (BlendingPrevious)	連続軌跡 (BlendingPrevious)	連続軌跡 (BlendingPrevious)	連続位置決め (ContinuousPositioning)	位置決め終了 (PositioningComplete)
制御方式		絶対値直線補間	絶対値円弧補間	絶対値直線補間	相対値円弧補間	相対値直線補間	絶対値直線補間
補間軸	補間軸1	Axis0001	Axis0001	Axis0001	Axis0001	Axis0001	Axis0001
	補間軸2	Axis0002	Axis0002	Axis0002	Axis0002	Axis0002	Axis0002
円弧補間モード		—	中心点指定	—	中心点指定	—	—
経路選択		—	CW	—	CW	—	—

項目		位置決めデータNo.					
		1	2	3	4	5	6
目標位置/移動量/終点	補間軸1	200.0	500.0	1000.0	300.0	0.0	200.0
	補間軸2	400.0	700.0	700.0	-300.0	-200.0	200.0
方向選択	補間軸1	正方向	—	正方向	—	—	正方向
	補間軸2	正方向	—	正方向	—	—	正方向
境界点/中心点/半径	補間軸1	—	500.0	—	0.0	—	—
	補間軸2	—	400.0	—	-300.0	—	—
速度モード		合成速度指定	—	合成速度指定	—	合成速度指定	合成速度指定
速度		100.0	150.0	100.0	150.0	100.0	100.0
加減速方式		加減速度指定方式	—	—	—	—	—
加速度/減速度時間		400.0	0.0	0.0	0.0	0.0	0.0
減速度		500.0	0.0	0.0	0.0	0.0	0.0
ジャーク		500.0	0.0	0.0	0.0	0.0	0.0
円弧補間誤差許容差値		—	—	—	—	—	—
リングカウンタを超えた目標位置指定		許可しない	—	許可しない	—	—	許可しない
ドウェルタイム		500.0	0.0	0.0	0.0	0.3	0.0
Mコード		0	0	0	0	0	0
Mコード出力タイミングオーバーライド		—	—	—	—	—	—

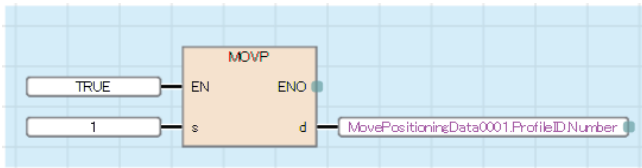
■使用するラベル

ラベル名	データ型	コメント
MCv_MovePositioningData_1	MCv_MovePositioningData	多軸位置決めデータ運転FB
bMovePosDataDone	ビット	実行完了
bMovePosDataBusy	ビット	実行中
bMovePosDataActive	ビット	制御中
bMovePosDataCommandAborted	ビット	実行中断
bMovePosDataError	ビット	エラー
uwMovePosDataErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
uwMovePosDataCurrentDataNo	ワード[符号なし]/ビット列[16ビット]	実行中位置決めデータNo.
uwMovePosDataPrereadingDataNo	ワード[符号なし]/ビット列[16ビット]	先読み位置決めデータNo.
uwMovePosDataRemainingLoopCount	ワード[符号なし]/ビット列[16ビット]	残ループ回数
MC_GroupEnable_1	MC_GroupEnable	軸グループ有効FB
bMovePositioningDataCMD	ビット	多軸位置決めデータ運転始動
bGroupEnableDone	ビット	軸グループ有効完了
bGroupEnableBusy	ビット	実行中
bGroupEnableError	ビット	エラー
uwGroupEnableErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
MC_GroupDisable_1	MC_GroupDisable	軸グループ無効FB
bGroupDisableDone	ビット	軸グループ無効完了
bGroupDisableBusy	ビット	実行中
bGroupDisableError	ビット	エラー
uwGroupDisableErrorID	ワード[符号なし]/ビット列[16ビット]	エラーコード
MovePositioningData0001*1	MC_POSITIONING_DATA_ID	多軸位置決めデータ0001

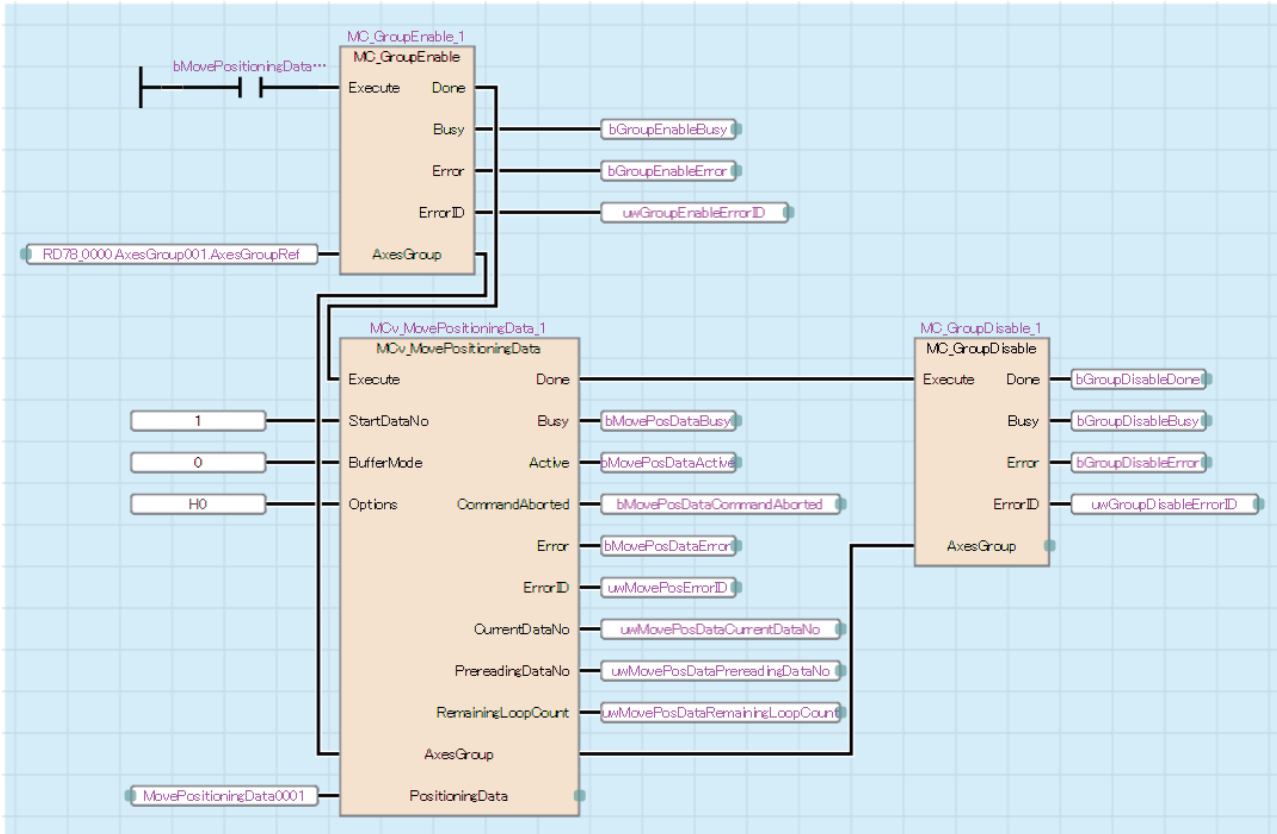
*1 FBD/LDのプログラムのみ使用(演算プロファイルをFBD/LDのプログラムで使用するために必要)

■FBD/LDのプログラム

- 2軸直線補間制御用データ設定



- 軸グループ有効/多軸位置決めデータ運転/軸グループ無効



■STのプログラム

//-----軸グループ有効-----

```
MC_GroupEnable_1(  
    AxesGroup:= AxesGroup001.AxesGroupRef ,  
    Execute:= bMovePositioningDataCMD ,  
    Done=> bGroupEnableDone ,  
    Busy=> bGroupEnableBusy ,  
    Error=> bGroupEnableError ,  
    ErrorID=> uwGroupEnableErrorID  
);
```

//-----多軸位置決めデータ運転-----

```
MCv_MovePositioningData_1(  
    AxesGroup:= AxesGroup001.AxesGroupRef ,  
    PositioningData:= PositioningData0001.PositioningData ,  
    Execute:= bGroupEnableDone ,  
    StartDataNo:= 1 ,  
    BufferMode:= MC_BUFFER_MODE__mcAborting ,  
    Options:= H00000000 ,  
    Done=> bMovePosDataDone ,  
    Busy=> bMovePosDataBusy ,  
    Active=> bMovePosDataActive ,  
    CommandAborted=> bMovePosDataCommandAborted ,  
    Error=> bMovePosDataError ,  
    ErrorID=> uwMovePosDataErrorID ,  
    CurrentDataNo=> uwMovePosDataCurrentDataNo ,  
    PrereadingDataNo=> uwMovePosDataPrereadingDataNo ,  
    RemainingLoopCount=> uwMovePosDataRemainingLoopCount  
);
```

//-----軸グループ無効-----

```
MC_GroupDisable_1(  
    AxesGroup:= AxesGroup001.AxesGroupRef ,  
    Execute:= bMovePosDataDone ,  
    Done=> bGroupDisableDone ,  
    Busy=> bGroupDisableBusy ,  
    Error=> bGroupDisableError ,  
    ErrorID=> uwGroupDisableErrorID  
);
```


42.25 絶対位置追従制御

42

MCv_PositionControl

指定された軸を制限速度，加速度，減速度，ジャークに従い目標位置まで移動させる機能です。また，指令が目標位置へ到達しても，再起動，連続更新で目標位置を変更し軸を移動できます。

ラダー	FBD/LD	ST
		<pre>MCv_PositionControl(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, ContinuousUpdate:= ?BOOL?, Position:= ?LREAL?, Velocity:= ?LREAL?, Acceleration:= ?LREAL?, Deceleration:= ?LREAL?, Jerk:= ?LREAL?, Direction:= ?INT?, BufferMode:= ?INT?, Options:= ?DWORD?, InAddress=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については，下記を参照してください。 📖 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	TRUE, FALSE	FALSE	TRUEで本FBを実行します。
ContinuousUpdate	連続更新	BOOL	起動時	TRUE, FALSE	FALSE	TRUEの間連続して目標位置，速度，加速度，減速度を変更可能です。
Position	目標位置	LREAL	起動時/再起動可 /連続更新可	■リングカウンタ無効時 -100000000000.0≤目標位置<100000000000.0 ■リングカウンタ有効時 リングカウンタ下限値≤ 目標位置<リングカウン タ上限値	0.0	目標位置を設定します。 詳細は，下記を参照してください。 📖 1822ページ 目標位置(Position)
Velocity	速度	LREAL	起動時/再起動可 /連続更新可	0.0, 0.0001~2500000000.0	0.0	速度を設定します。 詳細は，下記を参照してください。 📖 1824ページ 速度(Velocity)
Acceleration	加速度	LREAL	起動時/再起動可 /連続更新可	0.0000, 0.0001~2147483647.0	0.0	加速度を設定します。 詳細は，下記を参照してください。 📖 1824ページ 加速度(Acceleration)
Deceleration	減速度	LREAL	起動時/再起動可 /連続更新可	0.0000, 0.0001~2147483647.0	0.0	減速度を設定します。 詳細は，下記を参照してください。 📖 1824ページ 減速度(Deceleration)
Jerk	ジャーク	LREAL	起動時	0.0000, 0.0001~2147483647.0	0.0	ジャークを設定します。 詳細は，下記を参照してください。 📖 1824ページ ジャーク(Jerk)

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Direction	方向選択	INT (MC_DIRECTION [0..15])	起動時/再起動可 /連続更新可	1, 2, 3	3	方向選択を設定します。 詳細は、下記を参照してください。 📖 1825ページ 方向選択(Direction)
BufferMode	バッファ モード選択	INT(MC_BUFFER _MODE)	起動時	0,1,2,3,4,5	0	バッファモードを設定します。 詳細は、下記を参照してください。 📖 1828ページ バッファモード (BufferMode)
Options	オプション	DWORD(HEX)	起動時	00000000H~00070021H	00000000H	本FBの機能オプションを設定します。 詳細は、下記を参照してください。 📖 1829ページ オプション(Options)

■出力変数

出力変数	名称	データ型	初期値	内容
InAddress	目標位置到達	BOOL	FALSE	モーションシステムで計算した指令位置 (AxisName.Md.SetPosition)が目標位置に到達した ときに、TRUEになります。 連続更新(ContinuousUpdate)が有効(TRUE)時に目 標位置を変更した場合、変更後の目標位置に到達 するまでFALSEになります。
Busy	実行中	BOOL	FALSE	本FBが実行中のときに、TRUEになります。
Active	制御中	BOOL	FALSE	本FBが軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	他のFBにより実行中断したときに、TRUEになりま す。 異常が発生して、本FBが中止するとき、または異 常発生中に本FBを起動するときに、TRUEになりま す。 実行指令(Execute)がFALSEになると、FALSEになり ます。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 📖 使用するコントローラのユーザーズマニュアル

機能

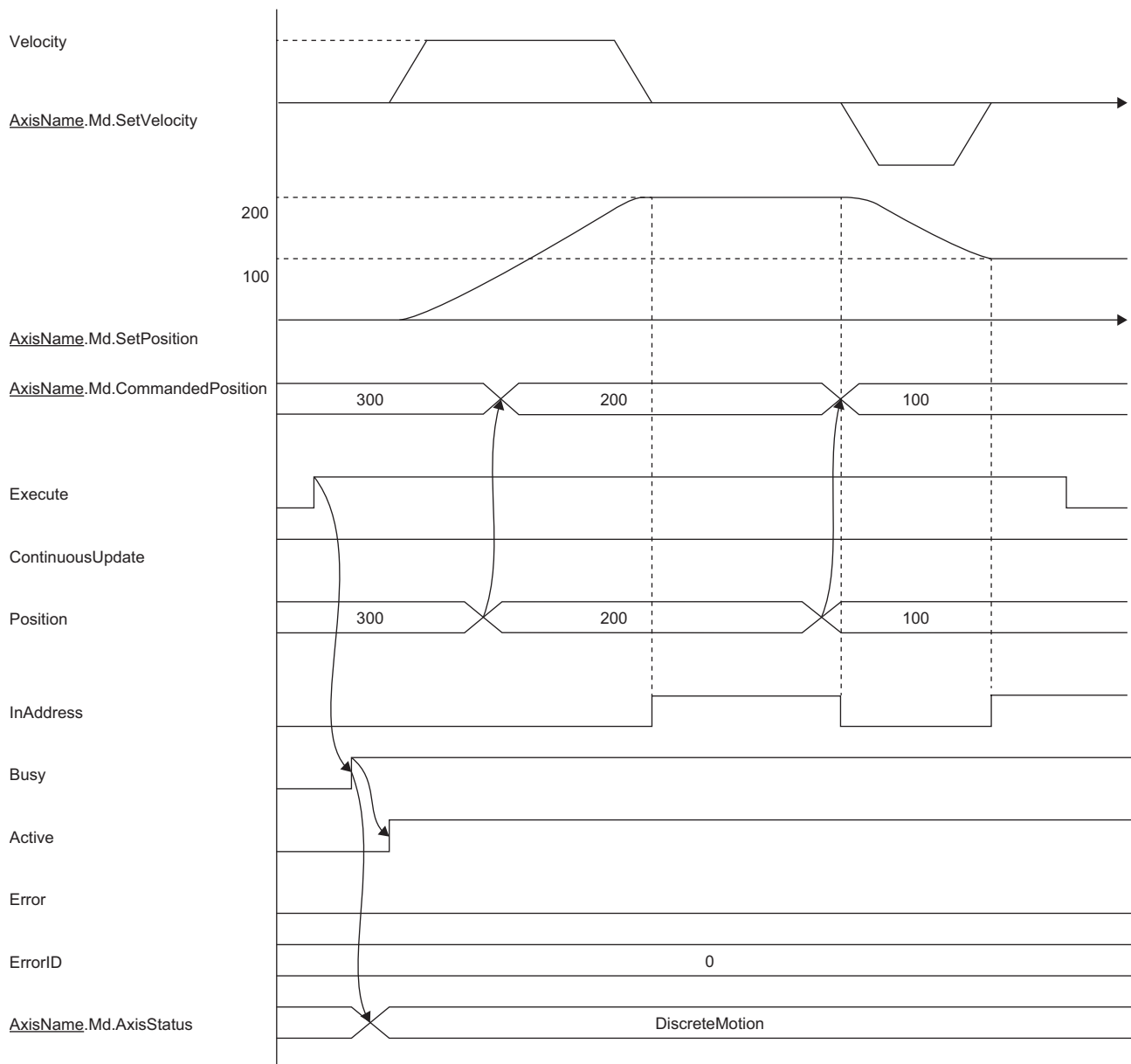
指定した軸(Axis)を制限速度、加速度(Acceleration)、減速度(Deceleration)、ジャーク(Jerk)に従い目標位置(Position)まで移動させます。また、指令が目標位置へ到達しても、再起動、連続更新(ContinuousUpdate)で目標位置(Position)を変更し軸を移動できます。

目標位置(Position)へ到達前に目標位置(Position)を変更した場合、変更後の目標位置へ移動します。

本FBは、MC_Stop(強制停止)または他の動作系FBを実行することで実行完了になります。

■タイミングチャート

・正常完了の場合



・異常完了の場合

異常完了時のタイミングチャートについては、下記を参照してください。

1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■目標位置(Position)

軸の単位に応じた絶対位置の目標位置を設定します。設定値がマイナス値の場合は、負方向へ移動します。0を設定した場合、軸は動作しません対象軸の軸状態(AxisName.Md.AxisStatus)は、「5: 位置決め運転中(DiscreteMotion)」になります。有効範囲は下記となります。範囲外の値を設定した場合、目標位置範囲外(エラーコード: 1A85H)となり始動しません。

リングカウンタの設定	有効範囲
無効	$-10000000000.0 \leq \text{目標位置} < 10000000000.0$
有効	リングカウンタ下限値 $\leq$ 目標位置 $<$ リングカウンタ上限値

ソフトウェアストロークリミット有効時(ソフトウェアストロークリミット対象(AxisName.Md.SwStrokeLimit_Target)が「-1: 無効(Invalid)」以外のとき)

- ソフトウェアストロークリミットオーバとなる位置を目標位置として設定した場合、ソフトウェアストロークリミットオーバ(目標位置)(エラーコード: 1A80H)となり始動しません。
- 方向選択(Direction)の設定を無視し、ソフトウェアストロークリミット範囲外をまたがない方向での位置決め制御を行います。
- 正方向および負方向でソフトウェアストロークリミット範囲外をまたがない場合、現在位置を基準に、目標位置(Position)へ近い方向(移動量の絶対値が短い方)に位置決め制御を行います。
- 正方向および負方向で距離が同じ場合、ひとつ前に実行したモーション制御FBの移動方向または前回実行時の目標位置変更時の移動方向を引き継いで動作します。前回の移動方向がない場合は正方向に動作します。

ソフトウェアストロークリミット無効時(ソフトウェアストロークリミット対象(AxisName.Md.SwStrokeLimit_Target)が「-1: 無効(Invalid)」のとき)、有効範囲内で位置決め制御を行います。

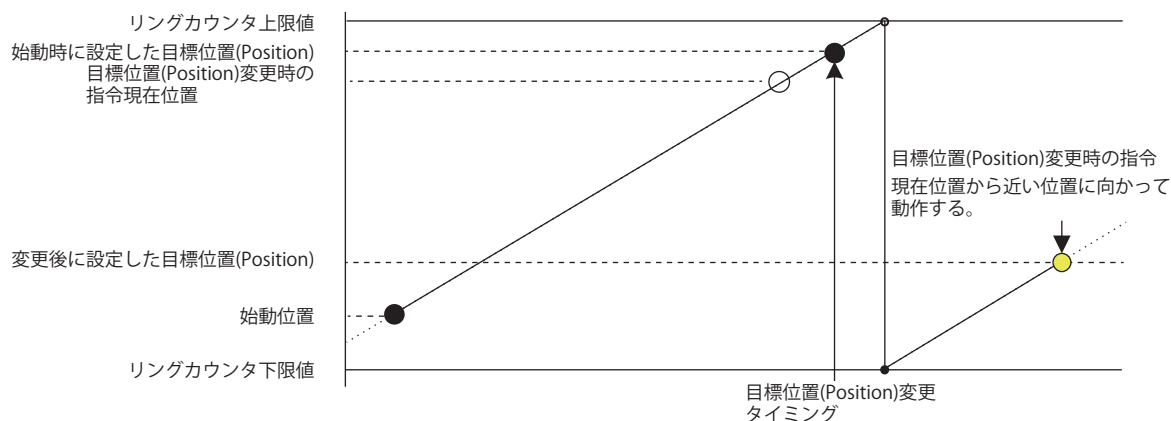
目標位置到達(InAddress)がTRUEになる1演算周期前は、目標位置変更が無視されます。

目標位置(Position)を変更した場合、方向選択(Direction)の設定により変更時の動作が異なります。

- モーション制御FB実行時に目標位置(Position)に範囲外の値を設定した場合、Position/Distance設定範囲外警告(エラーコード: 0D0DH)が発生し、変更前の設定で動作を継続します。
- Position/Distance設定範囲外警告(エラーコード: 0D0DH)発生時、方向選択(Direction)を変更している場合は、変更前の方向選択(Direction)の設定で動作を継続します。
- 方向選択範囲外警告(エラーコード: 0D0DH)発生時、変更前の目標位置(Position)の設定で動作を継続します。

例

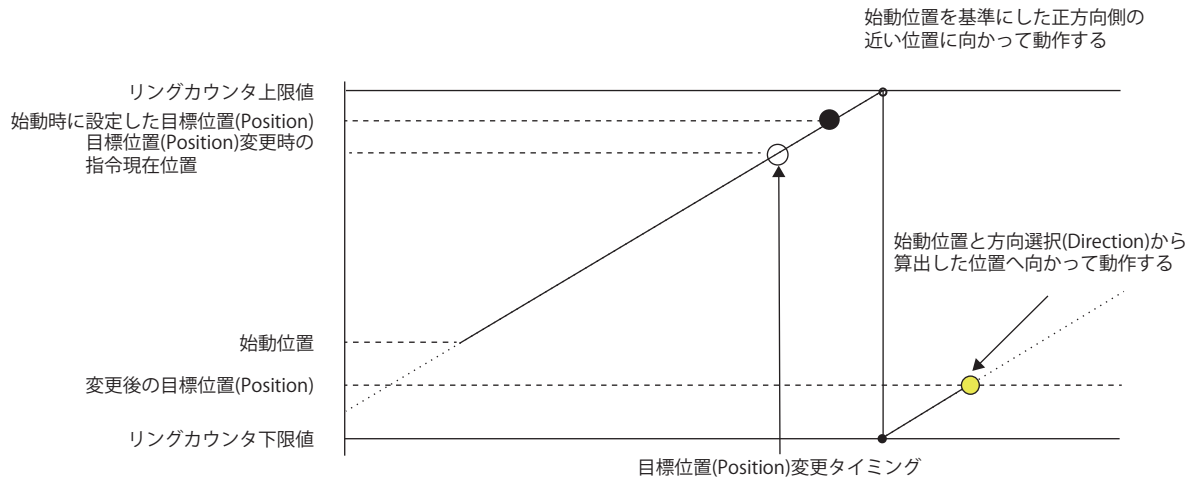
最短経路を設定する場合、変更時の指令現在位置から変更後の目標位置(Position)まで近い方向に向かって動作します。下記の「始動」とは初回後のモーション制御FBの起動および、前回の動作で目標位置到達(InAddress)がTRUEになった後の目標位置(Position)の変更を示します。



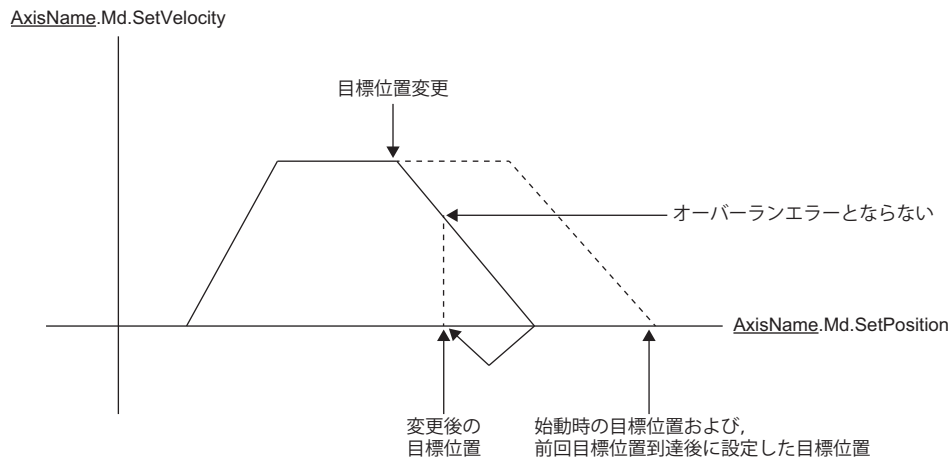
例

最短経路以外を設定する場合、変更時の指令現在位置と方向選択(Direction)から目標位置(Position)を算出します。

下記の「始動」は初回後のモーション制御FBの起動および、前回の動作で目標位置到達(InAddress)がTRUEになった後の目標位置(Position)の変更を示します。



目標位置(Position)の変更により反転動作になる場合や、変更の際に変更後の目標位置(Position)を通り過ぎてしまった場合は、減速停止を行います。減速停止完了後、変更された目標位置(Position)に向け動作を開始します。この場合オーバーランによるエラーは発生しません。



■速度(Velocity)

本FBでの指令速度を設定します。

設定範囲

0.0, 0.0001~2500000000.0

設定を省略したまたは0.0を設定した場合、軸は動作しませんが対象軸の軸状態(AxisName.Md.AxisStatus)は、「6: 連続動作運転中(ContinuousMotion)」になります。

範囲外の値を設定した場合、速度範囲外(エラーコード: 1ABAH)が発生し始動しません。

モーション制御FB実行中の動作は、使用するコントローラのユーザーズマニュアルを参照してください。

■加速度(Acceleration)

本FBでの加速度を設定します。加減速方式の詳細は使用するコントローラのユーザーズマニュアルを参照してください。

範囲外の値を設定した場合、加速度範囲外(エラーコード: 1A8DH)が発生し始動しません。

モーション制御FB実行中の動作は、使用するコントローラのユーザーズマニュアルを参照してください。

■減速度(Deceleration)

本FBでの減速度を設定します。加減速方式の詳細は使用するコントローラのユーザーズマニュアルを参照してください。

範囲外の値を設定した場合、減速度範囲外(エラーコード: 1A8EH)が発生し始動しません。

モーション制御FB実行中の動作は、使用するコントローラのユーザーズマニュアルを参照してください。

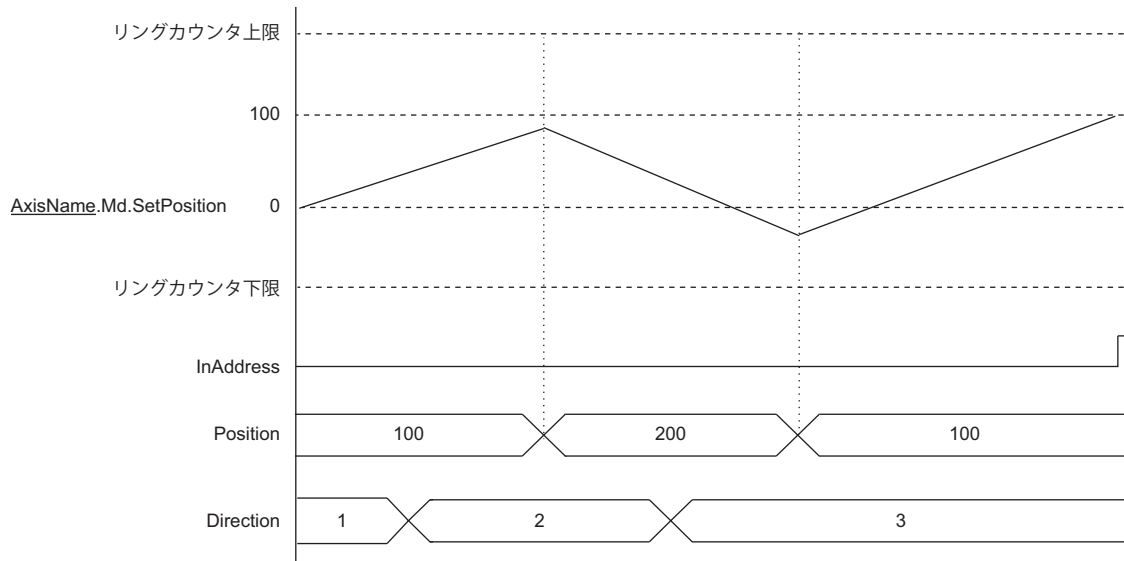
■ジャーク(Jerk)

本FBでのジャークを設定します。加減速方式の詳細は使用するコントローラのユーザーズマニュアルを参照してください。

範囲外の値を設定した場合、ジャーク範囲外(エラーコード: 1A8FH)が発生し始動しません。

■方向選択(Direction)

動作する方向を指定します。連続更新および再起動する場合、方向選択(Direction)は値を変更したタイミングでは反映されず、目標位置(Position)を変更したときに反映されます。



下記を選択できます。

- 1: 正方向(mcPositiveDirection)
- 2: 負方向(mcNegativeDirection)
- 3: 最短経路(mcShortestWay)

始動時、方向選択(Direction)に範囲外の値を設定すると、方向選択範囲外(エラーコード: 1AA5H)が発生し、始動しません。制御中、方向選択(Direction)に範囲外の値を設定すると、方向選択範囲外警告(エラーコード: 0D4DH)が発生し、変更前の動作を続けます。このとき目標位置(Position)を方向選択(Direction)と同時に変更している場合、変更前の目標位置(Position)の設定で動作を続けます。

ソフトウェアストロークリミット有効時は、方向選択(Direction)の設定を無視します。

ソフトウェアストロークリミット無効時は、方向選択(Direction)の「1: 正方向(mcPositiveDirection)」、「2: 負方向(mcNegativeDirection)」、「3: 最短経路(mcShortestWay)」から選択でき、現在位置から目標位置(Position)へ移動する方向を設定します。

• 1: 正方向(mcPositiveDirection)

現在位置から目標位置に向かって正方向(アドレス増加)の位置決めを行います。

設定範囲	動作
現在値 ≤ 目標位置 < リングカウンタ上限値	The graph shows a coordinate system with 'リングカウンタ 上限値' (Ring Counter Upper Limit) at 100.0 and 'リングカウンタ 下限値' (Ring Counter Lower Limit) at -100.0. A horizontal line at 0.0 represents the '現在値「0.0」' (Current Value '0.0'). A dashed diagonal line represents the target position. An arrow points from the current value towards the target position, indicating movement in the positive direction.
リングカウンタ下限値 ≤ 目標位置 < 現在値	The graph shows a coordinate system with 'リングカウンタ 上限値' (Ring Counter Upper Limit) at 100.0 and 'リングカウンタ 下限値' (Ring Counter Lower Limit) at -100.0. A horizontal line at 0.0 represents the '現在値「0.0」' (Current Value '0.0'). A dashed diagonal line represents the target position. An arrow points from the current value towards the target position, indicating movement in the positive direction.

• 2: 負方向(mcNegativeDirection)

現在位置から負方向(アドレス減少)の目標位置に向かって位置決めを行います。

設定範囲	動作
リングカウンタ下限値 ≤ 目標位置 ≤ 現在値	The graph shows a coordinate system with 'リングカウンタ 上限値' (Ring Counter Upper Limit) at 100.0 and 'リングカウンタ 下限値' (Ring Counter Lower Limit) at -100.0. A horizontal line at 0.0 represents the '現在値「0.0」' (Current Value '0.0'). A dashed diagonal line represents the target position. An arrow points from the current value towards the target position, indicating movement in the negative direction.
現在値 < 目標位置 < リングカウンタ上限値	The graph shows a coordinate system with 'リングカウンタ 上限値' (Ring Counter Upper Limit) at 100.0 and 'リングカウンタ 下限値' (Ring Counter Lower Limit) at -100.0. A horizontal line at 0.0 represents the '現在値「0.0」' (Current Value '0.0'). A dashed diagonal line represents the target position. An arrow points from the current value towards the target position, indicating movement in the negative direction.

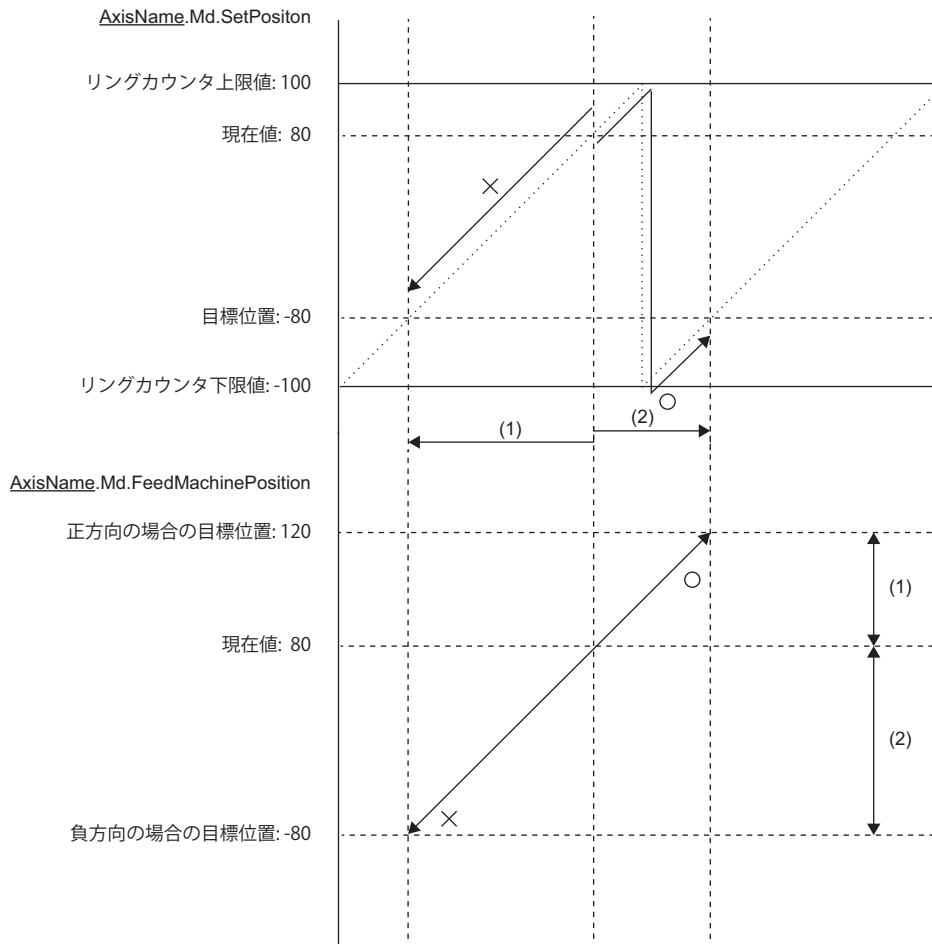
• 3: 最短経路(mcShortestWay)

現在位置を基準にして、目標位置へ近い方向(移動量の絶対値が短い方)での位置決め制御を行います。正方向、負方向で距離が同じ場合は現在方向で動作します。目標位置の設定範囲は、「リングカウンタ下限値 ≤ 目標位置 < リングカウンタ上限値」です。範囲外の値を設定した場合は、目標位置範囲外(エラーコード: 1A85H)となり始動しません。

例

下記の値を設定した場合の動作

- リングカウンタ上限値: 100.0
- リングカウンタ下限値: -100.0
- 現在位置: 80.0
- 目標位置: -80.0



(1)の場合は移動量が「160.0」、(2)の場合は移動量が「40.0」となるため、(2)の正方向で移動します。

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

本FBで設定可能なバッファモードを下記に示します。

設定値	内容
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。
1: Buffered(mcBuffered)	制御中のFBに次のFBをバッファリングします。 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFB完了にて、バッファリングFBを順次実行します。
2: BlendingLow(mcBlendingLow)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBとバッファリングFBの目標速度のうち低い方の速度を切換え速度とします。
3: BlendingPrevious(mcBlendingPrevious)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBの目標速度を切換え速度とします。
4: BlendingNext(mcBlendingNext)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、バッファリングFBの目標速度を切換え速度とします。
5: BlendingHigh(mcBlendingHigh)	制御中のFBに次のFBをバッファリングします。 ^{*1} 制御中のFBをすでにバッファリングしている場合、直前のFBの次にバッファリングします。(最大2つ) 制御中のFBが目標位置まで到達したのち、バッファリングFBを順次実行します。 制御中のFBが目標位置に到達したとき、制御中のFBとバッファリングFBの目標速度のうち高い方の速度を切換え速度とします。

*1 制御中のFBとバッファリングFB間で停止を行いません。

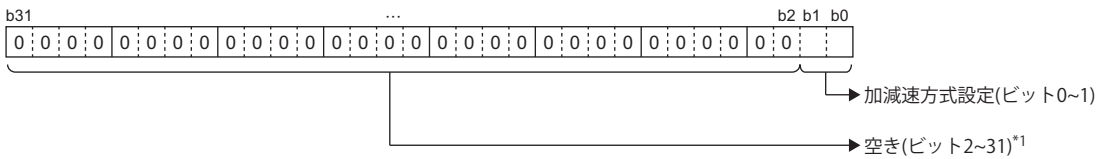
Point

多重起動(バッファモード)の詳細については、下記を参照してください。

 使用するコントローラのユーザーズマニュアル

■オプション(Options)

本FBで使用する機能オプションをビット指定で設定します。
ビット指定で設定する内容を下記に示します。



*1 空きには「0」を設定してください。「0」以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

ビット	名称	内容
0~1	加減速方式設定	制御を行うための加減速方式を設定します。 • 0: 加減速度指定方式(mcAccDec) • 1: 加減速時間一定方式(mcFixedTime)

• 加減速方式設定(ビット0~1)

設定値	内容
0: 加減速度指定方式(mcAccDec)	本FBで設定した加速度(Acceleration), 減速度(Deceleration), ジャーク(Jerk)を用いて, 加速/減速する方式です。
1: 加減速時間一定方式(mcFixedTime)	速度に関係なく, 本FBで設定した加減速時間を用いて, 加速/減速する方式です。 加減速時間は, 加速度(Acceleration)に設定し, 減速度(Deceleration), ジャーク(Jerk)は使用しません。

注意事項

目標位置(Position)と方向選択(Direction)を同時に変更した場合、泣き別れが発生する可能性があります。方向選択(Direction)を先に変更してから目標位置(Position)を変更することで、目標位置(Position)変更時に、両方の変更が適応されます。目標位置(Position)と方向選択(Direction)の値を同時に変更しなければならない場合は、本FBをプログラム実行周期を“定周期”で実行するか、本FBを再起動させてください。

プログラム例

本FBのプログラム例については、下記を参照してください。
📖使用するコントローラのユーザーズマニュアル

42.26 モーションサイクリック位置制御

MCv_CyclicPosition

指定した目標位置に従いドライブユニットに目標位置を送信します。

ラダー	FBD/LD	ST
		<pre>MCv_CyclicPosition(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, ContinuousUpdate:= ?BOOL?, Position:= ?LREAL?, Direction:= ?INT?, BufferMode:= ?INT?, Options:= ?DWORD?, InAddress=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、 下記を参照してください。 📖 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	—	FALSE	TRUEで本FBを実行します。
Continuous Update	連続更新	BOOL	起動時	—	FALSE	TRUEの間連続して目標位置を変更可能です。
Position	目標位置	LREAL	起動時/再起動可 /連続更新可	■リングカウンタが無効の場合 -10000000000.0≤設定値 <10000000000.0 ■リングカウンタが有効の場合 リングカウンタ下限値≤設定値 <リングカウンタ上限値	0	絶対位置の目標位置を位置決め 範囲で設定します。 詳細は、下記を参照してくだ さい。 📖 1833ページ 目標位置 (Position)
Direction	方向選択	MC_DIRECTION	起動時/再起動可 /連続更新可	1, 2, 3	0	ソフトウェアストロークリミ ット無効時に、現在位置から目標 位置へ移動する方向を指定しま す。 詳細は、下記を参照してくだ さい。 📖 1834ページ 方向選択 (Direction)
BufferMode	バッファモード	MC_BUFFER_MODE	起動時	mcAborting	mcAborting	バッファモードを設定します。 詳細は、下記を参照してくだ さい。 📖 1837ページ バッファモー ド(BufferMode)
Options	オプション	DWORD(HEX)	起動時	0固定	0	本FBでは設定できません。0に 設定してください。0以外を設 定した場合、Options範囲外(エ ラーコード: 1ABBH)となりま す。

■出力変数

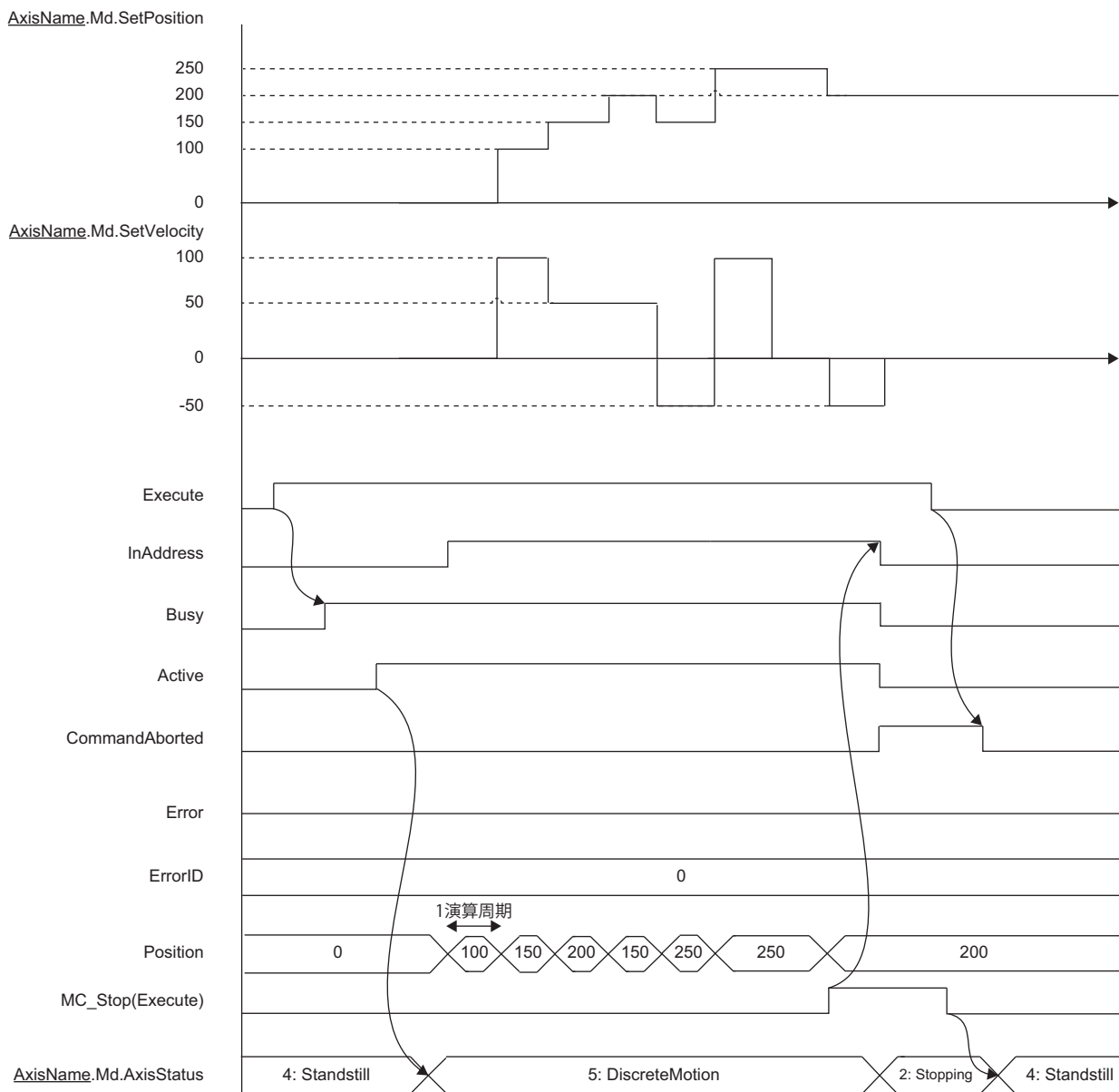
出力変数	名称	データ型	初期値	内容
InAddress	目標位置到達	BOOL	FALSE	AxisName.Md.SetPositionが目標位置に到達したときに、TRUEになります。 異常時または停止要因発生時にFALSEとなります。
Busy	実行中	BOOL	FALSE	本FBが動作中のときに、TRUEになります。
Active	制御中	BOOL	FALSE	本FBが軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	他のFBにより実行中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。  使用するコントローラのユーザーズマニュアル

機能

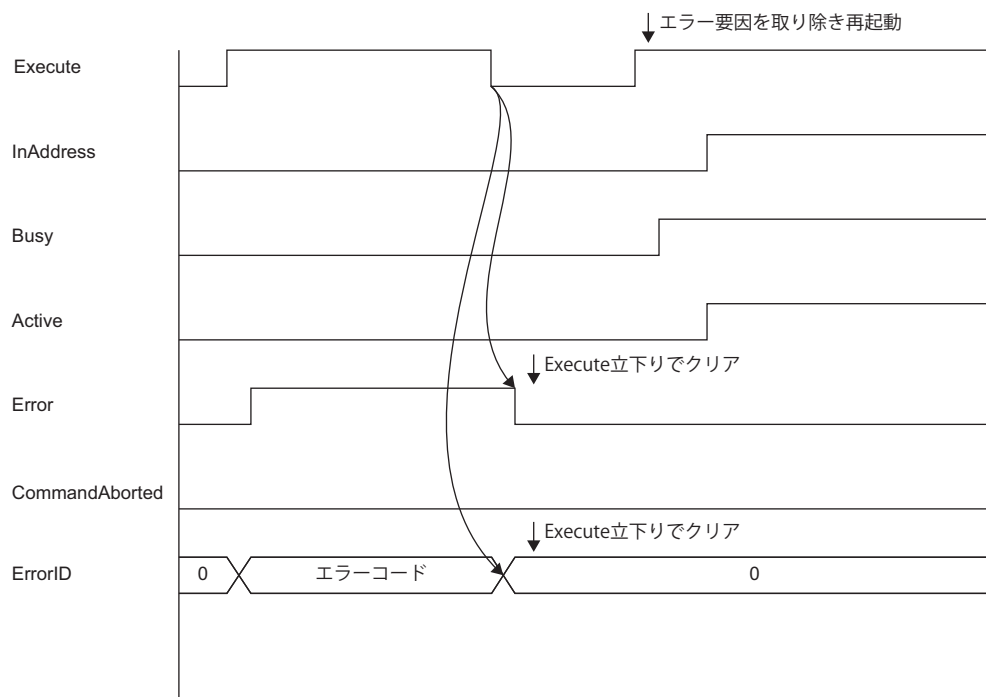
ドライバ側の制御モードをcspに切り換え、指令位置を送信します。本FBを停止するには、MC_Stop(強制停止)を使用してください。

■タイミングチャート

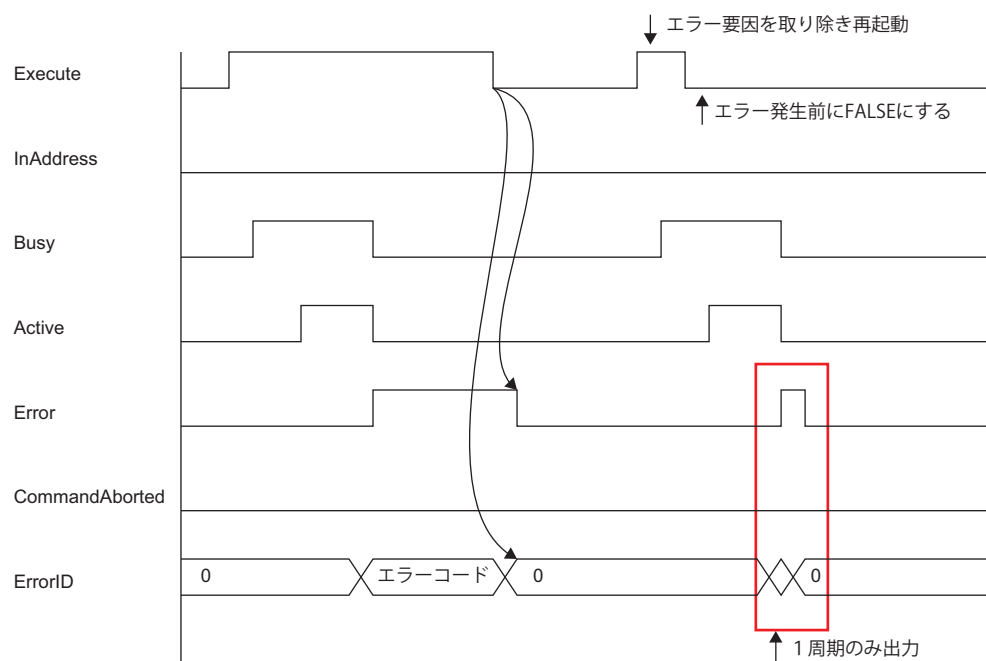
- ・ 正常完了の場合



・ 入出力変数異常の場合



・ 入力変数異常の場合



■目標位置(Position)

絶対位置の目標位置を位置決め範囲で設定します。有効範囲は下記となります。範囲外の値を設定した場合、目標位置範囲外(エラーコード: 1A85H)になります。

リングカウンタの設定	有効範囲
無効	-10000000000.0 ≤ 目標位置 < 10000000000.0
有効	リングカウンタ下限値 ≤ 目標位置 < リングカウンタ上限値

ソフトウェアストロークリミット有効時(ソフトウェアストロークリミット対象(AxisName.Md.SwStrokeLimit_Target)が「-1: 無効(Invalid)」以外のとき)、ソフトウェアストロークリミット範囲外をまたがない方向で位置決め制御を行います。正方向、負方向ともにソフトウェアストロークリミット範囲外をまたがない場合、現在位置を基準にして目標位置へ近い方向(移動量の絶対値が短い方)で位置決め制御を行います。正方向、負方向で距離が同じ場合は前回実行したモーション制御FBの移動方向または前回の目標位置変更時の動作方向を引き継いで動作します。また、有効範囲内であってもソフトウェアストロークリミットオーバとなる位置を目標位置として指定した場合、ソフトウェアストロークリミットオーバ(目標位置)(エラーコード: 1A80H)が発生します。

ソフトウェアストロークリミット無効時(ソフトウェアストロークリミット対象(AxisName.Md.SwStrokeLimit_Target)が「-1: 無効(Invalid)」のとき)の動作は、下記を参照してください。

1834ページ 方向選択(Direction)

目標位置(Position)を変更する場合、下記の範囲で変更してください。満たさず、電子ギア変換後の値が31ビットを超える値を設定した場合、ドライバ位置変換後の移動量範囲外(エラーコード: 1B3CH)が発生します。

目標位置の変化量 × ドライバ単位変換分子 / ドライバ単位変換分母 ≤ 2000000000

例

下記の条件で目標位置を計算例を示します。

項目	値
モータエンコーダ分解能	67108864[pulse/rev]
ボールネジピッチ	1000[μm]
外付けギア比	1/10

電子ギア設定がサーボモータ1回転あたりのパルス数が67108864[pulse]、サーボモータ1回転あたりの移動量が100.0[μm]のとき、目標位置の変化量は下記の範囲以下になります。

$$\text{目標位置の変化量} = 2000000000 \div \frac{\text{ドライバ単位変換分子}}{\text{ドライバ単位変換分母}} = 2000000000 \div \frac{1 \text{ 回転あたりのパルス数}}{1 \text{ 回転あたりの移動量}} = 2000000000 \div \frac{67108864}{100.0} = 2980.23...$$

■方向選択(Direction)

ソフトウェアストロークリミット無効時に、現在位置から目標位置へ移動する方向を指定します。

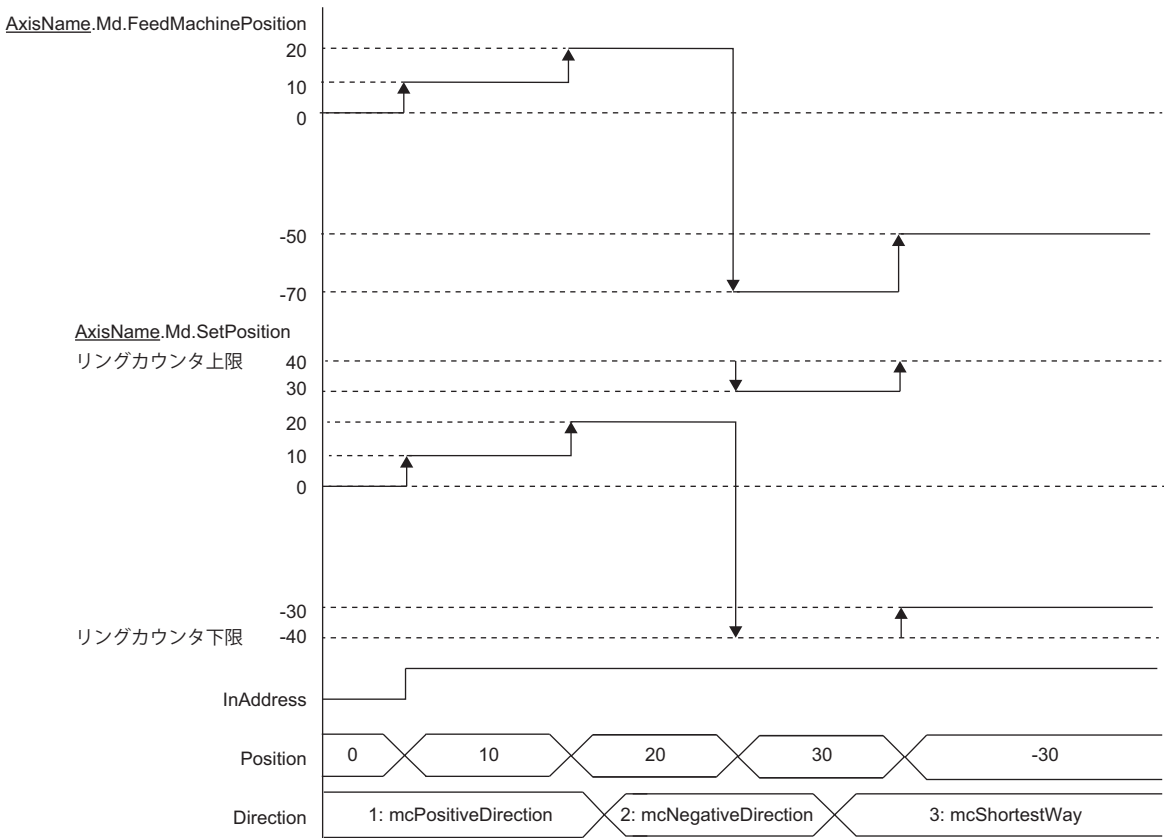
設定範囲は、「1: 正方向(mcPositiveDirection)」, 「2: 負方向(mcNegativeDirection)」, 「3: 最短経路(mcShortestWay)」から選択できます。

ソフトウェアストロークリミット有効時は、方向選択(Direction)の設定を無視します。

始動時に方向選択(Direction)を範囲外の値を設定した場合、方向選択範囲外(エラーコード: 1AA5H)が発生し始動しません。

制御中時に方向選択(Direction)を範囲外の値を設定した場合、方向選択範囲外警告(エラーコード: 0D4DH)が発生し、変更前の動作を続けます。

方向選択(Direction)による移動方向は、モーション制御FB起動時は起動時の指令現在位置、モーション制御FB起動後は目標位置変更時の指令現在位置を基準として動作します。



• 1: 正方向(mcPositiveDirection)

現在位置から目標位置に向かって正方向(アドレス増加)の位置決めを行います。

設定範囲	動作
現在値 ≤ 目標位置 < リングカウンタ上限値	The graph shows a sawtooth pattern for the ring counter. The y-axis is labeled 'リングカウンタ' (Ring Counter) with values -100.0, 0.0, and 100.0. The x-axis represents position. A solid line shows the current value moving from 0.0 towards the target position, which is within the upper limit of the ring counter. The target position is marked with a circle and labeled '現在値「0.0」'.
リングカウンタ下限値 ≤ 目標位置 < 現在値	The graph shows a sawtooth pattern for the ring counter. The y-axis is labeled 'リングカウンタ' (Ring Counter) with values -100, 0, and 100. The x-axis represents position. A solid line shows the current value moving from 0 towards the target position, which is within the lower limit of the ring counter. The target position is marked with a circle and labeled '現在値「0」'.

• 2: 負方向(mcNegativeDirection)

現在位置から負方向(アドレス減少)の目標位置に向かって位置決めを行います。

設定範囲	動作
リングカウンタ下限値 ≤ 目標位置 ≤ 現在値	The graph shows a sawtooth pattern for the ring counter. The y-axis is labeled 'リングカウンタ' (Ring Counter) with values -100.0, 0.0, and 100.0. The x-axis represents position. A solid line shows the current value moving from 0.0 towards the target position, which is within the lower limit of the ring counter. The target position is marked with a circle and labeled '現在値「0.0」'.
現在値 < 目標位置 < リングカウンタ上限値	The graph shows a sawtooth pattern for the ring counter. The y-axis is labeled 'リングカウンタ' (Ring Counter) with values -100, 0, and 100. The x-axis represents position. A solid line shows the current value moving from 0 towards the target position, which is within the upper limit of the ring counter. The target position is marked with a circle and labeled '現在値「0」'.

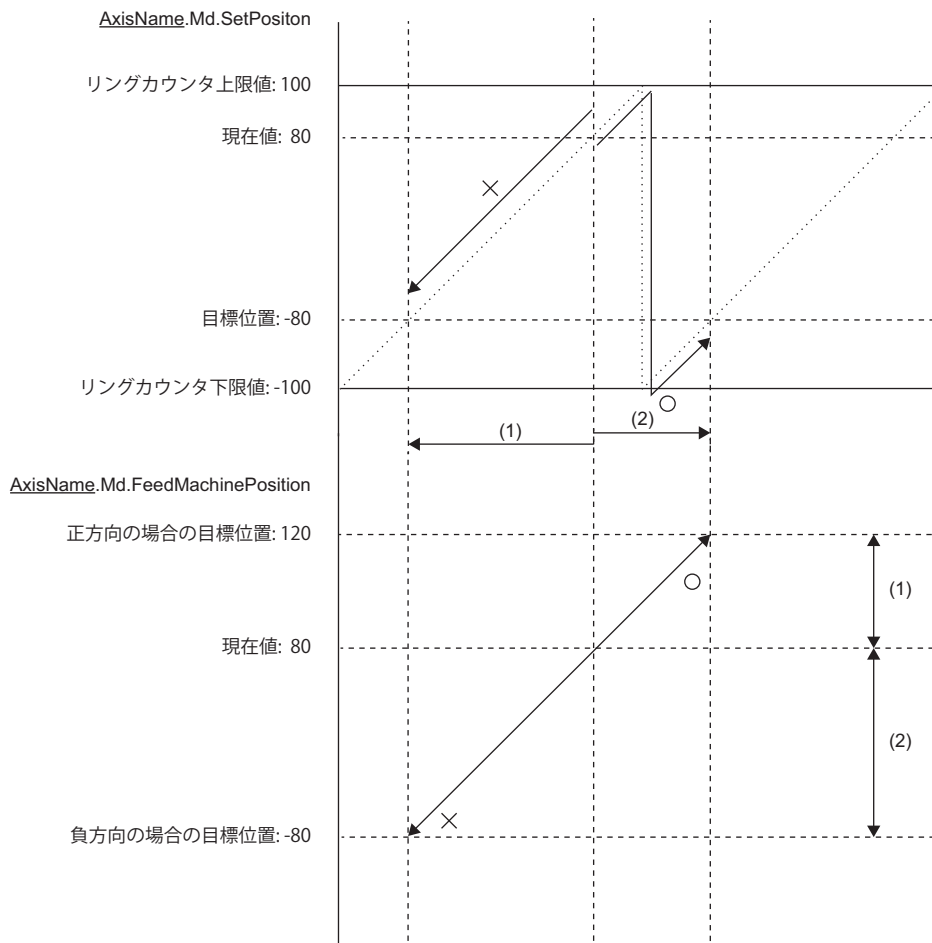
• 3: 最短経路(mcShortestWay)

現在位置を基準にして、目標位置へ近い方向(移動量の絶対値が短い方)での位置決め制御を行います。正方向、負方向で距離が同じ場合は現在方向で動作します。目標位置の設定範囲は、「リングカウンタ下限値 ≤ 目標位置 < リングカウンタ上限値」です。範囲外の値を設定した場合は、目標位置範囲外(エラーコード: 1A85H)となり始動しません。

例

下記の値を設定した場合の動作

- リングカウンタ上限値: 100.0
- リングカウンタ下限値: -100.0
- 現在位置: 80.0
- 目標位置: -80.0



(1)の場合は移動量が「160.0」、(2)の場合は移動量が「40.0」となるため、(2)の正方向で移動します。

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

本FBで設定可能なバッファモードを下記に示します。

設定値	内容
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。

Point

多重起動(バッファモード)の詳細については、下記を参照してください。

 使用するコントローラのユーザーズマニュアル

注意事項

指令現在速度が軸ごとの速度制限値(正方向であれば正方向速度制限値(AxisName.Pr.VelocityLimit_Positive), 負方向であれば負方向速度制限値(AxisName.Pr.VelocityLimit_Negative))を超える場合, 速度制限値オーバー時動作設定(AxisName.Pr.VelocityLimit_OverOperation)に従います。

また, 指令現在速度の絶対値が最大速度を超える場合, 制御中速度範囲オーバー(エラーコード: 1B1AH)となり, 運転を停止します。

プログラム例

本FBのプログラム例については、下記マニュアルを参照してください。

 使用するコントローラのユーザーズマニュアル

42.27 モーションサイクリック速度制御

MCv_CyclicVelocity

指定した目標速度に従いドライブユニットに目標速度を送信します。

ラダー	FBD/LD	ST
		<pre> MCv_CyclicVelocity(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, ContinuousUpdate:= ?BOOL?, Velocity:= ?LREAL?, Direction:= ?INT?, BufferMode:= ?INT?, Options:= ?DWORD?, InVelocity=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?); </pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数 (AxisName.AxisRef.) については、下記を参照してください。 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	—	FALSE	TRUEで本FBを実行します。
ContinuousUpdate	連続更新	BOOL	起動時	—	FALSE	TRUEの間連続して目標位置を変更可能です。
Velocity	目標速度	LREAL	起動時/再起動可/連続更新可	-2500000000.0~2500000000.0	0	目標速度を設定します。。 詳細は、下記を参照してください。 1840ページ 目標速度 (Velocity)
Direction	方向選択	MC_DIRECTION	起動時	mcPositiveDirection正方向(1), mcNegativeDirection負方向(2)	0	方向を指定します。 詳細は、下記を参照してください。 1841ページ 方向選択 (Direction)
BufferMode	バッファモード	MC_BUFFER_MODE	起動時	mcAborting	mcAborting	バッファモードを設定します。 詳細は、下記を参照してください。 1841ページ バッファモード(BufferMode)
Options	オプション	DWORD(HEX)	起動時	0固定	0	本FBでは設定できません。0に設定してください。0以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

■出力変数

出力変数	名称	データ型	初期値	内容
InVelocity	目標速度到達	BOOL	FALSE	AxisName.Md.IO_TargetVelocityが目標速度に到達したときに、TRUEになります。 異常時または停止要因発生時にFALSEとなります。
Busy	実行中	BOOL	FALSE	本FBが実行中のときに、TRUEになります。
Active	制御中	BOOL	FALSE	本FBが軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	他のFBにより実行中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。  使用するコントローラのユーザーズマニュアル

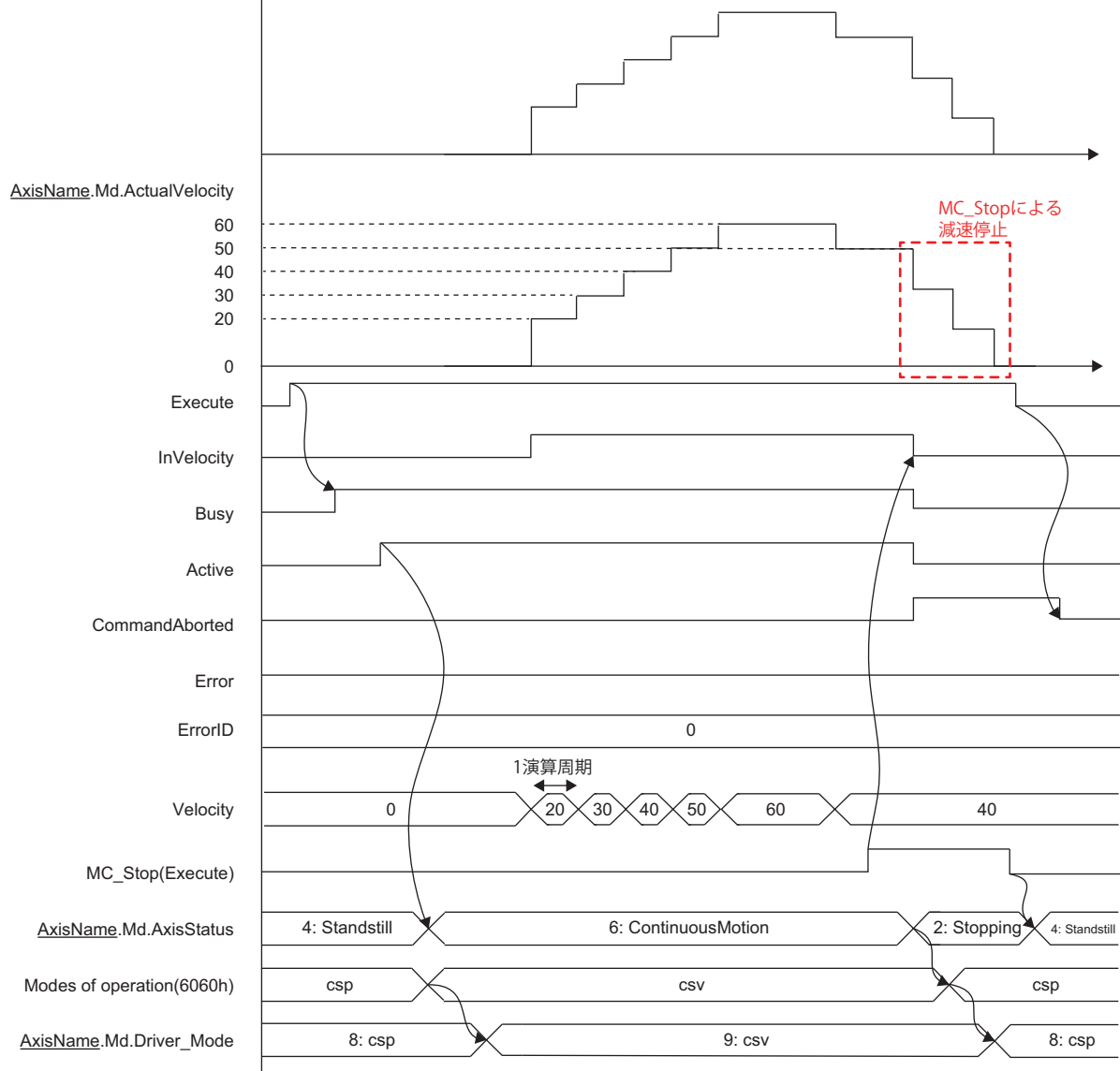
機能

ドライバ側の制御モードをcsvに切り換え、指令速度を送信します。本FBを停止するには、MC_Stop(強制停止)を使用してください。

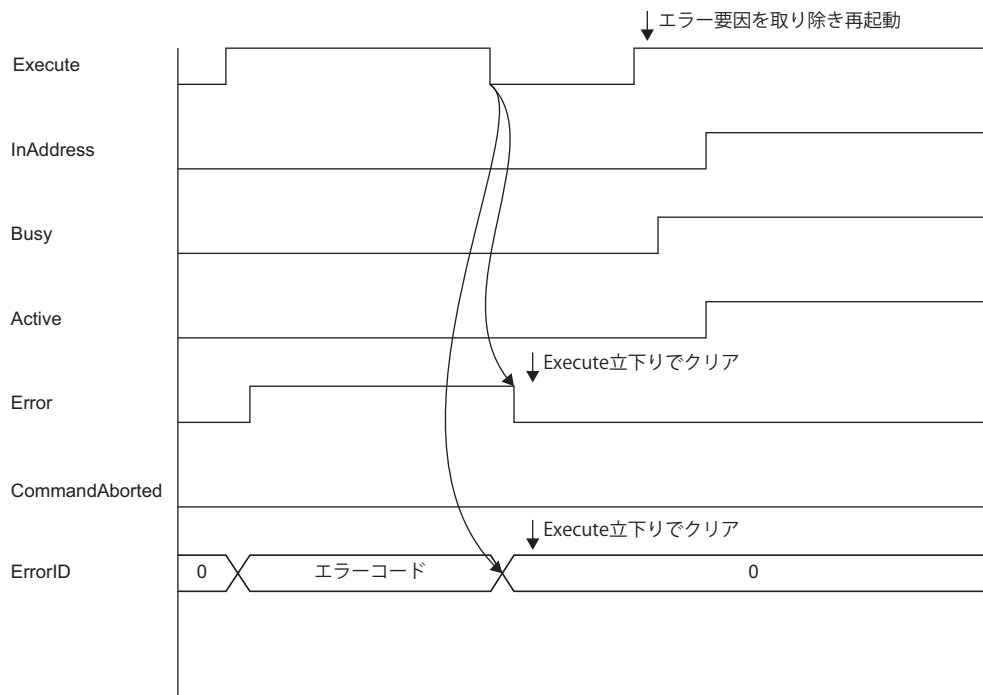
■タイミングチャート

- ・ 正常完了の場合

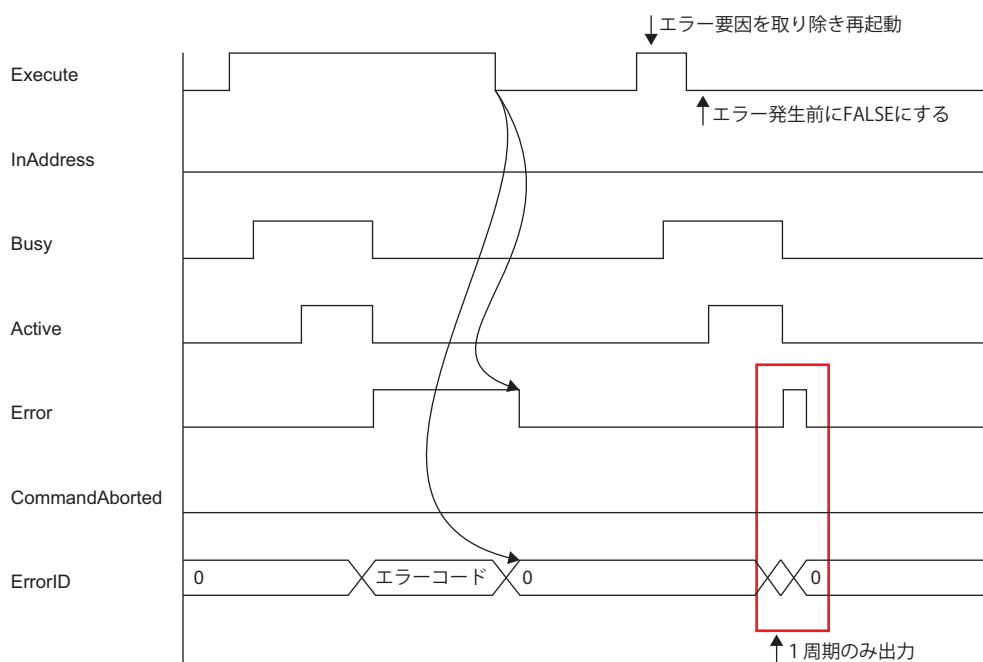
AxisName.Md.Io_TargetVelocity
(具体的な値は設定によるので省略)



・ 入出力変数異常の場合



・ 入力変数異常の場合



■目標速度(Velocity)

目標速度を設定します。始動時に範囲外の値を設定した場合、速度範囲外(エラーコード: 1ABAH)になります。制御中に範囲外を設定した場合、速度範囲外警告(エラーコード: 0D0FH)が発生します。

ドライバ速度単位変換後の指令速度が-2147483648~2147483647の範囲となるように設定してください。変換後の指令速度が範囲外の場合は動作状態に関わらず、ドライバ速度単位変換後の指令速度範囲外(エラーコード: 1A90H)が発生します。ドライバ速度単位変換後の指令速度は下記の式から求められます。

ドライバ速度単位変換後の指令速度=速度指令単位×(ドライバ単位変換分子÷ドライバ単位変換分母)

速度をマイナスに設定した場合、負方向に移動します。

始動時に0を設定した場合、軸は動作しません対象軸の軸状態(AxisName.Md.AxisStatus)は、「6: 連続動作運転中(ContinuousMotion)」になります。

本FBを多重起動し、かつ目標速度(Velocity)に0を設定した場合、前のモーション制御FBの指令速度として動作します。

■方向選択(Direction)

現在位置から目標位置へ移動する方向を指定します。

設定範囲は、「1: 正方向(mcPositiveDirection)」、「2: 負方向(mcNegativeDirection)」、「3: 最短経路(mcShortestWay)」から選択できます。「2: 負方向(mcNegativeDirection)」を設定し、目標速度(Velocity)をマイナスに設定した場合、モータの移動方向は正方向になります。

始動時に方向選択(Direction)を範囲外の値を設定した場合、方向選択範囲外(エラーコード: 1AA5H)が発生し始動しません。

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

本FBで設定可能なバッファモードを下記に示します。

設定値	内容
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。

Point

多重起動(バッファモード)の詳細については、下記を参照してください。

 使用するコントローラのユーザーズマニュアル

注意事項

目標速度を設定する際、現在の速度と目標の速度の差が大きくなるように値を設定してください。差が大きい場合、軸が急加速する場合があります。

本FBを始動中に停止要因が発生した場合、即停止します。

プログラム例

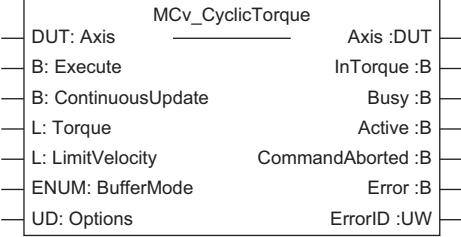
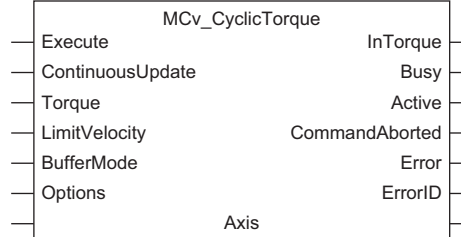
本FBのプログラム例については、下記を参照してください。

 使用するコントローラのユーザーズマニュアル

42.28 モーションサイクリックトルク制御

MCv_CyclicTorque

指定した目標トルクに従いドライブユニットに目標トルクを送信します。

ラダー	FBD/LD	ST
		<pre>MCv_CyclicTorque(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, ContinuousUpdate:= ?BOOL?, Torque:= ?LREAL?, LimitVelocity:= ?LREAL?, BufferMode:= ?INT?, Options:= ?DWORD?, InTorque=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
Axis	軸情報	AXIS_REF	起動時	—	省略不可	軸を設定します。 使用する変数(AxisName.AxisRef.)については、下記を参照してください。 📖 1378ページ AxisName.AxisRef.(軸情報)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	実行指令	BOOL	起動時	—	FALSE	TRUEで本FBを実行します。
Continuous Update	連続更新	BOOL	起動時	—	FALSE	TRUEの間連続して目標位置を変更可能です。
Torque	目標トルク	LREAL	起動時/再起動可/連続更新可	-1000.0~1000.0[%]	0	使用するサーボモータの定格トルクに対する比率で設定します 詳細は、下記を参照してください。 📖 1845ページ 目標トルク(Torque)
LimitVelocity	制限速度	LREAL	起動時/再起動可/連続更新可	-2500000000.0~2500000000.0	0	方向を指定します。 詳細は、下記を参照してください。 📖 1846ページ 制限速度(LimitVelocity)
BufferMode	バッファモード	MC_BUFFER_MODE	起動時	mcAborting	mcAborting	バッファモードを設定します。 詳細は、下記を参照してください。 📖 1846ページ バッファモード(BufferMode)
Options	オプション	DWORD(HEX)	起動時	0固定	0	本FBでは設定できません。0に設定してください。0以外を設定した場合、Options範囲外(エラーコード: 1ABBH)となります。

■出力変数

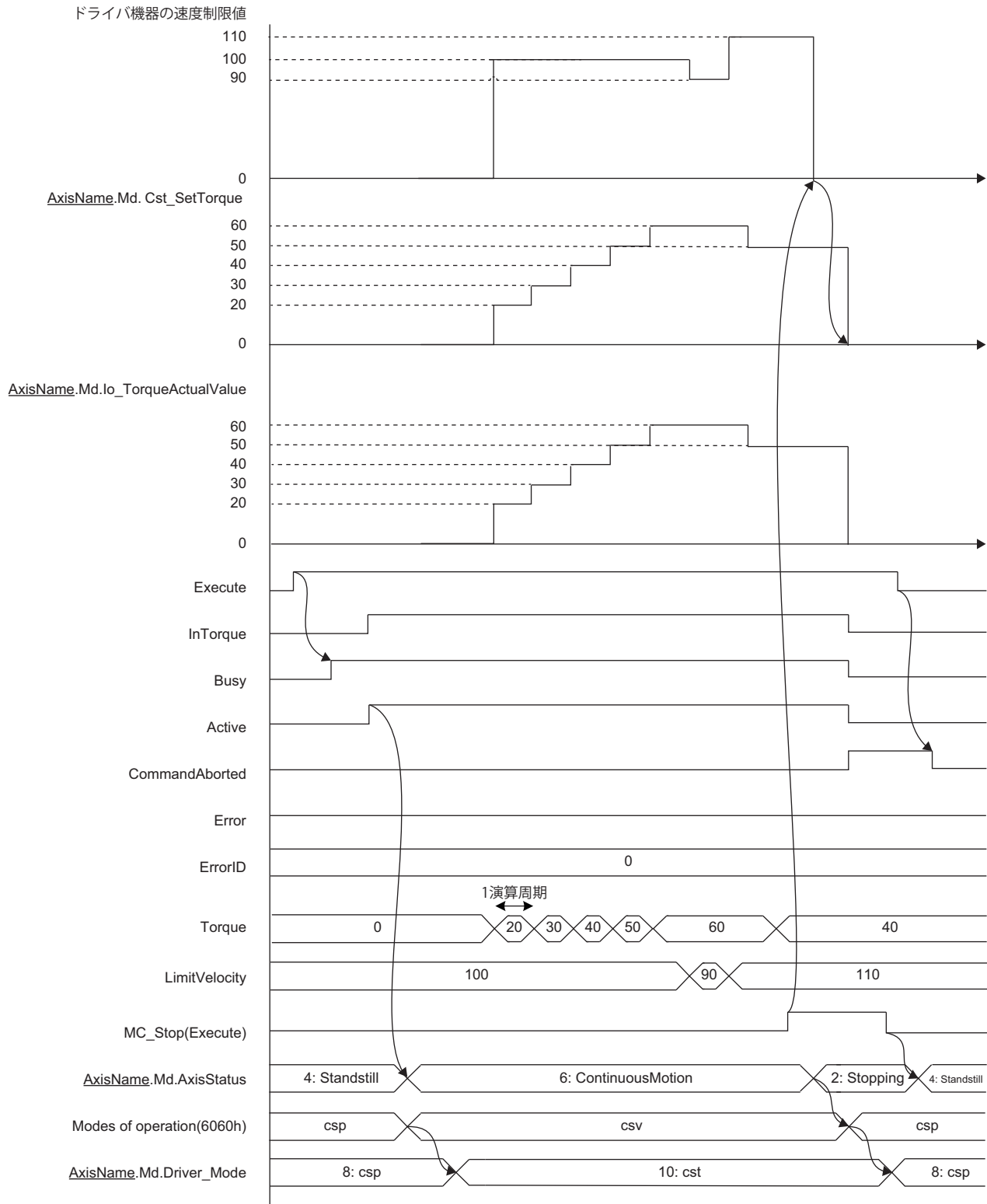
出力変数	名称	データ型	初期値	内容
InTorque	目標トルク到達	BOOL	FALSE	ドライブユニットの送信するCst_SetTorqueが目標トルクに到達したときに、TRUEになります。異常時または停止要因発生による軸停止後にFALSEとなります。
Busy	実行中	BOOL	FALSE	本FBが動作中のときに、TRUEになります。
Active	制御中	BOOL	FALSE	本FBが軸を制御中のときに、TRUEになります。
CommandAborted	実行中断	BOOL	FALSE	他のFBにより実行中断したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。エラーコードの詳細は、下記を参照してください。  使用するコントローラのユーザーズマニュアル

機能

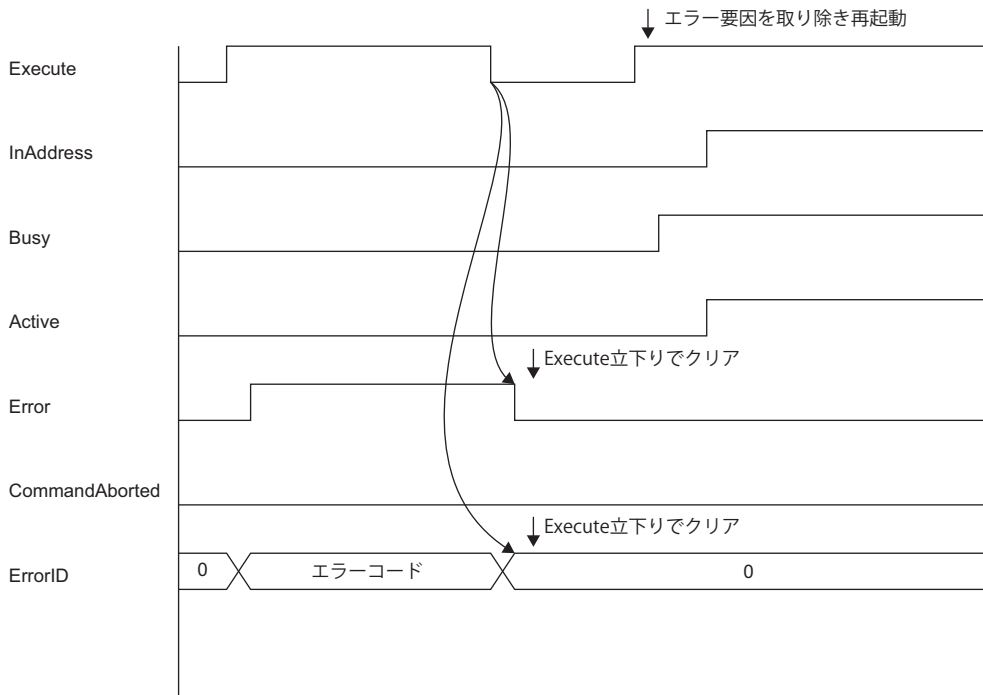
ドライバ側の制御モードをcstに切り換え、指令速度を送信します。本FBを停止するには、MC_Stop(強制停止)を使用してください。

■タイミングチャート

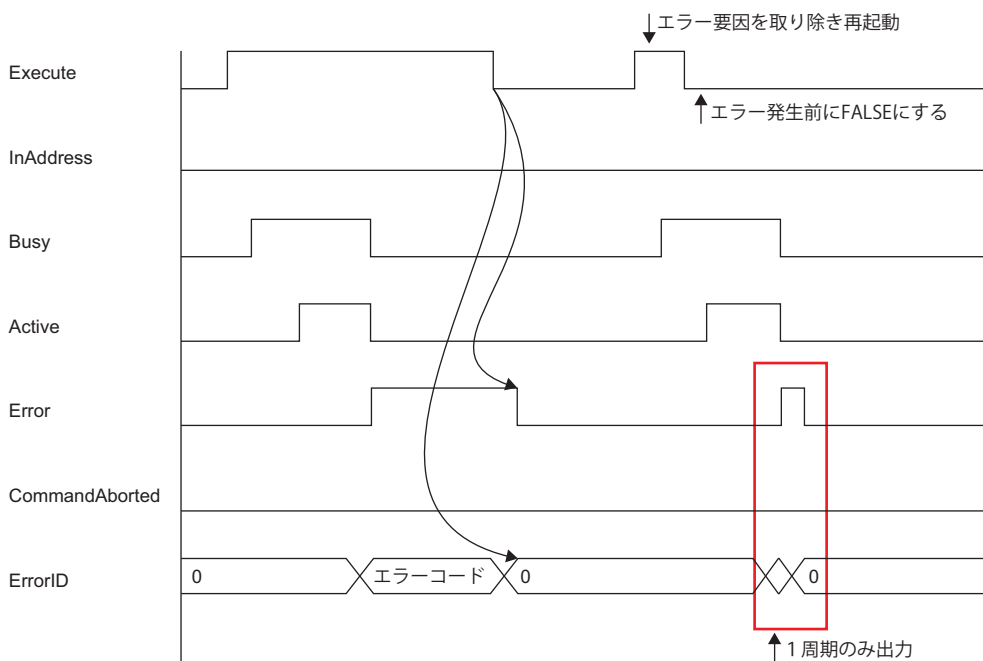
- ・ 正常完了の場合



・ 入出力変数異常の場合



・ 入力変数異常の場合



■目標トルク (Torque)

使用するサーボモータの定格トルクに対する比率を%単位で設定します。

始動時に範囲外の値を指定した場合、目標トルク範囲外(エラーコード: 1A96H)が発生します

制御中に範囲外の値を指定した場合、トルク設定範囲外警告(エラーコード: 0D16H)が発生します。

目標トルクの有効桁数は接続するドライブユニットのTarget Torqueの仕様により異なります。目標トルクに設定した値にTarget Torqueの有効桁数以下の値が含まれている場合、有効桁数以下の値は切り捨てられます。

指令トルクを正方向(アドレス増加方向)に出力する場合は正の値、負方向(アドレス減少方向)に出力する場合は負の値を設定してください。

安全のため目標トルクを反転する場合は一度目標トルクを0にしてください。

例

目標トルクを-10.0から10.0にする場合、下記の順番で変更します。

-10.0→0.0→10.0

■制限速度(LimitVelocity)

サイクリックトルクモード時の速度制限値を-2500000000.0~2500000000.0の範囲で設定します。範囲外を設定した場合、始動時は制限速度範囲外(エラーコード: 1A86H)、制御中は制限速度範囲外警告(エラーコード: 0D1BH)が発生します。

ドライバ速度単位変換後の指令速度が-2147483648~2147483647[0.01rpm]の範囲外とならないように設定してください。範囲外を設定した場合ドライバ速度単位変換後の指令速度範囲外(エラーコード: 1A90H)が発生します。

ドライバ速度単位変換後の指令速度は下記の式から求められます。

ドライバ速度単位変換後の指令速度=速度指令単位×(ドライバ単位変換分子÷ドライバ単位変換分母)

始動時に0を設定した場合、軸は動作しませんが対象軸の軸状態(AxisName.Md.AxisStatus)は、「6: 連続動作運転中(ContinuousMotion)」になります。

MC_CyclicTorque(トルク制御)を多重起動し、かつLimitVelocity(制限速度)に0が格納されている場合、前のモーション制御FBのモータ回転数を制限速度(LimitVelocity)として制御します。

制限速度(LimitVelocity)は、下記を満たしている場合に有効になります。

- 接続するドライブユニットのcst(サイクリックトルクモード)中の制限速度(LimitVelocity)に対応するオブジェクトをPDOマッピング割付する。
- vVelLimitValue(AxisName.PrConst.SlaveObject.vVelLimitValue)にマッピングしたオブジェクトデータを設定する。

制限速度(LimitVelocity)での速度制限を有効にする方法はドライブユニットによって設定方法が異なります。

- MR-J5(W)-Gを使用する場合、Velocity limit valueをPDOマッピング割付してください。
 - MR-J5(W)-G以外のドライブユニットを使用する場合、接続するドライブユニットのcst(サイクリックトルクモード)中の制限速度に対応するオブジェクトをPDOマッピング割付してください。さらにvVelLimitValue(AxisName.PrConst.SlaveObject.vVelLimitValue)にマッピングしたオブジェクトデータを設定してください。
- 制限速度(LimitVelocity)での速度制限が無効の場合、コントローラの速度指令は有効になりません。
- MR-J5(W)-Gを使用する場合、エンジニアリングツールからサーボパラメータ[Pr.PT67 速度制限]の設定値を設定してください。
 - MR-J5(W)-G以外のドライブユニットを使用する場合、接続するドライブユニット側の速度制限機能を使用してください。詳細は接続するドライブユニットのユーザーズマニュアルを参照してください。

■バッファモード(BufferMode)

多重起動(バッファモード)を行うための動作を設定します。

本FBで設定可能なバッファモードを下記に示します。

設定値	内容
0: Aborting(mcAborting)	制御中のFBを中断(キャンセル)して次のFBを即時実行します。

Point

多重起動(バッファモード)の詳細については、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

注意事項

- ・フォローアップにより指令現在位置，送り機械位置を更新します。
- ・本FBにより制御モードが切り替わるまでの時間は，使用するドライブユニットの仕様によって異なります。
- ・本FBを始動中に停止要因が発生した場合，即停止します。
- ・本FBの動作中に位置決め制御FBを起動しないでください。ドライバ制御モード(AxisName.Md.Driver_Mode)が「10: トルク制御(cst)」に切り換わったことを確認した後，位置決め制御FBを起動してください。
- ・目標トルクに設定したトルクが十分でない場合，機構や負荷によっては急激にトルクが抜ける場合があります。目標トルクを小さくする場合は，安全に十分注意を払って設定してください。
- ・軸が停止している場合，制御モードが切り換わってから制限速度(LimitVelocity)に値を格納してください。切り換わる前に制限速度(LimitVelocity)に値が格納されている場合，速度が急激に増加し危険な場合があります。
- ・cst中は，制限速度(LimitVelocity)を推奨します。制限速度(LimitVelocity)を使用しない場合，現在のトルクと目標トルクの差分が大きい場合に，機構や負荷のかかり方によっては，速度が急激に増加し危険な場合があります。
- ・cst中のトルク制限変更について，下記の変更をした場合の動作に注意してください。

変更内容	動作
再起動または連続更新で目標トルクをトルク制限値(正方向トルク制限値(<u>AxisName</u> .Md.TorqueLimit_Positive)，負方向トルク制限値(<u>AxisName</u> .Md.TorqueLimit_Negative))よりも大きな値に変更した	トルク制限値オーバ警告(エラーコード: 0D12H)が発生し，変更前の値で動作します。
トルク制御中に目標トルクよりも小さい正方向トルク制限値(<u>AxisName</u> .Cd.TorqueLimit_Positive)，負方向トルク制限値(<u>AxisName</u> .Cd.TorqueLimit_Negative)に変更した	目標トルクをトルク制限値に変更します。このとき，指令トルクは1演算周期で変更します。

- ・本FBの動作中にMC_TorqueControl(トルク制御)を多重起動して，押当て制御モードの切り換えを行うことはできません。制御モード切り換えを行った場合，制御モード切り換え不可警告(エラーコード: 0D31H)が発生し，現在の制御モードで動作します。
- ・押当て制御中に本FBを実行できません。切り換えを実施した場合，制御モード切り換え異常(エラーコード: 1A95H)が発生します。

プログラム例

本FBのプログラム例については，下記を参照してください。

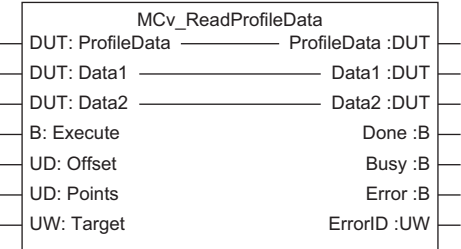
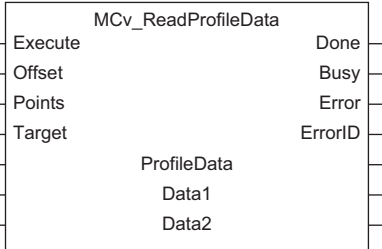
📖使用するコントローラのユーザーズマニュアル

43 一般ファンクションブロック

43.1 プロファイル読出

MCv_ReadProfileData

指定した演算プロファイルデータ(カムデータ)を、展開エリアまたはファイルから読み出します。

ラダー	FBD/LD	ST
		<pre>MCv_ReadProfileData(ProfileData:= ?PROFILE_DATA?, Data1:= ?TARGET_REF?, Data2:= ?TARGET_REF?, Execute:= ?BOOL?, Offset:= ?DWORD?, Points:= ?DWORD?, Target:= ?WORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
ProfileData	プロファイルデータ	PROFILE_DATA	起動時	—	省略不可	読み出す演算プロファイルを設定します。 コントローラ側でFBを使用する場合は ProfileData.ID.StartIOは設定不要です。 Data1.StartIOで設定した入出力番号を参照します。 演算プロファイルはPROFILE_DATA構造体で設定 します。PROFILE_DATA構造体については、下記 を参照してください。 📖 1424ページ PROFILE_DATA 読出し用データ構造体を定義した演算プロファイル 形式は、読出し操作ができます。
Data1	読出しデータ1	TARGET_REF	起動時	—	省略不可	読み出す演算プロファイル形式に応じた読出し用 データ構造体を設定します。 詳細は、下記を参照してください。 📖 1855ページ 読出しデータ1(Data1)/読出しデ ータ2(Data2)
Data2	読出しデータ2	TARGET_REF	起動時	—	省略不可	読み出す演算プロファイル形式に応じた読出し用 データ構造体を設定します。 詳細は、下記を参照してください。 📖 1855ページ 読出しデータ1(Data1)/読出しデ ータ2(Data2)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	起動	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMCv_ReadProfileData(プロファイル読出)を実行します。
Offset	オフセット	DWORD(UDINT)	起動時	0~分解能 (2~65535)	0	演算プロファイル先頭からのオフセットを設定します。 オフセット(Offset)が演算プロファイルの範囲を超えると、オフセット範囲外(エラーコード: 1B92H)となり、読出しは行いません。 オフセット(Offset)に「0」以外の値を設定する場合、読出しデータ数(Points)には「0」以外の値を設定してください。(読出しデータ数(Points)に「0」を設定した場合、オフセット・読出し/書込みデータ数不正(エラーコード: 1BADH)となります。) *: オフセット(Offset)と読出しデータ数(Points)の設定値により読み出す動作が異なります。詳細は、下記を参照してください。 📖 1850ページ オフセット(Offset)と読出しデータ数(Points)の設定
Points	読出しデータ数	DWORD(UDINT)	起動時	0~4294967295	0	読み出すデータ点数を設定します。 読出しデータ数(Points)が演算プロファイルの要素数を超える場合、演算プロファイルの要素数の範囲内で読出しを行います。 演算プロファイルの補間方法指定(Interpolate)が「1: 区間ごとに指定」、または「2: スプライン補間」の場合、「0」を設定してください。(「0」以外を設定した場合、オフセット・読出し/書込みデータ数不正(エラーコード: 1BADH)となります。)
Target	読出し先	WORD(UINT)	起動時	0, 1, 2	0	演算プロファイルの読出し先を設定します。 ・0: 展開エリア ・1: ファイル ・2: 展開エリア(モーションサービス処理) 詳細は、下記を参照してください。 📖 1857ページ 読出し先(Target)

■出力変数

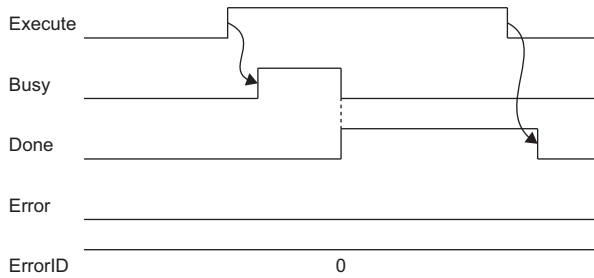
出力変数	名称	データ型	初期値	内容
Done	完了	BOOL	FALSE	制御が完了したときに、TRUEになります。 動作完了時に起動(Execute)の状態により下記となります。 ■起動(Execute)がTRUEの場合 起動(Execute)をFALSEにするまでTRUEのままになります。 ■起動(Execute)がFALSEの場合 1周期だけTRUEになります。
Busy	実行中	BOOL	FALSE	MCv_ReadProfileData(プロファイル読出)を実行したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 📖 使用するコントローラのユーザーズマニュアル

機能

- MCv_ReadProfileData(プロファイル読出)を使用して演算プロファイルの読出し操作を行います。
- プロファイルデータ(ProfileData), および読出し先(Target)で設定したファイル, または展開エリアの演算プロファイルに対し, オフセット(Offset)で設定したデータから読出しデータ数(Points)で設定した点数分を読み出します。
- 演算プロファイル全体を読出すには, オフセット(Offset)と読出しデータ数(Points)の両方に「0」を設定します。
- 読出したデータは, 読出しデータ1(Data1), 読出しデータ2(Data2)で設定した変数に格納します。

■タイミングチャート

- 正常完了の場合



- 異常完了の場合

異常完了時のタイミングチャートについては, 下記を参照してください。

📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■オフセット(Offset)と読出しデータ数(Points)の設定

オフセット(Offset)と読出しデータ数(Points)に設定した値により下記の動作となります。

オフセット(Offset)	読出しデータ数(Points)	演算プロファイルの要素数との大小関係	動作
0	0	—	演算プロファイルの全要素を読み出します。
	0以外	オフセット(Offset) + 読出しデータ数(Points) ≤ 演算プロファイルの要素数	先頭から読出しデータ数(Points)で設定した点数分演算プロファイルの要素を読み出します。
	0以外	オフセット(Offset) + 読出しデータ数(Points) > 演算プロファイルの要素数	演算プロファイルの全要素を読み出します。
0以外	0	オフセット(Offset) ≤ 演算プロファイルの要素数	オフセット・読出し/書き込みデータ数不正(エラーコード: 1BADH)が発生し, 読出しされません。
	0以外	オフセット(Offset) + 読出しデータ数(Points) ≤ 演算プロファイルの要素数	オフセット(Offset)から読出しデータ数(Points)で指定した点数分演算プロファイルの要素を読み出します。
	0以外	オフセット(Offset) + 読出しデータ数(Points) > 演算プロファイルの要素数	オフセット(Offset)から演算プロファイルの最終要素までを読み出します。
	—	オフセット(Offset) > 演算プロファイルの要素数	オフセット範囲外(エラーコード: 1B92H)が発生し, 読出しされません。

オフセット(Offset), 読出しデータ数(Points)に設定する値の内容は, 演算プロファイル形式により異なります。

・ 演算プロファイル形式が「カムデータ」の場合

補間方法指定 (Interpolate)	内容																																											
0: 直線補間	読み出す座標の先頭をオフセット(Offset), 座標数を読み出しデータ数(Points)で設定します。 読み出しデータ1(Data1)に格納されるデータは、補間方法指定(Interpolate)が「0: 直線補間」のPROFILE_CAM_DATA型のデータとなります。 読み出しデータ2(Data2)に格納されるデータは、設定した座標データ(入力値(1サイクル現在値)と出力値(ストローク)のペア)をLREAL型の二次元配列で定義したデータとなります。 <例> オフセット(Offset)に「2」、読み出しデータ数(Points)に「3」を設定したときの読み出しデータ2(Data2)の読み出し結果 (読み出しデータ2(Data2)に設定する変数をデータ型「LREAL[1..8,1..2]」のラベル「CamData2」とした場合) <変数(ラベル/デバイス)> LREAL型の二次元配列のラベル (CamData2[1..8,1..2]) <table><thead><tr><th></th><th>CamData[n][1]</th><th>CamData[n][2]</th></tr></thead><tbody><tr><td>n=1</td><td>2点目の入力値</td><td>2点目の出力値</td></tr><tr><td>n=2</td><td>3点目の入力値</td><td>3点目の出力値</td></tr><tr><td>n=3</td><td>4点目の入力値</td><td>4点目の出力値</td></tr><tr><td>n=4</td><td>(更新しない)</td><td>(更新しない)</td></tr><tr><td>n=5</td><td>(更新しない)</td><td>(更新しない)</td></tr><tr><td>n=6</td><td>(更新しない)</td><td>(更新しない)</td></tr><tr><td>n=7</td><td>(更新しない)</td><td>(更新しない)</td></tr><tr><td>n=8</td><td>(更新しない)</td><td>(更新しない)</td></tr></tbody></table> オフセット (Offset) 読み出しデータ数 (Points) <展開エリア/ファイル>*1 <table><thead><tr><th>0点目の入力値</th><th>0点目の出力値</th></tr></thead><tbody><tr><td>1点目の入力値</td><td>1点目の出力値</td></tr><tr><td>2点目の入力値</td><td>2点目の出力値</td></tr><tr><td>3点目の入力値</td><td>3点目の出力値</td></tr><tr><td>4点目の入力値</td><td>4点目の出力値</td></tr><tr><td>5点目の入力値</td><td>5点目の出力値</td></tr><tr><td>6点目の入力値</td><td>6点目の出力値</td></tr><tr><td>7点目の入力値</td><td>7点目の出力値</td></tr></tbody></table> *1: 読み出し先(Target)によります。		CamData[n][1]	CamData[n][2]	n=1	2点目の入力値	2点目の出力値	n=2	3点目の入力値	3点目の出力値	n=3	4点目の入力値	4点目の出力値	n=4	(更新しない)	(更新しない)	n=5	(更新しない)	(更新しない)	n=6	(更新しない)	(更新しない)	n=7	(更新しない)	(更新しない)	n=8	(更新しない)	(更新しない)	0点目の入力値	0点目の出力値	1点目の入力値	1点目の出力値	2点目の入力値	2点目の出力値	3点目の入力値	3点目の出力値	4点目の入力値	4点目の出力値	5点目の入力値	5点目の出力値	6点目の入力値	6点目の出力値	7点目の入力値	7点目の出力値
	CamData[n][1]	CamData[n][2]																																										
n=1	2点目の入力値	2点目の出力値																																										
n=2	3点目の入力値	3点目の出力値																																										
n=3	4点目の入力値	4点目の出力値																																										
n=4	(更新しない)	(更新しない)																																										
n=5	(更新しない)	(更新しない)																																										
n=6	(更新しない)	(更新しない)																																										
n=7	(更新しない)	(更新しない)																																										
n=8	(更新しない)	(更新しない)																																										
0点目の入力値	0点目の出力値																																											
1点目の入力値	1点目の出力値																																											
2点目の入力値	2点目の出力値																																											
3点目の入力値	3点目の出力値																																											
4点目の入力値	4点目の出力値																																											
5点目の入力値	5点目の出力値																																											
6点目の入力値	6点目の出力値																																											
7点目の入力値	7点目の出力値																																											
1: 区間ごとに指定	読み出し先(Target)により読み出し可能なデータが異なります。 ■読み出し先(Target)が「0: 展開エリア」、「2: 展開エリア(モーションサービス処理)」の場合 読み出す区間の先頭をオフセット(Offset), 区間数(NumberOfSections)を読み出しデータ数(Points)で設定します。 読み出しデータ1(Data1)に格納されるデータは、補間方法指定(Interpolate)が「0: 直線補間」のPROFILE_CAM_DATA型のデータとなります。 読み出しデータ2(Data2)に格納されるデータは、対象のカムデータをカム分解能の点数で等分割して定義したデータ(1サイクル現在値とストロークのペア)をLREAL型の二次元配列で定義したデータとなります。 <例> カム分解能が「256」の場合、オフセット(Offset)に「2」、読み出しデータ数(Points)に「3」を設定したときの読み出しデータ2(Data2)の読み出し結果 (読み出しデータ2(Data2)に設定する変数をデータ型「LREAL[1..8,1..2]」のラベル「CamData2」とした場合) <変数(ラベル/デバイス)> LREAL型の二次元配列のラベル (CamData2[1..8,1..2]) <table><thead><tr><th></th><th>CamData[n][1]</th><th>CamData[n][2]</th></tr></thead><tbody><tr><td>n=1</td><td>2点目の1サイクル現在値</td><td>2点目のストローク</td></tr><tr><td>n=2</td><td>3点目の1サイクル現在値</td><td>3点目のストローク</td></tr><tr><td>n=3</td><td>4点目の1サイクル現在値</td><td>4点目のストローク</td></tr><tr><td>n=4</td><td>(更新しない)</td><td>(更新しない)</td></tr><tr><td>n=5</td><td>(更新しない)</td><td>(更新しない)</td></tr><tr><td>n=6</td><td>(更新しない)</td><td>(更新しない)</td></tr><tr><td>n=7</td><td>(更新しない)</td><td>(更新しない)</td></tr><tr><td>n=8</td><td>(更新しない)</td><td>(更新しない)</td></tr></tbody></table> オフセット (Offset) 読み出しデータ数 (Points) <展開エリア> <table><thead><tr><th>開始点</th><th>初期ストローク</th></tr></thead><tbody><tr><td>1点目の1サイクル現在値</td><td>1点目のストローク</td></tr><tr><td>2点目の1サイクル現在値</td><td>2点目のストローク</td></tr><tr><td>3点目の1サイクル現在値</td><td>3点目のストローク</td></tr><tr><td>4点目の1サイクル現在値</td><td>4点目のストローク</td></tr><tr><td>5点目の1サイクル現在値</td><td>5点目のストローク</td></tr><tr><td>⋮</td><td>⋮</td></tr><tr><td>256点目の1サイクル現在値</td><td>256点目のストローク</td></tr></tbody></table> ↑ カムデータをカム分解能の点数で等分割したデータ		CamData[n][1]	CamData[n][2]	n=1	2点目の1サイクル現在値	2点目のストローク	n=2	3点目の1サイクル現在値	3点目のストローク	n=3	4点目の1サイクル現在値	4点目のストローク	n=4	(更新しない)	(更新しない)	n=5	(更新しない)	(更新しない)	n=6	(更新しない)	(更新しない)	n=7	(更新しない)	(更新しない)	n=8	(更新しない)	(更新しない)	開始点	初期ストローク	1点目の1サイクル現在値	1点目のストローク	2点目の1サイクル現在値	2点目のストローク	3点目の1サイクル現在値	3点目のストローク	4点目の1サイクル現在値	4点目のストローク	5点目の1サイクル現在値	5点目のストローク	⋮	⋮	256点目の1サイクル現在値	256点目のストローク
	CamData[n][1]	CamData[n][2]																																										
n=1	2点目の1サイクル現在値	2点目のストローク																																										
n=2	3点目の1サイクル現在値	3点目のストローク																																										
n=3	4点目の1サイクル現在値	4点目のストローク																																										
n=4	(更新しない)	(更新しない)																																										
n=5	(更新しない)	(更新しない)																																										
n=6	(更新しない)	(更新しない)																																										
n=7	(更新しない)	(更新しない)																																										
n=8	(更新しない)	(更新しない)																																										
開始点	初期ストローク																																											
1点目の1サイクル現在値	1点目のストローク																																											
2点目の1サイクル現在値	2点目のストローク																																											
3点目の1サイクル現在値	3点目のストローク																																											
4点目の1サイクル現在値	4点目のストローク																																											
5点目の1サイクル現在値	5点目のストローク																																											
⋮	⋮																																											
256点目の1サイクル現在値	256点目のストローク																																											
2: スプライン補間	■読み出し先(Target)が「1: ファイル」の場合 オフセット(Offset), および読み出しデータ数(Points)に「0」を設定して、全要素を読み出します。(「0」以外を設定した場合、オフセット・読み出し/書き込みデータ数不正(エラーコード: 1BADH)となります。) 読み出しデータ1(Data1)に格納されるデータは、補間方法指定(Interpolate)が「1: 区間ごとに指定」、または「2: スプライン補間」のPROFILE_CAM_DATA型のデータとなります。 読み出しデータ2(Data2)に格納されるデータは、PROFILE_CAM_ELEMENT型の配列で定義した区間データとなります。																																											

・演算プロファイル形式が「ロータリーカッター」の場合

読み出し先 (Target)	内容																																											
0: 展開エリア	読み出す区間の先頭をオフセット(Offset)、区間数(NumberOfSections)を読み出しデータ数(Points)で設定します。読み出しデータ1(Data1)に格納されるデータは、演算プロファイル形式が「カムデータ」、補間方法指定(Interpolate)が「0: 直線補間」のPROFILE_CAM_DATA型のデータとなります。読み出しデータ2(Data2)に格納されるデータは、対象のカムデータをカム分解能の点数で等分割して定義したデータ(1サイクル現在値とストロークのペア)をLREAL型の二次元配列で定義したデータとなります。 <例> オフセット(Offset)に「2」、読み出しデータ数(Points)に「3」を設定したときの読み出しデータ2(Data2)の読み出し結果(読み出しデータ2(Data2)に設定する変数をデータ型「LREAL[1..8,1..2]」のラベル「CamData2」とした場合) <変数(ラベル/デバイス)> LREAL型の二次元配列のラベル (CamData2[1..8,1..2]) <table><thead><tr><th></th><th>CamData[n][1]</th><th>CamData[n][2]</th></tr></thead><tbody><tr><td>n=1</td><td>2点目の1サイクル現在値</td><td>2点目のストローク</td></tr><tr><td>n=2</td><td>3点目の1サイクル現在値</td><td>3点目のストローク</td></tr><tr><td>n=3</td><td>4点目の1サイクル現在値</td><td>4点目のストローク</td></tr><tr><td>n=4</td><td>(更新しない)</td><td>(更新しない)</td></tr><tr><td>n=5</td><td>(更新しない)</td><td>(更新しない)</td></tr><tr><td>n=6</td><td>(更新しない)</td><td>(更新しない)</td></tr><tr><td>n=7</td><td>(更新しない)</td><td>(更新しない)</td></tr><tr><td>n=8</td><td>(更新しない)</td><td>(更新しない)</td></tr></tbody></table> オフセット (Offset) 読み出し データ数 (Points) <展開エリア> <table><thead><tr><th>0</th><th>0</th></tr></thead><tbody><tr><td>1点目の1サイクル現在値</td><td>1点目のストローク</td></tr><tr><td>2点目の1サイクル現在値</td><td>2点目のストローク</td></tr><tr><td>3点目の1サイクル現在値</td><td>3点目のストローク</td></tr><tr><td>4点目の1サイクル現在値</td><td>4点目のストローク</td></tr><tr><td>5点目の1サイクル現在値</td><td>5点目のストローク</td></tr><tr><td>⋮</td><td>⋮</td></tr><tr><td>255点目の1サイクル現在値</td><td>255点目のストローク</td></tr></tbody></table> ↑ カムデータをカム分解能の点数で等分割したデータ		CamData[n][1]	CamData[n][2]	n=1	2点目の1サイクル現在値	2点目のストローク	n=2	3点目の1サイクル現在値	3点目のストローク	n=3	4点目の1サイクル現在値	4点目のストローク	n=4	(更新しない)	(更新しない)	n=5	(更新しない)	(更新しない)	n=6	(更新しない)	(更新しない)	n=7	(更新しない)	(更新しない)	n=8	(更新しない)	(更新しない)	0	0	1点目の1サイクル現在値	1点目のストローク	2点目の1サイクル現在値	2点目のストローク	3点目の1サイクル現在値	3点目のストローク	4点目の1サイクル現在値	4点目のストローク	5点目の1サイクル現在値	5点目のストローク	⋮	⋮	255点目の1サイクル現在値	255点目のストローク
		CamData[n][1]	CamData[n][2]																																									
n=1		2点目の1サイクル現在値	2点目のストローク																																									
n=2	3点目の1サイクル現在値	3点目のストローク																																										
n=3	4点目の1サイクル現在値	4点目のストローク																																										
n=4	(更新しない)	(更新しない)																																										
n=5	(更新しない)	(更新しない)																																										
n=6	(更新しない)	(更新しない)																																										
n=7	(更新しない)	(更新しない)																																										
n=8	(更新しない)	(更新しない)																																										
0	0																																											
1点目の1サイクル現在値	1点目のストローク																																											
2点目の1サイクル現在値	2点目のストローク																																											
3点目の1サイクル現在値	3点目のストローク																																											
4点目の1サイクル現在値	4点目のストローク																																											
5点目の1サイクル現在値	5点目のストローク																																											
⋮	⋮																																											
255点目の1サイクル現在値	255点目のストローク																																											
1: ファイル	オフセット(Offset)、および読み出しデータ数(Points)の設定は無視します。 読み出しデータ1(Data1)に格納されるデータは、演算プロファイル形式が「ロータリーカッター」のPROFILE_ROTARY_CUTTER型のデータとなります。読み出しデータ2(Data2)に格納されるデータはありません。																																											

・演算プロファイル形式が「多軸位置決めデータ」の場合

読み出し先(Target)	内容																																																																																																										
0: 展開エリア	位置決めデータを読み出す区間の先頭をオフセット(Offset)、 区間数(NumberOfSections)を読み出しデータ数(Points)で指定してください。																																																																																																										
2: 展開エリア(モーションサービス処理)	Data1(TARGET_REF型)に指定したPROFILE_POSITIONING_DATA型の変数に、データを読み出します。 Data2(TARGET_REF型)にPROFILE_POSITIONING_DATA_ELEMENT型配列の変数を指定する場合は、指定した変数に位置決めデータを読み出します。 <例> オフセット(Offset)に「2」、読み出しデータ数(Points)に「3」を設定したときの読み出しデータ2(Data2)の読み出し結果(読み出しデータ2(Data2)に設定する変数をデータ型「PROFILE_POSITIONING_DATA_ELEMENT[1..8]」のラベル「PositioningData2」とした場合) <変数(ラベル/デバイス)> PROFILE_POSITIONING_DATA_ELEMENT型のラベル (PositioningData2[1..8]) <table><tr><td>n=1</td><td>3点目のDataNo(3)</td><td>3点目のOperationPattern</td><td>...</td></tr><tr><td>n=2</td><td>4点目のDataNo(4)</td><td>4点目のOperationPattern</td><td>...</td></tr><tr><td>n=3</td><td>5点目のDataNo(5)</td><td>5点目のOperationPattern</td><td>...</td></tr><tr><td>n=4</td><td>(更新しない)</td><td>(更新しない)</td><td>...</td></tr><tr><td>n=5</td><td>(更新しない)</td><td>(更新しない)</td><td>...</td></tr><tr><td>n=6</td><td>(更新しない)</td><td>(更新しない)</td><td>...</td></tr><tr><td>n=7</td><td>(更新しない)</td><td>(更新しない)</td><td>...</td></tr><tr><td>n=8</td><td>(更新しない)</td><td>(更新しない)</td><td>...</td></tr></table> オフセット(Offset) 読み出しデータ数(Points) <展開エリア> <table><tr><td>1点目のDataNo(1)</td><td>1点目のOperationPattern</td><td>...</td></tr><tr><td>2点目のDataNo(2)</td><td>2点目のOperationPattern</td><td>...</td></tr><tr><td>3点目のDataNo(3)</td><td>3点目のOperationPattern</td><td>...</td></tr><tr><td>4点目のDataNo(4)</td><td>4点目のOperationPattern</td><td>...</td></tr><tr><td>5点目のDataNo(5)</td><td>5点目のOperationPattern</td><td>...</td></tr><tr><td>6点目のDataNo(6)</td><td>6点目のOperationPattern</td><td>...</td></tr><tr><td>...</td><td>...</td><td>...</td></tr></table> 読み出す区間に、展開や書き込みで未設定の位置決めデータがある場合、該当のDataNoにあたる区間データに初期値を格納します。(DataNoは0となります。) <例> 上記の例で展開エリアの4点目(位置決めデータNo.4)が展開や書き込みで未設定の場合 <変数(ラベル/デバイス)> PROFILE_POSITIONING_DATA_ELEMENT型のラベル (PositioningData2[1..8]) <table><tr><td>n=1</td><td>3点目のDataNo(3)</td><td>3点目のOperationPattern</td><td>...</td></tr><tr><td>n=2</td><td>0</td><td>0</td><td>...</td></tr><tr><td>n=3</td><td>5点目のDataNo(5)</td><td>5点目のOperationPattern</td><td>...</td></tr><tr><td>n=4</td><td>(更新しない)</td><td>(更新しない)</td><td>...</td></tr><tr><td>n=5</td><td>(更新しない)</td><td>(更新しない)</td><td>...</td></tr><tr><td>n=6</td><td>(更新しない)</td><td>(更新しない)</td><td>...</td></tr><tr><td>n=7</td><td>(更新しない)</td><td>(更新しない)</td><td>...</td></tr><tr><td>n=8</td><td>(更新しない)</td><td>(更新しない)</td><td>...</td></tr></table> オフセット(Offset) 読み出しデータ数(Points) <展開エリア> <table><tr><td>1点目のDataNo(1)</td><td>1点目のOperationPattern</td><td>...</td></tr><tr><td>2点目のDataNo(2)</td><td>2点目のOperationPattern</td><td>...</td></tr><tr><td>3点目のDataNo(3)</td><td>3点目のOperationPattern</td><td>...</td></tr><tr><td>4点目のDataNo</td><td>4点目のOperationPattern</td><td>...</td></tr><tr><td>5点目のDataNo(5)</td><td>5点目のOperationPattern</td><td>...</td></tr><tr><td>6点目のDataNo(6)</td><td>6点目のOperationPattern</td><td>...</td></tr><tr><td>...</td><td>...</td><td>...</td></tr></table> Data2(TARGET_REF型)にPROFILE_POSITIONING_DATA_EXTENDED型の変数を指定する場合 変数のメンバConditionSignal、SkipSignalおよびPositioningDataにWSTRING型で指定した変数に、条件信号、スキップ信号、および位置決めデータをそれぞれ格納します。 条件信号、スキップ信号は、PROFILE_SIGNAL_SELECT_ELEMENT型配列のデータを読み出します。位置決めデータは、PROFILE_POSITIONING_DATA_ELEMENT型配列のデータを読み出します。(Data2(TARGET_REF型)にPROFILE_POSITIONING_DATA_ELEMENT型の配列を指定した場合と同様です。)	n=1	3点目のDataNo(3)	3点目のOperationPattern	...	n=2	4点目のDataNo(4)	4点目のOperationPattern	...	n=3	5点目のDataNo(5)	5点目のOperationPattern	...	n=4	(更新しない)	(更新しない)	...	n=5	(更新しない)	(更新しない)	...	n=6	(更新しない)	(更新しない)	...	n=7	(更新しない)	(更新しない)	...	n=8	(更新しない)	(更新しない)	...	1点目のDataNo(1)	1点目のOperationPattern	...	2点目のDataNo(2)	2点目のOperationPattern	...	3点目のDataNo(3)	3点目のOperationPattern	...	4点目のDataNo(4)	4点目のOperationPattern	...	5点目のDataNo(5)	5点目のOperationPattern	...	6点目のDataNo(6)	6点目のOperationPattern	...	...	...	...	n=1	3点目のDataNo(3)	3点目のOperationPattern	...	n=2	0	0	...	n=3	5点目のDataNo(5)	5点目のOperationPattern	...	n=4	(更新しない)	(更新しない)	...	n=5	(更新しない)	(更新しない)	...	n=6	(更新しない)	(更新しない)	...	n=7	(更新しない)	(更新しない)	...	n=8	(更新しない)	(更新しない)	...	1点目のDataNo(1)	1点目のOperationPattern	...	2点目のDataNo(2)	2点目のOperationPattern	...	3点目のDataNo(3)	3点目のOperationPattern	...	4点目のDataNo	4点目のOperationPattern	...	5点目のDataNo(5)	5点目のOperationPattern	...	6点目のDataNo(6)	6点目のOperationPattern	...	...	...	...
n=1	3点目のDataNo(3)	3点目のOperationPattern	...																																																																																																								
n=2	4点目のDataNo(4)	4点目のOperationPattern	...																																																																																																								
n=3	5点目のDataNo(5)	5点目のOperationPattern	...																																																																																																								
n=4	(更新しない)	(更新しない)	...																																																																																																								
n=5	(更新しない)	(更新しない)	...																																																																																																								
n=6	(更新しない)	(更新しない)	...																																																																																																								
n=7	(更新しない)	(更新しない)	...																																																																																																								
n=8	(更新しない)	(更新しない)	...																																																																																																								
1点目のDataNo(1)	1点目のOperationPattern	...																																																																																																									
2点目のDataNo(2)	2点目のOperationPattern	...																																																																																																									
3点目のDataNo(3)	3点目のOperationPattern	...																																																																																																									
4点目のDataNo(4)	4点目のOperationPattern	...																																																																																																									
5点目のDataNo(5)	5点目のOperationPattern	...																																																																																																									
6点目のDataNo(6)	6点目のOperationPattern	...																																																																																																									
...	...	...																																																																																																									
n=1	3点目のDataNo(3)	3点目のOperationPattern	...																																																																																																								
n=2	0	0	...																																																																																																								
n=3	5点目のDataNo(5)	5点目のOperationPattern	...																																																																																																								
n=4	(更新しない)	(更新しない)	...																																																																																																								
n=5	(更新しない)	(更新しない)	...																																																																																																								
n=6	(更新しない)	(更新しない)	...																																																																																																								
n=7	(更新しない)	(更新しない)	...																																																																																																								
n=8	(更新しない)	(更新しない)	...																																																																																																								
1点目のDataNo(1)	1点目のOperationPattern	...																																																																																																									
2点目のDataNo(2)	2点目のOperationPattern	...																																																																																																									
3点目のDataNo(3)	3点目のOperationPattern	...																																																																																																									
4点目のDataNo	4点目のOperationPattern	...																																																																																																									
5点目のDataNo(5)	5点目のOperationPattern	...																																																																																																									
6点目のDataNo(6)	6点目のOperationPattern	...																																																																																																									
...	...	...																																																																																																									

読み出し先(Target)	内容																																																																								
1: ファイル	オフセット(Offset), および読み出しデータ数(Points)に0を指定して, 全要素を読み出してください。 (0以外を設定した場合, オフセット・読み出し/書き込みデータ数不正(エラーコード: 1BADH)が発生します。) Data1(TARGET_REF型)に指定したPROFILE_POSITIONING_DATA型の変数に, データを読み出します。 Data2(TARGET_REF型)にPROFILE_POSITIONING_DATA_ELEMENT型配列の変数を指定する場合は, 指定した変数に位置決めデータを読み出します。 ファイルにDataNoを空けて位置決めデータを設定した場合, 空きのDataNoにあたる区間データに初期値を格納します。(DataNoは0となります。) <例> オフセット(Offset)に「0」, 読み出しデータ数(Points)に「0」を設定したときの読み出しデータ2(Data2)の読み出し結果 (読み出しデータ2(Data2)に設定する変数をデータ型「PROFILE_POSITIONING_DATA_ELEMENT[1..100]」のラベル「PositioningData2」とした場合) <変数(ラベル/デバイス)> PROFILE_POSITIONING_DATA_ELEMENT型のラベル (PositioningData2[1..100]) <table><tr><td>n=1</td><td>0</td><td>0</td><td>...</td></tr><tr><td>n=2</td><td>1点目のDataNo(2)</td><td>1点目のOperationPattern</td><td>...</td></tr><tr><td>n=3</td><td>2点目のDataNo(3)</td><td>2点目のOperationPattern</td><td>...</td></tr><tr><td>n=4</td><td>0</td><td>0</td><td>...</td></tr><tr><td>n=5</td><td>3点目のDataNo(5)</td><td>3点目のOperationPattern</td><td>...</td></tr><tr><td></td><td>...</td><td>...</td><td>...</td></tr><tr><td>n=100</td><td>4点目のDataNo(100)</td><td>4点目のOperationPattern</td><td>...</td></tr></table> <ファイル> <table><tr><td>1点目のDataNo(2)</td><td>1点目のOperationPattern</td><td>...</td></tr><tr><td>2点目のDataNo(3)</td><td>2点目のOperationPattern</td><td>...</td></tr><tr><td>3点目のDataNo(5)</td><td>3点目のOperationPattern</td><td>...</td></tr><tr><td>4点目のDataNo(100)</td><td>4点目のOperationPattern</td><td>...</td></tr></table> Data2(TARGET_REF型)にPROFILE_POSITIONING_DATA_EXTENDED型の変数を指定する場合 変数のメンバConditionSignal, SkipSignalおよびPositioningDataにWSTRING型で指定した変数に, 条件信号, スキップ信号, および位置決めデータをそれぞれ格納します。 条件信号, スキップ信号は, PROFILE_SIGNAL_SELECT_ELEMENT型配列のデータを読み出します。 位置決めデータは, PROFILE_POSITIONING_DATA_ELEMENT型配列のデータを読み出します。(Data2(TARGET_REF型)にPROFILE_POSITIONING_DATA_ELEMENT型配列の変数を指定した場合と同様です。)	n=1	0	0	...	n=2	1点目のDataNo(2)	1点目のOperationPattern	...	n=3	2点目のDataNo(3)	2点目のOperationPattern	...	n=4	0	0	...	n=5	3点目のDataNo(5)	3点目のOperationPattern	...		...	...	...	n=100	4点目のDataNo(100)	4点目のOperationPattern	...	1点目のDataNo(2)	1点目のOperationPattern	...	2点目のDataNo(3)	2点目のOperationPattern	...	3点目のDataNo(5)	3点目のOperationPattern	...	4点目のDataNo(100)	4点目のOperationPattern	...																																
n=1	0	0	...																																																																						
n=2	1点目のDataNo(2)	1点目のOperationPattern	...																																																																						
n=3	2点目のDataNo(3)	2点目のOperationPattern	...																																																																						
n=4	0	0	...																																																																						
n=5	3点目のDataNo(5)	3点目のOperationPattern	...																																																																						
	...	...	...																																																																						
n=100	4点目のDataNo(100)	4点目のOperationPattern	...																																																																						
1点目のDataNo(2)	1点目のOperationPattern	...																																																																							
2点目のDataNo(3)	2点目のOperationPattern	...																																																																							
3点目のDataNo(5)	3点目のOperationPattern	...																																																																							
4点目のDataNo(100)	4点目のOperationPattern	...																																																																							
・ 演算プロファイル形式が「デジタルカムスイッチデータ」の場合																																																																									
読み出し先(Target)	内容																																																																								
0: 展開エリア	読み出す区間の先頭をオフセット(Offset), 区間数(NumberOfSections)を読み出しデータ数(Points)で設定します。																																																																								
2: 展開エリア(モーションサービス処理)	読み出しデータ1(Data1)に格納されるデータは, PROFILE_CAMSWITCH_DATA型のデータとなります。 読み出しデータ2(Data2)に格納されるデータは, 指定したスイッチ設定のデータ(FirstOnPosition, LastOnPosition, Direction, CamSwitchMode, Duration)をPROFILE_CAMSWITCH_ELEMENT型の配列で定義したデータとなります。 <例> オフセット(Offset)に「2」, 読み出しデータ数(Points)に「3」を設定したときの読み出しデータ2(Data2)の読み出し結果 (読み出しデータ2(Data2)に設定する変数をデータ型「PROFILE_CAMSWITCH_ELEMENT[1..8]」のラベル「DigitalCamSwitch Data2」とした場合) <変数(ラベル/デバイス)> PROFILE_CAMSWITCH_ELEMENT型のラベル (PositioningData2[1..8]) <table><tr><td>n=1</td><td>3点目のFirstOnPosition</td><td>3点目のLastOnPosition</td><td>...</td><td>3点目のDuration</td></tr><tr><td>n=2</td><td>4点目のFirstOnPosition</td><td>4点目のLastOnPosition</td><td>...</td><td>4点目のDuration</td></tr><tr><td>n=3</td><td>5点目のFirstOnPosition</td><td>5点目のLastOnPosition</td><td>...</td><td>5点目のDuration</td></tr><tr><td>n=4</td><td>(更新しない)</td><td>(更新しない)</td><td>...</td><td>(更新しない)</td></tr><tr><td>n=5</td><td>(更新しない)</td><td>(更新しない)</td><td>...</td><td>(更新しない)</td></tr><tr><td>n=6</td><td>(更新しない)</td><td>(更新しない)</td><td>...</td><td>(更新しない)</td></tr><tr><td>n=7</td><td>(更新しない)</td><td>(更新しない)</td><td>...</td><td>(更新しない)</td></tr><tr><td>n=8</td><td>(更新しない)</td><td>(更新しない)</td><td>...</td><td>(更新しない)</td></tr></table> <展開エリア> <table><tr><td>1点目のFirstOnPosition</td><td>1点目のLastOnPosition</td><td>...</td><td>1点目のDuration</td></tr><tr><td>2点目のFirstOnPosition</td><td>2点目のLastOnPosition</td><td>...</td><td>2点目のDuration</td></tr><tr><td>3点目のFirstOnPosition</td><td>3点目のLastOnPosition</td><td>...</td><td>3点目のDuration</td></tr><tr><td>4点目のFirstOnPosition</td><td>4点目のLastOnPosition</td><td>...</td><td>4点目のDuration</td></tr><tr><td>5点目のFirstOnPosition</td><td>5点目のLastOnPosition</td><td>...</td><td>5点目のDuration</td></tr><tr><td>6点目のFirstOnPosition</td><td>6点目のLastOnPosition</td><td>...</td><td>6点目のDuration</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td>10000点目のFirstOnPosition</td><td>10000点目のLastOnPosition</td><td>...</td><td>10000点目のDuration</td></tr></table> オフセット(Offset) 読み出しデータ数(Points)	n=1	3点目のFirstOnPosition	3点目のLastOnPosition	...	3点目のDuration	n=2	4点目のFirstOnPosition	4点目のLastOnPosition	...	4点目のDuration	n=3	5点目のFirstOnPosition	5点目のLastOnPosition	...	5点目のDuration	n=4	(更新しない)	(更新しない)	...	(更新しない)	n=5	(更新しない)	(更新しない)	...	(更新しない)	n=6	(更新しない)	(更新しない)	...	(更新しない)	n=7	(更新しない)	(更新しない)	...	(更新しない)	n=8	(更新しない)	(更新しない)	...	(更新しない)	1点目のFirstOnPosition	1点目のLastOnPosition	...	1点目のDuration	2点目のFirstOnPosition	2点目のLastOnPosition	...	2点目のDuration	3点目のFirstOnPosition	3点目のLastOnPosition	...	3点目のDuration	4点目のFirstOnPosition	4点目のLastOnPosition	...	4点目のDuration	5点目のFirstOnPosition	5点目のLastOnPosition	...	5点目のDuration	6点目のFirstOnPosition	6点目のLastOnPosition	...	6点目のDuration	...	...	...	...	10000点目のFirstOnPosition	10000点目のLastOnPosition	...	10000点目のDuration
n=1	3点目のFirstOnPosition	3点目のLastOnPosition	...	3点目のDuration																																																																					
n=2	4点目のFirstOnPosition	4点目のLastOnPosition	...	4点目のDuration																																																																					
n=3	5点目のFirstOnPosition	5点目のLastOnPosition	...	5点目のDuration																																																																					
n=4	(更新しない)	(更新しない)	...	(更新しない)																																																																					
n=5	(更新しない)	(更新しない)	...	(更新しない)																																																																					
n=6	(更新しない)	(更新しない)	...	(更新しない)																																																																					
n=7	(更新しない)	(更新しない)	...	(更新しない)																																																																					
n=8	(更新しない)	(更新しない)	...	(更新しない)																																																																					
1点目のFirstOnPosition	1点目のLastOnPosition	...	1点目のDuration																																																																						
2点目のFirstOnPosition	2点目のLastOnPosition	...	2点目のDuration																																																																						
3点目のFirstOnPosition	3点目のLastOnPosition	...	3点目のDuration																																																																						
4点目のFirstOnPosition	4点目のLastOnPosition	...	4点目のDuration																																																																						
5点目のFirstOnPosition	5点目のLastOnPosition	...	5点目のDuration																																																																						
6点目のFirstOnPosition	6点目のLastOnPosition	...	6点目のDuration																																																																						
...	...	...	...																																																																						
10000点目のFirstOnPosition	10000点目のLastOnPosition	...	10000点目のDuration																																																																						
1: ファイル	オフセット(Offset), および読み出しデータ数(Points)を指定して, 全要素を読み出してください。 (0以外を設定した場合, オフセット・読み出し/書き込みデータ数不正(エラーコード: 1BADH)が発生します。) 読み出しデータ1(Data1)に格納されるデータは, PROFILE_CAMSWITCH_DATA型のデータとなります。 読み出しデータ2(Data2)に格納されるデータは, 指定したスイッチ設定のデータ(FirstOnPosition, LastOnPosition, Direction, CamSwitchMode, Duration)をPROFILE_CAMSWITCH_ELEMENT型の配列で定義したデータとなります。																																																																								

■読出しデータ1(Data1)/読出しデータ2(Data2)

演算プロファイルの読出しデータをTARGET_REF構造体で設定します。使用できるデータ種別は、[VAR]、[DEV]です。TARGET_REF構造体についての詳細は、下記を参照してください。

1421ページ TARGET_REF(入力信号)

使用できるデータ型は、各演算プロファイル形式の読出し用データ構造体の型です。

・読出しデータ1(Data1)

読出しデータ1(Data1)に設定する演算プロファイル形式、およびデータ型を示します。

演算プロファイル形式	データ型	参照
カムデータ	PROFILE_CAM_DATA	1426ページ PROFILE_CAM_DATA
ロータリーカッター用カムデータ	PROFILE_ROTARY_CUTTER	1428ページ PROFILE_ROTARY_CUTTER
多軸位置決めデータ	PROFILE_POSITIONING_DATA	1430ページ PROFILE_POSITIONING_DATA
デジタルカムスイッチデータ	PROFILE_CAMSWITCH_DATA	1429ページ PROFILE_CAMSWITCH_DATA

例

デバイス、ラベルを使用してPROFILE_CAM_DATA型のラベル(CamData1)を読み出す場合、下記のように記述します。

Data1.Target := "[VAR]CamData1"

・読出しデータ2(Data2)

読出しデータ2(Data2)に設定する演算プロファイル形式、およびデータ型を示します。読出し先(Target)、および対象となるカムデータの補間方法指定(Interpolate)により、設定するデータ型の構造体が異なります。

なお、演算プロファイル全体を読出しする場合、指定する配列の要素数が(分解能 +1/座標数)より小さい場合は、読出し/書き込みデータ数不一致(エラーコード: 1BACH)となります。

○: 対応, ×: 非対応

演算プロファイル形式	読出し先(Target)	データ型	補間方法指定(Interpolate)			参照
			0: 直線補間	1: 区間ごとに指定	2: スプライン補間	
カムデータ	0: 展開エリア 2: 展開エリア(モーションサービス処理)	LREAL[]* ¹	○	○* ³	○* ³	—
	1: ファイル	LREAL[]* ¹	○	×	×	—
		PROFILE_CAM_ELEMENT[]* ²	×	○	○* ⁴	1427ページ PROFILE_CAM_ELEMENT
ロータリーカッター用カムデータ	設定は不要です。 設定しても無視します。					—
多軸位置決めデータ	—	PROFILE_POSITIONING_DATA_ELEMENT[]* ⁵	—	—	—	1430ページ PROFILE_POSITIONING_DATA
デジタルカムスイッチデータ	—	PROFILE_CAMSWITCH_ELEMENT[]* ⁶	—	—	—	1430ページ PROFILE_CAMSWITCH_ELEMENT

*1 LREAL型の2次元配列は、下記のように設定します。

LREAL[m..n,o..p]

項目	内容
要素数(n-m+1)	配列要素が読み出す点数分以上となるように設定します。 演算プロファイル全体を読出しする場合(オフセット(Offset)、読出しデータ数(Points)を「0」とした場合)は、対象のカムデータの分解能+1/座標数以上となるように設定してください。 ただし、読出しデータ1(Data1)にて読出した分解能+1/座標数より大きいエリアは更新されません。
次元数(p-o+1)	各ポイントの1サイクル現在位置とストロークを読み出します。 配列要素が「2」となるように設定してください。 配列要素を「3」以上とした場合、読出しは可能ですが、データが前詰めされるため意図した結果が得られない可能性があるため、必ず「2」となるように設定してください。

*2 PROFILE_CAM_ELEMENTの配列は、下記のように設定します。
PROFILE_CAM_ELEMENT[m..n]

項目	内容
要素数(n-m+1)	配列要素が読み出す点数分以上となるように設定します。 演算プロファイル全体を読み出す場合(オフセット(Offset), 読出しデータ数(Points)を「0」とした場合)は、対象のカムデータの区間数(NumberOfSections)以上となるように設定してください。 ただし、読出しデータ1(Data1)にて読出した区間数(NumberOfSections)より大きいエリアは更新されません。

- *3 展開エリアに展開したストロークデータ(1サイクル現在値とストロークのベア)として読み出します。1ポイント目のデータのXは開始点、Yは初期ストローク量に設定した値となります。
2ポイント目以降のデータは、分解能で分割したストロークデータとなります。そのため、展開エリアでの読出し点数は「分解能+1」となります。
- *4 終了点、ストロークのみ使用します。
- *5 PROFILE_POSITIONING_DATA_ELEMENT[m..n]の配列は、下記のように設定します。

項目	内容
要素数(n-m+1)	配列要素が読み出す点数分以上となるように設定します。 演算プロファイル全体を読み出す場合(オフセット(Offset), 読出しデータ数(Points)を「0」とした場合)は、配列の要素数が位置決めデータ設定数(PROFILE_POSITIONING_DATA.TotalNumberOfData)以上となるように設定してください。 配列の要素数が少ない場合は、読出し/書き込みデータ数不一致(エラーコード: 1BACH)となります。

*6 PROFILE_CAMSWITCH_ELEMENT[m..n]の配列は、下記のように設定します。

項目	内容
要素数(n-m+1)	配列要素が読み出す点数分以上となる正数を設定します。 演算プロファイル全体を読み出す場合(オフセット(Offset), 読出しデータ数(Points)を「0」とした場合)は、デジタルカムスイッチデータのスイッチ数(PROFILE_CAMSWITCH_DATA.NumberOfSwitches)以上となるように設定してください。 ただし、Data1で読み出したスイッチ数より大きいエリアは更新されません。

例

補間方法指定(Interpolate)によるLREAL型の場合

補間方法指定(Interpolate)	設定例
0: 直線補間	TARGET_REF構造体の対象(Target)に、座標数「100」の2次元ラベルを設定する例を下記に示します。 Data2.Target := "[VAR]CamData2" *: CamData2は、グローバルラベルのラベル名に「CamData2」、データ型に「LREAL[1..100,1..2]」のラベルを宣言し、対象(Target)には文字列を指定します。読出しするデータは下記ようになります。 ・CamData2[n,1]: nポイント目の1サイクル現在位置 ・CamData2[n,2]: nポイント目のストローク
1: 区間ごとに指定	TARGET_REF構造体の対象(Target)に、分解能「100」の2次元ラベルを設定する例を下記に示します。
2: スプライン補間	TARGET_REF.Target := "[VAR]CamData2" *: CamData2は、グローバルラベルのラベル名に「CamData2」、データ型に「LREAL[1..101,1..2]」のラベルを宣言し、対象(Target)には文字列を指定します。読出しするデータは下記ようになります。 ・CamData2[1,1]: 開始点に設定した値 ・CamData2[1,2]: 初期ストローク量の設定した値 ・CamData2[n+1,1]: nポイント目の1サイクル現在位置 ・CamData2[n+1,2]: nポイント目のストローク

例

PROFILE_CAM_ELEMENT型の場合

設定例
TARGET_REF構造体の対象(Target)に、区間数「10」の2次元ラベルを設定する例を下記に示します。 Data2.Target := "[VAR]CamData2" *: CamData2は、グローバルラベルのラベル名に「CamData2」、データ型に「PROFILE_CAM_ELEMENT[1..10]」のラベルを宣言し、対象(Target)には文字列を指定します。読出しするデータは下記ようになります。 ・CamData2[n]: nポイント目の区間データ(PROFILE_CAM_ELEMENT型)

Point

- 設定した読出し用データ構造体の型が演算プロファイル形式ごとに要求する型と一致しない場合は、読出し/書き込みデータ型不正(エラーコード: 1BAAH)となります。
- 読出しデータ1(Data1)/読出しデータ2(Data2)に設定する演算プロファイルのカムデータについての詳細は、下記を参照してください。

📖使用するコントローラのユーザーズマニュアル

- [VAR]を使用すると読出し処理に時間がかかる場合があります。[DEV]を使用することで[VAR]より高速に読出しができるため、読出しの処理時間を短縮したい場合は、[DEV]を使用してください。

■読出し先(Target)

演算プロファイルの読出し先を設定します。

設定値	内容
0: 展開エリア 2: 展開エリア(モーションサービス処理)	• PROFILE_DATA構造体のプロファイルID(ID)の値を参照して、PROFILE_ID構造体のプロファイルID番号(Number)が示す展開エリアの演算プロファイルに対し、読出しを行います。 • 演算プロファイルが存在しない場合(未展開の場合)、演算プロファイル無し(エラーコード: 1B8EH)となります。
1: ファイル	• PROFILE_DATA構造体の演算プロファイル格納場所(Location)の値を参照して、ファイル名称とパスが示すファイルの演算プロファイルに対し、読出しを行います。 • 演算プロファイルが存在しない場合、演算プロファイル無し(エラーコード: 1B8EH)となります。

Point

- 読出し先(Target)が「1: ファイル」、「2: 展開エリア(モーションサービス処理)」の場合、モーションサービス処理で実行するため処理が完了するまでに時間がかかる場合があります。必要に応じてDone(実行完了)をインタロックに使用してください。
- Target = 「0: 展開エリア」の場合、FBを呼び出したタスク内で処理を行うため即時完了します。読み出すデータ点数が多いとプログラムの実行時間が延びるため、必要に応じて複数回に分けて読出しを実行するか、Target = 「2: 展開エリア(モーションサービス処理)」を使用してください。

注意事項

- 一度に読出しできるデータ点数には限りがあります。1回のFB実行で読出しが完了しないときは、複数回に分けて読出しを実行してください。

プログラム例

「LREAL型(2次元配列)の形式」のカムデータ1(ProfileData0001)と「PROFILE_CAM_ELEMENT型の形式」のカムデータ2(ProfileData0002)をファイルから読み出すプログラム例を下記に示します。

■演算プロファイル

プロファイルID	ラベル名	データ型	コメント
1	ProfileData0001	MC_CAM_REF	カムデータ1
2	ProfileData0002	MC_CAM_REF	カムデータ2

・演算プロファイル(カムデータ)の設定

項目	ProfileData0001	ProfileData0002
補間方法指定	直線補間	区間ごとに指定
分解能	—	256
1サイクル長設定	単位	degree
	1サイクル長	360.00000
ストローク量	単位	%
	ストローク量	100.0
カム1サイクル時間	—	1.000[s]
初期ストローク値	—	0.0000000
カム1サイクル最小値	0.00000	—
カム1サイクル最大値	360.00000	—

・ProfileData0001

区間No.	入力値[degree]	出力値[pulse]
1	0.00000	0.0
2	90.00000	60.0
3	180.00000	100.0
4	270.00000	30.0
5	360.00000	0.0

・ProfileData0002

区間No.	開始点[degree]	終了点[degree]	ストローク[%]	カム曲線種別
1	0.00000	90.00000	60.0000000	単弦
2	90.00000	180.00000	100.0000000	単弦
3	180.00000	270.00000	30.0000000	単弦
4	270.00000	0.00000	0.0000000	単弦

■使用するラベル

ラベル名	データ型	コメント
MCv_ReadProfileData_1	MCv_ReadProfileData	プロファイル読出FB 1
ReadData1_1	TARGET_REF	読出しデータ1 プロファイル1
ReadData2_1	TARGET_REF	読出しデータ2 プロファイル1
ReadData1_Lreal	PROFILE_CAM_DATA	プロファイルデータ カムデータ1
ReadData2_Lreal	倍精度実数(0..4,0..1)	入力値/出力値データ LREAL型
bReadProfile1	ビット	プロファイル読出指令1
bDone1	ビット	完了1
bBusy1	ビット	実行中1
bError1	ビット	エラー 1
uwErrorID1	ワード[符号なし]/ビット列[16ビット]	エラーコード1
Cam_ProfileData0001 ^{*1}	PROFILE_DATA	プロファイルデータ0001
MCv_ReadProfileData_2	MCv_ReadProfileData	プロファイル読出FB 2
ReadData1_2	TARGET_REF	読出しデータ1 プロファイル2
ReadData2_2	TARGET_REF	読出しデータ2 プロファイル2
ReadData1_Element	PROFILE_CAM_DATA	プロファイルデータ カムデータ2
ReadData2_Element	PROFILE_CAM_ELEMENT(0..3)	入力値/出力値データ PROFILE_CAM_ELEMENT型
bReadProfile2	ビット	プロファイル読出指令2
bDone2	ビット	完了2
bBusy2	ビット	実行中2
bError2	ビット	エラー 2
uwErrorID2	ワード[符号なし]/ビット列[16ビット]	エラーコード2
Cam_ProfileData0002 ^{*1}	PROFILE_DATA	プロファイルデータ0002

*1 FBD/LDのプログラムのみ使用(演算プロファイルをFBD/LDのプログラムで使用するために必要)

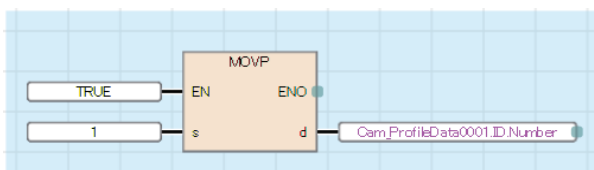
■使用するグローバルラベル

ラベル名	データ型	コメント
G_ProfileCamData ^{*1}	PROFILE_CAM_DATA	プロファイルカムデータ
G_ProfileCamELEMENT ^{*1}	PROFILE_CAM_ELEMENT	プロファイルカムエレメント

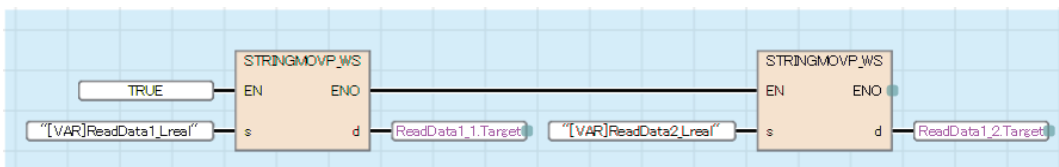
*1 FBD/LDのプログラムのみ使用(演算プロファイルをFBD/LDのプログラムで使用するために必要)

■FBD/LDのプログラム

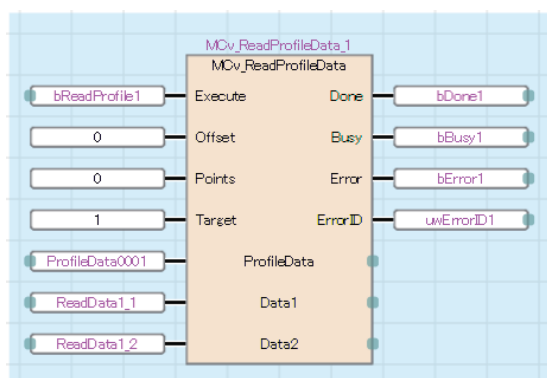
- プロファイルデータ0001



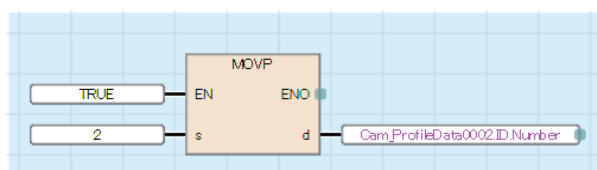
- 読出しデータ1



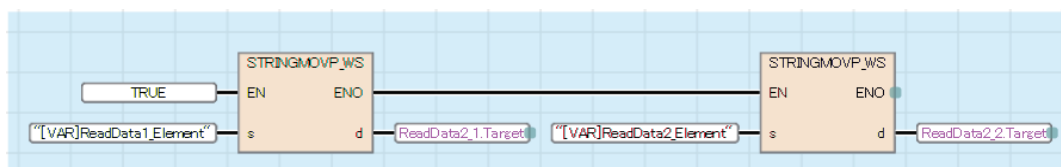
- プロファイル読出1(LREAL型(2次元配列)の形式でカムデータを読出し)



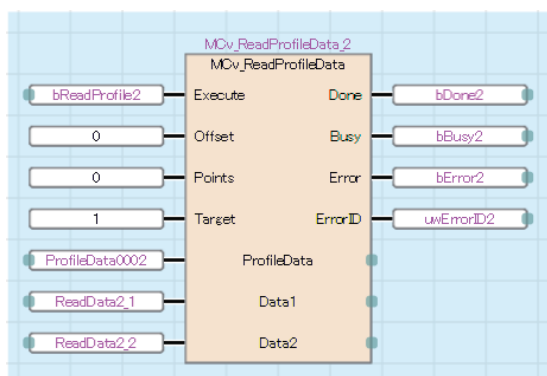
- プロファイルデータ0002



- 読出しデータ2



- プロファイル読出2(PROFILE_CAM_ELEMENT型の形式でカムデータを読出し)



■STのプログラム

```
//-----読出しデータ1-----
ReadData1_1.Target := "[VAR]ReadData1_Lreal";
ReadData2_1.Target := "[VAR]ReadData2_Lreal";

//-----プロファイル読出1-----
//-----LREAL型(2次元配列)の形式でカムデータを読出し-----
MCv_ReadProfileData_1(
  ProfileData:= ProfileData0001.ProfileData ,
  Data1:= ReadData1_1 ,
  Data2:= ReadData2_1 ,
  Execute:= bReadProfile1 ,
  Offset:= 0 ,
  Points:= 0 ,
  Target:= 1 ,
  Done=> bDone1 ,
  Busy=> bBusy1 ,
  Error=> bError1 ,
  ErrorID=> uwErrorID1
);

//-----読出しデータ2-----
ReadData1_2.Target := "[VAR]ReadData1_Element";
ReadData2_2.Target := "[VAR]ReadData2_Element";

//-----プロファイル読出2-----
//-----PROFILE_CAM_ELEMENT型の形式でカムデータを読出し-----
MCv_ReadProfileData_2(
  ProfileData:= ProfileData0002.ProfileData ,
  Data1:= ReadData1_2 ,
  Data2:= ReadData2_2 ,
  Execute:= bReadProfile2 ,
  Offset:= 0 ,
  Points:= 0 ,
  Target:= 1 ,
  Done=> bDone2 ,
  Busy=> bBusy2 ,
  Error=> bError2 ,
  ErrorID=> uwErrorID2
);
```

43.2 プロファイル書込

MCv_WriteProfileData

指定した演算プロファイルデータを、展開エリアまたはファイルに書き込みます。

ラダー	FBD/LD	ST
		<pre> MCv_WriteProfileData(ProfileData:= ?PROFILE_DATA?, Data1:= ?TARGET_REF?, Data2:= ?TARGET_REF?, Execute:= ?BOOL?, Offset:= ?DWORD?, Points:= ?DWORD?, Target:= ?WORD?, ExecutionMode:= ?INT?, Done=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?, ProfileID=> ?PROFILE_ID?); </pre>

設定データ

■入出力変数

入出力変数	名称	データ型	入力取込	有効範囲	初期値	内容
ProfileData	プロファイルデータ	PROFILE_DATA	起動時	—	省略不可	書き込む演算プロファイルを設定します。 コントローラ側でFBを使用する場合は ProfileData.ID.StartIOは設定不要です。 Data1.StartIOで設定した入出力番号を参照します。 演算プロファイルはPROFILE_DATA構造体で設定 します。PROFILE_DATA構造体については、下記 を参照してください。 ☞ 1424ページ PROFILE_DATA 書き込み用データ構造体を定義した演算プロファイル 形式は、書き込み操作ができます。
Data1	書き込みデータ1	TARGET_REF	起動時	—	省略不可	書き込む演算プロファイル形式に応じた書き込み用 データ構造体を設定します。 詳細は、下記を参照してください。 ☞ 1867ページ 書き込みデータ1(Data1)/書き込みデ ータ2(Data2)
Data2	書き込みデータ2	TARGET_REF	起動時	—	省略不可	書き込む演算プロファイル形式に応じた書き込み用 データ構造体を設定します。 詳細は、下記を参照してください。 ☞ 1867ページ 書き込みデータ1(Data1)/書き込みデ ータ2(Data2)

■入力変数

入力変数	名称	データ型	取込	有効範囲	初期値	内容
Execute	起動	BOOL	起動時	TRUE, FALSE	FALSE	TRUEでMCv_WriteProfileData(プロファイル書込)を実行します。
Offset	オフセット	DWORD(UDINT)	起動時	0~分解能 (2~65535)	0	演算プロファイル先頭からのオフセットを設定します。 オフセット(Offset)が演算プロファイルの範囲を超えると、オフセット範囲外(エラーコード: 1B92H)となり、書込みは行いません。 オフセット(Offset)に「0」以外の値を設定する場合、書込みデータ数(Points)には「0」以外の値を設定してください。(書込みデータ数(Points)に「0」を設定した場合、オフセット・読出し/書込みデータ数不正(エラーコード: 1BADH)となります。) *: オフセット(Offset)と書込みデータ数(Points)の設定値により書き込む動作が異なります。詳細は、下記を参照してください。 📖 1864ページ オフセット(Offset)と書込みデータ数(Points)の設定
Points	書込みデータ数	DWORD(UDINT)	起動時	0~4294967295	0	書き込むデータ点数を設定します。 書込みデータ数(Points)が演算プロファイルの要素数を超える場合、演算プロファイルの要素数の範囲内で書込みを行います。 演算プロファイル書込みにて、書込み先(Target)が「1: ファイル」の場合は、「0」を設定してください。(「0」以外を設定した場合、オフセット・読出し/書込みデータ数不正(エラーコード: 1BADH)となります。)
Target	書込み先	WORD(UINT)	起動時	0, 1, 2	0	演算プロファイルの書込み先を設定します。 • 0: 展開エリア • 1: ファイル • 2: 展開エリア(モーションサービス処理) *: 書込み先(Target)に「1: ファイル」を設定した場合、オフセット(Offset)と書込みデータ数(Points)には、「0」を設定してください。(「0」以外を設定した場合は、オフセット・読出し/書込みデータ数不正(エラーコード: 1BADH)となります。) 詳細は、下記を参照してください。 📖 1870ページ 書込み先(Target)
ExecutionMode	起動モード	INT (MC_EXECUTION_MODE)	起動時	0, 1, 3	0	MCv_WriteProfileData(プロファイル書込)の実行タイミングを設定します。 • 0: 直ちに実行(mcImmediately) • 1: 完了待ちして実行(mcQueued) • 3: 投機的に実行(mcSpeculatively) 詳細は、下記を参照してください。 📖 1870ページ 起動モード(ExecutionMode)

■出力変数

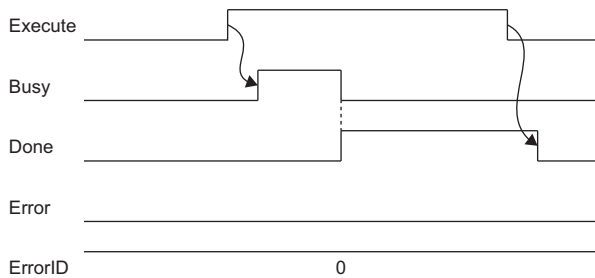
出力変数	名称	データ型	初期値	内容
Done	完了	BOOL	FALSE	制御が完了したときに、TRUEになります。 動作完了時に起動(Execute)の状態により下記となります。 ■起動(Execute)がTRUEの場合 起動(Execute)をFALSEにするまでTRUEのままになります。 ■起動(Execute)がFALSEの場合 1周期だけTRUEになります。
Busy	実行中	BOOL	FALSE	MCv_WriteProfileData(プロファイル書込)を実行したときに、TRUEになります。
Error	エラー	BOOL	FALSE	異常が発生したときに、TRUEになります。
ErrorID	エラーコード	WORD(UINT)	0	異常が発生したときに、エラーコードを返します。 エラーコードの詳細は、下記を参照してください。 📖 使用するコントローラのユーザーズマニュアル
ProfileID	プロファイルID	PROFILE_ID	0	プロファイルIDを出力します。

機能

- MCv_WriteProfileData(プロファイル書込)を使用して演算プロファイルの書込み操作を行います。
- プロファイルデータ(ProfileData), および書込み先(Target)で設定したファイル, または展開エリアの演算プロファイルに対し, オフセット(Offset)で設定したデータから書込みデータ数(Points)で設定した点数分を書き込みます。
- 演算プロファイル全体を書込むには, オフセット(Offset)と書込みデータ数(Points)の両方に「0」を設定します。
- 書き込むデータは, 書込みデータ1(Data1), 書込みデータ2(Data2)で設定した変数で設定します。

■タイミングチャート

- 正常完了の場合



- 異常完了の場合

異常完了時のタイミングチャートについては, 下記を参照してください。

📖 1368ページ 実行指令(Execute)型によるモーション制御FBの基本動作

■オフセット(Offset)と書込みデータ数(Points)の設定

オフセット(Offset)と書込みデータ数(Points)に設定した値により下記の動作となります。

オフセット(Offset), 書込みデータ数(Points)に設定する値の内容は, 演算プロファイル形式により異なります。

- 演算プロファイル形式が「カムデータ」の場合

補間方法指定 (Interpolate)	内容																																													
0: 直線補間	書き込み対象の座標の先頭をオフセット(Offset), 座標数を書込みデータ数(Points)で指定してください。 書き込みデータ1(Data1)は, 補間方法指定(Interpolate)が「0: 直線補間」のPROFILE_CAM_DATA型のデータを設定します。書き込みデータ2(Data2)は, 設定した座標データ(1サイクル現在値とストロークのペア)をLREAL型の二次元配列で定義したデータを設定します。 <例> オフセット(Offset)に「2」, 書き込みデータ数(Points)に「3」を設定したときの書き込みデータ2(Data2)の書き込み結果(書き込みデータ2(Data2)に設定する変数をデータ型「LREAL[1..8,1..2]」のラベル「CamData2」とした場合) <変数(ラベル/デバイス)> LREAL型の二次元配列のラベル (CamData2[1..8,1..2]) <table><thead><tr><th></th><th>CamData[n][1]</th><th>CamData[n][2]</th></tr></thead><tbody><tr><td>n=1</td><td>2点目の入力値</td><td>2点目の出力値</td></tr><tr><td>n=2</td><td>3点目の入力値</td><td>3点目の出力値</td></tr><tr><td>n=3</td><td>4点目の入力値</td><td>4点目の出力値</td></tr><tr><td>n=4</td><td>(更新しない)</td><td>(更新しない)</td></tr><tr><td>n=5</td><td>(更新しない)</td><td>(更新しない)</td></tr><tr><td>n=6</td><td>(更新しない)</td><td>(更新しない)</td></tr><tr><td>n=7</td><td>(更新しない)</td><td>(更新しない)</td></tr><tr><td>n=8</td><td>(更新しない)</td><td>(更新しない)</td></tr></tbody></table> オフセット (Offset) 書き込み データ数 (Points) <table><thead><tr><th colspan="2"><展開エリア/ファイル>*1</th></tr></thead><tbody><tr><td>0点目の入力値</td><td>0点目の出力値</td></tr><tr><td>1点目の入力値</td><td>1点目の出力値</td></tr><tr><td>2点目の入力値</td><td>2点目の出力値</td></tr><tr><td>3点目の入力値</td><td>3点目の出力値</td></tr><tr><td>4点目の入力値</td><td>4点目の出力値</td></tr><tr><td>5点目の入力値</td><td>5点目の出力値</td></tr><tr><td>6点目の入力値</td><td>6点目の出力値</td></tr><tr><td>7点目の入力値</td><td>7点目の出力値</td></tr></tbody></table> *1: 読み出し先(Target)によります。		CamData[n][1]	CamData[n][2]	n=1	2点目の入力値	2点目の出力値	n=2	3点目の入力値	3点目の出力値	n=3	4点目の入力値	4点目の出力値	n=4	(更新しない)	(更新しない)	n=5	(更新しない)	(更新しない)	n=6	(更新しない)	(更新しない)	n=7	(更新しない)	(更新しない)	n=8	(更新しない)	(更新しない)	<展開エリア/ファイル>*1		0点目の入力値	0点目の出力値	1点目の入力値	1点目の出力値	2点目の入力値	2点目の出力値	3点目の入力値	3点目の出力値	4点目の入力値	4点目の出力値	5点目の入力値	5点目の出力値	6点目の入力値	6点目の出力値	7点目の入力値	7点目の出力値
	CamData[n][1]	CamData[n][2]																																												
n=1	2点目の入力値	2点目の出力値																																												
n=2	3点目の入力値	3点目の出力値																																												
n=3	4点目の入力値	4点目の出力値																																												
n=4	(更新しない)	(更新しない)																																												
n=5	(更新しない)	(更新しない)																																												
n=6	(更新しない)	(更新しない)																																												
n=7	(更新しない)	(更新しない)																																												
n=8	(更新しない)	(更新しない)																																												
<展開エリア/ファイル>*1																																														
0点目の入力値	0点目の出力値																																													
1点目の入力値	1点目の出力値																																													
2点目の入力値	2点目の出力値																																													
3点目の入力値	3点目の出力値																																													
4点目の入力値	4点目の出力値																																													
5点目の入力値	5点目の出力値																																													
6点目の入力値	6点目の出力値																																													
7点目の入力値	7点目の出力値																																													
1: 区間ごとに 指定	部分的に書き込むことはできません。 オフセット(Offset), および書き込みデータ数(Points)に「0」を設定し, 演算プロファイル全体に対して書き込みを行います。「0」以外を設定した場合, オフセット・読み出し/書き込みデータ数不正(エラーコード: 1BADH)となります。書き込みデータ1(Data1)は, 補間方法指定(Interpolate)が「1: 区間ごとに指定」, または「2: スプライン補間」のPROFILE_CAM_DATA型のデータを設定します。書き込みデータ2(Data2)は, 区間データをPROFILE_CAM_ELEMENT型の配列で定義したデータを設定します。																																													
2: スプライン 補間																																														

・演算プロファイル形式が「ロータリーカッター」の場合

部分的に書き込むことはできません。オフセット(Offset), および書き込みデータ数(Points)に「0」を設定し, 演算プロファイル全体に対して書き込みを行います。オフセット(Offset), および書き込みデータ数(Points)の設定は無視します。書き込みデータ1(Data1)は, 演算プロファイル形式が「ロータリーカッター」のPROFILE_ROTARY_CUTTER型のデータを設定します。書き込みデータ2(Data2)は, 使用しません。

・演算プロファイル形式が「多軸位置決めデータ」の場合

書き込み先(Target)	内容																																																																																																										
0: 展開エリア	位置決めデータを書込む区間の先頭をオフセット(Offset), 区間数(NumberOfSections)を書込みデータ数(Points)で指定してください。Data1(TARGET_REF型)に指定したPROFILE_POSITIONING_DATA型の変数に, データを書き込みます。																																																																																																										
2: 展開エリア(モーションサービス処理)	Data2(TARGET_REF型)にPROFILE_POSITIONING_DATA_ELEMENT型配列の変数を指定する場合は, 指定した変数から位置決めデータを書き込みます。 <例> オフセット(Offset)に「2」, 書き込みデータ数(Points)に「3」を設定したときの書き込みデータ2(Data2)の書き込み結果(書き込みデータ2(Data2)に設定する変数をデータ型「PROFILE_POSITIONING_DATA_ELEMENT[1..8]」のラベル「PositioningData2」とした場合) <変数(ラベル/デバイス)> PROFILE_POSITIONING_DATA_ELEMENT型のラベル (PositioningData2[1..8]) <table><tr><td>n=1</td><td>3点目のDataNo(3)</td><td>3点目のOperationPattern</td><td>...</td></tr><tr><td>n=2</td><td>4点目のDataNo(4)</td><td>4点目のOperationPattern</td><td>...</td></tr><tr><td>n=3</td><td>5点目のDataNo(5)</td><td>5点目のOperationPattern</td><td>...</td></tr><tr><td>n=4</td><td>(使用しない)</td><td>(使用しない)</td><td>...</td></tr><tr><td>n=5</td><td>(使用しない)</td><td>(使用しない)</td><td>...</td></tr><tr><td>n=6</td><td>(使用しない)</td><td>(使用しない)</td><td>...</td></tr><tr><td>n=7</td><td>(使用しない)</td><td>(使用しない)</td><td>...</td></tr><tr><td>n=8</td><td>(使用しない)</td><td>(使用しない)</td><td>...</td></tr></table> オフセット(Offset) 書き込みデータ数 (Points) <展開エリア> <table><tr><td>1点目のDataNo(1)</td><td>1点目のOperationPattern</td><td>...</td></tr><tr><td>2点目のDataNo(2)</td><td>2点目のOperationPattern</td><td>...</td></tr><tr><td>3点目のDataNo(3)</td><td>3点目のOperationPattern</td><td>...</td></tr><tr><td>4点目のDataNo(4)</td><td>4点目のOperationPattern</td><td>...</td></tr><tr><td>5点目のDataNo(5)</td><td>5点目のOperationPattern</td><td>...</td></tr><tr><td>6点目のDataNo(6)</td><td>6点目のOperationPattern</td><td>...</td></tr><tr><td>...</td><td>...</td><td>...</td></tr></table> 指定した変数の各位置決めデータのDataNoは, 書き込み先のDataNoと同じとなるように指定してください。 • DataNoが「0」の位置決めデータがある場合, 未設定とみなし, 該当のDataNoにあたる区間データに初期値を格納します。(DataNoは0となります。) <例> 上記の例で変数n=2のDataNoを0に設定した場合 <変数(ラベル/デバイス)> PROFILE_POSITIONING_DATA_ELEMENT型のラベル (PositioningData2[1..8]) <table><tr><td>n=1</td><td>3点目のDataNo(3)</td><td>3点目のOperationPattern</td><td>...</td></tr><tr><td>n=2</td><td>DataNo(0)</td><td>(使用しない)</td><td>...</td></tr><tr><td>n=3</td><td>5点目のDataNo(5)</td><td>5点目のOperationPattern</td><td>...</td></tr><tr><td>n=4</td><td>(使用しない)</td><td>(使用しない)</td><td>...</td></tr><tr><td>n=5</td><td>(使用しない)</td><td>(使用しない)</td><td>...</td></tr><tr><td>n=6</td><td>(使用しない)</td><td>(使用しない)</td><td>...</td></tr><tr><td>n=7</td><td>(使用しない)</td><td>(使用しない)</td><td>...</td></tr><tr><td>n=8</td><td>(使用しない)</td><td>(使用しない)</td><td>...</td></tr></table> オフセット(Offset) 書き込みデータ数 (Points) <展開エリア> <table><tr><td>1点目のDataNo(1)</td><td>1点目のOperationPattern</td><td>...</td></tr><tr><td>2点目のDataNo(2)</td><td>2点目のOperationPattern</td><td>...</td></tr><tr><td>3点目のDataNo(3)</td><td>3点目のOperationPattern</td><td>...</td></tr><tr><td>4点目のDataNo(0)</td><td>4点目のOperationPattern(0)</td><td>(初期値)</td></tr><tr><td>5点目のDataNo(5)</td><td>5点目のOperationPattern</td><td>...</td></tr><tr><td>6点目のDataNo(6)</td><td>6点目のOperationPattern</td><td>...</td></tr><tr><td>...</td><td>...</td><td>...</td></tr></table> DataNoが「0」または書き込み先のDataNoのいずれでもない位置決めデータがある場合, オフセット・読み出し/書き込みデータ数不正(エラーコード: 1BADH)が発生します。 Data2(TARGET_REF型)にPROFILE_POSITIONING_DATA_EXTENDED型の変数を指定する場合 変数のメンバConditionSignal, SkipSignalおよびPositioningDataにWSTRING型で指定した変数から, 条件信号, スキップ信号, および位置決めデータを書き込みます。 条件信号, スキップ信号は, PROFILE_SIGNAL_SELECT_ELEMENT型配列のデータを指定してください。 位置決めデータは, PROFILE_POSITIONING_DATA_ELEMENT型配列のデータを指定してください。(Data2(TARGET_REF型)にPROFILE_POSITIONING_DATA_ELEMENT型配列の変数を指定した場合と同様です。)	n=1	3点目のDataNo(3)	3点目のOperationPattern	...	n=2	4点目のDataNo(4)	4点目のOperationPattern	...	n=3	5点目のDataNo(5)	5点目のOperationPattern	...	n=4	(使用しない)	(使用しない)	...	n=5	(使用しない)	(使用しない)	...	n=6	(使用しない)	(使用しない)	...	n=7	(使用しない)	(使用しない)	...	n=8	(使用しない)	(使用しない)	...	1点目のDataNo(1)	1点目のOperationPattern	...	2点目のDataNo(2)	2点目のOperationPattern	...	3点目のDataNo(3)	3点目のOperationPattern	...	4点目のDataNo(4)	4点目のOperationPattern	...	5点目のDataNo(5)	5点目のOperationPattern	...	6点目のDataNo(6)	6点目のOperationPattern	...	...	...	...	n=1	3点目のDataNo(3)	3点目のOperationPattern	...	n=2	DataNo(0)	(使用しない)	...	n=3	5点目のDataNo(5)	5点目のOperationPattern	...	n=4	(使用しない)	(使用しない)	...	n=5	(使用しない)	(使用しない)	...	n=6	(使用しない)	(使用しない)	...	n=7	(使用しない)	(使用しない)	...	n=8	(使用しない)	(使用しない)	...	1点目のDataNo(1)	1点目のOperationPattern	...	2点目のDataNo(2)	2点目のOperationPattern	...	3点目のDataNo(3)	3点目のOperationPattern	...	4点目のDataNo(0)	4点目のOperationPattern(0)	(初期値)	5点目のDataNo(5)	5点目のOperationPattern	...	6点目のDataNo(6)	6点目のOperationPattern	...	...	...	...
n=1	3点目のDataNo(3)	3点目のOperationPattern	...																																																																																																								
n=2	4点目のDataNo(4)	4点目のOperationPattern	...																																																																																																								
n=3	5点目のDataNo(5)	5点目のOperationPattern	...																																																																																																								
n=4	(使用しない)	(使用しない)	...																																																																																																								
n=5	(使用しない)	(使用しない)	...																																																																																																								
n=6	(使用しない)	(使用しない)	...																																																																																																								
n=7	(使用しない)	(使用しない)	...																																																																																																								
n=8	(使用しない)	(使用しない)	...																																																																																																								
1点目のDataNo(1)	1点目のOperationPattern	...																																																																																																									
2点目のDataNo(2)	2点目のOperationPattern	...																																																																																																									
3点目のDataNo(3)	3点目のOperationPattern	...																																																																																																									
4点目のDataNo(4)	4点目のOperationPattern	...																																																																																																									
5点目のDataNo(5)	5点目のOperationPattern	...																																																																																																									
6点目のDataNo(6)	6点目のOperationPattern	...																																																																																																									
...	...	...																																																																																																									
n=1	3点目のDataNo(3)	3点目のOperationPattern	...																																																																																																								
n=2	DataNo(0)	(使用しない)	...																																																																																																								
n=3	5点目のDataNo(5)	5点目のOperationPattern	...																																																																																																								
n=4	(使用しない)	(使用しない)	...																																																																																																								
n=5	(使用しない)	(使用しない)	...																																																																																																								
n=6	(使用しない)	(使用しない)	...																																																																																																								
n=7	(使用しない)	(使用しない)	...																																																																																																								
n=8	(使用しない)	(使用しない)	...																																																																																																								
1点目のDataNo(1)	1点目のOperationPattern	...																																																																																																									
2点目のDataNo(2)	2点目のOperationPattern	...																																																																																																									
3点目のDataNo(3)	3点目のOperationPattern	...																																																																																																									
4点目のDataNo(0)	4点目のOperationPattern(0)	(初期値)																																																																																																									
5点目のDataNo(5)	5点目のOperationPattern	...																																																																																																									
6点目のDataNo(6)	6点目のOperationPattern	...																																																																																																									
...	...	...																																																																																																									

書き込み先(Target)	内容																																																
1: ファイル	オフセット(Offset), および書き込みデータ数(Points)に0を指定して、演算プロファイル全体に対して書き込んでください。 (0以外を設定した場合、オフセット・読出し/書き込みデータ数不正(エラーコード: 1BADH)が発生します。) Data1(TARGET_REF型)に指定した変数から、PROFILE_POSITIONING_DATA型のデータを書き込みます。 Data2(TARGET_REF型)にPROFILE_POSITIONING_DATA_ELEMENT型配列の変数を指定する場合は、指定した変数から位置決めデータを書き込みます。 書き込む区間に、位置決めデータNo.が「0」の位置決めデータがある場合、該当のDataNoにあたる区間データを書き込みません。 <例> オフセット(Offset)に「0」、書き込みデータ数(Points)に「0」を設定したときの書き込みデータ2(Data2)の書き込み結果 (書き込みデータ2(Data2)に設定する変数をデータ型「PROFILE_POSITIONING_DATA_ELEMENT[1..100]」のラベル「PositioningData2」 とした場合) <変数(ラベル/デバイス)> PROFILE_POSITIONING_DATA_ELEMENT型のラベル (PositioningData2[1..100]) <table><tr><td>n=1</td><td>DataNo(0) (使用しない)</td><td>(使用しない)</td><td>...</td></tr><tr><td>n=2</td><td>1点目のDataNo(2)</td><td>1点目の OperationPattern</td><td>...</td></tr><tr><td>n=3</td><td>2点目のDataNo(3)</td><td>2点目の OperationPattern</td><td>...</td></tr><tr><td>n=4</td><td>DataNo(0) (使用しない)</td><td>(使用しない)</td><td>...</td></tr><tr><td>n=5</td><td>3点目のDataNo(5)</td><td>3点目の OperationPattern</td><td>...</td></tr><tr><td>n=6</td><td>DataNo(0) (使用しない)</td><td>(使用しない)</td><td>...</td></tr><tr><td></td><td>...</td><td>...</td><td>...</td></tr><tr><td>n=99</td><td>DataNo(0) (使用しない)</td><td>(使用しない)</td><td>...</td></tr><tr><td>n=100</td><td>4点目の DataNo(100)</td><td>4点目の OperationPattern</td><td>...</td></tr></table> <ファイル> <table><tr><td>1点目のDataNo(2)</td><td>1点目の OperationPattern</td><td>...</td></tr><tr><td>2点目のDataNo(3)</td><td>2点目の OperationPattern</td><td>...</td></tr><tr><td>3点目のDataNo(5)</td><td>3点目の OperationPattern</td><td>...</td></tr><tr><td>4点目のDataNo (100)</td><td>4点目の OperationPattern</td><td>...</td></tr></table> Data2(TARGET_REF型)にPROFILE_POSITIONING_DATA_EXTENDED型の変数を指定する場合 変数のメンバConditionSignal, SkipSignalおよびPositioningDataにWSTRING型で指定した変数から、条件信号、スキップ信号、および位置決めデータを書き込みます。 条件信号, スキップ信号は、PROFILE_SIGNAL_SELECT_ELEMENT型配列のデータを指定してください。 位置決めデータは、PROFILE_POSITIONING_DATA_ELEMENT型配列のデータを指定してください。(Data2(TARGET_REF型)にPROFILE_POSITIONING_DATA_ELEMENT型配列の変数を指定した場合と同様です。)	n=1	DataNo(0) (使用しない)	(使用しない)	...	n=2	1点目のDataNo(2)	1点目の OperationPattern	...	n=3	2点目のDataNo(3)	2点目の OperationPattern	...	n=4	DataNo(0) (使用しない)	(使用しない)	...	n=5	3点目のDataNo(5)	3点目の OperationPattern	...	n=6	DataNo(0) (使用しない)	(使用しない)	...		...	...	...	n=99	DataNo(0) (使用しない)	(使用しない)	...	n=100	4点目の DataNo(100)	4点目の OperationPattern	...	1点目のDataNo(2)	1点目の OperationPattern	...	2点目のDataNo(3)	2点目の OperationPattern	...	3点目のDataNo(5)	3点目の OperationPattern	...	4点目のDataNo (100)	4点目の OperationPattern	...
n=1	DataNo(0) (使用しない)	(使用しない)	...																																														
n=2	1点目のDataNo(2)	1点目の OperationPattern	...																																														
n=3	2点目のDataNo(3)	2点目の OperationPattern	...																																														
n=4	DataNo(0) (使用しない)	(使用しない)	...																																														
n=5	3点目のDataNo(5)	3点目の OperationPattern	...																																														
n=6	DataNo(0) (使用しない)	(使用しない)	...																																														
	...	...	...																																														
n=99	DataNo(0) (使用しない)	(使用しない)	...																																														
n=100	4点目の DataNo(100)	4点目の OperationPattern	...																																														
1点目のDataNo(2)	1点目の OperationPattern	...																																															
2点目のDataNo(3)	2点目の OperationPattern	...																																															
3点目のDataNo(5)	3点目の OperationPattern	...																																															
4点目のDataNo (100)	4点目の OperationPattern	...																																															

・ 演算プロファイル形式が「デジタルカムスイッチデータ」の場合

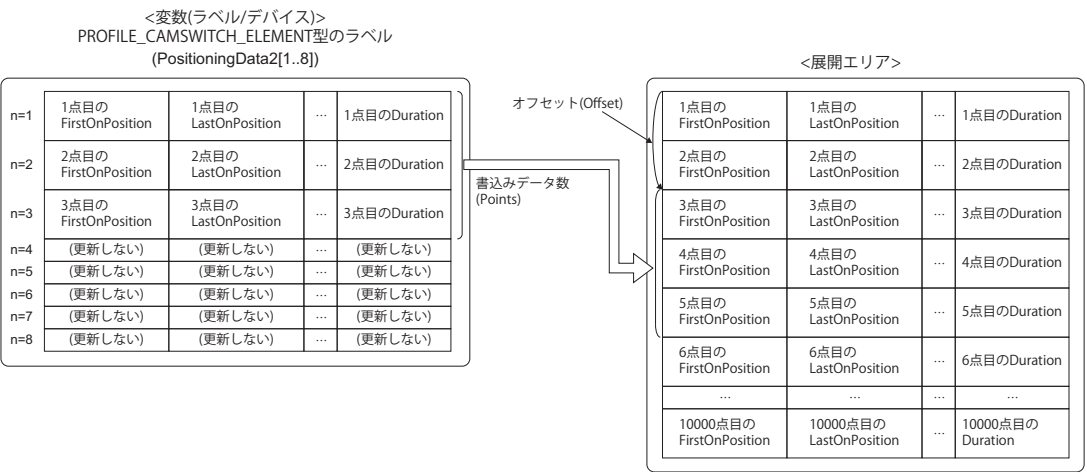
書き込む区間の先頭をオフセット(Offset), 区間数(NumberOfSections)を書込みデータ数(Points)で設定します。

書き込みデータ1(Data1)は、PROFILE_CAMSWITCH_DATA型のデータを指定してください。

書き込みデータ2(Data2)は、スイッチ設定のデータ(FirstOnPosition, LastOnPosition, Direction, CamSwitchMode, Duration)をPROFILE_CAMSWITCH_ELEMENT型の配列で定義したデータを指定してください。

<例>

オフセット(Offset)に「0」、書き込みデータ数(Points)に「3」を設定したときの書き込みデータ2(Data2)の書き込み結果
(書き込みデータ2(Data2)に設定する変数をデータ型「PROFILE_CAMSWITCH_ELEMENT[1..8]」のラベル
「DigitalCamSwitchData2」とした場合)



■書き込みデータ1(Data1)/書き込みデータ2(Data2)

演算プロファイルの書き込みデータをTARGET_REF構造体で設定します。使用できるデータ種別は、[VAR]、[DEV]です。TARGET_REF構造体についての詳細は、下記を参照してください。

📖 1421ページ TARGET_REF(入力信号)

使用できるデータ型は、各演算プロファイル形式の書き込み用データ構造体の型です。

・書き込みデータ1(Data1)

書き込みデータ1(Data1)に設定する演算プロファイル形式、およびデータ型を示します。

演算プロファイル形式	データ型	参照
カムデータ	PROFILE_CAM_DATA	📖 1426ページ PROFILE_CAM_DATA
ロータリーカッター用カムデータ	PROFILE_ROTARY_CUTTER	📖 1428ページ PROFILE_ROTARY_CUTTER
多軸位置決めデータ	PROFILE_POSITIONING_DATA	📖 1430ページ PROFILE_POSITIONING_DATA
デジタルカムスイッチデータ	PROFILE_CAMSWITCH_DATA	📖 1429ページ PROFILE_CAMSWITCH_DATA

例

デバイス、ラベルを使用してPROFILE_CAM_DATA型のラベル(CamData1)を書き込む場合、下記のように記述します。

Data1.Target := "[VAR]CamData1"

・書き込みデータ2(Data2)

書き込みデータ2(Data2)に設定する演算プロファイル形式、およびデータ型を示します。書き込み先(Target)、および対象となるカムデータの補間方法指定(Interpolate)により、設定するデータ型の構造体が異なります。

演算プロファイル全体を書き込みする場合、指定する配列の要素数が(分解能 +1/座標数)より小さい場合は、読出し/書き込みデータ数不一致(エラーコード: 1BACH)となります。

○: 対応, ×: 非対応

演算プロファイル形式	書き込み先(Target)	データ型	補間方法指定(Interpolate)			参照
			0: 直線補間	1: 区間ごとに指定	2: スプライン補間	
カムデータ	0: 展開エリア 2: 展開エリア(モーションサービス処理)	LREAL[J] ^{*1}	○	×	×	—
		PROFILE_CAM_ELEMENT[J] ^{*2}	×	○	○ ^{*3}	📖 1427ページ PROFILE_CAM_ELEMENT
	1: ファイル	LREAL[J] ^{*1}	○	×	×	—
		PROFILE_CAM_ELEMENT[J] ^{*2}	×	○	○ ^{*3}	📖 1427ページ PROFILE_CAM_ELEMENT
ロータリーカッター用カムデータ	設定は不要です。 設定しても無視します。					—
多軸位置決めデータ	—	PROFILE_POSITIONING_DATA_ELEMENT[J] ^{*4}	—			📖 1430ページ PROFILE_POSITIONING_DATA
デジタルカムスイッチデータ	—	PROFILE_CAMSWITCH_ELEMENT[J] ^{*5}	—			📖 1430ページ PROFILE_CAMSWITCH_ELEMENT

*1 LREAL型の2次元配列は、下記のように設定します。
LREAL[m..n,o..p]

項目	内容
要素数(n-m+1)	配列要素が読み出す点数分以上となるように設定します。 演算プロファイル全体を書き込みする場合(オフセット(Offset)、書き込みデータ数(Points)を「0」とした場合)は、対象のカムデータの分解能+1/座標数以上となるように設定してください。 ただし、書き込みデータ1(Data1)にて指定した分解能+1/座標数より大きいエリアのデータは使用しません。
次元数(p-o+1)	各ポイントの1サイクル現在位置とストロークを書き込みます。 配列要素が「2」となるように設定してください。 配列要素を「3」以上とした場合、書き込みは可能ですが、データが前詰めされるため意図した結果が得られない可能性があるため、必ず「2」となるように設定してください。

*2 PROFILE_CAM_ELEMENTの配列は、下記のように設定します。
PROFILE_CAM_ELEMENT[m..n]

項目	内容
要素数(n-m+1)	配列要素が読み出す点数分以上となるように設定します。 演算プロファイル全体を読み出す場合(オフセット(Offset)、読み出しデータ数(Points)を「0」とした場合)は、対象のカムデータの区間数(NumberOfSections)以上となるように設定してください。 ただし、書き込みデータ1(Data1)にて指定した区間数(NumberOfSections)より大きいエリアのデータは使用しません。

*3 終了点、ストロークのみ使用します。

*4 PROFILE_POSITIONING_DATA_ELEMENT[m..n]の配列は、下記のように設定します。

項目	内容
要素数(n-m+1)	配列要素が読み出す点数分以上となるように設定します。 演算プロファイル全体を読み出す場合(オフセット(Offset)、読み出しデータ数(Points)を「0」とした場合)は、配列の要素数が位置決めデータ設定数(PROFILE_POSITIONING_DATA.TotalNumberOfData)以上となるように設定してください。 配列の要素数が少ない場合は、読み出し/書き込みデータ数不一致(エラーコード: 1BACH)となります。

*5 PROFILE_CAMSWITCH_ELEMENT[m..n]の配列は、下記のように設定します。

項目	内容
要素数(n-m+1)	配列要素が読み出す点数分以上となる正数を設定します。 演算プロファイル全体を読み出す場合(オフセット(Offset)、読み出しデータ数(Points)を「0」とした場合)は、デジタルカムスイッチデータのスイッチ数(PROFILE_CAMSWITCH_DATA.NumberOfSwitches)以上となるように設定してください。 Data1で指定したスイッチ数より大きいエリアのデータは使用できません。

例

LREAL型の場合

補間方法指定(Interpolate)	設定例
0: 直線補間	TARGET_REF構造体の対象(Target)に、座標数「100」の2次元デバイス/ラベルを設定する例を下記に示します。 ■デバイス指定 TARGET_REF.Target := "[DEV](LREAL[1..100,1..2])D0" ■ラベル指定 Data2.Target := "[VAR]CamData2" *: CamData2は、グローバルラベルのラベル名に「CamData2」、データ型に「LREAL[1..100,1..2]」のラベルを宣言し、対象(Target)には文字列を指定します。書き込むデータは下記ようになります。 - CamData2[n,1]: nポイント目の1サイクル現在位置 - CamData2[n,2]: nポイント目のストローク
1: 区間ごとに指定	TARGET_REF構造体の対象(Target)に、分解能「100」の2次元デバイス/ラベルを設定する例を下記に示します。 ■デバイス指定 TARGET_REF.Target := "[DEV](LREAL[1..101,1..2])D0" ■ラベル指定 TARGET_REF.Target := "[VAR]CamData2" *: CamData2は、グローバルラベルのラベル名に「CamData2」、データ型に「LREAL[1..101,1..2]」のラベルを宣言し、対象(Target)には文字列を指定します。書き込むデータは下記ようになります。 - CamData2[1,1]: 開始点に設定した値 - CamData2[1,2]: 初期ストローク量の設定した値 - CamData2[n+1,1]: nポイント目の1サイクル現在位置 - CamData2[n+1,2]: nポイント目のストローク
2: スプライン補間	

例

PROFILE_CAM_ELEMENT型の場合

設定例
TARGET_REF構造体の対象(Target)に、区間数「10」の2次元デバイス/ラベルを設定する例を下記に示します。 ■デバイス指定 TARGET_REF.Target := "[DEV](PROFILE_CAM_ELEMENT[1..10])D0" ■ラベル指定 Data2.Target := "[VAR]CamData2" *: CamData2は、グローバルラベルのラベル名に「CamData2」、データ型に「PROFILE_CAM_ELEMENT[1..10]」のラベルを宣言し、対象(Target)には文字列を指定します。書き込むデータは下記ようになります。 CamData2[n]: nポイント目の区間データ(PROFILE_CAM_ELEMENT型)

Point

- 設定した書き込み用データ構造体の型が演算プロファイル形式ごとに要求する型と一致しない場合は、読出し/書き込みデータ型不正(エラーコード: 1BAAH)となります。
- 書き込みデータ1(Data1)/書き込みデータ2(Data2)に設定する演算プロファイルのカムデータについての詳細は、下記を参照してください。
 使用するコントローラのユーザーズマニュアル
- LREAL型の配列でMCv_WriteProfileData(プロファイル書込)の書き込み先(Target)に「1: ファイル」を設定して書き込みした場合、カムデータ形式の補間方法指定(Interpolate)を「0: 直線補間」としてファイルへ出力します。
- [VAR]を使用すると書き込み処理に時間がかかる場合があります。[DEV]を使用することで[VAR]より高速に書き込みができるため、書き込みの処理時間を短縮したい場合は、[DEV]を使用してください。

■書き込み先(Target)

演算プロファイルの書き込み先を設定し、書き込みデータ1(Data1)/書き込みデータ2(Data2)と演算プロファイル展開FBのプロファイルID(PROFILE_ID)に対し、書き込みを行います。

設定値	内容
0: 展開エリア 2: 展開エリア(モーションサービス処理)	- PROFILE_DATA構造体のプロファイルID(ID)の値を参照して、PROFILE_ID構造体のプロファイルID番号(Number)が示す展開エリアの演算プロファイルに対し、書き込みを行います。 ■PROFILE_ID構造体のプロファイルID番号(Number)に「0」を設定した場合 自動で空きIDを割り付け、プロファイルID番号(Number)にIDを格納します。 同一の演算プロファイルに対し、プロファイルID番号(Number)が「0」の展開FBを複数回実行すると、複数の異なる展開エリアに展開します。 空きIDがない場合、演算プロファイルID不足(エラーコード: 1BA0H)となり、展開処理は行いません。 ■PROFILE_ID構造体のプロファイルID番号(Number)に「1~60000」を設定した場合 設定したIDを変えずに展開を行います。 すでに展開済みのIDを設定した場合は展開データを上書きします。
1: ファイル	- PROFILE_DATA構造体のプロファイルID(ID)の値を参照して、PROFILE_ID構造体のプロファイルID番号(Number)が示す展開エリアの演算プロファイルに対し、書き込みを行います。その後、PROFILE_DATA構造体の演算プロファイル格納場所(Location)の値を参照して、FILE_LOCATION構造体のファイル名称(Filename)とフォルダ指定(Path)が示すファイルの演算プロファイルに対し、書き込みを行います。演算プロファイルのプロファイルIDには、出力変数のプロファイルID(ProfileID)の値をファイルに書き込みます。 - 設定したファイルが存在した場合、上書きをします。 - 設定したファイルが存在しない場合、オフセット(Offset)と書き込みデータ数(Points)の両方の値が「0」であれば新規作成し、文字コードは「Unicode(UTF-16LEBOM付き)」で書き込みを行います。 - ファイルを新規作成する場合、オフセット(Offset)と書き込みデータ数(Points)の両方の値が「0」でなければオフセット・読出し/書き込みデータ数不正(エラーコード: 1BADH)となります。(展開エリアへの書き込みは実行します。) 「1: ファイル」を設定した場合、オフセット(Offset)と書き込みデータ数(Points)には、「0」を設定してください。(0以外を設定した場合、オフセット・読出し/書き込みデータ数不正(エラーコード: 1BADH)となります。)

Point

- 読出し先(Target)が「1: ファイル」、「2: 展開エリア(モーションサービス処理)」の場合、モーションサービス処理で実行するため処理が完了するまでに時間がかかる場合があります。必要に応じてDone(実行完了)をインタロックに使用してください。
- Target = 「0: 展開エリア」の場合、FBを呼び出したタスク内で処理を行うため即時完了します。書き込むデータ点数が多いとプログラムの実行時間が延びるため、必要に応じて複数回に分けて書き込みを実行するか、Target = 「2: 展開エリア(モーションサービス処理)」を使用してください。

■起動モード(ExecutionMode)

MCv_WriteProfileData(プロファイル書込)を実行するタイミングを設定します。

設定値	内容
0: 直ちに実行(mclmmediately)	展開エリアの内容を即座に書き込みます。 実行中の制御に影響を及ぼす可能性があります。 ただし、FBの実行中に書き込みを行う場合は、演算プロファイルの形式、分解能が一致していないと、演算プロファイル操作中(エラーコード: 1B90H)となります。
1: 完了待ちして実行(mcQueued)	実行中のFBの実行完了を待ち、書き込みます。(Done = TRUE) 複数のFBが待ち状態となった場合、次に実行するFBは、優先度の高いタスクで実行しているFBとなります。 同じ優先度の場合は、起動順に実行します。
3: 投機的に実行(mcSpeculatively)	演算プロファイル操作中(エラーコード: 1B90H)となり、展開エリアは変更しません。

Point

実行中のFBがない場合は、起動モード(ExecutionMode)の設定によらず即座に実行します。
展開エリアにアクセスする各FB実行中は、各FBの実行中(Busy)がTRUEになります。必要に応じて起動モード(ExecutionMode)をインタロックに使用してください。

注意事項

- 一度に書き込みできるデータ点数には限りがあります。1回のFB実行で書き込みが完了しないときは、複数回に分けて書き込みを実行してください。

プログラム例

「LREAL型(2次元配列)の形式」のカムデータ1(ProfileData0001)と「PROFILE_CAM_ELEMENT型の形式」のカムデータ2(ProfileData0002)をファイルに書き込むプログラム例を下記に示します。

■演算プロファイル

プロファイルID	ラベル名	データ型	コメント
1	ProfileData0001	MC_CAM_REF	カムデータ1
2	ProfileData0002	MC_CAM_REF	カムデータ2

• 演算プロファイル(カムデータ)の設定

項目	ProfileData0001	ProfileData0002
補間方法指定	直線補間	区間ごとに指定
分解能	—	256
1サイクル長設定	単位	degree
	1サイクル長	360.00000
ストローク量	単位	%
	ストローク量	100.0000000
カム1サイクル時間	—	1.000[s]
初期ストローク値	—	0.0000000
カム1サイクル最小値	0.00000	—
カム1サイクル最大値	360.00000	—

• ProfileData0001

区間No.	入力値[degree]	出力値[pulse]
1	0.00000	0.0
2	90.00000	60.0
3	180.00000	100.0
4	270.00000	30.0
5	360.00000	0.0

• ProfileData0002

区間No.	開始点[degree]	終了点[degree]	ストローク[%]	カム曲線種別
1	0.00000	90.00000	60.0000000	単弦
2	90.00000	180.00000	100.0000000	単弦
3	180.00000	270.00000	30.0000000	単弦
4	270.00000	0.00000	0.0000000	単弦

■使用するラベル

ラベル名	データ型	コメント
MCv_WriteProfileData_1	MCv_WriteProfileData	プロファイル書込FB 1
WriteData1_1	TARGET_REF	書込みデータ1 プロファイル1
WriteData2_1	TARGET_REF	書込みデータ2 プロファイル1
WriteData1_Lreal	PROFILE_CAM_DATA	プロファイルデータ1 カムデータ1
WriteData2_Lreal	倍精度実数(0..4,0..1)	プロファイルデータ1 LREAL型
bWriteProfile1	ビット	プロファイル書込指令1
bDone1	ビット	完了1
bBusy1	ビット	実行中1
bError1	ビット	エラー 1
uwErrorID1	ワード[符号なし]/ビット列[16ビット]	エラーコード1
ProfileID1	PROFILE_ID	プロファイルID1
Cam_ProfileData0001*1	PROFILE_DATA	プロファイルデータ0001
MCv_WriteProfileData_2	MCv_WriteProfileData	プロファイル書込FB 2
WriteData1_2	TARGET_REF	書込みデータ1 プロファイル2
WriteData2_2	TARGET_REF	書込みデータ2 プロファイル2
WriteData1_Element	PROFILE_CAM_DATA	プロファイルデータ2 カムデータ2
WriteData2_Element	PROFILE_CAM_ELEMENT(0..3)	プロファイルデータ2 PROFILE_CAM_ELEMENT型
bWriteProfile2	ビット	プロファイル書込指令2
bDone2	ビット	完了2
bBusy2	ビット	実行中2
bError2	ビット	エラー 2
uwErrorID2	ワード[符号なし]/ビット列[16ビット]	エラーコード2
ProfileID2	PROFILE_ID	プロファイルID2
Cam_ProfileData0002*1	PROFILE_DATA	プロファイルデータ0002

*1 FBD/LDのプログラムのみ使用(演算プロファイルをFBD/LDのプログラムで使用するために必要)

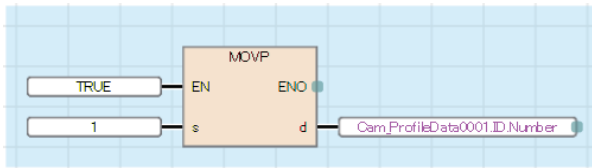
■使用するグローバルラベル

ラベル名	データ型	コメント	公開ラベル
G_ProfileCamData*1	PROFILE_CAM_DATA	プロファイルカムデータ	有効
G_ProfileCamELEMENT*1	PROFILE_CAM_ELEMENT	プロファイルカムエレメント	有効

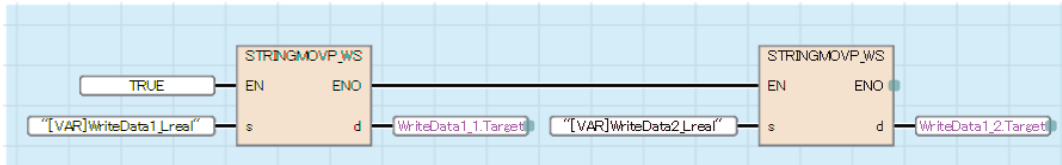
*1 FBD/LDのプログラムのみ使用(演算プロファイルをFBD/LDのプログラムで使用するために必要)

■FBD/LDのプログラム

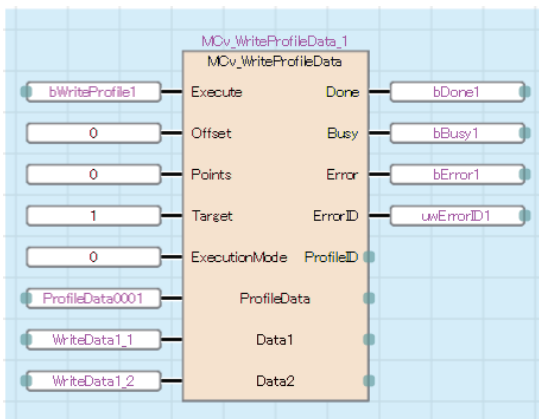
- プロファイルデータ0001



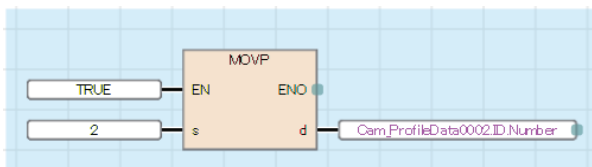
- 書き込みデータ1



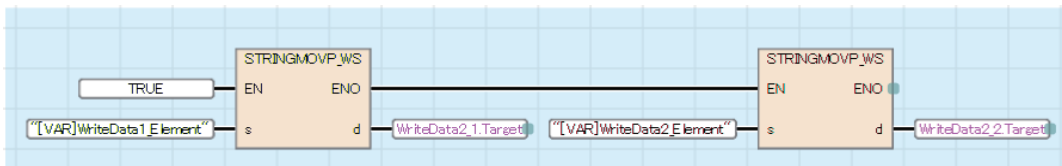
- プロファイル書込1(LREAL型(2次元配列)の形式でカムデータを書込み)



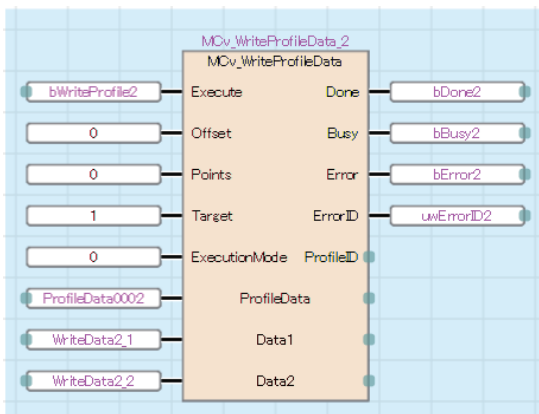
- プロファイルデータ0002



- 書き込みデータ2



- プロファイル書込2(PROFILE_CAM_ELEMENT型の形式でカムデータを書込み)



■STのプログラム

```
//-----書き込みデータ1-----
WriteData1_1.Target := "[VAR]WriteData1_Lreal";
WriteData1_2.Target := "[VAR]WriteData2_Lreal";

//-----プロファイル書込1-----
//-----LREAL型(2次元配列)の形式でカムデータを書込み-----
MCv_WriteProfileData_1(
    ProfileData:= ProfileData0001.ProfileData ,
    Data1:= WriteData1_1 ,
    Data2:= WriteData1_2 ,
    Execute:= bWriteProfile1 ,
    Offset:= 0 ,
    Points:= 0 ,
    Target:= 1 ,
    ExecutionMode:= MC_EXECUTION_MODE__mclImmediately ,
    Done=> bDone1 ,
    Busy=> bBusy1 ,
    Error=> bError1 ,
    ErrorID=> uwErrorID1 ,
    ProfileID=> ProfileID1
);

//-----書き込みデータ2-----
WriteData2_1.Target := "[VAR]WriteData1_Element";
WriteData2_2.Target := "[VAR]WriteData2_Element";

//-----プロファイル書込2-----
//-----PROFILE_CAM_ELEMENT型の形式でカムデータを書込み-----
MCv_WriteProfileData_2(
    ProfileData:= ProfileData0002.ProfileData ,
    Data1:= WriteData2_1 ,
    Data2:= WriteData2_2 ,
    Execute:= bWriteProfile2 ,
    Offset:= 0 ,
    Points:= 0 ,
    Target:= 1 ,
    ExecutionMode:= MC_EXECUTION_MODE__mclImmediately ,
    Done=> bDone2 ,
    Busy=> bBusy2 ,
    Error=> bError2 ,
    ErrorID=> uwErrorID2 ,
    ProfileID=> ProfileID2
);
```


第9部

ネットワークファンク
ションブロック

この部は下記の章構成となります。

44 Ethernet用ファンクションブロック

45 CC-Link IE TSN用ファンクションブロック

44 Ethernet用ファンクションブロック

Point

本章のファンクションブロックはコントローラのユニットFBです。エンジニアリングツールの部品選択ウィンドウの[ユニット]⇒[ユニットFB]から使用してください。

44.1 コネクションの確立/切断

M+MX-SQ_ConnectionOpen

コネクションをオープン(確立)します。

暗号化通信プロトコル(TLSとDTLS)を指定する場合は、相手機器も対応している必要があります。

ラダー	FBD/LD
M+MX-SQ_ConnectionOpen B: i_bEN o_bENO: B DUT: i_stModule o_bOK: B UW: i_uConnectionNo o_bErr: B o_uErrId: UW pbi_bUseParameters 0 pbi_uProtocol 0 pbi_uOpen_System 0 pbi_uPredefProtocol 0 pbi_uLocal_Port_No 4096 pbi_uTarget_Port_No 4096 pbi_u2IP_Address 0	M+MX-SQ_ConnectionOpen i_bEN o_bENO i_stModule o_bOK i_uConnectionNo o_bErr o_uErrId

ST

```
M_MX_SQ_ConnectionOpen(  
  i_bEN:= ?BOOL?,  
  i_stModule:= ?M+MX_SQ?,  
  i_uConnectionNo:= ?WORD?,  
  o_bENO=> ?BOOL?,  
  o_bOK=> ?BOOL?,  
  o_bErr=> ?BOOL?,  
  o_uErrId=> ?WORD?  
);
```

設定データ

■入力引数

変数名	名称	データ型	有効範囲	内容
i_bEN	実行指令	ビット	—	ON: FBを起動します。 OFF: FBを起動しません。
i_stModule	ユニットラベル	構造体	—	FBを実行するコントローラのユニットラベルを指定します。 (例: MX_SQ)
i_uConnectionNo	コネクションNo.	ワード[符号なし]/ ビット列[16ビット]	1~64	オープンするコネクションの番号を指定します。

■出力引数

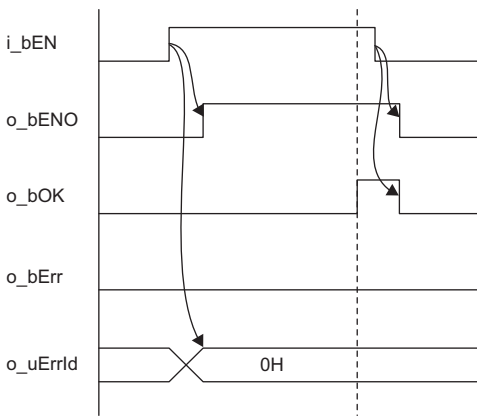
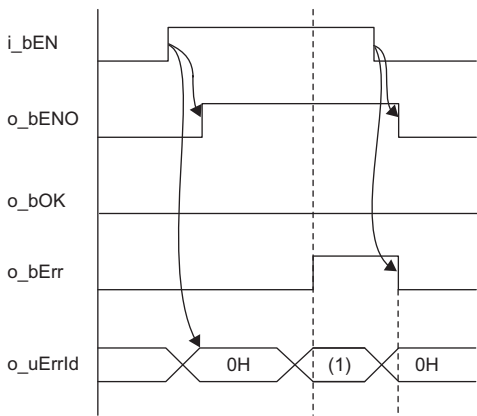
変数名	名称	データ型	内容	デフォルト値
o_bENO	実行状態	ビット	ON: 実行指令ON OFF: 実行指令OFF	OFF
o_bOK	正常完了	ビット	ONの場合、FBの処理が正常完了したことを示します。	OFF
o_bErr	異常完了	ビット	ONの場合、FBの処理が異常完了したことを示します。	OFF
o_uErrId	エラーコード	ワード[符号なし]/ ビット列[16ビット]	異常完了時にエラーコードが格納されます。	0

■動作パラメータ

変数名	名称	データ型	有効範囲	内容	デフォルト値
pbi_bUseParameters	使用パラメータ	ビット	ON, OFF	コネクションのオープン処理時に、エンジニアリングツールによるパラメータ設定値を使用するか、下記の動作パラメータの設定値を使用するかを選択します。 • OFF: エンジニアリングツールの相手機器接続構成設定で設定した内容でオープン処理を行う。(下記の動作パラメータの値は設定する必要はありません。設定しても値は無視されます。) • ON: 下記の動作パラメータに設定した内容でオープン処理を行う。	OFF
pbi_uProtocol	プロトコル	ワード[符号なし]/ビット列[16ビット]	0~3	オープンするコネクションのプロトコルを選択します。 • pbi_uPredefProtocol(通信プロトコル設定)が0の場合 0: TCP 1: UDP 2: TLS 3: DTLS • pbi_uPredefProtocol(通信プロトコル設定)が1の場合 0: TCP 1: UDP	0
pbi_uOpen_System	オープン方式	ワード[符号なし]/ビット列[16ビット]	0~2	コネクションのオープン方式を選択します。 • 0: ActiveオープンまたはUDPまたはDTLS • 1: Unpassiveオープン • 2: Fullpassiveオープン	0
pbi_uPredefProtocol	通信プロトコル設定	ワード[符号なし]/ビット列[16ビット]	0~1	通信プロトコル機能の使用有無を選択します。 • 0: 通信プロトコル支援機能を使用しない。(ソケット通信機能を使用する。) • 1: 通信プロトコル支援機能を使用する。	0
pbi_uLocal_Port_No	自ノードポート番号	ワード[符号なし]/ビット列[16ビット]	1~4999, 5010~65534	自ノードのポート番号を指定します。 ポート番号1~1023は一般的に予約されたポート番号(WELL KNOWN PORT NUMBERS)に割り当てられているため、ポート番号1024~4999, 5010~65534の使用を推奨します。	4096
pbi_uTarget_Port_No	相手機器ポート番号	ワード[符号なし]/ビット列[16ビット]	1~65534, 65535	相手機器のポート番号を指定します。 65535を指定したコネクション(プロトコルでUDPまたはDTLSを選択した場合のみ)では、全ポート番号を対象に受信を行います。65535を指定したコネクションではデータを送信できないため、データを送信する場合は1~65534を指定してください。	4096
pbi_u2IP_Address	相手機器IPアドレス	ワード[符号なし]/ビット列[16ビット](0..1)	0.0.0.1~255.255.255.255(00000001H~FFFFFFFH)	相手機器のIPアドレスを指定します。 1ワード目に第3~4オクテット、2ワード目に第1~2オクテットを指定します。 一斉通報通信を行う場合は、255.255.255.255(FFFFFFFH)を指定します。 b15b8b7b0 +0 (3)(4) +1 (1)(2) (1)~(4): IPアドレスのオクテットを示します。	192.168.1.1(C0A80101H)

機能

項目	内容
対象機器	対象ユニット
	対象CPU
	対象エンジニアリングツール
使用言語	ラダー
ラベル使用量	ラベル: 36点 ラッチラベル: 0点 プログラムに組み込んだラベル使用量は、使用するコントローラ、引数に指定したデバイスやGX Works3のオプション設定によって異なります。GX Works3のオプション設定については、下記を参照してください。  GX Works3 オペレーティングマニュアル

項目	内容
機能説明	i_bEN(実行指令)のONで、相手機器とのデータ交信用の接続をオープンします。
FBコンパイル方式	マクロ型
FB動作	パルス型(複数スキャン実行型)
入出力信号の動き	・正常完了の場合  ・異常完了の場合(ユニットエラーの場合も同様)  (1) エラーコード
注意事項	■FBの仕様について - 他のユニットFBまたは専用命令が使用中の接続に対して、本ユニットFBを実行できません。使用中の接続に対して本ユニットFBを実行した場合はエラーとなります。 - pbi_bUseParameters(使用パラメータ)をONにして動作パラメータの内容でオープン処理を行う場合、使用できる通信手段はソケット通信および通信プロトコルのみです。 - 相手機器接続構成設定でパラメータを設定済みの接続に対して本FBを実行した場合は、本FBで指定した動作パラメータに従いオープン処理を行います。 - 本FBは、プログラム別ラベル初期値を使用します。コントローラのブート運転において、本FBを使用したプログラムファイルをブートファイル設定で指定する場合、同様にプログラム別ラベル初期値ファイルもブートファイル設定で指定してください。なお、1878ページエラーコードに記載されていないエラーコードが発生した場合は、ブートファイル設定でプログラム別ラベル初期値ファイルが設定されていない可能性があります。ブートファイル設定でプログラム別ラベル初期値ファイルを指定してください。 - 本FBを実行した場合に、コントローラで演算異常が発生する場合があります。その場合は、イベント履歴の演算異常の詳細情報を確認して、該当ユニットFBの入力引数を見直してください。 ■FBの動作について - i_bEN(実行指令)は、o_bOK(正常完了)またはo_bErr(異常完了)がONした後にOFFさせてください。i_bEN(実行指令)のOFFによってo_bOK(正常完了)とo_bErr(異常完了)がOFFし、o_uErrld(エラーコード)が0クリアされます。

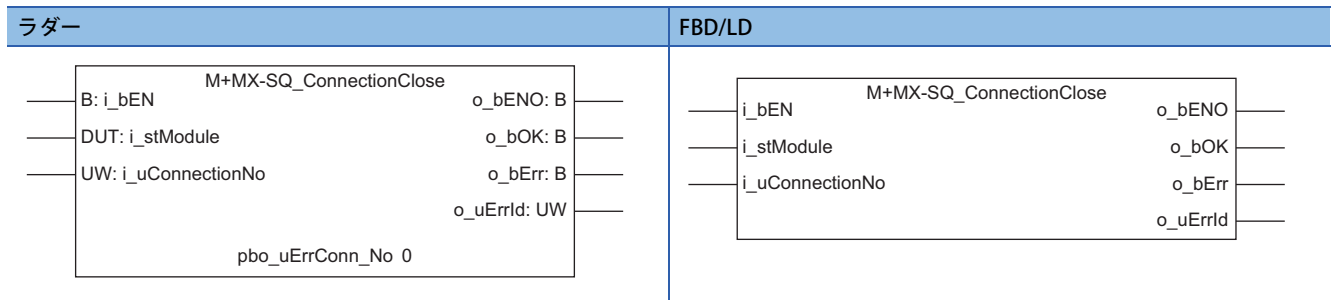
■エラーコード

本FBで発生するエラーコードはSOCOPEN命令と同じです。

📖 1084ページ GP.SOCOPEN, SP.SOCOPEN

M+MX-SQ_ConnectionClose

コネクションをクローズ(切断)します。



ST

```

M_MX_SQ_ConnectionClose(
  i_bEN:= ?BOOL?,
  i_stModule:= ?M+MX_SQ?,
  i_uConnectionNo:= ?WORD?,
  o_bENO=> ?BOOL?,
  o_bOK=> ?BOOL?,
  o_bErr=> ?BOOL?,
  o_uErrId=> ?WORD?
);

```

設定データ

■入力引数

変数名	名称	データ型	有効範囲	内容
i_bEN	実行指令	ビット	—	ON: FBを起動します。 OFF: FBを起動しません。
i_stModule	ユニットラベル	構造体	—	FBを実行するコントローラのユニットラベルを指定します。 (例: MX_SQ)
i_uConnectionNo	コネクションNo.	ワード[符号なし]/ ビット列[16ビット]	1~64, 65535	クローズするコネクションの番号を指定します。 65535(FFFFH)を指定した場合は、すべてのコネクションをクローズ します。

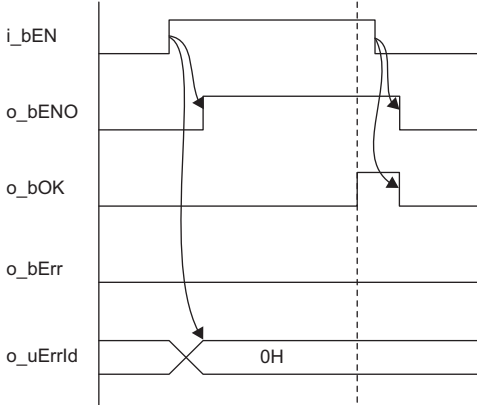
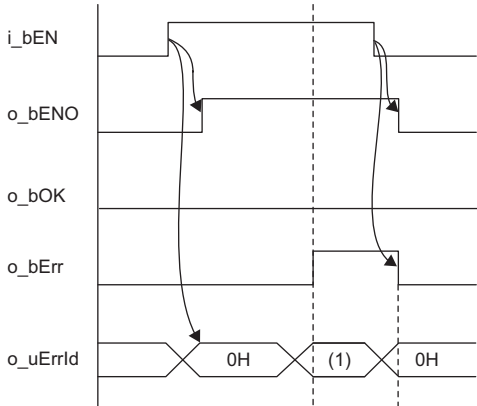
■出力引数

変数名	名称	データ型	内容	デフォルト値
o_bENO	実行状態	ビット	ON: 実行指令ON OFF: 実行指令OFF	OFF
o_bOK	正常完了	ビット	ONの場合、FBの処理が正常完了したことを示します。	OFF
o_bErr	異常完了	ビット	ONの場合、FBの処理が異常完了したことを示します。	OFF
o_uErrId	エラーコード	ワード[符号なし]/ ビット列[16ビット]	異常完了時にエラーコードが格納されます。	0

■公開変数

変数名	名称	データ型	内容	デフォルト値
pbo_uErrConn_No	異常発生コネクション No.	ワード[符号なし]/ ビット列[16ビット]	クローズ処理が異常完了したコネクションの番号が格納されます。 コネクションNo.(i_uConnectionNo)で65535(FFFFH)を指定した場 合は、最初にクローズ処理が異常完了したコネクションの番号が 格納されます。	0

機能

項目	内容	
対象機器	対象ユニット	MELSEC MXコントローラ(Ethernet用ポート)
	対象CPU	MELSEC MXコントローラ
	対象エンジニアリングツール	GX Works3
使用言語	ラダー	
ラベル使用量	ラベル: 28点 ラッチラベル: 0点 プログラムに組み込んだラベル使用量は、使用するコントローラ、引数に指定したデバイスやGX Works3のオプション設定によって異なります。GX Works3のオプション設定については、下記を参照してください。 GX Works3 オペレーティングマニュアル	
機能説明	- i_bEN(実行指令)のONで、相手機器とのデータ交信用の接続をクローズします。 - 入力引数の接続No.(i_uConnectionNo)に65535(FFFFH)を指定した場合は、すべての接続をクローズします。 - クローズ対象に指定した接続を1つでもクローズできなかった場合は、異常完了します。	
FBコンパイル方式	マクロ型	
FB動作	パルス型(複数スキャン実行型)	
入出力信号の動き	- 正常完了の場合  - 異常完了の場合(ユニットエラーの場合も同様)  (1) エラーコード	
注意事項	- 他のユニットFBまたは専用命令が使用中の接続に対して、本ユニットFBを実行できません。使用中の接続に対して本ユニットFBを実行した場合はエラーとなります。 - i_bEN(実行指令)は、o_bOK(正常完了)またはo_bErr(異常完了)がONした後にOFFさせてください。i_bEN(実行指令)のOFFによってo_bOK(正常完了)とo_bErr(異常完了)がOFFし、o_uErrId(エラーコード)が0クリアされます。 - 本FBを実行した場合に、コントローラで演算異常が発生する場合があります。その場合は、イベント履歴の演算異常の詳細情報を確認して、該当ユニットFBの入力引数を見直してください。	

エラー

本FBで発生するエラーコードはSOCCLOSE命令と同じです。

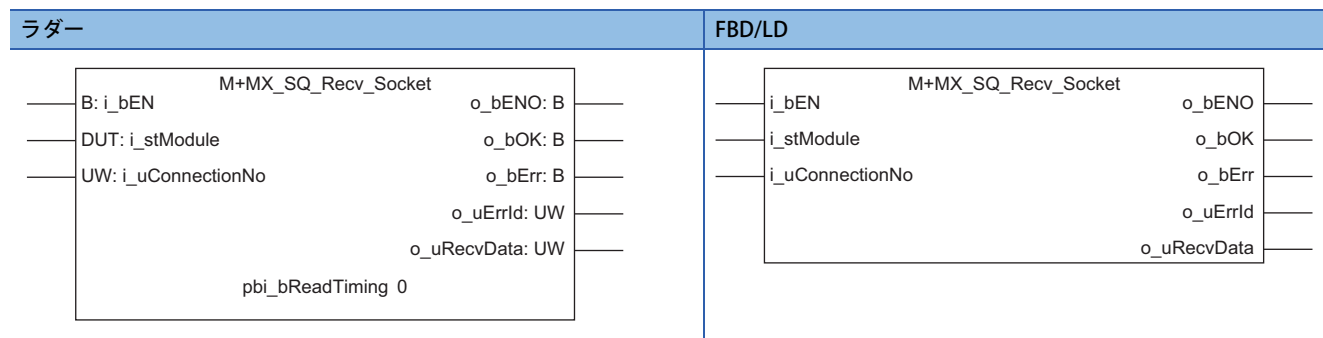
📖 1088ページ GP.SOCCLOSE, SP.SOCCLOSE

44.2 ソケット通信

44

M+MX-SQ_Recv_Socket

指定したコネクションにソケット通信で受信しているデータを読み出します。



ST

```
M_MX_SQ_Recv_Socket(  
  i_bEN:= ?BOOL?,  
  i_stModule:= ?M+MX_SQ?,  
  i_uConnectionNo:= ?WORD?,  
  o_bENO=> ?BOOL?,  
  o_bOK=> ?BOOL?,  
  o_bErr=> ?BOOL?,  
  o_uErrId:= ?WORD?,  
  o_uRecvData=> ?WORD?  
);
```

設定データ

■入力引数

変数名	名称	データ型	有効範囲	内容
i_bEN	実行指令	ビット	—	ON: FBを起動します。 OFF: FBを起動しません。
i_stModule	ユニットラベル	構造体	—	FBを実行するコントローラのユニットラベルを指定します。 (例: MX_SQ)
i_uConnectionNo	コネクションNo.	ワード[符号なし]/ ビット列[16ビット]	1~64	受信するコネクションの番号を指定します。

■出力引数

変数名	名称	データ型	内容	デフォルト値
o_bENO	実行状態	ビット	ON: 実行指令ON OFF: 実行指令OFF	OFF
o_bOK	正常完了	ビット	ONの場合、FBの処理が正常完了したことを示します。	OFF
o_bErr	異常完了	ビット	ONの場合、FBの処理が異常完了したことを示します。	OFF
o_uErrId	エラーコード	ワード[符号なし]/ ビット列[16ビット]	異常完了時にエラーコードが格納されます。	0

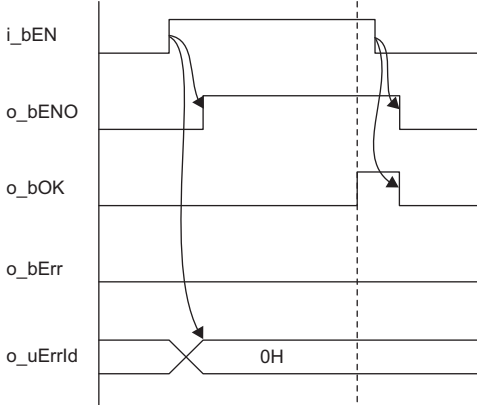
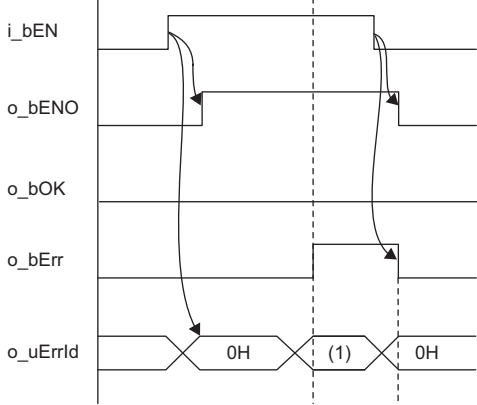
変数名	名称	データ型	内容	デフォルト値						
o_uRecvData	受信データ格納先	ワード[符号なし]/ ビット列[16ビット]	受信データ長と、受信データを格納するデバイスの先頭番号を指定します。下記のように、読み出したデータが若いアドレスから順次格納されます。 - データ単位がバイトの場合 1ワード目: 受信データ長(単位: バイト) 2~nワード目: b15...b8 b7...b0 <table><tr><td>(2)</td><td>(1)</td></tr><tr><td colspan="2">⋮</td></tr><tr><td>(4)</td><td>(3)</td></tr></table> (1)受信データ1 (2)受信データ2 (3)受信データm-1 (4)受信データm 受信データのデータ形式や単位、データ長の範囲は、ユニットの種類やコネクション番号により異なります。 受信データは、ワードエリアの前半(b0~7)→後半(b8~15)の順に格納されます。 引数として下記を指定できません。指定すると、エラー (2820H: デバイス・ラベル・バッファメモリ指定不正)が発生する場合があります。 - 動的指定した配列要素(例: wLabel[D0]) - 桁指定したラベル(例: K4bLabel) - 間接指定(例: @W0) - ローカルデバイス指定(例: #D0)	(2)	(1)	⋮		(4)	(3)	0
(2)	(1)									
⋮										
(4)	(3)									

■動作パラメータ

変数名	名称	データ型	有効範囲	内容	デフォルト値
pbi_bReadTiming	読出しタイミング	ビット	ON, OFF	データ読出し処理を実行するタイミングを指定します。 - OFF: ユニットFB起動後、すぐに読出しを開始する。 - ON: ユニットFB起動後、最初のEND処理時に読み出す。	OFF

機能

項目	内容
対象機器	対象ユニット
	MELSEC MXコントローラ(Ethernet用ポート)
	対象CPU
	MELSEC MXコントローラ
	対象エンジニアリングツール
	GX Works3
使用言語	ラダー
ラベル使用量	ラベル: 24点 ラッチラベル: 0点 プログラムに組み込んだラベル使用量は、使用するコントローラ、引数に指定したデバイスやGX Works3のオプション設定によって異なります。GX Works3のオプション設定については、下記を参照してください。  GX Works3 オペレーティングマニュアル
機能説明	i_bEN(実行指令)のONで、入力引数で指定したコネクションに受信しているデータを読み出します。
FBコンパイル方式	マクロ型
FB動作	パルス型(複数スキャン実行型)

項目	内容
入出力信号の動き	• 正常完了の場合  • 異常完了の場合(ユニットエラーの場合も同様)  (1) エラーコード
注意事項	- 他のユニットFBまたは専用命令が使用中のコネクションに対して、本ユニットFBを実行できません。使用中のコネクションに対して本ユニットFBを実行した場合はエラーとなります。 - i_bEN(実行指令)は、o_bOK(正常完了)またはo_bErr(異常完了)がONした後にOFFさせてください。i_bEN(実行指令)のOFFによってo_bOK(正常完了)とo_bErr(異常完了)がOFFし、o_uErrld(エラーコード)が0クリアされます。 - 本FBの実行指令は任意のタイミングで実行できますが、データ受信後に実行する場合は、G7889~G7892(ソケット通信受信状態信号)または該当するユニットラベルを、実行指令の条件に加える必要があります。 - 動作/パラメータの読出しタイミングでON(ユニットFB起動後、最初のEND処理時に読み出す)を指定して実行すると、本ユニットFBでは1回のEND処理内でデータの読出し処理を完了させるため、スキャンタイムが延びます。 - 本FBは、プログラム別ラベル初期値を使用します。コントローラのブート運転において、本FBを使用したプログラムファイルをブートファイル設定で指定する場合、同様にプログラム別ラベル初期値ファイルもブートファイル設定で指定してください。なお、1883ページ エラーコードに記載されていないエラーコードが発生した場合は、ブートファイル設定でプログラム別ラベル初期値ファイルが設定されていない可能性があります。ブートファイル設定でプログラム別ラベル初期値ファイルを指定してください。 - 本FBを実行した場合に、コントローラで演算異常が発生する場合があります。その場合は、イベント履歴の演算異常の詳細情報を確認して、該当ユニットFBの入力引数を見直してください。

■エラーコード

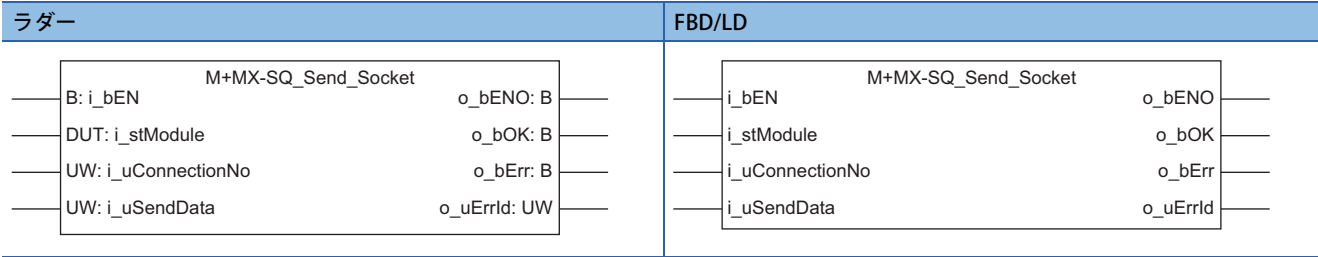
本FBで発生するエラーコードはSOCRCV命令、SOCRCVS命令と同じです。

📖 1090ページ GP.SOCRCV, SP.SOCRCV

📖 1094ページ G.SOCRCVS, S.SOCRCVS

M+MX-SQ_Send_Socket

指定した接続の相手機器へソケット通信でデータを送信します。



ST

```
M_MX_SQ_Send_Socket(  
  i_bEN:= ?BOOL? ,  
  i_stModule:= ?M+MX_SQ? ,  
  i_uConnectionNo:= ?WORD? ,  
  i_uSendData:= ?WORD? ,  
  o_bENO=> ?BOOL? ,  
  o_bOK=> ?BOOL? ,  
  o_bErr=> ?BOOL? ,  
  o_uErrId=> ?WORD?  
);
```

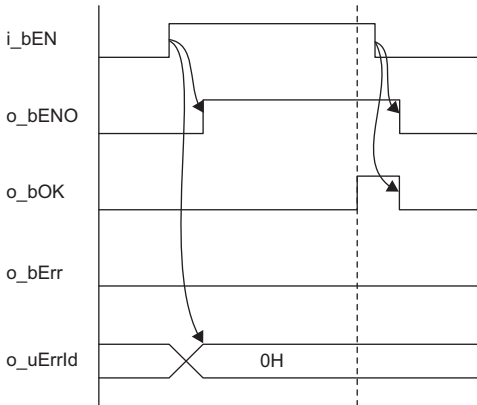
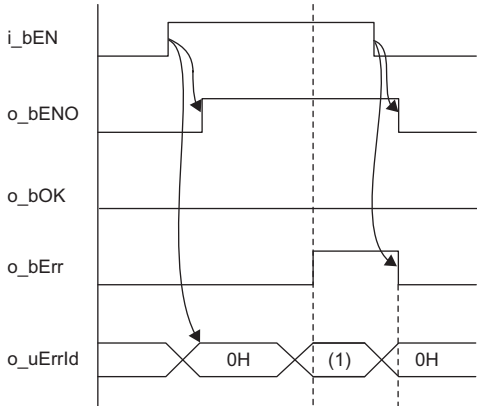
■入力引数

変数名	名称	データ型	有効範囲	内容								
i_bEN	実行指令	ビット	—	ON: FBを起動します。 OFF: FBを起動しません。								
i_stModule	ユニットラベル	構造体	—	FBを実行するコントローラのユニットラベルを指定します。 (例: MX_SQ)								
i_uConnectionNo	コネクションNo.	ワード[符号なし]/ ビット列[16ビット]	1~64	送信するコネクションの番号を指定します。								
i_uSendData	送信データ格納先	ワード[符号なし]/ ビット列[16ビット]	—	送信データ長と、送信するデータが格納されているデバイスの先頭番号を指定します。 ・データ単位がバイトの場合 1ワード目: 送信データ長(単位: バイト) 2~nワード目: <table><tr><td>b15・・・b8</td><td>b7・・・b0</td></tr><tr><td>(2)</td><td>(1)</td></tr><tr><td colspan="2">⋮</td></tr><tr><td>(4)</td><td>(3)</td></tr></table> (1)送信データ1 (2)送信データ2 (3)送信データm-1 (4)送信データm 送信データのデータ形式やデータ長の範囲は、ユニットの種類や使用するコネクションの設定により異なります。 ワードエリアの前半(b0~7)→後半(b8~15)の順にデータが送信されます。 引数として下記を指定できません。指定すると、エラー (2820H: デバイス・ラベル・バッファメモリ指定不正)が発生する場合があります。 ・動的指定した配列要素(例: wLabel[D0]) ・桁指定したラベル(例: K4bLabel) ・間接指定(例: @W0) ・ローカルデバイス指定(例: #D0)	b15・・・b8	b7・・・b0	(2)	(1)	⋮		(4)	(3)
b15・・・b8	b7・・・b0											
(2)	(1)											
⋮												
(4)	(3)											

■出力引数

変数名	名称	データ型	内容	デフォルト値
o_bENO	実行状態	ビット	ON: 実行指令ON OFF: 実行指令OFF	OFF
o_bOK	正常完了	ビット	ONの場合、FBの処理が正常完了したことを示します。	OFF
o_bErr	異常完了	ビット	ONの場合、FBの処理が異常完了したことを示します。	OFF
o_uErrId	エラーコード	ワード[符号なし]/ ビット列[16ビット]	異常完了時にエラーコードが格納されます。	0

機能

項目	内容	
対象機器	対象ユニット	MELSEC MXコントローラ(Ethernet用ポート)
	対象CPU	MELSEC MXコントローラ
	対象エンジニアリングツール	GX Works3
使用言語	ラダー	
ラベル使用量	ラベル: 24点 ラッチラベル: 0点 プログラムに組み込んだラベル使用量は、使用するコントローラ、引数に指定したデバイスやGX Works3のオプション設定によって異なります。GX Works3のオプション設定については、下記を参照してください。 GX Works3 オペレーティングマニュアル	
機能説明	i_bEN(実行指令)のONで、入力引数で指定した接続の相手機器へデータを送信します。	
FBコンパイル方式	マクロ型	
FB動作	パルス型(複数スキャン実行型)	
入出力信号の動き	・正常完了の場合  ・異常完了の場合(ユニットエラーの場合も同様)  (1) エラーコード	
注意事項	- 他のユニットFBまたは専用命令が使用中の接続に対して、本ユニットFBを実行できません。使用中の接続に対して本ユニットFBを実行した場合はエラーとなります。 - i_bEN(実行指令)は、o_bOK(正常完了)またはo_bErr(異常完了)がONした後にOFFさせてください。i_bEN(実行指令)のOFFによってo_bOK(正常完了)とo_bErr(異常完了)がOFFし、o_uErrld(エラーコード)が0クリアされます。 - 本FBを実行した場合に、コントローラで演算異常が発生する場合があります。その場合は、イベント履歴の演算異常の詳細情報を確認して、該当ユニットFBの入力引数を見直してください。	

エラー

本FBで発生するエラーコードはSOCSND命令と同じです。

1097ページ GP.SOCSND, SP.SOCSND

44.3 SLMP

M+MX-SQ_SLMP_DeviceRead_IP

IPアドレス指定でSLMP対応機器のデバイスデータを読み出します。
相手機器は、SLMPコマンド(Device Read)に対応している必要があります。
暗号化通信を行う場合は、相手機器が上記に加え、暗号化通信プロトコル(DTLS)に対応している必要があります。

ラダー	FBD/LD
M+MX-SQ_SLMP_DeviceRead_IP B: i_bENo_bENO: B DUT: i_stModuleo_bOK: B UW: i_u2IP_Addresso_bErr: B UW: i_uSubCommando_uErrId: UW UW: i_uDeviceCodeo_uReadData:UW UW: i_u2DeviceNo UW: i_uDevicePoints UW: i_uChannel UW: i_uTarget_Port_No pbi_uRequestModuleIO H3FF pbi_uResendCountMax 0 pbi_uMonitorTime 0 pbi_bEncryptionComm 0 pbi_uLocal_Port_No 4096 pbo_uResendCount 0 pbo_u4ErrTime 0 pbo_uErrIP_Address_3rd_4th 0 pbo_uErrIP_Address_1st_2nd 0	M+MX-SQ_SLMP_DeviceRead_IP i_bENo_bENO i_stModuleo_bOK i_u2IP_Addresso_bErr i_uSubCommando_uErrId i_uDeviceCodeo_uReadData i_u2DeviceNo i_uDevicePoints i_uChannel i_uTarget_Port_No

ST
<pre>M_MX_SQ_SLMP_DeviceRead_IP(i_bEN:= ?BOOL?, i_stModule:= ?M+MX_SQ?, i_u2IP_Address:= ?WORD(0..1)?, i_uSubCommand:= ?WORD?, i_uDeviceCode:= ?WORD?, i_u2DeviceNo:= ?WORD(0..1)?, i_uDevicePoints:= ?WORD?, i_uChannel:= ?WORD?, i_uTarget_Port_No:= ?WORD?, o_bENO=> ?BOOL?, o_bOK=> ?BOOL?, o_bErr=> ?BOOL?, o_uErrId=> ?WORD?, o_uReadData=> ?WORD?);</pre>

設定データ

■入力引数

変数名	名称	データ型	有効範囲	内容
i_bEN	実行指令	ビット	—	ON: FBを起動します。 OFF: FBを起動しません。
i_stModule	ユニットラベル	構造体	—	FBを実行するコントローラのユニットラベルを指定します。 (例: MX_SQ)
i_u2IP_Address	相手機器IPアドレス	ワード[符号なし]/ビット列 [16ビット](0..1)	0.0.0.1~253.255.255.254(00000001H~DFFFFFFEH)	相手機器のIPアドレスを指定します。 1ワード目に第3~4オクテット, 2ワード目に第1~2オクテットを指定します。 ただし, 第4オクテットが0または255(FFH)となる設定はできません。 b15b8b7b0 +0 (3)(4) +1 (1)(2) (1)~(4): IPアドレスのオクテットを示します。
i_uSubCommand	サブコマンド	ワード[符号なし]/ビット列 [16ビット]	—	読み出し単位およびデバイスの指定方法を指定します。 ■0ビット目: 読み出し単位 0: ワード単位 1: ビット単位 ■1ビット目: 読み出すデバイスの指定方法 0: デバイスコードを2桁, 先頭デバイス番号を6桁で指定(MELSEC-Q/Lシリーズ用) 1: デバイスコードを4桁, 先頭デバイス番号を8桁で指定(MELSEC iQ-Rシリーズ用)
i_uDeviceCode	デバイスコード*1	ワード[符号なし]/ビット列 [16ビット]	—	読み出すデバイスのデバイスコードをバイナリコードで指定します。 ・サブコマンドの1ビット目が0の場合: 2桁 ・サブコマンドの1ビット目が1の場合: 4桁
i_u2DeviceNo	先頭デバイス番号	ワード[符号なし]/ビット列 [16ビット](0..1)	—	読み出すデバイスの先頭デバイス番号をバイナリコードで指定します。 ・サブコマンドの1ビット目が0の場合: 6桁 ・サブコマンドの1ビット目が1の場合: 8桁
i_uDevicePoints	デバイス点数	ワード[符号なし]/ビット列 [16ビット]	—	読み出すデバイスのデバイス点数をバイナリコードで指定します。 ・サブコマンドの0ビット目が0の場合: 1~960 ・サブコマンドの0ビット目が1の場合: 1~3972
i_uChannel	自局使用チャンネル	ワード[符号なし]/ビット列 [16ビット]	1~9	自局が使用するチャンネルを指定します。 チャンネルにより要求伝文にシリアル番号*2を付加する/付加しないが決まるため, 用途に応じてチャンネルを指定します。 ・1: シリアル番号を付加しないチャンネル(3Eフレーム使用) ・2~9: シリアル番号を付加するチャンネル(4Eフレーム使用)
i_uTarget_Port_No	相手機器ポート番号	ワード[符号なし]/ビット列 [16ビット]	1~65534	相手機器のUDPポート番号またはDTLSポート番号を指定します。

*1 デバイスコードについては, 下記を参照してください。

📖 SLMPリファレンスマニュアル

*2 同一のSLMP対応機器に複数の要求伝文を送信する場合に付加します。付加するシリアル番号は, システムで自動的に決定されます。シリアル番号については, 下記を参照してください。

📖 SLMPリファレンスマニュアル

■出力引数

変数名	名称	データ型	内容	デフォルト値																																																				
o_bENO	実行状態	ビット	ON: 実行指令ON OFF: 実行指令OFF	OFF																																																				
o_bOK	正常完了	ビット	ONの場合、FBの処理が正常完了したことを示します。	OFF																																																				
o_bErr	異常完了	ビット	ONの場合、FBの処理が異常完了したことを示します。	OFF																																																				
o_uErrId	エラーコード	ワード[符号なし]/ ビット列[16ビット]	異常完了時にエラーコードが格納されます。	0																																																				
o_uReadData	読出しデータ格納先	ワード[符号なし]/ ビット列[16ビット]	読み出すデータを格納するデバイスの先頭デバイス番号を指定します。読み出したデータは、バイナリコードで格納されます。 ■サブコマンドの0ビット目が0の場合 ワード単位でデバイスデータを読み出します。 例: ワード単位でビットデバイスM100~M115(1ワード分)を読み出す場合 1ワード目: b15b8b7b0 <table><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td colspan="4">...</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr></table> M115...M100 例: ワード単位でワードデバイスD0~D2を読み出す場合 1ワード目: b15b8b7b0 <table><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr></table> D0 2ワード目: b15b8b7b0 <table><tr><td>0</td><td>0</td><td>0</td><td>2</td></tr></table> D1 3ワード目: b15b8b7b0 <table><tr><td>1</td><td>D</td><td>E</td><td>F</td></tr></table> D2 ■サブコマンドの0ビット目が1の場合 ビット単位でデバイスデータを読み出します。 例: ビット単位でビットデバイスM100~M107を読み出す場合 1ワード目: b15b8b7b0 <table><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr></table> M102M103M100M101 2ワード目: b15b8b7b0 <table><tr><td>1</td><td>1</td><td>0</td><td>0</td></tr></table> M106M107M104M105	1	2	3	4	...				0	0	0	1	0	0	1	0	0	0	1	0	0	0	1	1	0	1	0	1	0	1	0	0	1	2	3	4	0	0	0	2	1	D	E	F	0	1	0	0	1	1	0	0	0
1	2	3	4																																																					
...																																																								
0	0	0	1																																																					
0	0	1	0																																																					
0	0	1	0																																																					
0	0	1	1																																																					
0	1	0	1																																																					
0	1	0	0																																																					
1	2	3	4																																																					
0	0	0	2																																																					
1	D	E	F																																																					
0	1	0	0																																																					
1	1	0	0																																																					

■動作パラメータ

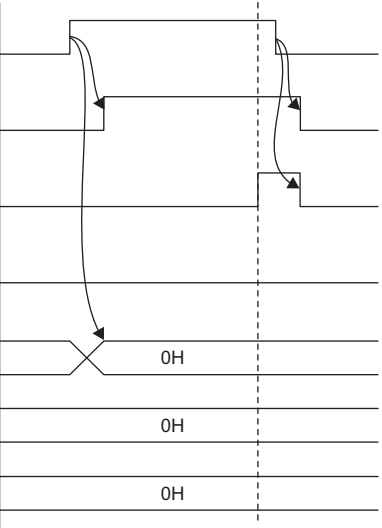
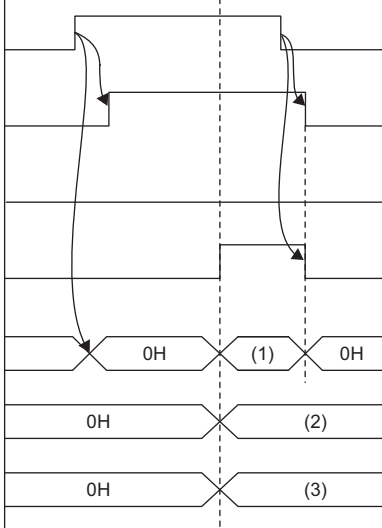
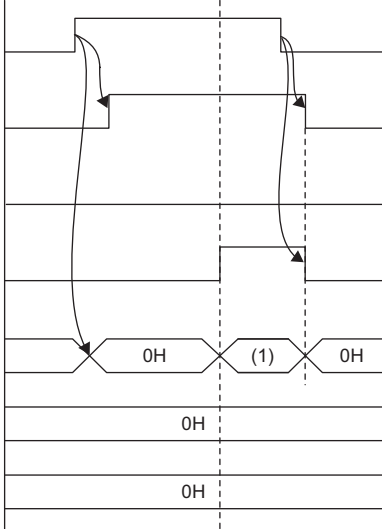
変数名	名称	データ型	有効範囲	内容	デフォルト値
pbi_uRequestModuleIO	要求先ユニット/ O番号	ワード[符号なし]/ ビット列[16ビット]	03E0H~03E3H, 03FFH	アクセス先のユニットを指定します。 • 03FFH: 自局, 管理CPU • 03E0H: マルチCPU1号機 • 03E1H: マルチCPU2号機 • 03E2H: マルチCPU3号機 • 03E3H: マルチCPU4号機	03FFH
pbi_uResendCountMax	最大再送回数	ワード[符号なし]/ ビット列[16ビット]	0~15	pbi_uMonitorTime(到達監視時間)で指定した監視時間内に完了しない場合に再送させる回数を設定します。 • 0回~15回	5
pbi_uMonitorTime	到達監視時間	ワード[符号なし]/ ビット列[16ビット]	0, 1~32767	処理完了までの監視時間を指定します。監視時間内に完了しない場合はpbi_uResendCountMax(最大再送回数)で指定した回数まで再送されます。 • 0: 10秒 • 1~32767: 1~32767秒	0
pbi_bEncryptionComm	暗号化通信有無	ビット列	ON, OFF	SLMPフレームの暗号化の有無を指定します。 • ON: 有効(暗号化する) • OFF: 無効(暗号化しない)	OFF
pbi_uLocal_Port_No	自ノードポート 番号	ワード[符号なし]/ ビット列[16ビット]	1~4999, 5010~65534	自ノードのポート番号を指定します。ポート番号1~1023は一般的に予約されたポート番号(WELL KNOWN PORT NUMBERS)に割り当てられているため、ポート番号1024~4999, 5010~65534の使用を推奨します。 指定した値は、pbi_bEncryptionComm(暗号化通信有無)がONの場合のみ有効になります。	4096

■公開変数

変数名	名称	データ型	内容	デフォルト値
pbo_uResendCount	再送回数	ワード[符号なし]/ビット列 [16ビット]	再送を行った回数(結果)が格納されます。異常を検知して再送を中断するまでに再送を行った回数(結果)が格納されます。	0
pbo_u4ErrTime	異常発生時刻	ワード[符号なし]/ビット列 [16ビット](0..3)	異常発生時の時計データが格納されます(異常時のみ)。 1ワード目 • 上位8ビット: 月(01H~12H) • 下位8ビット: 年(00H~99H)西暦下2桁 2ワード目 • 上位8ビット: 時(00H~23H) • 下位8ビット: 日(01H~31H) 3ワード目 • 上位8ビット: 秒(00H~59H) • 下位8ビット: 分(00H~59H) 4ワード目 • 上位8ビット: 年(00H~99H)西暦上2桁 • 下位8ビット: 曜日(00H(日)~06H(土))	0
pbo_uErrIP_Address_3rd_4th	異常検出局IPアドレス (第3, 4オクテット)	ワード[符号なし]/ビット列 [16ビット](0..1)	異常を検出した局のIPアドレス(第3オクテット, 第4オクテット)が格納されます。 ^{*1} 例: 192.168.1.2の場合は0102H	0
pbo_uErrIP_Address_1st_2nd	異常検出局IPアドレス (第1, 2オクテット)	ワード[符号なし]/ビット列 [16ビット](0..1)	異常を検出した局のIPアドレス(第1オクテット, 第2オクテット)が格納されます。 ^{*1} 例: 192.168.1.2の場合はC0A8H	0

*1 専用命令が異常完了した場合のみ値が格納されます。
異常検出機器IPアドレス(第3, 4オクテット)および異常検出機器IPアドレス(第1, 2オクテット)には入力引数の対象局アドレスに設定した値が格納されます。

項目	内容	
対象機器	対象ユニット	MELSEC MXコントローラ(Ethernet用ポート)
	対象CPU	MELSEC MXコントローラ
	対象エンジニアリングツール	GX Works3
使用言語	ラダー	
ラベル使用量	ラベル: 1096点 ラッチラベル: 0点 プログラムに組み込んだラベル使用量は、使用するコントローラ、引数に指定したデバイスやGX Works3のオプション設定によって異なります。GX Works3のオプション設定については、下記を参照してください。  GX Works3 オペレーティングマニュアル	
機能説明	・i_bEN(実行指令)のONで、SLMP対応機器のデバイスデータを読み出します。 ・本FBは、相手機器のIPアドレスを指定して実行します。 ・本FBは、SLMPのReadコマンド(コマンド: 0401)を使用します。SLMPのコマンドの伝文はバイナリコードになります。( SLMP リファレンスマニュアル) ・pbi_bEncryptionComm(暗号化通信有無)をONし本FBを実行すると、pbi_uLocal_Port_No(自ノードポート番号)に指定したポート番号を使用してDTLSで暗号化したSLMPフレームによる通信を行います。	
FBコンパイル方式	マクロ型	
FB動作	パルス型(複数スキャン実行型)	

項目	内容
入出力信号の動き	- 正常完了の場合  - 異常完了の場合(専用命令異常完了時)  (1) エラーコード (2) 異常発生時刻 (3) 異常検出機器IPアドレス - 異常完了の場合(専用命令正常完了/終了コード異常時)  (1) エラーコード

項目	内容
注意事項	- 本FBは、エラーから復帰する処理は含んでいません。エラー処理については、システムや要求動作に合わせ、別途作成してください。 - 本FBは、SLMPSND命令またはSLMPSNDC命令を使用します。SLMPSND命令とSLMPSNDC命令は、対象機器が異常応答を返した場合でも正常完了になります。そのため、本FBでは専用命令の完了状態に加え、応答フレームの終了コードで正常完了/異常完了を判定します。終了コードで異常完了と判定した場合は、o_uErrId(エラーコード)には終了コードを、pbo_u4ErrTime(異常発生時刻)及び、pbo_uErrIP_Address_3rd_4th(異常検出機器IPアドレス(第3, 4オクテット)), pbo_uErrIP_Address_1st_2nd(異常検出機器IPアドレス(第1, 2オクテット))に、0(初期値)が格納されているか確認してください。異常原因及び処置方法は使用しているSLMP対応機器のマニュアルを参照してください。(☞SLMPリファレンスマニュアル) - i_bEN(実行指令)は、o_bOK(正常完了)またはo_bErr(異常完了)がONした後でOFFしてください。i_bEN(実行指令)のOFFによってo_bOK(正常完了)とo_bErr(異常完了)がOFFし、o_uErrId(エラーコード)が0クリアされます。 - 本FBでは、SLMPの拡張指定でアクセスするデバイス(リンクダイレクトデバイスなど)の読み出しはできません。 - 本FBでは、他ネットワークの局を対象局とすることはできません。 - リモートパスワードが設定されている相手機器のポートに対して本FBを実行する場合は、リモートパスワードのアンロック処理を行ってから実行してください。リモートパスワードが設定されている相手機器のポートに対して本FBを実行した場合は、エラーとなります。 - 対象局は、SLMPコマンドの「Read(コマンド:0401)」に対応している必要があります。 - 本FBは、バイナリコードによる通信のみが対象です(ASCIIコードによる通信はできません)。 - 本FBは、UDPまたはDTLSで通信します。相手機器も同じ設定にしてください。 - 本FBは、プログラム別ラベル初期値を使用します。コントローラのブート運転において、本FBを使用したプログラムファイルをブートファイル設定で指定する場合、同様にプログラム別ラベル初期値ファイルもブートファイル設定で指定してください。なお、1893ページエラーコードに記載されていないエラーコードが発生した場合は、ブートファイル設定でプログラム別ラベル初期値ファイルが設定されていない可能性があります。ブートファイル設定でプログラム別ラベル初期値ファイルを指定してください。 - pbi_bEncryptionComm(暗号化通信有無)がON(有効)時、pbi_uLocal_Port_No(自ノードポート番号)に設定する値は、ユニットパラメータ(ポート3:Ethernet)の"相手機器接続構成設定"に設定した"シーケンサ ポート番号"に設定した自局のポート番号と同じにしないでください。設定したポート番号が重複している場合、本FBが異常完了する場合があります。 - 本FBを実行した場合に、コントローラで演算異常が発生する場合があります。その場合は、イベント履歴の演算異常の詳細情報を確認して、該当ユニットFBの入力引数を見直してください。

■エラーコード

エラーコード	内容	処置方法
100H	引数のデバイス点数(i_uDevicePoints)に範囲外の値が設定されています。	デバイス点数(i_uDevicePoints)の範囲を範囲内に修正してください。
100H以外	SLMPSND命令のエラーコードを確認し処置してください。 ☞ 1120ページ G(P).SLMPSND, J(P).SLMPSND, SP.SLMPSND 暗号化通信を行う場合は、SLMPSNDC命令のエラーコードを確認し処置してください。 ☞ 1129ページ G(P).SLMPSNDC	

M+MX-SQ_SLMP_DeviceWrite_IP

IPアドレス指定でSLMP対応機器にデバイスデータを書き込みます。
相手機器は、SLMPコマンド(Device Write)に対応している必要があります。
暗号化通信を行う場合は、相手機器が上記に加え、暗号化通信プロトコル(DTLS)に対応している必要があります。

ラダー	FBD/LD
M+MX-SQ_SLMP_DeviceWrite_IP B: i_bENo_bENO: B DUT: i_stModuleo_bOK: B UW: i_u2IP_Addresso_bErr: B UW: i_uSubCommando_uErrId: UW UW: i_uDeviceCode UW: i_u2DeviceNo UW: i_uDevicePoints UW: i_uWriteData UW: i_uChannel UW: i_uTarget_Port_No pbi_uRequestModuleIO H3FF pbi_uResendCountMax 5 pbi_uMonitorTime 0 pbi_bEncryptionComm 0 pbi_uLocal_Port_No 4096 pbo_uResendCount 0 pbo_u4ErrTime 0 pbo_uErrIP_Address_3rd_4th 0 pbo_uErrIP_Address_1st_2nd 0	M+MX-SQ_SLMP_DeviceWrite_IP i_bENo_bENO i_stModuleo_bOK i_u2IP_Addresso_bErr i_uSubCommando_uErrId i_uDeviceCode i_u2DeviceNo i_uDevicePoints i_uWriteData i_uChannel i_uTarget_Port_No

ST

```
M_MX_SQ_SLMP_DeviceWrite_IP(  
  i_bEN:= ?BOOL?,  
  i_stModule:= ?M+MX_SQ?,  
  i_u2IP_Address:= ?WORD(0..1)?,  
  i_uSubCommand:= ?WORD?,  
  i_uDeviceCode:= ?WORD?,  
  i_u2DeviceNo:= ?WORD(0..1)?,  
  i_uDevicePoints:= ?WORD?,  
  i_uWriteData:= ?WORD?,  
  i_uChannel:= ?WORD?,  
  i_uTarget_Port_No:= ?WORD?,  
  o_bENO=> ?BOOL?,  
  o_bOK=> ?BOOL?,  
  o_bErr=> ?BOOL?,  
  o_uErrId=> ?WORD?  
);
```

■入力引数

変数名	名称	データ型	有効範囲	内容						
i_bEN	実行指令	ビット	—	ON: FBを起動します。 OFF: FBを起動しません。						
i_stModule	ユニットラベル	構造体	—	FBを実行するコントローラのユニットラベルを指定します。 (例: MX_SQ)						
i_u2IP_Address	相手機器IPアドレス	ワード[符号なし]/ ビット列[16ビット](0..1)	0.0.0.1~253.255.255.254(00000001H ~DFFFFFFEH)	対象局のIPアドレスを指定します。 1ワード目に第3~4オクテット、2ワード目に第1~2オクテットを指定します。 ただし、第4オクテットが0または255(FFH)となる設定はできません。 b15b8b7b0 <table><tr><td>+0</td><td>(3)</td><td>(4)</td></tr><tr><td>+1</td><td>(1)</td><td>(2)</td></tr></table> (1)~(4): IPアドレスのオクテットを示します。	+0	(3)	(4)	+1	(1)	(2)
+0	(3)	(4)								
+1	(1)	(2)								
i_uSubCommand	サブコマンド	ワード[符号なし]/ ビット列[16ビット]	—	書き込み単位およびデバイスの指定方法を指定します。 ■0ビット目: 書き込み単位 0: ワード単位 1: ビット単位 ■書き込むデバイスの指定方法 0: デバイスコードを2桁、先頭デバイス番号を6桁で指定 (MELSEC-Q/Lシリーズ用) 1: デバイスコードを4桁、先頭デバイス番号を8桁で指定 (MELSEC iQ-Rシリーズ用)						
i_uDeviceCode	デバイスコード*1	ワード[符号なし]/ ビット列[16ビット]	—	書き込むデバイスのデバイスコードをバイナリコードで指定します。 ・サブコマンドの1ビット目が0の場合: 2桁 ・サブコマンドの1ビット目が1の場合: 4桁						
i_u2DeviceNo	先頭デバイス番号	ワード[符号なし]/ ビット列[16ビット](0..1)	—	書き込むデバイスの先頭デバイス番号をバイナリコードで指定します。 ・サブコマンドの1ビット目が0の場合: 6桁 ・サブコマンドの1ビット目が1の場合: 8桁						
i_uDevicePoints	デバイス点数	ワード[符号なし]/ ビット列[16ビット]	—	書き込むデバイスのデバイス点数をバイナリコードで指定します。 ・サブコマンドの0ビット目が0の場合: 1~960 ・サブコマンドの0ビット目が1の場合: 1~3972						

変数名	名称	データ型	有効範囲	内容																																																
i_uWriteData	書き込みデータ格納先	ワード[符号なし]/ビット列[16ビット]	—	書き込むデータを格納するデバイスの先頭デバイス番号を指定します。 ■サブコマンドの0ビット目が0の場合 ワード単位でデバイスデータを書込みます。 例: ワード単位でビットデバイスM100~M115(1ワード分)を書き込む場合 1ワード目: b15b8b7b0 <table><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td colspan="4">...</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr></table> M115...M100 例: ワード単位でワードデバイスD0~D2を書き込む場合 1ワード目: b15b8b7b0 <table><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr></table> D0 2ワード目: b15b8b7b0 <table><tr><td>0</td><td>0</td><td>0</td><td>2</td></tr></table> D1 3ワード目: b15b8b7b0 <table><tr><td>1</td><td>D</td><td>E</td><td>F</td></tr></table> D2 ■サブコマンドの0ビット目が1の場合 ビット単位でデバイスデータを書き込みます。 例: ビット単位でビットデバイスM100~M107を書き込む場合 1ワード目: b15b8b7b0 <table><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr></table> M102M103M100M101 2ワード目: b15b8b7b0 <table><tr><td>1</td><td>1</td><td>0</td><td>0</td></tr></table> M106M107M104M105	1	2	3	4	...				0	0	0	1	0	0	1	0	0	0	0	1	1	0	1	0	0	0	1	0	1	2	3	4	0	0	0	2	1	D	E	F	0	1	0	0	1	1	0	0
1	2	3	4																																																	
...																																																				
0	0	0	1																																																	
0	0	1	0																																																	
0	0	0	1																																																	
1	0	1	0																																																	
0	0	1	0																																																	
1	2	3	4																																																	
0	0	0	2																																																	
1	D	E	F																																																	
0	1	0	0																																																	
1	1	0	0																																																	
i_uChannel	自局使用チャンネル	ワード[符号なし]/ビット列[16ビット]	1~9	自局が使用するチャンネルを指定します。 チャンネルにより要求伝文にシリアル番号*2を付加する/付加しないが決まるため、用途に応じてチャンネルを指定します。 ・1: シリアル番号を付加しないチャンネル(3Eフレーム使用) ・2~9: シリアル番号を付加するチャンネル(4Eフレーム使用)																																																
i_uTarget_Port_No	相手機器ポート番号	ワード[符号なし]/ビット列[16ビット]	1~65534	相手機器のUDPポート番号またはDTLSポート番号を指定します。																																																

*1 デバイスコードについては、下記を参照してください。

📖 SLMPリファレンスマニュアル

*2 同一のSLMP対応機器に複数の要求伝文を送信する場合に付加します。付加するシリアル番号は、システムで自動的に決定されます。
シリアル番号については、下記を参照してください。

📖 SLMPリファレンスマニュアル

■出力引数

変数名	名称	データ型	内容	デフォルト値
o_bENO	実行状態	ビット	ON: 実行指令ON OFF: 実行指令OFF	OFF
o_bOK	正常完了	ビット	ONの場合、FBの処理が正常完了したことを示します。	OFF
o_bErr	異常完了	ビット	ONの場合、FBの処理が異常完了したことを示します。	OFF
o_uErrId	エラーコード	ワード[符号なし]/ ビット列[16ビット]	異常完了時にエラーコードが格納されます。	0

■動作パラメータ

変数名	名称	データ型	有効範囲	内容	デフォルト値
pbi_uRequestModuleIO	要求先ユニット I/O番号	ワード[符号なし]/ビット列 [16ビット]	03E0H~03E3H, 03FFH	アクセス先のユニットを指定します。 ・ 03FFH: 自局、管理CPU ・ 03E0H: マルチCPU1号機 ・ 03E1H: マルチCPU2号機 ・ 03E2H: マルチCPU3号機 ・ 03E3H: マルチCPU4号機	03FFH
pbi_uResendCountMax	最大再送回数	ワード[符号なし]/ビット列 [16ビット]	0~15	pbi_uMonitorTime(到達監視時間)で 指定した監視時間内に完了しない場 合に再送させる回数を設定します。 ・ 0回~15回	5
pbi_uMonitorTime	到達監視時間	ワード[符号なし]/ビット列 [16ビット]	0, 1~32767	処理完了までの監視時間を指定しま す。監視時間内に完了しない場合は pbi_uResendCountMax(最大再送回 数)で指定した回数まで再送されま す。 ・ 0: 10秒 ・ 1~32767: 1~32767秒	0
pbi_bEncryptionComm	暗号化通信有無	ビット列	ON, OFF	SLMPフレームの暗号化の有無を指定 します。 ・ ON: 有効(暗号化する) ・ OFF: 無効(暗号化しない)	OFF
pbi_uLocal_Port_No	自ノードポート 番号	ワード[符号なし]/ビット列 [16ビット]	1~4999, 5010~65534	自ノードのポート番号を指定します。 ポート番号1~1023は一般的に予約さ れたポート番号(WELL KNOWN PORT NUMBERS)に割り当てられているた め、ポート番号1024~4999, 5010~65534の使用を推奨します。 指定した値は、 pbi_bEncryptionComm(暗号化通信 有無)がONの場合のみ有効になりま す。	4096

■公開変数

変数名	名称	データ型	内容	デフォルト値
pbo_uResendCount	再送回数	ワード[符号なし]/ビット列 [16ビット]	再送を行った回数(結果)が格納されます。	0
pbo_u4ErrTime	異常発生時刻	ワード[符号なし]/ビット列 [16ビット](0..3)	異常発生時の時計データが格納されます(異常時のみ)。 1ワード目 ・上位8ビット: 月(01H~12H) ・下位8ビット: 年(00H~99H)西暦下2桁 2ワード目 ・上位8ビット: 時(00H~23H) ・下位8ビット: 日(01H~31H) 3ワード目 ・上位8ビット: 秒(00H~59H) ・下位8ビット: 分(00H~59H) 4ワード目 ・上位8ビット: 年(00H~99H)西暦上2桁 ・下位8ビット: 曜日(00H(日)~06H(土))	0
pbo_uErrIP_Address_3rd_4th	異常検出局IPアドレス (第3, 4オクテット)	ワード[符号なし]/ビット列 [16ビット](0..1)	異常を検出した局のIPアドレス(第3オクテット, 第4オクテット)が格納されます。*1 例: 192.168.1.2の場合 0102h	0
pbo_uErrIP_Address_1st_2nd	異常検出局IPアドレス (第1, 2オクテット)	ワード[符号なし]/ビット列 [16ビット](0..1)	異常を検出した局のIPアドレス(第1オクテット, 第2オクテット)が格納されます。*1 例: 192.168.1.2の場合 C0A8h	0

*1 専用命令が異常完了した場合のみ値が格納されます。

異常検出機器IPアドレス(第3, 4オクテット)および異常検出機器IPアドレス(第1, 2オクテット)には入力引数の対象局アドレスに設定した値が格納されます。

機能

項目	内容
対象機器	対象ユニット
	MELSEC MXコントローラ(Ethernet用ポート)
	対象CPU
	MELSEC MXコントローラ
	対象エンジニアリングツール
	GX Works3
使用言語	ラダー
ラベル使用量	ラベル: 1096点 ラッチラベル: 0点 プログラムに組み込んだラベル使用量は、使用するコントローラ、引数に指定したデバイスやGX Works3のオプション設定によって異なります。GX Works3のオプション設定については、下記を参照してください。 📖 GX Works3 オペレーティングマニュアル
機能説明	・i_bEN(実行指令)のONで、SLMP対応機器のデバイスデータを書き込みます。 ・本FBは、相手機器のIPアドレスを指定して実行します。 ・本FBは、SLMPのWriteコマンド(コマンド: 1401)を使用します。SLMPのコマンドの伝文はバイナリコードになります。(📖 SLMP リファレンスマニュアル) ・pbi_bEncryptionComm(暗号化通信有無)をONし本FBを実行すると、pbi_uLocal_Port_No(自ノードポート番号)に指定したポート番号を使用してDTLSで暗号化したSLMPフレームによる通信を行います。
FBコンパイル方式	マクロ型
FB動作	パルス型(複数スキャン実行型)

項目	内容
入出力信号の動き	• 正常完了の場合 • 異常完了の場合(専用命令異常完了時) (1) エラーコード (2) 異常発生時刻 (3) 異常検出機器IPアドレス • 異常完了の場合(専用命令正常完了/終了コード異常時) (1) エラーコード

項目	内容
注意事項	- 本FBは、エラーから復帰する処理は含んでいません。エラー処理については、システムや要求動作に合わせ、別途作成してください。 - 本FBは、SLMPSND命令またはSLMPSNDC命令を使用します。SLMPSND命令とSLMPSNDC命令は、対象機器が異常応答を返した場合でも正常完了になります。そのため、本FBでは専用命令の完了状態に加え、応答フレームの終了コードで正常完了/異常完了を判定します。終了コードで異常完了と判定した場合は、o_uErrId(エラーコード)には終了コードを、pbo_u4ErrTime(異常発生時刻)及び、pbo_uErrIP_Address_3rd_4th(異常検出機器IPアドレス(第3, 4オクテット)), pbo_uErrIP_Address_1st_2nd(異常検出機器IPアドレス(第1, 2オクテット))に、0(初期値)が格納されているか確認してください。異常原因及び処置方法は使用しているSLMP対応機器のマニュアルを参照してください。(☞SLMPリファレンスマニュアル) - i_bEN(実行指令)は、o_bOK(正常完了)またはo_bErr(異常完了)がONした後でOFFしてください。i_bEN(実行指令)のOFFによってo_bOK(正常完了)とo_bErr(異常完了)がOFFし、o_uErrId(エラーコード)が0クリアされます。 - 本FBでは、SLMPの拡張指定でアクセスするデバイス(リンクダイレクトデバイスなど)の書き込みはできません。 - 本FBでは、他ネットワークの局を対象局とすることはできません。 - リモートパスワードが設定されている相手機器のポートに対して本FBを実行する場合は、リモートパスワードのアンロック処理を行ってから実行してください。リモートパスワードが設定されている相手機器のポートに対して本FBを実行した場合は、エラーとなります。 - 対象局は、SLMPコマンドの「Write(コマンド: 1401)」に対応している必要があります。 - 本FBは、バイナリコードによる通信のみが対象です(ASCIIコードによる通信はできません)。 - 本FBは、UDPまたはDTLSで通信します。相手機器も同じ設定にしてください。 - 本FBは、プログラム別ラベル初期値を使用します。コントローラのブート運転において、本FBを使用したプログラムファイルをブートファイル設定で指定する場合、同様にプログラム別ラベル初期値ファイルもブートファイル設定で指定してください。なお、1900ページエラーコードに記載されていないエラーコードが発生した場合は、ブートファイル設定でプログラム別ラベル初期値ファイルが設定されていない可能性があります。ブートファイル設定でプログラム別ラベル初期値ファイルを指定してください。 - pbi_bEncryptionComm(暗号化通信有無)がON(有効)時、pbi_uLocal_Port_No(自ノードポート番号)に設定する値は、ユニットパラメータ(ポート3:Ethernet)の"相手機器接続構成設定"に設定した"シーケンサ ポート番号"に設定した自局のポート番号と同じにしないでください。設定したポート番号が重複している場合、本FBが異常完了する場合があります。 - 本FBを実行した場合に、コントローラで演算異常が発生する場合があります。その場合は、イベント履歴の演算異常の詳細情報を確認して、該当ユニットFBの入力引数を見直してください。

■エラーコード

エラーコード	内容	処置方法
100H	引数のデバイス点数(i_uDevicePoints)に範囲外の値が設定されています。	デバイス点数(i_uDevicePoints)の範囲を範囲内に修正してください。
100H以外	SLMPSND命令のエラーコードを確認し処置してください。 ☞ 1120ページ G(P).SLMPSND, J(P).SLMPSND, SP.SLMPSND 暗号化通信を行う場合は、SLMPSNDC命令のエラーコードを確認し処置してください。 ☞ 1129ページ G(P).SLMPSNDC	

M+MX-TSN_SLMP_DeviceRead_IP

IPアドレス指定でSLMP対応機器のデバイスデータを読み出します。
相手機器は、SLMPコマンド(Device Read)に対応している必要があります。

ラダー	FBD/LD
M+MX-TSN_SLMP_DeviceRead_IP B: i_bEN o_bENO: B DUT: i_stModule o_bOK: B UW: i_u2IP_Address o_bErr: B UW: i_uSubCommand o_uErrId: UW UW: i_uDeviceCode o_uReadData: UW UW: i_u2DeviceNo UW: i_uDevicePoints UW: i_uChannel UW: i_uTarget_Port_No pbi_uRequestModuleIO H3FF pbi_uResendCountMax 5 pbi_uMonitorTime 0 pbo_uResendCount 0 pbo_u4ErrTime 0 pbo_uErrIP_Address_3rd_4th 0 pbo_uErrIP_Address_1st_2nd 0	M+MX-TSN_SLMP_DeviceRead_IP i_bEN o_bENO i_stModule o_bOK i_u2IP_Address o_bErr i_uSubCommand o_uErrId i_uDeviceCode o_uReadData i_u2DeviceNo i_uDevicePoints i_uChannel i_uTarget_Port_No

ST

```

M_MX_TSN_SLMP_DeviceRead_IP(
  i_bEN:= ?BOOL?,
  i_stModule:= ?M+MX_TSN?,
  i_u2IP_Address:= ?WORD(0..1)?,
  i_uSubCommand:= ?WORD?,
  i_uDeviceCode:= ?WORD?,
  i_u2DeviceNo:= ?WORD(0..1)?,
  i_uDevicePoints:= ?WORD?,
  i_uChannel:= ?WORD?,
  i_uTarget_Port_No:= ?WORD?,
  o_bENO=> ?BOOL?,
  o_bOK=> ?BOOL?,
  o_bErr=> ?BOOL?,
  o_uErrId=> ?WORD?,
  o_uReadData=> ?WORD?
);

```

設定データ

■入力引数

変数名	名称	データ型	有効範囲	内容															
i_bEN	実行指令	ビット	—	ON: FBを起動します。 OFF: FBを起動しません。															
i_stModule	ユニットラベル	構造体	—	FBを実行するコントローラのユニットラベルを指定します。 (例: MX_TSN)															
i_u2IP_Address	相手機器IPアドレス	ワード[符号なし]/ビット列 [16ビット](0..1)	0.0.0.1~253.255.255.254(00000001H~DFFFFFFEH)	相手機器のIPアドレスを指定します。 1ワード目に第3~4オクテット, 2ワード目に第1~2オクテットを指定します。 ただし, 第4オクテットが0または255(FFH)となる設定はできません。 <table style="margin-left: auto; margin-right: auto;"> <tr> <td></td><td style="text-align: center;">b15</td><td style="text-align: center;">b8</td><td style="text-align: center;">b7</td><td style="text-align: center;">b0</td></tr> <tr> <td>+0</td><td style="border: 1px solid black; text-align: center;">(3)</td><td colspan="2" style="border: 1px solid black; text-align: center;">(4)</td><td></td></tr> <tr> <td>+1</td><td style="border: 1px solid black; text-align: center;">(1)</td><td colspan="2" style="border: 1px solid black; text-align: center;">(2)</td><td></td></tr> </table> (1)~(4): IPアドレスのオクテットを示します。		b15	b8	b7	b0	+0	(3)	(4)			+1	(1)	(2)		
	b15	b8	b7	b0															
+0	(3)	(4)																	
+1	(1)	(2)																	

変数名	名称	データ型	有効範囲	内容
i_uSubCommand	サブコマンド	ワード[符号なし]/ビット列 [16ビット]	—	読出し単位およびデバイスの指定方法を指定します。 ■0ビット目: 読出し単位 0: ワード単位 1: ビット単位 ■1ビット目: 読み出すデバイスの指定方法 0: デバイスコードを2桁, 先頭デバイス番号を6桁で指定(MELSEC-Q/Lシリーズ用) 1: デバイスコードを4桁, 先頭デバイス番号を8桁で指定(MELSEC iQ-Rシリーズ用)
i_uDeviceCode	デバイスコード*1	ワード[符号なし]/ビット列 [16ビット]	—	読み出すデバイスのデバイスコードをバイナリコードで指定します。 ・サブコマンドの1ビット目が0の場合: 2桁 ・サブコマンドの1ビット目が1の場合: 4桁
i_u2DeviceNo	先頭デバイス番号	ワード[符号なし]/ビット列 [16ビット](0..1)	—	読み出すデバイスの先頭デバイス番号をバイナリコードで指定します。 ・サブコマンドの1ビット目が0の場合: 6桁 ・サブコマンドの1ビット目が1の場合: 8桁
i_uDevicePoints	デバイス点数	ワード[符号なし]/ビット列 [16ビット]	—	読み出すデバイスのデバイス点数をバイナリコードで指定します。 ・サブコマンドの0ビット目が0の場合: 1~960 ・サブコマンドの0ビット目が1の場合: 1~3972
i_uChannel	自局使用チャンネル	ワード[符号なし]/ビット列 [16ビット]	1~9	自局が使用するチャンネルを指定します。 ・1: シリアル番号を付加しないチャンネル(3Eフレーム使用) ・2~9: シリアル番号を付加するチャンネル(4Eフレーム使用) ・10~17: 局番拡張フレーム
i_uTarget_Port_No	相手機器ポート番号	ワード[符号なし]/ビット列 [16ビット]	1~65534	相手機器のUDPポート番号を指定します。

*1 デバイスコードについては, 下記を参照してください。

📖 SLMPリファレンスマニュアル

設定データ

■出力引数

変数名	名称	データ型	内容	デフォルト値
o_bENO	実行状態	ビット	ON: 実行指令ON OFF: 実行指令OFF	OFF
o_bOK	正常完了	ビット	ONの場合, FBの処理が正常完了したことを示します。	OFF
o_bErr	異常完了	ビット	ONの場合, FBの処理が異常完了したことを示します。	OFF
o_uErrId	エラーコード	ワード[符号なし]/ ビット列[16ビット]	異常完了時にエラーコードが格納されます。	0

変数名	名称	データ型	内容	デフォルト値																																
o_uReadData	読出しデータ格納先	ワード[符号なし]/ ビット列[16ビット]	読み出すデータを格納するデバイスの先頭デバイス番号を指定します。読み出したデータは、バイナリコードで格納されます。 ■サブコマンドの0ビット目が0の場合 ワード単位でデバイスデータを読み出します。 例: ワード単位でビットデバイスM100~M115(1ワード分)を読み出す場合 1ワード目: b15b8b7b0 <table><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td>0001</td><td>0010</td><td>0011</td><td>0100</td></tr></table> M115...M100 例: ワード単位でワードデバイスD0~D2を読み出す場合 1ワード目: b15b8b7b0 <table><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr></table> D0 2ワード目: b15b8b7b0 <table><tr><td>0</td><td>0</td><td>0</td><td>2</td></tr></table> D1 3ワード目: b15b8b7b0 <table><tr><td>1</td><td>D</td><td>E</td><td>F</td></tr></table> D2 ■サブコマンドの0ビット目が1の場合 ビット単位でデバイスデータを読み出します。 例: ビット単位でビットデバイスM100~M107を読み出す場合 1ワード目: b15b8b7b0 <table><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr></table> M102M103M100M101 2ワード目: b15b8b7b0 <table><tr><td>1</td><td>1</td><td>0</td><td>0</td></tr></table> M106M107M104M105	1	2	3	4	...	...	...	...	0001	0010	0011	0100	1	2	3	4	0	0	0	2	1	D	E	F	0	1	0	0	1	1	0	0	0
1	2	3	4																																	
...	...	...	...																																	
0001	0010	0011	0100																																	
1	2	3	4																																	
0	0	0	2																																	
1	D	E	F																																	
0	1	0	0																																	
1	1	0	0																																	

■動作パラメータ

変数名	名称	データ型	有効範囲	内容	デフォルト値
pbi_uRequestModuleIO	要求先ユニットI/O番号	ワード[符号なし]/ビット列[16ビット]	03E0H~03E3H, 03FFH	アクセス先のユニットを指定します。 • 03FFH: 自局, 管理CPU • 03E0H: マルチCPU1号機 • 03E1H: マルチCPU2号機 • 03E2H: マルチCPU3号機 • 03E3H: マルチCPU4号機	03FFH
pbi_uResendCountMax	最大再送回数	ワード[符号なし]/ビット列[16ビット]	0~15	pbi_uMonitorTime(到達監視時間)で指定した監視時間内に完了しない場合に再送させる回数を設定します。 • 0回~15回	5
pbi_uMonitorTime	到達監視時間	ワード[符号なし]/ビット列[16ビット]	0, 1~32767	処理完了までの監視時間を指定します。監視時間内に完了しない場合はpbi_uResendCountMax(最大再送回数)で指定した回数まで再送されます。 • 0: 10秒 • 1~32767: 1~32767秒	0

■公開変数

変数名	名称	データ型	内容	デフォルト値
pbo_uResendCount	再送回数	ワード[符号なし]/ビット列[16ビット]	再送を行った回数(結果)が格納されます。異常を検知して再送を中断するまでに再送を行った回数(結果)が格納されます。	0
pbo_u4ErrTime	異常発生時刻	ワード[符号なし]/ビット列[16ビット](0..3)	異常発生時の時計データが格納されます(異常時のみ)。 1ワード目 • 上位8ビット: 月(01H~12H) • 下位8ビット: 年(00H~99H)西暦下2桁 2ワード目 • 上位8ビット: 時(00H~23H) • 下位8ビット: 日(01H~31H) 3ワード目 • 上位8ビット: 秒(00H~59H) • 下位8ビット: 分(00H~59H) 4ワード目 • 上位8ビット: 年(00H~99H)西暦上2桁 • 下位8ビット: 曜日(00H(日)~06H(土))	0
pbo_uErrIP_Address_3rd_4th	異常検出局IPアドレス(第3, 4オクテット)	ワード[符号なし]/ビット列[16ビット](0..1)	異常を検出した局のIPアドレス(第3オクテット, 第4オクテット)が格納されます。 ^{*1} 例: 192.168.1.2の場合 0102h	0
pbo_uErrIP_Address_1st_2nd	異常検出局IPアドレス(第1, 2オクテット)	ワード[符号なし]/ビット列[16ビット](0..1)	異常を検出した局のIPアドレス(第1オクテット, 第2オクテット)が格納されます。 ^{*1} 例: 192.168.1.2の場合 C0A8h	0

*1 専用命令が異常完了した場合のみ値が格納されます。

異常検出機器IPアドレス(第3, 4オクテット)および異常検出機器IPアドレス(第1, 2オクテット)には入力引数の対象局アドレスに設定した値が格納されます。

項目	内容	
対象機器	対象ユニット	MELSEC MXコントローラ(CC-Link IE TSN用ポート)
	対象CPU	MELSEC MXコントローラ
	対象エンジニアリングツール	GX Works3
使用言語	ラダー	
ラベル使用量	ラベル: 1076点 ラッチラベル: 0点 プログラムに組み込んだラベル使用量は、使用するコントローラ、引数に指定したデバイスやGX Works3のオプション設定によって異なります。GX Works3のオプション設定については、下記を参照してください。  GX Works3 オペレーティングマニュアル	
機能説明	・i_bEN(実行指令)のONで、SLMP対応機器のデバイスデータを読み出します。 ・本FBは、相手機器のIPアドレスを指定して実行します。 ・本FBは、SLMPのReadコマンド(コマンド: 0401)を使用します。SLMPのコマンドの伝文はバイナリコードになります。( SLMP リファレンスマニュアル)	
FBコンパイル方式	マクロ型	
FB動作	パルス型(複数スキャン実行型)	

項目	内容
入出力信号の動き	• 正常完了の場合 • 異常完了の場合(専用命令異常完了時) (1) エラーコード (2) 異常発生時刻 (3) 異常検出機器IPアドレス • 異常完了の場合(専用命令正常完了/終了コード異常時) (1) エラーコード

項目	内容
注意事項	- 本FBは、エラーから復帰する処理は含んでいません。エラー処理については、システムや要求動作に合わせ、別途作成してください。 - 本FBは、SLMPSND命令を使用します。SLMPSND命令は、対象機器が異常応答を返した場合でも正常完了になります。そのため、本FBでは専用命令の完了状態に加え、応答フレームの終了コードで正常完了/異常完了を判定します。終了コードで異常完了と判定した場合は、o_uErrId(エラーコード)には終了コードを、pbo_u4ErrTime(異常発生時刻)及び、pbo_uErrIP_Address_3rd_4th(異常検出機器IPアドレス(第3, 4オクテット)), pbo_uErrIP_Address_1st_2nd(異常検出機器IPアドレス(第1, 2オクテット))に、0(初期値)が格納されているか確認してください。異常原因及び処置方法は使用しているSLMP対応機器のマニュアルを参照してください。(SLMPリファレンスマニュアル) - i_bEN(実行指令)は、o_bOK(正常完了)またはo_bErr(異常完了)がONした後でOFFしてください。i_bEN(実行指令)のOFFによってo_bOK(正常完了)とo_bErr(異常完了)がOFFし、o_uErrId(エラーコード)が0クリアされます。 - 本FBでは、SLMPの拡張指定でアクセスするデバイス(リンクダイレクトデバイスなど)の読出しはできません。 - 本FBでは、他ネットワークの局を対象局とすることはできません。 - リモートパスワードが設定されている相手機器のポートに対して本FBを実行する場合は、リモートパスワードのアンロック処理を行ってから実行してください。リモートパスワードが設定されている相手機器のポートに対して本FBを実行した場合は、エラーとなります。 - 対象局は、SLMPコマンドの「Read(コマンド:0401)」に対応している必要があります。 - 本FBは、バイナリコードによる通信のみが対象です(ASCIIコードによる通信はできません)。 - 本FBは、UDPで通信します。相手機器も同じ設定にしてください。 - 本FBは、プログラム別ラベル初期値を使用します。コントローラのブート運転において、本FBを使用したプログラムファイルをブートファイル設定で指定する場合、同様にプログラム別ラベル初期値ファイルもブートファイル設定で指定してください。なお、1907ページエラーコードに記載されていないエラーコードが発生した場合は、ブートファイル設定でプログラム別ラベル初期値ファイルが設定されていない可能性があります。ブートファイル設定でプログラム別ラベル初期値ファイルを指定してください。 - 本FBを実行した場合に、コントローラで演算異常が発生する場合があります。その場合は、イベント履歴の演算異常の詳細情報を確認して、該当ユニットFBの入力引数を見直してください。

■エラーコード

エラーコード	内容	処置方法
100H	引数のデバイス点数(i_uDevicePoints)に範囲外の値が設定されています。	デバイス点数(i_uDevicePoints)の範囲を範囲内に修正してください。
100H以外	SLMPSND命令のエラーコードを確認し処置してください。  1120ページ G(P).SLMPSND, J(P).SLMPSND, SP.SLMPSND	

M+MX-TSN_SLMP_DeviceWrite_IP

IPアドレス指定でSLMP対応機器にデバイスデータを書き込みます。
相手機器は、SLMPコマンド(Device Write)に対応している必要があります。

ラダー	FBD/LD
M+MX-TSN_SLMP_DeviceWrite_IP B: i_bEN o_bENO: B DUT: i_stModule o_bOK: B UW: i_u2IP_Address o_bErr: B UW: i_uSubCommand o_uErrId: UW UW: i_uDeviceCode UW: i_u2DeviceNo UW: i_uDevicePoints UW: i_uWriteData UW: i_uChannel UW: i_uTarget_Port_No pbi_uRequestModuleIO H3FF pbi_uResendCountMax 5 pbi_uMonitorTime 0 pbo_uResendCount 0 pbo_u4ErrTime 0 pbo_uErrIP_Address_3rd_4th 0 pbo_uErrIP_Address_1st_2nd 0	M+MX-TSN_SLMP_DeviceWrite_IP i_bEN o_bENO i_stModule o_bOK i_u2IP_Address o_bErr i_uSubCommand o_uErrId i_uDeviceCode i_u2DeviceNo i_uDevicePoints i_uWriteData i_uChannel i_uTarget_Port_No

ST

```
M_MX_TSN_SLMP_DeviceWrite_IP(  
  i_bEN:= ?BOOL?,  
  i_stModule:= ?M+MX_TSN?,  
  i_u2IP_Address:= ?WORD(0..1)?,  
  i_uSubCommand:= ?WORD?,  
  i_uDeviceCode:= ?WORD?,  
  i_u2DeviceNo:= ?WORD(0..1)?,  
  i_uDevicePoints:= ?WORD?,  
  i_uWriteData:= ?WORD?,  
  i_uChannel:= ?WORD?,  
  i_uTarget_Port_No:= ?WORD?,  
  o_bENO=> ?BOOL?,  
  o_bOK=> ?BOOL?,  
  o_bErr=> ?BOOL?,  
  o_uErrId=> ?WORD?  
);
```

■入力引数

変数名	名称	データ型	有効範囲	内容
i_bEN	実行指令	ビット	—	ON: FBを起動します。 OFF: FBを起動しません。
i_stModule	ユニットラベル	構造体	—	FBを実行するコントローラのユニットラベルを指定します。 (例: MX_TSN)
i_u2IP_Address	相手機器IPアドレス	ワード[符号なし]/ビット列 [16ビット](0..1)	0.0.0.1~253.255.255.254(00000001H ~DFFFFFFEH)	対象局のIPアドレスを指定します。 1ワード目に第3~4オクテット, 2ワード目に第1~2 オクテットを指定します。 ただし, 第4オクテットが0または255(FFH)となる 設定はできません。 b15b8b7b0 +0 (3)(4) +1 (1)(2) (1)~(4): IPアドレスのオクテットを示します。
i_uSubCommand	サブコマンド	ワード[符号なし]/ビット列 [16ビット]	—	書き込み単位およびデバイスの指定方法を指定しま す。 ■0ビット目: 書き込み単位 0: ワード単位 1: ビット単位 ■書き込むデバイスの指定方法 0: デバイスコードを2桁, 先頭デバイス番号を6桁 で指定(MELSEC-Q/Lシリーズ用) 1: デバイスコードを4桁, 先頭デバイス番号を8桁 で指定(MELSEC iQ-Rシリーズ用)
i_uDeviceCode	デバイス コード*1	ワード[符号なし]/ビット列 [16ビット]	—	書き込むデバイスのデバイスコードをバイナリコー ドで指定します。 ・サブコマンドの1ビット目が0の場合: 2桁 ・サブコマンドの1ビット目が1の場合: 4桁
i_u2DeviceNo	先頭デバ イス番号	ワード[符号なし]/ビット列 [16ビット](0..1)	—	書き込むデバイスの先頭デバイス番号をバイナリ コードで指定します。 ・サブコマンドの1ビット目が0の場合: 6桁 ・サブコマンドの1ビット目が1の場合: 8桁
i_uDevicePoints	デバイス 点数	ワード[符号なし]/ビット列 [16ビット]	—	書き込むデバイスのデバイス点数をバイナリコード で指定します。 ■サブコマンドの0ビット目が0の場合 ・1~960 ■サブコマンドの0ビット目が1の場合: 1~3972 ・自局使用チャンネルが1~9: 1~960 ・自局使用チャンネルが10~17: 1~3960

変数名	名称	データ型	有効範囲	内容																																
i_uWriteData	書込みデータ格納先	ワード[符号なし]/ビット列 [16ビット]	—	書き込むデータを格納するデバイスの先頭デバイス番号を指定します。 ■サブコマンドの0ビット目が0の場合 ワード単位でデバイスデータを書込みます。 例: ワード単位でビットデバイスM100~M115(1ワード分)を書き込む場合 1ワード目: b15b8b7b0 <table><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td>0001</td><td>0010</td><td>0001</td><td>1010</td></tr></table> M115...M100 例: ワード単位でワードデバイスD0~D2を書き込む場合 1ワード目: b15b8b7b0 <table><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr></table> D0 2ワード目: b15b8b7b0 <table><tr><td>0</td><td>0</td><td>0</td><td>2</td></tr></table> D1 3ワード目: b15b8b7b0 <table><tr><td>1</td><td>D</td><td>E</td><td>F</td></tr></table> D2 ■サブコマンドの0ビット目が1の場合 ビット単位でデバイスデータを書き込みます。 例: ビット単位でビットデバイスM100~M107を書き込む場合 1ワード目: b15b8b7b0 <table><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr></table> M102M103M100M101 2ワード目: b15b8b7b0 <table><tr><td>1</td><td>1</td><td>0</td><td>0</td></tr></table> M106M107M104M105	1	2	3	4	...	...	...	...	0001	0010	0001	1010	1	2	3	4	0	0	0	2	1	D	E	F	0	1	0	0	1	1	0	0
1	2	3	4																																	
...	...	...	...																																	
0001	0010	0001	1010																																	
1	2	3	4																																	
0	0	0	2																																	
1	D	E	F																																	
0	1	0	0																																	
1	1	0	0																																	
i_uChannel	自局使用チャンネル	ワード[符号なし]/ビット列 [16ビット]	1~17	自局が使用するチャンネルを指定します。 • 1: シリアル番号を付加しないチャンネル(3Eフレーム使用) • 2~9: シリアル番号を付加するチャンネル(4Eフレーム使用) • 10~17: 局番拡張フレーム																																
i_uTarget_Port_No	相手機器ポート番号	ワード[符号なし]/ビット列 [16ビット]	1~65534	相手機器のUDPポート番号を指定します。																																

*1 デバイスコードについては、下記を参照してください。

📖 SLMPリファレンスマニュアル

■出力引数

変数名	名称	データ型	内容	デフォルト値
o_bENO	実行状態	ビット	ON: 実行指令ON OFF: 実行指令OFF	OFF
o_bOK	正常完了	ビット	ONの場合、FBの処理が正常完了したことを示します。	OFF
o_bErr	異常完了	ビット	ONの場合、FBの処理が異常完了したことを示します。	OFF
o_uErrId	エラーコード	ワード[符号なし]/ ビット列[16ビット]	異常完了時にエラーコードが格納されます。	0

■動作パラメータ

変数名	名称	データ型	有効範囲	内容	デフォルト値
pbi_uRequestModuleIO	要求先ユニットI/O番号	ワード[符号なし]/ ビット列[16ビット]	03E0H~03E3H, 03FFH	アクセス先のユニットを指定します。 ・03FFH: 自局、管理CPU ・03E0H: マルチCPU1号機 ・03E1H: マルチCPU2号機 ・03E2H: マルチCPU3号機 ・03E3H: マルチCPU4号機	03FFH
pbi_uResendCountMax	最大再送回数	ワード[符号なし]/ ビット列[16ビット]	0~15	pbi_uMonitorTime(到達監視時間)で 指定した監視時間内に完了しない場 合に再送させる回数を設定します。 ・0回~15回	5
pbi_uMonitorTime	到達監視時間	ワード[符号なし]/ ビット列[16ビット]	0, 1~32767	処理完了までの監視時間を指定しま す。監視時間内に完了しない場合は pbi_uResendCountMax(最大再送回 数)で指定した回数まで再送されま す。 ・0: 10秒 ・1~32767: 1~32767秒	0

■公開変数

変数名	名称	データ型	内容	デフォルト値
pbo_uResendCount	再送回数	ワード[符号なし]/ビット列 [16ビット]	再送を行った回数(結果)が格納されます。	0
pbo_u4ErrTime	異常発生時刻	ワード[符号なし]/ビット列 [16ビット](0..3)	異常発生時の時計データが格納されます(異常時のみ)。 1ワード目 ・上位8ビット: 月(01H~12H) ・下位8ビット: 年(00H~99H)西暦下2桁 2ワード目 ・上位8ビット: 時(00H~23H) ・下位8ビット: 日(01H~31H) 3ワード目 ・上位8ビット: 秒(00H~59H) ・下位8ビット: 分(00H~59H) 4ワード目 ・上位8ビット: 年(00H~99H)西暦上2桁 ・下位8ビット: 曜日(00H(日)~06H(土))	0
pbo_uErrIP_Address_ 3rd_4th	異常検出局IPアドレス (第3, 4オクテット)	ワード[符号なし]/ビット列 [16ビット](0..1)	異常を検出した局のIPアドレス(第3オクテット, 第4オ クテット)が格納されます。 ^{*1} 例: 192.168.1.2の場合 0102h	0
pbo_uErrIP_Address_ 1st_2nd	異常検出局IPアドレス (第1, 2オクテット)	ワード[符号なし]/ビット列 [16ビット](0..1)	異常を検出した局のIPアドレス(第1オクテット, 第2オ クテット)が格納されます。 ^{*1} 例: 192.168.1.2の場合 C0A8h	0

*1 専用命令が異常完了した場合のみ値が格納されます。

異常検出機器IPアドレス(第3, 4オクテット)および異常検出機器IPアドレス(第1, 2オクテット)には入力引数の対象局アドレスに設定した値が格納されます。

機能

項目	内容	
対象機器	対象ユニット	MELSEC MXコントローラ(CC-Link IE TSN用ポート)
	対象CPU	MELSEC MXコントローラ
	対象エンジニアリングツール	GX Works3
使用言語	ラダー	
ラベル使用量	ラベル: 1076点 ラッチラベル: 0点 プログラムに組み込んだラベル使用量は、使用するコントローラ、引数に指定したデバイスやGX Works3のオプション設定によって異なります。GX Works3のオプション設定については、下記を参照してください。  GX Works3 オペレーティングマニュアル	
機能説明	・i_bEN(実行指令)のONで、SLMP対応機器のデバイスデータを書き込みます。 ・本FBは、相手機器のIPアドレスを指定して実行します。 ・本FBは、SLMPのWriteコマンド(コマンド: 1401)を使用します。SLMPのコマンドの伝文はバイナリコードになります。( SLMPリファレンスマニュアル)	
FBコンパイル方式	マクロ型	
FB動作	パルス型(複数スキャン実行型)	

項目	内容
入出力信号の動き	• 正常完了の場合 • 異常完了の場合(専用命令異常完了時) (1) エラーコード (2) 異常発生時刻 (3) 異常検出機器IPアドレス • 異常完了の場合(専用命令正常完了/終了コード異常時) (1) エラーコード

項目	内容
注意事項	- 本FBは、エラーから復帰する処理は含んでいません。エラー処理については、システムや要求動作に合わせ、別途作成してください。 - 本FBは、SLMPSND命令を使用します。SLMPSND命令は、対象機器が異常応答を返した場合でも正常完了になります。そのため、本FBでは専用命令の完了状態に加え、応答フレームの終了コードで正常完了/異常完了を判定します。終了コードで異常完了と判定した場合は、o_uErrId(エラーコード)には終了コードを、pbo_u4ErrTime(異常発生時刻)及び、pbo_uErrIP_Address_3rd_4th(異常検出機器IPアドレス(第3, 4オクテット)), pbo_uErrIP_Address_1st_2nd(異常検出機器IPアドレス(第1, 2オクテット))に、0(初期値)が格納されているか確認してください。異常原因及び処置方法は使用しているSLMP対応機器のマニュアルを参照してください。( SLMPリファレンスマニュアル) - i_bEN(実行指令)は、o_bOK(正常完了)またはo_bErr(異常完了)がONした後でOFFしてください。i_bEN(実行指令)のOFFによってo_bOK(正常完了)とo_bErr(異常完了)がOFFし、o_uErrId(エラーコード)が0クリアされます。 - 本FBでは、SLMPの拡張指定でアクセスするデバイス(リンクダイレクトデバイスなど)の書き込みはできません。 - 本FBでは、他ネットワークの局を対象局とすることはできません。 - リモートパスワードが設定されている相手機器のポートに対して本FBを実行する場合は、リモートパスワードのアンロック処理を行ってから実行してください。リモートパスワードが設定されている相手機器のポートに対して本FBを実行した場合は、エラーとなります。 - 対象局は、SLMPコマンドの「Write(コマンド: 1401)」に対応している必要があります。 - 本FBは、バイナリコードによる交信のみが対象です(ASCIIコードによる交信はできません)。 - 本FBは、UDPで通信します。相手機器も同じ設定にしてください。 - 本FBは、プログラム別ラベル初期値を使用します。コントローラのブート運転において、本FBを使用したプログラムファイルをブートファイル設定で指定する場合、同様にプログラム別ラベル初期値ファイルもブートファイル設定で指定してください。なお、1914ページエラーコードに記載されていないエラーコードが発生した場合は、ブートファイル設定でプログラム別ラベル初期値ファイルが設定されていない可能性があります。ブートファイル設定でプログラム別ラベル初期値ファイルを指定してください。 - 本FBを実行した場合に、コントローラで演算異常が発生する場合があります。その場合は、イベント履歴の演算異常の詳細情報を確認して、該当ユニットFBの入力引数を見直してください。

■エラーコード

エラーコード	内容	処置方法
100H	引数のデバイス点数(i_uDevicePoints)に範囲外の値が設定されています。	デバイス点数(i_uDevicePoints)の範囲を範囲内に修正してください。
100H以外	SLMPSND命令のエラーコードを確認し処置してください。  1120ページ G(P).SLMPSND, J(P).SLMPSND, SP.SLMPSND	

Point

本章のファンクションブロックはコントローラのユニットFBです。エンジニアリングツールの部品選択ウィンドウの[ユニット]⇒[ユニットFB]から使用してください。

45.1 リモートリセット

M+MX-TSN_RemoteReset2_IP

IPアドレス指定により対象局に対して、SLMP要求のリモートリセットを送信します。

ラダー	FBD/LD
M+MX-TSN_RemoteReset2_IP B: i_bEN o_bENO: B DUT: i_stModule o_bOK: B UW: i_u2TargetAddress o_bErr: B UW: i_uTarget_Port_No o_uErrId: UW UW: i_uChannel pbo_u4ErrTime 0 pbo_uErrIP_Address_3rd_4th 0 pbo_uErrIP_Address_1st_2nd 0	M+MX-TSN_RemoteReset2_IP i_bEN o_bENO i_stModule o_bOK i_u2TargetAddress o_bErr i_uTarget_Port_No o_uErrId i_uChannel

ST

```
M_MX_TSN_RemoteReset2_IP(  
  i_bEN:= ?BOOL?,  
  i_stModule:= ?M+MX_TSN?,  
  i_u2TargetAddress:= ?WORD(0..1)?,  
  i_uTarget_Port_No:= ?WORD?,  
  i_uChannel:= ?WORD?,  
  o_bENO=> ?BOOL?,  
  o_bOK=> ?BOOL?,  
  o_bErr=> ?BOOL?,  
  o_uErrId=> ?WORD?  
);
```

設定データ

■入力引数

変数名	名称	データ型	有効範囲	内容
i_bEN	実行指令	ビット	—	ON: FBを起動します。 OFF: FBを起動しません。
i_stModule	ユニットラベル	構造体	—	FBを実行するコントローラのユニットラベルを指定します。 (例: MX_TSN)
i_u2TargetAddress	相手機器IPアドレス	ワード[符号なし]/ビット列 [16ビット](0..1)	0.0.0.1~253.255.255.254(00000001H~DFFFFFFEH)	相手機器のIPアドレスを指定します。ラベルで指定する場合、データ型に配列を使用してください。 ・00000001H~DFFFFFFEH 第4オクテットは1~254(FEH)を指定してください。 b15b8b7b0 +0 34 +1 12 1~4: IPアドレスのオクテットを示します。
i_uTargetPort_No	相手機器ポート番号	ワード[符号なし]/ビット列 [16ビット]	1~65534	相手機器のUDPポート番号を指定します。 指定するポート番号は、相手機器のマニュアルを確認してください。
i_uChannel	自局使用チャンネル	ワード[符号なし]/ビット列 [16ビット]	1~17	自局が使用するチャンネルを指定します。*1

*1 本FBでシリアル番号なしフレームによる通信を行う際は、自局使用チャンネルに1を指定します。2~9を指定した場合は、シリアル番号ありフレームによる通信をします。10~17を指定した場合は、局番拡張フレームにより通信します。

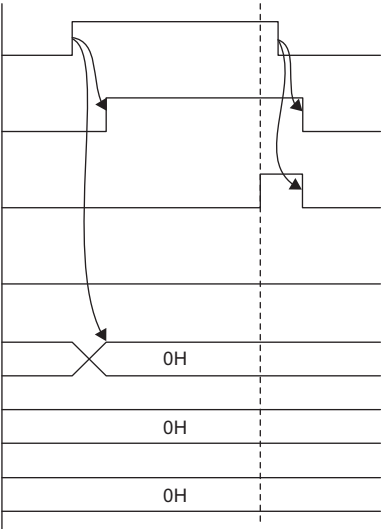
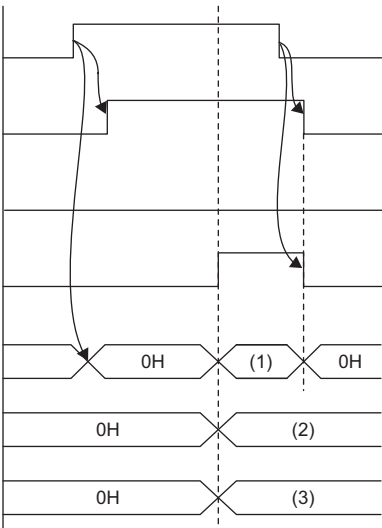
■出力引数

変数名	名称	データ型	内容	デフォルト値
o_bENO	実行状態	ビット	ON: 実行指令ON OFF: 実行指令OFF	OFF
o_bOK	正常完了	ビット	ONの場合、FBの処理が正常完了したことを示します。	OFF
o_bErr	異常完了	ビット	ONの場合、FBの処理が異常完了したことを示します。	OFF
o_uErrId	エラーコード	ワード[符号なし]/ビット列 [16ビット]	異常完了時にエラーコードが格納されます。	0

■公開変数

変数名	名称	データ型	内容	デフォルト値
pbo_u4ErrTime	異常発生時刻	ワード[符号なし]/ビット列 [16ビット](0..3)	異常発生時の時計データが格納されます(異常時のみ)。*1 1ワード目 ・上位8ビット: 月(01H~12H) ・下位8ビット: 年(00H~99H)西暦下2桁 2ワード目 ・上位8ビット: 時(00H~23H) ・下位8ビット: 日(01H~31H) 3ワード目 ・上位8ビット: 秒(00H~59H) ・下位8ビット: 分(00H~59H) 4ワード目 ・上位8ビット: 年(00H~99H)西暦上2桁 ・下位8ビット: 曜日(00H(日)~06H(土))	0
pbo_uErrIP_Address_3rd_4th	異常検出機器IPアドレス(第3, 4オクテット)	ワード[符号なし]/ビット列 [16ビット]	異常を検出した局のIPアドレス(第3オクテット, 第4オクテット)が格納されます。*1 例: 192.168.1.2の場合 0102h	0
pbo_uErrIP_Address_1st_2nd	異常検出機器IPアドレス(第1, 2オクテット)	ワード[符号なし]/ビット列 [16ビット]	異常を検出した局のIPアドレス(第1オクテット, 第2オクテット)が格納されます。*1 例: 192.168.1.2の場合 C0A8h	0

*1 専用命令が異常完了した場合のみ値が格納されます。
異常検出機器IPアドレス(第3, 4オクテット)および異常検出機器IPアドレス(第1, 2オクテット)には入力引数の対象局アドレスに設定した値が格納されます。

項目	内容	
対象機器	対象ユニット	MELSEC MXコントローラ(CC-Link IE TSN用ポート)
	対象CPU	MELSEC MXコントローラ
	対象エンジニアリングツール	GX Works3
使用言語	ラダー	
ラベル使用量	ラベル: 72点 ラッチラベル: 0点 プログラムに組み込んだラベル使用量は、使用するコントローラ、引数に指定したデバイスやGX Works3のオプション設定によって異なります。GX Works3のオプション設定については、下記を参照してください。 GX Works3 オペレーティングマニュアル	
機能説明	i_bEN(実行指令)のONで、対象局に対して、SLMP要求のリモートリセットを送信します。	
FBコンパイル方式	マクロ型	
FB動作	パルス型(複数スキャン実行型)	
FB_ENの入力条件	なし	
入出力信号の動き	- 正常完了の場合  - 異常完了の場合(専用命令異常完了時)  (1) エラーコード (2) 異常発生時刻 (3) 異常検出局アドレス	

項目	内容
注意事項	• 本FBは、エラーから復帰する処理は含んでいません。エラー処理については、システムや要求動作に合わせ、別途作成してください。 • 本FBは、SLMPSND命令を使用します。SLMPSND命令は、対象機器が異常応答を返した場合でも正常完了されます。 • SLMPSND命令が正常完了した場合は、公開変数の異常発生時刻、pbo_uErrIP_Address_3rd_4th(異常検出機器IPアドレス(第3, 4オクテット)), pbo_uErrIP_Address_1st_2nd(異常検出機器IPアドレス(第1, 2オクテット))には、値が格納されません。^{*1} • i_bEN(実行指令)は、o_bOK(正常完了)またはo_bErr(異常完了)がONした後でOFFしてください。i_bEN(実行指令)のOFFによってo_bOK(正常完了)とo_bErr(異常完了)がOFFし、o_uErrId(エラーコード)が0クリアされます。 • o_bOK(正常完了)は、リモートリセットの要求送信が正常完了すると、ONします。実際に対象局がリモートリセットされるかは対象局の状態で異なります。 • 本FBでは、他ネットワークの局を対象局とすることができません。 • 相手機器はSLMPコマンドの「Remote Reset(1006H)」に対応している必要があります。 • 本FBは、UDPで通信します。相手機器も同じ設定にしてください。

*1 異常発生時刻、異常検出機器IPアドレス(第3, 4オクテット)および異常検出機器IPアドレス(第1, 2オクテット)に0(初期値)が格納されている場合は、使用しているSLMP対応機器のマニュアルで確認および処置を行ってください。

エラー

本FBで発生するエラーコードはSLMPSND命令と同じです。

📖 1120ページ G(P).SLMPSND, J(P).SLMPSND, SP.SLMPSND

付1 命令処理時間

各命令の処理時間を示します。

処理時間はソース、ディスティネーションの内容や、プログラムに応じた処理の最適化によって変動するため、表中の値は目安として参照してください。

命令の分類			命令名	条件	処理時間[ns]	
大分類	中分類	小分類			MXR300-16, MXR300-32, MXR300-64	MXR500-128, MXR500-256
					最小	最小
シーケンス命令	接点命令	—	LD	ビットデバイス	0.38	0.28
				ビット型ラベル	0.40	0.29
	出力命令	—	OUT	ビットデバイス	15.29	11.12
				ビット型ラベル	0.33	0.24
			OUT T	—	28.50	20.70
			OUT LT	—	98.90	73.80
基本命令	データ転送命令	—	MOV	—	0.68	0.5
			BMOV	n=1	106	76.6
				n=1024	262	191
			FMOV	n=1	64.2	46.7
				n=1024	110	79.7
			算術演算命令	インクリメント	INC	—
	四則演算命令(整数)	+		—	0.64	0.46
		-		—	0.64	0.46
		*		—	0.63	0.46
		/	—	0.63	0.46	

命令の分類			命令名	条件	処理時間[ns]	
大分類	中分類	小分類			MXR300-16, MXR300-32, MXR300-64	MXR500-128, MXR500-256
					最小	最小
応用命令	浮動小数点命令	四則演算命令(単精度実数)	E+	s1=2 ¹²⁶ , s2=2 ¹²⁷	46.8	34
			E-	s1=2 ¹²⁶ , s2=2 ¹²⁷	46.8	34
			E*	s1=2 ⁶³ , s2=2 ⁶⁴	46.8	34
			E/	s1=2 ¹²⁶ , s2=2 ¹²⁷	57.7	41.9
		四則演算命令(倍精度実数)	ED+	s1=2 ¹⁰²² , s2=2 ¹⁰²³	44.1	32.1
			ED-	s1=2 ¹⁰²² , s2=2 ¹⁰²³	44.1	32.1
			ED*	s1=2 ⁵¹¹ , s2=2 ⁵¹²	44.3	32.1
			ED/	s1=2 ¹⁰²² , s2=2 ¹⁰²³	57.7	41.9
		三角関数(単精度実数)	SIN	s=E0.7853982(45度)	83.3	60.5
		三角関数(倍精度実数)	SIND	s=E0.7853982(45度)	62.6	45.5
		実数データ転送(単精度実数)	EMOV	s=内部デバイス, d=内部デバイス	0.69	0.5
				s=E1.23, d=内部デバイス	0.63	0.46
		実数データ転送(倍精度実数)	EDMOV	s=内部デバイス, d=内部デバイス	0.68	0.5
				s=E1.23, d=内部デバイス	0.63	0.46
	PID演算命令	—	PID	—	103	74.9
	PID制御命令	不完全微分	S.PIDINIT	1ループ	358	261
				32ループ	1250	905
			S.PIDCONT	1ループ	115	83.6
				32ループ	2140	1360
	複数データの文字列変換結合	—	SPF	s1=%5d, s2=12345	638	467
				s1=%5s, s2="12345"	601	436
				s1=%5wf, s2=1.2345	1810	1330
				s1=%5d%5s, s2+0=12345, s2+1="12345"	1230	883
	構造化命令	繰り返し命令	FOR	—	0.13	0.09
			NEXT	—	0.04	0.03
		サブルーチンコール命令	CALL	ローカルポインタ	352	243
				グローバルポインタ	395	278
			RET	自プログラムまでのリターン	235	163
				他プログラムまでのリターン	264	186
	ユニットアクセス命令	—	TO	n=1	4110	3740
				n=1024	24500	24000
			FROM	n=1	5890	5470
				n=1024	43500	41400
専用命令	内蔵Ethernet用命令	SLMP伝文送信	GP.SLMPSND	Read(コマンド: 0401H)(ワード単位 読出し): 読出し点数 =1点	778	567
	インテリジェント機能ユニット用命令	他局シーケンサのデータ読出し	GP.READ	—	66900	64900

命令の分類			命令名	条件	処理時間[ns]	
大分類	中分類	小分類			MXR300-16, MXR300-32, MXR300-64	MXR500-128, MXR500-256
					最小	最小
FB/FUN呼出し	FB呼出	FB呼出	—	ローカルFB	286	208
				グローバルFB	279	205
	FUN呼出	FUN呼出	—	—	274	204
	インラインST呼出	インラインST呼出	—	—	2.72	1.98
ST制御構文	条件分岐	—	IF	—	0.65	0.47
	繰り返し	—	CASE	—	2.52	1.84
			ST_FOR	—	7.45	5.44

命令処理時間の加算時間

各命令の処理時間に対し、条件に応じて加算される処理時間を示します。

実行条件チェック時間

実行条件が“立上り実行”または“立下り実行”となっている命令は、実行条件のチェックに1命令あたり下表に示す処理時間がかかります。

(たとえば、MOVP命令の処理時間は、“MOV命令の処理時間+実行条件チェック加算時間”となります。

実行条件	処理時間[ns]	
	MXR300-16, MXR300-32, MXR300-64	MXR500-128, MXR500-256
立上り実行/立下り実行	28.62	20.80

デバイスアクセス時間

下表に示すデバイスを引数に指定した場合、1デバイスあたり下表に示す処理時間が加算されます。

デバイス名	指定箇所	処理時間[ns]	
		MXR300-16, MXR300-32, MXR300-64	MXR500-128, MXR500-256
ユニットアクセスデバイス(U¥G)およびユニットラベル(ラベル名に_Dが付くもの)	ソース	6.37	5.45
	デスティネーション	3.82	3.74
リンクダイレクトデバイス(J¥)およびユニットラベル(ラベル名に_Dが付くリンクデバイス)	ソース	12.90	12.10
	デスティネーション	10.30	9.25

付2 ユニット専用命令使用可否

付

コントローラでMELSEC iQ-Rシリーズのユニットを使用する場合の、ユニット専用命令の使用可否を示します。(2025年5月時点)

ユニット専用命令の詳細は、下記を参照してください。

📖 MELSEC iQ-R プログラミングマニュアル(ユニット専用命令編)

Ethernet, CC-Link IE, MELSECNET/H共通命令

○: 使用可, ×: 使用不可

■リンク専用命令

命令記号	使用可否
JP.READ	○
GP.READ	○
JP.SREAD	○
GP.SREAD	○
JP.WRITE	○
GP.WRITE	○
JP.SWRITE	○
GP.SWRITE	○
JP.SEND	○
GP.SEND	○
JP.RECV	○
GP.RECV	○
G.RECVS	○
Z.RECVS	○
J.ZNRD	○
JP.ZNRD	○
J.ZNWR	○
JP.ZNWR	○
J.REQ	○
JP.REQ	○
G.REQ	○
GP.REQ	○
J.REQ	○
JP.REQ	○
G.REQ	○
GP.REQ	○

■CC-Link専用命令

命令記号	使用可否
J.RIRD	○
JP.RIRD	○
G.RIRD	○
GP.RIRD	○
J.RIWT	○
JP.RIWT	○
G.RIWT	○
GP.RIWT	○

Ethernet用命令

○: 使用可, ×: 使用不可

■オープン/クローズ処理命令

命令記号	使用可否
GP.CONOPEN	○
GP.CONCLOSE	○
GP.OPEN	○
ZP.OPEN	○
GP.CLOSE	○
ZP.CLOSE	○

■ソケット通信用命令

命令記号	使用可否
GP.SOCRCV	○
G.SOCRCVS	○
GP.SOCSND	○

■固定バッファ交信用命令

命令記号	使用可否
GP.BUFRCV	○
ZP.BUFRCV	○
G.BUFRCVS	○
Z.BUFRCVS	○
GP.BUFSND	○
ZP.BUFSND	○

■再イニシャル処理

命令記号	使用可否
G.UINI	○
GP.UINI	○
Z.UINI	○
ZP.UINI	○

■通信プロトコル支援機能の登録プロトコル実行

命令記号	使用可否
GP.ECPRTCL	○

■エラー情報クリア

命令記号	使用可否
GP.ERRCLEAR	○
ZP.ERRCLEAR	○

■エラー情報読出し

命令記号	使用可否
GP.ERRRD	○
ZP.ERRRD	○

CC-Link IE TSN用命令

○: 使用可, ×: 使用不可

■リモート局のデータ読出し

命令記号	使用可否
JP.REMFR	○
ZP.REMFR	○
JP.REMFRD	○
GP.REMFRIP	○
GP.REMFRDIP	○

■リモート局へデータ書込み

命令記号	使用可否
JP.REMTO	○
ZP.REMTO	○
JP.REMTOD	○
GP.REMTOIP	○
GP.REMTODIP	○

■パラメータ設定

命令記号	使用可否
G.CCPASET	○
GP.CCPASET	○
G.CCPASET	○
GP.CCPASET	○

■自局局番/IPアドレス設定

命令記号	使用可否
G.UINI	○
GP.UINI	○
Z.UINI	○
ZP.UINI	○

■SLMP伝文送信

命令記号	使用可否
J.SLMPSND	○
JP.SLMPSND	○
G.SLMPSND	○
GP.SLMPSND	○

CC-Link IEコントローラネットワーク，MELSECNET/H共通命令

○: 使用可，×: 使用不可

■リモートRUN

命令記号	使用可否
J.RRUN	○
JP.RRUN	○
G.RRUN	○
GP.RRUN	○
Z.RRUN	○
ZP.RRUN	○

■リモートSTOP

命令記号	使用可否
J.RSTOP	○
JP.RSTOP	○
G.RSTOP	○
GP.RSTOP	○
Z.RSTOP	○
ZP.RSTOP	○

■他局シーケンサの時計データ読出し

命令記号	使用可否
J.RTMRD	○
JP.RTMRD	○
G.RTMRD	○
GP.RTMRD	○
Z.RTMRD	○
ZP.RTMRD	○

■他局シーケンサの時計データ書込み

命令記号	使用可否
J.RTMWR	○
JP.RTMWR	○
G.RTMWR	○
GP.RTMWR	○
Z.RTMWR	○
ZP.RTMWR	○

■自局局番設定

命令記号	使用可否
G.UINI	○
GP.UINI	○
Z.UINI	○
ZP.UINI	○

CC-Link IEフィールドネットワーク用命令

○: 使用可, ×: 使用不可

■インテリジェントデバイス局/リモートデバイス局のデータ読出し

命令記号	使用可否
JP.REMFR	○
ZP.REMFR	○
JP.REMFRD	○

■インテリジェントデバイス局/リモートデバイス局へデータ書込み

命令記号	使用可否
JP.REMTO	○
ZP.REMTO	○
JP.REMTOD	○

■インテリジェントデバイス局のシステム構成ユニット形名情報読出し

命令記号	使用可否
J.SINFYRD	○
JP.SINFYRD	○
G.SINFYRD	○
GP.SINFYRD	○

■インテリジェントデバイス局のシステム構成ユニット状態情報読出し

命令記号	使用可否
J.SINFSTRD	○
JP.SINFSTRD	○
G.SINFSTRD	○
GP.SINFSTRD	○

■パラメータ設定

命令記号	使用可否
G.CCPASET	○
GP.CCPASET	○
G.CCPASETR	×
GP.CCPASETR	×

■自局局番設定

命令記号	使用可否
G.UINI	○
GP.UINI	○
Z.UINI	○
ZP.UINI	○

■SLMPフレーム送信

命令記号	使用可否
J.SLMPREQ	○
JP.SLMPREQ	○
G.SLMPREQ	○
GP.SLMPREQ	○

CC-Link用命令

○: 使用可, ×: 使用不可

■対象局からのデータ読出し

命令記号	使用可否
G.RIRD	○
GP.RIRD	○

■対象局へのデータ書込み

命令記号	使用可否
G.RIWT	○
GP.RIWT	○

■インテリジェントデバイス局バッファメモリからのデータ読出し

命令記号	使用可否
G.RIRCV	○
GP.RIRCV	○

■インテリジェントデバイス局バッファメモリへのデータ書込み

命令記号	使用可否
G.RISEND	○
GP.RISEND	○

■自動更新バッファからのデータ読出し

命令記号	使用可否
G.RIFR	○
GP.RIFR	○

■自動更新バッファへのデータ書込み

命令記号	使用可否
G.RITO	○
GP.RITO	○

■ネットワークパラメータ設定

命令記号	使用可否
G.RLPASET	○
GP.RLPASET	○

■リモートデバイス局宛のメッセージ伝送

命令記号	使用可否
G.RDMSG	○
GP.RDMSG	○

シリアルコミュニケーション用命令

○: 使用可, ×: 使用不可

■オンデマンド機能によるデータ送信

命令記号	使用可否
G.ONDEMAND	○
GP.ONDEMAND	○

■通信プロトコル支援機能の登録プロトコル実行

命令記号	使用可否
G.CPRTCL	○
GP.CPRTCL	○

■無手順プロトコルによるデータ送信

命令記号	使用可否
G.OUTPUT	○
GP.OUTPUT	○

■無手順プロトコルによるデータ受信

命令記号	使用可否
G.INPUT	○

■双方向プロトコルによるデータ送信

命令記号	使用可否
G.BIDOUT	○
GP.BIDOUT	○

■双方向プロトコルによるデータ受信

命令記号	使用可否
G.BIDIN	○
GP.BIDIN	○

■データ送信/受信データ読出し

命令記号	使用可否
G.SPBUSY	○
GP.SPBUSY	○

■割込みプログラムでのデータ受信

命令記号	使用可否
Z.BUFRCVS	○

■ユーザ登録フレームのデータ送信

命令記号	使用可否
G.PRR	○
GP.PRR	○

■受信データクリア

命令記号	使用可否
ZP.CSET	○

■シーケンサCPU監視登録/解除

命令記号	使用可否
ZP.CSET	○

■初期設定

命令記号	使用可否
ZP.CSET	○

■ユーザ登録フレームの登録

命令記号	使用可否
G.PUTE	○
GP.PUTE	○

■ユーザ登録フレームの読出し

命令記号	使用可否
G.GETE	○
GP.GETE	○

■モード切換え

命令記号	使用可否
ZP.UINI	○

アナログ用命令

○: 使用可, ×: 使用不可

■モード切換え

命令記号	使用可否
G.OFFGAN	○
GP.OFFGAN	○

■ユーザレンジ設定の読出し

命令記号	使用可否
G.OGLOAD	○
GP.OGLOAD	○

■ユーザレンジ設定の復元

命令記号	使用可否
G.OGSTOR	○
GP.OGSTOR	○

位置決め用命令

○: 使用可, ×: 使用不可

■絶対位置復元

命令記号	使用可否
G.ABRST1	○
G.ABRST2	○
G.ABRST3	○
G.ABRST4	○
Z.ABRST1	○
Z.ABRST2	○
Z.ABRST3	○
Z.ABRST4	○

■位置決め始動

命令記号	使用可否
GP.PSTR1	○
GP.PSTR2	○
GP.PSTR3	○
GP.PSTR4	○
ZP.PSTR1	○
ZP.PSTR2	○
ZP.PSTR3	○
ZP.PSTR4	○

■ティーチング

命令記号	使用可否
GP.TEACH1	○
GP.TEACH2	○
GP.TEACH3	○
GP.TEACH4	○
ZP.TEACH1	○
ZP.TEACH2	○
ZP.TEACH3	○
ZP.TEACH4	○

■ユニットバックアップ(フラッシュ ROM書込み)

命令記号	使用可否
GP.PFWRT	○
ZP.PFWRT	○

■ユニット初期化

命令記号	使用可否
GP.PINIT	○
ZP.PINIT	○

高速データロガー用命令

○: 使用可, ×: 使用不可

■ファイルアクセス命令

命令記号	使用可否
Z.RCPWRITE	×
ZP.RCPWRITE	×
Z.RCPREAD	×
ZP.RCPREAD	×

C言語インテリジェント機能ユニット用命令

○: 使用可, ×: 使用不可

■ユーザ関数実行命令

命令記号	使用可否
G.CEEXECUTE	×
GP.CEEXECUTE	×

■データ解析関数実行命令

命令記号	使用可否
G.DIGIFLTR	×
GP.DIGIFLTR	×
G.ENVCALC	×
GP.ENVCALC	×
G.FFTSPECT	×
GP.FFTSPECT	×

■統計解析関数実行命令

命令記号	使用可否
G.LEASTSQ	×
GP.LEASTSQ	×
G.MTUNIT	×
GP.MTUNIT	×
G.MTMD	×
GP.MTMD	×
G.MULTIRG	×
GP.MULTIRG	×

デバイスネット用命令

○: 使用可, ×: 使用不可

■マスタ機能受信データエリア読出し

命令記号	使用可否
G.DNTMRD	○

■マスタ機能送信データエリア書込み

命令記号	使用可否
G.DNTMWR	○

■スレーブ機能受信データエリア読出し

命令記号	使用可否
G.DNTSRD	○

■スレーブ機能送信データエリア書込み

命令記号	使用可否
G.DNTSWR	○

付3 PIDの3定数の求め方

PID演算命令のオートチューニング機能は、リミットサイクル法とステップ応答法の2種類の方法があります。それぞれの方法の概要を説明します。

リミットサイクル法の概要

入力値の振幅(a)、振動周期(τ , τ_{on})を求め、下記「動作特性と3定数」の式によって、比例ゲイン(K_p)、積分時間(T_I)、微分時間(T_D)を算出する方法としてリミットサイクル法について説明します。

■リミットサイクル法とは

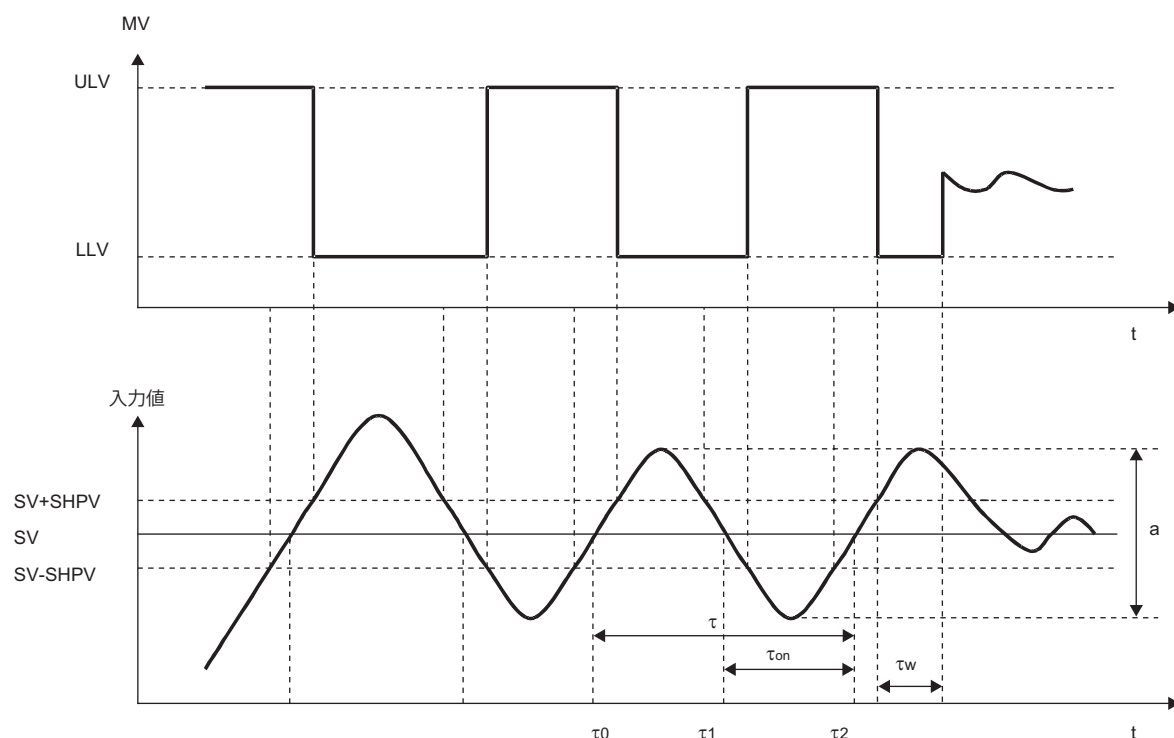
二位値制御(偏差により、出力上限値(ULV)と出力下限値(LLV)を切り換えて出力)を行ったときの、入力値の変化を測定し、PIDの3定数を求める方法です。

■動作特性(逆動作の例)

チューニングサイクル終了後、 τ_W の間は、出力値(MV)は出力下限値(LLV)を保持し、通常のPID制御に移ります。

$\tau_W = (50 + K_W) / 100 \times (\tau - \tau_{on})$ で求められ、ウェイト設定パラメータ(K_W)は(s3)+28で設定できます。

(設定範囲 $K_W = -50 \sim 32717[\%]$, 異常範囲指定時は $\tau_W = 0$ として動作)



ULV: 出力上限値

LLV: 出力下限値

SV: 目標値

t: 時間

SHPV: PV値スレッシュホールド(ヒステリシス)幅

■動作特性と3定数

制御形態	比例ゲイン(K _p)[%]	積分時間(T _i)[×100ms]	微分時間(T _d)[×10ms]
比例制御(P動作)のみ	$\frac{1}{a} (\text{ULV-LLV}) \times 100$	—	—
PI制御(PI動作)	$\frac{0.9}{a} (\text{ULV-LLV}) \times 100$	$33 \times \tau_{\text{on}} \left(1 - \frac{\tau_{\text{on}}}{\tau}\right)$	—
PID制御(PID動作)	$\frac{1.2}{a} (\text{ULV-LLV}) \times 100$	$20 \times \tau_{\text{on}} \left(1 - \frac{\tau_{\text{on}}}{\tau}\right)$	$50 \times \tau_{\text{on}} \left(1 - \frac{\tau_{\text{on}}}{\tau}\right)$

ステップ応答法の概要

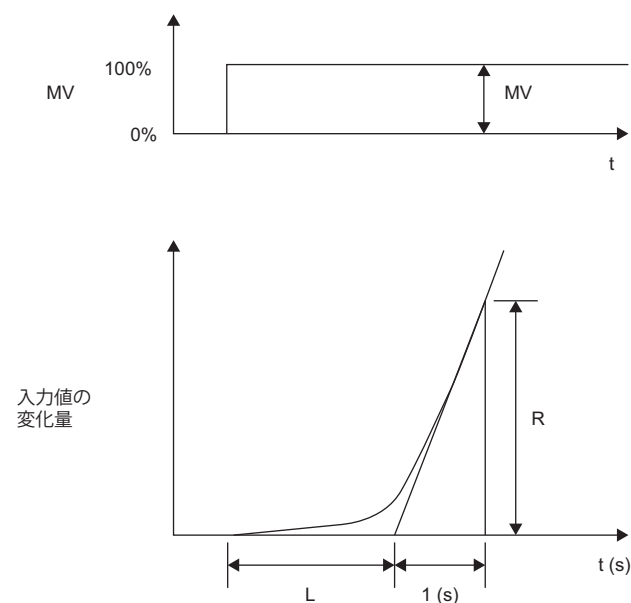
PIDの3定数(比例ゲイン(K_p), 積分時間(T_i), 微分時間(T_d))の最適値を求める方法としてステップ応答法について説明します。

■ステップ応答法とは

制御システムに対して, 0→100%^{*1}のステップ状の出力を与えることによって, 入力変化から求められる動作特性(最大傾斜(R), 無駄時間(L))よりPIDの3定数を求める方法です。

*1 ステップ状の出力は, 0→75%や0→50%でも求めることができます。

■動作特性



MV: 出力値
L: 無駄時間
R: 最大傾斜
t: 時間

■動作特性と3定数

制御形態	比例ゲイン(K _p)[%]	積分時間(T _i)[×100ms]	微分時間(T _d)[×10ms]
比例制御(P動作)のみ	$\frac{1}{RL} \times MV \times 100$	—	—
PI制御(PI動作)	$\frac{0.9}{RL} \times MV \times 100$	33L	—
PID制御(PID動作)	$\frac{1.2}{RL} \times MV \times 100$	20L	50L

索引

Symbols

-	93
*	93
**	93
/	93
&	93
+	93
<	93
<=	93
<>	93
=	93
>	93
>=	93

0~9

16ビットデータ(ワードデータ)	32
32ビットデータ(ダブルワードデータ)	35

A

AND	93
ASCII	104,116

C

CASE	98
CC-Link IE TSNマスタ・ローカルユニット	24
CC-Link IEコントローラネットワーク搭載ユニット	24

E

EN	62,74
ENO	62,74
EXIT	98

F

FBファイル	66,75
FOR...DO	98,102
FUNファイル	61,63

I

IF THEN	98
IF...ELSE	98
IF...ELSIF	98
iQSS対応機器	24

M

MOD	93

N

NOT	93

O

OR	93

P

PID制御命令	1047
PID命令の種類	1031
PROFILE_POSITIONING_DATA_EXTENDED	1433

R

REPEAT...UNTIL	98
RETURN	97

S

STRING	104,116

U

Unicode	104

W

WHILE...DO	98
WSTRING	104,116

X

XOR	93

い

インスタンス	68

お

応用命令	529

か

外部変数	67
型指定	105,116
型変換	96,109

き

基本命令	252
局サブID番号	24

さ

サブルーチン型ファンクションブロック	66
サブルーチンプログラム	60

し

シーケンス命令	200
実行条件	43
実数データ(浮動小数点データ)	38

シフトJIS	104,116
出力変数	62,66

す

ステートメント	90

そ

ソース(s)	26

た

代入文	95
単精度実数データ	38

て

データリンク	24
ディスティネーション(d)	26
デバイス局	21
デバイス数/転送数/データ数/文字列数などの数値(n)	27

な

内部変数	67

に

入出力変数	66
入力変数	62,66

の

ノート	90

は

倍精度実数データ	39
バッファメモリ	21

ひ

ビットデータ	31

ふ

ファンクション(FUN)	59,61
ファンクションブロック(FB)	59,65
ファンクションブロック呼び出し文	97
ファンクション呼び出し文	97
プログラム	63,75
プログラムブロック	60

ま

マスタ局	21

め

命令の構成	26
メインルーチンプログラム	60

も

文字列	104,116
文字列[Unicode]	104,116
文字列データ	41

よ

予約語	92

ら

ラダー言語	85

り

リンクデバイス	22

ろ

ローカル局	22

わ

割込みプログラム	60

命令索引

Symbols

-(P)(_U)	266,268
*(P)(_U)	278
/(P)(_U)	280
+(P)(_U)	262,264
\$+(P)	719,721
\$MOV(P)	723
\$MOV(P)_WS	725

A

ABS(_E)	1253
ACOS(_E)	1264
ACOS(P)	867
ACOSD(P)	879
ADD(_E)	1266
ADDONEXE(P)	1079
ADDON_MONI	1425
ADDON_PARAM	1425
ADD_TIME(_E)	1324
ADRSET(P)	953
ADV_CAM_MONI	1445
ADV_CAM_PARAM	1444
ADV_CLUTCH_CMD	1441
ADV_CLUTCH_MONI	1441
ADV_CLUTCH_PARAM	1437
ADV_GEAR_PARAM	1437
ADV_INPUT_MONI	1413
ADV_INPUT_PRM	1413
ADV_INPUT_PRM_CONST	1413
ADV_OUTPUT_CMD	1418
ADV_OUTPUT_MONI	1418
ADV_OUTPUT_PRM	1416
ADV_OUTPUT_PRM_CONST	1415
ADV_RESTORE_PARAM	1446
ADV_SPEEDCHANGE_GEAR_PARAM	1443
ANB	208
AND	200
AND(_E)	1289
AND<(_U)	252
AND<=(_U)	252
AND<>(_U)	252
AND=(_U)	252
AND>(_U)	252
AND>=(_U)	252
AND\$<	716
AND\$<=	716
AND\$<>	716
AND\$=	716
AND\$>	716
AND\$>=	716
ANDD<(_U)	254
ANDD<=(_U)	254
ANDD<>(_U)	254
ANDD=(_U)	254
ANDD>(_U)	254
ANDD>=(_U)	254
ANDD_EQ(_U)	254
ANDD_GE(_U)	254
ANDD_GT(_U)	254
ANDD_LE(_U)	254

ANDD_LT(_U)	254
ANDD_NE(_U)	254
ANDDT<	979
ANDDT<=	979
ANDDT<>	979
ANDDT=	979
ANDDT>	979
ANDDT>=	979
ANDDT_EQ	979
ANDDT_GE	979
ANDDT_GT	979
ANDDT_LE	979
ANDDT_LT	979
ANDDT_NE	979
ANDE<	807
ANDE<=	807
ANDE<>	807
ANDE=	807
ANDE>	807
ANDE>=	807
ANDED<	809
ANDED<=	809
ANDED<>	809
ANDED=	809
ANDED>	809
ANDED>=	809
ANDED_EQ	809
ANDED_GE	809
ANDED_GT	809
ANDED_LE	809
ANDED_LT	809
ANDED_NE	809
ANDE_EQ	807
ANDE_GE	807
ANDE_GT	807
ANDE_LE	807
ANDE_LT	807
ANDE_NE	807
AND_EQ(_U)	252
ANDF	202
ANDFI	205
AND_GE(_U)	252
AND_GT(_U)	252
AND_LE(_U)	252
AND_LT(_U)	252
AND_NE(_U)	252
ANDP	202
ANDPI	205
ANDSTRING_EQ	716
ANDSTRING_GE	716
ANDSTRING_GT	716
ANDSTRING_LE	716
ANDSTRING_LT	716
ANDSTRING_NE	716
ANDTM<	983
ANDTM<=	983
ANDTM<>	983
ANDTM=	983
ANDTM>	983
ANDTM>=	983
ANDTM_EQ	983
ANDTM_GE	983

ANDTM_GT	983
ANDTM_LE	983
ANDTM_LT	983
ANDTM_NE	983
ANI	200
ASC2INT(P)	471
ASIN(_E)	1263
ASIN(P)	865
ASIND(P)	877
ATAN(_E)	1265
ATAN(P)	869
ATAND(P)	881
AXES_GROUP_CMD	1407
AXES_GROUP_MONI	1404
AXES_GROUP_PRM	1402
AXES_GROUP_PRM_CONST	1402
AXES_GROUP_REF	1402
AXIS_CMD	1394
AXIS_MONI	1387
AXIS_PRM	1383
AXIS_PRM_CONST	1379
AXIS_REF	1378,1412,1414

B

B-(P)	289,290
B*(P)	298
B/(P)	300
B+(P)	286,287
BACOS(P)	891
BAND(P)(_U)	662
BASIN(P)	889
BATAN(P)	893
BCD(P)	387
BCDDA(P)	751
BCD_TO_DINT(_E)	1200
BCD_TO_INT(_E)	1198
BCD_TO_STRING(_E)	1203
BCOS(P)	885
BDIVISION(P)	300
BDSQRT(P)	917
BIN(P)	391
BINDA(P)(_U)	727
BINHA(P)	737
BITARR_TO_DINT(_E)	1242
BITARR_TO_INT(_E)	1241
BK-(P)(_U)	308
BK+(P)(_U)	306
BKAND(P)	328
BKBCD(P)	443
BKBIN(P)	445
BKCMP<(P)(_U)	256
BKCMP<=(P)(_U)	256
BKCMP<>(P)(_U)	256
BKCMP=(P)(_U)	256
BKCMP>(P)(_U)	256
BKCMP>=(P)(_U)	256
BKCMP_EQ(P)(_U)	256
BKCMP_GE(P)(_U)	256
BKCMP_GT(P)(_U)	256
BKCMP_LE(P)(_U)	256
BKCMP_LT(P)(_U)	256
BKCMP_NE(P)(_U)	256
BKMINUS(P)(_U)	308
BKOR(P)	337
BKPLUS(P)(_U)	306

BKRST(P)	365
BKXNR(P)	355
BKXOR(P)	346
BLKMOVB(P)	527
BMINUS(P)	290
BMOV(P)	506
BMOVL(P)	508
BMULTI(P)	298
BOOL_TO_DINT(_E)	1156
BOOL_TO_DWORD(_E)	1154
BOOL_TO_INT(_E)	1155
BOOL_TO_STRING(_E)	1158
BOOL_TO_TIME(_E)	1157
BOOL_TO_WORD(_E)	1152
BPLUS(P)	287
BREAK(P)	550
BRST(P)	359
BSET(P)	357
BSFL(P)	373
BSFR(P)	371
BSIN(P)	883
BSQRT(P)	915
BTAN(P)	887
BTOW(P)	494
BXCH(P)	522

C

CALL(P)	552
CJ	530
CML(P)	500
CMLB(P)	504
COM(P)	999
CONCAT(_E)	1313
COS(_E)	1261
COS(P)	861
COSD(P)	873
COUNTER_FB_M	1349
CPY_BITARR(_E)	1245
CPY_BIT_OF_INT(_E)	1250
CRC(P)	712
CTD(_E)	1344
CTU(_E)	1342
CTUD(_E)	1346
CYCLE_MONI	1425
CYCLE_PARAM	1425

D

D-(P)(_U)	274,276
D*(P)(_U)	282
D/(P)(_U)	284
D+(P)(_U)	270,272
DABCD(P)	459
DABIN(P)(_U)	447
DAND(P)	324,326
DATE-(P)	969
DATE+(P)	967
DATE2SEC(P)(_U)	975
DATEMINUS(P)	969
DATEPLUS(P)	967
DATERD(P)	963
DATEWR(P)	965
DB-(P)	295,296
DB*(P)	302
DB/(P)	304

EDNEG(P).....	856
EDPLUS(P).....	822
EDSQRT(P).....	905
EFCALL(P).....	572
EGF.....	211
EGP.....	211
El.....	534
EMAX(P).....	927
EMIN(P).....	931
EMINUS(P).....	818
EMOD(P).....	473
EMOV(P).....	857
EMULTI(P).....	828
ENCO(P).....	479
END.....	248
ENEG(P).....	855
EPLUS(P).....	814
EQ(_E).....	1302
EREXP(P).....	853
ESQRT(P).....	903
ESTR(P).....	760
EVAL(P).....	849
EXP(_E).....	1259
EXP(P).....	907
EXPD(P).....	909
EXPT(_E).....	1278

F

FCALL(P).....	559
FDEL(P).....	600
FEND.....	247
FF.....	238
FIFR(P).....	592
FIFW(P).....	596
FILE_LOCATION.....	1424
FIND(_E).....	1322
FINS(P).....	598
FLT2DBL(P).....	848
FLT2DINT(P).....	399
FLT2INT(P).....	395
FLT2UDINT(P).....	401
FLT2UINT(P).....	397
FMOV(P).....	510
FMOVL(P).....	512
FOR.....	547
FPOP(P).....	594
FROM(P).....	1003
FROMD(P).....	1011
F_TRIG(_E).....	1340

G

G.SOCRCVS.....	1094
G(P).SLMPSND.....	1120
G(P).SLMPSNDC.....	1129
G(P).SOCRDATA.....	1110
G(P)_SLMPSND.....	1120
G(P)_SLMPSNDC.....	1129
G(P)_SOCRDATA.....	1110
GBIN(P)(_U).....	439
GE(_E).....	1302
GET_BIT_OF_INT(_E).....	1246
GET_BOOL_ADDR.....	1252
GET_INT_ADDR.....	1252
GET_WORD_ADDR.....	1252

GOEND.....	533
GP.ECPRTCL.....	1112
GP.FTPCGET.....	1144
GP.FTPCPUT.....	1138
GP.SOCCINF.....	1101
GP.SOCCLOSE.....	1088
GP.SOCCSET.....	1104
GP.SOCOPEN.....	1084
GP.SOCRCV.....	1090
GP.SOCRMODE.....	1106
GP.SOCSND.....	1097
GP_ECPRTCL.....	1112
GP_FTPCGET.....	1144
GP_FTPCPUT.....	1138
GP_SOCCINF.....	1101
GP_SOCCLOSE.....	1088
GP_SOCCSET.....	1104
GP_SOCOPEN.....	1084
GP_SOCRCV.....	1090
GP_SOCRMODE.....	1106
GP_SOCSND.....	1097
GRY(P)(_U).....	435
G_SOCRCVS.....	1094
GT(_E).....	1302

H

HABIN(P).....	454

I

IMASK.....	541
INC(P)(_U).....	316
INSERT(_E).....	1315
INSTANCE_ID.....	1425
INSTR(P).....	801
INT2ASC(P).....	766
INT2DBL(P).....	844
INT2DINT(P).....	413
INT2FLT(P).....	836
INT2UDINT(P).....	415
INT2UINT(P).....	411
INT_TO_BCD(_E).....	1179
INT_TO_BITARR(_E).....	1243
INT_TO_BOOL(_E).....	1174
INT_TO_DINT(_E).....	1178
INT_TO_DWORD(_E).....	1176
INT_TO_LREAL(_E).....	1182
INT_TO_REAL(_E).....	1181
INT_TO_STRING(_E).....	1184
INT_TO_TIME(_E).....	1183
INT_TO_WORD(_E).....	1175
INV.....	209
IRET.....	545

J

J(P).SLMPSND.....	1120
J(P)_SLMPSND.....	1120
JMP.....	530

L

LD.....	200
LD<(_U).....	252
LD<=(_U).....	252

LD<>(_U)	252
LD=(_U)	252
LD>(_U)	252
LD>=(_U)	252
LD\$<	716
LD\$<=	716
LD\$<>	716
LD\$=	716
LD\$>	716
LD\$>=	716
LDD<(_U)	254
LDD<=(_U)	254
LDD<>(_U)	254
LDD=(_U)	254
LDD>(_U)	254
LDD>=(_U)	254
LDD_EQ(_U)	254
LDD_GE(_U)	254
LDD_GT(_U)	254
LDD_LE(_U)	254
LDD_LT(_U)	254
LDD_NE(_U)	254
LDDT<	979
LDDT<=	979
LDDT<>	979
LDDT=	979
LDDT>	979
LDDT>=	979
LDDT_EQ	979
LDDT_GE	979
LDDT_GT	979
LDDT_LE	979
LDDT_LT	979
LDDT_NE	979
LDE<	807
LDE<=	807
LDE<>	807
LDE=	807
LDE>	807
LDE>=	807
LEDE<	809
LEDE<=	809
LEDE<>	809
LEDE=	809
LEDE>	809
LEDE>=	809
LEDE_EQ	809
LEDE_GE	809
LEDE_GT	809
LEDE_LE	809
LEDE_LT	809
LEDE_NE	809
LDE_EQ	807
LDE_GE	807
LDE_GT	807
LDE_LE	807
LDE_LT	807
LDE_NE	807
LD_EQ(_U)	252
LDF	202
LDFI	205
LD_GE(_U)	252
LD_GT(_U)	252
LDI	200
LD_LE(_U)	252
LD_LT(_U)	252

LD_NE(_U)	252
LDP	202
LDPI	205
LDSTRING_EQ	716
LDSTRING_GE	716
LDSTRING_GT	716
LDSTRING_LE	716
LDSTRING_LT	716
LDSTRING_NE	716
LDTM<	983
LDTM<=	983
LDTM<>	983
LDTM=	983
LDTM>	983
LDTM>=	983
LDTM_EQ	983
LDTM_GE	983
LDTM_GT	983
LDTM_LE	983
LDTM_LT	983
LDTM_NE	983
LE(_E)	1302
LEDR	714
LEFT(_E)	1308
LEFT(P)	794
LEN(_E)	1306
LEN(P)	790
LIMIT(_E)	1297
LIMIT(P)(_U)	658
LN(_E)	1256
LOG(_E)	1257
LOG(P)	911
LOG10(P)	923
LOG10D(P)	925
LOGD(P)	913
LREAL_TO_DINT(_E)	1216
LREAL_TO_INT(_E)	1214
LREAL_TO_REAL(_E)	1218
LT(_E)	1302

M

M+MX-SQ_ConnectionClose	1879
M+MX-SQ_ConnectionOpen	1876
M+MX-SQ_Recv_Socket	1881
M+MX-SQ_Send_Socket	1884
M+MX-SQ_SLMP_DeviceRead_IP	1887
M+MX-SQ_SLMP_DeviceWrite_IP	1894
M+MX-TSN_RemoteReset2_IP	1915
M+MX-TSN_SLMP_DeviceRead_IP	1901
M+MX-TSN_SLMP_DeviceWrite_IP	1908
MAX(_E)	1295
MAX(P)(_U)	690
MC	243
MC_AbortTrigger	1522
MC_ABS_SYSTEM	1451
MC_ACC_ZERO_MODE	1452
MC_AXES_GROUP_STATUS	1449
MC_AXIS_STATUS	1449
MC_AXIS_TYPE	1451
MC_BUFFER_MODE	1448
MC_CAM_CURVE_TYPE	1450
MC_CAM_CYCLE_METHOD	1455
MC_CAM_ID	1419
MC_CamIn	1705
MC_CAM_REF	1419

MC_CAM_REFERENCE_METHOD	1455
MC_CAM_RESTORE_METHOD.....	1455
MC_CAMSWITCH_DATA_REF	1420
MC_CAMSWITCH_REF	1420
MC_CAMSWITCH_SOURCE_REF	1420
MC_CamTableSelect	1527
MC_CIRC_MODE	1449
MC_CIRC_PATHCHOICE	1449
MC_CLUTCH_METHOD.....	1454
MC_CLUTCH_REFERENCE.....	1454
MC_CLUTCH_SMOOTHING_METHOD	1454
MC_CombineAxes	1742
MC_COMBINE_MODE.....	1448
MC_CONTROL_METHOD	1453
MC_DECEL_STOP_MODE.....	1451
MC_DigitalCamSwitch	1562
MC_DIRECTION	1448
MC_DRIVE_MODE	1450
MC_DRIVE_STATE	1451
MC_ENCODER_AXIS_TYPE.....	1451
MC_ERROR_CLASS	1452
MC_EXECUTION_MODE	1448
MC_EXECUTION_STATE	1452
MC_GearIn	1732
MC_GEAR_RESTORE_METHOD	1454
MC_GroupDisable	1460
MC_GroupEnable.....	1456
MC_GroupReset.....	1503
MC_GroupSetOverride	1545
MC_GroupStop	1584
MC_Home.....	1569
MC_INPUT_DIRECTION.....	1454
MC_INPUT_REF	1420
MC_INTERPOLATE_SPEED_MODE	1450
MC_LENGTH_PER_CYCLE_CHANGE.....	1454
MC_MCODE_OUTPUT_OVERRIDE.....	1453
MC_MoveAbsolute	1590
MC_MoveRelative	1602
MC_MoveVelocity	1618
MC_OPERATION_PATTERN	1453
MC_OUTPUT_REF	1420
MC_POSITIONING_DATA_ID	1419
MC_POSITIONING_DATA_REF.....	1419
MC_POS_RESTORE_STATUS.....	1452
MC_POS_SOURCE	1452
MC_Power.....	1463
MCR	243
MC_ReadParameter.....	1487
MC_RECORD_MODE.....	1450
MC_Reset	1500
MC_SetOverride.....	1483
MC_SetPosition	1467
MC_SIGNAL_LOGIC.....	1452
MC_SOURCE	1448
MC_START_MODE.....	1449
MC_STEP_MODE	1453
MC_Stop	1579
MC_STOP_CURVE_MODE.....	1451
MC_SYNC_CHANGE_COMMAND	1454
MC_TorqueControl	1625
MC_TouchProbe	1507
MC_TRACK_REF	1420
MC_TRIGGER_REF	1420
MCv_AdvancedSync	1804
MCv_AdvCamSetPositionCalc.....	1559
MCv_AdvPositionPerCycleCalc	1556

MCv_AllPower	1542
MCv_BacklashCompensationFilter.....	1752
MCv_ChangeCycle	1534
MCv_CyclicPosition.....	1830
MCv_CyclicTorque	1842
MCv_CyclicVelocity.....	1838
MCv_DirectionFilter	1774
MC_VELOCITY_LIMIT_MODE.....	1450
MCv_GroupMoveWait	1797
MCv_Jog,MCv	1610
MCv_MotionErrorReset	1549
MCv_MoveCircularInterpolateAbsolute	1671
MCv_MoveCircularInterpolateRelative	1688
MCv_MoveLinearInterpolateAbsolute.....	1643
MCv_MoveLinearInterpolateRelative.....	1658
MCv_MovePositioningData.....	1810
MCv_MoveWait	1790
MCv_PositionControl	1819
MCv_ReadProfileData	1848
MCv_SetTorqueLimit	1475
MCv_SmoothingFilter.....	1765
MCv_SpeedControl.....	1635
MCv_SpeedLimitFilter	1782
MCv_SyncOperationCycles	1553
MCv_WriteProfileData	1862
MC_WriteParameter	1494
MEAN(P)(_U).....	706
MEF	210
MEP	210
MID(_E)	1311
MIDR(P)	796
MIDW(P)	798
MIN(_E)	1295
MIN(P)(_U)	694
MINUS(P)(_U)	268
MOD(_E)	1276
MOV(P)	496
MOVB(P)	526
MOVE(_E)	1279
MUL(_E).....	1269
MULTI(P)(_U)	278
MUL_TIME(_E)	1328
MUX(_E)	1300

N

NDIS(P)	488
NE(_E)	1304
NEG(P)	475
NEXT	547
NOPLF	251
NOT(_E)	1292
NUNI(P)	490

O

OR.....	200
OR(_E)	1289
OR<(_U)	252
OR<=(_U)	252
OR<>(_U)	252
OR=(_U)	252
OR>(_U)	252
OR>=(_U)	252
OR\$<	716
OR\$<=.....	716

OR\$<>	716
OR\$=	716
OR\$>	716
OR\$>=	716
ORB	208
ORD<(_U)	254
ORD<=(_U)	254
ORD<>(_U)	254
ORD=(_U)	254
ORD>(_U)	254
ORD>=(_U)	254
ORD_EQ(_U)	254
ORD_GE(_U)	254
ORD_GT(_U)	254
ORD_LE(_U)	254
ORD_LT(_U)	254
ORD_NE(_U)	254
ORDT<	979
ORDT<=	979
ORDT<>	979
ORDT=	979
ORDT>	979
ORDT>=	979
ORDT_EQ	979
ORDT_GE	979
ORDT_GT	979
ORDT_LE	979
ORDT_LT	979
ORDT_NE	979
ORE<	807
ORE<=	807
ORE<>	807
ORE=	807
ORE>	807
ORE>=	807
ORED<	809
ORED<=	809
ORED<>	809
ORED=	809
ORED>	809
ORED>=	809
ORED_EQ	809
ORED_GE	809
ORED_GT	809
ORED_LE	809
ORED_LT	809
ORED_NE	809
ORE_EQ	807
ORE_GE	807
ORE_GT	807
ORE_LE	807
ORE_LT	807
ORE_NE	807
OR_EQ(_U)	252
ORF	202
ORFI	205
OR_GE(_U)	252
OR_GT(_U)	252
ORI	200
OR_LE(_U)	252
OR_LT(_U)	252
OR_NE(_U)	252
ORP	202
ORPI	205
ORSTRING_EQ	716
ORSTRING_GE	716

ORSTRING_GT	716
ORSTRING_LE	716
ORSTRING_LT	716
ORSTRING_NE	716
ORTM<	983
ORTM<=	983
ORTM<>	983
ORTM=	983
ORTM>	983
ORTM>=	983
ORTM_EQ	983
ORTM_GE	983
ORTM_GT	983
ORTM_LE	983
ORTM_LT	983
ORTM_NE	983
OUT	213
OUT C	221
OUT F	225
OUT LC	223
OUT LST	218
OUT LT	218
OUT ST	215
OUT T	215
OUT_C	221
OUTH	215
OUTH ST	215
OUTH T	215
OUT_T	215

P

PID	1043
PIDCONT(P)	1072
PIDINIT(P)	1070
PIDPRMW(P)	1077
PIDRUN(P)	1076
PIDSTOP(P)	1075
PLF	236
PLS	234
PLUS(P)(_U)	264
POW(P)	919
POWD(P)	921
PROFILE_CAM_DATA	1426
PROFILE_CAM_ELEMENT	1427
PROFILE_CAMSWITCH_DATA	1429
PROFILE_CAMSWITCH_ELEMENT	1430
PROFILE_DATA	1424
PROFILE_ID	1424
PROFILE_POSITIONING_DATA	1430
PROFILE_POSITIONING_DATA_ELEMENT	1430
PROFILE_ROTARY_CUTTER	1428
PROFILE_SIGNAL_SELECT_ELEMENT	1434

R

RAD(P)	895
RADD(P)	899
RCL(P)	585
RCR(P)	582
REAL_TO_DINT(_E)	1207
REAL_TO_INT(_E)	1205
REAL_TO_LREAL(_E)	1209
REAL_TO_STRING(_E)	1211
REPLACE(_E)	1319
RET	558

RFS(P).....	997
RIGHT(_E).....	1308
RIGHT(P).....	792
RND(P).....	935
ROL(_E).....	1285
ROL(P).....	585
ROR(_E).....	1287
ROR(P).....	582
RS(_E).....	1336
RSET(P).....	947
RST.....	228
RST F.....	232
R_TRIG(_E).....	1338

S

S.SOCRCVS.....	1094
S(P).DATE-.....	992
S(P).DATE+.....	990
S(P).DATERD.....	988
S(P).DEVLD.....	603
S(P).PIDCONT.....	1061
S(P).PIDINIT.....	1058
S(P).PIDPRMW.....	1066
S(P).PIDRUN.....	1065
S(P).PIDSTOP.....	1064
S(P).SOCRDATA.....	1110
S(P).ZCOM.....	1001
S(P)_DATEMINUS.....	992
S(P)_DATEPLUS.....	990
S(P)_DATERD.....	988
S(P)_DEVLD.....	603
S(P)_PIDCONT.....	1061
S(P)_PIDINIT.....	1058
S(P)_PIDPRMW.....	1066
S(P)_PIDRUN.....	1065
S(P)_PIDSTOP.....	1064
S(P)_SOCRDATA.....	1110
S(P)_ZCOM.....	1001
SCJ.....	530
SCL(P)(_U).....	670
SCL2(P)(_U).....	676
SEC2DATE(P)(_U).....	977
SEC2TIME(P).....	973
SEG(P).....	481
SEL(_E).....	1293
SERDATA(P).....	680
SERMM(P).....	684
SET.....	226
SET F.....	230
SET_BIT_OF_INT(_E).....	1248
SFL(P).....	369
SFR(P).....	367
SFT(P).....	241
SFTBL(P).....	381
SFTBR(P).....	379
SFTWL(P).....	385
SFTWR(P).....	383
SHL(_E).....	1281
SHR(_E).....	1283
SIGNAL_SELECT.....	1424
SIMASK.....	543
SIN(_E).....	1260
SIN(P).....	859
SIND(P).....	871
SJIS2WS(P).....	786

SJIS2WSB(P).....	788
SLAVE_OBJECT_REAL.....	1434
SLAVE_OBJECT_VIRTUAL_ENCODER.....	1436
SORTD(_U).....	698
SP.DEVST.....	605
SP.ECPRTCL.....	1112
SP.FCOPY.....	638
SP.FDELETE.....	635
SP.FMOVE.....	643
SP.FREAD.....	607
SP.FRENAME.....	648
SP.FSTATUS.....	652
SP.FTPGET.....	1144
SP.FTPPUT.....	1138
SP.FWRITE.....	621
SP.SLMPSND.....	1120
SP.SOCCINF.....	1101
SP.SOCCLOSE.....	1088
SP.SOCCSET.....	1104
SP.SOCOPEN.....	1084
SP.SOCRCV.....	1090
SP.SOCRMODE.....	1106
SP.SOCSND.....	1097
SP_DEVST.....	605
SP_ECPRTCL.....	1112
SPF(P).....	768
SP_FCOPY.....	638
SP_FDELETE.....	635
SP_FMOVE.....	643
SP_FREAD.....	607
SP_FRENAME.....	648
SP_FSTATUS.....	652
SP_FTPGET.....	1144
SP_FTPPUT.....	1138
SP_FWRITE.....	621
SP_SLMPSND.....	1120
SP_SOCCINF.....	1101
SP_SOCCLOSE.....	1088
SP_SOCCSET.....	1104
SP_SOCOPEN.....	1084
SP_SOCRCV.....	1090
SP_SOCRMODE.....	1106
SP_SOCSND.....	1097
SQRT(_E).....	1255
SQRT(P).....	710
SR(_E).....	1334
SRND(P).....	936
S_SOCRCVS.....	1094
STMR.....	961
STOP.....	250
STR(P)(_U).....	745
STRDEL(P).....	805
STRINGMOV(P).....	723
STRINGMOV(P)_WS.....	725
STRINGPLUS(P).....	721
STRING_TO_BCD(_E).....	1234
STRING_TO_BOOL(_E).....	1227
STRING_TO_DINT(_E).....	1232
STRING_TO_DWORD(_E).....	1229
STRING_TO_INT(_E).....	1230
STRING_TO_REAL(_E).....	1236
STRING_TO_TIME(_E).....	1239
STRING_TO_WORD(_E).....	1228
STRINS(P).....	803
SUB(_E).....	1271
SUB_TIME(_E).....	1326

SUM(P).....	688
SWAP(P).....	524
SYS_CMD.....	1411
SYS_CONST.....	1408
SYS_MONI.....	1410,1419
SYS_PRM.....	1409

T

TAN(_E).....	1262
TAN(P).....	863
TAND(P).....	875
TARGET_REF.....	1421
TCMP(P).....	986
TEST(P).....	361
TIMCHK.....	996
TIME2SEC(P).....	971
TIMER_100_FB_M.....	1358
TIMER_10_FB_M.....	1358
TIMER_CONT_FB_M.....	1358
TIMER_CONTHFB_M.....	1358
TIMER_HIGH_FB_M.....	1358
TIMER_LOW_FB_M.....	1358
TIME_TO_BOOL(_E).....	1220
TIME_TO_DINT(_E).....	1224
TIME_TO_DWORD(_E).....	1222
TIME_TO_INT(_E).....	1223
TIME_TO_STRING(_E).....	1225
TIME_TO_WORD(_E).....	1221
TO(P).....	1006
TOD(P).....	1015
TOF(_E).....	1356
TON(_E).....	1354
TP(_E).....	1351
TTMR.....	959
TYPERD(P).....	1020

U

UDCNT1.....	955
UDCNT2.....	957
UDINT2DBL(P).....	847
UDINT2DINT(P).....	433
UDINT2FLT(P).....	840
UDINT2INT(P).....	429
UDINT2UINT(P).....	431
UINT2DBL(P).....	845
UINT2DINT(P).....	419
UINT2FLT(P).....	837
UINT2INT(P).....	417
UINT2UDINT(P).....	421
UNI(P).....	486
UNIINFRD(P).....	1024

V

VAL(P)(_U).....	465

W

WAND(P).....	321,322
WDT(P).....	546
WOR(P).....	330,331
WORD_TO_BOOL(_E).....	1159
WORD_TO_DINT(_E).....	1162
WORD_TO_DWORD(_E).....	1160

WORD_TO_INT(_E).....	1161
WORD_TO_STRING(_E).....	1165
WORD_TO_TIME(_E).....	1164
WS2SJIS(P).....	784
WSUM(P)(_U).....	702
WTOB(P).....	492
WXNR(P).....	348,349
WXOR(P).....	339,340

X

XCALL.....	577
XCH(P).....	518
XOR(_E).....	1289

Z

ZONE(P)(_U).....	666
ZPOP(P).....	940,945
ZPOP(P)_2.....	945
ZPUSH(P).....	937,941
ZPUSH(P)_2.....	941
ZRRDB(P).....	949
ZRWRB(P).....	951

改訂履歴

*取扱説明書番号は、本説明書の裏表紙の左下に記載してあります。

改訂年月	*取扱説明書番号	改訂内容
2025年5月	SH(名)-082643-A	初版
2025年5月	SH(名)-082643-B	誤記修正

本書によって、工業所有権その他の権利の実施に対する保証、または実施権を許諾するものではありません。また本書の掲載内容の使用により起因する工業所有権上の諸問題については、当社は一切その責任を負うことができません。

© 2025 MITSUBISHI ELECTRIC CORPORATION

製品の適用について

- (1) 当社シーケンサをご使用いただくにあたりましては、万一シーケンサに故障・不具合などが発生した場合でも重大な事故にいたらない用途であること、および故障・不具合発生時にはバックアップやフェールセーフ機能が機器外部でシステム的に実施されていることをご使用の条件とさせていただきます。
- (2) 当社シーケンサは、一般工業などへの用途を対象とした汎用品として設計・製作されています。
したがって、以下のような機器・システムなどの特殊用途へのご使用については、当社シーケンサの適用を除外させていただきます。万一使用された場合は当社として当社シーケンサの品質、性能、安全に関する一切の責任（債務不履行責任、瑕疵担保責任、品質保証責任、不法行為責任、製造物責任を含むがそれらに限定されない）を負わないものとさせていただきます。
 - ・各電力会社殿の原子力発電所およびその他発電所向けなどの公共への影響が大きい用途
 - ・鉄道各社殿および官公庁殿など、特別な品質保証体制の構築を当社にご要求になる用途
 - ・航空宇宙、医療、鉄道、燃焼・燃料装置、乗用移動体、有人搬送装置、娯楽機械、安全機械など
生命、身体、財産に大きな影響が予測される用途ただし、上記の用途であっても、具体的に用途を限定すること、特別な品質（一般仕様を超えた品質等）をご要求されないこと等を条件に、当社の判断にて当社シーケンサの適用可とする場合もございますので、詳細につきましては当社窓口へご相談ください。
- (3) DoS攻撃、不正アクセス、コンピュータウイルスその他のサイバー攻撃により発生するシーケンサ、およびシステムトラブル上の諸問題に対して、当社はその責任を負わないものとさせていただきます。

保証について

ご使用に際しましては、以下の製品保証内容をご確認いただきますよう、よろしくお願いいたします。

1. 無償保証期間と無償保証範囲

無償保証期間中に、製品に当社側の責任による故障や瑕疵（以下併せて「故障」と呼びます）が発生した場合、当社はお買い上げいただきました販売店または当社サービス会社を通じて、無償で製品を修理させていただきます。

ただし、国内および海外における出張修理が必要な場合は、技術者派遣に要する実費を申し受けます。

また、故障ユニットの取替えに伴う現地再調整・試運転は当社責務外とさせていただきます。

【無償保証期間】

製品の無償保証期間は、お客様にてご購入後またはご指定場所に納入後 36 ヶ月とさせていただきます。

ただし、当社製品出荷後の流通期間を最長 6 ヶ月として、製造から 42 ヶ月を無償保証期間の上限とさせていただきます。

また、修理品の無償保証期間は、修理前の無償保証期間を超えて長くなることはありません。

【無償保証範囲】

(1) 一次故障診断は、原則として貴社にて実施をお願い致します。

ただし、貴社要請により当社、または当社サービス網がこの業務を有償にて代行することができます。この場合、故障原因が当社側にある場合は無償と致します。

(2) 使用状態・使用方法、および使用環境などが、取扱説明書、ユーザーズマニュアル、製品本体注意ラベルなどに記載された条件・注意事項などに従った正常な状態で使用されている場合に限定させていただきます。

(3) 無償保証期間内であっても、以下の場合には有償修理とさせていただきます。

① お客様における不適切な保管や取扱い、不注意、過失などにより生じた故障およびお客様のハードウェアまたはソフトウェア設計内容に起因した故障。

② お客様にて当社の了解なく製品に改造などの手を加えたことに起因する故障。

③ 当社製品がお客様の機器に組み込まれて使用された場合、お客様の機器が受けている法的規制による安全装置または業界の通念上備えられているべきと判断される機能・構造などを備えていれば回避できたと認められる故障。

④ 取扱説明書などに指定された消耗部品が正常に保守・交換されていれば防げたと認められる故障。

⑤ 消耗部品（バッテリー、リレー、ヒューズなど）の交換。

⑥ 火災、異常電圧などの不可抗力による外部要因および地震、雷、風水害などの天変地異による故障。

⑦ 当社出荷当時の科学技術の水準では予見できなかった事由による故障。

⑧ その他、当社の責任外の場合またはお客様が当社責任外と認めた故障。

2. 生産中止後の有償修理期間

(1) 当社が有償にて製品修理を受け付けることができる期間は、その製品の生産中止後 7 年間です。

生産中止に関しましては、当社テクニカルニュースなどにて報じさせていただきます。

(2) 生産中止後の製品供給（補用品も含む）はできません。

3. 海外でのサービス

海外においては、当社の各地域 FA センターで修理受付をさせていただきます。ただし、各 FA センターでの修理条件などが異なる場合がありますのでご了承ください。

4. 機会損失、二次損失などへの保証責務の除外

無償保証期間の内外を問わず、以下については当社責務外とさせていただきます。

(1) 当社の責に帰することができない事由から生じた障害。

(2) 当社製品の故障に起因するお客様での機会損失、逸失利益。

(3) 当社の予見の有無を問わず特別の事情から生じた損害、二次損害、事故補償、当社製品以外への損傷。

(4) お客様による交換作業、現地機械設備の再調整、立上げ試運転その他の業務に対する補償。

5. 製品仕様の変更

カタログ、マニュアルもしくは技術資料などに記載の仕様は、お断りなしに変更させていただく場合がありますので、あらかじめご承知おきください。

以 上

サービスのお問い合わせ

修理・サービスに関するお問い合わせはこちらにお問い合わせください。

三菱電機システムサービス株式会社

北日本支社	(022) 353-7814	北陸支店	(076) 252-9519
北海道支店	(011) 890-7515	関西支社	(06) 6458-9728
首都圏第2支社	(03) 3454-5521	京滋機器サービスステーション	(075) 874-3614
神奈川機器サービスステーション	(045) 938-5420	姫路機器サービスステーション	(079) 269-8845
関越機器サービスステーション	(048) 708-5910	中四国支社	(082) 285-2111
新潟機器サービスステーション	(025) 241-7261	岡山機器サービスステーション	(086) 242-1900
中部支社	(052) 722-7601	四国支店	(087) 831-3186
静岡機器サービスステーション	(054) 287-8866	九州支社	(092) 483-8208

商標

CiA[®]およびCANopen[®]は、CAN in Automation e.V.のEUにおける登録商標です。
Microsoft, Microsoft Access, Excel, SQL Server, Visual Basic, Visual C++, Visual Studio, Windows, Windows NT, Windows Server, Windows Vista, およびWindows XPは、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。
Unicodeは、Unicode, Inc.の米国およびその他の国における登録商標または商標です。
本文中における会社名、システム名、製品名などは、一般に各社の登録商標または商標です。
本文中で、商標記号(™, ®)は明記していない場合があります。

三菱電機株式会社 〒100-8310 東京都千代田区丸の内2-7-3 (東京ビル)

お問い合わせは下記へどうぞ

本社機器営業部	〒100-8310	東京都千代田区丸の内2-7-3 (東京ビル)	(03) 3218-2606
関越機器営業部	〒330-6034	さいたま市中央区新都心11-2 (明治安田生命さいたま新都心ビル)	(048) 600-5835
新潟支店	〒950-8504	新潟市中央区東大通2-4-10 (日本生命新潟ビル)	(025) 241-7227
神奈川機器営業部	〒220-8118	横浜市西区みなとみらい2-2-1 (横浜ランドマークタワー)	(045) 224-2624
北海道支社	〒060-8693	札幌市中央区大通西3-11 (北洋ビル)	(011) 212-3793
東北支社	〒980-0013	仙台市青葉区花京院1-1-20 (花京院スクエア)	(022) 216-4546
北陸支社	〒920-0031	金沢市広岡3-1-1 (金沢パークビル)	(076) 233-5502
中部支社	〒450-6423	名古屋市中村区名駅3-28-12 (大名古屋ビルヂング)	(052) 565-3314
豊田支店	〒471-0034	豊田市小坂本町1-5-10 (矢作豊田ビル)	(0565) 34-4112
関西支社	〒530-8206	大阪市北区大深町4-20 (グランフロント大阪タワーA)	(06) 6486-4122
中国支社	〒730-8657	広島市中区中町7-32 (ニッセイ広島ビル)	(082) 248-5348
四国支社	〒760-8654	高松市寿町1-1-8 (日本生命高松駅前ビル)	(087) 825-0055
九州支社	〒810-8686	福岡市中央区天神2-12-1 (天神ビル)	(092) 721-2247

三菱電機 FA

検索

www.MitsubishiElectric.co.jp/fa

メンバー
登録無料!

インターネットによる情報サービス「三菱電機FAサイト」

三菱電機FAサイトでは、製品や事例などの技術情報に加え、トレーニングスクール情報や各種お問い合わせ窓口をご提供しています。また、メンバー登録いただくとマニュアルやCADデータ等のダウンロード、eラーニングなどの各種サービスをご利用いただけます。

仕様・機能に関するお問い合わせ

製品ごとにお問い合わせを受け付けております。
三菱電機FAサイト - 仕様・機能に関するお問い合わせ
www.MitsubishiElectric.co.jp/fa/contact-us/spec/



本マニュアル対象機種の電話技術相談窓口

音声ガイダンスに従って「お客様相談内容に関する代理店、商社への提供可否」の回答後、機種選択番号を入力してください。
「お客様相談内容に関する代理店、商社への提供可否」の回答後であれば、ガイダンスの途中でも機種選択番号を入力できます。

対象機種		共通電話番号	機種選択番号	受付時間※1
MELSEC MXコントローラ MX-Rモデル	機能全般 (CC-Link IE TSN機能、 モーション機能を除く) MELSOFT GXシリーズ	052-712-2444	2→2	月曜～木曜 9:00～19:00 金曜～日曜・祝日 9:00～17:00
	CC-Link IE TSN機能		2→3	月曜～木曜 9:00～19:00 金曜～日曜・祝日 9:00～17:00
	モーション機能		1	月曜～木曜 9:00～19:00 金曜～日曜・祝日 9:00～17:00

お問い合わせの際には、今一度電話番号をお確かめの上、お掛け間違いのないようお願いいたします。

※1: 春季・夏季・年末年始の休日を除く

SH(名)-082643-B(2505)MEE

形名: MXR-P-J

2025年5月作成

本マニュアルは、お断りなしに仕様を変更することがありますのでご了承ください。
本マニュアルは、輸出する場合、経済産業省への役務取引許可申請は不要です。