

三菱电机FA整合控制器

MELSEC MX Controller

MELSEC MX控制器MX-F型
编程手册

安全注意事项

(使用之前请务必阅读)

使用三菱电机FA整合控制器之前，应仔细阅读各产品手册及各产品手册中所介绍的关联手册，同时在充分注意安全的前提下正确地操作。

请妥善保管本手册以备需要时阅读，并应将本手册交给最终用户。

前言

在此非常感谢贵方购买了三菱电机FA整合控制器控制器产品。

本手册是用于让用户了解进行编程时必要的指令、函数和FB相关内容的手册。

在使用之前应熟读本手册及关联手册，在充分了解控制器的功能・性能的基础上正确地使用本产品。

此外，将本手册中介绍的程序示例应用于实际系统的情况下，应充分验证对象系统中不存在控制方面的问题。
应将本手册交给最终用户。

目录

安全注意事项	1
前言	2
关联手册	21
术语	22
总称/简称	24

第1部分 编程概要

第1章 指令概要	26
1.1 指令的构成	26
1.2 数据的指定方法	28
位数据	31
16位数据(字数据)	32
32位数据(双字数据)	34
实数数据(浮点数据)	37
字符串数据	40
1.3 执行条件	42
1.4 编程时的注意事项	43
指令通用的错误	43
执行指令时的软元件、标签的范围检查	43
使用长定时器、长累计定时器软元件时的动作	49
将分配到同步站的链接软元件用于指令参数时的注意事项	51
使用同一软元件的OUT指令、SET/RST指令、PLS/PLF指令时的动作	51
使用文件寄存器时的限制事项	57
第2章 程序部件	58
2.1 程序块	59
2.2 函数(FUN)	60
2.3 功能块(FB)	64
2.4 注意事项	75
使用函数的情况	75
使用FB的情况	75
使用MC/MCR指令控制EN的动作	79
第3章 梯形图语言	84
3.1 配置	84
梯形图符号	84
程序执行顺序	85
以梯形图语言使用FB时的注意事项	86
3.2 内嵌ST	87
3.3 声明/注解	89
第4章 ST语言	90
4.1 配置	91
分隔符	92
运算符	92

指令	93
语法	94
常数	103
标签与软元件	104
注释	106

第5章 FBD/LD语言 107

5.1 配置	107
部件	108
常数	115
标签与软元件	115
5.2 内嵌ST	117
5.3 程序执行顺序	118
部件的执行顺序	118

第2部分 指令/FUN/FB一览

第6章 指令一览 122

6.1 顺控程序指令	122
6.2 基本指令	126
6.3 应用指令	144
程序控制	144
数据处理	146
调试、故障诊断	153
字符串处理	153
实数处理	156
随机数	163
软元件操作	164
定时器、计数器	165
时钟	166
模块访问	170
PID运算指令	171
PID控制指令	171
高速I/O控制指令	172
定位功能专用指令	172
高速计数器功能专用指令	173
脉冲系统指令	173

第7章 网络指令一览 174

7.1 以太网用指令	174
7.2 外部设备通信指令	176

第8章 通用FUN/通用FB一览 177

8.1 通用函数	177
8.2 通用FB	188

第9章 运动控制FB一览 190

9.1 管理系统FB	190
9.2 运行系统FB	193

9.3	一般FB	196
第10章 网络FB一览		197
10.1	以太网用FB	197
10.2	CC-Link IE TSN用FB	198
第3部分 顺控程序指令/基本指令		
第11章 顺控程序指令		200
11.1	触点指令	200
	运算开始、串联连接、并联连接	200
	脉冲运算开始、脉冲串联连接、脉冲并联连接	202
	脉冲否定运算开始、脉冲否定串联连接、脉冲否定并联连接	205
11.2	合并指令	208
	梯形图块串联连接、并联连接	208
	运算结果取反	209
	运算结果脉冲化	210
	变址继电器运算结果脉冲化	211
11.3	输出指令	213
	OUT(定时器、计数器、报警器除外)	213
	定时器	215
	长定时器	218
	计数器	221
	长计数器	223
	报警器	225
	软元件的设置(报警器除外)	226
	软元件的复位(报警器除外)	228
	报警器的设置	229
	报警器的复位	231
	上升沿输出	233
	下降沿输出	235
	位软元件输出取反	237
	直接输出的脉冲化	238
11.4	移位指令	240
	位软元件移位	240
11.5	主控制指令	242
	主控制的设置、复位	242
11.6	结束指令	246
	主程序结束	246
	顺控程序结束	247
11.7	停止指令	249
	顺控程序停止	249
11.8	无处理指令	250
	无处理(NOPLF)	250
第12章 基本指令		251
12.1	比较运算指令	251
	BIN16位数据比较	251
	BIN32位数据比较	253
	BIN16位块数据比较	255

	BIN32位块数据比较	258
12.2	算术运算指令	261
	BIN16位加法运算	261
	BIN16位减法运算	265
	BIN32位加法运算	269
	BIN32位减法运算	273
	BIN16位乘法运算	277
	BIN16位除法运算	279
	BIN32位乘法运算	281
	BIN32位除法运算	283
	BCD4位加法运算	285
	BCD4位减法运算	288
	BCD8位加法运算	291
	BCD8位减法运算	294
	BCD4位乘法运算	297
	BCD4位除法运算	299
	BCD8位乘法运算	301
	BCD8位除法运算	303
	BIN16位块数据加法运算	305
	BIN16位块数据减法运算	307
	BIN32位块数据加法运算	309
	BIN32位块数据减法运算	312
	16位BIN数据递增	315
	16位BIN数据递减	316
	32位BIN数据递增	317
	32位BIN数据递减	318
12.3	逻辑运算指令	319
	16位数据逻辑与	319
	32位数据逻辑与	322
	16位块数据逻辑与	325
	16位数据逻辑或	327
	32位数据逻辑或	330
	16位块数据逻辑或	333
	16位数据异或	335
	32位数据异或	338
	16位块数据异或	341
	16位数据异或非	343
	32位数据异或非	345
	16位块数据异或非	348
12.4	位处理指令	350
	字软元件的位设置	350
	字软元件的位复位	352
	16位测试	353
	32位测试	355
	位软元件的批量复位	357
12.5	数据移位指令	358
	BIN16位数据的n位右移	358
	BIN16位数据的n位左移	360
	n位数据的1位右移	362
	n位数据的1位左移	364
	n字数据的1字右移	366

n字数据的1字左移	368
n位数据的n位右移	370
n位数据的n位左移	372
n字数据的n字右移	374
n字数据的n字左移	376
12.6 数据转换指令	378
BIN数据→BCD4位转换	378
BIN数据→BCD8位转换	380
BCD4位→BIN16位数据转换	382
BCD8位→BIN32位数据转换	384
单精度实数→带符号BIN16位数据	386
单精度实数→无符号BIN16位数据	388
单精度实数→带符号BIN32位数据	390
单精度实数→无符号BIN32位数据	392
双精度实数→带符号BIN16位数据	394
双精度实数→无符号BIN16位数据	396
双精度实数→带符号BIN32位数据	398
双精度实数→无符号BIN32位数据	400
带符号BIN16位数据→无符号BIN16位数据转换	402
带符号BIN16位数据→带符号BIN32位数据转换	404
带符号BIN16位数据→无符号BIN32位数据转换	406
无符号BIN16位数据→带符号BIN16位数据转换	408
无符号BIN16位数据→带符号BIN32位数据转换	410
无符号BIN16位数据→无符号BIN32位数据转换	412
带符号BIN32位数据→带符号BIN16位数据转换	414
带符号BIN32位数据→无符号BIN16位数据转换	416
带符号BIN32位数据→无符号BIN32位数据转换	418
无符号BIN32位数据→带符号BIN16位数据转换	420
无符号BIN32位数据→无符号BIN16位数据转换	422
无符号BIN32位数据→带符号BIN32位数据转换	424
BIN16位数据→BIN格雷码数据转换	426
BIN32位数据→格雷码数据转换	428
BIN16位格雷码数据→BIN16位数据转换	430
BIN32位格雷码数据→BIN32位数据转换	432
BIN16位数据块→BCD4位数据转换块	434
块BCD4位数据→块BIN16位转换数据	436
10进制ASCII数据→BIN16位数据转换	438
10进制ASCII数据→BIN32位数据转换	441
16进制ASCII数据→BIN16位数据转换	445
16进制ASCII数据→BIN32位数据转换	447
10进制ASCII数据→BCD4位数据转换	450
10进制ASCII数据→BCD8位数据转换	453
10进制字符串→BIN16位数据转换	456
10进制字符串→BIN32位数据转换	459
16进制ASCII→16进制数据BIN转换	462
单精度实数→BCD分解	464
BIN16位数据2的补数(符号取反)	466
BIN32位数据2的补数(符号取反)	467
8→256位解码	468
256→8位编码	470
7段解码	472

BIN16位数据的4位分离	474
BIN16位数据的4位合并	476
任意数据的位分离	478
任意数据的位合并	480
字节单位数据分离	482
字节单位数据合并	484
12.7 数据传送指令	486
BIN16位数据传送	486
BIN32位数据传送	488
BIN16位数据否定传送	490
BIN32位数据否定传送	491
1位数据取反传送	493
BIN16位数据块传送(16位)	494
BIN16位数据块传送(32位)	496
同一BIN16位数据块传送(16位)	498
同一BIN16位数据块传送(32位)	500
同一BIN32位数据块传送(16位)	502
同一BIN32位数据块传送(32位)	504
BIN16位数据交换	506
BIN32位数据交换	507
BIN16位数据块交换	508
BIN16位数据上下字节交换	510
BIN32位数据上下字节交换	511
1位数据传送	512
n位数据传送	513

第4部分 应用指令

第13章 程序控制	516
13.1 程序分支指令	516
指针分支	516
跳转至END	519
13.2 程序执行控制指令	520
中断禁止、中断允许	520
指定优先级及其以下的中断禁止	523
中断程序屏蔽	528
指定中断指针的禁止/允许	530
从中断程序返回	532
WDT复位	533
13.3 结构化指令	534
FOR~NEXT	534
FOR~NEXT强制结束	537
子程序调用	539
从子程序的返回	545
子程序的输出OFF调用	546
程序文件之间子程序调用	552
程序文件之间子程序输出OFF调用	559
子程序调用	564

第14章 数据处理	569
14.1 旋转指令	569
BIN16位数据的右旋转	569
BIN16位数据的左旋转	572
BIN32位数据的右旋转	575
BIN32位数据的左旋转	577
14.2 数据表操作指令	579
从数据表的先输入数据的读取	579
从数据表的后输入数据的读取	581
至数据表的数据写入	583
数据表的数据插入	585
数据表的数据删除	587
14.3 数据读取/写入指令	589
从数据存储器的数据读取	590
至数据存储器的数据写入	592
14.4 文件操作指令	594
从指定文件的数据读取	594
至指定文件的数据写入	608
指定文件的删除	621
指定文件的复制	624
指定文件的移动	629
指定文件名的更改	634
指定文件状态的获取	637
文件操作指令中发生的错误代码	640
14.5 数据控制指令	641
BIN16位数据上下限位控制	641
BIN32位数据上下限位控制	643
BIN16位数据死区控制	645
BIN32位数据死区控制	647
BIN16位数据区域控制	649
BIN32位数据区域控制	651
BIN16位单位标度(各点坐标数据)	653
BIN32位单位标度(各点坐标数据)	656
BIN16位单位标度(各X/Y坐标数据)	659
BIN32位单位标度(各X/Y坐标数据)	661
14.6 数据处理指令	663
BIN16位数据查找	663
BIN32位数据查找	665
BIN16位数据查找(最小·相同·最大)	667
BIN32位数据查找(最小·同一·最大)	669
BIN16位数据位检查	671
BIN32位数据位检查	672
BIN16位数据最大值搜索	673
BIN32位数据最大值搜索	675
BIN16位数据最小值搜索	677
BIN32位数据最小值搜索	679
BIN16位数据排序	681
BIN32位数据排序	683
BIN16位数据合计值计算	685
BIN32位数据合计值计算	687

BIN16位数据平均值计算	689
BIN32位数据平均值计算	691
BIN16位平方根计算	693
BIN32位平方根计算	694
CRC运算	695

第15章 调试、故障诊断 **697**

15.1 调试、故障诊断指令	697
错误显示或报警器复位	697

第16章 字符串处理 **699**

16.1 字符串处理指令	699
字符串比较	699
字符串的合并	702
字符串传送	706
Unicode对应字符串传送	708
BIN16位数据→10进制ASCII转换	710
BIN32位数据→10进制ASCII转换	714
BIN16位数据→16进制ASCII转换	719
BIN32位数据→16进制ASCII转换	723
BIN16位数据→字符串转换	727
BIN32位数据→字符串转换	730
BCD4位数据→10进制ASCII代码转换	733
BCD8位数据→10进制ASCII代码转换	737
单精度实数→字符串转换	742
16进制BIN数据→16进制ASCII代码转换	747
多个数据的字符串转换合并	749
Unicode字符串→移位JIS字符串转换	765
移位JIS字符串→Unicode字符串转换(无字节顺序标志)	767
移位JIS字符串→Unicode转换(有字节顺序标志)	769
字符串的长度检测	771
从字符串的右侧开始提取	773
从字符串的左侧开始提取	775
字符串中的任意提取	777
字符串中的任意替换	779
字符串查找	782
字符串插入	784
字符串删除	786

第17章 实数处理 **788**

17.1 浮点指令	788
单精度实数比较	788
双精度实数比较	790
单精度实数加法运算	793
单精度实数减法运算	797
双精度实数加法运算	801
双精度实数减法运算	805
单精度实数乘法运算	809
单精度实数除法运算	811
双精度实数乘法运算	813

双精度实数除法运算	815
带符号BIN16位数据→单精度实数转换	817
无符号BIN16位数据→单精度实数转换	818
带符号BIN32位数据→单精度实数转换	819
无符号BIN32位数据→单精度实数转换	821
双精度实数→单精度实数转换	823
带符号BIN16位数据→双精度实数转换	825
无符号BIN16位数据→双精度实数转换	826
带符号BIN32位数据→双精度实数转换	827
无符号BIN32位数据→双精度实数转换	828
单精度实数→双精度实数转换	829
字符串→单精度实数转换	830
BCD格式数据→单精度实数数据转换	834
单精度实数符号取反	836
双精度实数符号取反	837
单精度实数数据传送	838
双精度实数数据传送	839
单精度实数SIN运算	840
单精度实数COS运算	842
单精度实数TAN运算	844
单精度实数 SIN^{-1} 运算	846
单精度实数 COS^{-1} 运算	848
单精度实数 TAN^{-1} 运算	850
双精度实数SIN运算	852
双精度实数COS运算	854
双精度实数TAN运算	856
双精度实数 SIN^{-1} 运算	858
双精度实数 COS^{-1} 运算	860
双精度实数 TAN^{-1} 运算	862
BCD型SIN运算	864
BCD型COS运算	866
BCD型TAN运算	868
BCD型 SIN^{-1} 运算	870
BCD型 COS^{-1} 运算	872
BCD型 TAN^{-1} 运算	874
单精度实数角度→弧度转换	876
单精度实数弧度→角度转换	878
双精度实数角度→弧度转换	880
双精度实数弧度→角度转换	882
单精度实数平方根	884
双精度实数平方根	886
单精度实数指数运算	888
双精度实数指数运算	890
单精度实数自然对数运算	892
双精度实数自然对数运算	894
BCD4位平方根	896
BCD8位平方根	898
单精度实数幂运算	900
双精度实数幂运算	902
单精度实数常用对数运算	904
双精度实数常用对数运算	906

单精度实数最大值搜索	908
双精度实数最大值搜索	910
单精度实数最小值搜索	912
双精度实数最小值搜索	914

第18章 随机数 **916**

18.1 随机数指令	916
随机数发生	916
系列更改	917

第19章 软元件操作 **918**

19.1 变址寄存器指令	918
变址寄存器的批量保存	918
变址寄存器的批量恢复	921
变址寄存器/长变址寄存器选择保存	922
变址寄存器/长变址寄存器选择恢复	926
19.2 文件寄存器操作指令	928
文件寄存器的块No. 切换	928
19.3 文件寄存器的1字节单位的读取/写入指令	930
文件寄存器的1字节数据读取	930
文件寄存器的1字节数据写入	932
19.4 间接地址读取指令	934
间接地址读取	934

第20章 定时器、计数器 **936**

20.1 特殊计数器指令	936
单相输入升值/降值计数器	936
2相输入升值/降值计数器	938
20.2 特殊定时器指令	940
示教定时器	940
特殊功能定时器	942

第21章 时钟 **944**

21.1 时钟用指令	944
时钟数据的读取	944
时钟数据的写入	946
时钟数据的加法运算	948
时钟数据的减法运算	950
时间数据的转换(时分秒→秒)	952
时间数据的转换(秒→时分秒)	954
日期时间数据的转换(日期时间→秒)	956
日期时间数据的转换(秒→日期时间)	958
日期比较	960
时间比较	964
时间比较输出	967
扩展时钟数据的读取	969
扩展时钟数据的加法运算	971
扩展时钟数据的减法运算	973
21.2 时机计测指令	975
时机脉冲发生	975

指定数据的时间计测	977
第22章 模块访问	978
22.1 模块访问指令	978
I/O刷新	978
从模块中的1字/2字数据读取(16位指定)	980
至模块的1字/2字数据写入(16位指定)	983
从模块中的1字/2字数据读取(32位指定)	988
至模块的1字/2字数据写入(32位指定)	992
模块固有信息读取	996
第23章 PID运算指令	1000
23.1 概要	1000
控制用数据	1002
自动调谐	1010
23.2 PID运算指令	1012
第24章 PID控制指令	1016
24.1 概要	1016
运算方式	1016
PID控制步骤	1017
其它功能	1023
24.2 PID控制指令(不完全微分)	1025
PID控制用数据的设置	1027
PID运算	1030
指定环路No. 的运算停止	1033
指定环路No. 的运算开始	1034
指定环路No. 的参数更改	1035
24.3 PID控制指令(完全微分)	1037
PID控制用数据的设置	1039
PID运算	1041
指定环路No. 的运算停止	1044
指定环路No. 的运算开始	1045
指定环路No. 的参数更改	1046
第25章 高速I/O控制指令	1048
25.1 高速输入输出的开始/停止	1048
16位数据高速输入输出功能的开始/停止	1048
32位数据高速输入输出功能的开始/停止	1051
第26章 定位功能专用指令	1054
26.1 机械原点复位	1054
26.2 中断1速定位	1061
26.3 可变速度运行	1067
26.4 1速定位(相对)	1072
26.5 1速定位(绝对)	1077
26.6 表格运行(多个表格)	1082
26.7 中断输入1触发	1084
26.8 绝对位置恢复	1087

第27章 高速计数器功能专用指令 1090

27.1 高速IO用SM/SD的读取/写入(更新)	1090
16位数据读取/写入(更新)	1090
32位数据读取/写入(更新)	1093

第28章 脉冲系统指令 1095

28.1 脉冲宽度调制	1095
BIN16位脉冲宽度调制	1095
BIN32位脉冲宽度调制	1100
28.2 脉冲密度的测定	1105
16位数据脉冲密度的测定	1105
32位数据脉冲密度的测定	1107

第5部分 网络指令

第29章 以太网用指令 1110

29.1 打开/关闭处理指令	1110
连接的建立	1110
连接的断开	1114
29.2 套接字通信指令	1116
接收数据的END处理时读取	1116
执行接收数据的指令时读取	1119
数据发送	1122
连接信息的读取	1126
连接的通信目标更改(UDP/IP)	1129
连接的接收模式更改	1131
套接字通信接收数据读取	1136
29.3 通信协议支持功能指令	1138
登录协议执行	1138
29.4 SLMP帧发送指令	1146
SLMP帧发送	1146
加密的SLMP帧发送	1155
29.5 文件传送功能用指令	1163
FTP客户端文件发送	1163
FTP客户端文件获取	1168

第30章 外部设备通信指令 1173

30.1 特殊适配器用指令	1173
30.2 串行数据传送2	1176

第6部分 通用FUN

第31章 类型转换函数 1180

31.1 BOOL型→WORD型转换	1180
31.2 BOOL型→DWORD型转换	1181
31.3 BOOL型→INT型转换	1182
31.4 BOOL型→DINT型转换	1183
31.5 BOOL型→TIME型转换	1184

31.6	BOOL型→STRING型转换	1185
31.7	WORD型→BOOL型转换	1186
31.8	WORD型→DWORD型转换	1187
31.9	WORD型→INT型转换	1188
31.10	WORD型→DINT型转换	1189
31.11	WORD型→TIME型转换	1190
31.12	WORD型→STRING型转换	1191
31.13	DWORD型→BOOL型转换	1192
31.14	DWORD型→WORD型转换	1193
31.15	DWORD型→INT型转换	1194
31.16	DWORD型→DINT型转换	1195
31.17	DWORD型→TIME型转换	1196
31.18	DWORD型→STRING型转换	1197
31.19	INT型→BOOL型转换	1198
31.20	INT型→WORD型转换	1199
31.21	INT型→DWORD型转换	1200
31.22	INT型→DINT型转换	1201
31.23	INT型→BCD型转换	1202
31.24	INT型→REAL型转换	1203
31.25	INT型→LREAL型转换	1204
31.26	INT型→TIME型转换	1205
31.27	INT型→STRING型转换	1206
31.28	DINT型→BOOL型转换	1208
31.29	DINT型→WORD型转换	1209
31.30	DINT型→DWORD型转换	1210
31.31	DINT型→INT型转换	1211
31.32	DINT型→BCD型转换	1212
31.33	DINT型→REAL型转换	1214
31.34	DINT型→LREAL型转换	1215
31.35	DINT型→TIME型转换	1216
31.36	DINT型→STRING型转换	1217
31.37	BCD型→INT型转换	1219
31.38	BCD型→DINT型转换	1221
31.39	BCD型→STRING型转换	1223
31.40	REAL型→INT型转换	1225
31.41	REAL型→DINT型转换	1226
31.42	REAL型→LREAL型转换	1227
31.43	REAL型→STRING型转换	1228
31.44	LREAL型→INT型转换	1231
31.45	LREAL型→DINT型转换	1232
31.46	LREAL型→REAL型转换	1233
31.47	TIME型→BOOL型转换	1234
31.48	TIME型→WORD型转换	1235
31.49	TIME型→DWORD型转换	1236
31.50	TIME型→INT型转换	1237
31.51	TIME型→DINT型转换	1238
31.52	TIME型→STRING型转换	1239
31.53	STRING型→BOOL型转换	1240
31.54	STRING型→WORD型转换	1241
31.55	STRING型→DWORD型转换	1242
31.56	STRING型→INT型转换	1243

31.57	STRING型→DINT型转换	1244
31.58	STRING型→BCD型转换	1246
31.59	STRING型→REAL型转换	1248
31.60	STRING型→TIME型转换	1251
31.61	位数组→INT型转换	1252
31.62	位数组→DINT型转换	1253
31.63	INT型→位数组转换	1254
31.64	DINT型→位数组转换	1255
31.65	位数组的复制	1256
31.66	字标签的指定位读取	1257
31.67	字标签的指定位写入	1259
31.68	字标签的指定位复制	1261
31.69	起始数据的获取	1263
第32章 单数值变量函数		1264
32.1	绝对值	1264
32.2	平方根	1266
32.3	自然对数运算	1267
32.4	常用对数运算	1268
32.5	指数运算	1270
32.6	SIN运算	1271
32.7	COS运算	1272
32.8	TAN运算	1273
32.9	SIN^{-1} 运算	1274
32.10	COS^{-1} 运算	1275
32.11	TAN^{-1} 运算	1276
第33章 算术运算函数		1277
33.1	加法运算	1277
33.2	乘法运算	1280
33.3	减法运算	1282
33.4	除法运算	1285
33.5	余数	1287
33.6	幂	1289
33.7	代入	1290
第34章 位移函数		1292
34.1	n位左移	1292
34.2	n位右移	1294
34.3	n位左旋	1296
34.4	n位右旋	1298
第35章 位型布尔函数		1300
35.1	逻辑与、逻辑或、异或	1300
35.2	逻辑否	1303
第36章 选择函数		1304
36.1	选择值	1304
36.2	最大值、最小值选择	1306
36.3	上下限位控制	1308

36.4	多路复用器	1311
第37章 比较函数		1313
37.1	比较	1313
37.2	比较	1315
第38章 字符串函数		1317
38.1	字符串的长度检测	1317
38.2	从字符串的左侧、右侧提取	1319
38.3	字符串的提取	1322
38.4	字符串的合并	1324
38.5	字符串的插入	1326
38.6	字符串的删除	1328
38.7	字符串的替换	1330
38.8	字符串的搜索	1333
第39章 时间数据类型函数		1335
39.1	加法运算	1335
39.2	减法运算	1337
39.3	乘法运算	1339
39.4	除法运算	1341
第7部分 通用FB		
第40章 双稳态FB		1344
40.1	双稳态FB(设置优先)	1344
40.2	双稳态FB(复位优先)	1346
第41章 边缘检测FB		1348
41.1	上升沿边缘检测	1348
41.2	下降沿边缘检测	1350
第42章 计数器/定时器FB		1352
42.1	升值计数器	1352
42.2	降值计数器	1354
42.3	升值降值计数器	1356
42.4	计数器FB	1359
42.5	脉冲定时器	1361
42.6	ON延迟定时器	1363
42.7	OFF延迟定时器	1365
42.8	定时器FB	1367
第8部分 运动控制FB		
第43章 运动控制FB的概要		1372
43.1	运动控制FB	1372
	运动控制FB的使用方法	1372
	各系统状态的运动控制FB的动作	1372
	运动控制FB的类型	1373

运动控制的类型	1379
错误处理	1379
控制中使用的单位	1379
运动控制FB的输入输出变量	1380
运动控制FB的注意事项	1382
43.2 运动控制FB的配置	1383
梯形图	1383
FBD/LD(功能块图表/梯形图)	1384
ST(结构化文本)	1384
第44章 运动控制FB变量一览	1385
44.1 变量一览	1385
轴变量	1387
轴组变量	1408
系统变量	1414
输入轴变量	1418
输出轴变量	1420
系统变量(高级同步控制)	1425
其他结构体的变量	1426
其它结构体的变量(高级同步控制)	1445
44.2 列举符一览	1455
ENUM列举符	1455
第45章 管理系统的FB	1463
45.1 轴组有效	1463
45.2 轴组无效	1467
45.3 允许运行	1471
45.4 当前位置更改	1475
45.5 转矩限制值	1482
45.6 倍率修调值设置	1490
45.7 参数读取	1495
45.8 参数写入	1502
45.9 轴错误复位	1508
45.10 轴组错误复位	1511
45.11 触摸探针有效	1515
45.12 触摸探针无效	1529
45.13 凸轮表选择	1534
45.14 1周期当前值更改	1542
45.15 全部轴可运行	1550
45.16 轴组倍率修调值设置	1553
45.17 运动错误复位	1558
45.18 控制运算周期同步	1562
45.19 高级同步控制1周期当前位置计算	1565
45.20 高级同步控制凸轮指令当前位置计算	1568
45.21 数字凸轮开关输出	1571
第46章 运行系统的FB	1577
46.1 原点复位	1577
46.2 强制停止	1586
46.3 组强制停止	1591

46.4	绝对值定位	1597
46.5	相对值定位	1610
46.6	JOG运行	1619
46.7	速度控制	1629
46.8	转矩控制	1637
46.9	速度控制(包含位置循环)	1647
46.10	绝对值直线插补控制	1656
46.11	相对值直线插补控制	1670
46.12	绝对值圆弧插补控制	1683
46.13	相对值圆弧插补控制	1699
46.14	凸轮动作开始	1716
46.15	齿轮动作开始	1742
46.16	加减法定位	1752
46.17	齿隙补偿滤波器	1762
46.18	平滑滤波器	1776
46.19	移动方向限制滤波器	1786
46.20	速度限制滤波器	1795
46.21	轴控制开始等待	1803
46.22	轴组控制开始等待	1810
46.23	高级同步控制	1817
46.24	多轴定位数据运行	1823
46.25	绝对位置跟踪控制	1832
46.26	运动循环位置控制	1844
46.27	运动循环速度控制	1853
46.28	运动循环转矩控制	1857
第47章 一般FB		1863
47.1	配置文件读取	1863
47.2	配置文件写入	1877
第9部分 网络FB		
第48章 以太网用FB		1892
48.1	连接的建立/断开	1892
48.2	套接字通信	1897
48.3	SLMP	1903
第49章 CC-Link IE TSN用FB		1931
49.1	远程复位	1931
附录		1935
附1	指令处理时间	1935
索引		1938
指令索引		1940
修订记录		1949
关于产品的应用		1950
质保		1951

资讯与服务	1952
商标	1952

关联手册

要取得最新的e-Manual及手册PDF，请向当地三菱电机代理店咨询。

手册名称[手册编号]	内容
MELSEC MX控制器 MX-F型编程手册 [SH-082692CHN] (本手册)	记载了程序语言的规格、控制器的指令、通用函数/FB、运动控制FB、网络FB的规格的有关内容。
MELSEC MX控制器 MX-F型用户手册 [SH-082635CHN]	记载了控制器运行前的步骤、规格、软元件、存储器、功能、参数、故障排除等有关内容。
GX Works3 操作手册 [SH-081271CHN]	对GX Works3的系统配置、参数设置、在线功能的操作方法等有关内容进行说明。

术语


在本手册中，除非特别标明，否则将使用下述术语进行说明。

术语	内容
智能功能模块	是A/D、D/A转换模块等具有输入输出以外功能的模块。
对象	CANopen对应的设备保有的数据。
凸轮表	用于凸轮控制的运算配置文件。 (例)凸轮数据形式的运算配置文件、旋转刀具用凸轮数据形式的运算配置文件
凸轮表的第1点	根据凸轮表的插补方法指定不同意思也不同。 <ul style="list-style-type: none"> 在各区间指定，样条插补：表No. 1的点 直线插补：点No. 1的点
宗机	在使用了PTP(Precision Time Protocol)的时间同步中，作为时间同步源的设备。
全局标签	工程内创建多个程序数据时，对于所有程序数据皆为有效的标签。 全局标签中，有GX Works3自动生成的模块固有标签(模块标签)和对于任意指定的软元件可以创建的标签。
循环数据传送处理	在1个网络中，全部的站从开始循环传送至完成为止的动作。与控制器的顺序扫描为非同步进行。 循环数据传送处理时间根据数据量或瞬时传送请求等而变动。
CPU功能部	通过控制器进行可编程控制器控制及以太网通信的部分。
CC-Link IE TSN功能部	通过控制器进行CC-Link IE TSN通信的部分。
运动功能部	通过控制器进行运动控制的部分。
核心	搭载在控制器上的运算梯形图。控制器上搭载了多个核心。
基本核心	执行程序的核心。
扩展核心	可分配执行功能的核心。
信号流	为程序及FB的各步中运算的前次执行/非执行的状态。
系统错误	与轴不相关的错误或警告运动控制FB的轴信息(Axis)/轴组信息(AxesGroup)不正确时变为系统错误。
Jerk	加速度或者减速度的时间性变化比率。
软元件	是用于存储数据的控制器的存储器。根据用途有X/Y/M/D等。
设备站	主站以外的站(本地站、远程站)。
瞬时传送组No.	向任意的站进行瞬时传送的编号。对瞬时传送的对象站进行组指定，可以发送数据至相同组No.的站。
缓冲存储器	是用于存储设定值、监视值等数据的存储器。存储网络功能的设定值、监视值等。
缓冲	多重起动时，运动控制FB则进入待机状态。
缓冲FB	进行多重起动，变为执行等待状态(Busy为TRUE)的运动控制FB。
反馈位置	通过环形计数器对设备返回位置地址进行化整后的值。受当前位置更改的影响。
反馈速度	将实际轴中设备的反馈值转换为轴单位的值。
程序块	配置程序的程序部件的集合。
程序部件	是按各功能分别定义的程序单位。通过程序的部件化，在程序分级时可按处理内容及功能将低位处理分为若干个单位，创建各单位的程序。
主站	是控制网络整体的站。可以与全部的站进行循环传送及瞬时传送。1个网络中仅存在1个。
运动控制FB	运动控制FB相关的FB。名称从MC_或MCv_开始。
单播模式	对于1个站进行循环数据发送的通信模式。
模块标签	是各模块将特有定义的存储器(输入输出信号及缓冲存储器)以，任意字符串表示的标签。 对于使用的模块，GX Works3将自动生成标签，且该标签可作为全局标签使用。
标签	指在输入输出数据或内部处理中指定任意字符串的变量。
远程站	对位单位的输入输出信号和字单位的输入输出数据进行循环传送的站。也可以进行瞬时传送。
链接软元件	是CC-Link IE TSN功能部或网络模块内部具有的软元件(RX、RY、RWr、RWw)。
路由	与其他网络通信时对通信路径进行的控制。有自动选择通信路径的动态路由和设置任意通信路径的静态路由。
虚拟编码器轴	从变量生成指令位置的轴。用于单轴同步控制的主轴。
虚拟驱动轴	可以虚拟生成指令的虚拟轴。
虚拟轴	不与网络上的设备关联的轴。
虚拟连接轴	单轴同步控制中连接在FB之间传达指令的轴。
切断	数据链接异常时停止该站的数据链接的处理。
外部信号高精度输入	与设备站的信号检测时间关联的输入信号。
指定位置	运行系统FB中指定的指令位置地址。
指定速度	运行系统FB中指定的指令速度。
指令当前位置	通过环形计数器进行化整后的指令位置地址。受当前位置更改的影响。
指令当前速度	根据运动运算生成的当前的控制值。

术语	内容
轴	进行运动控制的对象。
轴组变量	包括轴组相关的参数及数据，AXES_GROUP型变量的实例。
轴报警	轴相关的错误及警告。
轴变量	包括轴相关的参数及数据，AXIS_*型变量的实例。
实际编码器轴	从与设备连接的编码器的当前位置生成指令位置的轴。用于单轴同步控制的主轴。
实际驱动轴	与对应CiA402驱动配置文件的csp/csv/cst模式(依次指令)的CC-Link IE TSN对应设备关联的轴。
实际轴	与网络上的设备关联的轴。
主轴(输入)绝对坐标	MC_CamTableSelect(凸轮表选择)的输入变量主轴绝对坐标(MasterAbsolute)的别称。
从轴(输出)绝对坐标	MC_CamTableSelect(凸轮表选择)的输入变量从轴绝对坐标(SlaveAbsolute)的别称。
输出变量	FB的输出参数。
切换速度	<ul style="list-style-type: none"> • Aborting指定时执行中的运动控制FB的指令当前速度。 • Blending指定时执行中的运动控制FB到达目标位置时的指令速度。
专用指令	为更容易进行用于使用智能功能模块功能的编程的指令。
进给机械位置	未通过环形计数器进行化整的指令位置地址。将原点作为基准，不受当前位置更改的影响。
多轴插补控制	直线插补及圆弧插补等多个轴相关运行的控制。指定通过轴组协调的轴。
多重起动	表示运动控制FB执行中，对于同一个轴执行另外的运动控制FB。
单轴同步	输出与Master(主轴)同步的Slave(从轴)1轴的位置信息(指令)的控制。
中继站	1个可编程控制器安装多个网络模块，向其他网络中继瞬时传送的站。
通信协议支持功能	<p>下述功能可以在GX Works3(通信协议支持功能)使用。</p> <ul style="list-style-type: none"> • 与对象设备相符合的协议的设置 • 协议设置数据的读取及写入
运行系统FB	将轴或者轴组作为参数，执行时轴状态或者轴组状态发生变化的运动控制FB。
输入变量	FB的输入参数。
通用HUB	CC-Link协会认定的CC-Link IE TSN Class A的交换集线器。
恢复	异常站变为正常时，重启该站的数据链接的处理。
目标位置	运行系统FB中指定的指令位置地址。
保留地址	根据RFC6890定义的用于特殊用途的保留IP地址。与全局IP网络连接时无法使用。
累计当前位置	未通过环形计数器进行化整的指令位置地址。受当前位置更改的影响。
连续更新	表示运动控制FB的ContinuousUpdate输入为TRUE期间，输入值连续反映到控制。
CC-Link IE TSN Class	CC-Link协会根据CC-Link IE TSN对应的设备及交换集线器的功能・性能进行的等级区分。关于CC-Link IE TSN Class，请参阅CC-Link协会发行的CC-Link IE TSN Installation Manual(BAP-C3007ENG-001)。
CC-Link IE TSN协议版本1.0	是以时间同步中使用了IEEE 802.1AS或IEEE1588的时间分割方式进行通信的协议。
CC-Link IE TSN协议版本2.0	<p>是以时间同步中使用了IEEE 802.1AS的时间分割方式加上时间管理・轮询方式进行通信的协议。</p> <p>如果对应此版本，系统配置为“Mixture of CC-Link IE TSN Class B/A or CC-Link IE TSN Class A only(CC-Link IE TSN Class B/A混合，或仅CC-Link IE TSN Class A)”时的限制将得到缓和，可连接设备会增多。</p>
CPU模块记录设置工具	用于数据记录的设置及维护的软件。
FB实例	粘贴到顺控程序中的FB。
GX LogViewer	用于显示通过数据记录收集的数据的软件。
GX Works3	是MELSEC可编程控制器软件包的产品名。
iQSS	表示iQ Sensor Solution。易于进行传感器的启动及维护，是传感器、可编程控制器、显示器、工程工具的联动。(见iQ Sensor Solution 参考手册)
MR Configurator2	伺服设置软件的产品名。
RWr	链接软元件的远程寄存器。设备站以16位(1字)单位向主站输入。
RWw	链接软元件的远程寄存器。主站以16位(1字)单位向设备站输出。
RX	链接软元件的远程输入。设备站以位单位向主站输入。
RY	链接软元件的远程输出。主站以位单位向设备站输出。
SB	链接特殊继电器的简称。表示CC-Link IE的模块运行状态、数据链接状态的位单位的信息。
SLMP	<p>表示SeamLess Message Protocol。</p> <p>是用于从外部设备对SLMP对应设备及SLMP对应设备上连接的可编程控制器进行访问的协议。</p>
SW	链接特殊寄存器的简称。表示CC-Link IE的模块运行状态、数据链接状态的16位(1字)单位的信息。
TSN HUB	CC-Link协会认定的CC-Link IE TSN Class B的交换集线器。

总称/简称

在本手册中，除特别标明外，皆使用下述总称/简称进行说明。

总称/简称	说明
工程工具	是进行控制器及伺服放大器的设置、编程、调试及维护用的工具。
控制器	MXF100-8-N32、MXF100-8-P32、MXF100-16-N32、MXF100-16-P32的总称。
数据链接	循环传送及瞬时传送的总称。
驱动器模块	伺服放大器等的电机驱动设备的总称。
运算配置文件	在各种控制中使用的波形数据的总称。
站子ID编号	表示连接CC-Link对应的通信模块的传感器的ID编号。
输入输出模块	表示输入模块、输出模块、输入输出模块。
CC-Link IE	下述总称。 <ul style="list-style-type: none"> • CC-Link IE TSN • CC-Link IE控制网络 • CC-Link IE现场网络
csp	循环位置模式(Cyclic synchronous position mode)的简称(驱动器模块侧控制模式的一种)。
cst	循环转矩模式(Cyclic synchronous torque mode)的简称(驱动器模块侧控制模式的一种)。
csv	循环速度模式(Cyclic synchronous velocity mode)的简称(驱动器模块侧控制模式的一种)。
ct	手动控制模式(Continuous operation to torque control mode)的简称(驱动器模块侧控制模式的一种)。
以太网设备	IP通信对应设备的总称。(计算机、视觉传感器、条形码阅读器等)
以太网搭载模块	下述模块的总称。 <ul style="list-style-type: none"> • 控制器(以太网功能使用时)
GOT	三菱电机图形操作终端GOT1000、GOT2000系列的总称。
hm	原点回归模式(Homing mode)的简称(驱动器模块侧控制模式的一种)。
ICMP	Internet Control Message Protocol的简称。用于交换IP网络上发生的错误及网络相关信息的协议。
iQSS对应设备	表示iQ Sensor Solution对应的设备。  iQ Sensor Solution 参考手册
MCFB	运动控制FB的简称。
MR-J5(W)-G	MR-J5_G/MR-J5_G-RJ/MR-J5_G-LL/MR-J5W_-G/MR-J5D_-G型伺服放大器的总称。
PDO	Process Data Object的简称。在多个CANopen节点之间周期性传送的应用程序对象的集合体。
PTP	Precision Time Protocol的简称。用于网络内设备间时间同步的通信协议。
SDO	Service Data Object的简称。用于访问任意CANopen节点的对象字典内的对象条目的信息。在站之间非周期性发送接收。
SLMP	Seamless Message Protocol的简称。用于从计算机及显示器等外部设备及SLMP对应的模块(以太网搭载模块及CC-Link IE TSN的模块等)访问SLMP对应设备的协议。

第1部分 编程概要

本部分由下述章构成。

1 指令概要

2 程序部件

3 梯形图语言

4 ST语言

5 FBD/LD语言

1 指令概要

1.1 指令的构成

控制器中可使用的指令的大多可分为指令部及操作数部。

指令部及操作数部的用途如下所示。

- 指令部：该指令的功能如下所示。
- 操作数部：指令中使用的数据如下所示。

操作数部分为源数据、目标数据、数值数据。


源(s)

源是运算中使用的数据。

根据各指令中指定的标签及软元件，其情况如下所示。

种类	内容
常数	指定运算中使用的数值。 由于是在创建程序时进行设置，因此在程序执行中不能更改。 将常数在可变数据中使用的情况下，应进行变址修饰。*1
软元件 标签	指定存储运算中使用的数据的软元件或标签。 在执行运算之前需要预先将数据存储到指定的软元件或标签中。 程序执行过程中，通过更改指定的软元件或标签中存储的数据，可以更改该指令中使用的数据。

*1 关于变址修饰，请参阅下述手册。

所使用的控制器的用户手册

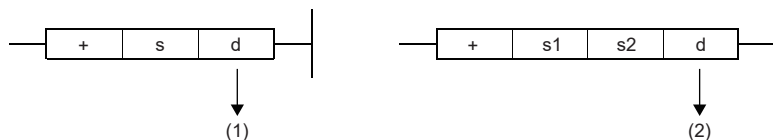
目标(d)

目标中存储运算后的数据。

但是，根据指令，有时在运算前需要在目标中存储运算中使用的数据。

例

BIN16位数据的加法运算指令的情况下



(1) 执行运算之前，存储使用的数据。

(2) 仅存储运算结果。

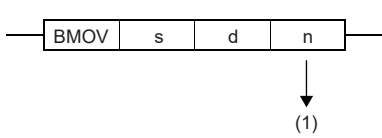
目标必须设置用于存储数据的标签及软元件。

软元件数/传送数/数据数/字符串数等的数值(n)

软元件数、传送数、数据数、字符串数等的数值，指定使用多个软元件的指令及指定重复次数、指定处理的数据数・字符串数等的指令中使用的软元件数、传送数、数据数、字符串数等。

例

块传送指令的情况下



(1) 指定通过BMOV指令传送的传送数。

指定软元件数、传送数、字符串数等的容量时，可设置范围为0~65535、0~4294967295。^{*1}

但是，软元件数、传送数、字符串数等的容量指定为0的情况下，该指令将变为无处理。

此外，根据软元件存储器容量或文件存储容量，可能变为小于上述指定容量上限值的值。^{*2}

*1 设置范围根据指令而有所不同。详细内容，请参阅各指令。

*2 根据机型，软元件存储器容量或文件存储容量将有所不同。关于详细内容，请参阅下述手册。

所使用的控制器的用户手册

要点

传送数等的数值中使用较大值的情况下，扫描时间将延迟，应加以注意。

1.2 数据的指定方法

控制器的各指令中可使用的数据有下述几种类型。

数据	分类
位数据	位数据
16位数据(字数据)	带符号BIN16位数据
	无符号BIN16位数据
32位数据(双字数据)	带符号BIN32位数据
	无符号BIN32位数据
实数数据(浮点数据)	单精度实数数据
	双精度实数数据
BCD数据	BCD4位数据
	BCD8位数据
	BCD16位数据
字符串数据	字符串
	Unicode字符串

作为软元件处理的数据

将设置数据通过软元件进行指定的情况下，可指定的软元件/常数的类型如下所示。

数据类型	内容	可指定的软元件/常数*1
位	可以处理位数据。 ☞ 31页 位数据	<ul style="list-style-type: none"> • 位软元件 • 字软元件的位指定
字	可以处理字数据。 ☞ 32页 16位数据(字数据)	<ul style="list-style-type: none"> • 字软元件 • 位软元件的位指定(K1~K4)*2
带符号BIN16位	可以处理16位数据。 在带符号及无符号中，值的范围有所不同。 ☞ 32页 16位数据(字数据)	<ul style="list-style-type: none"> • 10进制常数 • 16进制常数
无符号BIN16位		
双字	可以处理双字数据。 ☞ 34页 32位数据(双字数据)	<ul style="list-style-type: none"> • 字软元件 • 双字软元件 • 位软元件的位指定(K1~K8)*2 • 10进制常数 • 16进制常数
带符号BIN32位	32位数据或16位数据可以处理2个连续的数据。 在带符号及无符号中，值的范围有所不同。 ☞ 34页 32位数据(双字数据)	
无符号BIN32位		
BCD4位	可以处理BCD4位数据。 将16位数据以4位进行分割，将各位以0~9进行指定。	<ul style="list-style-type: none"> • 字软元件 • 位软元件的位指定(K1~K4)*2 • 10进制常数 • 16进制常数
BCD8位	可以处理BCD8位数据。 将32位数据以8位进行分割，将各位以0~9进行指定。	<ul style="list-style-type: none"> • 字软元件 • 双字软元件 • 位软元件的位指定(K1~K8)*2 • 10进制常数 • 16进制常数
单精度实数	可以处理单精度实数数据(单精度浮点数据)。 ☞ 37页 单精度实数数据的构成	<ul style="list-style-type: none"> • 字软元件 • 双字软元件 • 实数常数
双精度实数	可以处理双精度实数数据(双精度浮点数据)。 ☞ 38页 双精度实数数据的构成	<ul style="list-style-type: none"> • 字软元件 • 双字软元件 • 实数常数
字符串	可以处理ASCII代码、移位JIS代码的字符串数据。 ☞ 40页 字符串数据	<ul style="list-style-type: none"> • 字软元件 • 字符串常数
Unicode字符串	可以处理Unicode的字符串数据。 ☞ 40页 字符串数据	<ul style="list-style-type: none"> • 字软元件 • 字符串常数
软元件名	可以直接指定软元件。	<ul style="list-style-type: none"> • 可以使用的软元件中对应的软元件名

*1 通过指令进行源(s)及数值数据(n)指定的数据中，可以使用常数。

*2 关于指定方法，请参阅各数据类型的详细页面。

作为标签处理的数据


将设置数据通过标签进行指定的情况下，可指定的标签类型如下所示。

■基本数据类型

数据类型(标签)	可指定的标签
位 (BOOL)	<ul style="list-style-type: none"> • 位型标签 • 字[无符号]/位串[16位]型标签的位指定 • 字[带符号]型标签的位指定 • 定时器/累计定时器/长定时器/长累计定时器型标签的触点・线圈 • 计数器/长计数器型标签的触点・线圈
字[无符号]/位串[16位] (WORD)	<ul style="list-style-type: none"> • 字[无符号]/位串[16位]型标签 • 位型标签的位指定(K1~K4) • 定时器/累计定时器型标签的当前值 • 计数器型标签的当前值
双字[无符号]/位串[32位] (DWORD)	<ul style="list-style-type: none"> • 双字[无符号]/位串[32位]型标签 • 位型标签的位指定(K1~K8) • 长定时器/长累计定时器型标签的当前值 • 长计数器型标签的当前值
字[带符号] (INT)	<ul style="list-style-type: none"> • 字[带符号]型标签 • 位型标签的位指定(K1~K4) • 定时器/累计定时器型标签的当前值 • 计数器型标签的当前值
双字[带符号] (DINT)	<ul style="list-style-type: none"> • 双字[带符号]型标签 • 位型标签的位指定(K1~K8) • 长定时器/长累计定时器型标签的当前值 • 长计数器型标签的当前值
单精度实数 (REAL)	<ul style="list-style-type: none"> • 单精度实数型标签
双精度实数 (LREAL)	<ul style="list-style-type: none"> • 双精度实数型标签
时间 (TIME)	<ul style="list-style-type: none"> • 时间型标签
字符串 (STRING)	<ul style="list-style-type: none"> • 字符串型标签
字符串[Unicode] (WSTRING)	<ul style="list-style-type: none"> • 字符串[Unicode]型标签
指针 (POINTER)	<ul style="list-style-type: none"> • 指针型标签

要点

关于各标签的内容，请参阅下述手册。

 所使用的控制器的用户手册

■总称数据类型

是汇总了若干个基本数据类型的标签的数据类型。

允许函数及FB的自变量、恢复值等多个数据类型的情况下，使用总称数据类型。

定义为总称数据类型的标签，在低位的数据类型的任意类型中均可使用。

数据类型(标签)				可指定的数据类型			
ANY* ¹	ANY_ELEMENTARY	ANY_BIT		ANY_BOOL	位		
				ANY_BITADDR* ¹	位		
				ANY16_U	字[无符号]/位串[16位]		
				ANY32_U	双字[无符号]/位串[32位]		
		ANY_WORDADDR	ANY_NUM	ANY_INT	ANY16	ANY16_S	字[带符号]
					ANY16_U	字[无符号]/位串[16位]	
				ANY_REAL	ANY32	ANY32_S	双字[带符号]、时间
					ANY32_U	双字[无符号]/位串[32位]	
					ANY_REAL_32	单精度实数	
			ANY_REAL_64	双精度实数			
			ANY_STRING		ANYSTRING_SINGLE	字符串	
			ANY_STRING		ANYSTRING_DOUBLE	字符串[Unicode]	
			ANY16_OR_STRING_SINGLE		ANY16_S	字[带符号]	
					ANY16_U	字[无符号]/位串[16位]	
		ANYSTRING_SINGLE			字符串		
		ANY_DT		字[带符号]、字[无符号]/位串[16位]			
		ANY_TM		字[带符号]、字[无符号]/位串[16位]			
ANY_STRUCT* ¹				结构体			
STRUCT				结构体			

*1 也可作为数组使用。

■总称数据类型(数组)

关于下述总称数据类型，应定义数组要素数。

数据类型(标签)			可指定的数据类型	
ANYBIT_ARRAY			位的数组	
ANYWORD_ARRAY	ANY16_ARRAY		ANY16_S_ARRAY	字[带符号]的数组
			ANY16_U_ARRAY	字[无符号]/位串[16位]的数组
	ANY32_ARRAY		ANY32_S_ARRAY	双字[带符号]的数组、时间的数组
			ANY32_U_ARRAY	双字[无符号]/位串[32位]的数组
	ANY_REAL_ARRAY		ANY_REAL_32_ARRAY	单精度实数的数组
			ANY_REAL_64_ARRAY	双精度实数的数组
	ANY_STRING_ARRAY		ANY_STRING_SINGLE_ARRAY	字符串的数组
			ANY_STRING_DOUBLE_ARRAY	字符串[Unicode]的数组
STRUCT_ARRAY			结构体的数组	

位数据

数据容量及数据的范围

位数据是将触点及线圈等以1位单位处理的数据。

数据名称	数据容量	值的范围
位数据	1位	0、1

位软元件/标签中位数据的处理

对1点字软元件/标签，可处理1位数据。

字软元件中位数据的处理

对字软元件通过进行位No. 指定，可以处理指定位No. 的位数据。

位指定的表示方法变为“字软元件编号.位No.”。

位No. 的指定是以16进制数在0~F的范围内进行指定。

例如，D0的位5(b5)以“D0.5”进行指定，D0的位10(b10)以“D0.A”进行指定。

可进行字软元件的位指定的软元件如下所示。

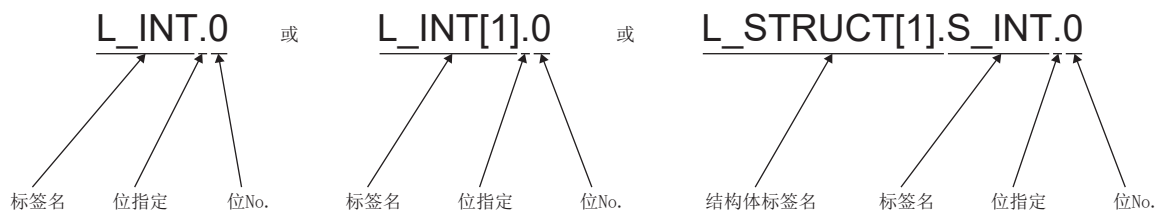
项目	软元件
可进行字软元件的位指定的软元件	<ul style="list-style-type: none"> • 数据寄存器(D) • 链接寄存器(W、J□\W) • 链接特殊寄存器(SW、J□\SW) • 函数寄存器(FD) • 特殊寄存器(SD) • 模块访问软元件(U□\G) • CPU缓冲存储器访问软元件(U3E□\G) • 文件寄存器(R、ZR) • 模块刷新寄存器(RD)

字型标签中位数据的处理

在字型标签中，通过进行位No. 指定，可以处理指定位No. 的位数据。

位指定的表示方法变为“标签名.位No.”。

例



可进行位指定的标签的数据类型如下所示。

项目	数据类型
可进行位指定的标签的数据类型	<ul style="list-style-type: none"> • 字[带符号](INT型) • 字[无符号]/位串[16位](WORD型) • 定时器(TIMER型)的当前值(N)*1 • 累计定时器(RETENTIVETIMER型)的当前值(N)*1 • 计数器(COUNTER型)的当前值(N)*1

*1 梯形图的情况下不能指定。

16位数据(字数据)

数据容量及数据的范围

16位数据中，有带符号16位数据、无符号16位数据。

对于带符号16位数据，负数以2的补数表示。

数据名称	数据容量	值的范围	
		10进制表示	16进制表示
带符号16位数据	16位(1字)	-32768~32767	0000H~FFFFH
无符号16位数据		0~65535	

位软元件中16位数据的处理

通过对位软元件进行位指定，可以作为16位数据处理。

项目		表示	示例
位软元件	链接直接软元件以外	K□位软元件起始编号 □: 位数(在1~4的范围内指定)	K4X40 K2M113
	链接直接软元件	J□\K□位软元件起始编号 左侧□: 网络No. 右侧□: 位数(在1~4的范围内指定)	J1\K3B10 J10\K2Y40

位型数组的标签中16位数据的处理

对位型数组的标签通过进行位指定，可以作为16位数据处理。

对位型数组的标签进行位指定作为16位数据处理时的表示方法如下所示。

项目	表示	示例
位型数组的标签	K□标签名 □: 位数(在1~4的范围内指定) 标签的位指定以无要素指定方式进行指定。	K1L_BOOL

位指定的范围

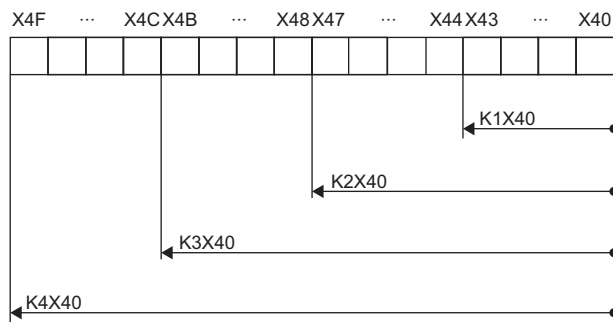
各位指定的16位数据的范围如下所示。

位指定	10进制表示	16进制表示
K1	0~15	0H~FH
K2	0~255	00H~FFH
K3	0~4095	000H~FFFH
K4	带符号16位数据: -32768~32767 无符号16位数据: 0~65535	0000H~FFFFH

例

对X40进行了位指定情况下的对象点数如下所示。

- K1X40→X40~X43的4点为对象
- K2X40→X40~X47的8点为对象
- K3X40→X40~X4B的12点为对象
- K4X40→X40~X4F的16点为对象



■源(s)侧指定了位指定的位软元件的情况下

对指令源侧指定已进行位指定的位软元件的情况下，目标侧字软元件的高位中将存储0，该高位是源侧中已进行位指定定位以后的位。

梯形图示例	处理
<p>• 16位数据指令的情况下</p>	

■目标(d)侧指定了位指定的位软元件的情况下

指令的目标侧有位指定的情况下，位指定的点数将成为目标侧的对象。

位指定点数以后的高位位软元件不变化。

梯形图示例	处理
<p>• 源数据为字软元件的情况下</p>	<p>(1) 不变化。</p>

字软元件/标签中16位数据的处理

■字软元件

1点字软元件可处理16位数据。

■字型标签

1点字型标签中可处理16位数据。

32位数据(双字数据)

数据容量及数据的范围

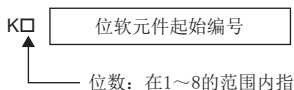
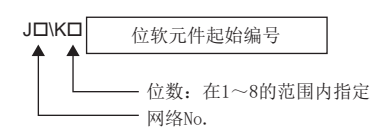
32位数据中，有带符号32位数据、无符号32位数据。

对于带符号32位数据，负数以2的补数表示。

数据名称	数据容量	值的范围	
		10进制表示	16进制表示
带符号32位数据	32位(2字)	-2147483648~2147483647	00000000H~FFFFFFFH
无符号32位数据		0~4294967295	

位软元件中32位数据的处理

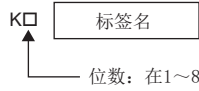
对位软元件通过进行位指定，可以作为32位数据处理。

项目	表示	示例
位软元件 链接直接软元件以外		K8X80 K6B018
链接直接软元件		J1\K7B30 J10\K5X128

位型数组的标签中32位数据的处理

对位型数组的标签通过进行位指定，可以作为32位数据处理。

对位型数组的标签进行位指定作为32位数据处理时的表示方法如下所示。

项目	表示	示例
位型数组的标签	 <p>标签的位指定以无要素指定方式进行指定。 梯形图以外的语言时，应在K5~K8的范围内指定。(梯形图以外的语言的情况下，K1~K4范围内的位指定将被作为ANY16处理。)</p>	K8L_BOOL

位指定的范围

各位指定的32位数据的范围如下所示。

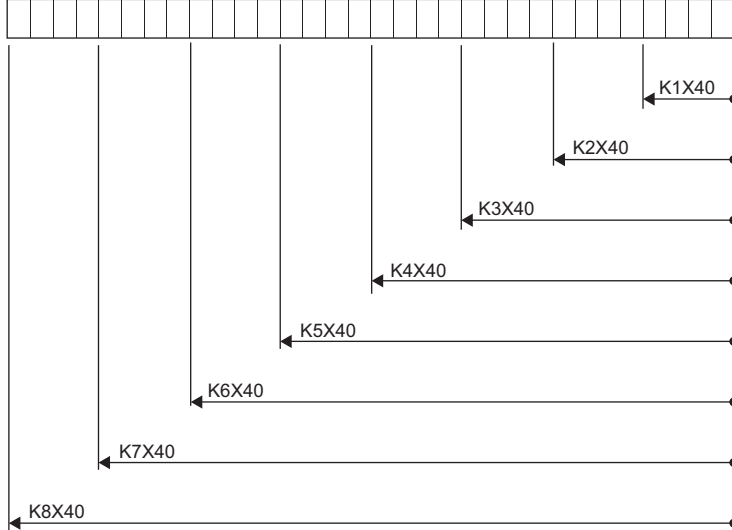
位指定	10进制表示	16进制表示
K1	0~15	0H~FH
K2	0~255	00H~FFH
K3	0~4095	000H~FFFH
K4	0~65535	0000H~FFFFH
K5	0~1048575	00000H~FFFFFH
K6	0~16777215	000000H~FFFFFFH
K7	0~268435455	0000000H~FFFFFFFH
K8	带符号32位数据: -2147483648~2147483647 无符号32位数据: 0~4294967295	00000000H~FFFFFFFFH

例

对X40进行了位指定情况下的对象点数如下所示。

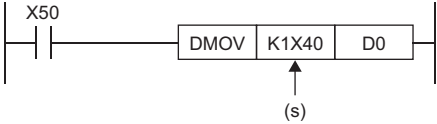
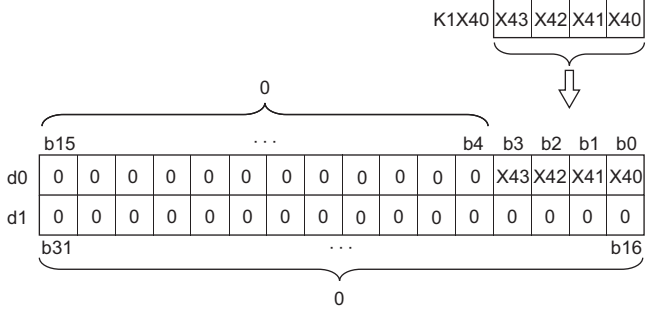
- K1X40→X40~X43的4点为对象
- K2X40→X40~X47的8点为对象
- K3X40→X40~X4B的12点为对象
- K4X40→X40~X4F的16点为对象
- K5X40→X40~X53的20点为对象
- K6X40→X40~X57的24点为对象
- K7X40→X40~X5B的28点为对象
- K8X40→X40~X5F的32点为对象

X5F ... X5CX5B ... X58X57 ... X54X53 ... X50X4F ... X4CX4B ... X48X47 ... X44X43 ... X40



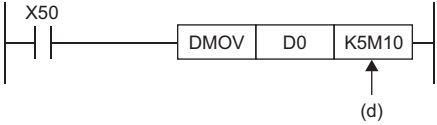
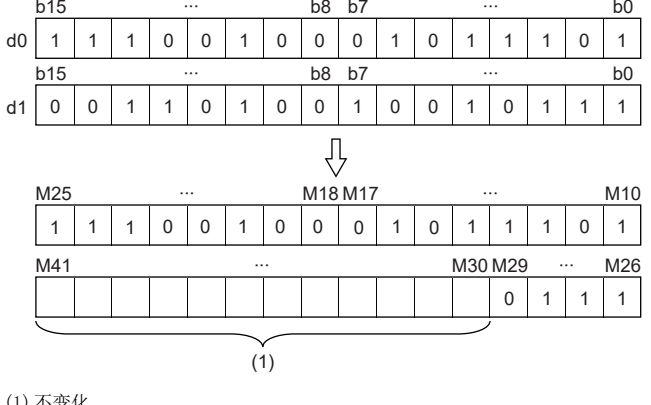
■源(s)侧指定了位指定的位软元件的情况下

对指令源侧指定已进行位指定的位软元件的情况下，目标侧字软元件的高位中将存储0，该高位是源侧中已进行位指定定位以后的位。

梯形图示例	处理
<p>• 32位数据的指令的情况下</p> 	

■目标(d)侧指定了位指定的位软元件的情况下

指令的目标侧有位指定的情况下，位指定的点数将成为目标侧的对象。位指定点数以后的高位位软元件不变化。

梯形图示例	处理
<p>• 源数据为字软元件的情况下</p> 	

字软元件/标签中32位数据的处理

■字软元件

2点的字软元件可作为32位数据处理。

但是，对于下述软元件，1点软元件可处理32位数据。

- 长定时器(LT)
- 长累计定时器(LST)
- 长计数器(LC)
- 长变址寄存器(LZ)

■双字型标签

双字型标签的1点可作为32位数据处理。

实数数据(浮点数据)

数据容量及数据的范围

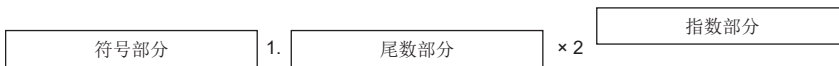
实数数据中, 有以32位表示的单精度实数数据及以64位表示的双精度实数数据。

实数数据只能存储到位软元件以外的软元件或单精度实数型标签、双精度实数型标签中。

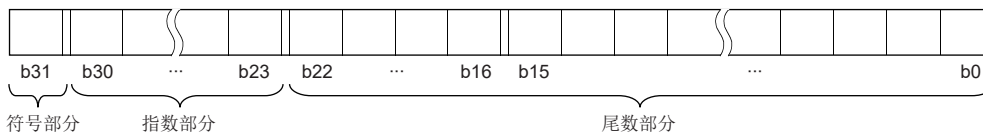
数据名称		数据容量	值的范围
单精度实数数据(单精度浮点数据)	正数	32位(2字)	$2^{-126} \leq \text{实数} < 2^{128}$
	零		0
	负数		$-2^{128} < \text{实数} \leq -2^{-126}$
双精度实数数据(双精度浮点数据)	正数	64位(4字)	$2^{-1022} \leq \text{实数} < 2^{1024}$
	零		0
	负数		$-2^{1024} < \text{实数} \leq -2^{-1022}$

单精度实数数据的构成

单精度实数数据由符号部、尾数部、指数部所构成, 按以下方式表示。



单精度实数数据内部表示时的位构成及含义如下所示。



■符号部(1位)

数值的正负以1位表示。0表示正数或0。1表示负数。

■尾数部(23位)

是指将单精度实数以2进制数 $1.XXXXX \dots \times 2^N$ 表示时的XXXXX...

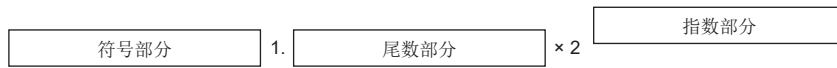
■指数部(8位)

是指将单精度实数以2进制数 $1.XXXXX \dots \times 2^N$ 表示时的N。单精度实数中指数部的值与N的关系如下所示。

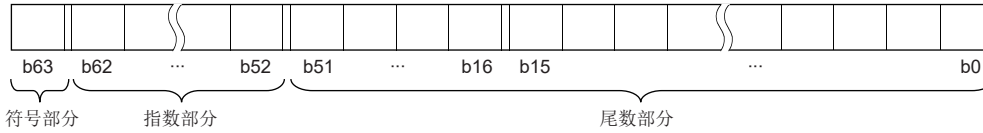
指数部(b24~b30)	FFH	FEH	FDH	...	81H	80H	7FH	7EH	...	02H	01H	00H
N	未使用	127	126	...	2	1	0	-1	...	-125	-126	未使用

双精度实数数据的构成

双精度实数数据由符号部、尾数部、指数部所构成，按下述方式表示。



双精度实数数据内部表示时的位构成及含义如下所示。



■符号部 (1位)

数值的正负以1位表示。0表示正数或0。1表示负数。

■尾数部 (52位)

是指将单精度实数以2进制数 $1.XXXXX\dots\times 2^N$ 表示时的XXXXX...

■指数部 (11位)

是指将单精度实数以2进制数 $1.XXXXX\dots\times 2^N$ 表示时的N。单精度实数中指数部的值与N的关系如下所示。

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

注意事项

■通过工程工具设置单精度实数的输入值的情况下

在工程工具中，对单精度实数数据以32位的单精度进行，因此有效位数约为7位。

单精度实数数据的输入值超过7位的情况下，将被化整为对第8位进行了四舍五入后的值。

因此，四舍五入后的值超过了-2147483648~2147483647的范围时，将不会变为所期望的值。

例

将输入值设置为“2147483647”的情况下，第8位的“6”将被四舍五入，因此以“2147484000”处理。

例

将输入值设置为“E1.1754943562”的情况下，第8位的“3”将被四舍五入，因此以“E1.175494”处理。

应将输入值设到下述范围内。如果超过下述范围，将变为转换错误。

小数表现： $0.0000000001 \leq \text{实数数据的绝对值} \leq 999999900000.0$

指数表现： $1.175494351E-38 \leq \text{实数数据的绝对值} \leq 3.402823466E+38$

■通过工程工具设置双精度实数的输入值的情况下

在工程工具中，对双精度实数数据以64位的双精度进行处理，因此有效位数约为15位。

双精度实数数据的输入值超过15位的情况下，将被化整为对第16位进行了四舍五入后的值。

因此，四舍五入后的值超过了-2147483648~2147483647的范围时，将不会变为所期望的值。

例

将输入值设置为“2147483646.12345678”的情况下，第16位的“6”将被四舍五入，因此以“2147483646.12346”处理。

例

将输入值设置为“E1.7976931348623157+307”的情况下，第16位的“5”将被四舍五入，因此以“E1.79769313486232+307”处理。

应将输入值设到下述范围内。如果超过下述范围，将变为转换错误。

小数表现： $0.00000000000000000001 \leq \text{实数数据的绝对值} \leq 9999999999999999000000.0$

指数表现： $2.22507385850721E-308 \leq \text{实数数据的绝对值} \leq 1.79769313486231E+308$

在工程工具的监视功能中，可以监视控制器的实数数据。

实数数据中表示“0”时，将下述范围全部设置为0。


- 单精度实数数据的情况下：b0~b31
- 双精度实数数据的情况下：b0~b63

实数数据的设置范围如下所示。^{*1}

- 单精度实数数据的情况下： $-2^{128} < [\text{单精度实数数据}] \leq -2^{-126}$ 、 0 、 $2^{-126} \leq [\text{单精度实数数据}] < 2^{128}$
- 双精度实数数据的情况下： $-2^{1024} < [\text{双精度实数数据}] \leq -2^{-1022}$ 、 0 、 $2^{-1022} \leq [\text{双精度实数数据}] < 2^{1024}$

实数数据中请勿指定“-0”（仅最高位1）。以-0进行实数运算时将变为运算错误。

*1 关于上溢及下溢时的动作及输入了特殊值时的动作，请参阅下述手册。

 所使用的控制器的用户手册

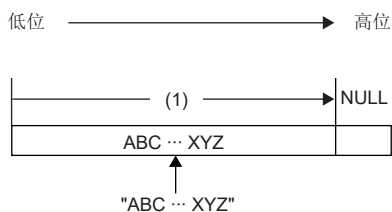
字符串数据

字符串数据的形式

字符串数据中有下述类型，可以将以NULL代码结束的字符代码串作为字符串处理。

种类	字符代码	最终字符
字符串	ASCII代码、移位JIS代码	NULL (00H)
Unicode字符串	Unicode (UTF-16(小端字节序))	NULL (0000H)

将字符串数据存储到软元件或数组中时，将从软元件编号或数组的要素编号的低位至高位的顺序存储字符串数据。



(1) 字符代码列

字符串的表示方法

梯形图程序中字符串的表示方法如下所示。

数据类型		表示方法	表示示例
字符串	STRING	用双引号(")将字符串(ASCII代码、移位JIS代码)、Unicode字符串围住。	"ABC"
字符串[Unicode]	WSTRING		

ST程序中的字符串的表示方法如下所示。

数据类型		表示方法	表示示例
字符串	STRING	用单引号(')将字符串(ASCII代码、移位JIS代码)围住。	'ABC'
字符串[Unicode]	WSTRING	将Unicode字符串用双引号(")围住。	"ABC"

FBD/LD程序中字符串的表示方法如下所示。

数据类型		表示方法	表示示例
字符串	STRING	用单引号(')将字符串(ASCII代码、移位JIS代码)围住。	
字符串[Unicode]	WSTRING	将Unicode字符串用双引号(")围住。	

数据的范围

字符串数据的范围如下所示。

种类	标签中可设置的最多字符数	字符串常数中可使用的最多字符数
字符串	半角255字符(不包括终端的NULL。)	半角255字符(不包括终端的NULL。)
Unicode字符串*1	255字符(不包括终端的NULL。)	255字符(不包括终端的NULL。)

*1 Unicode字符串可在基本多语言中使用。

数据存储中必要的字数

字符串数据可存储到字软元件中。

字符串数据的存储必要字数如下所示。

字符串的字节数	用于存储字符串的必要字数	用于存储Unicode字符串的必要字数
0字节	1[字]	1[字]
奇数字节	(字符串的字节数+1)÷2[字]	—(由于1字符为偶数字节。)
偶数字节	(字符串的字节数+2)÷2[字]	字符数+1[字]

字符串数据的存储目标

字符串数据的存储目标示意图如下所示。

■字符串的情况下

在字符串的存储示意图中，NULL表示NULL代码(00H)。

存储的字符串	从D0开始存储了字符串数据时的示意图	从字型标签的数组arrayA[0]开始存储了字符串数据时的示意图												
空字符串 ("或'")	D0 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td><td>NULL</td></tr></table>	NULL	NULL	arrayA[0] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td><td>NULL</td></tr></table>	NULL	NULL								
NULL	NULL													
NULL	NULL													
ABC	D0 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>B</td><td>A</td></tr></table> D1 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td><td>C</td></tr></table>	B	A	NULL	C	arrayA[0] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>B</td><td>A</td></tr></table> arrayA[1] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td><td>C</td></tr></table>	B	A	NULL	C				
B	A													
NULL	C													
B	A													
NULL	C													
ABCD	D0 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>B</td><td>A</td></tr></table> D1 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>D</td><td>C</td></tr></table> D2 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td><td>NULL</td></tr></table>	B	A	D	C	NULL	NULL	arrayA[0] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>B</td><td>A</td></tr></table> arrayA[1] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>D</td><td>C</td></tr></table> arrayA[2] <table border="1" style="display: inline-table; vertical-align: middle;"><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字符串的情况下





在Unicode字符串的存储示意图中，NULL表示NULL代码(0000H)。

存储的字符串	从D0开始存储了字符串数据时的示意图	从字型标签的数组arrayA[0]开始存储了字符串数据时的示意图										
空字符串("")	D0 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td></tr></table>	NULL	arrayA[0] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td></tr></table>	NULL								
NULL												
NULL												
ABCD	D0 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>A</td></tr></table> D1 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>B</td></tr></table> D2 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>C</td></tr></table> D3 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>D</td></tr></table> D4 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td></tr></table>	A	B	C	D	NULL	arrayA[0] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>A</td></tr></table> arrayA[1] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>B</td></tr></table> arrayA[2] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>C</td></tr></table> arrayA[3] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>D</td></tr></table> arrayA[4] <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>NULL</td></tr></table>	A	B	C	D	NULL
A												
B												
C												
D												
NULL												
A												
B												
C												
D												
NULL												

1.3 执行条件

执行条件的类型

指令的执行条件有以下类型。

执行条件		说明*1
ON时执行		是ON中执行型的指令，仅在指令的前条件为ON期间执行该指令。前条件为OFF的情况下，不执行该指令，不进行处理。
上升沿执行		是仅在ON时执行1次型的指令，仅在指令的前条件的上升沿时 (OFF→ON) 执行指令，以后即使条件为ON也不执行该指令，不进行处理。
OFF时执行		是OFF中执行型的指令，仅在指令的前条件为OFF期间执行该指令。前条件为ON的情况下，不执行该指令，不进行处理。
下降沿执行		是OFF时执行1次型的指令，仅在指令的前条件的下降沿时 (ON→OFF) 执行指令，以后即使条件为OFF也不执行该指令，不进行处理。
常时执行	—	是常时执行的指令，与指令的前条件的ON/OFF无关，始终执行。前条件为OFF的情况下，该指令将进行OFF处理。

*1 在ST语言、FBD/LD语言中，指令中的EN将成为指令的前条件。

各指令的执行条件

各指令中执行条件有所不同。关于各指令的执行条件，请参阅各指令的页面。

在ST语言、FBD/LD语言中，EN将成为执行条件，仅在EN为TRUE的情况下才执行。ENO变为与EN相同的状态。

通用函数/FB，根据EN的有无其执行条件有所不同。无EN的情况下将常时执行。关于带EN情况，请参阅通用函数/FB的页面。

有关运动控制FB执行条件的详细内容，请参阅下述章节。

☞ 1375页 执行指令(Execute)类型/有效(Enable)类型

1.4 编程时的注意事项

指令通用的错误

执行指令时错误的条件如下所示。

错误内容*1	错误代码
指定了超出范围的输入输出编号(000H~FFFH、3E0H~3E3H以外)。	2800H
指定了不存在对象模块的输入输出编号。	2801H
指定了指令中不能指定的模块的输入输出编号。	2803H
指定了超出范围的网路编号(1~239以外)。	2804H
指定了不存在的网路编号。	2805H
<ul style="list-style-type: none"> 指令中指定的软元件或标签超出了允许使用范围。 在CPU参数的文件设置中未设置文件寄存器，或在程序中未设置使用的文件寄存器的状态下，访问了文件寄存器。 	2820H
<ul style="list-style-type: none"> 超出指令中指定的模块具有的缓冲存储器的范围。 指令中指定的模块是不具有缓冲存储器的模块。 	2823H

*1 在触点指令中未检测出错误，运算结果变为非导通状态。

执行指令时的软元件、标签的范围检查

软元件的范围检查

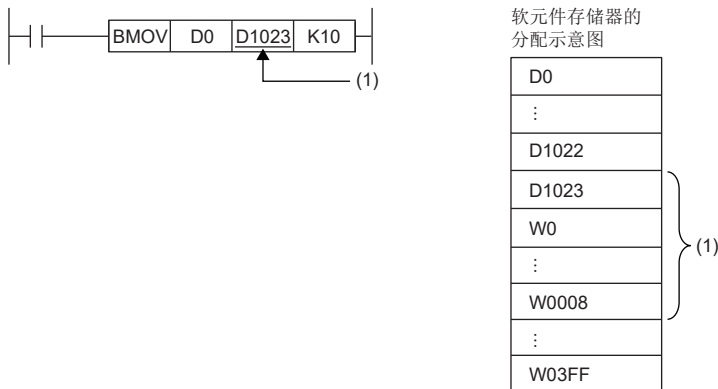
通过指令指定了软元件的情况下，将进行范围检查，因此应创建其运算结果不会超出相应软元件的范围的程序。

指定了超出该软元件范围的情况下，将发生错误(错误代码：2820H)。

将软元件中分配的标签通过程序中的指令进行了指定的情况下也相同。

例

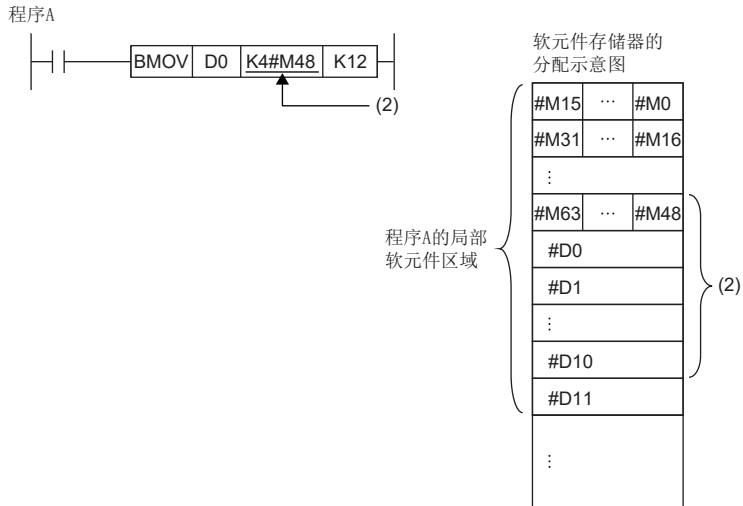
分配软元件存储器时，全局软元件D1023的下一个已变为W0的情况下



(1) 传送目标范围为D1023~D1032。D1024~D1032不存在，因此将变为错误。

例

已在M0~M63、D0~D11的范围内设置局部元件的情况下



(2) 传送目标范围为程序A的局部软件元件#M48~#M239。#M64~#M239不存在，因此将变为错误。

文件寄存器的范围检查

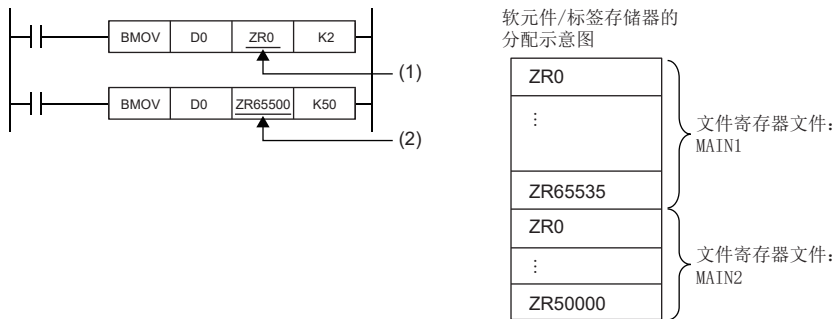
通过指令指定了文件寄存器的情况下，将进行范围检查，因此应创建其运算结果不会超出相应文件寄存器的范围的程序。

指定时超出文件寄存器 (ZR) 范围的情况下，将发生错误 (错误代码：2820H)。

指定时超出文件寄存器 (R) 中使用的块No. 的文件寄存器范围的情况下，将发生错误 (错误代码：2820H)。

例

指定了文件寄存器 (ZR) 的情况下

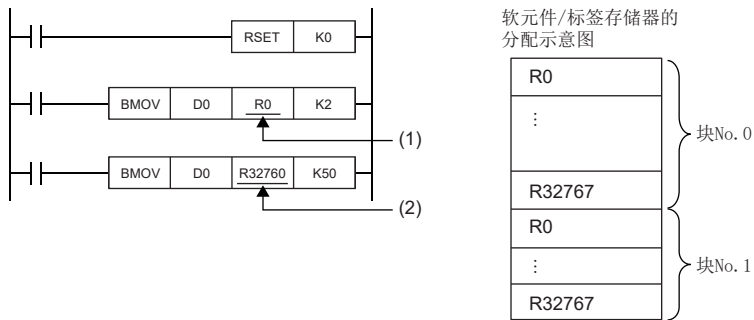


(1) 传送目标在文件寄存器MAIN1的范围内。数据将被写入ZR0~ZR1。

(2) 传送目标超出文件寄存器MAIN1的范围。由于超出文件寄存器MAIN1的区域范围，因此发生错误。

例

指定了文件寄存器 (R) 的情况下



(1) 传送目标在块No. 0的R元件的范围内。数据将被写入R0~R1。

(2) 传送目标超出了块No. 0的R元件的范围。由于超出块No. 0的R元件的区域范围，因此发生错误。

全局(锁存)标签、局部(锁存)标签的范围检查

通过指令指定了全局(锁存)标签、局部(锁存)标签的情况下,要进行范围的检查,因此应创建其运算结果不会超出相应标签的范围的程序。此外,位数指定或位指定时也应应在该标签的范围内指定。

指定了超出该软元件范围的情况下,将发生错误(错误代码:2820H)。

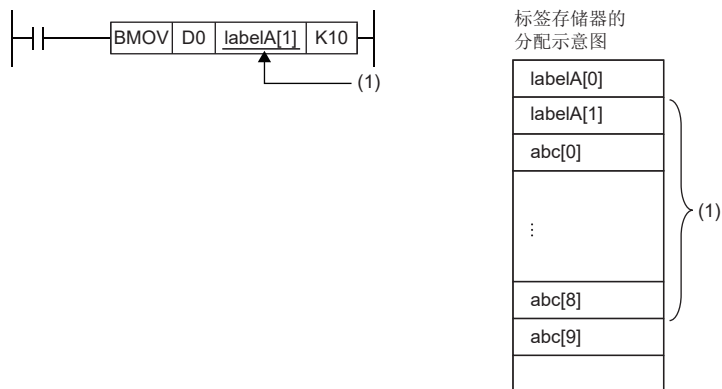
■数组型标签

参数中指定了数组型标签时,将进行数组型标签全体检查。此外,即使指定了数组型标签的要素,也要进行数组型标签范围内检查。

例

使用下述数组设置标签的情况下

标签名	数据类型	数组要素数
labelA	位串[16位]	2
abc	位串[16位]	10



(1) 传送目标范围为从labelA[1]开始的10点标签。labelA[2]~labelA[9]不存在,因此将变为错误。

■结构体型标签

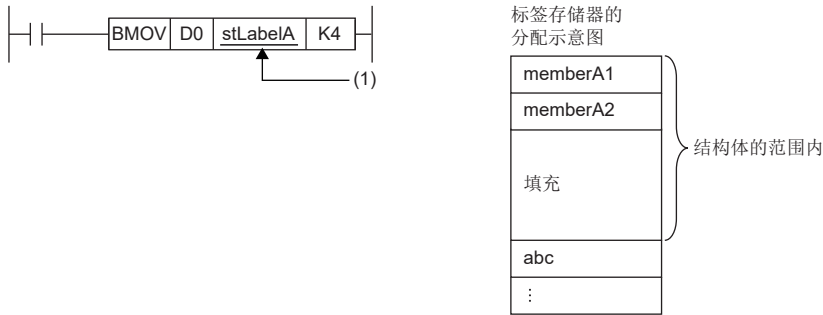
参数中指定了结构体型标签时，将进行结构体型标签全体检查。此外，指定结构体构件时，要对该构件的数据类型相应的范围内进行检查。

例

使用下述数组设置标签的情况下

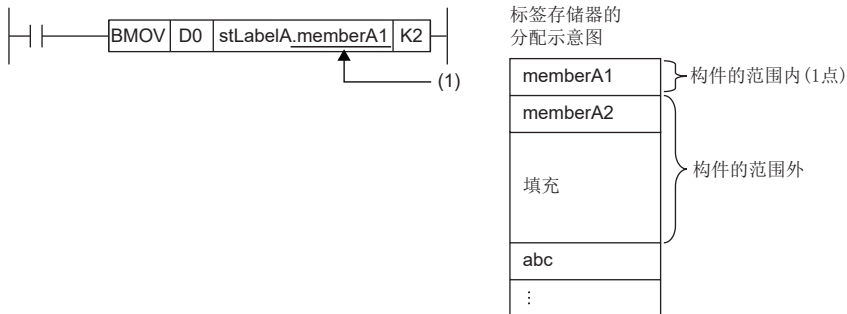
标签名	数据类型		构件
stLabelA	结构体A	字/位串[16位]	memberA1
		字/位串[16位]	memberA2
abc	字/位串[16位]		—

- 指定了结构体型标签的情况下



(1) 传送目标为从结构体型标签“stLabelA”的起始开始到4点的范围。因为要访问包括结构体的填充2点的范围，数据将被写入。

- 指定了构件“memberA1”的情况下



(1) 传送目标范围为memberA1的起始开始到2点的范围。因memberA1的范围为1点，变为错误。

模块标签、网络标签的范围检查

通过指令指定了模块标签、网络标签的情况下，要在网络标签/模块标签分配区域进行范围检查，因此应创建其运算结果不会超出网络标签/模块标签分配区域的范围的程序。

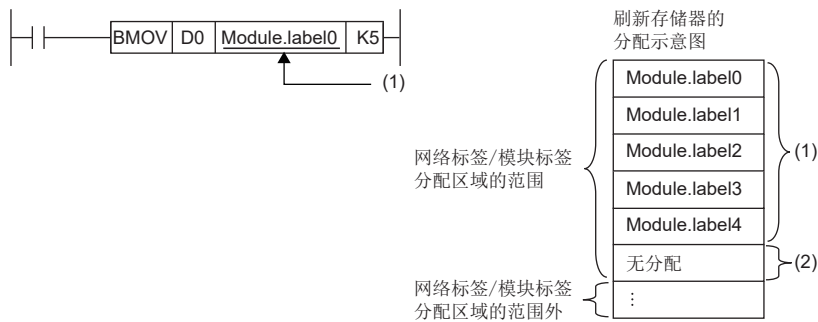
指定了超出该网络标签/模块标签分配区域的情况下，将发生错误(错误代码：2820H)。

网络标签/模块标签分配区域的设置方法请确认所使用的控制器的用户手册。

例

使用下述项目设置模块标签的情况下

实例名	标签名	数据类型	分配软元件
Module	label10	字[无符号]	UV:0
	label11	字[无符号]	UV:1
	label12	字[无符号]	UV:2
	label13	字[无符号]	UV:3
	label14	字[无符号]	UV:4



- (1) 传送目标在网络标签/模块标签分配区域的范围内。写入到Module.label0~Module.label4中。此外，即使数据类型不同，也将被写入。
- (2) 在传送目标的范围外，但在网络标签/模块标签分配区域的范围内，因此可访问此区域。

使用长定时器、长累计定时器软元件时的动作

长定时器、长累计定时器，在处理数据超出当前值的宽度(32位)的情况下，不仅是当前值的区域，还将使用上次值、触点·线圈的区域执行动作。

软元件	构成
定时器(T)	
累计定时器(ST)	
计数器(C)	
长定时器(LT)	
长累计定时器(LST)	
长计数器(LC)	

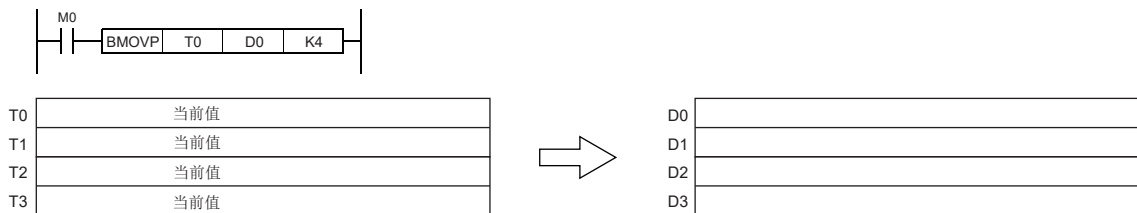
当前值的批量传送中使用BMOV指令的情况下，仅当前值无法进行批量传送，因此对当前值、触点、线圈进行批量传送后，传送后的数据中只应使用当前值。

当前值的批量传送中使用DMOV指令的情况下，仅将当前值的传送通过FOR~NEXT指令重复进行，应传送当前值。

例

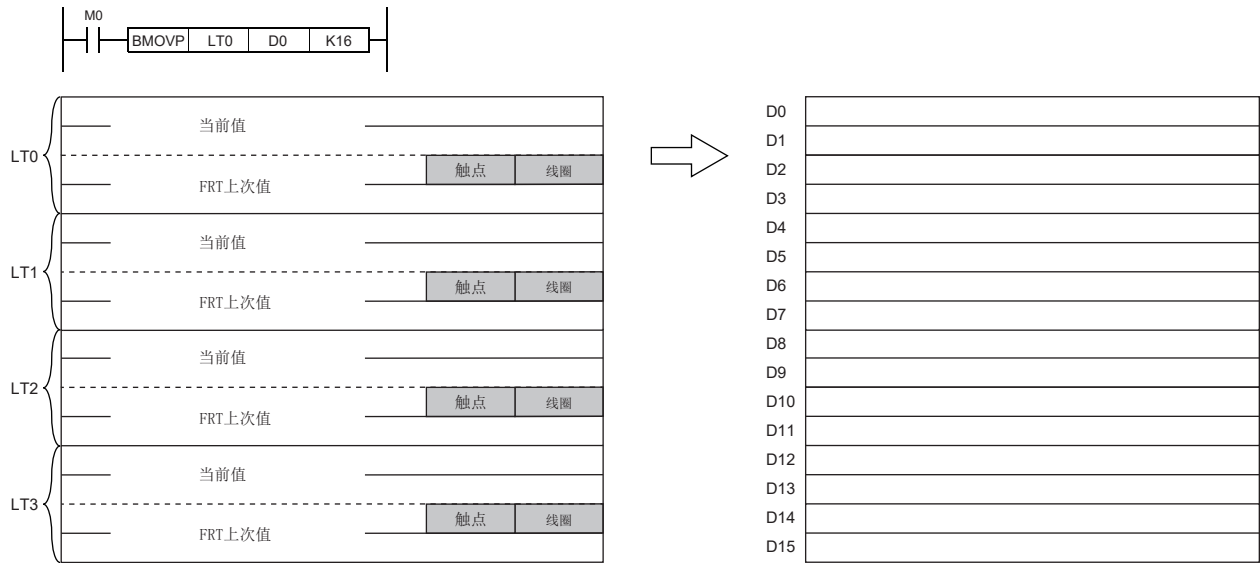
对定时器软元件的当前值进行批量传送的情况下

当前值的批量传送中使用BMOV指令的情况下仅批量传送当前值。

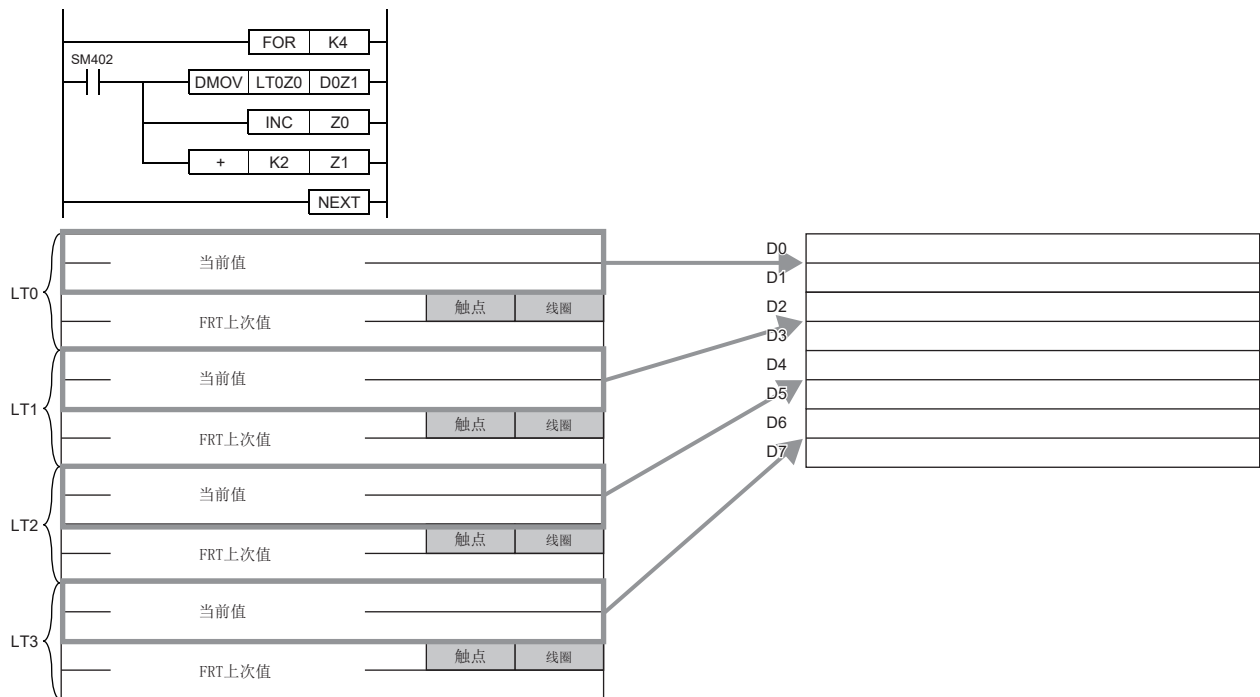


例

对长定时器软元件的当前值进行批量传送的情况下
 当前值的批量传送中使用BMOV指令的情况下批量传送当前值、触点、线圈。



当前值的批量传送中使用DMOV指令的情况下仅批量传送当前值。



将分配到同步站的链接软元件用于指令参数时的注意事项

将分配到CC-Link IE TSN同步站的链接软元件用于指令参数时，应设置链接刷新后，指定链接刷新目标的软元件或者标签。指令参数中指定了直接访问软元件(U□\G□、J□\□)时，链接软元件无法正确运行。

关于CC-Link-IE TSN的详细情况请确认所使用的控制器的用户手册。

使用同一软元件的OUT指令、SET/RST指令、PLS/PLF指令时的动作

以下介绍将使用了同一软元件的OUT指令、SET/RST指令、PLS/PLF指令在1个扫描中执行多次情况下的动作有关内容。

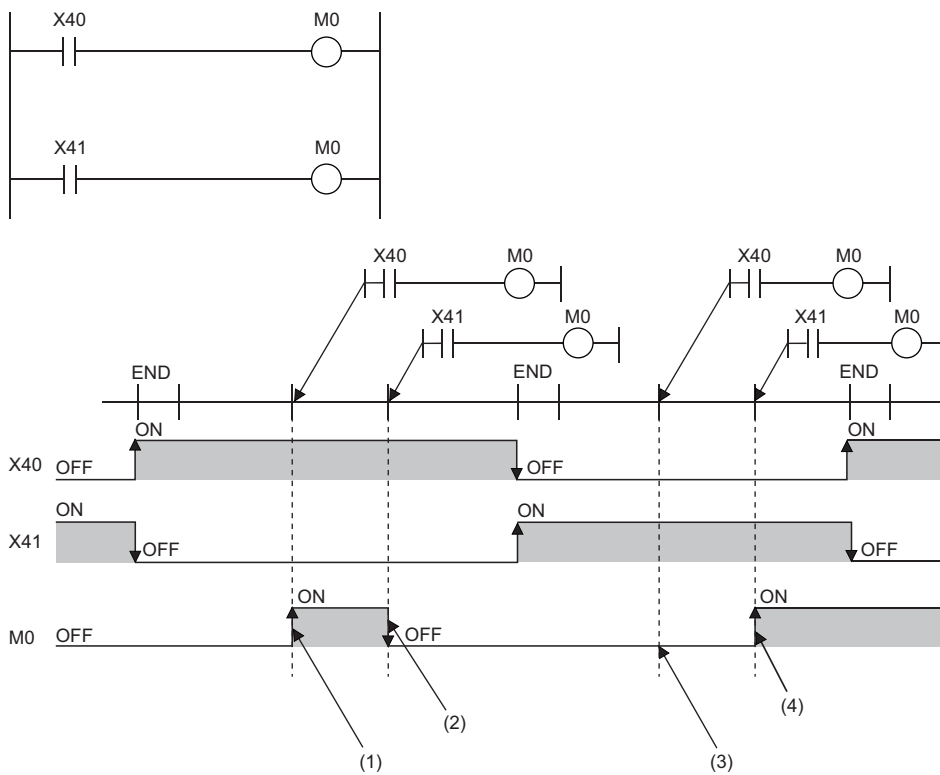
同一软元件的OUT指令的情况下

1个扫描中请勿多次执行同一软元件的OUT指令。

1个扫描中多次执行了同一软元件的OUT指令的情况下，执行各个OUT指令时，根据OUT指令为止的运算结果，指定软元件将变为ON/OFF。

执行各个OUT指令时，指定软元件的ON/OFF是确定的，因此1个扫描中有可能重复进行ON/OFF。

输入的X40及X41中，创建了将同一内部继电器(M0)置为ON/OFF的梯形图情况下的动作如下所示。



- (1) X40为ON，因此M0变为ON。
- (2) X41为OFF，因此M0变为OFF。
- (3) X40为OFF，因此M0保持OFF不变。
- (4) X41为ON，因此M0变为ON。

通过OUT指令指定输出(Y)时，1个扫描的最后执行的OUT指令的ON/OFF状态将被输出。

使用了同一软元件的SET/RST指令的情况下

■SET指令的情况下

SET指令在执行指令为ON时将指定软元件置为ON，在执行指令为OFF时无处理。

因此将同一软元件的SET指令在1个扫描中执行了多次的情况下，如果1个执行指令为ON，指定软元件将变为ON。

■RST指令的情况下

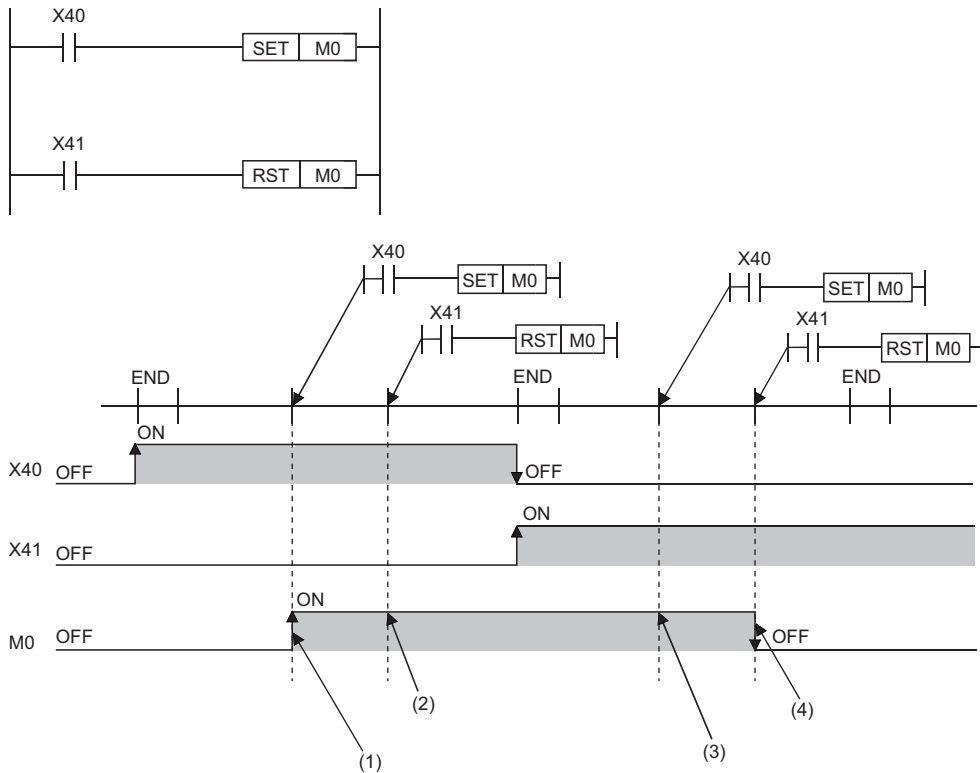
RST指令在执行指令为ON时将指定软元件置为OFF，在执行指令为OFF时无处理。

因此将同一软元件的RST指令在1个扫描中执行了多次的情况下，如果1个执行指令为ON，指定软元件将变为OFF。

■1个扫描中存在有同一软元件的SET指令及RST指令的情况下

1个扫描中存在有同一软元件的SET指令及RST指令的情况下，SET指令在执行指令为ON时将指定软元件置为ON，RST指令在执行指令为ON时将指定软元件置为OFF。

SET指令及RST指令的执行指令为OFF的情况下，指定软元件的ON/OFF状态不变化。



- (1) X40为ON，因此M0变为ON。
- (2) X41为OFF，因此M0保持ON不变。(RST指令将变为无处理。)
- (3) X40为OFF，因此M0保持ON不变。(SET指令将变为无处理。)
- (4) X41为ON，因此M0变为OFF。

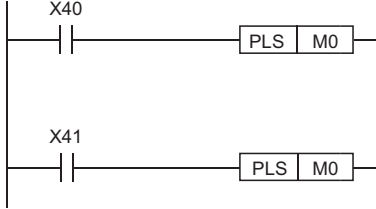
通过SET/RST指令指定输出(Y)时，1个扫描的最后执行的SET/RST指令的ON/OFF状态将被输出。

使用了同一软元件的PLS指令的情况下

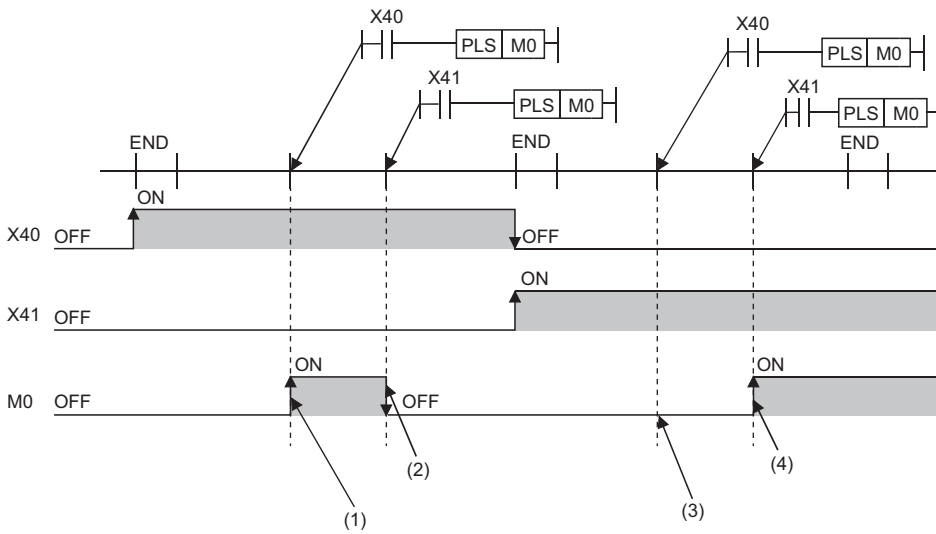
PLS指令在执行指令的OFF→ON时将指定软元件置为ON。OFF→ON以外(OFF→OFF、ON→ON、ON→OFF)时，将指定软元件置为OFF。

在1个扫描中多次执行了同一软元件的PLS指令的情况下，在各PLS指令的执行指令的OFF→ON时，将指定软元件置为ON。OFF→ON以外时将指定软元件置为OFF。

因此，在1个扫描中多次执行了同一软元件的PLS指令的情况下，通过PLS指令置为ON的软元件有可能不执行1个扫描ON。

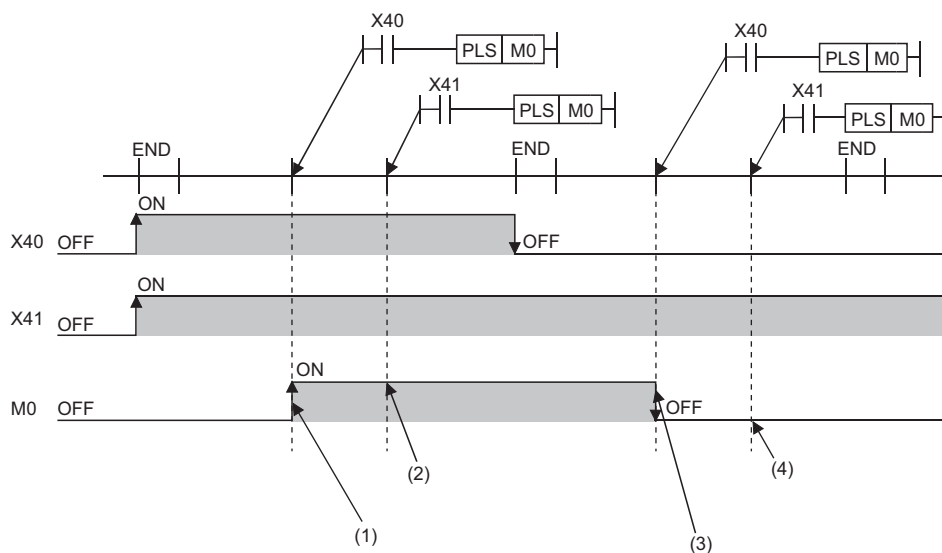


- X40与X41的ON/OFF时机不相同(指定软元件1个扫描不ON。)



- (1) X40为OFF→ON，因此M0变为ON。
- (2) X41为OFF→ON以外，因此M0变为OFF。
- (3) X40为OFF→ON以外，因此M0保持OFF不变。
- (4) X41为OFF→ON，因此M0变为ON。

• X40及X41的OFF→ON为同一时机时



- (1) X40为OFF→ON，因此M0变为ON。
- (2) X41为OFF→ON，因此M0保持OFF不变。
- (3) X40为OFF→ON以外，因此M0变为OFF。
- (4) X41为OFF→ON以外，因此M0保持OFF不变。

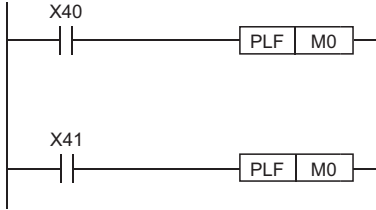
通过PLS指令指定输出(Y)时，1个扫描的最后执行的PLS指令的ON/OFF状态将被输出。

使用了同一软元件的PLF指令的情况下

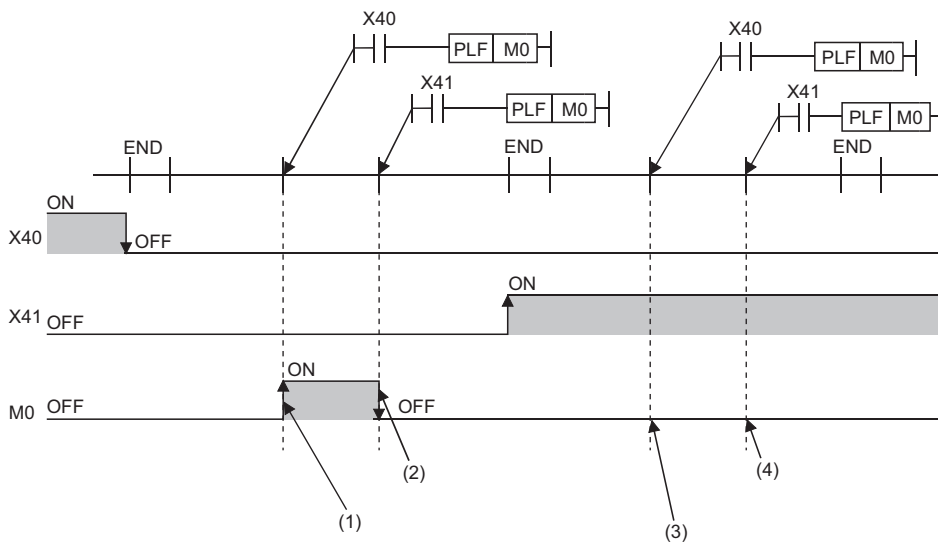
PLF指令在执行指令的ON→OFF时将指定软元件置为ON。ON→OFF以外(OFF→OFF、OFF→ON、ON→ON)时，将指定软元件置为OFF。

在1个扫描中多次执行了同一软元件的PLF指令的情况下，各PLF指令的执行指令为ON→OFF时，将指定软元件置为ON。ON→OFF以外时将指定软元件置为OFF。

因此，在1个扫描中多次执行了同一软元件的PLF指令的情况下，通过PLF指令置为ON的软元件有可能不执行1个扫描ON。

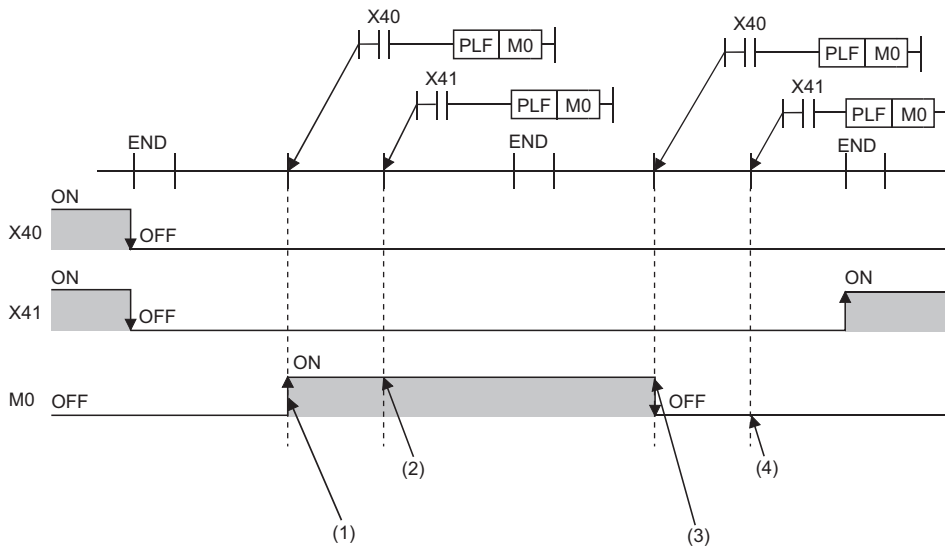


- X40与X41的ON/OFF时机不相同(指定软元件1个扫描不ON。)



- (1) X40为ON→OFF，因此M0变为ON。
- (2) X41为ON→OFF以外，因此M0变为OFF。
- (3) X40为ON→OFF以外，因此M0保持OFF不变。
- (4) X41为ON→OFF以外，因此M0保持OFF不变。

• X40及X41的ON→OFF为同一时机时



- (1) X40为ON→OFF，因此M0变为ON。
- (2) X41为ON→OFF，因此M0保持ON不变。
- (3) X40为ON→OFF以外，因此M0变为OFF。
- (4) X41为ON→OFF以外，因此M0保持OFF不变。

通过PLF指令指定输出(Y)时，1个扫描最后执行的PLF指令的ON/OFF状态将被输出。

使用文件寄存器时的限制事项

刷新软件中指定了文件寄存器的情况下，应注意下述限制事项。

通过RSET指令切换了块No. 的情况下

通过RSET指令切换了块No. 的情况下，应注意以下事项。

- 将被刷新为切换后的块No. 的文件寄存器(R)。
- 将被刷新为刷新之前的块No. 的文件寄存器(R)。

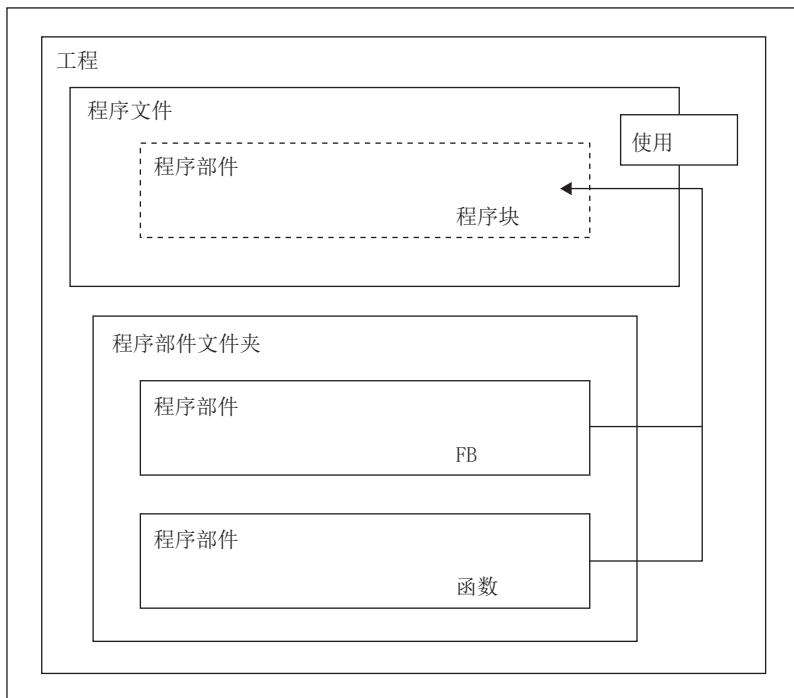
希望读写刷新数据的情况下，应指定为刷新之前的块No.。

2 程序部件

程序部件有下述几种类型。

- 程序块
- 函数
- 功能块

各程序部件可以通过符合控制的程序语言记述处理。在函数及FB中，可以通过梯形图语言、ST语言或FBD/LD语言记述处理。从程序块调用函数及FB后执行。



要点

程序的部件化是指按照各处理内容及功能，将程序阶层化时的低位处理分为若干单位，创建各单位的程序。通过将程序部件化提高独立性，可以设计出更容易添加及更换的程序。

对以下处理进行部件化较好。

- 在程序中重复被记述的处理
- 作为一个功能可被分开的处理

本项使用标签对各程序部件进行说明。

各程序部件的程序本体(工作表)中也可以使用软元件。关于软元件的详细内容，请参阅下述手册。

📖所使用的控制器的用户手册

要点

ST语言与FBD/LD语言中，一个程序部件内最多可以创建32个工作表。

多个工作表的执行顺序，从工程工具的“工作表执行顺序设置”画面设置。(📖GX Works3 操作手册)

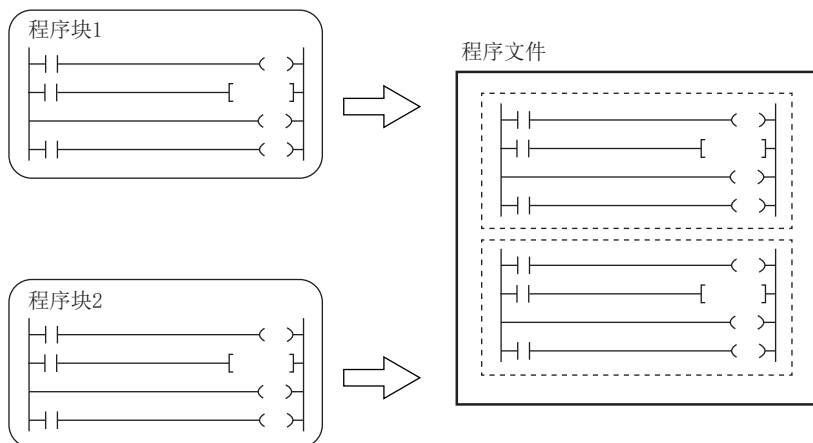
2.1 程序块

程序块为构成程序的单位。

在程序文件内可以创建多个程序块，并按照程序文件设置中指定的顺序执行。程序文件设置中未指定顺序的情况下，按照程序块名的顺序(升序)执行。

如果对各功能及处理划分程序块，可以让程序的顺序更改及更换变得更容易。

将程序块的程序本体存储到各登录目标程序中的程序文件中。



程序块的分割

可分别对各程序块创建主程序、子程序、中断程序。

类型	内容
主程序	是从程序的步0开始到FEND指令为止的程序。
子程序	是指针(P)开始到RET指令为止的程序。 只在通过子程序调用指令(CALL指令、ECALL指令等)调用的情况下执行。
中断程序	是从中断指针(I)开始到IRET指令为止的程序。 如果发生中断原因，执行与该中断指针编号相对应的中断程序。

要点

- 子程序以及中断程序是在FEND指令以后进行创建。FEND指令及其以后的程序不作为主程序执行。例如，在第二个程序块的最后使用了FEND指令的情况下，第三个程序块以后将变为子程序或中断程序。
- 为了创建易懂的程序，应在一个程序块中使用成对的FOR指令与NEXT指令、MC指令与MCR指令。
- 简单程序的情况下，在一个程序块内仅记述主程序便可在控制器中执行。

关于子程序、中断程序的详细内容，请参阅下述手册。

📖 所使用的控制器的用户手册

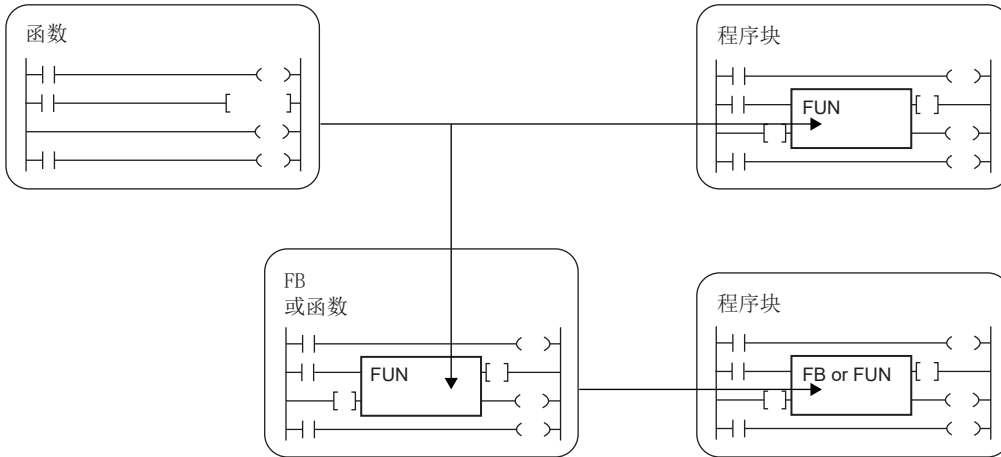
2.2 函数(FUN)

函数是在程序块、FB以及其它的功能中使用的程序部件。

函数执行完成后将值交接至调用源。该值称为返回值。

对于同样的输入，函数将作为处理结果始终输出相同的返回值。

如果预先定义经常使用的单纯独立的程序算法，可以有效地再利用。



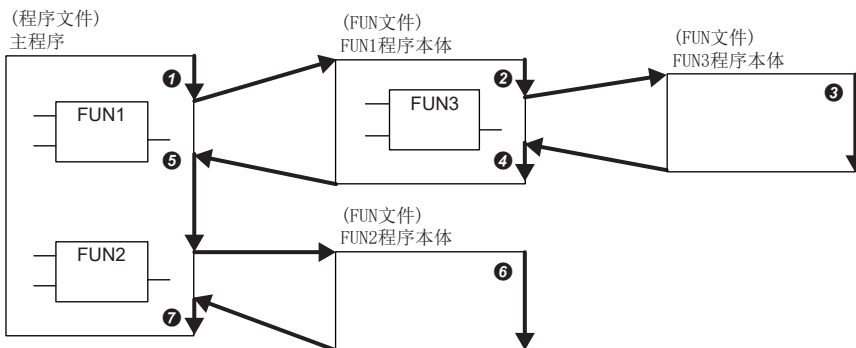
动作概要

将程序本体存储在FUN文件内，通过从调用源程序调用FUN文件内的程序本体来执行函数。

例

从主程序调用FUN1及FUN2，且FUN1又调用FUN3的情况(嵌套数：3次)

①~⑦表示执行的流程(顺序)。

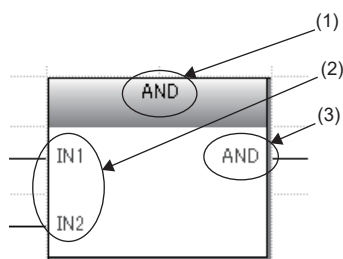


子程序型FB、函数全部合计可进行32次嵌套。

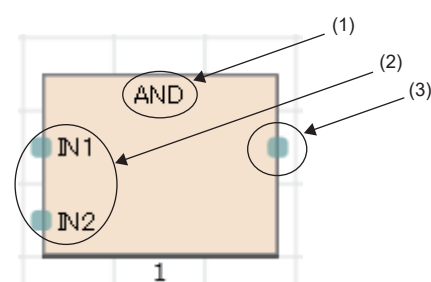
输入变量与输出变量

函数中可以定义输入变量与输出变量。输出变量可以分配与返回值不同的其它输出数据。

梯形图语言的情况



FBD/LD语言的情况



- (1) 函数名
- (2) 输入变量
- (3) 输出变量

不显示函数的返回值名称。

在VAR_INPUT的分类中设置输入变量，在VAR_OUTPUT的分类中设置输出变量。

要点

在函数中定义的变量于每次调用函数时被覆盖。

每次调用时希望保持变量值的情况下，应通过使用FB或将输出变量保存为不同的变量等进行编程。

EN/ENO

通过在函数中附加EN(允许输入)、ENO(允许输出)，可以控制执行处理。

- 在EN中设置作为函数执行条件的布尔型变量。
- 带EN的函数只在EN的执行条件为TRUE的情况下执行。
- 在ENO中设置输出函数执行结果的布尔型变量。

EN状态的ENO与运算结果的内容如下所示。


EN	ENO	运算结果
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE	不定值

要点


- 在梯形图语言、FBD/LD语言的程序中，不需要设置至ENO的输出标签。
- 在通用函数中使用EN/ENO的情况下，带EN的函数将变为“函数名_E”。

创建程序

创建函数程序的情况下，执行下述操作。

 [导航窗口]⇒[FB/FUN]⇒右击⇒[新建数据]
在“基本设置”的“数据类型”中选择“函数”

创建的程序存储在FUN文件中。

 [CPU参数]⇒[程序设置]⇒[FB/FUN文件设置]

1个FUN文件中最多可以存储64个创建的程序。

无法在函数内使用上升沿/下降沿执行指令。

创建程序有关内容，请参阅下述手册。

项目	参照
函数的创建方法	 GX Works3 操作手册
可写入至控制器的FB/FUN文件数	 所使用的控制器的用户手册

■可使用的软元件/标签

函数程序中可使用的软元件及标签一览如下所示。

○：可以使用，△：只可在指令中使用(作为表示程序的步的标签时，禁止使用)，×：禁止使用

软元件/标签的类型		能否使用
标签(指针型以外)	全局标签	×
	局部标签	○*1
标签(指针型)	指针型全局标签	×
	指针型局部标签	○
软元件	全局软元件	○
	局部软元件	×
指针	全局指针	△
	局部指针	×

*1 不能使用下述数据类型。

定时器、累计定时器、计数器、长定时器、长累计定时器、长计数器

要点

- 函数的返回值，可以通过在函数内将函数名作为标签编程进行设置。函数名不需要作为标签进行设置。在函数的属性中，可以使用“返回值的类型”中设置的数据类型。

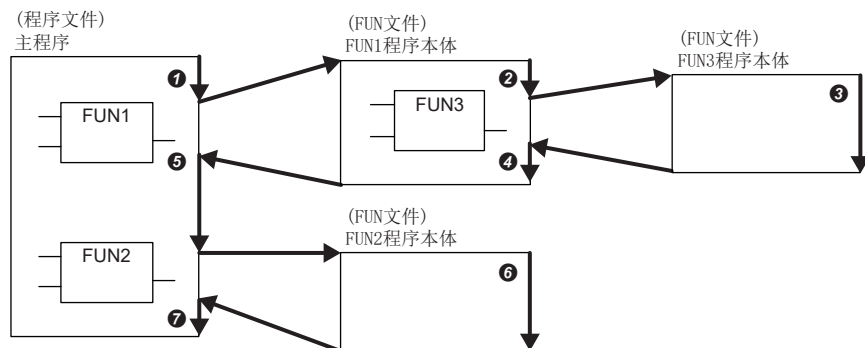
函数中定义的标签

对于在函数中定义的标签分配目标，在执行函数时将确保到存储器内的临时区域(临时工作区)中，并在执行完成时解除。

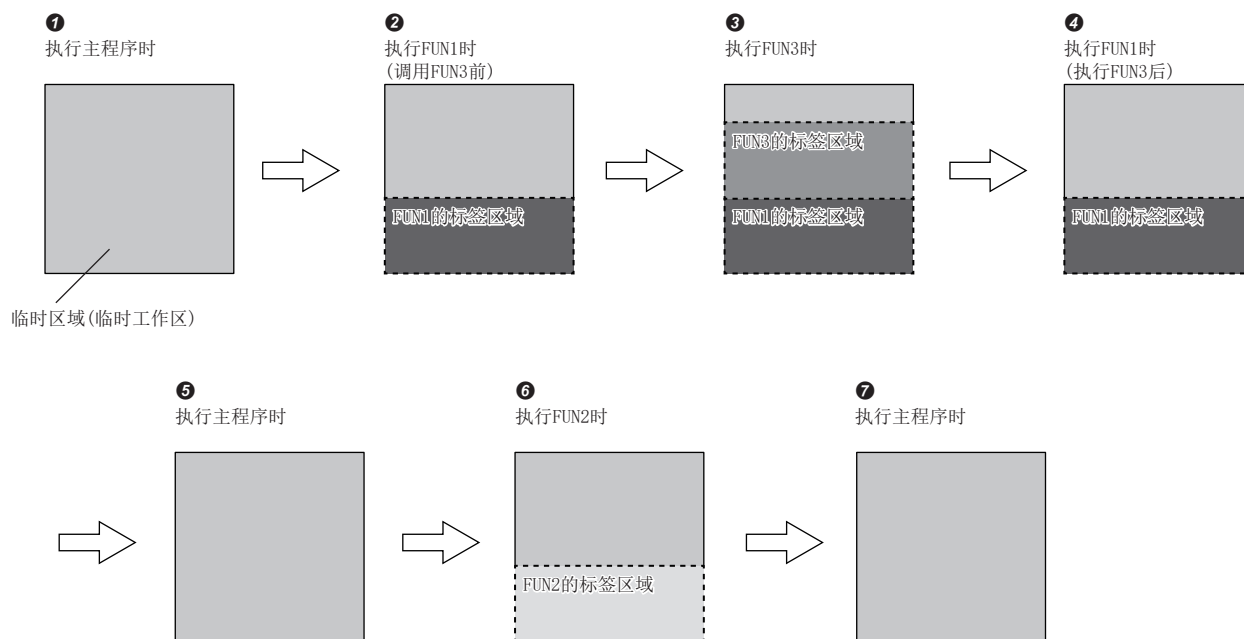
例

从主程序调用FUN1及FUN2，且FUN1又调用FUN3的情况

(①~⑦表示执行的流程(顺序))



对于上述函数执行动作的标签分配状态如下所示。



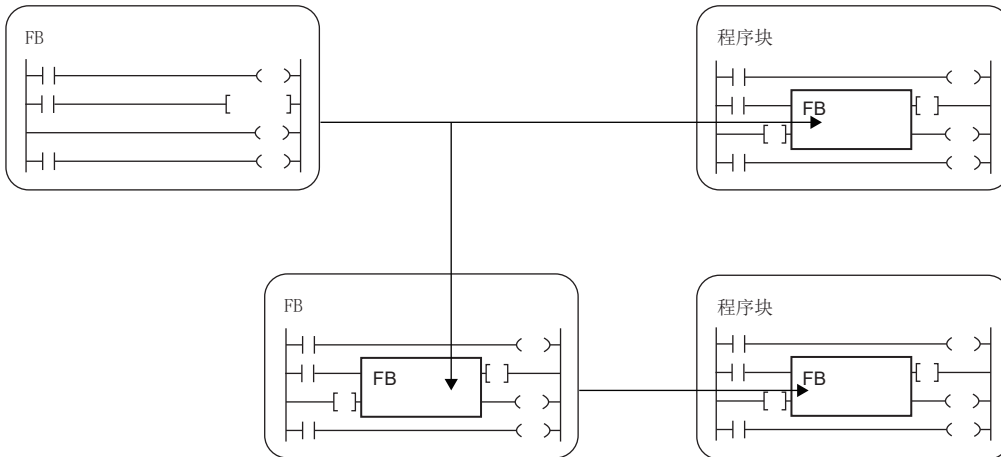
函数中可定义的标签的分类为VAR、VAR_CONSTANT、VAR_INPUT、VAR_OUTPUT。

要点

由于在函数中定义的标签变为不定值，最初访问时需要通过程序进行初始化。

2.3 功能块(FB)

FB是在程序块及其它的FB中使用的程序部件。



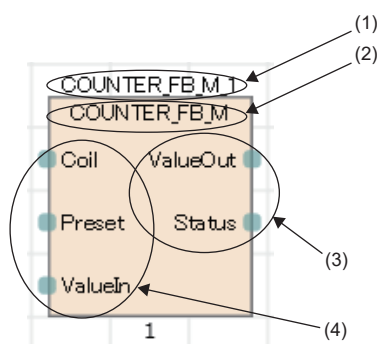
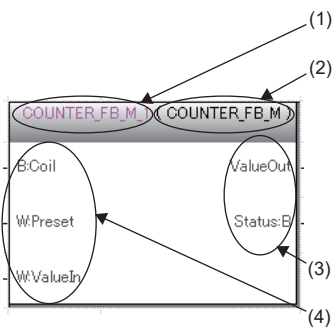
FB与函数不同，不能保持返回值。

FB能将值保存在变量中，因此也能保持输入状态及处理结果。

在下次处理中使用保持后的值，因此即使为相同的输入值也不一定每次都输出相同的结果。

梯形图语言的情况

FBD/LD语言的情况



- (1) 实例名
- (2) FB名
- (3) 输出变量
- (4) 输入变量

此外，为了在程序上使用FB，需要定义实例。

☞ 67页 实例

要点

关于模块FB的详细内容，请参阅下述手册。

📖 所使用的模块的FB的参考

动作概要

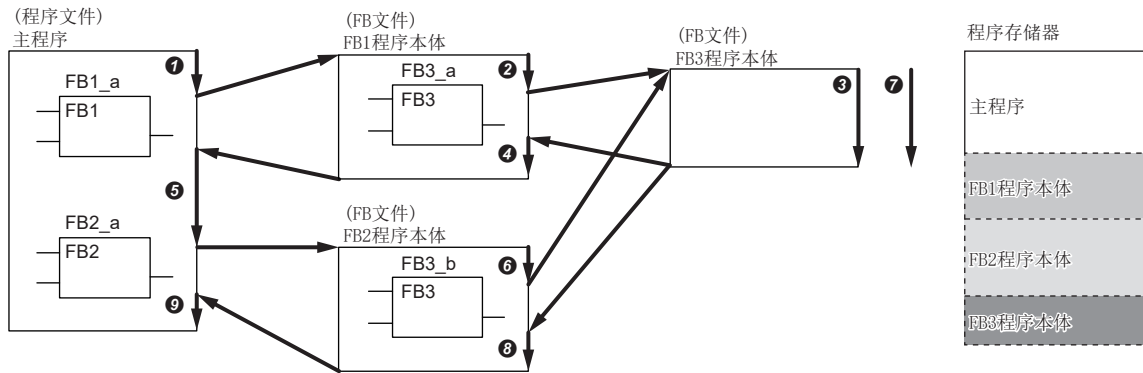
■子程序型FB

将程序本体存储在FB文件中，执行时从调用源程序调用FB文件内的程序本体后再执行子程序型FB。

例

从主程序调用FB1_a及FB2_a，且又从FB1_a调用FB3_a，从FB2_a调用FB3_b的情况(嵌套数：3次)

①~⑨表示执行的流程(顺序)。

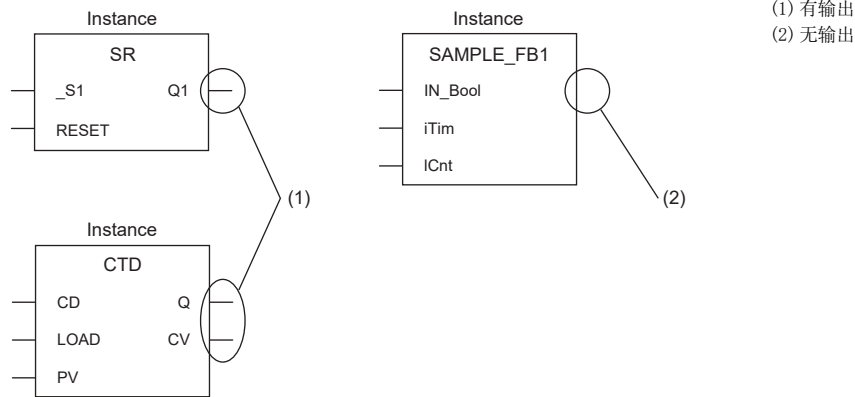


子程序型FB、函数全部合计可进行32次嵌套。

输入变量、输出变量、输入输出变量

FB中需要定义输入变量、输出变量、输入输出变量。

FB可以输出多个运算结果。也可以不使其输出。

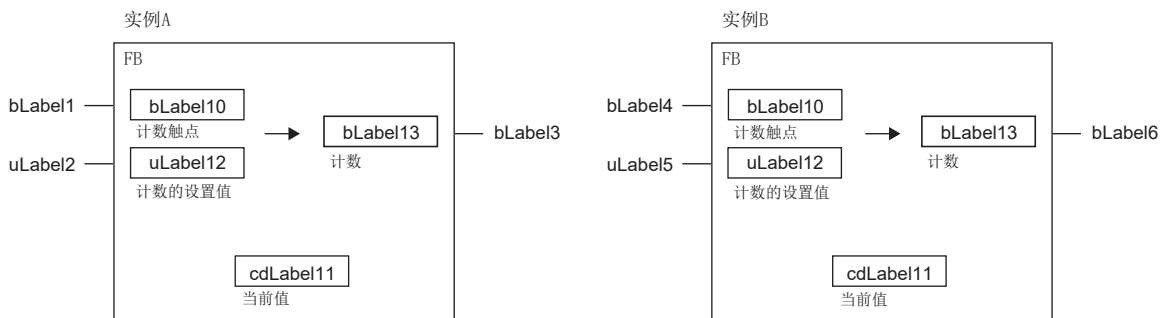


在VAR_INPUT的分类中设置输入变量，在VAR_OUTPUT及VAR_OUTPUT_RETAIN的分类中设置输出变量，在VAR_IN_OUT的分类中设置输入输出变量。

内部变量

FB使用内部变量。内部变量按FB的实例将标签分配到不同的区域中。即使是同样的标签名，各实例可保持不同的状态。

例



输入变量变为ON时开始计数，当内部变量中保持的当前值达到设定值时，将输出变量置为ON的FB。即使是同一个FB，实例A与实例B保持着各自独立的状态，因此输出的时机有所不同。

在VAR、VAR_CONSTANT、VAR_RETAIN的分类中设置内部变量。

外部变量及公开变量

FB可以使用外部变量(全局标签)及公开变量。

在VAR_PUBLIC、VAR_PUBLIC_RETAIN的分类中设置公开变量。

实例

■实例含义

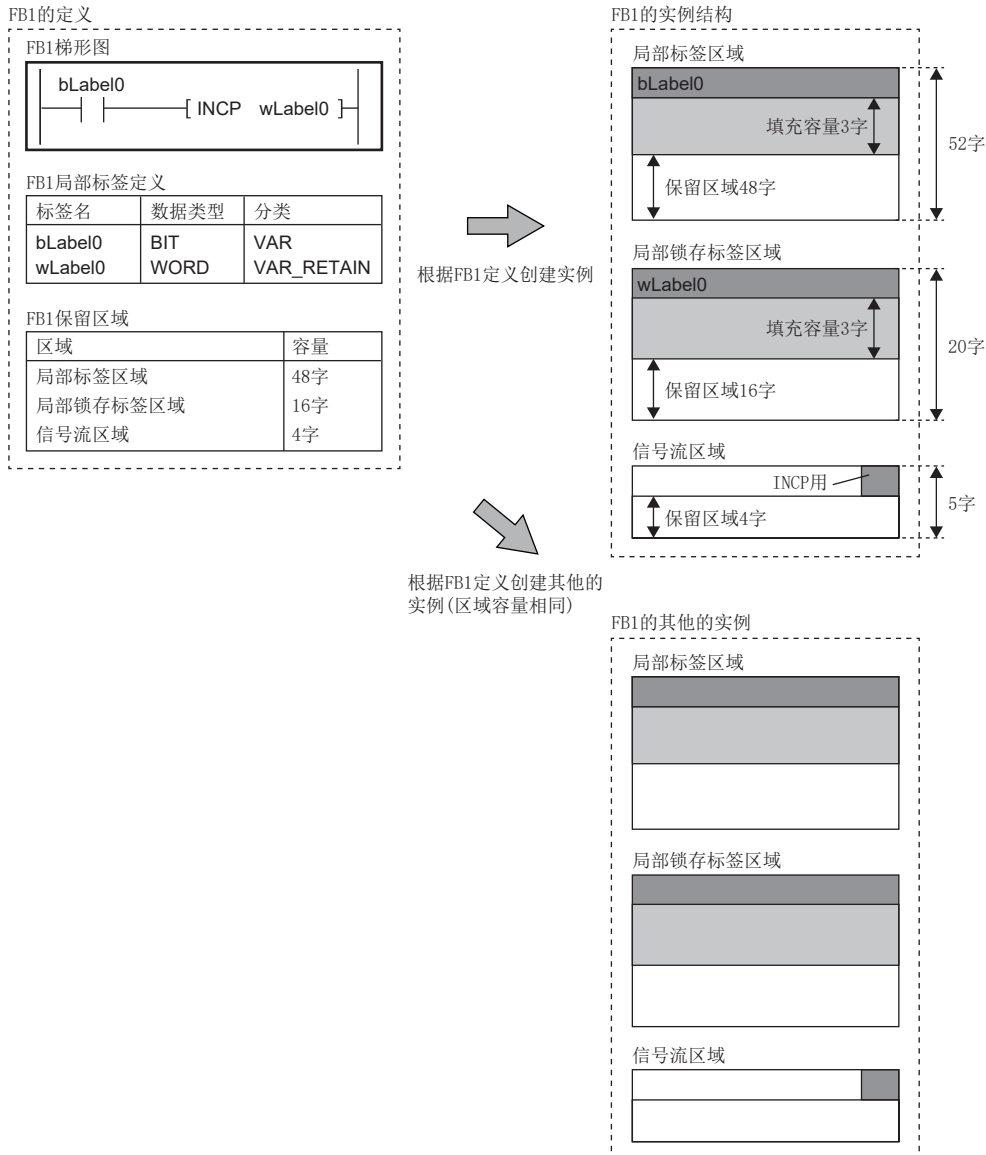
FB的实例是指，在FB定义的基础上分配的标签。可以从一个FB定义创建多个实例。

实例的配置如下。

项目	内容
局部标签区域	分配FB的局部标签的区域。
局部锁存标签区域	对FB的锁存属性的局部标签进行分配的区域。
信号流区域	对FB定义内的指令所使用的信号流进行分配的区域。

例

实例的配置



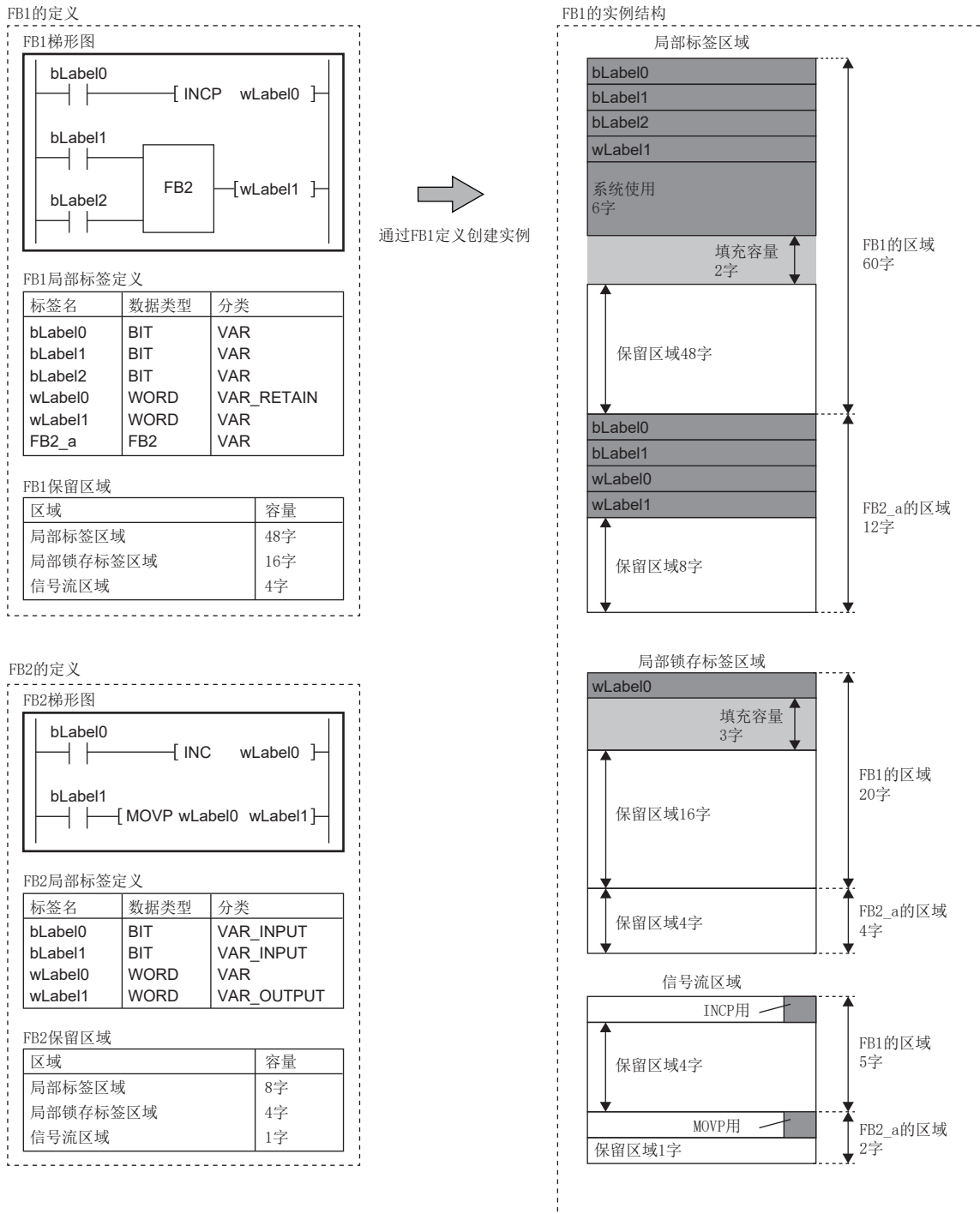
局部标签区域及局部锁存标签区域，确保4字单位的标签使用区域，因此上述的示例已确保3字(填充容量)。

各区域确保保留区域。保留区域是指在转换或RUN中写入等时维持标签的分配不变，用于对参照局部标签或信号流存储器的指令、FB的实例进行添加/更改的区域。

无法确保与添加/更改对象的数据类型相应的区域时，需要全部转换或转换+RUN中写入。

例

对FB进行了嵌套的实例的配置(已变更FB2的保留区域的情况)



如上所示，FB嵌套时，在声明源的实例区域外将确保声明为局部标签的FB的实例。此外，声明源的局部标签区域将确保系统中所使用的区域(6字)。

在上图的FB1中添加或更改FB型的局部标签时，FB1的局部标签区域中需要系统使用部分的容量(6字)，作为FB1声明源的程序的实例区域中需要FB2的实例区域部分的容量(对于局部标签区域、局部锁存区域、信号流区域的各容量)。

希望在无需全部转换且保持分配不变的状态下添加/更改局部标签及FB实例时，应预先确保保留区域中将来添加/更改的容量，或执行转换+RUN中写入。关于保留区域的设置方法，请参阅下述手册。

📖 GX Works3 操作手册

■创建实例

为了使用FB，需要创建实例。

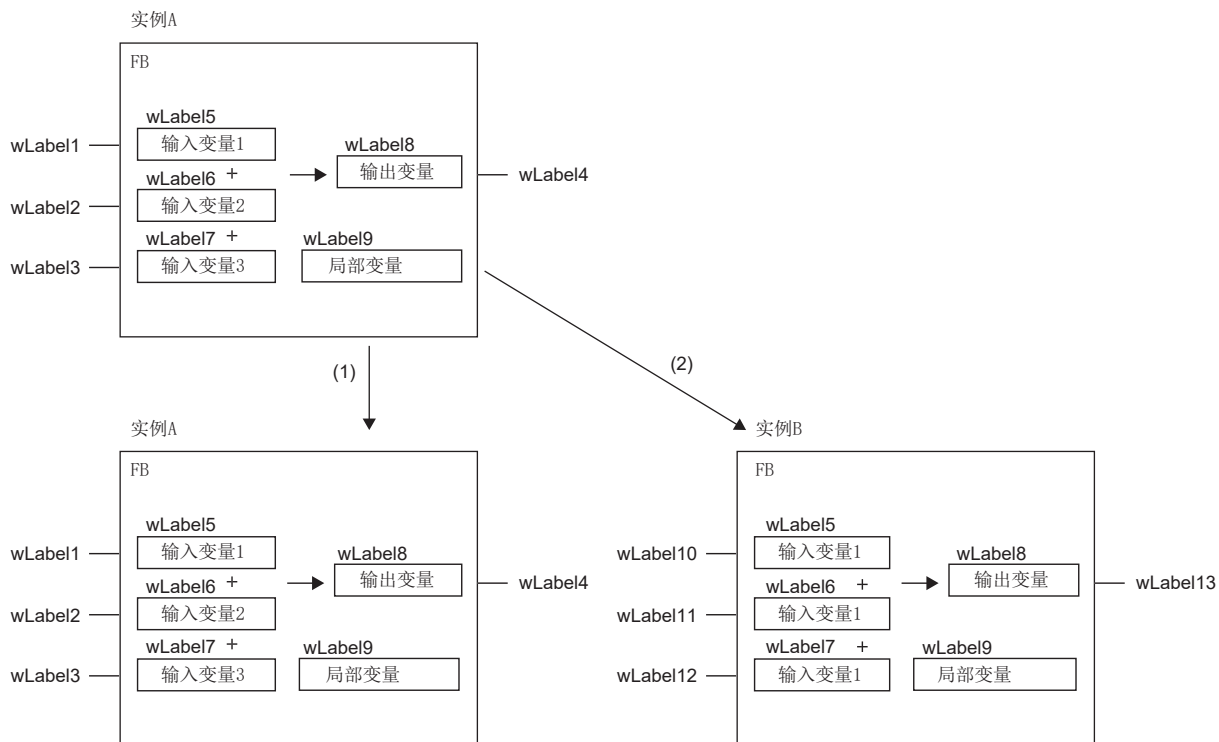
通过创建FB的实例，可以从程序块及其他的FB调用使用。

在全局标签或局部标签中声明实例。

标签的类型	实例的类型	分类
全局标签	全局FB	VAR_GLOBAL
局部标签*1	局部FB	VAR

*1 可以作为程序块或FB的局部标签进行声明。在函数中无法声明。

在一个程序部件中，同一个FB可以在不同的实例中使用。



(1) 相同实例的情况下，使用相同的内部变量。

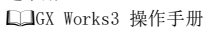
(2) 不同实例的情况下，使用不同区域的内部变量。

■实例的容量

关于实例的各数据区域的容量，计算方法如下所示。


- 局部标签区域的容量

实例的局部标签区域的容量 = 锁存属性以外的局部标签的数据容量(总和) + 保留区域容量

项目	内容
局部标签容量(锁存属性的局部标签除外)	作为局部标签使用的数据的总和。 实际使用的区域的容量，根据标签的存储器分配而有所不同。标签的存储器分配的详细内容，请参阅下述手册。  GX Works3 操作手册
保留区域容量	默认为48字。可以以4字单位设置。

- 局部锁存标签区域的容量

实例的局部锁存标签区域的容量 = 锁存属性的局部标签的数据容量(总和) + 保留区域容量

项目	内容
锁存属性局部标签容量	作为锁存属性的局部标签使用的数据的总和。 实际使用的区域的容量，根据标签的存储器分配而有所不同。标签的存储器分配的详细内容，请参阅下述手册。  GX Works3 操作手册
保留区域容量	默认为16字。可以以4字单位设置。

- 信号流区域的容量

实例的信号流区域的容量=FB定义内的指令所使用的信号流区域的总和+保留区域容量

项目	内容
信号流区域容量	FB定义内的指令所使用的信号流区域的总和。
保留区域容量	默认为4字。可以以1字单位设置。

要点

应在保留区域容量中设置RUN中写入预计会添加/更改的局部标签及信号流存储器的容量。关于设置方法，请参阅下述手册。

 GX Works3 操作手册

在FB定义的局部标签中包含部分结构体标签的情况下，转换+RUN中写入中无法添加/更改超出预期区域容量，需要全部转换。详细情况请确认所使用的控制器的用户手册。

对于下述的FB，通过将保留区域设置为比默认值还小的值，可以更有效率地使用控制器的存储器。

- 不进行局部标签的添加/更改或程序更改的已调试完成的子程序型FB
- 大量声明实例的子程序型FB

初始值的设置

■FB的局部标签的初始值

FB的局部标签可以设置FB定义及各实例的初始值。

可设置初始值的局部标签的分类为VAR、VAR_RETAIN、VAR_INPUT、VAR_OUTPUT、VAR_OUTPUT_RETAIN、VAR_PUBLIC、VAR_PUBLIC_RETAIN。

■实例的初始值

实例的初始值的类型如下所示。

类型	内容																												
默认初始值	<p>是各数据类型决定的初始值。FB的局部标签中未设置初始值的情况下，默认初始值也适用。</p> <p>全局标签定义</p> <table border="1"> <thead> <tr> <th>标签名</th> <th>初始值</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> <p>FB1的局部标签定义</p> <table border="1"> <thead> <tr> <th>标签名</th> <th>初始值</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> <p>程序块1的局部标签定义</p> <table border="1"> <thead> <tr> <th>标签名</th> <th>初始值</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> <p>(1) FB1定义的局部标签中未设置初始值。 (2) 默认初始值也适用。</p>	标签名	初始值	FB1_a	-	-- AAA	0	-- BBB	0	-- CCC	0	标签名	初始值	AAA		BBB		CCC		标签名	初始值	FB1_b	-	-- AAA	0	-- BBB	0	-- CCC	0
标签名	初始值																												
FB1_a	-																												
-- AAA	0																												
-- BBB	0																												
-- CCC	0																												
标签名	初始值																												
AAA																													
BBB																													
CCC																													
标签名	初始值																												
FB1_b	-																												
-- AAA	0																												
-- BBB	0																												
-- CCC	0																												
FB定义初始值	<p>是在定义FB的局部标签时设置的初始值。设置了FB定义初始值的情况下，对于所有的实例，相同的定义初始值也适用。</p> <p>全局标签定义</p> <table border="1"> <thead> <tr> <th>标签名</th> <th>初始值</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> <p>FB1的局部标签定义</p> <table border="1"> <thead> <tr> <th>标签名</th> <th>初始值</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> <p>程序块1的局部标签定义</p> <table border="1"> <thead> <tr> <th>标签名</th> <th>初始值</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> <p>(1) FB1定义的局部标签中已设有初始值。 (2) FB1的所有实例根据相同的定义初始值进行初始化。</p>	标签名	初始值	FB1_a	-	-- AAA	1111	-- BBB	2222	-- CCC	3333	标签名	初始值	AAA	1111	BBB	2222	CCC	3333	标签名	初始值	FB1_b	-	-- AAA	1111	-- BBB	2222	-- CCC	3333
标签名	初始值																												
FB1_a	-																												
-- AAA	1111																												
-- BBB	2222																												
-- CCC	3333																												
标签名	初始值																												
AAA	1111																												
BBB	2222																												
CCC	3333																												
标签名	初始值																												
FB1_b	-																												
-- AAA	1111																												
-- BBB	2222																												
-- CCC	3333																												
实例初始值	<p>是在包含全局标签及程序块的局部标签定义的实例中所设置的初始值。</p> <p>全局标签定义</p> <table border="1"> <thead> <tr> <th>标签名</th> <th>初始值</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> <p>FB1的局部标签定义</p> <table border="1"> <thead> <tr> <th>标签名</th> <th>初始值</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> <p>程序块1的局部标签定义</p> <table border="1"> <thead> <tr> <th>标签名</th> <th>初始值</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> <p>(1) 可以在FB1定义的各实例中设置初始值。</p>	标签名	初始值	FB1_a	-	-- AAA	3333	-- BBB	4444	-- CCC	5555	标签名	初始值	AAA	1111	BBB	2222	CCC	3333	标签名	初始值	FB1_b	-	-- AAA	7777	-- BBB	8888	-- CCC	9999
标签名	初始值																												
FB1_a	-																												
-- AAA	3333																												
-- BBB	4444																												
-- CCC	5555																												
标签名	初始值																												
AAA	1111																												
BBB	2222																												
CCC	3333																												
标签名	初始值																												
FB1_b	-																												
-- AAA	7777																												
-- BBB	8888																												
-- CCC	9999																												

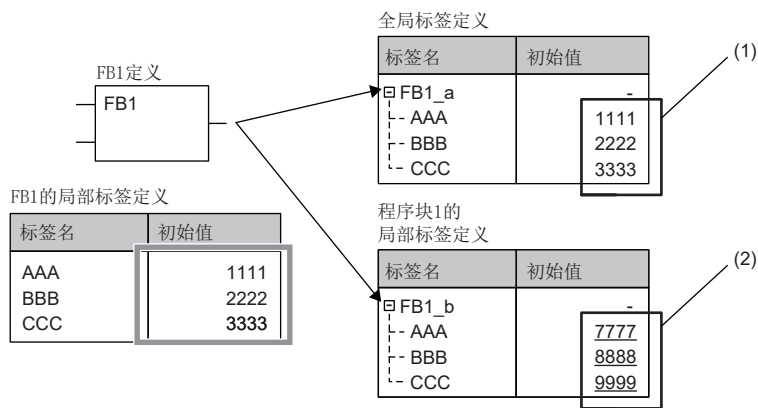
可以设置FB定义初始值与实例初始值两者的FB的初始值。

设置两个初始值的情况下，适用的初始值的优先顺序如下所述。

优先顺序	类型
高 ↑	实例初始值
↓	FB定义初始值
低	默认初始值

要点 🔍

创建两个设置有FB定义初始值的FB的实例，只对其中一个设置了实例初始值的情况下，FB定义初始值适用于未设置实例初始值的实例，实例初始值适用于设置了实例初始值的实例。

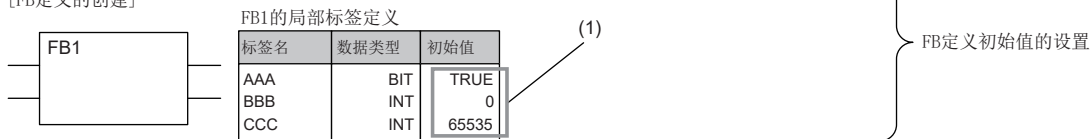


- (1) 未设置实例初始值的情况下，按照定义初始值进行初始化。
- (2) 设置了实例初始值的情况下，按照实例初始值进行初始化。

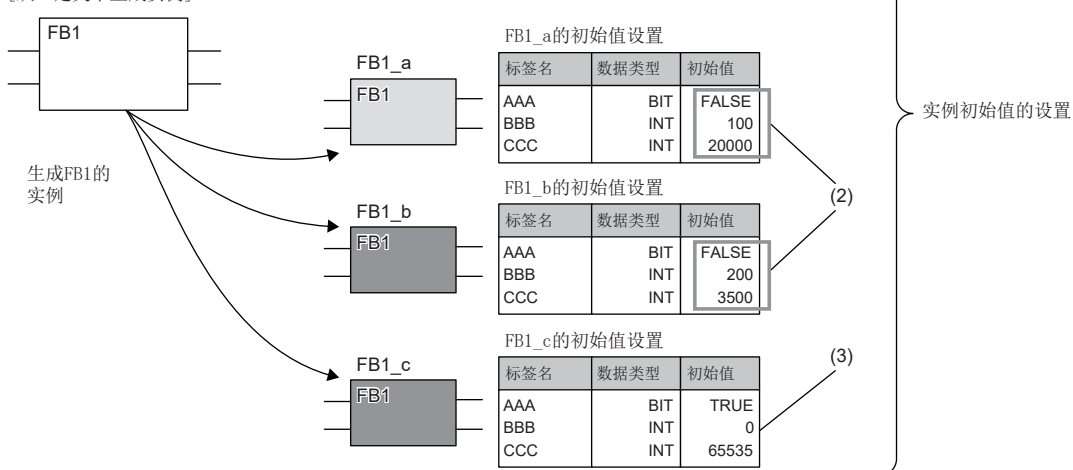
■使用示例

FB的初始值的使用示例如下所示。

[FB定义的创建]



[从FB定义中生成实例]



- (1) 在全部实例中设置通用的初始值。
- (2) 可以对各实例设置个别的初始值。
- (3) 未设置个别初始值的情况下，通用的初始值也适用。

EN/ENO

FB与函数一样通过附带EN(允许输入)、ENO(允许输出)，可以进行执行处理的控制。(☞ 61页 EN/ENO)


调用附带EN/ENO的FB的实例时，必须对EN分配实自变量。

要点 🔍


有关运动控制FB执行条件的详细内容，请参阅下述手册。
☞ 1375页 执行指令(Execute)类型/有效(Enable)类型

创建程序

创建FB的程序的情况下，实施下述操作。



 [导航窗口]⇒[FB/FUN]⇒右击⇒[新建数据]
在“基本设置”的“数据类型”中选择“FB”

创建的程序存储在FB文件中。

 [CPU参数]⇒[程序设置]⇒[FB/FUN文件设置]

1个FB文件中最多可以存储64个创建的程序。

创建程序有关内容，请参阅下述手册。


项目	参照
FB的创建方法	 GX Works3 操作手册
可写入至控制器的FB/FUN文件数	 所使用的控制器的用户手册

■程序的类型

FB有下述几种类型，FB程序本体的存储方式不同。


- 子程序型FB



详细内容，请参阅下述章节。

 65页 动作概要

模块FB、通用函数、通用FB、运动控制FB无法进行上述选择。

■固有属性设置

创建FB的程序的情况下，可以实施下述设置。（ GX Works3 操作手册）

项目	内容
使用MC/MCR控制EN	选择“是”时，使用MC/MCR指令控制EN；选择“否”时，使用CJ指令控制EN。在FB内使用了上升沿/下降沿时，应选择“是”。此外，根据选择，FB内所使用的定时器/计数器及OUT指令的动作将有所不同。详细内容，请参阅下述章节。  83页 使用MC/MCR指令控制EN的动作
使用EN/ENO	选择“是”时，变为附带EN/ENO的FB，即使EN/ENO标签未登录至局部标签，也能在程序中使用。选择“否”时，变为不附带EN/ENO的FB。 关于EN/ENO的详细内容，请参阅下述章节。  73页 EN/ENO

■可使用的软元件/标签

在FB程序中可使用的软元件及标签一览如下所示。

○：可以使用，△：只可在指令中使用（作为表示程序的步的标签时，禁止使用），×：禁止使用

软元件/标签的类型	能否使用	
标签(指针型以外)	全局标签	○
	局部标签	○
标签(指针型)	指针型全局标签	△
	指针型局部标签	○
软元件	全局软元件	○
	局部软元件	×
指针	全局指针	△
	局部指针	×

2.4 注意事项

使用函数的情况

关于指针

局部指针、指针型的全局标签不能使用。

使用FB的情况

关于指针

局部指针不能使用。

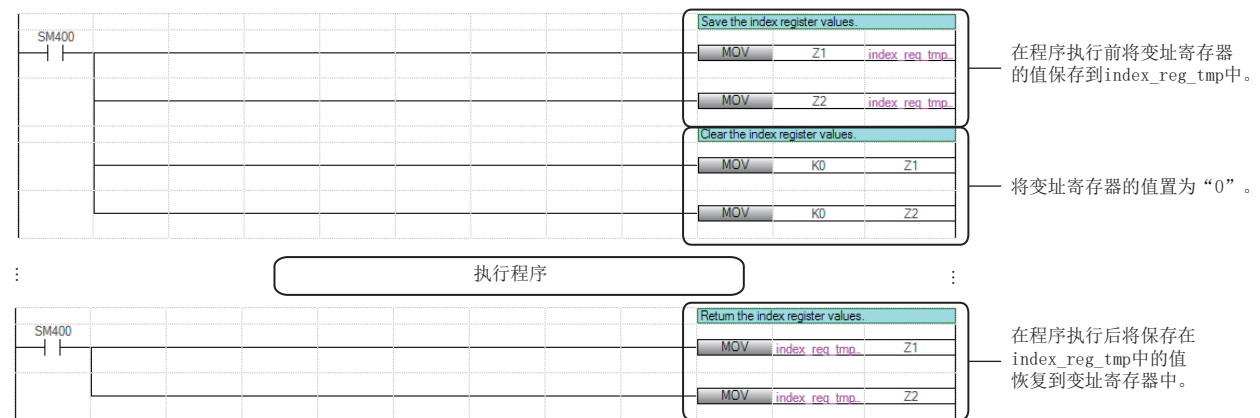
使用变址寄存器的情况

在FB的程序中使用变址寄存器的情况下，为了保护变址寄存器的值，需要保存梯形图与恢复梯形图。

保存变址寄存器时通过将变址寄存器的值设为0，可以防止变址修饰的整合性检查(软元件编号是否超过软元件范围)的错误。

例

执行程序前保存变址寄存器Z1、Z2，在程序执行后恢复保存的变址寄存器的情况



智能功能模块的起始I/O No. 的指定

访问智能功能模块的缓冲存储器或输入输出信号的情况下，应使用变址寄存器指定起始输入输出编号。

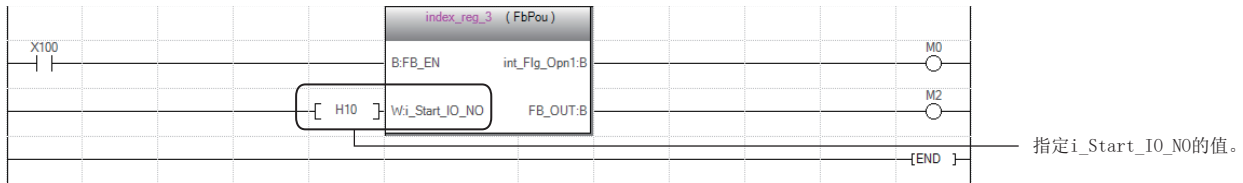
通过将起始输入输出编号作为输入自变量进行接收，则在安装位置不同的多个智能功能模块中，可以不用更改起始输入输出编号而直接使用通用FB。

例

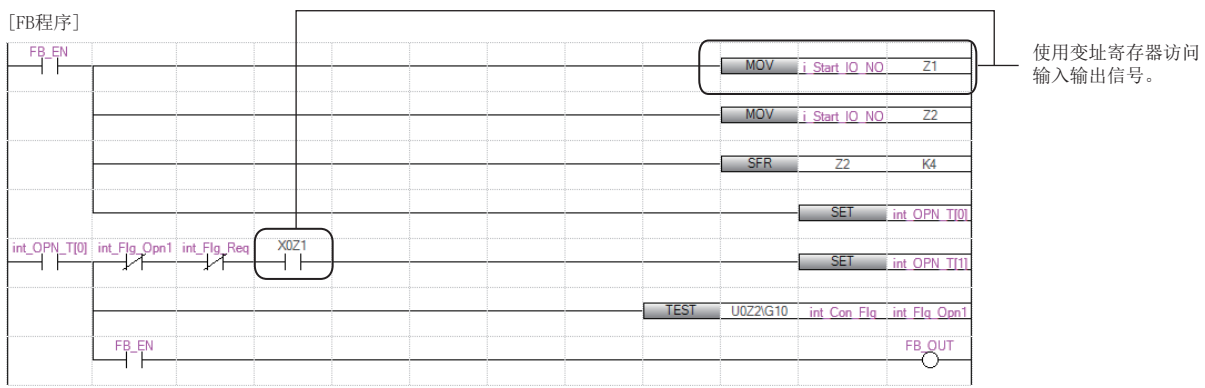
访问智能功能模块的输入输出信号的情况

通过使用变址寄存器，可以访问对象智能功能模块的输入输出信号。

[顺控程序]



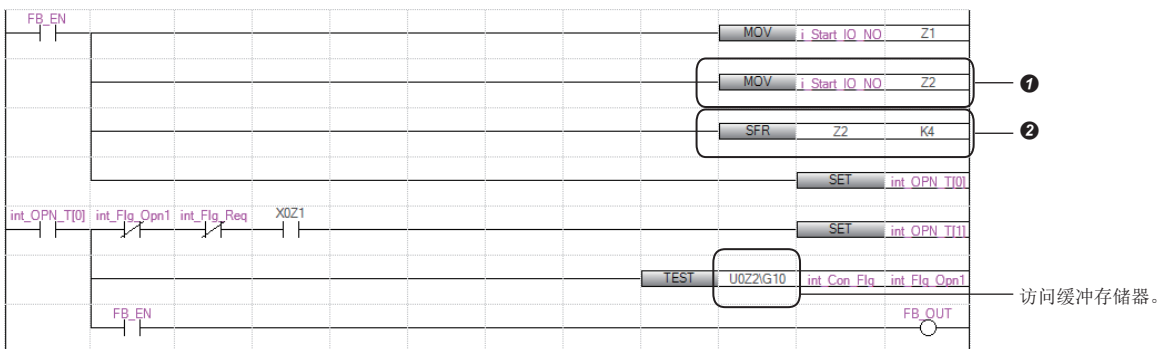
[FB程序]



例

访问智能功能模块的缓冲存储器的情况

- ① 将对象智能功能模块的起始输入输出编号输入到变址寄存器中。
- ② 使用SFR指令，把值向右移四位，或者使用将值除以16所得的商。



关于宏型FB的复制

使用多个宏型FB时，应从部件选择窗口拖放。

FB编译方式为在程序编辑器上复制了宏型FB时，复制的FB将更改为子程序型FB。

复制模块FB或FB库时，应确认所使用的FB的参考手册，确认FB编译方式中是否记载为子程序类型后再使用。

要点

宏型FB是模块FB、FB库中使用的FB的类型。用户无法定义。

动作为在编程时对调用源程序展开调用对象的程序本体。执行时，将执行与常规程序同样展开的程序。

关于宏型FB的调用

作为指针指定目标使用指针型标签的宏型FB的各实例在工程内应仅调用一次。作为指针指定目标使用指针型标签的宏型FB的同一实例被多次调用时，将发生转换错误。

模块FB的限制事项

使用模块FB的情况下，存在以下限制事项。

- 在MC指令至MCR指令之间调用模块FB的情况下，请勿将MC指令的触点置为OFF。
- 请勿进行会使得在CJ指令、SCJ指令、JMP指令中无法调用模块FB的跳转。
- 在子程序内调用模块FB的情况下，每次扫描应执行1次子程序。此外，请勿在FCALL(P)指令、EFCALL(P)指令、XCALL指令中执行子程序的非执行处理。
- 在中断程序、初始执行类型程序、恒定周期执行类型程序、事件执行类型程序内，请勿调用模块FB。
- 在FOR~NEXT指令间、内嵌ST或ST语言的控制语句内(IF语句、FOR语句、CASE语句等)，请勿调用模块FB。
- 如果指定了部分模块FB的输入参数、输出参数中修饰的软元件，则可能会变为错误。希望将修饰的软元件指定为模块FB的输入参数、输出参数时，应记述调用模块FB后传送值的处理。

模块FB的动作参数的更改

模块FB的输入标签及输出标签以外的动作参数(外部变量)的更改可以在标签设置中更改。

- 将模块FB的实例置为局部标签的情况下,可以在“局部标签设置”中更改。

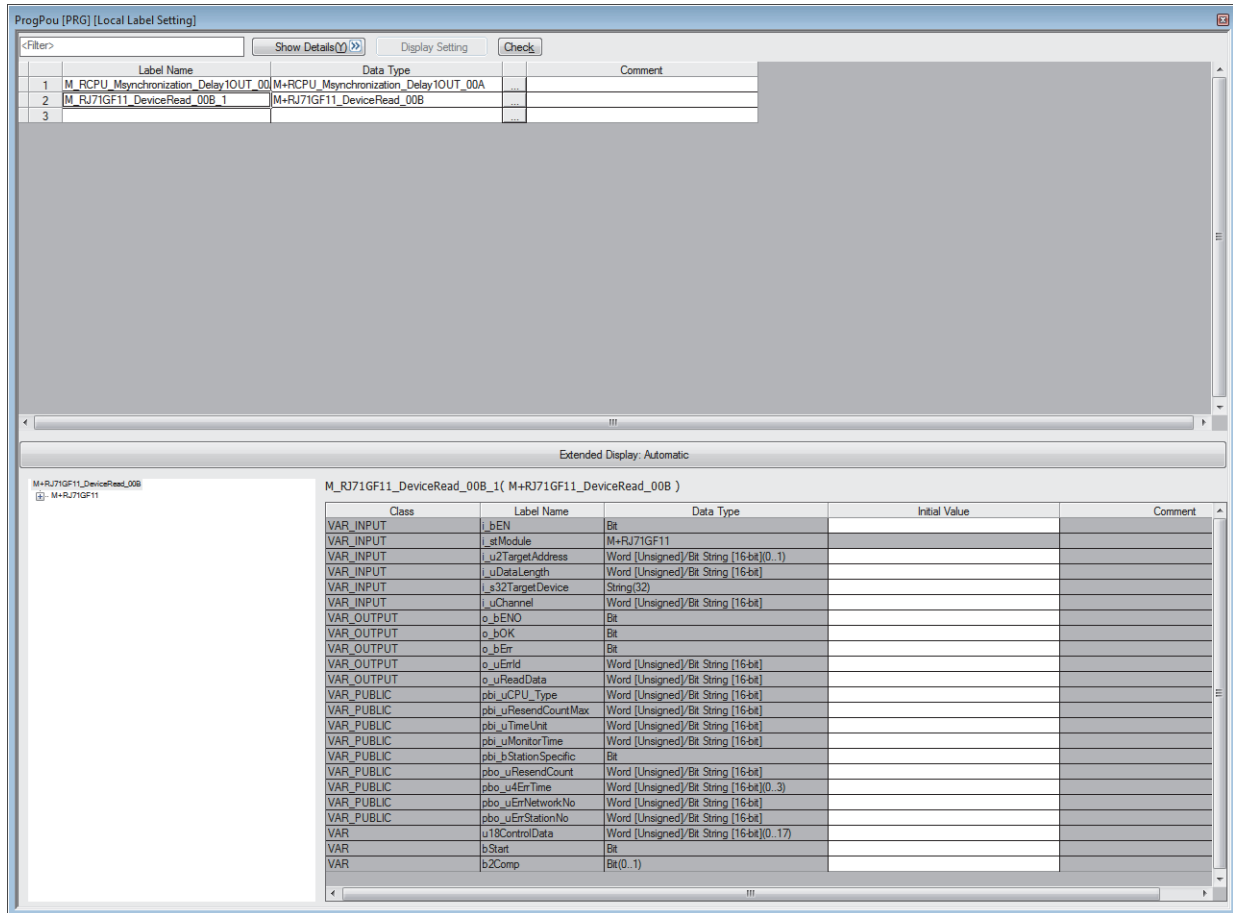
☞ [导航窗口]⇒[程序]⇒执行类型⇒程序文件⇒程序块⇒[局部标签]

- 将模块FB的实例置为全局标签的情况下,可以在“全局标签设置”中更改。

☞ [导航窗口]⇒[标签]⇒[全局标签]

例

设置局部标签的情况



The screenshot shows the 'Local Label Setting' dialog box for the module 'M_RJ71GF11_DeviceRead_00B_1'. The dialog has a table with the following columns: Label Name, Data Type, and Comment. The table contains the following entries:

Label Name	Data Type	Comment
M_RCPU_Myynchronization_DelayIOUT_00	M+RCPU_Myynchronization_DelayIOUT_00A	
M_RJ71GF11_DeviceRead_00B_1	M+R71GF11_DeviceRead_00B	

Below the table, there is a section for 'Extended Display: Automatic' showing a detailed list of variables for the module 'M_RJ71GF11_DeviceRead_00B_1'. The variables are listed with their Class, Label Name, Data Type, and Initial Value.

Class	Label Name	Data Type	Initial Value	Comment
VAR_INPUT	i_bEN	Bit		
VAR_INPUT	i_stModule	M+R71GF11		
VAR_INPUT	i_u2TargetAddress	Word [Unsigned]/Bit String [16-bit][0..1]		
VAR_INPUT	i_uDataLength	Word [Unsigned]/Bit String [16-bit]		
VAR_INPUT	i_s32TargetDevice	String(32)		
VAR_INPUT	i_uChannel	Word [Unsigned]/Bit String [16-bit]		
VAR_OUTPUT	o_bENO	Bit		
VAR_OUTPUT	o_bOK	Bit		
VAR_OUTPUT	o_bErr	Bit		
VAR_OUTPUT	o_uErrId	Word [Unsigned]/Bit String [16-bit]		
VAR_OUTPUT	o_uReadData	Word [Unsigned]/Bit String [16-bit]		
VAR_PUBLIC	pbi_uCPU_Type	Word [Unsigned]/Bit String [16-bit]		
VAR_PUBLIC	pbi_uResendCountMax	Word [Unsigned]/Bit String [16-bit]		
VAR_PUBLIC	pbi_uTimeUnit	Word [Unsigned]/Bit String [16-bit]		
VAR_PUBLIC	pbi_uMonitorTime	Word [Unsigned]/Bit String [16-bit]		
VAR_PUBLIC	pbi_bStationSpecific	Bit		
VAR_PUBLIC	pbo_uResendCount	Word [Unsigned]/Bit String [16-bit]		
VAR_PUBLIC	pbo_u4ErrTime	Word [Unsigned]/Bit String [16-bit][0..3]		
VAR_PUBLIC	pbo_uErrNetworkNo	Word [Unsigned]/Bit String [16-bit]		
VAR_PUBLIC	pbo_uErrStationNo	Word [Unsigned]/Bit String [16-bit]		
VAR	u18ControlData	Word [Unsigned]/Bit String [16-bit][0..17]		
VAR	bStart	Bit		
VAR	b2Comp	Bit(0..1)		

在“初始值”栏中设置动作参数。

使用MC/MCR指令控制EN的动作

在FB的固有属性设置中将“使用MC/MCR控制EN”设置为有效时在FB内使用的指令、软元件/标签的动作如下所示。

FB内使用的指令、软元件/标签	FB内使用的指令、软元件/标签的状态	
	将“使用MC/MCR控制EN”选为“是”时	将“使用MC/MCR控制EN”选为“否”时
上升沿/下降沿指令(PLS指令、脉冲化指令(□P)*1)	在下一个EN变为ON时，条件触点若成立，则执行指令。	但在下一个EN变为ON时，即使条件触点成立，也有可能发生不执行指令的情况。
定时器(低速/高速)、长定时器	计数值变为0，且线圈、触点也变为OFF。	保持现状。
累计定时器(低速/高速)、长累计定时器、计数器、长计数器	线圈变为OFF，但计数值、触点仍保持现状。	保持现状。
OUT指令的软元件部中指定的软元件	强制变为OFF。	保持现状。

*1 线圈侧中指定的指令为对象。

限制事项

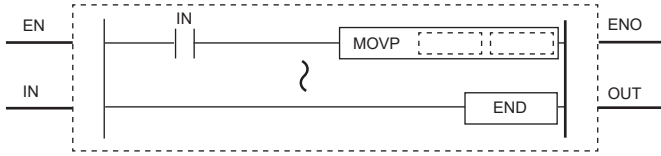
将“使用MC/MCR控制EN”选为“是”的情况下，在该FB处于执行中时，请勿使用MC/MCR指令。使用了MC/MCR指令的情况下，EN的控制可能无法正确动作。

上升沿/下降沿指令的动作

上升沿/下降沿指令的动作如下所示。

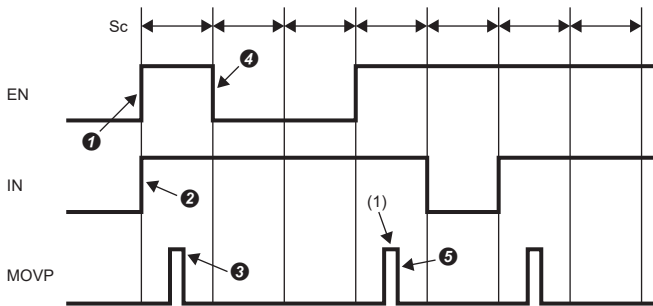
例

使用上升沿指令的子程序型FB



■将“使用MC/MCR控制EN”选为“是”时

在EN变为ON时，条件触点若成立，则执行指令。(图中(1))

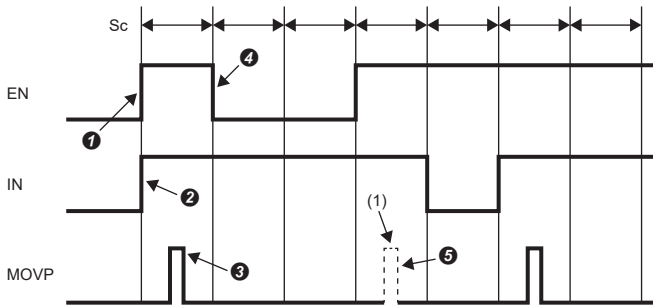


Sc: 扫描

- 1 将EN置为ON。(用户操作)
- 2 将IN置为ON。(用户操作)
- 3 执行MOVP指令。(控制器动作)
- 4 将EN置为OFF。(用户操作)
- 5 执行MOVP指令。(控制器动作)

■将“使用MC/MCR控制EN”选为“否”时

EN为OFF时，根据条件触点状态，指令的动作将有所不同。(图中(1))



Sc: 扫描

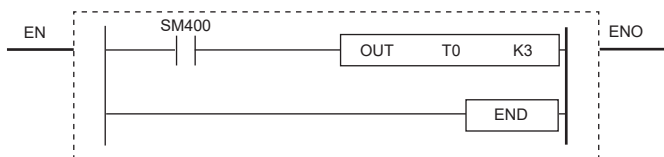
- 1 将EN置为ON。(用户操作)
- 2 将IN置为ON。(用户操作)
- 3 执行MOVP指令。(控制器动作)
- 4 将EN置为OFF。(用户操作)
- 5 4中，若条件触点在EN变为OFF前变为OFF，则执行MOVP指令。(控制器动作) (4中，若条件触点在EN变为OFF前变为ON，则不执行MOVP指令。)

定时器(低速/高速)、长定时器的动作

定时器(低速/高速)、长定时器的动作如下所示。

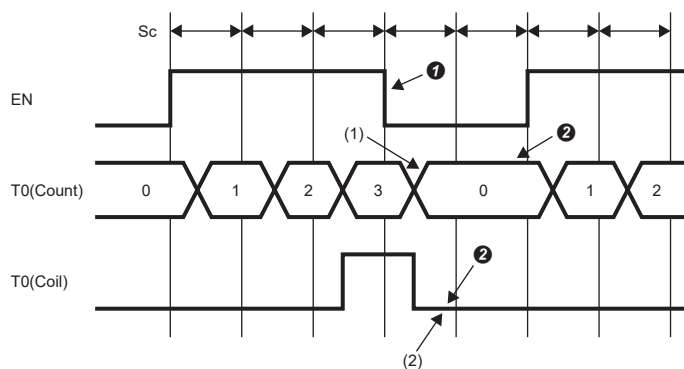
例

使用低速定时器的子程序型FB



■将“使用MC/MCR控制EN”选为“是”时

计数值变为0。(图中(1))此外,线圈变为OFF。(图中(2))



Sc: 扫描

T0(Count): T0(计数值)

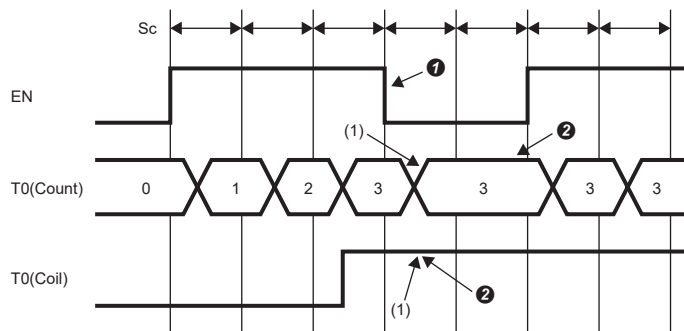
T0(Coil): T0(线圈)

① 将EN置为OFF。(用户操作)

② 线圈变为OFF,清除定时器值、计数值。(控制器动作)

■将“使用MC/MCR控制EN”选为“否”时

计数值和线圈皆保持现状。(图中(1))



Sc: 扫描

T0(Count): T0(计数值)

T0(Coil): T0(线圈)

① 将EN置为OFF。(用户操作)

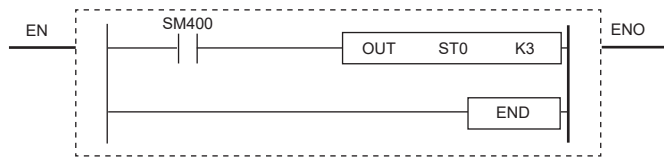
② 保持值。(控制器动作)

累计定时器(低速/高速)、长累计定时器、计数器、长计数器的动作

累计定时器(低速/高速)、长累计定时器、计数器、长计数器的动作如下所示。

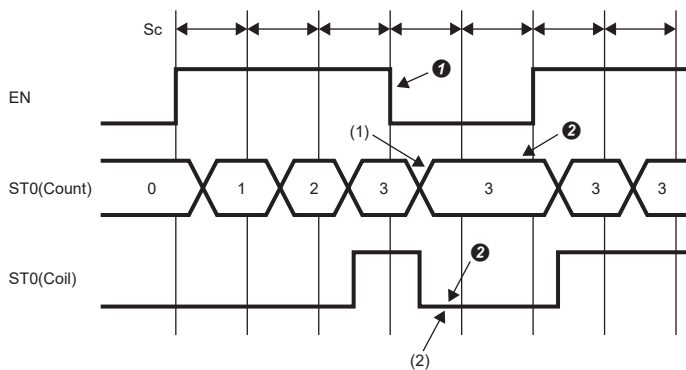
例

使用低速累计定时器的子程序型FB



■将“使用MC/MCR控制EN”选为“是”时

计数值保持现状。(图中(1))此外,线圈变为OFF。(图中(2))



Sc: 扫描

ST0(Count): T0(计数值)

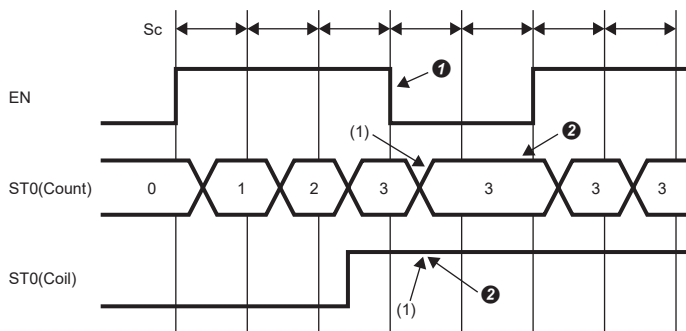
ST0(Coil): T0(线圈)

① 将EN置为OFF。(用户操作)

② 线圈变为OFF,但计数值、触点仍保持当前的状态。(控制器动作)

■将“使用MC/MCR控制EN”选为“否”时

计数值和线圈皆保持现状。(图中(1))



Sc: 扫描

ST0(Count): T0(计数值)

ST0(Coil): T0(线圈)

① 将EN置为OFF。(用户操作)

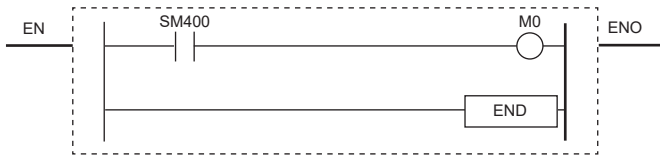
② 保持值。(控制器动作)

OUT指令的软元件部中指定的软元件的动作

OUT指令的软元件部中指定的软元件的动作如下所示。

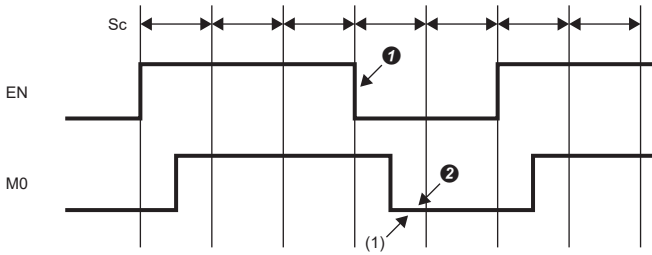
例

OUT指令的软元件部中使用M0的子程序型FB



■将“使用MC/MCR控制EN”选为“是”时

M0强制变为OFF。(图中(1))

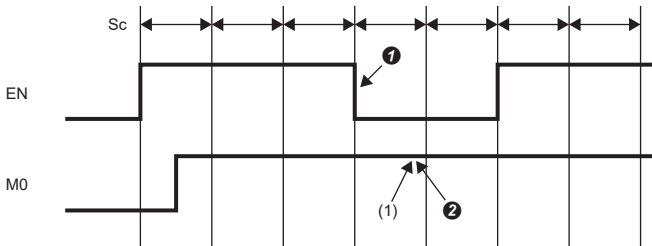


Sc: 扫描

- ① 将EN置为OFF。(用户操作)
- ② 将线圈输出置为OFF。(控制器动作)

■将“使用MC/MCR控制EN”选为“否”时

M0保持现状。(图中(1))



Sc: 扫描

- ① 将EN置为OFF。(用户操作)
- ② 保持线圈。(控制器动作)

3 梯形图语言

是在由触点与线圈构成的梯形图中，表示以串联与并联的组合进行AND/OR逻辑运算，记述顺控程序控制的语言。

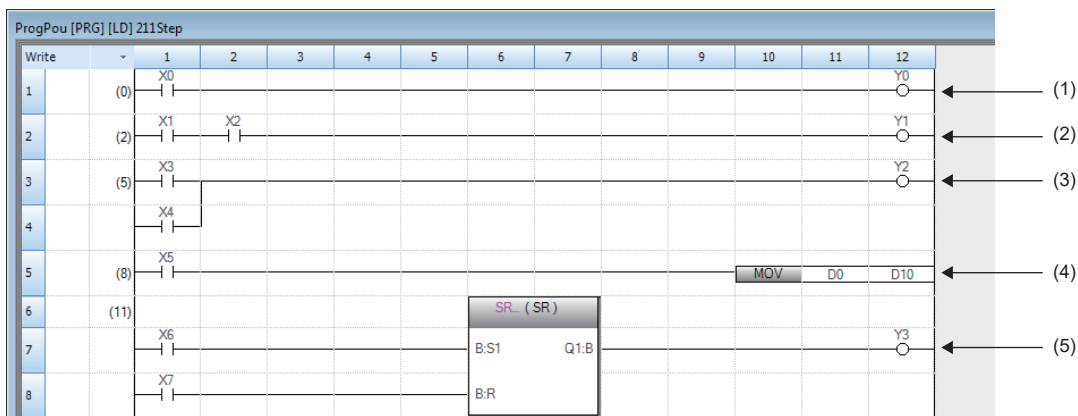
要点

在本章中，对梯形图语言的动作及规格有关内容进行说明。关于创建梯形图程序时的操作方法，请参阅下述手册。

📖 GX Works3 操作手册

3.1 配置








梯形图语言中，可以创建下述梯形图。


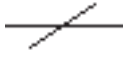


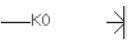
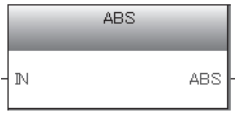
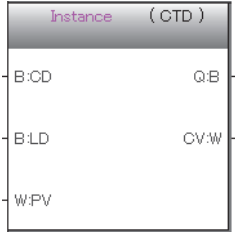


- (1) 由触点与线圈组成的梯形图
- (2) 由串联组成的梯形图
- (3) 由并联组成的梯形图
- (4) 使用了指令的梯形图
- (5) 使用了通用函数/FB的梯形图

梯形图符号

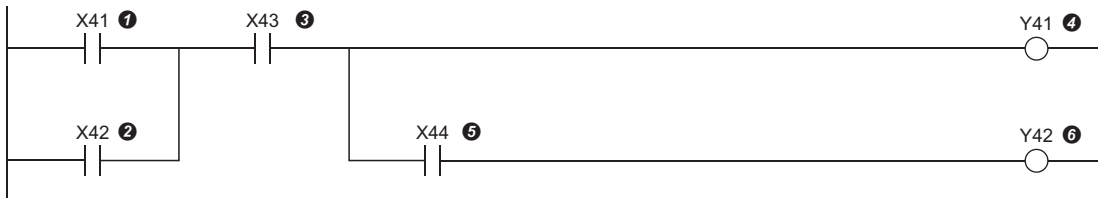
可以在梯形图语言的编程中使用的梯形图符号如下所示。

项目	内容
常开触点	 指定软元件或标签变为ON时导通。
常闭触点	 指定软元件或标签变为OFF时导通。
上升沿脉冲	 指定软元件或标签上升沿时 (OFF→ON) 导通。
下降沿脉冲	 指定软元件或标签下降沿时 (ON→OFF) 导通。
上升沿脉冲否定	 指定软元件或标签OFF时、ON时及下降沿时 (ON→OFF) 导通。
下降沿脉冲否定	 指定软元件或标签OFF、ON及上升沿时 (OFF→ON) 导通。
运算结果上升沿脉冲化	 运算结果上升沿时 (OFF→ON) 导通。运算结果在上升沿以外的情况不导通。

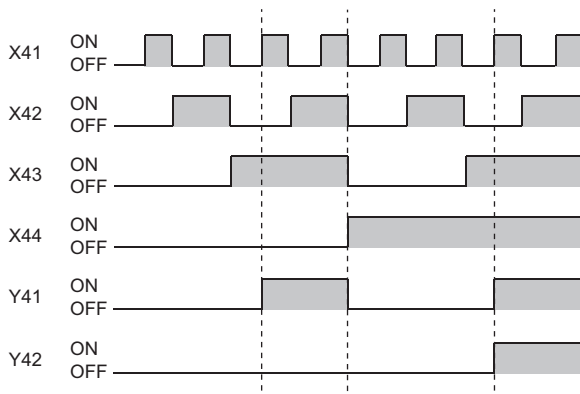
项目		内容
运算结果下降沿脉冲化		运算结果下降沿时 (ON→OFF) 导通。运算结果在下降沿以外的情况不导通。
运算结果取反		将之前的运算结果取反。
线圈		将运算结果输出至指定软元件或标签。
指令		执行 [] 内指定的指令。
换行		触点数超过了1个梯形图的行中可创建的点数的情况下，创建换行源的符号及换行目标的符号，执行梯形图的换行。
函数		执行函数。 <ul style="list-style-type: none"> 函数的创建方法 (☞ GX Works3 操作手册) 通用功能 (☞ 177页 通用函数)
FB		执行FB。 <ul style="list-style-type: none"> FB的创建方法 (☞ GX Works3 操作手册) 通用FB (☞ 188页 通用FB) 运动控制FB (☞ 190页 运动控制FB一览) 模块FB (☞ 所使用的模块的FB参考)

程序执行顺序

按照下述的编号顺序执行。



执行了上述程序的情况下，根据X41~X44的ON/OFF，Y41、Y42变为ON的时机如下所示。

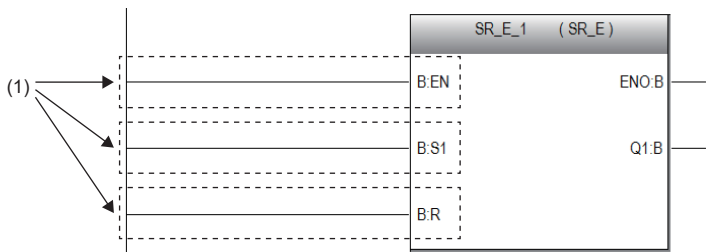


以梯形图语言使用FB时的注意事项

左母线已直接连接FB实例时的注意事项

在FB实例的输入梯形图部，EN及输入变量(位型)与左母线直接连接时，ON/OFF的状态不发生变化。

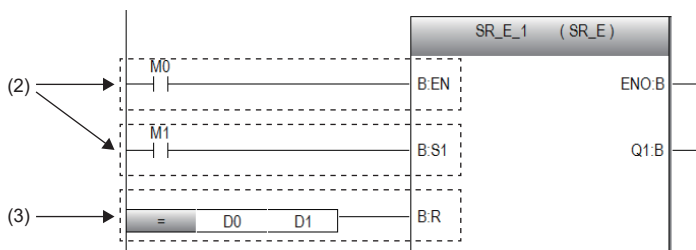
(1) ON/OFF的状态不发生变化。



若要使EN及输入变量(位型)的ON/OFF的状态发生变化，应使用触点或相当于触点的指令。

(2) 触点

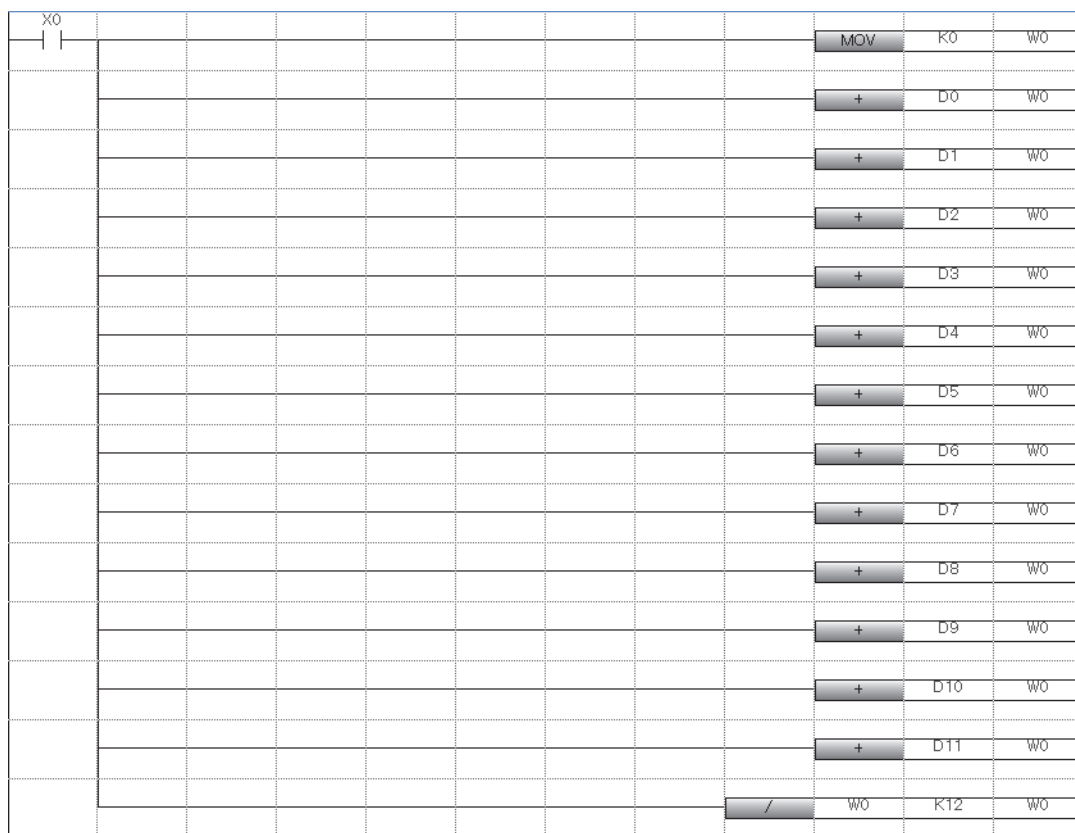
(3) 相当于触点的指令



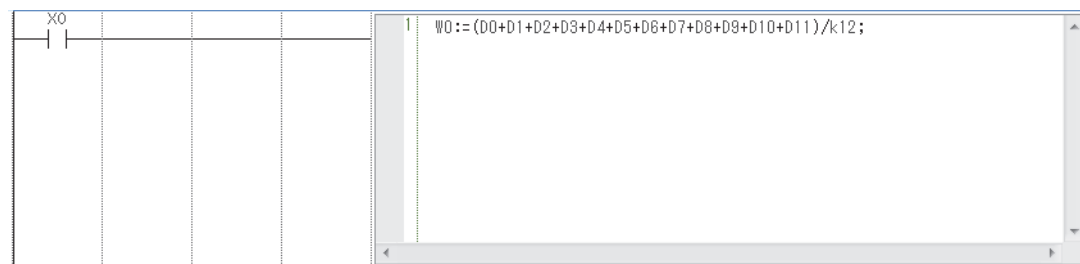
3.2 内嵌ST

内嵌ST是指在梯形图编辑器内创建在与线圈相当的指令单元格中显示ST程序的内嵌ST盒子，进行编辑/监视的功能。由此，可以轻松地在梯形图程序内创建数值运算及字符串处理。

- 不使用内嵌ST情况下的程序



- 使用了内嵌ST情况下的程序



规格

关于内嵌ST中记述的ST程序的规格，请参阅ST语言的规格。


☞ 90页 ST语言

注意事项

- 梯形图程序的1行中只能创建一个内嵌ST。
- 在梯形图程序的1行中无法使用函数/FB与内嵌ST盒两者。
- 如果在触点相应的指令位置创建内嵌ST盒，在线圈相应的指令位置也创建内嵌ST盒。
- 内嵌ST内最多可输入的字符数为20000个字符。（换行作为2个字符进行计数。）
- 如果在内嵌ST内使用“RETURN语句”，不是结束程序块的处理，而是结束内嵌ST盒内的处理。
- 在函数的程序中使用内嵌ST的情况下，无法从内嵌ST内调用FB。
- 在内嵌ST中，转换时使用CJ指令控制程序的动作。内嵌ST的触点为OFF的情况下，内嵌ST内的处理将不通过CJ指令执行。因此，即使通过内嵌ST内的代入语句变为ON的软元件不执行内嵌ST，也将保持输出状态。关于CJ指令的详细内容，请参阅下述手册。

 516页 CJ、SCJ、JMP

- 无法在内嵌ST内使用OUT、OUTH及参照信号流存储器上一次执行状态的指令。

 所使用的控制器的用户手册

3.3 声明/注解

在梯形图中，可以显示声明及注解。

声明

通过使用声明，可以对梯形图块添加注释。通过进行添加，处理等流程变得易懂。

声明中有行间声明/P声明/I声明。

行间声明可以在导航窗口的树状图上显示。

■行间声明

对整个梯形图块添加注释。

■P声明

对指针软元件添加注释。

■I声明

对中断指针软元件添加注释。

注解

通过使用注解，可以对程序中的线圈及指令添加注释。

通过添加注释，线圈及指令的内容等变得易懂。

声明/注解的类型

声明与注解的类型有“全体”与“外围”两种。


类型	种类	内容
全体	<ul style="list-style-type: none"> • 行间声明 • P声明 • I声明 • 注解 	说明文的字符串转换时不置入到程序内，而是作为程序的附加信息保存。 一行使用一步。
外围	<ul style="list-style-type: none"> • 行间声明 • P声明 • I声明 • 注解 	说明文的字符串不置入到程序内，而是作为程序的附加信息保存。 一行使用一步。 在输入的文本前自动添加*印记。

4 ST语言

ST语言是规定逻辑记述方法的标准IEC61131-3所定义的语言。ST语言是具有与C语言等相似的语法结构的文本形式的程序语言。适用于对难以梯形图语言表现的复杂处理进行编程的情况下。

要点

在本章中，对ST语言的动作及规格有关内容进行说明。关于创建ST程序时的操作方法，请参阅下述手册。

 GX Works3 操作手册

ST语言支持控制语法、运算式、功能块(FB)、函数(FUN)，可以按以下方式记述。

例

条件语句的选择分支、重复语句等的控制语法

```
(*以线A~C进行控制*)
CASE 线 OF
  1:
    开始开关 := TRUE; (*传送带运行*)
  2:
    开始开关 := FALSE; (*传送带停止*)
  3:
    开始开关 := TRUE; (*传送带停止 警告*)
ELSE
  警告指示灯 := TRUE;
END_CASE;
IF 开始开关 = TRUE THEN (*传送带运行 处理100次*)
  FOR 处理次数 := 0 TO 100 BY 1 DO
    处理数 := 处理数 +1;
  END_FOR;
END_IF;
```

例

使用运算符(*、/、+、-、<、>、=等)的表达式

```
DO := D1 * D2 + D3 / D4 - D5 ;
IF DO > D10 THEN
  DO := D10 ;
END_IF;
```

例

定义的FB的调用

```
//FB数据名: LINE1_FB
//输入变量: I_Test
//输出变量: O_Test
//输入输出变量: IO_Test
//FB标签名: FB1
FB1(I_Test:= D0 , O_Test => D1 ,IO_Test:= D100);
```

例

通用函数的调用

```
(* 将BOOL型数据转换为INT型/DINT型数据 *)
wLabel12 := BOOL_TO_INT(bLabel11);
```

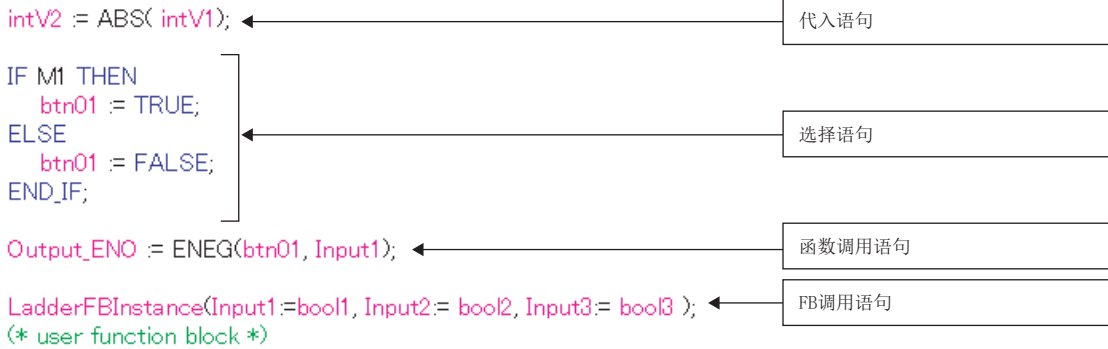
例

汉字等全角字符的使用

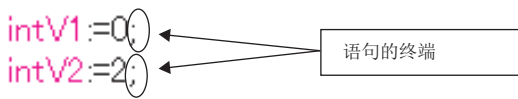
```
//油箱限制器ON时关闭阀，OFF时打开阀
IF 油箱限制器 = TRUE THEN
  阀 := FALSE ; (* 由于限制器变为ON，关闭阀 *)
ELSE
  阀 := TRUE ; /* 由于限制器变为OFF，打开阀 */
END_IF;
```

4.1 配置

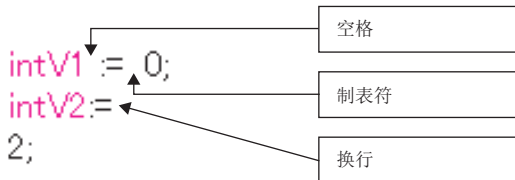
ST语言中的编程由运算符与语法组成。



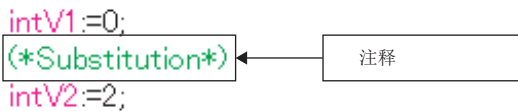
句子的终端必须添加“;”（分号）。



空格、制表符、换行可以插入到运算符及数据中。



可以在程序中插入注释。在注释语句的前后记述“(*注释语句*)”。



程序的结构要素

ST程序由以下要素构成。

项目	示例	参照目标	
分隔符	、 (、)	92页 分隔符	
运算符	+、-、<、>、=	92页 运算符	
保留字	语法	IF、CASE、WHILE、RETURN	
	软元件	X40、Y50、M100、ZRO	所使用的控制器的用户手册
	数据类型	BOOL、DWORD	所使用的控制器的用户手册
	通用函数	ADD、REAL_TO_STRING_E	177页 通用函数
常数	123、“abc”	103页 常数	
标签	Switch_A	104页 标签与软元件	
注释	(*置为ON*)	106页 注释	
其它符号	半角空格、换行代码、TAB代码	—	

- 分隔符、运算符号、保留字应用半角记述。
- 关于保留字的详细内容，请参阅下述手册。

GX Works3 操作手册

分隔符

在ST语言中为了明确表示程序的结构，有下述的分隔符。

符号	内容
()	圆括弧式
[]	数组要素编号的指定
. (句号)	结构体、FB构件的指定
, (逗号)	自变量的分隔
:(冒号)	软元件型指定符、CASE语句的分隔
;(分号)	语句的终端
"(双引号)	Unicode字符串的标记
'(单引号)	字符串(ASCII、移位JIS)的标记
.. (两个句号)	整数范围指定

运算符

在ST程序中使用的运算符、对象数据类型与运算结果的数据类型如下所示。

运算符	对象数据类型	运算结果类型
*, /, +, -	ANY_NUM*1	ANY_NUM
<, >, <=, >=, =, <>	ANY_ELEMENTARY*2	位
MOD	ANY_INT	ANY_INT
AND, &, XOR, OR, NOT	ANY_BIT	ANY_BIT
**	ANY_REAL(底) ANY_NUM(指数)*1	ANY_REAL

*1 不可以指定“WORD#”或“DWORD#”的常数标记。(☞ 103页 常数的标记方法)

*2 不可以指定WSTRING型的Unicode字符串。

运算符的优先顺序如下所示。

运算符	内容	示例	优先顺序
()	圆括弧式	(2+3)*(4+5)	1
函数()	函数调用方式	CONCAT(' AB', ' CD')	2
-	符号取反	-10	3
NOT	位型补数	NOT TRUE	
**	次方	3.0**4	4
*	乘法	10*20	5
/	除法	20/10	
MOD	求余数	17 MOD 10	
+	加法	1.4+2.5	6
-	减法	3-2	
<, >, <=, >=	比较	10>20	7
=	一致	T#26h=T#1d2h	8
<>	不一致	8#15<>13	
&, AND	逻辑与	TRUE AND FALSE	9
XOR	异或	TRUE XOR FALSE	10
OR	逻辑或	TRUE OR FALSE	11

- 一个公式中有多个相同优先顺序的运算符的情况下，从左侧的运算符开始运算。
- 一个公式中，最多可记述的运算符使用个数为1024个。

指令

ST程序中使用的指令如下所示。

```

      (1)      (2)      (3)
      ↓        ↓        ┌──────────┴──────────┐
bResult := BMOV( TRUE, wValue0, 1, wValue1);
  
```

- (1) 指定为返回值。
(在上述示例中，将ENO代入bResult。指令的情况下，返回值的记述可以省略。)
- (2) 指定为指令名。
- (3) 指定为指令的参数。

指令的EN的参数指定省略

指令的情况下，EN的参数指定可以省略。

此时，由于视为TRUE分配给EN而进行动作，因此可以进行如下程序记述。

```

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

在通用函数等指令以外不能省略EN。

语法

可在ST程序中使用的语法如下所示。

项目	内容	参照
代入语句	代入语句	94页 代入语句
子程序控制语句	FB调用语句、函数调用语句	96页 子程序控制语句
	RETURN语句	
选择语句	IF语句(IF THEN、IF ELSE、IF ELSIF)	97页 选择语句
	CASE语句	
重复语句	FOR语句	97页 重复语句
	WHILE语句	
	REPEAT语句	
	EXIT语句	

应用半角字符记述语法。

代入语句

书写格式	内容	记述示例
<左边>:=<右边>;	具有将右边公式的结果代入到左边的标签及软元件中的功能。 右边公式的结果需与左边的数据类型相同。	intV1:=0; intV2:=2;

使用数组型标签及结构体标签的情况下，应注意代入语句的左边与右边的数据类型。
数组型标签的情况下，左边与右边的数据类型与要素数需要相同。此外，请勿指定要素。

例

```
intAry1:=intAry2;
```

结构体标签的情况下，左边与右边的数据类型(结构体的数据类型)需要相同。

例

```
dutVar1:=dutVar2;
```

右边为函数调用方式的情况下，函数的返回值将代入左边。将返回值代入标签的示例如下所示。

函数	记述示例
具有1个输入变量的函数的情况(例: ABS)	Output1 := ABS(Input1);
具有3个输入变量的函数的情况(例: MAX)	Output1 := MAX(Input1, Input2, Input3);
具有EN/ENO的函数(通用函数)的情况(例: MAX_E)	Output1 := MAX_E(boolEN, boolENO, Input1, Input2, Input3);
通用函数以外的情况(例: MOV)	boolENO := MOV(boolEN, Input1, Output1); (执行完函数的结果为ENO, 第一自变量(变量1)变为EN。)

■数据类型的自动转换

在ST语言中记述不同数据类型的代入或算术运算公式时，可以自动转换数据类型。

例

自动转换示例

```
dintLabel1 := intLabel1 ;
//代入语句：将INT型(intLabel1)的值自动转换为DINT型，代入至左边的DINT型(dintLabel1)

dintLabel1 := dintLabel2 + intLabel1 ;
//算术运算公式：INT型(intLabel1)的值自动转换为DINT型，执行DINT型的加法运算

DMOV(TRUE, wordLabel1, dwordLabel1);
//指令、函数、FB调用语句：将WORD型输入自变量(wordLabel1)的值自动转换为DWORD型，执行传送
```

类型转换在代入语句、向FB及函数(包括指令、通用函数、通用FB)交接输入自变量(VAR_INPUT部)、算术运算公式中进行。从数据类型容量小的开始往容量大的顺序进行，以确保类型转换时不丢失数据。类型转换以数据类型中的下述数据类型为对象。

数据类型	内容
字[带符号]	转换后变为双字[带符号]的情况下，自动转换为符号扩展值。 单精度实数或双精度实数的情况下，自动转换为与转换前的整数相同的值。*1
字[无符号]/位串[16位]	转换后为双字[无符号]/位串[32位]或双字[带符号]的情况下，自动转换为零扩展值。*2 单精度实数或双精度实数的情况下，自动转换为与转换前的整数相同的值。*1
双字[带符号]	转换后为双精度实数的情况下，自动转换为与转换前的整数相同的值。
双字[无符号]/位串[32位]	
单精度实数	转换后为双精度实数的情况下，自动转换为相同的值。

*1 数据类型若将16位数据(字[带符号]或字[无符号]/位串[16位])过渡为ANY_REAL的输入自变量，则自动转换为单精度实数。

*2 数据类型若将字[无符号]/位串[16位]数据过渡为ANY32的输入自变量下，则自动转换为双字[无符号]/位串[32位]。

上述以外的数据类型，应使用类型转换函数。

此外，下述情况也无法进行类型转换，应使用类型转换函数。

- 符号不同的整数型之间的类型转换
- 数据丢失型之间的类型转换

代入算术运算的结果时的注意事项，请参阅下述章节。

☞ 98页 代入算术运算式结果的情况

使用软元件时的注意事项，请参阅下述章节。

☞ 105页 使用软元件时的数据类型自动转换

子程序控制语句

■FB调用语句

书写格式	内容
实例名(输入变量1:=变量1,... 输出变量1=>变量2,...);	在实例名后,用“()”括住输入变量、输出变量的代入语句。 多个变量的情况下,各代入语句之间用“,”(逗号)隔开。
实例名.输入变量1:=变量1; : 实例名(); 变量2:=实例名.输出变量1;	在FB调用的前后列举输入自变量、输出自变量的代入语句。

在FB调用语句的自变量中所使用的符号与可分配表达式如下所示。

类型	内容	使用符号	可分配表达式
EN, VAR_INPUT	输入变量	:=	所有的表达式
ENO, VAR_OUTPUT, VAR_OUTPUT_RETAIN	输出变量	=>	只有变量
VAR_IN_OUT	输入输出变量	:=	只有变量

FB的执行结果通过在实例名后添加“.”(句号)指定输出变量且代入变量被存储。

FB	FB定义	记述示例
具有1个输入变量、1个输出变量的FB的情况	FB名: FBADD FB实例名: FBADD1 输入变量1: IN1 输出变量1: OUT1	FBADD1(IN1:= Input1); Output1 := FBADD1.OUT1;
具有3个输入变量、2个输出变量的FB的情况	FB名: FBADD FB实例名: FBADD1 输入变量1: IN1 输入变量2: IN2 输入变量3: IN3 输出变量1: OUT1 输出变量2: OUT2	FBADD1(IN1:= Input1, IN2:= Input2, IN3:= Input3); Output1 := FBADD1.OUT1; Output2 := FBADD1.OUT2;

■函数调用语句

无返回值的函数或参数中包含VAR_OUTPUT变量的函数,通过在函数调用方式的后方加上“;”(分号),可作为语句执行。

书写格式	内容
函数名(变量1,变量2,...);	用“()”将紧接在函数名后的自变量括起来。 多个自变量的情况下用“,”(逗号)隔开。

■RETURN语句

语法	书写格式	内容	记述示例
■RETURN	RETURN;	在中途结束程序、FB、函数时使用。 如果在程序中使用RETURN语句,将跳转到程序的最后语句的下一步。 如果在FB中使用RETURN语句,将从FB返回。 如果在函数中使用RETURN语句,将从函数返回。 对于1个RETURN语句,在系统中使用1点指针型标签。	IF boo11 THEN RETURN; END_IF;

选择语句

语法	书写格式	内容	记述示例
■IF THEN	IF<布尔表达式>THEN <语句...>; END_IF;	布尔表达式(条件表达式)为真(TRUE)时, 执行语句。布尔表达式为假(FALSE)时, 不执行语句。 对于布尔表达式, 只要单一的位型变量的状态或包含多个变量的复杂表达式的布尔运算结果为返回真(TRUE)或假(FALSE)的表达式, 则可以在任意表达式中使用。	IF bool1 THEN intV1 := intV1 + 1; END_IF;
■IF...ELSE	IF<布尔表达式>THEN <语句1...>; ELSE <语句2...>; END_IF;	布尔表达式(条件表达式)为真(TRUE)时, 执行语句1。 布尔表达式的值为假(FALSE)时, 执行语句2。	IF bool1 THEN intV3 := intV3 + 1; ELSE intV4 := intV4 + 1; END_IF;
■IF...ELSIF	IF<布尔表达式1>THEN <语句1...>; ELSIF<布尔表达式2>THEN <语句2...>; ELSIF<布尔表达式3>THEN <语句3...>; END_IF;	布尔表达式(条件表达式)1为真(TRUE)时, 执行语句1。布尔表达式1的值为假(FALSE)且布尔表达式2的值为真(TRUE)时, 执行语句2。 布尔表达式1、2的值都为假(FALSE)且布尔表达式3的值为真(TRUE)时, 执行语句3。	IF bool1 THEN intV1 := intV1 + 1; ELSIF bool2 THEN intV2 := intV2 + 2; ELSIF bool3 THEN intV3 := intV3 + 3; END_IF;
■CASE	CASE<整数表达式>OF <整数选择值1>: <语句1...>; <整数选择值2>: <语句2...>; : <整数选择值n>: <语句n...>; ELSE <语句n+1...>; END_CASE;	具有与整数表达式(条件表达式)的值一致的整数的选择值的语句被执行后, 若无一致, 则执行ELSE语句的下一语句。 例如, 根据单一的整数及复杂表达式的结果的整数, 可在执行选择语句时使用CASE语句。	CASE intV1 OF 1 : bool1 := TRUE; 2 : bool2 := TRUE; ELSE intV1 := intV1 + 1; END_CASE;

重复语句

语法	书写格式	内容	记述示例
■FOR...DO	FOR<重复变量初始化> TO<最终值> BY<增加表达式>DO <语句...>; END_FOR;	对作为反复变量使用的数据初始化。 根据增加公式对初始化后的反复变量进行加法或减法运算, 重复执行自DO到END_FOR间的一个及其以上的语句, 直至超出最终值为止。 FOR...DO语句结束后的反复变量, 保持在结束时的值。	FOR intV1 := 0 TO 30 BY 1 DO intV3 := intV1 + 1; END_FOR;
■WHILE...DO	WHILE<布尔表达式>DO <语句...>; END_WHILE;	布尔表达式(条件表达式)为真(TRUE)时, 执行1个及其以上的语句。 布尔表达式在执行语句之前被判定, 布尔表达式为假(FALSE)时, 不执行DO...END_WHILE中的语句。WHILE语句中的<布尔表达式>, 无论返回的结果为真或假, 只要有返回即可, 因此IF语句中的<布尔表达式>中可指定的表达式全可以使用。	WHILE intV1 = 30 DO intV1 := intV1 + 1; END_WHILE;
■REPEAT...UNTIL	REPEAT <语句...>; UNTIL<布尔表达式> END_REPEAT;	布尔表达式(条件表达式)为假(FALSE)时, 执行1个及其以上语句。 布尔表达式在执行语句后被判定, 值为真(TRUE)时则不执行REPEAT...UNTIL内的语句。REPEAT语句中的<布尔表达式>, 无论返回的结果为真或假, 只要有返回即可, 因此IF语句中的<布尔表达式>中的可指定的表达式全可以使用。	REPEAT intV1 := intV1 + 1; UNTIL intV1 = 30 END_REPEAT;
■EXIT	EXIT;	只可在重复语句中使用的语法, 于中途结束重复语句。 如果在执行重复环路过程中达到EXIT语句, 则不执行EXIT语句之后的重复环路处理。从终止重复语句后的下一行开始继续执行程序。	FOR intV1 := 0 TO 10 BY 1 DO IF intV1 > 10 THEN EXIT; END_IF; END_FOR;

注意事项

■使用代入语句时

- 字符串的代入的最大字符串长为255字符。代入的字符串超过最大字符串长时，将变为转换错误。
- 定时器型、计数器型的触点与线圈无法在代入语句的左边使用。
- FB的实例无法在代入语句的左边使用。应在代入式的左边使用实例的输入变量、输出变量、外部变量。

■代入算术运算式结果的情况

将算术运算表达式的结果代入到数据容量较大的数据类型中的情况下，应预先把算术运算表达式的变量转换为左边的数据类型之后再行运算。

例

把数据容量16位(INT型)的算术运算结果代入到32位的数据类型(DINT型)的情况

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

算术运算表达式的运算结果将变为与输入操作数的数据类型相同的数据类型。因此在上述的程序中，varInt1*10的运算结果超出了INT型的范围(-32768~32767)的情况下，会将上溢或下溢的运算结果代入到varDint1中。

在这种情况下，应预先将运算表达式的操作数转换到左边的数据类型之后再行运算。

```
varDint2 := INT_TO_DINT(varInt1); //将INT型变量转换为DINT型变量  
varDint1 := varDint2 * 10; //以DINT型进行乘法运算, 代入运算结果
```

■在算术运算式中使用符号取反运算符的情况

对数据类型的最小值，使用符号取反运算符(-)时，将变为相同的值。

例如，INT型最小值的情况下，变为-(-32768)=-32768。

因此，在数据类型的自动转换的对象变量中使用符号取反运算符时，可能无法变为希望的结果。

例

varInt1(INT型)的值为-32768、varDint1(DINT型)的值为0的情况

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

此时，(-varInt1)的值将保持-32768不变，varDint2中将被代入-32768。

在算术运算式中使用符号取反运算符的情况下，应预先在算术运算前进行数据类型的自动转换或创建不使用符号取反运算符的程序。

例

算术运算之前进行数据类型自动转换的情况

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

例

不使用符号取反运算符的情况

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

■从单精度实数转换成双精度实数数据型的情况

从单精度实数转换成双精度实数型(REAL_TO_LREAL)时，转换结果可能会发生误差。

因此，在进行数据类型自动转换、在代入语句的右边或算术运算式的操作数中使用了实数型函数(SIN等)的返回值的的情况下，可能无法变为希望的结果。

例

发生误差的情况

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

上述情况下，ABS(varReal1)的返回值变为单精度实数，由于将该值型转换为双精度实数后代入varLReal1中，因此发生误差。这种情况下，应使用与代入目标相同的数据类型(双精度实数)创建执行函数的程序。

例

不发生误差的情况


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

■使用位型标签时

选择语句或重复语句中布尔表达式(条件表达式)一旦成立, 将<语句>内的位型标签置为ON状态时, 则该位型标签将变为常时ON。

例


常时ON的程序

ST程序	ST程序同等处理的梯形图程序
<pre>IF bLabel1 THEN bLabel2 := TRUE; END_IF;</pre>	

为避免常时ON, 应按下述方式对将位型标签置为OFF的程序进行添加。

例

避免常时ON的程序

ST程序*1	ST程序同等处理的梯形图程序
<pre>IF bLabel1 THEN bLabel2 := TRUE; ELSE bLabel2 := FALSE; END_IF;</pre>	

*1 上述程序可以按下述方式记述。

bLabel2:=bLabel1;

或

OUT(bLabel1, bLabel2);

但是, 在<语句>内使用了OUT指令的情况下, 变为与常时ON程序同样的状态。

■使用定时器FB、计数器FB时

选择语句中的布尔表达式(条件表达式), 与定时器FB、计数器FB的执行条件不同。

例

定时器FB的情况

更改前程序示例

```
IF bLabel1 THEN
  TIMER_100_FB_M_1(Coil:= bLabel2, Preset:= wLabel3, ValueIn:= wLabel4, ValueOut=> wLabel5, Status=> bLabel6);
END_IF;
(* bLabel1=ON且bLabel2=ON时, 开始计数。*)
(* bLabel1=ON且bLabel2=OFF时, 清除计数。*)
(* bLabel1=OFF且bLabel2=ON时, 停止计数。不清除计数值。*)
(* bLabel1=OFF且bLabel2=OFF时, 停止计数。不清除计数值。*)
```

更改后的程序示例

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

例

计数器FB的情况

更改前程序示例

```
IF bLabel1 THEN
  COUNTER_FB_M_1(Coil:= bLabel2, Preset:= wLabel3, ValueIn:= wLabel4, ValueOut=> wLabel5, Status=> bLabel6);
END_IF;
(* bLabel1=ON且bLabel2=ON/OFF时, 将计数+1。*)
(* bLabel1=OFF且bLabel2=ON/OFF时, 不进行计数。*)
(* bLabel1=ON/OFF与计数+1不联动。*)
```

更改后的程序示例

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

上述更改前的程序示例是在选择语句不成立的情况下, 为了不执行与定时器、计数器相关联的语句而创建的。

通过bLabel1条件与bLabel1的AND条件, 使定时器、计数器动作的情况下, 不使用控制语句, 仅使用FB。

通过使用更改后的程序, 可以使定时器、计数器动作。

■使用FOR...DO语句时

- 无法在重复自变量中使用结构体构件及数组要素。
- 应让重复自变量中使用的类型与<最终值的表达式>、<增加表达式>的类型一致。
- <增加表达式>可以省略。省略的情况下<增加表达式>作为1执行。
- 如果向<增加表达式>中代入0，则FOR语法以下可能不被执行或变为无限循环。
- FOR...DO语法中，执行FOR语法中的<语句...>后进行重复变量的计数处理。执行了超过重复变量的数据类型的最大值或低于最小值的计数处理的情况下，发生无限循环。

■使用上升沿指令、下降沿指令时

- IF语句及CASE语句的执行条件变为FALSE或CASE不一致的情况下，上次扫描时的IF语句及CASE语句的上升沿指令、下降沿指令的执行结果不进行初始化，在下次执行时将被沿用。
- 重复语句(FOR语句、WHILE语句或REPEAT语句)结束后，上次扫描时重复语句中的上升沿指令、下降沿指令的执行结果不进行初始化，在下次执行时将被沿用。
- 在IF语句及CASE语句中使用上升沿指令、下降沿指令时的动作如下所示。

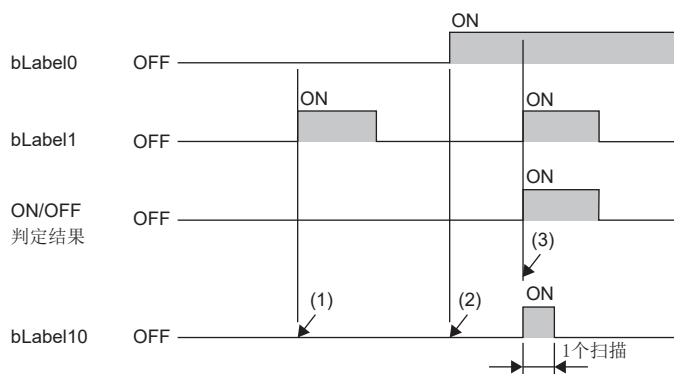
条件			动作结果		
IF语句、CASE语句的条件表达式	指令执行条件(EN)	上一次扫描时指令ON/OFF判定结果	指令ON/OFF判定结果	上升沿指令	下降沿指令
TRUE或CASE一致	TRUE	ON	ON	不执行	不执行
		OFF	ON	执行	不执行
	FALSE	ON	OFF	不执行	执行
		OFF	OFF	不执行	不执行
FALSE或CASE不一致	TRUE	ON	OFF	不执行	不执行*1
		OFF	OFF	不执行	不执行
	FALSE	ON	OFF	不执行	不执行*1
		OFF	OFF	不执行	不执行

*1 虽然为下降沿(ON→OFF)，但是由于IF或CASE语句条件不成立，指令不执行。

例

在IF语句中使用了PLS指令(上升沿执行)的情况

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



- (1) bLabel0 = OFF的情况下(IF语句的条件表达式为FALSE)，ON/OFF判定结果变为OFF，不执行PLS指令。(保持bLabel10 = OFF不变)
- (2) bLabel0 = ON(IF语句的条件表达式为TRUE)且，bLabel1 = OFF(指令执行条件为OFF)的情况下，ON/OFF判定结果变为OFF，不执行PLS指令。(保持bLabel10 = OFF不变)
- (3) bLabel0 = ON(IF语句的条件表达式为TRUE)且bLabel1 = ON(指令执行条件为ON)的情况下，ON/OFF判定结果变为OFF→ON(上升沿条件成立)，执行PLS指令。(bLabel10仅1个扫描ON)

- 以重复语句(FOR语句、WHILE语句或REPEAT语句), 执行上升沿指令或下降沿指令的情况下, 使用变址继电器(V)以及变址修饰。此时, 对每个使用了变址继电器(V)指令, 在系统中使用1点变址继电器(V)。因此, 在重复语句中加入使用的点数, 确保正在使用的指令数的变址继电器(V)。

例

在FOR语句中使用了上升沿指令及下降沿指令的情况

在1个地方使用变址继电器(V)的示例

(最多可使用的变址继电器(V)合计为11点(INC指令中V0~V10)。)

```
FOR ZO := 0 TO 9 BY 1 DO
  INC(EGP(M100ZO, V0ZO), D100ZO);
END_FOR;
```

在2个地方使用变址继电器(V)的示例

(最多可使用的变址继电器(V)合计为22点(INC指令中V0~V10、DEC指令中V11~V21)。)

```
FOR ZO := 0 TO 9 BY 1 DO
  INC(EGP(M100ZO, V0ZO), D100ZO);
  DEC(EGF(M200ZO, V11ZO), D200ZO);
END_FOR;
```

■使用主控制指令时

主控制OFF时的动作如下所示。

- 选择语句(IF语句或CASE语句)或重复语句(FOR语句、WHILE语句或REPEAT语句)中的语句变为无处理。
- 选择语句或重复语句之外的情况下, 代入语句变为无处理、代入语句以外的语句变为非执行处理。

例

选择语句(IF语句)中的语句

```
MC(M0, N1, M1); //主控制OFF
IF M2 THEN
  M3 := M4; //主控制OFF时为无处理, 因此M3保持之前扫描时的值
END_IF;
M20 := MCR(M0, N1);
```

例

选择语句或重复语句之外的语句(位代入语句的情况)

```
MC(M0, N1, M1); //主控制OFF
M3 := M4; //主控制OFF时为无处理, 因此M3保持之前扫描时的值
M20 := MCR(M0, N1);
```

例

选择语句或重复语句之外的语句(OUT指令的情况)

```
MC(M0, N1, M1); //主控制OFF
OUT(M2, M3); //主控制OFF时为非执行处理, 因此将M3置为OFF
M20 := MCR(M0, N1);
```

常数的标记方法


在ST程序中，除了可用普通的常数标记外，还可用下述常数标记。

可对应的数据类型	类型	标记方法*1	标记示例
位	引导	在使用的引导值前添加“BOOL#”。	BOOL#1、BOOL#0 BOOL#TRUE、BOOL#FALSE
字[无符号]/位串[16位]	2进制数	在2进制数前添加“UINT#2#”或“WORD#2#”。*2	UINT#2#10101010 WORD#2#10101010
	8进制数	在8进制数前添加“UINT#8#”或“WORD#8#”。*2	UINT#8#3370 WORD#8#3370
	10进制数	在10进制数前添加“UINT#”或“WORD#”。*2	UINT#123 WORD#123
	16进制数	在16进制数前添加“UINT#16#”或“WORD#16#”。*2	UINT#16#FF WORD#16#FF
双字[无符号]/位串[32位]	2进制数	在2进制数前添加“UDINT#2#”或“DWORD#2#”。*2	UDINT#2#1100110011001100 DWORD#2#1100110011001100
	8进制数	在8进制数前添加“UDINT#8#”或“DWORD#8#”。*2	UDINT#8#33703370 DWORD#8#33703370
	10进制数	在10进制数前添加“UDINT#”或“DWORD#”。*2	UDINT#456789 DWORD#456789
	16进制数	在16进制数前添加“UDINT#16#”或“DWORD#16#”。*2	UDINT#16#FFFF DWORD#16#FFFF
字[带符号]	2进制数	在2进制数前添加“INT#2#”。	INT#2#01010101
	8进制数	在8进制数前添加“INT#8#”。	INT#8#3370
	10进制数	在10进制数前添加“INT#”。	INT#-123
	16进制数	在16进制数前添加“INT#16#”。	INT#16#1F
双字[带符号]	2进制数	在2进制数前添加“DINT#2#”。	DINT#2#0011001100110011
	8进制数	在8进制数前添加“DINT#8#”。	DINT#8#33703370
	10进制数	在10进制数前附上“DINT#”。	DINT#-456789
	16进制数	在16进制数前添加“DINT#16#”。	DINT#16#1F1F
单精度实数	实数	在实数前添加“REAL#”。	REAL#2.34
	实数(指数表现)		REAL#1.OE6
双精度实数	实数	在实数前添加“LREAL#”。	LREAL#-2.34
	实数(指数表现)		LREAL#1.001E16
字符串	STRING	用单引号(')括住字符串(ASCII、移位JIS)。	'ABC'
字符串[Unicode]	WSTRING	用双引号(")括住Unicode字符串。	"ABC"

*1 大小写无区别。此外，不可与使用K、H、E的常数标记同时使用。

*2 ANY_NUM算术运算符的操作数、函数调用语句、FB调用语句、函数调用方式的自变量的情况下，应常数标记为“UINT#”或“UDINT#”。
常数标记为“WORD#”或“DWORD#”的情况下，将被认为是位串，转换时会发生错误。

上述以外的常数标记方法，请参阅下述手册。

 所使用的控制器的用户手册

要点

2进制数、8进制数、10进制数、16进制数、实数的标记时，可使用下划线(_)划分数值使程序更易读。例如，双字[无符号]的2进制标记时，可记载如下。

```
UDINT#2#1100_1100_1100_1100
```


程序处理时，下划线(_)的数值划分将被忽略。

标签与软元件

指定方法

在ST程序中可以直接记述并使用标签与软元件。标签与软元件可以在表达式的左边、右边、通用函数/FB的自变量、返回值等中使用。

标签与软元件的详细内容，请参阅下述手册。

所使用的控制器的用户手册

■带类型指定软元件标记

通过在软元件名中添加软元件型指定符，可将字软元件作为任意数据类型使用。

软元件型指定符	数据类型	示例	示例的说明
无	总称数据类型ANY16 在算术运算式中只使用软元件的情况下，变为字[带符号]。 但是，在FUN/FB的自变量部分被指定作为无类型指定的软元件的情况下，变为自变量定义的数据类型。	D0	D0中不附加类型指定的情况
:U	字[无符号]/位串[16位]	D0:U	将D0作为字[无符号]/位串[16位]的值
:D	双字[带符号]	D0:D	将D0、D1作为双字[带符号]的值
:UD	双字[无符号]/位串[32位]	D0:UD	将D0、D1作为双字[无符号]/位串[32位]的值
:E	单精度实数	D0:E	将D0、D1作为单精度实数的值
:ED	双精度实数	D0:ED	将D0、D1、D2、D3作为双精度实数的值




软元件类型指定可使用的软元件如下所示。

- 数据寄存器(D)
- 链接寄存器(W)
- 链接直接软元件(J□\W□)
- 模块访问软元件(U□\G□)
- CPU缓冲存储器访问软元件(U3E□\G□)
- 文件寄存器(R/ZR)
- 刷新数据寄存器(RD)


位数指定或间接指定的软元件中不能赋予软元件型指定符。

■软元件的指定方法

关于软元件的指定可以使用下述方法。

- 变址修饰
- 间接指定
- 位指定( 31页 位数据)
- 位数指定( 32页 16位数据(字数据)、 34页 32位数据(双字数据))

关于变址修饰和间接指定的详细内容，请参阅下述手册。

所使用的控制器的用户手册

注意事项

- 在ST程序中无法使用指针型。
- 用当前值使用定时器、计数器、累计定时器软元件时的数据类型为字[无符号]/位串[16位]。用当前值使用长定时器、长计数器、长累计定时器软元件时的数据类型为双字[无符号]/位串[32位]。
- 使用位数指定代入的情况下，应使右边和左边的数据类型一致。

例

D0:=K5X40;

在上述情况下，K5X40为双字型、D0变为字型，因此程序变为错误。

- 使用位数指定代入的情况下，右边>左边时，数据将传送至左边的对象点数范围内。

例

K5X40:=2#1011_1101_1111_0111_0011_0001;

在上述情况下，K5X40的对象点数20点，因此向K5X40代入1101_1111_0111_0011_0001(20位)。

- 以字[无符号]/位串[16位]以外的类型使用计数器(C)、定时器(T)、累计定时器(ST)的当前值(TNn等)，或以双字[无符号]/位串[32位]以外的类型使用长计数器(LC)、长定时器(LT)、长累计定时器(LST)的当前值(LTNn等)的情况下，应使用类型转换函数。

例

varInt:=WORD_TO_INT(T0);(*使用类型转换函数*)

■使用软元件时的数据类型自动转换

以字[带符号]以外的数据类型使用字软元件的情况下，应赋予软元件类型指定符。(☞ 104页 带类型指定软元件标记)

例

将D2、D3的值传送至双字[无符号]的标签dwordLabel1的情况

//赋予软元件类型指定符，以正确的数据类型传送的示例

```
dwordLabel1 := D2:UD;
```

//赋予软元件类型指定符的D2:UD为双字[无符号]，因此会将D2、D3的值传送至dwordLabel1。

//出现意料外的传送结果的示例

```
dwordLabel1 := D2;
```

//没有软元件类型指定符的D2为字[带符号]，因此会将数据类型自动转换为双字[无符号]后，传送至dwordLabel1。

//因此，仅传送D2的值，不传送D3的值。

注释

可以在ST程序中使用的注释如下所示。

注释形式	注释符号	内容	记述示例
单一行注释	//	将从开始符号“//”到行尾的内容作为注释。	//注释内容
多行注释	(**)	将从开始符号“(* ”起到结束符号“(*)”为止的内容作为注释处理。 可以在注释中输入换行。	■无换行 (*注释内容*) ■有换行 (*第1行注释内容 第2行注释内容*)
	/**/	将从开始符号“/*”起到结束符号“*/”为止的内容作为注释处理。 可以在注释中输入换行。	■无换行 /*注释内容*/ ■有换行 /*第1行注释内容 第2行注释内容*/

在多行注释中请勿记述含有结束符号的注释。

是通过按照数据及信号的流向对特定处理的块、变量部件、常数部件进行连接，记述程序的图表语言。

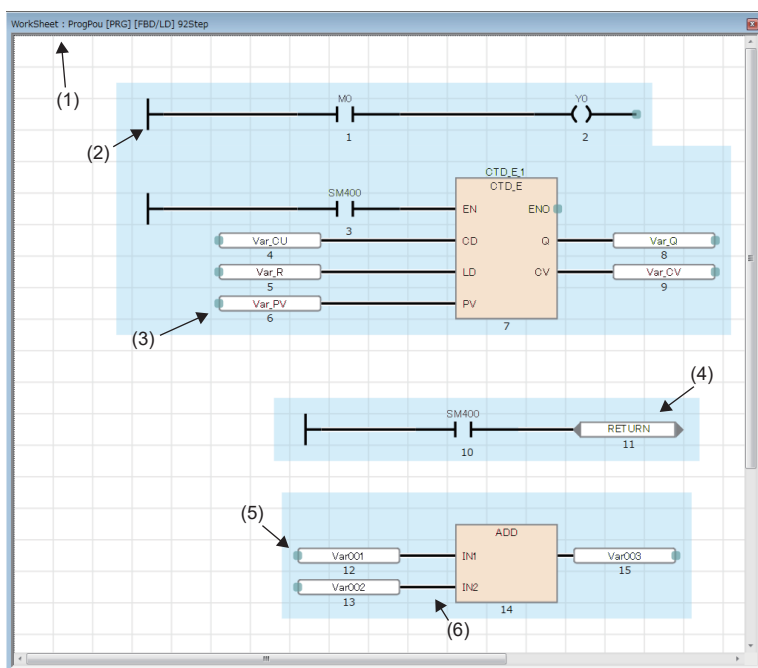
要点

- 在本章中，对FBD/LD语言的动作及规格有关内容进行说明。关于创建FBD/LD程序时的操作方法，请参阅下述手册。

📖 GX Works3 操作手册

5.1 配置

FBD/LD语言中，可创建以下程序。



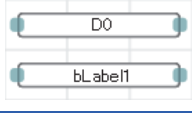
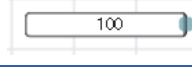
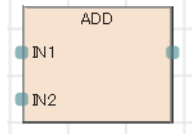
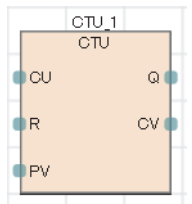
- (1) 工作表
- (2) LD部件
- (3) FBD部件
- (4) 通用部件
- (5) 连接点
- (6) 连接线

在FBD/LD语言的程序中，数据从功能块(FB)、函数(FUN)、变量部件(标签与软元件)、常数部件的输出点流至其它FB及变量部件等的输入点。

部件

FBD部件

配置FBD/LD程序的FBD部件如下所示。

项目		内容
变量		用于存储各个值(数据)时使用变量。变量中规定数据类型, 仅存储该数据类型的值(数据)。在变量中, 可以指定标签或软元件。
常数		输出所指定的常数值。
函数(FUN)		执行函数。 <ul style="list-style-type: none"> 函数的创建方法(□□GX Works3 操作手册) 通用功能(☞ 177页 通用函数)
功能块(FB)		执行FB。 <ul style="list-style-type: none"> FB的创建方法(□□GX Works3 操作手册) 通用FB(☞ 188页 通用FB) 运动控制FB(☞ 190页 运动控制FB一览) 模块FB(□□所使用的模块的FB参考)

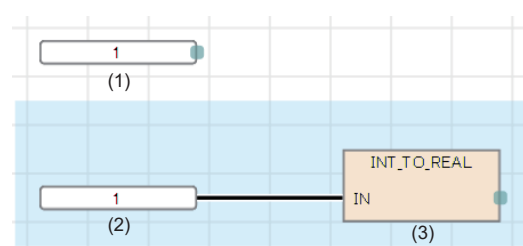
■常数部件的数据类型

常数部件的情况下, 在输入常数值时尚未决定常数值的数据类型。用连接线将常数部件与FBD部件连接在一起时, 决定数据类型。常数值的数据类型变为与用连接线连接的目标的FBD部件相同的数据类型。

例

在常数值中输入1的情况

由于数据类型后补中存在BOOL型、WORD型、DWORD型、INT型、DINT型、REAL型、LREAL型, 因此无法决定数据类型。用连接线将常数部件与FBD部件连接时, 将变为连接目标中的部件输入点的数据类型。



- (1) 数据类型未决定
- (2) INT型
- (3) INT型

■数据类型的自动转换

连接的部件的数据类型不同时, 可能会自动转换数据类型。

类型转换时, 只从容量小的数据类型转换至容量大的数据类型, 以确保不丢失数据。FBD/LD语言中数据类型的自动转换的动作与ST语言相同。详细内容, 请参阅下述章节。

☞ 95页 数据类型的自动转换

■函数的输入输出点

- 需要预先将函数的所有输入点与其它部件连接。
- 函数的输入变量与输出变量的数据类型确定后，连接至输入点与输出点的部件也需与这些数据类型相符。

■带EN的函数、带EN的FB的输出变量与其它部件连接时的注意事项

带EN的函数及带EN的FB的ENO为FALSE(运算停止)时，根据连接部件的不同，程序动作也有所不同。

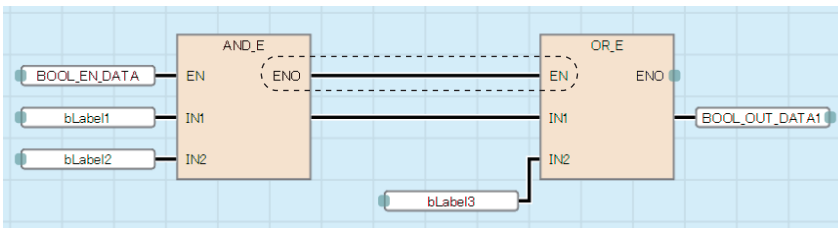
连接部件	带EN的函数及带EN的FB的ENO为FALSE(运算停止)时的程序动作
<ul style="list-style-type: none"> • 函数 • FB 	连接至带EN的函数及带EN的FB的输出变量的输入变量值变为不定值。
线圈	连接至带EN函数及带EN的FB的输出变量的线圈值为FALSE。此外，取反型线圈的情况下为TRUE。
上述以外	连接至带EN的函数及带EN的FB的输出变量的部件值不被更改。(变为前次值。)

因此，在将带EN的FB的输出变量直接连接至另外的函数或FB时，可能发生预想之外的动作。

为防止使用不定值，应通过下述示例中的任意一种方法进行连接。

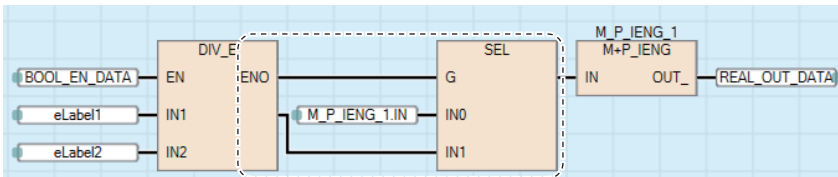
例

将连接目标作为带EN的函数或带EN的FB，对连接源的ENO与连接目标的EN进行连接。










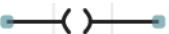
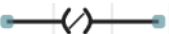
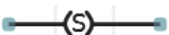

例

使用选择函数SEL，在ENO变为FALSE时使之输入前次值。



LD部件

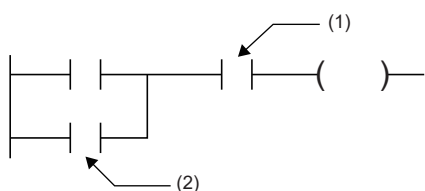
在FBD/LD程序中可使用的梯形图部件如下所示。

项目		内容
左母线		是表示母线的部件。变为创建梯形图时的起点。 左母线的输出将变为常时ON。
常开触点		指定软元件或标签变为ON时导通。
常闭触点		指定软元件或标签变为OFF时导通。
上升沿脉冲		指定软元件或标签上升沿时 (OFF→ON) 导通。
下降沿脉冲		指定软元件或标签下降沿时 (ON→OFF) 导通。
上升沿脉冲否定		指定软元件或标签OFF时、ON时以及下降沿时 (ON→OFF) 导通。
下降沿脉冲否定		指定软元件或标签OFF时、ON时以及上升沿时 (OFF→ON) 导通。
线圈		将运算结果输出至指定软元件或标签。
取反型线圈		运算结果变为OFF时，指定软元件或标签变为ON。
设置线圈		运算结果变为ON时，指定软元件或标签变为ON。 即使运算结果变为OFF，已ON的软元件或标签也将保持ON状态不变。
复位线圈		运算结果变为ON时，指定软元件或标签变为OFF。 运算结果变为OFF的情况下，软元件或标签的状态不变化。

■触点符号的AND运算与OR运算

触点符号，根据梯形图的连接状态进行AND运算、OR运算，并反映至运算结果。

- 串联连接(1)的情况下，进行之前的运算结果与AND运算，并作为运算结果。
- 并联连接(2)的情况下，进行之前的运算结果与OR运算，并作为运算结果。



- (1) 串联连接的触点
- (2) 并联连接的触点






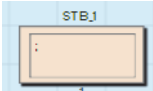
■将其它部件连接到线圈的输出连接点上的情况

将其它部件直接连接到线圈的输出连接点上时，动作将变为与并联连接时相同。

项目	内容
程序示例	
上述示例的动作	

通用部件

配置在FBD/LD程序上的通用部件如下所示。

项目		内容
跳转		从跳转部件开始到跳转标签为止跳转执行处理。不执行已跳转的部分。 根据至跳转部件的ON/OFF信息，控制是否进行跳转。 ON：在跳转标签中跳转执行处理。 OFF：不跳转，进行普通的执行处理。
跳转标签		为来自于同一程序内的跳转部件的跳转目标。进行了跳转的情况下，通过跳转标签及其以后的执行顺序的程序执行处理。
连接器		作为连接线的替代品使用。 处理将转移到成对的连接器部件中。 对一个输出连接器，可以使用一个或多个输入连接器。
返回		中断程序上的返回部件及其以后的处理。在不执行返回部件及其以后的程序、函数及FB处理的情况下使用。 根据返回至部件的ON/OFF信息，控制是否进行返回处理。 ON：执行返回处理。 OFF：不进行返回处理，进行普通的执行处理。
注释		记载注释的情况下使用。
内嵌ST		在FBD/LD编辑器内显示ST程序。 通过双击插入的内嵌ST部件，显示ST编辑器并进行ST程序的编辑、监视。 详细内容，请参阅下述内容。 ☞ 117页 内嵌ST

■跳转部件

- 在跳转部件中使线圈处于ON状态的定时器跳转时，将变得无法正常进行测量。
- 在跳转部件的上侧(执行顺序在前)可配置跳转标签。该情况下，应创建包含脱离循环方法的程序，以确保不超出看门狗定时器的设定值。
- 在跳转部件与跳转标签中仅可指定指针型的局部标签。而且不可以使用结构体的构件。
- 不可以使用指针分支指令(CJ、SCJ、JMP)。进行跳转的情况下，应使用跳转部件。
- 不可以跳转至程序块的外侧或从外侧跳转。

跳转关联的动作	执行可否
跳转至程序块的外侧*1	不可以执行
从程序块的外侧跳转*1	不可以执行
子程序的调用	可以执行
作为子程序被调用	不可以执行

*1 包含通过BREAK指令进行的分支。

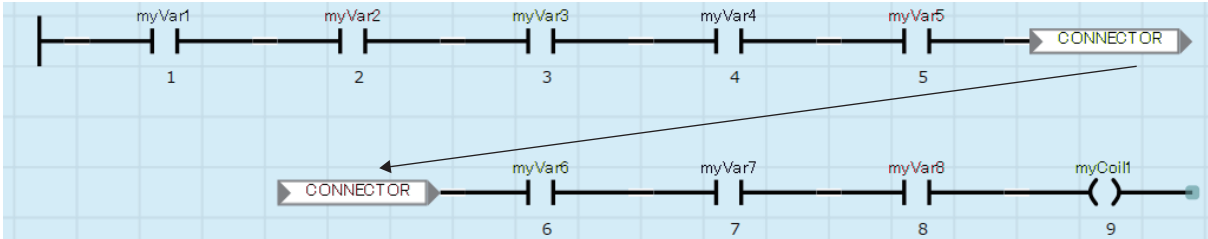
■返回部件

返回部件的动作根据使用的程序及函数、FB而有所不同。

所使用的程序部件	内容
程序	结束程序部件的执行。
函数	结束函数，返回至调用了函数指令的下一步。
FB	结束FB，返回至调用了FB指令的下一步。

■连接器部件

在FBD/LD编辑器的显示范围内或印刷范围内，希望配置程序的情况下使用连接器部件。



连接线



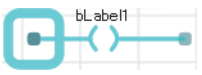



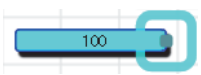
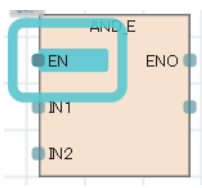
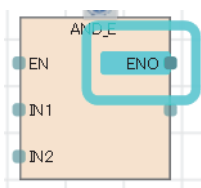
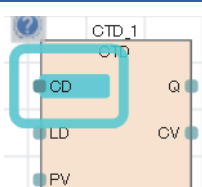
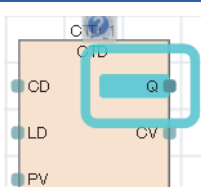
是连接FBD部件、LD部件、通用部件的连接点间的线。

连接部件从左端向右端交接数据。处于连接状态的部件的数据类型需要一致，或数据类型为可自动转换的类型。

连接点

是用连接线连接FBD部件、LD部件、通用部件时的端点。

各部件左侧点表示输入侧，右侧点表示输出侧。

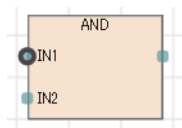
项目	输入连接点	输出连接点
触点		
线圈		
变量		
常数	—	
函数		 不显示函数的返回值名称。
FB		

连接点被连接时，将变为隐藏。

■输入输出点的取反

通过连接点可对至部件的输入或来自于部件的输出进行取反。

用黑圆圈括住取反状态的连接点，取反输入或输出的数据 (FALSE→TRUE或TRUE→FALSE)。



可取反的数据类型为BOOL、WORD、DWORD、ANY_BIT、ANY_BOOL。

常数的显示方法

FBD/LD程序中的字符串的显示方法如下所示。

数据类型		显示方法	显示示例
字符串	STRING	将字符串 (ASCII、移位JIS)用单引号(')括住。	
字符串 [Unicode]	WSTRING	用双引号(")括住Unicode字符串。	

上述以外常数的显示方法，请参阅下述手册。

所使用的控制器的用户手册

标签与软元件

指定方法

在FBD/LD程序上，可以直接记述且使用标签与软元件。可以对部件的输入点、输出点、通用函数/FB的自变量、返回值等使用标签与软元件。

标签与软元件的详细内容，请参阅下述手册。

所使用的控制器的用户手册

■带类型指定软元件标记

通过对软元件名附加软元件型指定符，可以以任意数据类型使用字软元件。

FBD/LD语言中的软元件型指定符与可使用的软元件与ST语言相同。详细内容，请参阅下述章节。

104页 带类型指定软元件标记

不指定字软元件的数据类型时，数据类型由软元件的类型决定。

字软元件	数据类型
定时器软元件的当前值 (TN)、累计定时器软元件的当前值 (STN)、计数器软元件的当前值 (CN)	WORD
长定时器软元件的当前值 (LTN)、长累计定时器软元件的当前值 (LSTN)、长计数器软元件的当前值 (LCN)	DWORD
长变址寄存器 (LZ)	DINT
上述以外	ANY16

注意事项

■使用标签的情况


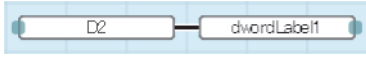
在数组的下标中，不能使用局部软元件。但是，将下标中使用的软元件代入其它软元件中、将代入目标的软元件指定为下标后，变为可使用。

■使用软元件时的数据类型自动转换

以字[带符号]以外的数据类型使用字软元件的情况下，应赋予软元件类型指定符。(☞ 115页 带类型指定软元件标记)

例

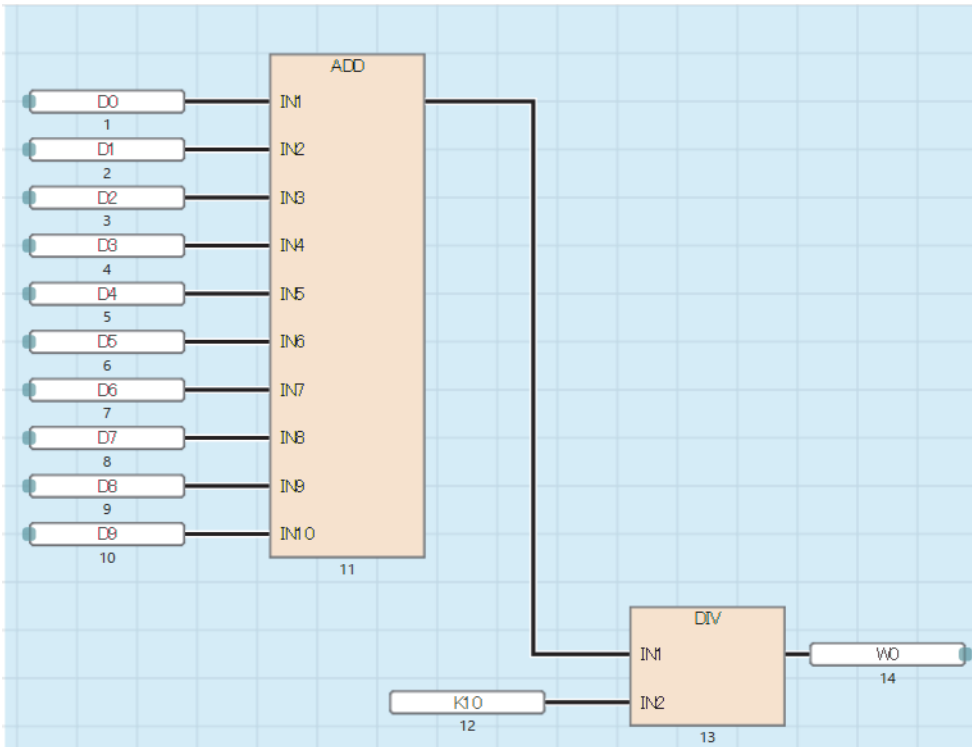
将D2、D3的值传送至双字[无符号]的标签dwordLabel1的情况

赋予软元件类型指定符，以正确的数据类型传送的示例	出现意料外的传送结果的示例
 <p>赋予了软元件类型指定符的D2：UD为双字[无符号]，因此会将D2、D3的值传送至dwordLabel1。</p>	 <p>没有软元件类型指定符的D2为字[带符号]，因此会在将数据类型自动转换为双字[无符号]后，传送至dwordLabel1。因此，仅传送D2的值，不传送D3的值。</p>

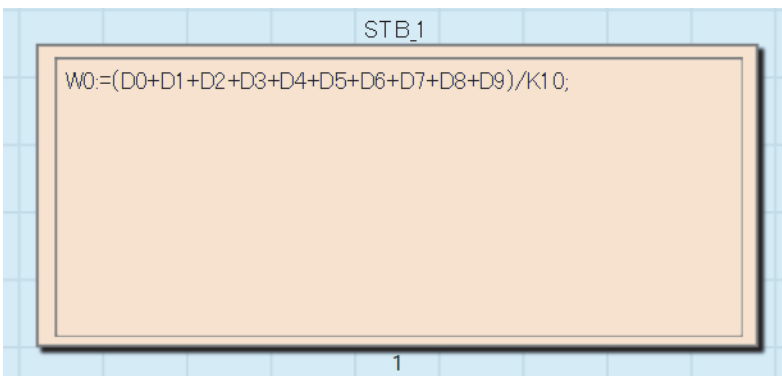
5.2 内嵌ST

内嵌ST是对在FBD/LD编辑器内显示ST程序的内嵌ST部件进行创建、编辑/监视的功能。由此，可以轻松地在FBD/LD程序内创建复杂的数值运算及字符串处理。

- 不使用内嵌ST情况下的程序



- 使用了内嵌ST情况下的程序



规格

双击插入的内嵌ST部件，显示ST编辑器并对ST程序进行编辑、监视。

关于内嵌ST中记述的ST程序的规格，请参阅ST语言的规格。

☞ 90页 ST语言

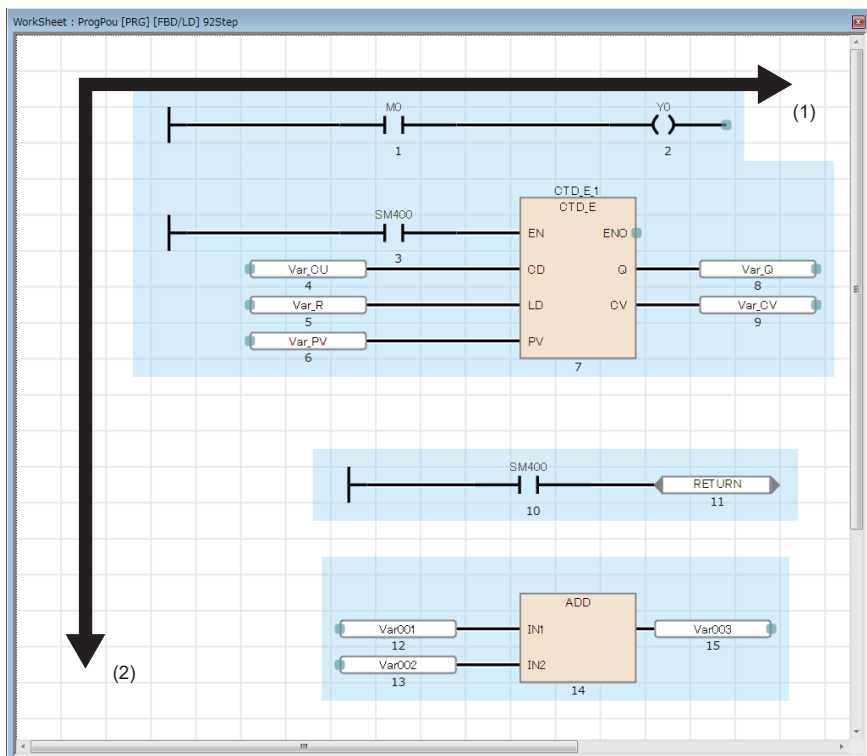
注意事项

- FBD/LD程序的一个程序部件内可插入的内嵌ST部件数最多为64个。
- 如果在内嵌ST内使用“RETURN语句”，不是结束程序块的处理，而是结束内嵌ST部件内的处理。
- 因FBD/LD程序中插入的内嵌ST部件没有连接点，所以始终执行。

5.3 程序执行顺序

部件的执行顺序

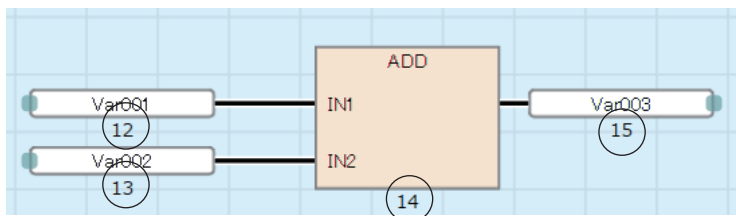
FBD/LD编辑器上部件的执行顺序依部件位置关系与连接状况而定。



(1) 从左至右执行。

(2) 从上至下执行。

配置在FBD/LD编辑器上的各部件显示执行顺序的编号。如果转换程序，则显示确定的执行顺序。

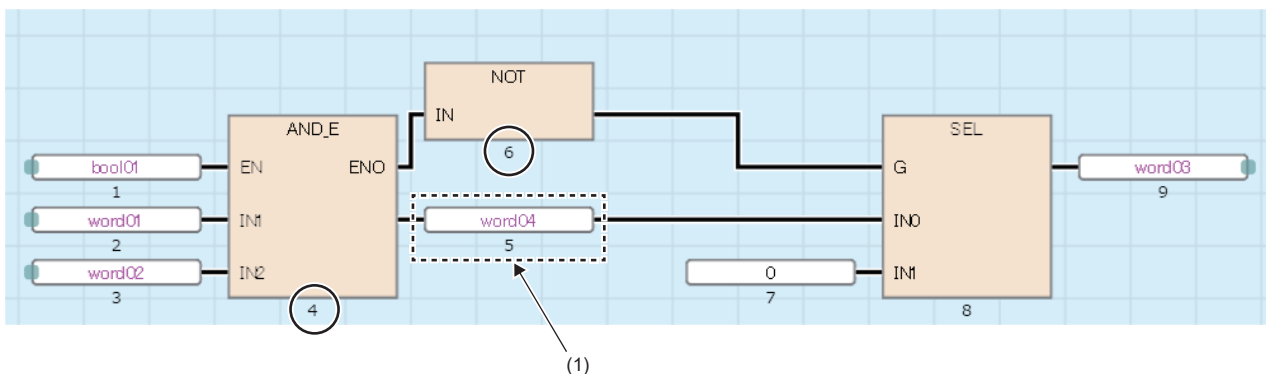


注意事项

如果使用函数的程序，请勿将连接函数的返回值与其他函数的输入变量直接连接，而是在之间连接变量部件。

例

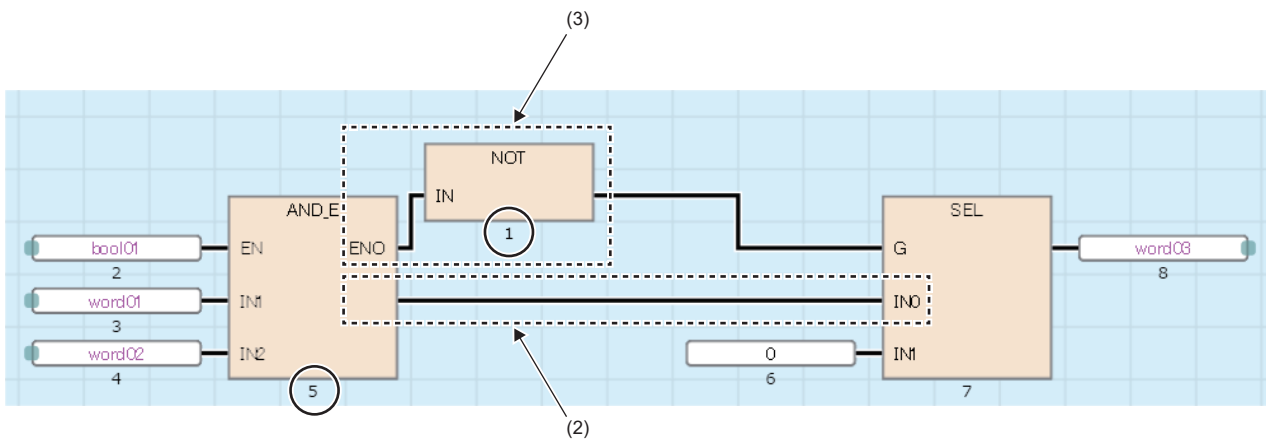
在返回值与输入变量之间连接了变量部件(1)的情况



如果直接连接函数的返回值与输入变量，可能不会变为预想内的执行顺序。

例

不变为预想内的执行顺序的情况



配置在左方的部件的返回值(2)与经由其他部件的输出变量(3)连接了配置在右方的部件的输入，所以执行顺序不同。

第2部分 指令/FUN/FB一览

本部分由下述章构成。

6 指令一览

7 网络指令一览

8 通用FUN/通用FB一览

9 运动控制FB一览

10 网络FB一览

6 指令一览

一览表在阅读方法如下所示。

项目	内容
指令符号	表示指令名。
处理内容	表示指令的概要。
参阅目标	表示详细说明的参阅目标。

6.1 顺控程序指令

触点指令

■运算开始、串联连接、并联连接

指令符号	处理内容	参阅目标
LD	将指定软元件的ON/OFF信息作为运算结果。(常开触点运算开始指令)	200页 LD、LDI、AND、ANI、OR、ORI
LDI	将指定软元件的ON/OFF信息作为运算结果。(常闭触点运算开始指令)	
AND	将指定软元件的ON/OFF信息与至当时为止的运算结果的AND运算作为运算结果。(常开触点串联连接指令)	
ANI	将指定软元件的ON/OFF信息与至当时为止的运算结果的AND运算作为运算结果。(常闭触点串联连接指令)	
OR	将指定软元件的ON/OFF信息与至当时为止的运算结果的OR运算作为运算结果。(1个常开触点的并联连接指令)	
ORI	将指定软元件的ON/OFF信息与至当时为止的运算结果的OR运算作为运算结果。(1个常闭触点的并联连接指令)	

■脉冲运算开始、脉冲串联连接、脉冲并联连接

指令符号	处理内容	参阅目标
LDP	仅指定软元件的上升沿时(OFF→ON)导通。(上升沿脉冲运算开始指令)	202页 LDP、LDF、ANDP、ANDF、ORP、ORF
LDF	仅指定软元件的下降沿时(ON→OFF)导通。(下降沿脉冲运算开始指令)	
ANDP	与至当时为止的运算结果进行AND运算。(上升沿脉冲串联连接指令)	
ANDF	与至当时为止的运算结果进行AND运算。(下降沿脉冲串联连接指令)	
ORP	与至当时为止的运算结果进行OR运算。(上升沿脉冲并联连接指令)	
ORF	与至当时为止的运算结果进行OR运算。(下降沿脉冲并联连接指令)	

■脉冲否定运算开始、脉冲否定串联连接、脉冲否定并联连接

指令符号	处理内容	参阅目标
LDPI	指定软元件的OFF时、ON时、下降沿时(ON→OFF)的情况下导通。(上升沿脉冲否定运算开始指令)	205页 LDPI、LDFI、ANDPI、ANDFI、ORPI、ORFI
LDFI	指定软元件的上升沿时(OFF→ON)、OFF时、ON时导通。(下降沿脉冲否定运算开始指令)	
ANDPI	与至当时为止的运算结果进行AND运算。(上升沿脉冲否定串联连接指令)	
ANDFI	与至当时为止的运算结果进行AND运算。(下降沿脉冲否定串联连接指令)	
ORPI	与至当时为止的运算结果进行OR运算。(上升沿脉冲否定并联连接指令)	
ORFI	与至当时为止的运算结果进行OR运算。(下降沿脉冲否定并联连接指令)	

合并指令

■梯形图块串联连接、并联连接

指令符号	处理内容	参阅目标
ANB	进行逻辑块之间的AND运算(逻辑块之间的串联连接)。	208页 ANB、ORB
ORB	进行逻辑块之间的OR运算(逻辑块之间的串联连接)。	

■运算结果取反

指令符号	处理内容	参阅目标
INV	对INV指令之前的运算结果进行取反。	209页 INV

■运算结果脉冲化

指令符号	处理内容	参阅目标
MEP	MEP指令之前的运算结果的上升沿时(OFF→ON)导通。	210页 MEP、MEF
MEF	MEF指令之前的运算结果的下降沿时(ON→OFF)导通。	

■变址继电器运算结果脉冲化

指令符号	处理内容	参阅目标
EGP	将EGP指令之前的运算结果通过变址继电器(V)进行存储。在运算结果的上升沿时(OFF→ON)导通。	211页 EGP、EGF
EGF	将EGF指令之前的运算结果通过变址继电器(V)进行存储。在运算结果的下降沿时(ON→OFF)导通。	

输出指令

■OUT(定时器、计数器、报警器除外)

指令符号	处理内容	参阅目标
OUT	将运算结果输出至指定的软元件。	213页 OUT

■定时器、长定时器

指令符号	处理内容	参阅目标
OUT T	OUT指令之前的运算结果为ON时，线圈ON后进行定时器的计测。如果时限到，常开触点将导通，常闭触点将变为非导通。 <ul style="list-style-type: none"> • OUT T：低速定时器指令 • OUTH T：高速定时器指令 • OUT ST：低速累计定时器指令 • OUTH ST：高速累计定时器指令 • OUT LT：低速长定时器指令 • OUT LST：低速长累计定时器指令 	215页 OUT T、OUTH T、OUT ST、OUTH ST
OUTH T		
OUT ST		
OUTH ST		
OUT LT		
OUT LST	218页 OUT LT、OUT LST	

■计数器、长计数器

指令符号	处理内容	参阅目标
OUT C	OUT指令之前的运算结果由OFF→ON变化时，计数器的当前值(计数值)将被+1。如果计数到，计数器的常开触点将导通，常闭触点将变为非导通。 <ul style="list-style-type: none"> • OUT C：计数器 • OUT LC：长计数器 	221页 OUT C
OUT LC		223页 OUT LC

■报警器

指令符号	处理内容	参阅目标
OUT F	将OUT F指令之前的运算结果输出至指定的报警器。	225页 OUT F

■软元件的设置(报警器除外)

指令符号	处理内容	参阅目标
SET	将指定位置为ON。	226页 SET

■软元件的复位(报警器除外)

指令符号	处理内容	参阅目标
RST	将指定软元件置为0(OFF)。对于定时器、计数器,将当前值设置为0,将触点/线圈置为OFF。	228页 RST

■报警器的设置、复位

指令符号	处理内容	参阅目标
SET F	将指定的报警器置为ON。	229页 SET F
RST F	将指定的报警器置为OFF。	231页 RST F

■上升沿、下降沿输出

指令符号	处理内容	参阅目标
PLS	PLS指令OFF→ON时使指定软元件1个扫描ON。	233页 PLS
PLF	PLF指令的ON→OFF时使指定软元件1个扫描ON。	235页 PLF

■位软元件输出取反

指令符号	处理内容	参阅目标
FF	将指定的软元件状态取反。	237页 FF

■直接输出的脉冲化

指令符号	处理内容	参阅目标
DELTA	对指定的直接访问输出(DY)进行脉冲输出。	238页 DELTA(P)
DELTAP		

移位指令

■位软元件移位

指令符号	处理内容	参阅目标
SFT	将1个小编号的软元件的ON/OFF状态移位至指定的软元件后,将1个小编号的软元件置为OFF。	240页 SFT(P)
SFTP		

主控制指令

■主控制的设置、复位

指令符号	处理内容	参阅目标
MC	开始主控制。	242页 MC、MCR
MCR	结束主控制。	

结束指令

■主程序结束

指令符号	处理内容	参阅目标
FEND	将主程序与子程序、中断程序分割时使用。	246页 FEND

■顺控程序结束

指令符号	处理内容	参阅目标
END	表示程序的最终。	247页 END

停止指令

■顺控程序停止

指令符号	处理内容	参阅目标
STOP	停止控制器的运算。(与将开关置为STOP侧的情况相同。)	249页 STOP

无处理指令

■无处理

指令符号	处理内容	参阅目标
NOPLF	在无处理的指令中，对至目前为止的运算不造成任何影响。	250页 NOPLF

6.2 基本指令

比较运算指令

■BIN16位数据比较

指令符号	处理内容	参阅目标
LD=、AND=、OR=	对指定的2个BIN16位数据通过常开触点处理进行比较。	251页 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位数据比较

指令符号	处理内容	参阅目标
LDD=、ANDD=、ORD=	对指定的2个BIN32位数据通过常开触点处理进行比较。	253页 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位块数据比较

指令符号	处理内容	参阅目标
BKCMP=、BKCMP<>、 BKCMP>、BKCMP<=、 BKCMP<、BKCMP>=	对指定的2个BIN16位块数据进行比较。	255页 BKCMP□(P) (_U)
BKCMP=P、BKCMP<>P、 BKCMP>P、BKCMP<=P、 BKCMP<P、BKCMP>=P		
BKCMP=_U、 BKCMP<>_U、 BKCMP>_U、 BKCMP<=_U、 BKCMP<_U、BKCMP>=_U		
BKCMP=P_U、 BKCMP<>P_U、 BKCMP>P_U、 BKCMP<=P_U、 BKCMP<P_U、 BKCMP>=P_U		

■BIN32位块数据比较

指令符号	处理内容	参阅目标
DBKCMPI=、DBKCMPI<>、 DBKCMPI>、DBKCMPI<=、 DBKCMPI<、DBKCMPI>=	对指定的2个BIN32位块数据进行比较。	258页 DBKCMPI□(P) (_U)
DBKCMPI=P、 DBKCMPI<>P、 DBKCMPI>P、 DBKCMPI<=P、 DBKCMPI<P、DBKCMPI>=P		
DBKCMPI=_U、 DBKCMPI<>_U、 DBKCMPI>_U、 DBKCMPI<=_U、 DBKCMPI<_U、 DBKCMPI>=_U		
DBKCMPI=P_U、 DBKCMPI<>P_U、 DBKCMPI>P_U、 DBKCMPI<=P_U、 DBKCMPI<P_U、 DBKCMPI>=P_U		

算术运算指令

■BIN16位加减法运算

指令符号	处理内容	参阅目标
+	对指定的2个BIN16位数据进行加法运算。(操作数为2个的情况下)	261页 +(P) (_U) [操作数为2个的情况下]
+P		
+_U		
+P_U		
+	对指定的2个BIN16位数据进行加法运算。(操作数为3个的情况下)	263页 +(P) (_U) [操作数为3个的情况下]
+P		
+_U		
+P_U		
-	对指定的2个BIN16位数据进行减法运算。(操作数为2个的情况下)	265页 -(P) (_U) [操作数为2个的情况下]
-P		
-_U		
-P_U		
-	对指定的2个BIN16位数据进行减法运算。(操作数为3个的情况下)	267页 -(P) (_U) [操作数为3个的情况下]
-P		
-_U		
-P_U		

■BIN32位加减法运算

指令符号	处理内容	参阅目标
D+	对指定的2个BIN32位数据进行加法运算。(操作数为2个的情况下)	269页 D+(P) (_U) [操作数为2个的情况下]
D+P		
D+_U		
D+P_U		
D+	对指定的2个BIN32位数据进行加法运算。(操作数为3个的情况下)	271页 D+(P) (_U) [操作数为3个的情况下]
D+P		
D+_U		
D+P_U		
D-	对指定的2个BIN32位数据进行减法运算。(操作数为2个的情况下)	273页 D-(P) (_U) [操作数为2个的情况下]
D-P		
D-_U		
D-P_U		
D-	对指定的2个BIN32位数据进行减法运算。(操作数为3个的情况下)	275页 D-(P) (_U) [操作数为3个的情况下]
D-P		
D-_U		
D-P_U		

■BIN16位乘除法运算

指令符号	处理内容	参阅目标
*	对指定的2个BIN16位数据进行乘法运算。	277页 *(P) (_U)
*P		
*_U		
*P_U		
/	对指定的2个BIN16位数据进行除法运算。	279页 /(P) (_U)
/P		
/_U		
/P_U		

■BIN32位乘除法运算

指令符号	处理内容	参阅目标
D*	对指定的2个BIN32位数据进行乘法运算。	281页 D*(P) (_U)
D*P		
D*_U		
D*P_U		
D/	对指定的2个BIN32位数据进行除法运算。	283页 D/(P) (_U)
D/P		
D/_U		
D/P_U		

■BCD4位加减法运算

指令符号	处理内容	参阅目标
B+	对指定的2个BCD4位数据进行加法运算。(操作数为2个的情况下)	285页 B+(P) [操作数为2个的情况下]
B+P		
B+	对指定的2个BCD4位数据进行加法运算。(操作数为3个的情况下)	286页 B+(P) [操作数为3个的情况下]
B+P		
B-	对指定的2个BCD4位数据进行减法运算。(操作数为2个的情况下)	288页 B-(P) [操作数为2个的情况下]
B-P		
B-	对指定的2个BCD4位数据进行减法运算。(操作数为3个的情况下)	289页 B-(P) [操作数为3个的情况下]
B-P		

■BCD8位加减法运算

指令符号	处理内容	参阅目标
DB+	对指定的2个BCD8位数据进行加法运算。(操作数为2个的情况下)	291页 DB+(P) [操作数为2个的情况下]
DB+P		
DB+	对指定的2个BCD8位数据进行加法运算。(操作数为3个的情况下)	292页 DB+(P) [操作数为3个的情况下]
DB+P		
DB-	对指定的2个BCD8位数据进行减法运算。(操作数为2个的情况下)	294页 DB-(P) [操作数为2个的情况下]
DB-P		
DB-	对指定的2个BCD8位数据进行减法运算。(操作数为3个的情况下)	295页 DB-(P) [操作数为3个的情况下]
DB-P		

■BCD4位乘除法运算

指令符号	处理内容	参阅目标
B*	对指定的2个BCD4位数据进行乘法运算。	297页 B*(P)
B*P		
B/	对指定的2个BCD4位数据进行除法运算。	299页 B/(P)
B/P		

■BCD8位乘除法运算

指令符号	处理内容	参阅目标
DB*	对指定的2个BCD8位数据进行乘法运算。	301页 DB*(P)
DB*P		
DB/	对指定的2个BCD8位数据进行除法运算。	303页 DB/(P)
DB/P		

■BIN16位块数据加减法运算

指令符号	处理内容	参阅目标
BK+	对指定的2个BIN16位数据块进行加法运算。	305页 BK+(P) (_U)
BK+P		
BK+_U		
BK+P_U		
BK-	对指定的2个BIN16位数据块进行减法运算。	307页 BK-(P) (_U)
BK-P		
BK-_U		
BK-P_U		

■BIN32位块数据加减法运算

指令符号	处理内容	参阅目标
DBK+	对指定的2个BIN32位数据块进行加法运算。	309页 DBK+(P) (_U)
DBK+P		
DBK+_U		
DBK+P_U		
DBK-	对指定的2个BIN32位数据块进行减法运算。	312页 DBK-(P) (_U)
DBK-P		
DBK-_U		
DBK-P_U		

■16位BIN数据增量、减量

指令符号	处理内容	参阅目标
INC	对指定的BIN16位数据进行+1。	315页 INC(P) (_U)
INCP		
INC_U		
INCP_U		
DEC	对指定的BIN16位数据进行-1。	316页 DEC(P) (_U)
DECP		
DEC_U		
DECP_U		

■32位BIN数据增量、减量

指令符号	处理内容	参阅目标
DINC	对指定的BIN32位数据进行+1。	317页 DINC(P) (_U)
DINCP		
DINC_U		
DINCP_U		
DDEC	对指定的BIN32位数据进行-1。	318页 DDEC(P) (_U)
DDECP		
DDEC_U		
DDECP_U		

逻辑运算指令

■16位/32位数据逻辑与

指令符号	处理内容	参阅目标
WAND	进行指定的2个BIN16位数据的逻辑与。(操作数为2个的情况下)	319页 WAND(P) [操作数为2个的情况下]
WANDP		
WAND	进行指定的2个BIN16位数据的逻辑与。(操作数为3个的情况下)	320页 WAND(P) [操作数为3个的情况下]
WANDP		
DAND	进行指定的2个BIN32位数据的逻辑与。(操作数为2个的情况下)	322页 DAND(P) [操作数为2个的情况下]
DANDP		
DAND	进行指定的2个BIN32位数据的逻辑与。(操作数为3个的情况下)	323页 DAND(P) [操作数为3个的情况下]
DANDP		

■16位块数据逻辑与

指令符号	处理内容	参阅目标
BKAND	对指定的2个BIN16位数据块的逻辑与。 	325页 BKAND(P)
BKANDP		

■16位/32位数据逻辑或

指令符号	处理内容	参阅目标
WOR	进行指定的2个BIN16位数据的逻辑或。(操作数为2个的情况下)	327页 WOR(P) [操作数为2个的情况下]
WORP		
WOR	进行指定的2个BIN16位数据的逻辑或。(操作数为3个的情况下)	328页 WOR(P) [操作数为3个的情况下]
WORP		
DOR	进行指定的2个BIN32位数据的逻辑或。(操作数为2个的情况下)	330页 DOR(P) [操作数为2个的情况下]
DORP		
DOR	进行指定的2个BIN32位数据的逻辑或。(操作数为3个的情况下)	331页 DOR(P) [操作数为3个的情况下]
DORP		

■16位块数据逻辑或

指令符号	处理内容	参阅目标
BKOR	进行指定的2个BIN16位数据块的逻辑或。 	333页 BKOR(P)
BKORP		

■16位/32位数据异或

指令符号	处理内容	参阅目标
WXOR	进行指定的2个BIN16位数据的异或。(操作数为2个的情况下)	335页 WXOR(P) [操作数为2个的情况下]
WXORP		
WXOR	进行指定的2个BIN16位数据的异或。(操作数为3个的情况下)	336页 WXOR(P) [操作数为3个的情况下]
WXORP		
DXOR	进行指定的2个BIN32位数据的异或。(操作数为2个的情况下)	338页 DXOR(P) [操作数为2个的情况下]
DXORP		
DXOR	进行指定的2个BIN32位数据的异或。(操作数为3个的情况下)	339页 DXOR(P) [操作数为3个的情况下]
DXORP		

■16位块数据异或

指令符号	处理内容	参阅目标
BKXOR	进行指定的2个BIN16位数据块的异或。	341页 BKXOR (P)
BKXORP		

■16位/32位数据异或非

指令符号	处理内容	参阅目标
WXNR	进行指定的2个BIN16位数据的异或非。(操作数为2个的情况下)	343页 WXNR (P) [操作数为2个的情况下]
WXNRP		
WXNR	进行指定的2个BIN16位数据的异或非。(操作数为3个的情况下)	344页 WXNR (P) [操作数为3个的情况下]
WXNRP		
DXNR	进行指定的2个BIN32位数据的异或非。(操作数为2个的情况下)	345页 DXNR (P) [操作数为2个的情况下]
DXNRP		
DXNR	进行指定的2个BIN32位数据的异或非。(操作数为3个的情况下)	346页 DXNR (P) [操作数为3个的情况下]
DXNRP		

■16位块数据异或非

指令符号	处理内容	参阅目标
BKXNR	进行指定的2个BIN16位数据块的异或非。	348页 BKXNR (P)
BKXNRP		

位处理指令

■字软元件的位设置/复位

指令符号	处理内容	参阅目标
BSET	将指定的字软元件内的第n位进行设置(1)。	350页 BSET (P)
BSETP		
BRST	将指定的字软元件内的第n位进行复位(0)。	352页 BRST (P)
BRSTP		

■位测试

指令符号	处理内容	参阅目标
TEST	提取指定的字软元件内的第n位。	353页 TEST (P)
TESTP		
DTEST	提取指定的2字软元件内的第n位。	355页 DTEST (P)
DTESTP		

■位软元件的批量复位

指令符号	处理内容	参阅目标
BKRST	将指定的位软元件开始的n点的位软元件进行复位。	357页 BKRST (P)
BKRSTP		

数据转换指令

■BIN数据→BCD4位/8位转换

指令符号	处理内容	参阅目标
BCD	将指定的BIN16位数据 (0~9999) 转换为BCD4位数据。	378页 BCD (P)
BCDP		
DBC	将指定的BIN32位数据 (0~99999999) 转换为BCD8位数据。	380页 DBC (P)
DBCDP		

■BCD4位/8位→BIN16位/32位数据转换

指令符号	处理内容	参阅目标
BIN	将指定的BCD4位数据 (0~9999) 转换为BIN16位数据。	382页 BIN (P)
BINP		
DBIN	将指定的BCD8位数据 (0~99999999) 转换为BIN32位数据。	384页 DBIN (P)
DBINP		

■单精度实数→带符号BIN16位/32位数据

指令符号	处理内容	参阅目标
FLT2INT	将指定的单精度实数 (-32768~32767) 转换为带符号BIN16位数据。	386页 FLT2INT (P)
FLT2INTP	 (1) 实数	
FLT2DINT	将指定的单精度实数 (-2147483648~2147483647) 转换为带符号BIN32位数据。	390页 FLT2DINT (P)
FLT2DINTP	 (1) 实数	

■单精度实数→无符号BIN16位/32位数据

指令符号	处理内容	参阅目标
FLT2UINT	将指定的单精度实数 (0~65535) 转换为无符号BIN16位数据。	388页 FLT2UINT (P)
FLT2UINTP	 (1) 实数	
FLT2UDINT	将指定的单精度实数 (0~4294967295) 转换为无符号BIN32位数据。	392页 FLT2UDINT (P)
FLT2UDINTP	 (1) 实数	

■双精度实数→带符号BIN16位/32位数据

指令符号	处理内容	参阅目标
DBL2INT DBL2INTP	将指定的双精度实数(-32768~32767)转换为带符号BIN16位数据。 $(s)+3, (s)+2, (s)+1, (s) \longrightarrow (d)$ ↑ (1) BIN	394页 DBL2INT(P)
DBL2DINT DBL2DINTP	将指定的双精度实数(-2147483648~2147483647)转换为带符号BIN32位数据。 $(s)+3, (s)+2, (s)+1, (s) \longrightarrow (d)+1, (d)$ ↑ (1) BIN	398页 DBL2DINT(P)

■双精度实数→无符号BIN16位/32位数据

指令符号	处理内容	参阅目标
DBL2UINT DBL2UINTP	将指定的双精度实数(0~65535)转换为无符号BIN16位数据。 $(s)+3, (s)+2, (s)+1, (s) \longrightarrow (d)$ ↑ (1) BIN	396页 DBL2UINT(P)
DBL2UDINT DBL2UDINTP	将指定的双精度实数(0~4294967295)转换为无符号BIN32位数据。 $(s)+3, (s)+2, (s)+1, (s) \longrightarrow (d)+1, (d)$ ↑ (1) BIN	400页 DBL2UDINT(P)

■带符号BIN16位数据→无符号BIN16位/32位数据转换

指令符号	处理内容	参阅目标
INT2UINT INT2UINTP	将(s)中指定的带符号BIN16位数据转换为无符号BIN16位数据后, 存储到(d)中。	402页 INT2UINT(P)
INT2UDINT INT2UDINTP	将(s)中指定的带符号BIN16位数据转换为无符号BIN32位数据后, 存储到(d)中。	406页 INT2UDINT(P)

■带符号BIN16位数据→带符号BIN32位数据转换

指令符号	处理内容	参阅目标
INT2DINT INT2DINTP	将(s)中指定的带符号BIN16位数据转换为带符号BIN32位数据后, 存储到(d)中。	404页 INT2DINT(P)

■无符号BIN16位数据→带符号BIN16位/32位数据转换

指令符号	处理内容	参阅目标
UINT2INT UINT2INTP	将(s)中指定的无符号BIN16位数据转换为带符号BIN16位数据后, 存储到(d)中。	408页 UINT2INT(P)
UINT2DINT UINT2DINTP	将(s)中指定的无符号BIN16位数据转换为带符号BIN32位数据后, 存储到(d)中。	410页 UINT2DINT(P)

■无符号BIN16位数据→无符号BIN32位数据转换

指令符号	处理内容	参阅目标
UINT2UDINT UINT2UDINTP	将(s)中指定的无符号BIN16位数据转换为无符号BIN32位数据后, 存储到(d)中。	412页 UINT2UDINT(P)

■带符号BIN32位数据→带符号BIN16位数据转换

指令符号	处理内容	参阅目标
DINT2INT	将(s)中指定的带符号BIN32位数据转换为带符号BIN16位数据后, 存储到(d)中。	414页 DINT2INT(P)
DINT2INTP		

■带符号BIN32位数据→无符号BIN16位/32位数据转换

指令符号	处理内容	参阅目标
DINT2UINT	将(s)中指定的带符号BIN32位数据转换为无符号BIN16位数据后, 存储到(d)中。	416页 DINT2UINT(P)
DINT2UINTP		
DINT2UDINT	将(s)中指定的带符号BIN32位数据转换为无符号BIN32位数据后, 存储到(d)中。	418页 DINT2UDINT(P)
DINT2UDINTP		


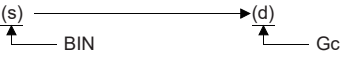


■无符号BIN32位数据→带符号BIN16位/32位数据转换

指令符号	处理内容	参阅目标
UDINT2INT	将(s)中指定的无符号BIN32位数据转换为带符号BIN16位数据后, 存储到(d)中。	420页 UDINT2INT(P)
UDINT2INTP		
UDINT2DINT	将(s)中指定的无符号BIN32位数据转换为带符号BIN32位数据后, 存储到(d)中。	424页 UDINT2DINT(P)
UDINT2DINTP		

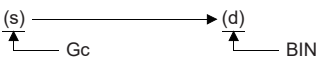
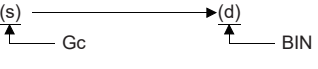
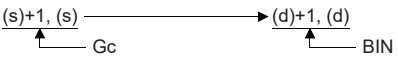

■无符号BIN32位数据→无符号BIN16位数据转换

指令符号	处理内容	参阅目标
UDINT2UINT	将(s)中指定的无符号BIN32位数据转换为无符号BIN16位数据后, 存储到(d)中。	422页 UDINT2UINT(P)
UDINT2UINTP		

■BIN16位/32位数据→BIN16位/32位格雷码数据转换

指令符号	处理内容	参阅目标
GRY	将指定的BIN16位数据(-32768~32767)转换为BIN16位格雷码数据。 	426页 GRY(P) (_U)
GRYP		
GRY_U	将指定的BIN16位数据(0~65535)转换为BIN16位格雷码数据。 	
GRYP_U		
DGRY	将指定的BIN32位数据(-2147483648~2147483647)转换为BIN32位格雷码数据。 	428页 DGRY(P) (_U)
DGRYP		
DGRY_U	将指定的BIN32位数据(0~4294967295)转换为BIN32位格雷码数据。 	
DGRYP_U		

■BIN16位/32位格雷码数据→BIN16位/32位数据转换

指令符号	处理内容	参阅目标
GBIN	将指定的BIN16位格雷码数据(-32768~32767)转换为BIN16位数据。	430页 GBIN(P) (_U)
GBINP	 <p>Gc: 格雷码</p>	
GBIN_U	将指定的BIN16位格雷码数据(0~65535)转换为BIN16位数据。	
GBINP_U	 <p>Gc: 格雷码</p>	
DGBIN	将指定的BIN32位格雷码数据(-2147483648~2147483647)转换为BIN32位数据。	432页 DGBIN(P) (_U)
DGBINP	 <p>Gc: 格雷码</p>	
DGBIN_U	将指定的BIN32位格雷码数据(0~4294967295)转换为BIN32位数据。	
DGBINP_U	 <p>Gc: 格雷码</p>	

■BIN16位数据块→BCD4位数据转换块

指令符号	处理内容	参阅目标
BKBCD	将从(s)开始的(n)点的BIN数据批量转换为BCD数据后, 存储到(d)及其以后。	434页 BKBCD(P)
BKBCDP		

■块BCD4位数据→块BIN16位转换数据

指令符号	处理内容	参阅目标
BKBIN	将从(s)开始的(n)点的BCD数据批量转换为BIN数据后, 存储到(d)及其以后。	436页 BKBIN(P)
BKBINP		

■10进制ASCII代码数据→BIN16位/32位数据转换

指令符号	处理内容	参阅目标
DABIN	将(s)中指定的10进制ASCII代码值转换为1字BIN值后, 存储到(d)中指定的字软元件编号中。	438页 DABIN(P) (_U)
DABINP		
DABIN_U		
DABINP_U		
DDABIN	将(s)中指定的10进制ASCII代码值转换为2字BIN值后, 存储到(d)中指定的字软元件编号中。	441页 DDABIN(P) (_U)
DDABINP		
DDABIN_U		
DDABINP_U		

■16进制ASCII代码数据→BIN16位/32位数据转换

指令符号	处理内容	参阅目标
HABIN	将(s)中指定的16进制ASCII代码值转换为1字BIN值后, 存储到(d)中指定的字软元件编号中。	445页 HABIN(P)
HABINP		
DHABIN	将(s)中指定的16进制ASCII代码值转换为2字BIN值后, 存储到(d)中指定的字软元件编号中。	447页 DHABIN(P)
DHABINP		

■10进制ASCII代码数据→BCD4位/8位转换

指令符号	处理内容	参阅目标
DABCD	将 (s) 中指定的10进制ASCII代码值转换为1字BCD值后, 存储到 (d) 中指定的字软元件编号中。	450页 DABCD (P)
DABCDP		
DDABCD	将 (s) 中指定的10进制ASCII代码值转换为2字BCD值后, 存储到 (d) 中指定的字软元件编号中。	453页 DDABCD (P)
DDABCDP		

■10进制字符串→BIN16位/32位数据转换

指令符号	处理内容	参阅目标
VAL	将 (s) 中指定包含小数点的字符串转换为1字BIN值及小数部位数后, 存储到 (d1)、(d2) 中指定的软元件中。	456页 VAL (P) (_U)
VALP		
VAL_U		
VALP_U		
DVAL	将 (s) 中指定包含小数点的字符串转换为2字BIN值及小数部位数后, 存储到 (d1)、(d2) 中指定的软元件中。	459页 DVAL (P) (_U)
DVALP		
DVAL_U		
DVALP_U		


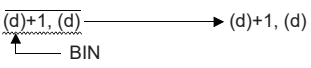
■16进制ASCII代码→16进制数据BIN转换

指令符号	处理内容	参阅目标
ASC2INT	将 (s) 中指定的字软元件及其以后的16进制ASCII代码数据以 (n) 中指定的字符数转换为BIN值后, 存储到 (d) 中指定的软元件编号及其以后。	462页 ASC2INT (P)
ASC2INTP		

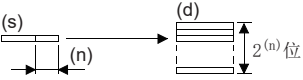
■单精度实数→BCD分解

指令符号	处理内容	参阅目标
EMOD	将 (s1) 的32位浮点数据转换为 (s2) 中指定的小数部位数的BCD后, 存储到 (d) 中指定的软元件中。	464页 EMOD (P)
EMODP		

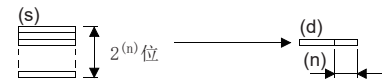
■BIN16位/32位数据2的补数(符号取反)

指令符号	处理内容	参阅目标
NEG	对BIN16位软元件的符号进行取反。 	466页 NEG (P)
NEGP		
DNEG	对BIN32位软元件的符号进行取反。 	467页 DNEG (P)
DNEGP		


■8→256位解码

指令符号	处理内容	参阅目标
DECO	对指定的软元件的低 (n) 位进行解码。 	468页 DECO (P)
DECOP		

■256→8位编码

指令符号	处理内容	参阅目标
ENCO	对2的 (n) 次方的数据进行编码。 	470页 ENCO (P)
ENCOP		

■7段解码

指令符号	处理内容	参阅目标
SEG	将软元件的低4位中指定的0~F的数据解码为7段显示数据。	472页 SEG(P)
SEGP		

■BIN16位数据的4位分离

指令符号	处理内容	参阅目标
DIS	将(s)中指定的16位数据以4位单位分离, 存储到(d)开始的(n)点的低4位中。(n<4)	474页 DIS(P)
DISP		

■BIN16位数据的4位合并

指令符号	处理内容	参阅目标
UNI	将(s)中指定的软元件开始的(n)点的低4位数据合并后, 存储到(d)中指定的软元件中。(n<4)	476页 UNI(P)
UNIP		

■任意数据的位分离、合并

指令符号	处理内容	参阅目标
NDIS	将(s1)中指定的软元件及其以后的数据分离为(s2)及其以后中指定的位后, 从(d)中指定的软元件开始依次存储。	478页 NDIS(P)
NDISP		
NUNI	将(s1)中指定的软元件及其以后的数据合并到(s2)及其以后中指定的各位中后, 从(d)中指定的软元件开始依次存储。	480页 NUNI(P)
NUNIP		

■字节单位数据分离、合并

指令符号	处理内容	参阅目标
WTOB	从(s)中指定的软元件开始, 将(n)点的16位数据分解为8位单位后, 从(d)中指定的软元件开始依次存储。	482页 WTOB(P)
WTOBP		
BTOW	从(s)中指定的软元件开始, 将(n)点的16位数据的低8位合并为16位后, 从(d)中指定的软元件开始依次存储。	484页 BTOW(P)
BTOWP		

数据移位指令

■BIN16位数据的n位右移、左移

指令符号	处理内容	参阅目标
SFR	将指定软元件的BIN16位数据右移。移位后的空余数据中存储0。	358页 SFR(P)
SFRP		
SFL	将指定软元件的BIN16位数据左移。移位后的空余数据中存储0。	360页 SFL(P)
SFLP		

■n位数据的1位右移、左移

指令符号	处理内容	参阅目标
BSFR	从指定的软元件开始的n点向右移1位。移位后的空余数据中存储0。	362页 BSFR(P)
BSFRP		
BSFL	从指定的软元件开始的n点向左移1位。移位后的空余数据中存储0。	364页 BSFL(P)
BSFLP		

■n字数据的1字右移、左移

指令符号	处理内容	参阅目标
DSFR	从指定的软元件开始将n点右移1字。移位后的空余数据中存储0。	366页 DSFR(P)
DSFRP		
DSFL	从指定的软元件开始将n点左移1字。移位后的空余数据中存储0。	368页 DSFL(P)
DSFLP		

■n位数据的n位右移、左移

指令符号	处理内容	参阅目标
SFTBR	从指定的软元件开始将n位的数据右移n位。移位后的空余数据中存储0。	370页 SFTBR(P)
SFTBRP		
SFTBL	从指定的软元件开始将n位的数据左移n位。移位后的空余数据中存储0。	372页 SFTBL(P)
SFTBLP		

■n字数据的n字右移、左移

指令符号	处理内容	参阅目标
SFTWR	从指定的软元件开始将n字的数据右移n字。移位后的空余数据中存储0。	374页 SFTWR(P)
SFTWRP		
SFTWL	从指定的软元件开始将n字的数据左移n字。移位后的空余数据中存储0。	376页 SFTWL(P)
SFTWLP		

数据传送指令

■BIN16位/32位数据传送

指令符号	处理内容	参阅目标
MOV	对指定的软元件的BIN16位数据进行传送。	486页 MOV (P)
MOVP	(s) \longrightarrow (d)	
DMOV	对指定的软元件的BIN32位数据进行传送。	488页 DMOV (P)
DMOV P	(s)+1, (s) \longrightarrow (d)+1, (d)	

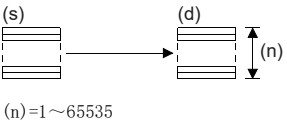
■BIN16位/32位数据否定传送

指令符号	处理内容	参阅目标
CML	将指定的BIN16位数据逐位取反后进行传送。	490页 CML (P)
CMLP	$\overline{(s)}$ \longrightarrow (d)	
DCML	将指定的BIN32位数据逐位取反后进行传送。	491页 DCML (P)
DCMLP	$\overline{(s)+1}, \overline{(s)}$ \longrightarrow (d)+1, (d)	

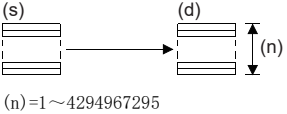
■1位数据取反传送

指令符号	处理内容	参阅目标
CMLB	将 (s) 中指定的位数据取反后, 存储到 (d) 中。	493页 CMLB (P)
CMLBP		

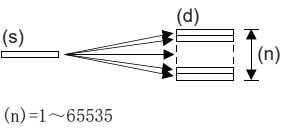
■BIN16位数据块传送(16位)

指令符号	处理内容	参阅目标
BMOV	对指定的软元件开始的 (n) 点的BIN16位数据进行批量传送。	494页 BMOV (P)
BMOV P	 (n)=1~65535	

■BIN16位数据块传送(32位)

指令符号	处理内容	参阅目标
BMOVL	对指定的软元件开始的 (n) 点的BIN16位数据进行批量传送。	496页 BMOVL (P)
BMOVLP	 (n)=1~4294967295	

■同一BIN16位数据块传送(16位)

指令符号	处理内容	参阅目标
FMOV	将BIN16位数据传送到指定的软元件开始的 (n) 点处。	498页 FMOV (P)
FMOV P	 (n)=1~65535	

■同一BIN16位数据块传送(32位)

指令符号	处理内容	参阅目标
FMOVL	将BIN16位数据传送到指定的软元件开始的(n)点处。	500页 FMOVL(P)
FMOVLP	<p>(n)=1~4294967295</p>	

■同一BIN32位数据块传送(16位)

指令符号	处理内容	参阅目标
DFMOV	将BIN32位数据传送到指定的软元件开始的(n)点处。	502页 DFMOV(P)
DFMOVLP	<p>(n)=1~65535</p>	

■同一BIN32位数据块传送(32位)

指令符号	处理内容	参阅目标
DFMOVL	将BIN32位数据传送到指定的软元件开始的(n)点处。	504页 DFMOVL(P)
DFMOVLP	<p>(n)=1~4294967295</p>	

■BIN16位/32位数据交换

指令符号	处理内容	参阅目标
XCH	对指定的BIN16位数据进行交换。	506页 XCH(P)
XCHP	(d1) ←→ (d2)	
DXCH	对指定的BIN32位数据进行交换。	507页 DXCH(P)
DXCHP	(d1)+1, (d1) ←→ (d2)+1, (d2)	

■BIN16位数据块交换

指令符号	处理内容	参阅目标
BXCH	对指定的软元件开始的(n)点的BIN16位数据进行交换。	508页 BXCH(P)
BXCHP		

■BIN16位数据高低字节交换

指令符号	处理内容	参阅目标
SWAP	对指定的软元件的高低各8位的值进行交换。	510页 SWAP(P)
SWAPP		

■BIN32位数据高低字节交换

指令符号	处理内容	参阅目标
DSWAP	对指定的软元件的高低各8位的值进行交换。	511页 DSWAP(P)
DSWAPP		

■1位数据传送

指令符号	处理内容	参阅目标
MOVB	将(s)中指定的位数据存储到(d)中。	512页 MOVBP
MOVBP		

■n位数据传送

指令符号	处理内容	参阅目标
BLKMOVB	将从(s)开始的(n)点的位数据批量传送到(d)开始的(n)点的位数据中。	513页 BLKMOVBP
BLKMOVBP		

6.3 应用指令

程序控制

程序分支指令

■指针分支

指令符号	处理内容	参阅目标
CJ	执行同一程序文件内指定指针编号的程序。	516页 CJ、SCJ、JMP
SCJ	从下一个扫描开始执行同一程序文件内指定指针编号的程序。	
JMP	无条件执行同一程序文件内指定指针编号的程序。	

■跳转至END

指令符号	处理内容	参阅目标
GOEND	跳转至同一程序文件内的FEND或END指令处。	519页 GOEND

程序执行控制指令

■中断禁止、中断允许

指令符号	处理内容	参阅目标
DI	禁止中断程序的执行。	520页 DI、EI
EI	解除中断禁止状态。	

■指定优先度及其以下的中断禁止

指令符号	处理内容	参阅目标
DI	禁止指定的优先度及其以下的中断程序的执行。	523页 DI

■中断程序屏蔽

指令符号	处理内容	参阅目标
IMASK	将指定的中断指针编号的中断程序置为允许执行状态/禁止执行状态。	528页 IMASK

■指定中断指针的禁止/允许

指令符号	处理内容	参阅目标
SIMASK	将指定的中断指针编号的中断程序置为允许执行状态/禁止执行状态。	530页 SIMASK

■从中断程序的恢复

指令符号	处理内容	参阅目标
IRET	表示中断程序处理的结束。	532页 IRET

■WDT复位

指令符号	处理内容	参阅目标
WDT	对看门狗定时器进行复位。	533页 WDT(P)
WDTP		

结构化指令

■FOR~NEXT

指令符号	处理内容	参阅目标
FOR	将FOR~NEXT指令之间的处理执行(n)次。	534页 FOR、NEXT
NEXT		

■FOR~NEXT强制结束

指令符号	处理内容	参阅目标
BREAK	强制结束通过FOR~NEXT指令进行的重复处理，执行将转移至FOR~NEXT指令及以后的处理。	537页 BREAK(P)
BREAKP		

■子程序调用

指令符号	处理内容	参阅目标
CALL	通过输入条件成立执行(P)的子程序。 (s1)~(s5)为至子程序的自变量。)	539页 CALL(P)
CALLP		

■从子程序返回

指令符号	处理内容	参阅目标
RET	表示子程序的结束。	545页 RET

■子程序的输出OFF调用

指令符号	处理内容	参阅目标
FCALL	输入条件不成立时进行(P)的子程序的非执行处理。 (s1)~(s5)为至子程序的自变量。)	546页 FCALL(P)
FCALLP		

■程序文件之间子程序调用

指令符号	处理内容	参阅目标
ECALL	通过输入条件成立执行指定程序的(P)的子程序。 (s1)~(s5)为至子程序的自变量。)	552页 ECALL(P)
ECALLP		

■程序文件之间子程序输出OFF调用

指令符号	处理内容	参阅目标
EFCALL	输入条件不成立时进行指定程序的(P)的子程序的非执行处理。(s1)~(s5)为至子程序的自变量。)	559页 EFCALL(P)
EFCALLP		

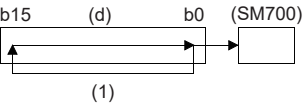
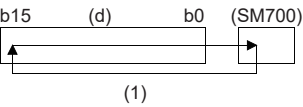
■子程序调用

指令符号	处理内容	参阅目标
XCALL	通过输入条件成立执行(P)的子程序。 输入条件不成立时进行(P)的子程序的非执行处理。 (s1)~(s5)为至子程序的自变量。)	564页 XCALL

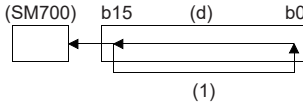
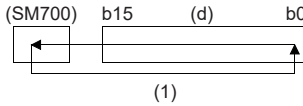
数据处理

旋转指令

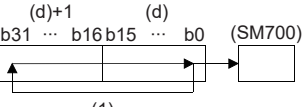
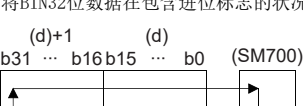
■BIN16位数据的右旋

指令符号	处理内容	参阅目标
ROR	将BIN16位数据在不包含进位标志的状况下右旋(n)位。	569页 ROR(P)、RCR(P)
RORP	 <p>(1)</p> <p>(1) (n)位右旋</p>	
RCR	将BIN16位数据在包含进位标志的状况下右旋(n)位。	
RCRP	 <p>(1)</p> <p>(1) (n)位右旋</p>	

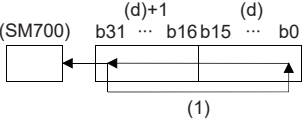
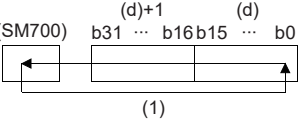
■BIN16位数据的左旋

指令符号	处理内容	参阅目标
ROL	将BIN16位数据在不包含进位标志的状况下左旋(n)位。	572页 ROL(P)、RCL(P)
ROLP	 <p>(1)</p> <p>(1) (n)位左旋</p>	
RCL	将BIN16位数据在包含进位标志的状况下左旋(n)位。	
RCLP	 <p>(1)</p> <p>(1) (n)位左旋</p>	

■BIN32位数据的右旋

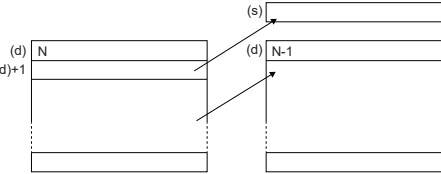
指令符号	处理内容	参阅目标
DROR	将BIN32位数据在不包含进位标志的状况下右旋(n)位。	575页 DROR(P)、DRCR(P)
DRORP	 <p>(1)</p> <p>(1) (n)位右旋</p>	
DRCR	将BIN32位数据在包含进位标志的状况下右旋(n)位。	
DRCRP	 <p>(1)</p> <p>(1) (n)位右旋</p>	

■BIN32位数据的左旋

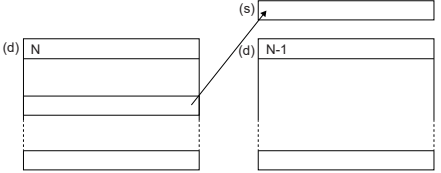
指令符号	处理内容	参阅目标
DROL DROLP	<p>将BIN32位数据在不包含进位标志的状况下左旋(n)位。</p>  <p>(1) (n)位左旋</p>	577页 DROL(P)、DRCL(P)
DRCL DRCLP	<p>将BIN32位数据在包含进位标志的状况下左旋(n)位。</p>  <p>(1) (n)位左旋</p>	

数据表操作指令

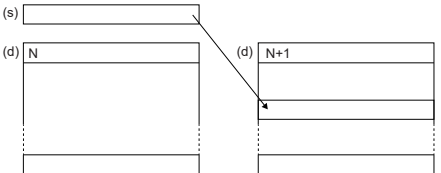
■从数据表的先入数据读取

指令符号	处理内容	参阅目标
FIFR FIFRP	<p>将最先存储到表中的数据存储到指定的软件元件中。</p>  <p>N: 数据数</p>	579页 FIFR(P)

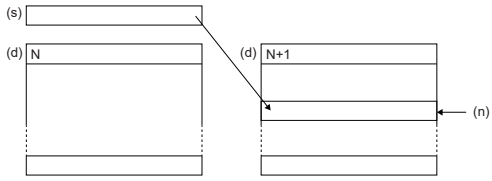
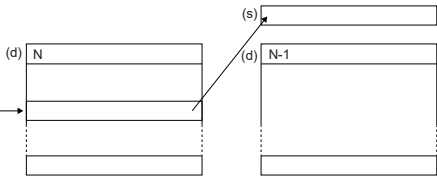
■从数据表的后入数据读取

指令符号	处理内容	参阅目标
FPOP FPOPP	<p>将最后存储到表中的数据存储到指定的软件元件中。</p>  <p>N: 数据数</p>	581页 FPOP(P)

■至数据表的数据写入

指令符号	处理内容	参阅目标
FIFW FIFWP	<p>将BIN16位数据存储到指定的数据表中。</p>  <p>N: 数据数</p>	583页 FIFW(P)

■数据表的数据插入、删除

指令符号	处理内容	参阅目标
FINS FINSP	<p>将BIN16位数据插入到指定数据表的第(n)个中。</p>  <p>N: 数据数</p>	585页 FINS (P)
FDEL FDELP	<p>将数据表的第(n)个数据删除。</p>  <p>N: 数据数</p>	587页 FDEL (P)

数据读取/写入指令

■从数据存储器的数据读取

指令符号	处理内容	参阅目标
S.DEVLD SP.DEVLD	从数据存储器上的软件数据存储用文件读取数据。	590页 S (P).DEVLD

■至数据存储器的数据写入

指令符号	处理内容	参阅目标
SP.DEVST	将数据写入到数据存储器上的软件数据存储用文件中。	592页 SP.DEVST

文件操作指令

■从指定文件的数据读取

指令符号	处理内容	参阅目标
SP. FREAD	从指定的文件读取数据。	594页 SP. FREAD

■至指定文件的数据写入

指令符号	处理内容	参阅目标
SP. FWRITE	将数据写入到指定的文件中。	608页 SP. FWRITE

■指定文件的删除

指令符号	处理内容	参阅目标
SP. FDELETE	删除指定的文件或文件夹。	621页 SP. FDELETE

■指定文件的复制

指令符号	处理内容	参阅目标
SP. FCOPY	复制指定的文件或文件夹。	624页 SP. FCOPY

■指定文件的移动

指令符号	处理内容	参阅目标
SP. FMOVE	移动指定的文件或文件夹。	629页 SP. FMOVE

■指定文件名的更改

指令符号	处理内容	参阅目标
SP. FRENAME	更改指定的文件名或文件夹名。	634页 SP. FRENAME

■指定文件状态的获取

指令符号	处理内容	参阅目标
SP. FSTATUS	获取指定文件或文件夹的状态。	637页 SP. FSTATUS

数据控制指令

■BIN16位/32位数据上下限限位控制

指令符号	处理内容	参阅目标
LIMIT	根据指定的输入值 (BIN16位值) 是否在上下限限位值的范围内, 对输出值进行控制。	641页 LIMIT(P) (_U)
LIMITP		
LIMIT_U		
LIMITP_U		
DLIMIT	根据指定的输入值 (BIN32位值) 是否在上下限限位值的范围内, 对输出值进行控制。	643页 DLIMIT(P) (_U)
DLIMITP		
DLIMIT_U		
DLIMITP_U		

■BIN16位/32位数据死区控制

指令符号	处理内容	参阅目标
BAND	根据指定的输入值 (BIN16位值) 是否在死区的上下限范围内, 对输出值进行控制。	645页 BAND(P) (_U)
BANDP		
BAND_U		
BANDP_U		
DBAND	根据指定的输入值 (BIN32位值) 是否在死区的上下限范围内, 对输出值进行控制。	647页 DBAND(P) (_U)
DBANDP		
DBAND_U		
DBANDP_U		

■BIN16位/32位数据区域控制

指令符号	处理内容	参阅目标
ZONE	对指定的输入值 (BIN16位值) 附加偏置值。	649页 ZONE (P) (_U)
ZONEP		
ZONE_U		
ZONEP_U		
DZONE	对指定的输入值 (BIN32位值) 附加偏置值。	651页 DZONE (P) (_U)
DZONEP		
DZONE_U		
DZONEP_U		

■BIN16位/32位单位标度 (各点坐标数据)

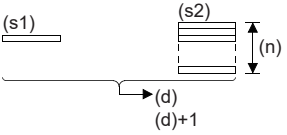
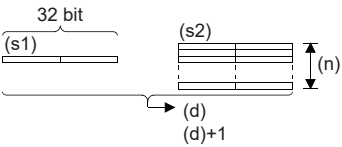
指令符号	处理内容	参阅目标
SCL	对 (s2) 中指定的标度用转换数据 (16位数据单位) 以 (s1) 中指定的输入值进行标度, 将运算结果存储到 (d) 中指定的软件元件中。 根据 (s2) 指定的软件元件及其以后中存储的标度用转换数据进行标度转换。	653页 SCL (P) (_U)
SCLP		
SCL_U		
SCLP_U		
DSCL	对 (s2) 中指定的标度用转换数据 (32位数据单位) 以 (s1) 中指定的输入值进行标度, 将运算结果存储到 (d) 中指定的软件元件中。 根据 (s2) 指定的软件元件及其以后中存储的标度用转换数据进行标度转换。	656页 DSCL (P) (_U)
DSCLP		
DSCL_U		
DSCLP_U		

■BIN16位/32位单位标度 (各X/Y坐标数据)

指令符号	处理内容	参阅目标
SCL2	对 (s2) 中指定的标度用转换数据 (16位数据单位) 以 (s1) 中指定的输入值进行标度, 将运算结果存储到 (d) 中指定的软件元件中。 根据 (s2) 指定的软件元件及其以后中存储的标度用转换数据进行标度转换。	659页 SCL2 (P) (_U)
SCL2P		
SCL2_U		
SCL2P_U		
DSCL2	对 (s2) 中指定的标度用转换数据 (32位数据单位) 以 (s1) 中指定的输入值进行标度, 将运算结果存储到 (d) 中指定的软件元件中。 根据 (s2) 指定的软件元件及其以后中存储的标度用转换数据进行标度转换。	661页 DSCL2 (P) (_U)
DSCL2P		
DSCL2_U		
DSCL2P_U		

数据处理指令

■BIN16位/32位数据查找

指令符号	处理内容	参阅目标
SERDATA	从 (s2) 中指定的BIN16位数据中查找 (n) 点的 (s1) 中指定的BIN16位数据。 	663页 SERDATA (P)
SERDATAP		
DSERDATA	从 (s2) 中指定的BIN32位数据中查找 (n) 点的 (s1) 中指定的BIN32位数据。 	665页 DSERDATA (P)
DSERDATAP		

■BIN16位/32位数据查找(最小・相同・最大)

指令符号	处理内容	参阅目标
SERMM SERMMP	<p>从(s1)中指定的BIN16位数据中查找(n)点的与(s2)中指定的BIN16位数据相同的数据以及最大值、最小值。</p>	667页 SERMM(P)
DSERMM DSERMMP	<p>从(s1)中指定的BIN32位数据中查找(n)点的与(s2)中指定的BIN32位数据相同的数据以及最大值、最小值。</p>	669页 DSERMM(P)

■BIN16位/32位数据位检查

指令符号	处理内容	参阅目标
SUM SUMP	<p>存储指定的软元件的BIN16位数据中, 变为1的位的总数。</p> <p>(d): 1的总数</p>	671页 SUM(P)
DSUM DSUMP	<p>存储指定的软元件的BIN32位数据中, 变为1的位的总数。</p> <p>(d): 1的总数</p>	672页 DSUM(P)

■BIN16位/32位数据最大值搜索

指令符号	处理内容	参阅目标
MAX MAXP MAX_U MAXP_U	<p>将从(s)中指定的软元件开始的(n)点的数据以16位单位进行搜索, 将最大值存储到(d)中指定的软元件中。</p>	673页 MAX(P) (_U)
DMAX DMAXP DMAX_U DMAXP_U	<p>将从(s)中指定的软元件开始的(n)点的数据以32位单位进行搜索, 将最大值存储到(d)中指定的软元件中。</p>	675页 DMAX(P) (_U)

■BIN16位/32位数据最小值搜索

指令符号	处理内容	参阅目标
MIN MINP MIN_U MINP_U	<p>将从(s)中指定的软元件开始的(n)点的数据以16位单位进行搜索, 将最小值存储到(d)中指定的软元件中。</p>	677页 MIN(P) (_U)
DMIN DMINP DMIN_U DMINP_U	<p>将从(s)中指定的软元件开始的(n)点的数据以32位单位进行搜索, 将最小值存储到(d)中指定的软元件中。</p>	679页 DMIN(P) (_U)

■BIN16位/32位数据排序

指令符号	处理内容	参阅目标
SORTD	将从(s1)中指定的软元件开始的(n)点的数据以16位单位进行排序。 (需要 $(n) \times ((n)-1) \div 2$ 个扫描)	681页 SORTD(_U)
SORTD_U		
DSORTD	将从(s1)中指定的软元件开始的(n)点的数据以32位单位进行排序。 (需要 $(n) \times ((n)-1) \div 2$ 个扫描)	683页 DSORTD(_U)
DSORTD_U		

■BIN16位数据合计值计算

指令符号	处理内容	参阅目标
WSUM	将从(s)中指定的软元件开始的(n)点的16位BIN数据全部进行加法运算后, 存储到(d)中指定的软元件中。	685页 WSUM(P)(_U)
WSUM_U		
WSUMP		
WSUMP_U		

■BIN32位数据合计值计算

指令符号	处理内容	参阅目标
DWSUM	将从(s)中指定的软元件开始的(n)点的32位BIN数据全部进行加法运算后, 存储到(d)中指定的软元件中。	687页 DWSUM(P)(_U)
DWSUM_U		
DWSUMP		
DWSUMP_U		

■BIN16位/32位数据平均值计算

指令符号	处理内容	参阅目标
MEAN	对从(s)中指定的软元件开始的(n)点的(16位单位)的平均值进行计算后, 存储到(d)中指定的软元件中。	689页 MEAN(P)(_U)
MEANP		
MEAN_U		
MEANP_U		
DMEAN	对从(s)中指定的软元件开始的(n)点的(32位单位)的平均值进行计算后, 存储到(d)中指定的软元件中。	691页 DMEAN(P)(_U)
DMEANP		
DMEAN_U		
DMEANP_U		

■BIN16位/32位平方根计算

指令符号	处理内容	参阅目标
SQRT	对指定的BIN16位数据的平方根进行运算。 $\sqrt{(s)} \rightarrow (d)$	693页 SQRT(P)
SQRTP		
DSQRT	对指定的BIN32位数据的平方根进行运算。 $\sqrt{(s)+1}, (s) \rightarrow (d)$	694页 DSQRT(P)
DSQRTP		

■CRC运算

指令符号	处理内容	参阅目标
CRC	对于(s)中指定的软元件开始的(n)点的8位数据(字节单位), 生成CRC值后存储到(d)中。	695页 CRC(P)
CRCP		

调试、故障诊断

调试、故障诊断指令

■错误显示或报警器复位

指令符号	处理内容	参阅目标
LEDR	对控制器的报警器显示及可继续运行的自诊断错误的显示进行复位。	697页 LEDR

字符串处理

字符串处理指令

■字符串比较

指令符号	处理内容	参阅目标
LD\$=、AND\$=、OR\$=	对 (s1) 的字符串与 (s2) 的字符串进行逐个字符的比较。	699页 LD\$□、AND\$□、OR\$□
LD\$<>、AND\$<>、OR\$<>		
LD\$>、AND\$>、OR\$>		
LD\$<=、AND\$<=、OR\$<=		
LD\$<、AND\$<、OR\$<		
LD\$>=、AND\$>=、OR\$>=		

■字符串的合并

指令符号	处理内容	参阅目标
\$+	将 (d) 中指定的字符串与 (s) 中指定的字符串合并后，存储到 (d) 及其以后。	702页 \$(P) [操作数为2个的情况下]
\$(P)		
\$+	将 (s1) 中指定的字符串与 (s2) 中指定的字符串合并后，存储到 (d) 及其以后。	704页 \$(P) [操作数为3个的情况下]
\$(P)		

■字符串传送

指令符号	处理内容	参阅目标
\$MOV	将 (s) 中指定的字符串传送到 (d) 中指定的软元件及其以后。	706页 \$MOV (P)
\$MOVP		
\$MOV_WS	将 (s) 中指定的Unicode字符串传送到 (d) 中指定的软元件及其以后。	708页 \$MOV (P)_WS
\$MOVP_WS		

■BIN16位/32位数据→10进制ASCII代码转换

指令符号	处理内容	参阅目标
BINDA	将 (s) 中指定的1字BIN值转换为10进制ASCII代码值后，存储到 (d) 中指定的字软元件编号及其以后。	710页 BINDA (P) (_U)
BINDAP		
BINDA_U		
BINDAP_U		
DBINDA	将 (s) 中指定的2字BIN值转换为10进制ASCII代码值后，存储到 (d) 中指定的字软元件编号及其以后。	714页 DBINDA (P) (_U)
DBINDAP		
DBINDA_U		
DBINDAP_U		

■BIN16位/32位数据→16进制ASCII代码转换

指令符号	处理内容	参阅目标
BINHA	将 (s) 中指定的1字BIN值转换为16进制ASCII代码值后, 存储到 (d) 中指定的字软件元件编号及其以后。	719页 BINHA (P)
BINHAP		
DBINHA	将 (s) 中指定的2字BIN值转换为16进制ASCII代码值后, 存储到 (d) 中指定的字软件元件编号及其以后。	723页 DBINHA (P)
DBINHAP		

■BIN16位/32位数据→字符串转换

指令符号	处理内容	参阅目标
STR	将 (s2) 中指定的1字BIN值转换为 (s1) 中指定的全部位数及小数部位数的10进制字符串后, 存储到 (d) 中指定的软件元件中。	727页 STR (P) (_U)
STRP		
STR_U		
STRP_U		
DSTR	将 (s2) 中指定的2字BIN值转换为 (s1) 中指定的全部位数及小数部位数的10进制字符串后, 存储到 (d) 中指定的软件元件中。	730页 DSTR (P) (_U)
DSTRP		
DSTR_U		
DSTRP_U		

■BCD4位/8位数据→10进制ASCII代码转换

指令符号	处理内容	参阅目标
BCDDA	将 (s) 中指定的1字BCD值转换为10进制ASCII代码值后, 存储到 (d) 中指定的字软件元件编号及其以后。	733页 BCDDA (P)
BCDDAP		
DBCDDA	将 (s) 中指定的2字BCD值转换为10进制ASCII代码值后, 存储到 (d) 中指定的字软件元件编号及其以后。	737页 DBCDDA (P)
DBCDDAP		

■单精度实数→字符串转换

指令符号	处理内容	参阅目标
ESTR	将 (s1) 中指定的单精度实数数据转换为字符串后, 存储到 (d) 中指定的软件元件中。	742页 ESTR (P)
ESTRP		

■16进制BIN数据→16进制ASCII代码转换

指令符号	处理内容	参阅目标
INT2ASC	将 (s) 中指定的软件元件编号及其以后的1字BIN值转换为16进制ASCII代码后, 以 (n) 中指定的字符数存储到 (d) 中指定的字软件元件编号及其以后。	747页 INT2ASC (P)
INT2ASCP		

■多个数据的字符串转换合并

指令符号	处理内容	参阅目标
SPF	按照指定的显示格式将多个数据转换为字符串, 与任意字符串连接后作为1个字符串输出。	749页 SPF (P)
SPFP		

■Unicode字符串→移位JIS字符串转换

指令符号	处理内容	参阅目标
WS2SJIS	将 (s) 中指定的Unicode字符串转换为移位JIS字符串后, 存储到 (d) 中。	765页 WS2SJIS (P)
WS2SJISP		

■移位JIS字符串→Unicode字符串转换(无字节顺序标志)

指令符号	处理内容	参阅目标
SJIS2WS	将 (s) 中指定的移位JIS字符串转换为Unicode字符串后, 存储到 (d) 中。	767页 SJIS2WS (P)
SJIS2WSP		

■移位JIS→Unicode转换(有字节顺序标记)

指令符号	处理内容	参阅目标
SJIS2WSB	将 (s) 中指定的移位JIS字符串转换为Unicode字符串后, 在起始处附加字节顺序标记后存储到 (d) 中。	769页 SJIS2WSB (P)
SJIS2WSBP		

■字符串的长度检测

指令符号	处理内容	参阅目标
LEN	将 (s) 中指定的软件件中存储的字符串数据的长度 (字符数) 存储到 (d) 中指定的软件件中。	771页 LEN (P)
LENP		

■从字符串的右侧/左侧提取

指令符号	处理内容	参阅目标
RIGHT	将 (s) 中指定的字符串的最终字符开始的 (n) 字符存储到 (d) 的指定软件件中。	773页 RIGHT (P)
RIGHTP		
LEFT	将 (s) 中指定的字符串的起始字符开始的 (n) 字符存储到 (d) 的指定软件件中。	775页 LEFT (P)
LEFTP		

■字符串中的任意提取、替换

指令符号	处理内容	参阅目标
MIDR	从 (s1) 中指定的字符串的 (s2) 中指定的位置开始, 将指定的字符数存储到 (d) 中指定的软件件中。	777页 MIDR (P)
MIDRP		
MIDW	将从 (s1) 的字符串开始的指定的字符数存储到 (d) 的字符串的 (s2) 中指定的位置处。	779页 MIDW (P)
MIDWP		

■字符串查找

指令符号	处理内容	参阅目标
INSTR	从 (s2) 的字符串的第 (s3) 字符开始搜索 (s1) 的字符串, 将一致的位置存储到 (d) 中。	782页 INSTR (P)
INSTRP		

■字符串插入

指令符号	处理内容	参阅目标
STRINS	将 (s1) 中指定的字符串数据, 插入到从 (d) 中指定的字符串数据的起始开始的第 (s2) 字符 (插入位置) 处。	784页 STRINS (P)
STRINSP		

■字符串删除

指令符号	处理内容	参阅目标
STRDEL	从 (d) 中指定的字符串数据的起始开始, 根据第 (s) 字符中指定的位置 (删除开始位置), 删除 (n) 字符的数据。	786页 STRDEL (P)
STRDELP		

实数处理

浮点指令

■单精度实数比较

指令符号	处理内容	参阅目标
LDE=、ANDE=、ORE=	对单精度实数通过常开触点处理进行比较运算。	788页 LDE□、ANDE□、ORE□
LDE<>、ANDE<>、ORE<>		
LDE>、ANDE>、ORE>		
LDE<=、ANDE<=、ORE<=		
LDE<、ANDE<、ORE<		
LDE>=、ANDE>=、ORE>=		

■双精度实数比较

指令符号	处理内容	参阅目标
LDED=、ANDED=、ORED=	对双精度实数通过常开触点处理进行比较运算。	790页 LDED□、ANDED□、ORED□
LDED<>、ANDED<>、ORED<>		
LDED>、ANDED>、ORED>		
LDED<=、ANDED<=、ORED<=		
LDED<、ANDED<、ORED<		
LDED>=、ANDED>=、ORED>=		

■单精度实数加减法运算

指令符号	处理内容	参阅目标
E+	对单精度实数进行加法运算。(操作数为2个的情况下)	793页 E+(P) [操作数为2个的情况下]
E+P		
E+	对单精度实数进行加法运算。(操作数为3个的情况下)	795页 E+(P) [操作数为3个的情况下]
E+P		
E-	对单精度实数进行减法运算。(操作数为2个的情况下)	797页 E-(P) [操作数为2个的情况下]
E-P		
E-	对单精度实数进行减法运算。(操作数为3个的情况下)	799页 E-(P) [操作数为3个的情况下]
E-P		

■双精度实数加减法运算

指令符号	处理内容	参阅目标
ED+	对双精度实数进行加法运算。(操作数为2个的情况下)	801页 ED+(P) [操作数为2个的情况下]
ED+P		
ED+	对双精度实数进行加法运算。(操作数为3个的情况下)	803页 ED+(P) [操作数为3个的情况下]
ED+P		
ED-	对双精度实数进行减法运算。(操作数为2个的情况下)	805页 ED-(P) [操作数为2个的情况下]
ED-P		
ED-	对双精度实数进行减法运算。(操作数为3个的情况下)	807页 ED-(P) [操作数为3个的情况下]
ED-P		

■单精度实数乘法运算

指令符号	处理内容	参阅目标
E*	对单精度实数进行乘法运算。	809页 E*(P)
E*P		
E/	对单精度实数进行除法运算。	811页 E/(P)
E/P		

■双精度实数乘法运算

指令符号	处理内容	参阅目标
ED*	对双精度实数进行乘法运算。	813页 ED*(P)
ED*P		
ED/	对双精度实数进行除法运算。	815页 ED/(P)
ED/P		

■带符号BIN16位/32位数据→单精度实数转换

指令符号	处理内容	参阅目标
INT2FLT	将(s)中指定的带符号BIN16位数据转换为单精度实数后, 存储到(d)中。	817页 INT2FLT(P)
INT2FLTP		
DINT2FLT	将(s)中指定的带符号BIN32位数据转换为单精度实数后, 存储到(d)中。	819页 DINT2FLT(P)
DINT2FLTP		

■无符号BIN16位/32位数据→单精度实数转换

指令符号	处理内容	参阅目标
UINT2FLT	将(s)中指定的无符号BIN16位数据转换为单精度实数后, 存储到(d)中。	818页 UINT2FLT(P)
UINT2FLTP		
UDINT2FLT	将(s)中指定的无符号BIN32位数据转换为单精度实数后, 存储到(d)中。	821页 UDINT2FLT(P)
UDINT2FLTP		

■双精度→单精度转换

指令符号	处理内容	参阅目标
DBL2FLT	将(s)中指定的双精度实数转换为单精度实数后, 存储到(d)中。	823页 DBL2FLT(P)
DBL2FLTP		

■带符号BIN16位/32位数据→双精度实数转换

指令符号	处理内容	参阅目标
INT2DBL	将(s)中指定的带符号BIN16位数据转换为双精度实数后, 存储到(d)中。	825页 INT2DBL(P)
INT2DBLP		
DINT2DBL	将(s)中指定的带符号BIN32位数据转换为双精度实数后, 存储到(d)中。	827页 DINT2DBL(P)
DINT2DBLP		

■无符号BIN16位/32位数据→双精度实数转换

指令符号	处理内容	参阅目标
UINT2DBL	将(s)中指定的无符号BIN16位数据转换为双精度实数后, 存储到(d)中。	826页 UINT2DBL(P)
UINT2DBLP		
UDINT2DBL	将(s)中指定的无符号BIN32位数据转换为双精度实数后, 存储到(d)中。	828页 UDINT2DBL(P)
UDINT2DBLP		

■单精度→双精度转换

指令符号	处理内容	参阅目标
FLT2DBL	将(s)中指定的单精度实数转换为双精度实数后, 存储到(d)中。	829页 FLT2DBL(P)
FLT2DBLP		

■字符串→单精度实数转换

指令符号	处理内容	参阅目标
EVAL	将(s)中指定的字符串转换为单精度实数后, 存储到(d)中指定的软件元件中。	830页 EVAL(P)
EVALP		

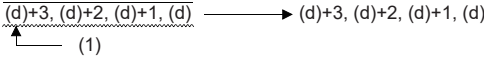
■BCD格式数据→单精度实数数据转换

指令符号	处理内容	参阅目标
EREXP	将(s1)的BCD数据以(s2)中指定的小数部位数转换为单精度实数后, 存储到(d)中指定的软件元件中。	834页 EREXP(P)
EREXPP		


■单精度实数符号取反

指令符号	处理内容	参阅目标
ENEG	对单精度实数数据的符号进行取反。 $\overline{(d)+1, (d)} \longrightarrow (d)+1, (d)$  (1) 实数	836页 ENEG(P)
ENEGP		

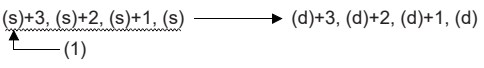
■双精度实数符号取反

指令符号	处理内容	参阅目标
EDNEG	对双精度实数数据的符号进行取反。 $\overline{(d)+3, (d)+2, (d)+1, (d)} \longrightarrow (d)+3, (d)+2, (d)+1, (d)$  (1) 实数	837页 EDNEG(P)
EDNEGP		

■单精度实数数据传送

指令符号	处理内容	参阅目标
EMOV	将单精度实数数据传送至指定的软件元件中。 $(s)+1, (s) \longrightarrow (d)+1, (d)$  (1) 实数	838页 EMOV(P)
EMOVP		

■双精度实数数据传送

指令符号	处理内容	参阅目标
EDMOV	将双精度实数数据传送至指定的软件元件中。 $(s)+3, (s)+2, (s)+1, (s) \longrightarrow (d)+3, (d)+2, (d)+1, (d)$  (1) 实数	839页 EDMOV(P)
EDMOVP		

■单精度实数SIN运算

指令符号	处理内容	参阅目标
SIN	对单精度实数中指定角度的SIN(正弦)值进行运算。	840页 SIN(P)
SINP		

■单精度实数COS运算

指令符号	处理内容	参阅目标
COS	对单精度实数中指定角度的COS(余弦)值进行运算。	842页 COS(P)
COSP		

■单精度实数TAN运算

指令符号	处理内容	参阅目标
TAN	对单精度实数中指定角度的TAN(正切)值进行运算。	844页 TAN(P)
TANP		

■单精度实数 SIN^{-1} 运算

指令符号	处理内容	参阅目标
ASIN	通过单精度实数中指定的SIN值进行角度运算。	846页 ASIN (P)
ASINP		

■单精度实数 COS^{-1} 运算

指令符号	处理内容	参阅目标
ACOS	通过单精度实数中指定的COS值进行角度运算。	848页 ACOS (P)
ACOSP		

■单精度实数 TAN^{-1} 运算

指令符号	处理内容	参阅目标
ATAN	通过单精度实数中指定的TAN值进行角度运算。	850页 ATAN (P)
ATANP		

■双精度实数SIN运算

指令符号	处理内容	参阅目标
SIND	对双精度实数中指定的角度的SIN(正弦)值进行运算。	852页 SIND (P)
SINDP		

■双精度实数COS运算

指令符号	处理内容	参阅目标
COSD	对双精度实数中指定的角度的COS(余弦)值进行运算。	854页 COSD (P)
COSDP		

■双精度实数TAN运算

指令符号	处理内容	参阅目标
TAND	对双精度实数中指定的角度的TAN(正接)值进行运算。	856页 TAND (P)
TANDP		

■双精度实数 SIN^{-1} 运算

指令符号	处理内容	参阅目标
ASIND	通过双精度实数中指定的SIN值进行角度运算。	858页 ASIND (P)
ASINDP		

■双精度实数 COS^{-1} 运算

指令符号	处理内容	参阅目标
ACOSD	通过双精度实数中指定的COS值进行角度运算。	860页 ACOSD (P)
ACOSDP		

■双精度实数 TAN^{-1} 运算

指令符号	处理内容	参阅目标
ATAND	通过双精度实数中指定的TAN值进行角度运算。	862页 ATAND (P)
ATANDP		

■BCD型SIN运算

指令符号	处理内容	参阅目标
BSIN	对以BCD值指定的角度的SIN(正弦)值进行运算。	864页 BSIN (P)
BSINP		

SIN (s) \longrightarrow

(d)
(d)+1
(d)+2

(d): 符号
(d)+1: 整数部
(d)+2: 小数部

■BCD型COS运算

指令符号	处理内容	参阅目标
BCOS	对以BCD值指定的角度的COS(余弦)值进行运算。	866页 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运算

指令符号	处理内容	参阅目标
BTAN	对以BCD值指定的角度的TAN(正接)值进行运算。	868页 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} 运算

指令符号	处理内容	参阅目标
BASIN	以BCD值指定的值的 SIN^{-1} 对(反正弦)值进行运算。	870页 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} 运算

指令符号	处理内容	参阅目标
BACOS	以BCD值指定的值的 COS^{-1} 对(反余弦)值进行运算。	872页 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} 运算

指令符号	处理内容	参阅目标
BATAN	以BCD值指定的值的 TAN^{-1} 对(反正切)值进行运算。	874页 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	

■单精度实数角度→弧度转换

指令符号	处理内容	参阅目标
RAD	将角度的大小单位从单精度实数中指定的度单位转换为弧度单位。	876页 RAD (P)
RADP	度→弧度转换 $(s)+1, (s) \longrightarrow (d)+1, (d)$	

■单精度实数弧度→角度转换

指令符号	处理内容	参阅目标
DEG	将角度的大小单位从单精度实数中指定的弧度单位转换为度单位。	878页 DEG (P)
DEGP	弧度→度转换 $(s)+1, (s) \longrightarrow (d)+1, (d)$	

■双精度实数角度→弧度转换

指令符号	处理内容	参阅目标
RADD	将角度的大小单位从双精度实数中指定的度单位转换为弧度单位。	880页 RADD (P)
RADDP	度→弧度转换 $(s)+3, (s)+2, (s)+1, (s) \longrightarrow (d)+3, (d)+2, (d)+1, (d)$	

■双精度实数弧度→角度转换

指令符号	处理内容	参阅目标
DEGD	将角度的大小单位从双精度实数中指定的弧度单位转换为度单位。	882页 DEGD (P)
DEGDP	弧度→度转换 $(s)+3, (s)+2, (s)+1, (s) \longrightarrow (d)+3, (d)+2, (d)+1, (d)$	

■单精度实数平方根

指令符号	处理内容	参阅目标
ESQRT	对单精度实数中指定的值的平方根进行运算。	884页 ESQRT (P)
ESQRTP	$\sqrt{(s)+1, (s)} \longrightarrow (d)+1, (d)$	

■双精度实数平方根

指令符号	处理内容	参阅目标
EDSQRT	对双精度实数中指定的值的平方根进行运算。	886页 EDSQRT (P)
EDSQRTP	$\sqrt{(s)+3, (s)+2, (s)+1, (s)} \longrightarrow (d)+3, (d)+2, (d)+1, (d)$	

■单精度实数指数运算

指令符号	处理内容	参阅目标
EXP	对单精度实数中指定的值的指数进行运算。	888页 EXP (P)
EXPP		

■双精度实数指数运算

指令符号	处理内容	参阅目标
EXPD	对双精度实数中指定的值的指数进行运算。	890页 EXPD (P)
EXPDP		

■单精度实数自然对数运算

指令符号	处理内容	参阅目标
LOG	对以单精度实数中指定的值的自然对数(e)为底时的对数进行运算。	892页 LOG (P)
LOGP		

■双精度实数自然对数运算

指令符号	处理内容	参阅目标
LOGD	对以双精度实数中指定的值的自然对数 (e) 为底时的对数进行运算。	894页 LOGD (P)
LOGDP		

■BCD4位/8位平方根

指令符号	处理内容	参阅目标		
BSQRT	对以BCD值 (4位) 指定的值的平方根进行运算。 $\sqrt{(s)}$ \longrightarrow <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>(d)</td></tr> <tr><td>(d)+1</td></tr> </table> (d): 整数部 (d)+1: 小数部	(d)	(d)+1	896页 BSQRT (P)
(d)				
(d)+1				
BSQRTP				
BDSQRT	对以BCD值 (8位) 指定的值的平方根进行运算。 $\sqrt{(s)+1, (s)}$ \longrightarrow <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>(d)</td></tr> <tr><td>(d)+1</td></tr> </table> (d): 整数部 (d)+1: 小数部	(d)	(d)+1	898页 BDSQRT (P)
(d)				
(d)+1				
BDSQRTP				

■单精度实数幂运算

指令符号	处理内容	参阅目标
POW	对单精度实数的幂进行运算。	900页 POW (P)
POWP		

■双精度实数幂运算

指令符号	处理内容	参阅目标
POWD	对双精度实数的幂进行运算。	902页 POWD (P)
POWDP		

■单精度实数常用对数运算

指令符号	处理内容	参阅目标
LOG10	对单精度实数中指定的值的常用对数 (以10为底的对数) 进行运算。	904页 LOG10 (P)
LOG10P		

■双精度实数常用对数运算

指令符号	处理内容	参阅目标
LOG10D	对双精度实数中指定的值的常用对数 (以10为底的对数) 进行运算。	906页 LOG10D (P)
LOG10DP		

■单精度实数最大值搜索

指令符号	处理内容	参阅目标
EMAX	将从 (s) 开始的 (n) 点中指定的单精度实数的块数据的最大值存储到 (d) 中指定的搜索结果 (最大值) 中。	908页 EMAX (P)
EMAXP		

■双精度实数最大值搜索

指令符号	处理内容	参阅目标
EDMAX	将从 (s) 开始的 (n) 点中指定的双精度实数的块数据的最大值存储到 (d) 中指定的搜索结果 (最大值) 中。	910页 EDMAX (P)
EDMAXP		

■单精度实数最小值搜索

指令符号	处理内容	参阅目标
EMIN	将从(s)开始的(n)点中指定的单精度实数的块数据的最小值存储到(d)中指定的搜索结果(最小值)中。	912页 EMIN(P)
EMINP		

■双精度实数最小值搜索

指令符号	处理内容	参阅目标
EDMIN	将从(s)开始的(n)点中指定的双精度实数的块数据的最小值存储到(d)中指定的搜索结果(最小值)中。	914页 EDMIN(P)
EDMINP		

随机数

随机数指令

■随机数发生、系列更改

指令符号	处理内容	参阅目标
RND	发生0到未滿32767的随机数，存储到(d)中指定的软元件中。	916页 RND(P)
RNDP		
SRND	根据(s)中指定的软元件中存储的16位BIN数据的内容，更改随机数系列。	917页 SRND(P)
SRNDP		

软元件操作

变址寄存器指令

■变址寄存器的批量保存、恢复

指令符号	处理内容	参阅目标
ZPUSH	将变址寄存器的内容保存到(d)中指定的区域中。	918页 ZPUSH(P)
ZPUSHP		
ZPOP	将(d)中指定的区域中保存的数据读取到变址寄存器中。	921页 ZPOP(P)
ZPOPP		

■变址寄存器/长变址寄存器的选择保存、恢复

指令符号	处理内容	参阅目标
ZPUSH	将(s)中指定的变址寄存器、长变址寄存器的内容保存到(d)中指定的区域中。	922页 ZPUSH(P)
ZPUSHP		
ZPOP	将(d)中指定的区域中保存的数据读取到变址寄存器、长变址寄存器中。	926页 ZPOP(P)
ZPOPP		

文件寄存器操作指令

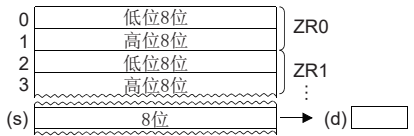
■文件寄存器的块No. 切换

指令符号	处理内容	参阅目标
RSET	将程序中使用的文件寄存器的块No. 更改为(s)中指定的软元件中存储的块No.。	928页 RSET(P)
RSETP		

文件寄存器的1字节单位的读取/写入指令

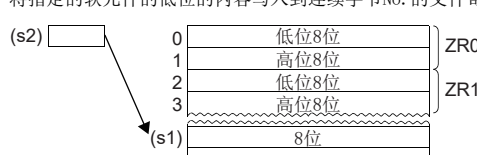
■文件寄存器的1字节数据读取

指令符号	处理内容	参阅目标
ZRRDB	读取指定的连续字节No. 的文件寄存器的内容。	930页 ZRRDB(P)
ZRRDBP		



■文件寄存器的1字节数据写入

指令符号	处理内容	参阅目标
ZRWRB	将指定的软元件的低位的内容写入到连续字节No. 的文件寄存器中。	932页 ZRWRB(P)
ZRWRBP		



间接地址读取指令

■间接地址读取

指令符号	处理内容	参阅目标
ADRSET	读取指定的软元件的间接地址。	934页 ADRSET (P)
ADRSETP	<p>(1) 指定软元件的间接地址 (2) 软元件名</p>	

定时器、计数器

特殊计数器指令

■单相输入升值/降值计数器

指令符号	处理内容	参阅目标
UDCNT1	更新指定计数器的当前值。 <p>(1) Cn当前值 (2) Cn触点</p>	936页 UDCNT1

■2相输入升值/降值计数器

指令符号	处理内容	参阅目标
UDCNT2	根据指定A相脉冲及B相脉冲的状态，更新计数器的当前值。 <p>(1) Cn当前值 (2) Cn触点</p>	938页 UDCNT2

特殊定时器指令

■示教定时器

指令符号	处理内容	参阅目标
TTMR	将计测指令ON中的时间以秒为单位进行测定后，存储与乘数相乘后的值。 <p>(s)=0:1, (s)=1:10, (s)=2:100 T_{ON}: TTMR的ON时间</p>	940页 TTMR

■特殊功能定时器

指令符号	处理内容	参阅目标
STMR	根据STMR指令的输入条件的ON/OFF，(d)中指定的位软元件开始的4点将进行以下动作。 <ul style="list-style-type: none"> • (d)+0: OFF延迟定时器输出 • (d)+1: OFF后单次触发定时器输出 • (d)+2: ON后单次触发定时器输出 • (d)+3: ON延迟+OFF延迟定时器 	942页 STMR

时钟

时钟用指令

■时钟数据的读取

指令符号	处理内容	参阅目标							
DATERD	从控制器的时钟单元中读取“年、月、日、时、分、秒、星期”。	944页 DATERD(P)							
DATERDP	<table border="1"> <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> <p>(d)：年 (d)+1：月 (d)+2：日 (d)+3：时 (d)+4：分 (d)+5：秒 (d)+6：星期</p>	(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									

■时钟数据的写入

指令符号	处理内容	参阅目标							
DATEWR	将指定的软元件及其以后存储的时钟数据，写入到控制器的时钟单元中。	946页 DATEWR(P)							
DATEWRP	<table border="1"> <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> <p>(s)：年 (s)+1：月 (s)+2：日 (s)+3：时 (s)+4：分 (s)+5：秒 (s)+6：星期</p>	(s)	(s)+1	(s)+2	(s)+3	(s)+4	(s)+5	(s)+6	
(s)									
(s)+1									
(s)+2									
(s)+3									
(s)+4									
(s)+5									
(s)+6									

■时钟数据的加法运算

指令符号	处理内容	参阅目标
DATE+	对时间数据进行加法运算。	948页 DATE+(P)
DATE+P	$ \begin{array}{ c } \hline (s1) \\ \hline \text{hour} \\ \hline \text{minute} \\ \hline \text{second} \\ \hline \end{array} + \begin{array}{ c } \hline (s2) \\ \hline \text{hour} \\ \hline \text{minute} \\ \hline \text{second} \\ \hline \end{array} \rightarrow \begin{array}{ c } \hline (d) \\ \hline \text{hour} \\ \hline \text{minute} \\ \hline \text{second} \\ \hline \end{array} $	

■时钟数据的减法运算

指令符号	处理内容	参阅目标
DATE-	对时间数据进行减法运算。	950页 DATE-(P)
DATE-P	$ \begin{array}{ c } \hline (s1) \\ \hline \text{hour} \\ \hline \text{minute} \\ \hline \text{second} \\ \hline \end{array} - \begin{array}{ c } \hline (s2) \\ \hline \text{hour} \\ \hline \text{minute} \\ \hline \text{second} \\ \hline \end{array} \rightarrow \begin{array}{ c } \hline (d) \\ \hline \text{hour} \\ \hline \text{minute} \\ \hline \text{second} \\ \hline \end{array} $	

■时间数据的转换(时分秒→秒)

指令符号	处理内容	参阅目标
TIME2SEC	将时间数据(时、分、秒)换算为秒数据。	952页 TIME2SEC(P)
TIME2SECP		

■时间数据的转换(秒→时分秒)

指令符号	处理内容	参阅目标
SEC2TIME	将秒数据换算为时间数据(时、分、秒)。	954页 SEC2TIME(P)
SEC2TIMEP		

■日期时间数据的转换

指令符号	处理内容	参阅目标
DATE2SEC	将日期时间数据(年、月、日、时、分、秒)换算为秒数据。	956页 DATE2SEC(P) (_U)
DATE2SECP		
DATE2SEC_U		
DATE2SECP_U		
DATE2SECP_U		
SEC2DATE	将秒数据(年、月、日、时、分、秒、星期)换算为日期时间数据。	958页 SEC2DATE(P) (_U)
SEC2DATEP		
SEC2DATE_U		
SEC2DATEP_U		
SEC2DATEP_U		

■日期比较

指令符号	处理内容	参阅目标
LDDT=、ANDDT=、ORDT=	进行指定日期数据的比较，或对日期数据与当前日期进行比较。	960页 LDDT□、ANDDT□、ORDT□
LDDT<>、ANDDT<>、ORDT<>		
LDDT>、ANDDT>、ORDT>		
LDDT<=、ANDDT<=、ORDT<=		
LDDT<、ANDDT<、ORDT<		
LDDT>=、ANDDT>=、ORDT>=		
LDDT>、ANDDT>、ORDT>		

■时间比较

指令符号	处理内容	参阅目标
LDTM=、ANDTM=、ORTM=	进行指定时间数据的比较，或对指定的时间数据与当前时间进行比较。	964页 LDTM□、ANDTM□、ORTM□
LDTM<>、ANDTM<>、ORTM<>		
LDTM>、ANDTM>、ORTM>		
LDTM<=、ANDTM<=、ORTM<=		
LDTM<、ANDTM<、ORTM<		
LDTM>=、ANDTM>=、ORTM>=		

■时间比较输出

指令符号	处理内容	参阅目标
TCMP	对 (s1)、(s2)、(s3) 中指定的比较时间数据与 (s4) 中指定的时间数据进行比较，根据其结果(小、一致、大) (d)、(d)+1、(d)+2 中之一将变为0N。	967页 TCMP(P)
TCMPP		

■扩展时钟数据的读取

指令符号	处理内容	参阅目标
S. DATERD	从控制器内部的时钟单元中读取包含有1/1000秒的时钟数据。	969页 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: 时
 (d)+4: 分
 (d)+5: 秒
 (d)+6: 星期
 (d)+7: 1/1000秒

■扩展时钟数据的加法运算

指令符号	处理内容	参阅目标
S. DATE+	对时间数据进行加法运算。	971页 S(P).DATE+
SP. DATE+		

(s1)		(s2)	+	(d)
hour		hour		hour
minute		minute		minute
second		second		second
—		—		—
1/1000 second		1/1000 second		1/1000 second

■扩展时钟数据的减法运算

指令符号	处理内容	参阅目标
S. DATE-	对时间数据进行减法运算。	973页 S(P).DATE-
SP. DATE-		

(s1)		(s2)	-	(d)
hour		hour		hour
minute		minute		minute
second		second		second
—		—		—
1/1000 second		1/1000 second		1/1000 second

时机计测指令

■发生时机脉冲

指令符号	处理内容	参阅目标
DUTY	将用户用时机时钟进行指定扫描的ON、指定扫描的OFF。 (n1): (n1)扫描 (n2): (n2)扫描 (d): SM420~SM424	975页 DUTY

■指定数据的时间计测

指令符号	处理内容	参阅目标
TIMCHK	对输入条件的ON时间进行计测，如果连续ON达到设置的时间，则将(d)中指定的软元件置为ON。	977页 TIMCHK

模块访问

模块访问指令

■I/O刷新

指令符号	处理内容	参阅目标
RFS	在1个扫描的途中进行相应的输入输出的部分刷新。	978页 RFS(P)
RFSP		

■从模块中进行1字/2字数据读取(16位指定)

指令符号	处理内容	参阅目标
FROM	从智能功能模块内的缓冲存储器中，以16位单位读取(n)字的数据。	980页 FROM(P)、DFROM(P)
FROMP		
DFROM	从智能功能模块内的缓冲存储器中，以16位单位读取(n)×2字的数据。	
DFROMP		

■至模块的1字/2字数据写入(16位指定)

指令符号	处理内容	参阅目标
TO	在智能功能模块内的缓冲存储器中，以16位单位写入(n)字的数据。	983页 TO(P)、DTO(P)
TOP		
DTO	在智能功能模块的缓冲存储器中，以16位单位写入(n)×2字的数据。	
DTOP		

■从模块中进行1字/2字数据读取(32位指定)

指令符号	处理内容	参阅目标
FROMD	从智能功能模块内的缓冲存储器中，以32位单位读取(n)字的数据。	988页 FROMD(P)、DFROMD(P)
FROMDP		
DFROMD	从智能功能模块内的缓冲存储器中，以32位单位读取(n)×2字的数据。	
DFROMDP		

■至模块的1字/2字数据写入(32位指定)

指令符号	处理内容	参阅目标
TOD	在智能功能模块内的缓冲存储器中，以32位单位写入(n)字的数据。	992页 TOD(P)、DTOD(P)
TODP		
DTOD	在智能功能模块内的缓冲存储器中，以32位单位写入(n)×2字的数据。	
DTODP		

■模块固有信息读取

指令符号	处理内容	参阅目标
UNLINF RD	将模块信息存储到(d)中指定的软元件及其以后。	996页 UNLINF RD(P)
UNLINF RDP		

PID运算指令

指令符号	处理内容	参阅目标
PID	根据 (s1)、(s2)、(s3) 中设置的值执行PID运算，并将各采样时间的运算结果存储到 (d) 中。	1012页 PID

PID控制指令

PID控制指令(不完全微分)

■PID控制用数据的设置

指令符号	处理内容	参阅目标
S. PIDINIT	将 (s) 中指定的软件编号及其以后设置的使用环路数的PID控制用数据批量登录到控制器内部，置为可PID控制状态。	1027页 S(P).PIDINIT
SP. PIDINIT		

■PID运算

指令符号	处理内容	参阅目标
S. PIDCONT	进行采样周期的计测及PID运算。	1030页 S(P).PIDCONT
SP. PIDCONT		

■指定环路No. 的运算停止/开始

指令符号	处理内容	参阅目标
S. PIDSTOP	停止 (s) 中指定的环路No. 的PID运算。	1033页 S(P).PIDSTOP
SP. PIDSTOP		
S. PIDRUN	开始 (s) 中指定的环路No. 的PID运算。	1034页 S(P).PIDRUN
SP. PIDRUN		

■指定环路No. 的参数更改

指令符号	处理内容	参阅目标
S. PIDPRMW	将 (s1) 中指定的环路No. 的运算参数更改为 (s2) 中指定的软件编号及其以后存储的PID控制用数据。	1035页 S(P).PIDPRMW
SP. PIDPRMW		

PID控制指令(完全微分)

■PID控制用数据的设置

指令符号	处理内容	参阅目标
PIDINIT	将 (s) 中指定的软件编号及其以后设置的使用环路数的PID控制用数据批量登录到控制器内部，置为可PID控制状态。	1039页 PIDINIT(P)
PIDINITP		

■PID运算

指令符号	处理内容	参阅目标
PIDCONT	进行采样周期的计测及PID运算。	1041页 PIDCONT(P)
PIDCONTP		

■指定环路No. 的运算停止/开始

指令符号	处理内容	参阅目标
PIDSTOP	停止 (s) 中指定的环路No. 的PID运算。	1044页 PIDSTOP(P)
PIDSTOPP		
PIDRUN	开始 (s) 中指定的环路No. 的PID运算。	1045页 PIDRUN(P)
PIDRUNP		

■指定环路No. 的参数更改

指令符号	处理内容	参阅目标
PIDPRMW	将 (s1) 中指定的环路No. 的运算参数更改为 (s2) 中指定的软件编号及其以后存储的PID控制用数据。	1046页 PIDPRMW(P)
PIDPRMWP		

高速I/O控制指令

高速输入输出的开始/停止

指令符号	处理内容	参阅目标
HIOEN(P)	控制高速输入输出功能的开始/停止。	1048页 HIOEN(P)
DHIOEN(P)		1051页 DHIOEN(P)

定位功能专用指令

机械原点复位

指令符号	处理内容	参阅目标
DSZR	使用该指令进行机械式原点复位。	1054页 DSZR/DDSZR
DDSZR		

中断1速定位

指令符号	处理内容	参阅目标
DVIT	该指令执行中断1速定长进给。	1061页 DVIT/DDVIT
DDVIT		

可变速度运行

指令符号	处理内容	参阅目标
PLSV	输出带旋转方向输出的可变速脉冲。	1067页 PLSV/DPLSV
DPLSV		

1速定位(相对)

指令符号	处理内容	参阅目标
DRVI	该指令通过相对地址进行1速定位。	1072页 DRVI/DDRVI
DDRVI		

1速定位(绝对)

指令符号	处理内容	参阅目标
DRVA	该指令通过绝对地址进行1速定位。	1077页 DRVA/DDRVA
DDRVA		

表格运行(多个表格)

指令符号	处理内容	参阅目标
DRVTL	该指令通过工程工具设置的数据表格，以1个指令连续运行或步进运行多个表格。	1082页 DRVTL

中断输入1触发

指令符号	处理内容	参阅目标
DITRG	该指令将发生与输入中断1信号同等的触发，不使用X软件进行触发。	1084页 DITRG

绝对位置恢复

指令符号	处理内容	参阅目标
DABRST	与支持绝对位置恢复功能的伺服放大器连接，读取伺服放大器的绝对位置(ABS)数据。读取数据时转换为脉冲值。	1087页 DABRST

高速计数器功能专用指令

16位数据读取/写入(更新)

指令符号	处理内容	参阅目标
HCMOV (P)	该指令进行高速计数器/脉冲宽度测定/PWM/定位用特殊继电器/特殊寄存器的读取、写入(更新)。	1090页 HCMOV (P)

32位数据读取/写入(更新)

指令符号	处理内容	参阅目标
DHCMOV (P)	该指令进行高速计数器/脉冲宽度测定/PWM/定位用特殊继电器/特殊寄存器的读取、写入(更新)。	1093页 DHCMOV (P)

脉冲系统指令

BIN16位脉冲宽度调制

指令符号	处理内容	参阅目标
PWMH	将(s1)中指定的ON时间(16位数据单位)及(s2)中指定的周期的脉冲(16位数据单位)输出到(d)中指定的输出目标中。	1095页 PWMH

BIN32位脉冲宽度调制

指令符号	处理内容	参阅目标
DPWMH	将(s1)中指定的ON时间(32位数据单位)及(s2)中指定的周期的脉冲(32位数据单位)输出到(d)中指定的输出目标中。	1100页 DPWMH

16位数据脉冲密度的测定

指令符号	处理内容	参阅目标
SPDH	通过中断输入测定指定时间的输入脉冲。	1105页 SPDH

32位数据脉冲密度的测定

指令符号	处理内容	参阅目标
DSPDH	通过中断输入测定指定时间的输入脉冲。	1107页 DSPDH

7 网络指令一览

7.1 以太网用指令

打开/关闭处理指令

■连接的建立

指令符号	处理内容	参阅目标
GP. SOCOPEN	进行 (s1) 中指定的连接的打开处理。	1110页 GP. SOCOPEN、 SP. SOCOPEN
SP. SOCOPEN		

■连接的切断

指令符号	处理内容	参阅目标
GP. SOCCLOSE	对 (s1) 中指定的连接进行关闭处理。(连接的切断)	1114页 GP. SOCCLOSE、 SP. SOCCLOSE
SP. SOCCLOSE		

套接字通信用指令

■接收数据的END处理时读取

指令符号	处理内容	参阅目标
GP. SOCRCV	END处理时将 (s1) 中指定的连接的接收数据从套接字通信接收数据区域读取。	1116页 GP. SOCRCV、SP. SOCRCV
SP. SOCRCV		

■执行接收数据的指令时读取

指令符号	处理内容	参阅目标
G. SOCRCVS	执行指令时将 (s) 中指定的连接的接收数据从套接字通信接收数据区域读取。	1119页 G. SOCRCVS、S. SOCRCVS
S. SOCRCVS		

■数据发送

指令符号	处理内容	参阅目标
GP. SOCSND	将 (s3) 中设置的数据发送至 (s1) 中指定的连接的对象设备中。	1122页 GP. SOCSND、SP. SOCSND
SP. SOCSND		

■连接信息的读取

指令符号	处理内容	参阅目标
GP. SOCCINF	读取 (s1) 中指定的连接的连接信息。	1126页 GP. SOCCINF、 SP. SOCCINF
SP. SOCCINF		

■连接的通信目标更改(UDP/IP)

指令符号	处理内容	参阅目标
GP. SOCCSET	更改 (s1) 中指定的连接的通信对象IP地址、通信对象端口编号。 (仅UDP/IP通信时)	1129页 GP. SOCCSET、 SP. SOCCSET
SP. SOCCSET		

■连接的接收模式更改

指令符号	处理内容	参阅目标
GP. SOCRMODE	对 (s1) 中指定的连接, 更改TCP/TLS接收模式及接收数据容量。	1131页 GP. SOCRMODE、 SP. SOCRMODE
SP. SOCRMODE		

■套接字通信接收数据读取

指令符号	处理内容	参阅目标
G. SOCRDATA	从(s1)中指定的连接的套接字通信接收数据区域中, 读取(n)中指定的字的数据后, 存储到(d)中指定的软元件及其以后。	1136页 G(P). SOCRDATA、 S(P). SOCRDATA
GP. SOCRDATA		
S. SOCRDATA		
SP. SOCRDATA		

通信协议支持功能指令

■登录协议执行

指令符号	处理内容	参阅目标
GP. ECPRTCL	执行工程工具的通信协议支持功能中设置的协议。	1138页 GP. ECPRTCL、 SP. ECPRTCL
SP. ECPRTCL		

SLMP帧发送指令

■SLMP帧发送

指令符号	处理内容	参阅目标
G. SLMPSND	对SLMP对应设备发送SLMP的报文。	1146页 G(P). SLMPSND、 J(P). SLMPSND、SP. SLMPSND
GP. SLMPSND		
J. SLMPSND		
JP. SLMPSND		
SP. SLMPSND		

■加密的SLMP帧发送

指令符号	处理内容	参阅目标
G. SLMPSNDC	对SLMP对应设备发送加密的SLMP报文。	1155页 G(P). SLMPSNDC
GP. SLMPSNDC		

文件传送功能用指令

■FTP客户端文件发送

指令符号	处理内容	参阅目标
GP. FTPCPUT	将(s2)中指定的控制器的文件发送至(s3)中指定的FTP服务器的文件夹路径。	1163页 GP. FTPCPUT、SP. FTPPUT
SP. FTPPUT		

■FTP客户端文件获取

指令符号	处理内容	参阅目标
GP. FTPCGET	将(s2)中指定的FTP服务器文件获取到(s3)中指定的控制器的文件夹路径中。	1168页 GP. FTPCGET、SP. FTPGET
SP. FTPGET		

7.2 外部设备通信指令

特殊适配器用指令

指令符号	处理内容	参阅目标
ADPRW	通过主站对应的功能代码，可以与从站进行通信(数据的读取/写入)。	1173页 ADPRW

串行数据传送2

指令符号	处理内容	参阅目标
RS2	经由安装在控制器上的通信适配器，通过无顺序通信进行数据的发送接收。	1176页 RS2

一览表的阅读方法如下所示。

项目	内容
函数符号、FB符号	表示函数、FB名。
处理内容	表示函数、FB的概要。
参阅目标	表示详细说说的参阅目标。

8.1 通用函数

类型转换函数

■BOOL型→WORD型/DWORD型转换

函数符号	处理内容	参阅目标
BOOL_TO_WORD	将BOOL型数据转换为WORD型数据。	1180页 BOOL_TO_WORD(_E)
BOOL_TO_WORD_E		
BOOL_TO_DWORD	将BOOL型数据转换为DWORD型数据。	1181页 BOOL_TO_DWORD(_E)
BOOL_TO_DWORD_E		

■BOOL型→INT型/DINT型转换

函数符号	处理内容	参阅目标
BOOL_TO_INT	将BOOL型数据转换为INT型数据。	1182页 BOOL_TO_INT(_E)
BOOL_TO_INT_E		
BOOL_TO_DINT	将BOOL型数据转换为DINT型数据。	1183页 BOOL_TO_DINT(_E)
BOOL_TO_DINT_E		

■BOOL型→TIME型转换

函数符号	处理内容	参阅目标
BOOL_TO_TIME	将BOOL型数据转换为TIME型数据。	1184页 BOOL_TO_TIME(_E)
BOOL_TO_TIME_E		

■BOOL型→STRING型转换

函数符号	处理内容	参阅目标
BOOL_TO_STRING	将BOOL型数据转换为STRING型数据。	1185页 BOOL_TO_STRING(_E)
BOOL_TO_STRING_E		

■WORD型→BOOL型转换

函数符号	处理内容	参阅目标
WORD_TO_BOOL	将WORD型数据转换为BOOL型数据。	1186页 WORD_TO_BOOL(_E)
WORD_TO_BOOL_E		

■WORD型→DWORD型转换

函数符号	处理内容	参阅目标
WORD_TO_DWORD	将WORD型数据转换为DWORD型数据。	1187页 WORD_TO_DWORD(_E)
WORD_TO_DWORD_E		

■WORD型→INT型/DINT型转换

函数符号	处理内容	参阅目标
WORD_TO_INT	将WORD型数据转换为INT型数据。	1188页 WORD_TO_INT(_E)
WORD_TO_INT_E		
WORD_TO_DINT	将WORD型数据转换为DINT型数据。	1189页 WORD_TO_DINT(_E)
WORD_TO_DINT_E		

■WORD型→TIME型转换

函数符号	处理内容	参阅目标
WORD_TO_TIME	将WORD型数据转换为TIME型数据。	1190页 WORD_TO_TIME(_E)
WORD_TO_TIME_E		

■WORD型→STRING型转换

函数符号	处理内容	参阅目标
WORD_TO_STRING	将WORD型数据转换为STRING型数据。	1191页 WORD_TO_STRING(_E)
WORD_TO_STRING_E		

■DWORD型→BOOL型转换

函数符号	处理内容	参阅目标
DWORD_TO_BOOL	将DWORD型数据转换为BOOL型数据。	1192页 DWORD_TO_BOOL(_E)
DWORD_TO_BOOL_E		

■DWORD型→WORD型转换

函数符号	处理内容	参阅目标
DWORD_TO_WORD	将DWORD型数据转换为WORD型数据。	1193页 DWORD_TO_WORD(_E)
DWORD_TO_WORD_E		

■DWORD型→INT型/DINT型转换

函数符号	处理内容	参阅目标
DWORD_TO_INT	将DWORD型数据转换为INT型数据。	1194页 DWORD_TO_INT(_E)
DWORD_TO_INT_E		
DWORD_TO_DINT	将DWORD型数据转换为DINT型数据。	1195页 DWORD_TO_DINT(_E)
DWORD_TO_DINT_E		

■DWORD型→TIME型转换

函数符号	处理内容	参阅目标
DWORD_TO_TIME	将DWORD型数据转换为TIME型数据。	1196页 DWORD_TO_TIME(_E)
DWORD_TO_TIME_E		

■DWORD型→STRING型转换

函数符号	处理内容	参阅目标
DWORD_TO_STRING	将DWORD型数据转换为STRING型数据。	1197页 DWORD_TO_STRING(_E)
DWORD_TO_STRING_E		

■INT型→BOOL型转换

函数符号	处理内容	参阅目标
INT_TO_BOOL	将INT型数据转换为BOOL型数据。	1198页 INT_TO_BOOL(_E)
INT_TO_BOOL_E		

■INT型→WORD型/DWORD型转换

函数符号	处理内容	参阅目标
INT_TO_WORD	将INT型数据转换为WORD型数据。	1199页 INT_TO_WORD(_E)
INT_TO_WORD_E		
INT_TO_DWORD	将INT型数据转换为DWORD型数据。	1200页 INT_TO_DWORD(_E)
INT_TO_DWORD_E		

■INT型→DINT型转换

函数符号	处理内容	参阅目标
INT_TO_DINT	将INT型数据转换为DINT型数据。	1201页 INT_TO_DINT(_E)
INT_TO_DINT_E		

■INT型→BCD型转换

函数符号	处理内容	参阅目标
INT_TO_BCD	将INT型数据转换为BCD型数据。	1202页 INT_TO_BCD(_E)
INT_TO_BCD_E		

■INT型→REAL型/LREAL型转换

函数符号	处理内容	参阅目标
INT_TO_REAL	将INT型数据转换为REAL型数据。	1203页 INT_TO_REAL(_E)
INT_TO_REAL_E		
INT_TO_LREAL	将INT型数据转换为LREAL型数据。	1204页 INT_TO_LREAL(_E)
INT_TO_LREAL_E		

■INT型→TIME型转换

函数符号	处理内容	参阅目标
INT_TO_TIME	将INT型数据转换为TIME型数据。	1205页 INT_TO_TIME(_E)
INT_TO_TIME_E		

■INT型→STRING型转换

函数符号	处理内容	参阅目标
INT_TO_STRING	将INT型数据转换为STRING型数据。	1206页 INT_TO_STRING(_E)
INT_TO_STRING_E		

■DINT型→BOOL型转换

函数符号	处理内容	参阅目标
DINT_TO_BOOL	将DINT型数据转换为BOOL型数据。	1208页 DINT_TO_BOOL(_E)
DINT_TO_BOOL_E		

■DINT型→WORD型/DWORD型转换

函数符号	处理内容	参阅目标
DINT_TO_WORD	将DINT型数据转换为WORD型数据。	1209页 DINT_TO_WORD(_E)
DINT_TO_WORD_E		
DINT_TO_DWORD	将DINT型数据转换为DWORD型数据。	1210页 DINT_TO_DWORD(_E)
DINT_TO_DWORD_E		

■DINT型→INT型转换

函数符号	处理内容	参阅目标
DINT_TO_INT	将DINT型数据转换为INT型数据。	1211页 DINT_TO_INT(_E)
DINT_TO_INT_E		

■DINT型→BCD型转换

函数符号	处理内容	参阅目标
DINT_TO_BCD	将DINT型数据转换为BCD型数据。	1212页 DINT_TO_BCD(_E)
DINT_TO_BCD_E		

■DINT型→REAL型/LREAL型转换

函数符号	处理内容	参阅目标
DINT_TO_REAL	将DINT型数据转换为REAL型数据。	1214页 DINT_TO_REAL(_E)
DINT_TO_REAL_E		
DINT_TO_LREAL	将DINT型数据转换为LREAL型数据。	1215页 DINT_TO_LREAL(_E)
DINT_TO_LREAL_E		

■DINT型→TIME型转换

函数符号	处理内容	参阅目标
DINT_TO_TIME	将DINT型数据转换为TIME型数据。	1216页 DINT_TO_TIME(_E)
DINT_TO_TIME_E		

■DINT型→STRING型转换

函数符号	处理内容	参阅目标
DINT_TO_STRING	将DINT型数据转换为STRING型数据。	1217页 DINT_TO_STRING(_E)
DINT_TO_STRING_E		

■BCD型→INT型/DINT型转换

函数符号	处理内容	参阅目标
BCD_TO_INT	将BCD型数据转换为INT型数据。	1219页 BCD_TO_INT(_E)
BCD_TO_INT_E		
BCD_TO_DINT	将BCD型数据转换为DINT型数据。	1221页 BCD_TO_DINT(_E)
BCD_TO_DINT_E		

■BCD型→STRING型转换

函数符号	处理内容	参阅目标
BCD_TO_STRING	将BCD型数据转换为STRING型数据。	1223页 BCD_TO_STRING(_E)
BCD_TO_STRING_E		

■REAL型→INT型/DINT型转换

函数符号	处理内容	参阅目标
REAL_TO_INT	将REAL型数据转换为INT型数据。	1225页 REAL_TO_INT(_E)
REAL_TO_INT_E		
REAL_TO_DINT	将REAL型数据转换为DINT型数据。	1226页 REAL_TO_DINT(_E)
REAL_TO_DINT_E		

■REAL型→LREAL型转换

函数符号	处理内容	参阅目标
REAL_TO_LREAL	将REAL型数据转换为LREAL型数据。	1227页 REAL_TO_LREAL(_E)
REAL_TO_LREAL_E		

■REAL型→STRING型转换

函数符号	处理内容	参阅目标
REAL_TO_STRING	将REAL型数据转换为STRING型(指数型式)数据。	1228页 REAL_TO_STRING(_E)
REAL_TO_STRING_E		

■LREAL型→INT型/DINT型转换

函数符号	处理内容	参阅目标
LREAL_TO_INT	将LREAL型数据转换为INT型数据。	1231页 LREAL_TO_INT(_E)
LREAL_TO_INT_E		
LREAL_TO_DINT	将LREAL型数据转换为DINT型数据。	1232页 LREAL_TO_DINT(_E)
LREAL_TO_DINT_E		

■LREAL型→REAL型转换

函数符号	处理内容	参阅目标
LREAL_TO_REAL	将LREAL型数据转换为REAL型数据。	1233页 LREAL_TO_REAL(_E)
LREAL_TO_REAL_E		

■TIME型→BOOL型转换

函数符号	处理内容	参阅目标
TIME_TO_BOOL	将TIME型数据转换为BOOL型数据。	1234页 TIME_TO_BOOL(_E)
TIME_TO_BOOL_E		

■TIME型→WORD型/DWORD型转换

函数符号	处理内容	参阅目标
TIME_TO_WORD	将TIME型数据转换为WORD型数据。	1235页 TIME_TO_WORD(_E)
TIME_TO_WORD_E		
TIME_TO_DWORD	将TIME型数据转换为DWORD型数据。	1236页 TIME_TO_DWORD(_E)
TIME_TO_DWORD_E		

■TIME型→INT型/DINT型转换

函数符号	处理内容	参阅目标
TIME_TO_INT	将TIME型数据转换为INT型数据。	1237页 TIME_TO_INT(_E)
TIME_TO_INT_E		
TIME_TO_DINT	将TIME型数据转换为DINT型数据。	1238页 TIME_TO_DINT(_E)
TIME_TO_DINT_E		

■TIME型→STRING型转换

函数符号	处理内容	参阅目标
TIME_TO_STRING	将TIME型数据转换为STRING型数据。	1239页 TIME_TO_STRING(_E)
TIME_TO_STRING_E		

■STRING型→BOOL型转换

函数符号	处理内容	参阅目标
STRING_TO_BOOL	将STRING型数据转换为BOOL型数据。	1240页 STRING_TO_BOOL(_E)
STRING_TO_BOOL_E		

■STRING型→WORD型/DWORD型转换

函数符号	处理内容	参阅目标
STRING_TO_WORD	将STRING型数据转换为WORD型数据。	1241页 STRING_TO_WORD(_E)
STRING_TO_WORD_E		
STRING_TO_DWORD	将STRING型数据转换为DWORD型数据。	1242页 STRING_TO_DWORD(_E)
STRING_TO_DWORD_E		

■STRING型→INT型/DINT型转换

函数符号	处理内容	参阅目标
STRING_TO_INT	将STRING型数据转换为INT型数据。	1243页 STRING_TO_INT(_E)
STRING_TO_INT_E		
STRING_TO_DINT	将STRING型数据转换为DINT型数据。	1244页 STRING_TO_DINT(_E)
STRING_TO_DINT_E		

■STRING型→BCD型转换

函数符号	处理内容	参阅目标
STRING_TO_BCD	将STRING型数据转换为BCD型数据。	1246页 STRING_TO_BCD(_E)
STRING_TO_BCD_E		

■STRING型→REAL型转换

函数符号	处理内容	参阅目标
STRING_TO_REAL	将STRING型数据转换为REAL型数据。	1248页 STRING_TO_REAL(_E)
STRING_TO_REAL_E		

■STRING型→TIME型转换

函数符号	处理内容	参阅目标
STRING_TO_TIME	将STRING型数据转换为TIME型数据。	1251页 STRING_TO_TIME(_E)
STRING_TO_TIME_E		

■位数组→INT型/DINT型转换

函数符号	处理内容	参阅目标
BITARR_TO_INT	将通过位数组指定的位数转换为INT型数据。	1252页 BITARR_TO_INT(_E)
BITARR_TO_INT_E		
BITARR_TO_DINT	将通过位数组指定的位数转换为DINT型数据。	1253页 BITARR_TO_DINT(_E)
BITARR_TO_DINT_E		

■INT型/DINT型→位数组转换

函数符号	处理内容	参阅目标
INT_TO_BITARR	将INT型数据的低n位输出到位数组中。	1254页 INT_TO_BITARR(_E)
INT_TO_BITARR_E		
DINT_TO_BITARR	将DINT型数据的低n位输出到位数组中。	1255页 DINT_TO_BITARR(_E)
DINT_TO_BITARR_E		

■位数组的复制

函数符号	处理内容	参阅目标
CPY_BITARR	将位数组进行指定位容量的复制。	1256页 CPY_BITARR(_E)
CPY_BITARR_E		

■字标签的指定位读取/写入/复制

函数符号	处理内容	参阅目标
GET_BIT_OF_INT	读取字标签的指定位。	1257页 GET_BIT_OF_INT(_E)
GET_BIT_OF_INT_E		
SET_BIT_OF_INT	对字标签的指定位进行写入。	1259页 SET_BIT_OF_INT(_E)
SET_BIT_OF_INT_E		
CPY_BIT_OF_INT	将字标签的指定位复制到其它字标签的指定位。	1261页 CPY_BIT_OF_INT(_E)
CPY_BIT_OF_INT_E		

■起始数据的获取

函数符号	处理内容	参阅目标
GET_BOOL_ADDR	将指定数据的起始数据作为BOOL型、INT型或WORD型数据进行输出。	1263页 GET_BOOL_ADDR、 GET_INT_ADDR、GET_WORD_ADDR
GET_INT_ADDR		
GET_WORD_ADDR		

单数值变量函数

■绝对值

函数符号	处理内容	参阅目标
ABS	输出输入值的绝对值。	1264页 ABS(_E)
ABS_E		

■平方根

函数符号	处理内容	参阅目标
SQRT	对输入值的平方根进行运算。	1266页 SQRT(_E)
SQRT_E		

■自然对数运算

函数符号	处理内容	参阅目标
LN	输出输入值的自然对数运算结果。	1267页 LN(_E)
LN_E		

■常用对数运算

函数符号	处理内容	参阅目标
LOG	输出输入值的常用对数(以10为底的对数)的运算结果。	1268页 LOG(_E)
LOG_E		

■指数运算

函数符号	处理内容	参阅目标
EXP	输出输入值的指数运算结果。	1270页 EXP(_E)
EXP_E		

■SIN运算/COS运算/TAN运算

函数符号	处理内容	参阅目标
SIN	输出输入值的SIN(正弦)值。	1271页 SIN(_E)
SIN_E		
COS	输出输入值的COS(余弦)值。	1272页 COS(_E)
COS_E		
TAN	输出输入值的TAN(正切)值。	1273页 TAN(_E)
TAN_E		

■SIN⁻¹运算/COS⁻¹运算/TAN⁻¹运算

函数符号	处理内容	参阅目标
ASIN	输出输入值的SIN ⁻¹ (反正弦)值。	1274页 ASIN(_E)
ASIN_E		
ACOS	输出输入值的COS ⁻¹ (反正弦)值。	1275页 ACOS(_E)
ACOS_E		
ATAN	输出输入值的TAN ⁻¹ (反正弦)值。	1276页 ATAN(_E)
ATAN_E		

算术运算函数

■加法运算

函数符号	处理内容	参阅目标
ADD	输出输入值的和((s1)+(s2)+...+(s28))。	1277页 ADD(_E)
ADD_E		

■乘法运算

函数符号	处理内容	参阅目标
MUL	输出输入值的积((s1)×(s2)×...×(s28))。	1280页 MUL(_E)
MUL_E		

■减法运算

函数符号	处理内容	参阅目标
SUB	输出输入值的差((s1)-(s2))。	1282页 SUB(_E)
SUB_E		

■除法运算

函数符号	处理内容	参阅目标
DIV	输出输入值的商((s1)÷(s2))。	1285页 DIV(_E)
DIV_E		

■余数

函数符号	处理内容	参阅目标
MOD	输出输入值的余数((s1)÷(s2))。	1287页 MOD(_E)
MOD_E		

■幂

函数符号	处理内容	参阅目标
EXPT	输出输入值的幂。	1289页 EXPT(_E)
EXPT_E		

■代入

函数符号	处理内容	参阅目标
MOVE	输出输入值的代入。	1290页 MOVE(_E)
MOVE_E		

位移函数

■n位左移、右移

函数符号	处理内容	参阅目标
SHL	将输入值左移(n)位数后输出。	1292页 SHL(_E)
SHL_E		
SHR	将输入值右移(n)位数后输出。	1294页 SHR(_E)
SHR_E		

■n位左旋、右旋

函数符号	处理内容	参阅目标
ROL	将输入值左旋(n)位数后输出。	1296页 ROL(_E)
ROL_E		
ROR	将输入值右旋(n)位数后输出。	1298页 ROR(_E)
ROR_E		

位型布尔函数

■逻辑与、逻辑或、异或

函数符号	处理内容	参阅目标
AND	输出输入值的逻辑与。	1300页 AND(_E)、OR(_E)、XOR(_E)
AND_E		
OR		
OR_E	输出输入值的逻辑或。	
XOR		
XOR_E	输出输入值的异或。	

■逻辑否

函数符号	处理内容	参阅目标
NOT	输出输入值的逻辑否。	1303页 NOT(_E)
NOT_E		

选择函数

■选择值

函数符号	处理内容	参阅目标
SEL	输出选择的输入值。	1304页 SEL(_E)
SEL_E		

■最大值、最小值选择

函数符号	处理内容	参阅目标
MAX	输出输入值的最大值。	1306页 MAX(_E)、MIN(_E)
MAX_E		
MIN	输出输入值的最小值。	
MIN_E		

■上下限限位控制

函数符号	处理内容	参阅目标
LIMIT	输出上下限限位控制的输入值。	1308页 LIMIT(_E)
LIMIT_E		

■多路复用器

函数符号	处理内容	参阅目标
MUX	输出多个输入值中之一。	1311页 MUX(_E)
MUX_E		

比较函数

■比较

函数符号	处理内容	参阅目标
GT	输出输入值的数据比较结果。	1313页 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		1315页 NE(_E)
NE_E		

字符串函数

■字符串的长度检测

函数符号	处理内容	参阅目标
LEN	检测并输出输入的字符串的长度。	1317页 LEN(_E)
LEN_E		

■从字符串的左侧、右侧提取

函数符号	处理内容	参阅目标
LEFT	从输入的字符串数据的左侧开始输出指定字符。	1319页 LEFT(_E)、RIGHT(_E)
LEFT_E		
RIGHT	从输入的字符串数据的右侧开始输出指定字符。	
RIGHT_E		

■字符串的提取

函数符号	处理内容	参阅目标
MID	从输入的字符串的任意位置开始输出指定字符。	1322页 MID(_E)
MID_E		

■字符串的合并

函数符号	处理内容	参阅目标
CONCAT	合并字符串后输出。	1324页 CONCAT(_E)
CONCAT_E		

■字符串的插入

函数符号	处理内容	参阅目标
INSERT	在字符串之间插入字符串后输出。	1326页 INSERT(_E)
INSERT_E		

■字符串的删除

函数符号	处理内容	参阅目标
DELETE	删除字符串的任意范围后输出。	1328页 DELETE(_E)
DELETE_E		

■字符串的替换

函数符号	处理内容	参阅目标
REPLACE	替换字符串的任意范围后输出。	1330页 REPLACE(_E)
REPLACE_E		

■字符串的搜索

函数符号	处理内容	参阅目标
FIND	搜索字符串并输出搜索结果。	1333页 FIND(_E)
FIND_E		

时间数据类型函数

■加法运算

函数符号	处理内容	参阅目标
ADD_TIME	输出输入值(TIME型)的和((s1)+(s2))。	1335页 ADD_TIME(_E)
ADD_TIME_E		

■减法运算

函数符号	处理内容	参阅目标
SUB_TIME	输出输入值(TIME型)的差((s1)-(s2))。	1337页 SUB_TIME(_E)
SUB_TIME_E		

■乘法运算

函数符号	处理内容	参阅目标
MUL_TIME	输出输入值(TIME型)的积((s1)×(s2))。	1339页 MUL_TIME(_E)
MUL_TIME_E		

■除法运算

函数符号	处理内容	参阅目标
DIV_TIME	输出输入值(TIME型)的商((s1)÷(s2))。	1341页 DIV_TIME(_E)
DIV_TIME_E		

8.2 通用FB

双稳态FB

■双稳态FB(设置优先)

FB符号	处理内容	参阅目标
SR	判断2个输入值，输出1(TRUE)或者0(FALSE)。	1344页 SR(_E)
SR_E		

■双稳态FB(设置优先)

FB符号	处理内容	参阅目标
RS	判断2个输入值，输出1(TRUE)或者0(FALSE)。	1346页 RS(_E)
RS_E		

边缘检测FB

■上升沿检测

FB符号	处理内容	参阅目标
R_TRIG	检测信号的上升沿并输出脉冲信号。	1348页 R_TRIG(_E)
R_TRIG_E		

■下降沿检测

FB符号	处理内容	参阅目标
F_TRIG	检测信号的下降沿并输出脉冲信号。	1350页 F_TRIG(_E)
F_TRIG_E		

计数器FB

■升值计数器

FB符号	处理内容	参阅目标
CTU	对信号的上升沿次数进行递增计数。	1352页 CTU(_E)
CTU_E		

■贬值计数器

FB符号	处理内容	参阅目标
CTD	对信号的上升沿次数进行递减计数。	1354页 CTD(_E)
CTD_E		

■升值贬值计数器

FB符号	处理内容	参阅目标
CTUD	对信号的上升沿次数进行递增/递减计数。	1356页 CTUD(_E)
CTUD_E		

■计数器FB

FB符号	处理内容	参阅目标
COUNTER_FB_M	执行条件成立时，执行递增计数。	1359页 COUNTER_FB_M

定时器FB

■脉冲定时器

FB符号	处理内容	参阅目标
TP	在指定时间期间将信号置为ON。	1361页 TP(_E)
TP_E		

■ON延迟定时器

FB符号	处理内容	参阅目标
TON	指定的时间后将信号置为ON。	1363页 TON(_E)
TON_E		

■OFF延迟定时器

FB符号	处理内容	参阅目标
TOF	指定的时间后将信号置为OFF。	1365页 TOF(_E)
TOF_E		

■定时器FB

FB符号	处理内容	参阅目标
TIMER_10_FB_M	执行条件成立时，至设置的时间为止执行定时器计数。	1367页 TIMER_□_M
TIMER_100_FB_M		
TIMER_HIGH_FB_M		
TIMER_LOW_FB_M		
TIMER_CONT_FB_M		
TIMER_CONTHFB_M		

9 运动控制FB一览

以下介绍一览的阅读方法。

项目	内容
FB符号	表示FB名。
处理内容	表示FB的概要。
参阅目标	表示详细说明的参阅目标。

有关运动控制FB的概要请参阅下述章节。

☞ 1372页 运动控制FB的概要

9.1 管理系统FB

管理系统的运动控制FB一览如下所示。

轴组有效

FB符号	处理内容	参阅
MC_GroupEnable	将指定的轴组的状态从“0：轴组无效(GroupDisabled)”转变为“4：待机中(GroupStandby)”。	1463页 轴组有效

轴组无效

FB符号	处理内容	参阅
MC_GroupDisable	将指定的轴组的状态转变为“0：轴组无效(GroupDisabled)”。	1467页 轴组无效

允许运行

FB符号	处理内容	参阅
MC_Power	将指定的轴切换为允许运行状态。	1471页 允许运行

当前位置更改

FB符号	处理内容	参阅
MC_SetPosition	更改指定的轴的当前位置(指令位置、反馈位置)。	1475页 当前位置更改

转矩限制值

FB符号	处理内容	参阅
MCv_SetTorqueLimit	执行转矩限制值的更改。	1482页 转矩限制值

倍率修调值设置

FB符号	处理内容	参阅
MC_SetOverride	执行指定的轴的目标速度、目标加速度、目标减速度的更改。	1490页 倍率修调值设置

参数读取

FB符号	处理内容	参阅
MC_ReadParameter	进行设备的对象的读取。	1495页 参数读取

参数写入

FB符号	处理内容	参阅
MC_WriteParameter	进行设备的对象的写入。	1502页 参数写入

轴错误复位

FB符号	处理内容	参阅
MC_Reset	对轴的错误、警告进行复位。	1508页 轴错误复位

轴组错误复位

FB符号	处理内容	参阅
MC_GroupReset	对轴组及轴组中所属的各轴的错误、警告进行复位。	1511页 轴组错误复位

触摸探针有效

FB符号	处理内容	参阅
MC_TouchProbe	通过触发事件发生记录任意数据。	1515页 触摸探针有效

触摸探针无效

FB符号	处理内容	参阅
MC_AbortTrigger	将执行中的锁存置为无效。	1529页 触摸探针无效

凸轮表选择

FB符号	处理内容	参阅
MC_CamTableSelect	将指定的运算配置文件(凸轮数据)存储到展开区域中。	1534页 凸轮表选择

1周期当前值更改

FB符号	处理内容	参阅
MCv_ChangeCycle	更改指定的运算配置文件控制FB的1周期当前值。	1542页 1周期当前值更改

所有轴允许运行

FB符号	处理内容	参阅
MCv_AllPower	将所有轴切换为允许运行状态。	1550页 全部轴可运行

轴组倍率修调值设置

FB符号	处理内容	参阅
MC_GroupSetOverride	执行指定的轴组的目标速度、目标加速度、目标减速度的更改。	1553页 轴组倍率修调值设置

运动错误复位

FB符号	处理内容	参阅
MCv_MotionErrorReset	对运动系统的所有的错误、警告进行复位。	1558页 运动错误复位

控制运算周期同步

FB符号	处理内容	参阅
MCv_SyncOperationCycles	与设置了轴的控制运算周期的周期同步。	1562页 控制运算周期同步

高级同步控制1周期当前位置计算

FB符号	处理内容	参阅
MCv_AdvPositionPerCycleCalc	以指定的凸轮数据为基准，计算1周期当前位置。	1565页 高级同步控制1周期当前位置计算

高级同步控制凸轮指令当前位置计算

FB符号	处理内容	参阅
MCv_AdvCamSetPositionCalc	以指定的凸轮数据为基准，计算凸轮指令当前位置。	1568页 高级同步控制凸轮指令当前位置计算

数字凸轮开关输出

FB符号	处理内容	参阅
MC_DigitalCamSwitch	根据任意数据的值输出ON/OFF信号。	1571页 数字凸轮开关输出

9.2 运行系统FB

运行系统的运动控制FB一览如下所示。

原点复位

FB符号	处理内容	参阅
MC_Home	进行指定的轴的原点复位。	1577页 原点复位

强制停止

FB符号	处理内容	参阅
MC_Stop	对指定的轴进行减速停止。	1586页 强制停止

组强制停止

FB符号	处理内容	参阅
MC_GroupStop	对指定的轴组进行减速停止。	1591页 组强制停止

绝对值定位

FB符号	处理内容	参阅
MC_MoveAbsolute	指定绝对位置的目标位置，执行定位。	1597页 绝对值定位

相对值定位

FB符号	处理内容	参阅
MC_MoveRelative	指定相对位置的移动量，执行定位。	1610页 相对值定位

JOG运行

FB符号	处理内容	参阅
MCv_Jog	按照指令速度执行JOG运行。	1619页 JOG运行

速度控制

FB符号	处理内容	参阅
MC_MoveVelocity	将驱动器模块切换为csv，按照指定的速度进行速度控制。	1629页 速度控制

转矩控制

FB符号	处理内容	参阅
MC_TorqueControl	将驱动器模块切换为cst，按照指定的目标转矩进行转矩控制。	1637页 转矩控制

速度控制(包含位置循环)

FB符号	处理内容	参阅
MCv_SpeedControl	执行包含位置循环的速度控制。	1647页 速度控制(包含位置循环)

绝对值直线插补控制

FB符号	处理内容	参阅
MCv_MoveLinearInterpolateAbsolute	对指定的轴组的绝对位置的目标位置进行指定，并通过直线插补控制执行定位。	1656页 绝对值直线插补控制

相对值直线插补控制

FB符号	处理内容	参阅
MCv_MoveLinearInterpolateRelative	对指定的轴组的相对位置的移动量进行指定，并通过直线插补控制执行定位。	1670页 相对值直线插补控制

绝对值圆弧插补控制

FB符号	处理内容	参阅
MCv_MoveCircularInterpolateAbsolute	使用设置的轴组的配置轴，设置绝对位置的终点及辅助点，通过2轴的圆弧插补执行定位。	1683页 绝对值圆弧插补控制

相对值圆弧插补控制

FB符号	处理内容	参阅
MCv_MoveCircularInterpolateRelative	使用设置的轴组的配置轴，从启动时的当前位置向终点及辅助点设置相对位置，通过2轴的圆弧插补执行定位。	1699页 相对值圆弧插补控制

凸轮动作开始

FB符号	处理内容	参阅
MC_CamIn	按照指定的凸轮数据开始凸轮动作。	1716页 凸轮动作开始

齿轮动作开始

FB符号	处理内容	参阅
MC_GearIn	按照指定的齿轮比开始齿轮动作。	1742页 齿轮动作开始

加减法定位

FB符号	处理内容	参阅
MC_CombineAxes	将对指定的主轴2轴的移动量进行了加减法计算的值作为指令位置进行定位。	1752页 加减法定位

齿隙补偿滤波器

FB符号	处理内容	参阅
MCv_BacklashCompensationFilter	进行按照移动方向，对机械系统的齿隙量进行补偿的滤波器处理。	1762页 齿隙补偿滤波器

平滑滤波器

FB符号	处理内容	参阅
MCv_SmoothingFilter	按照指定的频率，进行滤波器处理。	1776页 平滑滤波器

移动方向限制滤波器

FB符号	处理内容	参阅
MCv_DirectionFilter	进行对设置的移动方向进行移动限制的滤波器处理。	1786页 移动方向限制滤波器

速度限制滤波器

FB符号	处理内容	参阅
MCv_SpeedLimitFilter	进行将速度限制为设置的限制值的滤波器处理。	1795页 速度限制滤波器

轴控制开始等待

FB符号	处理内容	参阅
MCv_MoveWait	通过允许执行信号的输入即时执行指定轴多重起动了的运行系统FB的运行。	1803页 轴控制开始等待

轴组控制开始等待

FB符号	处理内容	参阅
MCv_GroupMoveWait	通过允许执行信号的输入即时执行指定轴组多重起动了的运行系统FB的运行。	1810页 轴组控制开始等待

高级同步控制

FB符号	处理内容	参阅
MCv_AdvancedSync	按照指定的高级同步控制设置开始同步控制。	1817页 高级同步控制

多轴定位数据运行

FB符号	处理内容	参阅
MCv_MovePositioningData	按照指定的定位数据，对轴组执行动作。	1823页 多轴定位数据运行

绝对位置跟踪控制

FB符号	处理内容	参阅
MCv_PositionControl	按照限制速度、加速度、减速度、Jerk将指定的轴移动到目标位置的功能。此外，即使指令到达目标位置，也可通过重启、连续更新来更改目标位置及移动轴。	1832页 绝对位置跟踪控制

运动循环位置控制

FB符号	处理内容	参阅
MCv_CyclicPosition	按照指定的目标位置将目标位置发送到驱动器模块。	1844页 运动循环位置控制

运动循环速度控制

FB符号	处理内容	参阅
MCv_CyclicVelocity	按照指定的目标速度将目标速度发送到驱动器模块。	1853页 运动循环速度控制

运动循环转矩控制

FB符号	处理内容	参阅
MCv_CyclicTorque	按照指定的目标位置将目标转矩发送到驱动器模块。	1857页 运动循环转矩控制

9.3 一般FB

进行轴控制的运动控制FB一览如下所示。

配置文件读取

FB符号	处理内容	参阅
MCv_ReadProfileData	从展开区域或文件读取指定的运算配置文件。	1863页 配置文件读取

配置文件写入

FB符号	处理内容	参阅
MCv_WriteProfileData	将指定的运算配置文件写入到展开区域或文件中。	1877页 配置文件写入

10 网络FB一览

以下介绍一览的阅读方法。

项目	内容
FB符号	表示FB名。
处理内容	表示FB的概要。
参阅目标	表示详细说明的参阅目标。

10.1 以太网用FB

连接的建立/断开

FB符号	处理内容	参阅
M+MXF-SQ_ConnectionOpen	打开(建立)连接。	1892页 M+MXF-SQ_ConnectionOpen
M+MXF-SQ_ConnectionClose	关闭(断开)连接。	1895页 M+MXF-SQ_ConnectionClose

套接字通信用FB

FB符号	处理内容	参阅
M+MXF-SQ_Recv_Socket	读取指定的连接中通过套接字通信接收的数据。	1897页 M+MXF-SQ_Recv_Socket
M+MXF-SQ_Send_Socket	将套接字通信的数据发送到指定连接的对象设备。	1900页 M+MXF-SQ_Send_Socket

SLMP帧用FB

FB符号	处理内容	参阅
M+MXF-SQ_SLMP_DeviceRead_IP	通过IP地址指定读取SLMP对应设备的软元件数据。	1903页 M+MXF-SQ_SLMP_DeviceRead_IP
M+MXF-SQ_SLMP_DeviceWrite_IP	通过IP地址指定将软元件数据写入至SLMP对应设备。	1910页 M+MXF-SQ_SLMP_DeviceWrite_IP
M+MXF-TSN_SLMP_DeviceRead_IP	通过IP地址指定读取SLMP对应设备的软元件数据。	1917页 M+MXF-TSN_SLMP_DeviceRead_IP
M+MXF-TSN_SLMP_DeviceWrite_IP	通过IP地址指定将软元件数据写入至SLMP对应设备。	1924页 M+MXF-TSN_SLMP_DeviceWrite_IP

10.2 CC-Link IE TSN用FB

SLMP帧的远程复位

FB符号	处理内容	参阅
M+MXF-TSN_RemoteReset2_IP	通过IP地址指定，向对象站发送SLMP请求的远程复位。	1931页 M+MXF-TSN_RemoteReset2_IP

第3部分 顺控程序指令/基本指令

本部分由下述章构成。

11 顺控程序指令

12 基本指令

11 顺控程序指令

11.1 触点指令

运算开始、串联连接、并联连接

LD、LDI、AND、ANI、OR、ORI

- LD: 常开触点运算开始指令/LDI: 常闭触点运算开始指令

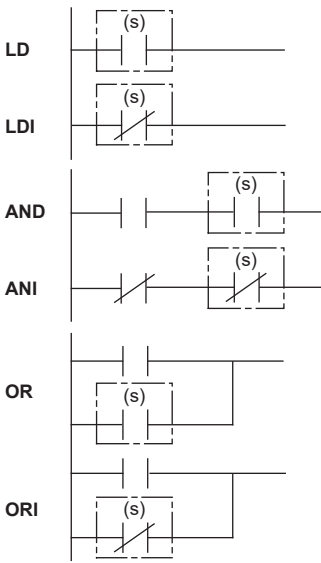
将指定软元件的ON/OFF信息作为运算结果。

- AND: 常开触点串联连接指令/ANI: 常闭触点串联连接指令

将指定软元件的ON/OFF信息与至当时为止的运算结果的AND运算作为运算结果。

- OR: 1个常开触点的并联连接指令/ORI: 1个常闭触点的并联连接指令

将指定软元件的ON/OFF信息与至当时为止的运算结果的OR运算作为运算结果。

梯形图	ST
	不对应。

FBD/LD

不对应。

■ 执行条件

指令	执行条件
LD	常时执行
LDI	
AND	
ANI	
OR	
ORI	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	作为触点使用的软元件	—	位	ANY_BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数				其它(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)	○	○	○	○	—	○	—	○	—	—	—	○	

功能

■LD、LDI

LD指令是常开触点运算开始指令，LDI指令是常闭触点运算开始指令，获取指定软元件值的ON/OFF信息*1，作为运算结果。

*1 字软元件的位指定时，根据指定位的1/0而ON/OFF。

■AND、ANI

AND指令是常开触点串联连接指令，ANI指令是常闭触点串联连接指令，获取指定位软元件的ON/OFF信息*1，与至当时为止的运算结果进行AND运算，将该值作为运算结果。

*1 字软元件的位指定时，根据指定位的1/0而ON/OFF。

• 在工程工具的梯形图编辑模式中其情况如下所示。

- 写入：AND指令、ANI指令为串联连接的情况下，最多可以创建24级的梯形图。
- 读取：AND指令、ANI指令为串联连接的情况下，最多可以显示24级的梯形图。超过了24级的情况下最多显示24级。

■OR、ORI

OR指令是1个常开触点的并联连接指令、ORI指令是1个常闭触点的并联连接指令，获取指定软元件的ON/OFF信息*1，与至当时为止的运算结果进行OR运算，将该值作为运算结果。

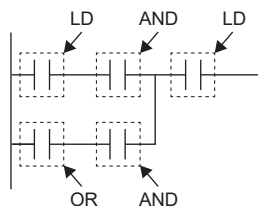
*1 字软元件的位指定时，根据指定位的1/0而ON/OFF。

• 在工程工具的梯形图编辑模式中其情况如下所示。

- 写入：OR指令、ORI指令最多可以创建23个连续连接的梯形图。
- 读取：OR指令、ORI指令最多可以显示23个连续连接的梯形图。超过了23个的梯形图将无法显示。

■由LD、LDI、AND、ANI、OR、ORI组合而成的动作

下述示例表示由LD、AND、OR组合而成的动作。LDI、ANI、ORI也相同。



要点

字软元件的位指定时，位的指定是以16进制数进行。(例如，D0的b11将变为“D0.0B”。)

出错

没有运算错误。

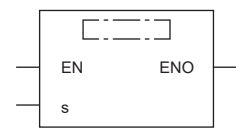
脉冲运算开始、脉冲串联连接、脉冲并联连接

LDP、LDF、ANDP、ANDF、ORP、ORF

- LDP: 上升沿脉冲运算开始指令
 仅在指定位软元件的上升沿时(OFF→ON)导通。
- LDF: 下降沿脉冲运算开始指令
 仅在指定位软元件的下降沿时(ON→OFF)导通。
- ANDP: 上升沿脉冲串联连接指令/ANDF: 下降沿脉冲串联连接指令
 与至当时为止的运算结果进行AND运算。
- ORP: 上升沿脉冲并联连接指令/ORF: 下降沿脉冲并联连接指令
 与至当时为止的运算结果进行OR运算。

梯形图	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



■ 执行条件

指令	执行条件
LDP LDF ANDP ANDF ORP ORF	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	作为触点使用的软元件	—	位	ANY_BOOL
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

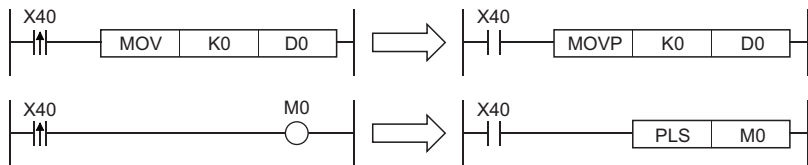
■可用软元件

操作数	位		字		双字		间接指定	常数			其它(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)	○	○	○	○	—	○	—	○	—	—	—	○

功能

■LDP、LDF

• LDP指令是上升沿脉冲运算开始指令，仅在指定位软元件的上升沿时(OFF→ON)导通。字软元件的位指定时，仅在指定位由0→1变化时导通。仅LDP指令的情况下，与ON中执行指令的脉冲化指令(□P)相同。将使用了LDP指令的梯形图替换为不使用LDP指令的梯形图时的情况如下所示。



- LDF指令是下降沿脉冲运算开始指令，在指定位软元件的下降沿时(ON→OFF)导通。字软元件的位指定时，指定位由1→0变化时导通。
- 将LDP指令通过ST及FBD/LD使用的情况下，指定位软元件(s)的上升沿时(OFF→ON)，ENO将ON。
- 将LDF指令通过ST及FBD/LD使用的情况下，指定位软元件(s)的下降沿时(ON→OFF)，ENO将ON。
- 通过ST使用了LDP、LDF指令的情况下，应将EN指定为始终ON。
- 通过FBD/LD使用了LDP、LDF指令的情况下，应将EN指定为左母线，或始终ON的变量部件/常数部件。

■ANDP、ANDF

• ANDP指令是上升沿脉冲串联连接指令，ANDF指令是下降沿脉冲串联连接指令，与至当时为止的运算结果进行AND运算并作为运算结果。ANDP指令、ANDF指令中使用的ON/OFF信息如下所示。

ANDP、ANDF中指定的软元件		ANDP的状态	ANDF的状态
位软元件	字软元件的位指定		
OFF→ON	0→1	ON	OFF
OFF	0	OFF	OFF
ON	1	OFF	OFF
ON→OFF	1→0	OFF	ON

- 将ANDP指令通过ST及FBD/LD使用的情况下，EN与指定位软元件(s)上升沿的AND运算结果为ON时，ENO将ON。EN不成为执行条件。
- 将ANDF指令通过ST及FBD/LD使用的情况下，EN与指定位软元件(s)下降沿的AND运算结果为ON时，ENO将ON。EN不成为执行条件。

■ORP、ORF

- ORP指令是上升沿脉冲并联连接指令、ORF指令是下降沿脉冲并联连接指令，与至当时为止的运算结果进行OR运算并作为运算结果。ORP指令、ORF指令中使用的ON/OFF信息如下所示。

ORP、ORF中指定的软元件		ORP的状态	ORF的状态
位软元件	字软元件的位指定		
OFF→ON	0→1	ON	OFF
OFF	0	OFF	OFF
ON	1	OFF	OFF
ON→OFF	1→0	OFF	ON

- 将ORP指令通过ST及FBD/LD使用的情况下，EN与指定位软元件(s)上升沿的OR运算结果为ON时，ENO将ON。EN不成为执行条件。
- 将ORF指令通过ST及FBD/LD使用的情况下，EN与指定位软元件(s)下降沿的OR运算结果为ON时，ENO将ON。EN不成为执行条件。

■由LDP、LDF、ANDP、ANDF、ORP、ORF组合而成的动作

由LDP、LDF、ANDP、ANDF、ORP、ORF组合而成的动作示例与由LD、AND、OR组合而成的动作示例相同。(☞ 200页 LD、LDI、AND、ANI、OR、ORI)

出错

没有运算错误。

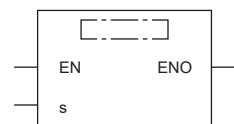
脉冲否定运算开始、脉冲否定串联连接、脉冲否定并联连接

LDPI、LDFI、ANDPI、ANDFI、ORPI、ORFI

- LDPI: 上升沿脉冲否定运算开始指令
在指定软元件的OFF时、ON时、下降沿时(ON→OFF)的情况下导通。
- LDFI: 下降沿脉冲否定运算开始指令
在指定软元件的上升沿时(OFF→ON)、OFF时、ON时导通。
- ANDPI: 上升沿脉冲否定串联连接指令/ANDFI: 下降沿脉冲否定串联连接指令
与至当时为止的运算结果进行AND运算。
- ORPI: 上升沿脉冲否定并联连接指令/ORFI: 下降沿脉冲否定并联连接指令
与至当时为止的运算结果进行OR运算。

梯形图	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



■执行条件

指令	执行条件
LDPI LDFI ANDPI ANDFI ORPI ORFI	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	作为触点使用的软元件	—	位	ANY_BOOL
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它(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)	○	○	○	○	—	○	—	○	—	—	—	○	

功能

■LDPI、LDFI

- LDPI指令是上升沿脉冲否定运算开始指令，在指定位软元件的OFF时、ON时、下降沿时(ON→OFF)的情况下导通。字软元件的位指定时，在指定位为0、1、发生了1→0变化的情况下导通。
- LDFI指令是下降沿脉冲否定运算开始指令，在指定位软元件的上升沿时(OFF→ON)、OFF时、ON时的情况下导通。字软元件的位指定时，在指定位为0、1、发生了0→1变化的情况下导通。LDPI指令、LDFI指令中使用的ON/OFF信息如下所示。

LDPI、LDFI中指定的软元件		LDPI的状态	LDFI的状态
位软元件	字软元件的位指定		
OFF→ON	0→1	OFF	ON
OFF	0	ON	ON
ON	1	ON	ON
ON→OFF	1→0	ON	OFF

- 将LDPI指令通过ST及FBD/LD使用的情况下，指定位软元件(s)的上升沿以外时，ENO将ON。
- 将LDFI指令通过ST及FBD/LD使用的情况下，指定位软元件(s)的下降沿以外时，ENO将ON。
- 通过ST使用了LDPI、LDFI指令的情况下，应将EN指定为始终ON。
- 通过FBD/LD使用了LDPI、LDFI指令的情况下，应将EN指定为左母线，或始终ON的变量部件/常数部件。

■ANDPI、ANDFI

- ANDPI指令是上升沿脉冲否定串联连接指令、ANDFI指令是下降沿脉冲否定串联连接指令，与至当时为止的运算结果进行AND运算并作为运算结果。ANDPI指令、ANDFI指令中使用的ON/OFF信息如下所示。

ANDPI、ANDFI中指定的软元件		ANDPI的状态	ANDFI的状态
位软元件	字软元件的位指定		
OFF→ON	0→1	OFF	ON
OFF	0	ON	ON
ON	1	ON	ON
ON→OFF	1→0	ON	OFF

- 将ANDPI指令通过ST及FBD/LD使用的情况下，EN与指定位软元件(s)上升沿的AND运算结果为ON以外时，ENO将ON。EN不成为执行条件。
- 将ANDFI指令通过ST及FBD/LD使用的情况下，EN与指定位软元件(s)下降沿的AND运算结果为ON以外时，ENO将ON。EN不成为执行条件。

■ORPI、ORFI

- ORPI指令是上升沿脉冲否定并联连接指令、ORFI指令是下降沿脉冲否定并联连接指令，与至当时为止的运算结果进行OR运算并作为运算结果。ORPI指令、ORFI指令中使用的ON/OFF信息如下所示。

ORPI、ORFI中指定的软元件		ORPI的状态	ORFI的状态
位软元件	字软元件的位指定		
OFF→ON	0→1	OFF	ON
OFF	0	ON	ON
ON	1	ON	ON
ON→OFF	1→0	ON	OFF

- 将ORPI指令通过ST及FBD/LD使用的情况下，EN与指定位软元件(s)上升沿的OR运算结果为ON以外时，ENO将ON。EN不成为执行条件。
- 将ORFI指令通过ST及FBD/LD使用的情况下，EN与指定位软元件(s)下降沿的OR运算结果为ON以外时，ENO将ON。EN不成为执行条件。

■由LDPI、LDFI、ANDPI、ANDFI、ORPI、ORFI组合而成的动作

由LDPI、LDFI、ANDPI、ANDFI、ORPI、ORFI组合而成的动作示例与由LD、AND、OR组合而成的动作示例相同。(☞ 200页 LD、LDI、AND、ANI、OR、ORI)

出错

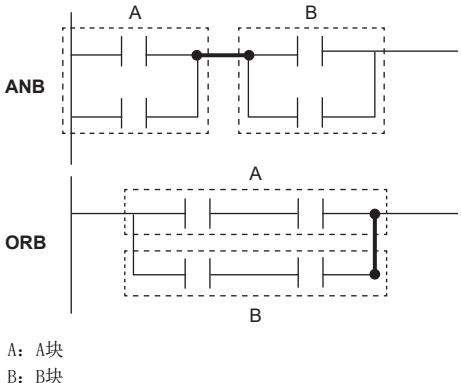
没有运算错误。

11.2 合并指令

梯形图块串联连接、并联连接

ANB、ORB

- ANB：梯形图块串联连接指令
进行A块与B块的AND运算。
- ORB：梯形图块并联连接指令
进行A块与B块的OR运算。

梯形图	ST
 <p>A: A块 B: B块</p>	不对应。

FBD/LD

不对应。

■执行条件

指令	执行条件
ANB ORB	常时执行

功能

■ANB

- 进行A块与B块的AND运算，并作为运算结果。
- ANB指令的符号不是触点符号，而是连接符号。

■ORB

- 进行A块与B块的OR运算，并作为运算结果。
- ORB指令对2触点及其以上的梯形图块进行并联连接。仅1个触点的并联连接使用OR指令、ORI指令，无需ORB指令。
- ORB指令的符号不是触点符号，而是连接符号。

出错

没有运算错误。

运算结果取反

INV

对到INV指令之前为止的运算结果进行取反。

梯形图	ST
	ENO:=INV(EN);

FBD/LD

■执行条件

指令	执行条件
INV	常时执行

功能

- 对到INV指令之前为止的运算结果进行取反。

到INV指令之前为止的运算结果	执行INV指令后的运算结果
OFF	ON
ON	OFF

出错

没有运算错误。

要点

- INV指令是以到INV指令之前为止的运算结果执行动作，因此应与AND(☞200页 运算开始、串联连接、并联连接)在同一位置使用。INV指令不能在LD、OR(☞200页 运算开始、串联连接、并联连接)的位置使用。
- 使用了梯形图块的情况下，以梯形图块的范围对运算结果进行取反。并用INV指令及ANB指令使梯形图动作的情况下，应注意取反的范围。



虚线部：取反范围

关于ANB指令的详细内容，请参阅下述内容。

☞208页 ANB、ORB

运算结果脉冲化

MEP、MEF

- MEP：运算结果脉冲化指令(上升沿)

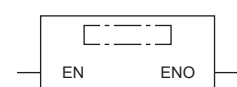
在MEP指令之前的运算结果的上升沿时(OFF→ON)导通。

- MEF：运算结果脉冲化指令(下降沿)

在MEF指令之前的运算结果的下降沿时(ON→OFF)导通。

梯形图	ST
	ENO:=MEP(EN); ENO:=MEF(EN);

FBD/LD



■执行条件

指令	执行条件
MEP MEF	常时执行

功能

■MEP

- 在MEP指令之前的运算结果为上升沿时(OFF→ON)，变为ON(导通状态)。MEP指令之前的运算结果为上升沿以外的情况下，变为OFF(非导通状态)。
- 使用MEP指令时，对多个触点进行了串联连接的情况下，脉冲化处理将易于进行。

■MEF

- 在MEF指令之前的运算结果为下降沿时(ON→OFF)，变为ON(导通状态)。MEF指令之前的运算结果为下降沿以外的情况下，变为OFF(非导通状态)。
- 使用MEF指令时，对多个触点进行了串联连接的情况下，脉冲化处理将易于进行。

出错

没有运算错误。

要点

- 对于MEP指令、MEF指令，如果通过子程序及FOR~NEXT指令等进行变址修饰后的触点的脉冲化，将可能无法正常动作。通过子程序、FOR~NEXT指令进行变址修饰后的触点的脉冲化的情况下，请参阅下述内容。
☞ 211页 EGP、EGF
- 对于MEP指令、MEF指令，以从MEP指令、MEF指令之前的LD指令开始到MEP指令、MEF指令之前为止的运算结果执行动作，因此应在与AND(☞ 200页 运算开始、串联连接、并联连接)在同一位置使用。MEP指令、MEF指令不能在LD、OR(☞ 200页 运算开始、串联连接、并联连接)的位置使用。

变址继电器运算结果脉冲化

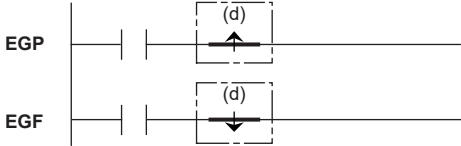
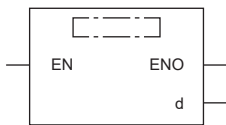
EGP、EGF

- EGP：变址继电器运算结果脉冲化指令(上升沿)



将EGP指令之前的运算结果通过变址继电器(V)存储。在运算结果的上升沿时(OFF→ON)导通。

- EGF：变址继电器运算结果脉冲化指令(下降沿)

将EGF指令之前的运算结果通过变址继电器(V)存储。在运算结果的下降沿时(ON→OFF)导通。

梯形图	ST
	ENO:=EGP(EN, d); ENO:=EGF(EN, d);
FBD/LD	
	

■执行条件

指令	执行条件
EGP	
EGF	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	存储运算结果的变址继电器编号	—	位	ANY_BOOL*1
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能使用软元件(V)中分配的位型标签。

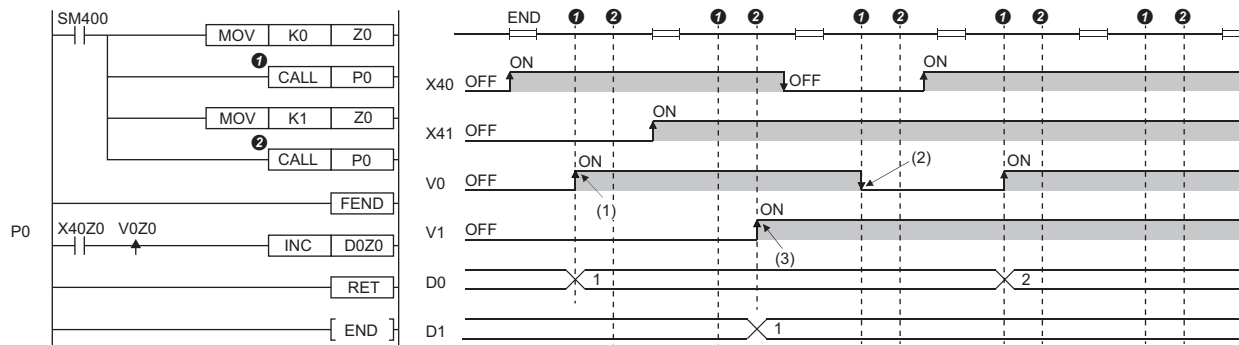
■可用软元件

操作数	位		字		双字		间接指定	常数				其它(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□\□	Z	LT、LST、LC		LZ	K、H	E	\$	
(d)	—	—	—	—	—	—	—	—	—	—	—	○

功能

EGP

- 将EGP指令之前的运算结果通过变址继电器(V)存储。
- 在EGP指令之前的运算结果为上升沿时(OFF→ON)，变为ON(导通状态)。EGP指令之前的运算结果为上升沿以外(ON→ON、ON→OFF、OFF→OFF)的情况下，变为OFF(非导通状态)。
- 对于EGP指令，在通过子程序及FOR~NEXT指令进行变址修饰后的程序的脉冲运算的情况下使用。
- EGP指令不能在AND指令处理中使用。
- 在子程序中使用了EGP指令的情况下，其动作如下所示。



- (1) X40的上升沿时置为ON。
- (2) X40的下降沿变为OFF。
- (3) X41的上升沿时置为ON。

EGF

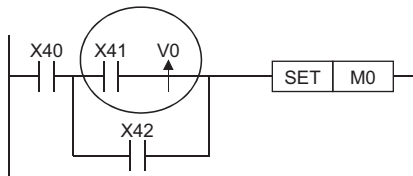
- 将EGF指令之前的运算结果通过变址继电器(V)存储。
- 在EGF指令之前的运算结果为下降沿时(ON→OFF)，变为ON(导通状态)。在EGF指令之前的运算结果为下降沿以外(OFF→ON、ON→ON、OFF→OFF)的情况下，变为OFF(非导通状态)。
- 对于EGF指令，通过子程序及FOR~NEXT指令进行变址修饰后的程序的脉冲运算的情况下使用。
- 对于EGF指令，不能在AND指令处理中使用。

出错

没有运算错误。

要点

- 对于EGP指令、EGF指令，以从位于EGP指令、EGF指令之前的LD指令开始到EGP指令、EGF指令之前为止的运算结果执行动作，因此应与AND(200页 运算开始、串联连接、并联连接)在同一位置使用。EGP指令、EGF指令不能在LD、OR(200页 运算开始、串联连接、并联连接)的位置使用。
- 对于EGP指令、EGF指令，不能在下述梯形图块的位置使用。



11.3 输出指令

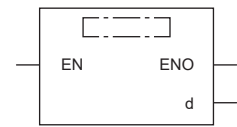
OUT(定时器、计数器、报警器除外)

OUT

将运算结果输出到指定的软元件中。

梯形图	ST
	ENO:=OUT(EN, d);

FBD/LD



■执行条件

指令	执行条件
OUT	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	ON/OFF的软元件编号	—	位	ANY_BOOL
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数			其它 (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	○	○*2	○	—	—	—	○	—	—	—	○

*1 使用F的情况下，请参阅下述指令。

☞ 225页 OUT F

*2 使用T、ST的情况下，请参阅下述指令。

☞ 215页 OUT T、OUTH T、OUT ST、OUTH ST

使用C的情况下，请参阅下述指令。

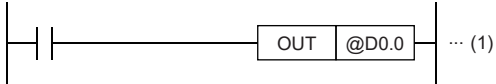
☞ 221页 OUT C

功能

- 将OUT指令之前的运算结果输出到指定的软元件中。

条件	运算结果	线圈/指定位
使用位软元件时	OFF	OFF
	ON	ON
字软元件的位指定时	OFF	0
	ON	1

- 使用间接指定的情况下，按以下方式进行位指定。



(1) 输出到D0中存储的间接地址的第0位。

出错

没有运算错误。

定时器

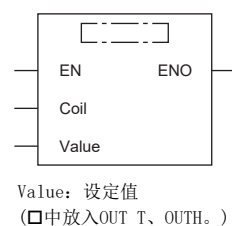
OUT T、OUTH T、OUT ST、OUTH ST

- OUT T: 低速定时器指令
- OUTH T: 高速定时器指令
- OUT ST: 低速累计定时器指令
- OUTH ST: 高速累计定时器指令

OUT指令之前的运算结果为ON时，线圈将ON并进行定时器的计测。如果时限到，常开触点将导通，常闭触点将变为非导通。

梯形图	ST
<p>Value: 设定值</p>	ENO:=OUT_T(EN,Coil,Value); ENO:=OUTH(EN,Coil,Value);

FBD/LD



■执行条件

指令	执行条件
OUT T OUTH T OUT ST OUTH ST	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	定时器软元件或定时器型标签	—	位	ANY_BOOL
Coil				
Value	定时器的设定值	0~65535	无符号BIN16位*1	ANY16*1
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 ST、FBD/LD的情况下，数据类型将变为ANY_INT。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它(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	—	—	—	—	—	—	—	—	—
Coil	—	—	—	—	—	—	—	—	—	—	—	—
Value	—	—	○*2	○	—	—	—	—	○*3	—	—	—

*1 只能使用T、ST。

*2 不能使用T、ST、C。

*3 只能使用10进制常数(K)。

功能

- OUT指令之前的运算结果为ON时，(d) (ST、FBD/LD的情况下为Coil)中指定的定时器的线圈将ON并进行计测直至设定值为止。如果时限到(当前值 \geq 设定值)，常开触点将导通，常闭触点将变为非导通。
- OUT指令之前的运算结果由ON \rightarrow OFF变化时其情况如下所示。

定时器的类型	定时器线圈	定时器的当前值	时限到前		时限到后	
			常开触点	常闭触点	常开触点	常闭触点
低速定时器	OFF	0	非导通	导通	非导通	导通
高速定时器						
低速累计定时器	OFF	保持当前值	非导通	导通	导通	非导通
高速累计定时器						


- 时限到后，通过RST指令进行累计定时器的当前值的清除及触点的OFF。
- 设定值为0时，执行OUT指令时将时限到。
- 执行OUT指令时，进行下述处理。
 - OUT T指令、OUTH T指令、OUT ST指令、OUTH ST指令的线圈的ON/OFF
 - OUT T指令、OUTH T指令、OUT ST指令、OUTH ST指令的触点的ON/OFF
 - OUT T指令、OUTH T指令、OUT ST指令、OUTH ST指令的当前值的更改
- OUT T指令、OUTH T指令、OUT ST指令、OUTH ST指令为ON时，通过JMP指令等跳过了OUT T指令的情况下，将不进行当前值的更新及触点的ON/OFF。
- 将同一OUT T指令、OUTH T指令、OUT ST指令、OUTH ST指令在同一扫描内执行了2次或其以上的情况下，将按照执行的次数进行当前值更新。

要点

定时器的时限设置是在工程工具的参数设置中进行。

- 低速定时器/低速累计定时器：1~10000ms (默认：100ms，设置单位：1ms)
- 高速定时器/高速累计定时器：0.01~100.0ms (默认：10.0ms，设置单位：0.01ms)

关于定时器的计数方法，请参阅下述手册。

所使用的控制器的用户手册

注意事项

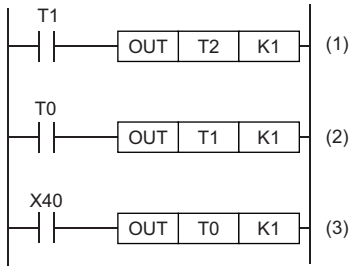
创建通过定时器的触点进行其它定时器的计测的程序的情况下，应从后计测的定时器开始依次编程。

定时器的设定值比扫描时间短的情况下，如果按计测顺序进行编程，所有的定时器将在同一扫描中变为ON。

- 设定值短于扫描时间时
- 设定值为1的情况下

例

将T0~T2的定时器按照从后计测的定时器开始的顺序进行了编程的情况下



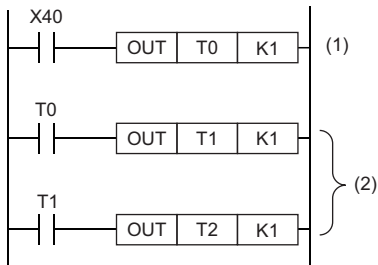
(3) X40为ON时，T0的定时器将开始计测。

(2) 从T0的触点变为ON的下一个扫描起，T1的定时器将开始计测。

(1) 从T1的触点变为ON的下一个扫描起，T2的定时器将开始计测。

例

将T0~T2的定时器按照计测顺序进行了编程的情况下



(1) X40为ON时，T0的定时器将开始计测。

(2) T0的触点变为ON时，T1、T2的定时器的触点也将变为ON。

出错

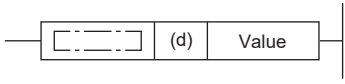
没有运算错误。

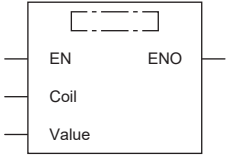
长定时器

OUT LT、OUT LST

- OUT LT：低速长定时器指令
- OUT LST：低速长累计定时器指令

OUT指令之前的运算结果为ON时，线圈将ON并进行定时器的计测。如果时限到，常开触点将导通，常闭触点将变为非导通。

梯形图	ST
 <p>Value: 设定值</p>	ENO:=OUT_T(EN,Coil,Value);

FBD/LD
 <p>Value: 设定值 (□中放入OUT_T。)</p>

■执行条件

指令	执行条件
OUT LT OUT LST	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	长定时器软件元件或长定时器型标签	—	位	ANY_BOOL
Coil				
Value	长定时器的设定值	0~4294967295	无符号BIN32位*1	ANY32*1
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 ST、FBD/LD的情况下，数据类型将变为ANY_INT。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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	—	—	—	—	—	—	—	—	—	—	—	—
Value	—	—	○*2	○	—	—	—	—	—	○*3	—	—

- *1 只能使用LT、LST。
- *2 不能使用T、ST、C。
- *3 只能使用10进制常数(K)。

功能

- OUT指令之前的运算结果为ON时，(d) (ST、FBD/LD的情况下为Coil)中指定的长定时器的线圈将ON并进行计测直至设定值为止。如果时限到(当前值 \geq 设定值)，常开触点将导通，常闭触点将变为非导通。
- OUT指令之前的运算结果由ON \rightarrow OFF变化时其情况如下所示。

定时器的类型	定时器线圈	定时器的当前值	时限到前		时限到后	
			常开触点	常闭触点	常开触点	常闭触点
长定时器	OFF	0	非导通	导通	非导通	导通
长累计定时器	OFF	保持当前值	非导通	导通	导通	非导通


- 时限到后，通过RST指令进行长累计定时器的当前值清除及触点的OFF。
- 设定值为0时，执行OUT指令时将时限到。
- 执行OUT指令时，进行下述处理。
 - OUT LT指令、OUT LST指令的线圈的ON/OFF
 - OUT LT指令、OUT LST指令的触点的ON/OFF
 - OUT LT指令、OUT LST指令的当前值的更改
- OUT LT指令、OUT LST指令为ON时，通过JMP指令等跳过了OUT LT指令的情况下，将不进行当前值的更新及触点的ON/OFF。
- 将同一OUT LT指令、OUT LST指令在同一扫描内执行了2次或其以上的情况下，将按照执行的次数进行当前值的更新。

要点

定时器的时限设置是在工程工具的参数设置中进行。

- 长定时器/长累计定时器：0.001~1000ms (默认：0.001ms，设置单位：0.001ms)

关于定时器的计数方法，请参阅下述手册。

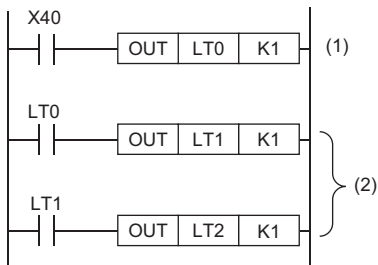
所使用的控制器的用户手册

注意事项

创建通过长定时器的触点进行其它长定时器的计测的程序的情况下，应按照从计测的长定时器开始的顺序进行编程。按照从后面计测的长定时器开始的顺序进行编程的话，定时器的计测误差有可能变大。长定时器的设定值比扫描时间短的情况下，变为如下状态。

例

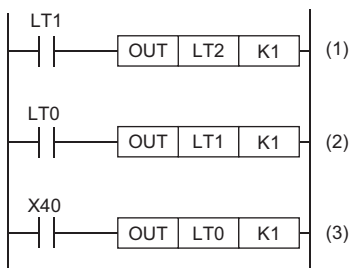
将LT0~LT2的定时器按照计测顺序进行了编程的情况下



- (1) X40变为ON时，LT0的定时器将开始计测。
- (2) LT0的触点变为ON时，按LT1、LT2的顺序定时器将开始计测。

例

将LT0~LT2的定时器按照从后计测的长定时器开始的顺序进行建立编程的情况下



- (3) X40为ON时，LT0的定时器将开始计测。
- (2) 从LT0的触点变为ON的下一个扫描起，LT1的定时器将开始计测。
- (1) 从LT1的触点变为ON的下一个扫描起，LT2的定时器将开始计测。

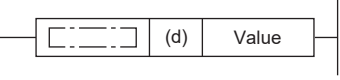
出错

没有运算错误。

计数器

OUT C

OUT指令之前的运算结果由OFF→ON变化时，计数器的当前值(计数值)将被+1。如果计数到，计数器的常开触点将导通，常闭触点将变为非导通。

梯形图	ST
 <p>Value: 设定值</p>	ENO:=OUT_C(EN,Coil,Value);

FBD/LD
 <p>Value: 设定值</p>

■执行条件

指令	执行条件
OUT C	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	计数器编号	—	位	ANY_BOOL*1
Coil				
Value	计数器的设定值	0~65535	无符号BIN16位*2	ANY16*2
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能使用计数器型标签。

*2 ST、FBD/LD的情况下，数据类型将变为ANY_INT。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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												
Value	—	—	○*2	○	—	—	—	—	○*3	—	—	—

*1 只能使用C。

*2 不能使用T、ST、C。

*3 只能使用10进制常数(K)。

功能

- OUT指令之前的运算结果由OFF→ON变化时，(d) (ST、FBD/LD的情况下为Coil)中指定的计数器的当前值(计数值)将+1。如果计数到(当前值≥设定值)，常开触点将导通，常闭触点将变为非导通。
- 设定值为0的情况下，其处理与上述处理相同。
- 运算结果为ON不变的状况下不进行计数。(计数输入无需脉冲化。)
- 计数到后，在执行RST指令之前计数值及触点的状态不变化。

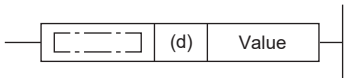
出错

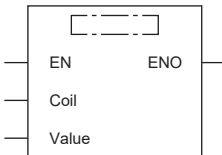
没有运算错误。

长计数器

OUT LC

OUT指令之前的运算结果由OFF→ON变化时，长计数器的当前值(计数值)将被+1。如果计数到，长计数器的常开触点将导通，常闭触点将变为非导通。

梯形图	ST
 <p>Value: 设定值</p>	ENO:=OUT_C(EN,Coil,Value);

FBD/LD
 <p>Value: 设定值 (□中放入OUT_C。)</p>

■执行条件

指令	执行条件
OUT LC	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	长计数器编号	—	位	ANY_BOOL*1
Coil				
Value	长计数器的设定值	0~4294967295	无符号BIN32位*2	ANY32*2
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能使用长计数器型标签。

*2 ST、FBD/LD的情况下，数据类型将变为ANY_INT。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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												
Value	—	—	○*2	○	—	—	—	—	—	—	—	—

*1 只能使用LC。

*2 不能使用T、ST、C。

*3 只能使用10进制常数(K)。

功能

- OUT指令之前的运算结果由OFF→ON变化时，(d) (ST、FBD/LD的情况下为Coil)中指定的长计数器的当前值(计数值)将+1。如果计数到(当前值≥设定值)，常开触点将导通，常闭触点将变为非导通。
- 设定值为0的情况下，其处理与上述处理相同。
- 运算结果为ON不变的状况下不进行计数。(计数输入无需脉冲化。)
- 计数到后，在执行RST指令之前计数值及触点的状态不变化。

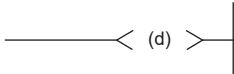
出错

没有运算错误。

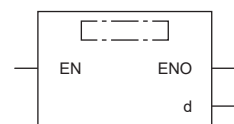
报警器

OUT F

将OUT F指令之前的运算结果输出到指定的报警器。

梯形图	ST
	ENO:=OUT(EN, d);

FBD/LD



(□中放入OUT。)

■执行条件

指令	执行条件
OUT F	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	变为ON的报警器编号	—	位	ANY_BOOL*1
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能使用报警器中分配的标签。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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 只能使用F。

功能

- 将OUT F指令之前的运算结果输出到指定的报警器。
- 通过OUT F指令将报警器(F)置为ON的情况如下所示。
 - 控制器前面的USER LED将亮灯。
 - 变为ON的报警器编号(F编号)将被存储到特殊寄存器(SD64~SD79)中。
 - SD63的内容将被+1。
- SD63的内容为16的情况下(报警器已16个ON的情况下), 即使新的报警器ON, 变为ON的报警器编号也不被存储到SD64~SD79中。
- 通过OUT F指令将报警器置为OFF的情况如下所示。
 - 线圈将变为OFF, 但USER LED的状态、SD63~SD79的内容不变化。
 - USER LED的熄灯、从SD63~SD79中将通过OUT F指令置为OFF的报警器删除的情况下, 通过RST F指令进行。

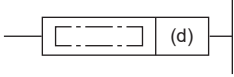
出错

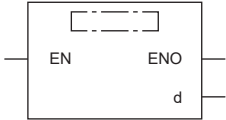
没有运算错误。

软元件的设置(报警器除外)

SET

将指定位置为ON。

梯形图	ST
	ENO:=SET(EN, d);

FBD/LD


■执行条件

指令	执行条件
SET	

■设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	设置(ON)的位软元件编号/字软元件的位指定	—	位	ANY_BOOL
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它(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 使用F的情况下，请参阅下述指令。

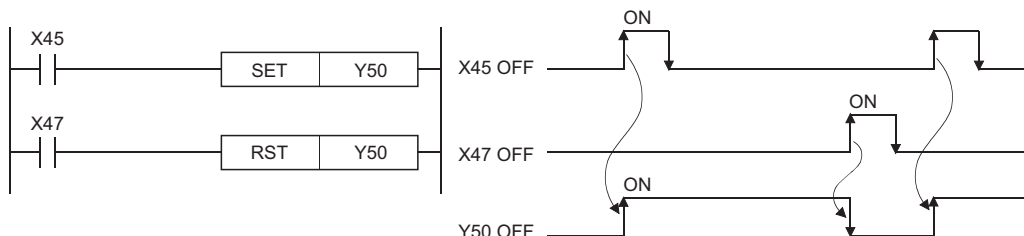
☞ 229页 SET F

功能

- 执行指令为ON时指定软元件将变为下述状态。

软元件	软元件的状态
位软元件	将线圈、触点置为ON。
字软元件的位指定	将指定位置为1。

- 对于置为ON的软元件，即使执行指令变为OFF也将保持ON不变。对于通过SET指令置为ON的软元件，可以通过RST指令置为OFF。



- 执行指令为OFF的情况下，软元件的状态不变化。

出错

没有运算错误。

要点

软元件中使用X的情况下，应使用实际输入中未使用的软元件编号。通过使用与实际输入软元件相同的编号，实际输入的数据将被通过SET指令指定的输入X所覆盖。

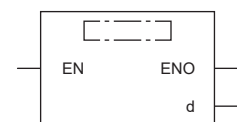
软元件的复位(报警器除外)

RST

将指定软元件置为0(OFF)。关于定时器、计数器，将当前值置为0，将触点/线圈置为OFF。

梯形图	ST
	ENO:=RST(EN, d);

FBD/LD



■执行条件

指令	执行条件
RST	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	复位的位软元件编号/字软元件的位指定或复位的字软元件编号	—	位/字/双字	ANY_ELEMENTARY
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它(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 使用F的情况下，请参阅下述指令。

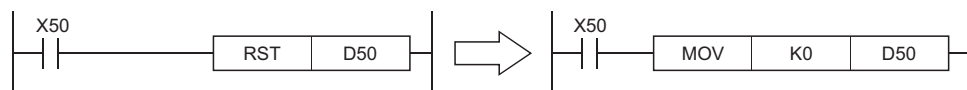
231页 RST F

功能

- 执行指令为ON时指定软元件将变为下述状态。

软元件	软元件的状态
位软元件	将线圈、触点置为OFF。
定时器、计数器	将当前值置为0，将线圈、触点置为OFF。
字软元件的位指定	将指定位置为0。
定时器、计数器以外的字软元件	将内容置为0。

- 执行指令为OFF的情况下，软元件的状态不变化。
- 对于通过RST指令指定字软元件时的功能，与下述梯形图相同。



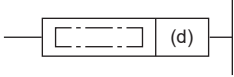
出错

没有运算错误。

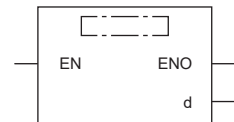
报警器的设置

SET F

将指定的报警器置为ON。

梯形图	ST
	ENO:=SET (EN, d);

FBD/LD



(□中放入SET。)

■执行条件

指令	执行条件
SET F	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	设置的报警器编号(F编号)	—	位	ANY_BOOL
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

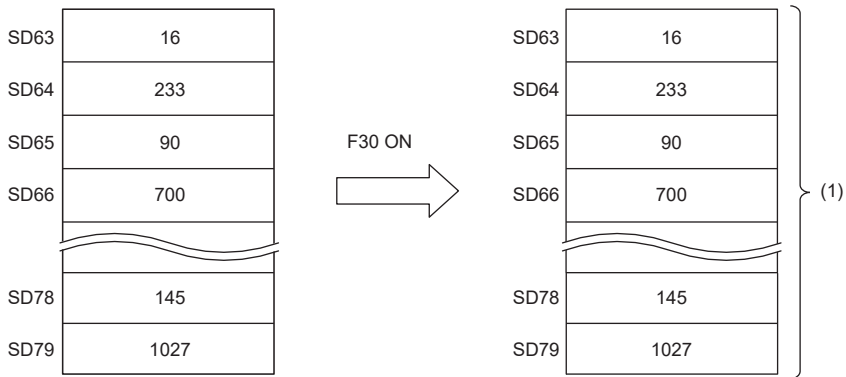
■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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 只能使用F。

功能

- 执行指令为ON时，将(d)中指定的报警器置为ON。
- 将报警器(F)置为ON的情况如下所示。
 - USER LED将亮灯。
 - 变为ON的报警器编号(F编号)将被存储到特殊寄存器(SD64~SD79)中。
 - SD63的内容将被+1。
- SD63的内容为16的情况下(报警器已16个ON的情况下)，即使新的报警器ON，变为ON的报警器编号也不被存储到SD64~SD79中。



(1) 不变化。

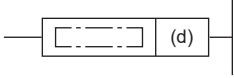
出错

没有运算错误。

报警器的复位

RST F

将指定的报警器置为OFF。

梯形图	ST
	ENO:=RST(EN, d);

FBD/LD
 <p>(□中放入RST。)</p>

■执行条件

指令	执行条件
RST F	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	复位的报警器编号(F编号)	—	位*1	ANY_BOOL*1
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 ST、FBD/LD的情况下，数据类型将变为ANY_ELEMENTARY。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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	—	—	—	—	—	—	—	—	—	—

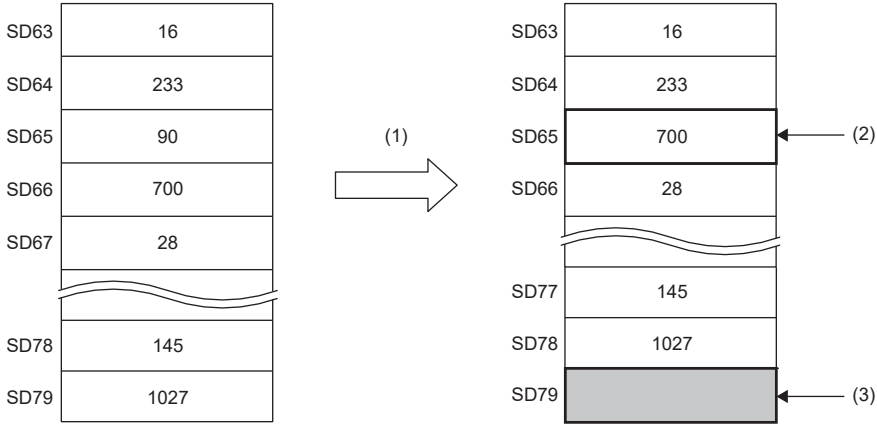
*1 只能使用F。

功能

- 执行指令变为ON时，将(d)中指定的报警器置为OFF。
- 变为OFF的报警器编号(F编号)将从特殊寄存器(SD64~SD79)中被删除，SD63的内容将被-1。
- SD63的内容为16时，通过RST指令报警器编号将从SD64~SD79中被删除。此外，SD64~SD79中未登录的编号的报警器变为ON时，该编号将被新建登录。如果将SD64~SD79的报警器编号全部置为OFF，控制器前面的USER LED将熄灯。

例

SD63为16，有未登录的报警器编号的情况下



出错

没有运算错误。

PLS

在PLS指令的OFF→ON时将指定软元件置为1个扫描ON。

梯形图	ST
	ENO:=PLS (EN, d);

FBD/LD

■执行条件

指令	执行条件
PLS	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	脉冲化的软元件编号	—	位	ANY_BOOL
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

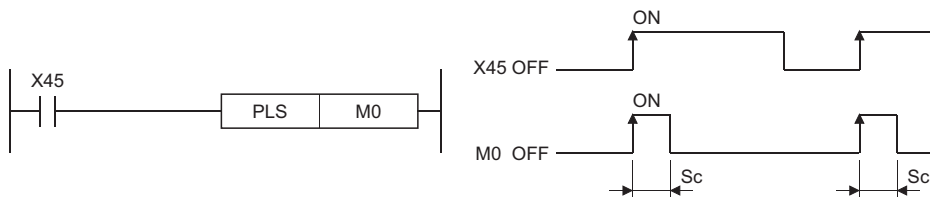
■可用软元件

操作数	位		字		双字		间接指定	常数			其它(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)	○	○	○	○	—	—	—	○	—	—	—	○

功能

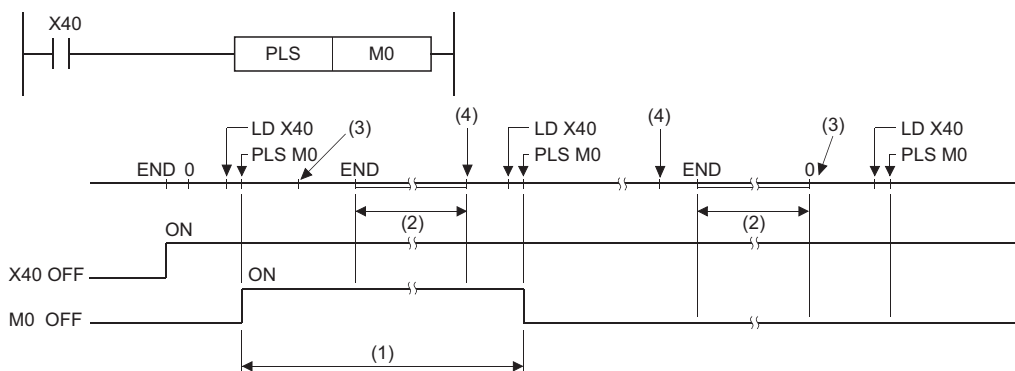
- 执行指令OFF→ON时将指定软元件置为ON，执行指令OFF→ON以外(ON→ON、ON→OFF、OFF→OFF)时不置为OFF。1个扫描中(d)中指定的软元件的PLS指令为1个的情况下，指定软元件将1个扫描ON。关于对同一软元件在1个扫描中多次执行了PLS指令情况下的动作，请参阅下述内容。

☞ 51页 使用同一软元件的OUT指令、SET/RST指令、PLS/PLF指令时的动作



Sc: 1个扫描

- 执行PLS指令后置为RUN→STOP，即使再次置为RUN也不执行PLS指令。



- (1) PLS M0的1个扫描
- (2) 控制器的运算停止时间
- (3) 将控制器的RUN/STOP/RESET开关置为RUN→STOP。
- (4) 将控制器的RUN/STOP/RESET开关置为STOP→RUN。

- 在执行指令中指定锁存继电器(L)，锁存继电器为ON的状态下将电源置为OFF→ON时，执行指令将在第1个扫描变为OFF→ON，执行PLS指令且指定软元件将变为ON。对于电源ON后第1个扫描中变为ON的软元件，通过下一个PLS指令(上升沿输出)将变为OFF。
- PLS指令在执行指令后执行下一个指令时进行OFF处理。

要点

- 通过CJ指令使PLS指令跳转，或执行的子程序未通过CALL(P)指令调用的情况下，(d)中指定的软元件有可能在1个或其以上的扫描中ON，应加以注意。

出错

没有运算错误。

下降沿输出

PLF

PLF指令ON→OFF时将指定软元件置为1个扫描ON。

梯形图	ST
	ENO:=PLF(EN, d);

FBD/LD

■执行条件

指令	执行条件
PLF	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	脉冲化的软元件编号	—	位	ANY_BOOL
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

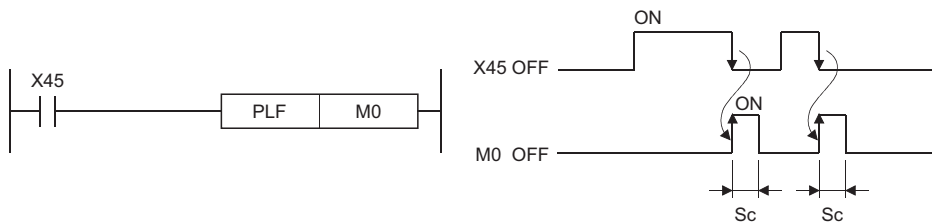
■可用软元件

操作数	位		字		双字		间接指定	常数			其它(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)	○	○	○	○	—	—	—	○	—	—	—	○

功能

- 执行指令ON→OFF时将指定软元件置为ON，执行指令为ON→OFF以外(OFF→OFF、OFF→ON、ON→ON)时将变为OFF。1个扫描中(d)中指定的软元件的PLF指令为1个的情况下，指定软元件将1个扫描ON。关于在1个扫描中对同一软元件执行了多次PLF指令时的动作，请参阅下述内容。

☞ 51页 使用同一软元件的OUT指令、SET/RST指令、PLS/PLF指令时的动作



Sc: 1个扫描

- 执行PLF指令后置为RUN→STOP，即使再次置为RUN也不执行PLF指令。
- PLF指令在执行指令后执行下一个指令时进行OFF处理。

要点

通过CJ指令使PLF指令跳转，或执行的子程序未通过CALL(P)指令调用的情况下，(d)中指定的软元件有可能在1个或其以上的扫描ON，因此应加以注意。

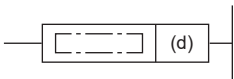
出错

没有运算错误。

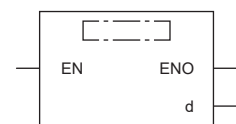
位软元件输出取反

FF


对指定的软元件状态进行取反。

梯形图	ST
	ENO:=FF (EN, d) ;

FBD/LD



■执行条件

指令	执行条件
FF	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	取反的软元件编号	—	位	ANY_BOOL
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它(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)	○	○	○	○	—	—	—	○	—	—	—	○	

功能

- 执行指令OFF→ON时，对(d)中指定的软元件状态进行取反。

软元件	软元件的状态	
	FF指令执行前	FF指令执行后
位软元件	OFF	ON
	ON	OFF
字软元件的位指定	0	1
	1	0

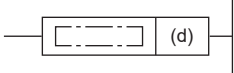
出错

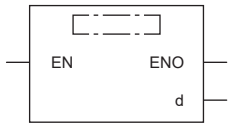
没有运算错误。

直接输出的脉冲化


DELTA (P)

对指定的直接访问输出 (DY) 进行脉冲输出。

梯形图	ST
	ENO:=DELTA (EN, d) ; ENO:=DELTAP (EN, d) ;

FBD/LD


■执行条件

指令	执行条件
DELTA	
DELTAP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(d)	脉冲化的软元件编号	—	位	ANY_BOOL*1
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能使用软元件 (DY) 中分配的标签。

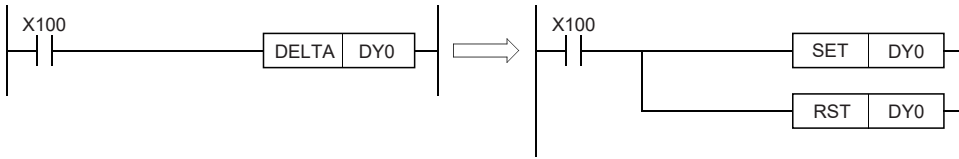
■可用软元件

操作数	位		字		双字		间接指定	常数			其它 (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)	—	—	—	—	—	—	—	—	—	—	—	○

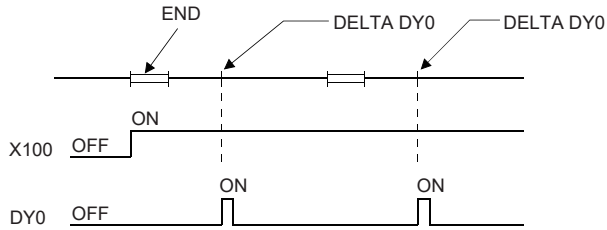
功能

- 对(d)中指定的直接访问输出(DY)进行脉冲输出。(d)指定了DY0的情况下,其动作将与使用了SET指令、RST指令的梯形图相同。

将使用了DELTA指令的梯形图替换为不使用SET/RST指令的梯形图的情况如下所示。



动作如下所示。



注意事项

对于DELTA(P)指令,用于至智能功能模块的上升沿执行指令。
 在使用了输出模块的实际输出指令中不能使用。

出错

没有运算错误。

要点

对于DELTA(P)指令,用于高速计数器模块的预设值的设置等。

11.4 移位指令

位软元件移位

SFT(P)

将小1号的软元件的ON/OFF状态移位到指定的软元件中，将小1号的软元件置为OFF。

梯形图	ST
	ENO:=SFT(EN, d); ENO:=SFTP(EN, d);

FBD/LD

■执行条件

指令	执行条件
SFT	
SFTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	移位的软元件	—	位	ANY_BOOL
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它(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)	○	○	○	○	—	—	—	○	—	—	—	○

功能

■位软元件的情况下

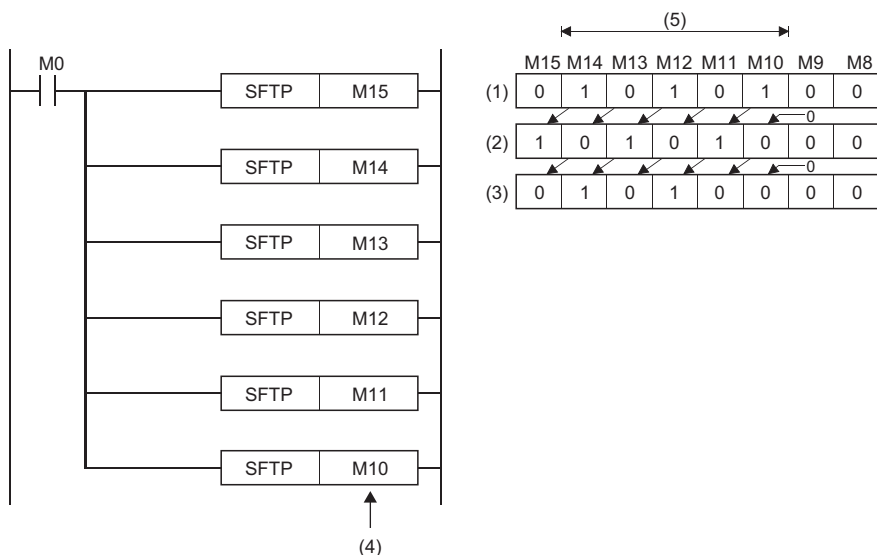
将(d)中指定的软元件的前一个软元件的ON/OFF状态移位到(d)中指定的软元件中。(d)中指定的软元件的前一个软元件将变为OFF。

要点

连续使用SFT(P)指令的情况下，应创建从软元件编号的大编号开始的程序。

例

通过SFTP指令将M10至M11的软元件进行了移位时变为如下状态。



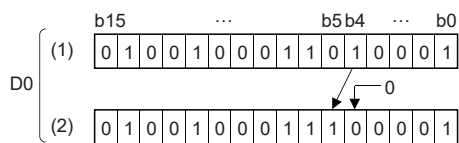
- (1) 初始状态
- (2) 第1次的移位输入后
- (3) 第2次的移位输入后
- (5) 移位范围

■字软元件的位指定的情况下

将(d)中指定位的前一个位的1/0状态移位至(d)中指定的位中。(d)中指定的位的前一个位将变为0。

例

通过SFT(P)指令指定了D0.5(D0的位5(b5))的情况下，执行SFT(P)指令时将D0的b4的1/0移位到b5中，将b4置为0。



- (1) 执行移位前
- (2) 执行移位后

出错

没有运算错误。

11.5 主控制指令

主控制的设置、复位

MC、MCR

- MC：开始主控制。
- MCR：结束主控制。

梯形图	ST
<p>(1) 主控制梯形图</p>	<pre>ENO:=MC(EN, N, d); ENO:=MCR(EN, N);</pre>

FBD/LD	MCR

■执行条件

指令	执行条件
MC	常时执行
MCR	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(N)	嵌套	N0~N14	软元件名	ANY16_S* ¹
(d)	置为ON的软元件编号	—	位	ANY_BOOL
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能使用软元件(N)中分配的标签或分配了常数的标签。

■可用软元件

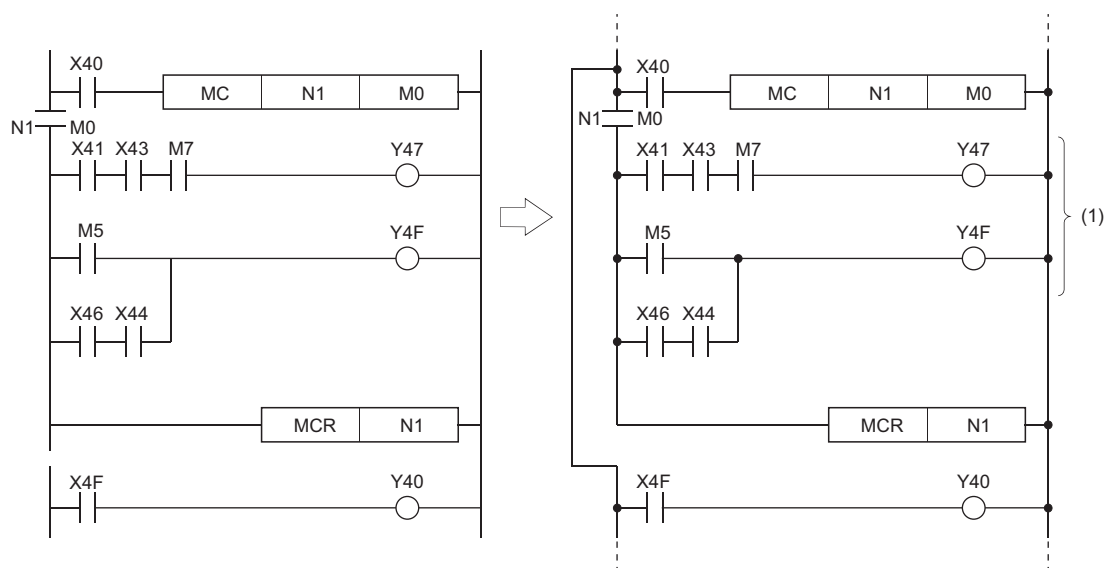
操作数	位		字		双字			间接指定	常数			其它		
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K、H	E	\$	N	DY	
(N)	—	—	—	—	—	—	—	—	—	—	—	—	—	—
(d)	○	○	○	—	—	—	—	○	—	—	—	—	—	○

功能

主控制指令是用于通过梯形图的公共母线的开闭，创建高效的梯形图切换程序的指令。

使用了主控制的梯形图如下所示。

(左：工程工具的显示，右：实际的动作梯形图)



(1) 仅在X40为ON时执行

MC

- 通过主控制的开始，MC指令的执行指令为ON的情况下，从MC指令开始到MCR指令之间的运算结果即为指令(梯形图)的执行结果。MC的执行指令为OFF的情况下，从MC指令开始到MCR指令之间的运算结果如下所示。

软元件	软元件的状态
高速定时器 低速定时器	计数值变为0，线圈、触点均变为OFF。
高速累计定时器 低速累计定时器 计数器	线圈变为OFF，但计数值、触点均保持当前的状态。
OUT指令中的软元件	强制置为OFF。
SET指令、RST指令中的软元件 SFT(P)指令中的软元件 基本指令、应用指令中的软元件	保持当前的状态。

- MC指令为OFF的情况下从MC指令开始到MCR指令之间的指令也将执行，因此扫描时间不会变短。

要点

- 使用了主控制的梯形图中，存在有不需要触点指令的指令(FOR~NEXT指令等)的情况下控制器与MC指令的执行指令无关，执行该指令。
- 为了创建易于理解的程序，成对的MC指令及MCR指令应在一个程序块内使用。
- 对于MC指令，通过改变(d)的软元件，可以多次使用同一嵌套(N)编号。
- MC指令为ON时，(d)中指定的软元件的线圈将变为ON。此外，OUT指令等中使用同一软元件时将变为双线圈，因此(d)中指定的软元件请勿在其它指令中使用。

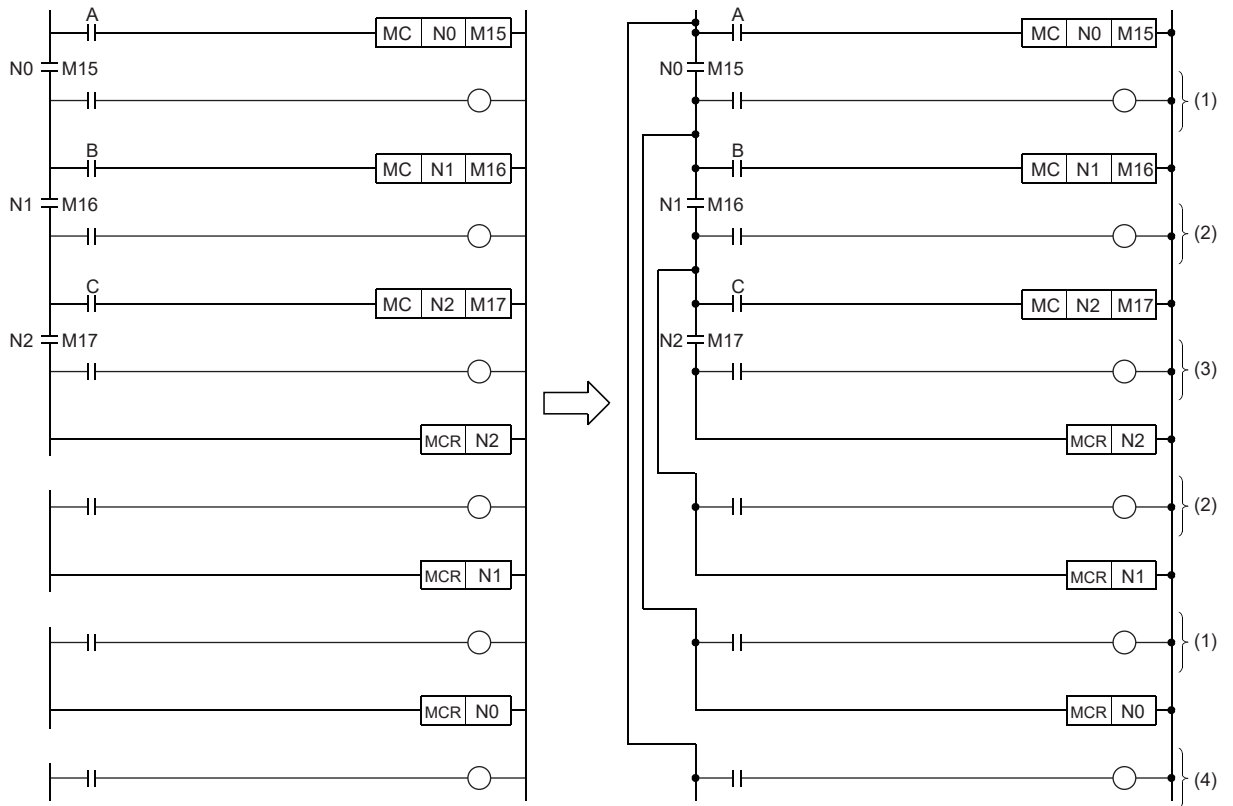
MCR

- 通过主控制的解除指令，表示主控制范围的结束。
- MCR指令的前面请勿附加触点指令。
- 使用时应设置同一嵌套编号的MC指令及MCR指令。但是，MCR指令集中于1个位置的嵌套结构时，通过一个最小号的嵌套(N)编号，可以结束所有的主控制。(参阅注意事项)

出错

没有运算错误。

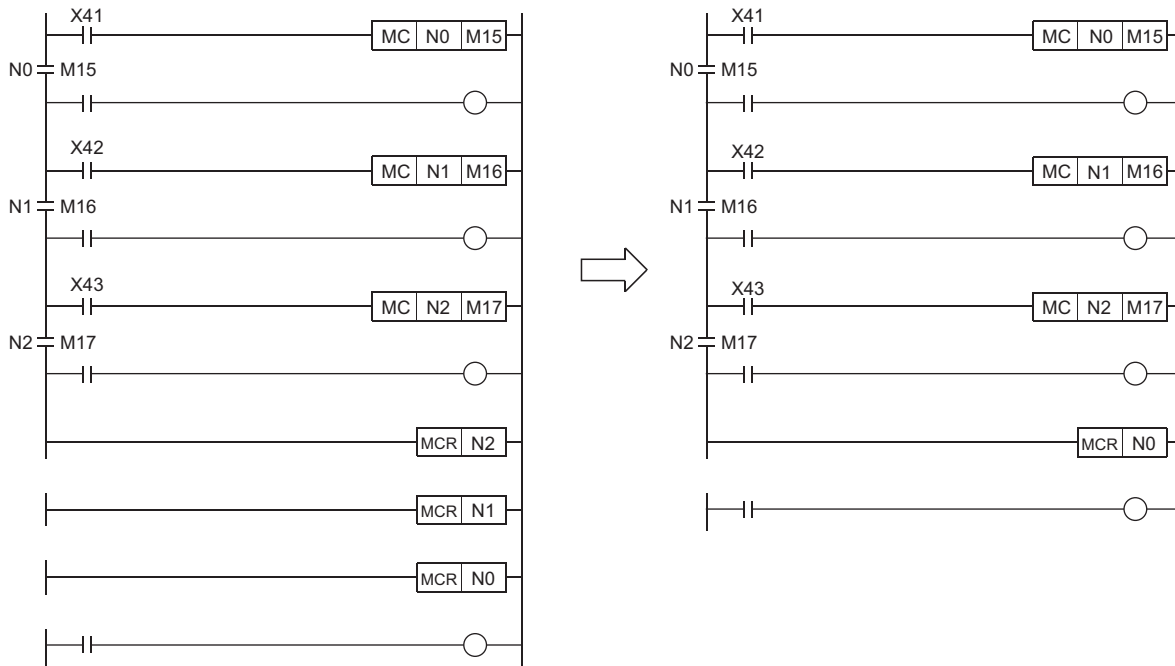
主控制指令可通过嵌套结构使用。各个主控制区间通过嵌套(N)进行区分。嵌套可使用N0~N14。
 通过使用嵌套结构，可以创建对程序的执行条件依次进行限制的梯形图。
 使用了嵌套结构的梯形图如下所示。
 (左：工程工具的显示，右：实际的动作梯形图)



- (1) A为ON时执行
- (2) A、B为ON时执行
- (3) A、B、C为ON时执行
- (4) 与A、B、C无关

注意事项

- 嵌套最多可以有15个(N0~N14)。进行嵌套的情况下，MC指令从嵌套(N)编号小的开始使用，而MCR指令则是从大编号开始使用。
- MCR指令集中于1个位置的嵌套结构时，通过一个最小号的嵌套(N)编号，可以结束所有的主控制。

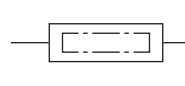


11.6 结束指令

主程序结束

FEND

用于在程序文件内将主程序与子程序、中断程序分开。

梯形图	ST
	不对应。

FBD/LD
不对应。

■执行条件

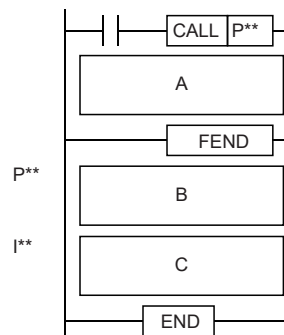
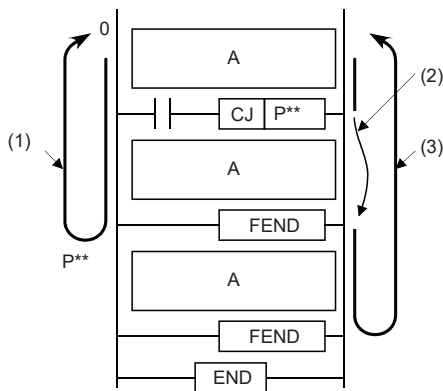
指令	执行条件
FEND	常时执行

功能

- 对于FEND指令，用于通过CJ指令等将顺控程序的运算分支的情况下，以及将主程序与子程序、通过中断指针(I)的中断程序分开时使用。
- 执行FEND指令时，控制器将结束执行的程序。
- FEND指令以后的顺控程序也可通过工程工具进行梯形图显示。

使用CJ指令的情况下

有子程序、中断程序的情况下



- A: 主程序
- B: 子程序
- C: 中断程序

- (1) 不执行CJ指令时的运算
- (2) 通过CJ指令进行跳转
- (3) 执行了CJ指令时的运算

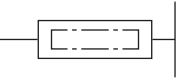
出错

错误代码 (SD0)	内容
3341H	执行CALL (P) 指令、FCALL (P) 指令、ECALL (P) 指令、EFCALL (P) 指令后，执行RET指令之前执行了FEND指令时。
3351H	在通过中断指针 (I) 的中断程序中在执行IRET指令之前执行了FEND指令时。

顺控程序结束

END

表示程序的最终。

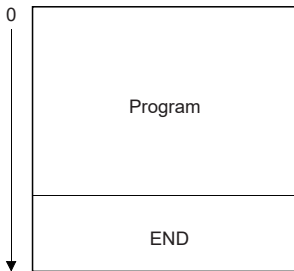
梯形图	ST
	不对应。
FBD/LD	
不对应。	

■ 执行条件

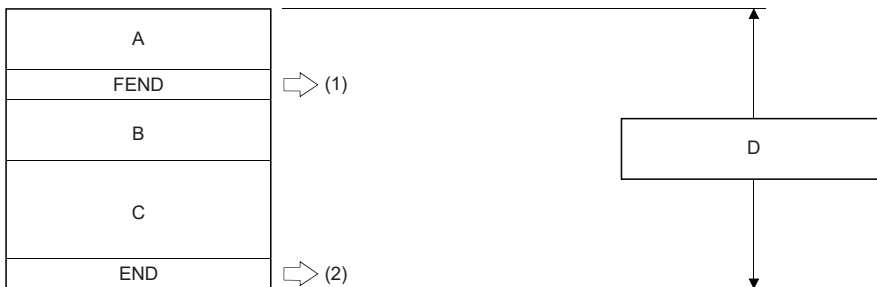
指令	执行条件
END	常时执行

功能

- 表示包含了主程序、子程序、中断程序的程序的最终。
- 执行END指令时，控制器将结束正在执行的程序。



- 程序的途中需要END处理的情况下，应使用FEND指令。
- 通过工程工具的梯形图编辑模式进行编程的情况下，END指令将被自动输入，不能进行编辑。
- 存在有主程序、子程序、中断程序的情况下的结束指令的使用用途如下所示。



- A: 主程序
 B: 子程序
 C: 中断程序
 D: 主顺控程序区域
 (1) 需要FEND指令。
 (2) 需要END指令。

要点

对于将程序分开为多个程序块情况下的END指令，表示程序块的结束。
 对于作为END处理执行的END指令，将变为程序设置中登录的程序的最后执行的程序。

出错

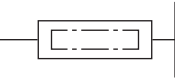
错误代码 (SD0)	内容
3341H	执行CALL(P)指令、FCALL(P)指令、ECALL(P)指令、EFCALL(P)指令后, 执行RET指令之前执行了END指令时。
3351H	通过中断指针(I)的中断程序中执行IRET指令之前执行了END指令时。

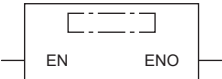
11.7 停止指令

顺控程序停止


STOP

停止控制器的运算。(与将开关置为STOP侧的情况相同。)

梯形图	ST
	ENO:=STOP(EN);

FBD/LD


■执行条件

指令	执行条件
STOP	

功能

- 执行指令为ON时，复位输出(Y)后，停止控制器的运算。(与将开关置为STOP侧的情况相同。)
- 执行STOP指令后，重启控制器运算时，将开关恢复为RUN→STOP后，再次置于RUN的位置。

出错

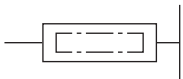
错误代码(SD0)	内容
3341H	执行CALL(P)指令、FCALL(P)指令、ECALL(P)指令、EFCALL(P)指令、XCALL指令后，执行RET指令前执行了STOP指令时。
3351H	通过中断指针(I)的中断程序中执行IRET指令前执行了STOP指令时。
3353H	恒定周期执行类型程序中执行了STOP指令时。
3354H	事件执行类型程序中执行了STOP指令时。

11.8 无处理指令

无处理 (NOPLF)

NOPLF

是无处理的指令，对至当时为止的运算不会产生任何影响。

梯形图	ST
	不对应。

FBD/LD

不对应。

■执行条件

指令	执行条件
NOPLF	常时执行

功能

是无处理的指令，对至当时为止的运算不会产生任何影响。

出错

没有运算错误。

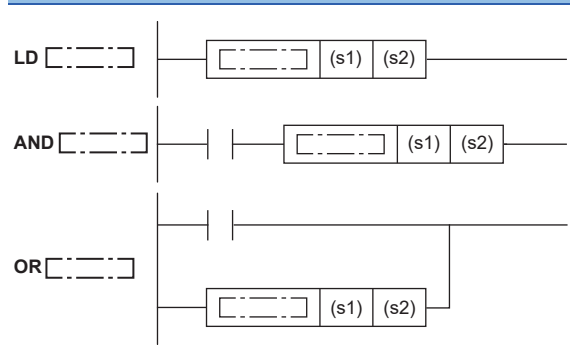
12 基本指令

12.1 比较运算指令

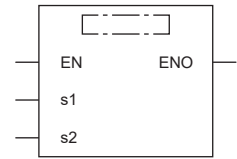
BIN16位数据比较

LD□(U)、AND□(U)、OR□(U)

将指定的2个BIN16位数据通过常开触点处理进行比较。

梯形图	ST	
	<pre>ENO:=LD_□(EN, s1, s2); ENO:=AND_□(EN, s1, s2); ENO:=OR_□(EN, s1, s2);</pre>	<pre>ENO:=LD_□_U(EN, s1, s2); ENO:=AND_□_U(EN, s1, s2); ENO:=OR_□_U(EN, s1, s2);</pre>
(□中放入=(_U)、<(_U)、>(_U)、<=(_U)、<(_U)、>=(_U)。)		
(□中为EQ、NE、GT、LE、LT、GE。)*1		

FBD/LD



(□中放入LD_、AND_、OR_与EQ(_U)、NE(_U)、GT(_U)、LE(_U)、LT(_U)、GE(_U)的组合。)*1

*1 EQ为=、NE为<>、GT为>、LE为<=、LT为<、GE为>=。

■执行条件

指令	执行条件
LD□(U)、AND□(U)、OR□(U)	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	LD□、AND□、OR□	比较数据或存储了比较数据的软元件	-32768~32767	带符号BIN16位	ANY16_S
	LD□_U、AND□_U、OR□_U		0~65535	无符号BIN16位	ANY16_U
(s2)	LD□、AND□、OR□	比较数据或存储了比较数据的软元件	-32768~32767	带符号BIN16位	ANY16_S
	LD□_U、AND□_U、OR□_U		0~65535	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 将(s1)中指定的软元件的BIN16位数据与(s2)中指定的软元件的BIN16位数据通过常开触点处理进行比较运算。
- 各指令的比较运算结果如下所示。

指令符号(梯形图、FBD/LD)	条件	比较运算结果
=(_U)、EQ(_U)	(s1)=(s2)	导通状态(ENO为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)	非导通状态(ENO为OFF)
=(_U)、EQ(_U)	(s1)≠(s2)	
<>(_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)	

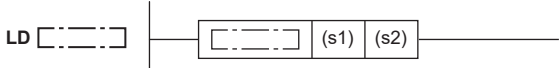
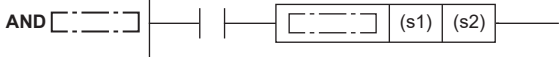
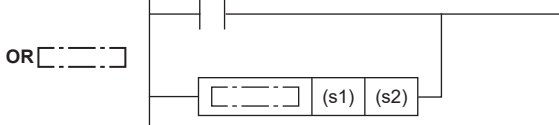
- LD□、AND□、OR□指令中，(s1)、(s2)中指定了16进制数常数的情况下，如果指定最高位(b15)为1的数值(8~F)，将被视为BIN值的负数，进行比较运算。
- 通过FBD/LD使用了LD□指令的情况下，应将EN指定为左母线，或始终ON的变量部件/常数部件。
- 通过FBD/LD使用了OR□指令的情况下，将EN指定为TRUE时ENO将变为ON。EN不成为执行条件。

出错

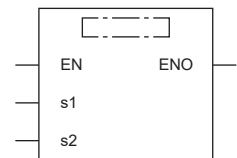
没有运算错误。

LDD□(U)、ANDD□(U)、ORD□(U)

将指定的2个BIN32位数据通过常开触点处理进行比较。

梯形图	ST	
  	<pre>ENO:=LDD_□(EN, s1, s2);</pre> <pre>ENO:=ANDD_□(EN, s1, s2);</pre> <pre>ENO:=ORD_□(EN, s1, s2);</pre>	<pre>ENO:=LDD_□_U(EN, s1, s2);</pre> <pre>ENO:=ANDD_□_U(EN, s1, s2);</pre> <pre>ENO:=ORD_□_U(EN, s1, s2);</pre>
<p>(□中放入D=(U)、D<>(U)、D>(U)、D<=(U)、D<(U)、D>=(U)。)</p>	<p>(□中为EQ、NE、GT、LE、LT、GE。)*1</p>	

FBD/LD



(□中放入LDD_、ANDD_、ORD_与EQ(U)、NE(U)、GT(U)、LE(U)、LT(U)、GE(U)的组合。)*1

*1 EQ为=、NE为<>、GT为>、LE为<=、LT为<、GE为>=。

■执行条件

指令	执行条件
LDD□(U)、ANDD□(U)、ORD□(U)	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	LDD□、ANDD□、ORD□	比较数据或存储了比较数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	LDD□_U、ANDD□_U、ORD□_U		0~4294967295	无符号BIN32位	ANY32_U
(s2)	LDD□、ANDD□、ORD□	比较数据或存储了比较数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	LDD□_U、ANDD□_U、ORD□_U		0~4294967295	无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	○	—	—	—	—

功能

- 将(s1)中指定的软元件的BIN32位数据与(s2)中指定的软元件的BIN32位数据通过常开触点处理进行比较运算。
- 各指令的比较运算结果如下所示。

指令符号(梯形图、FBD/LD)	条件	比较运算结果
D=(<u> </u>)、EQ(<u> </u>)	(s1)=(s2)	导通状态(ENO为ON)
D>(<u> </u>)、NE(<u> </u>)	(s1)≠(s2)	
D>(<u> </u>)、GT(<u> </u>)	(s1)>(s2)	
D<(<u> </u>)、LE(<u> </u>)	(s1)≤(s2)	
D<(<u> </u>)、LT(<u> </u>)	(s1)<(s2)	
D>= <u> </u>)、GE(<u> </u>)	(s1)≥(s2)	
D=(<u> </u>)、EQ(<u> </u>)	(s1)≠(s2)	非导通状态(ENO为OFF)
D>(<u> </u>)、NE(<u> </u>)	(s1)≤(s2)	
D>(<u> </u>)、GT(<u> </u>)	(s1)>(s2)	
D<(<u> </u>)、LE(<u> </u>)	(s1)<(s2)	
D<(<u> </u>)、LT(<u> </u>)	(s1)≥(s2)	
D>= <u> </u>)、GE(<u> </u>)	(s1)<(s2)	

- 在LDD□、ANDD□、ORD□指令中，(s1)、(s2)中指定了16进制数的常数的情况下，如果指定最高位(b31)为1的数值(8~F)，将被视为BIN值的负数，进行比较运算。
- 比较中使用的数据应通过32位数据处理指令(DMOV(P)指令等)进行指定。通过16位数据处理指令(MOV(P)指令等)指定的情况下，将无法正常进行大小比较。
- 通过FBD/LD使用了LDD□指令的情况下，应将EN指定为左母线，或始终ON的变量部件/常数部件。
- 通过FBD/LD使用了ORD□指令的情况下，将EN指定为TRUE时ENO将变为ON。EN不成为执行条件。

出错

没有运算错误。

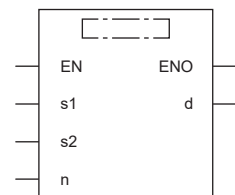
BIN16位块数据比较

BKCMP□(P)(_U)

对指定的2个BIN16位块数据进行比较。

梯形图	ST
<p>(□中放入BKCMP=(P)(_U)、BKCMP<>(P)(_U)、BKCMP>(P)(_U)、BKCMP<=(P)(_U)、BKCMP<(P)(_U)、BKCMP>=(P)(_U)。) </p>	ENO:=BKCMP_□(EN, s1, s2, n, d); ENO:=BKCMP_□P(EN, s1, s2, n, d); ENO:=BKCMP_□_U(EN, s1, s2, n, d); ENO:=BKCMP_□_UP(EN, s1, s2, n, d); (□中为EQ、NE、GT、LE、LT、GE。)*1

FBD/LD



(□中放入BKCMP_与EQ(P)(_U)、NE(P)(_U)、GT(P)(_U)、LE(P)(_U)、LT(P)(_U)、GE(P)(_U)的组合。)*1

*1 EQ为=、NE为<>、GT为>、LE为<=、LT为<、GE为>=。

■执行条件

指令	执行条件
BKCMP□(_U)	
BKCMP□P(_U)	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	BKCMP□(P)	比较数据或存储了比较数据的起始软元件	-32768~32767	带符号BIN16位	ANY16_S*1
	BKCMP□(P)_U		0~65535	无符号BIN16位	ANY16_U*1
(s2)	BKCMP□(P)	存储了比较数据的起始软元件	—	带符号BIN16位	ANY16_S*1
	BKCMP□(P)_U			无符号BIN16位	ANY16_U*1
(d)	存储比较运算结果的起始软元件	—	位	ANY_BOOL*1	
(n)	比较的数据数	0~65535	无符号BIN16位	ANY16	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

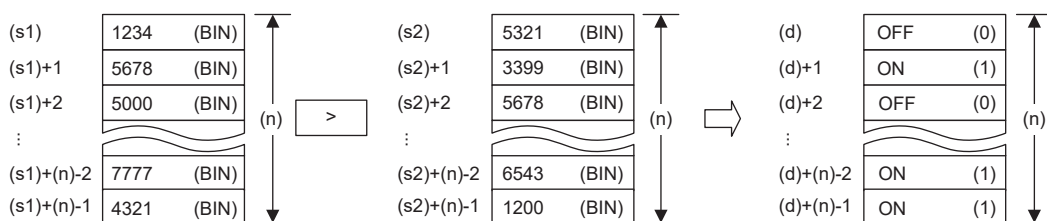
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

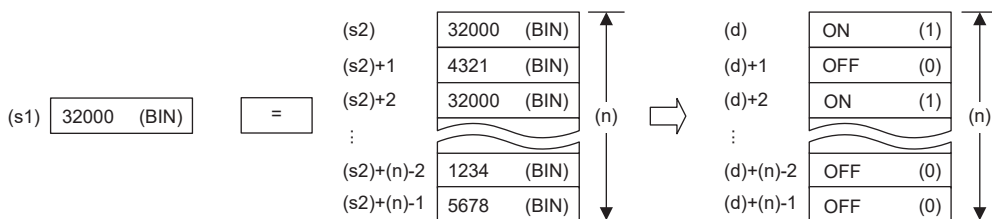
操作数	位		字		双字			间接指定	常数				其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	—	—	○	—	—	—	—	○	○	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	○	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

- 将(s1)中指定的软元件开始的(n)点的BIN16位数据与(s2)中指定的软元件开始的(n)点的BIN16位数据进行比较, 将运算结果存储到(d)中指定的软元件及其以后。
- 比较条件成立的情况下, (d)的相应软元件将变为ON, 比较条件不成立的情况下, (d)的相应软元件将变为OFF。



- 比较运算是以16位单位进行。
- (s1)可以指定常数。



- 各指令的比较运算结果如下所示。

指令符号(梯形图、FBD/LD)	条件	比较运算结果
BKCMPEQ(P) (U), BKCMPEQ(P) (U)	(s1) = (s2)	ON (1)
BKCMPLT(P) (U), BKCMPLT(P) (U)	(s1) ≠ (s2)	
BKCMPGT(P) (U), BKCMPGT(P) (U)	(s1) > (s2)	
BKCMPLT(P) (U), BKCMPLT(P) (U)	(s1) ≤ (s2)	
BKCMPLT(P) (U), BKCMPLT(P) (U)	(s1) < (s2)	
BKCMPLT(P) (U), BKCMPLT(P) (U)	(s1) ≥ (s2)	
BKCMPEQ(P) (U), BKCMPEQ(P) (U)	(s1) ≠ (s2)	OFF (0)
BKCMPLT(P) (U), BKCMPLT(P) (U)	(s1) = (s2)	
BKCMPLT(P) (U), BKCMPLT(P) (U)	(s1) ≤ (s2)	
BKCMPLT(P) (U), BKCMPLT(P) (U)	(s1) > (s2)	
BKCMPLT(P) (U), BKCMPLT(P) (U)	(s1) ≥ (s2)	
BKCMPLT(P) (U), BKCMPLT(P) (U)	(s1) < (s2)	

- (d)开始的(n)点中存储的比较运算结果全部为ON(1)时, SM704(块比较信号)将变为ON。

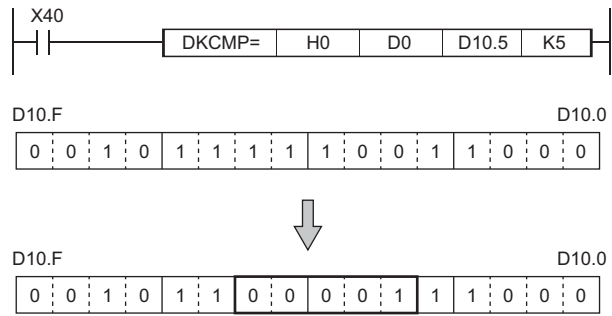
出错

错误代码 (SD0)	内容
2821H	(s1)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围重复时。
	(s2)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围重复时。

要点 

进行了字软元件的位指定的情况下，存储下述运算结果的位指定软元件以外不变化。

- 比较的数据数 (n)：K5
- 存储比较运算结果的软元件的起始编号 (d)：D10.5



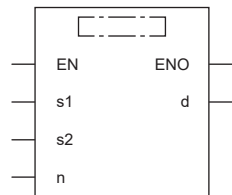
BIN32位块数据比较

DBKCM□(P)(_U)

对指定的2个BIN32位块数据进行比较。

梯形图	ST	
<p>(□中放入DBKCM□(P)(_U)、DBKCM□<(P)(_U)、DBKCM□>(P)(_U)、DBKCM□<=(P)(_U)、DBKCM□<(P)(_U)、DBKCM□>=(P)(_U)。) </p>	ENO:=DBKCM□(EN, s1, s2, n, d); ENO:=DBKCM□P(EN, s1, s2, n, d);	ENO:=DBKCM□_U(EN, s1, s2, n, d); ENO:=DBKCM□_UP(EN, s1, s2, n, d);
	(□中为EQ、NE、GT、LE、LT、GE。)*1	

FBD/LD



(□中放入DBKCM□与EQ(P)_U、NE(P)_U、GT(P)_U、LE(P)_U、LT(P)_U、GE(P)_U的组合。)*1

*1 EQ为=、NE为<>、GT为>、LE为<=、LT为<、GE为>=。

■执行条件

指令	执行条件
DBKCM□(P)_U	
DBKCM□P(P)_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	DBKCM□(P)	比较数据或存储了比较数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S*1
	DBKCM□(P)_U		0~4294967295	无符号BIN32位	ANY32_U*1
(s2)	DBKCM□(P)	存储了比较数据的起始软元件	—	带符号BIN32位	ANY32_S*1
	DBKCM□(P)_U			无符号BIN32位	ANY32_U*1
(d)	存储比较运算结果的起始软元件	—	位	ANY_BOOL*1	
(n)	比较的数据数	0~65535	无符号BIN16位	ANY16	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

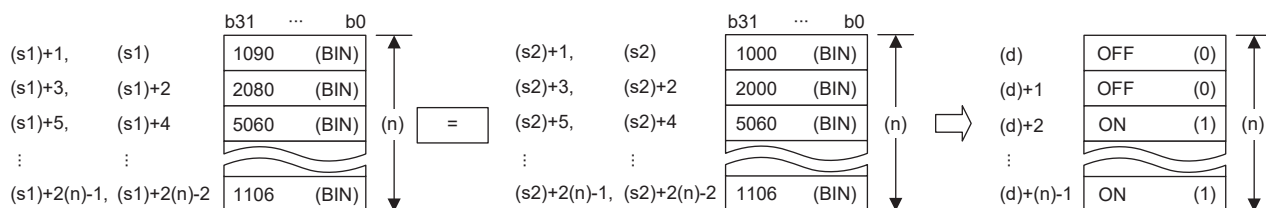
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

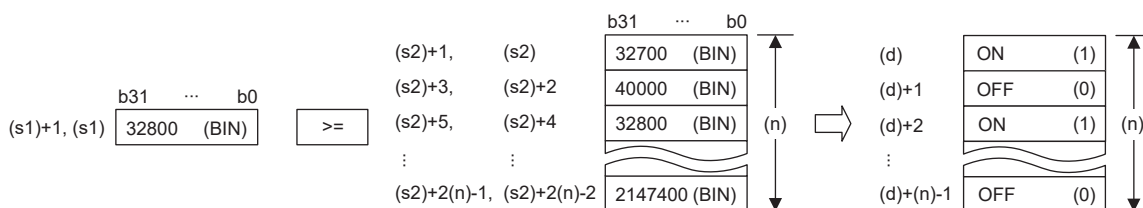
操作数	位		字		双字			间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	—	—	○	—	—	—	—	○	○	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	○	—	○	—	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 将(s1)中指定的软元件开始的(n)点的BIN32位数据与(s2)中指定的软元件开始的(n)点的BIN32位数据进行比较, 将运算结果存储到(d)中指定的软元件及其以后。
- 比较条件成立的情况下, (d)的相应软元件将变为ON, 比较条件不成立的情况下, (d)的相应软元件将变为OFF。



- 比较运算以32位单位进行。
- (s1)可以指定常数。



- (d)是在(s1)开始的(n)点的软元件范围及(s2)开始的(n)点的软元件范围以外进行指定。
- 各指令的比较运算结果如下所示。

指令符号(梯形图、FBD/LD)	条件	比较运算结果
DBKCMPEQ(P) (U)、DBKCMPEQ(P) (U)	(s1) = (s2)	ON (1)
DBKCMPEQ<>(P) (U)、DBKCMPEQ<>(P) (U)	(s1) ≠ (s2)	
DBKCMPEQ>(P) (U)、DBKCMPEQ>(P) (U)	(s1) > (s2)	
DBKCMPEQ<=(P) (U)、DBKCMPEQ<=(P) (U)	(s1) ≤ (s2)	
DBKCMPEQ<(P) (U)、DBKCMPEQ<(P) (U)	(s1) < (s2)	
DBKCMPEQ>=(P) (U)、DBKCMPEQ>=(P) (U)	(s1) ≥ (s2)	
DBKCMPEQ(P) (U)、DBKCMPEQ(P) (U)	(s1) ≠ (s2)	OFF (0)
DBKCMPEQ<>(P) (U)、DBKCMPEQ<>(P) (U)	(s1) = (s2)	
DBKCMPEQ>(P) (U)、DBKCMPEQ>(P) (U)	(s1) ≤ (s2)	
DBKCMPEQ<=(P) (U)、DBKCMPEQ<=(P) (U)	(s1) > (s2)	
DBKCMPEQ<(P) (U)、DBKCMPEQ<(P) (U)	(s1) ≥ (s2)	
DBKCMPEQ>=(P) (U)、DBKCMPEQ>=(P) (U)	(s1) < (s2)	

- (d)开始的(n)点中存储的比较运算结果全部为ON(1)时, SM704(块比较信号)将变为ON。
- (n)中指定的值为0的情况下, 将变为无处理。

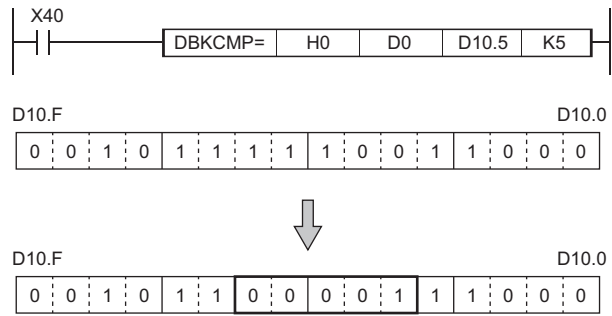
出错

错误代码 (SD0)	内容
2821H	(s1)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围重复时。
	(s2)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围重复时。

要点

进行了字软元件的位指定的情况下，存储下述运算结果的位指定软元件以外不变化。

- 比较的数据数 (n)：K5
- 存储比较运算结果的软元件的起始编号 (d)：D10.5



12.2 算术运算指令

BIN16位加法运算

+ (P) (_U) [操作数为2个的情况下]

对指定的2个BIN16位数据进行加法运算。

梯形图	ST
	不对应。 (☞ 263页 + (P) (_U) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 263页 + (P) (_U) [操作数为3个的情况下])

■ 执行条件

指令	执行条件
+ +_U	
+P +P_U	

设置数据

■ 内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s)	+ (P)	加法运算数据或存储了加法运算数据的软元件	-32768~32767	带符号BIN16位	ANY16_S
	+ (P)_U		0~65535	无符号BIN16位	ANY16_U
(d)	+ (P)	存储了加法运算数据的软元件	-32768~32767	带符号BIN16位	ANY16_S
	+ (P)_U		0~65535	无符号BIN16位	ANY16_U

■ 可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	—	—	—	—

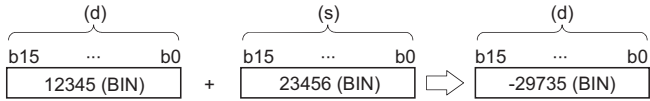
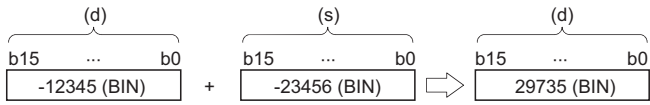
功能

- 将(d)中指定的BIN16位数据与(s)中指定的BIN16位数据进行加法运算，将加法运算结果存储到(d)中指定的软元件中。



- 加法运算结果中发生了上溢的情况下，位上升将被忽略。在此情况下，SM700(进位标志)不变为ON。

[+(P)的情况下]



[+(P)_U的情况下]

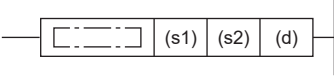


出错

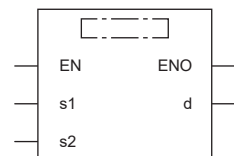
没有运算错误。

+ (P) (_U) [操作数为3个的情况下]

对指定的2个BIN16位数据进行加法运算。



梯形图	ST
	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) ;

FBD/LD



(□中放入PLUS、PLUSP、PLUS_U、PLUSP_U。)

■执行条件

指令	执行条件
+ +_U	
+P +P_U	

设置数据

■内容、范围、数据类型

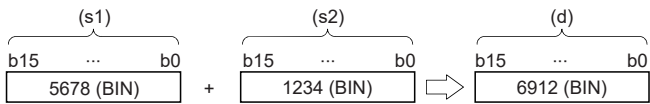
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	+ (P)	-32768~32767	带符号BIN16位	ANY16_S
	+ (P)_U	0~65535	无符号BIN16位	ANY16_U
(s2)	+ (P)	-32768~32767	带符号BIN16位	ANY16_S
	+ (P)_U	0~65535	无符号BIN16位	ANY16_U
(d)	+ (P)	—	带符号BIN16位	ANY16_S
	+ (P)_U	—	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—

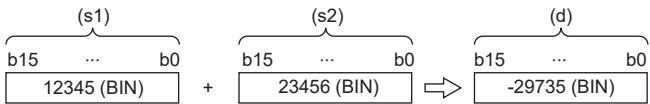
功能

- 将(s1)中指定的BIN16位数据与(s2)中指定的BIN16位数据进行加法运算，将加法运算结果存储到(d)中指定的软元件中。



- 加法运算结果中发生了上溢的情况下，位上升将被忽略。在此情况下，SM700(进位标志)不变为ON。

[+(P)的情况下]



[+(P)_U的情况下]



出错

没有运算错误。

BIN16位减法运算

-(P) (_U) [操作数为2个的情况下]

对指定的2个BIN16位数据进行减法运算。

梯形图	ST
	不对应。 (☞ 267页 -(P) (_U) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 267页 -(P) (_U) [操作数为3个的情况下])

■执行条件

指令	执行条件
- -_U	
-P -P_U	

设置数据

■内容、范围、数据类型

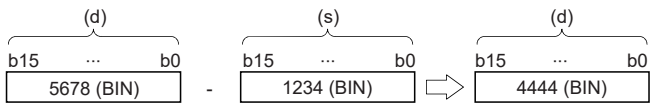
操作数	内容	范围	数据类型	数据类型(标签)	
(s)	-(P)	减数数据或存储了减数数据的软元件	-32768~32767	带符号BIN16位	ANY16_S
	-(P)_U		0~65535	无符号BIN16位	ANY16_U
(d)	-(P)	存储了被减数数据的软元件	-32768~32767	带符号BIN16位	ANY16_S
	-(P)_U		0~65535	无符号BIN16位	ANY16_U

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	—	—	—	—

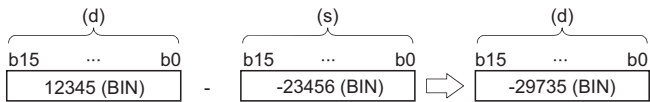
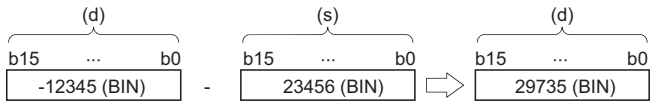
功能

- 将(d)中指定的BIN16位数据与(s)中指定的BIN16位数据进行减法运算，将减法运算结果存储到(d)中指定的软元件中。

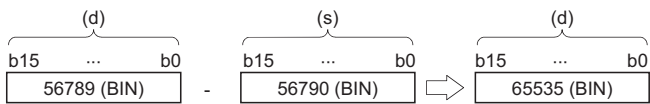


- 减法运算结果中发生了下溢的情况下，位下降将被忽略。在此情况下，SM700(进位标志)不变为0N。

[-(P)的情况下]



[-(P)_U的情况下]



出错

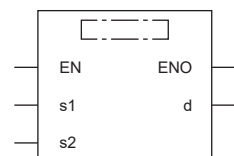
没有运算错误。

-(P) (_U) [操作数为3个的情况下]

对指定的2个BIN16位数据进行减法运算。

梯形图	ST
	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);

FBD/LD



(□中放入MINUS、MINUSP、MINUS_U、MINUSP_U。)

■执行条件

指令	执行条件
- -_U	
-P -P_U	

设置数据

■内容、范围、数据类型

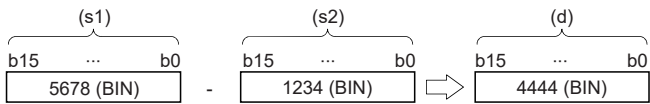
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	-(P)	-32768~32767	带符号BIN16位	ANY16_S
	-(P)_U	0~65535	无符号BIN16位	ANY16_U
(s2)	-(P)	-32768~32767	带符号BIN16位	ANY16_S
	-(P)_U	0~65535	无符号BIN16位	ANY16_U
(d)	-(P)	—	带符号BIN16位	ANY16_S
	-(P)_U	—	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—

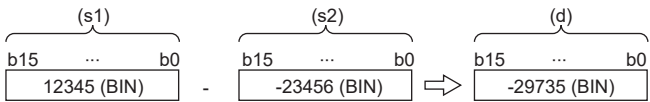
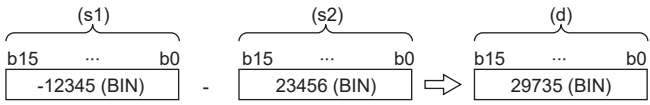
功能

- 将(s1)中指定的BIN16位数据与(s2)中指定的BIN16位数据进行减法运算，将减法运算结果存储到(d)中指定的软元件中。

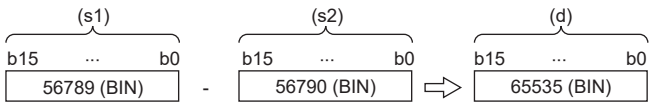


- 减法运算结果中发生了下溢的情况下，位下降将被忽略。在此情况下，SM700(进位标志)不变为0N。

[-(P)的情况下]



[-(P)_U的情况下]



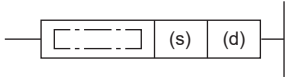
出错

没有运算错误。

BIN32位加法运算



D+(P) (_U) [操作数为2个的情况下]

对指定的2个BIN32位数据进行加法运算。

梯形图	ST
	不对应。 (☞ 271页 D+(P) (_U) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 271页 D+(P) (_U) [操作数为3个的情况下])

■执行条件

指令	执行条件
D+ D+_U	
D+P D+P_U	

设置数据

■内容、范围、数据类型

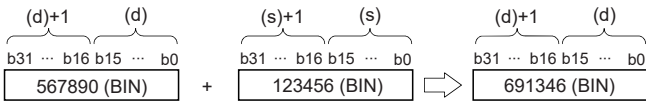
操作数	内容	范围	数据类型	数据类型(标签)	
(s)	D+(P)	加法运算数据或存储了加法运算数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	D+(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(d)	D+(P)	存储了加法运算数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	D+(P)_U		0~4294967295	无符号BIN32位	ANY32_U

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	○	○	—	—	—	—	—

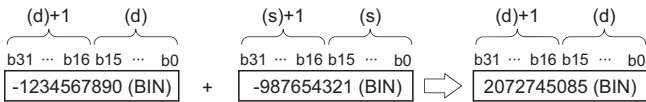
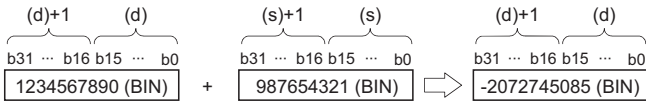
功能

- 将(d)中指定的BIN32位数据与(s)中指定的BIN32位数据进行加法运算，将加法运算结果存储到(d)中指定的软元件中。

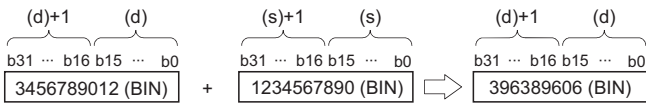


- 加法运算结果中发生了上溢的情况下，位上升将被忽略。在此情况下，SM700(进位标志)不变为ON。

[D+(P)的情况下]



[D+(P)_U的情况下]



出错

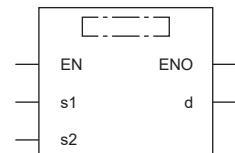
没有运算错误。

D+(P) (_U) [操作数为3个的情况下]

对指定的2个BIN32位数据进行加法运算。

梯形图	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



(□中放入DPLUS、DPLUSP、DPLUS_U、DPLUSP_U。)

■执行条件

指令	执行条件
D+ D+_U	
D+P D+P_U	

设置数据

■内容、范围、数据类型

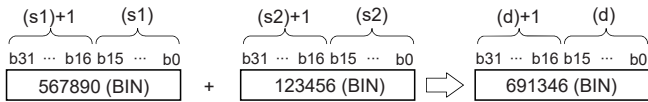
操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	D+(P)	加法运算的数据或存储了加法运算数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	D+(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(s2)	D+(P)	加法运算数据或存储了加法运算数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	D+(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(d)	D+(P)	存储运算结果的起始软元件	—	带符号BIN32位	ANY32_S
	D+(P)_U			无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—

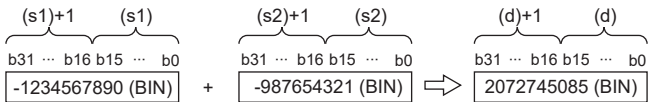
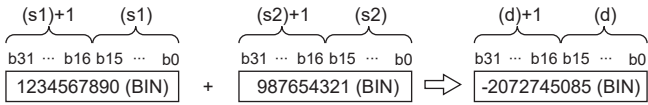
功能

- 将(s1)中指定的BIN32位数据与(s2)中指定的BIN32位数据进行加法运算，将加法运算结果存储到(d)中指定的软元件中。

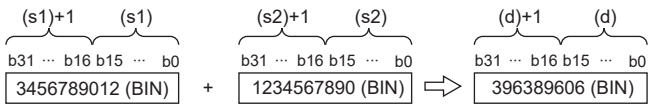


- 加法运算结果中发生了上溢的情况下，位上升将被忽略。在此情况下，SM700(进位标志)不变为ON。

[D+(P)的情况下]



[D+(P)_U的情况下]



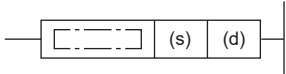
出错

没有运算错误。

BIN32位减法运算


D-(P) (_U) [操作数为2个的情况下]

对指定的2个BIN32位数据进行减法运算。

梯形图	ST
	不对应。 (☞ 275页 D-(P) (_U) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 275页 D-(P) (_U) [操作数为3个的情况下])

■执行条件

指令	执行条件
D- D-_U	
D-P D-P_U	

设置数据

■内容、范围、数据类型

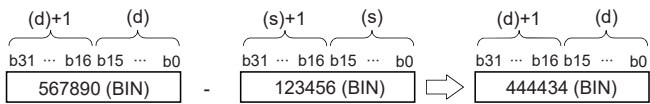
操作数	内容	范围	数据类型	数据类型(标签)	
(s)	D-(P)	减数数据或存储了减数数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	D-(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(d)	D-(P)	存储了被减数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	D-(P)_U		0~4294967295	无符号BIN32位	ANY32_U

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—	—

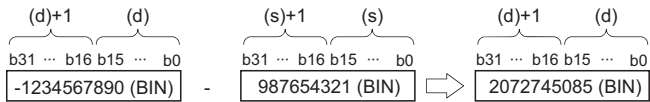
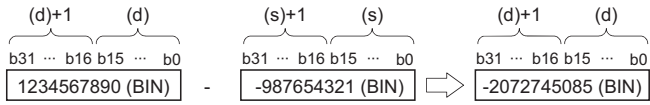
功能

- 将(d)中指定的BIN32位数据与(s)中指定的BIN32位数据进行减法运算，将减法运算结果存储到(d)中指定的软元件中。

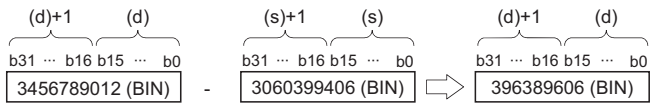


- 减法运算结果中发生了下溢的情况下，位上升将被忽略。在此情况下，SM700(进位标志)不变为ON。

[D-(P)的情况下]



[D-(P)_U的情况下]



出错

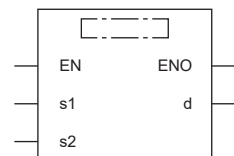
没有运算错误。

D-(P) (_U) [操作数为3个的情况下]

对指定的2个BIN32位数据进行减法运算。

梯形图	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



(□中放入DMINUS、DMINUSP、DMINUS_U、DMINUSP_U。)

■执行条件

指令	执行条件
D- D-_U	
D-P D-P_U	

设置数据

■内容、范围、数据类型

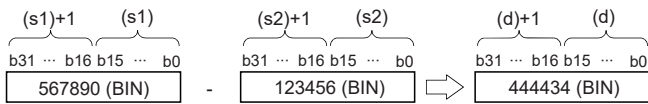
操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	D-(P)	被减数据或存储了被减数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	D-(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(s2)	D-(P)	减数数据或存储了减数数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	D-(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(d)	D-(P)	存储运算结果的起始软元件	—	带符号BIN32位	ANY32_S
	D-(P)_U			无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	○	○	—	—	—	—	—

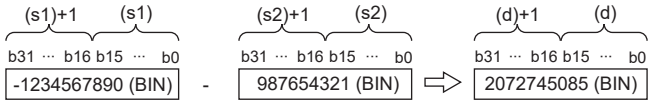
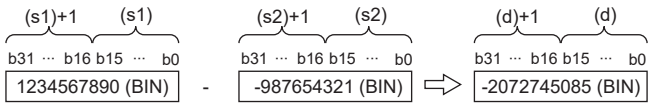
功能

- 将(s1)中指定的BIN32位数据与(s2)中指定的BIN32位数据进行减法运算，将减法运算结果存储到(d)中指定的软元件中。

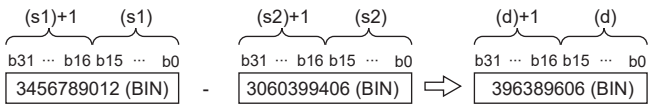


- 减法运算结果中发生了下溢的情况下，位上升将被忽略。在此情况下，SM700(进位标志)不变为ON。

[D-(P)的情况下]



[D-(P)_U的情况下]



出错

没有运算错误。

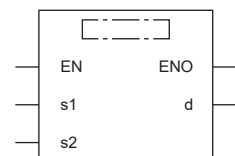
BIN16位乘法运算

* (P) (_U)

对指定的2个BIN16位数据进行乘法运算。

梯形图	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



(□中放入MULTI、MULTIP、MULTI_U、MULTIP_U。)

■执行条件

指令	执行条件
* *_U	
*P *P_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	* (P)	乘法运算数据或存储了乘法运算数据的软元件	-32768~32767	带符号BIN16位	ANY16_S
	* (P)_U		0~65535	无符号BIN16位	ANY16_U
(s2)	* (P)	乘法运算数据或存储了乘法运算数据的软元件	-32768~32767	带符号BIN16位	ANY16_S
	* (P)_U		0~65535	无符号BIN16位	ANY16_U
(d)	* (P)	存储运算结果的起始软元件	—	带符号BIN32位	ANY32_S
	* (P)_U			无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

- 将(s1)中指定的BIN16位数据与(s2)中指定的BIN16位数据进行乘法运算，将乘法运算结果存储到(d)中指定的软元件中。



- (d)为位软元件的情况下，指定将从低位的位开始。

例

(d)为位软元件情况下的乘法运算结果

- K1...低位的4位 (b0~b3)
- K4...低位的16位 (b0~b15)
- K8...低位的32位 (b0~b31)

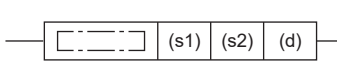
出错

没有运算错误。

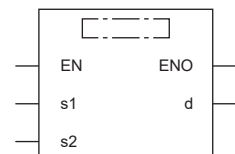
BIN16位除法运算

/ (P) (_U)

对指定的2个BIN16位数据进行除法运算。


梯形图	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



(□中放入DIVISION、DIVISIONP、DIVISION_U、DIVISIONP_U。)

■执行条件

指令	执行条件
/ /_U	
/P /P_U	

设置数据

■内容、范围、数据类型

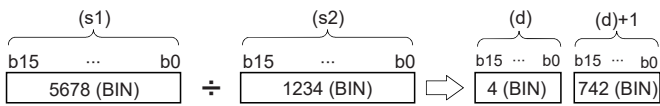
操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	/ (P)	被除数据或存储了被除数据的软元件	-32768~32767	带符号BIN16位	ANY16_S
	/ (P)_U		0~65535	无符号BIN16位	ANY16_U
(s2)	/ (P)	除数数据或存储了除数数据的软元件	-32768~32767	带符号BIN16位	ANY16_S
	/ (P)_U		0~65535	无符号BIN16位	ANY16_U
(d)	/ (P)	存储运算结果的起始软元件	—	带符号BIN32位	ANY16_S_ARRAY (要素数: 2)
	/ (P)_U			无符号BIN32位	ANY16_U_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—

功能

- 将(s1)中指定的BIN16位数据与(s2)中指定的BIN16位数据进行除法运算，将除法运算结果存储到(d)中指定的软元件中。



(d): 商

(d)+1: 余数

- 除法运算结果使用32位存储商及余数。位软元件的情况下，使用位指定中指定的位数，存储商及余数。
 - 商...被存储到低位16位中。
 - 余数...被存储到高位16位中。

出错

错误代码(SD0)	内容
3280H	(s2)中指定的值(除数)为0时。

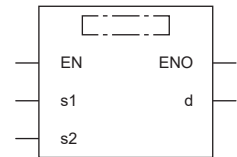
BIN32位乘法运算

D*(P) (_U)

对指定的2个BIN32位数据进行乘法运算。

梯形图	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



(□中放入DMULTI、DMULTIP、DMULTI_U、DMULTIP_U。)

■执行条件

指令	执行条件
D* D*_U	
D*P D*P_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	D*(P)	乘法运算数据或存储了乘法运算数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	D*(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(s2)	D*(P)	乘法运算数据或存储了乘法运算数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	D*(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(d)	D*(P)	存储运算结果的起始软元件	—	带符号BIN64位	ANY32_S_ARRAY (要素数: 2)
	D*(P)_U			无符号BIN64位	ANY32_U_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	—	○	—	—	○	—	○	—	—	—	—

功能

- 将(s1)中指定的BIN32位数据与(s2)中指定的BIN32位数据进行乘法运算，将乘法运算结果存储到(d)中指定的软元件中。



- (d)为位软元件的情况下，乘法运算结果的低位32位为止将成为对象，不能指定高位32位。位软元件中需要乘法运算结果的高位32位数据的情况下，应预先将数据存储到字软元件中一次，将字软元件的(d)+2、(d)+3的数据传送到指定位软元件中。

例

(d)为位软元件情况下的乘法运算结果

- K1...低位的4位(b0~b3)
- K4...低位的16位(b0~b15)
- K8...低位的32位(b0~b31)

出错

没有运算错误。

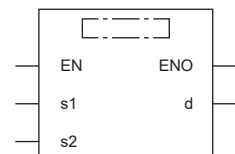
BIN32位除法运算

D/(P) (_U)

对指定的2个BIN32位数据进行除法运算。

梯形图	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



(□中放入DDIVISION、DDIVISIONP、DDIVISION_U、DDIVISIONP_U。)

■执行条件

指令	执行条件
D/ D/_U	
D/P D/P_U	

设置数据

■内容、范围、数据类型

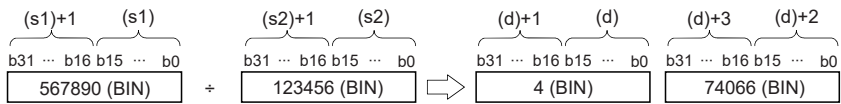
操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	D/(P)	被除数据或存储了被除数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	D/(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(s2)	D/(P)	除数数据或存储了除数数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	D/(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(d)	D/(P)	存储运算结果的起始软元件	—	带符号BIN64位	ANY32_S_ARRAY (要素数: 2)
	D/(P)_U			无符号BIN64位	ANY32_U_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	—	○	—	—	○	—	○	—	—	—	—	—

功能

- 将(s1)中指定的BIN32位数据与(s2)中指定的BIN32位数据进行除法运算，将除法运算结果存储到(d)中指定的软元件中。



- 字软元件的情况下，除法运算结果使用64位存储商及余数。商被存储到低位32位中，余数被存储到高位32位中。位软元件的情况下，只使用32位存储商。

出错

错误代码(SD0)	内容
3280H	(s2)中指定的值(除数)为0时。

BCD4位加法运算

B+(P) [操作数为2个的情况下]

对指定的2个BCD4位数据进行加法运算。

梯形图	ST
	不对应。 (☞ 286页 B+(P) [操作数为3个的情况下])
FBD/LD	
不对应。 (☞ 286页 B+(P) [操作数为3个的情况下])	

■执行条件

指令	执行条件
B+	
B+P	

设置数据

■内容、范围、数据类型

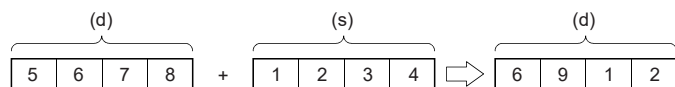
操作数	内容	范围	数据类型	数据类型(标签)
(s)	加法运算数据或存储了加法运算数据的软元件	0~9999	BCD4位	ANY16
(d)	存储了加法运算数据的软元件	0~9999	BCD4位	ANY16

■可用软元件

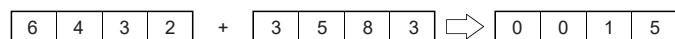
操作数	位		字		双字		间接指定	常数			其它	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K、H	E		\$
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 将(d)中指定的BCD4位数据与(s)中指定的BCD4位数据进行加法运算，将加法运算结果存储到(d)中指定的软元件中。



- 加法运算结果超过了9999的情况下，位上升将被忽略。在此情况下，SM700(进位标志)不变为ON。



出错

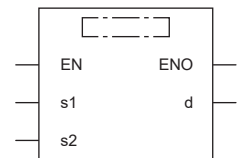
错误代码(SD0)	内容
3285H	(s)的BCD数据为0~9999以外时。
	(d)的BCD数据为0~9999以外时。

B+(P) [操作数为3个的情况下]

对指定的2个BCD4位数据进行加法运算。

梯形图	ST
	ENO:=BPLUS(EN, s1, s2, d); ENO:=BPLUSP(EN, s1, s2, d);

FBD/LD



(□中放入BPLUS、BPLUSP。)

■执行条件

指令	执行条件
B+	
B+P	

设置数据

■内容、范围、数据类型

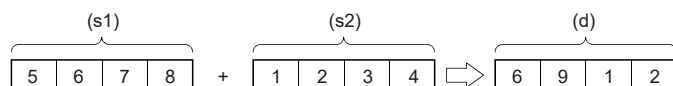
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	加法运算的数据或存储了加法运算数据的软元件	0~9999	BCD4位	ANY16
(s2)	加法运算数据或存储了加法运算数据的软元件	0~9999	BCD4位	ANY16
(d)	存储运算结果的软元件	—	BCD4位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

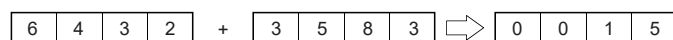
操作数	位		字		双字			间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 将(s1)中指定的BCD4位数据与(s2)中指定的BCD4位数据进行加法运算，将加法运算结果存储到(d)中指定的软元件中。



- 加法运算结果超过了9999的情况下，位上升将被忽略。在此情况下，SM700(进位标志)不变为ON。



错误代码 (SD0)	内容
3285H	(s1) 的BCD数据为0~9999以外时。
	(s2) 的BCD数据为0~9999以外时。

BCD4位减法运算

B-(P) [操作数为2个的情况下]

对指定的2个BCD4位数据进行减法运算。

梯形图	ST
	不对应。 (☞ 289页 B-(P) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 289页 B-(P) [操作数为3个的情况下])

■执行条件

指令	执行条件
B-	
B-P	

设置数据

■内容、范围、数据类型

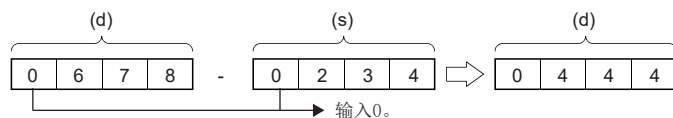
操作数	内容	范围	数据类型	数据类型(标签)
(s)	减数数据或存储了减数数据的软元件	0~9999	BCD4位	ANY16
(d)	存储了被减数据的软元件	0~9999	BCD4位	ANY16

■可用软元件

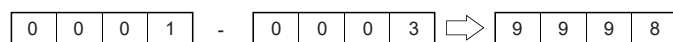
操作数	位		字		双字		间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K、H	E	
(s)	○	○	○	○	○	—	—	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—

功能

- 将(d)中指定的BCD4位数据与(s)中指定的BCD4位数据进行减法运算，将减法运算结果存储到(d)中指定的软元件中。



- 减法运算结果发生了下溢时，其情况如下所示。在此情况下，SM700(进位标志)不变为ON。



出错

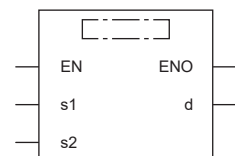
错误代码(SD0)	内容
3285H	(s)的BCD数据为0~9999以外时。 (d)的BCD数据为0~9999以外时。

B-(P) [操作数为3个的情况下]

对指定的2个BCD4位数据进行减法运算。

梯形图	ST
	ENO:=BMINUS(EN, s1, s2, d); ENO:=BMINUSP(EN, s1, s2, d);

FBD/LD



(□中放入BMINUS、BMINUSP。)

■执行条件

指令	执行条件
B-	
B-P	

设置数据

■内容、范围、数据类型

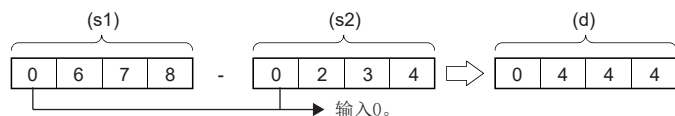
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	被减数据或存储了被减数据的软元件	0~9999	BCD4位	ANY16
(s2)	减数数据或存储了减数数据的软元件	0~9999	BCD4位	ANY16
(d)	存储运算结果的软元件	—	BCD4位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

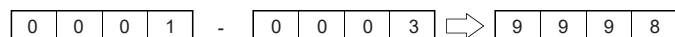
操作数	位		字		双字			间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	○	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 将(s1)中指定的BCD4位数据与(s2)中指定的BCD4位数据进行减法运算，将减法运算结果存储到(d)中指定的软元件中。



- 减法运算结果发生了下溢时，其情况如下所示。在此情况下，SM700(进位标志)不变为ON。



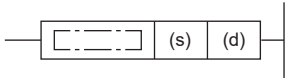
出错

错误代码 (SD0)	内容
3285H	(s1) 的BCD数据为0~9999以外时。
	(s2) 的BCD数据为0~9999以外时。



BCD8位加法运算

DB+(P) [操作数为2个的情况下]

对指定的2个BCD8位数据进行加法运算。

梯形图 	ST 不对应。 (☞ 292页 DB+(P) [操作数为3个的情况下])
FBD/LD 不对应。 (☞ 292页 DB+(P) [操作数为3个的情况下])	

■执行条件

指令	执行条件
DB+	
DB+P	

设置数据

■内容、范围、数据类型

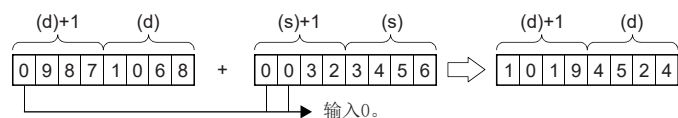
操作数	内容	范围	数据类型	数据类型(标签)
(s)	加法运算数据或存储了加法运算数据的起始软元件	0~99999999	BCD8位	ANY32
(d)	存储了加法运算数据的起始软元件	0~99999999	BCD8位	ANY32

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	○	○	○	○	○	○	○	—	—	—	—

功能

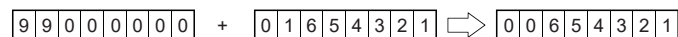
- 将(d)中指定的BCD8位数据与(s)中指定的BCD8位数据进行加法运算，将加法运算结果存储到(d)中指定的软元件中。



(d)+1, (s)+1: 高位4位

(d), (s): 低位4位

- 加法运算结果超过了99999999的情况下，进位将被忽略。在此情况下，SM700(进位标志)不变为0N。

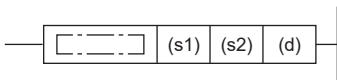


出错

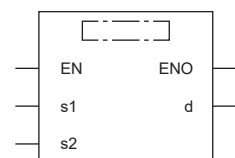
错误代码(SD0)	内容
3285H	(s)的BCD数据为0~99999999以外时。
	(d)的BCD数据为0~99999999以外时。

DB+ (P) [操作数为3个的情况下]

对指定的2个BCD8位数据进行加法运算。



梯形图	ST
	ENO:=DBPLUS (EN, s1, s2, d); ENO:=DBPLUSP (EN, s1, s2, d);

FBD/LD



(□中放入DBPLUS、DBPLUSP。)

■执行条件

指令	执行条件
DB+	
DB+P	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	加法运算的数据或存储了加法运算数据的起始软元件	0~99999999	BCD8位	ANY32
(s2)	加法运算数据或存储了加法运算数据的起始软元件	0~99999999	BCD8位	ANY32
(d)	存储运算结果的起始软元件	—	BCD8位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	○	○	○	—	—	—	—

功能

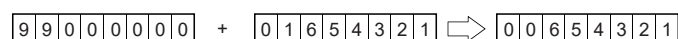
- 将(s1)中指定的BCD8位数据与(s2)中指定的BCD8位数据进行加法运算，将加法运算结果存储到(d)中指定的软元件中。



(d)+1, (s1)+1, (s2)+1: 高位4位

(d), (s1), (s2): 低位4位

- 加法运算结果超过了99999999的情况下，进位将被忽略。在此情况下，SM700(进位标志)不变为0N。



错误代码 (SD0)	内容
3285H	(s1) 的BCD数据为0~99999999以外时。
	(s2) 的BCD数据为0~99999999以外时。

BCD8位减法运算

DB-(P) [操作数为2个的情况下]

对指定的2个BCD8位数据进行减法运算。

梯形图	ST
	不对应。 (☞ 295页 DB-(P) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 295页 DB-(P) [操作数为3个的情况下])

■执行条件

指令	执行条件
DB-	
DB-P	

设置数据

■内容、范围、数据类型

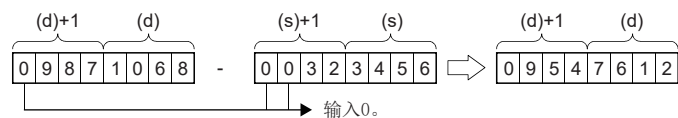
操作数	内容	范围	数据类型	数据类型(标签)
(s)	减数数据或存储了减数数据的起始软元件	0~99999999	BCD8位	ANY32
(d)	被减数据或存储了被减数据的起始软元件	0~99999999	BCD8位	ANY32

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	○	○	○	—	—	—	—

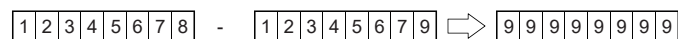
功能

- 将(d)中指定的BCD8位数据与(s)中指定的BCD8位数据进行减法运算，将减法运算结果存储到(d)中指定的软元件中。



(d)+1, (s)+1: 高位4位
 (d), (s): 低位4位

- 减法运算结果发生了下溢时，其情况如下所示。在此情况下，SM700(进位标志)不变为0N。



出错

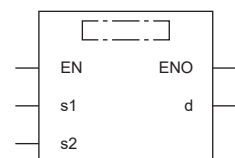
错误代码(SD0)	内容
3285H	(s)的BCD数据为0~99999999以外时。
	(d)的BCD数据为0~99999999以外时。

DB-(P) [操作数为3个的情况下]

对指定的2个BCD8位数据进行减法运算。

梯形图	ST
	ENO:=DBMINUS(EN, s1, s2, d); ENO:=DBMINUSP(EN, s1, s2, d);

FBD/LD



(□中放入DBMINUS、DBMINUSP。)

■执行条件

指令	执行条件
DB-	
DB-P	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	被减数据或存储了被减数据的起始软元件	0~99999999	BCD8位	ANY32
(s2)	减数数据或存储了减数数据的起始软元件	0~99999999	BCD8位	ANY32
(d)	存储运算结果的起始软元件	—	BCD8位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	○	○	○	○	—	—	—

功能

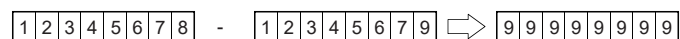
- 将(s1)中指定的BCD8位数据与(s2)中指定的BCD8位数据进行减法运算，将减法运算结果存储到(d)中指定的软元件中。



(d)+1, (s1)+1, (s2)+1: 高位4位

(d), (s1), (s2): 低位4位

- 减法运算结果发生了下溢时，其情况如下所示。在此情况下，SM700(进位标志)不变为ON。



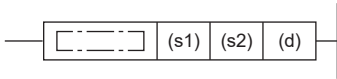
出错

错误代码 (SD0)	内容
3285H	(s1) 的BCD数据为0~99999999以外时。
	(s2) 的BCD数据为0~99999999以外时。

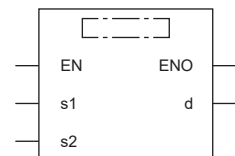
BCD4位乘法运算

B*(P)

对指定的2个BCD4位数据进行乘法运算。



梯形图	ST
	不对应。

FBD/LD



(□中放入BMULTI、BMULTIP。)

■执行条件

指令	执行条件
B*	
B*P	

设置数据

■内容、范围、数据类型

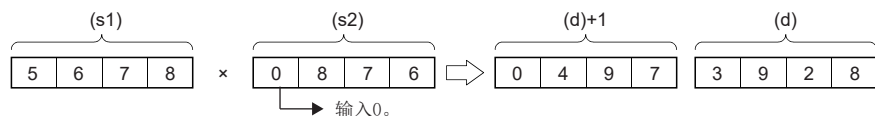
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	乘法运算数据或存储了乘法运算数据的软元件	0~9999	BCD4位	ANY16
(s2)	乘法运算数据或存储了乘法运算数据的软元件	0~9999	BCD4位	ANY16
(d)	存储运算结果的起始软元件	—	BCD8位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

将(s1)中指定的BCD4位数据与(s2)中指定的BCD4位数据进行乘法运算，将乘法运算结果存储到(d)中指定的软元件中。((d)+1为高位4位，(d)为低位4位)



出错

错误代码 (SD0)	内容
3285H	(s1) 的BCD数据为0~9999以外时。
	(s2) 的BCD数据为0~9999以外时。

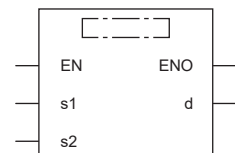
BCD4位除法运算

B/(P)

对指定的2个BCD4位数据进行除法运算。

梯形图	ST
	不对应。

FBD/LD



(□中放入BDIVISION、BDIVISIONP。)

■执行条件

指令	执行条件
B/	
B/P	

设置数据

■内容、范围、数据类型

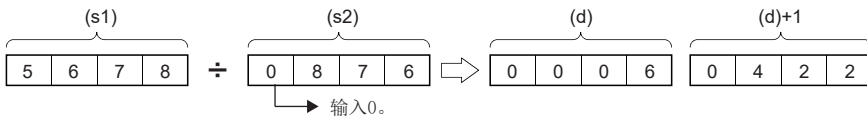
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	被除数据或存储了被除数据的软元件	0~9999	BCD4位	ANY16
(s2)	除数数据或存储了除数数据的软元件	0~9999	BCD4位	ANY16
(d)	存储运算结果的起始软元件	—	BCD8位	ANY16_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

- 将(s1)中指定的BCD4位数据与(s2)中指定的BCD4位数据进行除法运算，将除法运算结果存储到(d)中指定的软元件中。



(d): 商

(d)+1: 余数

- 除法运算结果使用32位存储商及余数。
- 商(BCD4位)…被存储到低位16位中。
- 余数(BCD4位)…被存储到高位16位中。

出错

错误代码(SD0)	内容
3280H	(s2)中指定的值(除数)为0时。
3285H	(s1)的BCD数据为0~9999以外时。
	(s2)的BCD数据为0~9999以外时。

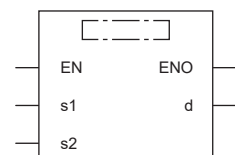
BCD8位乘法运算

DB*(P)

对指定的2个BCD8位数据进行乘法运算。

梯形图	ST
	不对应。

FBD/LD



(□中放入DBMULTI、DBMULTIP。)

■执行条件

指令	执行条件
DB*	
DB*P	

设置数据

■内容、范围、数据类型

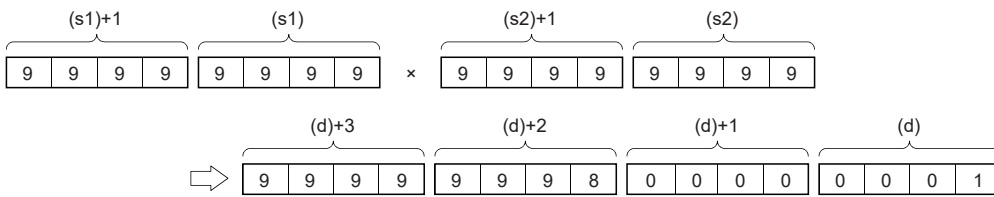
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	乘法运算数据或存储了乘法运算数据的起始软元件	0~99999999	BCD8位	ANY32
(s2)	乘法运算数据或存储了乘法运算数据的起始软元件	0~99999999	BCD8位	ANY32
(d)	存储运算结果的起始软元件	—	BCD16位	ANY32_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	—	○	—	—	○	—	○	—	—	—	—

功能

- 将(s1)中指定的BCD8位数据与(s2)中指定的BCD8位数据进行乘法运算，将乘法运算结果存储到(d)中指定的软元件中。



- (d)为位软元件的情况下，乘法运算结果的低位8位(低位32位)为止将成为对象，不能指定高位8位(高位32位)。

例

(d)为位软元件情况下的乘法运算结果

- K1...低位1位(b0~b3)
- K4...低位4位(b0~b15)
- K8...低位8位(b0~b31)

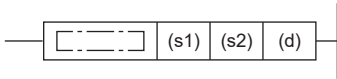
出错

错误代码(SD0)	内容
3285H	(s1)的BCD数据为0~99999999以外时。
	(s2)的BCD数据为0~99999999以外时。

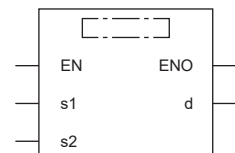
BCD8位除法运算

DB/(P)

对指定的2个BCD8位数据进行除法运算。


梯形图	ST
	不对应。

FBD/LD



(□中放入DBDIVISION、DBDIVISIONP。)

■执行条件

指令	执行条件
DB/	
DB/P	

设置数据

■内容、范围、数据类型

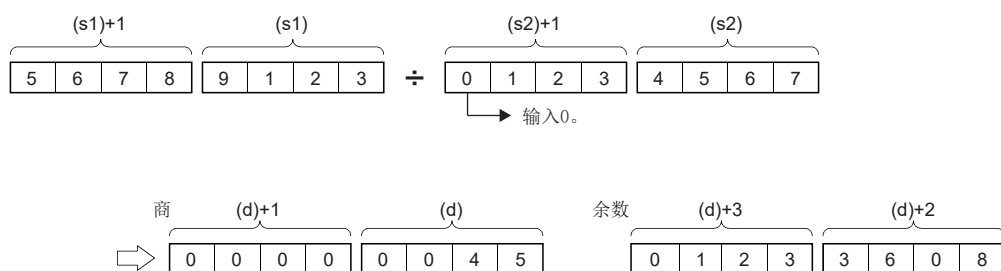
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	被除数据或存储了被除数据的起始软元件	0~99999999	BCD8位	ANY32
(s2)	除数数据或存储了除数数据的起始软元件	0~99999999	BCD8位	ANY32
(d)	存储运算结果的起始软元件	—	BCD16位	ANY32_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	—	○	—	—	○	—	○	—	—	—	—

功能

- 将(s1)中指定的BCD8位数据与(s2)中指定的BCD8位数据进行除法运算，将除法运算结果存储到(d)中指定的软元件中。



(d)+1, (d)+3: 高位4位

(d), (d)+2: 低位4位

- 除法运算结果使用64位存储商及余数。
- 商(BCD8位)…被存储到低位32位中。
- 余数(BCD8位)…被存储到高位32位中。
- 将(d)以位软元件进行了指定的情况下，除法运算结果的余数将不被存储。

出错

错误代码(SD0)	内容
3280H	(s2)中指定的值(除数)为0时。
3285H	(s1)的BCD数据为0~99999999以外时。
	(s2)的BCD数据为0~99999999以外时。

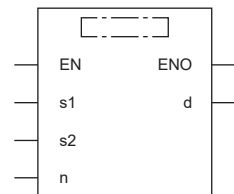
BIN16位块数据加法运算

BK+(P) (_U)

对指定的2个BIN16位数据块进行加法运算。

梯形图	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



(□中放入BKPLUS、BKPLUSP、BKPLUS_U、BKPLUSP_U。)

■执行条件

指令	执行条件
BK+ BK+_U	
BK+P BK+P_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	BK+(P)	-32768~32767	带符号BIN16位	ANY16_S*1
	BK+(P)_U	0~65535	无符号BIN16位	ANY16_U*1
(s2)	BK+(P)	-32768~32767	带符号BIN16位	ANY16_S*1
	BK+(P)_U	0~65535	无符号BIN16位	ANY16_U*1
(d)	BK+(P)	—	带符号BIN16位	ANY16_S*1
	BK+(P)_U	—	无符号BIN16位	ANY16_U*1
(n)	加法运算数据个数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

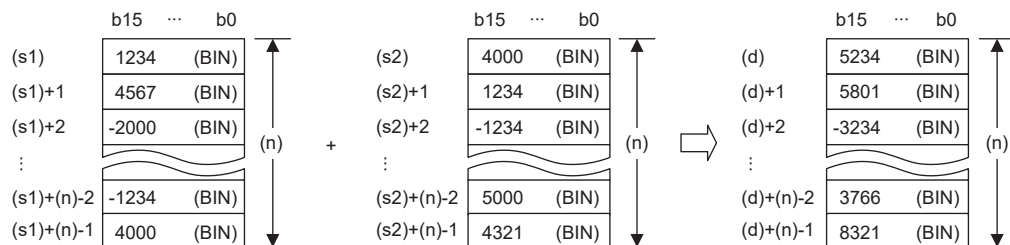
操作数	位		字			双字		间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	—	—	○	—	—	—	○	—	—	—	—	
(s2)	—	—	○	—	—	—	○	○	—	—	—	
(d)	—	—	○	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	○	○	—	—	—	

功能

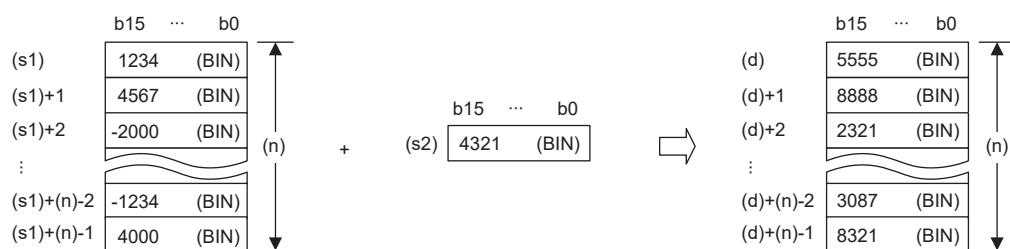
- 将(s1)中指定的软元件开始的(n)点的BIN16位数据与(s2)中指定的软元件开始的(n)点的BIN16位数据或常数进行加法运算，将加法运算结果存储到(d)中指定的软元件及其以后。
- 块加法运算以16位单位进行。

例

(s2)中指定了软元件的情况下(带符号指定时)



(s2)中指定了常数的情况下(带符号指定时)



- 运算结果发生了上溢时，其情况如下所示。在此情况下，SM700(进位标志)不变为0N。

指定了带符号的情况下	指定了无符号的情况下
$K32767 (H7FFF) + K2 (H0002) \Rightarrow K-32767 (H8001)$ $K-32767 (H8001) + K-2 (HFFFE) \Rightarrow K32767 (H7FFF)$	$K65535 (HFFFF) + K1 (H0001) \Rightarrow K0 (H0000)$

出错

错误代码(SD0)	内容
2821H	(s1)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围重复时。 (s1)及(d)中指定了同一软元件的情况下除外。)
	(s2)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围重复时。 (s2)及(d)中指定了同一软元件的情况下除外。)

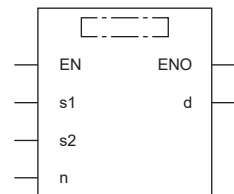
BIN16位块数据减法运算

BK-(P) (_U)

对指定的2个BIN16位数据块进行减法运算。

梯形图	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



(□中放入BKMINUS、BKMINUSP、BKMINUS_U、BKMINUSP_U。)

■执行条件

指令	执行条件
BK- BK-_U	
BK-P BK-P_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	BK-(P)	-32768~32767	带符号BIN16位	ANY16_S*1
	BK-(P)_U	0~65535	无符号BIN16位	ANY16_U*1
(s2)	BK-(P)	-32768~32767	带符号BIN16位	ANY16_S*1
	BK-(P)_U	0~65535	无符号BIN16位	ANY16_U*1
(d)	BK-(P)	—	带符号BIN16位	ANY16_S*1
	BK-(P)_U	—	无符号BIN16位	ANY16_U*1
(n)	减数数据个数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

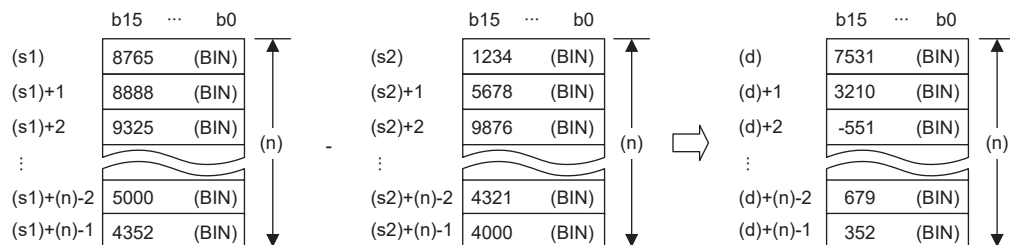
操作数	位		字		双字		间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K	H	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

功能

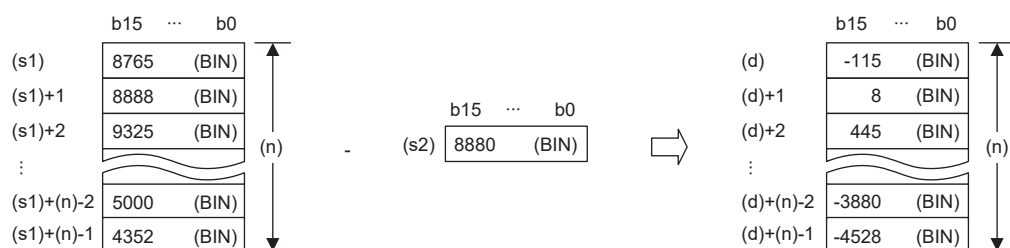
- 将(s1)中指定的软元件开始的(n)点的BIN16位数据与(s2)中指定的软元件开始的(n)点的BIN16位数据或常数进行减法运算，将减法运算结果存储到(d)中指定的软元件及其以后。
- 块减法运算以16位单位进行。

例

(s2)中指定了软元件的情况下



(s2)中指定了常数的情况下



- 运算结果发生了上溢时，其情况如下所示。在此情况下，SM700(进位标志)不变为0N。

指定了带符号的情况下		指定了无符号的情况下	
K-32768 (H8000)	- K2 (H0002)	⇒	K32766 (H7FFE)
K32767 (H7FFF)	- K-2 (HFFFE)	⇒	K-32767 (H8001)
			K0 (H0000) - K1 (H0001) ⇒ K65535 (HFFFF)

出错

错误代码(SD0)	内容
2821H	(s1)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围重复时。 (s1)及(d)中指定了同一软元件的情况下除外。)
	(s2)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围重复时。 (s2)及(d)中指定了同一软元件的情况下除外。)

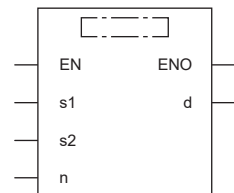
BIN32位块数据加法运算

DBK+(P) (_U)

对指定的2个BIN32位数据块进行加法运算。

梯形图	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



(□中放入DBKPLUS、DBKPLUSP、DBKPLUS_U、DBKPLUSP_U。)

■执行条件

指令	执行条件
DBK+ DBK+_U	
DBK+P DBK+P_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	DBK+(P)	加法运算的数据或存储了加法运算数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S*1
	DBK+(P)_U		0~4294967295	无符号BIN32位	ANY32_U*1
(s2)	DBK+(P)	加法运算数据或存储了加法运算数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S*1
	DBK+(P)_U		0~4294967295	无符号BIN32位	ANY32_U*1
(d)	DBK+(P)	存储运算结果的起始软元件	—	带符号BIN32位	ANY32_S*1
	DBK+(P)_U		无符号BIN32位	ANY32_U*1	
(n)	加法运算数据个数	0~65535	无符号BIN16位	ANY16	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

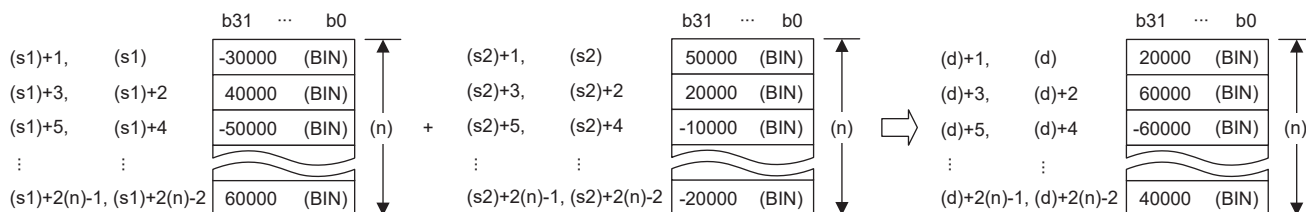
操作数	位		字		双字		间接指定	常数				其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K	H	E	
(s1)	—	—	○	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	○	○	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—	—

功能

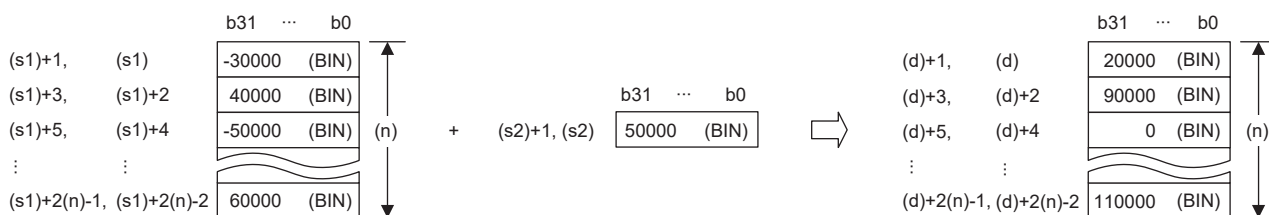
- 将(s1)中指定的软元件开始的(n)点的BIN32位数据与(s2)中指定的软元件开始的(n)点的BIN32位数据或常数进行加法运算，将加法运算结果存储到(d)中指定的软元件及其以后。
- 块加法运算以32位单位进行。

例

(s2)中指定了软元件的情况下(带符号指定时)



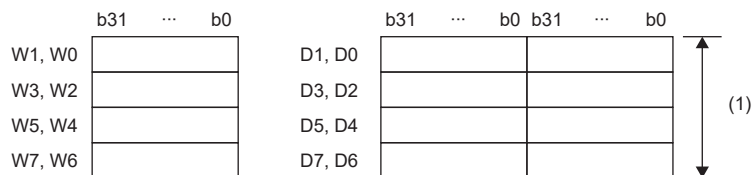
(s2)中指定了常数的情况下(带符号指定时)



- (s1)或(s2)与(d)以同一软元件(完全一致)指定的情况下，可以运算。但是，(s1)或(s2)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围有部分一致(重复)的情况下，将变为错误状态。

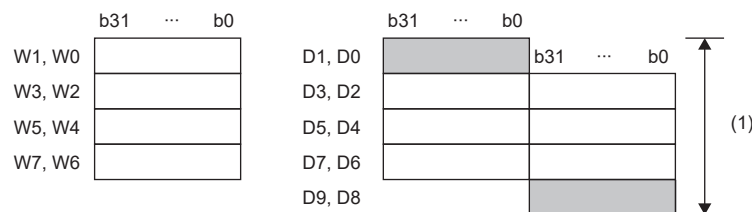
例

(s2)与(d)开始的4点的软元件完全一致的情况下



(1) 由于完全一致，因此可以运算。

(s2)、(d)开始的4点的软元件有部分一致的情况下



(1) 由于部分一致，因此变为运算错误。

- (n)中指定的值为0的情况下，将变为无处理。
- 运算结果发生了上溢时，其情况如下所示。在此情况下，SM700(进位标志)不变为ON。

指定了带符号的情况下		指定了无符号的情况下			
K2147483647 (H7FFFFFFF)	+ K2 (H00000002)	⇒ K-2147483647 (H80000001)	K4294967295 (HFFFFFFF)	+ K1 (H00000001)	⇒ K0 (H00000000)
K-2147483647 (H80000001)	+ K-2 (HFFFFFFFE)	⇒ K2147483647 (H7FFFFFFF)			

错误代码 (SD0)	内容
2821H	(s1) 开始的 (n) 点的软元件范围与 (d) 开始的 (n) 点的软元件范围重复时。 (s1) 及 (d) 中指定了同一软元件的情况下除外。)
	(s2) 开始的 (n) 点的软元件范围与 (d) 开始的 (n) 点的软元件范围重复时。 (s2) 及 (d) 中指定了同一软元件的情况下除外。)

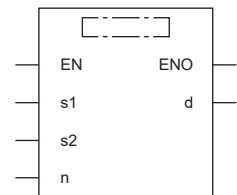
BIN32位块数据减法运算

DBK-(P) (_U)

对指定的2个BIN32位数据块进行减法运算。

梯形图	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) ;

FBD/LD



(□中放入DBKMINUS、DBKMINUSP、DBKMINUS_U、DBKMINUSP_U。)

■执行条件

指令	执行条件
DBK- DBK-_U	
DBK-P DBK-P_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	DBK-(P)	被减数据或存储了被减数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S*1
	DBK-(P)_U	被减数据或存储了被减数据的起始软元件	0~4294967295	无符号BIN32位	ANY32_U*1
(s2)	DBK-(P)	减数数据或存储了减数数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S*1
	DBK-(P)_U	减数数据或存储了减数数据的起始软元件	0~4294967295	无符号BIN32位	ANY32_U*1
(d)	DBK-(P)	存储运算结果的起始软元件	—	带符号BIN32位	ANY32_S*1
	DBK-(P)_U	存储运算结果的起始软元件	—	无符号BIN32位	ANY32_U*1
(n)	减数数据个数	0~65535	无符号BIN16位	ANY16	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

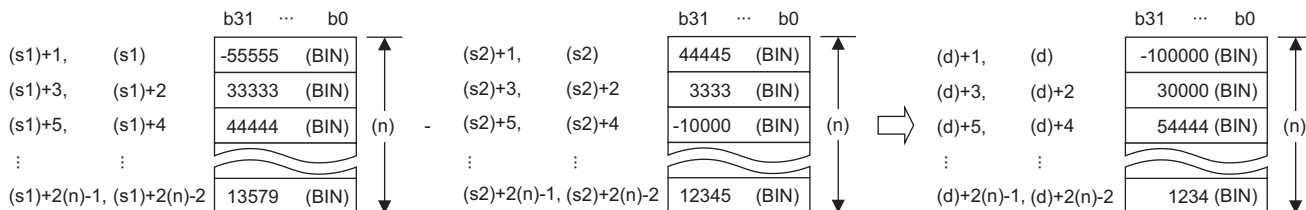
操作数	位		字		双字		间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K	H	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(s2)	—	—	○	—	—	—	○	○	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

功能

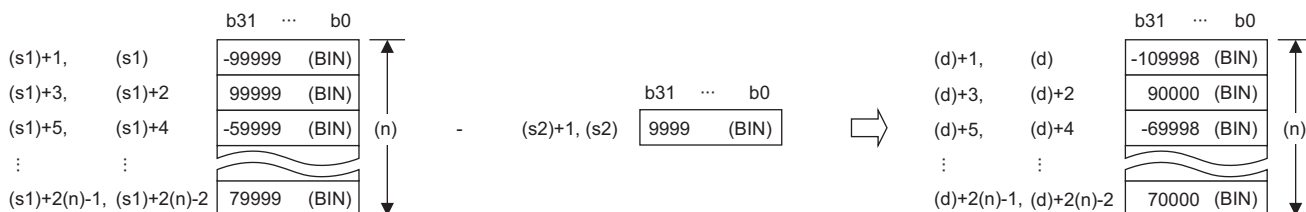
- 将(s1)中指定的软元件开始的(n)点的BIN32位数据与(s2)中指定的软元件开始的(n)点的BIN32位数据或常数进行减法运算，将减法运算结果存储到(d)中指定的软元件以后。
- 块减法运算以32位单位进行。

例

(s2)中指定了软元件的情况下(带符号指定时)



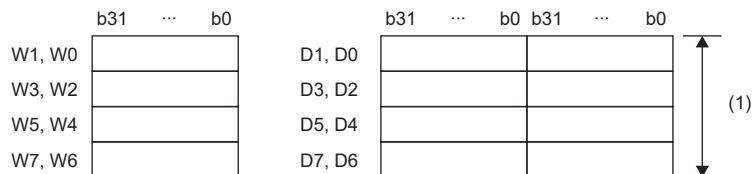
(s2)中指定了常数的情况下(带符号指定时)



- (s1)或(s2)与(d)以同一软元件(完全一致)指定的情况下，可以运算。但是，(s1)或(s2)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围有部分一致(重复)的情况下，将变为错误状态。

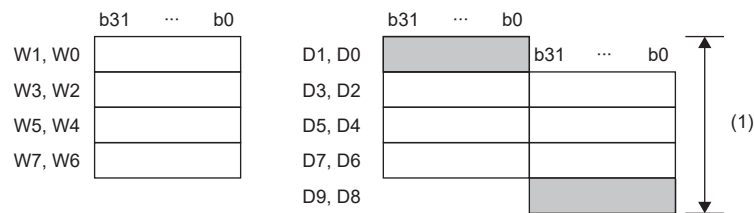
例

(s2)与(d)开始的4点的软元件完全一致的情况下



(1) 由于完全一致，因此可以运算。

(s2)、(d)开始的4点的软元件有部分一致的情况下



(1) 由于部分一致，因此变为运算错误。

- (n)中指定的值为0的情况下，将变为无处理。
- 运算结果发生了上溢时，其情况如下所示。在此情况下，SM700(进位标志)不变为0N。

指定了带符号的情况下		指定了无符号的情况下	
K2147483647 (H7FFFFFFF)	- K-2 (HFFFFFFFE)	⇒	K-2147483647 (H80000001)
K-2147483647 (H80000001)	- K2 (H00000002)	⇒	K2147483647 (H7FFFFFFF)
			K0 (H00000000) - K1 (H00000001) ⇒ K4294967295 (HFFFFFFF)

出错

错误代码 (SD0)	内容
2821H	(s1) 开始的 (n) 点的软元件范围与 (d) 开始的 (n) 点的软元件范围重复时。 ((s1) 及 (d) 中指定了同一软元件的情况下除外。)
	(s2) 开始的 (n) 点的软元件范围与 (d) 开始的 (n) 点的软元件范围重复时。 ((s2) 及 (d) 中指定了同一软元件的情况下除外。)

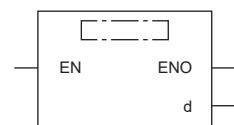
16位BIN数据递增

INC(P) (_U)

对指定的BIN16位数据进行+1。

梯形图	ST	
	ENO:=INC(EN, d); ENO:=INCP(EN, d);	ENO:=INC_U(EN, d); ENO:=INCP_U(EN, d);

FBD/LD



■执行条件

指令	执行条件
INC INC_U	
INCP INCP_U	

设置数据

■内容、范围、数据类型

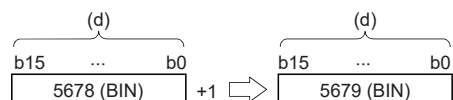
操作数	内容	范围	数据类型	数据类型(标签)	
(d)	INC(P)	进行+1的软元件	-32768~32767	带符号BIN16位	ANY16_S
	INC(P)_U		0~65535	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 对(d)中指定的软元件(BIN16位数据)进行+1。



- (d)中指定的软元件的内容为32767时执行了INC(P)指令的情况下，-32768将被存储到(d)中指定的软元件中。(指定了带符号的情况下)
- (d)中指定的软元件内容为65535时执行了INC(P)_U指令的情况下，0将被存储到(d)中指定的软元件中。(指定了无符号的情况下)

出错

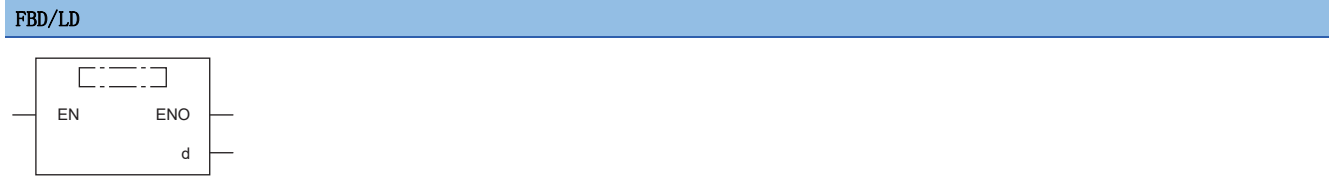
没有运算错误。

16位BIN数据递减

DEC(P) (_U)

对指定的BIN16位数据进行-1。

梯形图	ST	
	ENO:=DEC(EN, d); ENO:=DECP(EN, d);	ENO:=DEC_U(EN, d); ENO:=DECP_U(EN, d);



■执行条件

指令	执行条件
DEC DEC_U	
DECP DECP_U	

设置数据

■内容、范围、数据类型

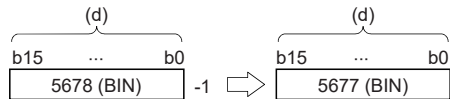
操作数	内容	范围	数据类型	数据类型(标签)
(d)	进行-1的软元件	-32768~32767	带符号BIN16位	ANY16_S
	DEC(P)_U	0~65535	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 对(d)中指定的软元件(BIN16位数据)进行-1。



- (d)中指定的软元件内容为-32768时执行了DEC(P)指令的情况下，32767将被存储到(d)中指定的软元件中。(指定了带符号的情况下)
- (d)中指定的软元件内容为0时执行了DEC(P)_U指令的情况下，65535将被存储到(d)中指定的软元件中。(指定了无符号的情况下)

出错

没有运算错误。

32位BIN数据递增

DINC(P) (_U)

对指定的BIN32位数据进行+1。

梯形图	ST	
	ENO:=DINC(EN, d); ENO:=DINCP(EN, d);	ENO:=DINC_U(EN, d); ENO:=DINCP_U(EN, d);

FBD/LD

■执行条件

指令	执行条件
DINC DINC_U	
DINCP DINCP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(d)	DINC(P)	进行+1的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	DINC(P)_U		0~4294967295	无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	○	○	○	—	—	—	—

功能

- 对(d)中指定的软元件内容(BIN32位数据)进行+1。



- (d)中指定的软元件内容为2147483647时执行了DINC(P) (_U)指令的情况下，-2147483648将被存储到(d)中指定的软元件中。(指定了带符号的情况下)
- (d)中指定的软元件内容为4294967295时执行了DINC(P) (_U)指令的情况下，0将被存储到(d)中指定的软元件中。(指定了无符号的情况下)

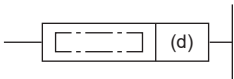
出错

没有运算错误。

32位BIN数据递减

DDEC(P) (_U)


对指定的BIN32位数据进行-1。

梯形图	ST	
	ENO:=DDEC(EN, d); ENO:=DDECP(EN, d);	ENO:=DDEC_U(EN, d); ENO:=DDECP_U(EN, d);

FBD/LD



■执行条件

指令	执行条件
DDEC DDEC_U	
DDECP DDECP_U	

设置数据

■内容、范围、数据类型

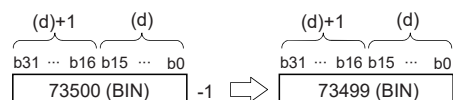
操作数	内容	范围	数据类型	数据类型(标签)	
(d)	DDEC(P)	进行-1的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	DDEC(P)_U		0~4294967295	无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	○	○	○	—	—	—	—

功能

- 对(d)中指定的软元件内容(BIN32位数据)进行-1。



- (d)中指定的软元件内容为-2147483648时执行了DDEC(P) (_U)指令的情况下，2147483647将被存储到(d)中指定的软元件中。(指定了带符号的情况下)
- (d)中指定的软元件内容为0时执行了DDEC(P) (_U)指令时，-1将被存储到(d)中指定的软元件中。(指定了带符号的情况下)
- (d)中指定的软元件内容为0时执行了DDEC(P) (_U)指令的情况下，4294967295将被存储到(d)中指定的软元件中。(指定了无符号的情况下)

出错

没有运算错误。

12.3 逻辑运算指令

16位数据逻辑与

WAND(P) [操作数为2个的情况下]

进行指定的2个BIN16位数据的逻辑与。

梯形图	ST
	不对应。 (☞ 320页 WAND(P) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 320页 WAND(P) [操作数为3个的情况下])

■执行条件

指令	执行条件
WAND	
WANDP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储进行逻辑与的数据或存储了数据的软元件	-32768~32767	带符号BIN16位	ANY16
(d)	存储逻辑与结果的软元件	-32768~32767	带符号BIN16位	ANY16

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 对(d)中指定的软元件的BIN16位数据与(s)中指定的软元件的BIN16位数据的各个位进行逻辑与运算，将结果存储到(d)中指定的软元件中。

(d)	b15	...	b8	b7	...	b0											
	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	0	1	0	0	0	0	0	0	1	0	0

- 位软元件的情况下，位指定的点数以后的位软元件将作为0进行运算。

出错

没有运算错误。

WAND(P) [操作数为3个的情况下]

进行指定的2个BIN16位数据的逻辑与。

梯形图	ST
	ENO:=WAND(EN, s1, s2, d); ENO:=WANDP(EN, s1, s2, d);

FBD/LD

■执行条件

指令	执行条件
WAND	
WANDP	

设置数据

■内容、范围、数据类型

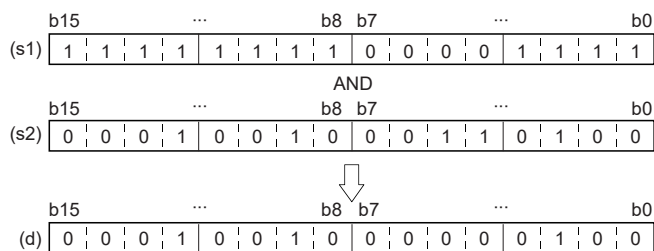
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储进行逻辑与的数据或存储了数据的软元件	-32768~32767	带符号BIN16位	ANY16
(s2)	存储进行逻辑与的数据或存储了数据的软元件	-32768~32767	带符号BIN16位	ANY16
(d)	存储逻辑与结果的软元件	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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	
(s1)	○	○	○	○	○	—	—	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—

功能

- 对(s1)中指定的软元件的BIN16位数据与(s2)中指定的软元件的BIN16位数据的各个位进行逻辑与运算，将结果存储到(d)中指定的软元件中。



- 位软元件的情况下，位指定的点数以后的位软元件将作为0进行运算。

出错

没有运算错误。

32位数据逻辑与

DAND(P) [操作数为2个的情况下]

对指定的2个BIN32位数据进行逻辑与。

梯形图	ST
	不对应。 (☞ 323页 DAND(P) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 323页 DAND(P) [操作数为3个的情况下])

■执行条件

指令	执行条件
DAND	
DANDP	

设置数据

■内容、范围、数据类型

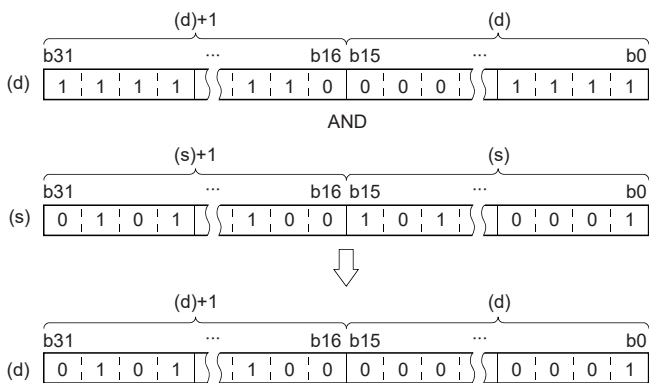
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行逻辑与的数据或存储了数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32
(d)	存储逻辑与结果的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

- 对(d)中指定的软元件的BIN32位数据与(s)中指定的软元件的BIN32位数据的各个位进行逻辑与运算，将结果存储到(d)中指定的软元件中。



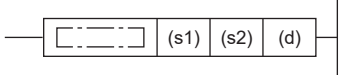
- 位软元件的情况下，位指定的点数以后的位软元件将作为0进行运算。

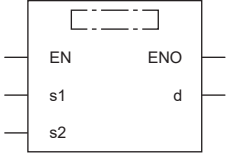
出错

没有运算错误。


DAND(P) [操作数为3个的情况下]

对指定的2个BIN32位数据进行逻辑与。

梯形图	ST
	ENO:=DAND(EN, s1, s2, d); ENO:=DANDP(EN, s1, s2, d);

FBD/LD


■执行条件

指令	执行条件
DAND	
DANDP	

设置数据

■内容、范围、数据类型

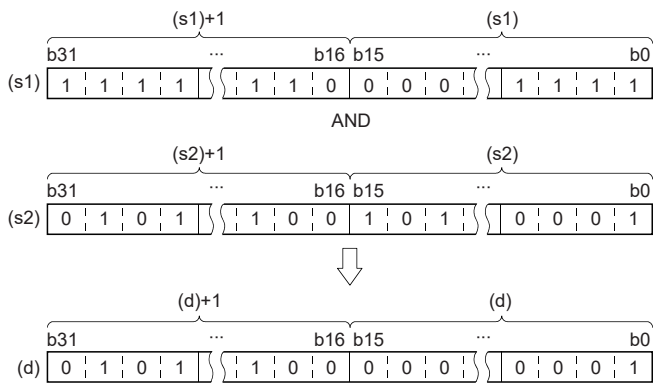
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	进行逻辑与的数据或存储了数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32
(s2)	进行逻辑与的数据或存储了数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32
(d)	存储逻辑与结果的起始软元件	—	带符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	○	○	○	○	—	—	—

功能

- 对(s1)中指定的软元件的BIN32位数据与(s2)中指定的软元件的BIN32位数据的各个位进行逻辑与运算，将结果存储到(d)中指定的软元件中。



- 位软元件的情况下，位指定的点数以后的位软元件将作为0进行运算。

出错

没有运算错误。

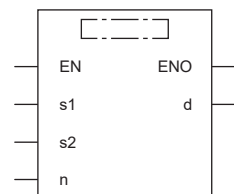
16位块数据逻辑与

BKAND(P)

对指定的2个BIN16位数据块进行逻辑与。

梯形图	ST
	ENO:=BKAND (EN, s1, s2, n, d); ENO:=BKANDP (EN, s1, s2, n, d);

FBD/LD



■执行条件

指令	执行条件
BKAND	
BKANDP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	进行逻辑与的数据或存储了进行逻辑与的数据的起始软元件	—	带符号BIN16位	ANY16*1
(s2)	进行逻辑与的数据或存储了数据的起始软元件	-32768~32767	带符号BIN16位	ANY16*1
(d)	存储逻辑与结果的起始软元件	—	带符号BIN16位	ANY16*1
(n)	运算数据个数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

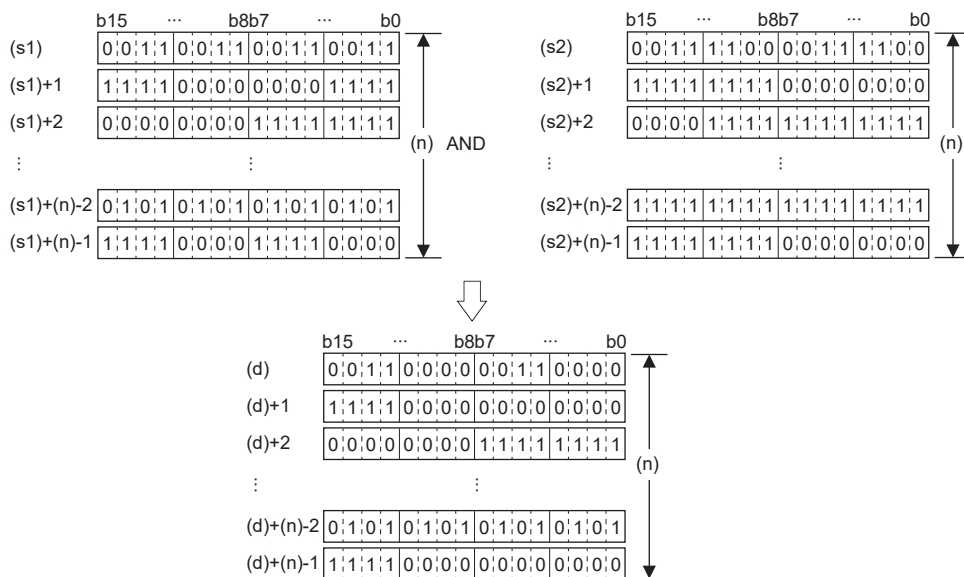
■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	—

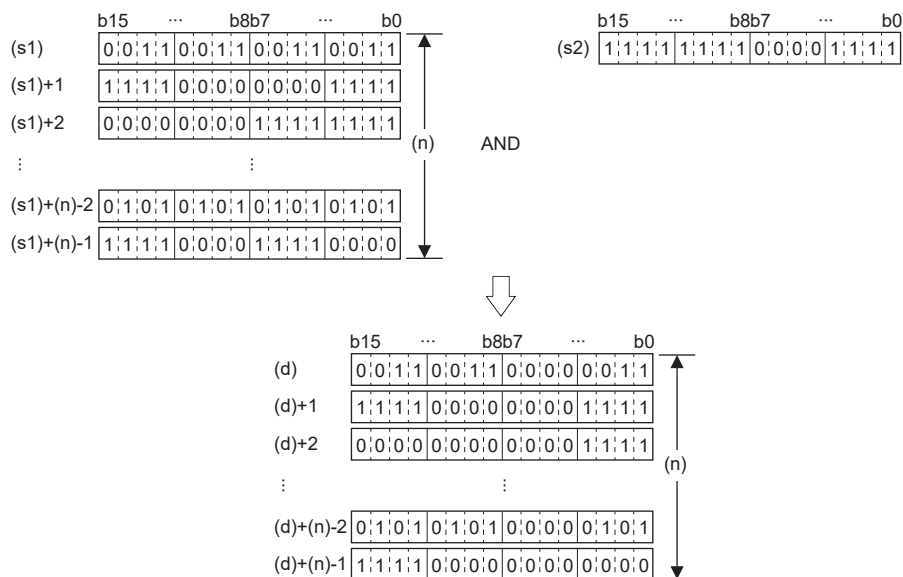
*1 (s1)与(d)或(s2)与(d)可以指定同一软元件编号。

功能

- 对(s1)中指定的软件元件开始的(n)点的内容与(s2)中指定的软件元件开始的(n)点的内容进行逻辑与运算，将结果存储到(d)中指定的软件元件及其以后。



- (s2)中可以指定-32768~32767(带符号BIN16位)的常数。



出错

错误代码(SD0)	内容
2821H	(s1)开始的(n)点的软件元件范围与(d)开始的(n)点的软件元件范围的一部分重复时。 (s1)及(d)中指定了同一软件元件的情况下除外。)
	(s2)开始的(n)点的软件元件范围与(d)开始的(n)点的软件元件范围的一部分重复时。 (s2)及(d)中指定了同一软件元件的情况下除外。)

16位数据逻辑或

WOR(P) [操作数为2个的情况下]

进行指定的2个BIN16位数据的逻辑或。

梯形图	ST
	不对应。 (☞ 328页 WOR(P) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 328页 WOR(P) [操作数为3个的情况下])

■执行条件

指令	执行条件
WOR	
WORP	

设置数据

■内容、范围、数据类型

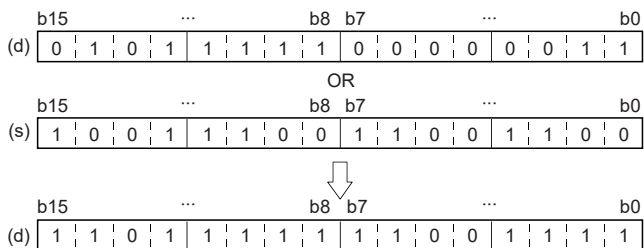
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行逻辑或的数据或存储了数据的软元件	-32768~32767	带符号BIN16位	ANY16
(d)	存储逻辑或结果的软元件	-32768~32767	带符号BIN16位	ANY16

■可用软元件

操作数	位		字		双字		间接指定	常数				其它	
	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)	○	○	○	○	○	—	—	○	—	—	—	—	—

功能

- 对(d)中指定的软元件的BIN16位数据与(s)中指定的软元件的BIN16位数据的各个位进行逻辑或运算，将结果存储到(d)中指定的软元件中。



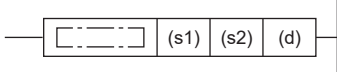
- 位软元件的情况下，位指定的点数以后的位软元件将作为0进行运算。

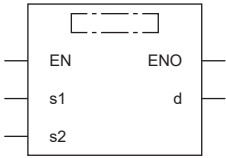
出错

没有运算错误。


WOR(P) [操作数为3个的情况下]

进行指定的2个BIN16位数据的逻辑或。

梯形图	ST
	ENO:=WOR(EN, s1, s2, d); ENO:=WORP(EN, s1, s2, d);

FBD/LD


■执行条件

指令	执行条件
WOR	
WORP	

设置数据

■内容、范围、数据类型

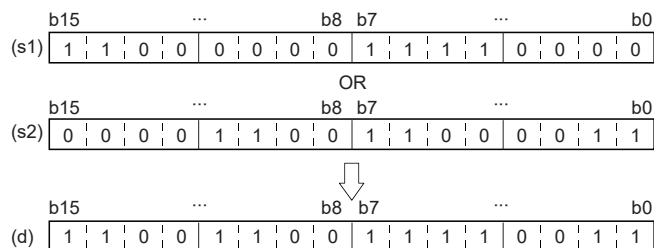
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	进行逻辑或的数据或存储了数据的软元件	-32768~32767	带符号BIN16位	ANY16
(s2)	进行逻辑或的数据或存储了数据的软元件	-32768~32767	带符号BIN16位	ANY16
(d)	存储逻辑或结果的软元件	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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	
(s1)	○	○	○	○	○	—	—	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	—	—	—
(d)	○	○	○	○	○	—	—	—	—	—	—

功能

- 对(s1)中指定的软元件的BIN16位数据与(s2)中指定的软元件的BIN16位数据的各个位进行逻辑或运算，将结果存储到(d)中指定的软元件中。



- 位软元件的情况下，位指定的点数以后的位软元件将作为0进行运算。

出错

没有运算错误。

32位数据逻辑或

DOR(P) [操作数为2个的情况下]

进行指定的2个BIN32位数据的逻辑或。

梯形图	ST
	不对应。 (☞ 331页 DOR(P) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 331页 DOR(P) [操作数为3个的情况下])

■执行条件

指令	执行条件
DOR	
DORP	

设置数据

■内容、范围、数据类型

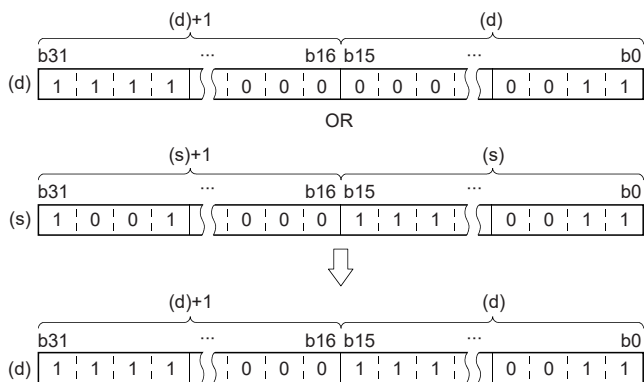
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行逻辑或的数据或存储了数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32
(d)	存储逻辑或结果的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32

■可用软元件

操作数	位		字		双字			间接指定	常数			其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—

功能

- 对(d)中指定的软元件的BIN32位数据与(s)中指定的软元件的BIN32位数据的各个位进行逻辑或运算，将结果存储到(d)中指定的软元件中。



- 位软元件的情况下，位指定的点数以后的位软元件将作为0进行运算。

出错

没有运算错误。

DOR(P) [操作数为3个的情况下]

进行指定的2个BIN32位数据的逻辑或。

梯形图	ST
	ENO:=DOR(EN, s1, s2, d); ENO:=DORP(EN, s1, s2, d);

FBD/LD

■执行条件

指令	执行条件
DOR	
DORP	

设置数据

■内容、范围、数据类型

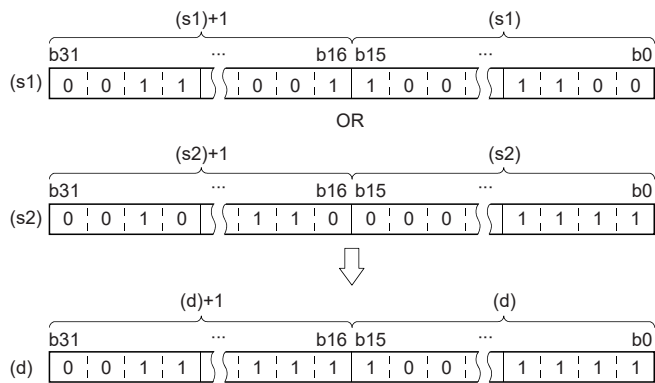
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	进行逻辑或的数据或存储了数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32
(s2)	进行逻辑或的数据或存储了数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32
(d)	存储逻辑或结果的起始软元件	—	带符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

- 对(s1)中指定的软元件的BIN32位数据与(s2)中指定的软元件的BIN32位数据的各个位进行逻辑或运算，将结果存储到(d)中指定的软元件中。



- 位软元件的情况下，位指定的点数以后的位软元件将作为0进行运算。

出错

没有运算错误。

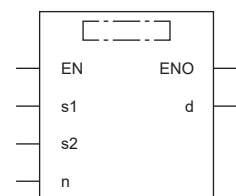
16位块数据逻辑或

BKOR (P)

进行指定的2个BIN16位数据块的逻辑或。

梯形图	ST
	ENO:=BKOR (EN, s1, s2, n, d); ENO:=BKORP (EN, s1, s2, n, d);

FBD/LD



■执行条件

指令	执行条件
BKOR	
BKORP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	进行逻辑或的数据或存储了进行逻辑或的数据的起始软元件	—	带符号BIN16位	ANY16*1
(s2)	进行逻辑或的数据或存储了数据的起始软元件	-32768~32767	带符号BIN16位	ANY16*1
(d)	存储逻辑或结果的起始软元件	—	带符号BIN16位	ANY16*1
(n)	运算数据个数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

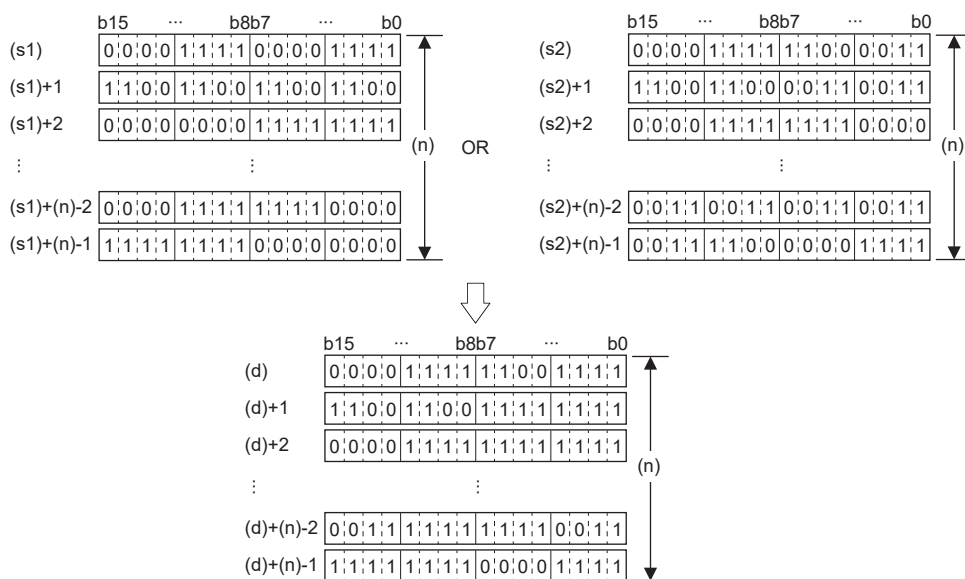
■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	

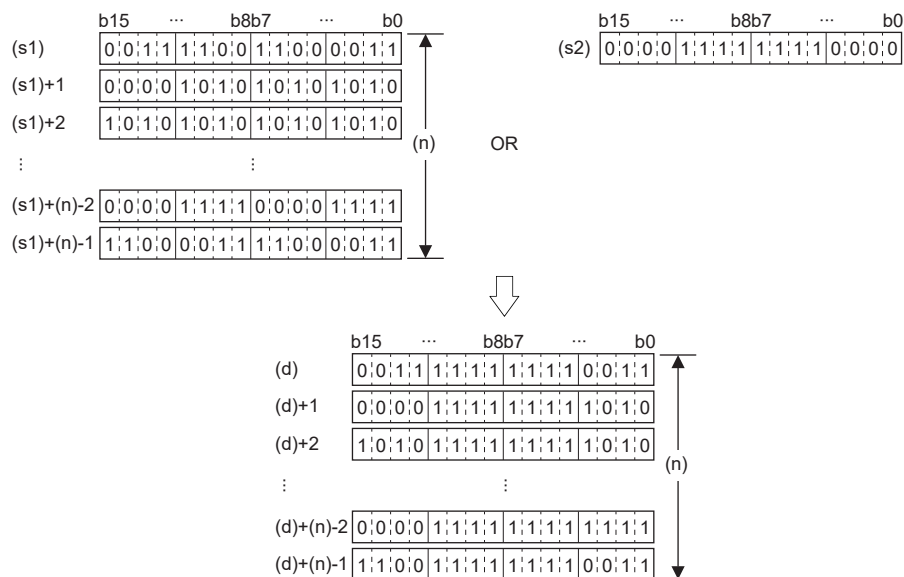
*1 (s1)与(d)或(s2)与(d)可以指定同一软元件编号。

功能

- 对(s1)中指定的软元件开始的(n)点的内容与(s2)中指定的软元件开始的(n)点的内容进行逻辑或运算，将结果存储到(d)中指定的软元件及其以后。



- (s2)中可以指定-32768~32767(带符号BIN16位)的常数。



出错

错误代码(SD0)	内容
2821H	(s1)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围的一部分重复时。 (s1)及(d)中指定了同一软元件的情况下除外。)
	(s2)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围的一部分重复时。 (s2)及(d)中指定了同一软元件的情况下除外。)

16位数据异或

WXOR(P) [操作数为2个的情况下]

进行指定的2个BIN16位数据的异或。

梯形图	ST
	不对应。 (☞ 336页 WXOR(P) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 336页 WXOR(P) [操作数为3个的情况下])

■执行条件

指令	执行条件
WXOR	
WXORP	

设置数据

■内容、范围、数据类型

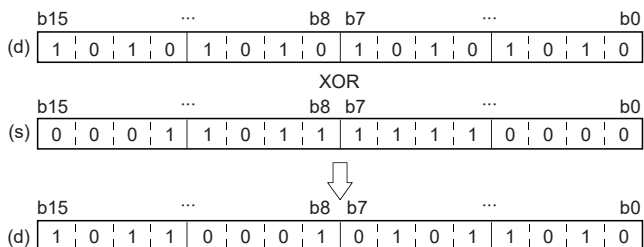
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行异或的数据或者存储了数据的软元件	-32768~32767	带符号BIN16位	ANY16
(d)	存储异或的结果的软元件	-32768~32767	带符号BIN16位	ANY16

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—	—

功能

- 对(d)中指定的软元件的BIN16位数据与(s)中指定的软元件的BIN16位数据的各个位进行异或运算，将结果存储到(d)中指定的软元件中。



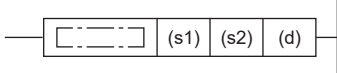
- 位软元件的情况下，位指定的点数以后的位软元件将作为0进行运算。

出错

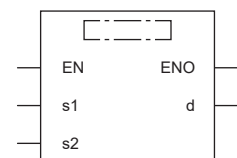
没有运算错误。

WXOR(P) [操作数为3个的情况下]

进行指定的2个BIN16位数据的异或。

梯形图	ST
	ENO:=WXOR(EN, s1, s2, d); ENO:=WXORP(EN, s1, s2, d);

FBD/LD



■执行条件

指令	执行条件
WXOR	
WXORP	

设置数据

■内容、范围、数据类型

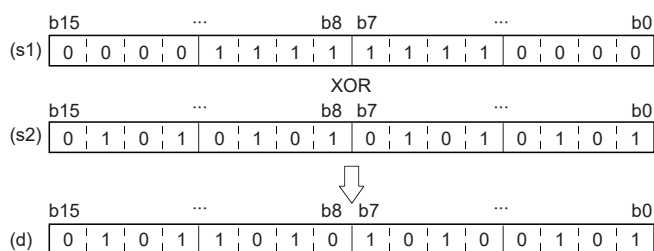
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	进行异或的数据或者存储了数据的软元件	-32768~32767	带符号BIN16位	ANY16
(s2)	进行异或的数据或者存储了数据的软元件	-32768~32767	带符号BIN16位	ANY16
(d)	存储异或的结果的软元件	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—	—

功能

- 对(s1)中指定的软元件的BIN16位数据与(s2)中指定的软元件的BIN16位数据的各个位进行异或运算，将结果存储到(d)中指定的软元件中。



- 位软元件的情况下，位指定的点数以后的位软元件将作为0进行运算。

出错

没有运算错误。

32位数据异或

DXOR(P) [操作数为2个的情况下]

进行指定的2个BIN32位数据的异或。

梯形图	ST
	不对应。 (☞ 339页 DXOR(P) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 339页 DXOR(P) [操作数为3个的情况下])

■执行条件

指令	执行条件
DXOR	
DXORP	

设置数据

■内容、范围、数据类型

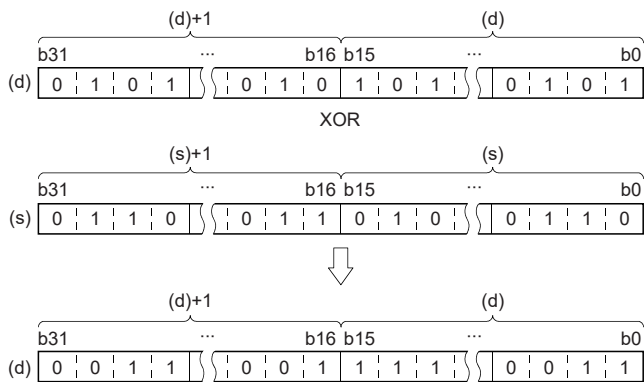
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行异或的数据或者存储了数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32
(d)	存储异或的结果的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

- 对(d)中指定的软元件的BIN32位数据与(s)中指定的软元件的BIN32位数据的各个位进行异或运算，将结果存储到(d)中指定的软元件中。



- 位软元件的情况下，位指定的点数以后的位软元件将作为0进行运算。

出错

没有运算错误。

DXOR(P) [操作数为3个的情况下]

进行指定的2个BIN32位数据的异或。

梯形图	ST
	ENO:=DXOR(EN, s1, s2, d); ENO:=DXORP(EN, s1, s2, d);

FBD/LD

■执行条件

指令	执行条件
DXOR	
DXORP	

设置数据

■内容、范围、数据类型

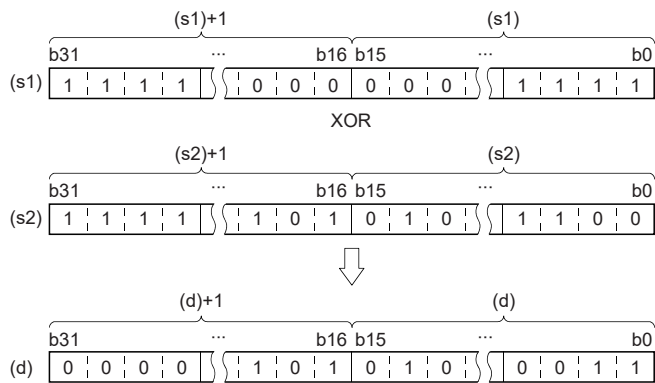
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	进行异或的数据或者存储了数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32
(s2)	进行异或的数据或者存储了数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32
(d)	存储异或的结果的起始软元件	—	带符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

- 对(s1)中指定的软元件的BIN32位数据与(s2)中指定的软元件的BIN32位数据的各个位进行异或运算，将结果存储到(d)中指定的软元件中。



- 位软元件的情况下，位指定的点数以后的位软元件将作为0进行运算。

出错

没有运算错误。

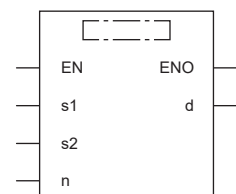
16位块数据异或

BKXOR(P)

进行指定的2个BIN16位数据块的异或。

梯形图	ST
	ENO:=BKXOR(EN, s1, s2, n, d); ENO:=BKXORP(EN, s1, s2, n, d);

FBD/LD



■执行条件

指令	执行条件
BKXOR	
BKXORP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储逻辑运算数据的起始软元件	—	带符号BIN16位	ANY16*1
(s2)	逻辑运算数据或者存储了逻辑运算数据的起始软元件	-32768~32767	带符号BIN16位	ANY16*1
(d)	存储运算结果的起始软元件	—	带符号BIN16位	ANY16*1
(n)	运算数据个数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

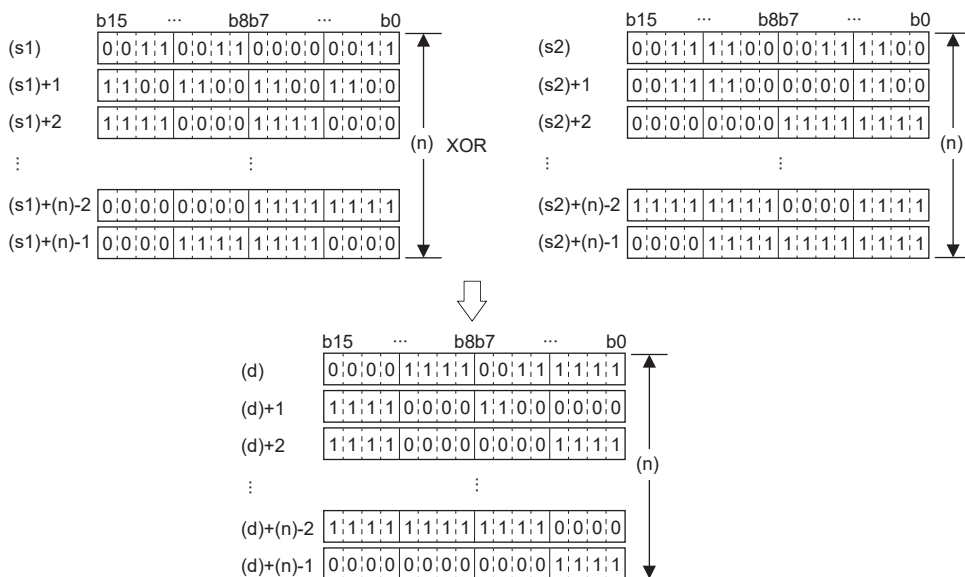
■可用软元件

操作数	位		字		双字			间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

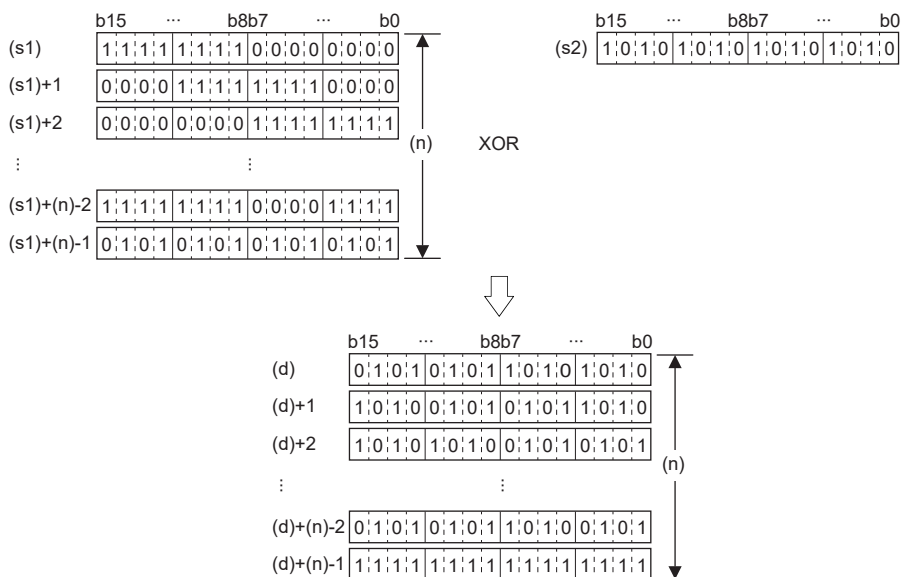
*1 (s1)与(d)或(s2)与(d)可以指定同一软元件编号。

功能

- 对(s1)中指定的软元件开始的(n)点的内容与(s2)中指定的软元件开始的(n)点的内容进行异或运算，将结果存储到(d)中指定的软元件及其以后。



- (s2)中可以指定-32768~32767(带符号BIN16位)的常数。



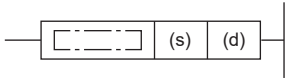
出错

错误代码(SD0)	内容
2821H	(s1)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围的一部分重复时。 (s1)及(d)中指定了同一软元件的情况下除外。)
	(s2)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围的一部分重复时。 (s2)及(d)中指定了同一软元件的情况下除外。)



16位数据异或非

WXNR(P) [操作数为2个的情况下]

进行指定的2个BIN16位数据的异或非。

梯形图	ST
	不对应。 (☞ 344页 WXNR(P) [操作数为3个的情况下])
FBD/LD	
不对应。 (☞ 344页 WXNR(P) [操作数为3个的情况下])	

■执行条件

指令	执行条件
WXNR	
WXNRP	

设置数据

■内容、范围、数据类型

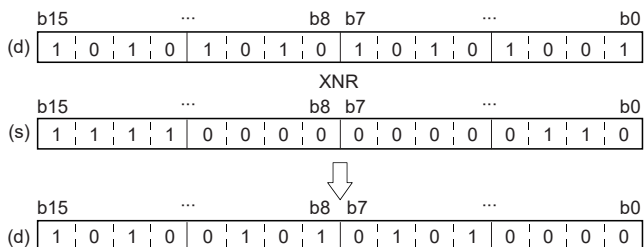
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行异或非的数据或者存储了数据的软元件	-32768~32767	带符号BIN16位	ANY16
(d)	存储异或非的结果的软元件	-32768~32767	带符号BIN16位	ANY16

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—	

功能

- 对(d)中指定的软元件的BIN16位数据与(s)中指定的软元件的BIN16位数据进行异或非运算，将结果存储到(d)中指定的软元件中。



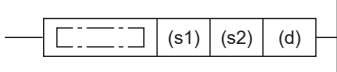
- 位软元件的情况下，位指定的点数以后的位软元件将作为0进行运算。

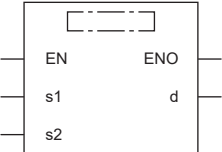
出错

没有运算错误。

WXNR(P) [操作数为3个的情况下]

进行指定的2个BIN16位数据的异或非。

梯形图	ST
	ENO:=WXNR(EN, s1, s2, d); ENO:=WXNRP(EN, s1, s2, d);

FBD/LD


■执行条件

指令	执行条件
WXNR	
WXNRP	

设置数据

■内容、范围、数据类型

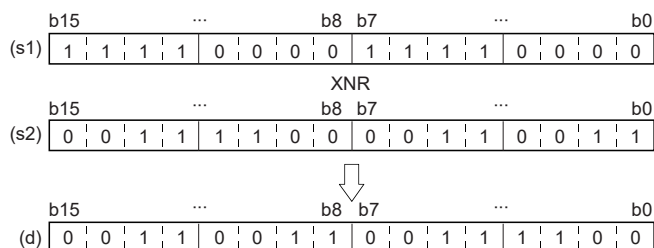
操作数	内容	范围	数据类型	数据类型(标签)
(s1)、(s2)	进行异或非运算的数据或者存储了数据的软元件	-32768~32767	带符号BIN16位	ANY16
(d)	存储异或非的结果的软元件	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 对(s1)中指定的软元件的BIN16位数据与(s2)中指定的软元件的BIN16位数据进行异或非运算，将结果存储到(d)中指定的软元件中。



- 位软元件的情况下，位指定的点数以后的位软元件将作为0进行运算。

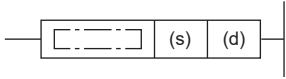
出错

没有运算错误。


32位数据异或非

DXNR(P) [操作数为2个的情况下]

进行指定的2个BIN32位数据的异或非。

梯形图	ST
	不对应。 (☞ 346页 DXNR(P) [操作数为3个的情况下])
FBD/LD	
不对应。 (☞ 346页 DXNR(P) [操作数为3个的情况下])	

■执行条件

指令	执行条件
DXNR	
DXNRP	

设置数据

■内容、范围、数据类型

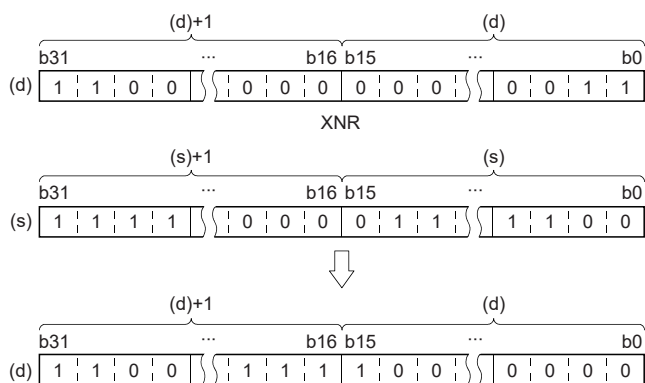
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行异或非的数据或者存储了数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32
(d)	存储异或非的结果的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	○	○	—	—	—	—	—

功能

- 对(d)中指定的软元件的BIN32位数据与(s)中指定的软元件的BIN32位数据进行异或非运算，将结果存储到(d)中指定的软元件中。



- 位软元件的情况下，位指定的点数以后的位软元件将作为0进行运算。

出错

没有运算错误。

DXNR (P) [操作数为3个的情况下]

进行指定的2个BIN32位数据的异或非。

梯形图	ST
	ENO:=DXNR (EN, s1, s2, d); ENO:=DXNRP (EN, s1, s2, d);

FBD/LD

■执行条件

指令	执行条件
DXNR	
DXNRP	

设置数据

■内容、范围、数据类型

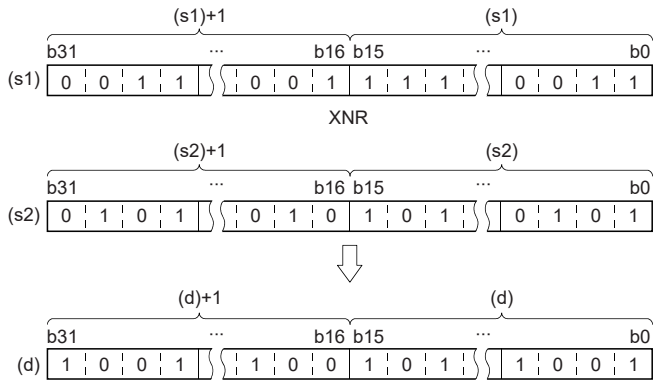
操作数	内容	范围	数据类型	数据类型(标签)
(s1)、(s2)	进行异或非运算的数据或者存储了数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32
(d)	存储异或非的结果的起始软元件	—	带符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	○	○	—	—	—	—	—

功能

- 对(s1)中指定的软元件的BIN32位数据与(s2)中指定的软元件的BIN32位数据进行异或非运算，将结果存储到(d)中指定的软元件中。



- 位软元件的情况下，位指定的点数以后的位软元件将作为0进行运算。

出错

没有运算错误。

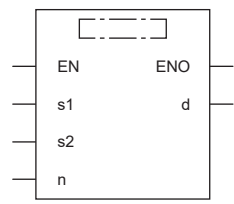
16位块数据异或非

BKXNR (P)

进行指定的2个BIN16位数据块的异或非。

梯形图	ST
	ENO:=BKXNR (EN, s1, s2, n, d) ; ENO:=BKXNRP (EN, s1, s2, n, d) ;

FBD/LD



■执行条件

指令	执行条件
BKXNR	
BKXNRP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储逻辑运算数据的起始软元件	—	带符号BIN16位	ANY16*1
(s2)	逻辑运算数据或者存储了逻辑运算数据的起始软元件	-32768~32767	带符号BIN16位	ANY16*1
(d)	存储运算结果的起始软元件	—	带符号BIN16位	ANY16*1
(n)	运算数据个数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

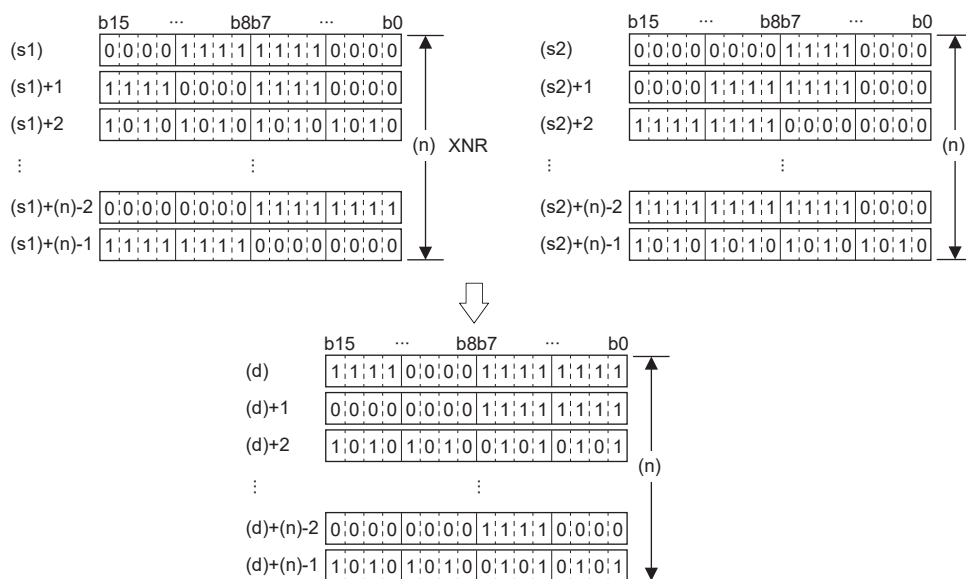
■可用软元件

操作数	位		字		双字			间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

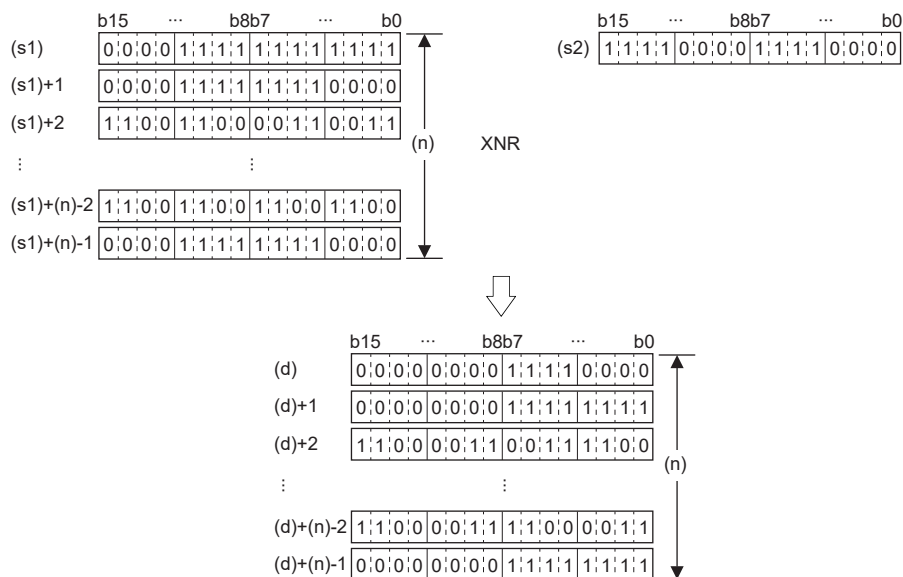
*1 (s1)与(d)或(s2)与(d)可以指定同一软元件编号。

功能

- 对(s1)中指定的软元件开始的(n)点的内容与(s2)中指定的软元件开始的(n)点的内容进行异或非运算，将结果存储到(d)中指定的软元件及其以后。



- (s2)中可以指定-32768~32767(带符号BIN16位)的常数。



出错

错误代码(SD0)	内容
2821H	(s1)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围的一部分重复时。 (s1)及(d)中指定了同一软元件的情况下除外。)
	(s2)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围的一部分重复时。 (s2)及(d)中指定了同一软元件的情况下除外。)

12.4 位处理指令

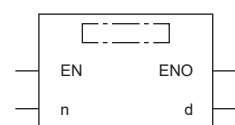
字软元件的位设置

BSET (P)

对指定的字软元件内的第n位进行设置(1)。

梯形图	ST
	ENO:=BSET (EN, n, d) ; ENO:=BSETP (EN, n, d) ;

FBD/LD



■执行条件

指令	执行条件
BSET	
BSETP	

设置数据

■内容、范围、数据类型

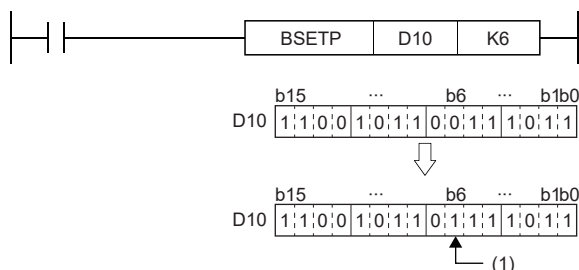
操作数	内容	范围	数据类型	数据类型(标签)
(d)	进行位设置的软元件	—	带符号BIN16位	ANY16
(n)	进行位设置的位的位置	0~15	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 对(d)中指定的字软元件的第(n)位进行设置(1)。



(1) 对D10的第6位进行设置(1)。

- (n)中超过了15的情况下,以低位4位的数据执行。

出错

没有运算错误。

字软元件的位复位

BRST (P)

对指定的字软元件内的第n位进行复位(0)。

梯形图	ST
	ENO:=BRST (EN, n, d) ; ENO:=BRSTP (EN, n, d) ;

FBD/LD

■执行条件

指令	执行条件
BRST	
BRSTP	

设置数据

■内容、范围、数据类型

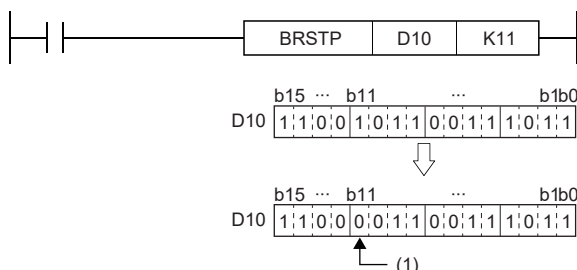
操作数	内容	范围	数据类型	数据类型(标签)
(d)	进行位复位的软元件	—	带符号BIN16位	ANY16
(n)	进行位复位的位的位置	0~15	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

- 对(d)中指定的字软元件的第(n)位进行复位(0)。



- (1) 对D10的第11位进行设置(0)。
- (n)中超过了15的情况下,以低位4位的数据执行。

出错

没有运算错误。

TEST (P)

对指定的字软元件内的第n位进行提取。

梯形图	ST
	ENO:=TEST (EN, s1, s2, d) ; ENO:=TESTP (EN, s1, s2, d) ;

FBD/LD

■执行条件

指令	执行条件
TEST	
TESTP	

设置数据

■内容、范围、数据类型

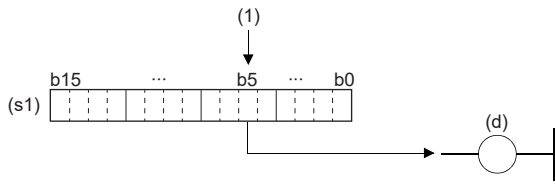
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储了要提取的位数据的软元件编号	—	带符号BIN16位	ANY16
(s2)	要提取的位数据的位置	0~15	无符号BIN16位	ANY16
(d)	存储已提取的位数据的位软元件编号	—	位	ANY_BOOL
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数			其它
	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)	○	○	○	○	—	—	—	○	—	—	—	—

功能

- 从(s1)中指定的字软元件开始，提取(s2)中指定的位置的位数据后，写入到(d)中指定的位软元件中。



(1) (s2)位((s2)=5的情况下)

- (d)中指定的位软元件，在相应位为“0”时变为OFF，为“1”时变为ON。
- 在(s2)中，指定1字数据的各个位位置(0~15)。(s2)中指定了16以上的情况下， $(s2) \div 16$ 的余数值将变为位位置。

例

(s2)=18时， $18 \div 16 = 1$ 余2，因此变为b2的数据。

出错

没有运算错误。

DTEST (P)

对指定的2字软件内的第n位进行提取。

梯形图	ST
	ENO:=DTEST (EN, s1, s2, d) ; ENO:=DTESTP (EN, s1, s2, d) ;

FBD/LD

■执行条件

指令	执行条件
DTEST	
DTESTP	

设置数据

■内容、范围、数据类型

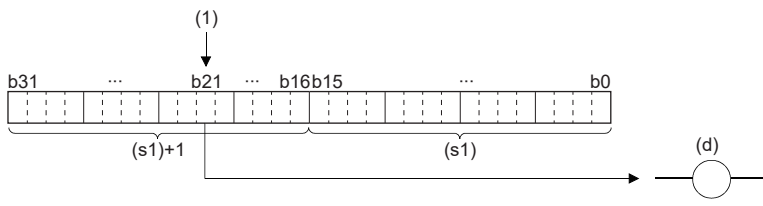
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储了要提取的位数据的软元件编号	—	带符号BIN32位	ANY32
(s2)	要提取的位数据的位置	0~31	无符号BIN16位	ANY16
(d)	存储已提取的位数据的位软元件编号	—	位	ANY_BOOL
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数			其它
	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)	○	○	○	○	—	—	—	○	—	—	—	—

功能

- 从(s1)中指定的双字软元件内，提取(s2)中指定的位置的位数据后，写入到(d)中指定的位软元件中。



(1) (s2)位((s2)=21的情况下)

- (d)中指定的位软元件，在相应位为“0”时变为OFF，为“1”时变为ON。
- 在(s2)中，指定2字数据的各个位位置(0~31)。(s2)中指定了32以上的情况下， $(s2) \div 32$ 的余数值将变为位位置。

例

(s2)=34时， $34 \div 32 = 1$ 余2，因此变为b2的数据。

出错

没有运算错误。

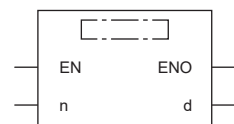
位软元件的批量复位

BKRST (P)

对从指定的位软元件开始的n点的位软元件进行复位。

梯形图	ST
	ENO:=BKRST (EN, n, d) ; ENO:=BKRSTP (EN, n, d) ;

FBD/LD



■执行条件

指令	执行条件
BKRST	
BKRSTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	进行复位的起始软元件	—	位	ANY_BOOL
(n)	进行复位的软元件数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 从(d)中指定的位软元件开始，对(n)点的位软元件进行复位。
- 位软元件的复位状态如下所示。

软元件编号	状态
报警器(F)	<ul style="list-style-type: none"> • 将从(d)中指定的报警器(F)编号开始的(n)点置为OFF。 • 将变为OFF的报警器编号从SD64~SD79中删除后，向前填充对齐。 • 将SD64~SD79中存储的报警器数存储到SD63中。
定时器(T)、累计定时器(ST)、计数器(C)、长定时器(LT)、长累计定时器(LST)、长计数器(LC)	<ul style="list-style-type: none"> • 从(d)中指定的定时器(T)、累计定时器(ST)、计数器(C)、长定时器(LT)、长累计定时器(LST)、长计数器(LC)编号中，将(n)点的当前值置为0，将线圈触点置为OFF。
上述以外的位软元件	<ul style="list-style-type: none"> • 将(d)中指定的软元件开始的(n)点的线圈、触点置为OFF。

- 指定的软元件为OFF的情况下，软元件的状态不变化。

出错


没有运算错误。

12.5 数据移位指令

BIN16位数据的n位右移

SFR(P)



将指定的软元件的BIN16位数据向右移动(n)位。移位后的空余数据中存储0。

梯形图	ST
	ENO:=SFR(EN, n, d); ENO:=SFRP(EN, n, d);

FBD/LD



■执行条件

指令	执行条件
SFR	
SFRP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	移位的软元件	—	带符号BIN16位	ANY16
(n)	移位数	0~15	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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	○	○	○	○	—	—	○	○	—	—	—

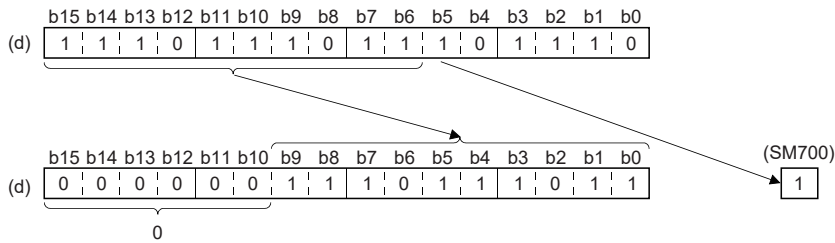
*1 FX、FY不能使用。

功能

- 将(d)中指定的软元件的BIN16位数据向右移动(n)位。
- 从最高位开始的(n)位将变为0。
- 存储移位范围右一个的1位至SM700(进位标志)中。

例

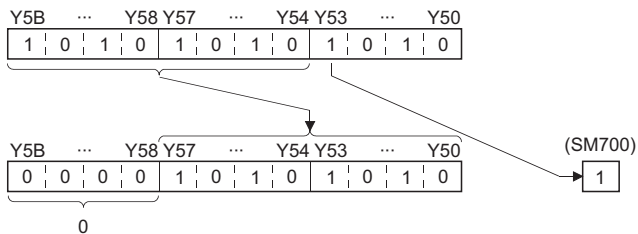
(n)=6的情况下



- (d)中指定了位软元件的情况下，将以位指定中指定的软元件范围进行右移。

例

(n)=4的情况下



- 实际移位的位数为 $(n) \div (\text{位指定中指定的点数})$ 的余数。例如， $(n)=15$ ，(位指定中指定的点数)=8位时， $15 \div 8=1$ 余7，因此移动7位。
- (n) 的指定范围为0~15。 (n) 中指定了16或其以上的值的情况下，将以 $(n) \div 16$ 的余数值进行右移。例如， $(n)=18$ 时， $18 \div 16=1$ 余2，因此右移2位。

出错

没有运算错误。

BIN16位数据的n位左移

SFL (P)

将指定的软元件的BIN16位数据向左移动(n)位。移位后的空余数据中存储0。

梯形图	ST
	ENO:=SFL (EN, n, d); ENO:=SFLP (EN, n, d);

FBD/LD

■执行条件

指令	执行条件
SFL	
SFLP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	移位的软元件	—	带符号BIN16位	ANY16
(n)	移位数	0~15	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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	○	○	○	○	—	—	○	○	—	—	—

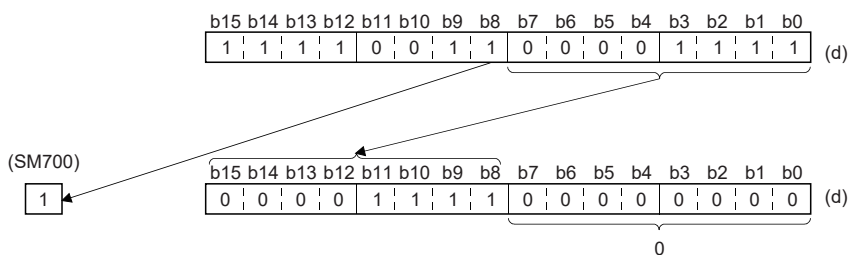
*1 FX、FY不能使用。

功能

- 将(d)中指定的软元件的BIN16位数据向左移动(n)位。
- 从最低位开始的(n)位将变为0。
- 存储移位范围左一个的1位至SM700(进位标志)中。

例

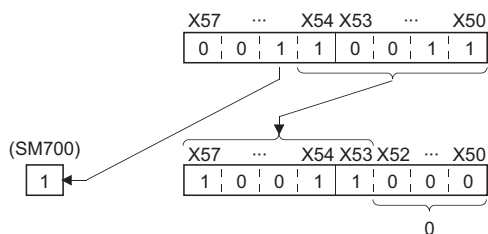
(n)=8的情况下



- (d)中指定了位软元件的情况下，将以位指定中指定的软元件范围进行左移。

例

(n)=5的情况下



- 实际移位的位数为(n)÷(位指定中指定的点数)的余数。例如，(n)=15，(位指定中指定的点数)=8位时， $15 \div 8 = 1$ 余7，因此移动7位。
- (n)的指定范围为0~15。(n)中指定了16或其以上的值的情况下，将以(n)÷16的余数值进行左移。例如，(n)=18时， $18 \div 16 = 1$ 余2，因此向左移动2位。

出错

没有运算错误。

n位数据的1位右移

BSFR (P)

将从指定的软元件开始的(n)点向右移动1位。移位后的空余数据中存储0。

梯形图	ST
	ENO:=BSFR (EN, n, d) ; ENO:=BSFRP (EN, n, d) ;

FBD/LD

■执行条件

指令	执行条件
BSFR	
BSFRP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	移位的软元件	—	位	ANY_BOOL
(n)	移位的数据的数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

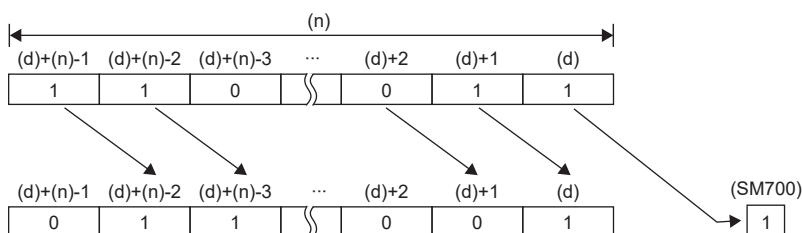
操作数	位		字		双字		间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K	H	
(d)	○*1	—	○	—	—	—	○	—	—	—	—
(n)	○*2	○	○	○	○	—	○	○	—	—	—

*1 T、C、ST不能使用。

*2 FX、FY不能使用。

功能

- 将从(d)中指定的软元件开始的(n)点向右移动1位。
- 从最高位开始的1位将变为0。
- 存储移位范围右一个的1位至SM700(进位标志)中。



- (n)中指定的值为0时变为无处理。

出错

没有运算错误。

n位数据的1位左移

BSFL (P)

将从指定的软元件开始的(n)点向左移动1位。移位后的空余数据中存储0。

梯形图	ST
	ENO:=BSFL(EN, n, d); ENO:=BSFLP(EN, n, d);

FBD/LD

■执行条件

指令	执行条件
BSFL	
BSFLP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	移位的软元件	—	位	ANY_BOOL
(n)	移位的数据的数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

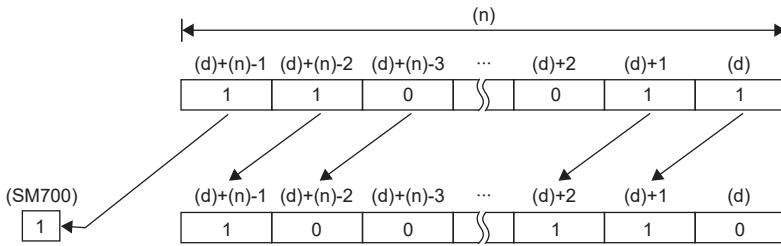
操作数	位		字		双字		间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K、H	E	
(d)	○*1	—	○	—	—	—	○	—	—	—	—
(n)	○*2	○	○	○	○	—	○	○	—	—	—

*1 T、C、ST不能使用。

*2 FX、FY不能使用。

功能

- 将从(d)中指定的软元件开始的(n)点向左移动1位。
- 从最低位开始的1位将变为0。
- 存储移位范围左一个的1位至SM700(进位标志)中。



- (n)中指定的值为0时变为无处理。

出错

没有运算错误。

n字数据的1字右移

DSFR (P)

将从指定的软元件开始的(n)点向右移动1字。移位后的空余数据中存储0。

梯形图	ST
	ENO:=DSFR (EN, n, d) ; ENO:=DSFRP (EN, n, d) ;

FBD/LD

■执行条件

指令	执行条件
DSFR	
DSFRP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	移位的软元件	—	字	ANY16
(n)	移位的数据的数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

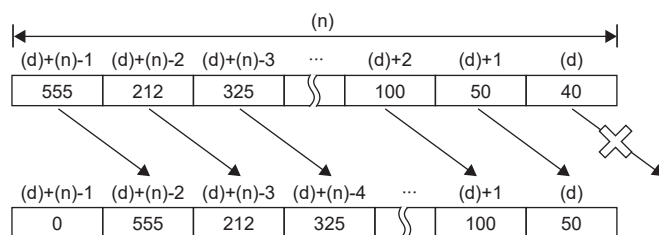
■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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不能使用。

功能

- 将(d)中指定的软元件开始的(n)点的数据向右移动1字。
- 从最高位开始的1字将变为0。



- (n)中指定的值为0时变为无处理。

出错

没有运算错误。

n字数据的1字左移

DSFL (P)

将指定的软元件开始的 (n) 点向左移动1字。移位后的空余数据中存储0。

梯形图	ST
	ENO:=DSFL (EN, n, d) ; ENO:=DSFLP (EN, n, d) ;

FBD/LD

■执行条件

指令	执行条件
DSFL	
DSFLP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	移位的软元件	—	字	ANY16
(n)	移位的数据的数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

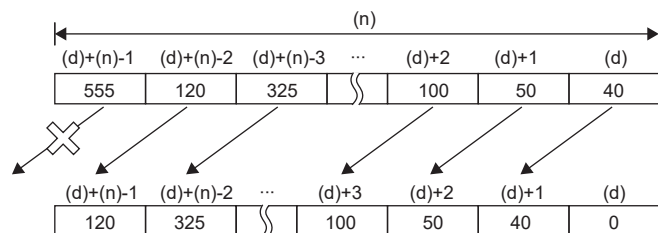
■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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不能使用。

功能

- 将 (d) 中指定的软元件开始的 (n) 点的数据向左移动1字。
- 从最低位开始的1字将变为0。



- (n) 中指定的值为0的情况下变为无处理。

出错

没有运算错误。

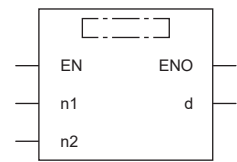
n位数据的n位右移

SFTBR (P)

在指定的软元件开始的(n1)位的数据范围内，向右移动(n2)位。移位后的空余数据中存储0。

梯形图	ST
	ENO:=SFTBR(EN, n1, n2, d); ENO:=SFTBRP(EN, n1, n2, d);

FBD/LD



■执行条件

指令	执行条件
SFTBR	
SFTBRP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	移位的软元件	—	位	ANY_BOOL
(n1)	移位的数据的数	0~64	无符号BIN16位	ANY16
(n2)	移位数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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不能使用。

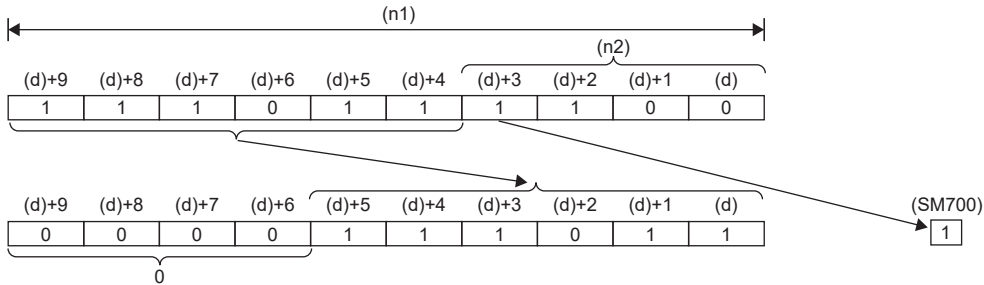
*2 FX、FY不能使用。

功能

- 在(d)中指定的软元件开始的(n1)位的数据范围内，向右移动(n2)位。
- 存储移位范围右一个的1位至SM700(进位标志)中。

例

(n1)=10, (n2)=4的情况下



- (n1)、(n2)的指定应满足 $(n1) > (n2)$ 。 $(n1) \leq (n2)$ 的情况下，将以 $(n2) \div (n1)$ 的余数值进行移位。但是， $(n2) \div (n1)$ 的余数值为0时，将变为无处理。
- (n1)的设置范围为1~64。
- 从最高位开始的(n2)位将变为0。 $(n1) < (n2)$ 的情况下， $(n2) \div (n1)$ 的余数值部分将变为0。
- (n1)或(n2)中指定的值为0时，将变为无处理。

出错

错误代码(SD0)	内容
3285H	(n1)超出0~64的范围时。

n位数据的n位左移

SFTBL (P)

在指定的软元件开始的(n1)位的数据范围内，向左移动(n2)位。移位后的空余数据中存储0。

梯形图	ST
	ENO:=SFTBL (EN, n1, n2, d) ; ENO:=SFTBLP (EN, n1, n2, d) ;

FBD/LD

■执行条件

指令	执行条件
SFTBL	
SFTBLP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	移位的软元件	—	位	ANY_BOOL
(n1)	移位的数据的数	0~64	无符号BIN16位	ANY16
(n2)	移位数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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不能使用。

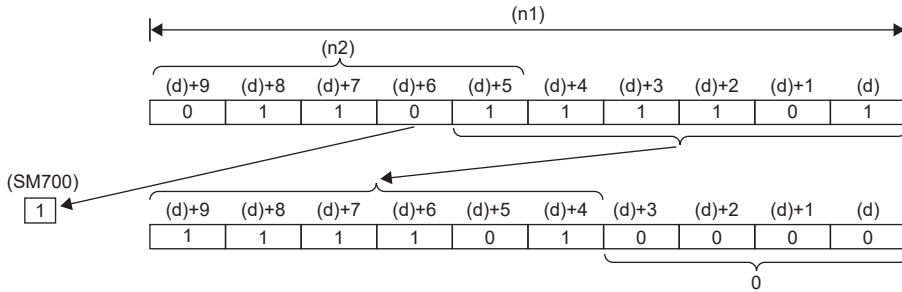
*2 FX、FY不能使用。

功能

- 在(d)中指定的软元件开始的(n1)位的数据范围内，向左移动(n2)位。
- 存储移位范围左一个的1位至SM700(进位标志)中。

例

(n1)=10, (n2)=4的情况下



- (n1)、(n2)的指定应满足 $(n1) > (n2)$ 。 $(n1) \leq (n2)$ 的情况下，将以 $(n2) \div (n1)$ 的余数值进行移位。但是， $(n2) \div (n1)$ 的余数值为0时，将变为无处理。
- (n1)的设置范围为1~64。
- 从最低位开始的(n2)位将变为0。 $(n1) < (n2)$ 的情况下， $(n2) \div (n1)$ 的余数值部分将变为0。
- (n1)或(n2)中指定的值为0时，将变为无处理。

出错

错误代码(SD0)	内容
3285H	(n1)超出0~64的范围时。

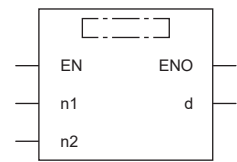
n字数据的n字右移

SFTWR(P)

在指定的软元件开始的(n1)字的数据范围内，向右移动(n2)字。移位后的空余数据中存储0。

梯形图	ST
	ENO:=SFTWR(EN, n1, n2, d); ENO:=SFTWRP(EN, n1, n2, d);

FBD/LD



■执行条件

指令	执行条件
SFTWR	
SFTWRP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	移位的软元件	—	字	ANY16
(n1)	移位的数据的数	0~65535	无符号BIN16位	ANY16
(n2)	移位数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—
(n1)	○*1	○	○	○	○	—	—	○	○	—	—	—
(n2)	○*1	○	○	○	○	—	—	○	○	—	—	—

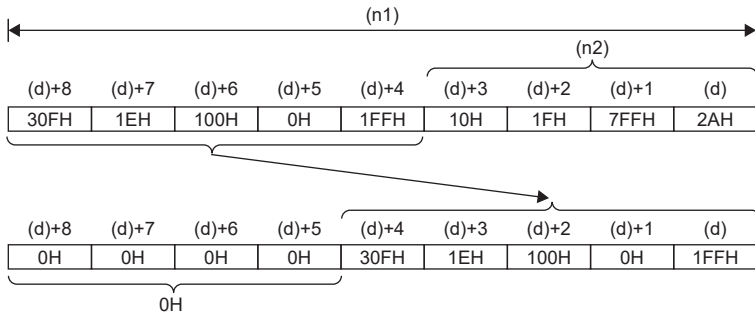
*1 FX、FY不能使用。

功能

- 在(d)中指定的软元件开始的(n1)字的数据范围内，向右移动(n2)字。
- 从最高位开始的(n2)字将变为0H。
- (n1)或(n2)中指定的值为0H时，将变为无处理。
- (n1)≤(n2)的情况下，从(d)中指定的软元件开始的(n1)字的数据将全部变为0H。

例

(n1)=9, (n2)=4的情况下



出错

没有运算错误。

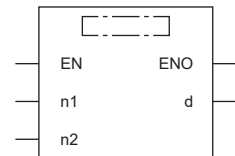
n字数据的n字左移

SFTWL (P)

在从指定的软元件开始的(n1)字的数据范围内，左移(n2)字。移位后的空余数据中存储0。

梯形图	ST
	ENO:=SFTWL (EN, n1, n2, d) ; ENO:=SFTWLP (EN, n1, n2, d) ;

FBD/LD



■执行条件

指令	执行条件
SFTWL	
SFTWLP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	移位的软元件	—	字	ANY16
(n1)	移位的数据的数	0~65535	无符号BIN16位	ANY16
(n2)	移位数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—
(n1)	○*1	○	○	○	○	—	—	○	○	—	—	—
(n2)	○*1	○	○	○	○	—	—	○	○	—	—	—

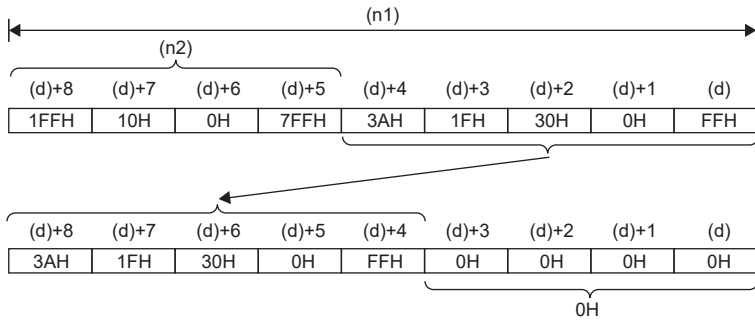
*1 FX、FY不能使用。

功能

- 在(d)中指定的软元件开始的(n1)字的数据范围内，向左移动(n2)字。
- 从最低位开始的(n2)字将变为0H。
- (n1)或(n2)中指定的值为0H时，将变为无处理。
- (n1)≤(n2)的情况下，从(d)中指定的软元件开始的(n1)字的数据将全部变为0H。

例

(n1)=9, (n2)=4的情况下



出错

没有运算错误。

12.6 数据转换指令

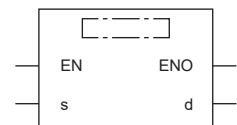
BIN数据→BCD4位转换

BCD(P)

将指定的BIN16位数据转换为BCD4位数据。

梯形图	ST
	ENO:=BCD(EN, s, d); ENO:=BCDP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
BCD	
BCDP	

设置数据

■内容、范围、数据类型

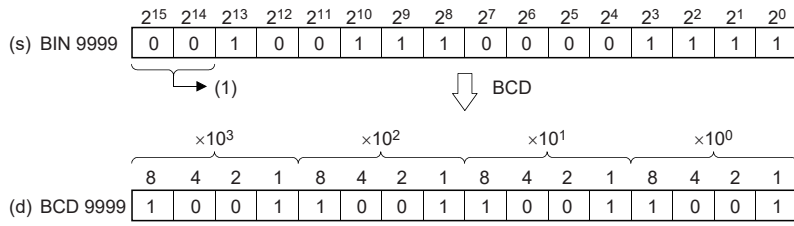
操作数	内容	范围	数据类型	数据类型(标签)
(s)	BIN数据或存储了BIN数据的软元件	0~9999	带符号BIN16位	ANY16
(d)	存储BCD数据的软元件	—	BCD4位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	○	○	○	○	—	—	○	—	—	—	—

功能

将(s)中指定的软元件的BIN16位数据(0~9999)转换为BCD4位数据后,存储到(d)中指定的软元件中。



(1) 必须设置为0。

出错

错误代码(SD0)	内容
3281H	(s)的数据为0~9999以外时。

BIN数据→BCD8位转换

DBCD (P)

将指定的BIN32位数据转换为BCD8位数据。

梯形图	ST
	ENO:=DBCD (EN, s, d) ; ENO:=DBC DP (EN, s, d) ;

FBD/LD

■执行条件

指令	执行条件
DBCD	
DBC DP	

设置数据

■内容、范围、数据类型

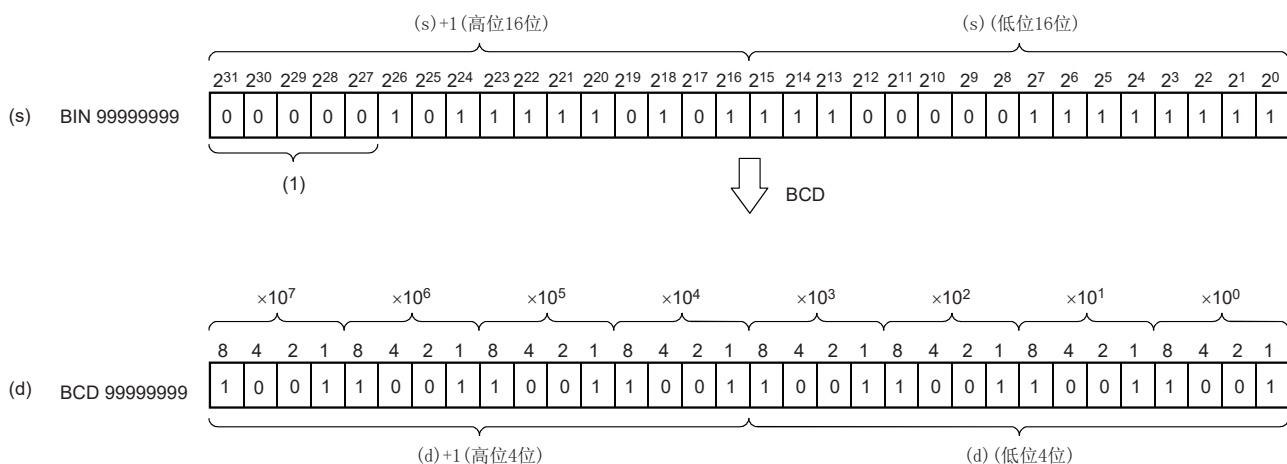
操作数	内容	范围	数据类型	数据类型(标签)
(s)	BIN数据或存储了BIN数据的起始软元件	0~99999999	带符号BIN32位	ANY32
(d)	存储BCD数据的起始软元件	—	BCD8位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	○	○	○	○	—	—	—

功能

将(s)中指定的软元件的BIN32位数据(0~99999999)转换为BCD8位数据后,存储到(d)中指定的软元件中。



(1) 必须设置为0。(高位5位)

出错

错误代码 (SD0)	内容
3281H	(s)的数据为0~99999999以外时。

BCD4位→BIN16位数据转换

BIN(P)

将指定的BCD4位数据转换为BIN16位数据。

梯形图	ST
	ENO:=BIN(EN, s, d); ENO:=BINP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
BIN	
BINP	

设置数据

■内容、范围、数据类型

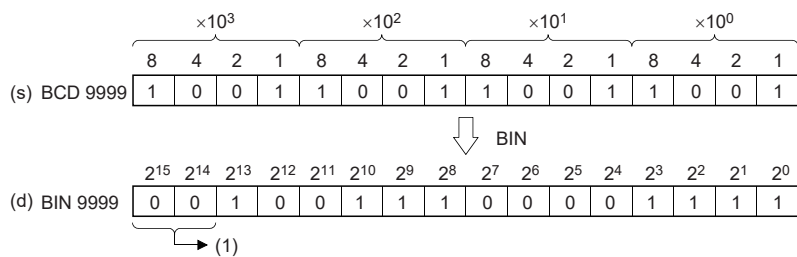
操作数	内容	范围	数据类型	数据类型(标签)
(s)	BCD数据或存储了BCD数据的软元件	0~9999	BCD4位	ANY16
(d)	存储BIN数据的软元件	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

将(s)中指定的软元件的BCD4位数据(0~9999)转换为BIN16位数据后,存储到(d)中指定的软元件中。



(1) 必定变为0。

出错

错误代码(SD0)	内容
3281H	(s)的各位中有0~9以外的值时。 ^{*1}

- *1 通过预先将SM754(BIN、DBIN指令错误禁止标志)置为ON,可以设置为不检测本错误。
 此外,设置了超出范围的数值的情况下,与SM754的ON/OFF无关,不执行指令。
 此外,BIN(P)指令的情况下,与错误的有无无关,在将指令(执行条件)置为OFF→ON之前,不执行下一个运算。

BCD8位→BIN32位数据转换

DBIN(P)

将指定的BCD8位数据转换为BIN32位数据。

梯形图	ST
	ENO:=DBIN (EN, s, d) ; ENO:=DBINP (EN, s, d) ;

FBD/LD

■执行条件

指令	执行条件
DBIN	
DBINP	

设置数据

■内容、范围、数据类型

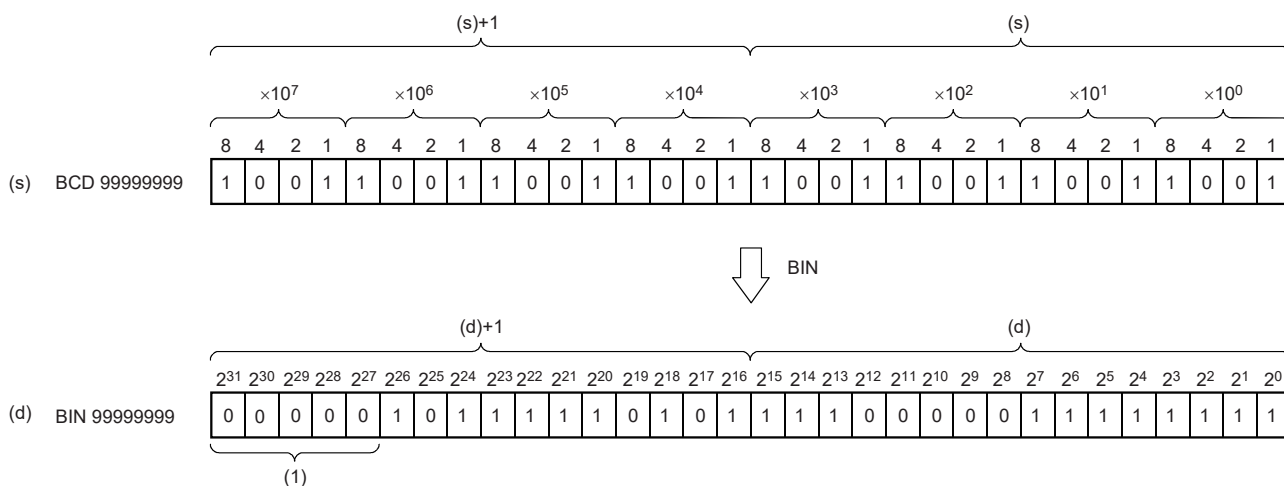
操作数	内容	范围	数据类型	数据类型(标签)
(s)	BCD数据或存储了BCD数据的起始软元件	0~99999999	BCD8位	ANY32
(d)	存储BIN数据的起始软元件	—	带符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—

功能

将(s)中指定的软元件的BCD8位数据(0~99999999)转换为BIN32位数据后,存储到(d)中指定的软元件中。



(1) 必定变为0。

出错

错误代码(SD0)	内容
3281H	(s)的各位中有0~9以外的值时。*1

- *1 通过预先将SM754(BIN、DBIN指令错误禁止标志)置为ON,可以设置为不检测本错误。
 此外,设置了超出范围的数值的情况下,与SM754的ON/OFF无关,不执行指令。
 此外,DBIN(P)指令的情况下,与错误的有无无关,在将指令(执行条件)置为OFF→ON之前,不执行下一个运算。

单精度实数→带符号BIN16位数据

FLT2INT(P)

将指定的单精度实数转换为带符号BIN16位数据。

梯形图	ST
	ENO:FLT2INT(EN, s, d); ENO:FLT2INTP(EN, s, d);

FBD/LD

■执行条件

指令	执行条件
FLT2INT	
FLT2INTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	单精度实数或存储了单精度实数的起始软元件	-32768~32767	单精度实数	ANYREAL_32
(d)	存储BIN数据的软元件	—	带符号BIN16位	ANY16_S
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—

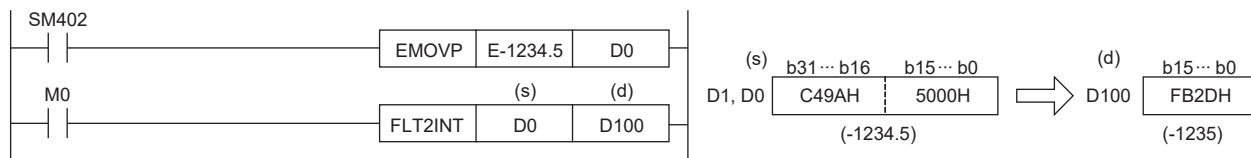
功能

- 将(s)中指定的单精度实数转换为带符号BIN16位数据后，存储到(d)中指定的软元件中。
- 转换后的数据，将变为(s)中指定的对单精度实数的小数点以下第1位进行了四舍五入后的值。
- 通过工程工具设置输入值的情况下，有可能会发生化整误差。关于通过工程工具设置输入值时的注意事项，请参阅下述内容。

☞ 38页 注意事项

例

M0变为ON时，将D0~D1中存储的单精度实数转换为带符号BIN16位数据后，存储到D100中。



出错

错误代码(SD0)	内容
3281H	(s)中设置的单精度实数超出了-32768~32767的范围时。
3282H	(s)中设置了特殊数时。 <ul style="list-style-type: none"> • 设置的单精度实数不在下述范围内。 $0, 2^{-126} \leq (s) < 2^{128}$ • 设置的软元件、标签的内容为-0、非正规化数、非数、±∞时。

单精度实数→无符号BIN16位数据

FLT2UINT (P)

将指定的单精度实数转换为无符号BIN16位数据。

梯形图	ST
	ENO:=FLT2UINT (EN, s, d) ; ENO:=FLT2UINTP (EN, s, d) ;

FBD/LD

■执行条件

指令	执行条件
FLT2UINT	
FLT2UINTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	单精度实数或存储了单精度实数的起始软元件	0~65535	单精度实数	ANYREAL_32
(d)	存储BIN数据的软元件	—	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—

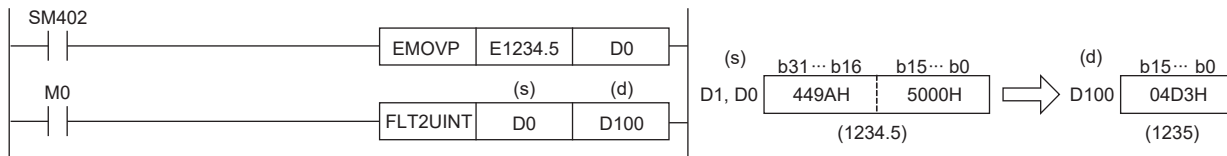
功能

- 将(s)中指定的单精度实数转换为无符号BIN16位数据后，存储到(d)中指定的软元件中。
- 转换后的数据，将变为(s)中指定的对单精度实数的小数点以下第1位进行了四舍五入后的值。
- 通过工程工具设置输入值的情况下，有可能会发生化整误差。关于通过工程工具设置输入值时的注意事项，请参阅下述内容。

☞ 38页 注意事项

例

M0变为ON时，将D0~D1中存储的单精度实数转换为无符号BIN16位数据后，存储到D100中。



出错

错误代码(SD0)	内容
3281H	(s)中设置的单精度实数超出了0~65535的范围时。
3282H	(s)中设置了特殊数时。 <ul style="list-style-type: none"> • 设置的单精度实数不在下述范围内。 $0, 2^{-126} \leq (s) < 2^{128}$ • 设置的软元件、标签的内容为-0、非正规化数、非数、±∞时。

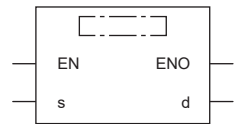
单精度实数→带符号BIN32位数据

FLT2DINT (P)

将指定的单精度实数转换为带符号BIN32位数据。

梯形图	ST
	ENO:=FLT2DINT (EN, s, d) ; ENO:=FLT2DINTP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
FLT2DINT	
FLT2DINTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	单精度实数或存储了单精度实数的起始软元件	-2147483648~2147483647	单精度实数	ANYREAL_32
(d)	存储BIN数据的起始软元件	—	带符号BIN32位	ANY32_S
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—

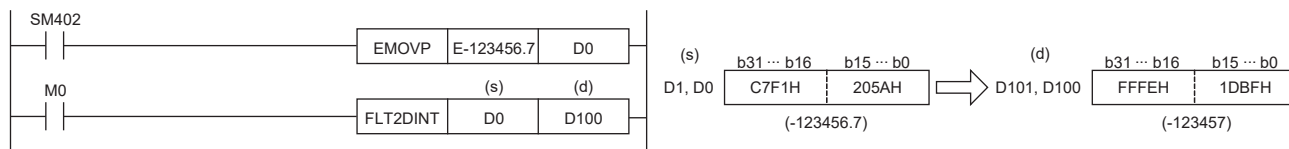
功能

- 将(s)中指定的单精度实数转换为带符号BIN32位数据后，存储到(d)中指定的软元件中。
- 转换后的数据，将变为(s)中指定的对单精度实数的小数点以下第1位进行了四舍五入后的值。
- 通过工程工具设置输入值的情况下，有可能会发生化整误差。关于通过工程工具设置输入值时的注意事项，请参阅下述内容。

☞ 38页 注意事项

例

M0变为ON时，将D0~D1中存储的单精度实数转换为带符号BIN32位数据后，存储到D100~D101中。



出错

错误代码(SD0)	内容
3281H	(s)中设置的单精度实数超出了-2147483648~2147483647的范围时。
3282H	(s)中设置了特殊数时。 <ul style="list-style-type: none"> • 设置的单精度实数不在下述范围内。 $0, 2^{-126} \leq (s) < 2^{128}$ • 设置的软元件、标签的内容为-0、非正规化数、非数、$\pm\infty$时。

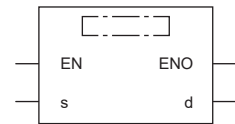
单精度实数→无符号BIN32位数据

FLT2UDINT (P)

将指定的单精度实数转换为无符号BIN32位数据。

梯形图	ST
	ENO:=FLT2UDINT (EN, s, d) ; ENO:=FLT2UDINTP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
FLT2UDINT	
FLT2UDINTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	单精度实数或存储了单精度实数的起始软元件	0~4294967295	单精度实数	ANYREAL_32
(d)	存储BIN数据的起始软元件	—	无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—

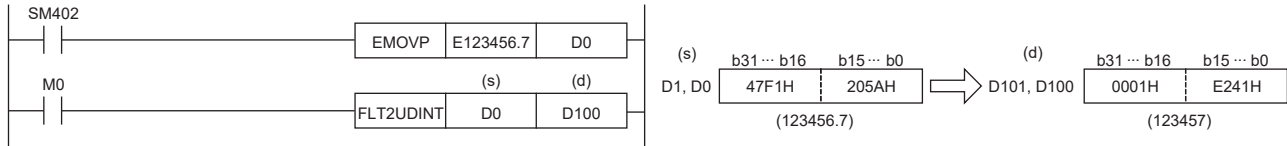
功能

- 将(s)中指定的单精度实数转换为无符号BIN32位数据后，存储到(d)中指定的软元件中。
- 转换后的数据，将变为(s)中指定的对单精度实数的小数点以下第1位进行了四舍五入后的值。
- 通过工程工具设置输入值的情况下，有可能会发生化整误差。关于通过工程工具设置输入值时的注意事项，请参阅下述内容。

☞ 38页 注意事项

例

M0变为ON时，将D0~D1中存储的单精度实数转换为无符号BIN32位数据后，存储到D100~D101中。



出错

错误代码 (SD0)	内容
3281H	(s)中设置的单精度实数超出了0~4294967295的范围时。
3282H	(s)中设置了特殊数时。 <ul style="list-style-type: none"> • 设置的单精度实数不在下述范围内。 $0 < 2^{-126} \leq (s) < 2^{128}$ • 设置的软元件、标签的内容为-0、非正规化数、非数、$\pm\infty$时。

双精度实数→带符号BIN16位数据

DBL2INT(P)

将指定的双精度实数转换为带符号BIN16位数据。

梯形图	ST
	ENO:=DBL2INT (EN, s, d); ENO:=DBL2INTP (EN, s, d);

FBD/LD



■执行条件

指令	执行条件
DBL2INT	
DBL2INTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	双精度实数或存储了双精度实数的起始软元件	-32768~32767	双精度实数	ANYREAL_64
(d)	存储BIN数据的软元件	—	带符号BIN16位	ANY16_S
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	—	○	—	○	—	—	○	—	—	—	—

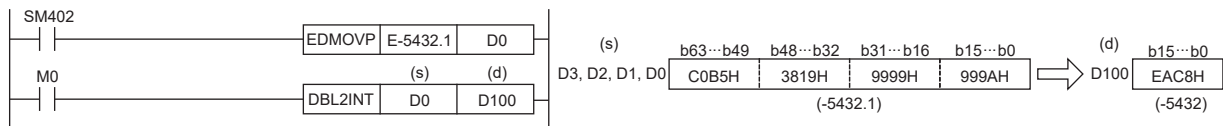
功能

- 将(s)中指定的双精度实数转换为带符号BIN16位数据后，存储到(d)中指定的软元件中。
- 转换后的数据，将变为(s)中指定的对双精度实数的小数点以下第1位进行了四舍五入后的值。
- 通过工程工具设置输入值的情况下，有可能会发生化整误差。关于通过工程工具设置输入值时的注意事项，请参阅下述内容。

☞ 38页 注意事项

例

M0变为ON时，将D0~D3中存储的双精度实数转换为带符号BIN16位数据后，存储到D100中。



出错

错误代码(SD0)	内容
3281H	(s)中设置的双精度实数超出了-32768~32767的范围时。
3282H	(s)中设置了特殊数时。 <ul style="list-style-type: none"> • 设置的双精度实数不在下述范围内。 $0 < 2^{-1022} \leq s < 2^{1024}$ • 设置的软元件、标签的内容为-0、非正规化数、非数、$\pm\infty$时。

双精度实数→无符号BIN16位数据

DBL2UINT (P)

将指定的双精度实数转换为无符号BIN16位数据。

梯形图	ST
	ENO:=DBL2UINT (EN, s, d) ; ENO:=DBL2UINTP (EN, s, d) ;

FBD/LD

■执行条件

指令	执行条件
DBL2UINT	
DBL2UINTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	双精度实数或存储了双精度实数的起始软元件	0~65535	双精度实数	ANYREAL_64
(d)	存储BIN数据的软元件	—	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	—	○	—	○	—	—	○	—	—	—	—

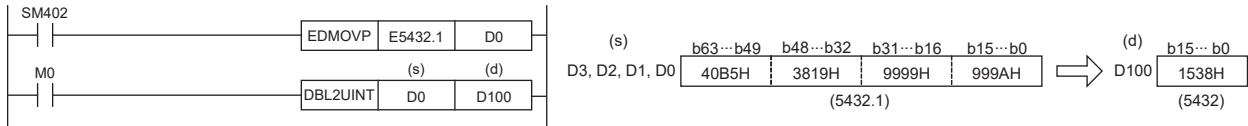
功能

- 将(s)中指定的双精度实数转换为无符号BIN16位数据后，存储到(d)中指定的软元件中。
- 转换后的数据，将变为(s)中指定的对双精度实数的小数点以下第1位进行了四舍五入后的值。
- 通过工程工具设置输入值的情况下，有可能会发生化整误差。关于通过工程工具设置输入值时的注意事项，请参阅下述内容。

☞ 38页 注意事项

例

M0变为ON时，将D0~D3中存储的双精度实数转换为无符号BIN16位数据后，存储到D100中。



出错

错误代码(SD0)	内容
3281H	(s)中设置的双精度实数超出了0~65535的范围时。
3282H	(s)中设置了特殊数时。 <ul style="list-style-type: none"> • 设置的双精度实数不在下述范围内。 $0 < 2^{-1022} \leq s < 2^{1024}$ • 设置的软元件、标签的内容为-0、非正规化数、非数、$\pm\infty$时。

双精度实数→带符号BIN32位数据

DBL2DINT (P)

将指定的双精度实数转换为带符号BIN32位数据。

梯形图	ST
	ENO:=DBL2DINT (EN, s, d) ; ENO:=DBL2DINTP (EN, s, d) ;

FBD/LD

■执行条件

指令	执行条件
DBL2DINT	
DBL2DINTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	双精度实数或存储了双精度实数的起始软元件	-2147483648~2147483647	双精度实数	ANYREAL_64
(d)	存储BIN数据的起始软元件	—	带符号BIN32位	ANY32_S
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	—	○	—	○	○	○	○	—	—	—	—

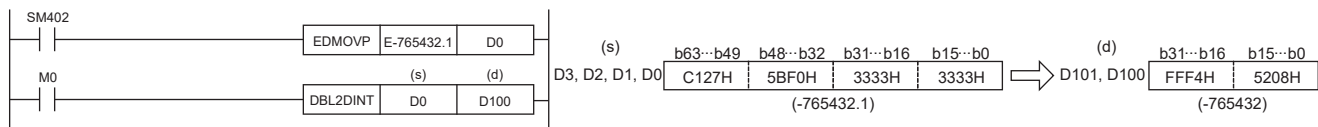
功能

- 将(s)中指定的双精度实数转换为带符号BIN32位数据后，存储到(d)中指定的软元件中。
- 转换后的数据，将变为(s)中指定的对双精度实数的小数点以下第1位进行了四舍五入后的值。
- 通过工程工具设置输入值的情况下，有可能会发生化整误差。关于通过工程工具设置输入值时的注意事项，请参阅下述内容。

☞ 38页 注意事项

例

M0变为ON时，将D0~D3中存储的双精度实数转换为带符号BIN32位数据后，存储到D100~D101中。



出错

错误代码(SD0)	内容
3281H	(s)中设置的双精度实数超出了-2147483648~2147483647的范围时。
3282H	(s)中设置了特殊数时。 <ul style="list-style-type: none"> • 设置的双精度实数不在下述范围内。 $0 < 2^{-1022} \leq (s) < 2^{1024}$ • 设置的软元件、标签的内容为-0、非正规化数、非数、$\pm\infty$时。

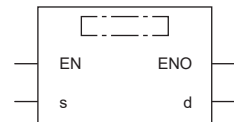
双精度实数→无符号BIN32位数据

DBL2UDINT (P)

将指定的双精度实数转换为无符号BIN32位数据。

梯形图	ST
	ENO:=DBL2UDINT (EN, s, d) ; ENO:=DBL2UDINTP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
DBL2UDINT	
DBL2UDINTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	双精度实数或存储了双精度实数的起始软元件	0~4294967295	双精度实数	ANYREAL_64
(d)	存储BIN数据的起始软元件	—	无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	—	○	—	○	○	○	○	—	—	—	—

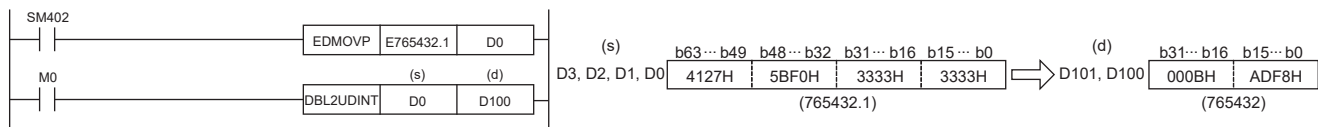
功能

- 将(s)中指定的双精度实数转换为无符号BIN32位数据后，存储到(d)中指定的软元件中。
- 转换后的数据，将变为(s)中指定的对双精度实数的小数点以下第1位进行了四舍五入后的值。
- 通过工程工具设置输入值的情况下，有可能会发生化整误差。关于通过工程工具设置输入值时的注意事项，请参阅下述内容。

☞ 38页 注意事项

例

M0变为ON时，将D0~D3中存储的双精度实数转换为无符号BIN32位数据后，存储到D100~D101中。



出错

错误代码(SD0)	内容
3281H	(s)中设置的双精度实数超出了0~4294967295的范围时。
3282H	(s)中设置了特殊数时。 <ul style="list-style-type: none"> • 设置的双精度实数不在下述范围内。 $0 < 2^{-1022} \leq s < 2^{1024}$ • 设置的软元件、标签的内容为-0、非正规化数、非数、$\pm\infty$时。

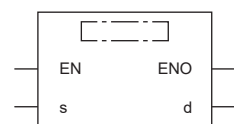
带符号BIN16位数据→无符号BIN16位数据转换

INT2UINT (P)

将指定的带符号BIN16位数据转换为无符号BIN16位数据。

梯形图	ST
	ENO:=INT2UINT (EN, s, d); ENO:=INT2UINTP (EN, s, d);

FBD/LD



■执行条件

指令	执行条件
INT2UINT	
INT2UINTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	BIN数据或存储了BIN数据的标签	-32768~32767	带符号BIN16位	ANY16_S
(d)	存储BIN数据的标签	—	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	—	—	—	—

要点

INT2UINT (P) 指令用于以下情况，通过标签进行编程时，使指令的操作数中可指定的数据类型与指定的标签的数据类型一致。

通过软件元件进行编程的情况下，无需使用INT2UINT (P) 指令。

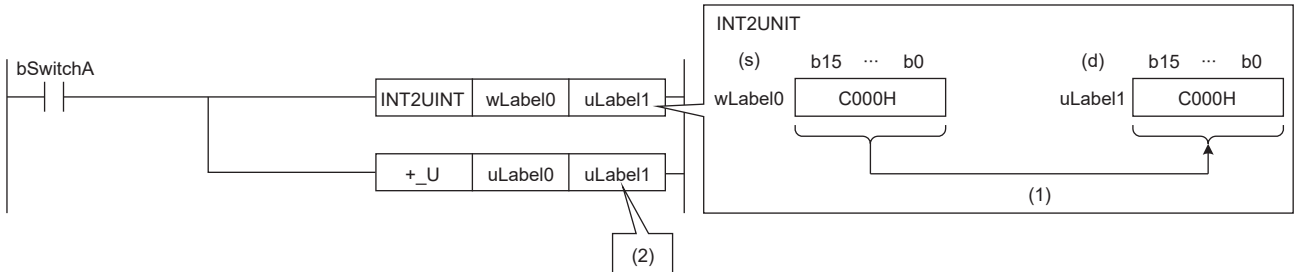
功能

- 将(s)中指定的带符号BIN16位数据(ANY16_S)转换为无符号BIN16位数据(ANY16_U)后, 存储到(d)中指定的标签中。
- INT2UINT(P)指令的使用示例如下所示。

例

+_U指令, 需要在操作数中指定ANY16_U, 因此执行+_U指令之前使用INT2UINT指令, 将ANY16_S的wLabel0转换为ANY16_U的uLabel1。

将wLabel0作为值直接存储到uLabel1中。



bSwitchA: 位

wLabel0: 字[带符号]

uLabel0、uLabel1: 字[无符号]/位串[16位]

(1) 原样不变地存储。

(2) 根据+_U指令的操作数的数据类型进行运算。

出错

没有运算错误。

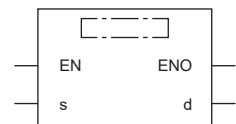
带符号BIN16位数据→带符号BIN32位数据转换

INT2DINT (P)

将指定的带符号BIN16位数据转换为带符号BIN32位数据。

梯形图	ST
	ENO:=INT2DINT (EN, s, d) ; ENO:=INT2DINTP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
INT2DINT	
INT2DINTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	BIN数据或存储了BIN数据的软元件	-32768~32767	带符号BIN16位	ANY16_S
(d)	存储BIN数据的起始软元件	—	带符号BIN32位	ANY32_S
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

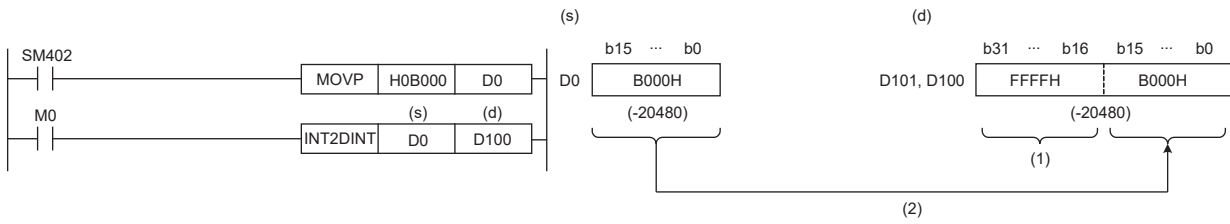
操作数	位		字			双字		间接指定	常数				其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

将(s)中指定的带符号BIN16位数据转换为带符号BIN32位数据后, 存储到(d)中指定的软元件中。

例

M0变为ON时, 将D0中存储的带符号BIN16位数据转换为带符号BIN32位数据后, 存储到D100~D101中。



- (1) 以转换前的最高位的值填充。
 (2) 存储至低位16位

出错

没有运算错误。

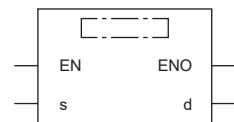
带符号BIN16位数据→无符号BIN32位数据转换

INT2UDINT (P)

将指定的带符号BIN16位数据转换为无符号BIN32位数据。

梯形图	ST
	<pre>ENO:=INT2UDINT (EN, s, d); ENO:=INT2UDINTP (EN, s, d);</pre>

FBD/LD



■执行条件

指令	执行条件
INT2UDINT	
INT2UDINTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	BIN数据或存储了BIN数据的软元件	-32768~32767	带符号BIN16位	ANY16_S
(d)	存储BIN数据的起始软元件	—	无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

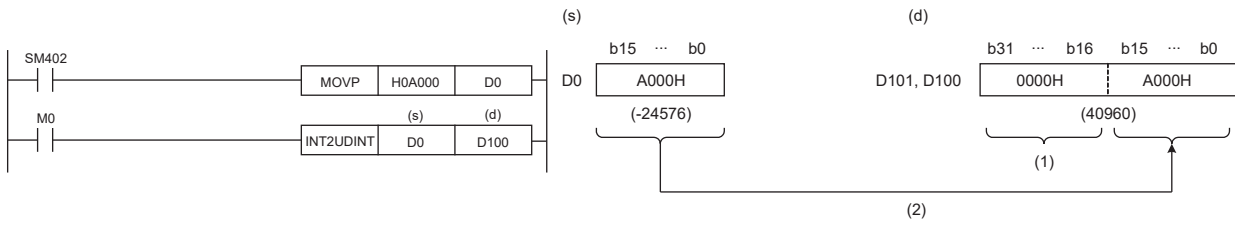
操作数	位		字			双字		间接指定	常数				其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

将(s)中指定的带符号BIN16位数据转换为无符号BIN32位数据后, 存储到(d)中指定的软元件中。

例

M0变为ON时, 将D0中存储的带符号BIN16位数据转换为无符号BIN32位数据后, 存储到D100~D101中。



(1) 存储0

(2) 存储至低位16位

出错

没有运算错误。

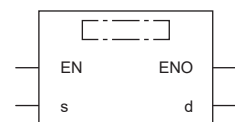
无符号BIN16位数据→带符号BIN16位数据转换

UINT2INT (P)

将指定的无符号BIN16位数据转换为带符号BIN16位数据。

梯形图	ST
	ENO:=UINT2INT (EN, s, d) ; ENO:=UINT2INTP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
UINT2INT	
UINT2INTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	BIN数据或存储了BIN数据的标签	0~65535	无符号BIN16位	ANY16_U
(d)	存储BIN数据的标签	—	带符号BIN16位	ANY16_S
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—	—

要点

UINT2INT (P) 指令用于以下情况，通过标签进行编程时，使指令的操作数中可指定的数据类型与指定的标签的数据类型一致。

通过软元件进行编程的情况下，无需使用UINT2INT (P) 指令。

功能

- 将(s)中指定的无符号BIN16位数据(ANY16_U)转换为带符号BIN16位数据(ANY16_S)后, 存储到(d)中指定的标签中。
- UINT2INT(P)指令的使用示例如下所示。

例

+指令, 需要在操作数中指定ANY16_S, 因此执行+指令之前使用UINT2INT指令, 将ANY16_U的uLabel0转换为ANY16_S的wLabel1。

将uLabel0作为值直接存储到wLabel1中。



bSwitchA: 位

wLabel0, wLabel1: 字[带符号]

uLabel0: 字[无符号]/位串[16位]

(1) 原样不变地存储。

(2) 根据+指令的操作数的数据类型进行运算。

出错

没有运算错误。

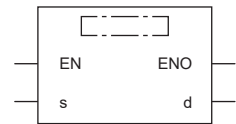
无符号BIN16位数据→带符号BIN32位数据转换

UINT2DINT (P)

将指定的无符号BIN16位数据转换为带符号BIN32位数据。

梯形图	ST
	ENO:=UINT2DINT (EN, s, d) ; ENO:=UINT2DINTP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
UINT2DINT	
UINT2DINTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	BIN数据或存储了BIN数据的软元件	0~65535	无符号BIN16位	ANY16_U
(d)	存储BIN数据的起始软元件	—	带符号BIN32位	ANY32_S
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

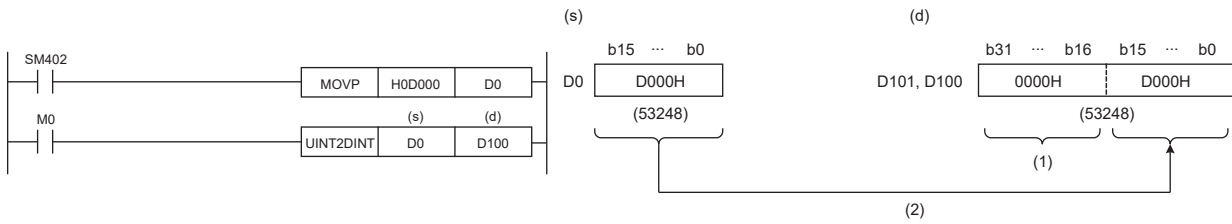
操作数	位		字			双字		间接指定	常数				其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

将(s)中指定的无符号BIN16位数据转换为带符号BIN32位数据后，存储到(d)中指定的软元件中。

例

M0变为ON时，将D0中存储的无符号BIN16位数据转换为带符号BIN32位数据后，存储到D100~D101中。



(1) 存储0

(2) 存储至低位16位

出错

没有运算错误。

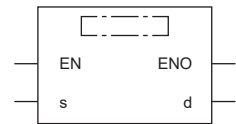
无符号BIN16位数据→无符号BIN32位数据转换

UINT2UDINT (P)

将指定的无符号BIN16位数据转换为无符号BIN32位数据。

梯形图	ST
	ENO:=UINT2UDINT (EN, s, d) ; ENO:=UINT2UDINTP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
UINT2UDINT	
UINT2UDINTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	BIN数据或存储了BIN数据的软元件	0~65535	无符号BIN16位	ANY16_U
(d)	存储BIN数据的起始软元件	—	无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

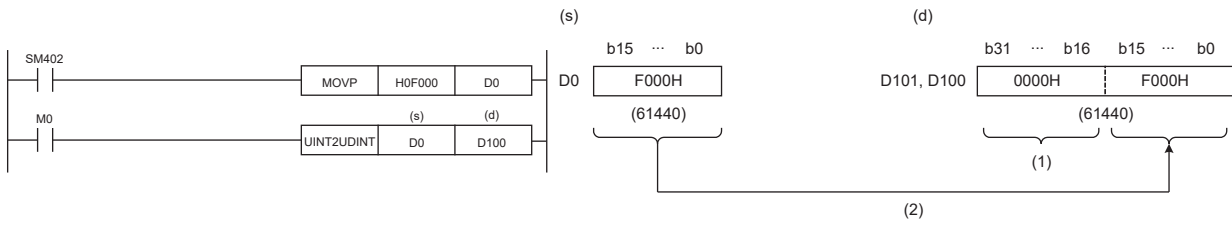
操作数	位		字			双字		间接指定	常数				其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	○	○	○	○	○	—	—	○	○	—	—	—	—
(d)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

将(s)中指定的无符号BIN16位数据转换为无符号BIN32位数据后，存储到(d)中指定的软元件中。

例

M0变为ON时，将D0中存储的无符号BIN16位数据转换为无符号BIN32位数据后，存储到D100~D101中。



(1) 存储0

(2) 存储至低位16位

出错

没有运算错误。

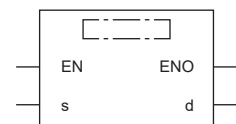
带符号BIN32位数据→带符号BIN16位数据转换

DINT2INT (P)

将指定的带符号BIN32位数据转换为带符号BIN16位数据。

梯形图	ST
	ENO:=DINT2INT (EN, s, d) ; ENO:=DINT2INTP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
DINT2INT	
DINT2INTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	BIN数据或存储了BIN数据的起始软元件	-32768~32767	带符号BIN32位	ANY32_S
(d)	存储BIN数据的软元件	—	带符号BIN16位	ANY16_S
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

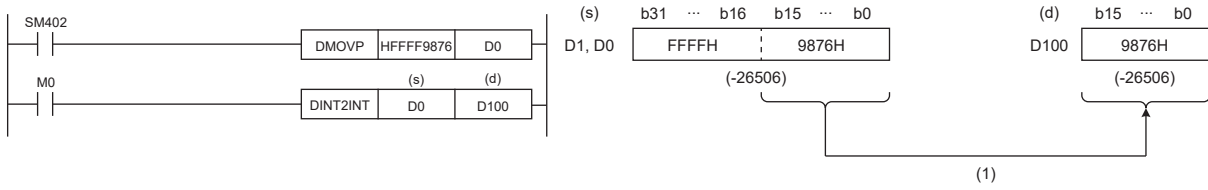
操作数	位		字		双字		间接指定	常数			其它	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K	H		E
(s)	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

将(s)中指定的带符号BIN32位数据转换为带符号BIN16位数据转换后，存储到(d)中指定的软元件中。

例

M0变为ON时，将D0~D1中存储的带符号BIN32位数据转换为带符号BIN16位数据后，存储到D100中。



(1) 存储至低位16位

出错

错误代码 (SD0)	内容
3281H	(s) 中设置的带符号BIN32位数据超出了-32768~32767的范围时。

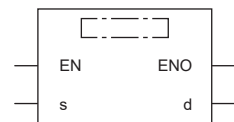
带符号BIN32位数据→无符号BIN16位数据转换

DINT2UINT (P)

将指定的带符号BIN32位数据转换为无符号BIN16位数据。

梯形图	ST
	ENO:=DINT2UINT (EN, s, d) ; ENO:=DINT2UINTP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
DINT2UINT	
DINT2UINTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	BIN数据或存储了BIN数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
(d)	存储BIN数据的软元件	—	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

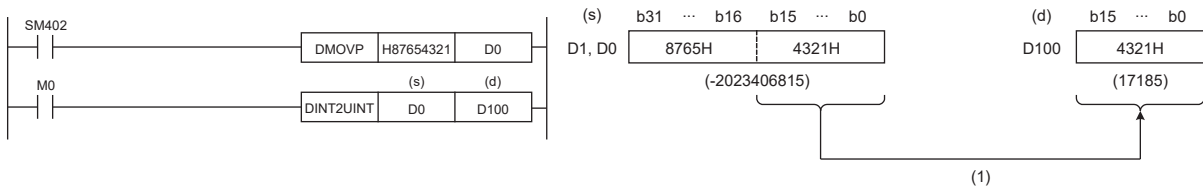
操作数	位		字			双字		间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

将(s)中指定的带符号BIN32位数据转换为无符号BIN16位数据后，存储到(d)中指定的软元件中。

例

M0变为ON时，将D0~D1中存储的带符号BIN32位数据转换为无符号BIN16位数据后，存储到D100中。



(1) 存储至低位16位

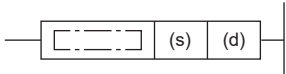
出错

没有运算错误。

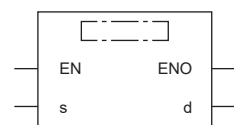
带符号BIN32位数据→无符号BIN32位数据转换

DINT2UDINT (P)



将指定的带符号BIN32位数据转换为无符号BIN32位数据。

梯形图	ST
	ENO:=DINT2UDINT (EN, s, d); ENO:=DINT2UDINTP (EN, s, d);

FBD/LD



■执行条件

指令	执行条件
DINT2UDINT	
DINT2UDINTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	BIN数据或存储了BIN数据的标签	-2147483648~2147483647	带符号BIN32位	ANY32_S
(d)	存储BIN数据的标签	—	无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	○	—	—	—

要点

DINT2UDINT (P) 指令，用于在通过标签进行编程的情况下，使指令的操作数中可指定的数据类型与指定标签的数据类型一致。

通过软元件进行编程的情况下，无需使用DINT2UDINT (P) 指令。

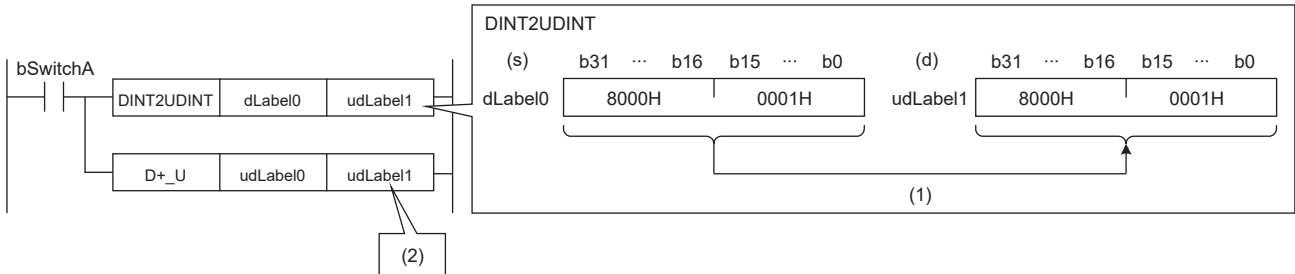
功能

- 将(s)中指定的带符号BIN32位数据(ANY32_S)转换为无符号BIN32位数据(ANY32_U)后, 存储到(d)中指定的标签中。
- DINT2UDINT(P)指令的使用示例如下所示。

例

D+_U指令, 需要在操作数中指定ANY32_U, 因此执行D+_U指令之前使用DINT2UDINT指令, 将ANY32_S的dLabel10转换为ANY32_U的udLabel11。

将dLabel10作为值直接存储到udLabel11中。



bSwitchA: 位

dLabel10: 双字[带符号]

udLabel10、udLabel11: 双字[无符号]/位串[32位]

(1) 原样不变地存储。

(2) 根据D+_U指令的操作数的数据类型进行运算。

出错

没有运算错误。

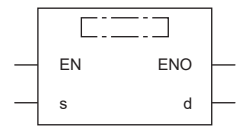
无符号BIN32位数据→带符号BIN16位数据转换

UDINT2INT (P)

将指定的无符号BIN32位数据转换为带符号BIN16位数据。

梯形图	ST
	ENO:=UDINT2INT (EN, s, d) ; ENO:=UDINT2INTP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
UDINT2INT	
UDINT2INTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s)	BIN数据或存储了BIN数据的起始软元件	0~4294967295	无符号BIN32位	ANY32_U
(d)	存储BIN数据的软元件	—	带符号BIN16位	ANY16_S
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

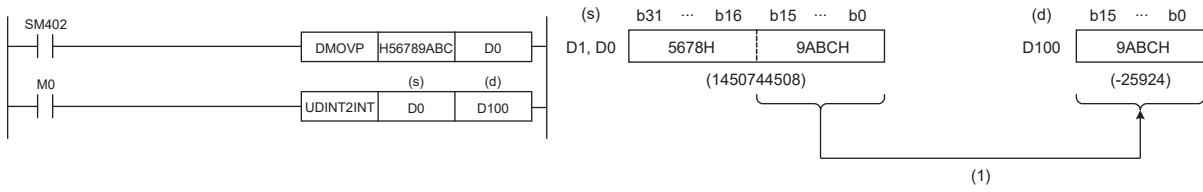
操作数	位		字		双字		间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K、H	E	
(s)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—

功能

将(s)中指定的无符号BIN32位数据转换为带符号BIN16位数据后，存储到(d)中指定的软元件中。

例

M0变为ON时，将D0~D1中存储的无符号BIN32位数据转换为带符号BIN16位数据后，存储到D100中。



(1) 存储至低位16位

出错

没有运算错误。

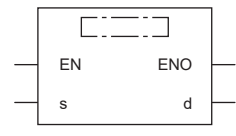
无符号BIN32位数据→无符号BIN16位数据转换

UDINT2UINT (P)

将指定的无符号BIN32位数据转换为无符号BIN16位数据。

梯形图	ST
	ENO:=UDINT2UINT (EN, s, d); ENO:=UDINT2UINTP (EN, s, d);

FBD/LD



■执行条件

指令	执行条件
UDINT2UINT	
UDINT2UINTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	BIN数据或存储了BIN数据的起始软元件	0~4294967295	无符号BIN32位	ANY32_U
(d)	存储BIN数据的软元件	—	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

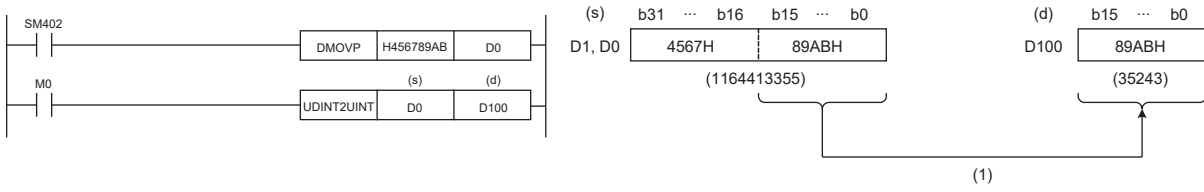
操作数	位		字		双字		间接指定	常数			其它	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K	H		E
(s)	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

将(s)中指定的无符号BIN32位数据转换为无符号BIN16位数据后，存储到(d)中指定的软元件中。

例

M0变为ON时，将D0~D1中存储的无符号BIN32位数据转换为无符号BIN16位数据后，存储到D100中。



(1) 存储至低位16位

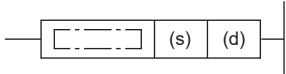
出错

没有运算错误。

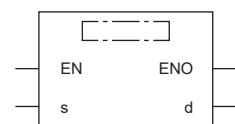
无符号BIN32位数据→带符号BIN32位数据转换

UDINT2DINT (P)


将指定的无符号BIN32位数据转换为带符号BIN32位数据。

梯形图	ST
	ENO:=UDINT2DINT (EN, s, d); ENO:=UDINT2DINTP (EN, s, d);

FBD/LD



■执行条件

指令	执行条件
UDINT2DINT	
UDINT2DINTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	BIN数据或存储了BIN数据的标签	0~4294967295	无符号BIN32位	ANY32_U
(d)	存储BIN数据的标签	—	带符号BIN32位	ANY32_S
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	○	○	○	○	○	○	○	○	—	—	—

要点

UDINT2DINT (P) 指令，用于在通过标签进行编程的情况下，使指令的操作数中可指定的数据类型与指定的标签的数据类型一致。

通过软元件进行编程的情况下，无需使用UDINT2DINT (P) 指令。

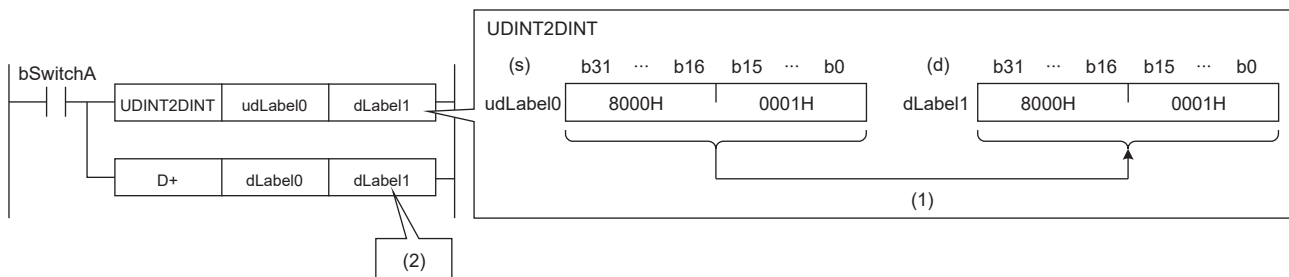
功能

- 将(s)中指定的无符号BIN32位数据(ANY32_U)转换为带符号BIN32位数据(ANY32_S)后, 存储到(d)中指定的标签中。
- UDINT2DINT(P)指令的使用示例如下所示。

例

D+指令, 需要在操作数中指定ANY32_S, 因此执行D+指令之前使用UDINT2DINT指令, 将ANY32_U的udLabel0转换为ANY32_S的dLabel1。

将udLabel0作为值直接存储到dLabel1中。



bSwitchA: 位

dLabel0、dLabel1: 双字[带符号]

udLabel0: 双字[无符号]/位串[32位]

(1) 原样不变地存储。

(2) 根据D+指令的操作数的数据类型进行运算。

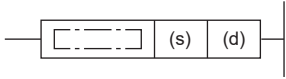
出错

没有运算错误。

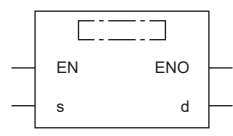
BIN16位数据→BIN格雷码数据转换

GRY(P) (_U)



将指定的BIN16位数据转换为BIN16位格雷码数据。

梯形图	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



■执行条件

指令	执行条件
GRY GRY_U	
GRYP GRYP_U	

设置数据

■内容、范围、数据类型

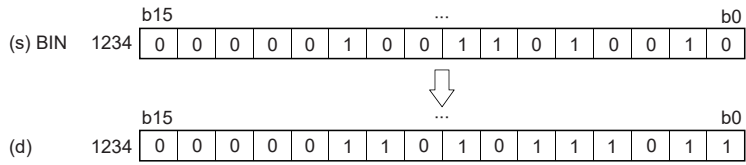
操作数	内容	范围	数据类型	数据类型(标签)	
(s)	GRY(P)	BIN数据或存储了BIN数据的软元件	0~32767	带符号BIN16位	ANY16_S
	GRY(P)_U		0~65535	无符号BIN16位	ANY16_U
(d)	GRY(P)	存储转换后的格雷码数据的软元件	—	带符号BIN16位	ANY16_S
	GRY(P)_U		—	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—	—

功能

将(s)中指定的BIN16位数据转换为BIN16位格雷码数据后，存储到(d)中指定的软元件中。



出错

错误代码(SD0)	内容
3281H	使用GRY(P)指令时，(s)中指定的值超出0~32767的范围时。

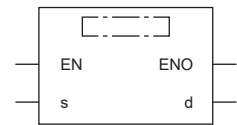
BIN32位数据→格雷码数据转换

DGRY(P) (_U)

将指定的BIN32位数据转换为BIN32位格雷码数据。

梯形图	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



■执行条件

指令	执行条件
DGRY DGRY_U	
DGRYP DGRYP_U	

设置数据

■内容、范围、数据类型

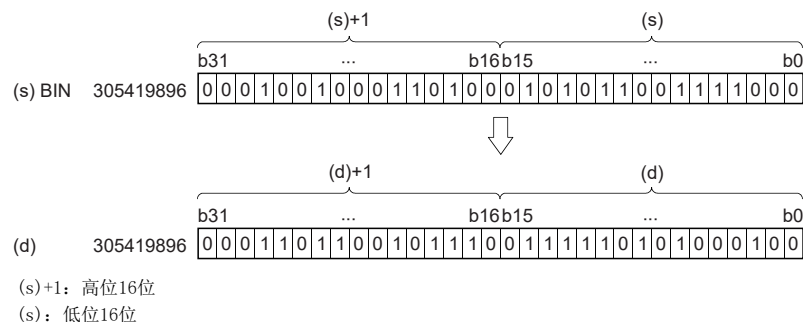
操作数	内容	范围	数据类型	数据类型(标签)	
(s)	DGRY(P)	BIN数据或存储了BIN数据的起始软元件	0~2147483647	带符号BIN32位	ANY32_S
	DGRY(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(d)	DGRY(P)	存储转换后的格雷码数据的起始软元件	—	带符号BIN32位	ANY32_S
	DGRY(P)_U			无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	○	—	—	—	—

功能

将(s)中指定的BIN32位数据转换为BIN32位格雷码数据后，存储到(d)中指定的软元件中。



错误代码 (SD0)	内容
3281H	使用DGRY (P) 指令时, (s) 中指定的值超出了0~2147483647的范围时。

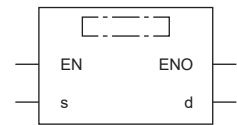
BIN16位格雷码数据→BIN16位数据转换

GBIN(P) (_U)

将指定的BIN16位格雷码数据转换为BIN16位数据。

梯形图	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



■执行条件

指令	执行条件
GBIN GBIN_U	
GBINP GBINP_U	

设置数据

■内容、范围、数据类型

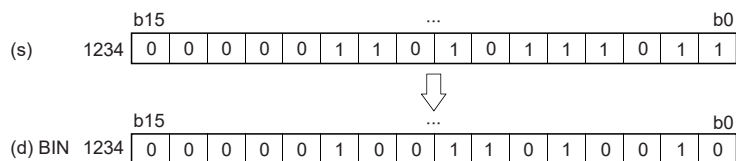
操作数	内容	范围	数据类型	数据类型(标签)	
(s)	GBIN(P)	格雷码数据或存储了格雷码数据的软元件	0~32767	带符号BIN16位	ANY16_S
	GBIN(P)_U		0~65535	无符号BIN16位	ANY16_U
(d)	GBIN(P)	存储转换后的BIN数据的软元件	—	带符号BIN16位	ANY16_S
	GBIN(P)_U			无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

将(s)中指定的软元件中存储的BIN16位格雷码数据转换为BIN16位数据后，存储到(d)中指定的软元件中。



错误代码(SD0)	内容
3281H	使用GBIN(P)指令时，(s)中指定的值超出了0~32767的范围时。

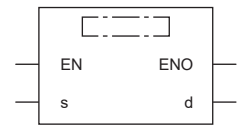
BIN32位格雷码数据→BIN32位数据转换

DGBIN(P) (_U)

将指定的BIN32位格雷码数据转换为BIN32位数据。

梯形图	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



■执行条件

指令	执行条件
DGBIN DGBIN_U	
DGBINP DGBINP_U	

设置数据

■内容、范围、数据类型

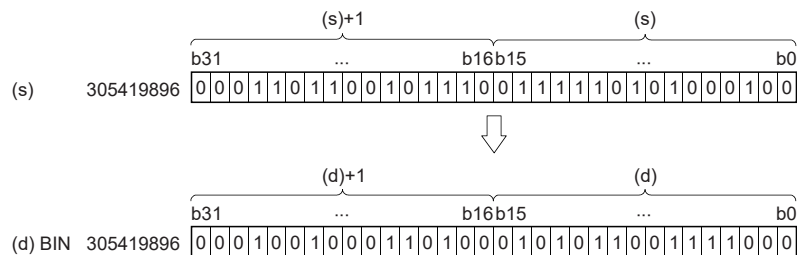
操作数	内容	范围	数据类型	数据类型(标签)	
(s)	DGBIN(P)	格雷码数据或存储了格雷码数据的起始软元件	0~2147483647	带符号BIN32位	ANY32_S
	DGBIN(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(d)	DGBIN(P)	存储转换后的BIN数据的起始软元件	—	带符号BIN32位	ANY32_S
	DGBIN(P)_U			无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—

功能

将(s)中指定的软元件中存储的BIN32位格雷码数据转换为BIN32位数据后，存储到(d)中指定的软元件中。



(s)+1、(d)+1: 高位16位
(s)、(d): 低位16位

错误代码 (SD0)	内容
3281H	使用DGBIN(P)指令时，(s)中指定的值超出了0~2147483647的范围时。

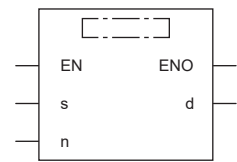
BIN16位数据块→BCD4位数据转换块

BKBCD(P)

对指定的软元件开始的(n)点的BIN数据(0~9999)进行BCD转换。

梯形图	ST
	ENO:=BKBCD(EN, s, n, d); ENO:=BKBCDP(EN, s, n, d);

FBD/LD



■执行条件

指令	执行条件
BKBCD	
BKBCDP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储了BIN数据的起始软元件	—	带符号BIN16位	ANY16*1
(d)	存储转换后的BCD数据的起始软元件	—	BCD4位	ANY16*1
(n)	变数数据数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

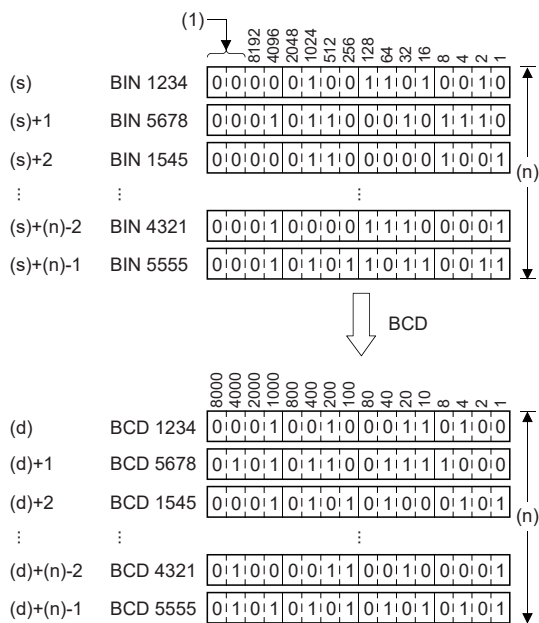
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

对从(s)中指定的软元件开始的(n)点的BIN16位数据(0~9999)进行BCD转换后,存储到(d)中指定的软元件及其以后。



(1) 必须设置为0。

出错

错误代码(SD0)	内容
2821H	(s)、(d)的软元件重复时。
3281H	(s)的软元件开始的(n)点的数据超出0~9999的范围时。

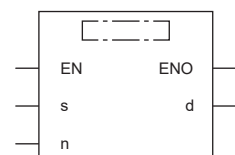
块BCD4位数据→块BIN16位转换数据

BKBIN(P)

对指定的软元件开始的 (n) 点的BCD数据 (0~9999) 进行BIN转换。

梯形图	ST
	ENO:=BKBIN(EN, s, n, d); ENO:=BKBINP(EN, s, n, d);

FBD/LD



■执行条件

指令	执行条件
BKBIN	
BKBINP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储了BCD数据的起始软元件	—	BCD4位	ANY16*1
(d)	存储转换后的BIN数据的起始软元件	—	带符号BIN16位	ANY16*1
(n)	变数数据数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

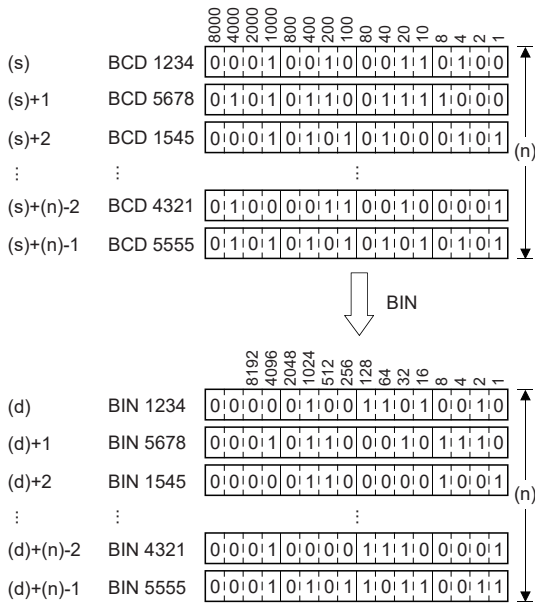
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

对从(s)中指定的软元件开始的(n)点的BCD数据(0~9999)进行BIN16位转换后, 存储到(d)中指定的软元件及其以后。



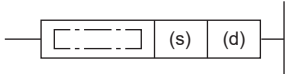
出错

错误代码(SD0)	内容
2821H	(s)、(d)的软元件重复时。
3281H	(s)的软元件开始的(n)点的数据超出0~9999的范围时。

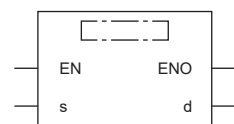
10进制ASCII数据→BIN16位数据转换

DABIN(P) (_U)



将10进制ASCII数据转换为BIN16位数据。

梯形图	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



■执行条件

指令	执行条件
DABIN DABIN_U	
DABINP DABINP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	转换为BIN值的ASCII数据, 或存储了ASCII数据的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	DABIN(P)	—	带符号BIN16位	ANY16_S
	DABIN(P)_U		无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—

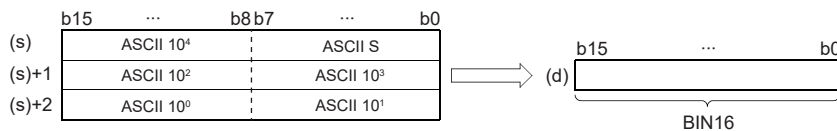
功能

- 将(s)中指定的软元件编号及其以后中存储的10进制ASCII数据转换为BIN16位数据后，存储到(d)中指定的软元件中。
- 根据SM705(变数位数切换)的状态，(s)中设置的10进制ASCII数据的设置方法将有所不同。

SM705的状态	(s)的设置方法	参阅目标
OFF	以固定位数(符号+数值部分5位)设置	439页 SM705(变数位数切换)为OFF时(s)的设置方法
ON	以任意位数(最大: 符号+数值部分5位)设置	440页 SM705(变数位数切换)为ON时(s)的设置方法

■SM705(变数位数切换)为OFF时(s)的设置方法

以固定位数将10进制ASCII数据设置到(s)~(s)+2中。

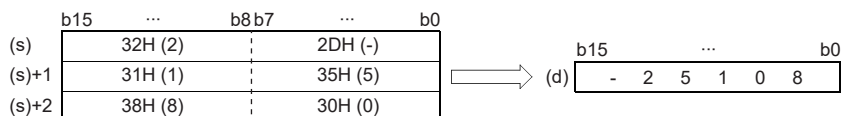


ASCII S: 符号的ASCII代码
 ASCII 10⁴: 万位的ASCII代码
 ASCII 10³: 千位的ASCII代码
 ASCII 10²: 百位的ASCII代码
 ASCII 10¹: 十位的ASCII代码
 ASCII 10⁰: 个位的ASCII代码

- (s)~(s)+2中指定的ASCII数据，在DABIN(P)指令的情况下范围为-32768~32767，DABIN(P)_U指令的情况下范围为0~65535。
- (s)+3及其以后的数据将被忽略。
- 符号数据中，转换的数据为正时设置20H(空白)、为负时设置2DH(-)。(设置了20H、2DH以外的情况下，将被作为正的数据处理。)
- 各位中设置的ASCII代码的范围为30H~39H。
- 各位中设置的ASCII代码为20H或00H时，将作为30H处理。

例

DABIN(P)指令中，在(s)中设置“-25108”时



■SM705(变数位数据切换)为ON时(s)的设置方法

在(s)中设置任意位数(包含00H(NULL代码))的10进制ASCII数据。只是,整数部分为最大位数(5位)时,不用设置00H(NULL代码)。

(s)的设置方法如下所示。

设置于s中的值	(s)~(s)+2的内容	设置于s中的值	(s)~(s)+2的内容																
<ul style="list-style-type: none"> • 0 • 正的数(数值部分1位) 	<ul style="list-style-type: none"> • 在(s)的高位字节中设置00H。 • (s)+1及其以后将被忽略。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8 b7 ... b0</td> </tr> <tr> <td>(s)+1</td> <td>00H ... ASCII 10⁰</td> </tr> <tr> <td>(s)+2</td> <td></td> </tr> </table>	(s)	b15 ... b8 b7 ... b0	(s)+1	00H ... ASCII 10 ⁰	(s)+2		<ul style="list-style-type: none"> • 正的数(数值部分2位) • 负的数(数值部分1位) 	<ul style="list-style-type: none"> • 在(s)+1的低位字节中设置00H。 • (s)+1的高位字节及其以后将被忽略。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8 b7 ... b0</td> </tr> <tr> <td>(s)+1</td> <td>ASCII 10⁰ ... ASCII 10¹ / 2DH (-)</td> </tr> <tr> <td>(s)+2</td> <td>00H</td> </tr> </table>	(s)	b15 ... b8 b7 ... b0	(s)+1	ASCII 10 ⁰ ... ASCII 10 ¹ / 2DH (-)	(s)+2	00H				
(s)	b15 ... b8 b7 ... b0																		
(s)+1	00H ... ASCII 10 ⁰																		
(s)+2																			
(s)	b15 ... b8 b7 ... b0																		
(s)+1	ASCII 10 ⁰ ... ASCII 10 ¹ / 2DH (-)																		
(s)+2	00H																		
⋮																			
负的数(数值部分4位)	在(s)+2的高位字节中设置00H。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8 b7 ... b0</td> </tr> <tr> <td>(s)+1</td> <td>ASCII 10³ ... 2DH (-)</td> </tr> <tr> <td>(s)+2</td> <td>ASCII 10¹ ... ASCII 10²</td> </tr> <tr> <td>(s)+3</td> <td>00H ... ASCII 10⁰</td> </tr> </table>	(s)	b15 ... b8 b7 ... b0	(s)+1	ASCII 10 ³ ... 2DH (-)	(s)+2	ASCII 10 ¹ ... ASCII 10 ²	(s)+3	00H ... ASCII 10 ⁰	正的数(数值部分5位)	(s)+2的高位字节及其以后将被忽略。由于为最大位数,不用设置00H。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8 b7 ... b0</td> </tr> <tr> <td>(s)+1</td> <td>ASCII 10³ ... ASCII 10⁴</td> </tr> <tr> <td>(s)+2</td> <td>ASCII 10¹ ... ASCII 10²</td> </tr> <tr> <td>(s)+3</td> <td>ASCII 10⁰</td> </tr> </table>	(s)	b15 ... b8 b7 ... b0	(s)+1	ASCII 10 ³ ... ASCII 10 ⁴	(s)+2	ASCII 10 ¹ ... ASCII 10 ²	(s)+3	ASCII 10 ⁰
(s)	b15 ... b8 b7 ... b0																		
(s)+1	ASCII 10 ³ ... 2DH (-)																		
(s)+2	ASCII 10 ¹ ... ASCII 10 ²																		
(s)+3	00H ... ASCII 10 ⁰																		
(s)	b15 ... b8 b7 ... b0																		
(s)+1	ASCII 10 ³ ... ASCII 10 ⁴																		
(s)+2	ASCII 10 ¹ ... ASCII 10 ²																		
(s)+3	ASCII 10 ⁰																		
负的数(数值部分5位)	(s)+3及其以后将被忽略。由于为最大位数,不用设置00H。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8 b7 ... b0</td> </tr> <tr> <td>(s)+1</td> <td>ASCII 10⁴ ... 2DH (-)</td> </tr> <tr> <td>(s)+2</td> <td>ASCII 10² ... ASCII 10³</td> </tr> <tr> <td>(s)+3</td> <td>ASCII 10⁰ ... ASCII 10¹</td> </tr> </table>	(s)	b15 ... b8 b7 ... b0	(s)+1	ASCII 10 ⁴ ... 2DH (-)	(s)+2	ASCII 10 ² ... ASCII 10 ³	(s)+3	ASCII 10 ⁰ ... ASCII 10 ¹	ASCII 10 ⁰ : 个位的ASCII代码 ASCII 10 ¹ : 十位的ASCII代码 ⋮ ASCII 10 ⁴ : 万位的ASCII代码									
(s)	b15 ... b8 b7 ... b0																		
(s)+1	ASCII 10 ⁴ ... 2DH (-)																		
(s)+2	ASCII 10 ² ... ASCII 10 ³																		
(s)+3	ASCII 10 ⁰ ... ASCII 10 ¹																		

- (s)~(s)+2中指定的ASCII数据,在DABIN(P)指令的情况下范围为-32768~32767, DABIN(P)_U指令的情况下范围为0~65535。
- 转换的数据为负的数时,作为符号数据将2DH(-)设置到(s)+0的低位字节中。转换的数据为0或正的数时,不用设置符号数据,直接设置最高位位的ASCII代码。
- 各位中设置的ASCII代码的范围为30H~39H。
- 正的数中数值部分为5位时,(s)+2的高位字节及其以后的数据将被忽略。负的数中数值部分为5位时,(s)+3及其以后的数据将被忽略。
- 各位中设置的ASCII代码为20H时,将作为30H处理。00H时,将作为10进制ASCII数据的终端处理。
- 下述情况下,(d)中将存储0。
 - 第1字符为00H(NULL)
 - 第1字符为2DH(-)、第2字符为00H(NULL)

出错

错误代码(SD0)	内容
3281H	(s)~(s)+2中输入了不能转换的非法数据时。 <ul style="list-style-type: none"> • 第1字符的ASCII代码为2DH、30H~39H、20H、00H以外。^{*1} • 第2字符及其以后的ASCII代码为30H~39H、20H、00H以外。 • 使用DABIN(P)指令时,ASCII数据为-32768~32767以外。 • 使用DABIN(P)_U指令时,ASCII数据为0~65535以外。

*1 SM705(转换位数据切换)为OFF时,第1个字符的ASCII代码中无论设置为何皆不会变为错误。

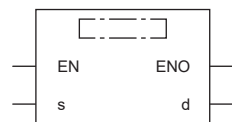
10进制ASCII数据→BIN32位数据转换

DDABIN(P) (_U)

将10进制ASCII数据转换为BIN32位数据。

梯形图	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



■执行条件

指令	执行条件
DDABIN DDABIN_U	
DDABINP DDABINP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	转换为BIN值的ASCII数据, 或存储了ASCII数据的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储转换结果的起始软元件	—	带符号BIN32位	ANY32_S
			无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	○	○	○	○	○	○	○	—	—	—	—

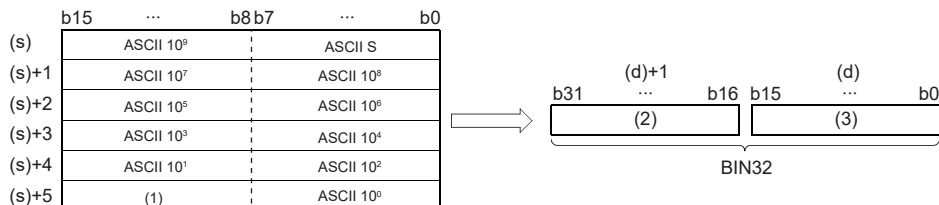
功能

- 将(s)中指定的软元件编号及其以后中存储的10进制ASCII数据转换为BIN32位数据后，存储到(d)中指定的软元件编号中。
- 根据SM705(变数位数切换)的状态，(s)中设置的10进制ASCII数据的设置方法将有所不同。

SM705的状态	(s)的设置方法	参阅目标
OFF	以固定位数(符号+数值部分10位)设置	442页 SM705(变数位数切换)为OFF时(s)的设置方法
ON	以任意位数(最大: 符号+数值部分10位)设置	443页 SM705(变数位数切换)为ON时(s)的设置方法

■SM705(变数位数切换)为OFF时(s)的设置方法

以固定位数将10进制ASCII数据设置到(s)~(s)+5中。



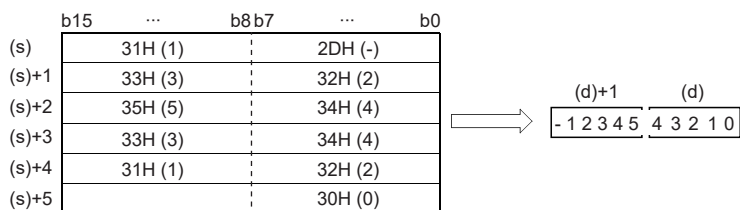
ASCII S: 符号的ASCII代码
 ASCII 10⁰: 个位的ASCII代码
 ASCII 10¹: 十位的ASCII代码
 ASCII 10²: 百位的ASCII代码
 ASCII 10³: 千位的ASCII代码
 ASCII 10⁴: 万位的ASCII代码
 ASCII 10⁵: 十万位的ASCII代码
 ASCII 10⁶: 百万位的ASCII代码
 ASCII 10⁷: 千万位的ASCII代码
 ASCII 10⁸: 亿位的ASCII代码
 ASCII 10⁹: 十亿位的ASCII代码

- (1) 忽略。
 (2) 高位16位
 (3) 低位16位

- (s)~(s)+5中指定的ASCII数据，在DDABIN(P)的情况下范围为-2147483648~2147483647，DDABIN(P)_U的情况下范围为0~4294967295。此外，(s)+5的高位字节及(s)+6及其以后中存储的数据将被忽略。
- 符号数据中，转换的数据为正时设置20H，为负时设置2DH。(设置了20H、2DH以外的情况下，将被作为正的数据处理。)
- 各位中设置的ASCII代码的范围为30H~39H。
- 各位中设置的ASCII代码为20H或00H时，将作为30H处理。

例

DDABIN(P)指令中，在(s)中设置“-1234543210”时



■SM705(变数位切换)为ON时(s)的设置方法

在(s)中设置任意位数(包含00H(NULL代码))的10进制ASCII数据。只是,整数部分为最大位数(10位)时,不用设置00H(NULL代码)。

(s)的设置方法如下所示。

设置于s中的值	(s)~(s)+5的内容	设置于s中的值	(s)~(s)+5的内容																																																																																				
<ul style="list-style-type: none"> 0 正的数(数值部分1位) 	<ul style="list-style-type: none"> 在(s)的高位字节中设置00H。 (s)+1及其以后将被忽略。 <table border="1"> <thead> <tr> <th></th> <th>b15</th> <th>...</th> <th>b8 b7</th> <th>...</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>(s)</td> <td colspan="2">00H</td> <td colspan="3">ASCII 10⁰</td> </tr> <tr> <td>(s)+1</td> <td colspan="5"></td> </tr> <tr> <td>(s)+2</td> <td colspan="5"></td> </tr> <tr> <td>(s)+3</td> <td colspan="5"></td> </tr> <tr> <td>(s)+4</td> <td colspan="5"></td> </tr> <tr> <td>(s)+5</td> <td colspan="5"></td> </tr> </tbody> </table>		b15	...	b8 b7	...	b0	(s)	00H		ASCII 10 ⁰			(s)+1						(s)+2						(s)+3						(s)+4						(s)+5						<ul style="list-style-type: none"> 正的数(数值部分2位) 负的数(数值部分1位) 	<ul style="list-style-type: none"> 在(s)+1的低位字节中设置00H。 (s)+1的高位字节及其以后将被忽略。 <table border="1"> <thead> <tr> <th></th> <th>b15</th> <th>...</th> <th>b8 b7</th> <th>...</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>(s)</td> <td colspan="2">ASCII 10⁰</td> <td colspan="3">ASCII 10¹ / 2DH (-)</td> </tr> <tr> <td>(s)+1</td> <td colspan="5">00H</td> </tr> <tr> <td>(s)+2</td> <td colspan="5"></td> </tr> <tr> <td>(s)+3</td> <td colspan="5"></td> </tr> <tr> <td>(s)+4</td> <td colspan="5"></td> </tr> <tr> <td>(s)+5</td> <td colspan="5"></td> </tr> </tbody> </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																																																																																							
<ul style="list-style-type: none"> 正的数(数值部分9位) 负的数(数值部分8位) 	<ul style="list-style-type: none"> 在(s)+4的高位字节中设置00H。 (s)+5将被忽略。 <table border="1"> <thead> <tr> <th></th> <th>b15</th> <th>...</th> <th>b8 b7</th> <th>...</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>(s)</td> <td colspan="2">ASCII 10⁷</td> <td colspan="3">ASCII 10⁸ / 2DH (-)</td> </tr> <tr> <td>(s)+1</td> <td colspan="2">ASCII 10⁵</td> <td colspan="3">ASCII 10⁶</td> </tr> <tr> <td>(s)+2</td> <td colspan="2">ASCII 10³</td> <td colspan="3">ASCII 10⁴</td> </tr> <tr> <td>(s)+3</td> <td colspan="2">ASCII 10¹</td> <td colspan="3">ASCII 10²</td> </tr> <tr> <td>(s)+4</td> <td colspan="2">00H</td> <td colspan="3">ASCII 10⁰</td> </tr> <tr> <td>(s)+5</td> <td colspan="5"></td> </tr> </tbody> </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						负的数(数值部分9位)	<ul style="list-style-type: none"> 在(s)+5的低位字节中设置00H。 (s)+5的高位字节将被忽略。 <table border="1"> <thead> <tr> <th></th> <th>b15</th> <th>...</th> <th>b8 b7</th> <th>...</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>(s)</td> <td colspan="2">ASCII 10⁸</td> <td colspan="3">2DH (-)</td> </tr> <tr> <td>(s)+1</td> <td colspan="2">ASCII 10⁶</td> <td colspan="3">ASCII 10⁷</td> </tr> <tr> <td>(s)+2</td> <td colspan="2">ASCII 10⁴</td> <td colspan="3">ASCII 10⁵</td> </tr> <tr> <td>(s)+3</td> <td colspan="2">ASCII 10²</td> <td colspan="3">ASCII 10³</td> </tr> <tr> <td>(s)+4</td> <td colspan="2">ASCII 10⁰</td> <td colspan="3">ASCII 10¹</td> </tr> <tr> <td>(s)+5</td> <td colspan="5">00H</td> </tr> </tbody> </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																																																																																						
正的数(数值部分10位)	(s)+5将被忽略。由于为最大位数,不用设置00H。 <table border="1"> <thead> <tr> <th></th> <th>b15</th> <th>...</th> <th>b8 b7</th> <th>...</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>(s)</td> <td colspan="2">ASCII 10⁸</td> <td colspan="3">ASCII 10⁹</td> </tr> <tr> <td>(s)+1</td> <td colspan="2">ASCII 10⁶</td> <td colspan="3">ASCII 10⁷</td> </tr> <tr> <td>(s)+2</td> <td colspan="2">ASCII 10⁴</td> <td colspan="3">ASCII 10⁵</td> </tr> <tr> <td>(s)+3</td> <td colspan="2">ASCII 10²</td> <td colspan="3">ASCII 10³</td> </tr> <tr> <td>(s)+4</td> <td colspan="2">ASCII 10⁰</td> <td colspan="3">ASCII 10¹</td> </tr> <tr> <td>(s)+5</td> <td colspan="5"></td> </tr> </tbody> </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						负的数(数值部分10位)	(s)+5的高位字节将被忽略。由于为最大位数,不用设置00H。 <table border="1"> <thead> <tr> <th></th> <th>b15</th> <th>...</th> <th>b8 b7</th> <th>...</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>(s)</td> <td colspan="2">ASCII 10⁹</td> <td colspan="3">2DH (-)</td> </tr> <tr> <td>(s)+1</td> <td colspan="2">ASCII 10⁷</td> <td colspan="3">ASCII 10⁸</td> </tr> <tr> <td>(s)+2</td> <td colspan="2">ASCII 10⁵</td> <td colspan="3">ASCII 10⁶</td> </tr> <tr> <td>(s)+3</td> <td colspan="2">ASCII 10³</td> <td colspan="3">ASCII 10⁴</td> </tr> <tr> <td>(s)+4</td> <td colspan="2">ASCII 10¹</td> <td colspan="3">ASCII 10²</td> </tr> <tr> <td>(s)+5</td> <td colspan="5">ASCII 10⁰</td> </tr> </tbody> </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: 个位的ASCII代码

ASCII 10¹: 十位的ASCII代码

⋮

ASCII 10⁹: 十亿位的ASCII代码

- (s)~(s)+5中指定的ASCII数据,在DDABIN(P)指令的情况下范围为-2147483648~2147483647,DDABIN(P)_U指令的情况下范围为0~4294967295。
- 转换的数据为负的数时,作为符号数据将2DH(-)设置到(s)+0的低位字节中。转换的数据为0或正的数时,不用设置符号数据,直接设置最高位位的ASCII代码。
- 各位中设置的ASCII代码的范围为30H~39H。
- 正的数中数值部分为10位时,(s)+5及其以后中存储的数据将被忽略。负的数中数值部分为10位时,(s)+5的高位字节及其以后中存储的数据将被忽略。
- 各位中设置的ASCII代码为20H时,将作为30H处理。00H时,将作为10进制ASCII数据的终端处理。
- 下述情况下,(d)中将存储0。
 - 第1字符为00H(NULL)
 - 第1字符为2DH(-)、第2字符为00H(NULL)

出错

错误代码 (SD0)	内容
3281H	(s)~(s)+5中输入了不能转换的非法数据时。 <ul style="list-style-type: none">• 第1字符的ASCII代码为2DH、30H~39H、20H、00H以外。*1• 第2字符及其以后的ASCII代码为30H~39H、20H、00H以外。• 使用DDABIN(P)指令时, ASCII数据为-2147483648~2147483647以外。• 使用DDABIN(P)_U指令时, ASCII数据为0~4294967295以外。

*1 SM705 (转换位数切换) 为OFF时, 第1个字符的ASCII代码中无论设置为何皆不会变为错误。

16进制ASCII数据→BIN16位数据转换

HABIN(P)

将16进制ASCII数据转换为BIN16位数据。

梯形图	ST
	ENO:=HABIN(EN, s, d); ENO:=HABINP(EN, s, d);

FBD/LD

■执行条件

指令	执行条件
HABIN	
HABINP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	转换为BIN值的ASCII数据, 或存储了ASCII数据的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储转换结果的软元件	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	○	—	—	—	—

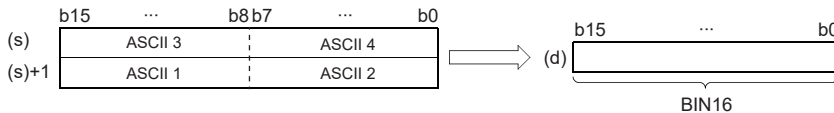
功能

- 将(s)中指定的软元件编号及其以后中存储的16进制ASCII数据转换为BIN16位数据后, 存储到(d)中指定的软元件中。
- 根据SM705(变数位数切换)的状态, (s)中设置的16进制ASCII数据的设置方法将有所不同。

SM705的状态	(s)的设置方法	参阅目标
OFF	以固定位数(4位)设置	446页 SM705(变数位数切换)为OFF时(s)的设置方法
ON	以任意位数(最大4位)设置	446页 SM705(变数位数切换)为ON时(s)的设置方法

■SM705(变数位数切换)为OFF时(s)的设置方法

- 以固定4位将16进制ASCII数据设置到(s)~(s)+1中。

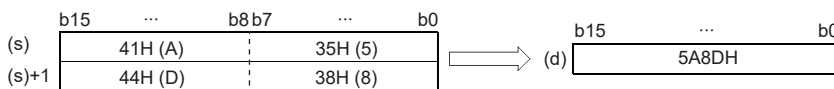


ASCII □: 第□位的ASCII代码

- (s)~(s)+1中指定的ASCII数据为0000H~FFFFH的范围内。
- (s)+2及其以后的数据将被忽略。
- 各位中设置的ASCII代码为30H~39H及41H~46H的范围内。

例

(s)中指定了5A8DH的情况下



■SM705(变数位数切换)为ON时(s)的设置方法

在(s)中设置任意位数(包含00H(NULL代码))的16进制ASCII数据。只是, 最大位数(4位)时, 不用设置00H(NULL代码)。

(s)的设置方法如下所示。

设置于s中的值	(s)~(s)+1的内容	设置于s中的值	(s)~(s)+1的内容
0H~FH	<ul style="list-style-type: none"> • 在(s)+0的高位字节中设置00H。 • (s)+1及其以后将被忽略。 	10H~FFH	<ul style="list-style-type: none"> • 在(s)+1的低位字节中设置00H。 • (s)+1的高位字节及其以后将被忽略。
100H~FFFH	<ul style="list-style-type: none"> • 在(s)+1的高位字节中设置00H。 • (s)+2及其以后将被忽略。 	1000H~FFFFH	(s)+2及其以后将被忽略。

ASCII □: 第□位的ASCII代码

- (s)~(s)+1中指定的ASCII数据为0000H~FFFFH的范围内。
- (s)+2及其以后的数据将被忽略。
- 各位中设置的ASCII代码为30H~39H及41H~46H的范围内。
- 各位中设置的ASCII代码为00H时, 将被作为16进制ASCII数据的终端处理。

出错

错误代码(SD0)	内容
3281H	(s)~(s)+1中输入了不能转换的非法数据时。 • 各位的ASCII代码为30H~39H、41H~46H以外。

16进制ASCII数据→BIN32位数据转换

DHABIN(P)

将16进制ASCII数据转换为BIN32位数据。

梯形图	ST
	ENO:=DHABIN(EN, s, d); ENO:=DHABINP(EN, s, d);

FBD/LD

■执行条件

指令	执行条件
DHABIN	
DHABINP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	转换为BIN值的ASCII数据, 或存储了ASCII数据的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储转换结果的起始软元件	—	带符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	○	○	○	○	○	○	○	—	—	—	—

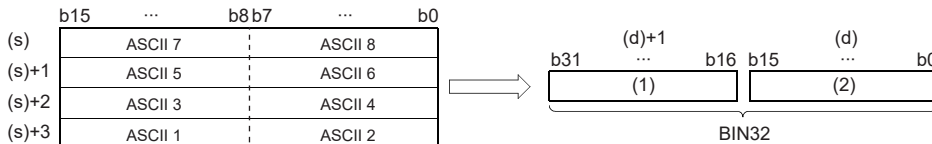
功能

- 将(s)中指定的软元件编号及其以后中存储的16进制ASCII数据转换为BIN32位数据后, 存储到(d)中指定的软元件编号中。
- 根据SM705(变数位数切换)的状态, (s)中设置的16进制ASCII数据的设置方法将有所不同。

SM705的状态	(s)的设置方法	参阅目标
OFF	以固定位数(8位)设置	448页 SM705(变数位数切换)为OFF时(s)的设置方法
ON	以任意位数(最大8位)设置	448页 SM705(变数位数切换)为ON时(s)的设置方法

■SM705(变数位数切换)为OFF时(s)的设置方法

- 以固定8位将16进制ASCII数据设置到(s)~(s)+3中。



ASCII □: 第□位的ASCII代码

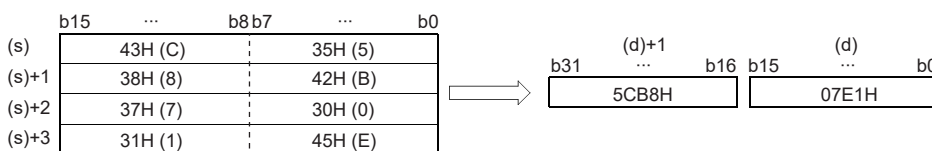
(1) 高位16位

(2) 低位16位

- (s)~(s)+3中指定的ASCII数据为00000000H~FFFFFFFH的范围内。
- (s)+4及其以后的数据将被忽略。
- 各位中设置的ASCII代码为30H~39H及41H~46H的范围内。

例

(s)中指定了5CB807E1H的情况下



■SM705(变数位数切换)为ON时(s)的设置方法

在(s)中设置任意位数(包含00H(NULL代码))的16进制ASCII数据。只是, 最大位数(8位)时, 不用设置00H(NULL代码)。

(s)的设置方法如下所示。

设置于s中的值	(s)~(s)+3的内容	设置于s中的值	(s)~(s)+3的内容																														
0H~FH	<ul style="list-style-type: none"> • 在(s)的高位字节中设置00H。 • (s)+1及其以后将被忽略。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8</td> <td>b7 ... b0</td> </tr> <tr> <td></td> <td>00H</td> <td>ASCII 1</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)	b15 ... b8	b7 ... b0		00H	ASCII 1	(s)+1			(s)+2			(s)+3			10H~FFH	<ul style="list-style-type: none"> • 在(s)+1的低位字节中设置00H。 • (s)+1的高位字节及其以后将被忽略。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8</td> <td>b7 ... b0</td> </tr> <tr> <td></td> <td>ASCII 1</td> <td>ASCII 2</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)	b15 ... b8	b7 ... b0		ASCII 1	ASCII 2	(s)+1		00H	(s)+2			(s)+3		
(s)	b15 ... b8	b7 ... b0																															
	00H	ASCII 1																															
(s)+1																																	
(s)+2																																	
(s)+3																																	
(s)	b15 ... b8	b7 ... b0																															
	ASCII 1	ASCII 2																															
(s)+1		00H																															
(s)+2																																	
(s)+3																																	
1000000H~FFFFFFFH	<ul style="list-style-type: none"> • 在(s)+3的高位字节中设置00H。 • (s)+4及其以后将被忽略。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8</td> <td>b7 ... b0</td> </tr> <tr> <td></td> <td>ASCII 6</td> <td>ASCII 7</td> </tr> <tr> <td>(s)+1</td> <td>ASCII 4</td> <td>ASCII 5</td> </tr> <tr> <td>(s)+2</td> <td>ASCII 2</td> <td>ASCII 3</td> </tr> <tr> <td>(s)+3</td> <td>00H</td> <td>ASCII 1</td> </tr> </table>	(s)	b15 ... b8	b7 ... b0		ASCII 6	ASCII 7	(s)+1	ASCII 4	ASCII 5	(s)+2	ASCII 2	ASCII 3	(s)+3	00H	ASCII 1	10000000H~FFFFFFFH	(s)+4及其以后将被忽略。 <table border="1"> <tr> <td>(s)</td> <td>b15 ... b8</td> <td>b7 ... b0</td> </tr> <tr> <td></td> <td>ASCII 7</td> <td>ASCII 8</td> </tr> <tr> <td>(s)+1</td> <td>ASCII 5</td> <td>ASCII 6</td> </tr> <tr> <td>(s)+2</td> <td>ASCII 3</td> <td>ASCII 4</td> </tr> <tr> <td>(s)+3</td> <td>ASCII 1</td> <td>ASCII 2</td> </tr> </table>	(s)	b15 ... b8	b7 ... b0		ASCII 7	ASCII 8	(s)+1	ASCII 5	ASCII 6	(s)+2	ASCII 3	ASCII 4	(s)+3	ASCII 1	ASCII 2
(s)	b15 ... b8	b7 ... b0																															
	ASCII 6	ASCII 7																															
(s)+1	ASCII 4	ASCII 5																															
(s)+2	ASCII 2	ASCII 3																															
(s)+3	00H	ASCII 1																															
(s)	b15 ... b8	b7 ... b0																															
	ASCII 7	ASCII 8																															
(s)+1	ASCII 5	ASCII 6																															
(s)+2	ASCII 3	ASCII 4																															
(s)+3	ASCII 1	ASCII 2																															

ASCII □: 第□位的ASCII代码

- (s)~(s)+3中指定的ASCII数据为00000000H~FFFFFFFH的范围内。
- (s)+4及其以后的数据将被忽略。
- 各位中设置的ASCII代码为30H~39H及41H~46H的范围内。
- 各位中设置的ASCII代码为00H时, 将被作为16进制ASCII数据的终端处理。

错误代码 (SD0)	内容
3281H	(s)~(s)+3中输入了不能转换的非法数据时。 • 各位的ASCII代码为30H~39H、41H~46H以外。

10进制ASCII数据→BCD4位数据转换

DABCD(P)

将10进制ASCII数据转换为BCD4位数据。

<p>梯形图</p>	<p>ST</p> <p>ENO:=DABCD(EN, s, d); ENO:=DABCDP(EN, s, d)</p>
-------------------	---

<p>FBD/LD</p>

■执行条件

指令	执行条件
DABCD	
DABCDP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	转换为BCD值的ASCII数据，或存储了ASCII数据的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储转换结果的软元件	—	BCD4位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	○	—	—	—	—

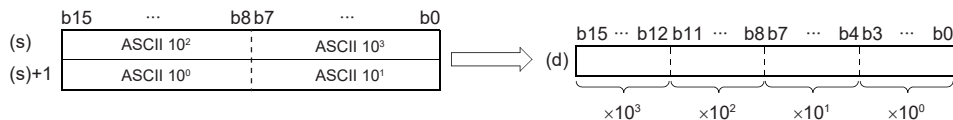
功能

- 将(s)中指定的软元件编号及其以后中存储的10进制ASCII数据转换为BCD4位数据后, 存储到(d)中指定的软元件中。
- 根据SM705(变数位数切换)的状态, (s)中设置的10进制ASCII数据的设置方法将有所不同。

SM705的状态	(s)的设置方法	参阅目标
OFF	以固定位数(4位)设置	451页 SM705(变数位数切换)为OFF时(s)的设置方法
ON	以任意位数(最大4位)设置	451页 SM705(变数位数切换)为ON时(s)的设置方法

■SM705(变数位数切换)为OFF时(s)的设置方法

- 以固定4位将10进制ASCII数据设置到(s)~(s)+1中。

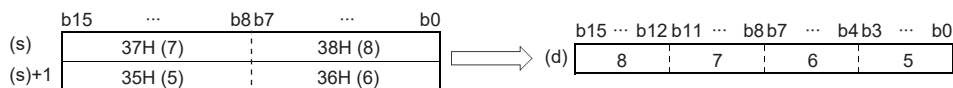


ASCII 10³: 千位的ASCII代码
 ASCII 10²: 百位的ASCII代码
 ASCII 10¹: 十位的ASCII代码
 ASCII 10⁰: 个位的ASCII代码

- (s)~(s)+1中指定的ASCII数据为0~9999的范围内。
- (s)+2及其以后的数据将被忽略。
- 各位中设置的ASCII代码的范围为30H~39H。
- 各位中设置的ASCII代码为20H或00H时, 将作为30H处理。

例

(s)中指定了8765的情况下



■SM705(变数位数切换)为ON时(s)的设置方法

在(s)中设置任意位数(包含00H(NULL代码))的10进制ASCII数据。只是, 最大位数(4位)时, 不用设置00H(NULL代码)。

(s)的设置方法如下所示。

设置于s中的值	(s)~(s)+1的内容	设置于s中的值	(s)~(s)+1的内容
0~9	<ul style="list-style-type: none"> • 在(s)+0的高位字节中设置00H。 • (s)+1及其以后将被忽略。 	10~99	<ul style="list-style-type: none"> • 在(s)+1的低位字节中设置00H。 • (s)+1的高位字节及其以后将被忽略。
100~999	<ul style="list-style-type: none"> • 在(s)+1的高位字节中设置00H。 • (s)+2及其以后将被忽略。 	1000~9999	(s)+2及其以后将被忽略。

ASCII 10³: 千位的ASCII代码
 ASCII 10²: 百位的ASCII代码
 ASCII 10¹: 十位的ASCII代码
 ASCII 10⁰: 个位的ASCII代码

- (s)~(s)+1中指定的ASCII数据为0~9999的范围内。
- (s)+2及其以后的数据将被忽略。
- 各位中设置的ASCII代码的范围为30H~39H。
- 各位中设置的ASCII代码为20H时, 将作为30H处理。00H时, 将作为10进制ASCII数据的终端处理。

出错

错误代码 (SD0)	内容
3281H	(s) 中输入了不能转换的非法数据时。 <ul style="list-style-type: none">• 数据中有0~9以外的字符。

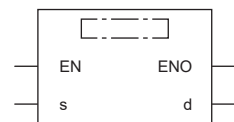
10进制ASCII数据→BCD8位数据转换

DDABCD (P)

将10进制ASCII数据转换为BCD8位数据。

梯形图	ST
	ENO:=DDABCD (EN, s, d) ; ENO:=DDABCDP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
DDABCD	
DDABCDP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	转换为BCD值的ASCII数据, 或存储了ASCII数据的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储转换结果的起始软元件	—	BCD8位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	○	○	○	○	○	○	○	○	—	—	—

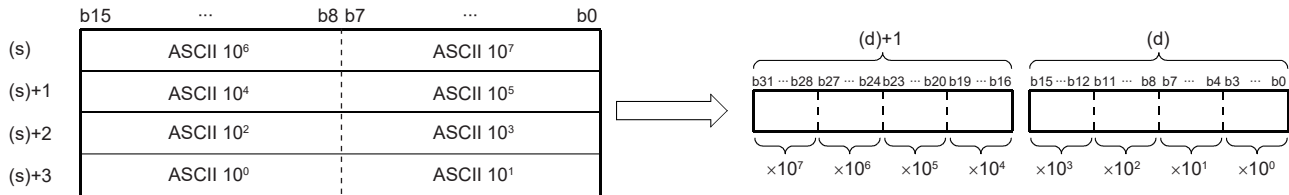
功能

- 将(s)中指定的软元件编号及其以后中存储的10进制ASCII数据转换为BCD8位数据后，存储到(d)中指定的软元件编号中。
- 根据SM705(变数位数切换)的状态，(s)中设置的10进制ASCII数据的设置方法将有所不同。

SM705的状态	(s)的设置方法	参阅目标
OFF	以固定位数(8位)设置	454页 SM705(变数位数切换)为OFF时(s)的设置方法
ON	以任意位数(最大8位)设置	455页 SM705(变数位数切换)为ON时(s)的设置方法

■SM705(变数位数切换)为OFF时(s)的设置方法

- 以固定8位将10进制ASCII数据设置到(s)~(s)+3中。

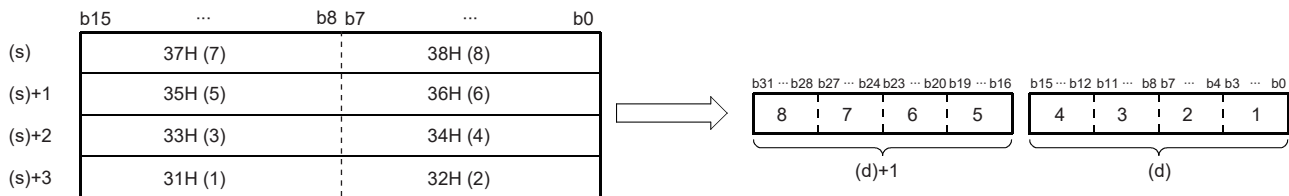


ASCII 10⁷: 千万位的ASCII代码
 ASCII 10⁶: 百万位的ASCII代码
 ASCII 10⁵: 十万位的ASCII代码
 ASCII 10⁴: 万位的ASCII代码
 ASCII 10³: 千位的ASCII代码
 ASCII 10²: 百位的ASCII代码
 ASCII 10¹: 十位的ASCII代码
 ASCII 10⁰: 个位的ASCII代码

- (s)~(s)+3中指定的ASCII数据为0~99999999的范围内。
- (s)+4及其以后的数据将被忽略。
- 各位中设置的ASCII代码的范围为30H~39H。
- 各位中设置的ASCII代码为20H或00H时，将作为30H处理。

例

(s)中指定了87654321的情况下



■SM705(变数位切换)为ON时(s)的设置方法

在(s)中设置任意位数(包含00H(NULL代码))的10进制ASCII数据。只是,最大位数(8位)时,不用设置00H(NULL代码)。

(s)的设置方法如下所示。

设置于s中的值	(s)~(s)+3的内容	设置于s中的值	(s)~(s)+3的内容																																																												
0~9	<ul style="list-style-type: none"> 在(s)的高位字节中设置00H。 (s)+1及其以后将被忽略。 <table border="1"> <tr> <td></td> <td>b15</td> <td>...</td> <td>b8b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(s)</td> <td colspan="2">00H</td> <td colspan="3">ASCII 10⁰</td> </tr> <tr> <td>(s)+1</td> <td colspan="5">[Greyed out]</td> </tr> <tr> <td>(s)+2</td> <td colspan="5">[Greyed out]</td> </tr> <tr> <td>(s)+3</td> <td colspan="5">[Greyed out]</td> </tr> </table>		b15	...	b8b7	...	b0	(s)	00H		ASCII 10 ⁰			(s)+1	[Greyed out]					(s)+2	[Greyed out]					(s)+3	[Greyed out]					10~99	<ul style="list-style-type: none"> 在(s)+1的低位字节中设置00H。 (s)+1的高位字节及其以后将被忽略。 <table border="1"> <tr> <td></td> <td>b15</td> <td>...</td> <td>b8b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(s)</td> <td colspan="2">ASCII 10⁰</td> <td colspan="3">ASCII 10¹</td> </tr> <tr> <td>(s)+1</td> <td colspan="5">[Greyed out]</td> </tr> <tr> <td>(s)+2</td> <td colspan="5">[Greyed out]</td> </tr> <tr> <td>(s)+3</td> <td colspan="5">[Greyed out]</td> </tr> </table>		b15	...	b8b7	...	b0	(s)	ASCII 10 ⁰		ASCII 10 ¹			(s)+1	[Greyed out]					(s)+2	[Greyed out]					(s)+3	[Greyed out]				
	b15	...	b8b7	...	b0																																																										
(s)	00H		ASCII 10 ⁰																																																												
(s)+1	[Greyed out]																																																														
(s)+2	[Greyed out]																																																														
(s)+3	[Greyed out]																																																														
	b15	...	b8b7	...	b0																																																										
(s)	ASCII 10 ⁰		ASCII 10 ¹																																																												
(s)+1	[Greyed out]																																																														
(s)+2	[Greyed out]																																																														
(s)+3	[Greyed out]																																																														
⋮																																																															
1000000~9999999	<ul style="list-style-type: none"> 在(s)+3的高位字节中设置00H。 (s)+4及其以后将被忽略。 <table border="1"> <tr> <td></td> <td>b15</td> <td>...</td> <td>b8b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(s)</td> <td colspan="2">ASCII 10⁵</td> <td colspan="3">ASCII 10⁶</td> </tr> <tr> <td>(s)+1</td> <td colspan="2">ASCII 10³</td> <td colspan="3">ASCII 10⁴</td> </tr> <tr> <td>(s)+2</td> <td colspan="2">ASCII 10¹</td> <td colspan="3">ASCII 10²</td> </tr> <tr> <td>(s)+3</td> <td colspan="2">00H</td> <td colspan="3">ASCII 10⁰</td> </tr> </table>		b15	...	b8b7	...	b0	(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	<ul style="list-style-type: none"> (s)+4及其以后将被忽略。 <table border="1"> <tr> <td></td> <td>b15</td> <td>...</td> <td>b8b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td>(s)</td> <td colspan="2">ASCII 10⁶</td> <td colspan="3">ASCII 10⁷</td> </tr> <tr> <td>(s)+1</td> <td colspan="2">ASCII 10⁴</td> <td colspan="3">ASCII 10⁵</td> </tr> <tr> <td>(s)+2</td> <td colspan="2">ASCII 10²</td> <td colspan="3">ASCII 10³</td> </tr> <tr> <td>(s)+3</td> <td colspan="2">ASCII 10⁰</td> <td colspan="3">ASCII 10¹</td> </tr> </table>		b15	...	b8b7	...	b0	(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		
	b15	...	b8b7	...	b0																																																										
(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																																																												
	b15	...	b8b7	...	b0																																																										
(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: 千万位的ASCII代码

ASCII 10⁶: 百万位的ASCII代码

ASCII 10⁵: 十万位的ASCII代码

ASCII 10⁴: 万位的ASCII代码

ASCII 10³: 千位的ASCII代码

ASCII 10²: 百位的ASCII代码

ASCII 10¹: 十位的ASCII代码

ASCII 10⁰: 个位的ASCII代码

- (s)~(s)+3中指定的ASCII数据为0~99999999的范围内。
- (s)+4及其以后的数据将被忽略。
- 各位中设置的ASCII代码的范围为30H~39H。
- 各位中设置的ASCII代码为20H时,将作为30H处理。00H时,将作为10进制ASCII数据的终端处理。

出错

错误代码(SD0)	内容
3281H	(s)中输入了不能转换的非法数据时。 <ul style="list-style-type: none"> 数据中有0~9以外的字符。

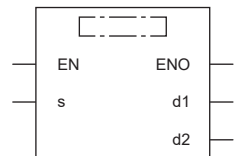
10进制字符串→BIN16位数据转换

VAL(P) (_U)

将字符串转换为BIN16位数据。

梯形图	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



■执行条件

指令	执行条件
VAL VAL_U	
VALP VALP_U	

设置数据

■内容、范围、数据类型

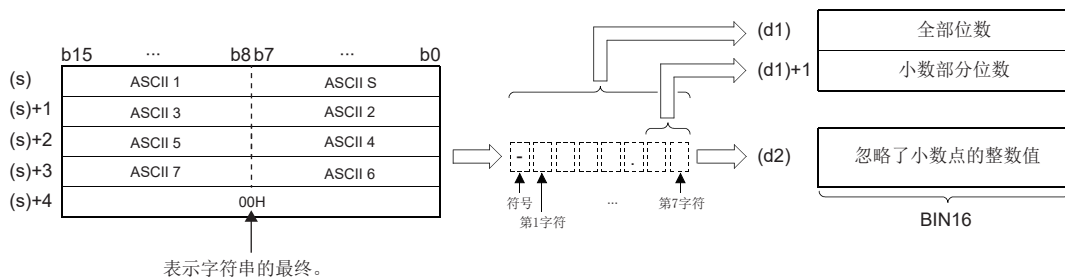
操作数	内容	范围	数据类型	数据类型(标签)
(s)	转换为BIN数据的字符串或存储了字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(d1)	VAL(P)	—	带符号BIN16位	ANY16_S_ARRAY (要素数: 2)
	VAL(P)_U		无符号BIN16位	ANY16_U_ARRAY (要素数: 2)
(d2)	VAL(P)	—	带符号BIN16位	ANY16_S
	VAL(P)_U		无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—	

功能

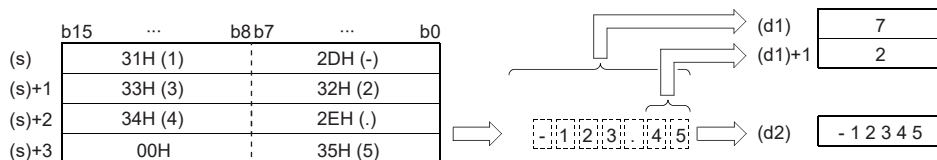
- 将(s)中指定的软件编号及其以后中存储的字符串转换为BIN16位数据后，将位数存储到(d1)中，将BIN数据存储在(d2)中。在字符串→BIN转换中，将从(s)中指定的软件编号开始至存储了00H的软件编号为止的数据作为字符串处理。



ASCII S: 符号的ASCII代码
 ASCII □: 第□字符的ASCII代码

例

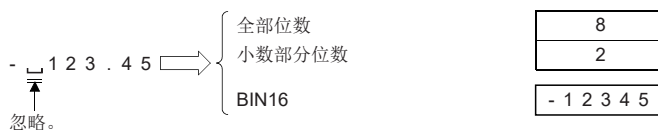
(s)及其以后指定了“-123.45”的字符串的情况下(指定了带符号的情况下)



- (s)中指定的字符串的全部字符数为2~8字符。
- (s)中指定的字符串中，小数部分的字符数为0~5字符。但是，指定时，不应超过“全部位数-3”。
- 可转换为BIN值的数值的字符串的范围，在以忽略小数点的值指定了带符号的情况下范围为-32768~32767，指定了无符号的情况下范围为0~65535。此外，符号及小数点除外的数值的字符串，只能在30H~39H的范围内指定。(忽略小数点的值...“-12345.6”的情况下，将变为“-123456”。)
- 表示正数值的情况下在符号中设置20H，表示负数值的情况下设置2DH。
- 小数点中设置2EH。
- (d1)中存储的全部位数存储表示数值的所有字符(包括符号、小数点)的数。(d1)+1中存储的小数位数存储表示2EH(.)以后的小数部的字符数。(d2)中存储的BIN16位数据，将忽略小数点的字符串转换为BIN值后存储。
- 在(s)中指定的字符串中，在符号与首个0以外的数值之间，存在有20H(空格)或30H(0)的情况下，将忽略20H、30H而转换为BIN值。

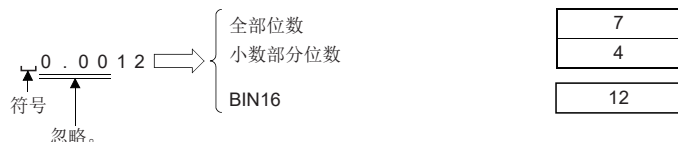
例

符号与首个0以外的数值之间存在有20H的情况下(指定了带符号的情况下)



例

符号与首个0以外的数值之间存在有30H的情况下



出错

错误代码 (SD0)	内容
2820H	从 (s) 中指定的软元件编号开始至相应软元件的最终软元件编号为止，未设置00H时。
3281H	(s) 中输入了不能转换的非法数据时。 <ul style="list-style-type: none">• 字符数为2~8以外。• 小数部字符数为0~5以外。• 全部字符数与小数部字符数的关系超出了下述范围。 全部字符数-3≥小数部字符数• 使用VAL(P)指令时，符号中设置了20H、2DH以外的ASCII代码。• 使用VAL(P)_U指令时，符号中设置了20H以外的ASCII代码。• 各数字的位中设有30H~39H及2EH(小数点)以外的ASCII代码。• 设置了多个小数点。 转换的BIN值的值超过了各指令中可转换的范围时。
3285H	(s) 中指定的字符串的字符数超过了16383字符时。

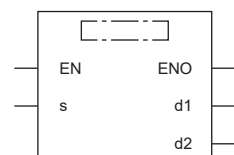
10进制字符串→BIN32位数据转换

DVAL(P) (_U)

将字符串转换为BIN32位数据。

梯形图	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



■执行条件

指令	执行条件
DVAL DVAL_U	
DVALP DVALP_U	

设置数据

■内容、范围、数据类型

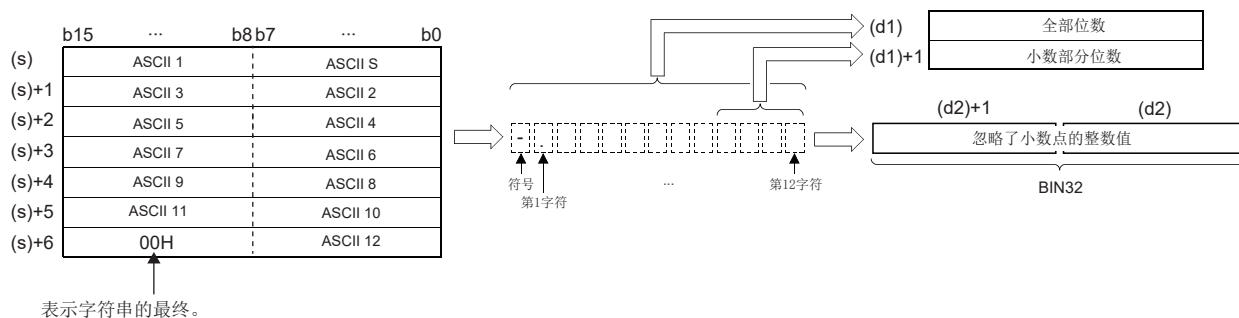
操作数	内容	范围	数据类型	数据类型(标签)
(s)	转换为BIN数据的字符串或存储了字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(d1)	DVAL(P)	—	带符号BIN16位	ANY16_S_ARRAY (要素数: 2)
	DVAL(P)_U		无符号BIN16位	ANY16_U_ARRAY (要素数: 2)
(d2)	DVAL(P)	—	带符号BIN32位	ANY32_S
	DVAL(P)_U		无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—

功能

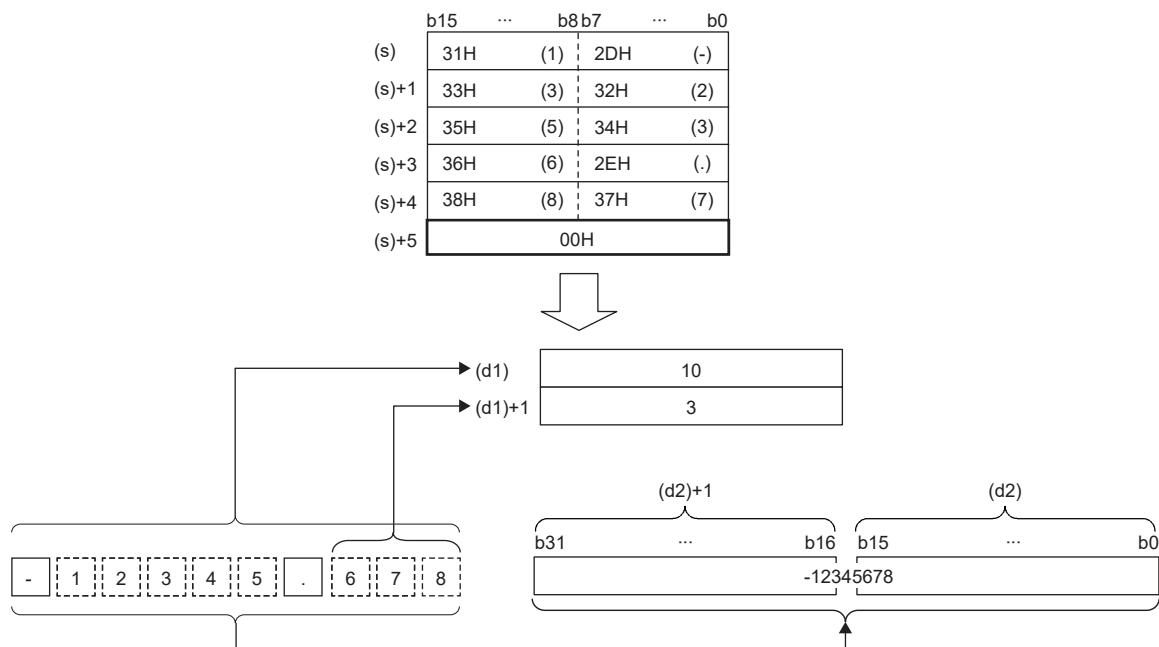
- 将(s)中指定的软元件编号及其以后中存储的字符串转换为BIN32位数据后，将位数存储到(d1)中，将BIN数据存储在(d2)中。在字符串→BIN转换中，将从(s)中指定的软元件编号开始至存储了00H的软元件编号为止的数据作为字符串处理。



ASCII S: 符号的ASCII代码
ASCII □: 第□字符的ASCII代码

例

(s)及其以后指定了“-12345.678”的字符串的情况下(指定了带符号的情况下)

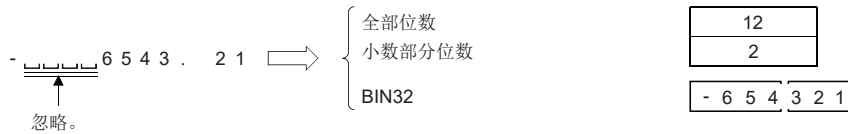


- (s)中指定的字符串的全部字符数为2~13字符。
- (s)中指定的字符串中，小数部分的字符数为0~10字符。但是，指定时，不应超过“全部位数-3”。
- 可转换为BIN值的数值的字符串的范围如下所示。此外，符号及小数点除外的数值的字符串，只能在30H~39H的范围内指定。(忽略小数点的值...“-12345.6”的情况下，将变为“-123456”。)
- 以忽略小数点的值指定了带符号的情况下：-2147483648~2147483647
- 以忽略小数点的值指定了无符号的情况下：0~4294967295
- 表示正数值的情况下在符号中设置20H，表示负数值的情况下设置2DH。
- 小数点中设置2EH。
- (d1)中存储的全部位数存储表示数值的所有字符(包括符号、小数点)的数。(d1)+1中存储的小数部位数存储表示2EH(.)以后的小数部的字符数。(d2)中存储的BIN32位数据，将忽略小数点的字符串转换为BIN值后存储。

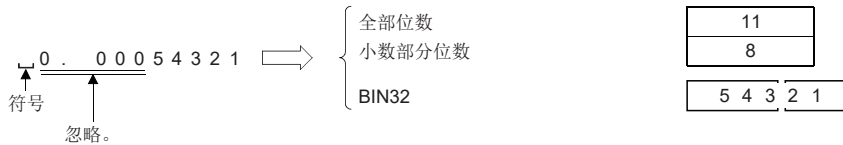
- 在(s)中指定的字符串中，在符号与首个0以外的数值之间，存在有20H(空格)或30H(0)的情况下，将忽略20H、30H而转换为BIN值。

例

符号与首个0以外的数值之间存在有20H的情况下(指定了带符号的情况下)

**例**

符号与首个“0”以外的数值之间存在有30H的情况下

**出错**

错误代码(SD0)	内容
2820H	从(s)中指定的软件编号开始至相应软件件的最终软件编号为止，未设置00H时。
3281H	<p>(s)中输入了不能转换的非法数据时。</p> <ul style="list-style-type: none"> 字符串的字符数为2~13以外。 字符串的小数部字符数为0~10以外。 全部字符数与小数部字符数的关系超出了下述范围。 <p>全部字符数-3≥小数部字符数</p> <ul style="list-style-type: none"> 使用DVAL(P)指令时，符号中设置了20H、2DH以外的ASCII代码。 使用DVAL(P)_U指令时，符号中设置了20H以外的ASCII代码。 各数字的位中设有30H~39H及2EH(小数点)以外的ASCII代码。 设置了多个小数点。 <p>转换的BIN值的值超过了各指令中可转换的范围时。</p>
3285H	(s)中指定的字符串的字符数超过了16383字符时。

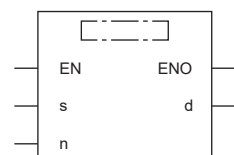
16进制ASCII→16进制数据BIN转换

ASC2INT (P)

将16进制ASCII数据转换为BIN值。

梯形图	ST
	ENO:=ASC2INT (EN, s, n, d) ; ENO:=ASC2INTP (EN, s, n, d) ;

FBD/LD



■执行条件

指令	执行条件
ASC2INT	
ASC2INTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s)	存储了转换为BIN数据的字符串的起始软元件	—	字符串	ANYSTRING_SINGLE* ¹
(d)	存储转换后的BIN数据的起始软元件	—	带符号BIN16位	ANY16* ¹
(n)	存储的字符数	0~16383	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

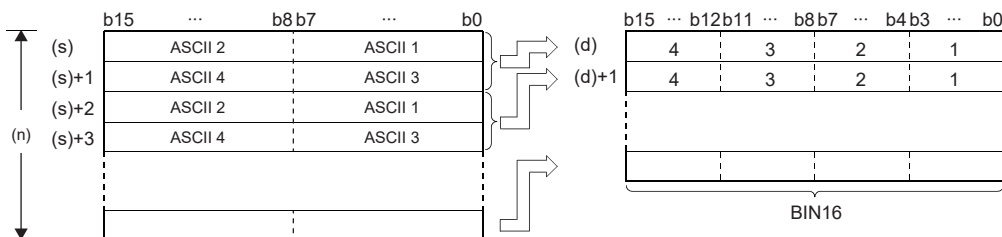
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 将(s)中指定的软元件编号及其以后，(n)中指定的字符数中存储的16进制ASCII数据转换为BIN值后，存储到(d)中指定的软元件编号及其以后。

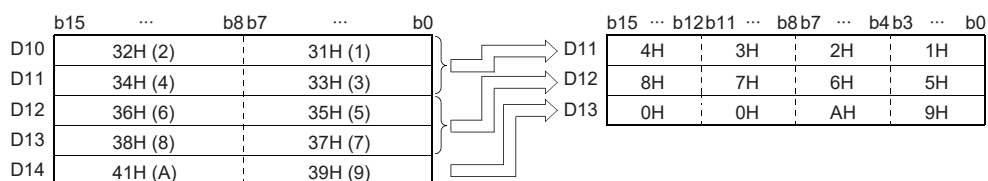


ASCII □: 第□位的ASCII代码

1~4: 第1~4位

(n): (n)中指定的字符数

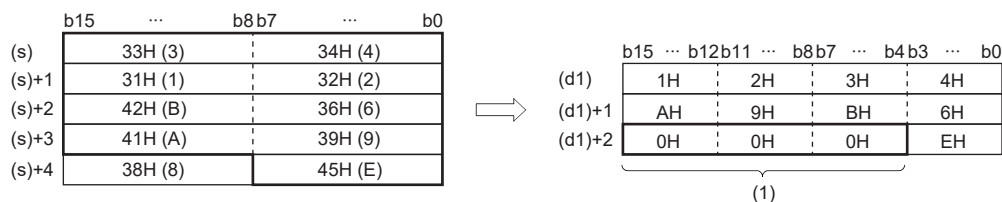
- (n)中设置字符数时，(s)中指定的字符串的范围及(d)中指定的存储BIN数据的软元件的范围将自动确定。
- 即使存储要转换的ASCII数据的软元件范围与存储转换后的BIN数据的软元件范围重复的情况下，也将作为正常处理。



- (n)中指定的字符数不是4的倍数的情况下，存储转换后BIN值的软元件编号中的最终软元件编号的指定字符数以后的位数将自动存储0H。

例

(n)的字符数为9的情况下



(1) 自动存储0H。

- (n)中指定的字符数为“0”的情况下，不进行转换处理。
- (s)中可指定的ASCII代码为30H~39H、41H~46H的范围内。

出错

错误代码(SD0)	内容
3281H	(s)中设置了16进制数值的字符串以外的字符(30H~39H、41H~46H以外的ASCII代码)。
3285H	(n)中输入了超出可指定范围的数据时。 <ul style="list-style-type: none"> 指定的字符数超出了0~16383的范围。

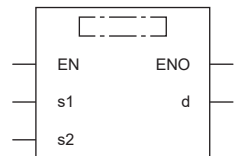
单精度实数→BCD分解

EMOD (P)

将单精度实数数据分解为BCD型浮点格式。

梯形图	ST
	ENO:=EMOD (EN, s1, s2, d) ; ENO:=EMODP (EN, s1, s2, d) ;

FBD/LD



■执行条件

指令	执行条件
EMOD	
EMODP	

设置数据

■内容、范围、数据类型

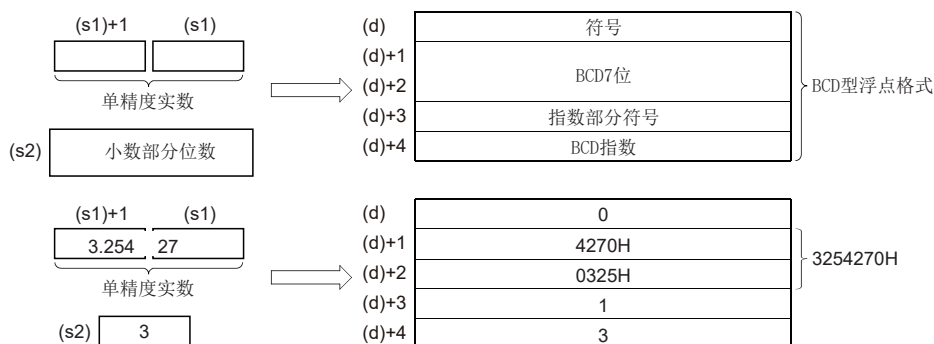
操作数	内容	范围	数据类型	数据类型 (标签)
(s1)	单精度实数数据或存储了单精度实数数据的起始软元件	$0 \cdot 2^{-126} \leq (s1) < 2^{128}$	单精度实数	ANYREAL_32
(s2)	小数部位数据	0~7	带符号BIN16位	ANY16
(d)	存储BCD分解后的数据的起始软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 5)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	—	—	○	—	—	—	—	○	—	—	—

功能

- 将(s1)中指定的软件中存储的单精度实数数据，分解为基于(s2)中指定的软件中存储的小数部位数的BCD型浮点格式后，存储到(d)中指定的软件编号及其以后。

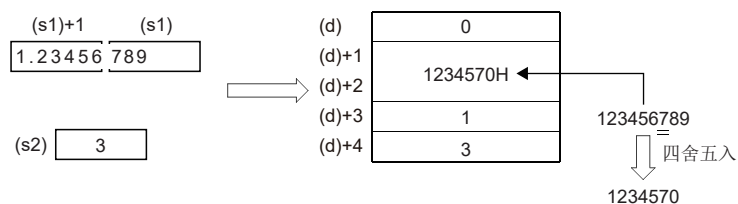


- (d)的符号及(d)+3的指数部符号中，正时存储0，负时存储1。
- (d)+4的BCD指数中存储0~38。
- 对于(s2)指定(s1)的单精度实数数据的小数部位数。在上图的示例中，变为以下情况。

3.25427

 (s2)=3

- (d)+1、(d)+2中存储的BCD的有效位为对第7位进行了四舍五入后的6位。



- (s2)的小数部位数中可设置0~7。
- 通过工程工具设置输入值的情况下，有可能会发生化整误差。关于通过工程工具设置输入值时的注意事项，请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3281H	(s2)中指定的小数部位数超出了0~7的范围时。
3282H	(s1)中设置的软件、标签的内容为-0、非正规化数、非数、±∞时。

BIN16位数据2的补数(符号取反)

NEG(P)

对BIN16位软元件的符号进行取反。

梯形图	ST
	ENO:=NEG(EN, d); ENO:=NEGP(EN, d);

FBD/LD



■执行条件

指令	执行条件
NEG	
NEGP	

设置数据

■内容、范围、数据类型

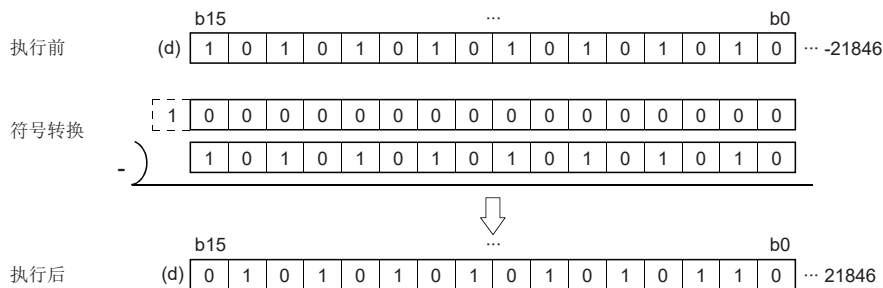
操作数	内容	范围	数据类型	数据类型(标签)
(d)	存储进行2的补数的数据的软元件	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 对(d)中指定的BIN16位软元件的符号进行取反后，存储到(d)中指定的软元件中。
- 用于对正负的符号进行取反。



出错

没有运算错误。

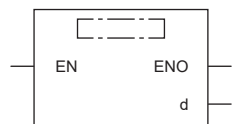
BIN32位数据2的补数(符号取反)

DNEG (P)

对BIN32位软元件的符号进行取反。

梯形图	ST
	ENO:=DNEG (EN, d); ENO:=DNEG (EN, d);

FBD/LD



■执行条件

指令	执行条件
DNEG	
DNEGP	

设置数据

■内容、范围、数据类型

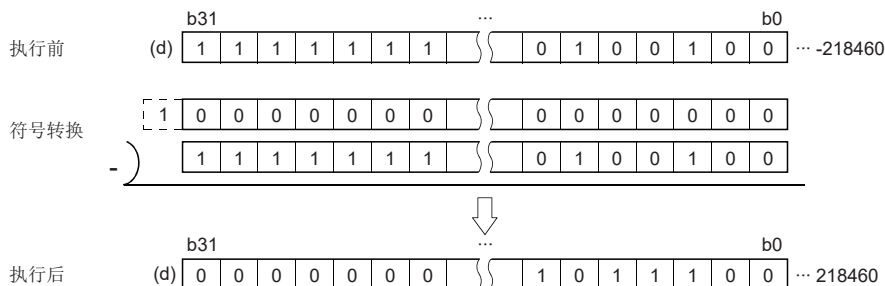
操作数	内容	范围	数据类型	数据类型(标签)
(d)	存储进行2的补数的数据的起始软元件	—	带符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	○	○	○	—	—	—	—

功能

- 对(d)中指定的BIN32位软元件的符号进行取反后, 存储到(d)中指定的软元件中。
- 用于对正负的符号进行取反。



出错

没有运算错误。

8→256位解码

DECO(P)

对指定的软元件的低位(n)位进行解码。

梯形图	ST
	ENO:=DECO(EN, s, n, d); ENO:=DECOP(EN, s, n, d);

FBD/LD

■执行条件

指令	执行条件
DECO	
DECOP	

设置数据

■内容、范围、数据类型

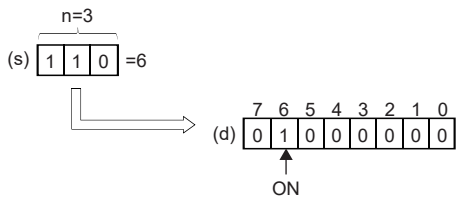
操作数	内容	范围	数据类型	数据类型(标签)
(s)	解码数据或存储了解码数据的软元件编号	—	位/带符号BIN16位	ANY_ELEMENTARY
(d)	存储解码结果的软元件	—	位/字	ANY_ELEMENTARY
(n)	有效位长	1~8	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 将(s)的低位(n)位中指定的BIN值对应的(d)的位的位置置为ON。



- (n)可指定1~8。
- (n)=0时将变为无处理，(d)中指定的软元件的内容不变化。
- 位软元件作为1位处理，字软元件作为16位处理。

出错

错误代码(SD0)	内容
3281H	(n)为0~8以外时。

256→8位编码

ENCO(P)

对2的(n)次方的数据进行编码。

梯形图	ST
	ENO:=ENCO(EN, s, n, d); ENO:=ENCOP(EN, s, n, d);

FBD/LD

■执行条件

指令	执行条件
ENCO	
ENCOP	

设置数据

■内容、范围、数据类型

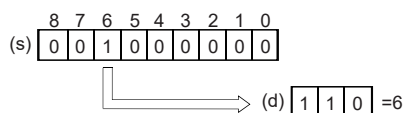
操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储编码数据的软元件	—	位/字	ANY_ELEMENTARY
(d)	存储编码结果的软元件编号	—	位/带符号BIN16位	ANY_ELEMENTARY
(n)	有效位长	1~8	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	—	○	○	—	—	—	—

功能

- 将(s)的2⁽ⁿ⁾位的数据开始处于1的位所对应的BIN值，存储到(d)中。



- (n)可指定1~8。
- (n)=0时将变为无处理，(d)中指定的软元件的内容不变化。
- 位软元件作为1位处理，字软元件作为16位处理。
- 多个位为1时以高位的位位置进行处理。

错误代码 (SD0)	内容
3281H	(n) 为0~8以外时。 从 (s) 开始的 $2^{(n)}$ 位的数据全部为0时。

7段解码

SEG (P)

将软元件的低位4位中指定的0~F的数据解码为7段显示数据。

梯形图	ST
	ENO:=SEG (EN, s, d) ; ENO:=SEGP (EN, s, d) ;

FBD/LD

■执行条件

指令	执行条件
SEG	
SEGP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	解码数据或存储了解码数据的软元件	-32768~32767	带符号BIN16位	ANY16
(d)	存储解码结果的软元件	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

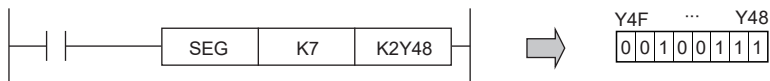
操作数	位		字			双字		间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s)	○	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 将(s)的低位4位中指定的0~F的数据解码为7段显示数据后, 存储到(d)中。
- 位软元件时(d)表示存储7段显示数据的起始软元件, 字软元件中存储的软元件编号如下所示。

例

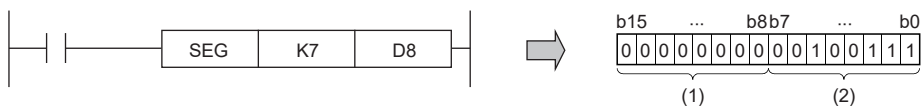
位软元件的情况下



Y48~Y4F在输出下一个数据之前不变化。

例

字软元件的情况下



- (1) 高位8位必定变为0。
- (2) 低位8位中存储7段显示数据。

- 7段解码表如下所示。

(s)		7段的构成	(d)								显示数据
16进制数	位模式		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

出错

没有运算错误。

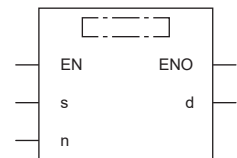
BIN16位数据的4位分离

DIS(P)

将指定的BIN16位数据的低位 (n) 位数的数据存储到指定的软元件中。

梯形图	ST
	ENO:=DIS (EN, s, n, d) ; ENO:=DISP (EN, s, n, d) ;

FBD/LD



■执行条件

指令	执行条件
DIS	
DISP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s)	存储分离的数据的软元件	—	带符号BIN16位	ANY16
(d)	存储分离后数据的起始软元件	—	带符号BIN16位	ANY16*1
(n)	分离数	1~4	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

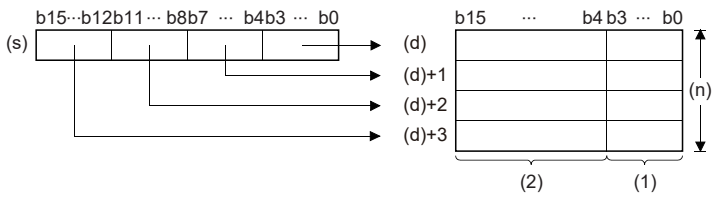
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 将(s)中指定的BIN16位数据的低位(n)位(1位4位)的数据存储到(d)中指定的软元件开始的(n)点的低位4位中。



- (1) 存储区域
(2) 存储0。

- (d)中指定的软元件开始的(n)点的高位12位将变为0。
- (n)可指定1~4。
- (n)=0时将变为无处理，(d)的软元件开始的(n)点的内容不变化。

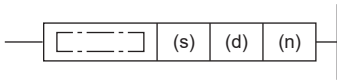
出错

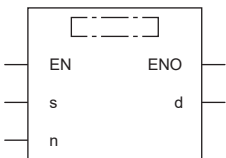
错误代码(SD0)	内容
3281H	(n)为0~4以外时。

BIN16位数据的4位合并



UNI (P)

将指定的软元件开始的 (n) 点的BIN16位数据的低位4位合并到16位软元件中。

梯形图	ST
	ENO:=UNI (EN, s, n, d) ; ENO:=UNIP (EN, s, n, d) ;

FBD/LD


■执行条件

指令	执行条件
UNI	
UNIP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s)	存储合并的数据的起始软元件	—	带符号BIN16位	ANY16*1
(d)	存储合并后的数据的软元件	—	带符号BIN16位	ANY16
(n)	合并数	1~4	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

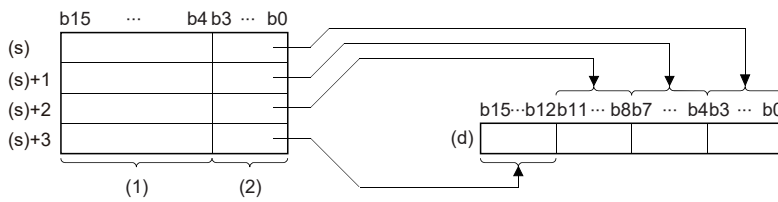
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 将(s)中指定的软元件开始的(n)点的BIN16位数据的低位4位，合并到(d)中指定的BIN16位软元件中。



(1) 忽略。

(2) 合并数据

- (d)中指定的软元件的高位(4-n)位的位将变为0。
- (n)可指定1~4。
- (n)=0时将变为无处理，(d)的软元件的内容不变化。

出错

错误代码(SD0)	内容
3281H	(n)为0~4以外时。

任意数据的位分离

NDIS (P)

将指定数据的各位逐个分离为任意的位。

梯形图	ST
	ENO:=NDIS (EN, s1, s2, d) ; ENO:=NDISP (EN, s1, s2, d) ;

FBD/LD

■执行条件

指令	执行条件
NDIS	
NDISP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s1)	存储分离的数据的起始软元件	—	带符号BIN16位	ANY16*1
(d)	存储分离后的数据的起始软元件	—	带符号BIN16位	ANY16*1
(s2)	存储分离单位的起始软元件	—	带符号BIN16位	ANY16*1
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

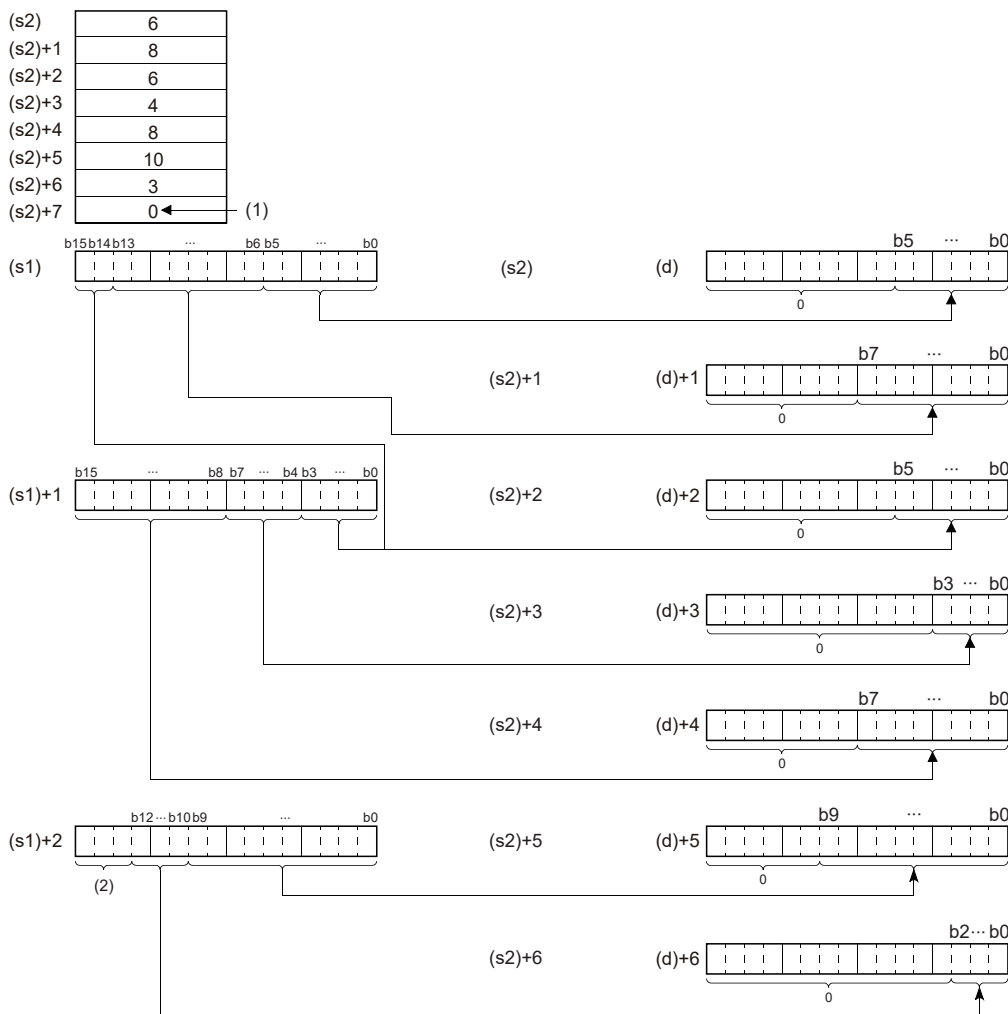
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 将(s1)中指定的软元件编号及其以后存储的数据的各位，逐个分离为(s2)中指定的位后，存储到(d)中指定的软元件编号及其以后。



(s2) ~ (s2)+6: (s2) ~ (s2)+6的指定位

(1) 设置结束指定

(2) 忽略。

- (s2)中指定的分离位数可在1~16位的范围内指定。
- 将从通过(s2)指定的软元件编号开始至存储了“0”的软元件编号为止作为分离位数处理。
- 应避免分离的数据的软元件范围((s1)~(s1)的结束范围)与存储分离后的数据的软元件范围((d)~(d)的结束范围)重复。重复的情况下，可能无法获得正确的运算结果。
- (s1)、(s2)、(d)中指定的软元件编号应避免重复。

出错

错误代码(SD0)	内容
2821H	(s1)、(s2)的软元件重复时。
	(s1)、(d)的软元件重复时。
	(s2)、(d)的软元件重复时。
3281H	(s2)中输入了不能转换的非法数据时。 <ul style="list-style-type: none"> 指定的分离位数指定没设置在1~16位的设置内。 从指定的软元件开始至超过相应软元件/标签的分配范围为止之间，未设置0。

任意数据的位合并

NUNI (P)

将指定数据的各位逐个合并为任意的位。

梯形图	ST
	ENO:=NUNI (EN, s1, s2, d) ; ENO:=NUNIP (EN, s1, s2, d) ;

FBD/LD

■执行条件

指令	执行条件
NUNI	
NUNIP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s1)	存储合并的数据的起始软元件	—	带符号BIN16位	ANY16*1
(d)	存储合并后的数据的起始软元件	—	带符号BIN16位	ANY16*1
(s2)	存储合并单位的起始软元件	—	带符号BIN16位	ANY16*1
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

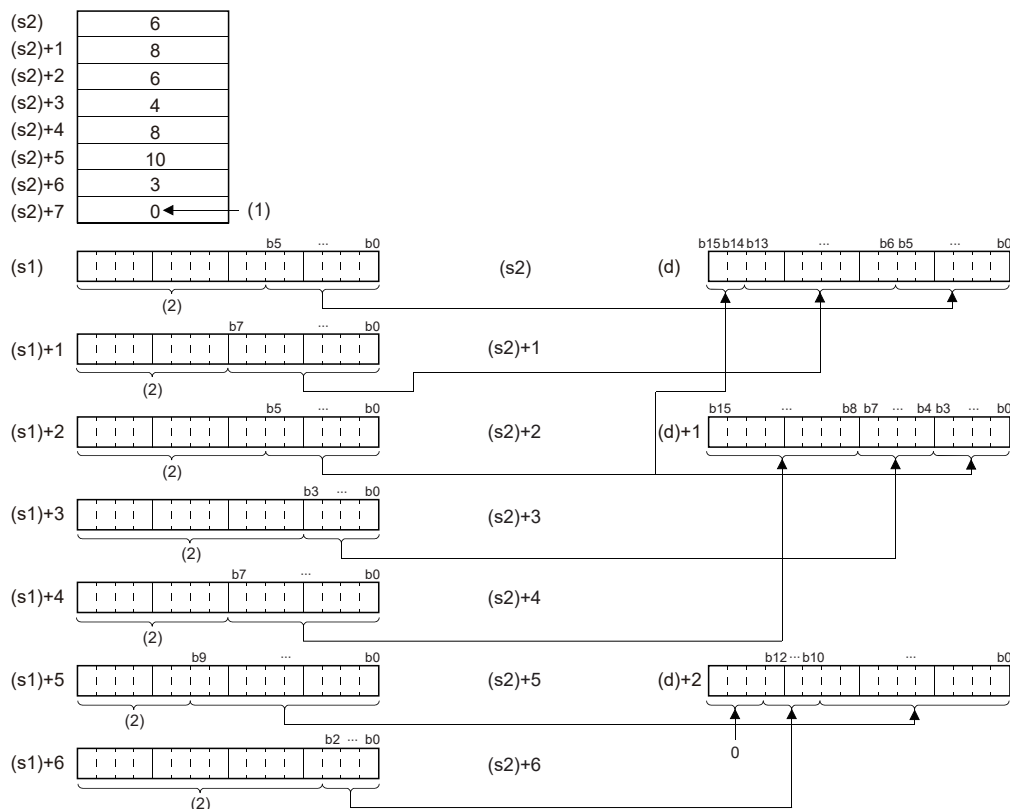
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 将(s1)中指定的软元件编号及其以后中存储的数据的各位，逐个合并为(s2)中指定的位后，存储到(d)中指定的软元件编号及其以后。



(s2) ~ (s2)+6: (s2) ~ (s2)+6 的指定位

(1) 设置结束指定

(2) 忽略。

- (s2) 中指定的合并位数可以在 1~16 位的范围内指定。
- 将从通过 (s2) 指定的软元件编号开始至存储了“0”的软元件编号为止作为合并位数处理。
- 合并的数据的软元件范围 ((s1) ~ (s1) 的结束范围) 与存储合并后的数据的软元件范围 ((d) ~ (d) 的结束范围) 应避免重复。重复的情况下，可能无法获得正确的运算结果。
- (s1)、(s2)、(d) 中指定的软元件编号应避免重复。

出错

错误代码 (SD0)	内容
2821H	(s1)、(s2) 的软元件重复时。
	(s1)、(d) 的软元件重复时。
	(s2)、(d) 的软元件重复时。
3281H	(s2) 中输入了不能转换的非法数据时。 <ul style="list-style-type: none"> 指定的合并位数指定没设置在 1~16 位的范围内。 从指定的软元件开始至超过相应软元件/标签的分配范围为止之间，未设置 0。

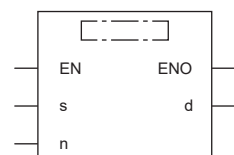
字节单位数据分离

WTOB(P)

将BIN16位数据分离为(n)字节。

梯形图	ST
	ENO:=WTOB(EN, s, n, d); ENO:=WTOBP(EN, s, n, d);

FBD/LD



■执行条件

指令	执行条件
WTOB	
WTOBP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储以字节单位分离的数据的起始软元件	—	带符号BIN16位	ANY16*1
(d)	存储以字节单位分离的结果的起始软元件	—	带符号BIN16位	ANY16*1
(n)	分离的字节数据的个数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

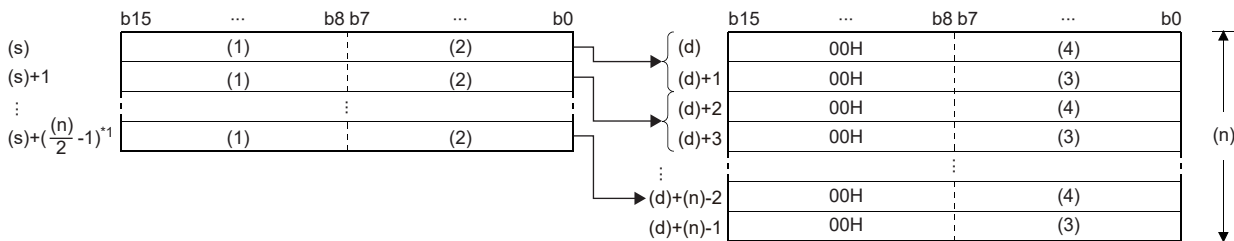
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 将(s)中指定的软元件编号及其以后中存储的BIN16位数据分离为(n)字节后, 存储到(d)中指定的软元件编号及其以后。

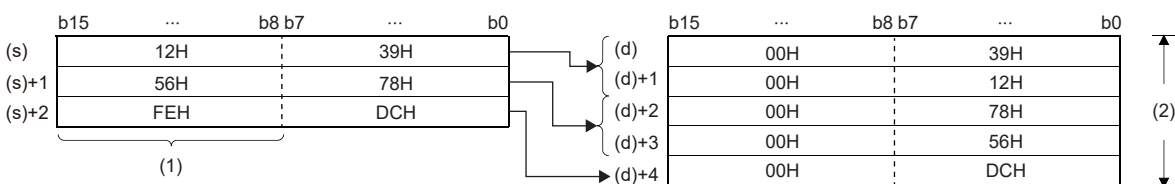


- (1) 高位字节
- (2) 低位字节
- (3) 高位字节的数据
- (4) 低位字节的数据

*1 小数点以下进位。

例

(n)=5的情况下, 将(s)~(s)+2的低位8位为止的数据存储到(d)~(d)+4中。

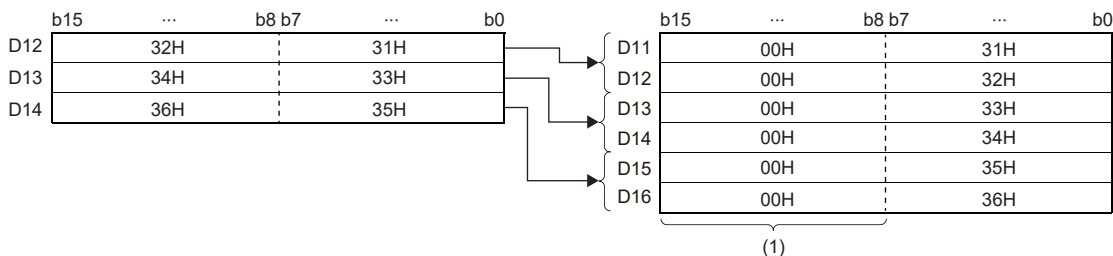


- (1) (n)=5的情况下忽略。
- (2) (n)=5的情况下

- 通过在(n)中设置字节数, (s)中指定的BIN16位数据的范围及存储(d)中指定的字节数据的软元件的范围将被自动确定。
- (n)中指定的字节数为0的情况下, 不进行处理。
- (d)中指定的字节数据存储软元件的高位8位中将被自动存储00H。

例

将D12~D14存储到D11~D16的低位8位中的情况下



- (1) 自动存储00H。

- 即使存储了分离的数据的软元件范围与存储分离后数据的软元件范围重复的情况下, 也将作为正常处理。

存储了分离的数据的软元件范围	存储分离后数据的软元件范围
$(s) \sim (s) + (\frac{n}{2} - 1)$	$(d) + 0 \sim (d) + (n) - 1$

出错

没有运算错误。

字节单位数据合并

BTOW (P)

将BIN16位数据的低位8位合并为字单位。

梯形图	ST
	ENO:=BTOW (EN, s, n, d) ; ENO:=BTOWP (EN, s, n, d) ;

FBD/LD

■执行条件

指令	执行条件
BTOW	
BTOWP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s)	存储了以字节单位合并的数据的起始软元件	—	带符号BIN16位	ANY16*1
(d)	存储以字节单位合并的结果的起始软元件	—	带符号BIN16位	ANY16*1
(n)	合并的字节数据的个数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

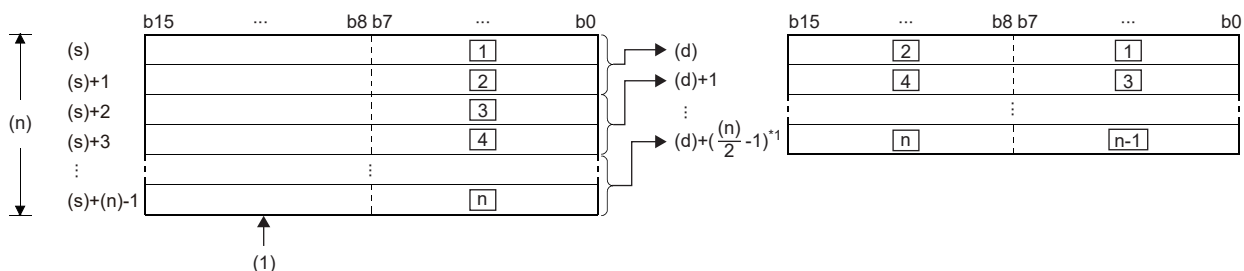
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 将(s)中指定的软元件编号及其以后中存储的(n)字的BIN16位数据的低位8位, 合并为字单位后, 存储到(d)中指定的软元件编号及其以后。
- (s)中指定的软元件编号及其以后中存储的(n)字的数据的高位8位将被忽略。此外, (n)为奇数的情况下, 存储了第(n)字节的数据的软元件的高位8位中将存储0。



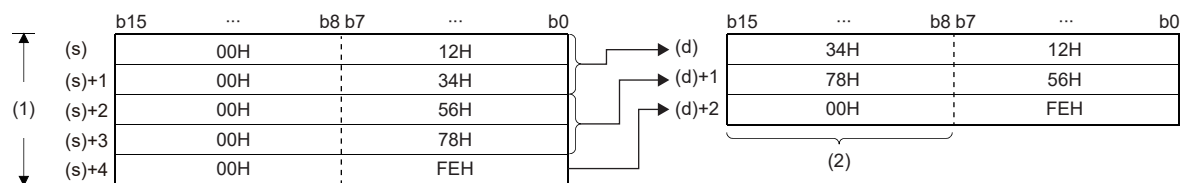
□: 第 □ 字节的数据

(1) 高位字节忽略。

*1 小数点以下进位。

例

(n)=5的情况下, 将(s)~(s)+4的低位8位为止的数据合并后存储到(d)~(d)+2中。



(1) (n)=5的情况下

(2) 存储00H。

- 通过在(n)中设置字节数, (s)中指定的字节数据的范围及存储(d)中指定的合并数据的软元件的范围将被自动确定。
- (n)中指定的字节数为0的情况下, 不进行处理。
- (s)中指定的字节数据存储软元件的高位8位将被忽略, 低位8位将成为对象。
- 即使存储了合并的数据的软元件范围与存储合并后的数据的软元件范围重复的情况下, 也将被作为正常处理。

存储了合并的数据存储的软元件范围

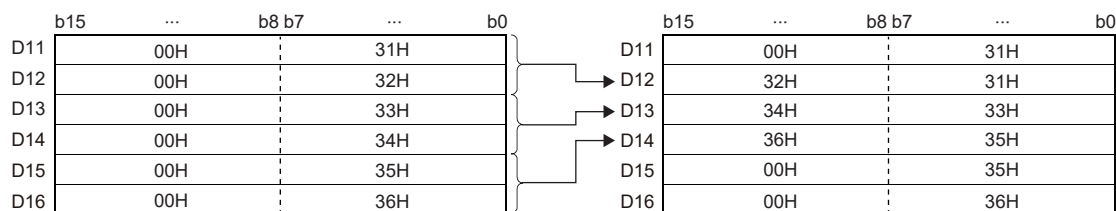
(s)+0~(s)+(n)-1

存储合并后数据的软元件范围

(d)~(d)+ $\frac{(n)}{2}-1$

例

将D11~D16的低位8位存储到D12~D14中的情况下



出错

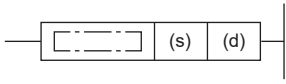
没有运算错误。

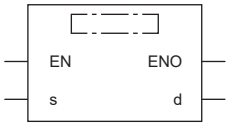
12.7 数据传送指令

BIN16位数据传送



MOV (P)

传送指定的软元件的BIN16位数据。

梯形图	ST
	ENO:=MOV (EN, s, d); ENO:=MOV (EN, s, d);

FBD/LD


■执行条件

指令	执行条件
MOV	
MOVP	

设置数据

■内容、范围、数据类型

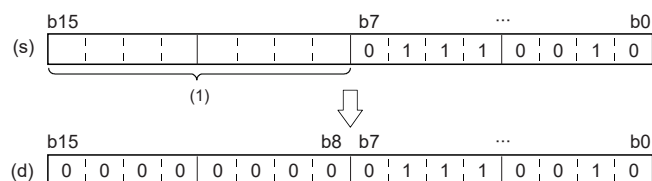
操作数	内容	范围	数据类型	数据类型 (标签)
(s)	传送源数据或存储了数据的软元件编号	-32768~32767	带符号BIN16位	ANY16
(d)	传送目标软元件编号	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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	
(s)	○	○	○	○	○	—	—	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—

功能

- 将(s)中指定的BIN16位数据传送到(d)中指定的软元件。
- (s)为进行了位指定的位软元件的情况下，位指定中指定的位将成为传送的对象。传送对象以外的位中将被作为指定了0而进行传送。



(1) (s)不足16位的情况下，将作为指定了0而进行传送。

出错

没有运算错误。

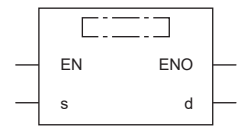
BIN32位数据传送

DMOV (P)

传送指定的软元件的BIN32位数据。

梯形图	ST
	ENO:=DMOV (EN, s, d) ; ENO:=DMOVP (EN, s, d)

FBD/LD



■执行条件

指令	执行条件
DMOV	
DMOVP	

设置数据

■内容、范围、数据类型

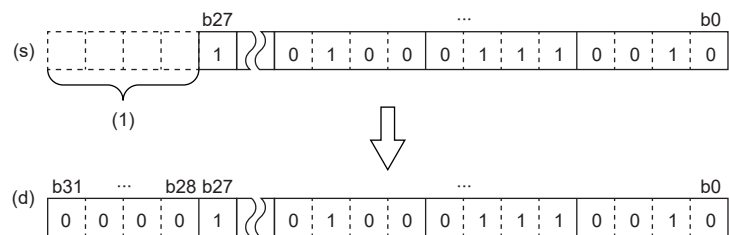
操作数	内容	范围	数据类型	数据类型(标签)
(s)	传送源数据或存储了数据的软元件编号	-2147483648~2147483647	带符号BIN32位	ANY32
(d)	传送目标软元件编号	—	带符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—

功能

- 将(s)中指定的BIN32位数据传送到(d)中指定的软元件。
- (s)为进行了位指定的位软元件的情况下，位指定中指定的位将成为传送的对象。传送对象以外的位中将被作为指定了0而进行传送。



(1) (s)不足32位的情况下，将作为指定了0而进行传送。

出错

没有运算错误。

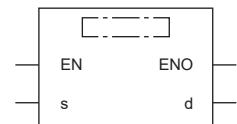
BIN16位数据否定传送

CML (P)

对指定的BIN16位数据执行逐位取反后进行传送。

梯形图	ST
	ENO:=CML (EN, s, d); ENO:=CMLP (EN, s, d);

FBD/LD



■执行条件

指令	执行条件
CML	
CMLP	

设置数据

■内容、范围、数据类型

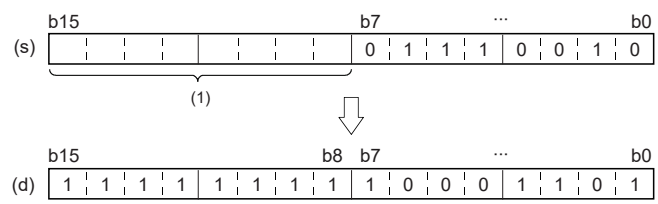
操作数	内容	范围	数据类型	数据类型(标签)
(s)	取反的数据或存储了数据的软件元件编号	-32768~32767	带符号BIN16位	ANY16
(d)	存储取反结果的软件元件编号	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—	—

功能

- 将(s)中指定的BIN16位数据进行逐位取反，将其结果传送到(d)中指定的软元件中。
- (s)为位指定的位软元件的情况下，位指定中指定的位将成为对象。对象以外的位中将被作为指定了0而取反后，进行传送。



(1) (s)不足16位的情况下，将被作为指定了0而取反。

出错

没有运算错误。

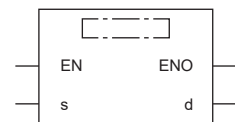
BIN32位数据否定传送

DCML (P)

对指定的BIN32位数据执行逐位取反后进行传送。

梯形图	ST
	ENO:=DCML (EN, s, d); ENO:=DCMLP (EN, s, d);

FBD/LD



■执行条件

指令	执行条件
DCML	
DCMLP	

设置数据

■内容、范围、数据类型

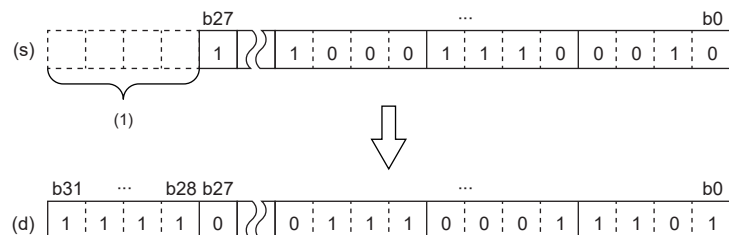
操作数	内容	范围	数据类型	数据类型 (标签)
(s)	取反的数据或存储了数据的软元件编号	-2147483648~2147483647	带符号BIN32位	ANY32
(d)	存储取反结果的软元件编号	—	带符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	○	○	○	—	—	—	—

功能

- 对(s)中指定的BIN32位数据进行逐位取反后, 将其结果传送到(d)中指定的软元件。
- (s)为位指定的位软元件的情况下, 位指定中指定的位将成为对象。对象以外的位中将被作为指定了0而取反后, 进行传送。



(1) (s)不足32位的情况下, 将被作为指定了0而取反。

出错

没有运算错误。

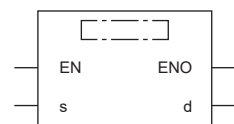
1位数据取反传送

CMLB(P)

对指定的位数据进行取反后传送。

梯形图	ST
	ENO:=CMLB(EN, s, d); ENO:=CMLBP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
CMLB	
CMLBP	

设置数据

■内容、范围、数据类型

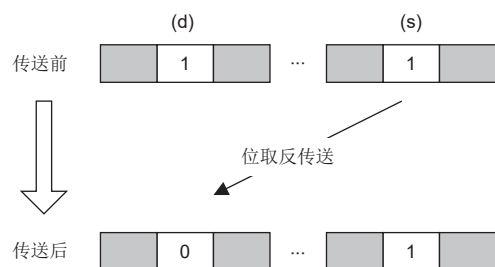
操作数	内容	范围	数据类型	数据类型(标签)
(s)	取反的数据或存储了数据的软元件编号	—	位	ANY_BOOL
(d)	传送目标软元件编号	—	位	ANY_BOOL
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	—	—	○	—	○	—	—	—	—

功能

- 对(s)中指定的位数据进行取反，将其结果存储到(d)中指定的软元件。



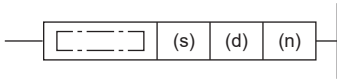
出错

没有运算错误。

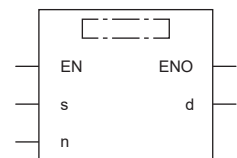
BIN16位数据块传送(16位)

BMOV (P)


将从指定的软元件开始的(n)点(0~65535)的BIN16位数据进行批量传送。

梯形图	ST
	ENO:=BMOV (EN, s, n, d) ; ENO:=BMOV P (EN, s, n, d) ;

FBD/LD



■执行条件

指令	执行条件
BMOV	
BMOV P	

设置数据


■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储了传送数据的起始软元件	—	带符号BIN16位	ANY16*1*2
(d)	传送目标的起始软元件	—	带符号BIN16位	ANY16*1*2
(n)	传送数	0~65535	无符号BIN16位	ANY16*2
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下,应在确保动作所需区域的前提下定义数组,指定该数组型标签的要素。

*2 也可以使用INT型/WORD型以外的数据类型。(BOOL型/POINTER型/STRUCT型(结构体除外))此外,在选项设置的“执行指令参数的数据类型检查”中选择“是”时,则可使用的数据类型仅为INT型/WORD型(ANY16的范围)。

有关选项设置,请参阅下述手册。

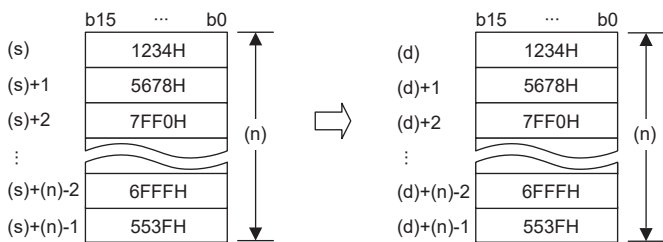
 GX Works3 操作手册

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

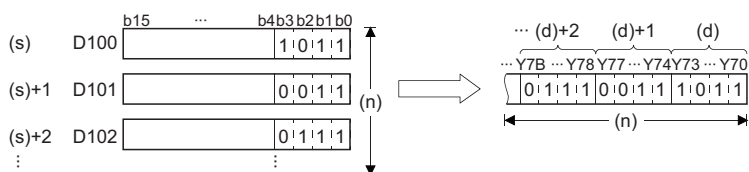
- 将(s)中指定的软元件开始的(n)点的BIN16位数据批量传送到(d)中指定的软元件。



- 传送源与传送目标软元件重复的情况下也可进行传送。向软元件编号的小编号向传送的情况下从(s)开始传送，向软元件编号的大编号方向传送的情况下从(s)+(n)-1开始传送。
- (s)为字软元件而(d)为位软元件的情况下，对于字软元件，位软元件的位指定中指定的位数将成为对象。

例

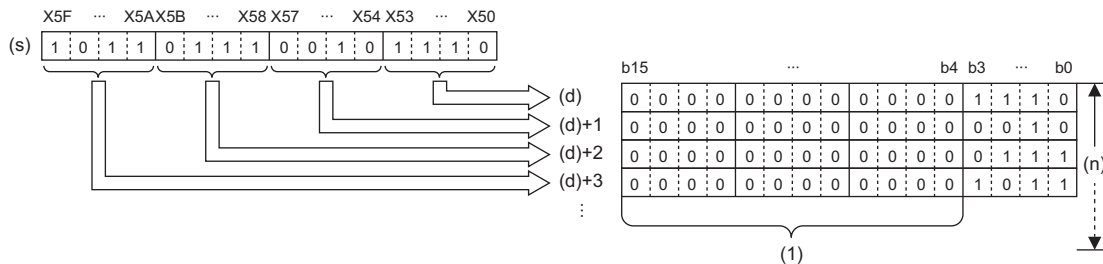
(d)中指定K1Y70的情况下，(s)中指定的字软元件的低位4位将成为对象。



- (s)为位指定的位软元件，(d)为字软元件的情况下，对于字软元件，位软元件的位指定中指定的位数将成为对象。

例

(s)中指定了K1X50的情况下，将传送到(d)的字软元件的低4位中。



(1) 存储0。

- (s)、(d)两方均为位软元件的情况下，必须将(s)、(d)的位数设置为相同。
- (s)、(d)中使用链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件的情况下，只应指定(s)或(d)中的一方。

出错

错误代码(SD0)	内容
3290H	(s)、(d)的两方中指定了链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件时。

BIN16位数据块传送(32位)

BMOV(L)(P)

将从指定的软元件开始的(n)点(1~4294967295)的BIN16位数据进行批量传送。

梯形图	ST
	ENO:=BMOV(L)(EN, s, n, d); ENO:=BMOVLP(EN, s, n, d);

FBD/LD

■执行条件

指令	执行条件
BMOV(L)	
BMOVLP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储了传送数据的起始软元件	—	带符号BIN16位	ANY16*1
(d)	传送目标的起始软元件	—	带符号BIN16位	ANY16*1
(n)	传送数	0~4294967295	无符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

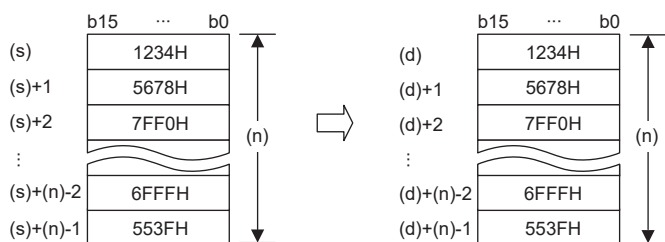
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	○	○	○	○	○	○	○	○	○	—	—

功能

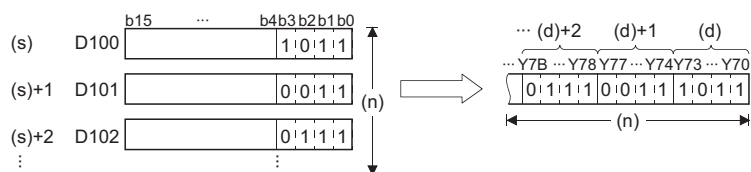
- 将从(s)中指定的软元件开始的(n)点的BIN16位数据, 以(d)中指定的软元件开始的(n)点进行批量传送。



- 传送源与传送目标软元件重复的情况下也可进行传送。向软元件编号的小编号向传送的情况下从(s)开始传送, 向软元件编号的大编号方向传送的情况下从(s)+(n)-1开始传送。
- (s)为字软元件而(d)为位软元件的情况下, 对于字软元件, 位软元件的位指定中指定的位数将成为对象。

例

(d)中指定K1Y70的情况下, (s)中指定的字软元件的低位4位将成为对象。



- (s)、(d)两方均为位软元件的情况下, 必须将(s)、(d)的位数设置为相同。
- (s)、(d)中使用链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件的情况下, 只应指定(s)或(d)中的一方。

出错

错误代码(SD0)	内容
3290H	(s)、(d)的两方中指定了链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件时。

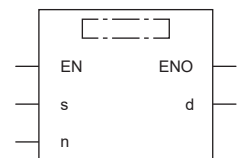
同一BIN16位数据块传送(16位)

FMOV(P)

将BIN16位数据传送到指定的软元件开始的(n)点(0~65535)中。

梯形图	ST
	ENO:=FMOV(EN, s, n, d); ENO:=FMOV(P)(EN, s, n, d);

FBD/LD



■执行条件

指令	执行条件
FMOV	
FMOV(P)	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	传送的数据或存储了传送数据的软元件	-32768~32767	带符号BIN16位	ANY16
(d)	传送目标的起始软元件	—	带符号BIN16位	ANY16*1
(n)	传送数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

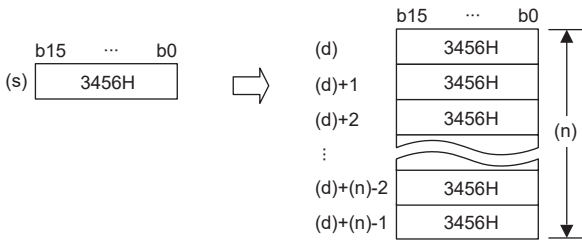
*1 通过标签进行设置的情况下,应在确保动作所需区域的前提下定义数组,指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

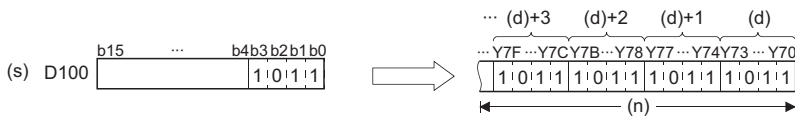
- 将与(s)中指定的软元件的BIN16位数据相同的数据，以(n)点传送到(d)中指定的软元件中。



- (s)为字软元件而(d)为位软元件的情况下，对于字软元件，位软元件的位指定中指定的位数将成为对象。

例

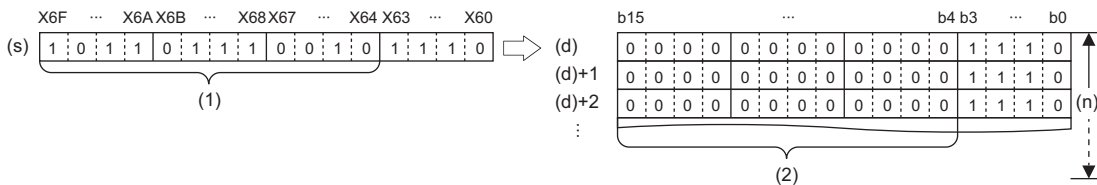
(d)中指定K1Y70的情况下，(s)中指定的字软元件的低位4位将成为对象。



- (s)为位指定的位软元件，(d)为字软元件的情况下，对于字软元件，位软元件的位指定中指定的位数将成为对象。

例

(s)中指定了K1X60的情况下，将传送到(d)的字软元件的低4位中。



- (1) 忽略。
- (2) 存储0。

- (s)、(d)两方均为位软元件的情况下，必须将(s)、(d)的位数设置为相同。

出错

没有运算错误。

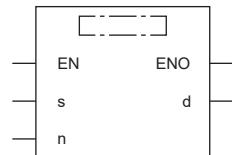
同一BIN16位数据块传送(32位)

FMOVL(P)

将BIN16位数据传送到指定的软元件开始的(n)点(1~4294967295)中。

梯形图	ST
	ENO:=FMOVL(EN, s, n, d); ENO:=FMOVLP(EN, s, n, d);

FBD/LD



■执行条件

指令	执行条件
FMOVL	
FMOVLP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	传送的数据或存储了传送数据的软元件	-32768~32767	带符号BIN16位	ANY16
(d)	传送目标的起始软元件	—	带符号BIN16位	ANY16*1
(n)	传送数	0~4294967295	无符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

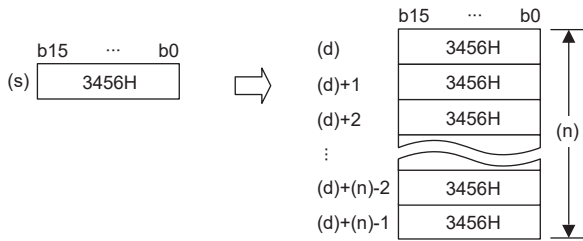
*1 通过标签进行设置的情况下,应在确保动作所需区域的前提下定义数组,指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	○	○	○	○	—	—	—

功能

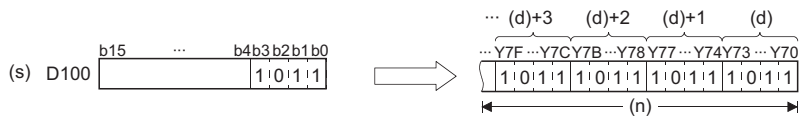
- 将与(s)中指定的软元件的BIN16位数据相同的数据，以(n)点传送到(d)中指定的软元件中。



- (s)为字软元件而(d)为位软元件的情况下，对于字软元件，位软元件的位指定中指定的位数将成为对象。

例

(d)中指定K1Y70的情况下，(s)中指定的字软元件的低位4位将成为对象。



- (s)、(d)两方均为位软元件的情况下，必须将(s)、(d)的位数设置为相同。

出错

没有运算错误。

同一BIN32位数据块传送(16位)

DFMOV (P)

将BIN32位数据传送至指定的软元件开始的(n)点(1~65535)。

梯形图	ST
	ENO:=DFMOV (EN, s, n, d) ; ENO:=DFMOV (EN, s, n, d) ;

FBD/LD

■执行条件

指令	执行条件
DFMOV	
DFMOV P	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	传送的数据或存储了传送数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32
(d)	传送目标的起始软元件	—	带符号BIN32位	ANY32*1
(n)	传送数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

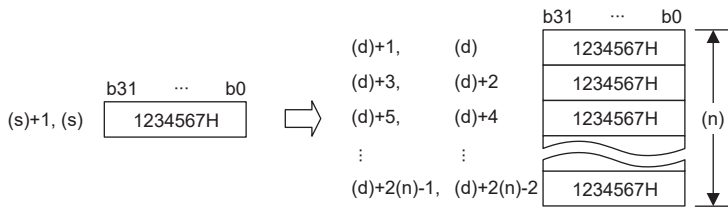
*1 通过标签进行设置的情况下,应在确保动作所需区域的前提下定义数组,指定该数组型标签的要素。

■可用软元件

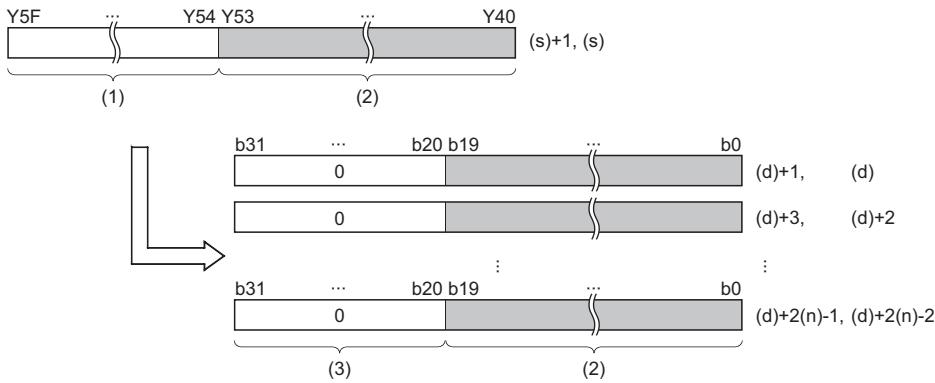
操作数	位		字		双字			间接指定	常数				其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	—	○	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

- 将与(s)中指定的软元件的BIN32位数据相同的数据以(n)点传送到(d)中指定的软元件。



- (s)中指定了位指定的情况下，传送的数据仅为位指定的量。(s)中指定了K5Y40的情况下，(s)的字软元件的低位20位(5位)将成为对象。

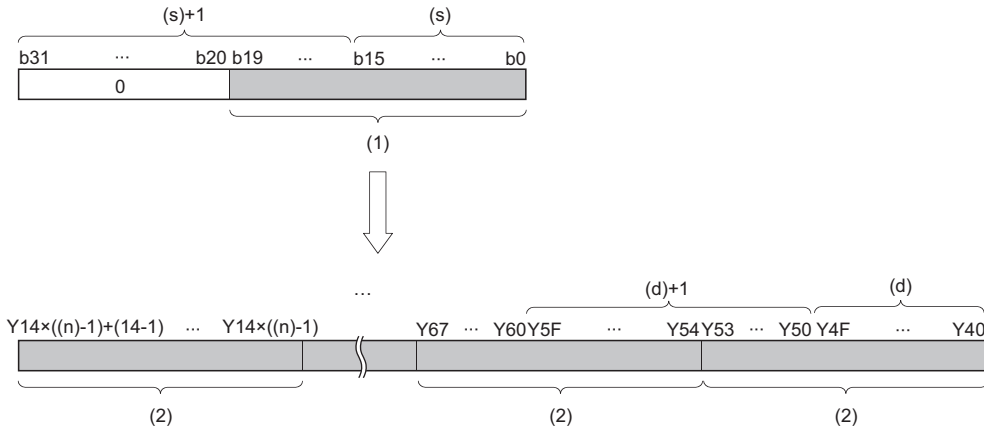


(1) 忽略。

(2) 20位(5位)的数据

(3) 存储0。

- (d)中指定了位指定的情况下，对(d)中指定的位指定的数据进行传送。(d)中指定K5Y40的情况下，(s)的软元件的低位20位将成为对象。(s)、(d)的两方中指定了位指定的情况下，与位数无关，对(d)中指定的位指定的数据进行传送。



(1) (d)中指定的数据(20位(5位))

(2) 20位(5位)的数据

- (n)中指定的值为0的情况下，将变为无处理。

出错

没有运算错误。

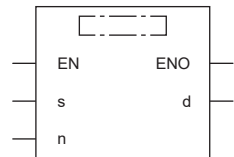
同一BIN32位数据块传送(32位)

DFMOVL(P)

将BIN32位数据传送至指定的软元件开始的(n)点(1~4294967295)。

梯形图	ST
	ENO:=DFMOVL(EN, s, n, d); ENO:=DFMOVL(P)(EN, s, n, d);

FBD/LD



■执行条件

指令	执行条件
DFMOVL	
DFMOVL(P)	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	传送的数据或存储了传送数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32
(d)	传送目标的起始软元件	—	带符号BIN32位	ANY32*1
(n)	传送数	0~4294967295	无符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

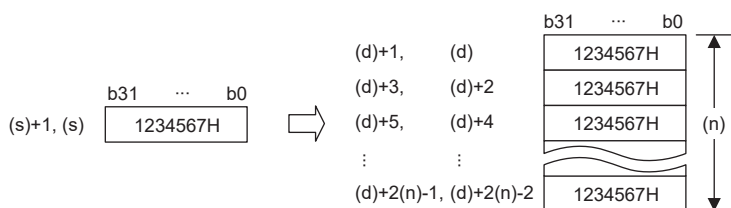
*1 通过标签进行设置的情况下,应在确保动作所需区域的前提下定义数组,指定该数组型标签的要素。

■可用软元件

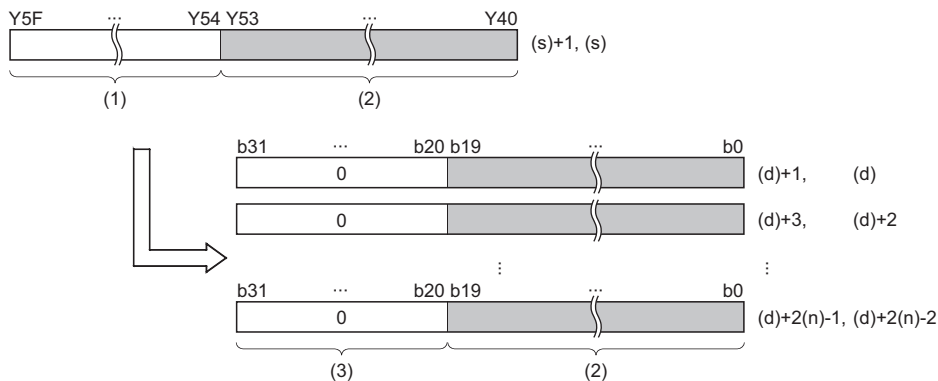
操作数	位		字			双字		间接指定	常数				其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	—	○	—	○	—	—	—	—	—
(n)	○	○	○	○	○	○	○	○	○	—	—	—	—

功能

- 将与(s)中指定的软元件的BIN32位数据相同的数据以(n)点传送到(d)中指定的软元件。

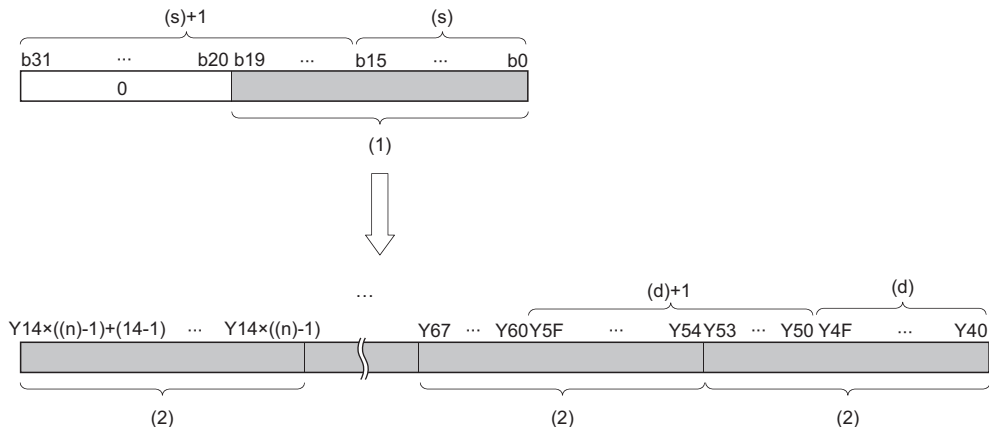


- (s)中指定了位指定的情况下，传送的数据仅为位指定的量。(s)中指定了K5Y40的情况下，(s)的软元件的低位20位(5位)将成为对象。



- (1) 忽略。
- (2) 20位(5位)的数据
- (3) 存储0。

- (d)中指定了位指定的情况下，对(d)中指定的位指定的数据进行传送。(d)中指定K5Y40的情况下，(s)的软元件的低位20位将成为对象。(s)、(d)的两方中指定了位指定的情况下，与位数无关，对(d)中指定的位指定的数据进行传送。



- (1) (d)中指定的数据(20位(5位))
- (2) 20位(5位)的数据

- (n)中指定的值为0的情况下，将变为无处理。

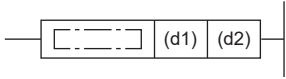
出错

没有运算错误。

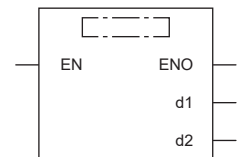
BIN16位数据交换

XCH(P)

对指定的BIN16位数据进行交换。

梯形图	ST
	ENO:=XCH(EN, d1, d2); ENO:=XCHP(EN, d1, d2);

FBD/LD



■执行条件

指令	执行条件
XCH	
XCHP	

设置数据

■内容、范围、数据类型

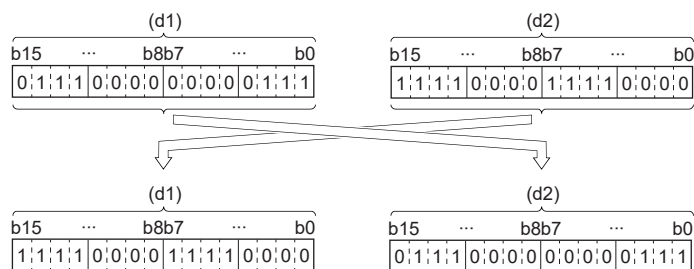
操作数	内容	范围	数据类型	数据类型(标签)
(d1)	存储了交换数据的软元件	—	带符号BIN16位	ANY16
(d2)	存储了交换数据的软元件	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

对(d1)与(d2)的BIN16位数据进行交换。



出错

没有运算错误。

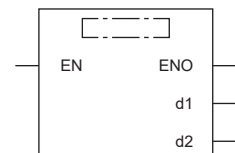
BIN32位数据交换

DXCH(P)

对指定的BIN32位数据进行交换。

梯形图	ST
	ENO:=DXCH(EN, d1, d2); ENO:=DXCHP(EN, d1, d2);

FBD/LD



■执行条件

指令	执行条件
DXCH	
DXCHP	

设置数据

■内容、范围、数据类型

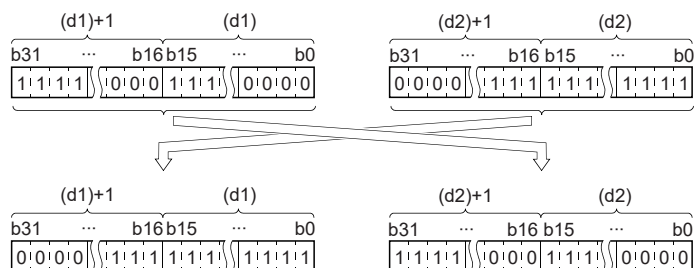
操作数	内容	范围	数据类型	数据类型(标签)
(d1)	存储交换数据的起始软元件	—	带符号BIN32位	ANY32
(d2)	存储交换数据的起始软元件	—	带符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—

功能

对(d1)与(d2)的BIN32位数据进行交换。



出错

没有运算错误。

BIN16位数据块交换

BXCH(P)

对指定的软元件开始的(n)点的BIN16位数据进行交换。

梯形图	ST
	ENO:=BXCH(EN, n, d1, d2); ENO:=BXCHP(EN, n, d1, d2);

FBD/LD

■执行条件

指令	执行条件
BXCH	
BXCHP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d1)	存储交换数据的起始软元件	—	字	ANY16*1
(d2)	存储交换数据的起始软元件	—	字	ANY16*1
(n)	交换数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

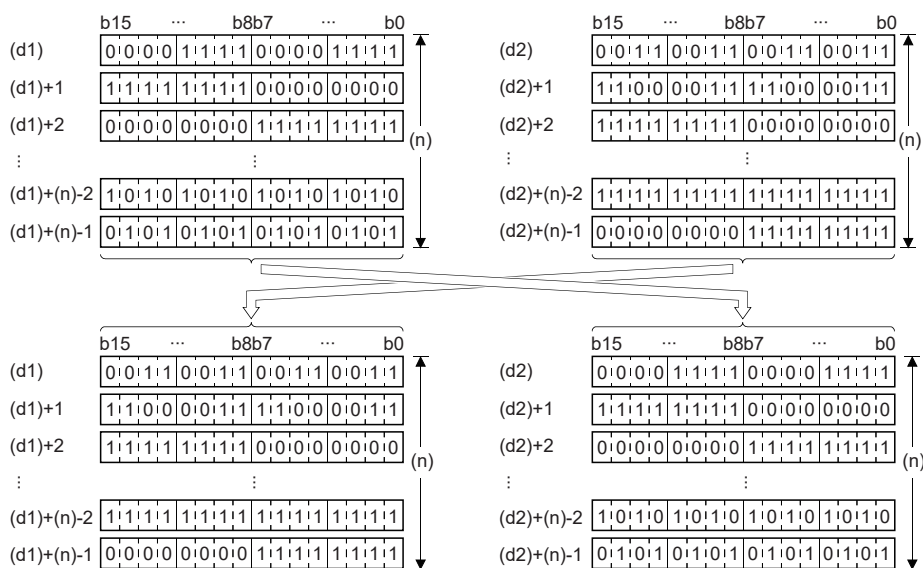
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

对 (d1) 中指定的软元件与 (d2) 中指定的软元件开始的 (n) 点的BIN16位数据进行交换。



出错

错误代码 (SD0)	内容
2821H	(d1)、(d2)的软元件重复时。

BIN16位数据上下字节交换

SWAP (P)

对指定软元件的上下各8位的值进行交换。

梯形图	ST
	ENO:=SWAP (EN, d) ; ENO:=SWAPP (EN, d) ;

FBD/LD

■执行条件

指令	执行条件
SWAP	
SWAPP	

设置数据

■内容、范围、数据类型

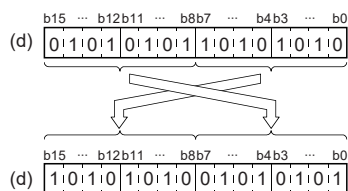
操作数	内容	范围	数据类型	数据类型 (标签)
(d)	存储进行上下各8位值交换的数据的起始软元件	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

(d)中指定的软元件进行上下各8位的值的交换。



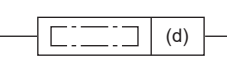
出错

没有运算错误。

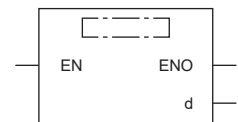
BIN32位数据上下字节交换

DSWAP (P)


对指定软元件的上下各8位的值进行交换。

梯形图	ST
	ENO:=DSWAP (EN, d) ; ENO:=DSWAPP (EN, d) ;

FBD/LD



■执行条件

指令	执行条件
DSWAP	
DSWAPP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(d)	存储进行上下各8位值交换的数据的起始软元件	—	带符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

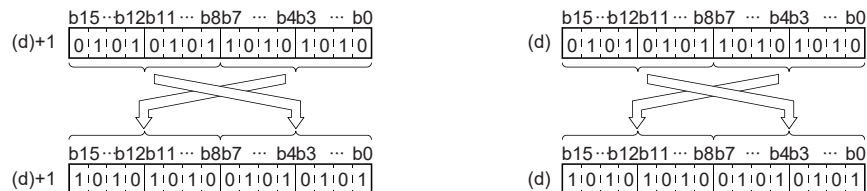
■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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 FX、FY不能使用。

功能

(d)中指定的软元件进行上下各8位的值的交换。



出错

没有运算错误。

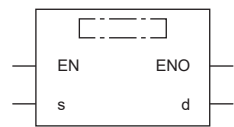
1位数据传送

MOV(B) (P)

对指定的1位数据进行传送。

梯形图	ST
	ENO:=MOV(B) (EN, s, d) ; ENO:=MOV(BP) (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
MOV(B)	
MOV(BP)	

设置数据

■内容、范围、数据类型

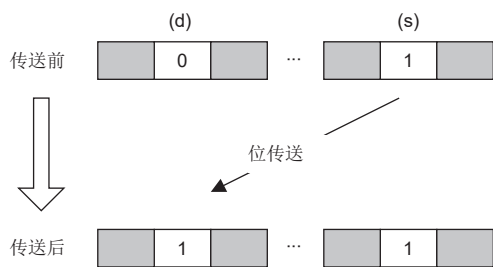
操作数	内容	范围	数据类型	数据类型 (标签)
(s)	存储了传送数据的软元件编号	—	位	ANY_BOOL
(d)	传送目标软元件编号	—	位	ANY_BOOL
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	—	—	○	—	○	—	—	—	—

功能

将(s)中指定的位数据传送到(d)中。



出错

没有运算错误。

n位数据传送

BLKMOVB(P)

对(n)点的位数据进行批量传送。

梯形图	ST
	ENO:=BLKMOVB(EN, s, n, d); ENO:=BLKMOVBP(EN, s, n, d);

FBD/LD

■执行条件

指令	执行条件
BLKMOVB	
BLKMOVBP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	传送源块数据	—	位	ANY_BOOL ^{*1}
(d)	传送目标块数据	—	位	ANY_BOOL ^{*1}
(n)	传送数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

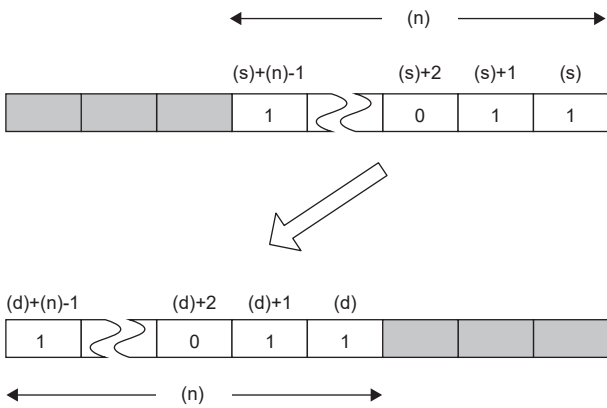
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 将从(s)开始的(n)点的位数据批量传送到(d)开始的(n)点的位数据。
- 传送源与传送目标重复的情况下, 也可进行传送。



出错

没有运算错误。

第4部分 应用指令

本部分由下述章构成。

13 程序控制

14 数据处理

15 调试、故障诊断

16 字符串处理

17 实数处理

18 随机数

19 软元件操作

20 定时器、计数器

21 时钟

22 模块访问

23 PID运算指令

24 PID控制指令

25 高速I/O控制指令

26 定位功能专用指令

27 高速计数器功能专用指令

28 脉冲系统指令

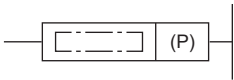
13 程序控制

13.1 程序分支指令

指针分支


CJ、SCJ、JMP

- CJ：执行同一程序文件内指定的指针编号的程序。
- SCJ：从下一个扫描开始执行同一程序文件内指定的指针编号的程序。
- JMP：无条件执行同一程序文件内指令的指针编号的程序。

梯形图	ST
	不对应。

FBD/LD
不对应。

■执行条件

指令	执行条件
CJ SCJ	
JMP	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(P)	跳转目标的指针编号	—	软元件名	POINTER

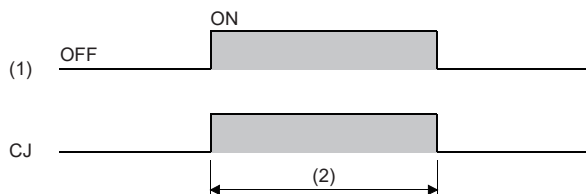
■可用软元件

操作数	位		字		双字		间接指定	常数			其它 (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)	—	—	—	—	—	—	—	—	—	—	○

功能

■CJ

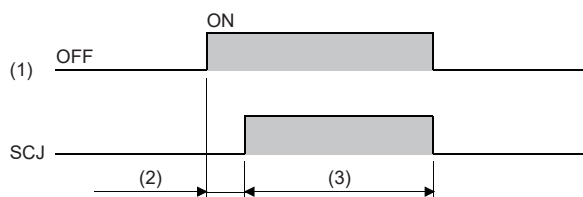
- 执行指令为ON时，执行同一程序文件内的指定指针号的程序。
- 执行指令为OFF时，执行下一步的程序。



- (1) 执行指令
(2) 每个扫描执行

■SCJ

- 从执行指令由OFF→ON变化的下一个扫描开始，执行同一程序文件内的指定指针号的程序。
- 执行指令为OFF及由ON→OFF变化时，执行下一个步的程序。



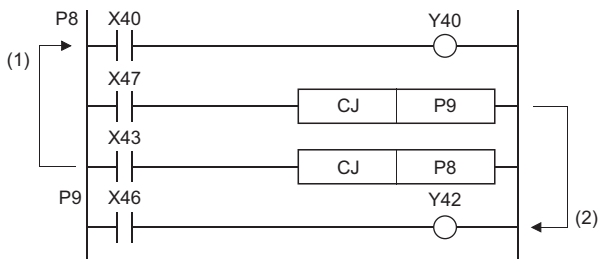
- (1) 执行指令
(2) 1个扫描
(3) 每个扫描执行

■JMP

无条件执行同一程序下一个文件内指定的指针编号的程序。

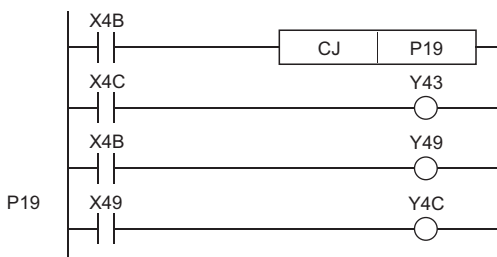
注意事项

- 将定时器的线圈置为ON后，通过CJ指令、SCJ指令、JMP指令对线圈为ON的定时器进行了跳转的情况下，将无法正常进行计测。
- 通过CJ指令、SCJ指令、JMP指令对OUT指令进行跳转时扫描时间将变短。
- 通过CJ指令、SCJ指令、JMP指令向后跳转时扫描时间将变短。
- CJ指令、SCJ指令、JMP指令可以跳转至比执行中的步号小的步之处。但是，为了避免看门狗定时器时限到，应考虑从环路中跳出的方法。



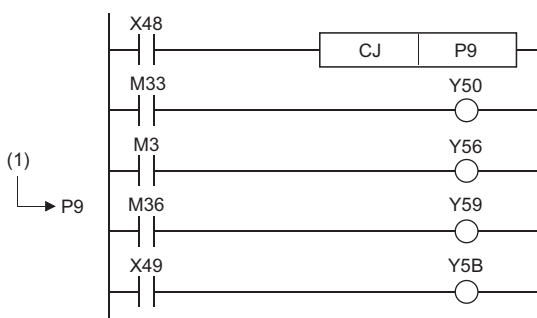
- (1) X43为ON期间，执行环路。
- (2) 将X47置为ON时，从环路中跳出。

- 通过CJ指令、SCJ指令、JMP指令跳过的软元件不变化。



- X4B为ON时，跳转至P19的标签。
- CJ指令执行中即使X4B、X4C变为ON/OFF，Y43、Y49也不变化。

- 标签(P□)占用1步。



- (1) 占用1步。

- 跳转指令只能指定同一程序文件内的指针编号。

出错

错误代码(SD0)	内容
3340H	(P)中指定的指针编号在END指令之前不存在时。
	(P)中指定的指针编号未在同一程序中作为标签使用时。
	(P)中指定的指针编号为其它程序中的全局指针时。


GOEND

跳转至同一程序文件内的FEND或END指令。

梯形图	ST
	不对应。

FBD/LD
不对应。

■执行条件

指令	执行条件
GOEND	

功能

跳转至同一程序文件内的FEND或END指令。

出错

错误代码 (SD0)	内容
3341H	执行CALL (P) 指令、ECALL (P) 指令后，执行RET指令前执行了GOEND指令时。
3351H	在通过中断指针 (I) 进行的中断程序中，执行IRET指令前执行了GOEND指令时。

13.2 程序执行控制指令

中断禁止、中断允许

DI、EI

- DI：禁止中断程序的执行。
- EI：解除中断禁止状态。

梯形图	ST
	ENO:=DI (EN); ENO:=EI (EN);

FBD/LD

■执行条件

指令	执行条件
DI EI	

功能

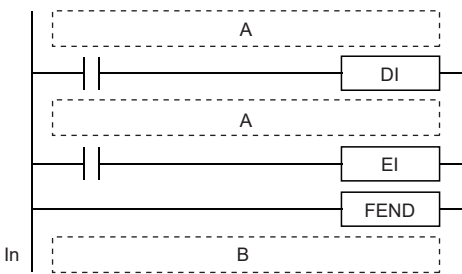
■DI

- 即使发生中断程序的启动原因，在执行EI指令之前禁止中断程序的执行。
 - 关于使用DI (指定优先级以下的中断禁止) 指令时的DI (中断禁止) 指令的动作，请参阅下述内容。
- ☞ 523页 DI
- DI (中断禁止) 指令不能通过中断程序执行。执行的情况下将变为无处理。

■EI

- 解除执行DI (中断禁止) 指令时的中断禁止状态，使通过IMASK指令置为允许的中断指针编号的中断程序置为允许执行状态。IMASK指令非执行时，I32~I43将变为中断禁止。
- 电源投入时或进行了控制器复位，RUN转移的情况下，将变为执行了EI指令后的状态。有初始执行类型程序的设置的情况下，RUN后第1个扫描的初始执行类型程序的执行中将变为执行了DI指令的状态，从执行第2个扫描的程序时开始变为执行了EI指令的状态。
- 关于使用DI (指定优先级以下的中断禁止) 指令时的EI指令的动作，请参阅下述内容。

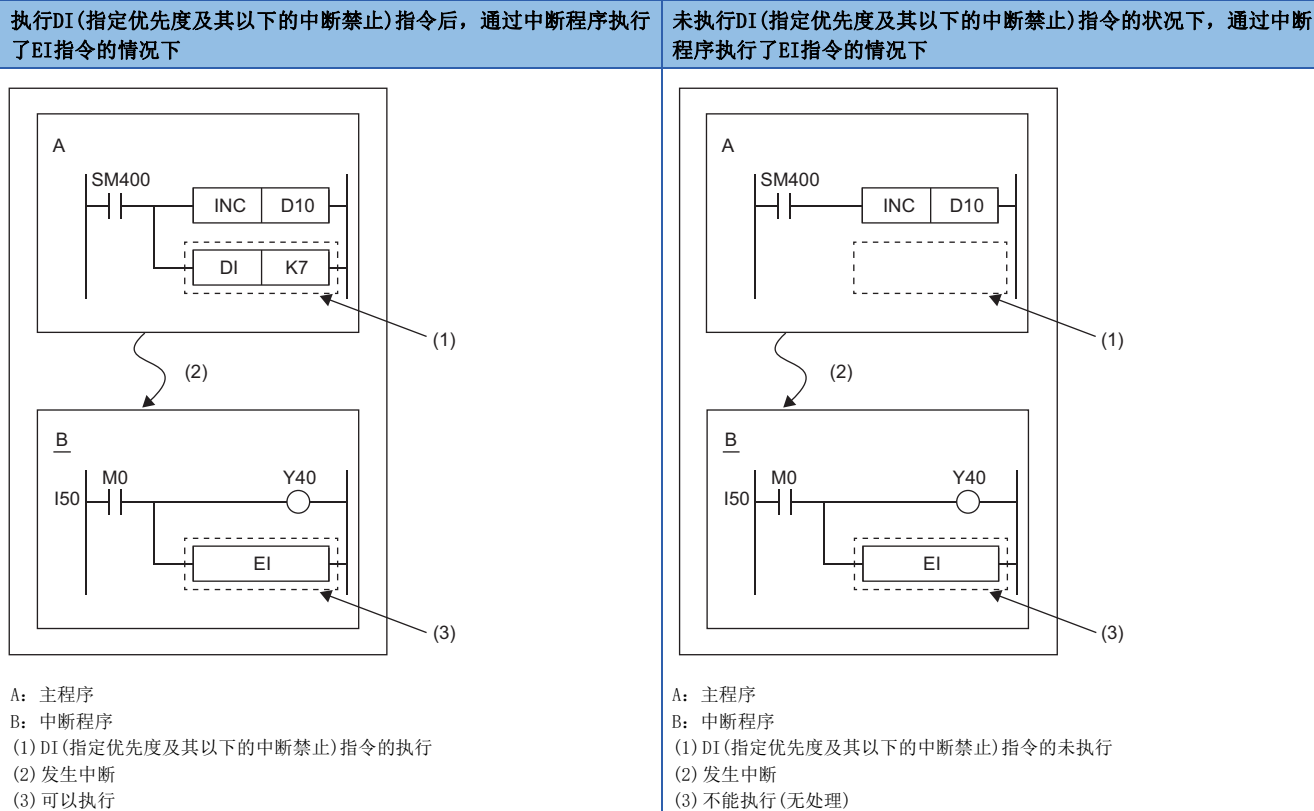
☞ 523页 DI



A: 顺控程序
B: 中断程序

DI~EI指令之间即使发生中断原因，在DI~EI指令之间的处理结束之前，中断程序将等待。

- 中断程序中的EI指令的动作，根据EI指令执行前的DI(指定优先级及其以下的中断禁止)指令的执行而变化。执行DI(指定优先级及其以下的中断禁止)指令后的中断程序中的EI指令可以执行。中断程序中的EI指令的执行可否如下所示。



- 初始执行类型程序中的EI指令、DI指令可以执行。但是，初始执行类型程序中执行了EI指令、DI指令时，在第2个扫描开始之前指定优先度的设置为有效。从执行第2个扫描的程序时开始变为执行了EI指令的状态。

要点

设置了超过网络通信周期同步中断的执行周期的中断禁止区间的情况下，可能会发生周期溢出。应使中断禁止区间最小限度进行编程，或者使用DI(指定优先级及其以下的中断禁止)指令使网络通信周期同步中断成为禁止对象外。使用DI指令将妨碍网络通信周期同步中断，成为引起下述功能误动作的原因。

- 运动功能
- 网络周期通信功能

出错

错误代码 (SD0)	内容
3332H	通过DI (中断禁止) 指令、DI (指定优先度及其以下的中断禁止) 指令进行的嵌套超过了16层的情况下。

如果DI指令的中断禁止区间过长，可能会发生下述错误或事件。

发生错误或事件的情况下，应使中断禁止区间最小限度进行编程，或者使用DI (指定优先度及其以下的中断禁止) 指令使网络通信周期同步中断成为禁止对象外。

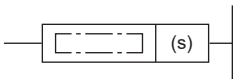
错误代码/事件代码	内容
00F08	运算处理及其他恒定周期处理未在设置的周期内完成。
11F0* ¹	同步中断程序的执行间隔超过了设定值。 网络通信周期同步程序的执行时间未在CC-Link IE TSN功能的通信周期间隔设置内完成。
11F1* ¹	同步中断程序的执行间隔超过了设定值。 网络通信周期同步程序检测出未执行的周期。
1860	发生了循环传送的遗漏。
1A00	运算处理及其他恒定周期处理未在设置的周期内完成。
2600	循环处理未能在下一个网络通信周期的开始时机之前完成。
3402	运算处理及其他恒定周期处理未在设置的周期内完成。
3AF0	以中速周期进行的循环处理未能在下一个网络通信周期的开始时机之前完成。
3AF1	以低速周期进行的循环处理未能在下一个网络通信周期的开始时机之前完成。

*1 将CPU参数的“执行间隔超出(网络通信周期同步)”设置为“检测”时发生。

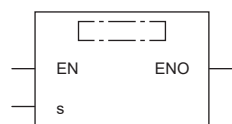
指定优先级及其以下的中断禁止

DI

禁止指定的优先级及其以下的中断程序的执行。


梯形图	ST
	ENO:=DI_1(EN, s);

FBD/LD



(□中放入DI_1。)

■执行条件

指令	执行条件
DI	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	禁止中断的优先级	1~32	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

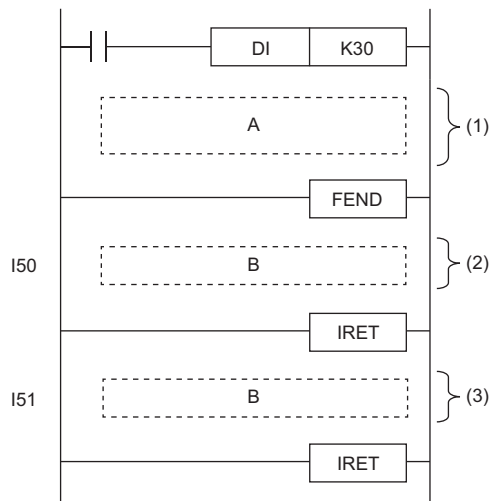
操作数	位		字		双字		间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K	H	
(s)	○	—	○	—	○	—	—	○	—	—	—

功能

- 禁止(s)中指定的中断优先级及其以下的中断指针编号的中断程序。

中断优先级设置

I No.	优先级
I50	29
I51	30

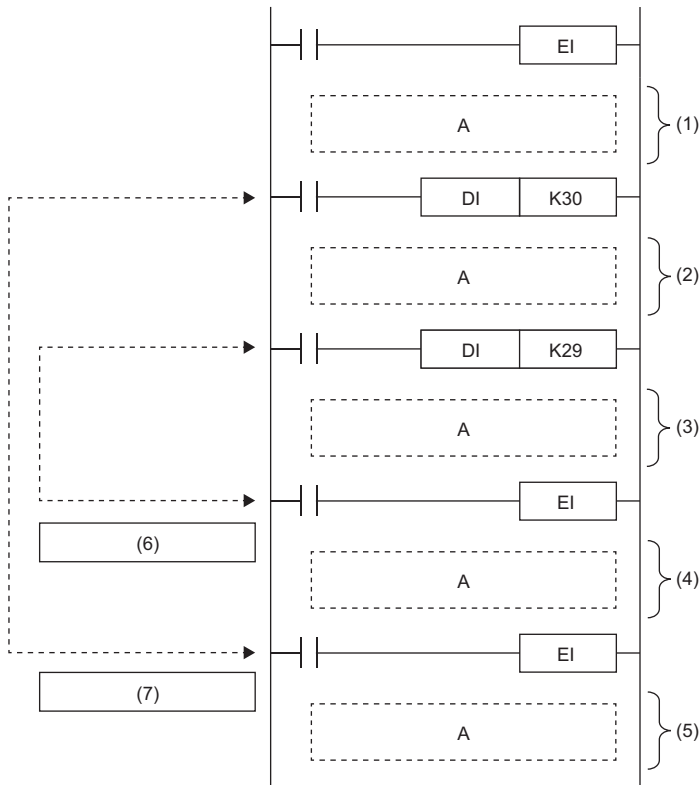


A: 顺控程序

B: 中断程序

- 优先级30及以下的中断禁止区间(优先级29及以上的中断允许区间)
- 优先级29的中断因此可以执行
- 优先级30的中断因此禁止执行

- 通过EI指令的执行，将通过DI指令禁止的优先度的中断置为允许。但是，仅通过DI(中断禁止)指令置为中断禁止的情况下，仅执行1次EI指令便将所有优先度的中断置为允许。

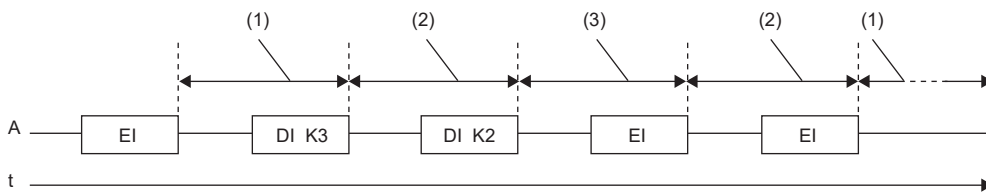


A: 顺控程序

- (1) 所有优先度的中断允许区间
- (2) 优先级30及以下的中断禁止区间(优先级29及以上的中断允许区间)
- (3) 优先级29及以下的中断禁止区间(优先级28及以上的中断允许区间)
- (4) 优先级30及以下的中断禁止区间(优先级29及以上的中断允许区间)
- (5) 所有优先度的中断允许区间
- (6) [DI K29]对应的EI指令
- (7) [DI K30]对应的EI指令

- 多次执行DI(指定优先级及其以下的中断禁止)指令时，指定的中断禁止优先级低于当前禁止的中断禁止优先级的情况下，当前的中断禁止优先级不被更改。
- DI指令的嵌套最多可达16层。
- 中断指针的中断优先级可通过参数进行设置。(📖(所使用的控制器的用户手册))
- 处于中断禁止的优先级，可以通过SD758(中断禁止优先级设定值)确认。
- 中断程序中执行DI(指定优先级以下的中断禁止)指令后，中断禁止优先级被更改的情况下SD758的值也将被更改。

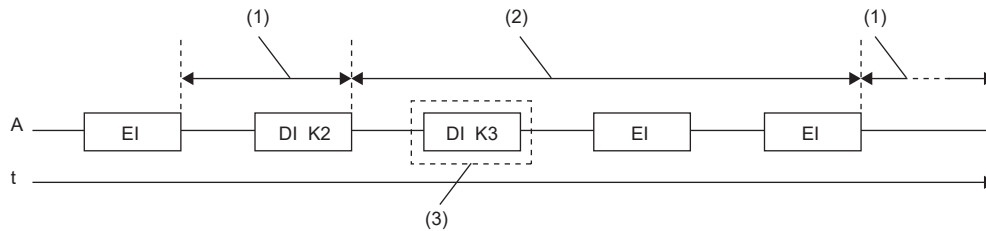
- 执行DI(中断禁止)指令、DI(指定优先级及其以下的中断禁止)指令及EI指令时的中断禁止区间如下所示。
- 多次执行了DI(指定优先级及其以下的中断禁止)指令的情况下(对高于当前禁止的中断优先级的优先度的中断执行了禁止指定的情况下)



A: 扫描执行类型程序

- (1) 所有优先度的中断允许区间
- (2) 优先级3及以下的中断禁止区间(优先级2及以上的中断允许区间)
- (3) 优先级2及以下的中断禁止区间(优先级1及以上的中断允许区间)

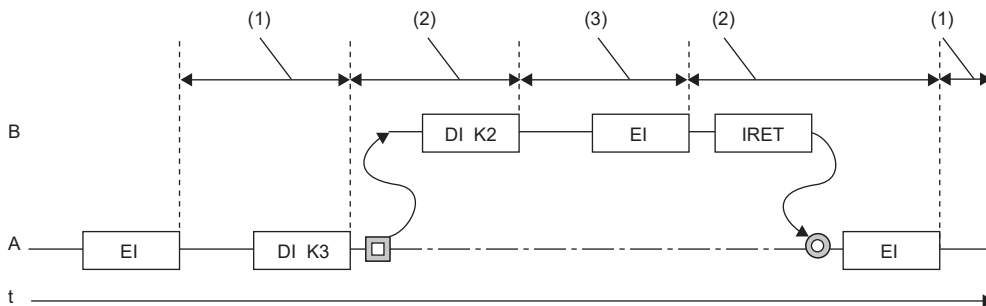
- 多次执行了DI(指定优先级及其以下的中断禁止)指令的情况下(对低于当前禁止的中断优先级的优先度的中断进行了禁止指定的情况下)



A: 扫描执行类型程序

- (1) 所有优先度的中断允许区间
- (2) 优先级2及以下的中断禁止区间(优先级1及以上的中断允许区间)
- (3) 优先级2及其以下的中断已被禁止, 因此中断禁止优先级不更改。

- 通过中断程序执行了DI(指定优先级及其以下的中断禁止)指令的情况下

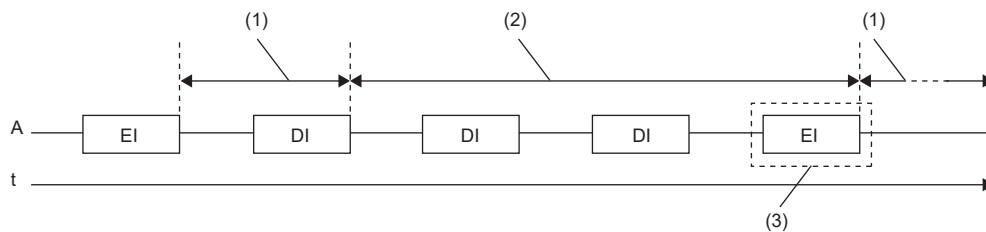


A: 扫描执行类型程序

B: 中断程序

- (1) 所有优先度的中断允许区间
- (2) 优先级3及以下的中断禁止区间(优先级2及以上的中断允许区间)
- (3) 优先级2及以下的中断禁止区间(优先级1及以上的中断允许区间)

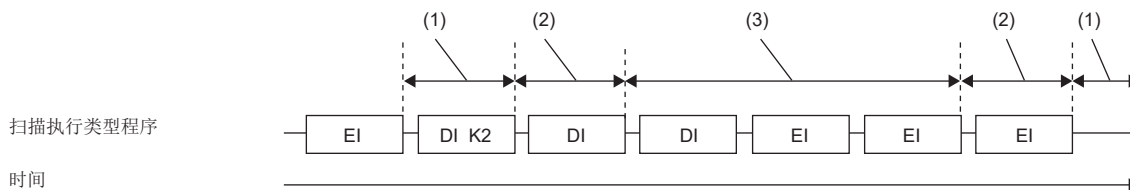
- 仅执行了DI(中断禁止)指令的情况下



A: 扫描执行类型程序

- (1) 所有优先度的中断允许区间
- (2) 优先级1及以下的中断禁止区间(所有的中断禁止区间)
- (3) 通过DI(中断禁止)指令置为中断禁止, 因此通过执行1次EI指令所有优先度的中断将被允许。

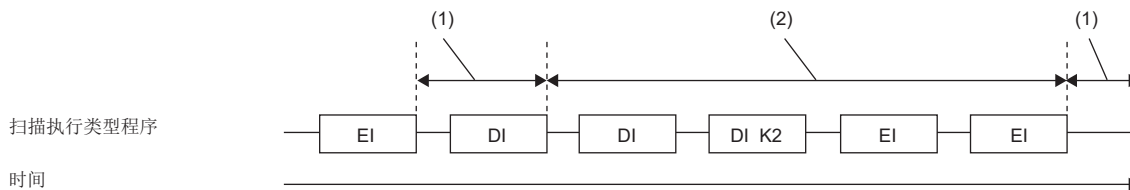
- 执行了DI (指定优先级及其以下的中断禁止) 指令及DI (中断禁止) 指令的情况下(按照DI (指定优先级及其以下的中断禁止) 指令→DI (中断禁止) 指令的顺序执行的情况下)



A: 扫描执行类型程序

- (1) 所有优先度的中断允许区间
- (2) 优先级2及以下的中断禁止区间 (优先级1及以上的中断允许区间)
- (3) 优先级1及以下的中断禁止区间 (所有的中断禁止区间)

- 执行了DI (指定优先级及其以下的中断禁止) 指令及DI (中断禁止) 指令的情况下(按照DI (中断禁止) 指令→DI (指定优先级及其以下的中断禁止) 指令的顺序执行的情况下)



A: 扫描执行类型程序

- (1) 所有优先度的中断允许区间
- (2) 优先级1及以下的中断禁止区间 (所有的中断禁止区间)

- 初始执行类型程序中的DI指令可以执行。但是，在初始执行类型程序中执行了DI指令时，在第2个扫描开始之前指定优先度的设置为有效。从执行第2个扫描的程序时开始变为执行了EI指令的状态。

出错

错误代码 (SD0)	内容
3285H	(s) 中指定的优先级为范围外时。
3332H	通过DI (中断禁止) 指令、DI (指定优先级及其以下的中断禁止) 指令进行的嵌套超过了16层的情况下。

中断程序屏蔽

IMASK

将指定的中断指针编号的中断程序置为执行允许状态/执行禁止状态。

梯形图	ST
	ENO:=IMASK(EN, s);

FBD/LD

■执行条件

指令	执行条件
IMASK	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	中断屏蔽数据或存储了中断屏蔽数据的起始软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 16)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 根据(s)中指定的软元件开始的16点的位模式，将指定的中断指针编号的中断程序置为执行允许状态或执行禁止状态。
- 1(ON)：中断程序的执行允许状态
- 0(OFF)：中断程序的执行禁止状态
- 各位对应的中断指针编号如下所示。

	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

- 电源投入时或进行了控制器复位的情况下，I0～I31、I44～I255的中断程序将变为执行允许状态，I32～I43的中断程序将变为执行禁止状态。
- (s)～(s)+15的软元件的状态将被存储到SD1400～SD1415(IMASK指令屏蔽模式)中。

要点

在IMASK指令中，可以将I0～I255的中断指针批量置为执行允许状态或执行禁止状态。

将I0～I255的中断指针置为执行允许状态或执行禁止状态的情况下，应通过程序将IMASK指令替换为SIMASK指令。

关于SIMASK指令的详细内容，请参阅下述内容。

☞ 530页 SIMASK

出错

没有运算错误。

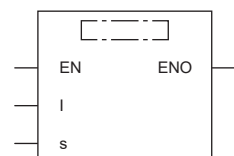
指定中断指针的禁止/允许

SIMASK

将指定的中断指针编号的中断程序置为执行允许状态/执行禁止状态。

梯形图	ST
	ENO:=SIMASK(EN, I, s);

FBD/LD



■执行条件

指令	执行条件
SIMASK	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(I)	对中断进行允许/禁止设置的中断指针No.	I0~I255	软元件名	POINTER*1
(s)	指定的中断指针No.的允许/禁止	0: 禁止 1: 允许	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能使用软元件(I)中分配的标签。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它(I)	
	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		\$
(I)	—	—	—	—	—	—	—	—	—	—	—	○
(s)	—	—	○	—	○	—	—	○	○	—	—	—

功能

- 将(I)中指定的中断指针编号的中断程序，按照(s)中指定的数据置为执行允许状态或执行禁止状态。
- (s)为1的情况下：中断程序的执行允许状态
- (s)为0的情况下：中断程序的执行禁止状态
- 电源投入时或进行了控制器复位的情况下，I0~I31、I44~I255的中断程序将变为执行允许状态，I32~I43的中断程序将变为执行禁止状态。
- 中断指针的执行允许状态/执行禁止状态将被存储到SD1400~SD1415(IMASK指令屏蔽模式)中。

要点

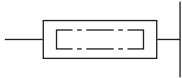
(I)可以进行变址修饰。通过使用变址修饰后的SIMASK指令，可以将I0~I255的中断指针置为执行允许状态或执行禁止状态。

错误代码 (SD0)	内容
3285H	(I) 中指定的中断指针No. 超过了中断指针No. 的范围时。 (s) 的值为中断禁止 (0)/中断允许 (1) 以外时。

从中断程序返回

IRET

表示中断程序的处理结束。

梯形图	ST
	不对应。

FBD/LD

不对应。

■执行条件

指令	执行条件
IRET	常时执行

功能

- 表示通过中断指针(I)进行的中断程序的处理结束。
- 执行IRET指令后，处理将返回至顺控程序。

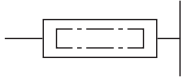
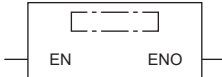
出错

错误代码(SD0)	内容
3350H	中断No. 中对应的指针不存在时。
3351H	发生中断后执行IRET指令前，执行了END指令、FEND指令、GOEND指令、STOP指令时。
3352H	在主程序中执行FEND指令前执行了IRET指令时。
3353H	恒定周期执行类型程序中执行了IRET指令时。
3354H	事件执行类型程序中执行了IRET指令时。



WDT复位

WDT (P)

复位看门狗定时器。

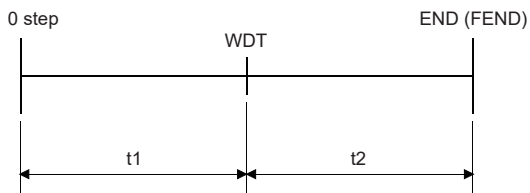
梯形图	ST
	ENO:=WDT (EN) ; ENO:=WDTP (EN) ;
FBD/LD	
	

■执行条件

指令	执行条件
WDT	
WDTP	

功能

- 通过程序对看门狗定时器进行复位。
- 在扫描时间根据条件超过了看门狗定时器的设定值的情况下使用。扫描时间超过了每个扫描看门狗定时器的设定值的情况下，应在工程工具的参数设置中更改看门狗定时器的设定值。
- 对从0步开始至WDT (P) 指令为止的t1、及从WDT (P) 指令开始至END (FEND) 指令为止的t2，请勿设为超过看门狗定时器的设定值。



- WDT (P) 指令在1个扫描中可以使用2次及其以上，但应注意至发生异常时的输出OFF为止需要耗费一定的时间。
- 即使执行WDT (P) 指令，特殊寄存器中存储的扫描时间的值也不被清除。因此，特殊寄存器的扫描时间值有可能会大于参数中设置的看门狗定时器的设定值。

出错

没有运算错误。

13.3 结构化指令

FOR~NEXT

FOR、NEXT

对FOR~NEXT指令之间的处理执行(n)次。

梯形图	ST
<p>A: 重复程序</p>	不对应。

FBD/LD	[NEXT]
[FOR]	[NEXT]

■执行条件

指令	执行条件
FOR NEXT	常时执行

设置数据

■内容、范围、数据类型

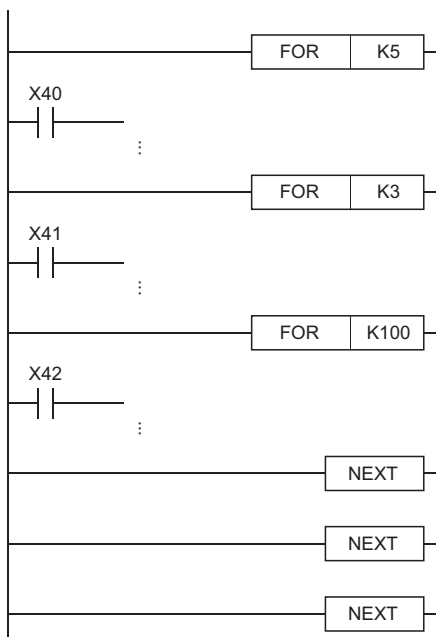
操作数	内容	范围	数据类型	数据类型(标签)
(n)	FOR~NEXT指令之间的重复次数	1~32767	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 将FOR~NEXT指令之间的处理无条件执行(n)次时,将进行NEXT指令的下一个步的处理。
- (n)可在1~32767的范围内指定。指定-32768~0的情况下,将进行与(n)=1相同的处理。
- 不希望执行FOR~NEXT指令之间的处理时,应通过CJ指令、SCJ指令跳转。
- FOR指令的嵌套最多可达16层。

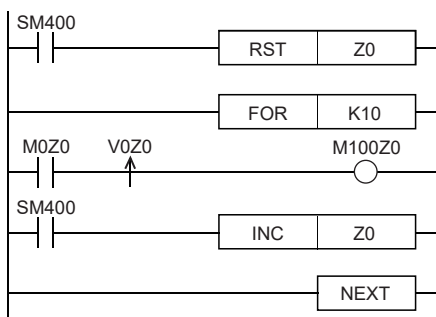


注意事项

- FOR~NEXT指令之间的重复执行中,中途结束的情况下,应使用BREAK指令。

☞ 537页 BREAK(P)

- FOR~NEXT指令之间进行变址修饰后的程序的脉冲运算的情况下,应使用EGP指令、EGF指令。但是,动作输出侧不能使用上升沿指令、下降沿指令。



- 从FOR~NEXT指令之间之外到FOR~NEXT指令内,从FOR~NEXT指令之间之内到FOR~NEXT指令外,无法通过JMP指令等进行分支,变为转换错误。
- FOR~NEXT指令内的JMP指令等中指定的指针软元件/标签进行了变址修饰的情况下,将无法分支,变为转换错误。
- 从FOR~NEXT指令之间的外部,无法通过CALL(P)指令在FOR~NEXT指令内进行分支,变为转换错误。但是,通过CALL(P)指令对分支目标中指定的指针软元件/标签进行变址修饰,变址修饰解决后指定了FOR~NEXT内部存在的指针时,则在CALL(P)指令执行时变为错误(错误代码:3340H)。CALL(P)指令的详细内容请参阅下述章节。

☞ 539页 CALL(P)

- FOR~NEXT指令内部为嵌套构造时,只有当JMP指令等指定的指针软元件/标签与执行源的JMP指令等为同一分层的情况下才可以记述。处于不同分层记述时将变为转换错误。

出错

没有运算错误。

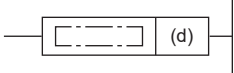
要点

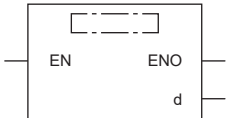
为了创建易于理解的程序，成对的FOR指令及NEXT指令，应在一个程序块内使用。

FOR~NEXT强制结束


BREAK (P)

强制结束通过FOR~NEXT指令进行的重复处理，执行将转移至FOR~NEXT指令及以后的处理。

梯形图	ST
	不对应。

FBD/LD


■执行条件

指令	执行条件
BREAK	
BREAKP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	存储重复剩余数的软元件编号	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

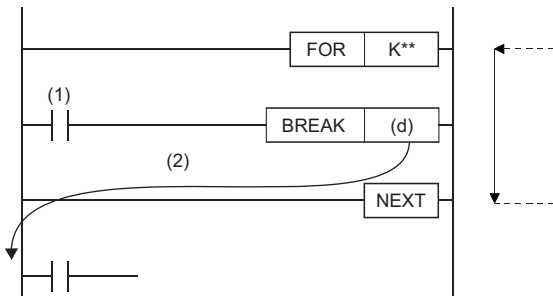
■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 强制结束通过FOR~NEXT指令进行的重复处理，执行将转移至FOR~NEXT指令及以后的处理。

未执行BREAK指令的情况下，以FOR指令中指定的次数返回至FOR指令。



(1) 强制结束条件

(2) 强制结束条件成立时

- (d) 中将存储强制结束时的FOR~NEXT指令中重复处理执行次数的剩余数。但是，重复处理的剩余数中还包含有BREAK(P)指令执行时的次数。
- BREAK(P)指令只能在FOR~NEXT指令之间使用。
- BREAK(P)指令只能对1个嵌套使用。强制结束多重嵌套的情况下，应执行嵌套重数对应次数的BREAK(P)指令。

要点

引用其他系列的程序时，应在NEXT指令结束后立即记述分支目标指针中记述的强制结束条件成立时的处理。

出错

没有运算错误。

子程序调用

CALL (P)

执行指定的指针的子程序。

梯形图	ST
	不对应。

FBD/LD

■执行条件

指令	执行条件
CALL	
CALLP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(P)	子程序的起始指针编号	—	软元件名	POINTER
(s1)~(s5)*1	子程序中作为自变量传送的软元件编号	-2147483648~2147483647	位/带符号BIN16位/带符号BIN32位	ANY
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 在FBD/LD中不能指定。

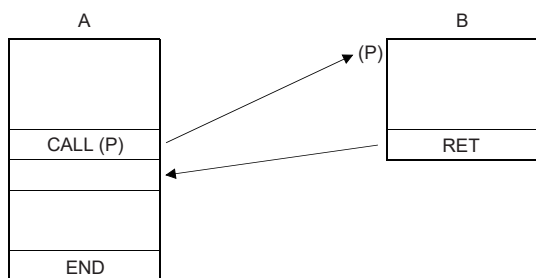
■可用软元件

操作数	位		字		双字			间接指定	常数			其它 (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以外可以使用。

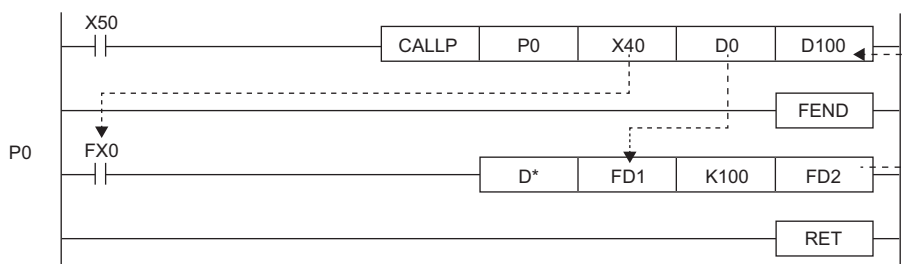
功能

- 执行CALL(P)指令时，将执行指针(P)的子程序。CALL(P)指令可以执行同一程序文件内的指针中指定的子程序及全局指针中指定的子程序。



A: 主程序
B: 子程序

- 通过子程序使用函数软元件(FX、FY、FD)时，在(s1)~(s5)中指定函数软元件对应的软元件。(s1)~(s5)中指定的软元件的内容如下所示。



- 执行子程序前位数据的内容将被传送到FX中，字数据的内容将被传送到FD中。
- 执行子程序后，将被传送到FY、FD的内容对应的软元件中。
- FX、FY、FD的数据传送大小根据(s1)~(s5)中指定的数据类型而有所不同。

■指定了软元件的情况下

FX、FY的函数软元件的传送单位为位单位。FD的函数软元件的传送单位为4字单位。根据参数中指定的软元件的类型，处理的数据大小有所不同。

函数软元件中指定的软元件，应预留出数据大小相应的区域。无法预留出数据大小相应的区域的情况下将变为错误状态。各函数软元件的数据大小如下所示。

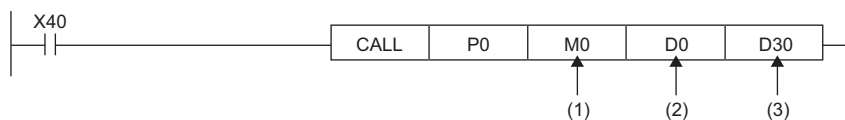
函数软元件	使用软元件	从(s1)~(s5)传送至函数软元件时的数据大小	从函数软元件传送至(s1)~(s5)时的数据大小
FX、FY	位软元件	1位	1位
	字软元件的位指定时	1位	1位
FD	位软元件的位指定时*1	4字	位指定的容量或4字*2
	字软元件	4字	4字

*1 位软元件的位指定时，即使(s1)~(s5)中指定的软元件编号不是16的倍数，也可以指定。

*2 软元件编号为16的倍数，且位指定为K4或K8的情况下传送4字大小。

例

指定了软元件的情况下，将变为如下动作。



- (1) 将M0传送至FX0。
- (2) 将D0~D3传送至FD1。
- (3) 将D30~D33传送至FD2。

■指定了标签的情况下

从(s1)~(s5)传送至函数软元件时，FX、FY的函数软元件的传送单位为位单位。FD的函数软元件的传送单位为4字单位。

从函数软元件传送至(s1)~(s5)时，FX、FY的函数软元件的传送单位为位单位。FD的函数软元件的传送单位为标签的分配大小单位。

各函数软元件的数据大小如下所示。

函数软元件	标签的分配大小	从(s1)~(s5)传送至函数软元件时的数据大小	从函数软元件传送至(s1)~(s5)时的数据大小
FX、FY	1位	1位	1位
FD	2位及以上	4字*1	标签的分配大小*2

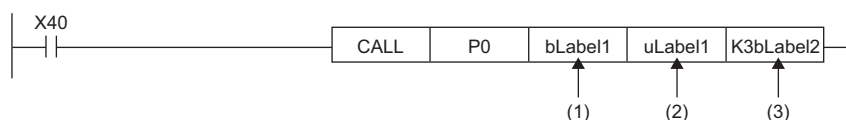
*1 指定的标签/结构体的分配大小不足4字的情况下，不足4字的区域将传送0。

*2 指定的标签/结构体的分配大小为4字及以上的情况下，将传送4字。

例

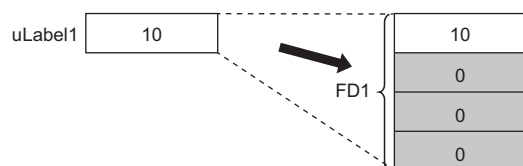
从(s1)~(s5)传送至函数软元件时，将执行以下动作。

标签名	标签的种类
bLabel1	位型标签
uLabel1	字型标签
bLabel2	数组位型标签

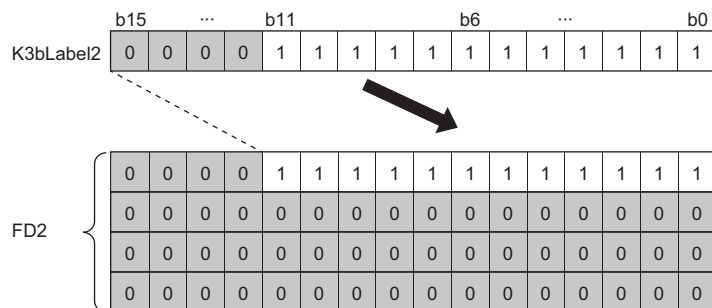


(1) 将bLabel1(1位)传送至FX0。

(2) 将uLabel1(1字)传送至FD1。分配大小不足4字，因此不足4字的区域将传送0。



(3) 将K3bLabel2的1字传送至FD2。分配大小不足4字，因此不足4字的区域将传送0。

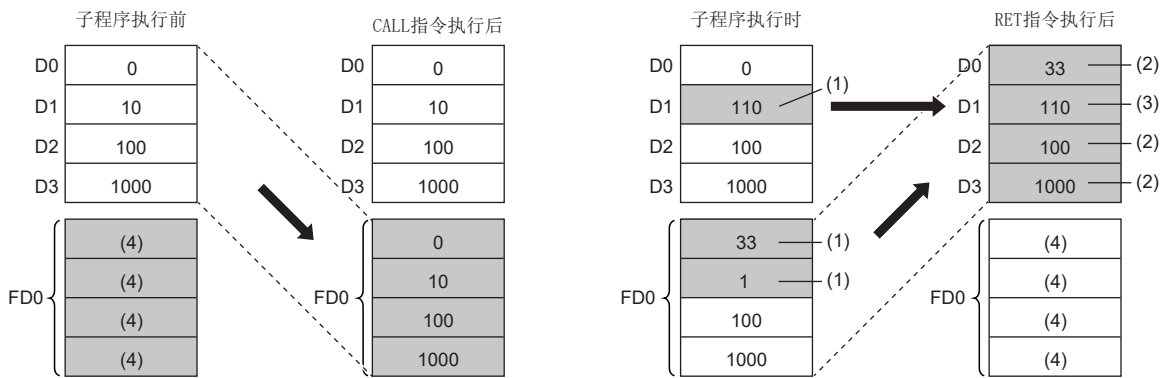
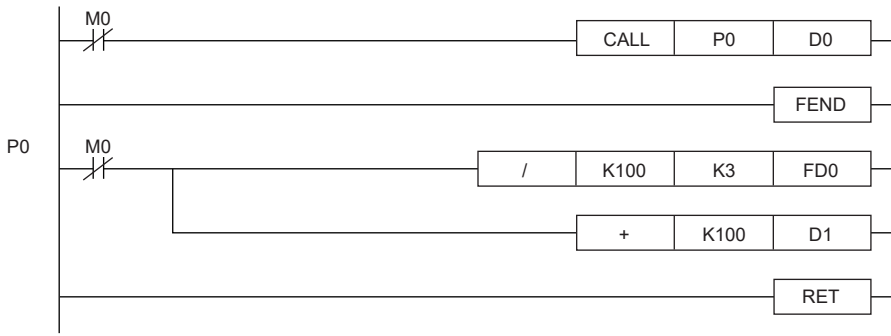


- 在CALL(P)指令中，可以使用(s1)~(s5)。
- 子程序中使用的函数软元件数及CALL(P)指令的自变量，应设置为相同的数。此外，应将函数软元件与CALL(P)指令的自变量的类型设置为相同。
- CALL(P)指令的自变量的软元件中使用了定时器或计数器的情况下，仅对当前值进行发送接收。
- CALL(P)指令中自变量中指定的软元件编号请勿重复。重复的情况下将无法正常运转。

• 请勿将CALL(P)指令的自变量中使用的软元件用于子程序。使用的情况下，将无法正常运转。

例

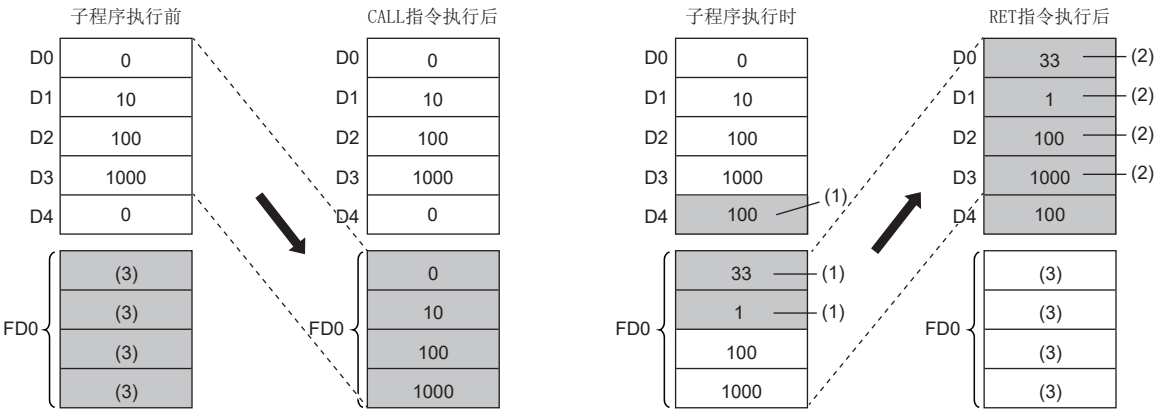
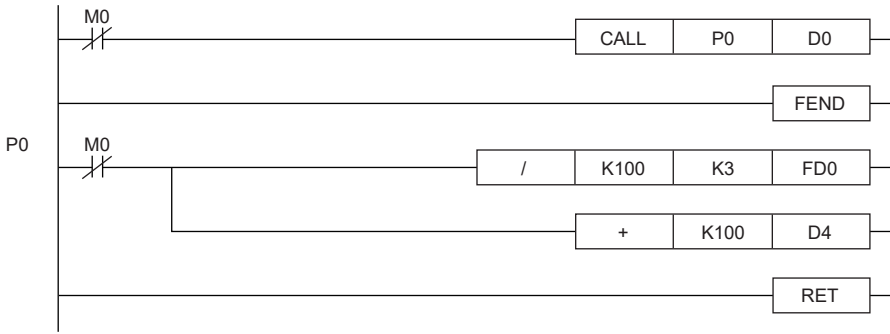
[错误的运算]子程序的FD0中指定D0，子程序中使用D1情况下的动作



- (1) 存储子程序的执行结果。
- (2) 替换为函数软元件的值。
- (3) D1不反映函数软元件的值。
- (4) 变为不定值。

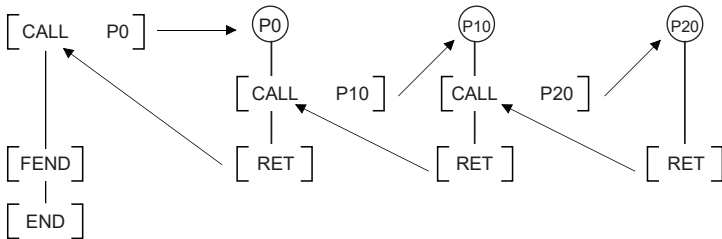
例

[正确运算]子程序的FD0中指定D0，通过子程序使用了D4时的动作



- (1) 存储子程序的执行结果。
- (2) 替换为函数软元件的值。
- (3) 变为不定值。

• CALL(P) 指令的嵌套最多可达16层。但是，嵌套的16层指的是CALL(P) 指令、FCALL(P) 指令、ECALL指令(P)、EFCALL(P) 指令、XCALL指令中的合计值。



• 子程序内置为ON的软元件，在子程序非执行时也将保持不变。执行子程序时置为ON的软元件，可以通过FCALL(P) 指令置为OFF。

注意事项

- FBD/LD程序不能创建为子程序。
- 在FBD/LD中，不能执行具有自变量的子程序。
- 如果(P)中指定的指针进行了变址修饰，则变址修饰解决后指定了FOR~NEXT内部存在的指针时，将变为错误(错误代码：3340H)。此外，不进行变址修饰，直接指定了FOR~NEXT内部存在的指针时，将变为转换错误。

出错

错误代码(SD0)	内容
2820H	(s1)~(s5)中自变量中指定的软元件无法预留数据大小区域时。
3330H	执行了第17重的嵌套时。
3340H	(P)中指定的指针的子程序不存在时。 或者对(P)中指定的指针进行变址修饰，则变址修饰解决后指定了FOR~NEXT内部存在的指针时。
3341H	执行CALL(P)指令后，执行RET指令前执行了END指令、FEND指令、GOEND指令、STOP指令时。
3342H	执行CALL(P)指令前执行了RET指令时。

从子程序的返回

RET

表示子程序的结束。

梯形图	ST
	不对应。

FBD/LD

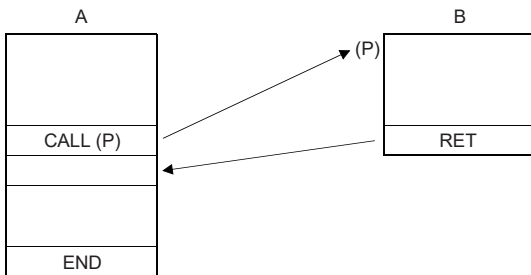
不对应。

■执行条件

指令	执行条件
RET	常时执行

功能

- 表示子程序的结束。
- 执行RET指令时，将返回至调用了子程序的CALL(P)指令、FCALL(P)指令、ECALL(P)指令、EFCALL(P)指令、XCALL指令的下一个步处。



A: 主程序
B: 子程序

出错

错误代码(SD0)	内容
3341H	执行CALL(P)指令、FCALL(P)指令、ECALL(P)指令、EFCALL(P)指令、XCALL指令执行后，执行RET指令前执行了END指令、FEND指令、GOEND指令、STOP指令时。
3342H	执行CALL(P)指令、FCALL(P)指令、ECALL(P)指令、EFCALL(P)指令、XCALL指令前执行了RET指令时。

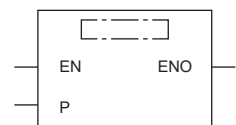
子程序的输出OFF调用

FCALL (P)

进行指定的指针的子程序的非执行处理。

梯形图	ST
 	不对应。

FBD/LD



■执行条件

指令	执行条件
FCALL	
FCALLP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(P)	子程序的起始指针编号	—	软元件名	POINTER
(s1) ... (s5)*1	子程序中作为自变量传送的软元件编号	-2147483648~2147483647	位/带符号BIN16位/带符号BIN32位	ANY
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 在FBD/LD中不能指定。

■可用软元件

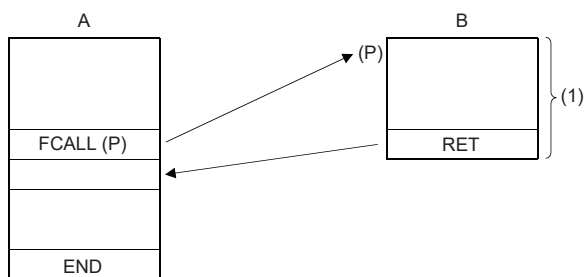
操作数	位		字		双字			间接指定	常数				其它(P)	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K	H	E	\$		
(P)	—	—	—	—	—	—	—	—	—	—	—	—	—	○
(s1) ... (s5)	○*1	○	○	○	○	○	○	○	○	○	—	—	—	—

*1 F以外可以使用。

功能

- 如果执行FCALL (P) 指令，将进行指针 (P) 的子程序的非执行处理*2。FCALL (P) 指令可以对同一程序文件内的指针中指定的子程序及全局指针中指定的子程序进行非执行。

*2 非执行处理时，将进行与将各线圈指令的条件设置为OFF状态时相同的处理。



A: 主程序

B: 子程序

(1) FCALL (P) 指令的指令由ON→OFF时进行非执行处理。

- 非执行处理后的各线圈指令的运算结果与条件触点的ON/OFF无关，其情况如下所示。

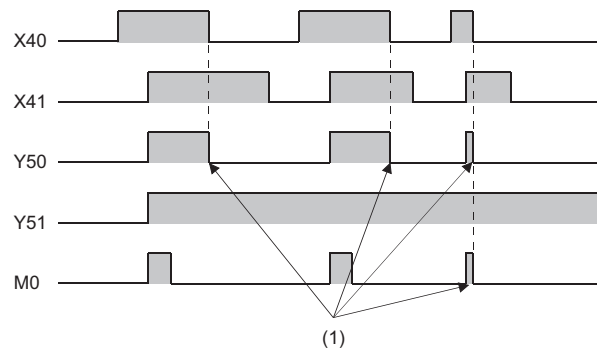
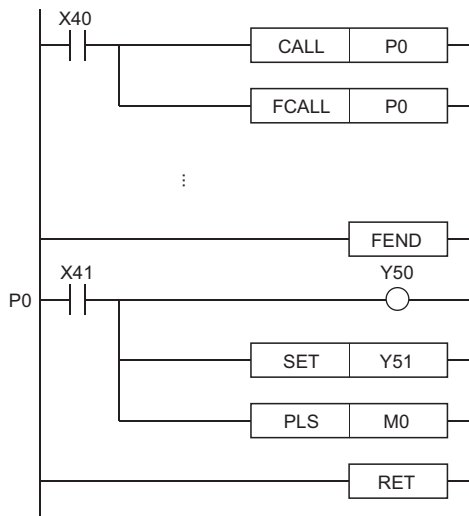
运算中使用的软元件	运算结果(软元件的状态)
高速定时器、低速定时器	变为0。
高速累计定时器、低速累计定时器、计数器	保持当前的状态。
OUT指令中的软元件	变为强制OFF。
SET指令、RST指令中的软元件、SFT (P) 指令中的软元件、基本/应用指令中的软元件	保持当前的状态。
PLS指令、脉冲化指令(□P)	变为与条件触点OFF相同的处理。

- FCALL (P) 指令与CALL (P) 指令组合使用。未将FCALL (P) 指令与CALL (P) 指令组合执行的情况下，即使将执行指令置为OFF也不进行子程序的非执行处理，因此各线圈指令的输出状态将保持。

- 执行指令变为OFF时进行子程序的非执行处理，因此可以将OUT指令、PLS指令(包括脉冲化指令)强制置为OFF。

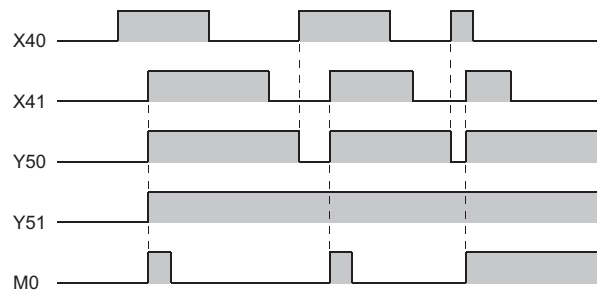
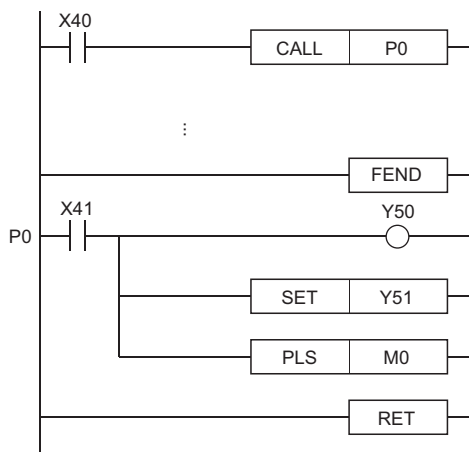
例

使用FCALL(P)指令时

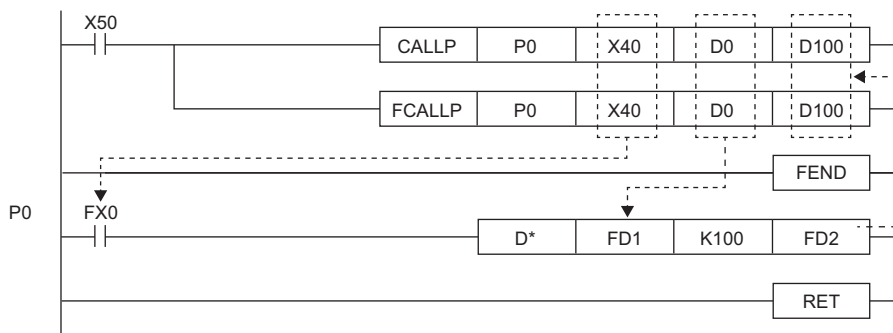


(1) 根据FCALL指令强制OFF

不使用FCALL(P)指令时



- 通过子程序使用函数软元件(FX、FY、FD)时，在(s1)~(s5)中指定函数软元件对应的软元件。(s1)~(s5)中指定的软元件的内容如下所示。



- 执行子程序前，位数据的内容将被传送到FX中，字数据的内容将被传送到FD中。
- 执行子程序后，将被传送到FY、FD的内容对应的软元件中。
- FX、FY、FD的数据传送大小根据(s1)~(s5)中指定的数据类型而有所不同。

■指定了软元件的情况下

FX、FY的函数软元件的传送单位为位单位。FD的函数软元件的传送单位为4字单位。根据参数中指定的软元件的类型，处理的数据大小有所不同。

函数软元件中指定的软元件，应预留出数据大小相应的区域。无法预留出数据大小相应的区域的情况下将变为错误状态。各函数软元件的数据大小如下所示。

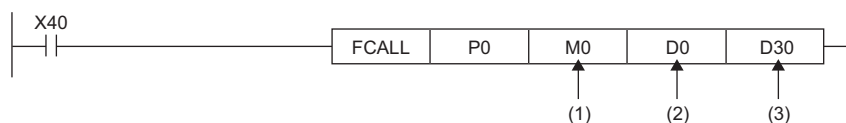
函数软元件	使用软元件	从(s1)~(s5)传送至函数软元件时的数据大小	从函数软元件传送至(s1)~(s5)时的数据大小
FX、FY	位软元件	1位	1位
	字软元件的位指定时	1位	1位
FD	位软元件的位指定时*1	4字	位指定的容量或4字*2
	字软元件	4字	4字

*1 位软元件的位指定时，即使(s1)~(s5)中指定的软元件编号不是16的倍数，也可以指定。

*2 软元件编号为16的倍数，且位指定为K4或K8的情况下传送4字大小。

例

指定了软元件的情况下，将变为如下动作。



(1) 将M0传送至FX0。

(2) 将D0~D3传送至FD1。

(3) 将D30~D33传送至FD2。

■指定了标签的情况下

从(s1)~(s5)传送至函数软元件时，FX、FY的函数软元件的传送单位为位单位。FD的函数软元件的传送单位为4字单位。

从函数软元件传送至(s1)~(s5)时，FX、FY的函数软元件的传送单位为位单位。FD的函数软元件的传送单位为标签的分配大小单位。

各函数软元件的数据大小如下所示。

函数软元件	标签的分配大小	从(s1)~(s5)传送至函数软元件时的数据大小	从函数软元件传送至(s1)~(s5)时的数据大小
FX、FY	1位	1位	1位
FD	2位及以上	4字*1	标签的分配大小*2

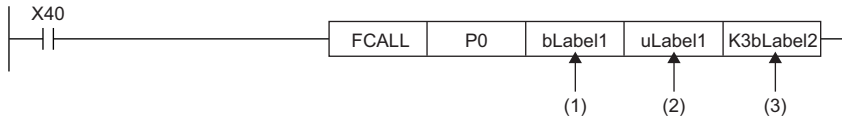
*1 指定的标签/结构体的分配大小不足4字的情况下，不足4字的区域将传送0。

*2 指定的标签/结构体的分配大小为4字及以上的情况下，将传送4字。

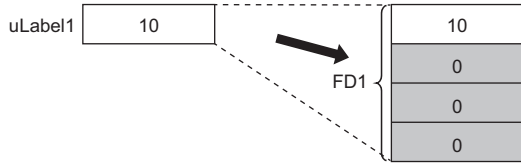
例

从(s1)~(s5) 传送至函数软元件时，将执行以下动作。

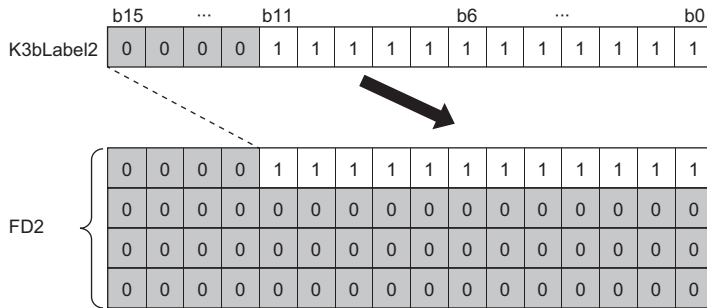
标签名	标签的种类
bLabel1	位型标签
uLabel1	字型标签
bLabel2	数组位型标签



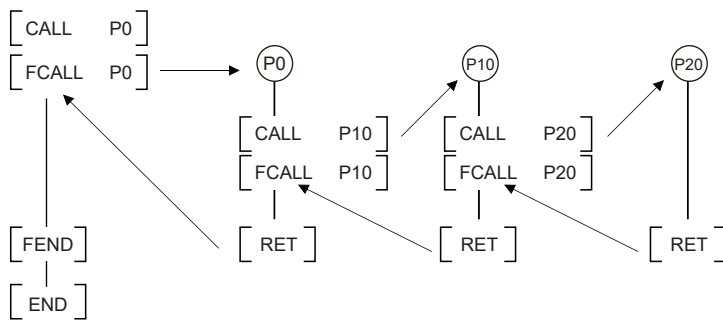
- (1) 将bLabel1(1位) 传送至FX0。
- (2) 将uLabel1(1字) 传送至FD1。分配大小不足4字，因此不足4字的区域将传送0。



- (3) 将K3bLabel2的1字传送至FD2。分配大小不足4字，因此不足4字的区域将传送0。



- 在FCALL(P) 指令中，可以使用(s1)~(s5)。
- FCALL(P) 指令的嵌套可达16层。但是，嵌套的16层指的是CALL(P) 指令、FCALL(P) 指令、ECALL(P) 指令、EFCALL(P) 指令、XCALL指令的合计值。



注意事项

- FBD/LD程序不能创建为子程序。
- 在FBD/LD中，不能执行具有自变量的子程序。

出错

错误代码 (SD0)	内容
2820H	(s1)~(s5)中自变量中指定的软元件无法预留数据大小区域时。
3330H	执行了第17重的嵌套时。
3340H	FCALL(P)指令中指定的指针的子程序不存在时。 或者对(P)中指定的指针进行变址修饰，则变址修饰解决后指定了FOR~NEXT内部存在的指针时。
3341H	执行FCALL(P)指令后，执行RET指令前执行了END指令、FEND指令、GOEND指令、STOP指令时。
3342H	执行FCALL(P)指令前执行了RET指令时。

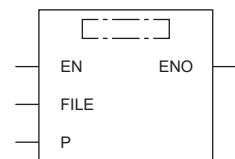
程序文件之间子程序调用

ECALL (P)

执行指定程序文件名的指定的指针的子程序。

梯形图	ST
<p>FILE: 文件名</p>	不对应。

FBD/LD



FILE: 文件名

■执行条件

指令	执行条件
ECALL	
ECALLP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(文件名)	调用的程序文件名	—	Unicode字符串	ANYSTRING_DOUBLE
(P)	子程序的起始指针编号	—	软元件名	POINTER
(s1) ... (s5)*1	子程序中作为自变量传送的软元件编号	-2147483648~2147483647	位/带符号BIN16位/带符号BIN32位	ANY
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 在FBD/LD中不能指定。

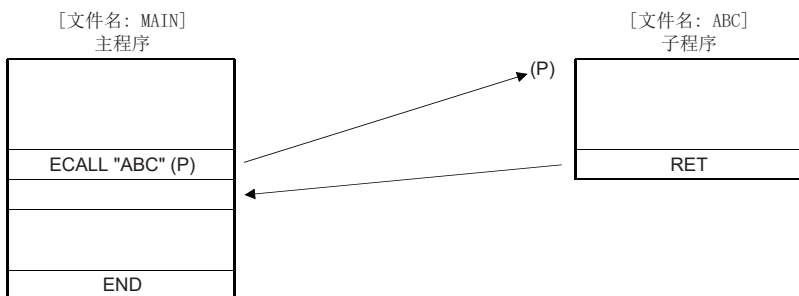
■可用软元件

操作数	位		字		双字			间接指定	常数				其它 (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	\$	
(文件名)	—	—	○	—	—	—	—	○	—	—	○	—	—
(P)	—	—	—	—	—	—	—	—	—	—	—	—	○
(s1) ... (s5)	○*1	○	○	○	○	○	○	○	○	—	—	—	—

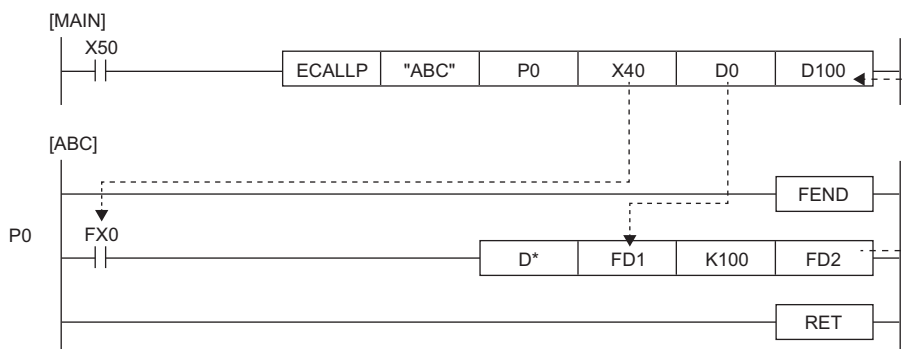
*1 F以外可以使用。

功能

- 执行ECALL (P) 指令时，执行指定程序文件名的指针 (P) 的子程序。在ECALL (P) 指令中，也可调用使用了其它程序文件的局部指针的子程序。



- 文件名中只能指定程序存储器中存储的程序文件。
- 文件名中无需指定扩展名 (.PRG)。 (仅 .PRG 文件作为对象。)
- 通过子程序使用函数软元件 (FX、FY、FD) 时，在 (s1) ~ (s5) 中指定函数软元件对应的软元件。 (s1) ~ (s5) 中指定的软元件的内容如下所示。



- 执行子程序前，位数据的内容将被传送到FX中，字数据的内容将被传送到FD中。
- 执行子程序后，将被传送到FY、FD的内容对应的软元件中。
- FX、FY、FD的数据传送大小根据 (s1) ~ (s5) 中指定的数据类型而有所不同。

■指定了软元件的情况下

FX、FY的函数软元件的传送单位为位单位。FD的函数软元件的传送单位为4字单位。根据参数中指定的软元件的类型，处理的数据大小有所不同。

函数软元件中指定的软元件，应预留出数据大小相应的区域。无法预留出数据大小相应的区域的情况下将变为错误状态。

各函数软元件的数据大小如下所示。

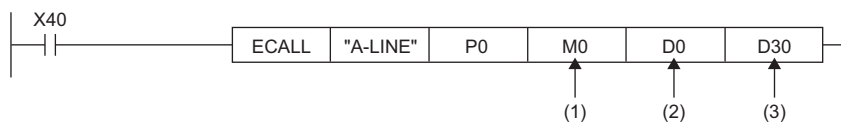
函数软元件	使用软元件	从(s1)~(s5)传送至函数软元件时的数据大小	从函数软元件传送至(s1)~(s5)时的数据大小
FX、FY	位软元件	1位	1位
	字软元件的位指定时	1位	1位
FD	位软元件的位指定时*1	4字	位指定的容量或4字*2
	字软元件	4字	4字

*1 位软元件的位指定时，即使(s1)~(s5)中指定的软元件编号不是16的倍数，也可以指定。

*2 软元件编号为16的倍数，且位指定为K4或K8的情况下传送4字大小。

例

指定了软元件的情况下，将变为如下动作。



(1) 将M0传送至FX0。

(2) 将D0~D3传送至FD1。

(3) 将D30~D33传送至FD2。

■指定了标签的情况下

从(s1)~(s5)传送至函数软元件时，FX、FY的函数软元件的传送单位为位单位。FD的函数软元件的传送单位为4字单位。

从函数软元件传送至(s1)~(s5)时，FX、FY的函数软元件的传送单位为位单位。FD的函数软元件的传送单位为标签的分配大小单位。

各函数软元件的数据大小如下所示。

函数软元件	标签的分配大小	从(s1)~(s5)传送至函数软元件时的数据大小	从函数软元件传送至(s1)~(s5)时的数据大小
FX、FY	1位	1位	1位
FD	2位及以上	4字*1	标签的分配大小*2

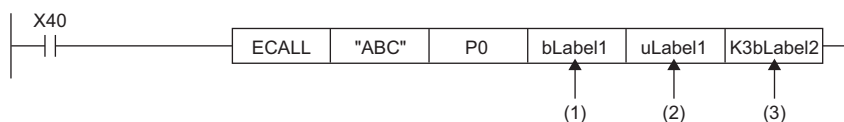
*1 指定的标签/结构体的分配大小不足4字的情况下，不足4字的区域将传送0。

*2 指定的标签/结构体的分配大小为4字及以上的情况下，将传送4字。

例

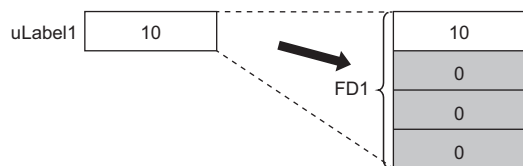
从(s1)~(s5)传送至函数软元件时，将执行以下动作。

标签名	标签的种类
bLabel1	位型标签
uLabel1	字型标签
bLabel2	数组位型标签

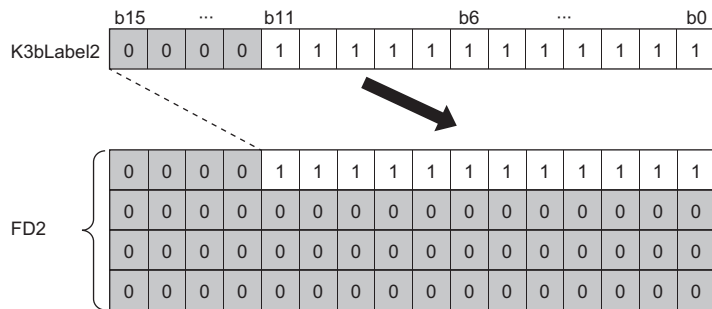


(1) 将bLabel1(1位)传送至FX0。

(2) 将uLabel1(1字)传送至FD1。分配大小不足4字，因此不足4字的区域将传送0。



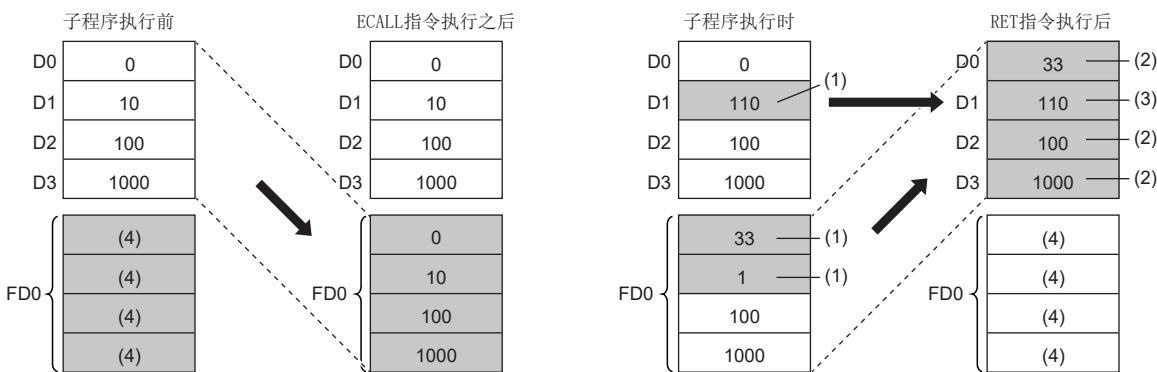
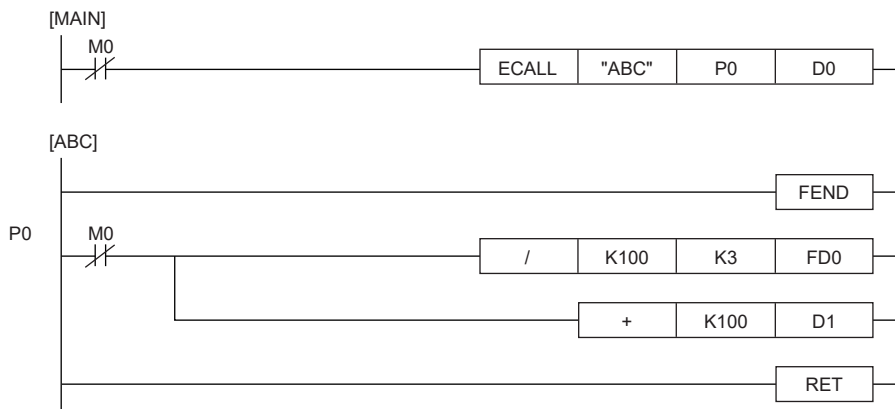
(3) 将K3bLabel2的1字传送至FD2。分配大小不足4字，因此不足4字的区域将传送0。



- 在ECALL (P) 指令中，可以使用(s1)~(s5)。
- 请勿将ECALL (P) 指令的自变量中使用的软元件用于子程序。将ECALL (P) 指令的自变量中使用的软元件用于子程序的情况下，将无法正常运算。

例

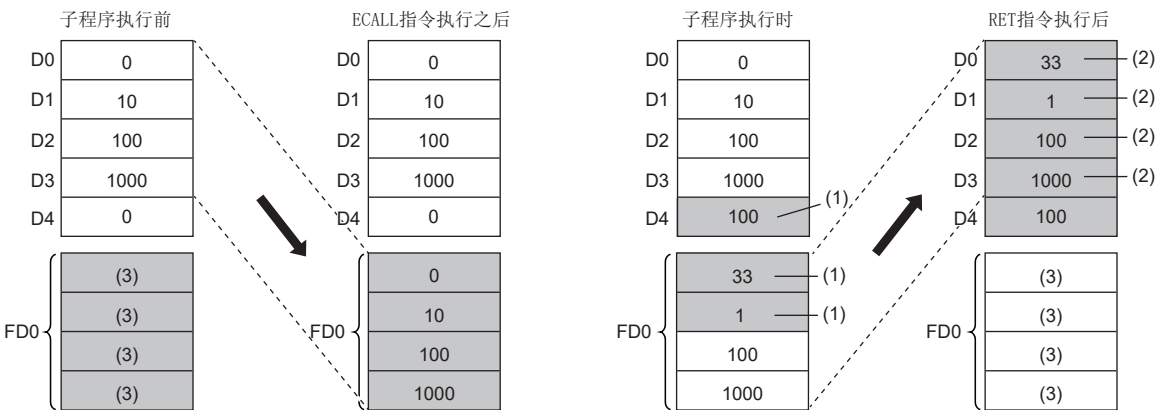
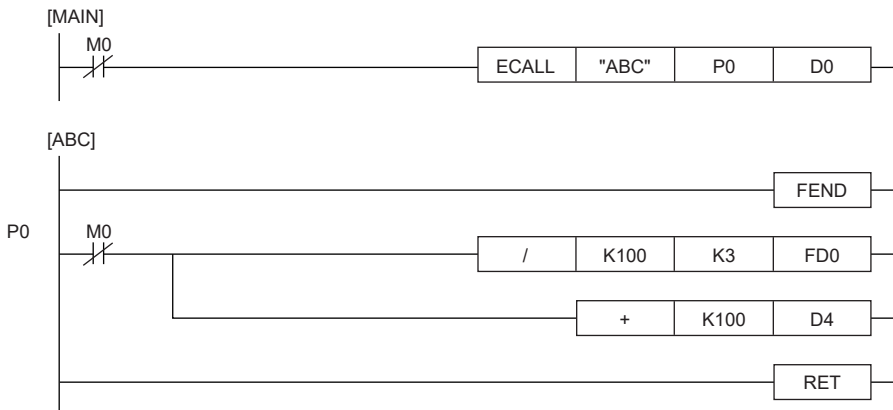
[错误的运算]子程序的FD0中指定D0，子程序中使用D1情况下的动作



- (1) 存储子程序的执行结果。
- (2) 替换为函数软元件的值。
- (3) D1不反映函数软元件的值。
- (4) 变为不定值。

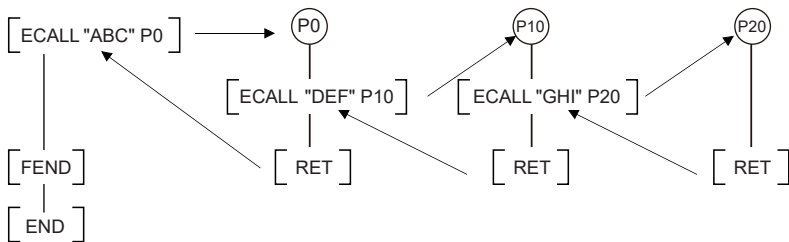
例

[正确运算]子程序的FD0中指定D0，通过子程序使用了D4时的动作



- (1) 存储子程序的执行结果。
- (2) 替换为函数软件的值。
- (3) 变为不定值。

- ECALL (P) 指令中自变量中指定的软元件编号请勿重复。重复的情况下，将无法正常运转。
- ECALL (P) 指令的嵌套可达16层。但是，嵌套的16层指的是CALL (P) 指令、FCALL (P) 指令、ECALL (P) 指令、EFCALL (P) 指令、XCALL指令的合计值。



- 子程序内置为ON的软元件，在子程序非执行时也将保持不变。对执行子程序时置为ON的软元件，可以通过EFCALL (P) 指令置为OFF。

注意事项

- FBD/LD程序不能创建为子程序。
- 在FBD/LD中，不能执行具有自变量的子程序。
- 如果(P)中指定的指针进行了变址修饰，则变址修饰解决后指定了FOR~NEXT内部存在的指针时，将变为错误(错误代码: 3340H)。此外，不进行变址修饰，直接指定了FOR~NEXT内部存在的指针时，将变为转换错误。

出错

错误代码 (SD0)	内容
2820H	(s1) ~ (s5) 中自变量中指定的软元件无法预留数据大小区域时。
2840H	(文件名) 中指定的文件不存在时。
2841H	(文件名) 中指定的文件无法执行时。
2842H	(文件名) 中指定的文件名的程序类型为不支持的程序时。
3330H	执行了第17重的嵌套时。
3340H	(P) 中指定的指针的子程序不存在时。 或者对 (P) 中指定的指针进行变址修饰, 则变址修饰解决后指定了FOR~NEXT内部存在的指针时。
3341H	执行ECALL (P) 指令后, 执行RET指令前执行了END指令、FEND指令、GOEND指令、STOP指令时。
3342H	执行ECALL (P) 指令前执行了RET指令时。

程序文件之间子程序输出OFF调用

EFCALL (P)

进行指定程序文件名的指定的指针的子程序的非执行处理。

梯形图	ST
<p>FILE: 文件名</p>	<p>不对应。</p>

FBD/LD
<p>FILE: 文件名</p>

■执行条件

指令	执行条件
EFCALL	
EFCALLP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(文件名)	进行非执行处理的程序文件名	—	Unicode字符串	ANYSTRING_DOUBLE
(P)	子程序的起始指针编号	—	软元件名	POINTER
(s1)~(s5)*1	子程序中作为自变量传送的软元件编号	-2147483648~2147483647	位/带符号BIN16位/带符号BIN32位	ANY
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 在FBD/LD中不能指定。

■可用软元件

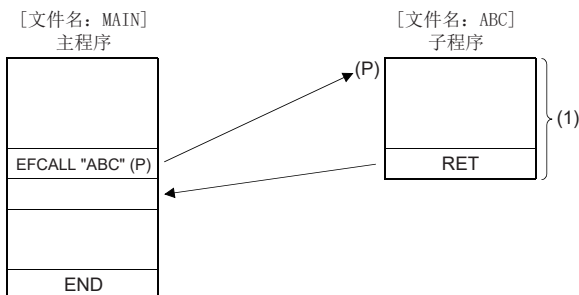
操作数	位		字		双字			间接指定	常数				其它(P)
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K	H	E	\$	
(文件名)	—	—	○	—	—	—	—	○	—	—	○	—	—
(P)	—	—	—	—	—	—	—	—	—	—	—	—	○
(s1)~(s5)	○*1	○	○	○	○	○	○	○	○	○	—	—	—

*1 F以外可以使用。

功能

- 如果执行EFCALL (P) 指令，将进行 (P) 中指定的指针的子程序的非执行处理*2。在EFCALL (P) 指令中，也可调用使用了其它程序文件的局部指针的子程序。

*2 非执行处理时，进行与将各线圈指令的条件触点置为OFF状态时相同的处理。



(1) EFCALL (P) 指令的指令由ON→OFF时进行非执行处理。

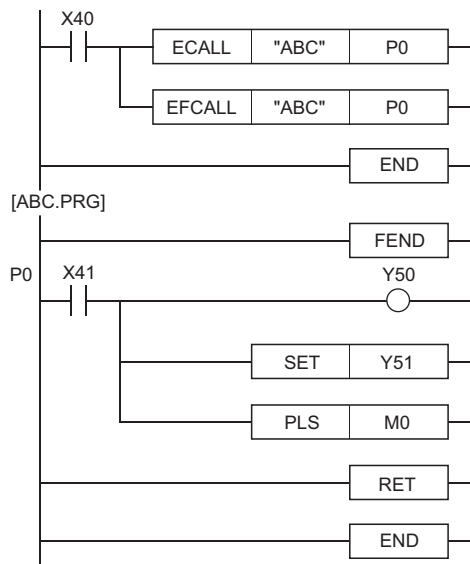
- 非执行处理后的各线圈指令的运算结果与条件触点的ON/OFF无关，其情况如下所示。

运算中使用的软元件	运算结果(软元件的状态)
高速定时器、低速定时器	变为0。
高速累计定时器、低速累计定时器、计数器	保持当前的状态。
OUT指令中的软元件	变为强制OFF。
SET指令、RST指令中的软元件、SFT (P) 指令中的软元件、基本/应用指令中的软元件	保持当前的状态。
PLS指令、脉冲化指令(□P)	变为与条件触点OFF相同的处理。

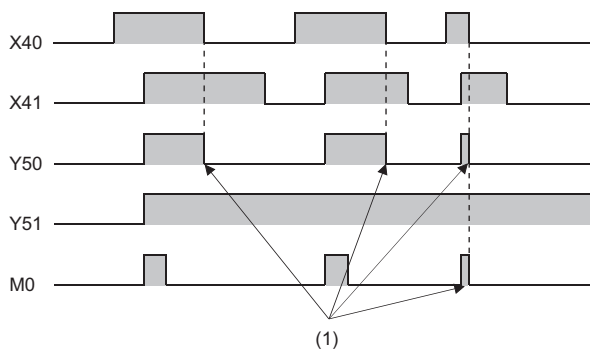
- EFCALL (P) 指令与ECALL (P) 指令组合使用。未与ECALL (P) 指令组合执行的情况下，即使将执行指令置为OFF也不进行子程序的非执行处理，因此各线圈指令的输出状态将被保持。
- 执行指令变为OFF时进行子程序的非执行处理，因此可以将OUT指令、PLS指令(包括脉冲化指令)强制置为OFF。

例

使用EFCALL指令时

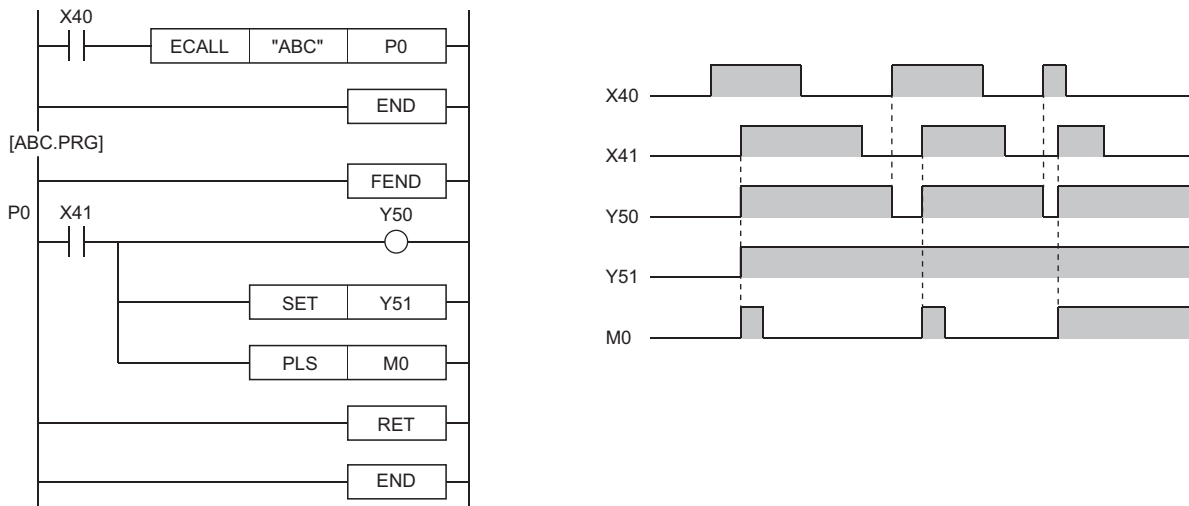


(1) 根据EFCALL指令强制OFF

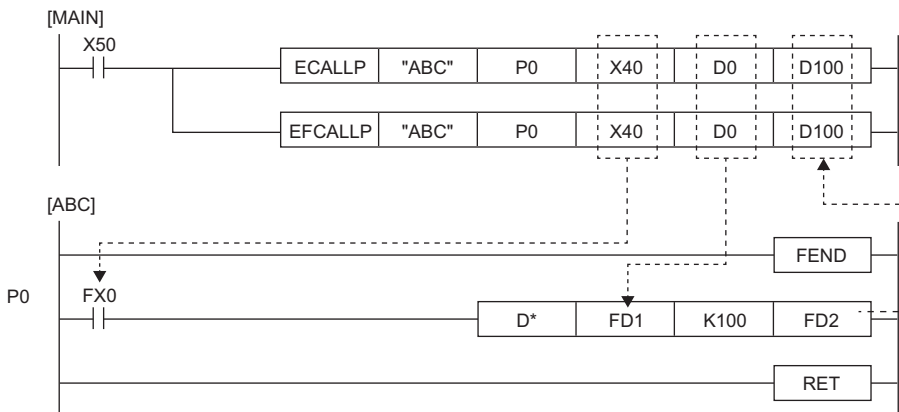


例

不使用EFCALL指令时



- 文件名中只能指定程序存储器中存储的程序文件。
- 文件名中无需指定扩展名(. PRG)。(仅. PRG文件作为对象。)
- 通过子程序使用函数软元件 (FX、FY、FD) 时，在 (s1) ~ (s5) 中指定函数软元件对应的软元件。(s1) ~ (s5) 中指定的软元件的内容如下所示。



- 执行子程序前，位数据的内容将被传送到FX中，字数据的内容将被传送到FD中。
- 执行子程序后，将被传送到FY、FD的内容对应的软元件中。
- FX、FY、FD的数据传送大小根据 (s1) ~ (s5) 中指定的数据类型而有所不同。

■指定了软元件的情况下

FX、FY的函数软元件的传送单位为位单位。FD的函数软元件的传送单位为4字单位。根据参数中指定的软元件的类型，处理的数据大小有所不同。

函数软元件中指定的软元件，应预留出数据大小相应的区域。无法预留出数据大小相应的区域的情况下将变为错误状态。

各函数软元件的数据大小如下所示。

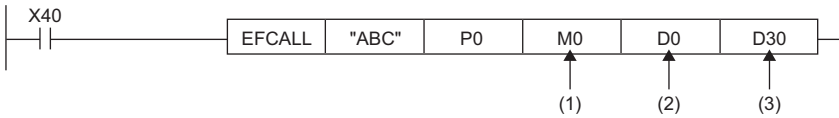
函数软元件	使用软元件	从 (s1) ~ (s5) 传送至函数软元件时的数据大小	从函数软元件传送至 (s1) ~ (s5) 时的数据大小
FX、FY	位软元件	1位	1位
	字软元件的位指定时	1位	1位
FD	位软元件的位指定时*1	4字	位指定的容量或4字*2
	字软元件	4字	4字

*1 位软元件的位指定时，即使 (s1) ~ (s5) 中指定的软元件编号不是16的倍数，也可以指定。

*2 软元件编号为16的倍数，且位指定为K4或K8的情况下传送4字大小。

例

指定了软元件的情况下，将变为如下动作。



- (1) 将M0传送至FX0。
- (2) 将D0~D3传送至FD1。
- (3) 将D30~D33传送至FD2。

■指定了标签的情况下

从(s1)~(s5)传送至函数软元件时，FX、FY的函数软元件的传送单位为位单位。FD的函数软元件的传送单位为4字单位。

从函数软元件传送至(s1)~(s5)时，FX、FY的函数软元件的传送单位为位单位。FD的函数软元件的传送单位为标签的分配大小单位。

各函数软元件的数据大小如下所示。

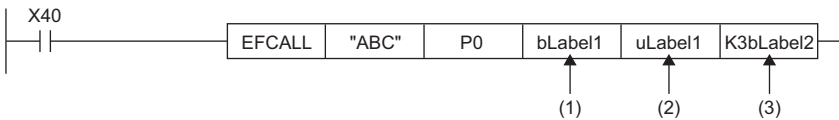
函数软元件	标签的分配大小	从(s1)~(s5)传送至函数软元件时的数据大小	从函数软元件传送至(s1)~(s5)时的数据大小
FX、FY	1位	1位	1位
FD	2位及以上	4字*1	标签的分配大小*2

- *1 指定的标签/结构体的分配大小不足4字的情况下，不足4字的区域将传送0。
- *2 指定的标签/结构体的分配大小为4字及以上的情况下，将传送4字。

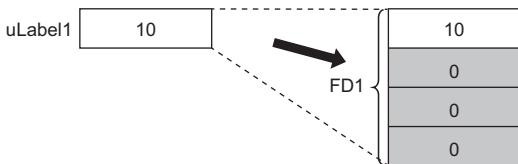
例

从(s1)~(s5)传送至函数软元件时，将执行以下动作。

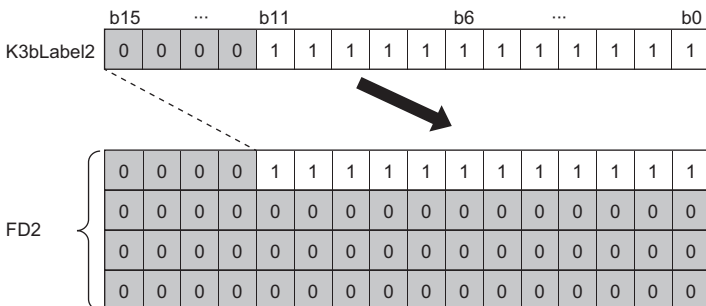
标签名	标签的种类
bLabel1	位型标签
uLabel1	字型标签
bLabel2	数组位型标签



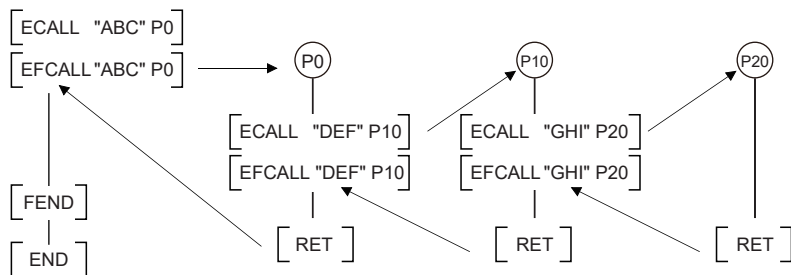
- (1) 将bLabel1(1位)传送至FX0。
- (2) 将uLabel1(1字)传送至FD1。分配大小不足4字，因此不足4字的区域将传送0。



- (3) 将K3bLabel2的1字传送至FD2。分配大小不足4字，因此不足4字的区域将传送0。



- 在EFCALL(P)指令中, 可以使用(s1)~(s5)。
- 子程序中使用的函数软元件数及EFCALL(P)指令的自变量, 需要设置为相同的数。此外, 应将函数软元件与EFCALL(P)指令的自变量的类型设置为相同。
- EFCALL(P)指令的嵌套可达16层。但是, 嵌套的16层指的是CALL(P)指令、FCALL(P)指令、ECALL(P)指令、EFCALL(P)指令、XCALL指令的合计值。



注意事项

- FBD/LD程序不能创建为子程序。
- 在FBD/LD中, 不能执行具有自变量的子程序。
- 如果(P)中指定的指针进行了变址修饰, 则变址修饰解决后指定了FOR~NEXT内部存在的指针时, 将变为错误(错误代码: 3340H)。此外, 不进行变址修饰, 直接指定了FOR~NEXT内部存在的指针时, 将变为转换错误。

出错

错误代码(SD0)	内容
2820H	(s1)~(s5)中自变量中指定的软元件无法预留数据大小区域时。
2840H	(文件名)中指定的文件不存在时。
2841H	(文件名)中指定的文件无法执行时。
3330H	执行了第17重的嵌套时。
3340H	(P)中指定的指针的子程序不存在时。 或者对(P)中指定的指针进行变址修饰, 则变址修饰解决后指定了FOR~NEXT内部存在的指针时。
3341H	执行EFCALL(P)指令后, 执行RET指令前执行了END指令、FEND指令、GOEND指令、STOP指令时。
3342H	执行EFCALL(P)指令前执行了RET指令时。

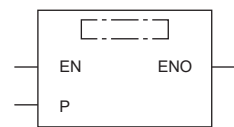
子程序调用

XCALL

进行子程序的执行及非执行处理。条件成立时进行子程序CALL，条件由ON→OFF时进行子程序的FCALL。

梯形图	ST
	不对应。

FBD/LD



■执行条件

指令	执行条件
XCALL	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(P)	子程序的起始指针编号	—	软元件名	POINTER
(s1) ... (s5)*1	子程序中作为自变量传送的软元件编号	-2147483648~2147483647	位/带符号BIN16位/带符号BIN32位	ANY
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 在FBD/LD中不能指定。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它 (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	○	○	○	○	○	○	○	○	○	—	—	—

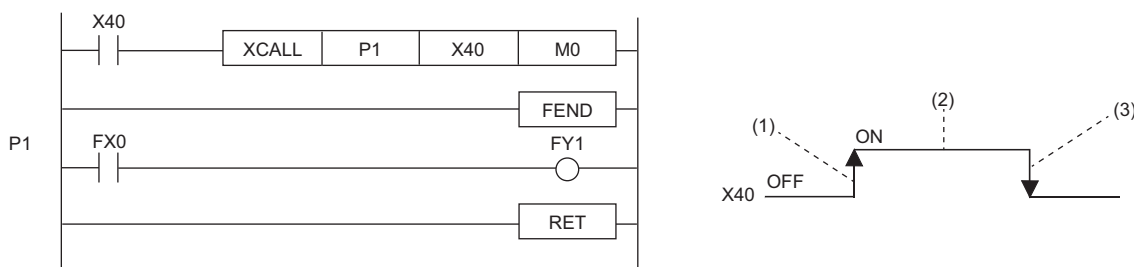
*1 F以外可以使用。

功能

- XCALL指令是进行子程序的执行及非执行处理的指令。
- 子程序的执行时，根据各线圈指令的条件触点的ON/OFF状态进行运算。
- 子程序的非执行处理时，进行与将各线圈指令的条件触点置为OFF状态时相同的处理。
- 非执行处理后的各线圈指令的运算结果与条件触点的ON/OFF无关，其情况如下所示。

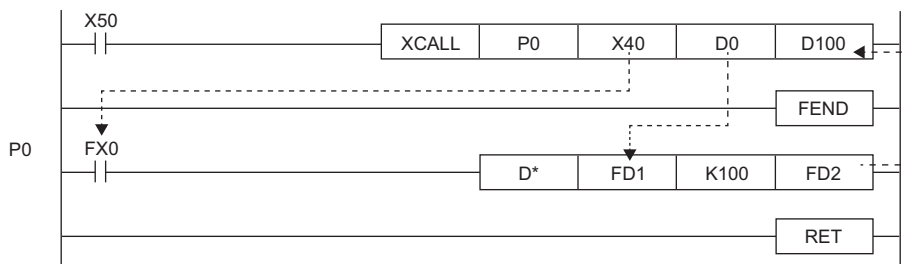
运算中使用的软元件	运算结果(软元件的状态)
高速定时器、低速定时器	变为0。
高速累计定时器、低速累计定时器、计数器	保持当前的状态。
OUT指令中的软元件	变为强制OFF。
SET指令、RST指令中的软元件、SFT(P)指令中的软元件、基本/应用指令中的软元件	保持当前的状态。
PLS指令、脉冲化指令(□P)	变为与条件触点OFF相同的处理。

- XCALL指令的动作如下所示。



- (1) X40的上升沿(OFF→ON): 执行P1的子程序。
- (2) X40的ON中: 执行P1的子程序。(X40的ON中, 不包含X40的上升沿。)
- (3) X40的下降沿(ON→OFF): 进行P1的子程序的非执行处理。

- 通过子程序使用函数软元件(FX、FY、FD)时，在(s1)~(s5)中指定函数软元件对应的软元件。(s1)~(s5)中指定的软元件的内容如下所示。



- 执行子程序前，位数据的内容将被传送到FX中，字数据的内容将被传送到FD中。
- 执行子程序后，将被传送到FY、FD的内容对应的软元件中。
- FX、FY、FD的数据传送大小根据(s1)~(s5)中指定的数据类型而有所不同。

■指定了软元件的情况下

FX、FY的函数软元件的传送单位为位单位。FD的函数软元件的传送单位为4字单位。根据参数中指定的软元件的类型，处理的数据大小有所不同。

函数软元件中指定的软元件，应预留出数据大小相应的区域。无法预留出数据大小相应的区域的情况下将变为错误状态。各函数软元件的数据大小如下所示。

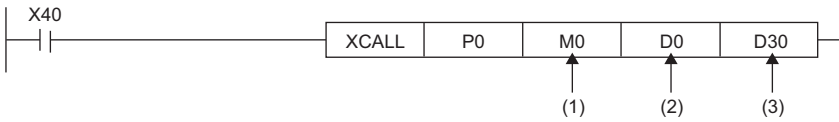
函数软元件	使用软元件	从(s1)~(s5)传送到函数软元件时的数据大小	从函数软元件传送到(s1)~(s5)时的数据大小
FX、FY	位软元件	1位	1位
	字软元件的位指定时	1位	1位
FD	位软元件的位指定时*1	4字	位指定的容量或4字*2
	字软元件	4字	4字

*1 位软元件的位指定时，即使(s1)~(s5)中指定的软元件编号不是16的倍数，也可以指定。

*2 软元件编号为16的倍数，且位指定为K4或K8的情况下传送4字大小。

例

指定了软元件的情况下，将变为如下动作。



- (1) 将M0传送至FX0。
- (2) 将D0~D3传送至FD1。
- (3) 将D30~D33传送至FD2。

■指定了标签的情况下

从(s1)~(s5)传送至函数软元件时，FX、FY的函数软元件的传送单位为位单位。FD的函数软元件的传送单位为4字单位。

从函数软元件传送至(s1)~(s5)时，FX、FY的函数软元件的传送单位为位单位。FD的函数软元件的传送单位为标签的分配大小单位。

各函数软元件的数据大小如下所示。

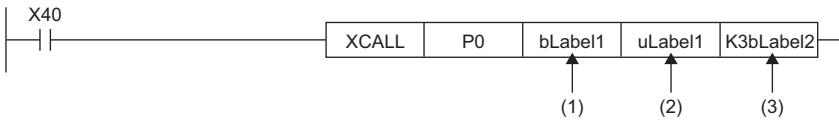
函数软元件	标签的分配大小	从(s1)~(s5)传送至函数软元件时的数据大小	从函数软元件传送至(s1)~(s5)时的数据大小
FX、FY	1位	1位	1位
FD	2位及以上	4字*1	标签的分配大小*2

- *1 指定的标签/结构体的分配大小不足4字的情况下，不足4字的区域将传送0。
- *2 指定的标签/结构体的分配大小为4字及以上的情况下，将传送4字。

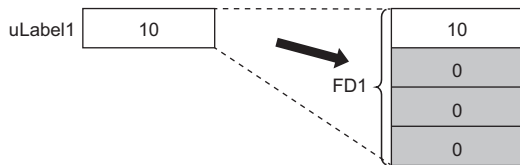
例

从(s1)~(s5)传送至函数软元件时，将执行以下动作。

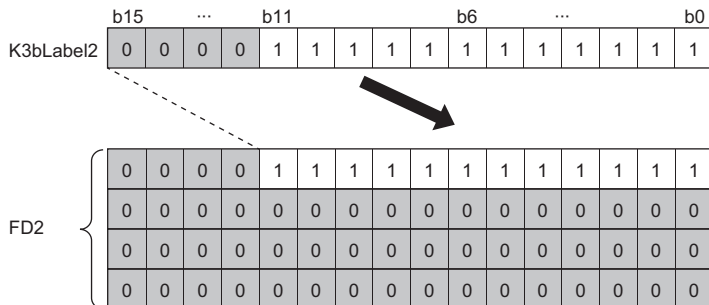
标签名	标签的种类
bLabel1	位型标签
uLabel1	字型标签
bLabel2	数组位型标签



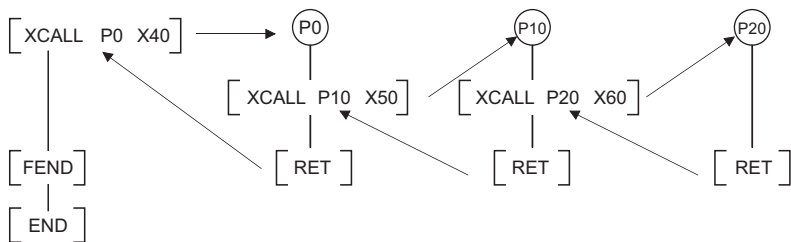
- (1) 将bLabel1(1位)传送至FX0。
- (2) 将uLabel1(1字)传送至FD1。分配大小不足4字，因此不足4字的区域将传送0。



- (3) 将K3bLabel2的1字传送至FD2。分配大小不足4字，因此不足4字的区域将传送0。



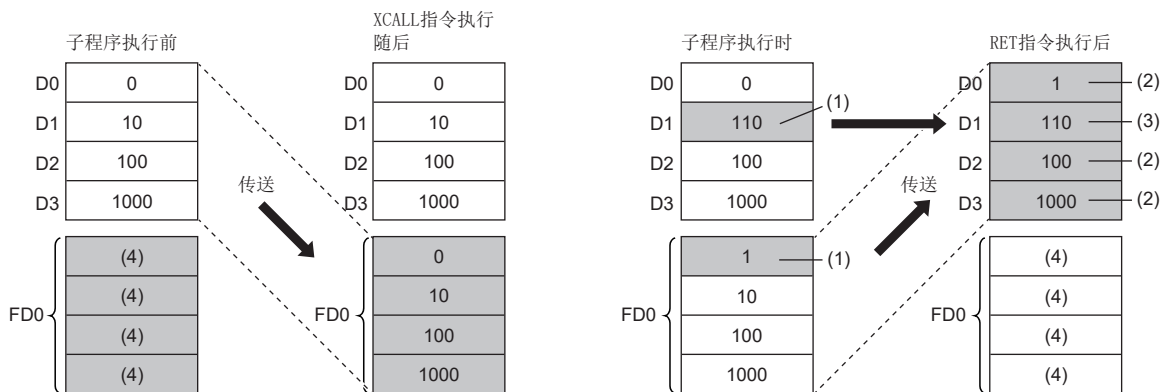
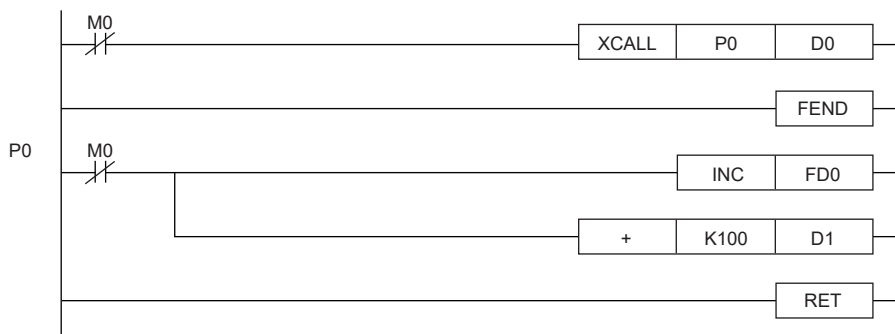
- 在XCALL指令中，可以使用(s1)~(s5)。
- 子程序中使用的函数软元件数与XCALL指令的自变量，需要设置为相同的数。此外，应将函数软元件与XCALL指令的自变量的类型设置为相同。
- XCALL指令中自变量中指定的软元件编号应避免重复。重复的情况下，将无法正常运转。
- XCALL指令的嵌套可达16层。但是，嵌套的16层指的是CALL(P)指令、FCALL(P)指令、ECALL(P)指令、EFCALL(P)指令、XCALL指令的合计值。



- 请勿将XCALL指令的自变量中使用的软元件用于子程序。将XCALL指令的自变量中使用的软元件用于子程序的情况下，将无法正常运转。

例

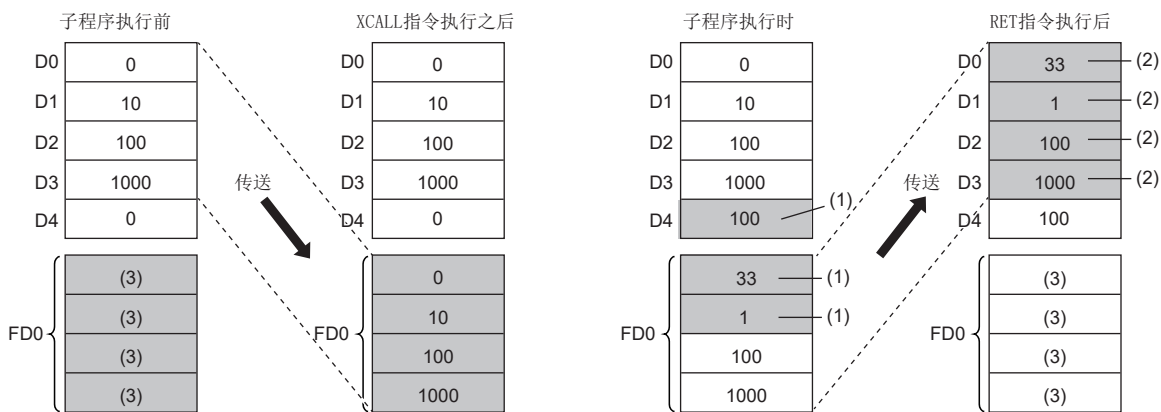
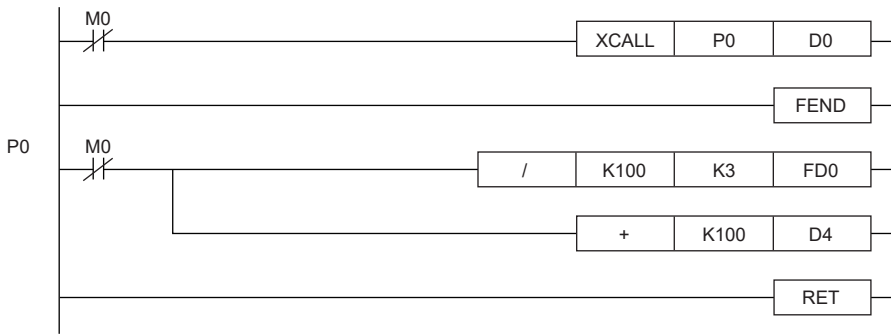
[错误的运算]子程序的FD0中指定D0，子程序中使用D1情况下的动作



- (1) 存储子程序的执行结果。
- (2) 替换为函数软元件的值。
- (3) D1不反映函数软元件的值。
- (4) 变为不定值。

例

[正确运算]子程序的FD0中指定D0，通过子程序使用了D4时的动作



- (1) 存储子程序的执行结果。
- (2) 替换为函数软元件的值。
- (3) 变为不定值。

注意事项

- FBD/LD程序不能创建为子程序。
- 在FBD/LD中，不能执行具有自变量的子程序。
- 如果(P)中指定的指针进行了变址修饰，则变址修饰解决后指定了FOR~NEXT内部存在的指针时，将变为错误(错误代码：3340H)。此外，不进行变址修饰，直接指定了FOR~NEXT内部存在的指针时，将变为转换错误。

出错

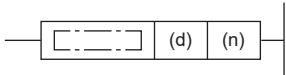
错误代码(SD0)	内容
2820H	(s1)~(s5)中自变量中指定的软元件无法预留数据大小区域时。
3330H	执行了第17重的嵌套时。
3340H	(P)中指定的指针的子程序不存在时。 或者对(P)中指定的指针进行变址修饰，则变址修饰解决后指定了FOR~NEXT内部存在的指针时。
3341H	执行XCALL指令后，执行RET指令前执行了END指令、FEND指令、GOEND指令、STOP指令时。
3342H	执行XCALL指令前执行了RET指令时。

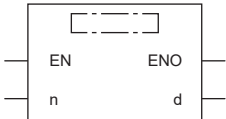
14.1 旋转指令

BIN16位数据的右旋转

ROR(P)、RCR(P)

- ROR(P)：将BIN16位数据在不包含进位标志的状况下进行(n)位右旋。
- RCR(P)：将BIN16位数据在包含进位标志的状况下进行(n)位右旋。



梯形图	ST*1
	ENO:=RORP(EN, n, d); ENO:=RCR(EN, n, d); ENO:=RCRP(EN, n, d);

FBD/LD*1


*1 在ST、FBD/LD中不支持ROR指令。应使用通用函数的ROR。

☞ 1298页 ROR(_E)

■执行条件

指令	执行条件
ROR RCR	
RORP RCRP	

设置数据

■内容、范围、数据类型

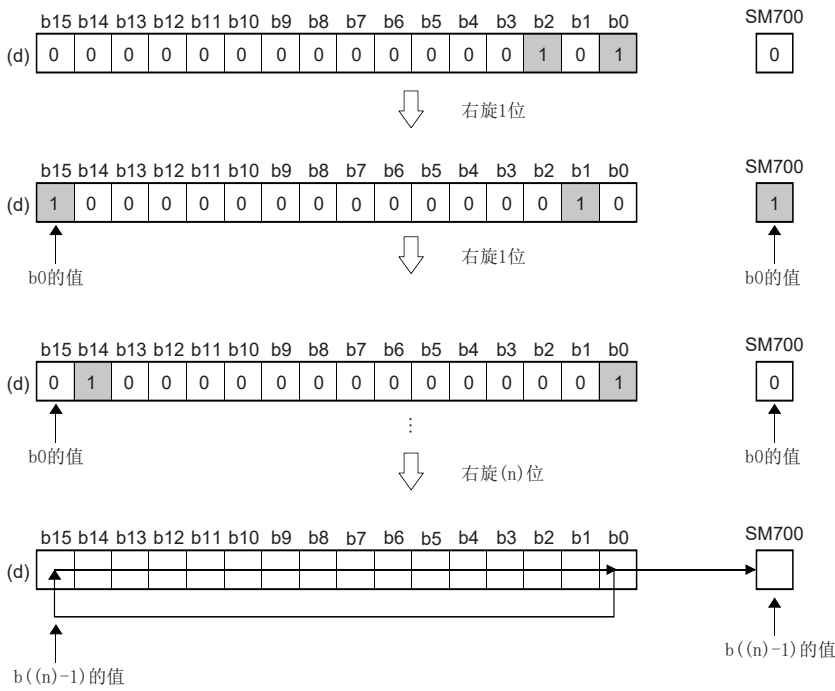
操作数	内容	范围	数据类型	数据类型(标签)
(d)	旋转的软元件	—	带符号BIN16位	ANY16
(n)	旋转的位数	0~15	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)

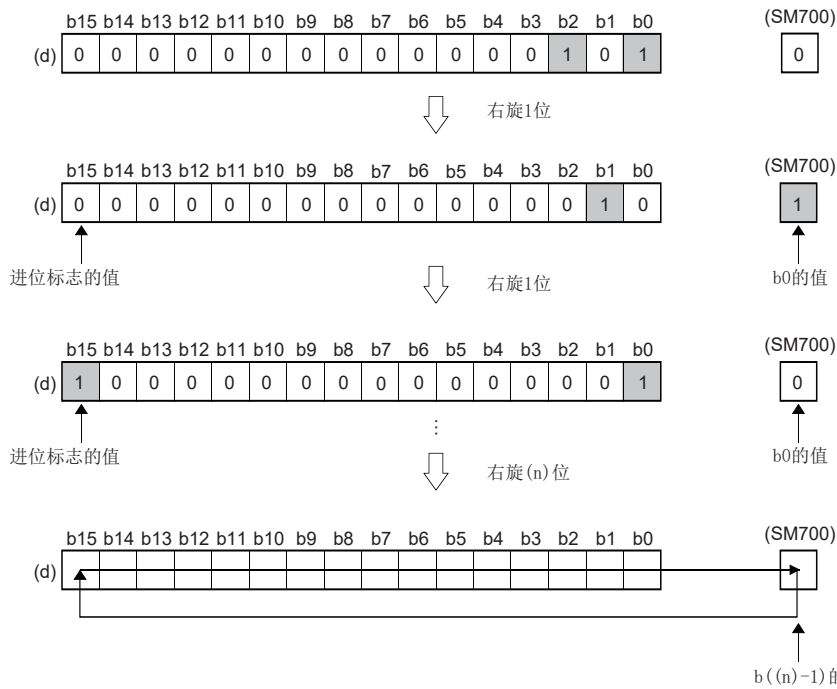
- 将(d)中指定的软元件的BIN16位数据，在不包含SM700(进位标志)的状况下进行(n)位右旋。SM700(进位标志)根据ROR(P)指令执行前的状态而处于ON或OFF状态。



- (d)中指定了位软元件的情况下，以位指定中指定的软元件范围进行旋转。此时实际旋转的位数将变为(n)÷(位指定中指定的点数)的余数。例如，(n)=15，(位指定中指定的点数)=12位时，15÷12=1余3，因此进行3位右旋。
- (n)以0~15进行指定。(n)中指定了16或其以上的值的情况下，以(n)÷16的余数值进行旋转。例如(n)=18时，18÷16=1余2，因此进行2位右旋。

■RCR(P)

- 将(d)中指定的软元件的BIN16位数据，在包含SM700(进位标志)的状况下进行(n)位右旋。SM700(进位标志)根据RCR(P)指令执行前的状态而处于ON或OFF状态。



- (d)中指定了位软元件的情况下，以位指定中指定的软元件范围进行旋转。此时实际旋转的位数将变为 $(n) \div (\text{位指定中指定的点数})$ 的余数。例如， $(n)=15$ ，(位指定中指定的点数)=12位时， $15 \div 12=1$ 余3，因此进行3位右旋。
- (n)以0~15进行指定。(n)中指定了16或其以上的值的情况下，以 $(n) \div 16$ 的余数值进行旋转。例如 $(n)=18$ 时， $18 \div 16=1$ 余2，因此进行2位右旋。

出错

没有运算错误。

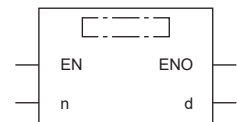
BIN16位数据的左旋转

ROL(P)、RCL(P)

- ROL(P)：将BIN16位数据在不包含进位标志的状况下进行(n)位左旋。
- RCL(P)：将BIN16位数据在包含进位标志的状况下进行(n)位左旋。

梯形图	ST*1
	ENO:=ROLP(EN, n, d); ENO:=RCL(EN, n, d); ENO:=RCLP(EN, n, d);

FBD/LD*1



*1 在ST、FBD/LD中不支持ROL指令。应使用通用函数的ROL。
 1296页 ROL(E)

■执行条件

指令	执行条件
ROL RCL	
ROLP RCLP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	旋转的软元件	—	带符号BIN16位	ANY16
(n)	旋转的位数	0~15	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

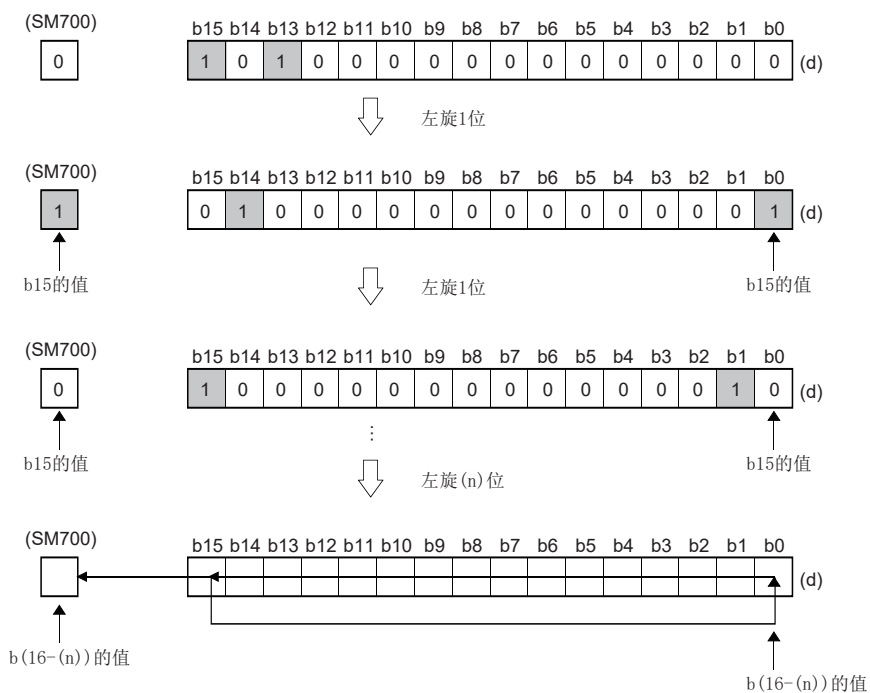
■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

■ROL(P)

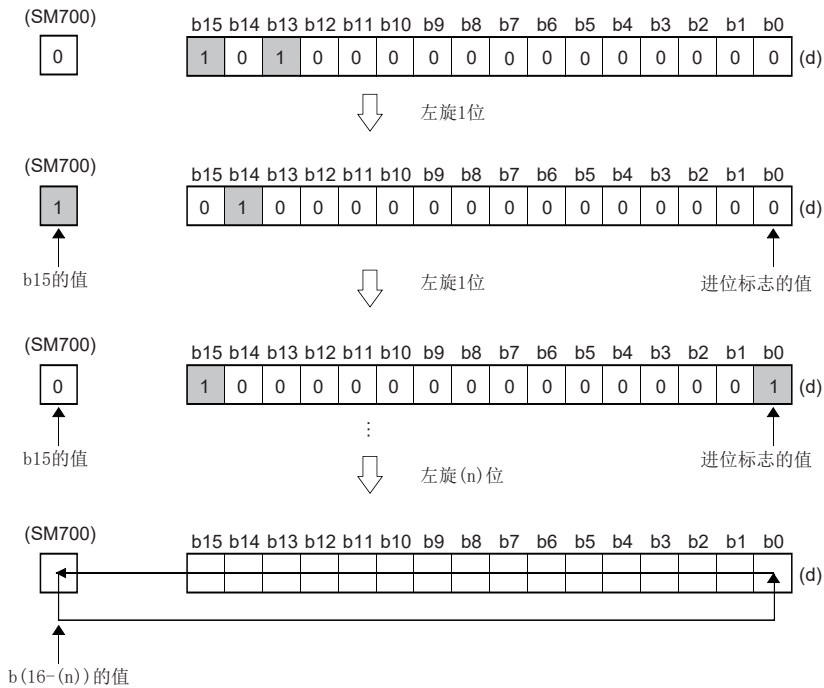
- 将(d)中指定的软元件的BIN16位数据，在不包含SM700(进位标志)的状况下进行(n)位左旋。SM700(进位标志)根据ROL(P)指令执行前的状态而处于ON或OFF状态。



- (d)中指定了位软元件的情况下，以位指定中指定的软元件范围进行旋转。此时实际旋转的位数将变为 $(n) \div (\text{位指定中指定的点数})$ 的余数。例如， $(n)=15$ ，(位指定中指定的点数)=12位时， $15 \div 12=1$ 余3，因此进行3位左旋。
- (n)以0~15进行指定。(n)中指定了16或其以上的值的情况下，以 $(n) \div 16$ 的余数值进行旋转。例如 $(n)=18$ 时， $18 \div 16=1$ 余2，因此进行2位左旋。

■RCL(P)

- 将(d)中指定的软元件的BIN16位数据，在包含SM700(进位标志)的状况下进行(n)位左旋。SM700(进位标志)根据RCL(P)指令执行前的状态而处于ON或OFF状态。



- (d)中指定了位软元件的情况下，以位指定中指定的软元件范围进行旋转。此时实际旋转的位数将变为 $(n) \div (\text{位指定中指定的点数})$ 的余数。例如， $(n)=15$ ，(位指定中指定的点数)=12位时， $15 \div 12=1$ 余3，因此进行3位左旋。
- (n)以0~15进行指定。(n)中指定了16或其以上的值的情况下，以 $(n) \div 16$ 的余数值进行旋转。例如 $(n)=18$ 时， $18 \div 16=1$ 余2，因此进行2位左旋。

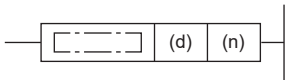
出错

没有运算错误。

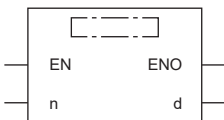
BIN32位数据的右旋转

DROR(P)、DRCR(P)

- DROR(P)：将BIN32位数据在不包含进位标志的状况下进行(n)位右旋。
- DRCR(P)：将BIN32位数据在包含进位标志的状况下进行(n)位右旋。

梯形图	ST*1
	ENO:=DRORP(EN, n, d); ENO:=DRCR(EN, n, d); ENO:=DRCRP(EN, n, d);


FBD/LD*1



*1 在ST、FBD/LD中不支持DROR指令。应使用通用函数的ROR。

☞ 1298页 ROR(_E)

■执行条件

指令	执行条件
DROR DRCR	
DRORP DRCRP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	旋转的起始软元件	—	带符号BIN32位	ANY32
(n)	旋转的位数	0~31	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

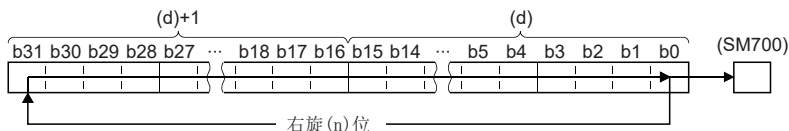
■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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	
(d)	○	○	○	○	○	○	○	—	—	—	—
(n)	○	○	○	○	○	—	—	○	—	—	—

功能

■DROR(P)

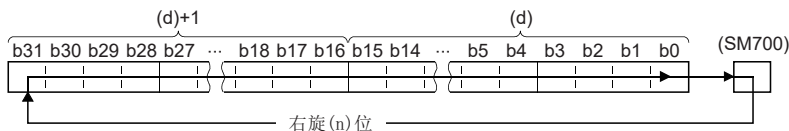
- 将(d)中指定的软元件的BIN32位数据,在不包含SM700(进位标志)的状况下进行(n)位右旋。SM700(进位标志)根据DROR(P)指令执行前的状态而处于ON或OFF状态。



- (d)中指定了位软元件的情况下,以位指定中指定的软元件范围进行旋转。此时实际旋转的位数将变为 $(n) \div (\text{位指定中指定的点数})$ 的余数。例如, $(n)=31$, (位指定中指定的点数)=24位时, $31 \div 24=1$ 余7,因此进行7位右旋。
- (n)以0~31进行指定。(n)中指定了32或其以上的值的情况下,以 $(n) \div 32$ 的余数值进行旋转。例如 $(n)=34$ 时, $34 \div 32=1$ 余2,因此进行2位右旋。

■DRCR(P)

- 将(d)中指定的软元件的BIN32位数据,在包含SM700(进位标志)的状况下进行(n)位右旋。SM700(进位标志)根据DRCR(P)指令执行前的状态而处于ON或OFF状态。



- (d)中指定了位软元件的情况下,以位指定中指定的软元件范围进行旋转。此时实际旋转的位数将变为 $(n) \div (\text{位指定中指定的点数})$ 的余数。例如, $(n)=31$, (位指定中指定的点数)=24位时, $31 \div 24=1$ 余7,因此进行7位右旋。
- (n)以0~31进行指定。(n)中指定了32或其以上的值的情况下,以 $(n) \div 32$ 的余数值进行旋转。例如 $(n)=34$ 时, $34 \div 32=1$ 余2,因此进行2位右旋。

出错

没有运算错误。

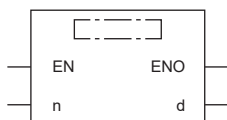
BIN32位数据的左旋转

DROL (P)、DRCL (P)

- DROL (P)：将BIN32位数据在不包含进位标志的状况下进行(n)位左旋。
- DRCL (P)：将BIN32位数据在包含进位标志的状况下进行(n)位左旋。

梯形图	ST*1
	ENO:=DROLP (EN, n, d) ; ENO:=DRCL (EN, n, d) ; ENO:=DRCLP (EN, n, d) ;

FBD/LD*1



*1 在ST、FBD/LD中不支持DROL指令。应使用通用函数的ROL。

☞ 1296页 ROL (E)

■执行条件

指令	执行条件
DROL DRCL	
DROLP DRCLP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	旋转的起始软元件	—	带符号BIN32位	ANY32
(n)	旋转的位数	0~31	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

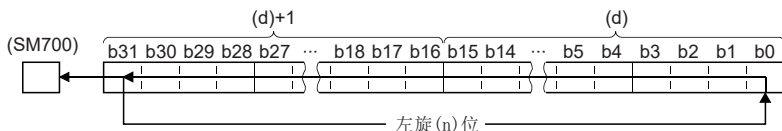
■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

■DROL(P)

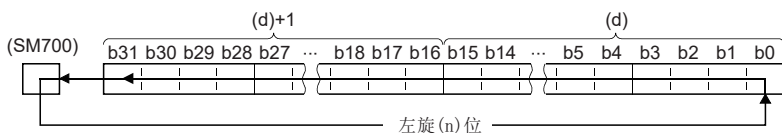
- 将(d)中指定的软元件的BIN32位数据，在不包含SM700(进位标志)的状况下进行(n)位左旋。SM700(进位标志)根据DROL(P)指令执行前的状态而处于ON或OFF状态。



- (d)中指定了位软元件的情况下，以位指定中指定的软元件范围进行旋转。此时实际旋转的位数将变为 $(n) \div (\text{位指定中指定的点数})$ 的余数。例如， $(n)=31$ ，(位指定中指定的点数)=24位时， $31 \div 24=1$ 余7，因此进行7位左旋。
- (n)以0~31进行指定。(n)中指定了32或其以上的值的情况下，以 $(n) \div 32$ 的余数值进行旋转。例如 $(n)=34$ 时， $34 \div 32=1$ 余2，因此进行2位左旋。

■DRCL(P)

- 将(d)中指定的软元件的BIN32位数据，在包含SM700(进位标志)的状况下进行(n)位左旋。SM700(进位标志)根据DRCL(P)指令执行前的状态而处于ON或OFF状态。



- (d)中指定了位软元件的情况下，以位指定中指定的软元件范围进行旋转。此时实际旋转的位数将变为 $(n) \div (\text{位指定中指定的点数})$ 的余数。例如， $(n)=31$ ，(位指定中指定的点数)=24位时， $31 \div 24=1$ 余7，因此进行7位左旋。
- (n)以0~31进行指定。(n)中指定了32或其以上的值的情况下，以 $(n) \div 32$ 的余数值进行旋转。例如 $(n)=34$ 时， $34 \div 32=1$ 余2，因此进行2位左旋。

出错

没有运算错误。

14.2 数据表操作指令

从数据表的先输入数据的读取

FIFR(P)

将表中最先存储的数据存储到指定的软元件中。

梯形图	ST
	ENO:=FIFR(EN, s, d); ENO:=FIFRP(EN, s, d);

FBD/LD

■执行条件

指令	执行条件
FIFR	
FIFRP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储从表读取的数据的软元件	—	带符号BIN16位	ANY16
(d)	表的起始软元件	—	字	ANY16*1
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

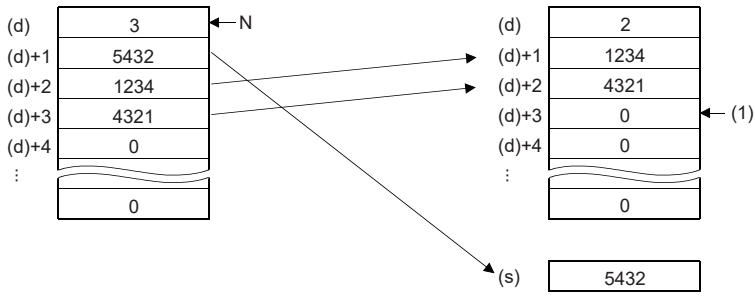
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 将(d)中指定的表的最先输入数据(d)+1存储到(s)中指定的软元件中。执行FIFR(P)指令后,数据表的数据将逐个向前依次填入。



N: 数据数

(1) 存储0。

- 应采用(d)中存储的值为0时,不执行FIFR(P)指令的互锁。
- 数据数(d)可存储的范围为0~65535。

出错

错误代码(SD0)	内容
3285H	(d)的值为0的状况下执行了FIFR(P)指令时。

从数据表的后输入数据的读取

FPOP (P)

将表的最后处存储的数据存储到指定的软元件中。

梯形图	ST
	ENO:=FPOP (EN, s, d) ; ENO:=FPOPP (EN, s, d) ;

FBD/LD

■执行条件

指令	执行条件
FPOP	
FPOPP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储从表读取的数据的软元件	—	带符号BIN16位	ANY16
(d)	表的起始软元件	—	字	ANY16* ¹
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

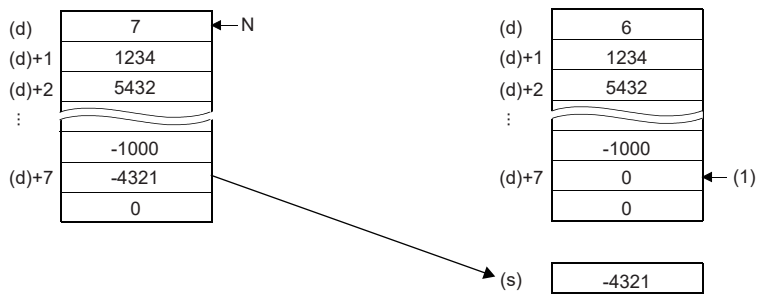
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 将(d)中指定的表的最后处存储的数据存储到(s)中指定的软元件中。执行FPOP(P)指令后，存储了通过FPOP(P)指令读取的数据的软元件将变为0。



N: 数据数

(1) 存储0。

- 应采用(d)中存储的值为0时，不执行FPOP(P)指令的互锁。
- 数据数(d)可存储的范围为0~65535。

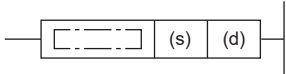
出错

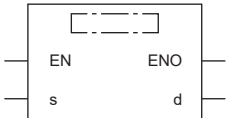
错误代码(SD0)	内容
3285H	(d)的值为0的状况下执行了FPOP(P)指令时。

至数据表的数据写入


FIFW(P)

将BIN16位数据存储到指定的数据表中。

梯形图	ST
	ENO:=FIFW(EN, s, d); ENO:=FIFWP(EN, s, d);

FBD/LD


■执行条件

指令	执行条件
FIFW	
FIFWP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	写入至表中的数据或存储了数据的软元件编号	—	带符号BIN16位	ANY16
(d)	表的起始软元件	—	字	ANY16* ¹
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

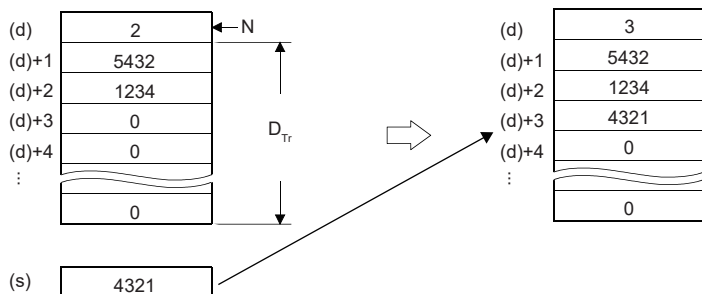
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 将(s)中指定的BIN16位数据存储到(d)中指定的数据表中。(d)中存储了表中存储的数据数后，(d)+1及其以后将依次存储(s)中指定的数据。



N: 数据数

D_{tr} : 数据表范围(由用户管理)

- 首次执行FIFW(P)指令的情况下，应将(d)中指定的软元件的值清除。
- (d)+1及其以后将依次存储数据，因此应注意数据表的范围。
- 数据数(d)可存储的范围为0~65535。

出错

错误代码(SD0)	内容
3285H	(d)的值为FFFFH的状况下执行了FIFW(P)指令时。

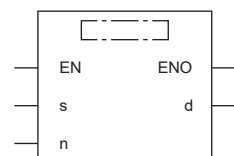
数据表的数据插入

FINS (P)

将BIN16位数据插入到指定的数据表的第(n)号。

梯形图	ST
	ENO:=FINS (EN, s, n, d) ; ENO:=FINS (EN, s, n, d) ;

FBD/LD



■执行条件

指令	执行条件
FINS	
FINS P	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储插入数据的软元件	—	带符号BIN16位	ANY16
(d)	表的起始软元件	—	字	ANY16*1
(n)	插入的表位置	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

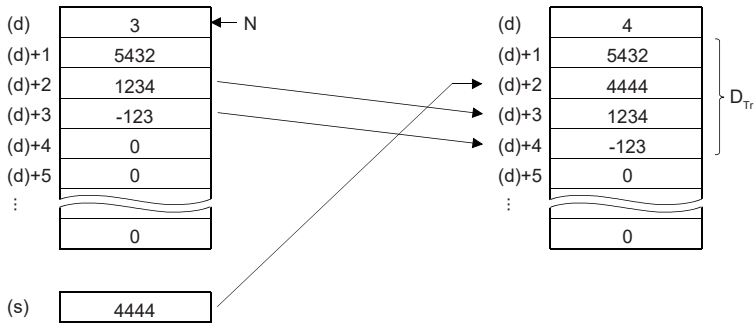
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 将(s)中指定的BIN16位数据插入到(d)中指定的数据表的第(n)号中。执行FINS(P)指令后，从数据表的第(n)号开始的数据将逐个往下顺延。



N: 数据数

D_{Tr} : 数据表范围(由用户管理)

(n)=2的情况下，将被插入到(d)+2中。

- 数据数(d)可存储的范围为0~65535。

出错

错误代码(SD0)	内容
3285H	(n)的值为0的状况下执行了FINS(P)指令时。
	(d)的值为FFFFH的状况下执行了FINS(P)指令时。
	(n)的值中指定超过(d)的数据数+1的值并执行了FINS(P)指令时。

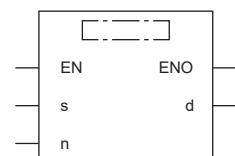
数据表的数据删除

FDEL (P)

删除数据表第(n)号的数据。

梯形图	ST
	ENO:=FDEL (EN, s, n, d) ; ENO:=FDELP (EN, s, n, d) ;

FBD/LD



■执行条件

指令	执行条件
FDEL	
FDELP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储删除数据的软元件	—	带符号BIN16位	ANY16
(d)	表的起始软元件	—	字	ANY16*1
(n)	删除的表位置	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

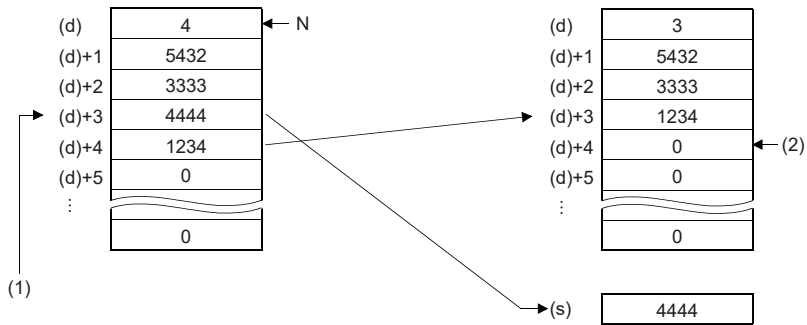
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 将(d)中指定的数据表的第(n)号的数据删除后, 存储到(s)中指定的软元件中。执行FDEL(P)指令后, 数据表的第(n)+1号及其以后的数据将逐个向前依次填入。



N: 数据数

(1) (n)=3的情况下, (d)+3将成为对象。

(2) 存储0。

- 数据数(d)可存储的范围为0~65535。

出错

错误代码(SD0)	内容
3285H	(n)的值为0的状况下执行了FDEL(P)指令时。
	(d)的值为0的状况下执行了FDEL(P)指令时。
	(n)的值中指定超过(d)的数据数+1的值并执行了FDEL(P)指令时。

14.3 数据读取/写入指令

数据存储器的数据读取/写入

至数据存储器的数据写入指令是将任意软元件数据写入到数据存储器中的指令。

通过将运算中使用的固定值及运算结果写入到数据存储器中，可以防止电池过低时数据丢失。

此外，可以使用通过数据存储器执行的数据读取指令，在任意时机对写入至数据存储器中的数据进行读取。


■执行方法

至软元件数据的数据存储器的写入是通过SP.DEVST指令进行。

此外，通过S(P).DEVLD指令，将数据存储器中写入的软元件数据读取到指定软元件中。

■设置方法

使用SP.DEVST指令、S(P).DEVLD指令的情况下，需要进行软元件数据存储用文件设置。

 [CPU参数]⇒[文件设置]⇒[软元件数据存储用文件设置]

软元件数据存储用文件设置	
<input type="checkbox"/> 使用有无设置	使用
容量	1 K字
文件名	DEVSTORE

项目	内容
容量	1K~512K(字)
文件名	DEVSTORE(固定)

数据存储器中没有创建软元件数据存储用文件的必要空余容量的情况下，将发生错误(错误代码：21A1H)。

此外，STOP→RUN时将进行实际软元件数据存储用文件与参数设置是否一致的检查，不一致的情况下将发生错误(错误代码：21A0H)。

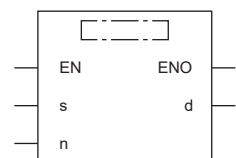
从数据存储器的数据读取

S(P).DEVLD

从数据存储器上的软元件数据存储用文件读取数据。

梯形图 	ST ENO:=S_DEVLD(EN, s, n, d); ENO:=SP_DEVLD(EN, s, n, d);
----------------	--

FBD/LD



■执行条件

指令	执行条件
S.DEVLD	
SP.DEVLD	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	从软元件数据存储用文件的读取偏置(以1点16位的单位指定)	0~524287	无符号BIN32位	ANY32
(d)	存储读取的数据的软元件	—	字	ANY16*1
(n)	读取点数	1~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

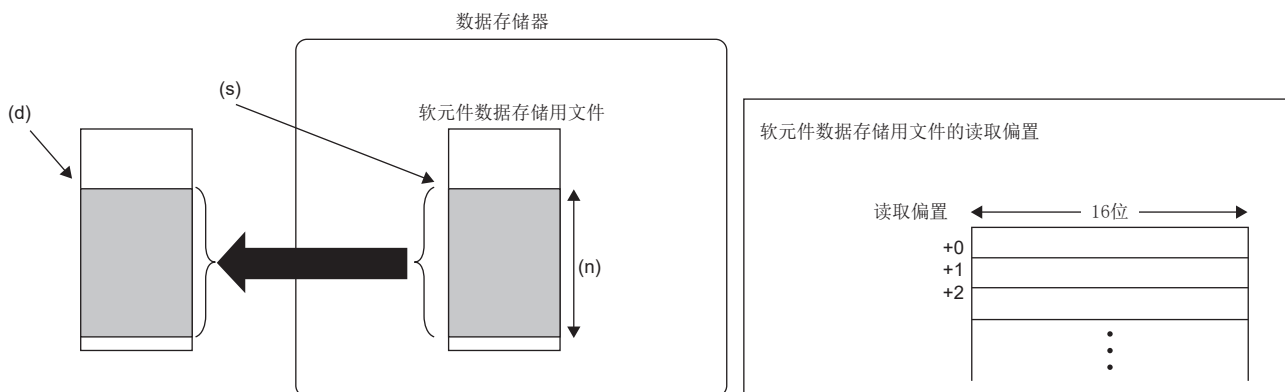
*1 通过标签进行设置的情况下,应在确保动作所需区域的前提下定义数组,指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	○	—	○	—	—	—	—	○	○	—	—	—	—

功能

- 从数据存储器上的软元件数据存储用文件的(s)中指定的读取偏置开始, 读取(n)中指定的点数的软元件数据, 存储到(d)中指定的软元件中。对(s), 在软元件数据存储用文件起始开始的偏置中, 以字偏置(每16位+1的单位)进行指定。



- 使用S(P).DEVLD指令的情况下, 需要进行软元件数据存储用文件设置。(☞ 589页 设置方法)
- 至软元件数据存储用文件的写入是通过SP.DEVST指令进行。(☞ 592页 SP.DEVST)

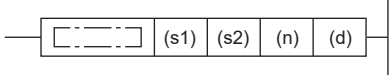
出错

错误代码(SD0)	内容
2840H	参数中未进行软元件数据存储用文件的设置时。
3285H	(n)的设置数据为0时。

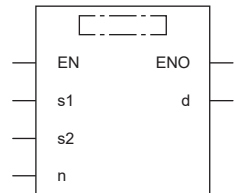
至数据存储器的数据写入

SP. DEVST


将指定点数的软元件数据写入到数据存储器上的软元件数据存储用文件中。

梯形图	ST
	<p>ENO:=SP_DEVST (EN, s1, s2, n, d) ;</p>

FBD/LD



■执行条件

指令	执行条件
SP. DEVST	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	软元件数据存储用文件的写入偏置(以1点16位的单位指定)	0~524287	无符号BIN32位	ANY32
(s2)	写入的起始软元件	—	字	ANY16*1
(n)	写入点数	1~65535	无符号BIN16位	ANY16
(d)	(d): 完成软元件, (d)+1: 异常完成软元件	—	位	ANYBIT_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

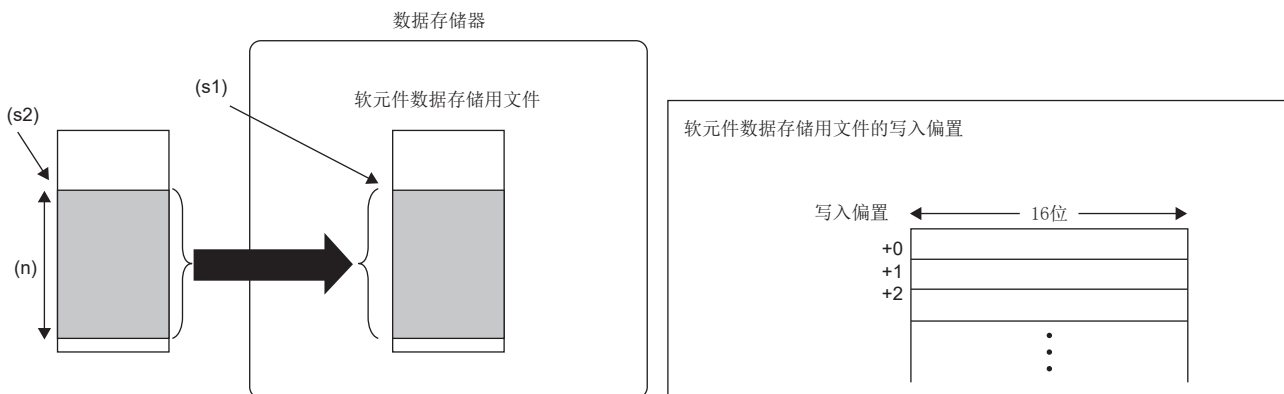
*1 通过标签进行设置的情况下, 应在确保动作所需区域的前提下定义数组, 指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—	—
(n)	○	—	○	—	—	—	—	○	○	—	—	—	—
(d)	○	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 将从(s2)中指定的软元件开始的(n)中指定的点数的软元件数据，写入到数据存储器上的软元件数据存储用文件的(s1)中指定的写入偏置中。(s1)是软元件数据存储用文件起始开始的偏置，以字偏置(每个16位+1的单位)进行指定。



- 完成软元件(d)，在检测出SP.DEVST指令处理完成的END指令执行时，将自动变为ON，由于在下一个扫描的END指令下变为OFF，作为SP.DEVST指令的执行完成标志使用。
- SP.DEVST指令异常完成时，异常完成软元件(d)+1将以与完成软元件(d)相同的时机置为ON/OFF，因此作为SP.DEVST指令的异常完成标志使用。
- SP.DEVST指令的执行中SM753(文件访问中)将变为ON。SM753已处于ON状态的情况下，不能执行SP.DEVST指令。(执行的情况下将变为无处理。)
- 执行SP.DEVST指令时检测到错误的情况下，完成软元件(d)、异常完成软元件(d)+1、SM753不变为ON。
- 使用SP.DEVST指令的情况下，需要进行软元件数据存储用文件设置。(☞ 589页 设置方法)
- 通过S(P).DEVLD指令，将软元件数据存储用文件中写入的软元件数据读取到指定软元件中。(☞ 590页 S(P).DEVLD)

注意事项

- 写入到数据存储器中的值将变为执行SP.DEVST指令时的值。
- 通过执行SP.DEVST指令，SD4124、SD4125(内置存储器改写次数指标)将增加。至控制器的数据存储器的写入次数有上限，数据存储器写入次数指标超过5万次时，将发生错误(错误代码：1080H)。
- 为了防止意外指令执行导致数据存储器写入次数增加，通过设置SD771(至数据存储器的写入指令执行次数指定)，可以对1天中的写入次数进行限制。写入次数的限制默认为36次。应根据需要，通过SD771更改限制次数。超过设置的写入次数时，将发生错误(错误代码：3291H)。此外，每天写入至数据存储器的指令执行次数将在下述时机被初始化为0。
 - 电源OFF→ON时，复位→复位解除时
 - 由于时间的推进，时钟数据的日期(年、月、日)更改时
 - 通过时钟数据的更改功能，控制器内部的时钟数据的年、月、日更改时
- 写入至软元件数据存储用文件的时机将变为执行END指令时。执行SP.DEVST指令之后的END指令执行时将写入到软元件数据存储用文件中。因此，根据写入点数，至软元件数据存储用文件的写入有可能会跨越多个扫描，因此应通过完成软元件判断写入是否完成。

出错

错误代码(SD0)	内容
2840H	参数中未进行软元件数据存储用文件的设置时。
3285H	(n)的设置数据为0时。
3291H	执行SP.DEVST指令时，当天的写入次数超过了SD771中指定的值时。 执行SP.DEVST指令时，SD771中设置了超过1~32767的范围的值时。

14.4 文件操作指令

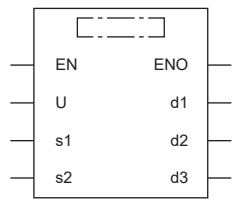
从指定文件的数据读取

SP. FREAD

从SD存储卡的指定文件读取软元件数据。

梯形图	ST
	<pre>ENO:=SP_FREAD(EN, U, s1, s2, d1, d2, d3);</pre>

FBD/LD



■执行条件

指令	执行条件
SP. FREAD	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(U)	虚拟	—	软元件名	ANY16
(s1)	驱动器指定	2(固定)*1	字	ANY16
(d1)	存储了控制数据的起始软元件	☞ 595页 控制数据 (d1)	字	ANY16_ARRAY (要素数: 8)
(s2)	存储了文件名的起始软元件	—	Unicode字符串	ANYSTRING_DOUBLE
(d2)	存储读取的数据的起始软元件	—	字	ANY16*2
(d3)	通过处理完成置为ON的位软元件 (但是, 异常完成时 (d3)+1也将ON)	—	位	ANYBIT_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能设置SD存储卡的驱动器2。

*2 通过标签进行设置的情况下, 应在确保动作所需区域的前提下定义数组, 指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它 (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)	○	—	○	—	—	—	—	—	—	—	—	—	—

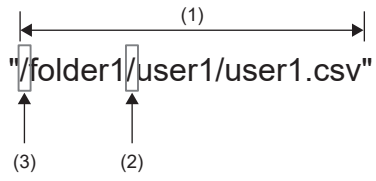
■控制数据 (d1)

操作数: (d1)				
软元件	项目	内容	设置范围	设置方
+0	执行/完成类型	指定执行类型。 ■00**H: 二进制读取 • 0000H: BIN16位数据 • 0001H: BIN32位数据 ■01**H: CSV格式转换读取 • 0100H: 10进制(16位数据) • 0110H: 10进制(32位数据) • 0120H: 16进制(16位数据) • 0121H: 16进制(32位数据) • 0130H: 字符串(ASCII数据) • 0140H: 浮点实数(单精度实数) • 0141H: 浮点实数(双精度实数)	0000H 0001H 0100H 0110H 0120H 0121H 0130H 0140H 0141H	用户
+1	完成状态	指令完成时存储完成状态。 • 0000H: 正常完成 • 0000H以外: 异常完成(错误代码(☞P.640页 文件操作指令中发生的错误代码))	—	系统
+2	请求读取数据数	指定希望读取的数据数。处理单位及设置范围, 根据(d1)+0的执行/完成类型而有所不同。 ■在(d1)+0中指定了“二进制读取”时 • 指定BIN16位数据时: 字单位(1~65535)*1*2 • 指定BIN32位数据时: 双字单位(1~32767) ■(d1)+0中指定了“CSV格式转换读取”时 • 指定10进制数(16位数据)时: 要素数(1~65535)*1*2 • 指定10进制数(32位数据)时: 要素数(1~32767) • 指定16进制数(16位数据)时: 要素数(1~65535) • 指定16进制数(32位数据)时: 要素数(1~32767) • 指定字符串(ASCII数据)时: 要素数(1~1023) • 指定浮点实数(单精度实数)时: 要素数(1~32767) • 指定浮点实数(双精度实数)时: 要素数(1~16383)	1~65535	用户
+3	最大读取数据数	■在(d1)+0中指定“0130H: 字符串(ASCII数据)”时 • 要素内的字符数合计大小 ■在(d1)+0中指定“0130H: 字符串(ASCII数据)”以外时 • 固定为0	0、1~65535	用户
+4 +5	文件位置	■在(d1)+0中指定了“二进制读取”时 • 00000000H: 从文件的起始开始 • 00000001H~FFFFFFFEH: 从指定位置开始(单位取决于(d1)+7的数据类型指定。) • FFFFFFFFH: 不能指定 ■(d1)+0中指定了“CSV格式转换读取”时 • 00000000H: 从文件的起始开始 • 00000001H~FFFFFFFEH: 从指定行开始 • FFFFFFFFH: 从上次的读取位置继续	00000000H~ FFFFFFFHH	用户
+6	列数指定	在(d1)+0指定了“二进制读取”时, 必须指定0。 在(d1)+0中指定了“CSV格式转换读取”时, 指定进行读取的列数。 • 0: 无列。变为1行。 • 0以外: 变为指定数的列。	0000H~FFFFH (0~65535)	用户
+7	数据类型指定	• 0: 字 • 1: 偶数字节*1 • 2: (d1)+0中指定的数据类型的单位 • 3: 奇数字节*1*2 *0: “字”及“1: 偶数字节”、“3: 奇数字节”仅在(d1)+0中指定了“0000H: BIN16位数据”或“0100H: 10进制(16位数据)”的情况下才能指定。	0、1、2、3	用户

*1 数据类型指定(d1)+7中指定“1: 偶数字节”或“3: 奇数字节”时, 请求读取数据数(d1)+2的设置范围为1~32767。

*2 指定奇数字节(d1)+7时, 对希望读取的奇数字字节数加1字节, 设为字单位。应在请求读取数据数(d1)+2中指定已设为字单位的数据数。

■文件名 (s2)

操作数: (s2)				
软元件	项目	内容	设置范围	设置方
+0~+□	文件名字符串	<p>指定存储有文件的文件夹路径、文件名的字符串。</p> <ul style="list-style-type: none"> 应在不超过253字符的范围内设置文件夹路径+文件名(包含扩展名)。 应在不超过244字符的范围内设置文件夹路径。(不含文件的分隔符) 应在不超过10的范围内设置文件夹路径的分层数。 省略文件名的扩展名时,应从“.”(点号)开始省略。 文件名(不包括扩展名)应在1个字符及以上60个字符及以内,扩展名应在点号+3个字符及以内。文件名(不包括扩展名)为61个字符及以上时,即使有扩展名也将被忽略,变为“.BIN”或“.CSV”。  <p>(1) 最多253字符 (2) 文件夹路径、文件的分割符使用“/”或“\”。 (3) 可以省略。省略时, (1) 最多为252字符。</p>	Unicode字符串	用户

■读取的数据 (d2)

操作数: (d2)				
软元件	项目	内容	设置范围	设置方
+0	读取结果数据数	对 (d1)+2中指定的数据数, 设置实际读取的数据数。值的单位取决于 (d1)+7的数据类型指定。	—	系统
+1~+□	读取数据	存储读取的数据。	—	系统

功能

- 从指定文件读取数据。根据控制数据的执行/完成类型, 指定文件的读取形式。
- 读取的对象仅为SD存储卡。
- 检测出SP. FREAD指令的处理完成后, 执行扫描的END指令时, 处理完成 (d3) 的位软元件将自动置为ON, 通过下一个扫描的END指令置为OFF。
- SP. FREAD指令异常完成时, 异常完成 (d3)+1的软元件将在与处理完成 (d3) 的软元件相同的时机变为ON/OFF。
- SP. FREAD指令执行中, SM753(文件访问中)将变为ON。
- SM753为ON时, 将不能执行SP. FREAD指令。(执行的情况下将变为无处理。)
- 执行指令时检测到错误的情况下, 处理完成 (d3)、异常完成 (d3)+1及SM753不变为ON。
- 指定奇数字节时, 如下存储数据。(指定的最后的b8~b15中已存储了指令执行前的数据。)

	b15	...	b8	b7	...	b0
(d2)						
(d2)+1						
(d2)+2						
(d2)+3						
(d2)+4						
(d2)+5						

- 数据的请求读取数据数(d1)+2、文件位置(d1)+4、(d1)+5以及读取结果数据数(d2)+0的处理单位，根据执行/完成类型(d1)+0及数据类型指定(d1)+7的组合进行指定。

执行/完成类型 (d1)+0	数据类型指定 (d1)+7	处理单位及设置范围			
		请求读取数据数 (d1)+2	文件位置 (d1)+4、(d1)+5	读取结果数据数 (d2)+0	
二进制读取	0000H: BIN16位数据	0: 字	字(1~65535)	字(00000000H~7FFFFFFFH)	字
		1: 偶数字节	字(1~32767)	字节(00000000H~FFFFFFFEH)	字节
		2: 执行/完成类型中指定的数据类型的单位	字(1~65535)	字(00000000H~7FFFFFFFH)	字
		3: 奇数字节	字(1~32767)	字节(00000000H~FFFFFFFEH)	字节
	0001H: BIN32位数据	0: 字 1: 偶数字节	(不能指定)		
		2: 执行/完成类型中指定的数据类型的单位	双字(1~32767)	双字(00000000H~3FFFFFFFH)	双字
3: 奇数字节		(不能指定)			
CSV格式转换读取	0100H: 10进制(16位数据)	0: 字	要素数(1~65535)	行数	字
		1: 偶数字节	要素数(1~32767)	行数	字节
		2: 执行/完成类型中指定的数据类型的单位	要素数(1~65535)	行数	字
		3: 奇数字节	要素数(1~32767)	行数	字节
	0110H: 10进制数(带符号32位数据)	0: 字 1: 偶数字节	(不能指定)		
		2: 执行/完成类型中指定的数据类型的单位	要素数(1~32767)	行数	双字
		3: 奇数字节	(不能指定)		
	0120H: 16进制(16位数据)	0: 字 1: 偶数字节	(不能指定)		
		2: 执行/完成类型中指定的数据类型的单位	要素数(1~65535)	行数	字
		3: 奇数字节	(不能指定)		
	0121H: 16进制(32位数据)	0: 字 1: 偶数字节	(不能指定)		
		2: 执行/完成类型中指定的数据类型的单位	要素数(1~32767)	行数	双字
		3: 奇数字节	(不能指定)		
	0130H: 字符串(ASCII数据)	0: 字 1: 偶数字节	(不能指定)		
		2: 执行/完成类型中指定的数据类型的单位	要素数(1~1023)	行数	要素数
		3: 奇数字节	(不能指定)		
	0140H: 浮点实数(单精度实数)	0: 字 1: 偶数字节	(不能指定)		
		2: 执行/完成类型中指定的数据类型的单位	要素数(1~32767)	行数	双字
		3: 奇数字节	(不能指定)		
	0141H: 浮点实数(双精度实数)	0: 字 1: 偶数字节	(不能指定)		
		2: 执行/完成类型中指定的数据类型的单位	要素数(1~16383)	行数	4字
		3: 奇数字节	(不能指定)		

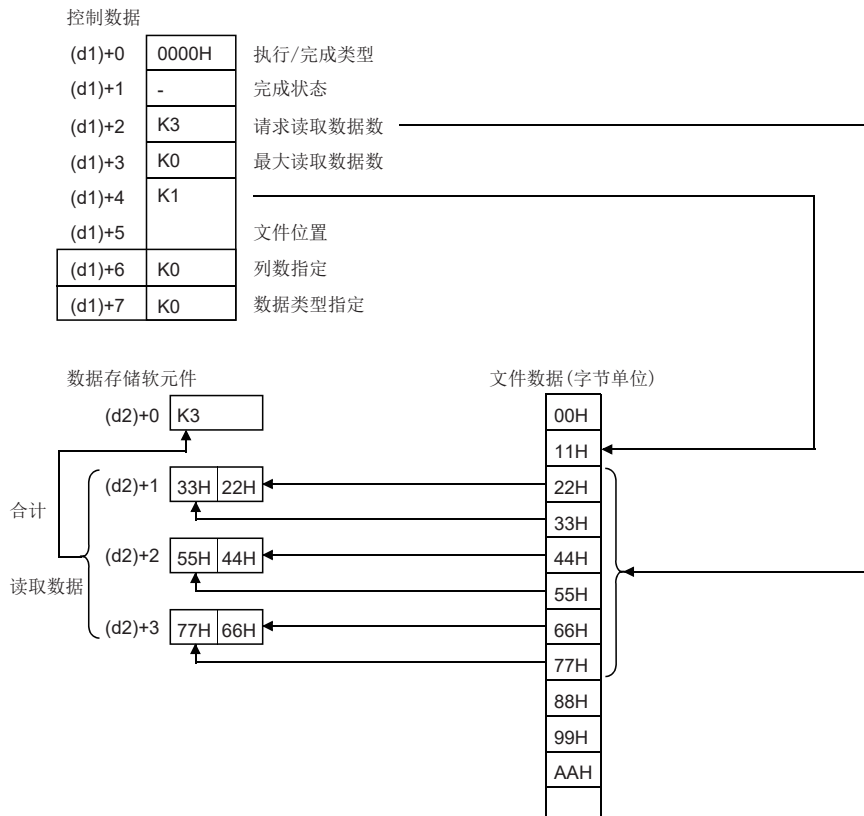
■二进制读取时

- 省略了对象文件的扩展名的情况下，扩展名将变为“.BIN”。
- 指定了不存在的文件的情况下将变为错误。
- 与现有文件的大小相比，指定了更大文件位置的情况下，将变为0点的读取，正常完成。

二进制读取的示例如下所示。

例

二进制读取(BIN16位数据)的示例



■CSV格式转换读取时

- 依照CSV格式文件的要素(Excel®的单元格)行进方向按顺序读取,并存储到软元件中。
- 省略了对象文件的扩展名的情况下,扩展名将变为“.CSV”。
- 指定了不存在的文件的情况下将变为错误。
- 与现有文件的大小相比,指定了更大文件位置的情况下,将变为0点的读取,正常完成。
- 从文件的起始开始请求读取数据数(d1)+2中指定的数据将被读取。读取指定数据数的数据之前,到达文件的最终数据的情况下,读取可读取的数据。
- 指定列数为0的情况下,将忽略CSV格式文件的换行进行读取。
- CSV文件内的字符串数据及读取后软元件中存储的值,根据执行/完成类型而定。

执行/完成类型	CSV内的数据(1要素)	软元件存储值	备注
0100H: 10进制(16位数据)	-32768~-1	-32768~-1 (32768~65535)	视为带符号16位数据时为-32768~-1,视为无符号16位数据时为32768~65535。作为软元件存储值也相同。
	0~32767	0~32767	—
	32768~65535	-32768~-1 (32768~65535)	视为带符号16位数据时为-32768~-1,视为无符号16位数据时为32768~65535。作为软元件存储值也相同。
	<ul style="list-style-type: none"> • 上述以外的数值 • 包含英文字母及符号的字符串 	0	不可转换,因此存储0。
0110H: 10进制(32位数据)	-2147483648~-1	-2147483648~-1 (2147483648~4294967295)	视为带符号32位数据时为-2147483648~-1,视为无符号32位数据时为2147483648~4294967295。作为软元件存储值也相同。
	0~2147483647	0~2147483647	—
	2147483648~4294967295	-2147483648~-1 (2147483648~4294967295)	视为带符号32位数据时为-2147483648~-1,视为无符号32位数据时为2147483648~4294967295。作为软元件存储值也相同。
	<ul style="list-style-type: none"> • 上述以外的数值 • 包含英文字母及符号的字符串 	0	不可转换,因此存储0。
0120H: 16进制(16位数据)	0H~FFFFH	0H~FFFFH	—
	<ul style="list-style-type: none"> • 上述以外的数值 • 包含A~F以外的英文字母及符号的字符串 	0000H	不可转换,因此存储0。
0121H: 16进制(32位数据)	0H~FFFFFFFFH	0H~FFFFFFFFH	—
	<ul style="list-style-type: none"> • 上述以外的数值 • 包含A~F以外的英文字母及符号的字符串 	00000000H	不可转换,因此存储0。
0130H: 字符串(ASCII数据)	字符串(最多1999字符)	字符串(最多1999字符)	于末尾添加NULL(00H)。若CSV内的字符串为偶数字节,则将于下一字中存储0000H。若CSV内的字符串中包含00H,则忽略此00H。
	字符串(2000字符及其以上)		1要素内的字符数若超过1999字符,则到1999字符的部分将作为1要素被读取。第2000个及其以上的字符不被读取,并执行下一要素的读取。
0140H: 浮点实数(单精度实数)	$-2^{128} < \text{数据} \leq -2^{126}$, 0, $2^{-126} \leq \text{数据} < 2^{128}$ 范围的值	左述范围的值	以小数点形式/指数形式任一形式转换。
	上述以外的数值	0	不可转换,因此存储0。
0141H: 浮点实数(双精度实数)	$-2^{1024} < \text{数据} \leq -2^{1022}$, 0, $2^{-1022} \leq \text{数据} < 2^{1024}$ 范围的值	左述范围的值	以小数点形式/指数形式任一形式转换。
	上述以外的数值	0	不可转换,因此存储0。

例

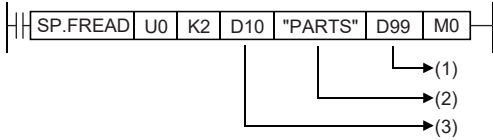
指定CSV格式转换读取(字符串(ASCII数据))时

[以CSV格式保存的数据]

PARTS.CSV

No.	Name	Value1	Value2	CR	LF
AA_0001	Prts_A	100	200	CR	LF
BB_0002	Prts_B	300	400	CR	LF

[软件中读入的数据]



[控制数据]

D10	H0130
D11	H0000
D12	K6
D13	K100
D14	K2
D15	K3
D16	K3
D17	K2

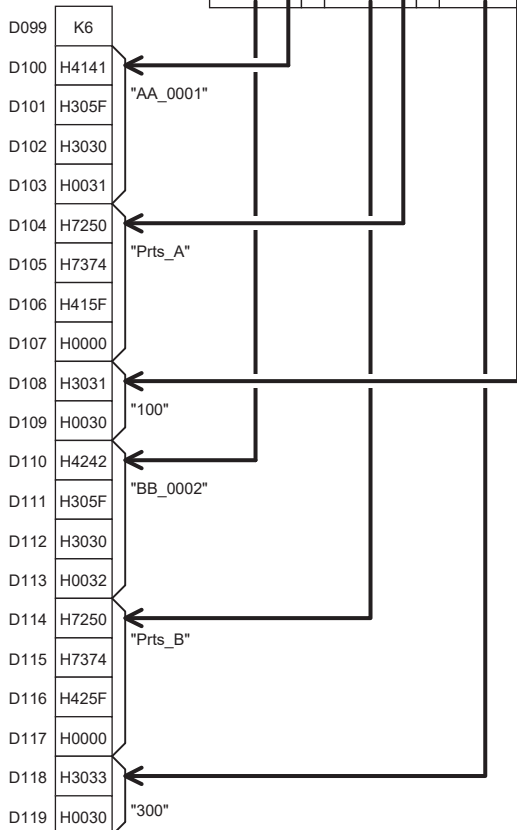
对左述框起来的部分进行读取。
(从PARTS.CSV的第2行开始以每行3列、合计6要素的方式进行读取)

- (1) 读取的数据
- (2) 文件名
- (3) 控制数据

- D10: 执行/完成类型
- D11: 完成状态
- D12: 请求读取数据数
- D13: 最大读取数据数
- D14、D15: 文件位置
- D16: 列数指定
- D17: 数据类型指定

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: 读取结果数据数
- D100~D103: 第2行第1列的字符串
- D104~D107: 第2行第2列的字符串
- D108~D109: 第2行第3列的字符串
- D110~D113: 第3行第1列的字符串
- D114~D117: 第3行第2列的字符串
- D118~D119: 第3行第3列的字符串

例

CSV格式转换读取(10进制数(16位数据))指定时指定数为0时

[通过Excel创建的数据]

	A	B	C
1	大/小项目		测定值
2	长度	1	3
3	温度	-21	

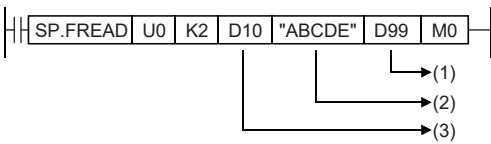


[以CSV格式保存的数据]

大/小项目	,	,	测定值	CR	LF
长度	,	1	,	3	CR LF
温度	,	-21	,		CR LF



[软件元件中读入的数据]



- (1) 读取的数据
- (2) 文件名
- (3) 控制数据

[控制数据]

D10	0100H
D11	-
D12	K9
D13	K0
D14	K0
D15	K0
D16	K0
D17	K0

- D10: 执行/完成类型
- D11: 完成状态
- D12: 请求读取数据数
- D13: 最大读取数据数
- D14、D15: 文件位置
- D16: 列数指定
- D17: 数据类型指定

[读入的数据]

D99	K9
D100	K0
D101	K0
D102	K0
D103	K0
D104	K1
D105	K3
D106	K0
D107	K-21
D108	K0

- D99: 读取结果数据数
- D100: “大/小项目”为数值以外的数据，因此存储转换数据(0)。
- D101: “”为数值以外的数据，因此存储转换数据(0)。
- D102: “测定值”为数值以外的数据，因此存储转换数据(0)。
- D103: “长度”为数值以外的数据，因此存储转换数据(0)。
- D104: “1”为数值，因此转换为二进制值。
- D105: “3”为数值，因此转换为二进制值。
- D106: “温度”为数值以外的数据，因此存储转换数据(0)。
- D107: “-21”为数值，因此转换为二进制值。
- D108: “”为数值以外的数据，因此存储转换数据(0)。

- 即使列数在各个行中不相同的情况下，也将忽略行进行读取。

例

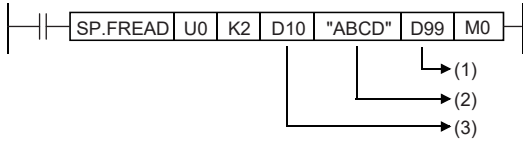
读取时的列数在各个行中不相同的情况下

[以CSV格式保存的数据]

大/小项目	,	,	测定值	,	余量	CR	LF
长度	CR	LF					
温度	,	-21	,	CR	LF		



[软件中读入的数据]



- (1) 读取的数据
- (2) 文件名
- (3) 控制数据

[控制数据]

D10	0100H
D11	-
D12	K7
D13	K0
D14	K0
D15	K0
D16	K0
D17	K0

- D10: 执行/完成类型
- D11: 完成状态
- D12: 请求读取数据数
- D13: 最大读取数据数
- D14、D15: 文件位置
- D16: 列数指定
- D17: 数据类型指定

[读入的数据]

D99	K7
D100	K0
D101	K0
D102	K0
D103	K0
D104	K0
D105	K0
D106	K-21

- D99: 读取结果数据数
- D100: “大/小项目”为数值以外的数据，因此存储转换数据(0)。
- D101: “,”为数值以外的数据，因此存储转换数据(0)。
- D102: “测定值”为数值以外的数据，因此存储转换数据(0)。
- D103: “余量”为数值以外的数据，因此存储转换数据(0)。
- D104: “长度”为数值以外的数据，因此存储转换数据(0)。
- D105: “温度”为数值以外的数据，因此存储转换数据(0)。
- D106: “-21”为数值，因此转换为二进制值。

要点

在Excel中，不能创建列数在各个行中不相同的文件。在修改了CSV文件的情况下发生。

- 指定列数为0以外的情况下，作为指定的列数的表读取CSV格式的文件。超过了指定列数的要素将被忽略。

例

CSV格式转换读取(10进制数(16位数据))指定时指定数为0以外时((d1)+6为2时)

[通过Excel创建的数据]

	A	B	C
1	大/小项目		测定值
2	长度	1	3
3	温度	-21	



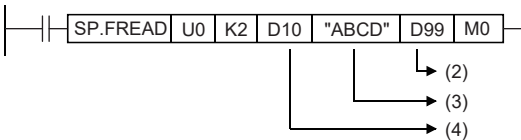
[以CSV格式保存的数据]

大/小项目	,	,	测定值	CR	LF
长度	,	1	,	3	CR LF
温度	,	-21	,		CR LF

(1)



[软件中读入的数据]



(1) 超过指定列数的要素将被忽略。

- (2) 读取的数据
- (3) 文件名
- (4) 控制数据

[控制数据]

D10	0100H
D11	-
D12	K6
D13	K0
D14	K0
D15	K0
D16	K2
D17	K0

- D10: 执行/完成类型
- D11: 完成状态
- D12: 请求读取数据数
- D13: 最大读取数据数
- D14、D15: 文件位置
- D16: 列数指定
- D17: 数据类型指定

[读入的数据]

D99	K6
D100	K0
D101	K0
D102	K0
D103	K1
D104	K0
D105	K-21

- D99: 读取结果数据数
- D100: “大/小项目”为数值以外的数据，因此存储转换数据(0)。
- D101: “,”为数值以外的数据，因此存储转换数据(0)。
- D102: “长度”为数值以外的数据，因此存储转换数据(0)。
- D103: “1”为数值，因此转换为二进制值。
- D104: “温度”为数值以外的数据，因此存储转换数据(0)。
- D105: “-21”为数值，因此转换为二进制值。

- 即使列数在各个行中不相同的情况下，超过了指定列数的要素将被忽略，不足指定列数的列将被补0。

例

读取时的列数在各个行中不相同的情况下

[以CSV格式保存的数据]

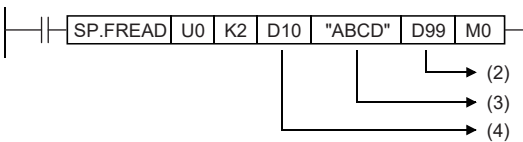
大 / 小项目	,	,	测定值	,	余量	CR	LF
长度	CR	LF					
温度	,	-21		CR	LF		

(1)

(1) 超过指定列数的要素将被忽略。



[软件中读入的数据]



- (2) 读取的数据
- (3) 文件名
- (4) 控制数据

[控制数据]

D10	0100H
D11	-
D12	K6
D13	K0
D14	K0
D15	K0
D16	K2
D17	K0

- D10: 执行/完成类型
- D11: 完成状态
- D12: 请求读取数据数
- D13: 最大读取数据数
- D14、D15: 文件位置
- D16: 列数指定
- D17: 数据类型指定

[读入的数据]

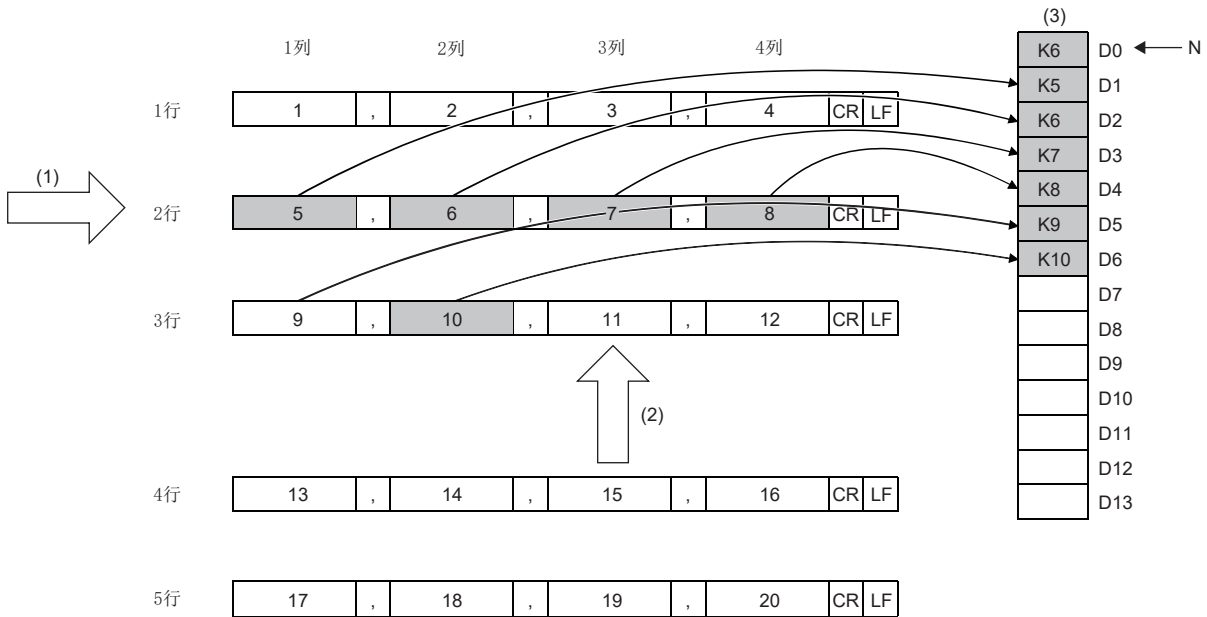
D99	K6
D100	K0
D101	K0
D102	K0
D103	K0
D104	K0
D105	K-21

- D99: 读取结果数据数
- D100: “大/小项目”为数值以外的数据，因此存储转换数据(0)。
- D101: “”为数值以外的数据，因此存储转换数据(0)。
- D102: “长度”为数值以外的数据，因此存储转换数据(0)。
- D103: 由于无相应的元素，以转换数据(0)补充。
- D104: “温度”为数值以外的数据，因此存储转换数据(0)。
- D105: “-21”为数值，因此转换为二进制值。

- 在CSV格式转换读取中，可以将数据分为多次进行读取。

[指定希望开始读取的行]

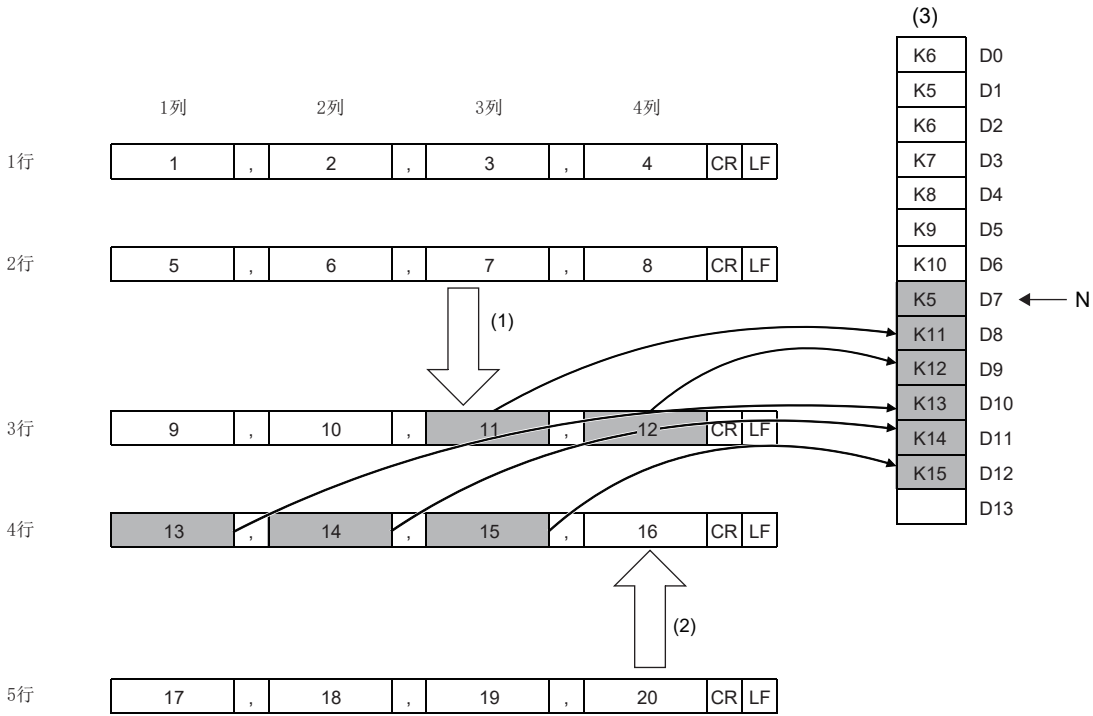
- 执行/完成类型：CSV格式转换读取(10进制数(BIN16位数据))
- 列数指定：4H
- 数据类型指定：字
- 文件位置：2H
- 读取起始软元件：D0
- 读取结果数据数：6H



- (1) 开始行
(2) 下一个开始位置
(3) 软元件存储数据(读取数据)
N: 数据数

[从上次读取位置开始继续读取]

- 执行/完成类型：CSV格式转换读取(10进制数(BIN16位数据))
- 列数指定：4H
- 数据类型指定：字
- 文件位置：FFFFFFFH(从上次读取位置开始继续)
- 读取起始软元件：D7
- 读取结果数据数：5H



- (1) 开始行
 (2) 下一个开始位置
 (3) 软元件存储数据(读取数据)
 N: 数据数

要点

- 从上次读取位置开始继续进行读取的情况下，将“执行/完成类型”、“列数指定”、“数据类型指定”设置为与上次不相同的情况下，将无法从上次的读取位置开始正常添加。
- 从上次读取位置开始继续读取数据的途中，如果执行其它设置的SP.FREAD指令及SP.FWRITE指令，将无法从上次的读取位置开始正常添加。

注意事项

- SP.FREAD指令，请勿通过中断程序执行。通过中断程序执行的情况下，可能导致误动作。
- CSV格式转换读取时，读取多个要素数时，应在预先确保读取的数据区域中各要素的合计容量后，再执行指令。由于读取的数据从(d2)+1开始存储，作为(d2)确保需要的字数((要素数的合计字数)+1)变为字。
- CSV格式转换读取(字符串(ASCII数据))指定时，应将合计容量(字单位)设置为(d1)+3的最大读取数据数。
 [例]从CSV文件对1要素的字符数为100字符的字符串读取100要素时
 (100(字符)+2(NULL))×100(要素)=10200字节=5100字
 因此，将最大读取的数据数((d1)+3)设为5100，确保(d2)有5101字的区域。
- 由于SP.FREAD指令发生文件读取，执行指令时扫描时间有可能会延长。
- SP.FREAD指令不支持带BOM的字符代码。执行使用了带BOM字符代码的文件时，将读取预期外的数据。读取通过EXCEL创建的带BOM的CSV文件的内容时，应通过文本编辑器等将字符代码更改为无BOM。

出错

错误代码(SD0)	内容
2820H	读取的数据容量超过读取软元件容量的情况下。
3285H	通过驱动器指定(s1)指定的驱动器为SD存储卡以外时。 控制数据(d1)及其以后设置的值超出设置范围时。 ☞ 595页 控制数据(d1)
	无法读取在(s2)中指定的文件名字符串时。 • 指定的文件名字符串的字符数超出了范围。 • 设置了不能使用的值。 • 对所指定的文件名字符串的末尾指定了分隔符。 • 指定的文件名字符串的文件名(不包括扩展名)为0字符时。 ☞ 595页 控制数据(d1)
3297H	(d1)+0的执行/完成类型及(d1)+7的数据类型指定变为不能指定的组合时。 ☞ 595页 控制数据(d1)

SP.FREAD指令异常完成的情况下，完成状态(d1)+1中指定的软元件中将存储错误代码。(指令发生运算错误时将不存储。)

关于完成状态(d1)+1中存储的错误代码，请参阅下述内容。

☞ 640页 文件操作指令中发生的错误代码

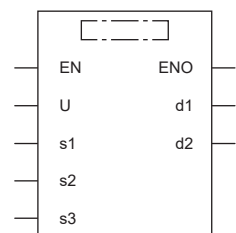
至指定文件的数据写入

SP.FWRITE

将软元件数据写入到SD存储卡的指定文件中。

梯形图	ST
	<pre>ENO:=SP_FWRITE(EN, U, s1, s2, s3, d1, d2);</pre>

FBD/LD



■执行条件

指令	执行条件
SP.FWRITE	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(U)	虚拟	—	软元件名	ANY16
(s1)	驱动器指定	2(固定)*1	字	ANY16
(d1)	存储了控制数据的起始软元件	☞ 609页 控制数据(d1)	字	ANY16_ARRAY (要素数: 8)
(s2)	存储了文件名的起始软元件	—	Unicode字符串	ANYSTRING_DOUBLE
(s3)	存储了数据的起始软元件	—	字	ANY16*2
(d2)	通过处理完成置为ON的位软元件	—	位	ANYBIT_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能设置SD存储卡的驱动器2。

*2 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它(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)	○	—	○	—	—	—	—	—	—	—	—	—	—	—

■控制数据 (d1)

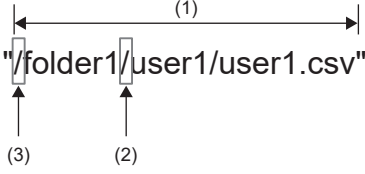
操作数: (d1)				
软元件	项目	内容	设置范围	设置方
+0	执行/完成类型	指定执行类型。 ■00**H: 二进制写入 • 0000H: BIN16位数据 • 0001H: BIN32位数据 ■01**H: CSV格式转换写入 • 0100H: 10进制数(带符号16位数据) • 0101H: 10进制数(无符号16位数据) • 0110H: 10进制数(带符号32位数据) • 0111H: 10进制数(无符号32位数据) • 0120H: 16进制(16位数据) • 0121H: 16进制(32位数据) • 0130H: 字符串(ASCII数据) • 0140H: 浮点实数(单精度实数) • 0141H: 浮点实数(双精度实数)	0000H 0001H 0100H 0101H 0110H 0111H 0120H 0121H 0130H 0140H 0141H	用户
+1	完成状态	指令完成时存储完成状态。 • 0000H: 正常完成 • 0000H以外: 异常完成(错误代码)( 640页 文件操作指令中发生的错误代码)	—	系统
+2	写入结果数据数	对(s3)中指定的数据, 输入实际写入的数据数。 值的单位取决于(d1)+7的数据类型指定。	—	系统
+3	使用用途设置区域	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> b15 b0 <div style="text-align: center; margin: 5px 0;">...</div> <div style="text-align: center; margin: 5px 0;">0</div> <div style="text-align: right; margin: 5px 0;">1/0</div> </div> b0: 设置写入开始位置*3 (d1)+0中指定了“二进制写入”时, 必须指定0。 (d1)+0中指定了“CSV格式转换写入”、在(d1)+4及(d1)+5中指定了“添加到文件的最后”时, 应指定写入开始位置。 • 0: 添加到文件的最后 • 1: 将文件最后的换行代码转换为逗号并添加(从最后的行接续写入)	如左所示	用户
+4 +5	文件位置	■(d1)+0中指定了“二进制写入”时 • 00000000H: 从文件的起始开始 • 00000001H~FFFFFFFH: 从指定位置开始(单位取决于(d1)+7的数据类型指定。) • FFFFFFFFH: 添加到文件的最后 ■(d1)+0中指定了“CSV格式转换写入”时 • 00000000H~FFFFFFFH: 从文件的起始开始 • FFFFFFFFH: 添加到文件的最后	00000000H~ FFFFFFFH	用户
+6	列数指定	(d1)+0中指定了“二进制写入”时, 必须指定0。 (d1)+0中指定了“CSV格式转换写入”时, 应指定进行写入的列数。 • 0: 无列。变为1行。 • 0以外: 变为指定数的列。	0000H~FFFFH (0~65535)	用户
+7	数据类型指定	• 0: 字 • 1: 偶数字节*1 • 2: (d1)+0中指定的数据类型的单位 • 3: 奇数字节*1*2 “0: 字”及“1: 偶数字节”、“3: 奇数字节”仅在(d1)+0中指定了“0000H: BIN16位数据”或“0100H: 10进制(带符号16位数据)”的情况下才能指定。	0、1、2、3	用户

*1 在数据类型指定(d1)+7中指定“1: 偶数字节”或“3: 奇数字节”时, 请求写入数据数(s3)+0的设置范围为1~32767。

*2 指定奇数字节(d1)+7时, 对希望写入的奇数字字节数加1字节, 设为字单位。应在请求写入数据数(s3)+0中指定已设为字单位的数据数。

*3 只有在执行/完成类型(d1)+0中指定了“CSV格式转换写入”并且在文件位置(d1)+4及(d1)+5中指定了“添加到文件的最后”时为有效。除此以外的情况下将变为错误状态。

■文件名 (s2)

操作数: (s2)				
软元件	项目	内容	设置范围	设置方
+0~+□	文件名字符串	<p>指定存储有文件的文件夹路径、文件名的字符串。</p> <ul style="list-style-type: none"> 应在不超过253字符的范围内设置文件夹路径+文件名(包含扩展名)。 应在不超过244字符的范围内设置文件夹路径。(不含文件的分隔符) 应在不超过10的范围内设置文件夹路径的分层数。 省略文件名的扩展名时,应从“.”(点号)开始省略。 文件名(不包括扩展名)应在1个字符及以上60个字符及以内,扩展名应在点号+3个字符及以内。文件名(不包括扩展名)为61个字符及以上时,即使有扩展名也将被忽略,变为“.BIN”或“.CSV”。  <p>(1) 最多253字符 (2) 文件夹路径、文件的分隔符使用“/”或“\”。 (3) 可以省略。省略时, (1)最多为252字符。</p>	Unicode字符串	用户

■写入数据 (s3)

操作数: (s3)				
软元件	项目	内容	设置范围	设置方
+0	请求写入数据数	<p>指定请求写入的数据数。</p> <p>处理单位及设置范围,根据(d1)+0的执行/完成类型的设置而有所不同。</p> <p>■(d1)+0中指定了“二进制写入”时</p> <ul style="list-style-type: none"> 指定BIN16位数据时:字单位(1~65535)*1*2 指定BIN32位数据时:双字单位(1~32767) <p>■(d1)+0中指定了“CSV格式转换写入”时:要素数</p> <ul style="list-style-type: none"> 指定10进制数(带符号16位数据)时:字单位(1~65535)*1*2 指定10进制数(无符号16位数据)时:字单位(1~65535) 指定10进制数(带符号32位数据)时:双字单位(1~32767) 指定10进制数(无符号32位数据)时:双字单位(1~32767) 指定16进制数(16位数据)时:字单位(1~65535) 指定16进制数(32位数据)时:双字单位(1~32767) 指定字符串(ASCII数据)时:要素数(1~1023) 指定浮点实数(单精度实数)时:双字单位(1~32767) 指定浮点实数(双精度实数)时:4字单位(1~16383) 	1~65535	用户
+1~+□	写入数据	存储请求写入的数据。	0000H~FFFFH	用户

*1 在数据类型指定(d1)+7中指定“1:偶数字节”或“3:奇数字节”时,请求写入数据数(s3)+0的设置范围为1~32767。

*2 指定奇数字节(d1)+7时,对希望写入的奇数字字节数加1字节,设为字单位。应在请求写入数据数(s3)+0中指定已设为字单位的数据数。

功能

- 将指定数据数的数据写入到指定的文件中。根据控制数据的执行/完成类型，指定文件的写入形式。
- 写入对象仅为SD存储卡。
- 检测出SP.FWRITE指令的处理完成后，执行扫描的END指令时，处理完成(d2)的位软元件将自动置为ON，通过下一个扫描的END指令置为OFF。
- SP.FWRITE指令异常完成时，异常完成(d2)+1的软元件将在与处理完成(d2)的软元件相同的时机变为ON/OFF。
- SP.FWRITE指令执行中，SM753(文件访问中)将变为ON。
- SM753为ON时，将不能执行SP.FWRITE指令。(执行的情况下将变为无处理。)
- 执行指令时检测出的错误，处理完成(d2)、异常完成(d2)+1及SM753不变为ON。
- 请求写入数据数((s3)+0)、文件位置((d1)+4、(d1)+5)以及写入结果数据数((d1)+2)的处理单位，根据执行/完成类型(d1)+0及数据类型指定(d1)+7的组合进行指定。

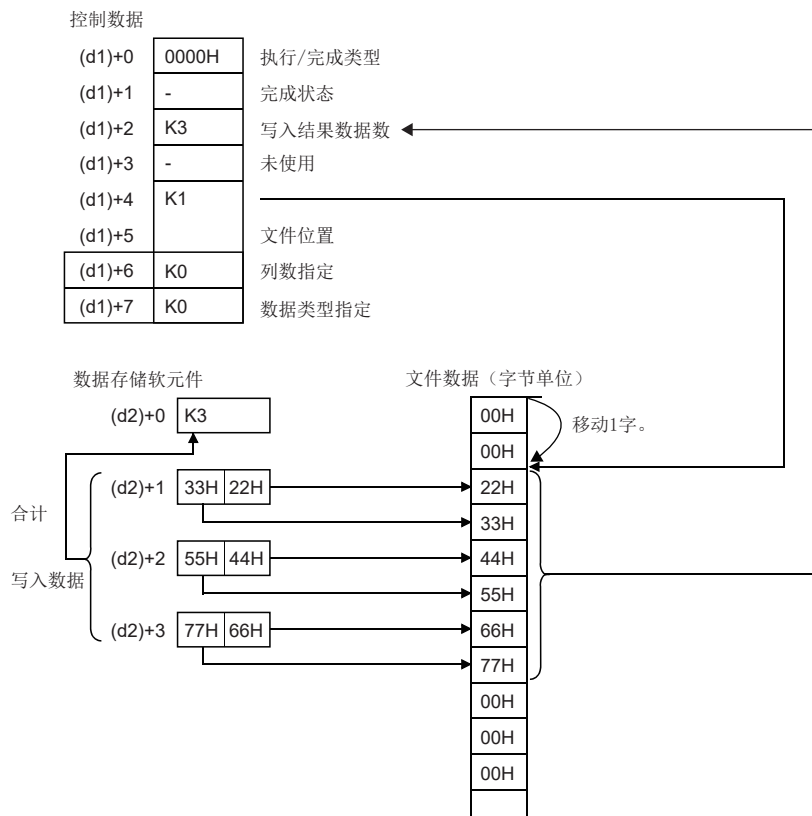
执行/完成类型 (d1)+0	数据类型指定 (d1)+7	处理单位及设置范围			
		请求写入数据数 (s3)+0	文件位置 (d1)+4、(d1)+5	写入结果数据数 (d1)+2	
二进制写入	0000H: BIN16位数据	0: 字	字(1~65535)	字(00000000H~7FFFFFFFH, FFFFFFFFH)	字
		1: 偶数字节	字(1~32767)	字节(00000000H~FFFFFFFH)	字节
		2: 执行/完成类型中指定的数据类型的单位	字(1~65535)	字(00000000H~7FFFFFFFH, FFFFFFFFH)	字
		3: 奇数字节	字(1~32767)	字节(00000000H~FFFFFFFH)	字节
	0001H: BIN32位数据	0: 字	(不能指定)		
		1: 偶数字节	(不能指定)		
		2: 执行/完成类型中指定的数据类型的单位	双字(1~32767)	双字(00000000H~3FFFFFFFH, FFFFFFFFH)	双字
	3: 奇数字节	(不能指定)			

执行/完成类型 (d1)+0	数据类型指定 (d1)+7	处理单位及设置范围			
		请求写入数据数 (s3)+0	文件位置 (d1)+4、(d1)+5	写入结果数据数 (d1)+2	
CSV格式转换写入	0100H: 10进制数(带符号16位数据)	0: 字	字(1~65535)	起始/末尾*3	字
		1: 偶数字节	字(1~32767)	起始/末尾*3	字节
		2: 执行/完成类型中指定的数据类型的单位	字(1~65535)	起始/末尾*3	字
		3: 奇数字节	字(1~32767)	起始/末尾*3	字节
0101H: 10进制数(无符号16位数据)	0: 字 1: 偶数字节	(不能指定)			
		2: 执行/完成类型中指定的数据类型的单位	字(1~65535)	起始/末尾*3	字
		3: 奇数字节	(不能指定)		
0110H: 10进制数(带符号32位数据)	0: 字 1: 偶数字节	(不能指定)			
		2: 执行/完成类型中指定的数据类型的单位	双字(1~32767)	起始/末尾*3	双字
		3: 奇数字节	(不能指定)		
0111H: 10进制数(无符号32位数据)	0: 字 1: 偶数字节	(不能指定)			
		2: 执行/完成类型中指定的数据类型的单位	双字(1~32767)	起始/末尾*3	双字
		3: 奇数字节	(不能指定)		
0120H: 16进制(16位数据)	0: 字 1: 偶数字节	(不能指定)			
		2: 执行/完成类型中指定的数据类型的单位	字(1~65535)	起始/末尾*3	字
		3: 奇数字节	(不能指定)		
0121H: 16进制(32位数据)	0: 字 1: 偶数字节	(不能指定)			
		2: 执行/完成类型中指定的数据类型的单位	双字(1~32767)	起始/末尾*3	双字
		3: 奇数字节	(不能指定)		
0130H: 字符串(ASCII数据)	0: 字 1: 偶数字节	(不能指定)			
		2: 执行/完成类型中指定的数据类型的单位	要素数(1~1023)	起始/末尾*3	要素数
		3: 奇数字节	(不能指定)		
0140H: 浮点实数(单精度实数)	0: 字 1: 偶数字节	(不能指定)			
		2: 执行/完成类型中指定的数据类型的单位	双字(1~32767)	起始/末尾*3	双字
		3: 奇数字节	(不能指定)		
0141H: 浮点实数(双精度实数)	0: 字 1: 偶数字节	(不能指定)			
		2: 执行/完成类型中指定的数据类型的单位	4字(1~16383)	起始/末尾*3	4字
		3: 奇数字节	(不能指定)		

*3 设定值为00000000H~FFFFFFEH(自文件的起始开始)或FFFFFFFH(添加到文件的最后)。

■二进制写入时

- 省略了对象文件的扩展名的情况下，扩展名将变为“.BIN”。
- 指定了不存在的文件的情况下，将新建相应文件后，从起始开始添加保存数据。此时新建文件的属性被设置为存档属性。
- 指定了存在的文件的情况下，将从存在的文件的起始开始进行保存。在写入数据的途中，超过了现有的大小的情况下，超过部分的数据将被添加保存。
- 与现有文件的大小相比，指定了更大文件位置的情况下，将变为以0点写入而正常完成。
- 在添加保存数据的途中，来自于媒介的空余区域不足的情况下将变为错误状态。此时，写入及添加保存成功的部分将变为原样写入状态，仅添加保存了可添加保存的部分后异常结束。
- 二进制写入(BIN16位数据)的情况下指定了请求写入数据数、文件位置时的数据写入方法如下所示。

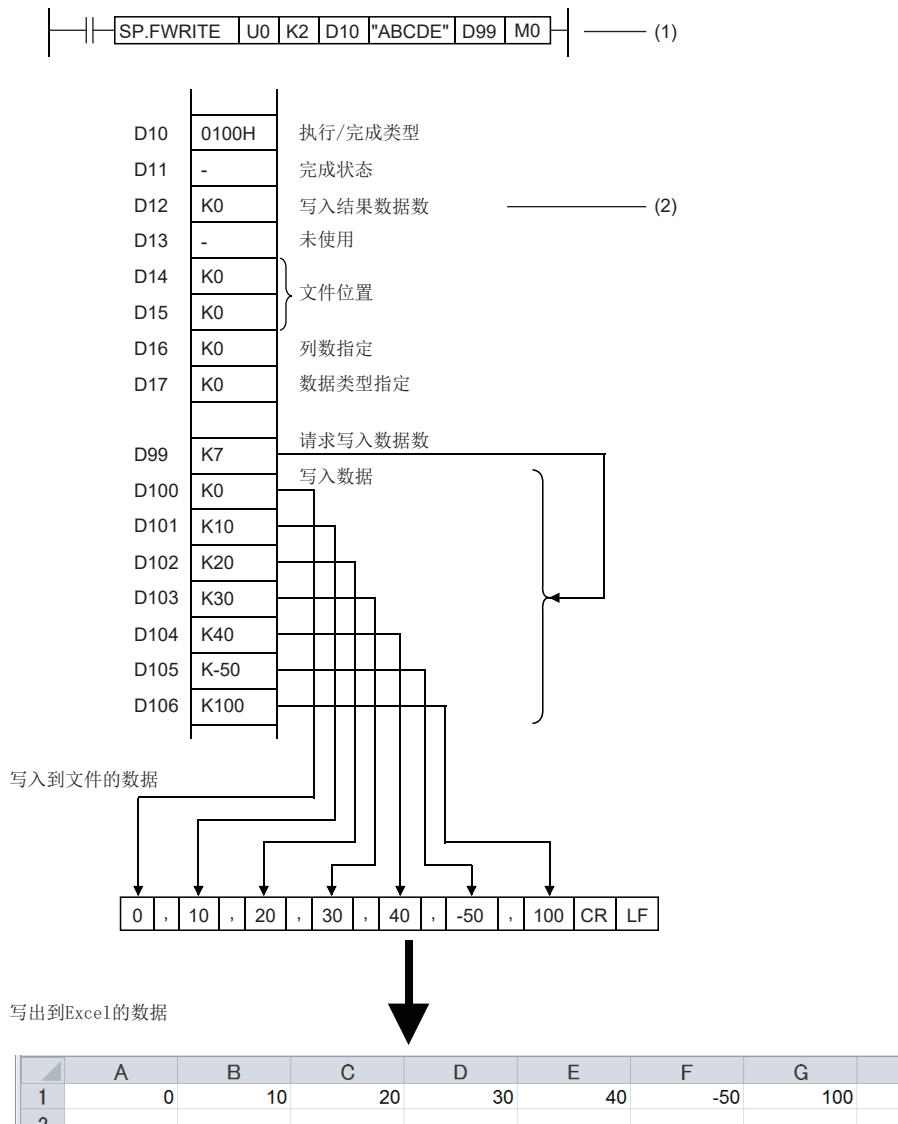


■CSV格式转换写入时

- 省略了扩展名的情况下，扩展名将变为“.CSV”。
- 指定了存在的文件时，其情况如下所示。
 - (d1)+4、(d1)+5中设置了FFFFFFFH以外时，将删除全部文件内容后从起始开始保存数据。
 - (d1)+4、(d1)+5中设置了FFFFFFFH时，将从文件的最后开始保存数据。此时，如果在(d1)+3中指定了1，则将文件最后的换行代码转换为逗号并保存数据。但是，文件的最后为换行代码以外时则不转换为逗号，而从文件的最后开始保存数据。
- 指定了不存在的文件的情况下，将新建相应文件后，从起始开始添加保存数据。此时的新建文件的属性将被设置为存档属性。
- 在数据添加保存的途中，来自于媒介的空余区域不足的情况下将变为错误状态。此时，添加保存成功的部分将变为原样写入状态，仅添加保存了可添加保存的部分后异常结束。
- 指定列数为0的情况下，作为1行的CSV格式文件保存。

例

CSV格式转换写入时指定列数为0的情况下



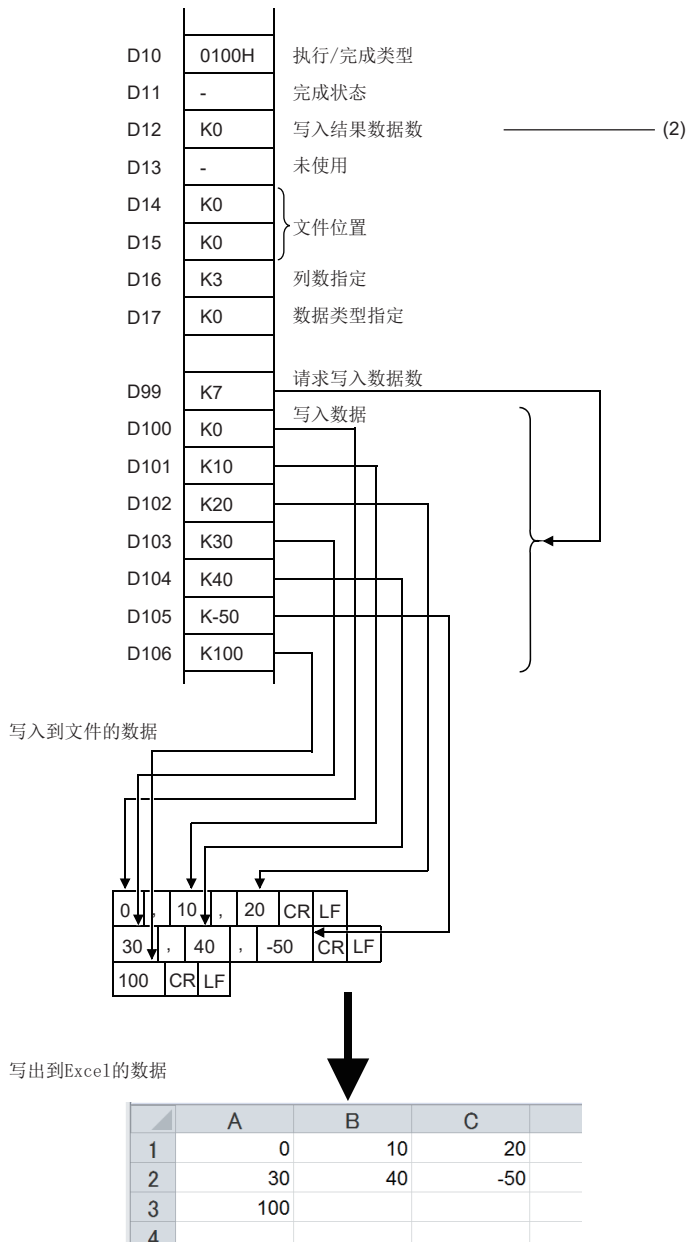
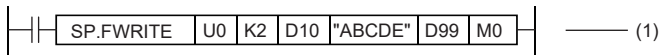
(1) 以字单位进行指定。

(2) 正常完成时，变为与写入数据数相同。

- 指定列数为0以外的情况下，作为指定列数的表保存CSV格式文件。

例

CSV格式转换写入时指定列数为0以外的情况下

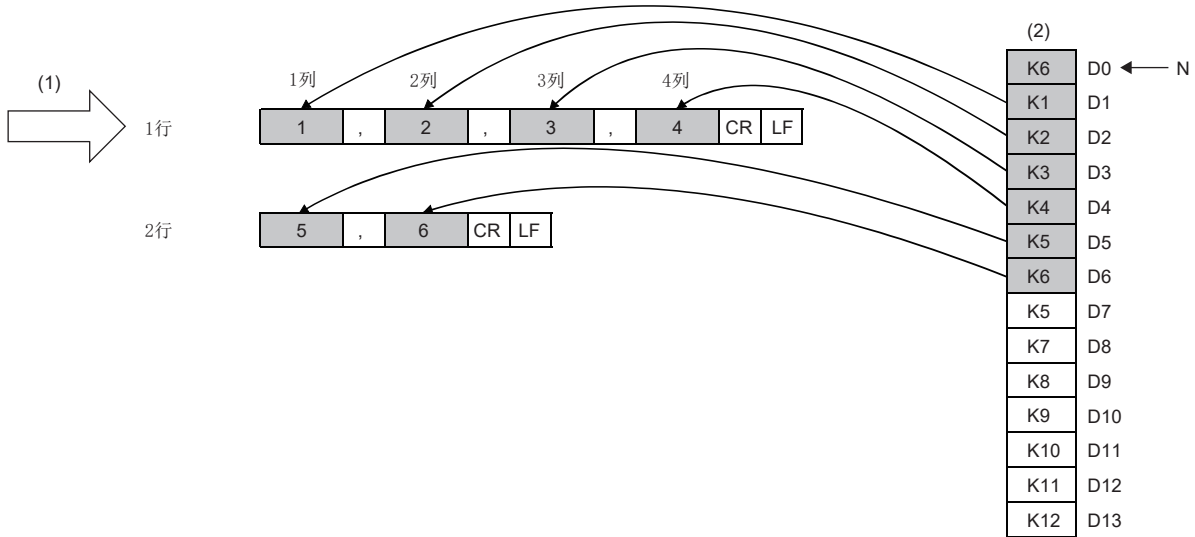


- (1) 以字单位进行指定。
- (2) 正常完成时，变为与写入数据数相同。

- 添加数据时的情况如下所示。

[指定进行写入的文件] (即使文件已存在, 也将其删除后再次新建。)

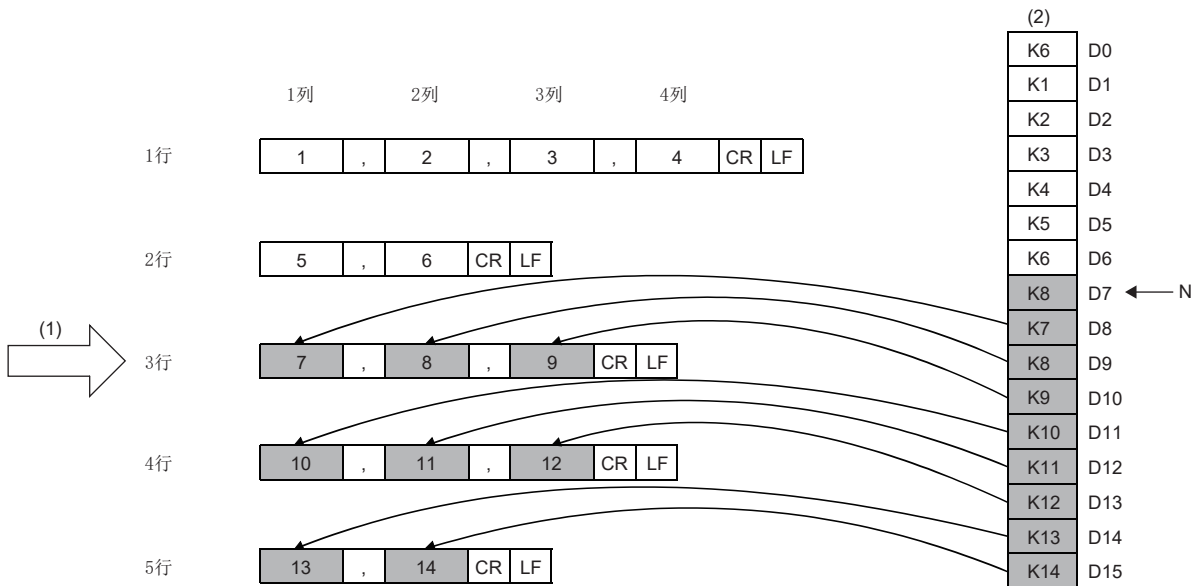
- 执行/完成类型: 写入CSV格式转换(10进制数(带符号16位数据))
- 文件位置: 00000000H(从文件的起始开始)
- 列数指定: 4H
- 数据类型指定: 字
- 写入起始软元件: D0
- 请求写入数据数: 6H



- (1) 开始行
- (2) 软元件存储数据(写入数据)
- N: 数据数

[添加到文件的最后]

- 执行/完成类型：写入CSV格式转换(10进制数(带符号16位数据))
- 使用用途设置区域：0H
- 文件位置：FFFFFFFH(添加到文件的最后)
- 列数指定：3H
- 数据类型指定：字
- 写入起始软元件：D7
- 请求写入数据数：8H



(1) 本次开始位置

(2) 软元件存储数据(写入数据)

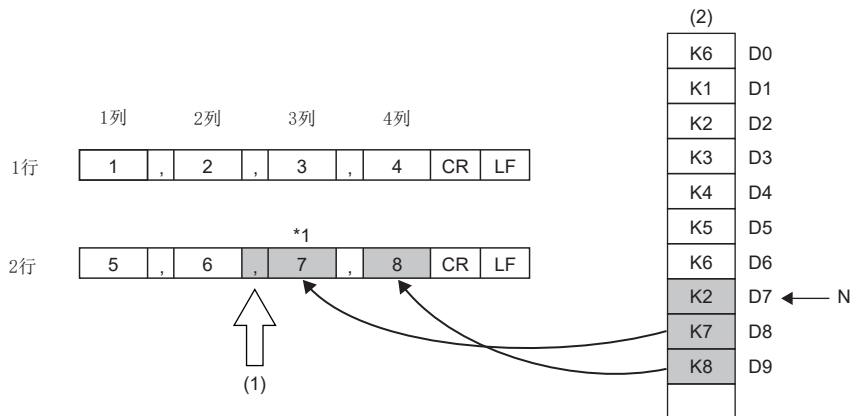
N: 数据数

要点

- “请求写入数据数”，应设置为“列数指定”的整数倍。不是整数倍的情况下，列数将错乱。
- 最后数据的后面必须输入换行代码，因此使用用途设置区域为0时，将从新的一行的起始开始添加。
- 添加到文件的最后的情况下，将“列数指定”从上次写入时进行了更改时，列数将错乱。

[从文件最后的行接续添加]

- 执行/完成类型：写入CSV格式转换(10进制数(带符号16位数据))
- 使用用途设置区域：1H
- 文件位置：FFFFFFFH(添加到文件的最后)
- 数据类型指定：字
- 写入起始软元件：D7
- 请求写入数据数：2H



(1) 本次开始位置

(2) 软元件存储数据(写入数据)

N: 数据数

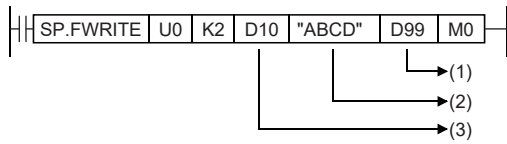
*1 使用用途设置区域设置为1时，文件末尾的换行代码将转换为逗号，并且可以从最后的行接续添加数据。

• 执行/完成类型中指定“字符串(ASCII数据)”时的示例如下所示。

例

写入CSV格式(字符串(ASCII数据))时

[写入到文件中的数据]

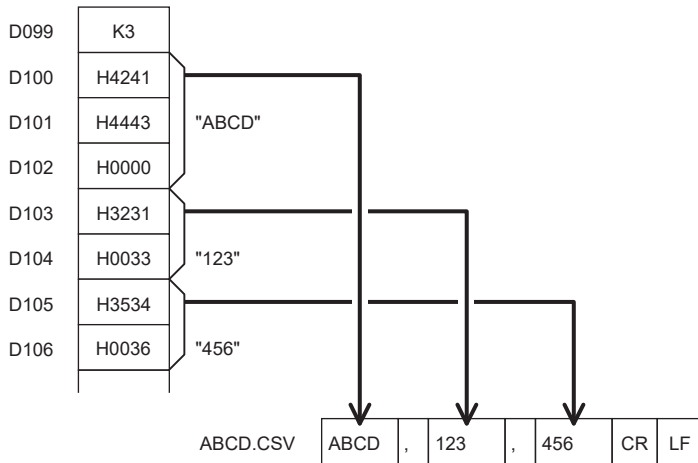


- (1) 写入的数据
- (2) 文件名
- (3) 控制数据

[控制数据]

D10	H0130
D11	H0000
D12	-
D13	-
D14	K0
D15	K0
D16	K0
D17	K2

- D10: 执行/完成类型: 字符串(ASCII数据)
- D11: 完成状态
- D12: 写入结果数据数
- D13: (未使用)
- D14、D15: 文件位置
- D16: 列数指定
- D17: 数据类型指定



- D99: 请求写入数据数
- D100~D102: 第1行第1列中写入的字符串
- D103~D104: 第1行第2列中写入的字符串
- D105~D106: 第1行第3列中写入的字符串

	A	B	C	D
1	ABCD	123	456	
2				

要点

- 1要素的字符串终端必须设置00H(NULL)。字符串为偶数字节的情况下,应在下一字中设置0000H(2字节的NULL)。
- 1要素的最大字符数为1999字符。若超过此字符数并未存储00H(NULL)的情况下,则第2000个及其以后的字符不被写入,并执行下一要素的写入。
- 1次指令执行中,最多可写入1023个要素。

- 执行/完成类型((d1)+0)中设置“0140H: 浮点实数(单精度实数)”或“0141H: 已设置浮点实数(双精度实数)”的情况下, (s3)+1及其以后中设置的值和CSV文件中写入的内容如下所示。

执行/完成类型((d1)+0)	写入数据((s3)+1及其以后)中设置的值	写入到CSV文件中的内容
0140H: 浮点实数(单精度实数)	$-2^{128} < \text{数据} \leq -2^{-126}$, 0, $2^{-126} \leq \text{数据} < 2^{128}$ 范围的值	左述的值(小数部位数0~7)以指数被写入。
	上述以外的值	不可转换, 因此写入0。
0141H: 浮点实数(双精度实数)	$-2^{1024} < \text{数据} \leq -2^{-1022}$, 0, $2^{-1022} \leq \text{数据} < 2^{1024}$ 范围的值	左述的值(小数部位数0~15)以指数被写入。
	上述以外的值	不可转换, 因此写入0。

- SD存储卡中写入CSV格式文件时的文件容量(合计字节数)的计算方法如下所示。

[合计字节数]=[最终行以外的合计字节数]+[最终行的字节数]

([各行的字节数]=[列数*2]+1+[行内各数据值的字节数的合计*3])

- *2 最终行以外将变为指定的列数。最终行的列数, 根据写入数据数指定的列数有可能不同, 因此按下述方式计算。

- 计算除去最终行的行数。(除去最终行的行数=请求写入数据数÷列数(余数舍去))
- 计算最终行的列数。(最终行的列数=请求写入数据数-(除去最终行的行数×列数))

- *3 各数据值的字节数按下表计算。

数据值的符号	各数据值的字节数	字节数的范围	例
正	位数	1~5(字指定时) 1~3(字节指定时)	• 12345→5字节 • 67→2字节
负	位数+1	2~6(字指定时) 2~4(字节指定时)	• -12345→6字节 • -67→3字节

注意事项

- SP.FWRITE指令, 请勿通过中断程序执行。通过中断程序执行的情况下, 可能导致误动作。
- 由于SP.FWRITE指令发生文件读取, 执行指令时扫描时间有可能会延长。
- SP.FWRITE指令不支持带BOM的字符代码。对于通过EXCEL创建的带BOM的CSV文件写入时, 应通过文本编辑器等将字符代码更改为无BOM。

出错

错误代码(SD0)	内容
2820H	请求写入数据数(s3)+0中指定的值超出设置范围时, 或超出(s3)+1及以后的相应软元件/标签的分配范围时。
3285H	通过驱动器指定(s1)指定的驱动器为SD存储卡以外时。
	控制数据(d1)及其以后设置的值超出设置范围时。 ☞ 609页 控制数据(d1)
3297H	无法读取在(s2)中指定的文件名字符串时。 • 指定的文件名字符串的字符数超出了范围。 • 设置了不能使用的值。 • 对所指定的文件名字符串的末尾指定了分隔符。 • 指定的文件名字符串的文件名(不包括扩展名)为0字符时。 ☞ 609页 控制数据(d1)
	(d1)+0的执行/完成类型及(d1)+7的数据类型指定变为不能指定的组合时。 (d1)的执行/完成类型、(d1)+3的写入开始位置设置及(d1)+4的文件位置指定变为不能指定的组合时。 ☞ 609页 控制数据(d1)

SP.FWRITE指令异常完成的情况下, 完成状态(d1)+1中指定的软元件中将存储错误代码。(指令发生运算错误时将不存储。)

关于完成状态(d1)+1中存储的错误代码, 请参阅下述内容。

☞ 640页 文件操作指令中发生的错误代码

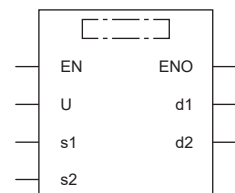
指定文件的删除

SP. FDELETE

删除SD存储卡内指定的文件或文件夹。

梯形图	ST
	ENO:=SP_FDELETE(EN, U, s1, s2, d1, d2);

FBD/LD



■执行条件

指令	执行条件
SP. FDELETE	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(U)	虚拟	—	软元件名	ANY16
(s1)	驱动器指定	2(固定)*1	字	ANY16
(d1)	存储了控制数据的起始软元件	☞ 622页 控制数据(d1)	字	ANY16_ARRAY (要素数: 2)
(s2)	存储了文件名/文件夹名的起始软元件	—	Unicode字符串	ANYSTRING_DOUBLE
(d2)	通过处理完成置为ON的位软元件	—	位	ANYBIT_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能设置SD存储卡的驱动器2。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它 (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)	○	—	○	—	—	—	—	—	—	—	—	—	—

■控制数据 (d1)

操作数: (d1)												
软元件	项目	内容	设置范围	设置方								
+0	使用用途设置区域	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%; border-bottom: 1px solid black;">b15</td> <td style="width: 30%; border-bottom: 1px solid black;">...</td> <td style="width: 20%; border-bottom: 1px solid black;">b1</td> <td style="width: 20%; border-bottom: 1px solid black;">b0</td> </tr> <tr> <td style="border-bottom: 1px solid black; text-align: center;">0</td> <td style="border-bottom: 1px solid black;"></td> <td style="border-bottom: 1px solid black; text-align: center;">1/0</td> <td style="border-bottom: 1px solid black; text-align: center;">1/0</td> </tr> </table> </div> <p>b0: 对象类型设置 将指定对象指定为文件或者是文件夹。</p> <ul style="list-style-type: none"> • 0: 文件指定 • 1: 文件夹指定 <p>b1: 空文件夹删除设置 指定删除文件夹时是否只是删除空文件夹。</p> <ul style="list-style-type: none"> • 0: 不是空文件夹时也删除 • 1: 仅在为空文件夹时删除 	b15	...	b1	b0	0		1/0	1/0	如左所示	用户
b15	...	b1	b0									
0		1/0	1/0									
+1	完成状态	<p>指令完成时存储完成状态。</p> <ul style="list-style-type: none"> • 0000H: 正常完成 • 0000H以外: 异常完成(错误代码)(640页 文件操作指令中发生的错误代码) 	—	系统								

■文件名/文件夹名 (s2)

操作数: (s2)				
软元件	项目	内容	设置范围	设置方
+0~+□	文件名/文件夹名	<p>指定文件的情况下, 对存储有所要删除文件的文件夹路径+文件名进行指定。 指定文件夹的情况下, 对所删除文件夹的文件夹路径进行指定。</p> <ul style="list-style-type: none"> • 文件名中存在扩展名时, 请勿省略。 • 应在不超过64字符的范围内设置文件名, 包括点号及扩展名。 • 应在不超过253字符的范围内设置文件夹路径+文件名(包含点号、扩展名)。 • 应在不超过244字符的范围内设置文件夹路径。(不含文件的分隔符) • 对文件名或文件夹路径应至少指定1个字符(分隔符除外)。 • 指定文件时, 请勿对字符串的末尾指定分隔符。 • 请勿对字符串的末尾或分隔符前面指定半角点号。 • 应在不超过10的范围内设置文件夹路径的分层数。 <p>■指定文件的情况下</p> <div style="text-align: center; margin-bottom: 10px;"> </div> <p>(1) 最多253字符 (2) 文件夹路径、文件的分割符使用“/”或“\”。 (3) 可以省略。省略时, (1)最多为252字符。</p> <p>■指定文件夹的情况下</p> <div style="text-align: center; margin-bottom: 10px;"> </div> <p>(4) 最多244字符 (5) 文件夹路径的分割符使用“/”或“\”。 (6) 可以省略。省略时, (4)最多为243字符。 (7) 可以省略</p>	Unicode字符串	用户

功能

- 删除 (s1) 指定驱动器的 (s2) 中指定的文件或文件夹。
- SP. FDELETE 指令执行中, SM753 (文件访问中) 将变为 ON。SM753 为 ON 时, 将不能执行 SP. FDELETE 指令。(执行的情况下将变为无处理。)
- 检测出 SP. FDELETE 指令的处理完成后, 执行扫描的 END 指令时, 处理完成 (d2) 的位软元件将自动置为 ON, 通过下一个扫描的 END 指令置为 OFF。
- SP. FDELETE 指令异常完成时, 异常完成 (d2)+1 的软元件将在与处理完成 (d2) 的软元件相同的时机变为 ON/OFF。
- 执行指令时检测出运算错误的情况下, 处理完成 (d2)、异常完成 (d2)+1 不会变为 ON。

注意事项

- 请勿通过中断程序执行 SP. FDELETE 指令。通过中断程序执行的情况下, 可能导致误动作。
- 指令执行过程中, 即使将控制器的状态更改为 RUN→STOP, 本指令也将继续进行处理。
- 指令在处理中途异常完成的情况下, 不能恢复已经删除的文件或文件夹。
- 如果删除对象文件的文件容量或文件数较大, 指令完成所花费的时间将变长。
- 请勿从其它功能访问正在以 SP. FDELETE 指令进行操作的文件。(可能导致文件破损或错误。)
- 请勿对其它功能正在访问的文件/文件夹进行操作。

出错

错误代码 (SD0)	内容
3285H	<p>通过驱动器指定 (s1) 指定的驱动器为 SD 存储卡以外时。</p> <p>无法读取 (s2) 中指定的文件名/文件夹名的字符串时。</p> <ul style="list-style-type: none"> • 文件名字符串未指定任何字符。 • 对所指定的文件名字符串指定了 254 个或其以上的字符。 • 对所指定的文件夹路径指定了 245 个或其以上的字符。 • 对所指定的文件夹路径指定了 11 或其以上的分层。 • 对所指定的文件名字符串的末尾, 或对各分隔符前面指定了半角点号。 • 指定文件时, 对所指定的文件名字符串的末尾指定了分隔符。 <p>在 (s2) 中指定的文件夹路径内指定了根文件夹正下方的系统文件夹 (\$MELPRJ\$) 时。</p>

SP. FDELETE 指令异常完成的情况下, 完成状态 (d1)+1 所指定的软元件中将存储错误代码。(指令发生运算错误时将不存储。)
关于完成状态 (d1)+1 中存储的错误代码, 请参阅下述内容。

☞ 640 页 文件操作指令中发生的错误代码

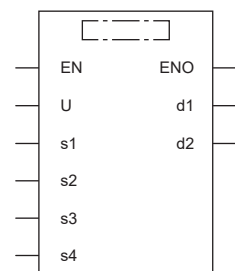
指定文件的复制

SP.FCOPY

复制SD存储卡内指定的文件或文件夹。指定文件夹时，复制整个文件夹或者指定文件夹内的所有文件、子文件夹。

梯形图	ST
	ENO:=SP_FCOPY(EN, U, s1, s2, s3, s4, d1, d2);

FBD/LD



■执行条件

指令	执行条件
SP.FCOPY	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(U)	虚拟	—	软元件名	ANY16
(d1)	存储了控制数据的起始软元件	☞ 625页 控制数据(d1)	字	ANY16_ARRAY (要素数: 2)
(s1)	复制源驱动器指定	2(固定)*1	字	ANY16
(s2)	存储了复制源文件名/文件夹名的起始软元件	—	Unicode字符串	ANYSTRING_DOUBLE
(s3)	复制目标驱动器指定	2(固定)*1	字	ANY16
(s4)	存储复制目标文件夹路径的起始软元件	—	Unicode字符串	ANYSTRING_DOUBLE
(d2)	通过处理完成置为ON的位软元件	—	位	ANYBIT_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

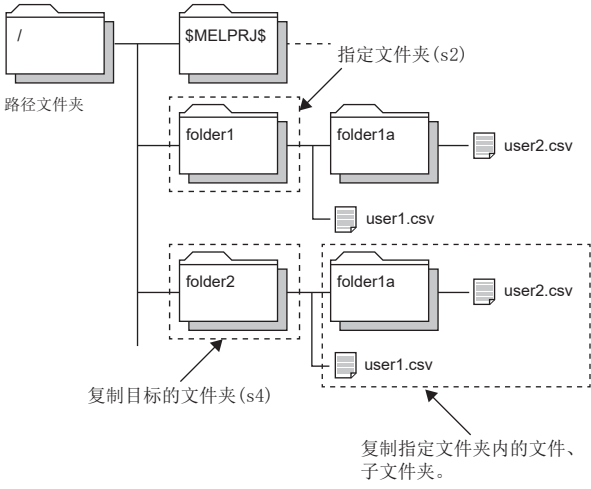
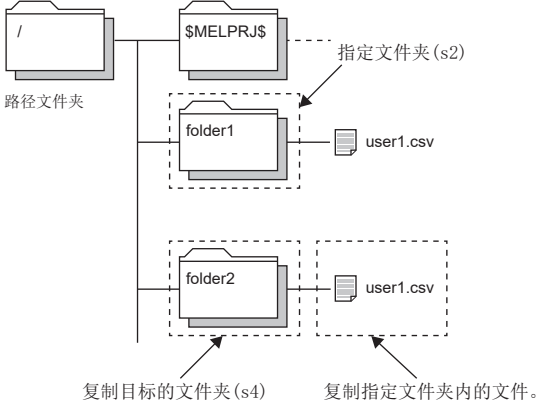

*1 只能设置SD存储卡的驱动器2。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它 (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)	○	—	○	—	—	—	—	○	—	—	—	—

■控制数据 (d1)

操作数: (d1)								
软元件	项目	内容	设置范围	设置方				
+0	使用用途设置 区域	<p>b15 ... b2 b1 b0</p> <table border="1"> <tr> <td>0</td> <td>1/0</td> <td>1/0</td> <td>1/0</td> </tr> </table> <p>■b0、b2: 对象类型设置 b0: 将指定对象指定为文件或者是文件夹。 0: 文件指定 1: 文件夹指定 b2: 指定复制对象。(b0为1(指定文件夹)时将生效。) 0: 复制整个指定文件夹 1: 复制指定文件夹内的所有文件、子文件夹 • b2为0(复制整个指定文件夹)时的动作 复制整个指定文件夹(包括文件、子文件夹)。</p> <p>路径文件夹</p> <p>指定文件夹 (s2)</p> <p>复制目标的文件夹 (s4)</p> <p>保持指定文件夹配置不变, 复制整个文件夹。</p>	0	1/0	1/0	1/0	如左所示	用户
0	1/0	1/0	1/0					

操作数: (d1)				
软件元件	项目	内容	设置范围	设置方
+0	使用用途设置区域	<p>• b2为1(复制指定文件夹内的所有文件、子文件夹)时的动作(指定文件夹内存在子文件夹的情况下) 复制指定文件夹内的文件、子文件夹。</p>  <p>• b2为1(复制指定文件夹内的所有文件、子文件夹)时的动作(指定文件夹内不存在子文件夹(只有文件)的情况下) 复制指定文件夹内的文件。</p>  <p>■b1: 覆盖设置 复制目标中存在有与复制源相同名称的文件或文件夹时对其动作进行指定。 • 0: 不覆盖 • 1: 覆盖 将b0指定为1(文件夹指定)、并将b1指定为0(不覆盖)时, 与复制目标同名的文件或文件夹将跳过复制。(不会异常完成。)</p>	如左所示	用户
+1	完成状态	<p>指令完成时存储完成状态。</p> <ul style="list-style-type: none"> • 0000H: 正常完成 • 0000H以外: 异常完成(错误代码)( 640页 文件操作指令中发生的错误代码) 	—	系统

■复制源文件名/文件夹名 (s2)

操作数: (s2)				
软元件	项目	内容	设置范围	设置方
+0~+□	文件名/文件夹名	<p>指定文件的情况下, 对存储有复制源文件的文件夹路径+文件名进行指定。 指定文件夹的情况下, 对复制源的文件夹路径进行指定。</p> <ul style="list-style-type: none"> • 文件名中存在扩展名时, 请勿省略。 • 应在不超过64字符的范围内设置文件名, 包括点号及扩展名。 • 应在不超过253字符的范围内设置文件夹路径+文件名(包含点号、扩展名)。 • 应在不超过244字符的范围内设置文件夹路径。(不含文件的分隔符) • 对文件名或文件夹路径应至少指定1个字符(分隔符除外)。 • 指定文件时, 请勿对字符串的末尾指定分隔符。 • 请勿对字符串的末尾或分隔符前面指定半角点号。 • 应在不超过10的范围内设置文件夹路径的分层数。 <p>■指定文件的情况下</p> <p>(1) 最多253字符 (2) 文件夹路径、文件的分割符使用“/”或“\”。 (3) 可以省略。省略时, (1)最多为252字符。</p> <p>■指定文件夹的情况下</p> <p>(4) 最多244字符 (5) 文件夹路径的分割符使用“/”或“\”。 (6) 可以省略。省略时, (4)最多为243字符。 (7) 可以省略</p>	Unicode字符串	用户

■复制目标文件夹路径 (s4)

操作数: (s4)				
软元件	项目	内容	设置范围	设置方
+0~+□	文件夹路径	<p>指定复制目标的文件夹路径。</p> <ul style="list-style-type: none"> • 应在不超过244字符的范围内设置文件夹路径。(不含文件的分隔符) • 对文件夹路径应至少指定1个字符(分隔符除外)。 • 请勿对字符串的末尾或分隔符前面指定半角空格。 • 请勿对字符串的末尾或分隔符前面指定半角点号。 • 应在不超过10的范围内设置文件夹路径的分层数。 <p>(1) 最多244字符 (2) 文件夹路径的分割符使用“/”或“\”。 (3) 可以省略。省略时, (1)最多为243字符。 (4) 可以省略</p>	Unicode字符串	用户

功能

- 将(s1)指定驱动器中的(s2)的指定文件或文件夹复制到(s3)指定驱动器中的(s4)的指定文件夹。(s2)中指定了文件夹时,复制目标中存在相同名称的文件及子文件夹的情况下,即使(d1)的位1的值(覆盖设置)为0,也不会变为异常完成。(跳过复制。)
- (s4)的指定复制目标文件夹不存在的情况下,将自动创建文件夹。
- SP.FCOPY指令执行中,SM753(文件访问中)将变为ON。SM753为ON时,将不能执行SP.FCOPY指令。(执行的情况下将变为无处理。)
- 检测出SP.FCOPY指令的处理完成后,执行扫描的END指令时,处理完成(d2)的位软元件将自动置为ON,通过下一个扫描的END指令置为OFF。
- SP.FCOPY指令异常完成时,异常完成(d2)+1的软元件将在与处理完成(d2)的软元件相同的时机变为ON/OFF。
- 执行指令时检测出运算错误的情况下,处理完成(d2)、异常完成(d2)+1不会变为ON。

注意事项

- 请勿通过中断程序执行SP.FCOPY指令。通过中断程序执行的情况下,可能导致误动作。
- 指令执行过程中,即使将控制器的状态更改为RUN→STOP,本指令也将继续进行处理。
- 复制后的文件夹路径+文件名以及文件夹路径不可超过字符数限制。(可能导致无法访问文件或错误。)
- 指令在处理中途异常完成的情况下,文件/文件夹可能以处于处理中的状态留存。
- 如果复制对象文件的文件容量或文件数较大,指令完成所花费的时间将变长。
- 请勿从其它功能访问正在以SP.FCOPY指令进行操作的文件。(可能导致文件破损或错误。)
- 请勿对其它功能正在访问的文件/文件夹进行操作。

出错

错误代码(SD0)	内容
3285H	通过驱动器指定(s1)、(s3)指定的驱动器不是SD存储卡时。
	无法读取(s2)中指定的文件名/文件夹名的字符串时。 <ul style="list-style-type: none">• 文件名字符串未指定任何字符。• 对所指定的文件名字符串指定了254个或以上的字符。• 对所指定的文件夹路径指定了245个或以上的字符。• 对所指定的文件夹路径指定了11或以上的分层。• 指定文件时,对所指定的文件名字符串的末尾指定了分隔符。• 对所指定的文件名字符串的末尾,或对各分隔符前面指定了半角点号。
	无法读取(s4)中指定的文件夹路径的字符串时。 <ul style="list-style-type: none">• 文件夹路径字符串未指定任何字符。• 对所指定的文件夹路径字符串指定了245个或以上的字符。• 对所指定的文件夹路径字符串指定了11或以上的分层。• 对所指定的文件名字符串的末尾,或对各分隔符前面指定了半角空格。• 对所指定的文件夹路径的末尾,或对各分隔符前面指定了半角点号。
	在(s2)、(s4)中指定的文件夹路径内指定了根文件夹正下方的系统文件夹(\$MELPRJ\$)时。

SP.FCOPY指令异常完成的情况下,完成状态(d1)+1所指定的软元件中将存储错误代码。(指令发生运算错误时将不存储。)

关于完成状态(d1)+1中存储的错误代码,请参阅下述内容。

☞ 640页 文件操作指令中发生的错误代码

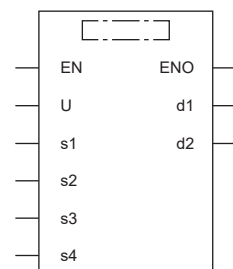
指定文件的移动

SP. FMOVE

移动SD存储卡内指定的文件或文件夹。指定文件夹时，移动整个文件夹或者指定文件夹内的所有文件、子文件夹。

梯形图	ST
	<pre>ENO:=SP_FMOVE(EN, U, s1, s2, s3, s4, d1, d2);</pre>

FBD/LD



■执行条件

指令	执行条件
SP. FMOVE	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(U)	虚拟	—	软元件名	ANY16
(d1)	存储了控制数据的起始软元件	☞ 630页 控制数据 (d1)	字	ANY16_ARRAY (要素数: 2)
(s1)	移动源驱动器指定	2(固定)*1	字	ANY16
(s2)	存储了移动源文件名/文件夹名的起始软元件	—	Unicode字符串	ANYSTRING_DOUBLE
(s3)	移动目标驱动器指定	2(固定)*1	字	ANY16
(s4)	存储移动目标文件夹路径的软元件	—	Unicode字符串	ANYSTRING_DOUBLE
(d2)	通过处理完成置为ON的位软元件	—	位	ANYBIT_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能设置SD存储卡的驱动器2。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它(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)	○	—	○	—	—	—	—	○	—	—	—	—

■控制数据(d1)

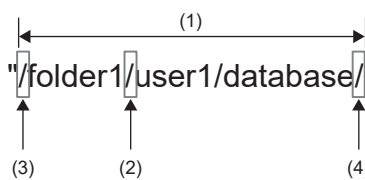
操作数：(d1)																
软元件	项目	内容	设置范围	设置方												
+0	使用用途设置区域	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%; text-align: right;">b15</td> <td style="text-align: center;">...</td> <td style="width: 10%;"></td> <td style="text-align: left;">b2</td> <td style="text-align: left;">b1</td> <td style="text-align: left;">b0</td> </tr> <tr> <td style="text-align: center;">0</td> <td></td> <td></td> <td style="text-align: center;">1/0</td> <td style="text-align: center;">1/0</td> <td style="text-align: center;">1/0</td> </tr> </table> </div> <p>■b0、b2：对象类型设置</p> <p>b0：将指定对象指定为文件或者是文件夹。 0：文件指定 1：文件夹指定</p> <p>b2：指定移动对象。(b0为1(指定文件夹)时有效。) 0：移动整个指定文件夹 1：移动指定文件夹内的所有文件、子文件夹</p> <ul style="list-style-type: none"> • b2为0(移动整个指定文件夹)时的动作 移动整个指定文件夹(包括文件、子文件夹)。 	b15	...		b2	b1	b0	0			1/0	1/0	1/0	如左所示	用户
b15	...		b2	b1	b0											
0			1/0	1/0	1/0											

操作数: (d1)				
软件	项目	内容	设置范围	设置方
+0	使用用途设置区域	<p>• b2为1 (移动指定文件夹内的所有文件、子文件夹)时的动作 (指定文件夹内存在子文件夹的情况下) 移动指定文件夹内的文件、子文件夹。</p> <p>• b2为1 (移动指定文件夹内的所有文件、子文件夹)时的动作 (指定文件夹内不存在子文件夹 (只有文件)的情况下) 移动指定文件夹内的文件。</p> <p>■b1: 覆盖设置 移动目标中存在有与移动源相同名称的文件或文件夹时对其动作进行指定。 • 0: 不覆盖 • 1: 覆盖</p>	如左所示	用户
+1	完成状态	<p>指令完成时存储完成状态。</p> <ul style="list-style-type: none"> • 0000H: 正常完成 • 0000H以外: 异常完成 (错误代码) (☞ 640页 文件操作指令中发生的错误代码) 	—	系统

■文件名/文件夹名(s2)

操作数: (s2)				
软元件	项目	内容	设置范围	设置方
+0~+□	文件名/文件夹名	<p>指定文件的情况下, 对存储有移动源文件的文件夹路径+文件名进行指定。 指定文件夹的情况下, 对移动源文件夹的文件夹路径进行指定。</p> <ul style="list-style-type: none"> • 文件名中存在扩展名时, 请勿省略。 • 应在不超过64字符的范围内设置文件名, 包括点号及扩展名。 • 应在不超过253字符的范围内设置文件夹路径+文件名(包含点号、扩展名)。 • 应在不超过244字符的范围内设置文件夹路径。(不含文件的分隔符) • 对文件名或文件夹路径应至少指定1个字符(分隔符除外)。 • 指定文件时, 请勿对字符串的末尾指定分隔符。 • 请勿对字符串的末尾或分隔符前面指定半角点号。 • 应在不超过10的范围内设置文件夹路径的分层数。 <p>■指定文件的情况下</p>  <p>(1) 最多253字符 (2) 文件夹路径、文件的分割符使用“/”或“\”。 (3) 可以省略。省略时, (1)最多为252字符。</p> <p>■指定文件夹的情况下</p>  <p>(4) 最多244字符 (5) 文件夹路径的分割符使用“/”或“\”。 (6) 可以省略。省略时, (4)最多为243字符。 (7) 可以省略</p>	Unicode字符串	用户

■移动目标文件夹路径(s4)

操作数: (s4)				
软元件	项目	内容	设置范围	设置方
+0~+□	文件夹路径	<p>指定移动目标的文件夹路径。</p> <ul style="list-style-type: none"> • 应在不超过244字符的范围内设置文件夹路径。(不含文件的分隔符) • 对文件夹路径应至少指定1个字符(分隔符除外)。 • 请勿对字符串的末尾或分隔符前面指定半角空格。 • 请勿对字符串的末尾或分隔符前面指定半角点号。 • 应在不超过10的范围内设置文件夹路径的分层数。  <p>(1) 最多244字符 (2) 文件夹路径、文件的分割符使用“/”或“\”。 (3) 可以省略。省略时, (1)最多为243字符。 (4) 可以省略</p>	Unicode字符串	用户

功能

- 将(s1)的指定驱动器中的(s2)的指定文件或文件夹移动到(s3)的指定驱动器中的(s4)的指定文件夹。(s4)的指定移动目标文件夹不存在的情况下,将自动创建文件夹。
- SP.FMOVE指令执行中,SM753(文件访问中)将变为ON。SM753为ON时,将不能执行SP.FMOVE指令。(执行的情况下将变为无处理。)
- 检测出SP.FMOVE指令的处理完成后,执行扫描的END指令时,处理完成(d2)的位软元件将自动置为ON,通过下一个扫描的END指令置为OFF。
- SP.FMOVE指令异常完成时,异常完成(d2)+1的软元件将在与处理完成(d2)的软元件相同的时机变为ON/OFF。
- 执行指令时检测出运算错误的情况下,处理完成(d2)、异常完成(d2)+1不会变为ON。

注意事项

- 请勿通过中断程序执行SP.FMOVE指令。通过中断程序执行的情况下,可能导致误动作。
- 指令执行过程中,即使将控制器的状态更改为RUN→STOP,本指令也将继续进行处理。
- 移动后的文件夹路径+文件名以及文件夹路径不可超过字符数限制。(可能导致文件破损或错误。)
- 发生覆盖时,指令完成所花费的时间可能有所延长。
- 发生覆盖时,应确保留有与操作对象文件同样大小的空余容量。
- 指令在处理中途异常完成的情况下,文件/文件夹可能以处于处理中的状态留存。
- 如果发生覆盖,且移动对象文件的文件容量或文件数较大,指令完成所花费的时间将变长。
- 请勿从其它功能访问正在以SP.FMOVE指令进行操作的文件。(可能导致无法访问文件或错误。)
- 请勿对其它功能正在访问的文件/文件夹进行操作。

出错

错误代码(SD0)	内容
3285H	<p>通过驱动器指定(s1)指定的驱动器为SD存储卡以外时。</p> <p>无法读取(s2)中指定的文件名/文件夹名的字符串时。</p> <ul style="list-style-type: none"> • 文件名字符串未指定任何字符。 • 对所指定的文件名字符串指定了254个或以上的字符。 • 对所指定的文件夹路径指定了245个或以上的字符。 • 对所指定的文件夹路径指定了11或以上的分层。 • 指定文件时,对所指定的文件名字符串的末尾指定了分隔符。 • 对所指定的文件名字符串的末尾,或对各分隔符前面指定了半角点号。 <p>无法读取(s4)中指定的文件夹路径的字符串时。</p> <ul style="list-style-type: none"> • 文件夹路径字符串未指定任何字符。 • 对所指定的文件夹路径字符串指定了245个或以上的字符。 • 对所指定的文件夹路径字符串指定了11或以上的分层。 • 对所指定的文件名字符串的末尾,或对各分隔符前面指定了半角空格。 • 对所指定的文件夹路径的末尾,或对各分隔符前面指定了半角点号。 <p>在(s2)、(s4)中指定的文件夹路径内指定了根文件夹正下方的系统文件夹(\$MELPRJ\$)时。</p>

SP.FMOVE指令异常完成的情况下,完成状态(d1)+1所指定的软元件中将存储错误代码。(指令发生运算错误时将不存储。)

关于完成状态(d1)+1中存储的错误代码,请参阅下述内容。

☞ 640页 文件操作指令中发生的错误代码

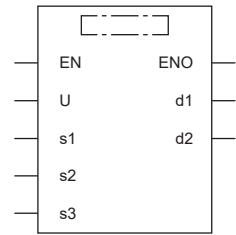
指定文件名的更改

SP. FRENAME

更改SD存储卡内指定的文件名或文件夹名。

梯形图	ST
	<pre>ENO:=SP_FRENAME (EN, U, s1, s2, s3, d1, d2);</pre>

FBD/LD



■执行条件

指令	执行条件
SP. FRENAME	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(U)	虚拟	—	软元件名	ANY16
(s1)	驱动器指定	2 (固定)*1	字	ANY16
(d1)	存储了控制数据的起始软元件	☞ 635页 控制数据 (d1)	字	ANY16_ARRAY (要素数: 2)
(s2)	存储了要更改的文件名/文件夹名的起始软元件	—	Unicode字符串	ANYSTRING_DOUBLE
(s3)	存储了更改后的文件名/文件夹名的起始软元件	—	Unicode字符串	ANYSTRING_DOUBLE
(d2)	通过处理完成置为ON的位软元件	—	位	ANYBIT_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能设置SD存储卡的驱动器2。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它 (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)	○	—	○	—	—	—	—	○	—	—	—	—	—	—

■控制数据 (d1)

操作数: (d1)				
软元件	项目	内容	设置范围	设置方
+0	使用用途设置区域	<div style="display: flex; justify-content: space-between; align-items: center;"> b15 ... b0 </div> <div style="border: 1px solid black; padding: 2px; width: 100%; text-align: center;"> 0 1/0 </div> <p>b0: 对象类型设置 将指定对象指定为文件或者是文件夹。</p> <ul style="list-style-type: none"> • 0: 文件指定 • 1: 文件夹指定 	如左所示	用户
+1	完成状态	指令完成时存储完成状态。 <ul style="list-style-type: none"> • 0000H: 正常完成 • 0000H以外: 异常完成(错误代码) (☞ 640页 文件操作指令中发生的错误代码) 	—	系统

■更改的文件名/文件夹名 (s2)

操作数: (s2)				
软元件	项目	内容	设置范围	设置方
+0~+□	文件名/文件夹名	<p>指定文件的情况下, 对存储了要更改名称的文件的文件夹路径+文件名进行指定。 指定文件夹的情况下, 对要更改名称的文件夹的文件夹路径进行指定。</p> <ul style="list-style-type: none"> • 文件名中存在扩展名时, 请勿省略。 • 应在不超过64字符的范围内设置文件名, 包括点号及扩展名。 • 应在不超过253字符的范围内设置文件夹路径+文件名(包含点号、扩展名)。 • 应在不超过244字符的范围内设置文件夹路径。(不含文件的分隔符) • 对文件名或文件夹路径应至少指定1个字符(分隔符除外)。 • 指定文件时, 请勿对字符串的末尾指定分隔符。 • 请勿对字符串的末尾或分隔符前面指定半角点号。 • 应在不超过10的范围内设置文件夹路径的分层数。 <p>■指定文件的情况下</p> <div style="text-align: center;"> <p>"/folder1/user1/user1.csv"</p> <p>(3) (2)</p> </div> <p>(1) 最多253字符 (2) 文件夹路径、文件的分割符使用“/”或“\”。 (3) 可以省略。省略时, (1) 最多为252字符。</p> <p>■指定文件夹的情况下</p> <div style="text-align: center;"> <p>"/folder1/user1/subfolder1/"</p> <p>(6) (5) (7)</p> </div> <p>(4) 最多244字符 (5) 文件夹路径的分割符使用“/”或“\”。 (6) 可以省略。省略时, (4) 最多为243字符。 (7) 可以省略</p>	Unicode字符串	用户

■更改后的文件名/文件夹名 (s3)

操作数: (s3)				
软元件	项目	内容	设置范围	设置方
+0~+□	文件名/文件夹名	指定更改后的文件名、文件夹名的字符串。 • 请勿指定文件夹路径。 • 应在不超过64字符的范围内设置文件名(包括点号及扩展名)。 • 应在不超过243字符的范围内设置文件夹名。 • 对文件名或文件夹名应至少指定1个字符。 • 请勿指定分隔符。 • 请勿对字符串的末尾指定半角点号。	Unicode字符串	用户

功能

- 将(s1)的指定驱动器中的(s2)的指定文件名或文件夹名更改为(s3)的指定文件名或文件夹名。(s3)的指定文件名或文件夹名已经存在的情况下, 将变为异常完成。
- SP.FRENAME指令执行中, SM753(文件访问中)将变为ON。SM753为ON时, 将不能执行SP.FRENAME指令。(执行的情况下将变为无处理。)
- 检测出SP.FRENAME指令的处理完成后, 执行扫描的END指令时, 处理完成(d2)的位软元件将自动置为ON, 通过下一个扫描的END指令置为OFF。
- SP.FRENAME指令异常完成时, 异常完成(d2)+1的软元件将在与处理完成(d2)的软元件相同的时机变为ON/OFF。
- 执行指令时检测出运算错误的情况下, 处理完成(d2)、异常完成(d2)+1不会变为ON。

注意事项

- 请勿通过中断程序执行SP.FRENAME指令。通过中断程序执行的情况下, 可能导致误动作。
- 指令执行过程中, 即使将控制器的状态更改为RUN→STOP, 本指令也将继续进行处理。
- 更改后的文件夹路径+文件名以及文件夹路径不可超过字符数限制。(可能导致无法访问文件或错误。)
- 请勿从其它功能访问正在以SP.FRENAME指令进行操作的文件。(可能导致文件破损或错误。)
- 请勿对其它功能正在访问的文件/文件夹进行操作。

出错

错误代码(SD0)	内容
3285H	通过驱动器指定(s1)指定的驱动器为SD存储卡以外时。 无法读取(s2)中指定的文件名/文件夹名的字符串时。 • 文件名字符串未指定任何字符。 • 对所指定的文件名字符串指定了254个或其以上的字符。 • 对所指定的文件夹路径指定了245个或其以上的字符。 • 对所指定的文件夹路径指定了11或其以上的分层。 • 指定文件时, 对所指定的文件名字符串的末尾指定了分隔符。 • 对所指定的文件名字符串的末尾, 或对各分隔符前面指定了半角点号。 无法读取(s3)中指定的文件名/文件夹名的字符串时。 • 文件名/文件夹名的字符串未指定任何字符。 • 对所指定的文件名字符串指定了253个或其以上的字符。 • 对所指定的文件夹名字符串指定了244个或其以上的字符。 • 对所指定的文件名字符串指定了分隔符。 • 对所指定的文件名字符串的末尾指定了半角点号。 在(s2)中指定的文件夹路径内指定了根文件夹正下方的系统文件夹(\$MELPRJ\$)时。 将(s3)的指定文件名/文件夹名指定为\$MELPRJ\$时。

SP.FRENAME指令异常完成的情况下, 完成状态(d1)+1中指定的软元件中将存储错误代码。(指令发生运算错误时将不存储。)
关于完成状态(d1)+1中存储的错误代码, 请参阅下述内容。

☞ 640页 文件操作指令中发生的错误代码

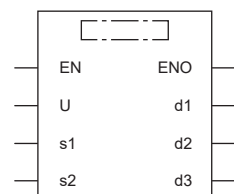
指定文件状态的获取

SP. FSTATUS

获取SD存储卡内指定的文件或文件夹。

梯形图	ST
	ENO:=SP_FSTATUS (EN, U, s1, s2, d1, d2, d3);

FBD/LD



■执行条件

指令	执行条件
SP. FSTATUS	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(U)	虚拟	—	软元件名	ANY16
(s1)	驱动器指定	2 (固定)*1	字	ANY16
(d1)	存储了控制数据的起始软元件	☞ 638页 控制数据 (d1)	字	ANY16_ARRAY (要素数: 2)
(s2)	存储了文件名/文件夹名的起始软元件	—	Unicode字符串	ANYSTRING_DOUBLE
(d2)	存储文件状态的起始软元件	—	字	ANY16_ARRAY (要素数: 10)
(d3)	通过处理完成置为ON的位软元件	—	位	ANYBIT_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能设置SD存储卡的驱动器2。

■可用软元件

操作数	位		字			双字		间接指定	常数				其它 (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)	○	—	○	—	—	—	○	—	—	—	—	—	—

■控制数据 (d1)

操作数: (d1)				
软元件	项目	内容	设置范围	设置方
+0	使用用途设置区域	<div style="display: flex; align-items: center; gap: 10px;"> <div style="border: 1px solid black; padding: 2px;">b15</div> <div style="flex-grow: 1; border-bottom: 1px solid black; text-align: center;">...</div> <div style="border: 1px solid black; padding: 2px;">b0</div> </div> <div style="border: 1px solid black; width: 100%; height: 15px; margin-top: 2px; text-align: center;">0</div> <div style="border: 1px solid black; width: 15px; height: 15px; margin-top: 2px; text-align: center; line-height: 15px;">1/0</div> <p>b0: 对象类型设置 将指定对象指定为文件或者是文件夹。</p> <ul style="list-style-type: none"> • 0: 文件指定 • 1: 文件夹指定 	如左所示	用户
+1	完成状态	指令完成时存储完成状态。 <ul style="list-style-type: none"> • 0000H: 正常完成 • 0000H以外: 异常完成(错误代码) (☞ 640页 文件操作指令中发生的错误代码) 	—	系统

■文件名/文件夹名 (s2)

操作数: (s2)				
软元件	项目	内容	设置范围	设置方
+0~+□	文件名/文件夹名	指定文件的情况下, 对存储有要获取状态的文件的文件夹路径+文件名进行指定。 指定文件夹的情况下, 对要获取状态的文件夹的文件夹路径进行指定。 <ul style="list-style-type: none"> • 文件名中存在扩展名时, 请勿省略。 • 应在不超过64字符的范围内设置文件名, 包括点号及扩展名。 • 应在不超过253字符的范围内设置文件夹路径+文件名(包含点号、扩展名)。 • 应在不超过244字符的范围内设置文件夹路径。(不含文件的分隔符) • 对文件名或文件夹路径应至少指定1个字符(分隔符除外)。 • 指定文件时, 请勿对字符串的末尾指定分隔符。 • 请勿对字符串的末尾或分隔符前面指定半角点号。 • 应在不超过10的范围内设置文件夹路径的分层数。 <p>■指定文件的情况下</p> <div style="text-align: center;"> </div> <p>(1) 最多253字符 (2) 文件夹路径、文件的分割符使用“/”或“\”。 (3) 可以省略。省略时, (1) 最多为252字符。</p> <p>■指定文件夹的情况下</p> <div style="text-align: center;"> </div> <p>(4) 最多244字符 (5) 文件夹路径的分割符使用“/”或“\”。 (6) 可以省略。省略时, (4) 最多为243字符。 (7) 可以省略</p>	Unicode字符串	用户

■文件状态 (d2)

操作数: (d2)			
软元件	项目	范围	设置方
+0	文件属性 bit0: 为只读文件时ON bit1: 为隐藏文件时ON bit2: 为系统文件时ON bit3: 保留(固定为0) bit4: 为目录时ON bit5: 为存档时ON bit6~15: 保留(固定为0)	如左所示	系统
+1	保留	0	
+2~+3	文件容量(字节单位)	0~4294967294*1	
+4	最终更新时间: 年	0、1980~2079*1	
+5	最终更新时间: 月	0~12	
+6	最终更新时间: 日	0~31	
+7	最终更新时间: 时	0~23	
+8	最终更新时间: 分	0~59	
+9	最终更新时间: 秒		

*1 在控制器以外的环境(OS)下访问文件/文件夹的情况下, 获取值的范围取决于该环境(OS)。

功能

- 获取(s1)的指定驱动器中的(s2)的指定文件或文件夹的状态, 并存储至(d2)及其以后的软元件。(s2)指定为文件夹的情况下, 将在(d2)+2~(d2)+3中存储0。
- SP.FSTATUS指令执行中, SM753(文件访问中)将变为ON。SM753为ON时, 将不能执行SP.FSTATUS指令。(执行的情况下将变为无处理。)
- 检测出SP.FSTATUS指令的处理完成后, 执行扫描的END指令时, 处理完成(d3)的位软元件将自动置为ON, 通过下一个扫描的END指令置为OFF。
- SP.FSTATUS指令异常完成时, 异常完成(d3)+1的软元件将在与处理完成(d3)的软元件相同的时机变为ON/OFF。
- 执行指令时检测出运算错误的情况下, 处理完成(d3)、异常完成(d3)+1不会变为ON。

注意事项

- 请勿通过中断程序执行SP.FSTATUS指令。通过中断程序执行的情况下, 可能导致误动作。
- 指令执行过程中, 即使将控制器的状态更改为RUN→STOP, 本指令也将继续进行处理。

出错

错误代码(SD0)	内容
3285H	通过驱动器指定(s1)指定的驱动器为SD存储卡以外时。 无法读取(s2)中指定的文件名/文件夹名的字符串时。 <ul style="list-style-type: none"> • 文件名字符串未指定任何字符。 • 对所指定的文件名字符串指定了254个或其以上的字符。 • 对所指定的文件夹路径指定了245个或其以上的字符。 • 对所指定的文件夹路径指定了11或其以上的分层。 • 指定文件时, 对所指定的文件名字符串的末尾指定了分隔符。 • 对所指定的文件名字符串的末尾, 或对各分隔符前面指定了半角点号。

SP.FSTATUS指令异常完成的情况下, 完成状态(d1)+1中指定的软元件中将存储错误代码。(指令发生运算错误时将不存储。)
关于完成状态(d1)+1中存储的错误代码, 请参阅下述内容。

☞ 640页 文件操作指令中发生的错误代码

文件操作指令中发生的错误代码

文件操作指令的完成状态中存储的错误代码一览如下所示。

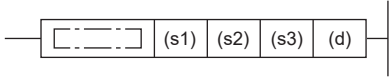
错误代码	错误内容	处理方法
8000H	SM606 (SD存储卡强制使用停止指示) 为0N。	SM606为0N时, 将SM606置为0FF, 并进行SD存储卡强制使用停止解除。
	未安装SD存储卡。	安装SD存储卡。
	SD存储卡未加载。	对SD存储卡进行加载。
8001H	对SD存储卡的访问失败。	<ul style="list-style-type: none"> • 确认文件名字符串的指定是否有误。 • 确认其它功能是否正在访问文件。 • 采取抗噪声措施。 • 将控制器复位后, 置为RUN。再次显示相同错误的情况下, 可能是SD存储卡的硬件异常。请向当地三菱电机代理店咨询。
	SD存储卡处于写保护状态。	将SD存储卡的写保护开关置为无效(允许写入)。
	文件属性处于只读状态。	解除只读的设置。
	SD存储卡中存储的文件容量超过了SD存储卡空余容量或最大容量。	<ul style="list-style-type: none"> • 删除SD存储卡内不要的文件, 确保留有空余容量。 • 将文件容量修改为不超过最大容量。
8002H	指定的文件或文件夹不存在。	<ul style="list-style-type: none"> • 确认指定的文件或文件夹是否存在。 • 确认指定的文件夹路径是否存在。
	SD存储卡的空余容量不足。	删除SD存储卡内不要的文件, 确保留有空余容量。
8003H	从文件读取的数据数的合计超过(d1)+3(最大读取数据数)。	调整(d1)+2(请求读取数据数)或(d1)+3(最大读取数据数)。
8004H	删除对象文件夹不是空文件夹。	确认控制数据的设置。
8005H	存在相同名称的文件或文件夹。	更改文件名或文件夹名。
8006H	<ul style="list-style-type: none"> • 指令执行后, 存在文件夹路径超过了244字符的文件夹。 • 指令执行后, 存在文件路径超过了253字符的文件。 	设置指令执行后的文件夹路径和文件路径, 使其不超过最多字符数。
	向复制目标或移动目标指定了下述文件夹/子文件夹。 <ul style="list-style-type: none"> • 同一个复制源的文件夹/子文件夹 • 同一个移动源的文件夹/子文件夹 	更改复制目标或移动目标的路径。

14.5 数据控制指令

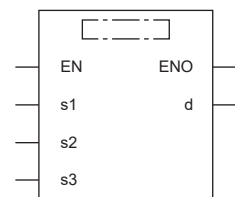
BIN16位数据上下限限位控制

LIMIT(P) (_U)

根据指定的输入值 (BIN16位值) 是否在上下限限位值的范围内, 对输出值进行控制。

梯形图	ST*1	
	ENO:=LIMITP(EN, s1, s2, s3, d);	ENO:=LIMIT_U(EN, s1, s2, s3, d);



FBD/LD*1



*1 在ST、FBD/LD中不支持LIMIT指令、LIMIT_U指令。应使用通用函数的LIMIT。

☞ 1308页 LIMIT(_E)

■执行条件

指令	执行条件
LIMIT LIMIT_U	
LIMITP LIMITP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	LIMIT(P)	下限限位值(最小输出极限值)	-32768~32767	带符号BIN16位	ANY16_S
	LIMIT(P)_U		0~65535	无符号BIN16位	ANY16_U
(s2)	LIMIT(P)	上限限位值(最大输出极限值)	-32768~32767	带符号BIN16位	ANY16_S
	LIMIT(P)_U		0~65535	无符号BIN16位	ANY16_U
(s3)	LIMIT(P)	通过上下限位控制进行控制的输入值	-32768~32767	带符号BIN16位	ANY16_S
	LIMIT(P)_U		0~65535	无符号BIN16位	ANY16_U
(d)	LIMIT(P)	存储通过上下限位控制进行控制的输出值的软元件	—	带符号BIN16位	ANY16_S
	LIMIT(P)_U			无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

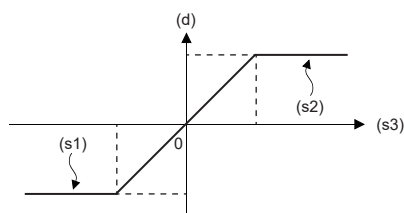
■可用软元件

操作数	位		字		双字			间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- (s3)中指定的输入值(BIN16位值)根据(s1)、(s2)中指定的上下限位值的范围,对(d)中指定的软元件中存储的输出值进行控制。输出值按下述方式被控制。

条件	输出值中存储的值
下限限位值(s1)>输入值(s3)时	下限限位值(s1)
上限限位值(s2)<输入值(s3)时	上限限位值(s2)
下限限位值(s1)≤输入值(s3)≤上限限位值(s2)时	输入值(s3)



- 仅进行上限限位值控制的情况下,在(s1)中指定的下限限位值中设置设置数据范围的最小值。
- 仅进行下限限位值控制的情况下,在(s2)中指定的上限限位值中设置设置数据范围的最大值。

出错

错误代码(SD0)	内容
3285H	(s1)中指定的下限限位值大于(s2)中指定的上限限位值时。

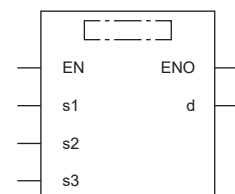
BIN32位数据上下限限位控制

DLIMIT(P) (_U)

根据指定的输入值 (BIN32位值) 是否在上下限限位值的范围内, 对输出值进行控制。

梯形图	ST*1	
	ENO:=DLIMITP (EN, s1, s2, s3, d);	ENO:=DLIMITP_U (EN, s1, s2, s3, d);

FBD/LD*1



*1 在ST、FBD/LD中不支持DLIMIT指令、DLIMIT_U指令。应使用通用函数的LIMIT。
 1308页 LIMIT(_E)

■执行条件

指令	执行条件
DLIMIT DLIMIT_U	
DLIMITP DLIMITP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	DLIMIT(P)	存储下限限位值(最小输出极限值)的软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	DLIMIT(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(s2)	DLIMIT(P)	存储上限限位值(最大输出极限值)的软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	DLIMIT(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(s3)	DLIMIT(P)	存储通过上下限限位控制进行控制的输入值的软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	DLIMIT(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(d)	DLIMIT(P)	存储通过上下限限位控制进行控制的输出值的起始软元件	—	带符号BIN32位	ANY32_S
	DLIMIT(P)_U			无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

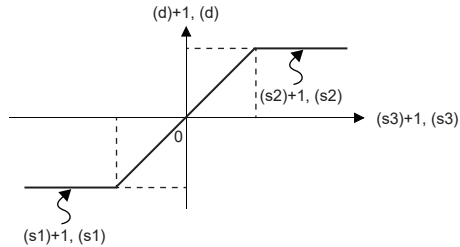
■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—

功能

- (s3)中指定的输入值(BIN32位值)根据(s1)、(s2)中指定的上下限限位值的范围,对(d)中指定的软件件中存储的输出值进行控制。输出值按下述方式被控制。

条件	输出值中存储的值
下限限位值((s1), (s1)+1) > 输入值((s3), (s3)+1) 时	下限限位值((s1), (s1)+1)
上限限位值((s2), (s2)+1) < 输入值((s3), (s3)+1) 时	上限限位值((s2), (s2)+1)
下限限位值((s1), (s1)+1) ≤ 输入值((s3), (s3)+1) ≤ 上限限位值((s2), (s2)+1) 时	输入值((s3), (s3)+1)



- 仅进行上限限位值控制的情况下,在(s1)中指定的下限限位值中设置设置数据范围的最小值。
- 仅进行下限限位值控制的情况下,在(s2)中指定的上限限位值中设置设置数据范围的最大值。

出错

错误代码(SD0)	内容
3285H	(s1)中指定的下限限位值大于(s2)中指定的上限限位值时。

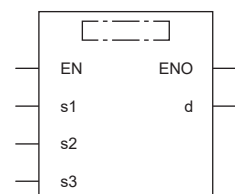
BIN16位数据死区控制

BAND(P) (_U)

根据指定的输入值 (BIN16位值) 是否在死区的上下限范围内, 对输出值进行控制。

梯形图	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



■执行条件

指令	执行条件
BAND BAND_U	
BANDP BANDP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	BAND(P)	死区(无输出区域)的下限值	-32768~32767	带符号BIN16位	ANY16_S
	BAND(P)_U		0~65535	无符号BIN16位	ANY16_U
(s2)	BAND(P)	死区(无输出区域)的上限值	-32768~32767	带符号BIN16位	ANY16_S
	BAND(P)_U		0~65535	无符号BIN16位	ANY16_U
(s3)	BAND(P)	通过死区控制进行控制的输入值	-32768~32767	带符号BIN16位	ANY16_S
	BAND(P)_U		0~65535	无符号BIN16位	ANY16_U
(d)	BAND(P)	存储通过死区控制进行控制的输出值的软元件	—	带符号BIN16位	ANY16_S
	BAND(P)_U			无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

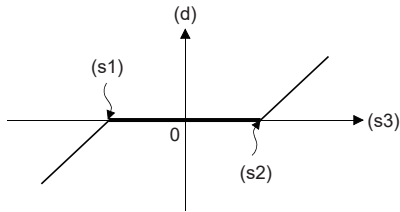
■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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	
(s1)	○	○	○	○	○	—	—	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	—	—	—
(s3)	○	○	○	○	○	—	—	○	—	—	—
(d)	○	○	○	○	○	—	—	—	—	—	—

功能

- (s3)中指定的输入值(BIN16位值)根据(s1)、(s2)中指定的死区的上下限范围,对(d)中指定的软元件中存储的输出值进行控制。输出值按下述方式被控制。

条件	输出值中存储的值
死区下限值(s1)>输入值(s3)时	输入值(s3)-死区下限值(s1)
死区上限值(s2)<输入值(s3)时	输入值(s3)-死区上限值(s2)
死区下限值(s1)≤输入值(s3)≤死区上限值(s2)时	0



- BAND(P)指令的情况下,运算结果超出-32768~32767的范围时,其情况如下例所示。

例

(s1)为10、(s3)为-32768时,则输出值=-32768-10=8000H-000AH=7FF6H=32758。

- BAND(P)_U指令的情况下,运算结果超出0~65535的范围时,其情况如下例所示。

例

(s1)为100、(s3)为50时,则输出值=50-100=0032H-0064H=FFCEH=65486。

出错

错误代码(SD0)	内容
3285H	(s1)中指定的下限值大于(s2)中指定的上限值时。

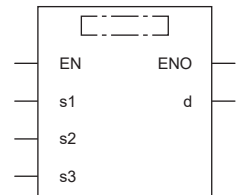
BIN32位数据死区控制

DBAND(P) (_U)

根据指定的输入值 (BIN32位值) 是否在死区的上下限范围内, 对输出值进行控制。

梯形图	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



■执行条件

指令	执行条件
DBAND DBAND_U	
DBANDP DBANDP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	DBAND (P)	存储死区(无输出区域)的下限值的软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	DBAND (P)_U		0~4294967295	无符号BIN32位	ANY32_U
(s2)	DBAND (P)	存储死区(无输出区域)的上限值的软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	DBAND (P)_U		0~4294967295	无符号BIN32位	ANY32_U
(s3)	DBAND (P)	存储通过死区控制进行控制的输入值的软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	DBAND (P)_U		0~4294967295	无符号BIN32位	ANY32_U
(d)	DBAND (P)	存储通过死区控制进行控制的输出值的起始软元件	—	带符号BIN32位	ANY32_S
	DBAND (P)_U			无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

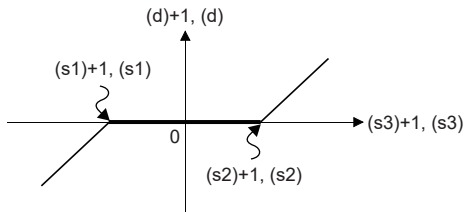
■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(s3)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—	—

功能

- (s3)中指定的输入值(BIN32位值)根据(s1)、(s2)中指定的死区的上下限范围,对(d)中指定的软元件中存储的输出值进行控制。输出值按下述方式被控制。

条件	输出值中存储的值
死区下限值((s1), (s1)+1) > 输入值((s3), (s3)+1) 时	输入值((s3), (s3)+1) - 死区下限值((s1), (s1)+1)
死区上限值((s2), (s2)+1) < 输入值((s3), (s3)+1) 时	输入值((s3), (s3)+1) - 死区上限值((s2), (s2)+1)
死区下限值((s1), (s1)+1) ≤ 输入值((s3), (s3)+1) ≤ 死区上限值((s2), (s2)+1) 时	0



- DBAND(P)指令的情况下,运算结果超出-2147483648~2147483647的范围时,其情况如下例所示。

例

(s1)、(s1)+1为1000, (s3)、(s3)+1为-2147483648时,则输出值=-2147483648-1000=80000000H-000003E8H=7FFFFC18H=2147482648。

- DBAND(P)_U指令的情况下,运算结果超出0~4294967295的范围时,其情况如下例所示。

例

(s1)、(s1)+1为100, (s3)、(s3)+1为50时,则输出值=50-100=00000032H-00000064H=FFFFFFCEH=4294967246。

出错

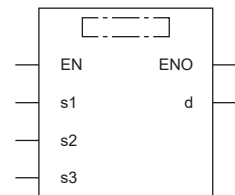
错误代码(SD0)	内容
3285H	(s1)中指定的下限值大于(s2)中指定的上限值时。

ZONE (P) (_U)

对指定的输入值 (BIN16位值) 附加偏置值。

梯形图	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



■执行条件

指令	执行条件
ZONE ZONE_U	
ZONEP ZONEP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)	
(s1)	ZONE (P)	输入值中进行加法运算的负的偏置值	-32768~32767	带符号BIN16位	ANY16_S
	ZONE (P)_U		0~65535	无符号BIN16位	ANY16_U
(s2)	ZONE (P)	输入值中进行加法运算的正的偏置值	-32768~32767	带符号BIN16位	ANY16_S
	ZONE (P)_U		0~65535	无符号BIN16位	ANY16_U
(s3)	ZONE (P)	用于进行区域控制的输入值	-32768~32767	带符号BIN16位	ANY16_S
	ZONE (P)_U		0~65535	无符号BIN16位	ANY16_U
(d)	ZONE (P)	存储通过区域控制进行控制的输出值的软元件	—	带符号BIN16位	ANY16_S
	ZONE (P)_U			无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

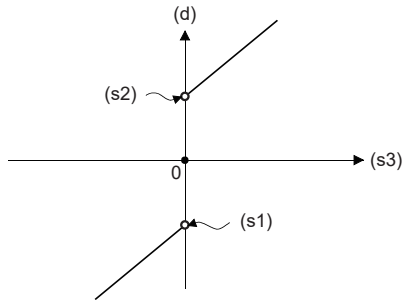
■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	—	—	—	—	—	—	—

功能

- 在(s3)中指定的输入值(BIN16位值)中附加(s1)或(s2)中指定的偏置值后,存储到(d)中指定的软元件编号中。偏置值按下述方式被控制。

条件	输出值中存储的值
输入值(s3)<0时	输入值(s3)+负的偏置值(s1)
输入值(s3)=0时	0
输入值(s3)>0时	输入值(s3)+正的偏置值(s2)



- ZONE(P)指令的情况下,运算结果超出-32768~32767的范围时,其情况如下例所示。

例

(s1)为-100、(s3)为-32768时,则输出值=-32768+(-100)=8000H-FF9CH=7F9CH=32668。

- ZONE(P)_U指令的情况下,运算结果超出0~65535的范围时,其情况如下例所示。

例

(s2)为100、(s3)为65535时,则输出值=65535+100=FFFFH-0064H=0063H=99。

- ZONE(P)_U指令的情况下,(s1)不作为虚拟数据处理使用。

出错

没有运算错误。

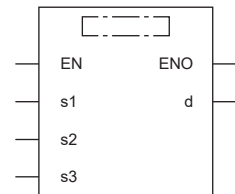
BIN32位数据区域控制

DZONE (P) (_U)

在指定的输入值(BIN32位值)中附加偏置值。

梯形图	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



■执行条件

指令	执行条件
DZONE DZONE_U	
DZONEP DZONEP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	DZONE (P)	存储在输入值中进行加法运算的负的偏置值的软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	DZONE (P)_U		0~4294967295	无符号BIN32位	ANY32_U
(s2)	DZONE (P)	存储在输入值中进行加法运算的正的偏置值的软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	DZONE (P)_U		0~4294967295	无符号BIN32位	ANY32_U
(s3)	DZONE (P)	存储用于进行区域控制的输入值的软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	DZONE (P)_U		0~4294967295	无符号BIN32位	ANY32_U
(d)	DZONE (P)	存储通过区域控制进行控制的输出值的起始软元件	—	带符号BIN32位	ANY32_S
	DZONE (P)_U			无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

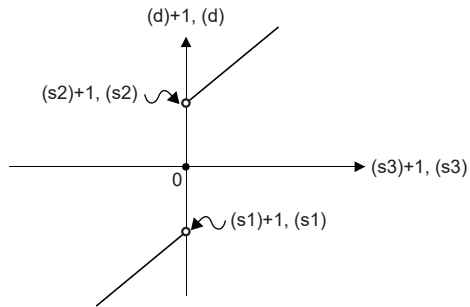
■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	○	—	—	—	—

功能

- 在 (s3) 中指定的输入值 (BIN32位值) 中附加 (s1) 或 (s2) 中指定的偏置值后, 存储到 (d) 中指定的软元件编号中。偏置值按下述方式被控制。

条件	输出值中存储的值
输入值 ((s3), (s3)+1) < 0 时	输入值 ((s3), (s3)+1) + 负的偏置值 (s1), (s1)+1
输入值 ((s3), (s3)+1) = 0 时	0
输入值 ((s3), (s3)+1) > 0 时	输入值 ((s3), (s3)+1) + 正的偏置值 (s2), (s2)+1



- DZONE (P) 指令的情况下, 运算结果超出 -2147483648 ~ 2147483647 的范围时, 其情况如下例所示。

例

(s1)、(s1)+1 为 -1000, (s3)、(s3)+1 为 -2147483648 时, 则输出值 = -2147483648 + (-1000) = 80000000H - FFFFFC18H = 7FFFFC18H = 2147482648。

- DZONE (P)_U 指令的情况下, 运算结果超出 0 ~ 4294967295 的范围时, 其情况如下例所示。

例

(s2)、(s2)+1 为 1000, (s3)、(s3)+1 为 4294967295 时, 则输出值 = 4294967295 + 1000 = FFFFFFFFH - 00003E8H = 000003E7H = 999。

- DZONE (P)_U 指令的情况下, (s1)、(s1)+1 不作为虚拟数据处理使用。

出错

没有运算错误。

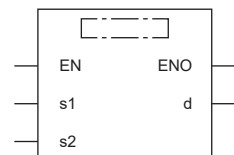
BIN16位单位标度(各点坐标数据)

SCL(P) (_U)

对标度用转换数据(16位数据单位)，按指定的输入值进行标度(各点坐标数据)。

梯形图	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



■执行条件

指令	执行条件
SCL SCL_U	
SCLP SCLP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	SCL(P)	进行标度的输入值或存储了输入值的软元件	-32768~32767	带符号BIN16位	ANY16_S
	SCL(P)_U		0~65535	无符号BIN16位	ANY16_U
(s2)	SCL(P)	存储了标度用转换数据的起始软元件	—	带符号BIN16位*1	ANY16_S*2
	SCL(P)_U			无符号BIN16位*1	ANY16_U*2
(d)	SCL(P)	存储通过标度进行控制的输出值的软元件	—	带符号BIN16位	ANY16_S
	SCL(P)_U			无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

*1 (s2)的坐标点数将变为无符号BIN16位。

*2 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

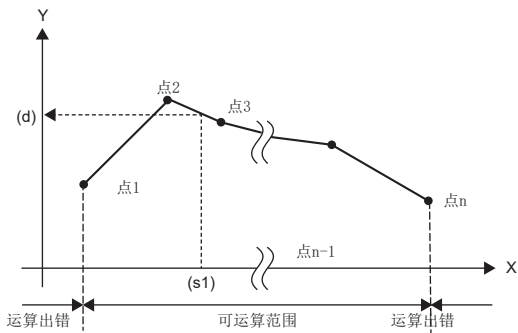
■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 对(s2)中指定的标度用转换数据(16位数据单位),通过(s1)中指定的输入值进行标度,将运算结果存储到(d)中指定的软元件编号中。根据(s2)中指定的软元件及其以后存储的标度用转换数据进行标度转换。

设置项目 (n表示(s2)中指定的坐标点数。)		软元件分配
坐标点数		(s2)
点1	X坐标	(s2)+1
	Y坐标	(s2)+2
点2	X坐标	(s2)+3
	Y坐标	(s2)+4
⋮		
点n	X坐标	(s2)+2n-1
	Y坐标	(s2)+2n



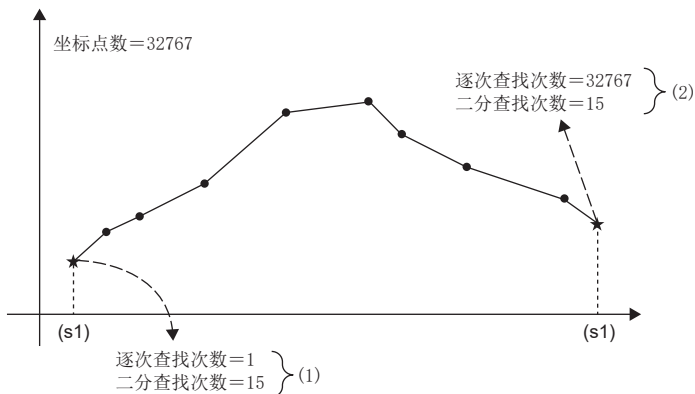
- 运算结果不是整数值的的情况下,对小数点以下第1位数进行四舍五入。
- 标度用转换数据的X坐标数据应设置为升序。
- (s1)应在标度用转换数据范围内((s2)的软元件值)进行设置。
- 多个点指定了同一X坐标的情况下,将输出点No. 为最大的点的Y坐标的值。
- (s2)中指定的标度用转换数据的坐标点数应在1~65535内。

注意事项

- 搜索方法及搜索次数范围根据SM755 (标度数据检查设置)的ON/OFF而有所不同。

SM755	搜索方法	搜索次数范围
OFF	逐次搜索	$1 \leq \text{次数} \leq 65535$
ON	二分搜索	$1 \leq \text{次数} \leq 16$

- 标度用转换数据以升序被排序的情况下，根据SM755的状态搜索方法有所不同，因此处理速度也不相同。处理速度取决于搜索次数，搜索次数越少则处理速度越快。
 - 逐次搜索的处理速度变快的情况下
坐标点数为最大，(s1)位于坐标点1~15之间的情况下，逐次搜索次数将变为 ≤ 15 ，因此逐次搜索的处理速度将变快。
 - 二分搜索的处理速度变快的情况下
最多搜索次数为16次，因此(s1)位于坐标点17以后的情况下，将变为二分搜索次数 \leq 逐次搜索次数，二分搜索的处理速度将变快。



- (1) 逐次搜索次数 < 二分搜索次数，因此逐次搜索的处理速度变快。
 (2) 逐次搜索次数 > 二分搜索次数，因此二分搜索的处理速度变快。

出错

错误代码 (SD0)	内容
3285H	比标度用转换数据的 (s1) 靠前的点的X坐标数据未设置为升序时。 (但是，SM755为ON时，不检测本错误。)
	(s1) 中指定的输入值超出设置的标度用转换数据的范围时。
	(s2) 的软件开始的坐标点数超出1~65535的范围时。

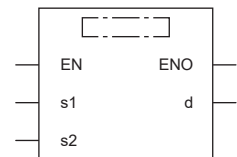
BIN32位单位标度(各点坐标数据)

DSCL(P) (_U)

对标度用转换数据(32位数据单位)，以指定的输入值进行标度(各点坐标数据)。

梯形图	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



■执行条件

指令	执行条件
DSCL DSCL_U	
DSCLP DSCLP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	DSCL(P)	进行标度的输入值或存储了输入值的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	DSCL(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(s2)	DSCL(P)	存储了标度用转换数据的起始软元件	—	带符号BIN32位*1	ANY32_S*2
	DSCL(P)_U			无符号BIN32位*1	ANY32_U*2
(d)	DSCL(P)	存储通过标度进行控制的输出值的起始软元件	—	带符号BIN32位	ANY32_S
	DSCL(P)_U			无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

*1 (s2)+0~(s2)+1的坐标点数将变为无符号BIN32位。

*2 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

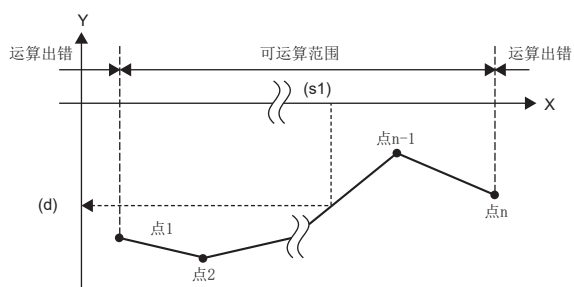
■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

- 对(s2)中指定的标度用转换数据(32位数据单位), 通过(s1)中指定的输入值进行标度, 将运算结果存储到(d)中指定的软元件编号中。根据(s2)中指定的软元件及其以后存储的标度用转换数据进行标度转换。

设置项目 (n表示(s2)中指定的坐标点数。)		软元件分配
坐标点数		(s2)+1, (s2)
点1	X坐标	(s2)+3, (s2)+2
	Y坐标	(s2)+5, (s2)+4
点2	X坐标	(s2)+7, (s2)+6
	Y坐标	(s2)+9, (s2)+8
⋮		
点n	X坐标	(s2)+4n-1, (s2)+4n-2
	Y坐标	(s2)+4n+1, (s2)+4n



- 运算结果不是整数值的情况下, 对小数点以下第1位数进行四舍五入。
- 标度用转换数据的X坐标数据应设置为升序。
- 对(s1), 应在标度用转换数据范围内((s2)、(s2)+1的软元件值)设置。
- 多个点指定了同一X坐标的情况下, 将输出点No. 为最大的点的Y坐标的值。
- (s2)中指定的标度用转换数据的坐标点数应在1~4294967295内。

注意事项

- 搜索方法及搜索次数范围根据SM755 (标度数据检查设置)的ON/OFF而有所不同。

SM755	搜索方法	搜索次数范围
OFF	逐次搜索	1≤次数≤4294967295
ON	二分搜索	1≤次数≤32

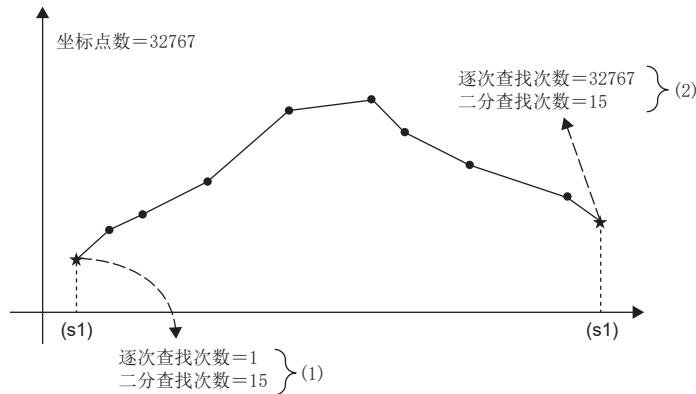
- 标度用转换数据以升序被排序的情况下，根据SM755的状态搜索方法有所不同，因此处理速度也不相同。处理速度取决于搜索次数，搜索次数越少则处理速度越快。

- 逐次搜索的处理速度变快的情况下

坐标点数为最大，(s1)位于坐标点1~15之间的情况下，逐次搜索次数将变为≤15，因此逐次搜索的处理速度将变快。

- 二分搜索的处理速度变快的情况下

最多搜索次数为32次，因此(s1)位于坐标点33以后的情况下，将变为二分搜索次数≤逐次搜索次数，二分搜索的处理速度将变快。



(1) 逐次搜索次数 < 二分搜索次数，因此逐次搜索的处理速度变快。

(2) 逐次搜索次数 > 二分搜索次数，因此二分搜索的处理速度变快。

出错

错误代码 (SD0)	内容
3285H	比标度用转换数据的 (s1) 靠前的点的X坐标数据未设置为升序时。 (但是，SM755为ON时，不检测本错误。)
	(s1) 中指定的输入值超出设置的标度用转换数据的范围时。
	(s2) 的软件开始的坐标点数超出1~4294967295的范围时。

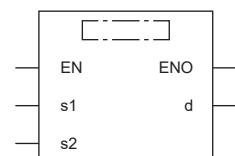
BIN16位单位标度(各X/Y坐标数据)

SCL2(P) (_U)

对标度用转换数据(16位数据单位)，以指定的输入值进行标度(各X/Y坐标数据)。

梯形图	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



■执行条件

指令	执行条件
SCL2 SCL2_U	
SCL2P SCL2P_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	SCL2(P)	进行标度的输入值或存储了输入值的软元件	-32768~32767	带符号BIN16位	ANY16_S
	SCL2(P)_U		0~65535	无符号BIN16位	ANY16_U
(s2)	SCL2(P)	存储了标度用转换数据的起始软元件	—	带符号BIN16位*1	ANY16_S*2
	SCL2(P)_U			无符号BIN16位*1	ANY16_U*2
(d)	SCL2(P)	存储通过标度进行控制的输出值的软元件	—	带符号BIN16位	ANY16_S
	SCL2(P)_U			无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

*1 (s2)的坐标点数将变为无符号BIN16位。

*2 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

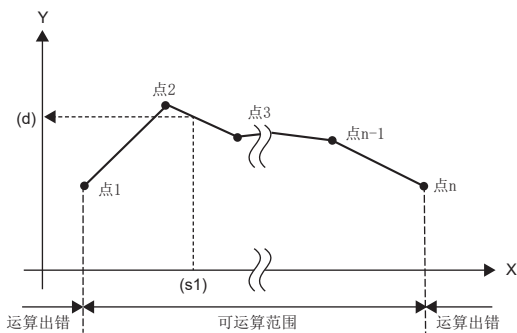
■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—	—

功能

- 对(s2)中指定的标度用转换数据(16位数据单位),通过(s1)中指定的输入值进行标度,将运算结果存储到(d)中指定的软元件编号中。根据(s2)中指定的软元件及其以后存储的标度用转换数据进行标度转换。

设置项目 (n表示(s2)中指定的坐标点数。)		软元件分配
坐标点数		(s2)
X坐标	点1	(s2)+1
	点2	(s2)+2
	⋮	⋮
	点n	(s2)+n
Y坐标	点1	(s2)+n+1
	点2	(s2)+n+2
	⋮	⋮
	点n	(s2)+2n



- 运算结果不是整数值的的情况下,对小数点以下第1位数进行四舍五入。
- 标度用转换数据的X坐标数据应设置为升序。
- (s1)应在标度用转换数据范围内((s2)的软元件值)进行设置。
- 多个点指定了同一X坐标的情况下,将输出点No. 为最大的点的Y坐标的值。
- 应将标度用转换数据的坐标点数设置在1~65535的范围内。

注意事项

标度用转换数据以升序排序的情况下,根据SM755(标度数据检查设置)的状态搜索方法有所不同,因此处理速度也不相同。详细内容,请参阅SCL(P) (_U)指令。

☞ 653页 SCL(P) (_U)

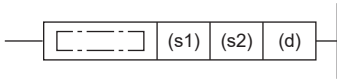
出错

错误代码(SD0)	内容
3285H	X坐标数据未设置为升序时。
	(s1)中指定的输入值超出设置的标度用转换数据的范围时。
	(s2)的软元件开始的坐标点数超出1~65535的范围时。

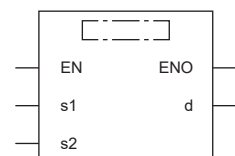
BIN32位单位标度(各X/Y坐标数据)

DSCL2(P) (_U)


对标度用转换数据(32位数据单位)，以指定的输入值进行标度(各X/Y坐标数据)。

梯形图	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



■执行条件

指令	执行条件
DSCL2 DSCL2_U	
DSCL2P DSCL2P_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	DSCL2(P)	进行标度的输入值或存储了输入值的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
	DSCL2(P)_U		0~4294967295	无符号BIN32位	ANY32_U
(s2)	DSCL2(P)	存储了标度用转换数据的起始软元件	—	带符号BIN32位*1	ANY32_S*2
	DSCL2(P)_U			无符号BIN32位*1	ANY32_U*2
(d)	DSCL2(P)	存储通过标度进行控制的输出值的起始软元件	—	带符号BIN32位	ANY32_S
	DSCL2(P)_U			无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

*1 (s2)~(s2)+1的坐标点数将变为无符号BIN32位。

*2 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

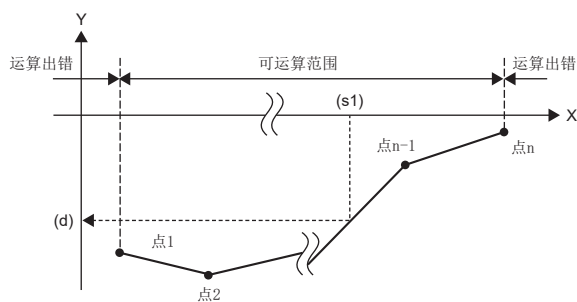
■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	○	○	○	—	—	—	—	—

功能

- 对(s2)中指定的标度用转换数据(32位数据单位),通过(s1)中指定的输入值进行标度,将运算结果存储到(d)中指定的软元件编号中。根据(s2)中指定的软元件及其以后存储的标度用转换数据进行标度转换。

设置项目 (n表示(s2)中指定的坐标点数。)		软元件分配
坐标点数		(s2)+1, (s2)
X坐标	点1	(s2)+3, (s2)+2
	点2	(s2)+5, (s2)+4
	⋮	⋮
	点n	(s2)+2n+1, (s2)+2n
Y坐标	点1	(s2)+2n+3, (s2)+2n+2
	点2	(s2)+2n+5, (s2)+2n+4
	⋮	⋮
	点n	(s2)+4n+1, (s2)+4n



- 运算结果不是整数值的的情况下,对小数点以下第1位数进行四舍五入。
- 标度用转换数据的X坐标数据应设置为升序。
- 对(s1),应在标度用转换数据范围内((s2)~(s2)+1的软元件值)设置。
- 多个点指定了同一X坐标的情况下,将输出点No.为最大的点的Y坐标的值。
- 应将标度用转换数据的坐标点数设置在1~4294967295的范围内。

注意事项

标度用转换数据以升序排序的情况下,根据SM755(标度数据检查设置)的状态搜索方法有所不同,因此处理速度也不相同。关于详细内容,请参阅DSCL(P)(U)指令。

☞ 656页 DSCL(P)(U)

出错

错误代码(SD0)	内容
3285H	X坐标数据未设置为升序时。
	(s1)中指定的输入值超出设置的标度用转换数据的范围时。
	(s2)的软元件开始的坐标点数超出1~4294967295的范围时。

14.6 数据处理指令

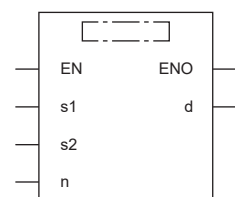
BIN16位数据查找

SERDATA (P)

从 (s2) 中指定的BIN16位数据开始查找 (n) 点的 (s1) 中指定的BIN16位数据。

梯形图	ST
	ENO:=SERDATA (EN, s1, s2, n, d); ENO:=SERDATAP (EN, s1, s2, n, d);

FBD/LD



■执行条件

指令	执行条件
SERDATA	
SERDATAP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s1)	查找数据或存储了查找数据的软元件	-32768~32767	带符号BIN16位	ANY16
(s2)	存储查找数据的起始软元件	—	带符号BIN16位	ANY16*1
(d)	存储查找结果的起始软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 2)
(n)	查找数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

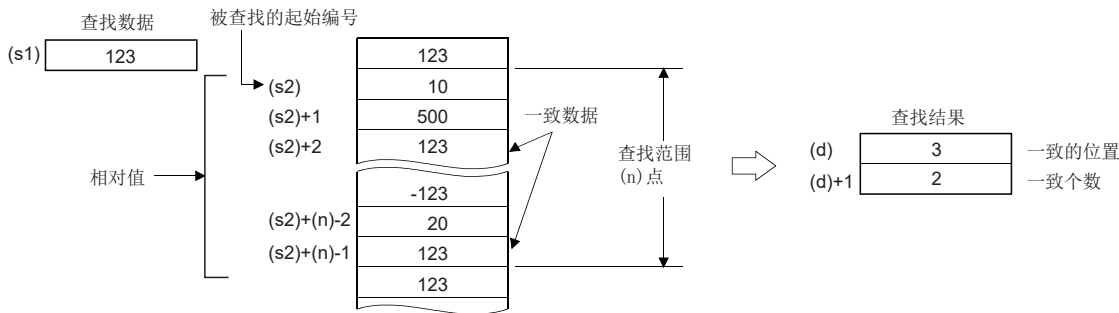
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

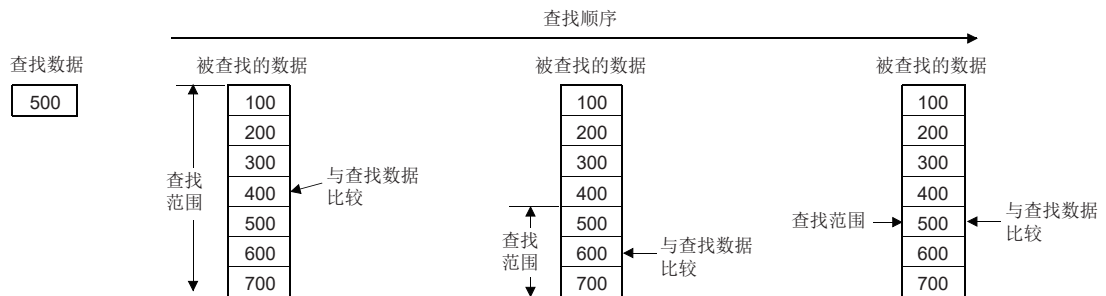
- 将(s1)中指定的软元件的BIN16位数据作为关键字，从(s2)中指定的软元件开始以BIN16位单位查找(n)点。将与关键字一致的个数存储到(d)+1中指定的软元件中，将最先一致的软元件编号从(s2)开始的相对值存储到(d)中指定的软元件中。



- (n)为0的情况下，将变为无处理。
- 查找的结果为未发现一致数据的情况下，(d)、(d)+1中指定的软元件将变为0。

要点

- 通过SERDATA(P)指令查找的数据为升序排序的情况下，将SM702(查找方法)*1置为ON时其处理将快于通过二分搜索法进行的查找。查找的数据未以升序排序的情况下如果将SM702置为ON，将无法获得正常的查找结果。通过二分搜索法进行查找的示例如下所示。



*1 SM702是用于设置查找方法的特殊继电器。

[SM702为OFF的情况下]: 变为逐次搜索法(线形搜索法)。是从查找的数据的起始开始与查找数据进行比较的方法。

[SM702为ON的情况下]: 变为二分搜索法。

对以升序排序的数据找出搜索范围中间的值，根据该值与希望查找的值的大小将搜索范围缩小至某一方。重复此操作，搜索要查找的数据的方法。

出错

没有运算错误。

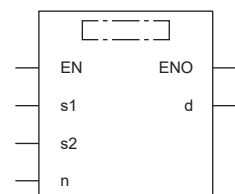
BIN32位数据查找

DSERDATA (P)

从(s2)中指定的BIN32位数据开始查找(n)点的(s1)中指定的BIN32位数据。

梯形图	ST
	ENO:=DSERDATA (EN, s1, s2, n, d) ; ENO:=DSERDATAP (EN, s1, s2, n, d) ;

FBD/LD



■执行条件

指令	执行条件
DSERDATA	
DSERDATAP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	查找数据或存储了查找数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32
(s2)	存储查找数据的起始软元件	—	带符号BIN32位	ANY32* ¹
(d)	存储查找结果的起始软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 2)
(n)	查找数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

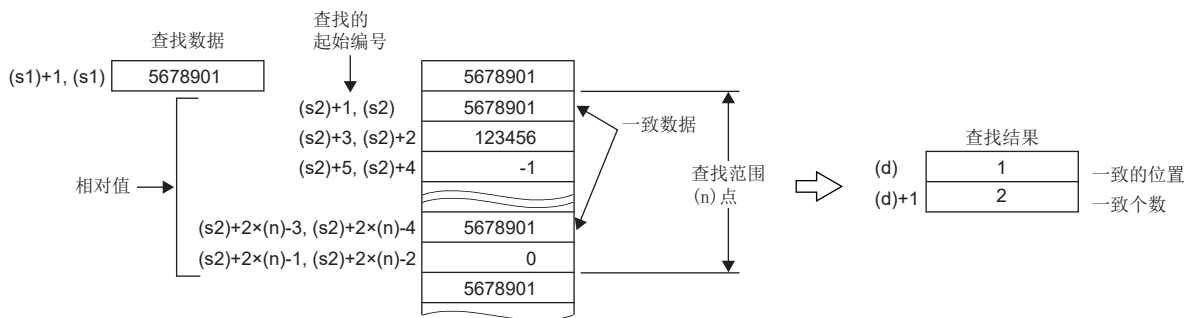
*1 通过标签进行设置的情况下,应在确保动作所需区域的前提下定义数组,指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

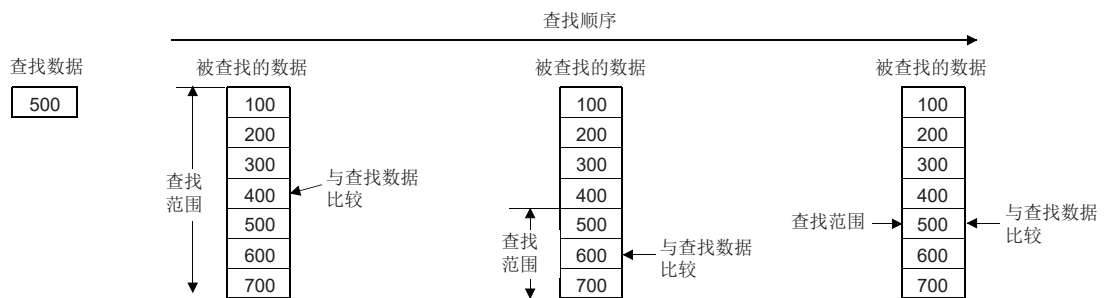
- 将(s1)中指定的软元件的BIN32位数据作为关键字，从(s2)中指定的软元件开始以BIN32位单位查找(n)点(以16位单位 $2 \times (n)$ 点)。将与关键字一致的个数存储到(d)+1中指定的软元件中，将最先一致的软元件编号从(s2)开始的相对值存储到(d)中指定的软元件中。



- (n)为0的情况下，将变为无处理。
- 查找结果为未发现一致数据的情况下，(d)、(d)+1中指定的软元件将变为0。

要点

- 通过DSERDATA(P)指令查找的数据为升序排序的情况下，将SM702(查找方法)*1置为ON时其处理将快于通过二分搜索法进行的查找。查找的数据未以升序排序的情况下如果将SM702置为ON，将无法获得正常的查找结果。通过二分搜索法进行查找的示例如下所示。



*1 SM702是用于设置查找方法的特殊继电器。

[SM702为OFF的情况下]: 变为逐次搜索法(线形搜索法)。

是从查找的数据的起始开始与查找数据进行比较的方法。

[SM702为ON的情况下]: 变为二分搜索法。

对以升序排序的数据找出搜索范围中间的值，根据该值与希望查找的值的大小将搜索范围缩小至某一方。重复此操作，搜索要查找的数据的方法。

出错

没有运算错误。

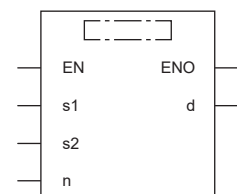
BIN16位数据查找(最小·相同·最大)

SERMM(P)

从(s1)中指定的BIN16位数据中查找(n)点的与(s2)中指定的BIN16位数据相同的数据以及最大值、最小值。

梯形图	ST
	ENO:=SERMM(EN, s1, s2, n, d); ENO:=SERMMP(EN, s1, s2, n, d);

FBD/LD



■执行条件

指令	执行条件
SERMM	
SERMMP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	同一数据、最大值、最小值的搜索起始软元件	—	带符号BIN16位	ANY16
(s2)	同一数据的搜索值或其存储目标软元件	—	带符号BIN16位	ANY16
(d)	同一数据、最大值、最小值搜索后存储这些个数的起始软元件	—	无符号BIN16位	ANY16_ARRAY (要素数: 5)
(n)	同一数据、最大值、最小值搜索个数	0~65535	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○*1	○	○	○	○	—	○	○	—	—	—	—
(d)	—	—	○	○	—	—	○	—	—	—	—	—
(n)	○*1	○	○	○	○	—	○	○	—	—	—	—

*1 FX、FY不能使用。

功能

- 对以(s1)为起始的(n)个的数据搜索与(s2)的BIN16位数据相同的数据，将结果存储到(d)~(d)+4中。
- 有同一数据的情况下，以(d)为起始的5点的软元件中，存储同一数据的个数、首次/最终位置及最大值、最小值的位置。
- 没有同一数据的情况下，以(d)为起始的5点的软元件中，存储同一数据的个数、首次/最终位置及最大值、最小值的位置。
但是，以(d)为起始的3点的软元件(同一数据的个数、首次/最终位置)中将存储0。
- 数据中有多个最小值、最大值时，将分别存储后面的位置。
- (n)中指定了0的情况下，将变为无处理。
- 搜索结果表的构成及数据示例如下所示。(n=10)

被搜索软元件 (s1)	被搜索数据 (s1)的值(例)	比较数据 (s2)的值(例)	数据的位置	搜索结果		
				最大值(d)+4	一致(d)	最小值(d)+3
(s1)	100	100	0	—	○(首次)	—
(s1)+1	111		1	—	—	—
(s1)+2	100		2	—	○	—
(s1)+3	98		3	—	—	—
(s1)+4	123		4	—	—	—
(s1)+5	66		5	—	—	○
(s1)+6	100		6	—	○(最终)	—
(s1)+7	95		7	—	—	—
(s1)+8	210		8	○	—	—
(s1)+9	88		9	—	—	—

- 上述示例的搜索结果表如下所示。

软元件编号	内容	搜索结果项目
(d)	3	同一数据的个数
(d)+1	0	同一数据的位置(首次)
(d)+2	6	同一数据的位置(最终)
(d)+3	5	最小值的最终位置
(d)+4	8	最大值的最终位置

出错

没有运算错误。

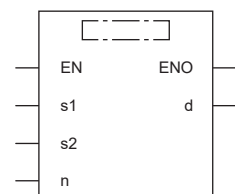
BIN32位数据查找(最小·同一·最大)

DSERMM(P)

从(s1)中指定的BIN32位数据中查找(n)点的与(s2)中指定的BIN32位数据相同的数据以及最大值、最小值。

梯形图	ST
	ENO:=DSERMM(EN, s1, s2, n, d); ENO:=DSERMP(EN, s1, s2, n, d);

FBD/LD



■执行条件

指令	执行条件
DSERMM	
DSERMP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	同一数据、最大值、最小值的搜索起始软元件	—	带符号BIN32位	ANY32
(s2)	同一数据的搜索值或其存储目标软元件	—	带符号BIN32位	ANY32
(d)	同一数据、最大值、最小值搜索后存储这些个数的起始软元件	—	无符号BIN32位	ANY32_ARRAY (要素数: 5)
(n)	同一数据、最大值、最小值搜索个数	0~65535	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○*1	○	○	○	○	○	○	○	○	—	—	—
(d)	—	—	○	○	—	—	—	○	—	—	—	—
(n)	○*1	○	○	○	○	—	—	○	○	—	—	—

*1 FX、FY不能使用。

功能

- 对以(s1)+1、(s1)为起始的(n)个的数据搜索与(s2)+1、(s2)的BIN32位数据相同的数据，将结果存储到[(d)+1, (d)]~[(d)+9, (d)+8]中。
- 有同一数据的情况下，以(d)+1, (d)为起始的5点的软元件中，存储同一数据的个数、首次/最终位置及最大值、最小值的位置。
- 每有同一数据的情况下，以(d)+1, (d)为起始的5点的软元件中，存储同一数据的个数、首次/最终位置及最大值、最小值的位置。但是，以(d)+1, (d)为起始的3点的软元件(同一数据的个数、首次/最终位置)中将存储0。
- 数据中有多个最小值、最大值时，将分别存储后面的位置。
- (n)中指定了0的情况下，将变为无处理。
- 搜索结果表的构成及数据示例如下所示。(n=10)

被搜索软元件 (s1)	被搜索数据 (s1)的值(例)	比较数据 (s2)的值(例)	数据的位置	搜索结果		
				最大值 (d)+9, (d)+8	一致(d)	最小值 (d)+7, (d)+6
(s1)+1, (s1)	100000	100000	0	—	○(首次)	—
(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	—	○(最终)	—
(s1)+15, (s1)+14	95000		7	—	—	—
(s1)+17, (s1)+16	910000		8	○	—	—
(s1)+19, (s1)+18	910000		9	○	—	—

- 上述示例的搜索结果表如下所示。

软元件编号	内容	搜索结果项目
(d)+1, (d)	3	同一数据的个数
(d)+3, (d)+2	0	同一数据的位置(首次)
(d)+5, (d)+4	6	同一数据的位置(最终)
(d)+7, (d)+6	5	最小值的最终位置
(d)+9, (d)+8	9	最大值的最终位置

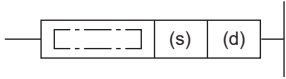
出错

没有运算错误。

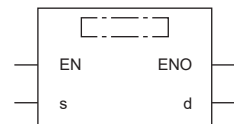
BIN16位数据位检查

SUM(P)


存储指定的软元件的BIN16位数据中处于1的位的总数。

梯形图	ST
	ENO:=SUM(EN, s, d); ENO:=SUMP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
SUM	
SUMP	

设置数据

■内容、范围、数据类型

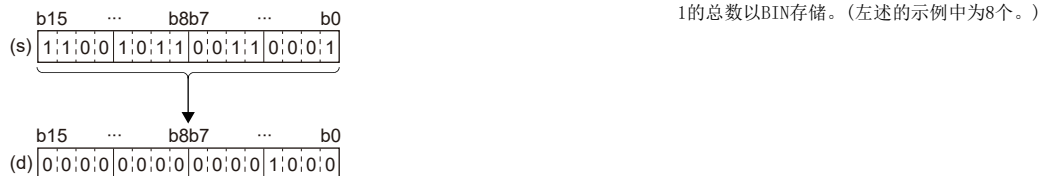
操作数	内容	范围	数据类型	数据类型(标签)
(s)	对处于1的位的总数进行计数的软元件	-32768~32767	带符号BIN16位	ANY16
(d)	存储位的总数的软元件	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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	
(s)	○	○	○	○	○	—	—	○	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—

功能

将(s)中指定的软元件的BIN16位数据中处于1的位的总数存储到(d)中指定的软元件中。



出错

没有运算错误。

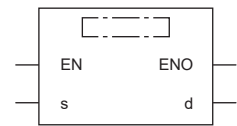
BIN32位数据位检查

DSUM(P)

存储指定的软元件的BIN32位数据中处于1的位的总数。

梯形图	ST
	ENO:=DSUM(EN, s, d); ENO:=DSUMP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
DSUM	
DSUMP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	对处于1的位的总数进行计数的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32
(d)	存储位的总数的软元件	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

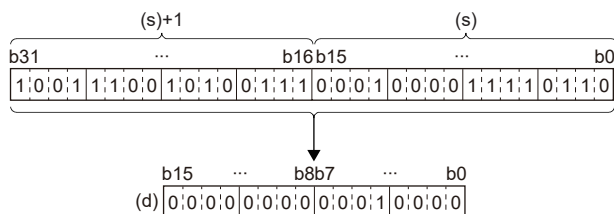
■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

将(s)中指定的软元件的BIN32位数据中，处于1的位的总数存储到(d)中指定的软元件中。

1的总数以BIN存储。(左述的示例中为16个。)



出错

没有运算错误。

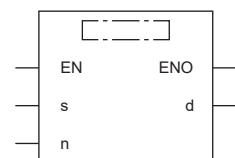
BIN16位数据最大值搜索

MAX(P) (_U)

从指定的软元件开始的(n)点的BIN16位数据中搜索最大值。

梯形图	ST*1	
	ENO:=MAXP(EN, s, n, d);	ENO:=MAXP_U(EN, s, n, d);

FBD/LD*1



*1 在ST、FBD/LD中不支持MAX指令、MAX_U指令。应使用通用函数的MAX。

☞ 1306页 MAX(_E)、MIN(_E)

■执行条件

指令	执行条件
MAX MAX_U	
MAXP MAXP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	MAX(P)	—	带符号BIN16位	ANY16_S*1
	MAX(P)_U		无符号BIN16位	ANY16_U*1
(d)	MAX(P)	—	带符号BIN16位	ANY16_S_ARRAY (要素数: 3)
	MAX(P)_U		无符号BIN16位	ANY16_U_ARRAY (要素数: 3)
(n)	查找数据数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

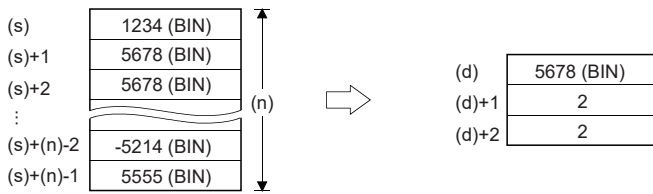
*1 通过标签进行设置的情况下,应在确保动作所需区域的前提下定义数组,指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

从(s)中指定的软元件开始的(n)点的BIN16位数据中搜索最大值后,将最大值存储到(d)中指定的软元件中。从(s)中指定的软元件中搜索,将存储了最先检测到的最大值的软元件编号是从(s)开始的第几点存储到(d)+1中,将最大值的个数存储到(d)+2中。



(d): 最大值
(d)+1: 位置
(d)+2: 个数

出错

没有运算错误。

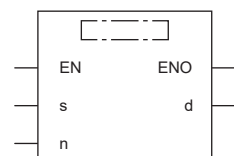
BIN32位数据最大值搜索

DMAX(P) (_U)

从指定的软元件开始的(n)点的BIN32位数据中搜索最大值。

梯形图	ST*1	
	ENO:=DMAXP(EN, s, n, d);	ENO:=DMAXP_U(EN, s, n, d);

FBD/LD*1



*1 在ST、FBD/LD中不支持DMAX指令、DMAX_U指令。应使用通用函数的MAX。

☞ 1306页 MAX(_E)、MIN(_E)

■执行条件

指令	执行条件
DMAX DMAX_U	
DMAXP DMAXP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	DMAX(P)	—	带符号BIN32位	ANY32_S*1
	DMAX(P)_U		无符号BIN32位	ANY32_U*1
(d)	DMAX(P)	—	带符号BIN32位	—*2 (ANY32_S_ARRAY)
	DMAX(P)_U		无符号BIN32位	—*2 (ANY32_U_ARRAY)
(n)	查找数据数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

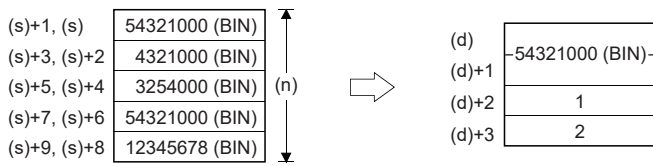
*2 不管使用的程序语言如何，通过软元件指定。请勿指定标签。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

从(s)中指定的软元件开始的(n)点的BIN32位数据中搜索最大值后,将最大值存储到(d)、(d)+1中指定的软元件中。从(s)中指定的软元件中搜索,将存储了最先检测到的最大值的软元件编号是从(s)开始的第几点存储到(d)+2中,将最大值的个数存储到(d)+3中。



(d)、(d)+1: 最大值

(d)+2: 位置

(d)+3: 个数

出错

没有运算错误。

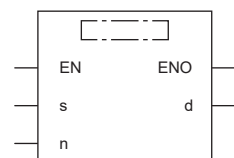
BIN16位数据最小值搜索

MIN(P) (_U)

从指定的软元件开始的 (n) 点的BIN16位数据中搜索最小值。

梯形图	ST*1	
	ENO:=MINP(EN, s, n, d);	ENO:=MINP_U(EN, s, n, d);

FBD/LD*1



*1 在ST、FBD/LD中不支持MIN指令、MIN_U指令。应使用通用函数的MIN。

☞ 1306页 MAX(_E)、MIN(_E)

■执行条件

指令	执行条件
MIN MIN_U	
MINP MINP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	MIN(P)	—	带符号BIN16位	ANY16_S*1
	MIN(P)_U		无符号BIN16位	ANY16_U*1
(d)	MIN(P)	—	带符号BIN16位	ANY16_S_ARRAY (要素数: 3)
	MIN(P)_U		无符号BIN16位	ANY16_U_ARRAY (要素数: 3)
(n)	查找数据数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

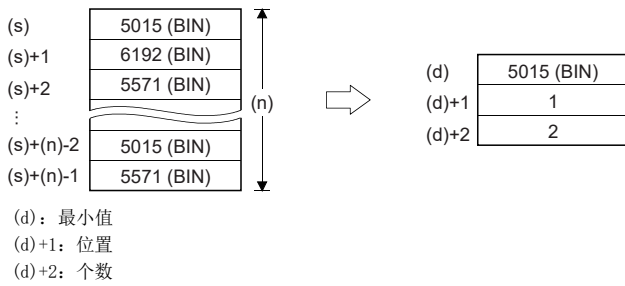
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

从(s)中指定的软元件开始至(n)点为止的BIN16位数据中搜索最小值，将最小值存储到(d)中指定的软元件中。从(s)中指定的软元件中搜索，将存储了最先检测到的最小值的软元件编号是从(s)开始的第几点存储到(d)+1中，将最小值的个数存储到(d)+2中。



出错

没有运算错误。

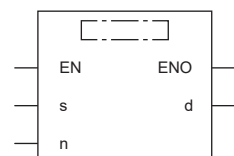
BIN32位数据最小值搜索

DMIN(P) (_U)

从指定的软元件开始的(n)点的BIN32位数据中搜索最小值。

梯形图	ST*1	
	ENO:=DMINP(EN, s, n, d);	ENO:=DMINP_U(EN, s, n, d);

FBD/LD*1



*1 在ST、FBD/LD中不支持DMIN指令、DMIN_U指令。应使用通用函数的MIN。

☞ 1306页 MAX(_E)、MIN(_E)

■执行条件

指令	执行条件
DMIN DMIN_U	
DMINP DMINP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	DMIN(P)	—	带符号BIN32位	ANY32_S*1
	DMIN(P)_U		无符号BIN32位	ANY32_U*1
(d)	DMIN(P)	—	带符号BIN32位	—*2 (ANY32_S_ARRAY)
	DMIN(P)_U		无符号BIN32位	—*2 (ANY32_U_ARRAY)
(n)	查找数据数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

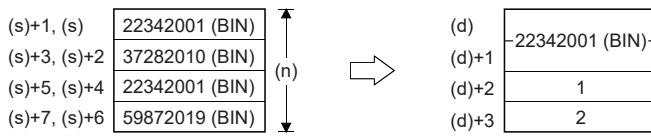
*2 不管使用的程序语言如何，通过软元件指定。请勿指定标签。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

从(s)中指定的软元件开始至(n)点为止的BIN32位数据中搜索最小值，将最小值存储到(d)、(d)+1中指定的软元件中。从(s)中指定的软元件中搜索，将存储了最先检测到最小值的软元件编号是从(s)开始的第几点存储到(d)+2中，将最小值的个数存储到(d)+3中。



(d)、(d)+1: 最小值

(d)+2: 位置

(d)+3: 个数

出错

没有运算错误。

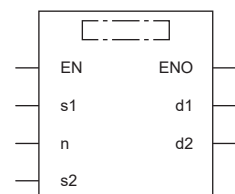
BIN16位数据排序

SORTD(_U)

将指定的 (n) 点的BIN16位数据以升序/降序进行排序 (按顺序排列)。

梯形图	ST	
	ENO:=SORTD(EN, s1, n, s2, d1, d2);	ENO:=SORTD_U(EN, s1, n, s2, d1, d2);

FBD/LD



■执行条件

指令	执行条件
SORTD SORTD_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s1)	SORTD	—	带符号BIN16位	ANY16_S*1
	SORTD_U		无符号BIN16位	ANY16_U*1
(n)	排序数据数	0~65535	无符号BIN16位	ANY16
(s2)	通过1次执行进行比较的数据数	1~65535	无符号BIN16位	ANY16
(d1)	通过排序完成置为ON的位软元件编号	—	位	ANY_BOOL
(d2)	系统使用软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

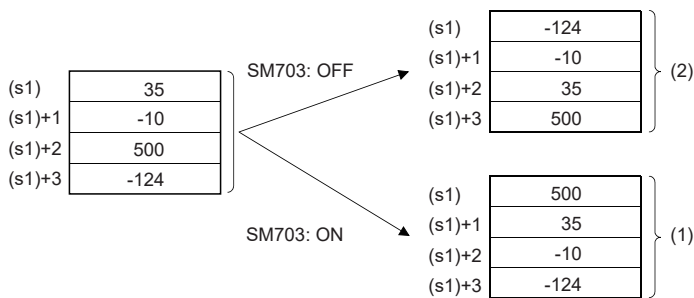
■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	
(d1)	○	—	○*1	—	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	—	○	—	—	—	—	

*1 T、C、ST不能使用。

功能

- 将从(s1)开始的(n)点的BIN16位数据以升序或降序进行排序(按顺序排列)。排序顺序, 在SM703(排序)为OFF时以升序进行排序, ON时以降序进行排序。



- (1) 按降序排序。
(2) 按升序排序。

- 通过SORTD(_U)指令进行的排序需要数个扫描。执行完成为止的扫描次数为, 将排序执行完成为止的最大执行次数用通过(s2)中指定的1次执行进行比较的数据数相除后的值(小数点以下进位)。增加(s2)的值时, 排序完成为止的扫描次数将变少, 但扫描时间将延长。
- 排序执行完成为止的最大执行次数通过以下公式计算: $(n) \times (n-1) \div 2$ (次)。例如, (n)=10的情况下, 变为 $10 \times (10-1) \div 2 = 45$ 次。此时设置为(s2)=2时, 排序完成为止变为 $45 \div 2 = 22.5$, 需要23个扫描。
- (d1)中指定的软元件(完成软元件), 通过SORTD(_U)指令执行开始变为OFF, 排序完成时变为ON。排序完成后, (d1)中指定的软元件将保持ON状态不变, 因此应根据需要将其置为OFF。
- 从(d2)中指定的软元件开始的2点, 在执行SORTD(_U)指令时由系统使用。请勿对(d2)中指定的软元件开始的2点进行更改。进行了更改的情况下, 可能错误。(错误代码: 3285H)
- 排序执行中更改了(n)的情况下, 将以更改后的排序数据数进行排序。
- 排序执行中将执行指令置为了OFF的情况下, 排序将中断。再次将执行指令置为ON的情况下, 将从头开始重新排序。
- 排序执行完成后, 连续进行下一个排序的情况下, 需要将执行指令置为一次OFF后, 再次将执行指令置为ON。

出错

错误代码(SD0)	内容
2821H	从(s1)开始的(n)点的软元件范围与从(d2)开始的2点的软元件范围重复时。
3285H	(s2)为0时。
	第2个扫描及其以后, 系统使用的(d2)的值大于等于(n)的值时。
	第2个扫描及其以后, 系统使用的(d2)的值处于 $(d2) < (d2)+1$ 的状态时。

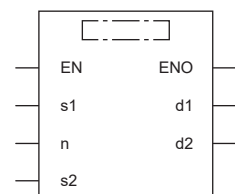
BIN32位数据排序

DSORTD(_U)

将指定的 (n) 点的BIN32位数据以升序/降序进行排序 (按顺序排列)。

梯形图	ST	
	ENO:=DSORTD(EN, s1, n, s2, d1, d2);	ENO:=DSORTD_U(EN, s1, n, s2, d1, d2);

FBD/LD



■执行条件

指令	执行条件
DSORTD DSORTD_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s1)	DSORTD	—	带符号BIN32位	ANY32_S*1
	DSORTD_U		无符号BIN32位	ANY32_U*1
(n)	排序数据数	0~65535	无符号BIN16位	ANY16
(s2)	通过1次执行进行比较的数据数	1~65535	无符号BIN16位	ANY16
(d1)	通过排序完成置为ON的位软元件编号	—	位	ANY_BOOL
(d2)	系统使用软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下, 应在确保动作所需区域的前提下定义数组, 指定该数组型标签的要素。

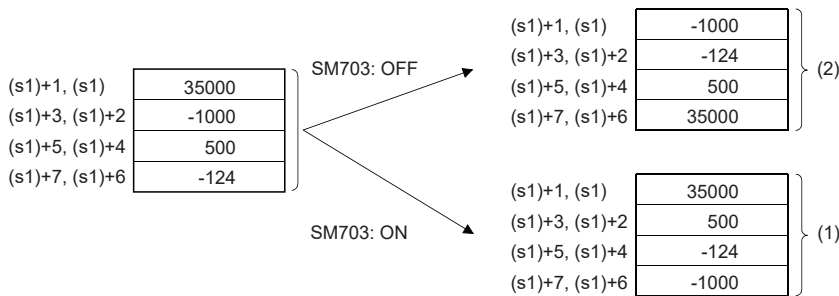
■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	
(d1)	○	—	○*1	—	—	—	—	○	—	—	—	—	
(d2)	—	—	○	—	—	—	—	○	—	—	—	—	

*1 T、C、ST不能使用。

功能

- 将从(s1)开始的(n)点的BIN32位数据以升序或降序进行排序(按顺序排列)。排列顺序, 在SM703(排序)为OFF时以升序进行排序, ON时以降序进行排序。



- (1) 按降序排序。
(2) 按升序排序。

- 通过DSORTD(_U)指令进行排序时, 需要数个扫描。执行完成为止的扫描次数为, 将排序执行完成为止的最大执行次数用通过(s2)中指定的1次执行进行比较的数据数相除后的值(小数点以下进位)。增加(s2)的值时, 排序完成为止的扫描次数将变少, 但扫描时间将延长。
- 排序执行完成为止的最大执行次数通过以下公式计算: $(n) \times (n-1) \div 2$ (次)。例如, (n)=10的情况下, 变为 $10 \times (10-1) \div 2 = 45$ 次。此时设置为(s2)=2时, 排序完成为止变为 $45 \div 2 = 22.5$, 需要23个扫描。
- (d1)中指定的软元件(完成软元件), 通过DSORTD(_U)指令开始执行变为OFF, 排序完成时将变为ON。排序完成后, (d1)中指定的软元件将保持ON状态不变, 因此应根据需要将其置为OFF。
- 从(d2)中指定的软元件开始的2点, 在执行DSORTD(_U)指令时由系统使用。请勿对(d2)中指定的软元件开始的2点进行更改。进行了更改的情况下, 可能错误。(错误代码: 3285H)
- 排序执行中更改了(n)的情况下, 将以更改后的排序数据数进行排序。
- 排序执行中将执行指令置为了OFF的情况下, 排序将中断。再次将执行指令置为ON的情况下, 将从头开始重新排序。
- 排序执行完成后, 连续进行下一个排序的情况下, 需要将执行指令置为一次OFF后, 再次将执行指令置为ON。

出错

错误代码(SD0)	内容
2821H	从(s1)开始的 $2 \times (n)$ 点的软元件范围与从(d2)开始的2点的软元件范围重复时。
3285H	(s2)为0时。
	第2个扫描及其以后, 系统使用的(d2)的值大于等于(n)的值时。
	第2个扫描及其以后, 系统使用的(d2)的值处于 $(d2) < (d2)+1$ 的状态时。

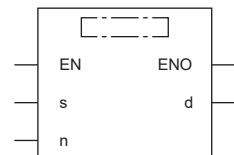
BIN16位数据合计值计算

WSUM(P) (_U)

将指定的软元件开始的(n)点的BIN16位数据全部进行加法运算。

梯形图	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



■执行条件

指令	执行条件
WSUM WSUM_U	
WSUMP WSUMP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	WSUM(P)	存储了进行合计值计算的数据的起始软元件	带符号BIN16位	ANY16_S*1
	WSUM(P)_U		无符号BIN16位	ANY16_U*1
(d)	WSUM(P)	存储合计值的起始软元件	带符号BIN32位	ANY32_S
	WSUM(P)_U		无符号BIN32位	ANY32_U
(n)	数据个数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

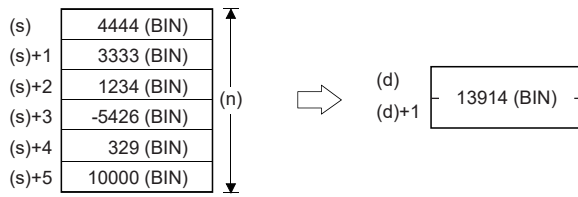
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

将(s)中指定的软元件开始的(n)点的BIN16位数据全部进行加法运算后, 存储到(d)中指定的软元件中。



出错

没有运算错误。

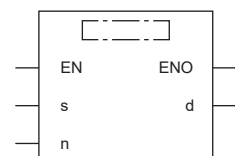
BIN32位数据合计值计算

DWSUM(P) (_U)

将从指定的软元件开始的(n)点的BIN32位数据全部进行加法运算。

梯形图	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



■执行条件

指令	执行条件
DWSUM DWSUM_U	
DWSUMP DWSUMP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	DWSUM(P)	存储了进行合计值计算的数据的起始软元件	带符号BIN32位	ANY32_S*1
	DWSUM(P)_U		无符号BIN32位	ANY32_U*1
(d)	DWSUM(P)	存储合计值的起始软元件	带符号BIN64位	ANY32_ARRAY (要素数: 2)
	DWSUM(P)_U		无符号BIN64位	
(n)	数据个数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

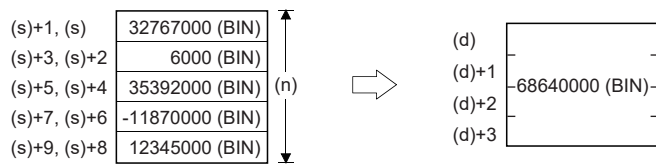
*1 通过标签进行设置的情况下,应在确保动作所需区域的前提下定义数组,指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

将从(s)中指定的软元件开始的(n)点的BIN32位数据全部进行加法运算后, 存储到(d)中指定的软元件中。



出错

没有运算错误。

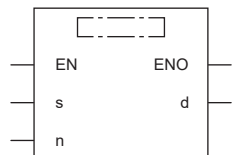
BIN16位数据平均值计算

MEAN(P) (_U)

对指定的软元件开始的(n)点的(16位数据单位)的平均值进行计算。

梯形图	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



■执行条件

指令	执行条件
MEAN MEAN_U	
MEANP MEANP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	MEAN(P)	存储了进行平均值计算的数据的起始软元件	带符号BIN16位	ANY16_S*1
	MEAN(P)_U		无符号BIN16位	ANY16_U*1
(d)	MEAN(P)	存储平均值的软元件	带符号BIN16位	ANY16_S
	MEAN(P)_U		无符号BIN16位	ANY16_U
(n)	数据数或存储了数据数的软元件编号	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

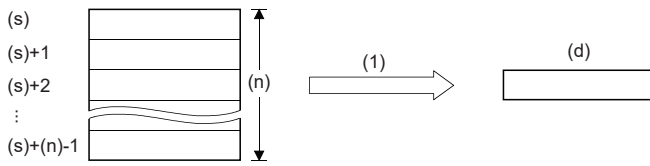
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

- 对(s)中指定的软元件开始的(n)点的BIN16位数据的平均值进行计算后, 存储到(d)中指定的软元件中。



(1) 平均值

- 计算结果不是整数值的情况下, 小数点以下将被舍去。
- (n)中指定的值为0的情况下, 将变为无处理。

出错

没有运算错误。

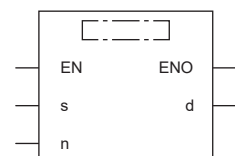
BIN32位数据平均值计算

DMEAN(P) (_U)

对指定的软元件开始的(n)点的(32位数据单位)的平均值进行计算。

梯形图	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



■执行条件

指令	执行条件
DMEAN DMEAN_U	
DMEANP DMEANP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	DMEAN(P)	存储了进行平均值计算的数据的起始软元件	带符号BIN32位	ANY32_S*1
	DMEAN(P)_U		无符号BIN32位	ANY32_U*1
(d)	DMEAN(P)	存储平均值的起始软元件	带符号BIN32位	ANY32_S
	DMEAN(P)_U		无符号BIN32位	ANY32_U
(n)	数据数或存储了数据数的软元件编号	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

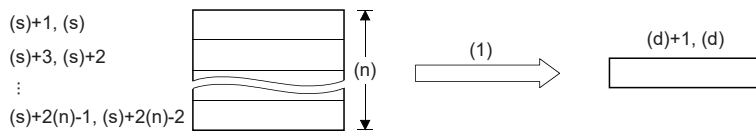
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

- 对(s)中指定的软元件开始的(n)点的BIN32位数据的平均值进行计算，存储到(d)中指定的软元件中。



(1) 平均值

- 计算结果不是整数值的的情况下，小数点以下将被舍去。
- (n)中指定的值为0的情况下，将变为无处理。

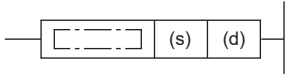
出错

没有运算错误。

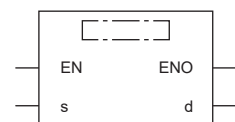
BIN16位平方根计算

SQRT(P)

对指定的BIN16位数据的平方根进行运算。


梯形图	ST*1
	ENO:=SQRTP(EN, s, d);

FBD/LD*1



*1 在ST语言、FBD/LD语言中不支持SQRT指令。应使用通用函数的SQRT。(参见1266页 SQRT(E))

■执行条件

指令	执行条件
SQRT	
SQRTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储平方根运算数据的软元件	0~65535	无符号BIN16位	ANY16
(d)	存储运算后平方根的软元件	—	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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不能使用。

功能

对(s)中指定的BIN16位数据的平方根进行运算，将运算结果存储到(d)中。运算结果为舍去小数点的整数。

$$\sqrt{(s)} \rightarrow (d)$$

出错

没有运算错误。

BIN32位平方根计算

DSQRT (P)

对指定的BIN32位数据的平方根进行运算。

梯形图	ST
	ENO:=DSQRT (EN, s, d) ; ENO:=DSQRTP (EN, s, d) ;

FBD/LD

■执行条件

指令	执行条件
DSQRT	
DSQRTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储平方根运算数据的软元件	0~4294967295	无符号BIN32位	ANY32
(d)	存储运算后平方根的软元件	—	无符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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	
(s)	○*1	○	○	○	○	○	○	○	—	—	—
(d)	○*1	○	○	○	○	○	○	—	—	—	—

*1 FX、FY不能使用。

功能

对(s)中指定的BIN32位数据的平方根进行运算，将运算结果存储到(d)中。运算结果为舍去小数点的整数。

$$\sqrt{(s)+1}, (s) \rightarrow (d)$$

出错

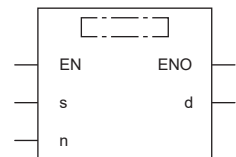
没有运算错误。

CRC(P)

以(s)中指定的软元件为起始的(n)点的8位数据(字节单位),生成CRC值后存储到(d)中。

梯形图	ST
	ENO:=CRC(EN, s, n, d); ENO:=CRCP(EN, s, n, d);

FBD/LD



■执行条件

指令	执行条件
CRC	
CRCP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储CRC值生成对象数据的起始软元件	—	带符号BIN16位	ANY16
(d)	生成的CRC值的存储目标软元件	—	带符号BIN16位	ANY16
(n)	计算CRC值的8位数据(字节单位)数,或存储了8位数据(字节单位)数的软元件	0~65535	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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	—	○	—	—	—	○	—	—	—	—
(n)	○*1	○	○	○	○	—	○	○	—	—	—

*1 FX、FY不能使用。

功能

- 以(s)中指定的软元件为起始的(n)点的8位数据(字节单位),生成CRC值后存储到(d)中。CRC值(CRC-16)的生成多项式中使用了对“ $X^{16}+X^{15}+X^2+1$ ”。该指令中,作为计算时使用的模式,有16位转换模式及8位转换模式。根据SM772(CCD/CRC指令转换模式指定)的ON/OFF可以选择转换模式。
- (n)中指定了0的情况下,将变为无处理。
- 各转换模式的动作如下所示。

■16位转换模式(SM772=OFF时)

对(s)的软元件的高位8位(字节单位)及低位8位(字节单位)进行CRC运算。(d)中指定的软元件1点的16位中将存储运算结果。

例

(n)=6的情况下

16位转换模式的情况下,下述阴影部分的6字节将成为运算对象。CRC值为“A57BH”,因此将“A57BH”存储到(d)的软元件中。

	10进制	16进制	
		高位	低位
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位转换模式(SM772=ON时)

仅对(s)的软元件的低位8位(低位字节)进行CRC运算。运算结果使用(d)中指定的软元件开始的2点,(d)中存储低位8位(字节单位),(d)+1中存储高位8位(字节单位)。

例

(n)=6的情况下

8位转换模式的情况下,下述阴影部分的6字节将成为运算对象。CRC值为“BDA1H”,因此将“A1H”存储到(d)中,将“BDH”存储到(d)+1中。

	10进制	16进制	
		高位	低位
D0	24932	61H	64H
D1	4219	10H	7BH
D2	-1333	FAH	CBH
D3	-1	FFH	FFH
D4	32761	7FH	F9H
D5	10000	27H	10H

出错

错误代码(SD0)	内容
3285H	在(s)、(d)中位软元件的位指定中,指定了4位指定以外时。

15.1 调试、故障诊断指令

错误显示或报警器复位

LEDR

对控制器的报警器显示及可继续运行的自诊断错误的显示进行复位。

梯形图	ST
	ENO:=LEDR(EN);

FBD/LD

■执行条件

指令	执行条件
LEDR	

功能

- 对控制器的报警器显示及可继续运行的自诊断错误的显示进行复位。通过执行1次指令，错误显示及报警器二者均将进行复位。
- 发生自诊断错误时的动作如下所示。
 - 发生了可继续运行的自诊断错误时。
控制器显示为可继续运行的自诊断错误时，控制器前面的ERROR LED将熄灯。
此时SM0、SM1(最新自诊断错误)、SDO(最新自诊断错误代码)的内容不被复位，因此应通过程序进行复位。
 - 发生了电池错误时。
更换了电池后执行LEDR指令时，控制器前面的BATTERY LED将熄灯。此时SM51(电池过低锁存)也将变为OFF。
- 报警器(F)为ON时的动作如下所示。
 - USER LED将熄灯。
 - SD62(报警器编号)、SD63(报警器个数)、SD64~79(报警器检测编号表)将全部被清除。

执行前	执行后
SD62 200	SD62 0
SD63 15	SD63 0
SD64 200	SD64 0
SD65 99	SD65 0
SD66 5	SD66 0
SD67 255	SD66 0
⋮	⋮
SD77	SD77 0
SD78 83	SD78 0
SD79 0	SD79 0

(1)

(1) 全部清除。

出错

没有运算错误。

16 字符串处理

16.1 字符串处理指令

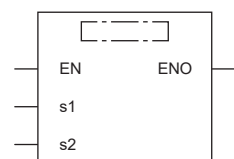
字符串比较

LD\$□、AND\$□、OR\$□

将字符串数据通过常开触点处理进行比较运算。

梯形图	ST
	<pre> ENO:=LDSTRING_□(EN, s1, s2); ENO:=ANDSTRING_□(EN, s1, s2); ENO:=ORSTRING_□(EN, s1, s2); (□中为EQ、NE、GT、LE、LT、GE。)*1 </pre>
<p>(□中输入\$=、\$<>、\$>、\$<=、\$<、\$>=。)</p>	

FBD/LD



(□中放入LDSTRING_、ANDSTRING_、ORSTRING_与EQ、NE、GT、LE、LT、GE的组合。)*1

*1 EQ为=、NE为<>、GT为>、LE为<=、LT为<、GE为>=。

■执行条件

指令	执行条件
LD\$□、AND\$□、OR\$□	常时执行

设置数据

■内容、范围、数据类型

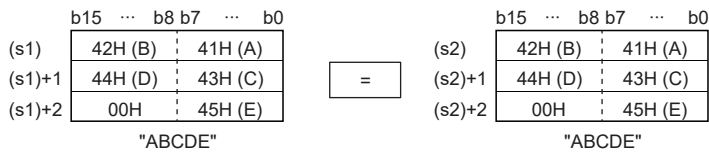
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	比较数据或存储了比较数据的起始软元件	—	字符串	ANYSTRING_SINGLE
(s2)	比较数据或存储了比较数据的起始软元件	—	字符串	ANYSTRING_SINGLE
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	—	—	○	—	—	—	○	—	—	○	—	—

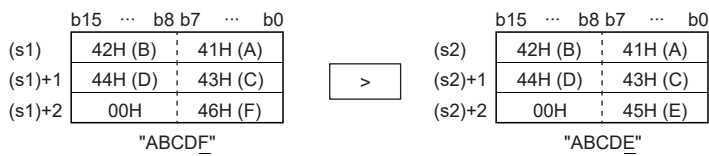
功能

- 将(s1)中指定的字符串数据与(s2)中指定的字符串数据通过常开触点处理进行比较运算。
- 比较运算时，将字符串的ASCII代码从字符串的起始处开始进行逐个字符比较。
- (s1)、(s2)的字符串的是从指定的软元件编号开始至存储了00H的软元件编号为止。
- 所有字符串一致的情况下，比较结果即为一致。



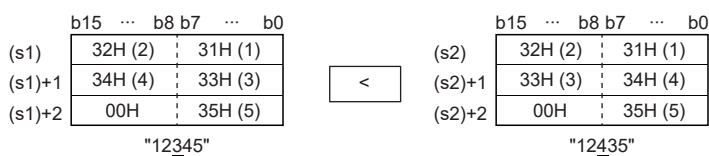
□内指令符号(梯形图、FBD/LD)	比较运算结果
\$=、EQ	导通状态 (ENO为ON)
\$<、NE	非导通状态 (ENO为OFF)
\$>、GT	非导通状态 (ENO为OFF)
\$<=、LE	导通状态 (ENO为ON)
\$<、LT	非导通状态 (ENO为OFF)
\$>=、GE	导通状态 (ENO为ON)

- 不同字符串的情况下，字符代码较大的字符串将变大。



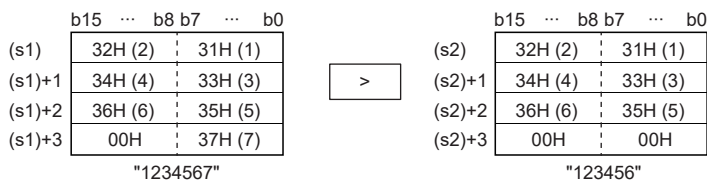
□内指令符号(梯形图、FBD/LD)	比较运算结果
\$=、EQ	非导通状态 (ENO为OFF)
\$<、NE	导通状态 (ENO为ON)
\$>、GT	导通状态 (ENO为ON)
\$<=、LE	非导通状态 (ENO为OFF)
\$<、LT	非导通状态 (ENO为OFF)
\$>=、GE	导通状态 (ENO为ON)

- 不同字符串的情况下，以首个不同字符代码的大小决定字符串的大小。



□内指令符号(梯形图、FBD/LD)	比较运算结果
\$=、EQ	非导通状态 (ENO为OFF)
\$<、NE	导通状态 (ENO为ON)
\$>、GT	非导通状态 (ENO为OFF)
\$<=、LE	导通状态 (ENO为ON)
\$<、LT	导通状态 (ENO为ON)
\$>=、GE	非导通状态 (ENO为OFF)

- (s1)与(s2)中的字符串数据的长度不同的情况下,较长的字符串数据将变大。



□内指令符号(梯形图、FBD/LD)	比较运算结果
\$=、EQ	非导通状态(ENO为OFF)
\$<、NE	导通状态(ENO为ON)
\$>、GT	导通状态(ENO为ON)
\$<=、LE	非导通状态(ENO为OFF)
\$<、LT	非导通状态(ENO为OFF)
\$>=、GE	导通状态(ENO为ON)

- (s1)、(s2)的字符串超过了16383字符的情况下,将变为非导通(ENO为OFF)。
- 通过FBD/LD使用LDSTRING_□指令的情况下,应将EN指定为左母线,或始终ON的变量部件/常数部件。
- 将ORSTRING_□指令通过FBD/LD使用的情况下,将EN指定为TRUE时ENO将ON。EN不成为执行条件。

出错

没有运算错误。

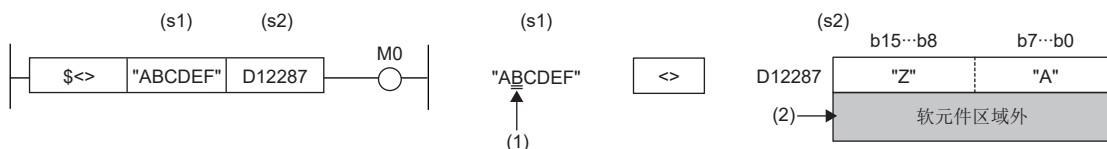
要点

在字符串比较指令中,在进行字符串比较的同时,还进行下述检查。

- 是否超出软元件区域范围的检查
- 字符串是否在16383字符以内的检查

因此,软元件区域内不存在00H的情况下,或字符串超过了16383字符的情况下,检测出字符不一致时,将在不变为非导通(ENO为OFF)的状况下输出比较运算结果。

下述示例表示软元件区域的最终软元件No.为D12287情况下的运算结果。



(1) (s1)的第2字符与(s2)不相同,(s1)≠(s2),因此运算结果为导通(ENO为ON)。

(2) D12287及其以后超出了软元件区域,因此对D12287为止的字符串数据实施比较处理。

检测出字符串的不一致,因此条件成立,结束处理。

字符串的合并

\$+(P) [操作数为2个的情况下]

连接字符串数据。

梯形图	ST
	不对应。 (☞ 704页 \$+(P) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 704页 \$+(P) [操作数为3个的情况下])

■执行条件

指令	执行条件
\$+	
\$+P	

设置数据

■内容、范围、数据类型

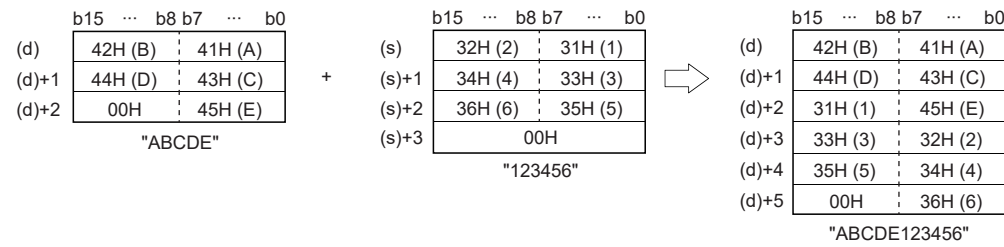
操作数	内容	范围	数据类型	数据类型(标签)
(s)	连接数据或存储了数据的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储了连接数据的起始软元件	—	字符串	ANYSTRING_SINGLE

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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	
(s)	—	—	○	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	○	—	—	—	—

功能

- 将(s)中指定的软元件编号及其以后存储的字符串连接到(d)中指定的软元件编号及其以后存储的字符串数据的后面，存储到(d)中指定的软元件编号及其以后。



- 字符串合并时，将忽略表示(d)中指定的字符串的结束的00H，在(d)的最终字符处连接(s)中指定的字符串。

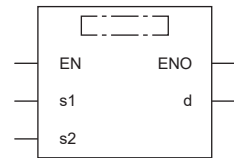
错误代码 (SD0)	内容
2820H	(s) 中指定的软元件编号及其以后, 相应软元件/标签的分配范围中不存在00H时。
	(d) 中指定的软元件编号及其以后, 相应软元件/标签的分配范围中不存在00H时。
2821H	(s) 与 (d) 中指定的字符串的存储软元件编号重复时。
3285H	(s) 的字符串超过了16383字符时。
	(d) 的字符串超过了16383字符时。
3286H	(s) + (d) 的字符串超过了16383字符时。
	(d) 中指定的软元件编号及其以后, 相应软元件/标签的分配范围中合并的字符串无法全部存储时。

\$+(P) [操作数为3个的情况下]

连接字符串数据。

梯形图	ST
	<pre>ENO:=STRINGPLUS(EN, s1, s2, d); ENO:=STRINGPLUSP(EN, s1, s2, d);</pre>

FBD/LD



(□中放入STRINGPLUS、STRINGPLUSP。)

■执行条件

指令	执行条件
\$+	
\$+P	

设置数据

■内容、范围、数据类型

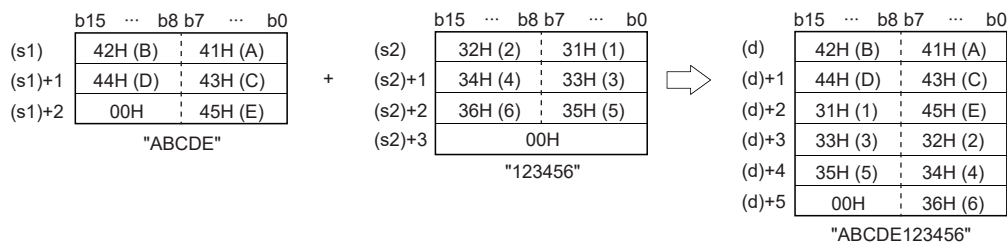
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	连接数据或存储了数据的起始软元件	—	字符串	ANYSTRING_SINGLE
(s2)	连接的数据或存储了连接的数据的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储连接结果的起始软元件	—	字符串	ANYSTRING_SINGLE
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 将(s2)中指定的软元件编号及其以后存储的字符串连接到(s1)中指定的软元件编号及其以后存储的字符串数据的后面，存储到(d)中指定的软元件编号及其以后。



- 字符串合并时，将忽略表示(s1)中指定的字符串的结束的00H，在(s1)的最终字符处连接(s2)中指定的字符串。

出错

错误代码(SD0)	内容
2820H	(s1)中指定的软元件编号及其以后，相应软元件/标签的分配范围中不存在00H时。
	(s2)中指定的软元件编号及其以后，相应软元件/标签的分配范围中不存在00H时。
2821H	(s2)与(d)中指定的字符串的存储软元件编号重复时。
3285H	(s1)的字符串超过了16383字符时。
	(s2)的字符串超过了16383字符时。
3286H	(d)中存储的字符串为超出可输出范围的数据时。
	• 合并后的字符串超过16383字符
	• 指定的软元件编号及其以后，相应软元件/标签的分配范围中合并的字符串无法全部存储

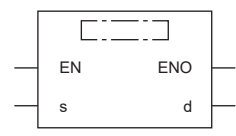
字符串传送

\$MOV(P)

将字符串数据传送到指定的软元件编号及其以后。

梯形图	ST
	ENO:=STRINGMOV(EN, s, d); ENO:=STRINGMOVP(EN, s, d);

FBD/LD



(□中放入STRINGMOV、STRINGMOVP。)

■执行条件

指令	执行条件
\$MOV	
\$MOVP	

设置数据

■内容、范围、数据类型

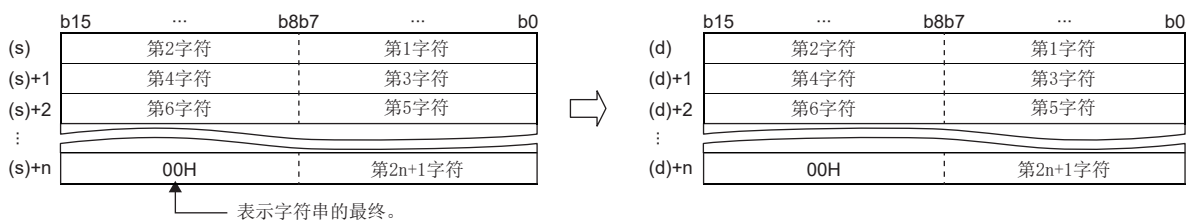
操作数	内容	范围	数据类型	数据类型(标签)
(s)	传送字符串(最大255字符)或存储了字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储传送字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

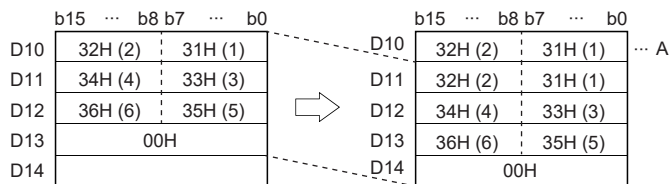
操作数	位		字		双字		间接指定	常数				其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K	H	E	
(s)	—	—	○	—	—	○	—	○	—	—	○	—
(d)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 将(s)中指定的字符串数据传送到(d)中指定的软元件编号及其以后。在字符串的传送中，对自(s)中指定的字符串或软元件编号开始至存储了00H的软元件编号为止的字符串进行一次传送。

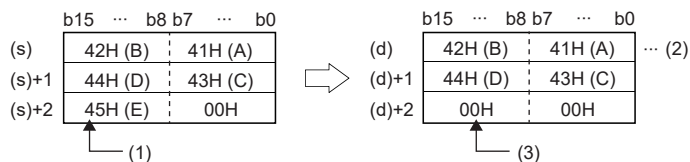


- 即使存储了传送的字符串数据的软元件范围(s)~(s)+n与存储传送后的字符串数据的软元件范围(d)~(d)+n重复的情况下，也将正常进行处理。例如，将D10~D13中存储的字符串传送到D11~D14中时，其情况如下所示。



A: 保持为传送前的字符串。

- (s)+n的低位字节中存储了00H的情况下，(d)+n的高位字节、低位字节均将存储00H。



- 不传送高位字节。
- 保持为传送前的字符串。
- 高位字节中自动存储00H。

出错

错误代码(SD0)	内容
2820H	(s)中指定的软元件编号及其以后，相应软元件/标签的分配范围中不存在00H时。
3285H	(s)的字符串超过了16383字符时。
3286H	(d)中指定的软元件编号及其以后，相应软元件/标签的分配范围中指定的字符串无法全部存储时。

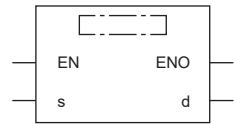
Unicode对应字符串传送

\$MOV(P)_WS

将Unicode字符串数据传送到指定的软元件编号及其以后。

梯形图	ST
	<pre>ENO:=STRINGMOV_WS(EN,s,d); ENO:=STRINGMOVP_WS(EN,s,d);</pre>

FBD/LD



(□中放入STRINGMOV_WS、STRINGMOVP_WS。)

■执行条件

指令	执行条件
\$MOV_WS	
\$MOVP_WS	

设置数据

■内容、范围、数据类型

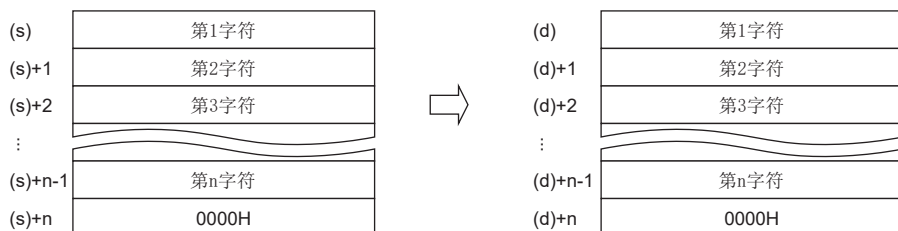
操作数	内容	范围	数据类型	数据类型(标签)
(s)	传送的Unicode字符串(最大255字符)或存储了Unicode字符串的起始软元件	—	Unicode字符串	ANYSTRING_DOUBLE
(d)	存储传送的Unicode字符串的起始软元件	—	Unicode字符串	ANYSTRING_DOUBLE
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—	

功能

- 将(s)中指定的Unicode字符串数据，传送到(d)中指定的软元件编号及其以后。在Unicode字符串的传送中，对自(s)中指定的Unicode字符串或软元件编号开始至存储了0000H的软元件编号为止的Unicode字符串进行一次传送。



- 即使存储了传送的Unicode字符串数据的软元件范围(s)~(s)+n与存储传送后的字符串数据的软元件范围(d)~(d)+n重复情况下，也将正常进行处理。例如，将D10~D13中存储的字符串传送到D11~D14中时，其情况如下所示。



A: 传送前的值的状态

出错

错误代码 (SD0)	内容
2820H	(s)中指定的软元件编号及其以后，相应软元件/标签的分配范围中不存在0000H时。
3285H	(s)的Unicode字符串超过了16383字符时。
3286H	(d)中指定的软元件编号及其以后，相应软元件/标签的分配范围中指定的Unicode字符串无法全部存储时。

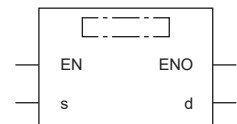
BIN16位数据→10进制ASCII转换

BINDA(P) (_U)

将BIN16位数据转换为10进制数ASCII代码。

梯形图	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



■执行条件

指令	执行条件
BINDA BINDA_U	
BINDAP BINDAP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s)	BINDA(P)	进行ASCII转换的BIN数据	-32768~32767	带符号BIN16位	ANY16_S
	BINDA(P)_U		0~65535	无符号BIN16位	ANY16_U
(d)	存储转换结果的起始软元件	—	字符串	ANYSTRING_SINGLE	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	—	—	○	—	—	—	—	○	—	—	—

功能

- 将(s)中指定的BIN16位数据，以10进制数表示时的数值转换为ASCII代码后，存储到(d)中指定的软元件编号及其以后。
- 根据SM705(变数位数切换)的状态，(d)中存储的10进制ASCII数据的格式将有所不同。

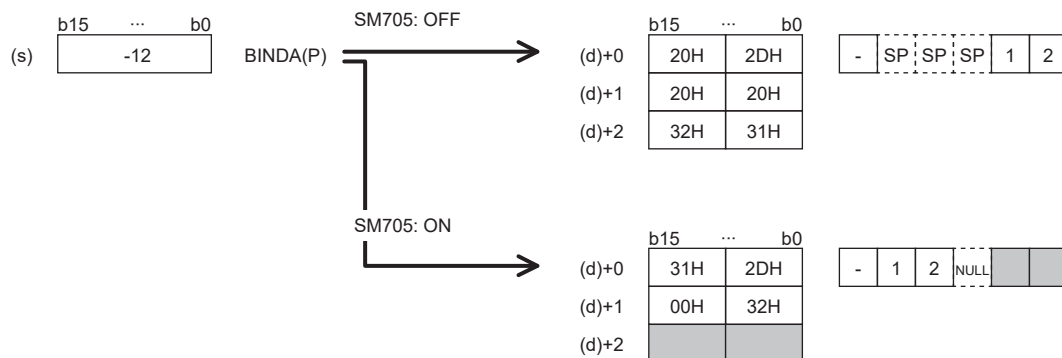
SM705的状态	(d)的存储形式	参阅目标
OFF	以固定位数(符号+5位)存储	712页 SM705(变数位数切换)为OFF时的动作
ON	根据(s)值，向前填充各位存储	712页 SM705(变数位数切换)为ON时的动作

■动作的概要

当SM705(变数位数切换)为OFF、ON时，其各自的动作如下所示。

例

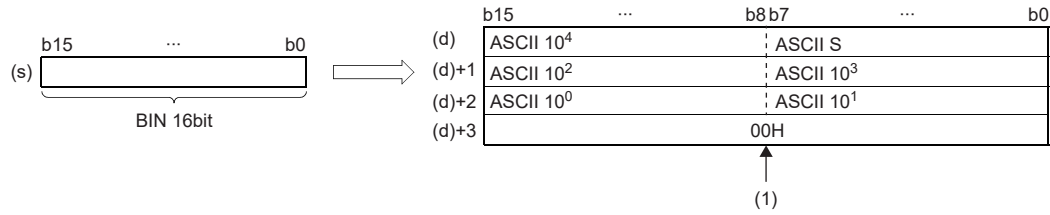
在(s)中存储有数值“-12”的状态下执行BINDA(P)指令的情况下



- SM705(变数位数切换)为OFF时，位数将固定。第1字符为符号，上述的示例将变为2DH(-) ((s)为0或正的数时，第1字符将变为20H(空白))。数值部分向右填充存储。数值部分不满5位时，高位的位的ASCII代码将变为20H(空白)。
- SM705(变数位数切换)为ON时将向前填充。数值部分不满5位时，终端将存储00H。

■SM705 (变数位数切换) 为OFF时的动作

10进制ASCII数据将以固定位数存储至 (d) ~ (d)+2。



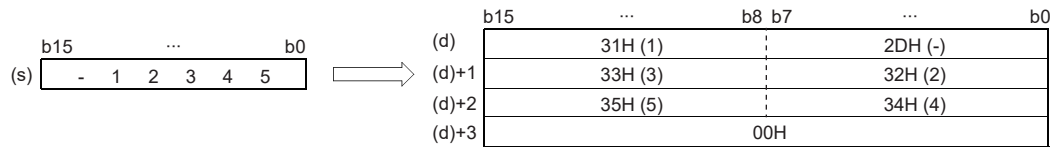
ASCII S: 符号的ASCII代码*1
 ASCII 10⁴: 万位的ASCII代码*2
 ASCII 10³: 千位的ASCII代码*3
 ASCII 10²: 百位的ASCII代码*4
 ASCII 10¹: 十位的ASCII代码*5
 ASCII 10⁰: 个位的ASCII代码

(1) SM701 (输出字符数切换) 为OFF时 (d)+3将存储00H, 为ON时则不变化。

- *1 为0或正的数时存储20H (空白), 为负的数时存储2DH (-)。
- *2 数值部分为4位或其以下时, 在ASCII 10⁴中存储20H (空白)。
- *3 数值部分为3位或其以下时, 在ASCII 10³中存储20H (空白)。
- *4 数值部分为2位或其以下时, 在ASCII 10²中存储20H (空白)。
- *5 数值部分为1位时, 在ASCII 10¹中存储20H (空白)。

例

BINDA (P) 指令中, 在 (s) 中指定-12345时



■SM705 (变数位数切换) 为ON时的动作

10进制ASCII数据将向前填充存储至 (d)。

(s) 的值与 (d) 中存储的值的示例如下所示。

(s) 的值	(d) ~ (d)+2 的内容	(s) 的值	(d) ~ (d)+2 的内容																																																						
<ul style="list-style-type: none"> • 0 • 正的数 (数值部分 1位) 	<ul style="list-style-type: none"> • (d) 的高位字节中将存储00H。 • (d)+1 ~ (d)+2 不变化。 <table border="1"> <tr> <td>(d)</td> <td>b15</td> <td>...</td> <td>b8 b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td></td> <td colspan="2">00H</td> <td colspan="3">ASCII 10⁰</td> </tr> <tr> <td>(d)+1</td> <td colspan="5" style="background-color: #cccccc;"></td> </tr> <tr> <td>(d)+2</td> <td colspan="5" style="background-color: #cccccc;"></td> </tr> </table>	(d)	b15	...	b8 b7	...	b0		00H		ASCII 10 ⁰			(d)+1						(d)+2						<ul style="list-style-type: none"> • 正的数 (数值部分 2位) • 负的数 (数值部分 1位) 	<ul style="list-style-type: none"> • (d)+1 中将存储00H。 • (d)+2 不变化。 <table border="1"> <tr> <td>(d)</td> <td>b15</td> <td>...</td> <td>b8 b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td></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" style="background-color: #cccccc;"></td> </tr> </table>	(d)	b15	...	b8 b7	...	b0		ASCII 10 ⁰		ASCII 10 ¹ / 2DH (-)			(d)+1	00H					(d)+2											
(d)	b15	...	b8 b7	...	b0																																																				
	00H		ASCII 10 ⁰																																																						
(d)+1																																																									
(d)+2																																																									
(d)	b15	...	b8 b7	...	b0																																																				
	ASCII 10 ⁰		ASCII 10 ¹ / 2DH (-)																																																						
(d)+1	00H																																																								
(d)+2																																																									
<ul style="list-style-type: none"> • 正的数 (数值部分 5位) • 负的数 (数值部分 4位) 	<ul style="list-style-type: none"> • (d)+2 的高位字节中将存储00H。 <table border="1"> <tr> <td>(d)</td> <td>b15</td> <td>...</td> <td>b8 b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td></td> <td colspan="2">ASCII 10³</td> <td colspan="3">ASCII 10⁴ / 2DH (-)</td> </tr> <tr> <td>(d)+1</td> <td colspan="2">ASCII 10¹</td> <td colspan="3">ASCII 10²</td> </tr> <tr> <td>(d)+2</td> <td colspan="2">00H</td> <td colspan="3">ASCII 10⁰</td> </tr> </table>	(d)	b15	...	b8 b7	...	b0		ASCII 10 ³		ASCII 10 ⁴ / 2DH (-)			(d)+1	ASCII 10 ¹		ASCII 10 ²			(d)+2	00H		ASCII 10 ⁰			<ul style="list-style-type: none"> • 负的数 (数值部分 5位) 	<ul style="list-style-type: none"> (1) 仅在SM701 (输出字符数切换) 为OFF时, 于 (d)+3 中存储00H。 <table border="1"> <tr> <td>(d)</td> <td>b15</td> <td>...</td> <td>b8 b7</td> <td>...</td> <td>b0</td> </tr> <tr> <td></td> <td colspan="2">ASCII 10⁴</td> <td colspan="3">2DH (-)</td> </tr> <tr> <td>(d)+1</td> <td colspan="2">ASCII 10²</td> <td colspan="3">ASCII 10³</td> </tr> <tr> <td>(d)+2</td> <td colspan="2">ASCII 10⁰</td> <td colspan="3">ASCII 10¹</td> </tr> <tr> <td>(d)+3</td> <td colspan="5">00H</td> </tr> </table> <p style="text-align: center;">↑ (1)</p>	(d)	b15	...	b8 b7	...	b0		ASCII 10 ⁴		2DH (-)			(d)+1	ASCII 10 ²		ASCII 10 ³			(d)+2	ASCII 10 ⁰		ASCII 10 ¹			(d)+3	00H				
(d)	b15	...	b8 b7	...	b0																																																				
	ASCII 10 ³		ASCII 10 ⁴ / 2DH (-)																																																						
(d)+1	ASCII 10 ¹		ASCII 10 ²																																																						
(d)+2	00H		ASCII 10 ⁰																																																						
(d)	b15	...	b8 b7	...	b0																																																				
	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: 万位的ASCII代码
 ASCII 10³: 千位的ASCII代码
 ASCII 10²: 百位的ASCII代码
 ASCII 10¹: 十位的ASCII代码
 ASCII 10⁰: 个位的ASCII代码

- 运算位数未满足最大位数 (符号+5位) 时, 无论SM701 (输出字符数切换) 为ON或OFF, 字符串的终端皆存储00H。字符串的终端为低位字节时, 高位字节也存储00H。
- 运算位数为最大位数 (符号+5位) 时, SM701 (输出字符数切换) 为OFF时, (d)+3 中将存储00H。SM701 (输出字符数切换) 为ON时, (d)+3 不变化。

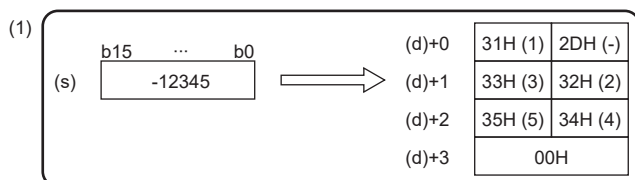
出错

没有运算错误。

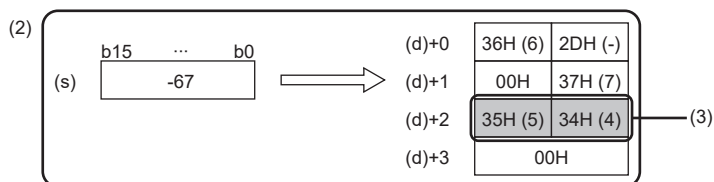
注意事项

SM705(变数位数切换)为ON时, (d)中仅存储有效位数的运算结果。因此, 连续执行BINDA(P) (_U)指令, 将运算结果存储到同一软元件时, (d)的一部分可能未被前次的运算结果成功覆盖而残留下来。

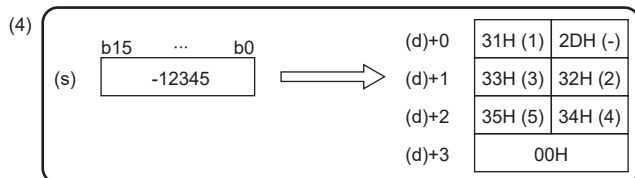
[例] 在(s)为“-12345H”时执行BINDA(P)指令、(s)为“-67H”时执行BINDA(P)指令的示例



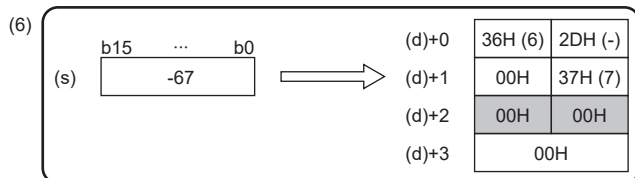
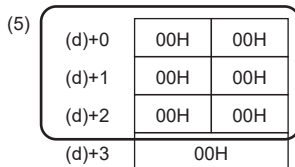
- (1) 将“-12345”转换为字符串
- (2) 将“-67”转换为字符串
- (3) 前次转换结果的一部分将残留至(d)+2中



若希望避免此情况, 应将存储区域(d)+0~(d)+2全部清除后, 创建执行BINDA(P) (_U)指令的程序。



- (4) 将“-12345”转换为字符串
- (5) 清除(d)+0~(d)+2
- (6) 将“-67”转换为字符串



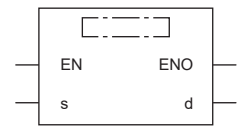
BIN32位数据→10进制ASCII转换

DBINDA(P) (_U)

将BIN32位数据转换为10进制数ASCII代码。

梯形图	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



■执行条件

指令	执行条件
DBINDA DBINDA_U	
DBINDAP DBINDAP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s)	DBINDA (P)	进行ASCII转换的BIN数据	-2147483648~2147483647	带符号BIN32位	ANY32_S
	DBINDA (P)_U		0~4294967295	无符号BIN32位	ANY32_U
(d)	存储转换结果的起始软元件	—	字符串	ANYSTRING_SINGLE	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 将(s)中指定的BIN32位数据，以10进制数表示时的数值转换为ASCII代码后，存储到(d)中指定的软元件编号及其以后。
- 根据SM705(变数位数切换)的状态，(d)中存储的10进制ASCII数据的格式将有所不同。

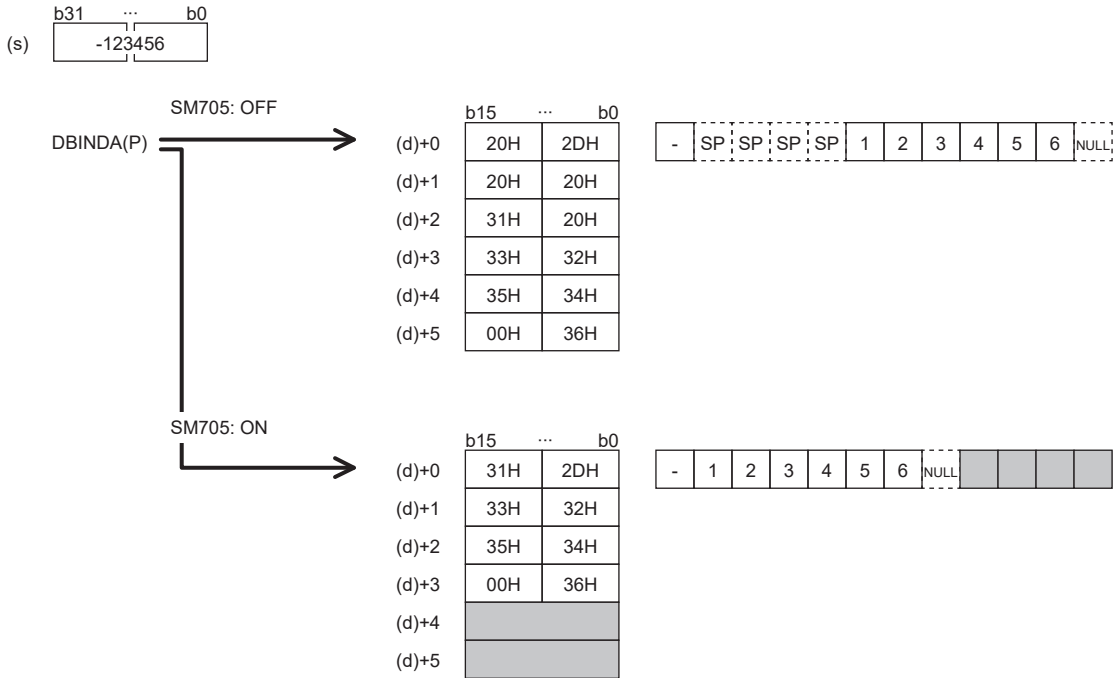
SM705的状态	(d)的存储形式	参阅目标
OFF	以固定位数(符号+10位)存储	716页 SM705(变数位数切换)为OFF时的动作
ON	根据(s)值，向前填充各位存储	717页 SM705(变数位数切换)为ON时的动作

■动作的概要

当SM705(变数位数切换)为OFF、ON时，其各自的动作如下所示。

例

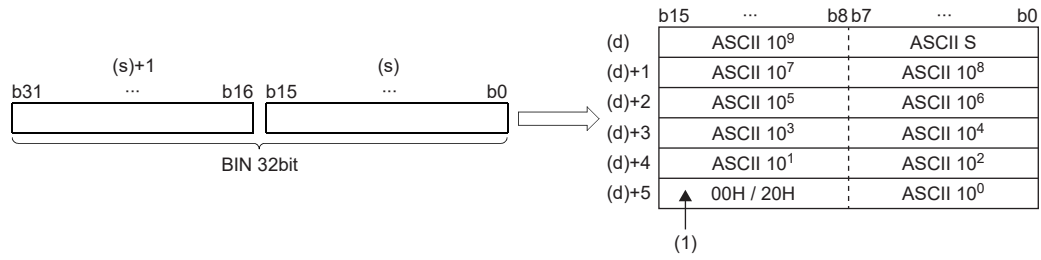
在(s)中存储有数值“-123456”的状态下执行DBINDA(P)指令的情况下



- SM705(变数位数切换)为OFF时，位数将固定。第1字符为符号，上述的示例将变为2DH(-) ((s)为0或正的数时，第1字符将变为20H(空白)。)。数值部分向右填充存储。数值部分不满10位时，高位的位的ASCII代码将变为20H(空白)。
- SM705(变数位数切换)为ON时将向前填充。数值部分不满10位时，终端将存储00H。

■SM705(变数位数切换)为OFF时的动作

10进制ASCII数据将以固定位数存储至(d)~(d)+5。



ASCII S: 符号的ASCII代码*1

ASCII 10⁹: 十亿位的ASCII代码*2

ASCII 10⁸: 亿位的ASCII代码*3

⋮

ASCII 10¹: 十位的ASCII代码*4

ASCII 10⁰: 个位的ASCII代码

(1) SM701(输出字符数切换)为OFF时(d)+5的高位字节将存储00H, 为ON时则存储20H(空白)。

*1 为0或正的数时存储20H(空白), 为负的数时存储2DH(-)。

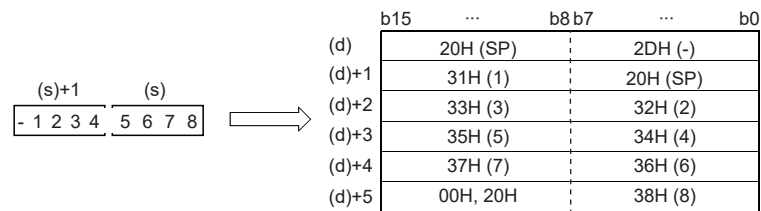
*2 数值部分为9位或其以下时, 在ASCII 10⁹中存储20H(空白)。

*3 数值部分为8位或其以下时, 在ASCII 10⁸中存储20H(空白)。

*4 数值部分为1位时, 在ASCII 10¹中存储20H(空白)。

例

(s)中指定了-12345678的情况下(指定了带符号的情况下)



■SM705(变数位数据切换)为ON时的动作

10进制ASCII数据将向前填充存储至(d)。

(s)的值与(d)中存储的值的示例如下所示。

(s)的值	(d)~(d)+2的内容	(s)的值	(d)~(d)+2的内容																																																																											
<ul style="list-style-type: none"> 0 正的数(数值部分1位) 	<ul style="list-style-type: none"> (d)的高位字节中将存储00H。 (d)+1及其以后不变化。 <table border="1"> <thead> <tr> <th>b15</th> <th>...</th> <th>b8 b7</th> <th>...</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td colspan="2">00H</td> <td>:</td> <td colspan="2">ASCII 10⁰</td> </tr> <tr> <td colspan="5">(d)+1</td> </tr> <tr> <td colspan="5">(d)+2</td> </tr> <tr> <td colspan="5">(d)+3</td> </tr> <tr> <td colspan="5">(d)+4</td> </tr> <tr> <td colspan="5">(d)+5</td> </tr> </tbody> </table>	b15	...	b8 b7	...	b0	00H		:	ASCII 10 ⁰		(d)+1					(d)+2					(d)+3					(d)+4					(d)+5					<ul style="list-style-type: none"> 正的数(数值部分2位) 负的数(数值部分1位) 	<ul style="list-style-type: none"> (d)+1中将存储00H。 (d)+2及其以后不变化。 <table border="1"> <thead> <tr> <th>b15</th> <th>...</th> <th>b8 b7</th> <th>...</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td colspan="2">ASCII 10⁰</td> <td>:</td> <td colspan="2">ASCII 10¹ / 2DH (-)</td> </tr> <tr> <td colspan="5">(d)+1</td> </tr> <tr> <td colspan="5">00H</td> </tr> <tr> <td colspan="5">(d)+2</td> </tr> <tr> <td colspan="5">(d)+3</td> </tr> <tr> <td colspan="5">(d)+4</td> </tr> <tr> <td colspan="5">(d)+5</td> </tr> </tbody> </table>	b15	...	b8 b7	...	b0	ASCII 10 ⁰		:	ASCII 10 ¹ / 2DH (-)		(d)+1					00H					(d)+2					(d)+3					(d)+4					(d)+5				
b15	...	b8 b7	...	b0																																																																										
00H		:	ASCII 10 ⁰																																																																											
(d)+1																																																																														
(d)+2																																																																														
(d)+3																																																																														
(d)+4																																																																														
(d)+5																																																																														
b15	...	b8 b7	...	b0																																																																										
ASCII 10 ⁰		:	ASCII 10 ¹ / 2DH (-)																																																																											
(d)+1																																																																														
00H																																																																														
(d)+2																																																																														
(d)+3																																																																														
(d)+4																																																																														
(d)+5																																																																														
<ul style="list-style-type: none"> 正的数(数值部分9位) 负的数(数值部分8位) 	<ul style="list-style-type: none"> (d)+4的高位字节中将存储00H。 (d)+5及其以后不变化。 <table border="1"> <thead> <tr> <th>b15</th> <th>...</th> <th>b8 b7</th> <th>...</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td colspan="2">ASCII 10⁷</td> <td>:</td> <td colspan="2">ASCII 10⁸ / 2DH (-)</td> </tr> <tr> <td colspan="2">ASCII 10⁵</td> <td>:</td> <td colspan="2">ASCII 10⁶</td> </tr> <tr> <td colspan="2">ASCII 10³</td> <td>:</td> <td colspan="2">ASCII 10⁴</td> </tr> <tr> <td colspan="2">ASCII 10¹</td> <td>:</td> <td colspan="2">ASCII 10²</td> </tr> <tr> <td colspan="2">00H</td> <td>:</td> <td colspan="2">ASCII 10⁰</td> </tr> <tr> <td colspan="5">(d)+5</td> </tr> </tbody> </table>	b15	...	b8 b7	...	b0	ASCII 10 ⁷		:	ASCII 10 ⁸ / 2DH (-)		ASCII 10 ⁵		:	ASCII 10 ⁶		ASCII 10 ³		:	ASCII 10 ⁴		ASCII 10 ¹		:	ASCII 10 ²		00H		:	ASCII 10 ⁰		(d)+5					<ul style="list-style-type: none"> 正的数(数值部分10位) 负的数(数值部分9位) 	<ul style="list-style-type: none"> (d)+5中将存储00H。 <table border="1"> <thead> <tr> <th>b15</th> <th>...</th> <th>b8 b7</th> <th>...</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td colspan="2">ASCII 10⁸</td> <td>:</td> <td colspan="2">ASCII 10⁹ / 2DH (-)</td> </tr> <tr> <td colspan="2">ASCII 10⁶</td> <td>:</td> <td colspan="2">ASCII 10⁷</td> </tr> <tr> <td colspan="2">ASCII 10⁴</td> <td>:</td> <td colspan="2">ASCII 10⁵</td> </tr> <tr> <td colspan="2">ASCII 10²</td> <td>:</td> <td colspan="2">ASCII 10³</td> </tr> <tr> <td colspan="2">ASCII 10⁰</td> <td>:</td> <td colspan="2">ASCII 10¹</td> </tr> <tr> <td colspan="5">(d)+5</td> </tr> <tr> <td colspan="5">00H</td> </tr> </tbody> </table>	b15	...	b8 b7	...	b0	ASCII 10 ⁸		:	ASCII 10 ⁹ / 2DH (-)		ASCII 10 ⁶		:	ASCII 10 ⁷		ASCII 10 ⁴		:	ASCII 10 ⁵		ASCII 10 ²		:	ASCII 10 ³		ASCII 10 ⁰		:	ASCII 10 ¹		(d)+5					00H				
b15	...	b8 b7	...	b0																																																																										
ASCII 10 ⁷		:	ASCII 10 ⁸ / 2DH (-)																																																																											
ASCII 10 ⁵		:	ASCII 10 ⁶																																																																											
ASCII 10 ³		:	ASCII 10 ⁴																																																																											
ASCII 10 ¹		:	ASCII 10 ²																																																																											
00H		:	ASCII 10 ⁰																																																																											
(d)+5																																																																														
b15	...	b8 b7	...	b0																																																																										
ASCII 10 ⁸		:	ASCII 10 ⁹ / 2DH (-)																																																																											
ASCII 10 ⁶		:	ASCII 10 ⁷																																																																											
ASCII 10 ⁴		:	ASCII 10 ⁵																																																																											
ASCII 10 ²		:	ASCII 10 ³																																																																											
ASCII 10 ⁰		:	ASCII 10 ¹																																																																											
(d)+5																																																																														
00H																																																																														
负的数(数值部分10位)	<ul style="list-style-type: none"> SM701(输出字符串切换)为OFF时, (d)+5的高位字节(1)将存储00H, 为ON时则存储20H(空白)。 <table border="1"> <thead> <tr> <th>b15</th> <th>...</th> <th>b8 b7</th> <th>...</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td colspan="2">ASCII 10⁹</td> <td>:</td> <td colspan="2">2DH (-)</td> </tr> <tr> <td colspan="2">ASCII 10⁷</td> <td>:</td> <td colspan="2">ASCII 10⁸</td> </tr> <tr> <td colspan="2">ASCII 10⁵</td> <td>:</td> <td colspan="2">ASCII 10⁶</td> </tr> <tr> <td colspan="2">ASCII 10³</td> <td>:</td> <td colspan="2">ASCII 10⁴</td> </tr> <tr> <td colspan="2">ASCII 10¹</td> <td>:</td> <td colspan="2">ASCII 10²</td> </tr> <tr> <td colspan="2">00H / 20H</td> <td>:</td> <td colspan="2">ASCII 10⁰</td> </tr> <tr> <td colspan="5">(d)+5</td> </tr> </tbody> </table> <p>(1)</p>	b15	...	b8 b7	...	b0	ASCII 10 ⁹		:	2DH (-)		ASCII 10 ⁷		:	ASCII 10 ⁸		ASCII 10 ⁵		:	ASCII 10 ⁶		ASCII 10 ³		:	ASCII 10 ⁴		ASCII 10 ¹		:	ASCII 10 ²		00H / 20H		:	ASCII 10 ⁰		(d)+5					ASCII 10 ⁹ : 十亿位的ASCII代码 ASCII 10 ⁸ : 一亿位的ASCII代码 ⋮ ASCII 10 ¹ : 十位的ASCII代码 ASCII 10 ⁰ : 个位的ASCII代码																																				
b15	...	b8 b7	...	b0																																																																										
ASCII 10 ⁹		:	2DH (-)																																																																											
ASCII 10 ⁷		:	ASCII 10 ⁸																																																																											
ASCII 10 ⁵		:	ASCII 10 ⁶																																																																											
ASCII 10 ³		:	ASCII 10 ⁴																																																																											
ASCII 10 ¹		:	ASCII 10 ²																																																																											
00H / 20H		:	ASCII 10 ⁰																																																																											
(d)+5																																																																														

- 运算位数未满足最大位数(符号+10位)时, 无论SM701(输出字符串切换)为ON或OFF, 字符串的终端皆存储00H。字符串的终端为低位字节时, 高位字节也存储00H。
- 运算位数为最大位数(符号+10位)的情况下, 在SM701(输出字符串切换)为OFF时, (d)+5的高位字节中将存储00H。SM701(输出字符串切换)为ON时在(d)+5的高位字节存储20H(空白)。

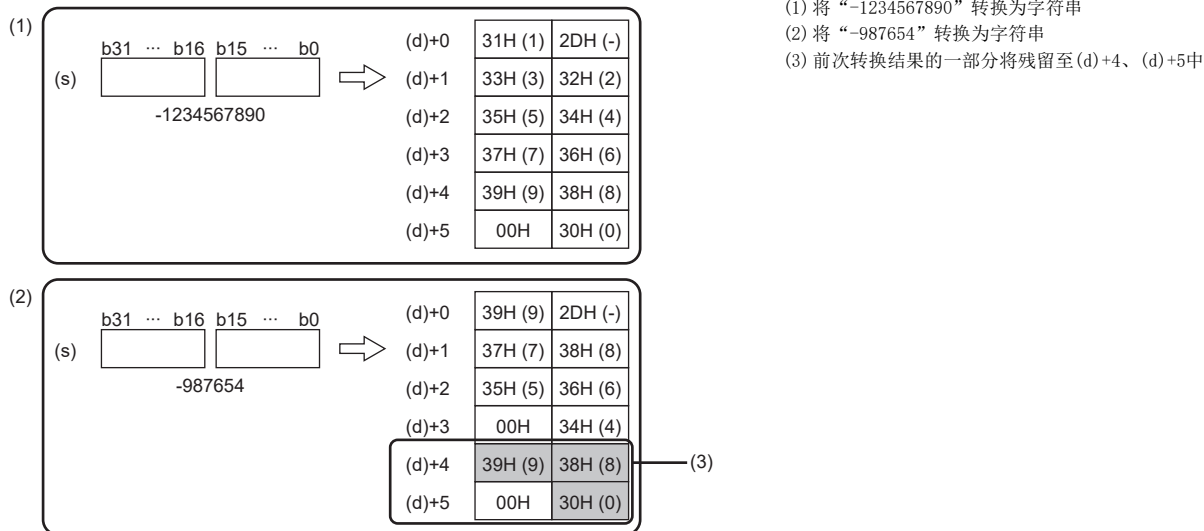
出错

没有运算错误。

注意事项

SM705(变数位数切换)为ON时，(d)中仅存储有效位数的运算结果。因此，连续执行DBINDA(P) (_U)指令，将运算结果存储到同一软元件时，(d)的一部分可能未被前次的运算结果成功覆盖而残留下来。

[例] 在(s)为“-1234567890”时执行DBINDA(P)指令、(s)为“-987654H”时执行DBINDA(P)指令的示例

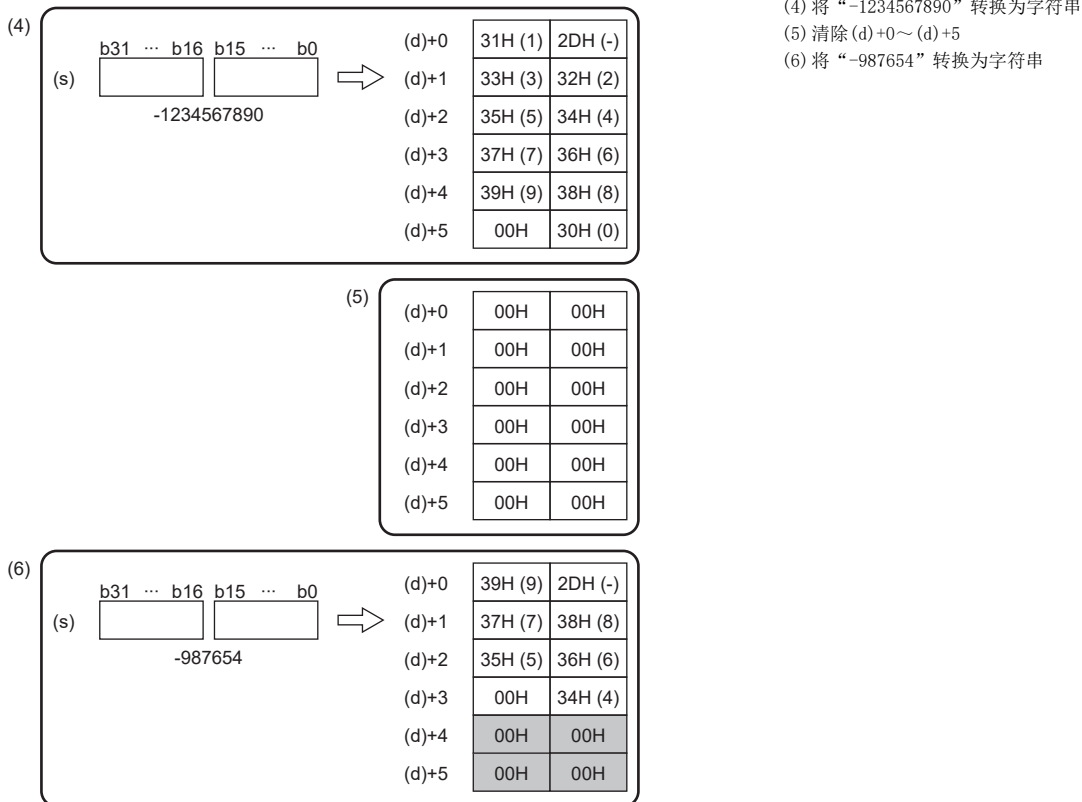


(1) 将“-1234567890”转换为字符串

(2) 将“-987654”转换为字符串

(3) 前次转换结果的一部分将残留至(d)+4、(d)+5中

若希望避开此情况，应将存储区域(d)+0~(d)+5全部清除后，创建执行DBINDA(P) (_U)指令的程序。



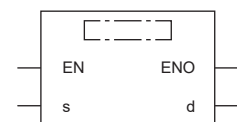
BIN16位数据→16进制ASCII转换

BINHA (P)

将BIN16位数据转换为16进制数ASCII代码。

梯形图	ST
	ENO:=BINHA (EN, s, d) ; ENO:=BINHAP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
BINHA	
BINHAP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行ASCII转换的BIN数据	-32768~32767	带符号BIN16位	ANY16
(d)	存储转换结果的起始软元件	—	字符串	ANYSTRING_SINGLE
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 将(s)中指定的BIN16位数据，以16进制数表示时的数值转换为ASCII代码后，存储到(d)中指定的软元件编号及其以后。
- 根据SM705(变数位数切换)的状态，(d)中存储的16进制ASCII数据的格式将有所不同。

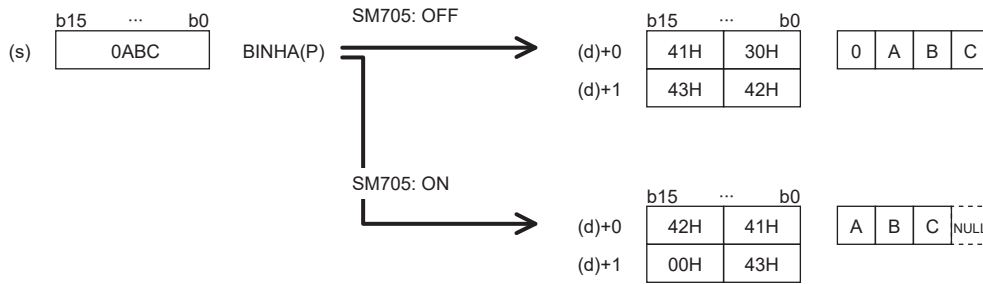
SM705的状态	(d)的存储形式	参阅目标
OFF	以固定位数(4位)存储	721页 SM705(变数位数切换)为OFF时的动作
ON	根据(s)值，向前填充各位存储	721页 SM705(变数位数切换)为ON时的动作

■动作的概要

当SM705(变数位数切换)为OFF、ON时，其各自的动作如下所示。

例

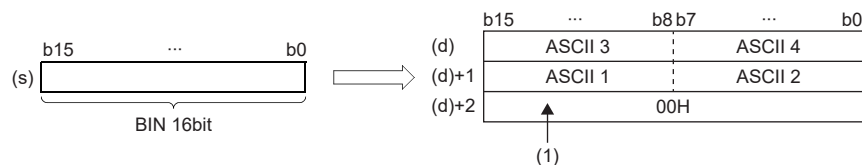
在(s)中存储有BIN16位数据“0ABC”的状态下执行BINHA(P)指令时



- SM705(变数位数切换)为OFF时，位数将固定。将“0ABC”的4位转换为ASCII数据存储。
- SM705(变数位数切换)为ON时将向前填充。将省略“0ABC”的起始0的“ABC”转换为ASCII数据存储，并在终端存储00H。

■SM705(变数位数切换)为OFF时的动作

16进制ASCII数据将以固定位数(4位)存储至(d)。



ASCII □: 第□位的ASCII代码

(1) SM701(输出字符数切换)为OFF时(d)+2将存储00H, 为ON时则不变化。

• (d)中存储的运算结果将被处理为4位的16进制数。因此,有效位数的左侧的0将被作为“0”处理。(进行0填充。)

例

(s)中指定了02A6H的情况下



■SM705(变数位数切换)为ON时的动作

将省略有效位数左侧0的位数(最大4位)的16进制ASCII代码向前填充, 储存在(d)中。

(s)的值与(d)中存储的值的示例如下所示。

(s)的值	(d)~(d)+1的内容	(s)的值	(d)~(d)+1的内容
0H~FH	<ul style="list-style-type: none"> (d)的高位字节中将存储00H。 (d)+1及其以后不变化。 	10H~FFH	(d)+1中将存储00H。
100H~FFFH	(d)+1的高位字节中将存储00H。	1000H~FFFFH	(1) SM701(输出字符数切换)为OFF时, (d)+2中将存储00H。SM701(输出字符数切换)为ON时, (d)+2不变化。

ASCII □: 第□位的ASCII代码

- 位数未满足最大位数(4位)时, 无论SM701(输出字符数切换)为ON或OFF, 字符串的终端将存储00H。字符串的终端为低位字节时, 高位字节也存储00H。
- 位数为最大位数(4位)时, SM701(输出字符数切换)为OFF时, (d)+2中将存储00H。SM701(输出字符数切换)为ON时, (d)+2不变化。

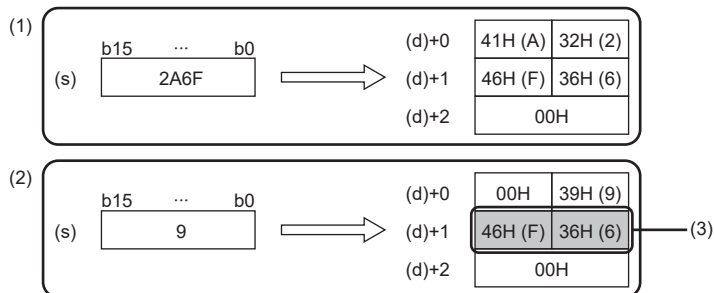
出错

没有运算错误。

注意事项

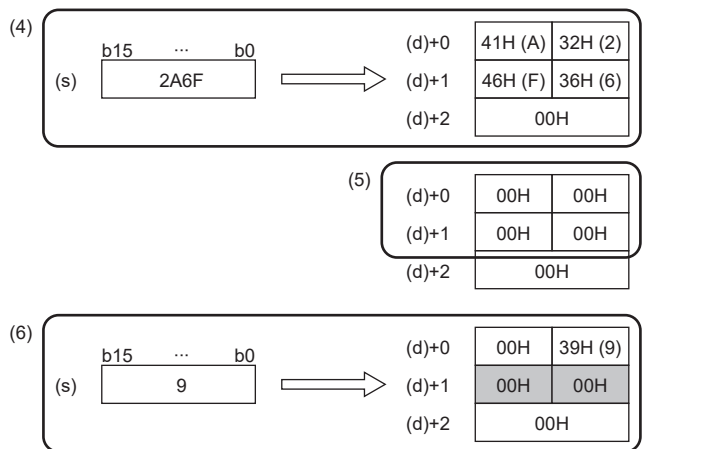
SM705(变數位數切换)为ON时, (d)中仅存储有效位数的运算结果。因此, 连续执行BINHA(P)指令, 将运算结果存储到同一软元件时, (d)的一部分可能未被前次的运算结果成功覆盖而残留下来。

[例] 在(s)为“2A6F”时执行BINHA(P)指令、(s)为“9”时执行BINHA(P)指令的示例



- (1) 将“2A6F”转换为字符串
- (2) 将“9”转换为字符串
- (3) 前次转换结果的一部分将残留至(d)+1中

若希望避开此情况, 应将存储区域(d)+0~(d)+1全部清除后, 创建执行BINHA(P)指令的程序。



- (4) 将“2A6F”转换为字符串
- (5) 清除(d)+0~(d)+1
- (6) 将“9”转换为字符串

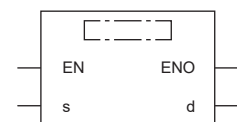
BIN32位数据→16进制ASCII转换

DBINHA (P)

将BIN32位数据转换为16进制数ASCII代码。

梯形图	ST
	ENO:=DBINHA (EN, s, d) ; ENO:=DBINHAP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
DBINHA	
DBINHAP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行ASCII转换的BIN数据	-2147483648~2147483647	带符号BIN32位	ANY32
(d)	存储转换结果的起始软元件	—	字符串	ANYSTRING_SINGLE
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 将(s)中指定的BIN32位数据，以16进制数表示时的数值转换为ASCII代码后，存储到(d)中指定的软元件编号及其以后。
- 根据SM705(变数位数切换)的状态，(d)中存储的10进制ASCII数据的格式将有所不同。

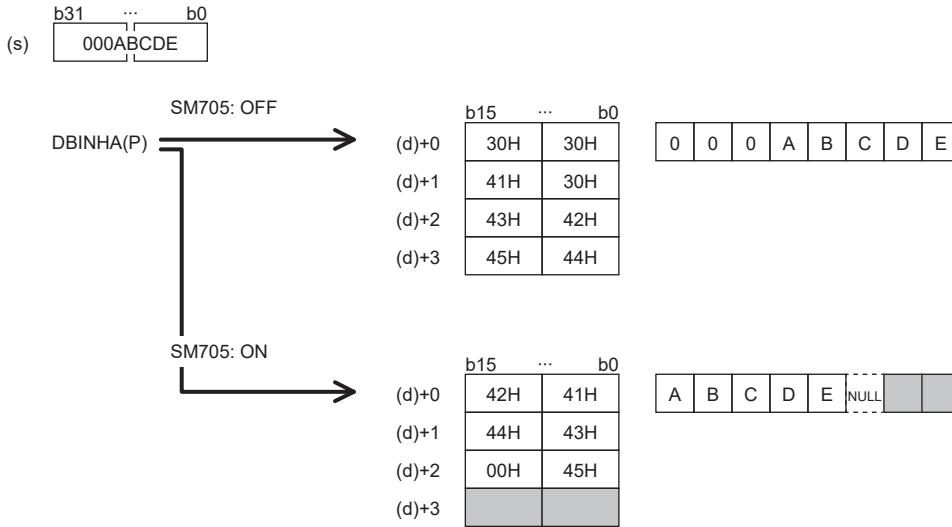
SM705的状态	(d)的存储形式	参阅目标
OFF	以固定位数(8位)存储	725页 SM705(变数位数切换)为OFF时的动作
ON	根据(s)值，向前填充各位存储	725页 SM705(变数位数切换)为ON时的动作

■动作的概要

当SM705(变数位数切换)为OFF、ON时，其各自的动作如下所示。

例

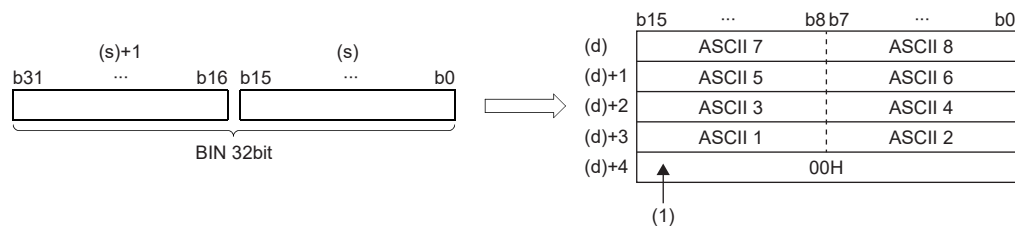
在(s)中存储有BIN32位数据“000ABCDEH”的状态下执行DBINHA(P)指令时



- SM705(变数位数切换)为OFF时，位数将固定。将“000ABCDE”的8位转换为ASCII数据存储。
- SM705(变数位数切换)为ON时将向前填充。仅将“000ABCDE”的有效位数部分“ABCDE”转换为ASCII数据存储，并在终端存储00H。

■SM705(变数位数切换)为OFF时的动作

16进制ASCII数据将以固定位数(8位)存储至(d)~(d)+3。



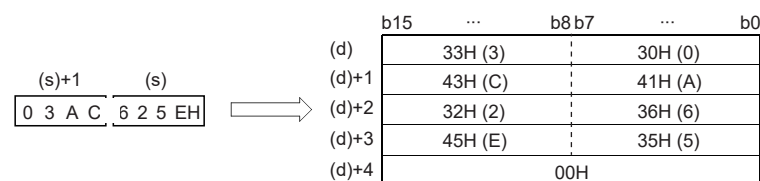
ASCII □: 第□位的ASCII代码

(1) SM701(输出字符数切换)为OFF时(d)+4将存储00H, 为ON时则不变化。

- (d)中存储的运算结果将被处理为8位的16进制数。因此,有效位数的左侧的0将被作为“0”处理。(进行0填充。)

例

(s)中指定了03AC625EH的情况下



■SM705(变数位数切换)为ON时的动作

将省略有效位数左侧0的位数(最大8位)的16进制ASCII代码向前填充, 储存到(d)中。

(s)的值与(d)中存储的值的示例如下所示。

(s)的值	(d)~(d)+3的内容	(s)的值	(d)~(d)+3的内容
0H~FH	<ul style="list-style-type: none"> • (d)的高位字节中将存储00H。 • (d)+1及其以后不变化。 	10H~FFH	<ul style="list-style-type: none"> • (d)+1中将存储00H。 • (d)+2及其以后不变化。
1000000H~FFFFFFFH	<ul style="list-style-type: none"> • (d)+3的高位字节中将存储00H。 	10000000H~FFFFFFFH	<p>(1) SM701(输出字符数切换)为OFF时, (d)+4中将存储00H。SM701(输出字符数切换)为ON时, (d)+4不变化。</p>

ASCII □: 第□位的ASCII代码

- 位数未满足最大位数(8位)时, 无论SM701(输出字符数切换)为ON或OFF, 字符串的终端将存储00H。字符串的终端为低位字节时, 高位字节也存储00H。
- 位数为最大位数(8位)时, SM701(输出字符数切换)为OFF时, (d)+4中将存储00H。SM701(输出字符数切换)为ON时, (d)+4不变化。

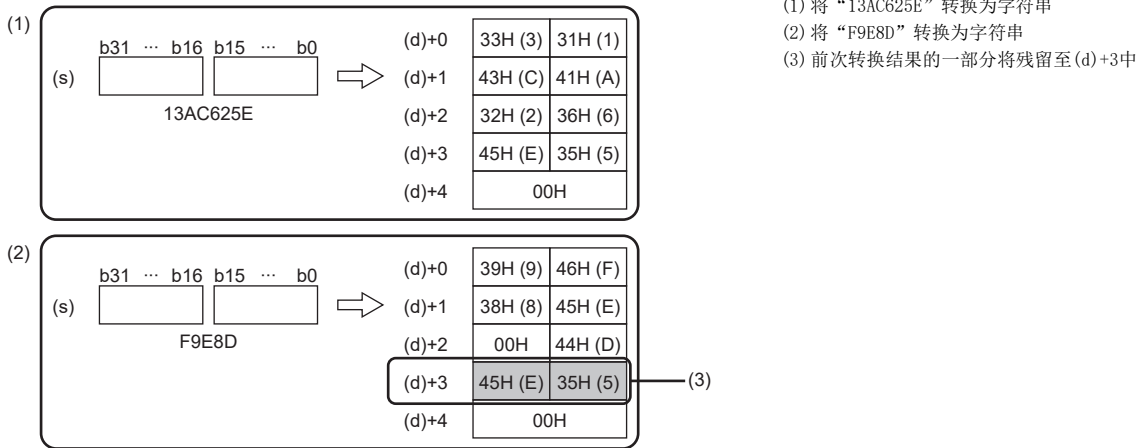
出错

没有运算错误。

注意事项

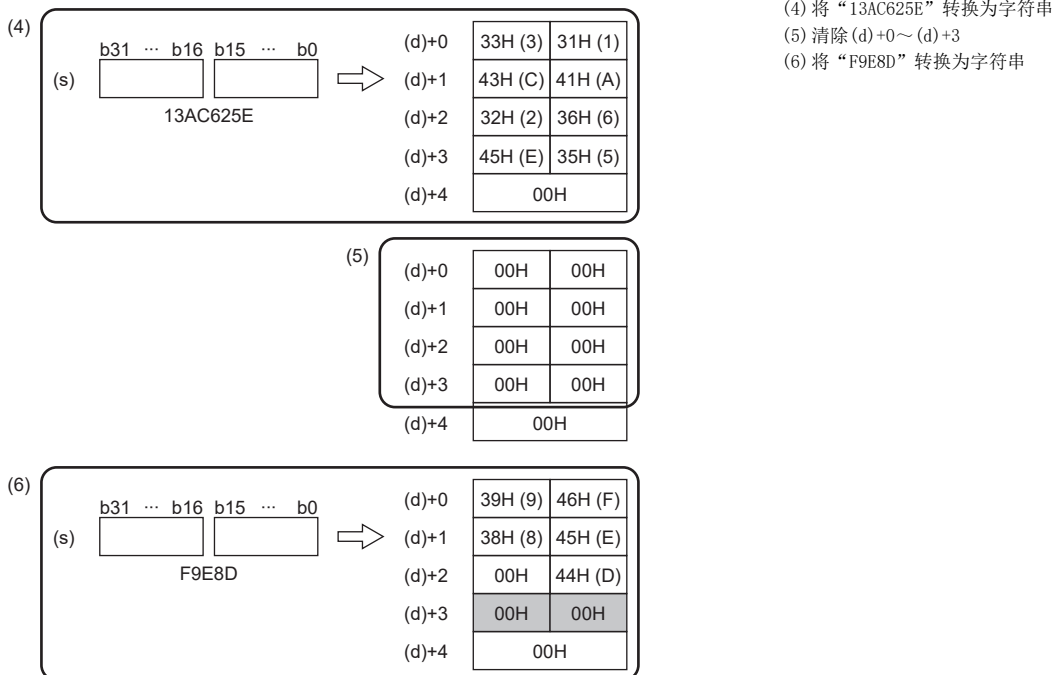
SM705(变数位数切换)为ON时，(d)中仅存储有效位数的运算结果。因此，连续执行DBINHA(P)指令，将运算结果存储到同一软元件时，(d)的一部分可能未被前次的运算结果成功覆盖而残留下来。

【例】为(s)为“13AC625E”时执行DBINHA(P)指令、(s)为“F9E8D”时执行DBINHA(P)指令的示例



- (1) 将“13AC625E”转换为字符串
- (2) 将“F9E8D”转换为字符串
- (3) 前次转换结果的一部分将残留至(d)+3中

若希望避免此情况，应将存储区域(d)+0~(d)+3全部清除后，创建执行DBINHA(P)指令的程序。



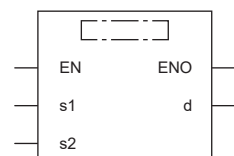
BIN16位数据→字符串转换

STR(P) (_U)

将BIN16位数据在指定的位置附加小数点后转换为字符串。

梯形图	ST	
	ENO:=STR(EN, s1, s2, d); ENO:=STRP(EN, s1, s2, d);	ENO:=STR_U(EN, s1, s2, d); ENO:=STRP_U(EN, s1, s2, d);

FBD/LD



■执行条件

指令	执行条件
STR STR_U	
STRP STRP_U	

设置数据

■内容、范围、数据类型

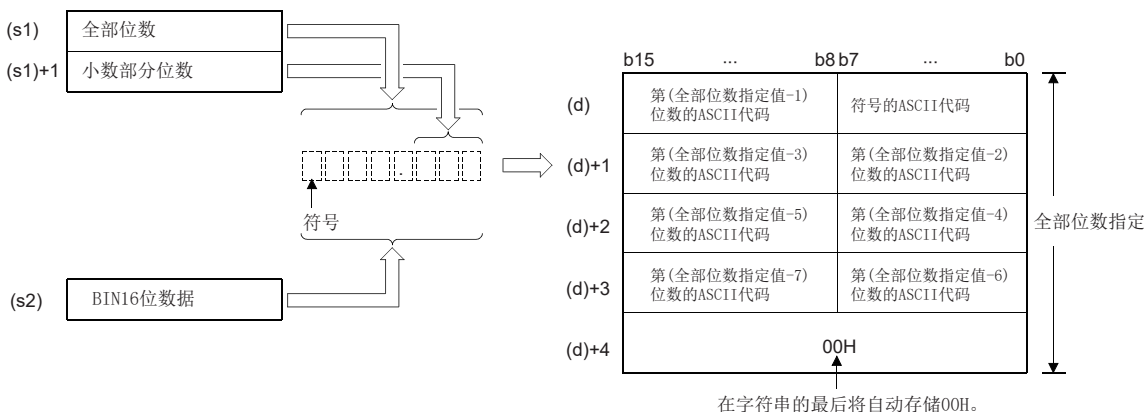
操作数	内容	范围	数据类型	数据类型(标签)	
(s1)	STR(P)	存储了转换数值的位数的起始软元件	—	带符号BIN16位	ANY16_S_ARRAY (要素数: 2)
	STR(P)_U		—	无符号BIN16位	ANY16_U_ARRAY (要素数: 2)
(s2)	STR(P)	转换的BIN数据	-32768~32767	带符号BIN16位	ANY16_S
	STR(P)_U		0~65535	无符号BIN16位	ANY16_U
(d)	存储转换后的字符串的起始软元件	—	—	字符串	ANYSTRING_SINGLE
EN	执行条件	—	—	位	BOOL
ENO	执行结果	—	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—	—

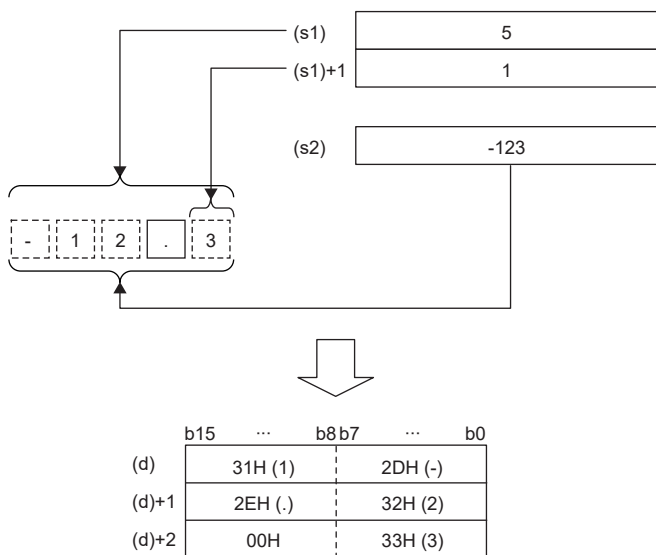
功能

- 将(s2)中指定的BIN16位数据，在(s1)中指定的位置处附加小数点后转换为字符串，存储到(d)中指定的软元件编号及其以后。



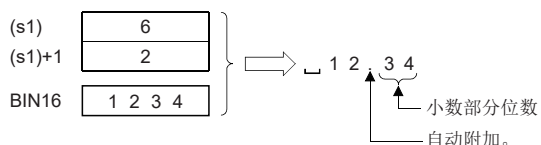
例

将(s2)的数据“-123”视为小数部分位数1位的值“-12.3”后转换成字符串时

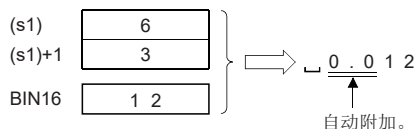


- (d)中存储小数部时，(s1)中可指定的全部位数为4~8位。(s1)+1中可指定的小数部位数为1~5位。但是，设置时应满足小数部位数 \leq (全部位数-3)的条件。
- (d)中不存储小数部时，(s1)中可指定的全部位数为2~8位。无法设置(s1)+1中可指定的小数部位数。

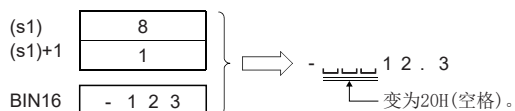
- 转换后的字符串数据将按下述方式被存储到(d)以后的软元件编号中。
- 在符号中, BIN16位数据为正时将存储20H(空白), 为负时将存储2DH(-)。
- 将小数部位数设置为0以外的情况下, 第指定的位数+1位中将自动存储2EH(.)。小数部位数为0时, 不存储2EH(.)。



- 小数部位数的值大于BIN16位数据的位数的情况下, 将自动附加0后, 向右对齐转换为“0.□□□□”。



- 全部位数的值中除去符号、小数点后的位数大于BIN16位数据的位数的情况下, 在符号与数值之间将存储20H(空白)。BIN16位数据的位数比较大的情况下, 将变为错误状态。



- 转换后的字符串的最后将自动存储00H。

出错

错误代码(SD0)	内容
3281H	<p>(s1)中输入了不能转换的非法数据时。</p> <ul style="list-style-type: none"> • 全部位数指定超出了2~8的范围 • (s1)+1的小数部位数指定超出了0~5的范围 • (s1)+1的小数部位数指定为1~5时, (s1)中指定的全部位数与(s1)+1中指定的小数部位数的指定值的关系不满足以下公式 全部位数-3≥小数部位数 • (s1)中指定的位数小于(s2)中指定的BIN16位数据的位数+2 [(s1)的位数]<[(s2)的不包含符号的BIN16位数据的位数+符号(+或-)的位数+小数点(.)的位数]
3286H	(d)中指定的软元件编号及其以后, 相应软元件/标签的分配范围中转换的字符串无法全部存储时。

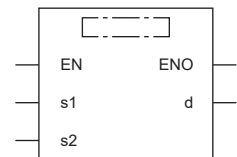
BIN32位数据→字符串转换

DSTR(P) (_U)

将BIN32位数据在指定的位置处附加小数点后转换为字符串。

梯形图	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



■执行条件

指令	执行条件
DSTR DSTR_U	
DSTRP DSTRP_U	

设置数据

■内容、范围、数据类型

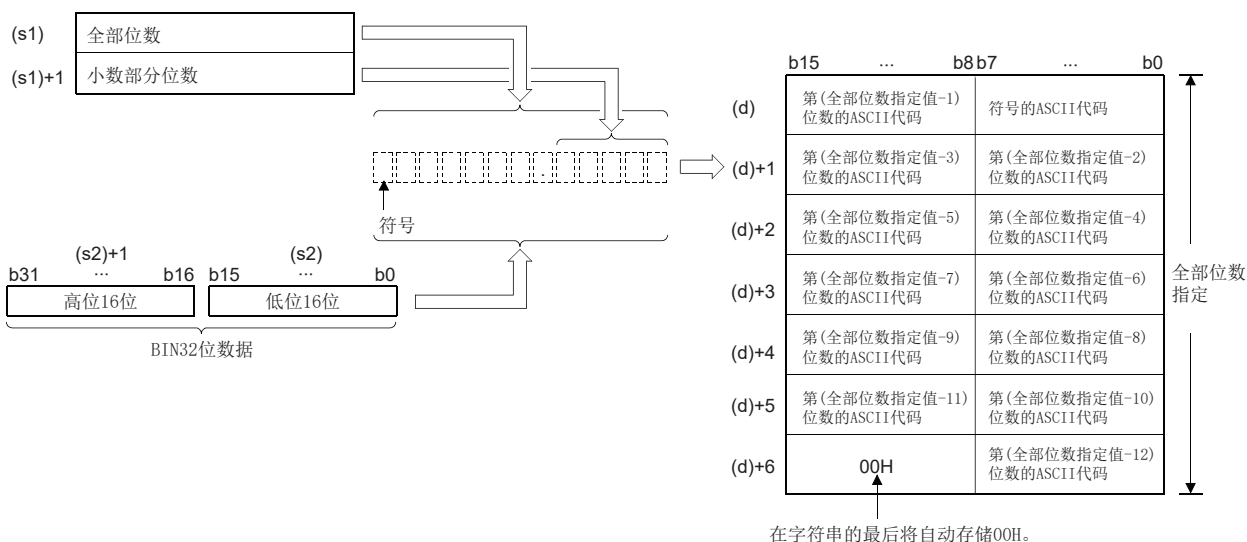
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	DSTR(P)	存储了转换数值的位数的起始软元件	带符号BIN16位	ANY16_S_ARRAY (要素数: 2)
	DSTR(P)_U		无符号BIN16位	ANY16_U_ARRAY (要素数: 2)
(s2)	DSTR(P)	-2147483648~2147483647	带符号BIN32位	ANY32_S
	DSTR(P)_U	0~4294967295	无符号BIN32位	ANY32_U
(d)	存储转换后的字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—

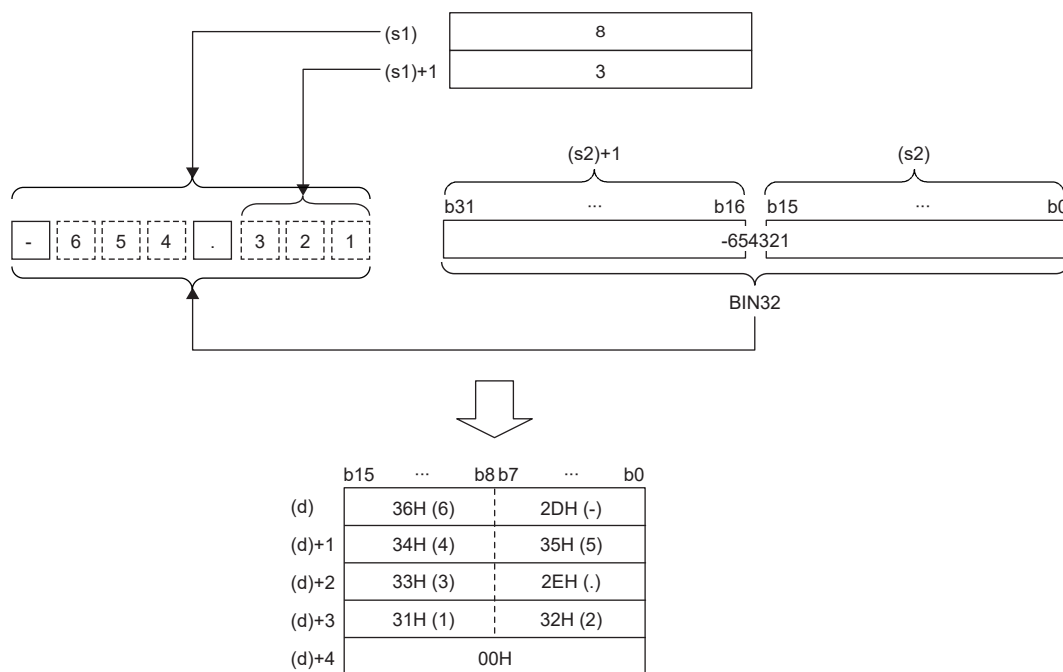
功能

- 将(s2)中指定的BIN32位数据，在(s1)中指定的位置附加小数点后转换为字符串，存储到(d)中指定的软件编号及其以后。



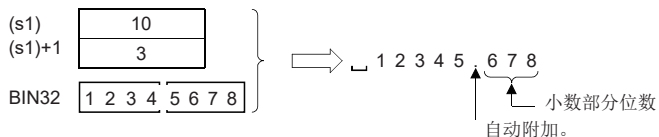
例

将(s2)的数据“-654321”视为小数部分位数3位的值“-654.321”后转换成字符串时

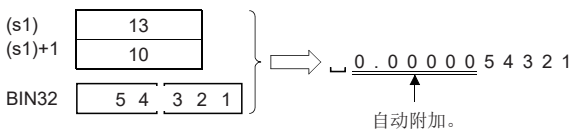


- (d)中存储小数部时，(s1)中可指定的全部位数为4~13位。(s1)+1中可指定的小数部位数为1~10位。但是，设置时应满足小数部位数≤(全部位数-3)的条件。
- (d)中不存储小数部时，(s1)中可指定的全部位数为2~13位。无法设置(s1)+1中可指定的小数部位数。

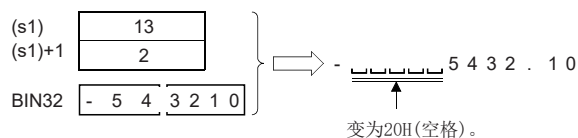
- 转换后的字符串数据将按下述方式被存储到(d)以后的软元件编号中。
- 在符号中, BIN32位数据为正时将存储20H(空白), 为负时将存储2DH(-)。
- 将小数部位数设置为0以外的情况下, 第指定的位数+1位中将自动存储2EH(.)。小数部位数为0时, 不存储2EH(.)。



- 小数部位数的值大于BIN32位数据的位数的情况下, 将自动附加0, 向右对齐转换为“0.□□□□”。



- 全部位数的值中除去符号、小数点后的位数大于BIN32位数据的位数的情况下, 在符号与数值之间将存储20H(空白)。BIN32位数据的位数比较大的情况下, 将变为错误状态。



- 转换后的字符串的最后将自动存储00H。

出错

错误代码 (SD0)	内容
3281H	(s1)中输入了不能转换的非法数据时。 <ul style="list-style-type: none"> • 全部位数指定超出了2~13的范围 • (s1)+1的小数部位数指定超出了0~10的范围 • (s1)+1的小数部位数指定为0~10时, (s1)中指定的全部位数与(s1)+1中指定的小数部位数的指定值的关系不满足以下公式 全部位数-3≥小数部位数 • (s1)中指定的位数小于(s2)中指定的BIN32位数据的位数+2 [(s1)的位数]<[(s2)不包含符号的BIN32位数据的位数+符号分(+或-)的位数+小数点(.)的位数]
3286H	(d)中指定的软元件编号及其以后, 相应软元件/标签的分配范围中转换的字符串无法全部存储时。

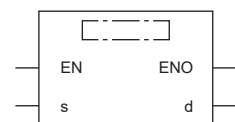
BCD4位数据→10进制ASCII代码转换

BCDDA (P)

将BCD4位数据转换为ASCII代码。

梯形图	ST
	ENO:=BCDDA(EN, s, d); ENO:=BCDDAP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
BCDDA	
BCDDAP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行ASCII转换的BCD数据	0~9999	BCD4位	ANY16
(d)	存储转换结果的起始软元件	—	字符串	ANYSTRING_SINGLE
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 将(s)中指定的BCD4位数据的各位的数值转换为ASCII代码后，存储到(d)中指定的软元件编号及其以后。
- 根据SM705(变数位数切换)的状态，(d)中存储的16进制ASCII数据的格式将有所不同。

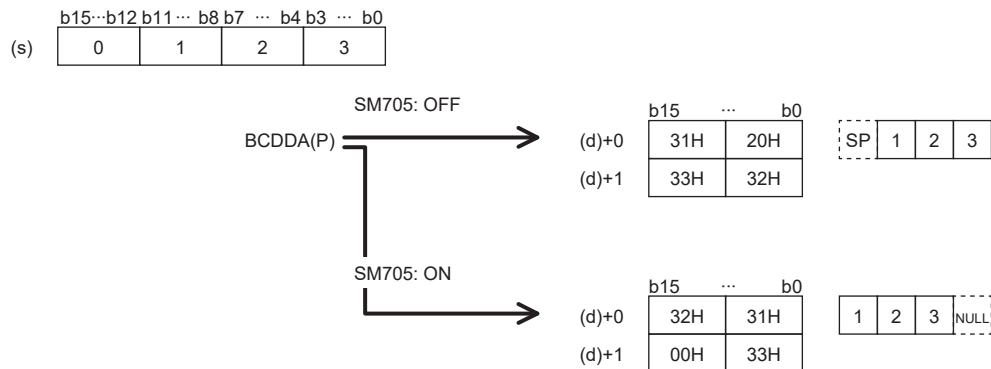
SM705的状态	(d)的存储形式	参阅目标
OFF	以固定位数(4位)存储	735页 SM705(变数位数切换)为OFF时的动作
ON	根据(s)值，向前填充各位存储	735页 SM705(变数位数切换)为ON时的动作

■动作的概要

当SM705(变数位数切换)为OFF、ON时，其各自的动作如下所示。

例

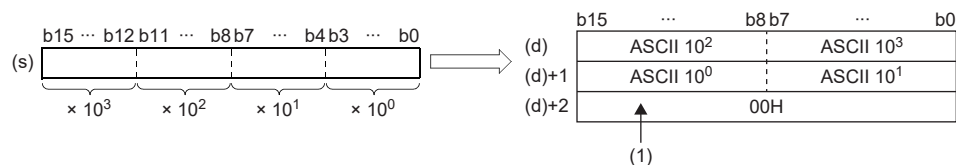
在(s)中存储有BCD4位数据“0123”的状态下执行BCDDA(P)指令时



- SM705(变数位数切换)为OFF时，位数将固定。“0123”的起始0转换为20H(空白)后存储。
- SM705(变数位数切换)为ON时将向前填充。将省略“0123”的起始0的“123”转换为ASCII数据存储，并在终端存储00H。

■SM705(变数位切换)为OFF时的动作

10进制ASCII数据将以固定位数(4位)存储至(d)~(d)+1。



ASCII 10³: 千位的ASCII代码

ASCII 10²: 百位的ASCII代码

ASCII 10¹: 十位的ASCII代码

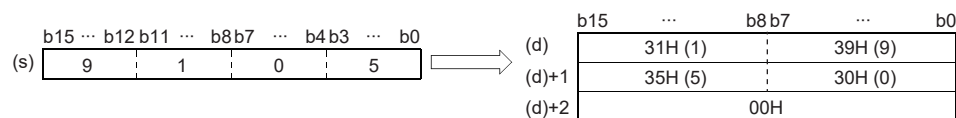
ASCII 10⁰: 个位的ASCII代码

(1) SM701(输出字符数切换)为OFF时(d)+2将存储00H, 为ON时则不变化。

- (d)中存储的运算结果中,有效位数的左侧的0处将存储20H(空白)。(进行0抑制。)例如,在“0050”的情况下,“00”将变为20H(空白)、“50”将变为有效位数。

例

(s)中指定了9105的情况下



■SM705(变数位切换)为ON时的动作

(d)中将存储有效位数的左侧0的位数(最大4位)的10进制ASCII代码。

(s)的值与(d)中存储的值的示例如下所示。

(s)的值	(d)~(d)+1的内容	(s)的值	(d)~(d)+1的内容
0H~9H	<ul style="list-style-type: none"> • (d)的高位字节中将存储00H。 • (d)+1及其以后不变化。 	10H~99H	(d)+1中将存储00H。
100H~999H	(d)+1的高位字节中将存储00H。 	1000H~9999H	(1) SM701(输出字符数切换)为OFF时, (d)+2中将存储00H。SM701(输出字符数切换)为ON时, (d)+2不变化。

ASCII 10³: 千位的ASCII代码

ASCII 10²: 百位的ASCII代码

ASCII 10¹: 十位的ASCII代码

ASCII 10⁰: 个位的ASCII代码

- 位数未满足最大位数(4位)时,无论SM701(输出字符数切换)为ON或OFF,字符串的终端将存储00H。字符串的终端为低位字节时,高位字节也存储00H。
- 位数为最大位数(4位)时,SM701(输出字符数切换)为OFF时,(d)+2中将存储00H。SM701(输出字符数切换)为ON时,(d)+2不变化。

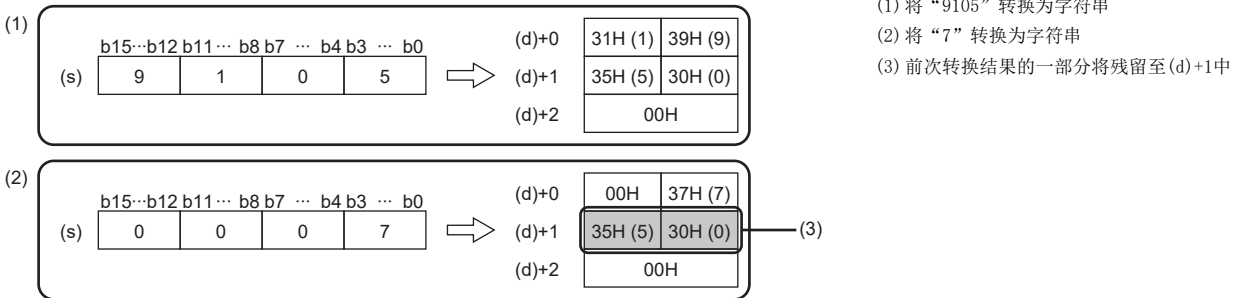
出错

错误代码 (SD0)	内容
3281H	(s) 的数据超出了0~9999的范围时。
3286H	(d) 中指定的软元件编号及其以后, 相应软元件/标签的分配范围中转换的字符串无法全部存储时。

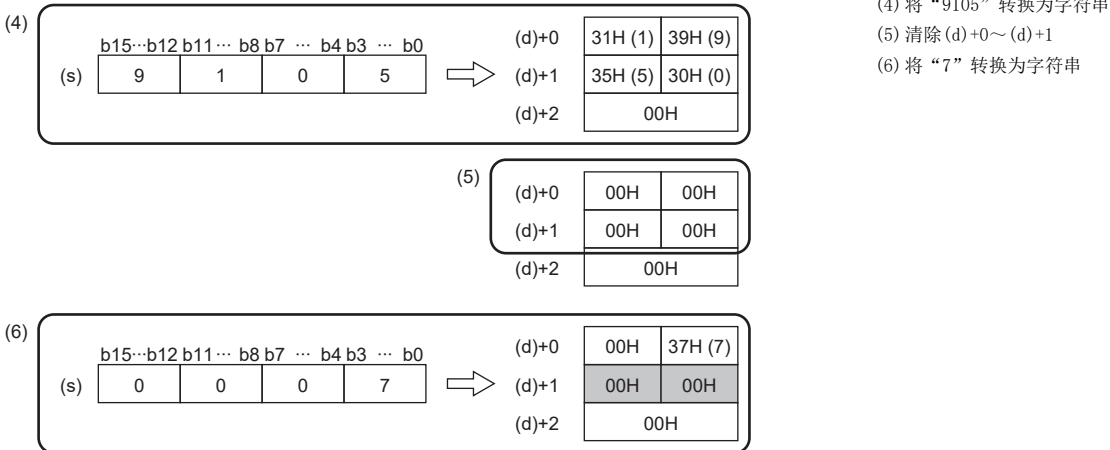
注意事项

SM705(变数位数切换)为ON时, (d) 中仅存储有效位数的运算结果。因此, 连续执行BCDDA(P)指令, 将运算结果存储到同一软元件时, (d) 的一部分可能未被前次的运算结果成功覆盖而残留下来。

[例] 在(s)为“9105”时执行BCDDA(P)指令、(s)为“0007”时执行BCDDA(P)指令的示例



若希望避免此情况, 应将存储区域(d)+0~(d)+1全部清除后, 创建执行BCDDA(P)指令的程序。



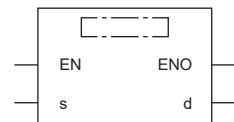
BCD8位数据→10进制ASCII代码转换

DBCDDA (P)

将BCD8位数据转换为ASCII代码。

梯形图	ST
	ENO:=DBCDDA (EN, s, d) ; ENO:=DBCDDAP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
DBCDDA	
DBCDDAP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行ASCII转换的BCD数据	0~99999999	BCD8位	ANY32
(d)	存储转换结果的起始软元件	—	字符串	ANYSTRING_SINGLE
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 将(s)中指定的BCD8位数据的各位的数值转换为ASCII代码后，存储到(d)中指定的软元件编号及其以后。
- 根据SM705(变数位数切换)的状态，(d)中存储的10进制ASCII数据的格式将有所不同。

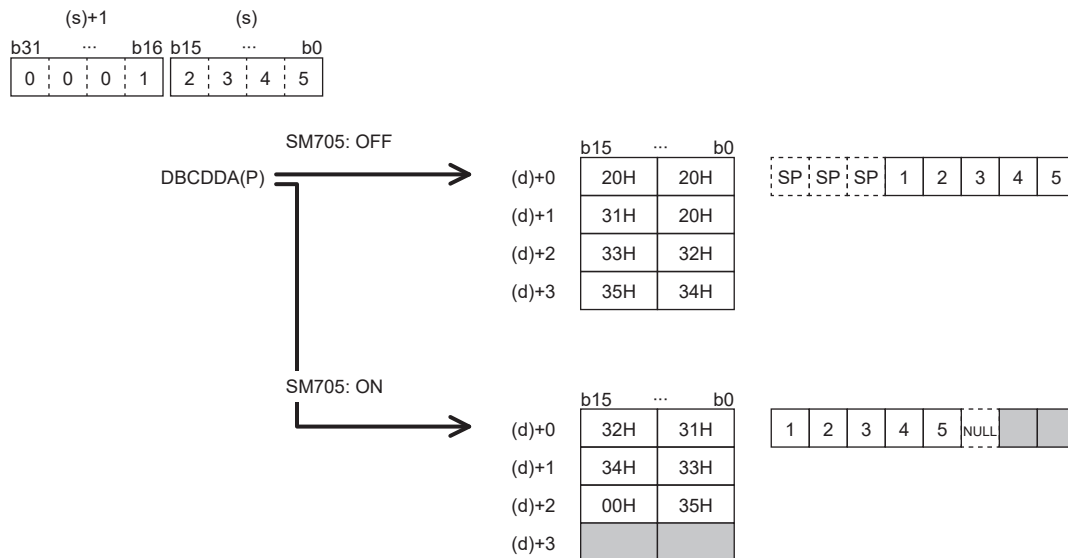
SM705的状态	(d)的存储形式	参阅目标
OFF	以固定位数(8位)存储	739页 SM705(变数位数切换)为OFF时的动作
ON	根据(s)值，向前填充各位存储	740页 SM705(变数位数切换)为ON时的动作

■动作的概要

当SM705(变数位数切换)为OFF、ON时，其各自的动作如下所示。

例

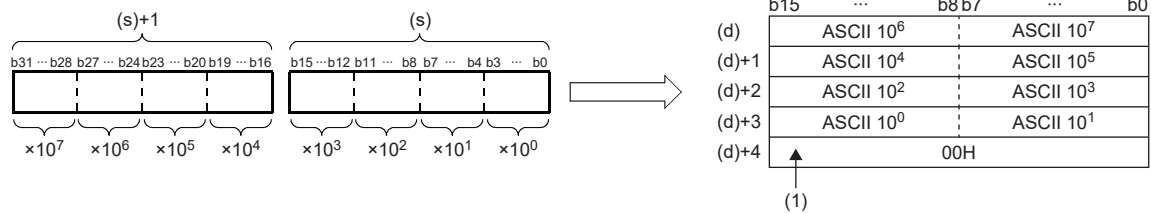
在(s)中存储有BCD8数据“00012345”的状态下执行DBCDDA(P)指令时



- SM705(变数位数切换)为OFF时，位数将固定。将“00012345”的0转换为20H(空白)后存储。
- SM705(变数位数切换)为ON时将向前填充。将省略“00012345”的起始0的“12345”转换为ASCII数据存储，并在终端存储00H。

■SM705(变数位切换)为OFF时的动作

10进制ASCII数据将以固定位数(8位)存储至(d)~(d)+3。



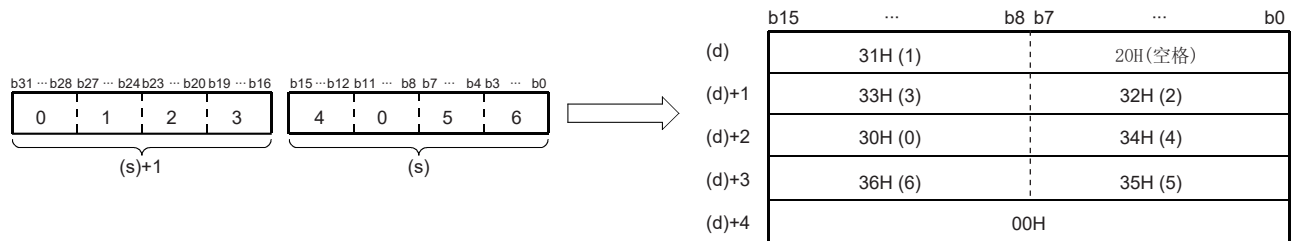
ASCII 10⁷: 千万位的ASCII代码
 ASCII 10⁶: 百万位的ASCII代码
 ⋮
 ASCII 10¹: 十位的ASCII代码
 ASCII 10⁰: 个位的ASCII代码

(1) SM701(输出字符数切换)为OFF时(d)+4将存储00H, 为ON时则不变化。

- (d)中存储的运算结果中, 有效位数的左侧的0处将存储20H(空白)。(进行0抑制。)例如, 在“00012098”的情况下, “000”将变为20H(空白)、“12098”将变为有效位数。

例

(s)中指定了01234056的情况下



■SM705 (变数位数切换) 为ON时的动作

(d) 中将存储有效位数的左侧0的位数 (最大8位) 的10进制ASCII代码。

(s) 的值与 (d) 中存储的值的示例如下所示。

(s) 的值	(d) ~ (d)+3 的内容	(s) 的值	(d) ~ (d)+3 的内容																																																							
0H~9H	<ul style="list-style-type: none"> (d) 的高位字节中将存储00H。 (d)+1 及其以后不变化。 <table border="1"> <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" style="background-color: #cccccc;"></td> </tr> <tr> <td>(d)+2</td> <td colspan="4" style="background-color: #cccccc;"></td> </tr> <tr> <td>(d)+3</td> <td colspan="4" style="background-color: #cccccc;"></td> </tr> </table>	b15	...	b8 b7	...	b0	(d)	00H	:	ASCII 10 ⁰		(d)+1					(d)+2					(d)+3					10H~99H	<ul style="list-style-type: none"> (d)+1 中将存储00H。 (d)+2 及其以后不变化。 <table border="1"> <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" style="background-color: #cccccc;">00H</td> </tr> <tr> <td>(d)+2</td> <td colspan="4" style="background-color: #cccccc;"></td> </tr> <tr> <td>(d)+3</td> <td colspan="4" style="background-color: #cccccc;"></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	<ul style="list-style-type: none"> (d)+3 的高位字节中将存储00H。 <table border="1"> <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	<ul style="list-style-type: none"> (1) SM701 (输出字符数切换) 为OFF时, (d)+4 中将存储00H。SM701 (输出字符数切换) 为ON时, (d)+4 不变化。 <table border="1"> <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" style="background-color: #cccccc;">00H</td> </tr> </table> <p>(1)</p>	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: 千万位的ASCII代码

ASCII 10⁶: 百万位的ASCII代码

⋮

ASCII 10¹: 十位的ASCII代码

ASCII 10⁰: 个位的ASCII代码

- 位数未满足最大位数 (8位) 时, 无论SM701 (输出字符数切换) 为ON或OFF, 字符串的终端将存储00H。字符串的终端为低位字节时, 高位字节也存储00H。
- 位数为最大位数 (8位) 时, SM701 (输出字符数切换) 为OFF时, (d)+4 中将存储00H。SM701 (输出字符数切换) 为ON时, (d)+4 不变化。

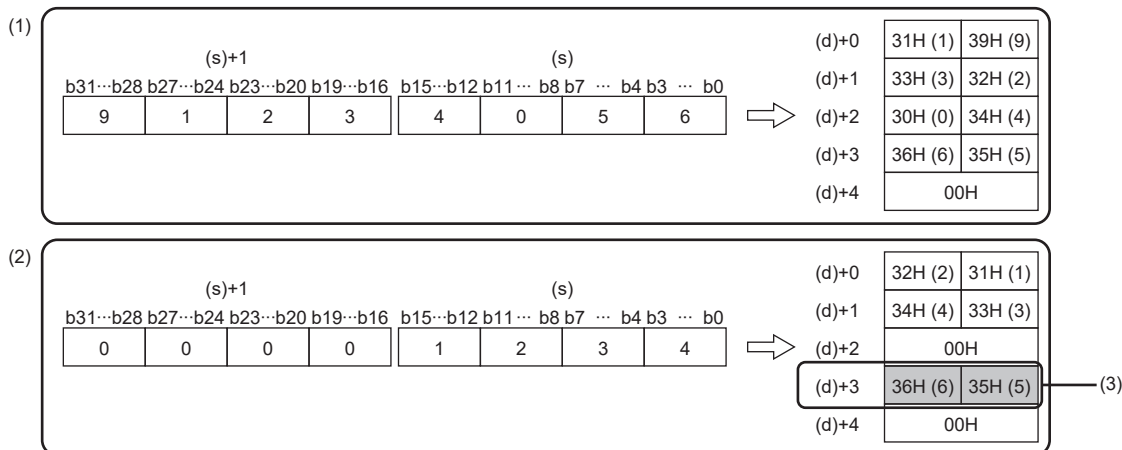
出错

错误代码 (SD0)	内容
3281H	(s) 的数据超出了 0~99999999 的范围时。
3286H	(d) 中指定的软元件编号及其以后，相应软元件/标签的分配范围中转换的字符串无法全部存储时。

注意事项

SM705(变数位数切换)为ON时，(d)中仅存储有效位数的运算结果。因此，连续执行DBCDDA(P)指令，将运算结果存储到同一软元件时，(d)的一部分可能为未被前次的运算结果成功覆盖而残留下来的情况。

[例] 在(s)为“91234056H”时执行DBCDDA(P)指令、(s)为“00001234H”时执行DBCDDA(P)指令的示例

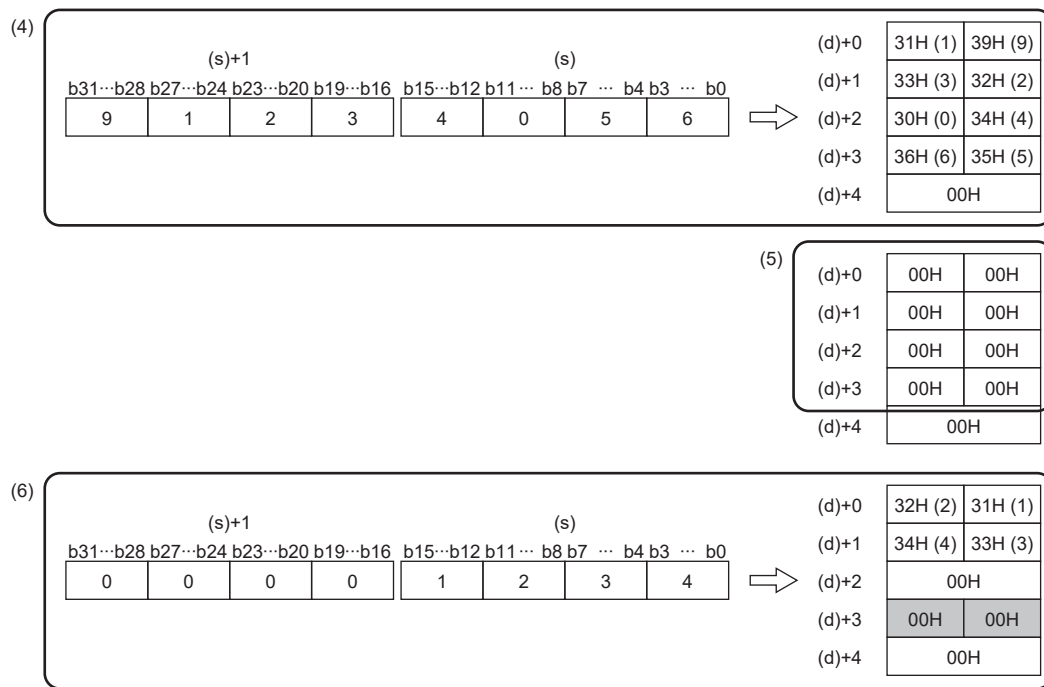


(1) 将“91234056”转换为字符串

(2) 将“1234”转换为字符串

(3) 前次转换结果的一部分将残留至(d)+3中

若希望避开此情况，应将存储区域(d)+0~(d)+3全部清除后，创建执行DBCDDA(P)指令的程序。



(4) 将“91234056”转换为字符串

(5) 清除(d)+0~(d)+3

(6) 将“1234”转换为字符串

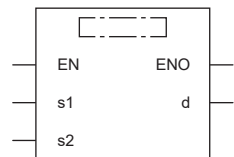
单精度实数→字符串转换

ESTR(P)

将单精度实数数据按照显示指定转换为字符串。

梯形图	ST
	ENO:=ESTR(EN, s1, s2, d); ENO:=ESTRP(EN, s1, s2, d);

FBD/LD



■执行条件

指令	执行条件
ESTR	
ESTRP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	转换的单精度实数或存储了数据的起始软元件	$0, 2^{-126} < s1 < 2^{128}$	单精度实数	ANYREAL_32
(s2)	存储了转换数值的显示指定的起始软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 3)
(d)	存储转换后的字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数			其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—

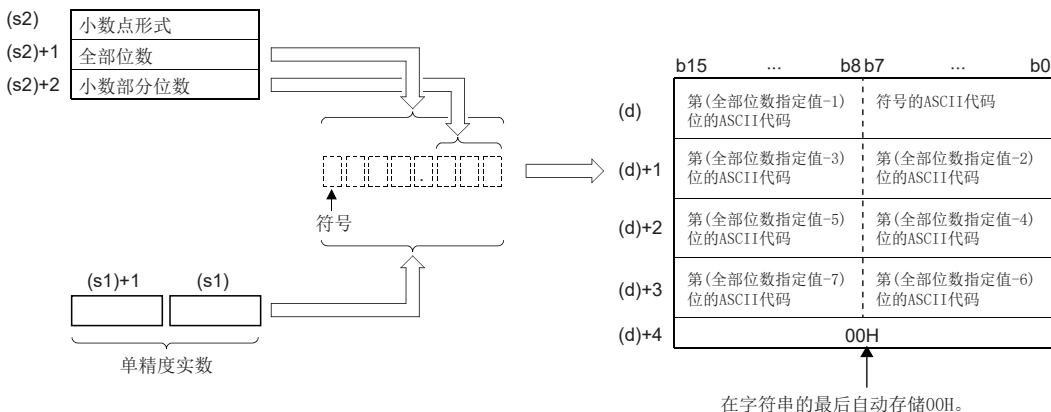
功能

- 将(s1)中指定的软件中存储的单精度实数数据，按照(s2)中指定的软件编号及其以后存储的显示指定转换为字符串后，存储到(d)中指定的软件编号及其以后。
- 根据(s2)中指定的显示指定转换后的数据有所不同。

(s2)	0: 小数点形式 1: 指数形式
(s2)+1	全部位数
(s2)+2	小数部分位数

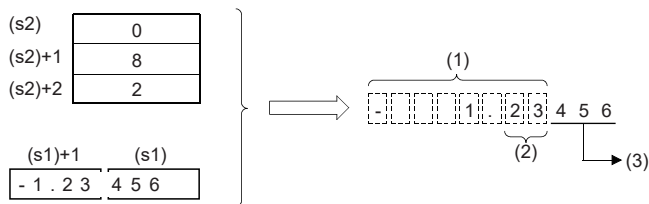
■ 小数点形式

- (s2)中指定了0的情况下，将变为小数点形式。

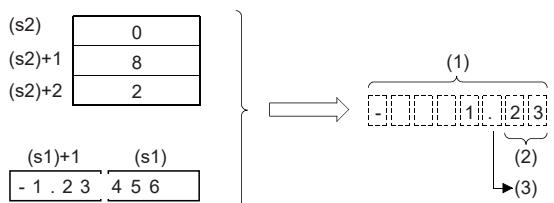


- (d)中存储小数部时，(s2)+1中可指定的全部位数为全部位数(最大24)≥(整数部位数*1+小数部位数+2)。(s2)+2中可指定的小数部位数为0~7位。但是，设置时应满足小数部位数≤(全部位数-3)的条件。

- *1 (s1)中指定的32位浮点型实数数据的整数部位数如下所示。
- (d)中不存储小数部时，(s2)+1中可指定的全部位数为全部位数(最大24)≥(整数部位数*2+1)。无法设置(s2)+2中可指定的小数部位数。
- *2 (s1)中指定的32位浮点型实数数据的整数部位数如下所示。
- 转换后的字符串数据将按下述方式被存储到(d)以后的软件编号中。
 - 在符号中，单精度实数数据为正时将存储20H(空白)，为负时将存储2DH(-)。
 - 小数部位数的范围内无法容纳单精度实数数据的小数部分的情况下，低位小数部将被四舍五入。

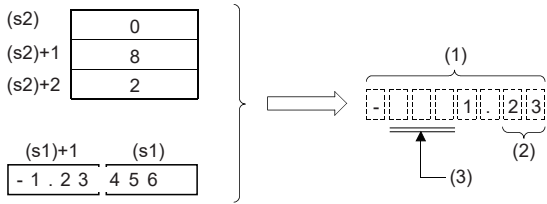


- (1) 全部位数
- (2) 小数部位数
- (3) 四舍五入。
- 将小数部位数设置为0以外的情况下，指定的小数部位数+1位中将自动存储2EH(.)。小数部位数为0时，不存储2EH(.)。



- (1) 全部位数
- (2) 小数部位数
- (3) 自动附加。

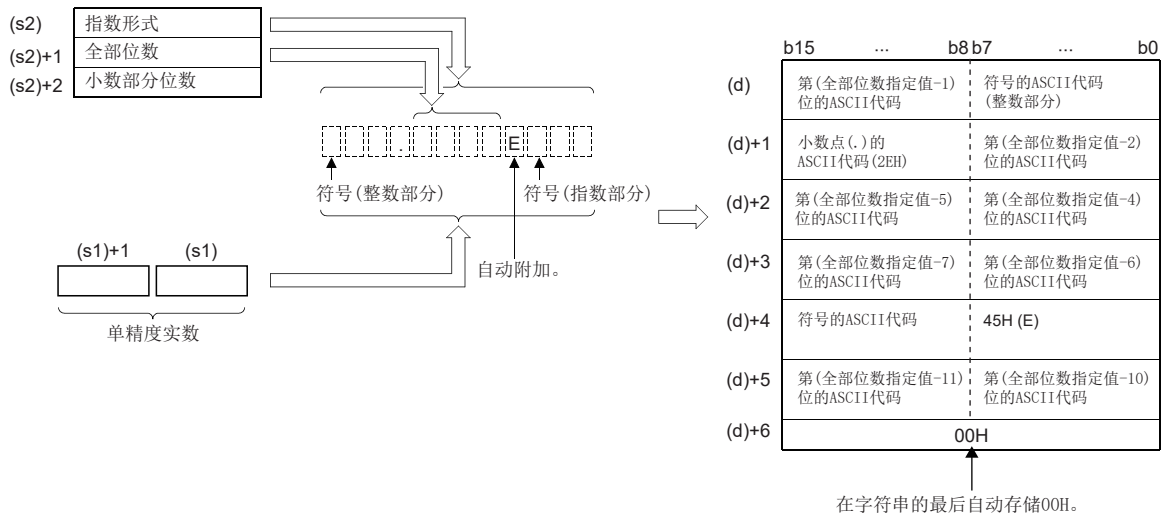
- 从全部位数中除去符号、小数点、小数部后的位数大于单精度实数数据的整数部的情况下，在符号与整数部之间将存储20H(空白)。



- (1) 全部位数
- (2) 小数部位数
- (3) 变为20H(空白)。
- 转换后的字符串的最后将自动存储00H。
- (s1)可指定的32位浮点型实数数据的整数部位数为1~16位。

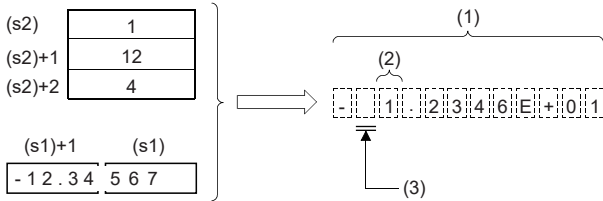
指数形式

- (s2)中指定了1的情况下，将变为指数形式。



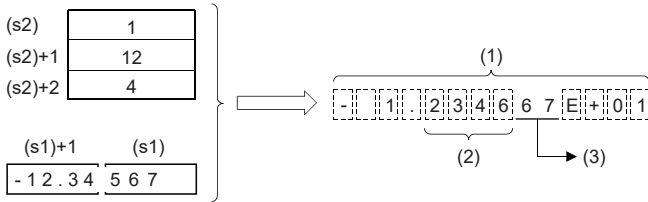
- (d)中存储小数部时，(s2)+1中可指定的全部位数为位数(最大24)≥(小数部位数+7)。(s2)+2中可指定的小数部位数为0~7位。但是，设置时应满足小数部位数≤(全部位数-7)的条件。
- (d)中不存储小数部时，(s2)+1中可指定的全部位数为位数(最大24)≥6。无法设置(s2)+2中可指定的小数部位数。

- 转换后的字符串数据将按下述方式被存储到(d)以后的软元件编号中。
- 整数部的符号中，单精度实数数据为正时将存储20H(空白)，为负时将存储2DH(-)。
- 整数部固定为1位。在整数部与符号之间存储20H(空白)。



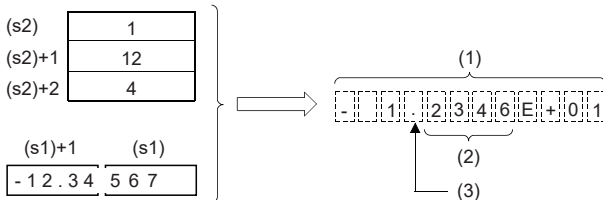
- (1) 全部位数
- (2) 固定为1位
- (3) 变为20H(空白)。

- 小数部位数的范围内无法容纳单精度实数数据的小数部分的情况下，低位小数部将被四舍五入。



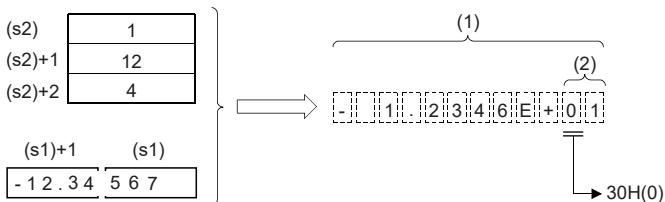
- (1) 全部位数
- (2) 小数部位数
- (3) 舍去。

- 将小数部位数设置为0以外的情况下，指定的小数部位数+1位中将自动存储2EH(.)。小数部位数为0时，不存储2EH(.)。



- (1) 全部位数
- (2) 小数部位数
- (3) 自动附加。

- 指数部的符号中，指数为正时将存储2BH(+), 为负时将存储2DH(-)。
- 指数部固定为2位。指数部为1位数的情况下，在指数部的符号之间将存储30H(0)。



- (1) 全部位数
- (2) 固定为2位

- 转换后的字符串的最后将自动存储00H。

- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

出错

错误代码 (SD0)	内容
3281H	<p>(s1) 超出下述范围时。 $0, 2^{-126} \leq (s1) < 2^{128}$</p> <p>(s2) 中设置了不能转换的非法数据时。</p> <ul style="list-style-type: none"> • (s2) 中指定的形式指定为0、1以外时。 • 小数点形式中, (s2)+1中指定的全部位数指定超出了下述范围时。 小数部位数为0时: 全部位数\geq整数部位数*1+1 小数部位数为0以外时: 全部位数\geq整数部位数*1+小数点位数+2 • 小数点形式中, (s2)+1中指定的全部位数及(s2)+2中指定的小数部位数指定超出了下述范围时。 小数部位数为0时: 全部位数\geq2 小数部位数为0以外时: 全部位数\geq(小数点位数+3) • 指数形式中, (s2)+1中指定的全部位数指定超出了下述范围时。 小数部位数为0时: 全部位数\geq6 小数部位数为0以外时: 全部位数\geq(小数部位数+7) • 指数形式中, (s2)+2中指定的小数部位数指定超出了下述范围时。 小数点形式时: 小数部位数\leq(全部位数-3) 指数形式时: 小数部位数\leq(全部位数-7) • 小数点形式的情况下, (s1) 中可指定的32位浮点实数数据的整数部位数超过了16位的情况下。 • (s2)+2中指定的小数部位数超出了0~7位时。 <p>全部位数指定了超出24的值时。</p>
3282H	(s1) 中指定的软元件的内容为-0、非正规化数、非数、 $\pm\infty$ 时。
3286H	(d) 中指定的软元件编号及其以后, 相应软元件/标签的分配范围中转换的字符串无法全部存储时。

*1 (s1) 中指定的32位浮点型实数数据的整数部位数如下所示。

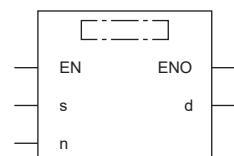
16进制BIN数据→16进制ASCII代码转换

INT2ASC(P)

将BIN16位数据转换为16进制数ASCII代码后，以任意范围存储。

梯形图	ST
	ENO:=INT2ASC(EN, s, n, d); ENO:=INT2ASCP(EN, s, n, d)

FBD/LD



■执行条件

指令	执行条件
INT2ASC	
INT2ASCP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储了转换为字符串的BIN数据的起始软元件	—	带符号BIN16位	ANY16*1
(d)	存储转换后的字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(n)	存储的字符数	0~16383	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

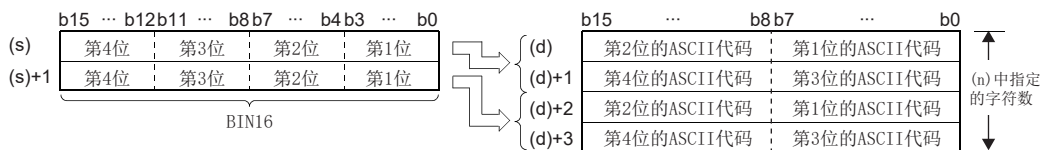
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

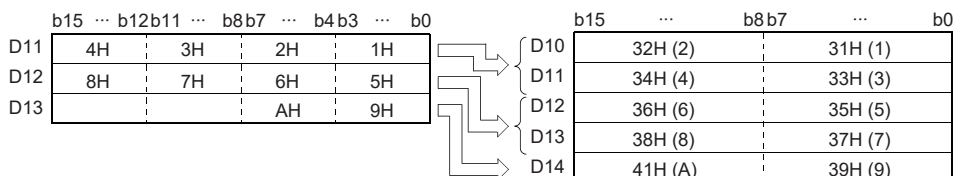
操作数	位		字		双字			间接指定	常数				其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

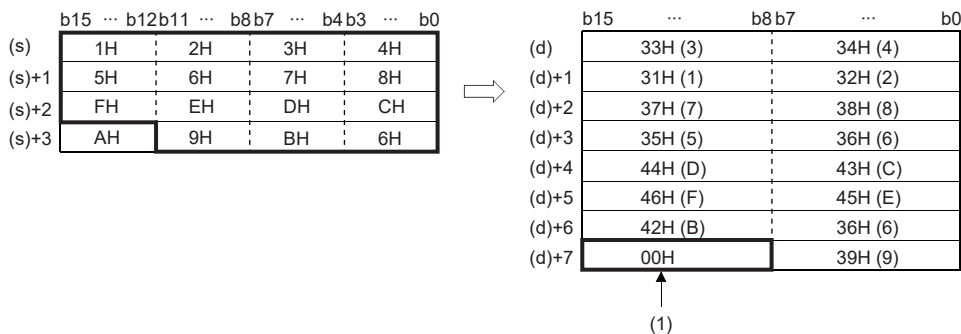
- 将(s)中指定的软元件编号及其以后存储的BIN16位数据，以16进制数处理转换为ASCII转换后，按照(n)中指定的字符数存储到(d)中指定的软元件编号及其以后。



- 根据(n)中设置的字符数，(s)中指定的BIN数据的范围及(d)中指定的字符串的存储软元件的范围将被自动确定。
- 即使存储了转换的BIN数据的软元件范围与存储转换后的ASCII数据的软元件范围重复的情况下，也将正常进行处理。



- (n)中指定的字符数为奇数的情况下，存储字符串的软元件范围的最终软元件编号的高位8位中，将自动存储00H。



(1) 自动存储00H。

- (n)中指定的字符数为0的情况下，不进行转换处理。

出错

错误代码(SD0)	内容
3285H	(n)中输入了超出允许指定范围的数据时。 <ul style="list-style-type: none"> 指定的字符数超出了0~16383的范围


多个数据的字符串转换合并

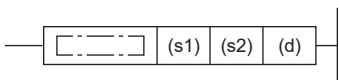
SPF (P)

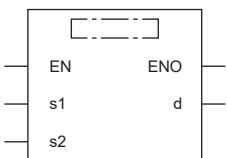
按照指定的显示格式将多个数据转换为字符串，与任意字符串连接后作为1个字符串输出。

限制事项



使用时，应确认控制器及工程工具的版本。

所使用的控制器的用户手册

梯形图	ST
	ENO:=SPF(EN, s1, s2, d); ENO:=SPFP(EN, s1, s2, d);

FBD/LD


执行条件

指令	执行条件
SPF	
SPFP	

设置数据

内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	字符串数据及显示格式或存储了这些的起始软元件	—	字符串	ANYSTRING_SINGLE
(s2)	存储了转换的数据的起始软元件	—	带符号BIN16位	ANY16
(d)	存储了转换及连接的字符串的起始软元件	—	字符串	ANYSTRING_SINGLE

可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 按照(s1)中指定的显示格式，将(s2)中指定的软元件编号及其以后存储的数据转换为字符串。与(s1)中存储的任意字符串连接，并作为1个字符串存储到(d)中指定的软元件编号及其以后。此外，也可以省略任意字符串，仅合并(s2)指定的软元件编号及其以后存储的数据。
- 显示格式在起始处记载“%”，按照[标志]、[最小字段宽度]、[精度]、[获取数据大小]、[转换指定符]的顺序组合指定符，并与要连接的任意字符串一起存储(s1)。可使用的显示格式选择如下所示。按照不同的顺序使用了显示格式的情况下，将变为错误状态。

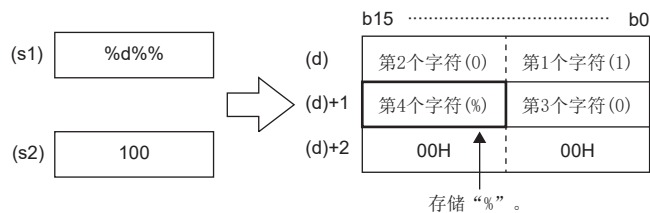
显示格式	指定符*1	说明	详细内容
标志	-, +, “ ” (半角空格)、#, 0	根据指定的指定符，对输出到(d)的字符串进行修饰。	752页 标志
最小字段宽度	任意数值	指定包括转换的字符串的符号、小数点、指数部及“ ” (半角空格)在内的输出到(d)的整个输出字符数。省略记载时输出获取数据的字符数。	753页 最小字段宽度
精度	“.” (点号)数值	从(s2)中存储的数据，指定通过[转换指定符]输出到(d)的字符数。	754页 精度
获取数据大小	w, q	指定转换(s2)中存储的数据的多少字及是否输出到(d)。省略记载时以1字单位获取。	756页 获取数据大小
转换指定符	d, u, x, X, o, s, c, f, e, E, g, G	指定输出到(d)的字符串的输出格式。	757页 转换指定符

*1 区分大小写。

- 可以省略[标志]、[最小字段宽度]、[精度]、[获取数据大小]。
- 仅将“%”作为字符输出时，可以省略[转换指定符]。将“%%”存储到(s1)，作为字符串输出时，“%”将作为字符串存储到(d)中。此外，“%%”之后存储的数据将作为字符输出，直至下一个显示格式为止。

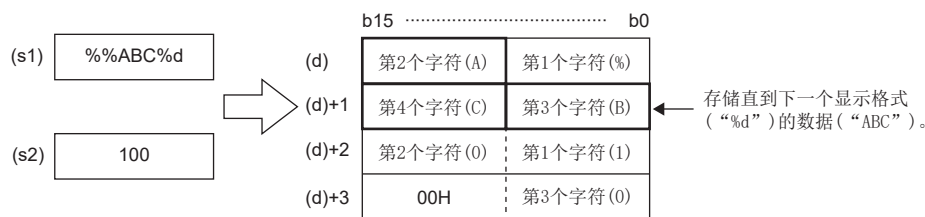
例

将“%d%%”存储到(s1)，将“100”存储到(s2)中指定的软元件时，(d)中指定的软元件中将存储“100%”。



例

将“%%ABC%d”存储到(s1)，将“100”存储到(s2)中指定的软元件时，(d)中指定的软元件中将存储“%ABC100”。

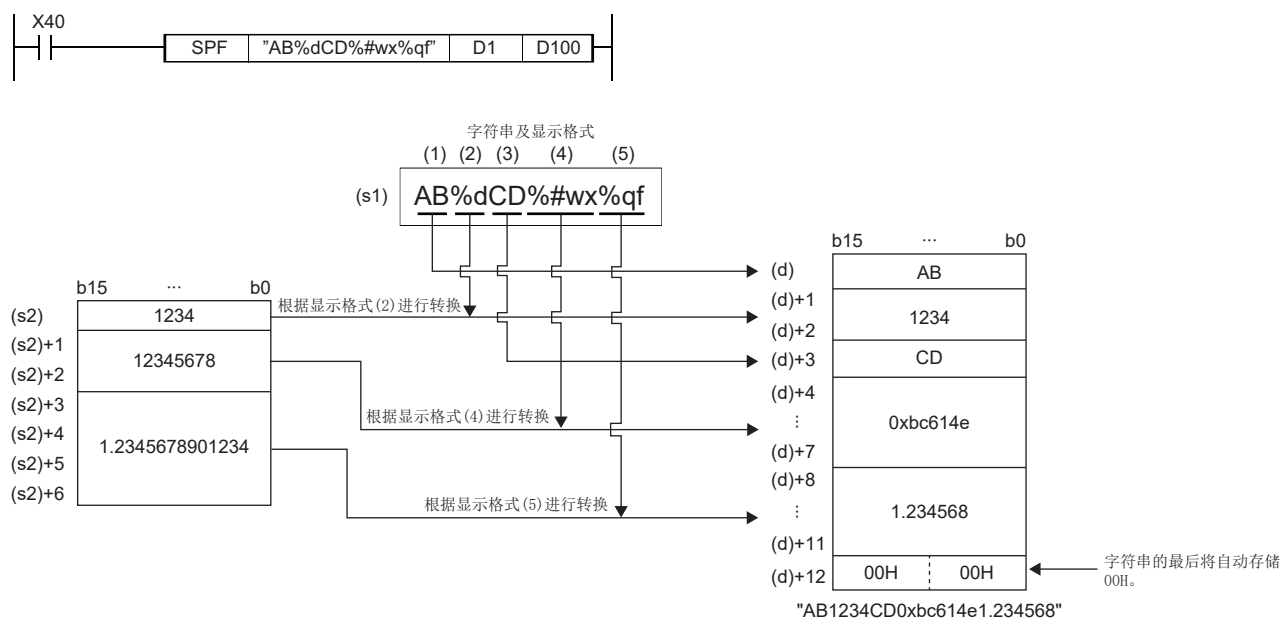


- 转换后字符串(d)为偶数字节结束时，字符串的最后的下一个软元件(高位8位、低位8位)中将分别存储NULL代码“00H”。此外，转换后字符串(d)为奇数字节结束时，字符串的最后的软元件(高位8位)中将存储NULL代码“00H”。

例

存储了下述数据的状态下，执行了SPF(P)指令的情况下，D100及其以后存储的字符串为“AB1234CD0xbc614e1.234568”。

操作数	存储的数据
(s1)	AB%dCD%#wx%qf(字符串) • 存储的字符串指定了下述显示格式。 AB(输出字符串)%d(D1的显示格式)CD(输出字符串)%#wx(D2~D3的显示格式)%qf(D4~D7的显示格式)
(s2)	D1 1234(10进制数) D2~D3 12345678(10进制数) D4~D7 1.2345678901234(双精度实数)
(d)	D100



- (1): 输出字符串AB。
- (2): 显示格式为[获取数据大小]=(省略记载)、[转换指定符]=d。从(s2)以1字单位获取数据，以带符号10进制数输出。
- (3): 输出字符串CD。
- (4): 显示格式为[标志]=#、[获取数据大小]=w、[转换指定符]=x。从(s2)+1以2字单位获取数据，在起始处附加“0x”，并以无符号16进制数输出。
- (5): 显示格式为[精度]=(省略记载)、[获取数据大小]=q、[转换指定符]=f。从(s2)+3以4字单位获取数据，以双精度实数的小数表现输出。此外，由于省略[精度]的记载，因此输出小数点以后的6位，对输出字符以后的第1位数值进行四舍五入。

■标志

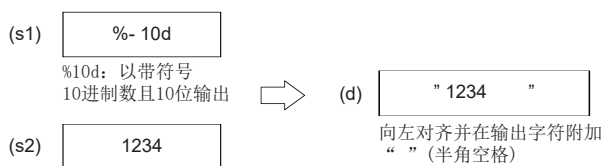
根据条件，对输出到(d)的字符串进行修饰。

指定符	说明
(省略记载)	右对齐输出。
-	<p>[最小字段宽度]大于从(s2)获取并转换的字符数时，左对齐输出。</p> <p>(s1) <code>%-10d</code> %10d: 以带符号10进制数且10位输出</p> <p>(s2) <code>1234</code></p> <p>⇒ (d) <code>"1234 "</code> 左对齐输出</p>
+	输出0或正数时，附加“+”。
“ ” (半角空格)	输出0或正数时，附加“ ” (半角空格)。
#	<p>根据设置的[转换指定符]，以下述输出格式输出。</p> <ul style="list-style-type: none"> “o”：在起始处附加“0”。 “x”、“X”：在起始处附加“0x”或“0X”。 “f”、“e”、“E”：输出实数时必须输出小数点。 <p>(s1) <code>%.0wf</code> %.0wf: 以单精度实数小数表现并输出小数点以后0位</p> <p>(s2) <code>1.234</code> (s2)+1</p> <p>⇒ (d) <code>"1."</code> 输出“.”(点号)</p> <p>• “g”、“G”：输出实数时必须输出小数点，并在末尾处存储有效位数“0”。</p> <p>(s1) <code>%.#wg</code> %.#wg: 以单精度实数小数表现并未输出小数点以下的“0”</p> <p>(s2) <code>1.0</code> (s2)+1</p> <p>⇒ (d) <code>"1.00000"</code> 由于省略了[精度]，因此将输出末尾的“0”至有效位数(6位)</p>
0	<p>[最小字段宽度]大于从(s2)获取并转换的字符数时，在输出字符串前输出“0”。</p> <p>(s1) <code>%.#06x</code> %.#6x: 以16进制数且6位输出</p> <p>(s2) <code>255</code></p> <p>⇒ (d) <code>"0x00ff"</code> 将“0”输出至输出字符串前</p>

可以指定多个[标志]。使用顺序没有限制。但同时使用了“+”及“ ” (半角空格)的情况下，优先输出“+”的指定符的内容。此外，使用了相同指定符时，第2个及其以后的指定将被忽略。同时指定了“-”和“0”的指定符时，“0”的指定符将被忽略。

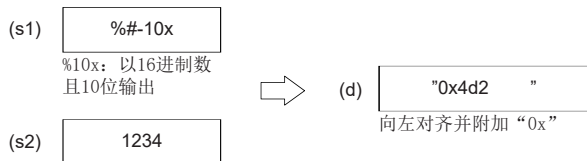
例

使用了“-”、“ ” (半角空格)的情况下

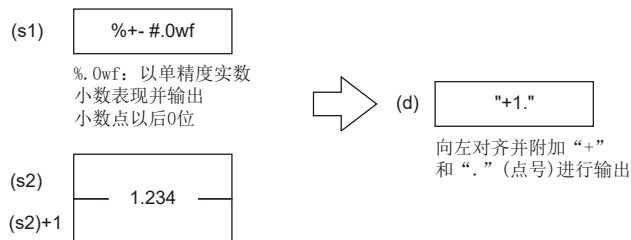


例

使用了“#”、“-”的情况下

**例**

使用了“+”、“-”、“ ”(半角空格)及“#”的情况下

**■最小字段宽度**

指定包括转换的字符串的符号、小数点、指数部及“ ”(半角空格)在内的输出到(d)的整个输出字符数。

指定符	说明
数值	<ul style="list-style-type: none"> 省略记载时输出获取数据的字符数。
(s1) <input type="text" value="%d"/>	%d: 以带符号10进制数输出 ⇒ (d) <input type="text" value="1234"/> 由于省略了数值的记载,因此按输出字符数(4位)输出
(s2) <input type="text" value="1234"/>	
(s1) <input type="text" value="%3d"/>	<ul style="list-style-type: none"> [最小字段宽度]为从(s2)获取并转换的字符数及其以下时,忽略[最小字段宽度]的指定。 %d: 以带符号10进制数输出 ⇒ (d) <input type="text" value="1234"/> 按输出字符数(4位)输出
(s2) <input type="text" value="1234"/>	
(s1) <input type="text" value="%10d"/>	<ul style="list-style-type: none"> [最小字段宽度]大于从(s2)获取并转换的字符数时,在数值前输出“ ”(半角空格)。 %d: 以带符号10进制数输出 ⇒ (d) <input type="text" value=" 1234"/> 在输出字符前输出“ ”(半角空格)
(s2) <input type="text" value="1234"/>	

■精度

从(s2)中存储的数据,指定通过[转换指定符]输出到(d)的字符数。

表示精度要在“.”(点号)后面记载数值。如果仅记述“.”(点号)并省略数值,则执行与指定为“.0”时相同的动作。

指定符	说明
“.”(点号)数值	<p>■[转换指定符]为“c”的情况下 忽略[精度]的指定。</p> <p>■[转换指定符]为“d”、“u”、“x”、“X”、“o”的情况下 指定输出字符数。</p> <ul style="list-style-type: none"> 指定的数值为从(s2)获取并转换的字符数及其以下时,忽略[精度]的指定。 <p>(s1) <input type="text" value="%3d"/> %d: 以带符号 10进制数输出</p> <p>(s2) <input type="text" value="1234"/></p> <p>⇒ (d) <input type="text" value="1234"/> 按输出字符数(4位)输出</p> <ul style="list-style-type: none"> 指定的数值大于从(s2)获取并转换的字符数时,在输出字符串前输出“0”。 <p>(s1) <input type="text" value="%5d"/> %d: 以带符号 10进制数输出</p> <p>(s2) <input type="text" value="1234"/></p> <p>⇒ (d) <input type="text" value="01234"/> 将“0”输出至输出字符串前</p> <ul style="list-style-type: none"> [精度]指定为“0”,并且(s2)中指定的转换数据也为“0”时,不输出任何内容。 <p>(s1) <input type="text" value="%0d"/> %d: 以带符号 10进制数输出</p> <p>(s2) <input type="text" value="0"/></p> <p>⇒ (d) <input type="text" value=""/></p> <p>■[转换指定符]为“f”、“e”、“E”的情况下 指定小数点以后的输出字符数。</p> <ul style="list-style-type: none"> 省略数值记载时,输出小数点以后的6位,对输出字符后的第1位数值进行四舍五入。 <p>(s1) <input type="text" value="%3wf"/> %wf: 以单精度实数 小数表现输出</p> <p>(s2) <input type="text" value="1.2345"/> (s2)+1</p> <p>⇒ (d) <input type="text" value="1.235"/> 由于省略了数值的记载, 因此输出小数点以后的3位</p> <ul style="list-style-type: none"> 指定的数值大于从(s2)获取并转换的字符数时,附加“0”输出。但是,根据(s2)中指定的转换数据,也可能输出“0”以外的字符。此外,小数表现时在输出字符串后面附加0,指数表现时在输出字符串后面以及“e”、“E”的前面附加0。 <p>(s1) <input type="text" value="%+- #.0wf"/> %we: 以单精度实数 指数表现输出</p> <p>(s2) <input type="text" value="1.234"/> (s2)+1</p> <p>⇒ (d) <input type="text" value="1.23400e+00"/> 在输出字符串后和 “e”前输出“0”</p>

指定符	说明																																								
“.” (点号) 数值	<p>■ [转换指定符] 为 “g”、“G” 的情况下 指定有效位数。</p> <ul style="list-style-type: none"> 省略数值记载时，输出有效位数6位，对输出字符以后的第1位数值进行四舍五入。 <p>(s1) <code>%3wg</code> %wg: 以单精度实数 小数表现输出</p> <p>⇒ (d) <code>"1.23"</code> 以3位有效位数输出</p> <p>(s2) <code>1.2345</code> (s2)+1</p> <ul style="list-style-type: none"> [精度] 指定为 “.0” 时，与指定为 “.1” 时相同。 (s2) 中指定为 1.234 时，按下述方式输出。 <p>(s1) <code>%0qg</code> %qg: 以双精度实数 指数表现输出</p> <p>⇒ (d) <code>"1"</code> 以1位有效数字输出</p> <p>(s2) <code>1.234</code> (s2)+1 (s2)+2 (s2)+3</p> <p>(s2) 中指定为 0.1234 时，按下述方式输出。</p> <p>(s1) <code>%0qg</code> %qg: 以双精度实数 指数表现输出</p> <p>⇒ (d) <code>"0.1"</code> 以1位有效数字输出</p> <p>(s2) <code>0.1234</code> (s2)+1 (s2)+2 (s2)+3</p> <p>■ [转换指定符] 为 “s” 的情况下 按照 (s2) 中存储的字符数据顺序，输出指定的输出字符数。</p> <ul style="list-style-type: none"> 输出字符数大于 (s2) 中获取的字符数据的字符数的情况下，输出 (s2) 中获取的字符串数据的字符数。 <p>(s1) <code>%10s</code></p> <p>⇒ (d) <code>"ABCDE"</code> 输出 (s2) 中获取的字符 数据的字符数 (5 字符)</p> <p>(s2) <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>b15</td><td>b8</td><td>b7</td><td>.....</td><td>b1</td></tr> <tr><td>42H(B)</td><td>:</td><td>:</td><td>:</td><td>41H(A)</td></tr> <tr><td>44H(D)</td><td>:</td><td>:</td><td>:</td><td>43H(C)</td></tr> <tr><td>00H</td><td>:</td><td>:</td><td>:</td><td>45H(E)</td></tr> </table></p> <ul style="list-style-type: none"> 输出字符数小于 (s2) 中获取的字符数据的字符数的情况下，输出输出字符数的字符数。 <p>(s1) <code>%3s</code></p> <p>⇒ (d) <code>"ABC"</code> 按输出字符数 (3 字符) 输出</p> <p>(s2) <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>b15</td><td>b8</td><td>b7</td><td>.....</td><td>b1</td></tr> <tr><td>42H(B)</td><td>:</td><td>:</td><td>:</td><td>41H(A)</td></tr> <tr><td>44H(D)</td><td>:</td><td>:</td><td>:</td><td>43H(C)</td></tr> <tr><td>00H</td><td>:</td><td>:</td><td>:</td><td>45H(E)</td></tr> </table></p>	b15	b8	b7	b1	42H(B)	:	:	:	41H(A)	44H(D)	:	:	:	43H(C)	00H	:	:	:	45H(E)	b15	b8	b7	b1	42H(B)	:	:	:	41H(A)	44H(D)	:	:	:	43H(C)	00H	:	:	:	45H(E)
b15	b8	b7	b1																																					
42H(B)	:	:	:	41H(A)																																					
44H(D)	:	:	:	43H(C)																																					
00H	:	:	:	45H(E)																																					
b15	b8	b7	b1																																					
42H(B)	:	:	:	41H(A)																																					
44H(D)	:	:	:	43H(C)																																					
00H	:	:	:	45H(E)																																					

■获取数据大小

指定转换 (s2) 中存储的数据的多少字及是否输出到 (d)。

指定符	说明
(省略记载)	<p>以1字单位获取 (s2)。</p> <p>(s1) %d %d: 以带符号 10进制数输出</p> <p style="text-align: center;">⇒ (d) "722"</p> <p>(s2) 02D2H ← 转换1字的数据 (s2)+1 4996H</p>
w	<p>以2字单位获取 (s2)。</p> <p>(s1) %wd %d: 以带符号 10进制数输出</p> <p style="text-align: center;">⇒ (d) "1234567890"</p> <p>(s2) 02D2H ← 转换2字的数据 (s2)+1 4996H</p>
q	以4字单位获取 (s2)。

转换指定符

指定输出到 (d) 的字符串的输出格式。

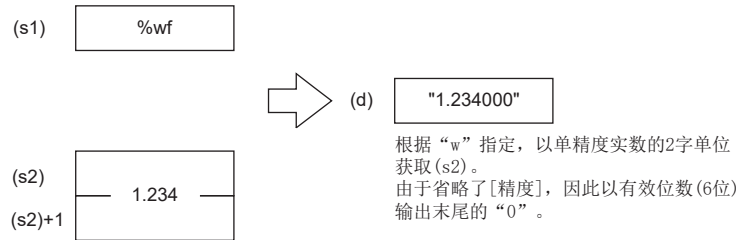
指定符	说明																								
d	以带符号10进制数输出。																								
u	以无符号10进制数输出。																								
x	以无符号16进制数输出。10~15以“a”~“f”显示。																								
X	以无符号16进制数输出。10~15以“A”~“F”显示。																								
o	以无符号8进制数输出。																								
s	<p>获取到 (s2) 中存储的字符串数据的00H(NULL)为止并输出。没有00H(NULL)时, 将变为错误(错误代码: 2820H)。</p> <p>(s1) <input type="text" value="%s"/></p> <p style="text-align: center;">➔ (d) <input abcd\""="" type="text" value="\"/></p> <table border="1" style="margin-left: 20px;"> <tr> <td style="padding: 2px;">(s2)</td> <td style="padding: 2px;">b15 b8 b7 b1</td> <td style="padding: 2px;">42H(B) 41H(A)</td> </tr> <tr> <td style="padding: 2px;">(s2)+1</td> <td style="padding: 2px;"></td> <td style="padding: 2px;">44H(D) 43H(C)</td> </tr> <tr> <td style="padding: 2px;">(s2)+2</td> <td style="padding: 2px;"></td> <td style="padding: 2px;">45H(E) 00H ← 获取至NULL</td> </tr> <tr> <td style="padding: 2px;">(s2)+3</td> <td style="padding: 2px;"></td> <td style="padding: 2px;">47H(G) 46H(F)</td> </tr> </table> <p>00H(NULL) 的存储位置为低位1字节 (b1~b7) 时, 下一个转换数据将从后续软件中指定的数据开始有效。</p> <p>(s1) <input type="text" value="%s%c"/></p> <p style="text-align: center;">➔ (d) <input abcdf\""="" type="text" value="\"/></p> <table border="1" style="margin-left: 20px;"> <tr> <td style="padding: 2px;">(s2)</td> <td style="padding: 2px;">b15 b8 b7 b1</td> <td style="padding: 2px;">42H(B) 41H(A)</td> </tr> <tr> <td style="padding: 2px;">(s2)+1</td> <td style="padding: 2px;"></td> <td style="padding: 2px;">44H(D) 43H(C)</td> </tr> <tr> <td style="padding: 2px;">(s2)+2</td> <td style="padding: 2px;"></td> <td style="padding: 2px;">45H(E) 00H</td> </tr> <tr> <td style="padding: 2px;">(s2)+3</td> <td style="padding: 2px;"></td> <td style="padding: 2px;">47H(G) 46H(F) ← %c指定的转换数据</td> </tr> </table>	(s2)	b15 b8 b7 b1	42H(B) 41H(A)	(s2)+1		44H(D) 43H(C)	(s2)+2		45H(E) 00H ← 获取至NULL	(s2)+3		47H(G) 46H(F)	(s2)	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指定的转换数据
(s2)	b15 b8 b7 b1	42H(B) 41H(A)																							
(s2)+1		44H(D) 43H(C)																							
(s2)+2		45H(E) 00H ← 获取至NULL																							
(s2)+3		47H(G) 46H(F)																							
(s2)	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指定的转换数据																							
c	<p>以1字单位作为ASCII代码(10进制数)获取 (s2) 中存储的字符串数据, 并进行字符转换后输出。</p> <p>(s1) <input type="text" value="%c"/></p> <p style="text-align: center;">➔ (d) <input a\""="" type="text" value="\"/></p> <p style="text-align: center;">以字符输出ASCII代码</p> <p>(s2) <input type="text" value="65"/></p> <p>(s2) 中存储的转换数据仅在低位1字节 (b1~b7) 有效。</p> <p>(s1) <input type="text" value="%c"/></p> <p style="text-align: center;">➔ (d) <input a\""="" type="text" value="\"/></p> <p style="text-align: center;">仅输出低位1字节的字符(“A”)</p> <p>(s2) <input type="text" value="16961"/></p> <p style="text-align: center;">表示“AB”的ASCII代码(10进制数)</p>																								
f	以小数表现输出。单精度实数时7位, 双精度实数时15位为最大有效位数。																								
e	以指数表现输出, 指数部以“e”显示。 单精度实数时7位, 双精度实数时15位为最大有效位数。																								
E	以指数表现输出, 指数部以“E”显示。 单精度实数时7位, 双精度实数时15位为最大有效位数。																								

指定符	说明
g	<p>根据数值大小以小数表现及指数表现输出。指数部以“e”显示。但是，(s2)中指定的转换数据的指数部的值大于或等于[精度]中指定的数值时，以指数表现输出。</p> <ul style="list-style-type: none"> • 数值的绝对值为0.0001及其以上：小数表现 • 数值未满足0.0001：指数表现 <p>单精度实数时7位，双精度实数时15位为最大有效位数。</p> <p>■(s2)中指定的转换数据为0.0001及其以上，并以小数表现输出的情况下</p> <p>(s1) <input type="text" value="%wg"/></p> <p>由于省略了[精度]，因此以有效位数6位输出</p> <p>(d) <input type="text" value="0.0001"/></p> <p>(s2) <input type="text" value="0.0001"/> (s2)+1 <input type="text" value="(1e-4)"/></p> <p>(s2)的指数部分-4小于[精度]的有效位数，因此以小数表现输出</p> <p>■(s2)中指定的转换数据为0.0001及其以上，并以指数表现输出的情况下</p> <p>(s1) <input type="text" value="%qg"/></p> <p>由于省略了[精度]，因此以有效位数6位输出</p> <p>(d) <input type="text" value="1.23457e+06"/></p> <p>(s2) <input type="text" value="1234567.12"/> (s2)+1 <input type="text" value="(1.23456712e+6)"/> (s2)+2 <input type="text" value=""/> (s2)+3 <input type="text" value=""/></p> <p>(s2)的指数部分6为[精度]的有效位数6位及其以上，因此以指数表现输出</p>
G	<p>根据数值大小以小数表现及指数表现输出。指数部以“E”显示。但是，(s2)中指定的转换数据的指数部的值大于或等于[精度]中指定的数值时，以指数表现输出。</p> <ul style="list-style-type: none"> • 数值的绝对值为0.0001及其以上：小数表现 • 数值未满足0.0001：指数表现 <p>单精度实数时7位，双精度实数时15位为最大有效位数。</p> <p>(s1) <input type="text" value="%wG"/></p> <p>(d) <input type="text" value="-9.99E-05"/></p> <p>以指数表现输出</p> <p>(s2) <input type="text" value="-0.0000999"/> (s2)+1 <input type="text" value=""/></p>

在[转换指定符]中使用“f”、“e”、“E”、“g”、“G”的情况下，可以根据[获取数据大小]的指定符选择单精度实数或双精度实数的输出格式。[获取数据大小]指定为“w”时以单精度实数输出，指定为“q”时以双精度实数输出。

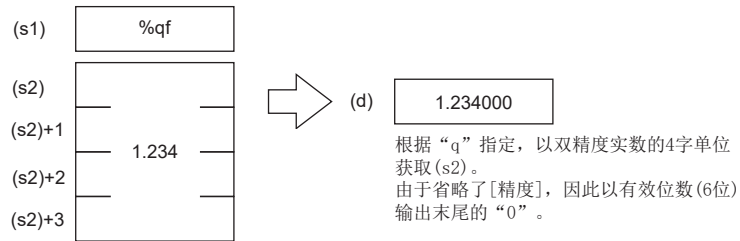
例

以单精度实数输出的情况下



例

以双精度实数输出的情况下

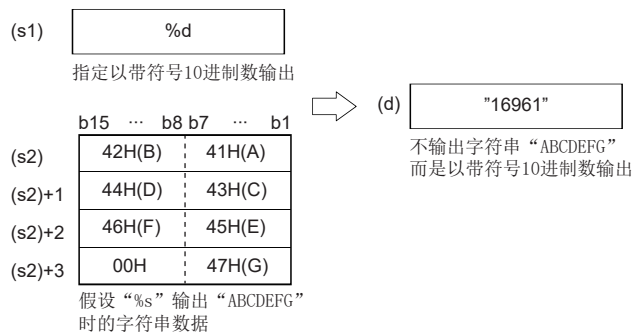


注意事项

显示格式的组合和要转换的数据的格式必须一致。将输出预期外的字符串。

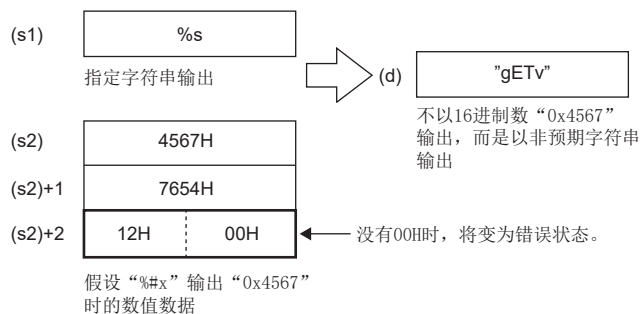
例

转换的数据设置为字符串数据，显示格式设置为数值数据的情况下



例

转换的数据设置为数值数据，显示格式设置为字符串数据的情况下

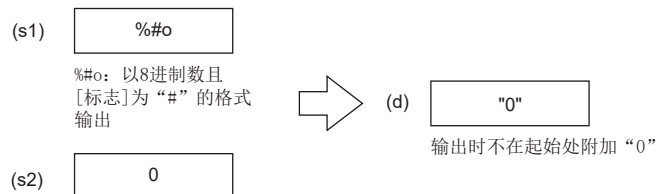


■标志

- [转换指定符]为“s”的情况下，忽略[标志]的指定。
- [最小字段宽度]为从(s2)获取并转换的字符数及其以下时，忽略“-”的指定符。
- [转换指定符]为“u”、“x”、“X”、“o”、“s”、“c”的情况下，忽略“+”和“ ”(半角空格)的指定符。
- [转换指定符]为“d”、“u”、“s”、“c”时或[转换指定符]指定为“o”、“x”、“X”，(s2)中指定为“0”时，忽略“#”的指定符。

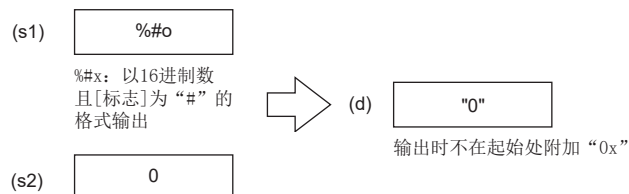
例

[转换指定符]指定为“o”，[标志]指定为“#”时，按下述方式输出。



例

[转换指定符]指定为“x”，[标志]指定为“#”时，按下述方式输出。



- 使用了“-”、“ ”(半角空格)的情况下，输出0或正数以外的值时将被忽略。
- 未指定“+”且输出负数时，在起始处附加“-”。

■最小字段值

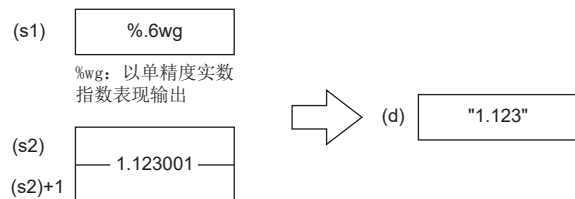
[最小字段宽度]指定为超过255个字符时，将变为错误(错误代码: 329CH)。

■精度

- [精度]指定为超过255个字符时，将变为错误(错误代码: 329CH)。
- [转换指定符]为“g”、“G”的情况下，不输出小数点以后的末尾的“0”。此外，小数点以后没有数值的情况下，“.”(点号)也不显示。

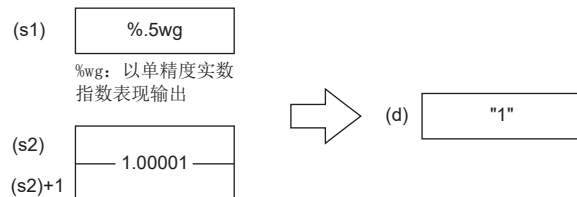
例

有效位数为6位时，小数点以后的末尾的0(小数点以后的第4位和第5位的0)不输出。



例

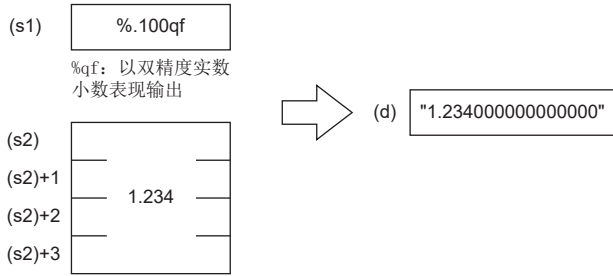
有效位数为5位时，小数点以后全部为0(无数值)，因此未输出“.”(点号)。



- [转换指定符]为“f”、“e”、“E”、“g”、“G”时，[精度]中指定的数值大于输出结果的最大有效位数时，以最大有效位数输出。

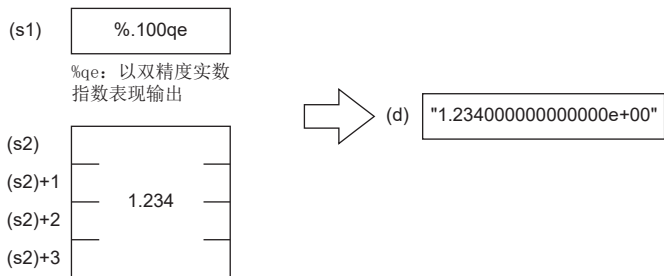
例

[转换指定符]指定为“f”，[精度]中指定的数值大于双精度实数的最大有效位数时，以双精度实数的最大有效位数15位输出。



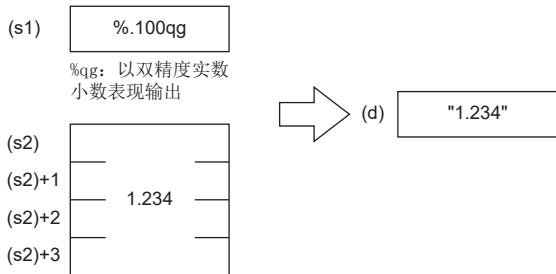
例

[转换指定符]指定为“e”，[精度]中指定的数值大于双精度实数的最大有效位数时，以双精度实数的最大有效位数15位输出。



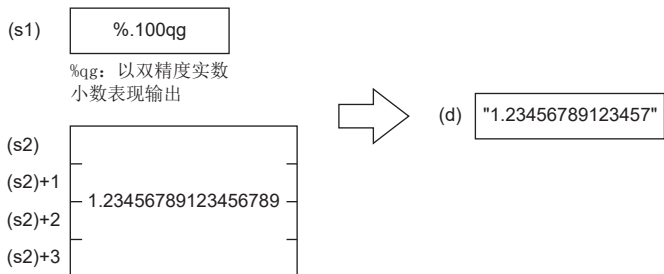
例

[转换指定符]指定为“g”，[精度]中指定的数值大于双精度实数的最大有效位数，且(s2)中存储的数值的位数为最大有效位数及其以下时，不输出小数点以后的末尾的“0”。



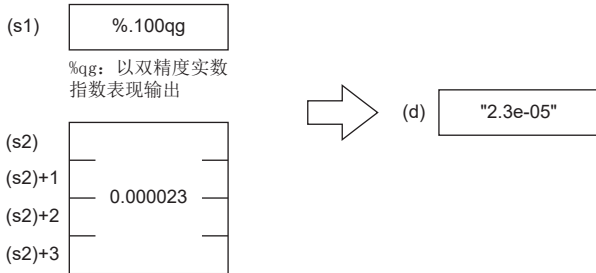
例

[转换指定符]指定为“g”，[精度]中指定的数值大于双精度实数的最大有效位数时，且(s2)中存储的数值的位数为最大有效位数及其以上时，对第16位进行了四舍五入后以最大有效位数15位输出。



例

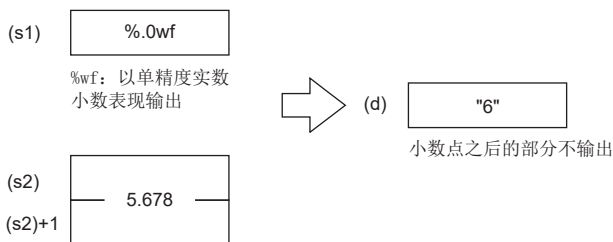
[转换指定符]指定为“g”，[精度]中指定的数值大于双精度实数的最大有效位数，且以指数表现输出时，不输出小数点以后的末尾的“0”。



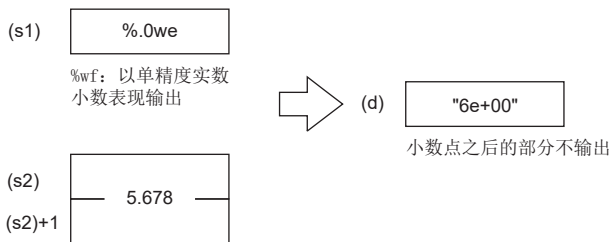
• [转换指定符]指定为“f”、“e”、“E”，[精度]指定为“.0”且[标志]未指定为“#”时，小数点以后不输出。

例

[转换指定符]为“f”的情况下，[标志]指定为“.0”时，按下述方式表现。

**例**

[转换指定符]为“e”的情况下，[标志]指定为“.0”时，按下述方式表现。



• [转换指定符]为“d”、“f”、“e”、“E”、“g”、“G”时，转换的字符串的符号不包含在[精度]中指定的输出字符串中。

■获取数据大小

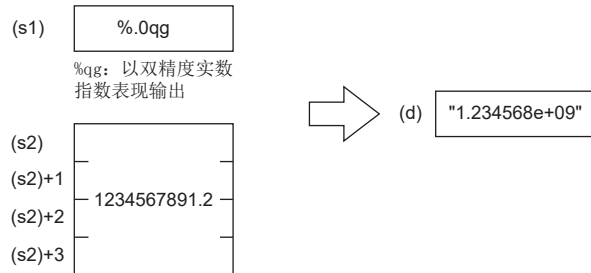
- [转换指定符]为“s”、“c”的情况下，忽略[获取数据大小]的指定。
- [转换指定符]为“f”、“e”、“E”、“g”、“G”的情况下，[获取数据大小]必须指定为“w”或“q”。省略了记载时，将变为错误(错误代码：3299H)。

■转换指定符

以指数表现输出时，指数以2位及以上输出。

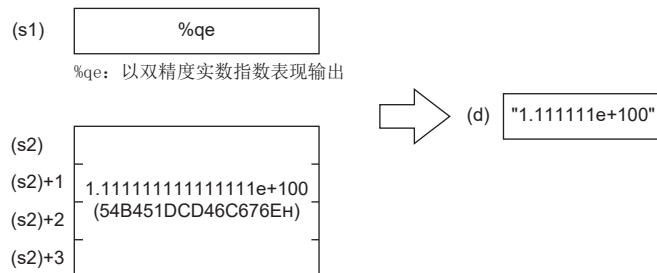
例

(s2)中存储的数据以指数表现时，如果指数部的位数为2位，则如下以2位输出指数部。



例

(s2)中存储的数据以指数表现时，如果指数部的位数为3位，则如下以3位输出指数部。



出错

错误代码 (SD0)	内容
2820H	从(s2)中指定的软件编号开始至相应软件的最终软件编号为止，不存在00H时。 ([转换指定符]中指定了“s”时)
	(s2)+[获取数据大小]中指定的数据大小超过(s2)的最终软件编号时。 ([转换指定符]中指定了“s”以外时)
	从(s1)中指定的软件编号开始至相应软件的最终软件编号为止，不存在00H时。
2821H	(s1)与(d)中指定的字符串的存储软件编号重复时。
3285H	(s1)的字符串超过了255字符时。
3298H	(s1)中如下所示使用了显示格式时。 <ul style="list-style-type: none"> • 使用可使用的指定符以外的指定符 • 以指定的格式以外的顺序使用显示格式 • 1个显示格式内多次使用[精度]、[获取数据大小]
3299H	(s1)中没有指定显示格式时。
329AH	(s2)中指定的字符串超过255字符时。 ([转换指定符]中指定了“s”时)
3286H	(d)中存储的字符串为超出可输出范围的数据时。 <ul style="list-style-type: none"> • 合并后的字符串超过255字符。 • 指定的软件编号及其以后，相应软件/标签的分配范围中转换的字符串无法全部存储时。
329CH	[最小字段宽度]或[精度]指定为超过255个字符时。

Unicode字符串→移位JIS字符串转换

WS2SJIS (P)

将Unicode字符串转换为移位JIS字符串。

梯形图	ST
	ENO:=WS2SJIS (EN, s, d) ; ENO:=WS2SJISP (EN, s, d) ;

FBD/LD

■执行条件

指令	执行条件
WS2SJIS	
WS2SJISP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s)	存储了转换的字符串的起始软元件	—	Unicode字符串	ANYSTRING_DOUBLE
(d)	存储转换后的字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

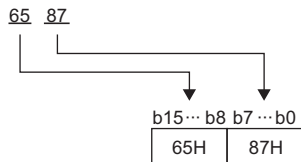
操作数	位		字		双字		间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K	H	
(s)	—	—	○	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	○	—	—	—	—

功能

- 将(s)中指定的Unicode字符串转换为移位JIS字符串后，存储到(d)中。
- (s)的Unicode字符串，应以小端字节序进行指定。

例

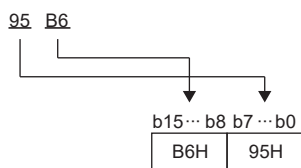
在Unicode字符串的“6587H”中指定“6587H”。



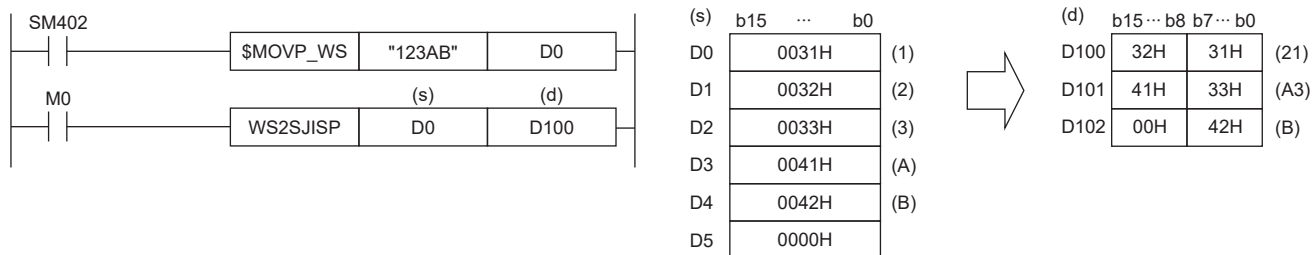
- (d)的移位JIS字符串以大端字节序存储。

例

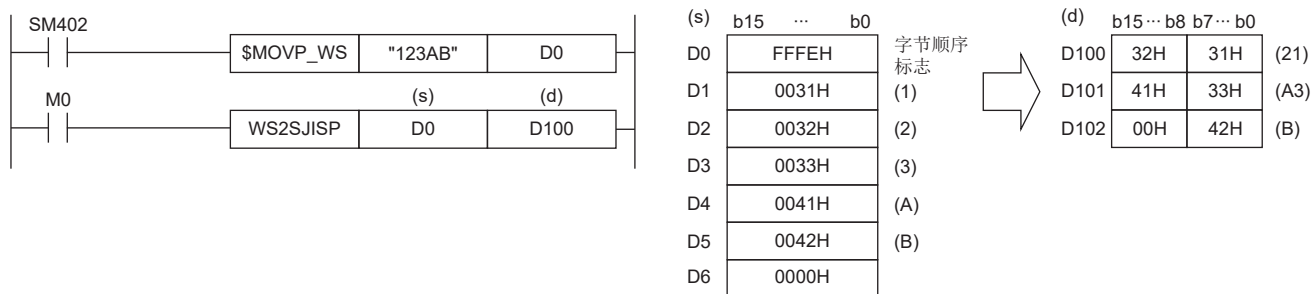
在移位JIS字符串的“95B6H”中指定“B695H”。



- 没有字节顺序标志情况下的Unicode→移位JIS转换的动作如下所示。



- 附加了字节顺序标志(FE FFH)情况下的Unicode→移位JIS转换的动作如下所示。



出错

错误代码(SD0)	内容
2821H	(s)、(d)中指定的范围重复的情况下。
3281H	(s)中指定的字符串中，附加了字节顺序标志FE FFH(大端字节序)的情况下。 (s)中指定的范围中，包含有不能转换的字符代码的情况下。
3285H	(s)中指定的字符串超过了16383字符的情况下。 ^{*1}
3286H	(d)中指定的软元件编号及其以后，相应软元件/标签的分配范围中转换的字符串无法全部存储时。

*1 以汉字等的移位JIS代码表示的情况下变为2字节数据的字符，每个字符应按2计数。

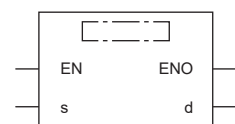
移位JIS字符串→Unicode字符串转换(无字节顺序标志)

SJIS2WS(P)

将移位JIS字符串转换为Unicode字符串。

梯形图	ST
	ENO:=SJIS2WS(EN, s, d); ENO:=SJIS2WSP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
SJIS2WS	
SJIS2WSP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储了转换的字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储转换后的字符串的起始软元件	—	Unicode字符串	ANYSTRING_DOUBLE
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

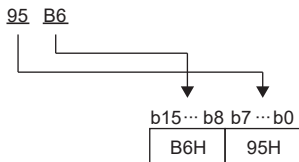
操作数	位		字		双字			间接指定	常数				其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	—	—	○	—	—	—	—	○	—	—	○	—	
(d)	—	—	○	—	—	—	—	○	—	—	—	—	

功能

- 将(s)中指定的移位JIS字符串转换为Unicode字符串后，存储到(d)中。
- (s)的移位JIS字符串，应以大端字节序进行指定。

例

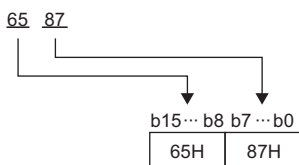
在移位JIS字符串的“95B6H”中指定“B695H”。



- (d)的Unicode字符串，以小端字节序存储。

例

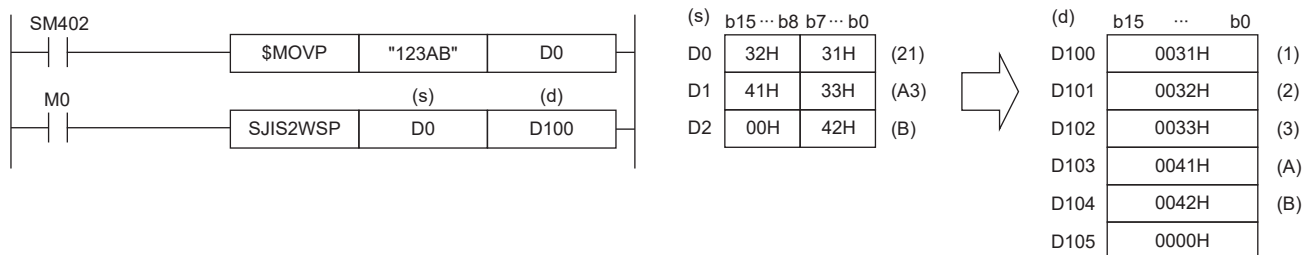
在Unicode字符串的“6587H”中指定“6587H”。



- 在SJIS2WS(P)指令中，不在(d)的起始处附加字节顺序标志。希望附加字节顺序标志的情况下，应使用SJIS2WSB(P)指令。

☞ 769页 SJIS2WSB(P)

- 移位JIS→Unicode转换的动作如下所示。



出错

错误代码(SD0)	内容
2821H	(s)、(d)中指定的范围重复的情况下。
3281H	(s)中指定的范围中，包含有不能转换的字符代码的情况下。
3285H	(s)中指定的字符串超过了16383字符的情况下。 ^{*1}
3286H	(d)中指定的软元件编号及其以后，相应软元件/标签的分配范围中转换的字符串无法全部存储时。

*1 以汉字等的移位JIS代码表示的情况下变为2字节数据的字符，每个字符应按2计数。

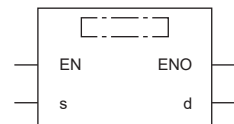
移位JIS字符串→Unicode转换(有字节顺序标志)

SJIS2WSB(P)

将移位JIS字符串转换为Unicode字符串后，在起始处附加字节顺序标志。

梯形图	ST
	ENO:=SJIS2WSB(EN, s, d); ENO:=SJIS2WSBP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
SJIS2WSB	
SJIS2WSBP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储了转换的字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储转换后的字符串的起始软元件	—	Unicode字符串	ANYSTRING_DOUBLE
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

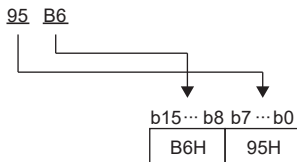
操作数	位		字		双字		间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K	H	
(s)	—	—	○	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	○	—	—	—	—

功能

- 将(s)中指定的移位JIS字符串转换为Unicode字符串后，在起始处附加字节顺序标志后存储到(d)中。
- (s)的移位JIS字符串，应以大端字节序进行指定。

例

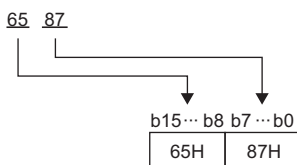
在移位JIS字符串的“95B6H”中指定“B695H”。



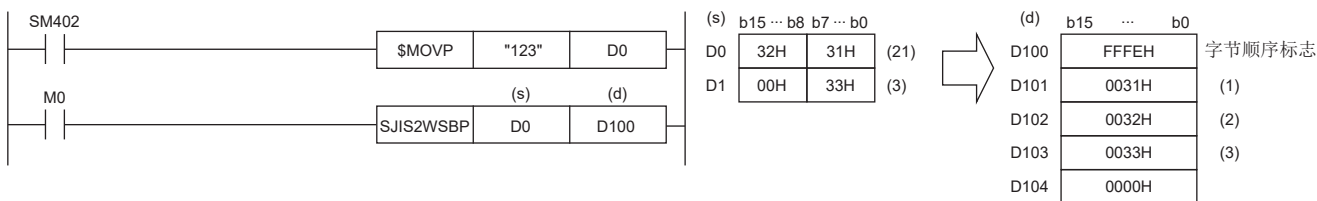
- (d)的Unicode字符串，以小端字节序存储。

例

在Unicode字符串的“6587H”中指定“6587H”。



- 移位JIS→Unicode转换的动作如下所示。



出错

错误代码(SD0)	内容
2821H	(s)、(d)中指定的范围重复的情况下。
3281H	(s)中指定的范围中，包含有不能转换的字符代码的情况下。
3285H	(s)中指定的字符串超过了16383字符的情况下。 ^{*1}
3286H	(d)中指定的软元件编号及其以后，相应软元件/标签的分配范围中转换的字符串无法全部存储时。

*1 以汉字等的移位JIS代码表示的情况下变为2字节数据的字符，每个字符应按2计数。

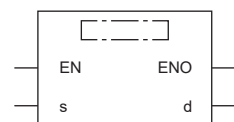
字符串的长度检测

LEN(P)

对指定的字符串的长度进行检测。

梯形图	ST*1
	ENO:=LENP(EN, s, d);

FBD/LD*1



*1 在ST、FBD/LD中不支持LEN指令。应使用通用函数的LEN。

☞ 1317页 LEN(E)

■执行条件

指令	执行条件
LEN	
LENP	

设置数据

■内容、范围、数据类型

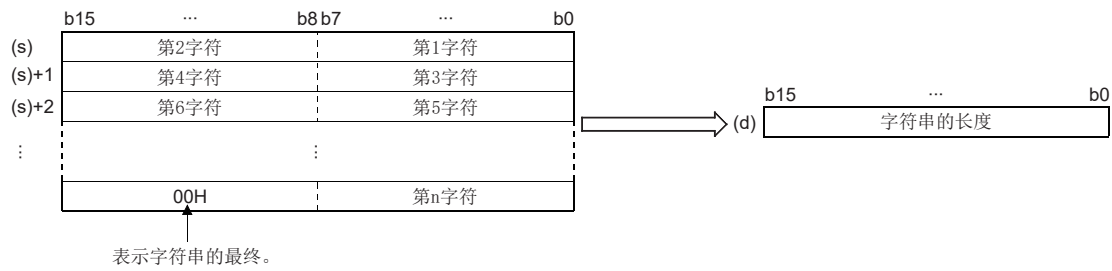
操作数	内容	范围	数据类型	数据类型(标签)
(s)	字符串或存储了字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储检测的字符串长度的软元件编号	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

检测 (s) 中指定的字符串的长度后，存储到 (d) 中指定的软元件编号及其以后。将 (s) 中指定的软元件编号开始，至存储的 00H 的软元件编号为止的数据作为字符串处理。



例

(s) 及其以后存储了“ABCDEFGH I”的情况下



出错

错误代码 (SD0)	内容
2820H	(s) 中指定的软元件编号及其以后，相应软元件/标签的分配范围中未设置 00H 时。
3285H	(s) 的字符串超过了 16383 字符时。

从字符串的右侧开始提取

RIGHT(P)

从字符串数据的右侧开始提取(n)字符的数据。

梯形图	ST*1
	ENO:=RIGHTP(EN, s, n, d);

FBD/LD*1

*1 在ST、FBD/LD中不支持RIGHT指令。应使用通用函数的RIGHT。

☞ 1319页 LEFT(E)、RIGHT(E)

■执行条件

指令	执行条件
RIGHT	
RIGHTP	

设置数据

■内容、范围、数据类型

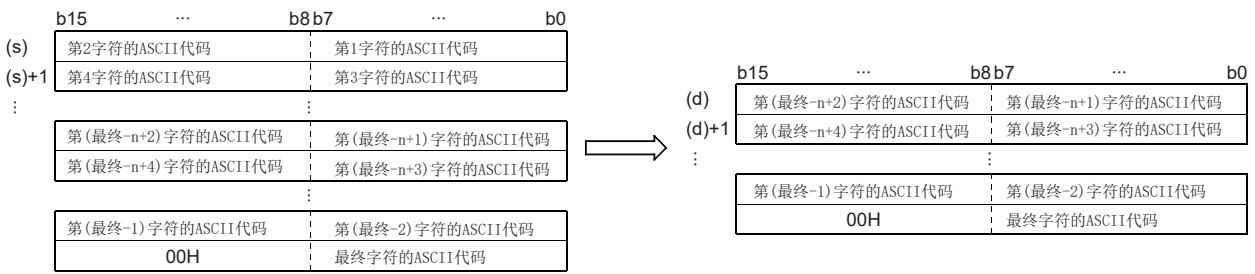
操作数	内容	范围	数据类型	数据类型(标签)
(s)	字符串或存储了字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储从(s)的右侧开始的(n)字符的字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(n)	提取的字符数	1~16383	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

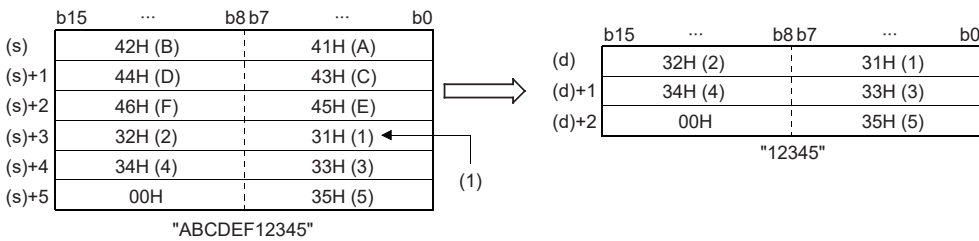
功能

- 对(s)中指定的软元件编号及其以后存储的字符串数据，将字符串的右侧(字符串的最终)开始的(n)字符的数据存储到(d)中指定的软元件编号及其以后。



例

(n)=5的情况下



(1) 第5字符的ASCII代码

- 表示字符串的最后的NULL代码(00H)将被自动附加到字符串数据的最后。
- (n)中指定的字符数为0的情况下，(d)中将存储NULL代码(00H)。

出错

错误代码(SD0)	内容
2820H	(s)中指定的软元件编号及其以后，相应软元件/标签的分配范围中未设置00H时。
3285H	(s)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> (s)的字符串超过了16383字符 (s)的字符串为0字符 (n)超过了(s)中指定的字符数时。
3286H	(d)中指定的软元件编号及其以后，相应软元件/标签的分配范围中获取的字符串无法全部存储时。

从字符串的左侧开始提取

LEFT(P)

将字符串数据的左侧开始的(n)字符的数据存储到(d)中指定的软元件编号及其以后。

梯形图	ST*1
	ENO:=LEFTP(EN, s, n, d);

FBD/LD*1

*1 在ST、FBD/LD中不支持LEFT指令。应使用通用函数的LEFT。

☞ 1319页 LEFT(E)、RIGHT(E)

■执行条件

指令	执行条件
LEFT	
LEFTP	

设置数据

■内容、范围、数据类型

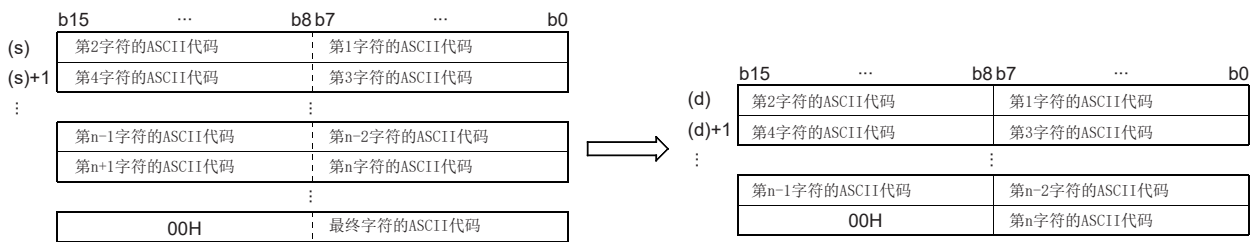
操作数	内容	范围	数据类型	数据类型(标签)
(s)	字符串或存储了字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储从(s)的左侧开始的(n)字符的字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(n)	提取的字符数	1~16383	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

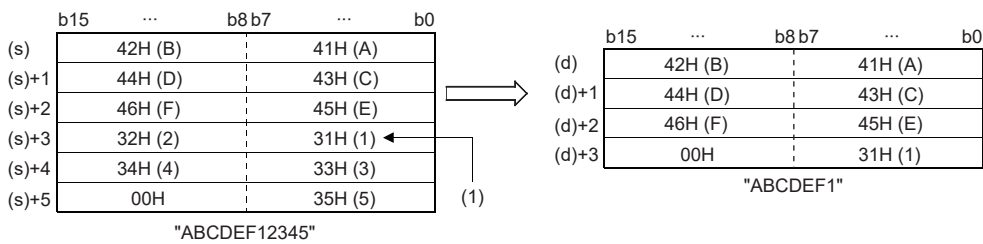
功能

- 对(s)中指定的软元件编号及其以后存储的字符串数据, 将字符串的左侧(字符串的起始)开始的(n)字符的数据存储到(d)中指定的软元件编号及其以后。



例

(n)=7的情况下



(1) 第7字符的ASCII代码

- 表示字符串的最后的NULL代码(00H)将被自动附加到字符串数据的最后。
- (n)中指定的字符数为0的情况下, (d)中将存储NULL代码(00H)。

出错

错误代码(SD0)	内容
2820H	(s)中指定的软元件编号及其以后, 相应软元件/标签的分配范围中未设置00H时。
3285H	(s)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> (s)的字符串超过了16383字符 (s)的字符串为0字符
	(n)超过了(s)中指定的字符数时。
3286H	(d)中指定的软元件编号及其以后, 相应软元件/标签的分配范围中获取的字符串无法全部存储时。

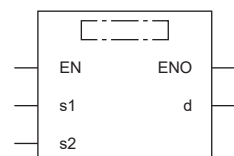
字符串中的任意提取

MIDR(P)

将字符串数据中的任意位置的数据存储到(d)中指定的软元件编号及其以后。

梯形图	ST
	ENO:=MIDR(EN, s1, s2, d); ENO:=MIDRP(EN, s1, s2, d);

FBD/LD



■执行条件

指令	执行条件
MIDR	
MIDRP	

设置数据

■内容、范围、数据类型

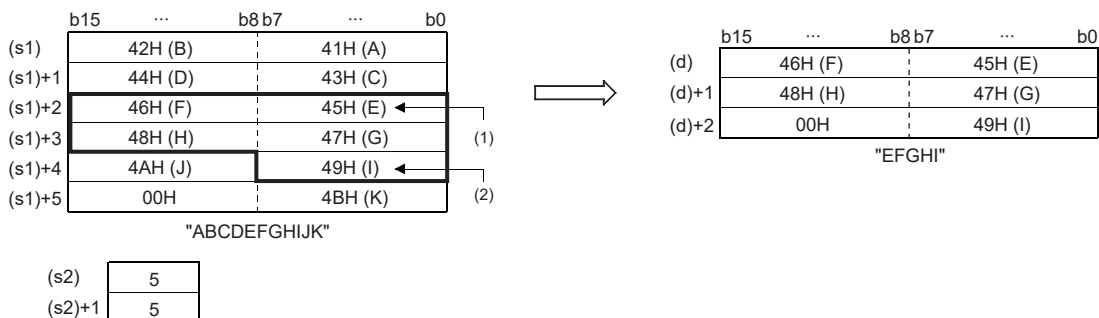
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	字符串或存储了字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储运算结果字符串数据的起始软元件	—	字符串	ANYSTRING_SINGLE
(s2)	存储起始字符位置及字符数的起始软元件 (s2): 起始字符的位置, (s2)+1: 字符数	—	带符号BIN16位	ANY16_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—

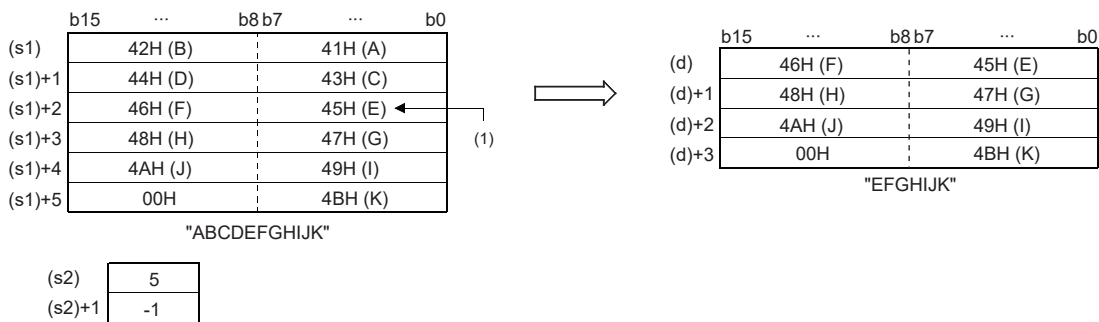
功能

- 对(s1)中指定的软元件编号及其以后存储的字符串数据，将(s2)中指定位置开始的(s2)+1中指定字符的数据存储到(d)中指定的软元件编号及其以后。



- (1) 第5字符的位置 (s2)
 (2) 第5字符的ASCII代码 (s2)+1

- 表示字符串的最后的NULL代码(00H)将被自动附加到字符串数据的最后。
- (s2)+1中指定的字符数为0的情况下不进行处理。
- (s2)+1中指定的字符数为-1的情况下，将(s1)中指定的最终字符数据为止的数据存储到(d)中指定的软元件及其以后。



- (1) 第5字符的位置 (s2)

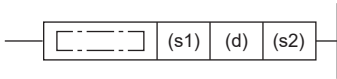
出错

错误代码(SD0)	内容
2820H	(s1)中指定的软元件编号及其以后，相应软元件/标签的分配范围中不存在00H时。
3285H	(s1)的字符串超过了16383字符时。 (s2)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> (s2)的值为0或其以下 (s2)+1的值为有效值(-1、0、1或其以上)以外 (s2)的值超过了(s1)的字符数 (s2)与(s2)+1的加法运算后的值超过了(s1)的字符数+1
3286H	(d)中指定的软元件编号及其以后，相应软元件/标签的分配范围中获取的字符串无法全部存储时。

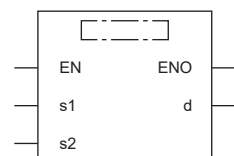
字符串中的任意替换

MIDW(P)


将字符串数据中任意位置的数据以任意字符串进行替换。

梯形图	ST
	ENO:=MIDW(EN, s1, s2, d); ENO:=MIDWP(EN, s1, s2, d);

FBD/LD



■执行条件

指令	执行条件
MIDW	
MIDWP	

设置数据

■内容、范围、数据类型

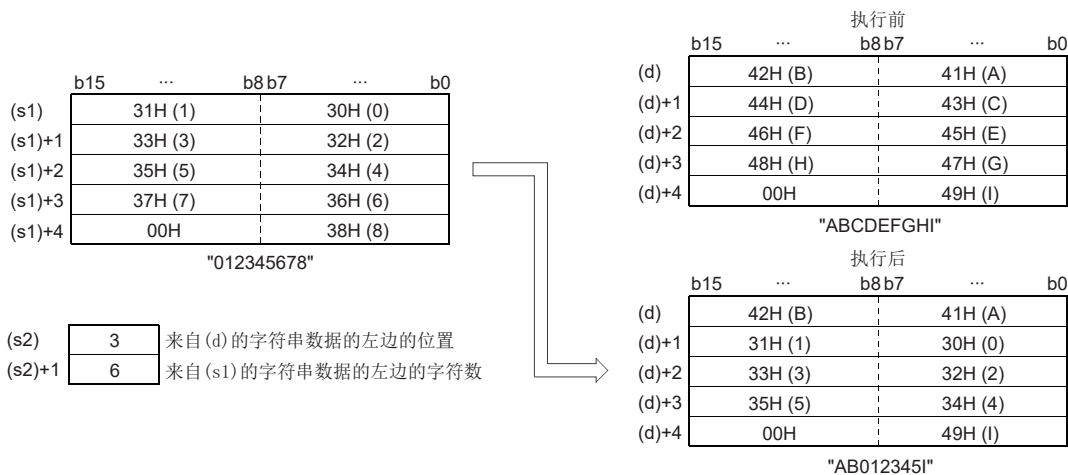
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	字符串或存储了字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储运算结果字符串数据的起始软元件	—	字符串	ANYSTRING_SINGLE
(s2)	存储起始字符位置及字符数的起始软元件 (s2): 起始字符的位置, (s2)+1: 字符数	—	带符号BIN16位	ANY16_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

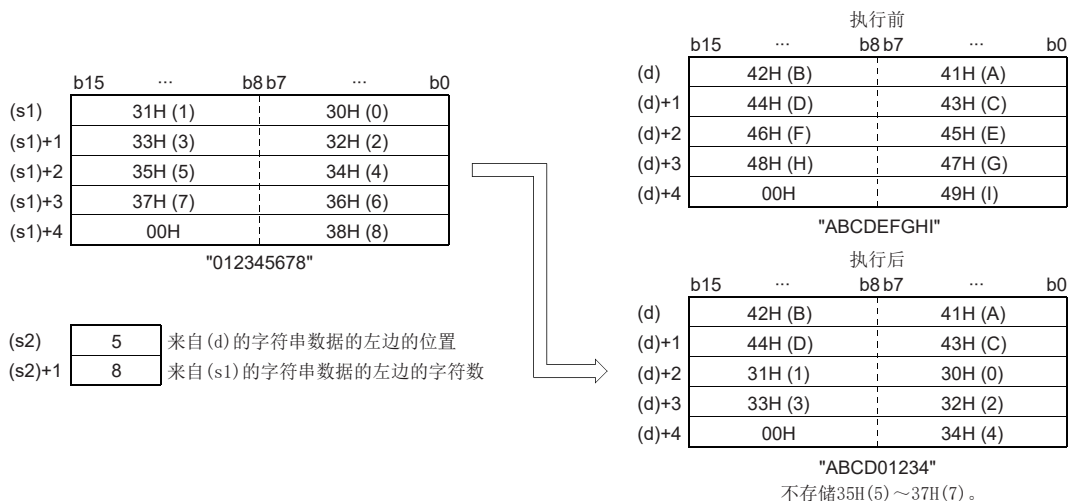
操作数	位		字		双字			间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	○	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	—	—	—	—

功能

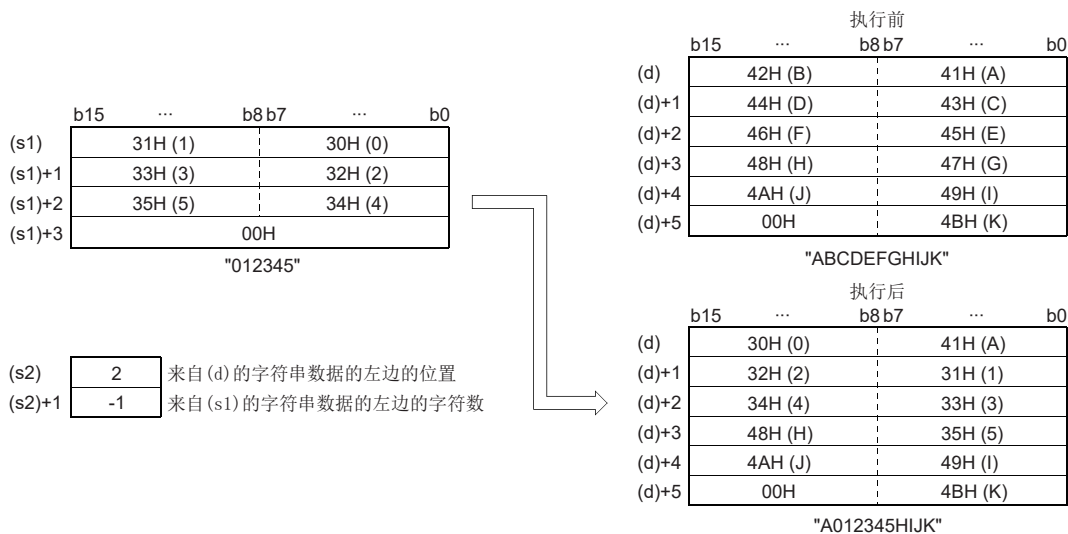
- 对(s1)中指定的软件编号及其以后存储的字符串数据, 将(s2)+1中指定的字符的数据, 存储到(d)中指定的软件编号及其以后存储的字符串数据的(s2)中指定的位置及其以后。



- 表示字符串的最后的NULL代码(00H)将被自动附加到字符串数据的最后。
- (s2)+1中指定的字符数为0的情况下不进行处理。
- (s2)+1中指定的字符数超过了(d)中指定的字符串数据的最终字符的情况下, 存储(d)的最终字符为止的数据。



- (s2)+1中指定的字符数为-1的情况下, 将(s1)中指定的最终字符数据为止的数据存储到(d)中指定的软件及其以后。

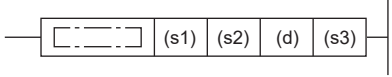


错误代码 (SD0)	内容
2820H	(s1)中指定的软元件编号及其以后，相应软元件/标签的分配范围中不存在00H时。
	(d)中指定的软元件编号及其以后，相应软元件/标签的分配范围中不存在00H时。
3285H	(s1)的字符串超过了16383字符时。
	(d)的字符串超过了16383字符时。
	(s2)中设置了超出允许指定范围的数据时。
	<ul style="list-style-type: none"> • (s2)的值为0或其以下 • (s2)+1的值为有效值(-1、0、1或其以上)以外 • (s2)的值超过了(d)的字符数 • (s2)+1的值超过了(s1)的字符数时。

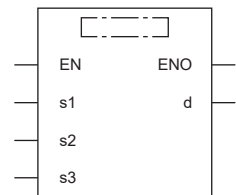
字符串查找

INSTR(P)


搜索字符串数据中的指定字符串。

梯形图	ST
	ENO:=INSTR(EN, s1, s2, s3, d); ENO:=INSTRP(EN, s1, s2, s3, d);

FBD/LD



■执行条件

指令	执行条件
INSTR	
INSTRP	

设置数据

■内容、范围、数据类型

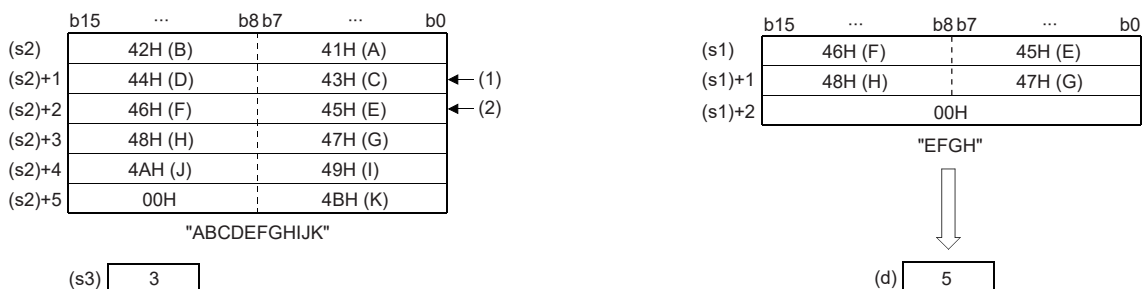
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	搜索字符串或存储了搜索字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(s2)	搜索的字符串或存储了搜索的字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储搜索结果的软元件	—	带符号BIN16位	ANY16
(s3)	搜索开始位置	1~16383	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 从(s2)中指定的软元件编号及其以后存储的字符串数据的左侧第(s3)字符开始，搜索(s1)中指定的软元件编号及其以后存储的字符串数据，将搜索结果存储到(d)中指定的软元件中。搜索结果将存储(s2)中指定的字符串数据的起始字符开始的第几字符。



(1) 开始查找位置(s3): 第3字符

(2) 从起始字符开始的第5字符

- 没有一致的字符串数据的情况下，(d)中将存储0。

出错

错误代码 (SD0)	内容
2820H	(s1)中指定的软元件及其以后，相应软元件/标签的分配范围中无NULL代码(00H)时。
	(s2)中指定的软元件及其以后，相应软元件/标签的分配范围中无NULL代码(00H)时。
3285H	(s1)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> 设置的字符串为0字符 设置的字符串超过了16383字符
	(s2)中设置的字符串超过了16383字符时。
	(s3)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> (s3)的值超过了(s2)的字符数 (s3)的值为负的数或0

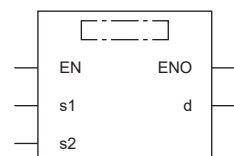
字符串插入

STRINS (P)

将任意字符串数据插入到字符串数据任意插入位置。

梯形图	ST
	ENO:=STRINS (EN, s1, s2, d); ENO:=STRINSP (EN, s1, s2, d);

FBD/LD



■执行条件

指令	执行条件
STRINS	
STRINSP	

设置数据

■内容、范围、数据类型

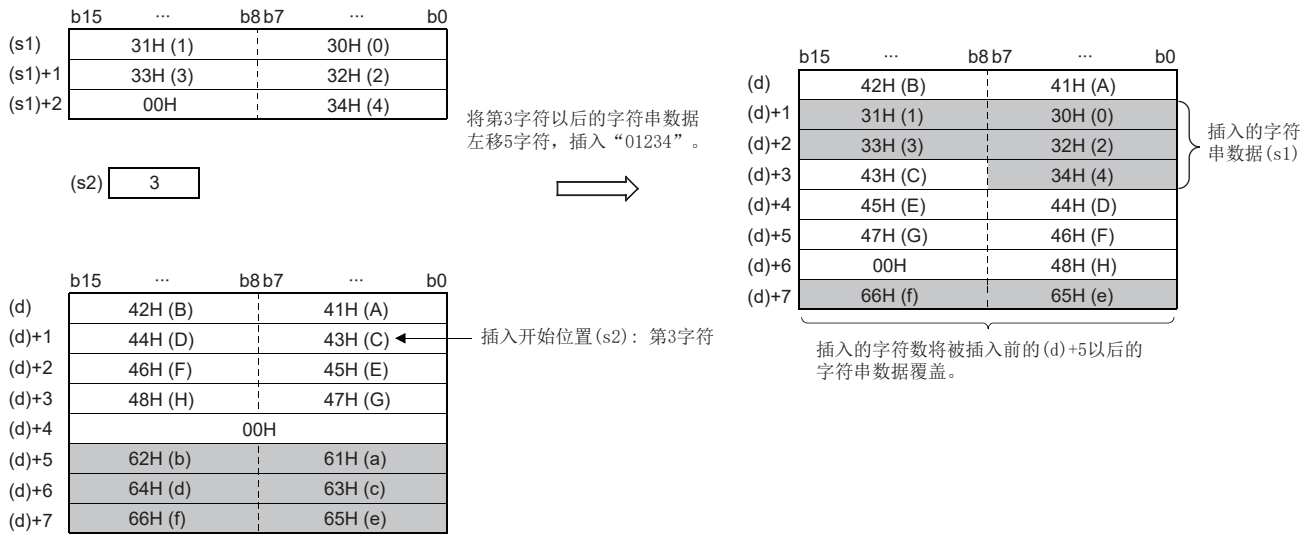
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	插入字符串或存储了插入字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储插入字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(s2)	插入位置(字节单位)	1~16383	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 将(s1)中指定的字符串数据，插入到(d)中指定的字符串数据的起始开始的第(s2)字符(插入位置)处。



- 插入后的字符串(s1)+(d)为偶数的情况下，字符串的最后的下一个软元件(1字)中将存储NULL代码(00H)。
- 插入后的字符串(s1)+(d)为奇数的情况下，字符串的最后的软元件(高位8位)中将存储NULL代码(00H)。
- (s2)中指定了(d)的字符数+1的情况下，将(d)的字符串的最后与(s1)的字符串合并。

出错

错误代码(SD0)	内容
2820H	(s1)中指定的软元件及其以后，相应软元件/标签的分配范围中无NULL代码(00H)时。
	(d)中指定的软元件及其以后，相应软元件/标签的分配范围中无NULL代码(00H)时。
2821H	字符串(s1)与字符串(d)的软元件有部分重复时。
	插入后的字符串(s1)+(d)与(s1)的字符串存储软元件重复时。
3285H	(s1)中指定的字符串的字符数超过了16383字符时。
	(s2)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> 指定的值超出了字符串(d)的字符数+1。 指定的值不在下述范围内。 $1 \leq (s2) \leq 16383$
	(d)中指定的字符串的字符数超过了16383字符时。
3286H	插入后的字符串(s1)+(d)为超出允许输出范围的数据时。 <ul style="list-style-type: none"> 插入后的字符串的字符数超过了16383字符。 (d)中指定的软元件编号及其以后，相应软元件/标签的分配范围中插入后的字符串无法全部存储时。

字符串删除

STRDEL (P)

从字符串数据的指定的位置删除(n)字符的数据。

梯形图	ST
	ENO:=STRDEL(EN, s, n, d); ENO:=STRDELP(EN, s, n, d);

FBD/LD



■执行条件

指令	执行条件
STRDEL	
STRDELP	

设置数据

■内容、范围、数据类型

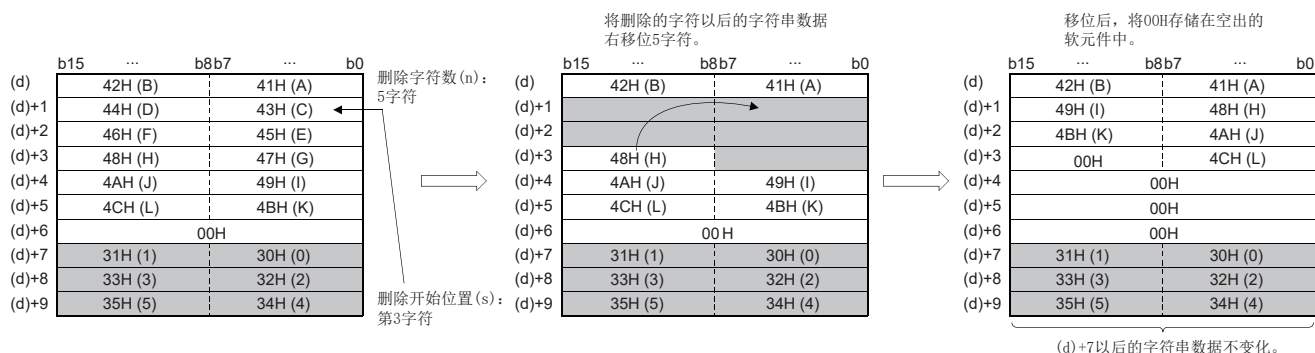
操作数	内容	范围	数据类型	数据类型(标签)
(d)	存储了删除字符串的起始软元件	—	字符串	ANYSTRING_SINGLE
(s)	删除开始位置	1~16383	带符号BIN16位	ANY16
(n)	删除字符数	0~16384-(s)	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字				双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	

功能

- 从(d)中指定的字符串数据的起始开始的第(s)字符中指定的位置(删除开始位置)开始,删除(n)字符的数据。



- 删除后, 字符串(d)为偶数的情况下, 字符串的最后的下一个软件件(1字)中将存储NULL代码(00H)。
- 删除后, 字符串(d)为奇数的情况下, 字符串的最后的软件件(高位8位)中将存储NULL代码(00H)。
- 将删除的字符串及其以后的字符串右移(n)字符后, 空出的软件件中存储NULL代码(00H)。

出错

错误代码(SD0)	内容
2820H	(d)中指定的软件件及其以后, 相应软件件/标签的分配范围中无NULL代码(00H)时。
3285H	(d)中指定的字符串的字符数超过了16383字符时。 (s)中设置了超出允许指定范围的数据时。 • 指定的值不在下述范围内 $1 \leq (s) \leq 16383$ • 指定的值超出了字符串(d)的字符数 (n)中设置了超出允许指定范围的数据时。 • 指定的值超过了字符串(d)的(s)开始至最终字符为止的字符数 • 指定的值为负数

17 实数处理

17.1 浮点指令

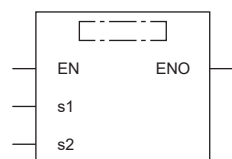
单精度实数比较

LDE□、ANDE□、ORE□

将单精度实数通过常开触点处理进行比较运算。

梯形图	ST
	<pre> ENO:=LDE_□(EN, s1, s2); ENO:=ANDE_□(EN, s1, s2); ENO:=ORE_□(EN, s1, s2); (□中为EQ、NE、GT、LE、LT、GE。)*1 </pre>
<p>(□输入E=、E<>、E>、E<=、E<、E>=。)</p>	

FBD/LD



(□中放入LDE_、ANDE_、ORE_与EQ、NE、GT、LE、LT、GE的组合。)*1

*1 EQ为=、NE为<>、GT为>、LE为<=、LT为<、GE为>=。

■执行条件

指令	执行条件
LDE□、ANDE□、ORE□	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	比较数据或存储了比较数据的起始软元件	$0, 2^{-126} \leq (s1) < 2^{128}$	单精度实数	ANYREAL_32
(s2)	比较数据或存储了比较数据的起始软元件	$0, 2^{-126} \leq (s2) < 2^{128}$	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	○	○	○	○	○	—	○	—	—	

功能

- 将(s1)中指定的软元件的单精度实数与(s2)中指定的软元件的单精度实数通过常开触点处理进行比较运算。
- 各指令的比较运算结果如下所示。

指令符号(梯形图、FBD/LD)	条件	比较运算结果
E=、EQ	$(s1)=(s2)$	导通状态(ENO为ON)
E<>、NE	$(s1)\neq(s2)$	
E>、GT	$(s1)>(s2)$	
E<=、LE	$(s1)\leq(s2)$	
E<、LT	$(s1)<(s2)$	
E>=、GE	$(s1)\geq(s2)$	
E=、EQ	$(s1)\neq(s2)$	非导通状态(ENO为OFF)
E<>、NE	$(s1)=(s2)$	
E>、GT	$(s1)\leq(s2)$	
E<=、LE	$(s1)>(s2)$	
E<、LT	$(s1)\geq(s2)$	
E>=、GE	$(s1)<(s2)$	

- (s1)、(s2)的内容超出设置数据的范围的情况下,变为非导通(ENO为OFF)。
- 通过FBD/LD使用了LDE_□指令的情况下,应将EN指定为左母线,或始终ON的变量部件/常数部件。
- 将ORE_□指令通过FBD/LD使用的情况下,将EN指定为TRUE时ENO将ON。EN不成为执行条件。
- 通过工程工具设置输入值的情况下,有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项,请参阅下述内容。

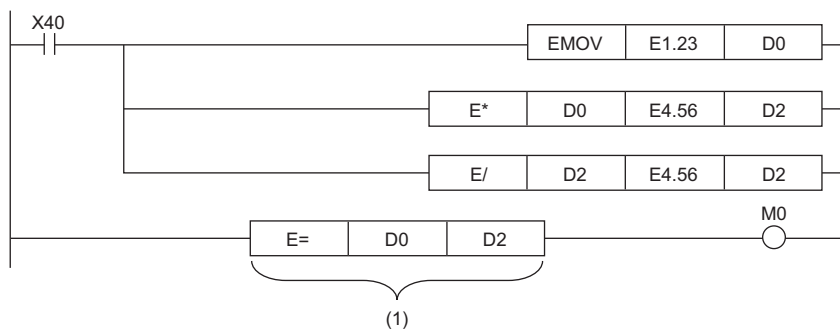
☞ 38页 注意事项

出错

没有运算错误。

要点

使用了E=指令的情况下,由于误差等可能导致不相等,应加以注意。



(1) 有时不相等。

双精度实数比较

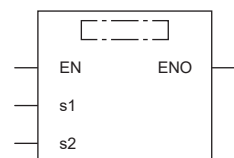
LDED□、ANDED□、ORED□

将双精度实数通过常开触点处理进行比较运算。

梯形图	ST
	ENO:=LDED_□(EN, s1, s2); ENO:=ANDED_□(EN, s1, s2); ENO:=ORED_□(EN, s1, s2); (□中为EQ、NE、GT、LE、LT、GE。)*1

(□中输入ED=、ED<>、ED>、ED<=、ED<、ED>=。)

FBD/LD



(□中放入LDED_、ANDED_、ORED_与EQ、NE、GT、LE、LT、GE的组合。)*1

*1 EQ为=、NE为<>、GT为>、LE为<=、LT为<、GE为>=。

■执行条件

指令	执行条件
LDED□、ANDED□、ORED□	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	比较数据或存储了比较数据的起始软元件	$0, 2^{-1022} \leq s1 < 2^{1024}$	双精度实数	ANYREAL_64
(s2)	比较数据或存储了比较数据的起始软元件	$0, 2^{-1022} \leq s2 < 2^{1024}$	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	○	—	○	—	○	—	—	—

功能

- 将(s1)中指定的软元件的双精度实数与(s2)中指定的软元件的双精度实数通过常开触点处理进行比较运算。
- 各指令的比较运算结果如下所示。

指令符号(梯形图、FBD/LD)	条件	比较运算结果
ED=、EQ	$(s1)=(s2)$	导通状态(ENO为ON)
ED<>、NE	$(s1)\neq(s2)$	
ED>、GT	$(s1)>(s2)$	
ED<=、LE	$(s1)\leq(s2)$	
ED<、LT	$(s1)<(s2)$	
ED>=、GE	$(s1)\geq(s2)$	
ED=、EQ	$(s1)\neq(s2)$	非导通状态(ENO为OFF)
ED<>、NE	$(s1)=(s2)$	
ED>、GT	$(s1)\leq(s2)$	
ED<=、LE	$(s1)>(s2)$	
ED<、LT	$(s1)\geq(s2)$	
ED>=、GE	$(s1)<(s2)$	

- (s1)、(s2)的内容超出设置数据的范围的情况下，变为非导通(ENO为OFF)。
- 通过FBD/LD使用了LDED_□指令的情况下，应将EN指定为左母线，或始终ON的变量部件/常数部件。
- 将ORED_□指令通过FBD/LD使用的情况下，将EN指定为TRUE时ENO将ON。EN不成为执行条件。
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

出错

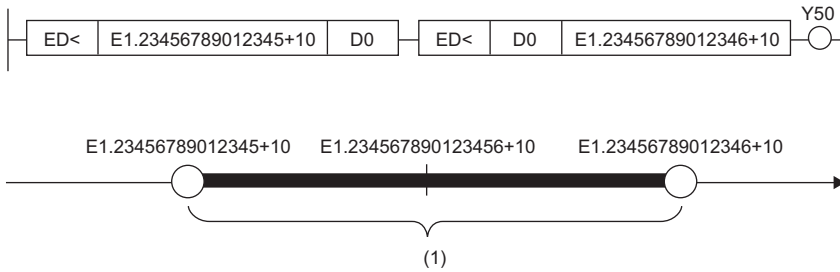
没有运算错误。

注意事项

通过工程工具可输入的实数的位数最大为15位，因此本项所示的指令中不能对有效位数16位及其以上的实数进行比较。对本项的指令判断有效位数16位及其以上的实数的一致、不一致的情况下，需要利用比较的实数前后的近似值大小来判断。

例

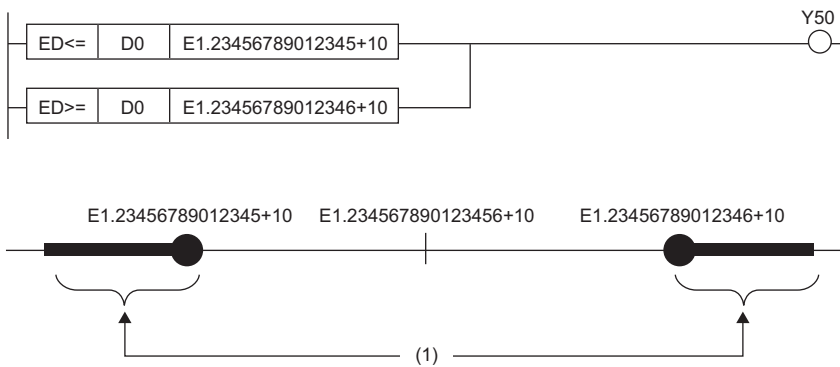
判断E1.23456789012345+10(有效位数16位)与双精度实数的一致情况下



(1) 检查D0~D3是否在该范围内。(边界值不包含在范围内。)

例

判断E1.23456789012345+10(有效位数16位)与双精度实数不一致的情况下



(1) 检查D0~D3是否在该范围内。(边界值包含在范围内。)

单精度实数加法运算

E+(P) [操作数为2个的情况下]

对单精度实数进行加法运算。

梯形图	ST
	不对应。 (☞ 795页 E+(P) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 795页 E+(P) [操作数为3个的情况下])

■执行条件

指令	执行条件
E+	
E+P	

设置数据

■内容、范围、数据类型

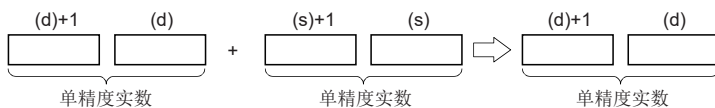
操作数	内容	范围	数据类型	数据类型(标签)
(s)	加法运算数据或存储了加法运算数据的起始软元件	$0, 2^{-126} \leq s < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储了加法运算数据的起始软元件	$0, 2^{-126} \leq d < 2^{128}$	单精度实数	ANYREAL_32

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	○	○	○	○	○	—	—	—	—

功能

- 对(d)中指定的单精度实数与(s)中指定的单精度实数进行加法运算，将加法运算结果存储到(d)中指定的软元件中。



- (s)、(d)中可以指定及存储 $0, 2^{-126} \leq |指定值(存储值)| < 2^{128}$ 的值。
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

出错

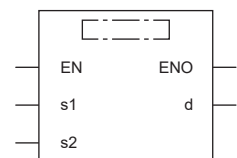
错误代码 (SD0)	内容
3282H	(s) 中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
	(d) 中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d) 超出下述范围时。(发生了上溢时。) $ d < 2^{128}$

E+(P) [操作数为3个的情况下]

对单精度实数进行加法运算。

梯形图	ST
	ENO:=EPLUS(EN, s1, s2, d); ENO:=EPLUSP(EN, s1, s2, d);

FBD/LD



(□中放入EPLUS、EPLUSP。)

■执行条件

指令	执行条件
E+	
E+P	

设置数据

■内容、范围、数据类型

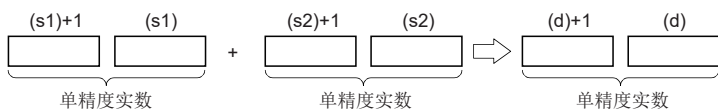
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	加法运算的数据或存储了加法运算数据的起始软元件	$0, 2^{-126} \leq (s1) < 2^{128}$	单精度实数	ANYREAL_32
(s2)	加法运算数据或存储了加法运算数据的起始软元件	$0, 2^{-126} \leq (s2) < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储运算结果的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—	—

功能

- 对(s1)中指定的单精度实数与(s2)中指定的单精度实数进行加法运算，将加法运算结果存储到(d)中指定的软元件中。



- (s1)、(s2)、(d)中可以指定及存储 $0, 2^{-126} \leq |指定值(存储值)| < 2^{128}$ 的值。

出错

错误代码 (SD0)	内容
3282H	(s1) 中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
	(s2) 中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d) 超出下述范围时。(发生了上溢时。) $ d < 2^{128}$

单精度实数减法运算

E-(P) [操作数为2个的情况下]

对单精度实数进行减法运算。

梯形图	ST
	不对应。 (☞ 799页 E-(P) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 799页 E-(P) [操作数为3个的情况下])

■执行条件

指令	执行条件
E-	
E-P	

设置数据

■内容、范围、数据类型

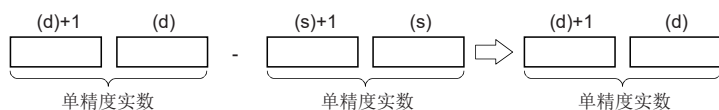
操作数	内容	范围	数据类型	数据类型(标签)
(s)	减数数据或存储了减数数据的起始软元件	$0, 2^{-126} \leq s < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储了被减数据的起始软元件	$0, 2^{-126} \leq d < 2^{128}$	单精度实数	ANYREAL_32

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	○	○	○	○	○	—	—	—	—

功能

- 对(d)中指定的单精度实数与(s)中指定的单精度实数进行减法运算，将减法运算结果存储到(d)中指定的软元件中。



- (s)、(d)中可以指定及存储 $0, 2^{-126} \leq |指定值(存储值)| < 2^{128}$ 的值。
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

出错

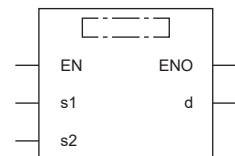
错误代码 (SD0)	内容
3282H	(s) 中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
	(d) 中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d) 超出下述范围时。(发生了上溢时。) $ d < 2^{128}$

E-(P) [操作数为3个的情况下]

对单精度实数进行减法运算。

梯形图	ST
	ENO:=EMINUS(EN, s1, s2, d); ENO:=EMINUSP(EN, s1, s2, d);

FBD/LD



(□中放入EMINUS、EMINUSP。)

■执行条件

指令	执行条件
E-	
E-P	

设置数据

■内容、范围、数据类型

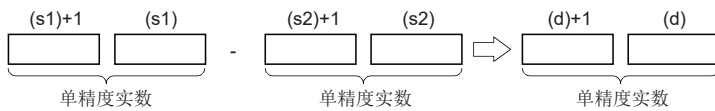
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	被减数据或存储了被减数据的起始软元件	$0, 2^{-126} \leq (s1) < 2^{128}$	单精度实数	ANYREAL_32
(s2)	减数数据或存储了减数数据的起始软元件	$0, 2^{-126} \leq (s2) < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储运算结果的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—

功能

- 对(s1)中指定的单精度实数与(s2)中指定的单精度实数进行减法运算，将减法运算结果存储到(d)中指定的软元件中。



- (s1)、(s2)、(d)中可以指定及存储 $0, 2^{-126} \leq |\text{指定值(存储值)}| < 2^{128}$ 的值。
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s1)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
	(s2)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d)超出下述范围时。(发生了上溢时。) $ (d) < 2^{128}$

双精度实数加法运算

ED+(P) [操作数为2个的情况下]

对双精度实数进行加法运算。

梯形图	ST
	不对应。 (☞ 803页 ED+(P) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 803页 ED+(P) [操作数为3个的情况下])

■执行条件

指令	执行条件
ED+	
ED+P	

设置数据

■内容、范围、数据类型

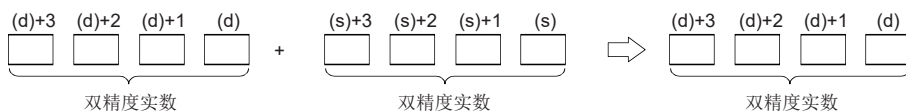
操作数	内容	范围	数据类型	数据类型(标签)
(s)	加法运算数据或存储了加法运算数据的起始软元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储了加法运算数据的起始软元件	$0, 2^{-1022} \leq (d) < 2^{1024}$	双精度实数	ANYREAL_64

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—

功能

- 对(d)中指定的双精度实数与(s)中指定的双精度实数进行加法运算，将加法运算结果存储到(d)中指定的软元件中。



- (s)、(d)中可以指定及存储 $0, 2^{-1022} \leq |指定值(存储值)| < 2^{1024}$ 的值。
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

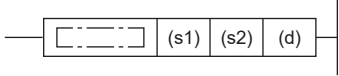
☞ 38页 注意事项

出错

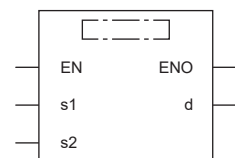
错误代码 (SD0)	内容
3282H	(s) 中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
	(d) 中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d) 超出下述范围时。(发生了上溢时。) $ d < 2^{1024}$

ED+(P) [操作数为3个的情况下]

对双精度实数进行加法运算。


梯形图	ST
	ENO:=EDPLUS(EN, s1, s2, d); ENO:=EDPLUSP(EN, s1, s2, d);

FBD/LD



(□中放入EDPLUS、EDPLUSP。)

■执行条件

指令	执行条件
ED+	
ED+P	

设置数据

■内容、范围、数据类型

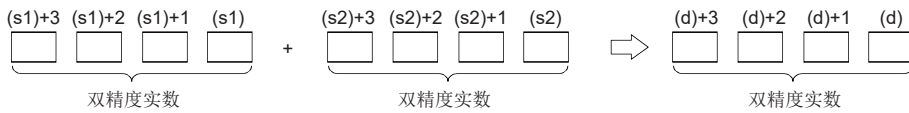
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	加法运算的数据或存储了加法运算数据的起始软元件	$0, 2^{-1022} \leq (s1) < 2^{1024}$	双精度实数	ANYREAL_64
(s2)	加法运算数据或存储了加法运算数据的起始软元件	$0, 2^{-1022} \leq (s2) < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储运算结果的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 对(s1)中指定的双精度实数与(s2)中指定的双精度实数进行加法运算，将加法运算结果存储到(d)中指定的软元件中。



- (s1)、(s2)、(d)中可以指定及存储 $0, 2^{-1022} \leq | \text{指定值(存储值)} | < 2^{1024}$ 的值。
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

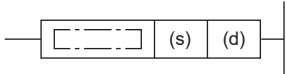
出错

错误代码(SD0)	内容
3282H	(s1)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
	(s2)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d)超出下述范围时。(发生了上溢时。) $ (d) < 2^{1024}$

双精度实数减法运算



ED-(P) [操作数为2个的情况下]

对双精度实数进行减法运算。

梯形图	ST
	不对应。 (☞ 807页 ED-(P) [操作数为3个的情况下])

FBD/LD
不对应。 (☞ 807页 ED-(P) [操作数为3个的情况下])

■执行条件

指令	执行条件
ED-	
ED-P	

设置数据

■内容、范围、数据类型

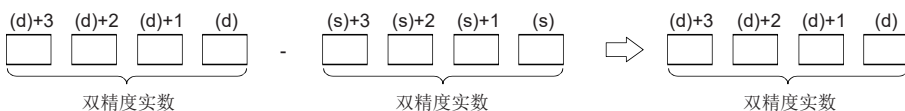
操作数	内容	范围	数据类型	数据类型(标签)
(s)	减数数据或存储了减数数据的起始软元件	$0, 2^{-1022} \leq s < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储了被减数据的起始软元件	$0, 2^{-1022} \leq d < 2^{1024}$	双精度实数	ANYREAL_64

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—

功能

- 对(d)中指定的双精度实数与(s)中指定的双精度实数进行减法运算，将减法运算结果存储到(d)中指定的软元件中。



- (s)、(d)中可以指定及存储 $0, 2^{-1022} \leq |指定值(存储值)| < 2^{1024}$ 的值。
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

出错

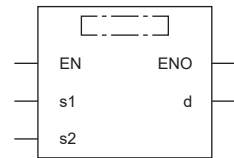
错误代码 (SD0)	内容
3282H	(s) 中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
	(d) 中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d) 超出下述范围时。(发生了上溢时。) $ d < 2^{1024}$

ED-(P) [操作数为3个的情况下]

对双精度实数进行减法运算。

梯形图	ST
	ENO:=EDMINUS(EN, s1, s2, d); ENO:=EDMINUSP(EN, s1, s2, d);

FBD/LD



(□中放入EDMINUS、EDMINUSP。)

■执行条件

指令	执行条件
ED-	
ED-P	

设置数据

■内容、范围、数据类型

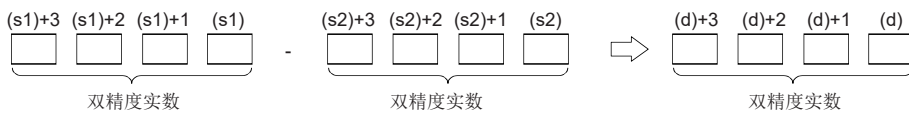
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	被减数据或存储了被减数据的起始软元件	$0, 2^{-1022} \leq (s1) < 2^{1024}$	双精度实数	ANYREAL_64
(s2)	减数数据或存储了减数数据的起始软元件	$0, 2^{-1022} \leq (s2) < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储运算结果的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 对(s1)中指定的双精度实数与(s2)中指定的双精度实数进行减法运算，将减法运算结果存储到(d)中指定的软元件中。



- (s1)、(s2)、(d)中可以指定及存储 $0, 2^{-1022} \leq | \text{指定值(存储值)} | < 2^{1024}$ 的值。
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

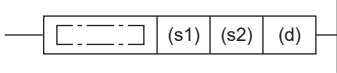
出错

错误代码(SD0)	内容
3282H	(s1)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
	(s2)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d)超出下述范围时。(发生了上溢时。) $ (d) < 2^{1024}$

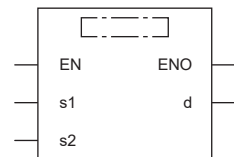
单精度实数乘法运算

E*(P)

对单精度实数进行乘法运算。


梯形图	ST
	ENO:=EMULTI(EN, s1, s2, d); ENO:=EMULTIP(EN, s1, s2, d);

FBD/LD



(□中放入EMULTI、EMULTIP。)

■执行条件

指令	执行条件
E*	
E*P	

设置数据

■内容、范围、数据类型

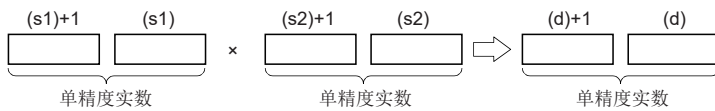
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	乘法运算数据或存储了乘法运算数据的起始软元件	$0, 2^{-126} \leq s1 < 2^{128}$	单精度实数	ANYREAL_32
(s2)	乘法运算数据或存储了乘法运算数据的起始软元件	$0, 2^{-126} \leq s2 < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储运算结果的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—

功能

- 对(s1)中指定的单精度实数与(s2)中指定的单精度实数进行乘法运算，将乘法运算结果存储到(d)中指定的软元件中。



- (s1)、(s2)、(d)中可以指定及存储 $0, 2^{-126} \leq |\text{指定值(存储值)}| < 2^{128}$ 的值。
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s1)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
	(s2)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d)超出下述范围时。(发生了上溢时。) $ (d) < 2^{128}$

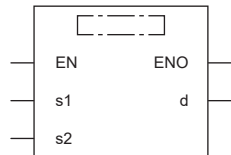
单精度实数除法运算

E/(P)

对单精度实数进行除法运算。

梯形图	ST
	ENO:=EDIVISION(EN, s1, s2, d); ENO:=EDIVISIONP(EN, s1, s2, d);

FBD/LD



(□中放入EDIVISION、EDIVISIONP。)

■执行条件

指令	执行条件
E/	
E/P	

设置数据

■内容、范围、数据类型

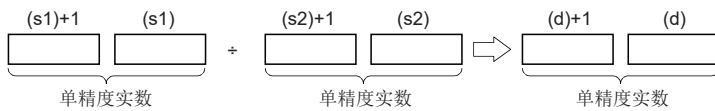
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	被除数据或存储了被除数据的起始软元件	$0, 2^{-126} \leq s1 < 2^{128}$	单精度实数	ANYREAL_32
(s2)	除数数据或存储了除数数据的起始软元件	$0, 2^{-126} \leq s2 < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储运算结果的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—

功能

- 对(s1)中指定的单精度实数与(s2)中指定的单精度实数进行除法运算，将除法运算结果存储到(d)中指定的软元件中。



- (s1)、(s2)、(d)中可以指定及存储 $0 < 2^{-126} \leq |指定值(存储值)| < 2^{128}$ 的值。
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

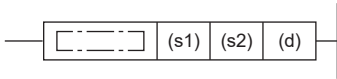
出错

错误代码(SD0)	内容
3280H	(s2)中指定的数据(除数)为0时。
3282H	(s1)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
	(s2)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d)超出下述范围时。(发生了上溢时。) $ (d) < 2^{128}$

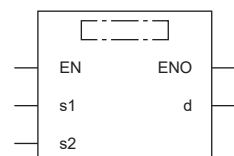
双精度实数乘法运算

ED*(P)

对双精度实数进行乘法运算。

梯形图	ST
	ENO:=EDMULTI (EN, s1, s2, d) ; ENO:=EDMULTIP (EN, s1, s2, d) ;

FBD/LD



(□中放入EDMULTI、EDMULTIP。)

■执行条件

指令	执行条件
ED*	
ED*P	

设置数据

■内容、范围、数据类型

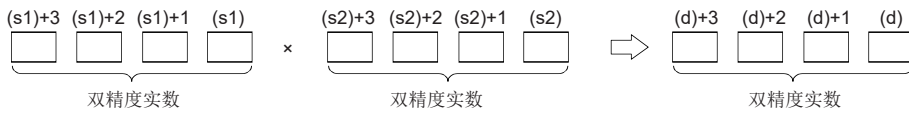
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	乘法运算数据或存储了乘法运算数据的起始软元件	$0, 2^{-1022} \leq s1 < 2^{1024}$	双精度实数	ANYREAL_64
(s2)	乘法运算数据或存储了乘法运算数据的起始软元件	$0, 2^{-1022} \leq s2 < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储运算结果的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 对(s1)中指定的双精度实数与(s2)中指定的双精度实数进行乘法运算，将乘法运算结果存储到(d)中指定的软元件中。



- (s1)、(s2)、(d)中可以指定及存储 $0, 2^{-1022} \leq | \text{指定值(存储值)} | < 2^{1024}$ 的值。
- 运算结果为-0或发生了下溢时，运算结果将变为0。
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

出错

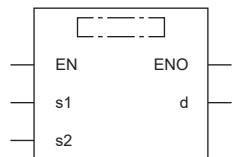
错误代码(SD0)	内容
3282H	(s1)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。 (s2)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d)超出下述范围时。(发生了上溢时。) $ (d) < 2^{1024}$

ED/(P)

对双精度实数进行除法运算。

梯形图	ST
	<pre>ENO:=EDDIVISION(EN, s1, s2, d); ENO:=EDDIVISIONP(EN, s1, s2, d);</pre>

FBD/LD



(□中放入EDDIVISION、EDDIVISIONP。)

■执行条件

指令	执行条件
ED/	
ED/P	

设置数据

■内容、范围、数据类型

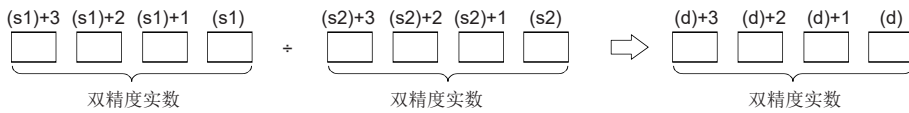
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	被除数据或存储了被除数据的起始软元件	$0, 2^{-1022} \leq s1 < 2^{1024}$	双精度实数	ANYREAL_64
(s2)	除数数据或存储了除数数据的起始软元件	$0, 2^{-1022} \leq s2 < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储运算结果的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 对(s1)中指定的双精度实数与(s2)中指定的双精度实数进行除法运算，将除法运算结果存储到(d)中指定的软元件中。



- (s1)、(s2)、(d)中可以指定及存储 0 、 $2^{-1022} \leq | \text{指定值(存储值)} | < 2^{1024}$ 的值。
- 运算结果为 -0 或发生了下溢时，运算结果将变为 0 。
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3280H	(s2)中指定的数据(除数)为 0 时。
3282H	(s1)中指定的数据为 -0 、非正规化数、非数、 $\pm\infty$ 时。 (s2)中指定的数据为 -0 、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d)超出下述范围时。(发生了上溢时。) $ (d) < 2^{1024}$

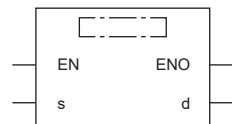
带符号BIN16位数据→单精度实数转换

INT2FLT(P)

将带符号BIN16位数据转换为单精度实数。

梯形图	ST
	ENO:=INT2FLT(EN, s, d); ENO:=INT2FLTP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
INT2FLT	
INT2FLTP	

设置数据

■内容、范围、数据类型

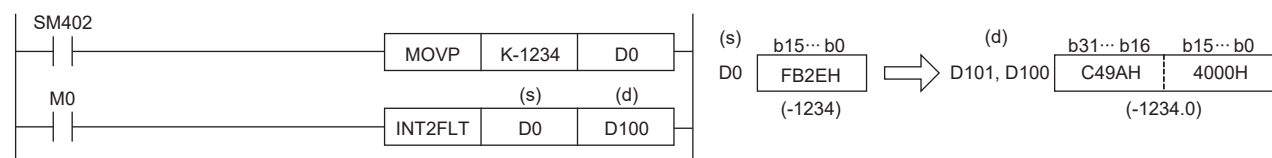
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行单精度实数转换的整数数据或存储了整数数据的软元件	-32768~32767	带符号BIN16位	ANY16_S
(d)	存储转换后单精度实数的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—	—

功能

将(s)中指定的带符号BIN16位数据转换为单精度实数后, 存储到(d)中。



出错

没有运算错误。

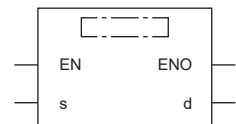
无符号BIN16位数据→单精度实数转换

UINT2FLT (P)

将无符号BIN16位数据转换为单精度实数。

梯形图	ST
	ENO:=UINT2FLT (EN, s, d) ; ENO:=UINT2FLTP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
UINT2FLT	
UINT2FLTP	

设置数据

■内容、范围、数据类型

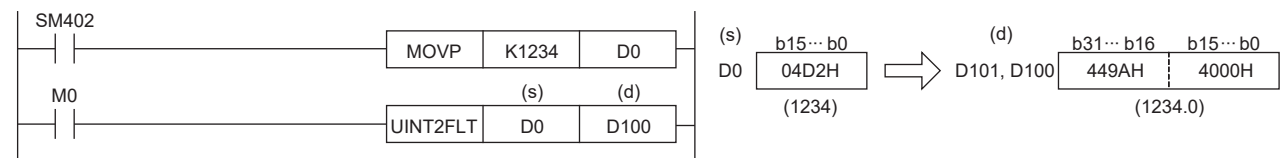
操作数	内容	范围	数据类型	数据类型 (标签)
(s)	进行单精度实数转换的整数数据或存储了整数数据的软元件	0~65535	无符号BIN16位	ANY16_U
(d)	存储转换后单精度实数的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—	—

功能

将(s)中指定的无符号BIN16位数据转换为单精度实数后，存储到(d)中。



出错

没有运算错误。

带符号BIN32位数据→单精度实数转换

DINT2FLT (P)

将带符号BIN32位数据转换为单精度实数。

梯形图	ST
	ENO:=DINT2FLT (EN, s, d) ; ENO:=DINT2FLTP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
DINT2FLT	
DINT2FLTP	

设置数据

■内容、范围、数据类型

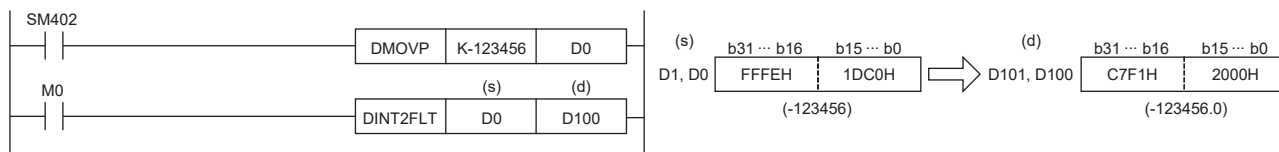
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行单精度实数转换的整数数据或存储了整数数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
(d)	存储转换后单精度实数的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

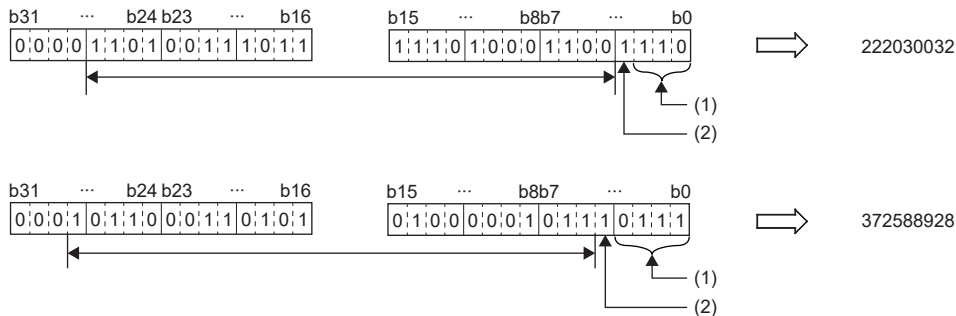
操作数	位		字		双字			间接指定	常数				其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K、H	E	\$		
(s)	○	○	○	○	○	○	○	○	○	—	—	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—	—

功能

- 将(s)中指定的带符号BIN32位数据转换为单精度实数后，存储到(d)中。



- 单精度实数是以32位的单精度进行处理，因此以2进制数表示时有效位数为24位，以10进制数表示时约为7位。因此，整数超过了 $-16777216 \sim 16777215$ (24位BIN值) 的范围的情况下，转换后的值将产生误差。转换结果为，对从整数的高位开始的第25位进行舍0进1，并舍去第26位及其以后的值。



- (1) 舍去。
(2) 舍去0进位1。

出错

没有运算错误。

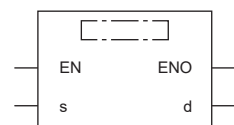
无符号BIN32位数据→单精度实数转换

UDINT2FLT (P)

将无符号BIN32位数据转换为单精度实数。

梯形图	ST
	ENO:=UDINT2FLT (EN, s, d); ENO:=UDINT2FLTP (EN, s, d);

FBD/LD



■执行条件

指令	执行条件
UDINT2FLT	
UDINT2FLTP	

设置数据

■内容、范围、数据类型

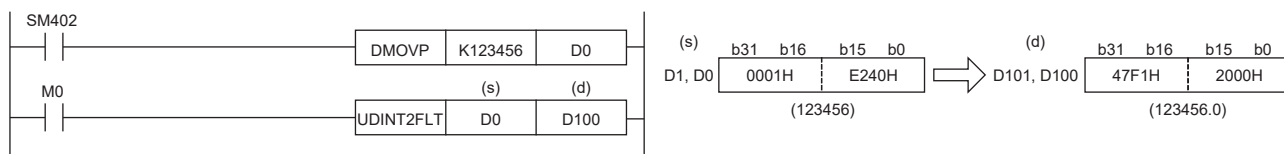
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行单精度实数转换的整数数据或存储了整数数据的起始软元件	0~4294967295	无符号BIN32位	ANY32_U
(d)	存储转换后单精度实数的起始软元件	—	单精度实数	ANYREAL_32

■可用软元件

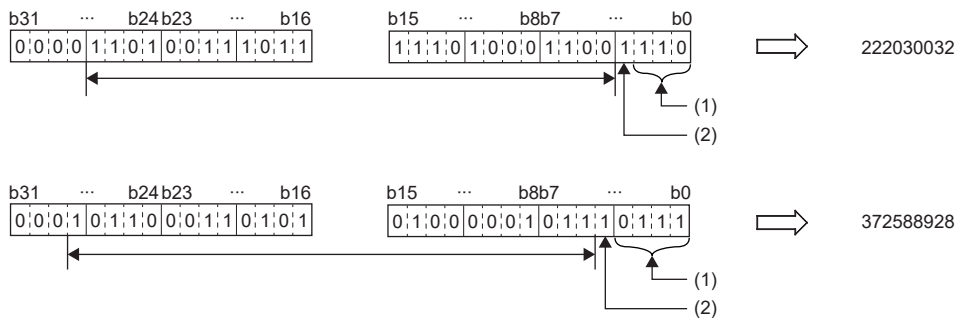
操作数	位		字		双字		间接指定	常数			其它	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K	H		E
(s)	○	○	○	○	○	○	○	○	—	—	—	—
(d)	—	—	○	○	○	○	○	○	—	—	—	—

功能

- 将(s)中指定的无符号BIN32位数据转换为单精度实数后，存储到(d)中。



- 单精度实数是以32位的单精度进行处理，因此以2进制数表示时有效位数为24位，以10进制数表示时约为7位。因此，整数超过了0~16777215(24位BIN值)的范围的情况下，转换后的值将产生误差。转换结果为，对从整数的高位开始的第25位进行舍0进1，并舍去第26位及其以后的值。



- (1) 舍去。
(2) 舍去0进位1。

出错

没有运算错误。

双精度实数→单精度实数转换

DBL2FLT(P)

将双精度实数转换为单精度实数。

梯形图	ST
	ENO:=DBL2FLT(EN, s, d); ENO:=DBL2FLTP(EN, s, d);

FBD/LD

■执行条件

指令	执行条件
DBL2FLT	
DBL2FLTP	

设置数据

■内容、范围、数据类型

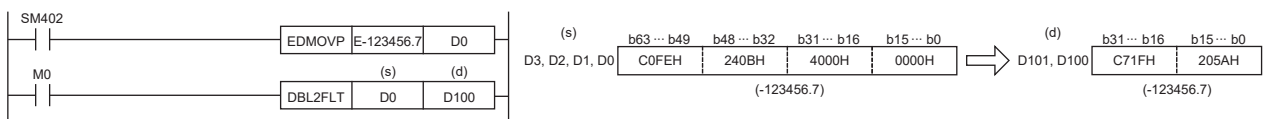
操作数	内容	范围	数据类型	数据类型(标签)
(s)	转换为单精度实数的双精度实数数据或存储了双精度实数数据的起始软元件	$0, 2^{-1022} \leq s < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储转换后单精度实数的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	—	○	○	○	○	—	—	—	—

功能

- 将(s)中指定的双精度实数转换为单精度实数后, 存储到(d)中。



- 通过工程工具设置输入值的情况下, 有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项, 请参阅下述内容。

☞ 38页 注意事项

出错

错误代码 (SD0)	内容
3282H	(s) 中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d) 超出下述范围时。(发生了上溢时。) $ d < 2^{128}$

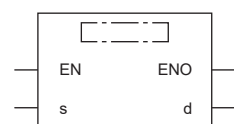
带符号BIN16位数据→双精度实数转换

INT2DBL (P)

将带符号BIN16位数据转换为双精度实数。

梯形图	ST
	ENO:=INT2DBL (EN, s, d) ; ENO:=INT2DBLP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
INT2DBL	
INT2DBLP	

设置数据

■内容、范围、数据类型

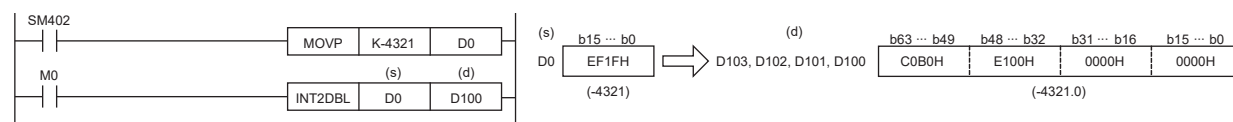
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行双精度实数转换的整数数据或存储了整数数据的软元件	-32768~32767	带符号BIN16位	ANY16_S
(d)	存储转换后双精度实数的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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	
(s)	○	—	○	—	○	—	—	○	—	—	—
(d)	—	—	○	—	—	○	—	○	—	—	—

功能

将(s)中指定的带符号BIN16位数据转换为双精度实数后, 存储到(d)中。



出错

没有运算错误。

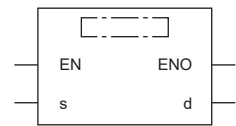
无符号BIN16位数据→双精度实数转换

UINT2DBL (P)

将无符号BIN16位数据转换为双精度实数。

梯形图	ST
	ENO:=UINT2DBL(EN, s, d); ENO:=UINT2DBLP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
UINT2DBL	
UINT2DBLP	

设置数据

■内容、范围、数据类型

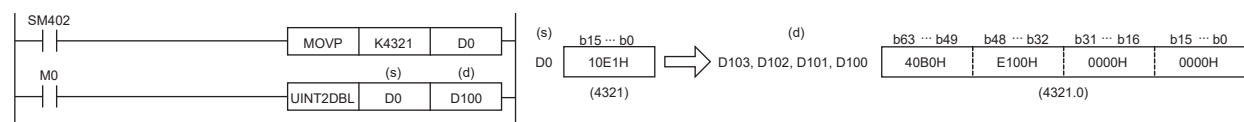
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行双精度实数转换的整数数据或存储了整数数据的软元件	0~65535	无符号BIN16位	ANY16_U
(d)	存储转换后双精度实数的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—	—

功能

将(s)中指定的无符号BIN16位数据转换为双精度实数后，存储到(d)中。



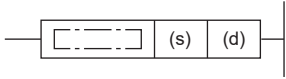
出错

没有运算错误。

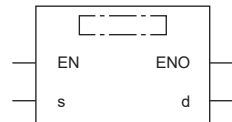
带符号BIN32位数据→双精度实数转换

DINT2DBL (P)

将带符号BIN32位数据转换为双精度实数。

梯形图	ST
	ENO:=DINT2DBL(EN, s, d); ENO:=DINT2DBLP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
DINT2DBL	
DINT2DBLP	

设置数据

■内容、范围、数据类型

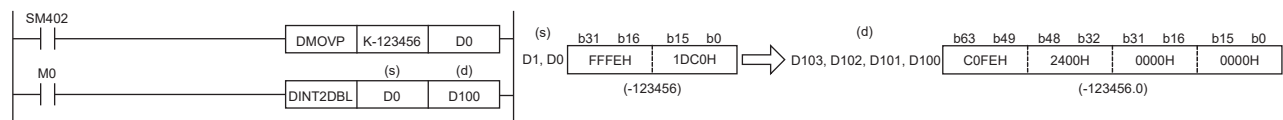
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行双精度实数转换的整数数据或存储了整数数据的起始软元件	-2147483648~2147483647	带符号BIN32位	ANY32_S
(d)	存储转换后双精度实数的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—

功能

将(s)中指定的带符号BIN32位数据转换为双精度实数后, 存储到(d)中。



出错

没有运算错误。

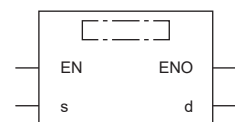
无符号BIN32位数据→双精度实数转换

UDINT2DBL (P)

将无符号BIN32位数据转换为双精度实数。

梯形图	ST
	ENO:=UDINT2DBL(EN, s, d); ENO:=UDINT2DBLP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
UDINT2DBL	
UDINT2DBLP	

设置数据

■内容、范围、数据类型

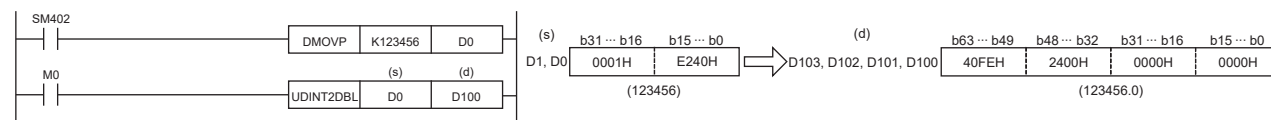
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行双精度实数转换的整数数据或存储了整数数据的起始软元件	0~4294967295	无符号BIN32位	ANY32_U
(d)	存储转换后双精度实数的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—	—

功能

将(s)中指定的无符号BIN32位数据转换为双精度实数后，存储到(d)中。



出错

没有运算错误。

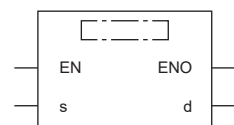
单精度实数→双精度实数转换

FLT2DBL (P)

将单精度实数转换为双精度实数。

梯形图	ST
	ENO:=FLT2DBL(EN, s, d); ENO:=FLT2DBLP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
FLT2DBL	
FLT2DBLP	

设置数据

■内容、范围、数据类型

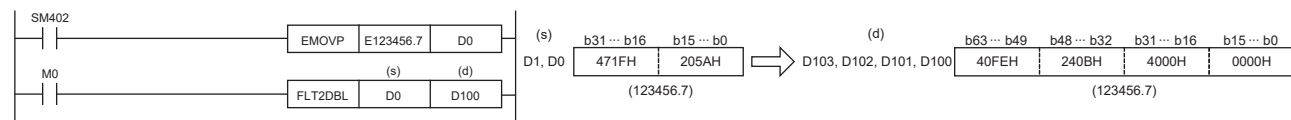
操作数	内容	范围	数据类型	数据类型(标签)
(s)	转换为双精度实数的单精度实数数据或存储了单精度实数数据的起始软元件	$0, 2^{-126} \leq (s) < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储转换后双精度实数的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—	—

功能

将(s)中指定的单精度实数转换为双精度实数后, 存储到(d)中。



出错

错误代码(SD0)	内容
3282H	(s)中指定的数据为-0、非规范化数、非数、±∞时。

字符串→单精度实数转换

EVAL (P)

将字符串转换为单精度实数。

<p>梯形图</p>	<p>ST</p> <pre>ENO:=EVAL (EN, s, d) ; ENO:=EVALP (EN, s, d) ;</pre>
-------------------	--

FBD/LD

■执行条件

指令	执行条件
EVAL	
EVALP	

设置数据

■内容、范围、数据类型

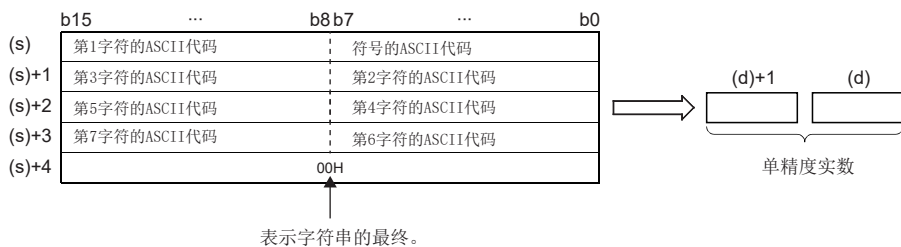
操作数	内容	范围	数据类型	数据类型 (标签)
(s)	进行单精度实数数据转换的字符串数据或存储了字符串数据的起始软元件	—	字符串	ANYSTRING_SINGLE
(d)	存储转换后单精度实数数据的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—	

功能

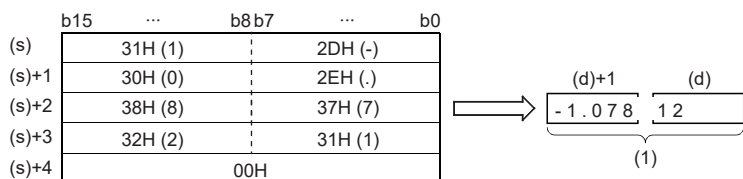
- 将(s)中指定的软件编号及其以后存储的字符串转换为单精度实数后, 存储到(d)中指定的软件中。
- 无论指定的字符串为小数点形式还是指数形式均可转换为单精度实数。



- 字符串中包含有20H(空白)的情况下, 将忽略20H进行转换。
- 字符串最多可设置24字符。字符串中的20H(空白)、30H(0)也作为1字符计数。

■ 小数点形式的情况下

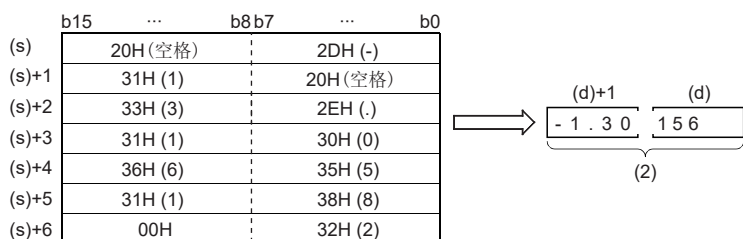
- (s)中指定的字符串为小数点形式时的情况如下所示。



[1][0][7][8][1][2]

(1) 单精度实数

- (s)中指定的字符串中, 将要转换为单精度实数的字符串除去符号、小数点、指数部, 6位为有效, 第7位及其以后被舍去后进行转换。



[][][][3][0][1][5][6][8][1][2]

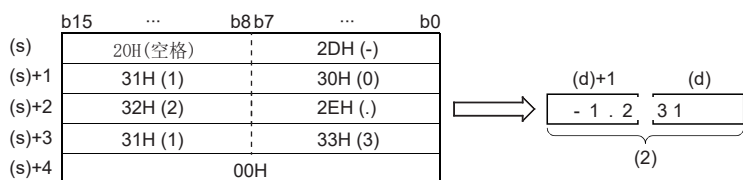
(1)

(1) 舍去。

(2) 单精度实数

- 在小数点形式中将符号指定为2BH(+)或省略符号时, 将作为正值进行转换。此外, 将符号指定为2DH(-)时, 将作为负值进行转换。

- (s)中指定的字符串中, 最初的0以外的数值之间存在有20H(空白)或30H(0)的情况下, 转换时将忽略20H、30H。



[][][0][1][2][3]

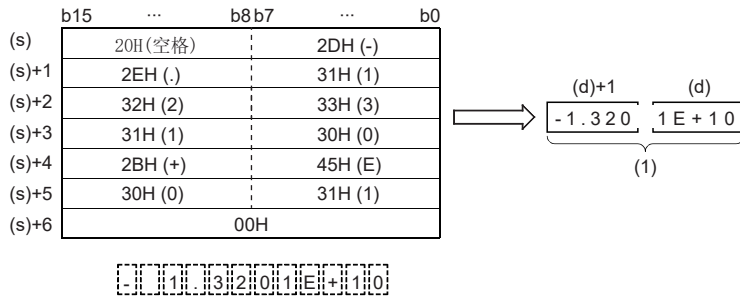
(1)

(1) 忽略。

(2) 单精度实数

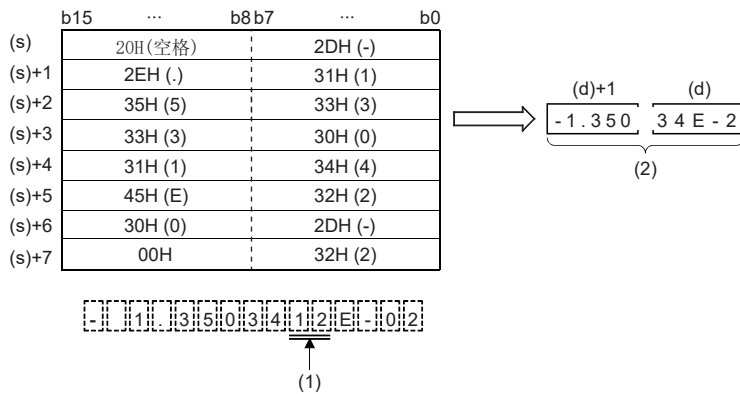
■指数形式的情况下

- (s) 中指定的字符串为指数形式的情况下，按下述方式执行。



(1) 单精度实数

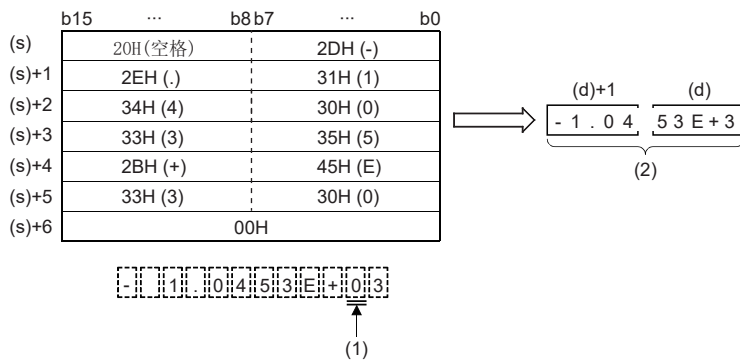
- (s) 中指定的字符串中，将要转换为单精度实数的字符串除去符号、小数点、指数部，6位为有效，第7位及其以后被舍去后进行转换。



(1) 舍去。

(2) 单精度实数

- 在指数形式中将指数部的符号指定为2BH(+)或省略符号时，将作为正值进行转换。将指数部的符号指定为2DH(-)时，将作为负值进行转换。
- (s) 中指定的字符串中，最初的0以外的数值之间存在有20H(空白)或30H(0)的情况下，转换时将忽略20H、30H。
- 在指数形式的字符串中，“E”与数值之间存储有30H(0)的情况下，转换时将忽略30H。



(1) 忽略。

(2) 单精度实数

- 字符串中包含有“20H”(空白)的情况下，转换时将忽略“20H”。
- 字符串最多可设置24字符。字符串中的“20H”(空白)、“30H”(0)也作为1个字符计数。

错误代码 (SD0)	内容
2820H	从 (s) 开始的相应软元件范围内没有 00H 时。
3281H	(s) 中设置了不能转换的非法数据时。 <ul style="list-style-type: none"> • 整数部、小数部中存在有 30H (0) ~ 39H (9) 以外的字符 • 指定的字符串中有 2 个或其以上的 2EH (.) • 指定的字符串的指数部中存在有 45H (E)、65H (e)、2BH (+)、2DH (-) 以外字符 • 指定的字符串中存在有多个 45H (E)、65H (e) 指数部 • 指定的字符串中，记载了 3 位或其以上的指数部数值 • 指定的字符串中存在有多个 2BH (+)、2DH (-) 指数部 • 指定的字符串中，在小数点形式的情况下正数部有多个 2BH (+)、2DH (-) 的符号，在指数形式的情况下尾数部中有多个 2BH (+)、2DH (-) 的符号
3283H	(d) 超出下述范围时。(发生了上溢时。) $ (d) < 2^{128}$
3285H	(s) 及其以后的字符数为 0 或超过 24 个字符时。

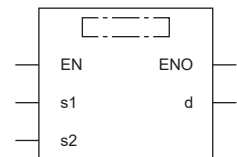
BCD格式数据→单精度实数数据转换

EREXP (P)

将BCD型浮点格式数据根据指定的小数部位数转换为单精度实数数据。

梯形图	ST
	ENO:=EREXP (EN, s1, s2, d) ; ENO:=EREXPP (EN, s1, s2, d) ;

FBD/LD



■执行条件

指令	执行条件
EREXP	
EREXPP	

设置数据

■内容、范围、数据类型

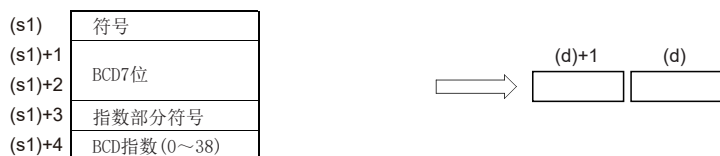
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储了BCD型浮点格式数据的起始软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 5)
(s2)	小数部位数数据	0~7	带符号BIN16位	ANY16
(d)	存储单精度实数的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	—	—	○	○	○	○	○	—	—	—	—

功能

- 将(s1)中指定的软元件编号及其以后存储的BCD型浮点格式数据，根据(s2)中指定的软元件中存储的小数位数转换为单精度实数数据后，存储到(d)中指定的软元件编号及其以后。



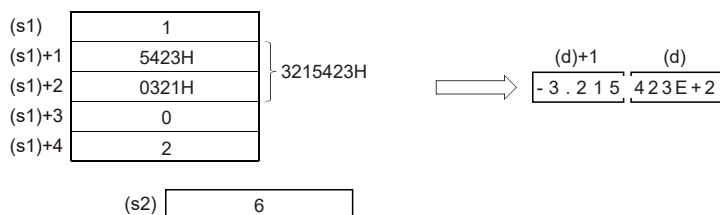
(s2) 小数部分位数(0~7)

(s1): 符号(正时0, 负时1)
 (s1)+1、(s1)+2: BCD7位
 (s1)+3: 指数部符号(正时0, 负时1)
 (s1)+4: BCD指数(0~38)
 (s2): 小数部位数(0~7)
 (d)+1、(d): 单精度实数

- (s1)的符号及(s1)+3的指数部符号中，为正时设置0，为负时设置1。
- (s1)+4的BCD指数中可以设置0~38。
- (s2)的小数部位数中可以设置0~7。

例

在(s2)中设置了6的情况下



出错

错误代码(SD0)	内容
3281H	(s1)中指定的形式指定为0、1以外时。
	(s1)+1、(s1)+2的各位中存在有0~9以外的值时。
	(s1)+3中指定的指数部符号为0、1以外时。
	(s1)+4中指定的指数数据为0~38以外时。
	(s2)中指定的小数部位数为0~7以外时。
3283H	(d)超出下述范围时。(发生了上溢时。)
	$ d < 2^{128}$

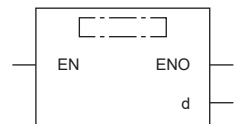
单精度实数符号取反

ENEG (P)

对单精度实数数据的符号进行取反。

梯形图	ST
	ENO:=ENEG (EN, d) ; ENO:=ENEGP (EN, d) ;

FBD/LD



■执行条件

指令	执行条件
ENEG	
ENEGP	

设置数据

■内容、范围、数据类型

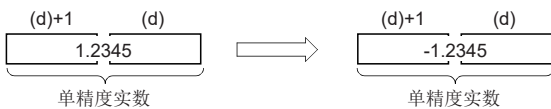
操作数	内容	范围	数据类型	数据类型 (标签)
(d)	存储了进行符号取反的单精度实数数据的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	—	—	○	○	○	○	○	—	—	—	—

功能

- 对 (d) 中指定的软元件的单精度实数的符号进行取反后，存储到 (d) 中指定的软元件中。



- 在对正负符号进行取反时使用。

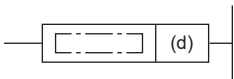
出错

错误代码 (SDO)	内容
3282H	(d) 中指定的内容为 -0、非正规化数、非数、±∞ 时。

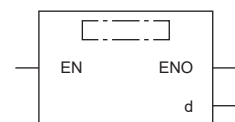
双精度实数符号取反

EDNEG(P)


对双精度实数数据的符号进行取反。

梯形图	ST
	ENO:=EDNEG(EN, d); ENO:=EDNEGP(EN, d);

FBD/LD



■执行条件

指令	执行条件
EDNEG	
EDNEGP	

设置数据

■内容、范围、数据类型

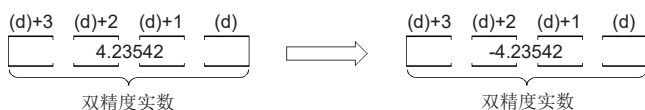
操作数	内容	范围	数据类型	数据类型(标签)
(d)	存储了进行符号取反的双精度实数数据的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—

功能

- 对(d)中指定的软元件的双精度实数数据的符号进行取反后, 存储到(d)中指定的软元件中。



- 在对正负符号进行取反时使用。

出错

错误代码(SD0)	内容
3282H	(d)中指定的内容为-0、非正规化数、非数、±∞时。

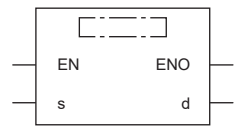
单精度实数数据传送

EMOV (P)

将单精度实数数据传送至指定的软元件。

梯形图	ST
	ENO:=EMOV (EN, s, d) ; ENO:=EMOVP (EN, s, d)

FBD/LD



■执行条件

指令	执行条件
EMOV	
EMOVP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s)	传送的数据或存储了传送数据的起始软元件	$0, 2^{-126} \leq (s) < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储传送目标数据的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	○	○	○	○	○	—	—	—	—

功能

将(s)中指定的软元件中存储的单精度实数数据，传送至(d)中指定的软元件。



出错

没有运算错误。

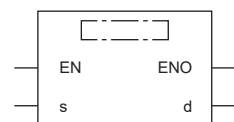
双精度实数数据传送

EDMOV (P)

将双精度实数数据传送至指定的软元件。

梯形图	ST
	ENO:=EDMOV (EN, s, d) ; ENO:=EDMOV (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
EDMOV	
EDMOV (P)	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s)	传送的数据或存储了传送数据的起始软元件	$0, 2^{-1022} \leq s < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储传送目标数据的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	—	—	○	—	○	—	—	—	—

功能

将(s)中指定的软元件中存储的双精度实数数据，传送至(d)中指定的软元件。



出错

没有运算错误。

单精度实数SIN运算

SIN(P)

对以单精度实数指定角度的SIN(正弦)值进行运算。

梯形图	ST*1
	ENO:=SINP(EN, s, d);

FBD/LD*1

*1 在ST、FBD/LD中不支持SIN指令。应使用通用函数的SIN。
 1271页 SIN(E)

■执行条件

指令	执行条件
SIN	
SINP	

设置数据

■内容、范围、数据类型

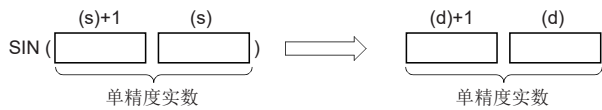
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行SIN(正弦)运算的角度数据或存储了角度数据的起始软元件	$0, 2^{-126} \leq (s) < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储运算结果的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—	—

功能

- 对 (s) 中指定角度的 SIN(正弦) 值进行运算后, 将运算结果存储到 (d) 中指定的软元件编号中。



- (s) 中指定的角度, 以弧度单位 (角度 $\times \pi \div 180$) 进行设置。
- 运算结果为 -0 或发生了下溢时, 运算结果将变为 0。
- 通过工程工具设置输入值的情况下, 有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项, 请参阅下述内容。

38页 注意事项

出错

错误代码 (SD0)	内容
3282H	(s) 中指定的值为 -0、非正规化数、非数、 $\pm\infty$ 时。

要点

关于角度 \leftrightarrow 弧度转换的有关内容, 请参阅 RAD(P) 指令、DEG(P) 指令。

876页 RAD(P)

878页 DEG(P)

单精度实数COS运算

COS(P)

对单精度实数中指定角度的COS(余弦)值进行运算。

梯形图	ST*1
	$ENO := \text{COSP}(EN, s, d);$

FBD/LD*1

*1 在ST、FBD/LD中不支持COS指令。应使用通用函数的COS。
 1272页 COS(_E)

■执行条件

指令	执行条件
COS	
COSP	

设置数据

■内容、范围、数据类型

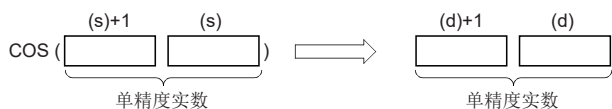
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行COS(余弦)运算的角度数据或存储了角度数据的起始软元件	$0, 2^{-126} \leq (s) < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储运算结果的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—	—

功能

- 对(s)中指定角度的COS(余弦)值进行运算后,将运算结果存储到(d)中指定的软元件编号中。



- (s)中指定的角度,以弧度单位(角度 $\times\pi\div 180$)进行设置。
- 运算结果为-0或发生了下溢时,运算结果将变为0。
- 通过工程工具设置输入值的情况下,有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项,请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、 $\pm\infty$ 时。

要点

关于角度 \leftrightarrow 弧度转换的有关内容,请参阅RAD(P)指令、DEG(P)指令。

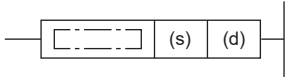
☞ 876页 RAD(P)

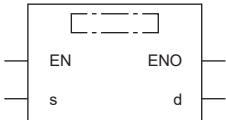
☞ 878页 DEG(P)

单精度实数TAN运算

TAN(P)

对单精度实数中指定角度的TAN(正切)值进行运算。

梯形图	ST*1
	ENO:=TANP(EN, s, d);

FBD/LD*1


*1 在ST、FBD/LD中不支持TAN指令。应使用通用函数的TAN。
 1273页 TAN(_E)

■执行条件

指令	执行条件
TAN	
TANP	

设置数据

■内容、范围、数据类型

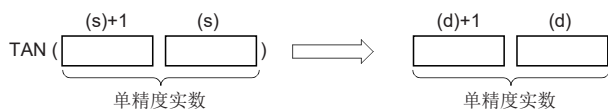
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行TAN(正切)运算的角度数据或存储了角度数据的起始软元件	$0, 2^{-126} \leq (s) < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储运算结果的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—	—

功能

- 对(s)中指定角度的TAN(正切)值进行运算, 将运算结果存储到(d)中指定的软元件编号中。



- (s)中指定的角度, 以弧度单位(角度 $\times\pi\div 180$)进行设置。
- 运算结果为-0或发生了下溢时, 运算结果将变为0。
- 通过工程工具设置输入值的情况下, 有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项, 请参阅下述内容。

☞ 38页 注意事项

注意事项

(s)中指定的角度为 $\pi/2$ 弧度、 $(3/2)\pi$ 弧度的情况下, 弧度值中将产生运算误差, 因此不发生错误。

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、 $\pm\infty$ 时。

要点

关于角度 \leftrightarrow 弧度转换的有关内容, 请参阅RAD(P)指令、DEG(P)指令。

☞ 876页 RAD(P)

☞ 878页 DEG(P)

单精度实数 SIN^{-1} 运算

ASIN(P)

通过单精度实数中指定的SIN值进行角度运算。

梯形图	ST*1
	$\text{ENO} := \text{ASINP}(\text{EN}, \text{s}, \text{d});$

FBD/LD*1

*1 在ST、FBD/LD中不支持ASIN指令。应使用通用函数的ASIN。
 1274页 ASIN(E)

■执行条件

指令	执行条件
ASIN	
ASINP	

设置数据

■内容、范围、数据类型

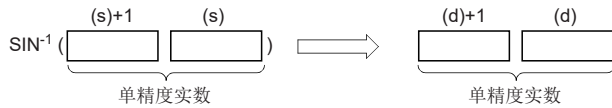
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行 SIN^{-1} (反正弦)运算的SIN值或存储了SIN值的起始软元件	-1.0~1.0	单精度实数	ANYREAL_32
(d)	存储运算结果的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—

功能

- 通过(s)中指定角度的SIN(正弦)值进行角度运算后,将运算结果存储到(d)中指定的软元件编号中。



- (s)中指定的SIN值可在-1.0~1.0的范围内设置。
- (d)中存储的角度(运算结果)以弧度单位存储。
- 运算结果为-0或发生了下溢时,运算结果将变为0。
- 通过工程工具设置输入值的情况下,有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项,请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、 $\pm\infty$ 时。
3285H	(s)中指定的值超出-1.0~1.0的范围时。

要点

关于弧度 \leftrightarrow 角度转换,请参阅RAD(P)指令、DEG(P)指令。

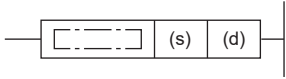
☞ 876页 RAD(P)

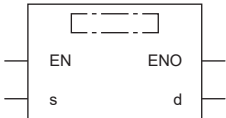
☞ 878页 DEG(P)

单精度实数 COS^{-1} 运算

ACOS(P)


通过单精度实数中指定的COS值进行角度运算。

梯形图	ST*1
	ENO:=ACOSP(EN, s, d)

FBD/LD*1


*1 在ST、FBD/LD中不支持ACOS指令。应使用通用函数的ACOS。
 1275页 ACOS(_E)

■执行条件

指令	执行条件
ACOS	
ACOSP	

设置数据

■内容、范围、数据类型

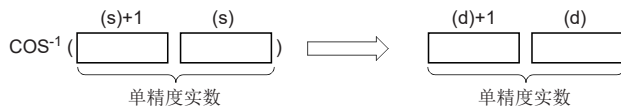
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行 COS^{-1} (反余弦)运算的COS值或存储了COS值的起始软元件	-1.0~1.0	单精度实数	ANYREAL_32
(d)	存储运算结果的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—	—

功能

- 通过(s)中指定的角度的COS(余弦)值进行角度运算后, 将运算结果存储到(d)中指定的软元件编号中。



- (s)中指定的COS值可在-1.0~1.0的范围内设置。
- (d)中存储的角度(运算结果)以弧度单位存储。
- 运算结果为-0或发生了下溢时, 运算结果将变为0。
- 通过工程工具设置输入值的情况下, 有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项, 请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、±∞时。
3285H	(s)中指定的值超出-1.0~1.0的范围时。

要点

关于弧度↔角度转换, 请参阅RAD(P)指令、DEG(P)指令。

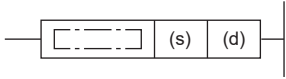
☞ 876页 RAD(P)

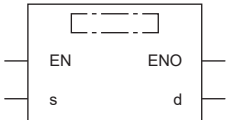
☞ 878页 DEG(P)

单精度实数 TAN^{-1} 运算

ATAN(P)

通过单精度实数中指定的TAN值进行角度运算。

梯形图	ST*1
	ENO:=ATANP (EN, s, d) ;

FBD/LD*1


*1 在ST、FBD/LD中不支持ATAN指令。应使用通用函数的ATAN。
 1276页 ATAN(_E)

■执行条件

指令	执行条件
ATAN	
ATANP	

设置数据

■内容、范围、数据类型

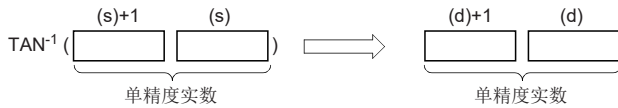
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行 TAN^{-1} (反正切)运算的TAN值或存储了TAN值的起始软元件	$0, 2^{-126} \leq (s) < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储运算结果的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—

功能

- 通过(s)中指定角度的TAN(正切)值进行角度运算后,将运算结果存储到(d)中指定的软元件编号中。



- (d)中存储的角度(运算结果)以弧度单位存储。
- 运算结果为-0或发生了下溢时,运算结果将变为0。
- 通过工程工具设置输入值的情况下,有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项,请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、±∞时。

要点

关于弧度↔角度转换,请参阅RAD(P)指令、DEG(P)指令。

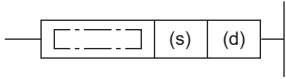
☞ 876页 RAD(P)

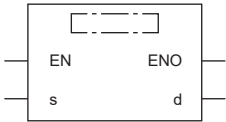
☞ 878页 DEG(P)

双精度实数SIN运算

SIND(P)



对双精度实数中指定角度的SIN(正弦)值进行运算。

梯形图	ST*1
	ENO:=SINDP(EN, s, d);

FBD/LD*1


*1 在ST、FBD/LD中不支持SIND指令。应使用通用函数的SIN。
 1271页 SIN(E)

■执行条件

指令	执行条件
SIND	
SINDP	

设置数据

■内容、范围、数据类型

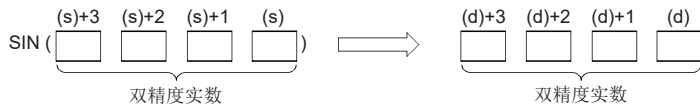
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行SIN(正弦)运算的角度数据或存储了角度数据的起始软元件	$0, 2^{-1022} \leq s < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储运算结果的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 对(s)中指定角度的SIN(正弦)值进行运算后,将运算结果存储到(d)中指定的软元件编号中。



- (s)中指定的角度,以弧度单位(角度 $\times\pi\div 180$)进行设置。
- 运算结果为-0或发生了下溢时,运算结果将变为0。
- 通过工程工具设置输入值的情况下,有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项,请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、 $\pm\infty$ 时。

要点

关于角度 \leftrightarrow 弧度转换的有关内容,请参阅RADD(P)指令、DEGD(P)指令。

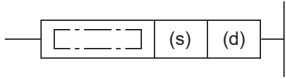
☞ 880页 RADD(P)

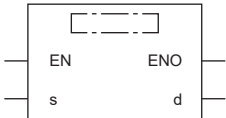
☞ 882页 DEGD(P)

双精度实数COS运算

COSD(P)

对双精度实数中指定角度的COS(余弦)值进行运算。

梯形图	ST*1
	ENO:=COSDP(EN, s, d);

FBD/LD*1


*1 在ST、FBD/LD中不支持COSD指令。应使用通用函数的COS。
 1272页 COS(_E)

■执行条件

指令	执行条件
COSD	
COSDP	

设置数据

■内容、范围、数据类型

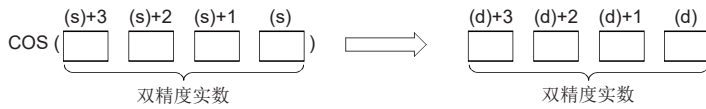
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行COS(余弦)运算的角度数据或存储了角度数据的起始软元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储运算结果的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 对(s)中指定角度的COS(余弦)值进行运算后,将运算结果存储到(d)中指定的软元件编号中。



- (s)中指定的角度,以弧度单位(角度 $\times\pi\div 180$)进行设置。
- 运算结果为-0或发生了下溢时,运算结果将变为0。
- 通过工程工具设置输入值的情况下,有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项,请参阅下述内容。

38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、 $\pm\infty$ 时。

要点

关于角度 \leftrightarrow 弧度转换的有关内容,请参阅RADD(P)指令、DEGD(P)指令。

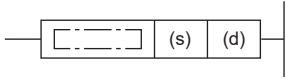
880页 RADD(P)

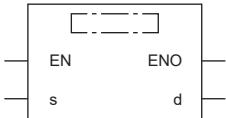
882页 DEGD(P)

双精度实数TAN运算

TAND(P)



对双精度实数中指定角度的TAN(正切)值进行运算。

梯形图	ST*1
	ENO:=TANDP(EN, s, d);

FBD/LD*1


*1 在ST、FBD/LD中不支持TAND指令。应使用通用函数的TAN。
 1273页 TAN(_E)

■执行条件

指令	执行条件
TAND	
TANDP	

设置数据

■内容、范围、数据类型

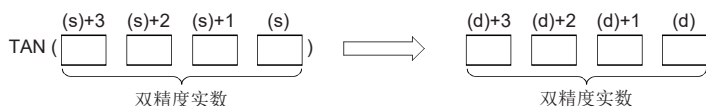
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行TAN(正切)运算的角度数据或存储了角度数据的起始软元件	$0, 2^{-1022} \leq s < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储运算结果的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—

功能

- 对(s)中指定角度的TAN(正切)值进行运算, 将运算结果存储到(d)中指定的软元件编号中。



- (s)中指定的角度, 以弧度单位(角度 $\times\pi\div 180$)进行设置。
- 运算结果为-0或发生了下溢时, 运算结果将变为0。
- 通过工程工具设置输入值的情况下, 有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项, 请参阅下述内容。

38页 注意事项

注意事项

(s)中指定的角度为 $\pi/2$ 弧度、 $(3/2)\pi$ 弧度的情况下, 弧度值中将产生运算误差, 因此不发生错误。

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、 $\pm\infty$ 时。

要点

关于角度 \leftrightarrow 弧度转换的有关内容, 请参阅RADD(P)指令、DEGD(P)指令。

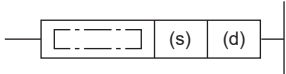
880页 RADD(P)

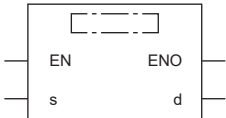
882页 DEGD(P)

双精度实数 SIN^{-1} 运算

ASIND(P)

通过以双精度实数指定的SIN值进行角度运算。



梯形图	ST*1
	$\text{ENO} := \text{ASINDP}(\text{EN}, \text{s}, \text{d})$

FBD/LD*1


*1 在ST、FBD/LD中不支持ASIND指令。应使用通用函数的ASIN。

☞ 1274页 ASIN(_E)

■执行条件

指令	执行条件
ASIND	
ASINDP	

设置数据

■内容、范围、数据类型

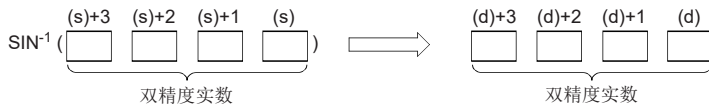
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行 SIN^{-1} (反正弦)运算的SIN值或存储了SIN值的起始软元件	-1.0~1.0	双精度实数	ANYREAL_64
(d)	存储运算结果的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 通过(s)中指定角度的SIN(正弦)值进行角度运算后,将运算结果存储到(d)中指定的软元件编号中。



- (s)中指定的SIN值可在-1.0~1.0的范围内设置。
- (d)中存储的角度(运算结果)以弧度单位存储。
- 运算结果为-0或发生了下溢时,运算结果将变为0。
- 通过工程工具设置输入值的情况下,有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项,请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、±∞时。
3285H	(s)中指定的值超出-1.0~1.0的范围时。

要点

关于弧度↔角度转换的有关内容,请参阅RADD(P)指令、DEGD(P)指令。

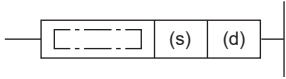
☞ 880页 RADD(P)

☞ 882页 DEGD(P)

双精度实数 \cos^{-1} 运算

ACOSD(P)

通过以双精度实数指定的COS值进行角度运算。

梯形图	ST*1
	$ENO := ACOSDP(EN, s, d);$

FBD/LD*1


*1 在ST、FBD/LD中不支持ACOSD指令。应使用通用函数的ACOS。
 1275页 ACOS(E)

■执行条件

指令	执行条件
ACOSD	
ACOSDP	

设置数据

■内容、范围、数据类型

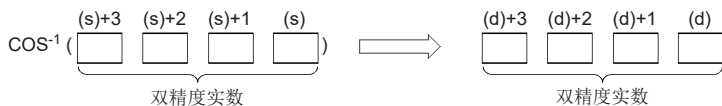
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行 \cos^{-1} (反余弦)运算的COS值或存储了COS值的起始软元件	-1.0~1.0	双精度实数	ANYREAL_64
(d)	存储运算结果的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—	

功能

- 通过(s)中指定的COS(余弦)值进行角度运算后,将运算结果存储到(d)中指定的软元件编号中。



- (s)中指定的COS值可在-1.0~1.0的范围内设置。
- (d)中存储的角度(运算结果)以弧度单位存储。
- 运算结果为-0或发生了下溢时,运算结果将变为0。
- 通过工程工具设置输入值的情况下,有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项,请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、±∞时。
3285H	(s)中指定的值超出-1.0~1.0的范围时。

要点

关于弧度↔角度转换的有关内容,请参阅RADD(P)指令、DEGD(P)指令。

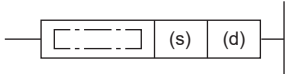
☞ 880页 RADD(P)

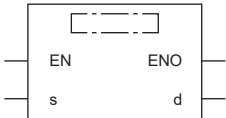
☞ 882页 DEGD(P)

双精度实数 TAN^{-1} 运算

ATAND(P)

通过以双精度实数指定的TAN值进行角度运算。



梯形图	ST*1
	ENO:=ATANDP(EN, s, d);

FBD/LD*1


*1 在ST、FBD/LD中不支持ATAND指令。应使用通用函数的ATAN。

☞ 1276页 ATAN(_E)

■执行条件

指令	执行条件
ATAND	
ATANDP	

设置数据

■内容、范围、数据类型

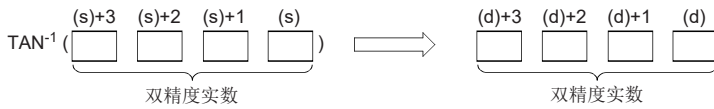
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行 TAN^{-1} (反正切)运算的TAN值或存储了TAN值的起始软元件	$0, 2^{-1022} \leq s < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储运算结果的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 通过(s)中指定角度的TAN(正切)值进行角度运算后,将运算结果存储到(d)中指定的软元件编号中。



- (d)中存储的角度(运算结果)以弧度单位存储。
- 运算结果为-0或发生了下溢时,运算结果将变为0。
- 通过工程工具设置输入值的情况下,有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项,请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、±∞时。

要点

关于弧度↔角度转换的有关内容,请参阅RADD(P)指令、DEGD(P)指令。

☞ 880页 RADD(P)

☞ 882页 DEGD(P)

BCD型SIN运算

BSIN(P)

对以BCD值指定的角度的SIN(正弦)值进行运算。

梯形图	ST
	ENO:=BSIN(EN, s, d); ENO:=BSINP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
BSIN	
BSINP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行SIN(正弦)运算的数据或存储了数据的软元件	0~9999	BCD4位	ANY16
(d)	存储运算结果的起始软元件	—	BCD4位	ANY16_ARRAY (要素数: 3)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 对(s)中指定的值(角度)的SIN(正弦)值进行运算, 将运算结果的符号存储到(d)中指定的软元件中, 将运算结果存储到(d)+1、(d)+2中指定的软元件中。

$$\text{SIN}(s) = \boxed{\text{(d)}} \boxed{\text{(d)+1}} . \boxed{\text{(d)+2}}$$

(d): 符号

(d)+1: 整数部

(d)+2: 小数部

- (s)中指定的值, 将0~360°(DEG. 单位)以BCD值进行设置。
- (d)中存储的运算结果的符号, 在运算结果为正时存储0, 为负时存储1。
- (d)+1、(d)+2中存储的运算结果为BCD值且在-1.000~1.000的范围内。
- 运算结果, 变为对小数部第5位进行了四舍五入后的值。

出错

错误代码(SD0)	内容
3285H	(s)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> 指定的数据不是BCD值 指定的数据超出了0~360的范围

BCD型COS运算

BCOS (P)

对以BCD值指定的角度的COS (余弦) 值进行运算。

梯形图	ST
	ENO:=BCOS (EN, s, d) ; ENO:=BCOSP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
BCOS	
BCOSP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s)	进行COS (余弦) 运算的数据或存储了数据的软元件	0~9999	BCD4位	ANY16
(d)	存储运算结果的起始软元件	—	BCD4位	ANY16_ARRAY (要素数: 3)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 对(s)中指定的值(角度)的COS(余弦)值进行运算,将运算结果的符号存储到(d)中指定的字软元件中,将运算结果存储到(d)+1、(d)+2中指定的字软元件中。

$$\text{COS}(s) = \boxed{\text{(d)}} \boxed{\text{(d)+1}} . \boxed{\text{(d)+2}}$$

(d): 符号

(d)+1: 整数部

(d)+2: 小数部

- (s)中指定的值,将0~360°(DEG.单位)以BCD值进行设置。
- (d)中存储的运算结果的符号,在运算结果为正时存储0,为负时存储1。
- (d)+1、(d)+2中存储的运算结果为BCD值且在-1.000~1.000的范围内。
- 运算结果,变为对小数部第5位进行了四舍五入后的值。

出错

错误代码(SD0)	内容
3285H	(s)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> 指定的数据不是BCD值 指定的数据超出了0~360的范围

BCD型TAN运算

BTAN(P)

对以BCD值指定的角度的TAN(正切)值进行运算。

梯形图	ST
	ENO:=BTAN(EN, s, d); ENO:=BTANP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
BTAN	
BTANP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行TAN(正切)运算的数据或存储了数据的软元件	0~9999	BCD4位	ANY16
(d)	存储运算结果的起始软元件	—	BCD4位	ANY16_ARRAY (要素数: 3)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—	

功能

- 对(s)中指定的值(角度)的TAN(正切)值进行运算, 将运算结果的符号存储到(d)中指定的软元件中, 将运算结果存储到(d)+1、(d)+2中指定的软元件中。

$$\text{TAN}(s) = \boxed{\text{(d)}} \boxed{\text{(d)+1}} . \boxed{\text{(d)+2}}$$

(d): 符号

(d)+1: 整数部

(d)+2: 小数部

- (s)中指定的值, 将0~360°(DEG. 单位)以BCD值进行设置。
- (d)中存储的运算结果的符号, 在运算结果为正时存储0, 为负时存储1。
- (d)+1、(d)+2中存储的运算结果为BCD值且在-57.2901~57.2903的范围内。
- 运算结果, 变为对小数部第5位进行了四舍五入后的值。

出错

错误代码(SD0)	内容
3285H	(s)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> 指定的数据不是BCD值 指定的数据超出了0~360的范围 指定的数据为90°、270°

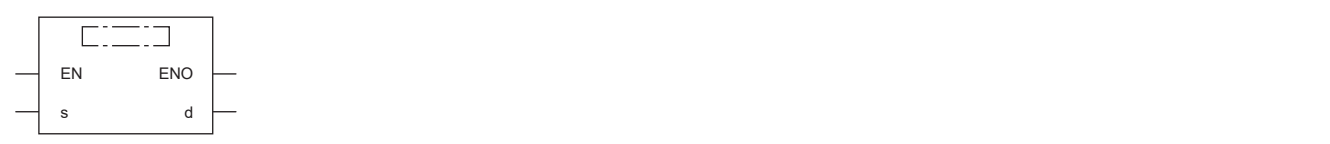
BCD型 SIN^{-1} 运算

BASIN(P)

对以BCD值指定的值的 SIN^{-1} (反正弦)值进行运算。

梯形图	ST
	ENO:=BASIN(EN, s, d); ENO:=BASINP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
BASIN	
BASINP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储进行 SIN^{-1} (反正弦)运算的数据的起始软元件	0~9999	BCD4位	ANY16_ARRAY (要素数: 3)
(d)	存储运算结果的软元件	—	BCD4位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—	—

功能

- 对(s)中指定的值的 SIN^{-1} (反正弦)值进行运算, 将运算结果(角度)存储到(d)中指定的软元件中。

$$\text{SIN}^{-1} \left(\overset{(s)}{\boxed{}} \overset{(s)+1}{\boxed{}} \overset{(s)+2}{\boxed{}} \right) = (d)$$

(s): 符号

(s)+1: 整数部

(s)+2: 小数部

- (s)中设置进行运算的数据的符号。运算数据为正时存储0, 为负时存储1。
- (s)+1、(s)+2中, 将进行运算的数据的整数部、小数部分别以BCD值进行存储。(可在0~1.0000的范围内设置。)
- (d)中存储的运算结果为, 在0~90°、270~360°(DEG. 单位)范围内的BCD值。
- 运算结果是对小数部进行了四舍五入后的值。

出错

错误代码(SD0)	内容
3285H	(s)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> 指定的数据不是BCD值 指定的数据超出了-1.0000~1.0000的范围

BCD型 COS^{-1} 运算

BACOS (P)

对以BCD值指定的值的 COS^{-1} (反余弦)值进行运算。

梯形图	ST
	ENO:=BACOS (EN, s, d) ; ENO:=BACOSP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
BACOS	
BACOSP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s)	存储进行 COS^{-1} (反余弦)运算的数据的起始软元件	0~9999	BCD4位	ANY16_ARRAY (要素数: 3)
(d)	存储运算结果的软元件	—	BCD4位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—	

功能

- 对(s)中指定的值的 \cos^{-1} 对(反余弦)值进行运算,将运算结果(角度)存储到(d)中指定的软元件中。

$$\cos^{-1} \left(\overset{(s)}{\boxed{}} \overset{(s)+1}{\boxed{}} \overset{(s)+2}{\boxed{}} \right) = (d)$$

(s): 符号

(s)+1: 整数部

(s)+2: 小数部

- (s)中设置进行运算的数据的符号。运算数据为正时存储0,为负时存储1。
- (s)+1、(s)+2中,将进行运算的数据的整数部、小数部分别以BCD值进行存储。(可在0~1.0000的范围内设置。)
- (d)中存储的运算结果为,在0~180°(DEG.单位)范围内的BCD值。
- 运算结果是对小数部进行了四舍五入后的值。

出错

错误代码(SD0)	内容
3285H	(s)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> 指定的数据不是BCD值 指定的数据超出了-1.0000~1.0000的范围

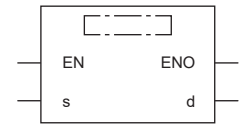
BCD型 TAN^{-1} 运算

BATAN(P)

对以BCD值指定的值的 TAN^{-1} (反正切)值进行运算。

梯形图	ST
	ENO:=BATAN(EN, s, d); ENO:=BATANP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
BATAN	
BATANP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储进行 TAN^{-1} (反正切)运算的数据的起始软元件	0~9999	BCD4位	ANY16_ARRAY (要素数: 3)
(d)	存储运算结果的软元件	—	BCD4位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—	

功能

- 对(s)中指定的值的 TAN^{-1} (反正切)值进行运算, 将运算结果(角度)存储到(d)中指定的软元件中。

$$\text{TAN}^{-1} \left(\begin{array}{c} \text{(s)} \\ \square \end{array} \begin{array}{c} \text{(s)+1} \\ \square \end{array} \begin{array}{c} \text{(s)+2} \\ \square \end{array} \right) = \text{(d)}$$

(s): 符号

(s)+1: 整数部

(s)+2: 小数部

- (s)中设置进行运算的数据的符号。运算数据为正时存储0, 为负时存储1。
- (s)+1、(s)+2中, 将进行运算的数据的整数部、小数部分别以BCD值进行存储。(可在0~9999.9999的范围内设置。)
- (d)中存储的运算结果为, 在 $0 \sim 90^\circ$ 、 $270 \sim 360^\circ$ (DEG. 单位)范围内的BCD值。
- 运算结果是对小数部进行了四舍五入后的值。

出错

错误代码(SD0)	内容
3285H	(s)中指定的数据不是BCD值时。

单精度实数角度→弧度转换

RAD(P)

将角度的大小单位从以单精度实数指定的度单位转换为弧度单位。

梯形图	ST
	ENO:=RAD(EN, s, d); ENO:=RADP(EN, s, d);

FBD/LD

■执行条件

指令	执行条件
RAD	
RADP	

设置数据

■内容、范围、数据类型

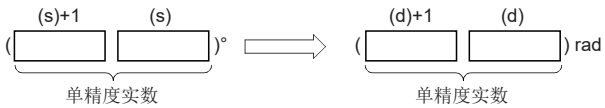
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行弧度单位转换的角度或存储了角度的起始软元件	$0, 2^{-126} \leq (s) < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储进行了弧度单位转换的值的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—	—

功能

- 将角度的大小单位从(s)中指定的度单位(DEG. 单位)转换为弧度单位后, 存储到(d)中指定的软元件编号中。



- 度单位→弧度单位转换按下述方式进行。

$$\text{弧度单位} = \text{度单位} \times \frac{\pi}{180}$$

- 通过工程工具设置输入值的情况下, 有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项, 请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、±∞时。
3283H	(d)超出下述范围时。(发生了上溢时。) (d) < 2 ¹²⁸

单精度实数弧度→角度转换

DEG (P)

将角度的大小单位从以单精度实数指定的弧度单位转换为度单位。

梯形图 	ST ENO:=DEG (EN, s, d); ENO:=DEGP (EN, s, d);
----------------	--

FBD/LD

■执行条件

指令	执行条件
DEG	
DEGP	

设置数据

■内容、范围、数据类型

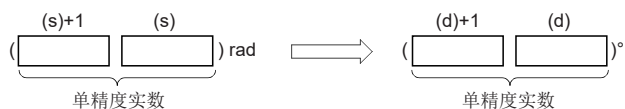
操作数	内容	范围	数据类型	数据类型 (标签)
(s)	进行度单位转换的弧度角度或存储了弧度角度的起始软元件	$0, 2^{-126} \leq (s) < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储进行了度单位转换的值的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—	

功能

- 将角度的大小单位从(s)中指定的弧度单位转换为度单位(DEG. 单位)后, 存储到(d)中指定的软元件编号中。



- 弧度单位→度单位转换按下述方式进行。

$$\text{度单位} = \text{弧度单位} \times \frac{180}{\pi}$$

- 通过工程工具设置输入值的情况下, 有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项, 请参阅下述内容。

☞ 38页 注意事项

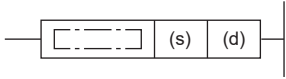
出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、±∞时。
3283H	(d)超出下述范围时。(发生了上溢时。) (d) < 2 ¹²⁸

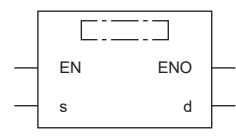
双精度实数角度→弧度转换

RADD(P)



将角度的大小单位从以双精度实数指定的度单位转换为弧度单位。

梯形图	ST
	ENO:=RADD(EN, s, d); ENO:=RADDP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
RADD	
RADDP	

设置数据

■内容、范围、数据类型

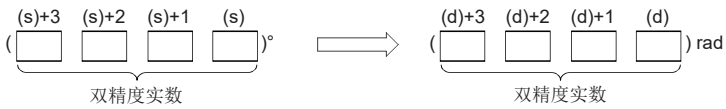
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行弧度单位转换的角度或存储了角度的起始软元件	$0, 2^{-1022} \leq s < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储进行了弧度单位转换的值的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 将角度的大小单位从(s)中指定的度单位(DEG. 单位)转换为弧度单位后, 存储到(d)中指定的软元件编号中。



- 度单位→弧度单位转换按下述方式进行。

$$\text{弧度单位} = \text{度单位} \times \frac{\pi}{180}$$

- 运算结果为-0或发生了下溢时, 运算结果将变为0。
- 通过工程工具设置输入值的情况下, 有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项, 请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、±∞时。
3283H	(d)超出下述范围时。(发生了上溢时。) (d) < 2 ¹⁰²⁴

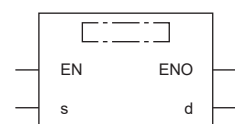
双精度实数弧度→角度转换

DEGD (P)

将角度的大小单位从以双精度实数指定的弧度单位转换为度单位。

梯形图	ST
	ENO:=DEGD (EN, s, d) ; ENO:=DEGDP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
DEGD	
DEGDP	

设置数据

■内容、范围、数据类型

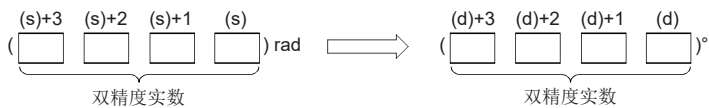
操作数	内容	范围	数据类型	数据类型 (标签)
(s)	进行度单位转换的弧度角度或存储了弧度角度的起始软元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储进行了度单位转换的值的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—	

功能

- 将角度的大小单位从(s)中指定的弧度单位转换为度单位(DEG. 单位)后, 存储到(d)中指定的软元件编号中。



- 弧度单位→度单位转换按下述方式进行。

$$\text{度单位} = \text{弧度单位} \times \frac{180}{\pi}$$

- 运算结果为-0或发生了下溢时, 运算结果将变为0。
- 通过工程工具设置输入值的情况下, 有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项, 请参阅下述内容。

☞ 38页 注意事项

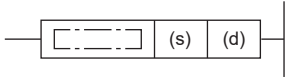
出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、±∞时。
3283H	(d)超出下述范围时。(发生了上溢时。) (d) < 2 ¹⁰²⁴

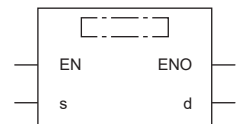
单精度实数平方根

ESQRT (P)

对以单精度实数指定的值的平方根进行运算。

梯形图	ST
	ENO:=ESQRT (EN, s, d) ; ENO:=ESQRTP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
ESQRT	
ESQRTP	

设置数据

■内容、范围、数据类型

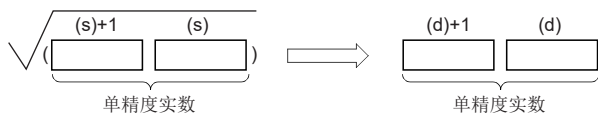
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行平方根运算的数据或存储了数据的起始软元件	$0, 2^{-126} \leq (s) < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储运算结果的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—	

功能

- 对(s)中指定的值的平方根进行运算后，将运算结果存储到(d)中指定的软元件编号中。



- (s)中指定的值只能设置为正的数。(不能以负的数进行运算。)
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

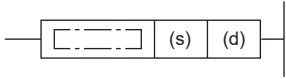
出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、 $\pm\infty$ 时。
3285H	(s)中指定的值为负的数时。

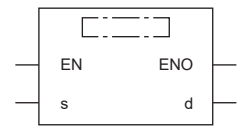
双精度实数平方根

EDSQRT (P)


对以双精度实数指定的值的平方根进行运算。

梯形图	ST
	<pre>ENO:=EDSQRT (EN, s, d); ENO:=EDSQRTP (EN, s, d);</pre>

FBD/LD



■执行条件

指令	执行条件
EDSQRT	
EDSQRTP	

设置数据

■内容、范围、数据类型

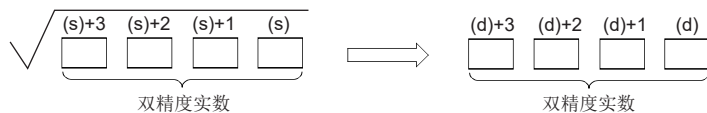
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行平方根运算的数据或存储了数据的起始软元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储运算结果的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 对(s)中指定的值的平方根进行运算后，将运算结果存储到(d)中指定的软元件编号中。



- (s)中指定的值只能设置为正的数。(不能以负的数进行运算。)
- 运算结果为-0或发生了下溢时，运算结果将变为0。
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

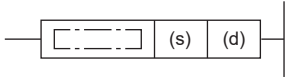
出错

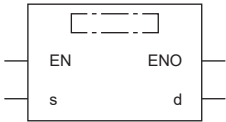
错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、 $\pm\infty$ 时。
3285H	(s)中指定的值为负的数时。

单精度实数指数运算

EXP (P)

对以单精度实数指定的值的指数进行运算。

梯形图	ST*1
	ENO:=EXPP (EN, s, d) ;

FBD/LD*1


*1 在ST、FBD/LD中不支持EXP指令。应使用通用函数的EXP。
 1270页 EXP (E)

■执行条件

指令	执行条件
EXP	
EXPP	

设置数据

■内容、范围、数据类型

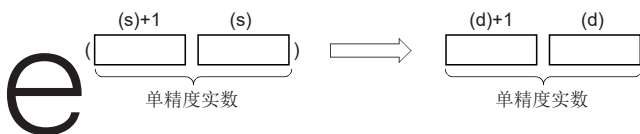
操作数	内容	范围	数据类型	数据类型 (标签)
(s)	进行指数运算的数据或存储了数据的起始软元件	$0, 2^{-126} \leq (s) < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储运算结果的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	○	○	○	○	○	—	—	—	—

功能

- 进行(s)中指定的值的指数运算后, 将运算结果存储到(d)中指定的软元件中。



- 在指数运算中, 将底(e)以“2.71828”进行运算。
- 通过工程工具设置输入值的情况下, 有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项, 请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、±∞时。
3283H	(d)超出下述范围时。(发生了上溢时。) (d) < 2 ¹²⁸

要点

- EXP(P)指令以自然对数进行运算。求出常用对数的值的情况下, 应将常用对数用0.43429相除后的值指定到(s)中。

$$10^X = e^{\frac{X}{0.43429}}$$

双精度实数指数运算

EXPDP (P)

对以双精度实数指定的值的指数进行运算。

梯形图	ST*1
	ENO:=EXPDP (EN, s, d) ;

FBD/LD*1

*1 在ST、FBD/LD中不支持EXPDP指令。应使用通用函数的EXP。

☞ 1270页 EXP (E)

■执行条件

指令	执行条件
EXPDP	
EXPDP	

设置数据

■内容、范围、数据类型

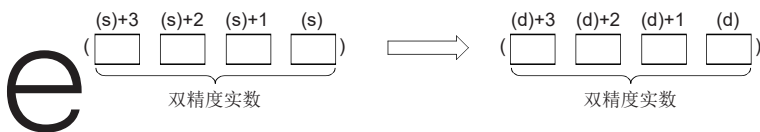
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行指数运算的数据或存储了数据的起始软元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储运算结果的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—

功能

- 进行(s)中指定的值的指数运算后，将运算结果存储到(d)中指定的软元件中。



- 在指数运算中，将底(e)以“2.71828”进行运算。
- 运算结果为-0或发生了下溢时，运算结果将变为0。
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、±∞时。
3283H	(d)超出下述范围时。(发生了上溢时。) (d) < 2 ¹⁰²⁴

要点

- 在EXPD(P)指令中，以自然对数进行运算。求出常用对数的值的情况下，应将常用对数用0.43429相除后的值指定到(s)中。

$$10^X = e^{\frac{X}{0.43429}}$$

单精度实数自然对数运算

LOG(P)

对以单精度实数指定的值的自然对数(e)为底时的对数进行运算。

梯形图	ST*1
	ENO:=LOGP(EN, s, d);

FBD/LD*1

*1 在ST、FBD/LD中不支持LOG指令。应使用通用函数的LOG。
 1268页 LOG(_E)

■执行条件

指令	执行条件
LOG	
LOGP	

设置数据

■内容、范围、数据类型

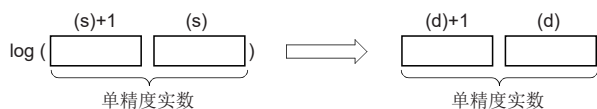
操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行自然对数运算的数据或存储了数据的起始软元件	$0, 2^{-126} \leq (s) < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储运算结果的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	—	—	○	○	○	○	○	—	—	—	—

功能

- 对(s)中指定的值的自然对数e为底时的对数进行运算后，将运算结果存储到(d)中指定的软元件中。



- (s)中指定的值只能设置为正的数。(不能以负的数进行运算。)
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

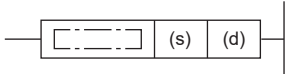
出错

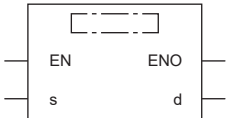
错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、±∞时。
3285H	(s)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> 指定的值为负的数 指定的值为0

双精度实数自然对数运算

LOGD (P)

对以双精度实数指定的值的自然对数 (e) 为底时的对数进行运算。



梯形图	ST*1
	ENO:=LOGDP (EN, s, d) ;

FBD/LD*1


*1 在ST、FBD/LD中不支持LOGD指令。应使用通用函数的LOG。

☞ 1268页 LOG (E)

■执行条件

指令	执行条件
LOGD	
LOGDP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s)	进行自然对数运算的数据或存储了数据的起始软元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储运算结果的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—

功能

- 对(s)中指定的值的自然对数e为底时的对数进行运算后, 将运算结果存储到(d)中指定的软元件中。



- (s)中指定的值只能设置为正的数。(不能以负的数进行运算。)
- 运算结果为-0或发生了下溢时, 运算结果将变为0。
- 通过工程工具设置输入值的情况下, 有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项, 请参阅下述内容。

☞ 38页 注意事项

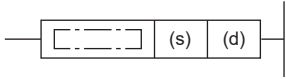
出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、 $\pm\infty$ 时。
3285H	(s)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> 指定的值为负的数 指定的值为0

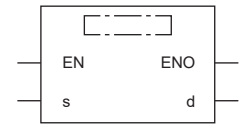
BCD4位平方根

BSQRT (P)

对以BCD值(4位)指定的值的平方根进行运算。

梯形图	ST
	ENO:=BSQRT (EN, s, d) ; ENO:=BSQRTP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
BSQRT	
BSQRTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行平方根运算的数据或存储了数据的软元件	0~9999	BCD4位	ANY16
(d)	存储运算结果的起始软元件	—	BCD4位	ANY16_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 对(s)中指定的BCD4位数据的平方根进行运算后, 将运算结果存储到(d)中指定的软元件中。

$$\sqrt{\boxed{(s)}} = \boxed{(d)}.\boxed{(d)+1}$$

(d): 整数部

(d)+1: 小数部

- (s)中指定的值是BCD值且最大4位(0~9999)。
- (d)的运算结果将分别以BCD值存储0~9999.9999。
- 运算结果将变为舍去了小数部第5位数后的值。

出错

错误代码(SD0)	内容
3285H	(s)中指定的数据为BCD值以外时。

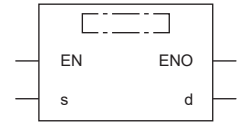
BCD8位平方根

BDSQRT (P)

对以BCD值 (8位) 指定的值的平方根进行运算。

梯形图	ST
	ENO:=BDSQRT (EN, s, d) ; ENO:=BDSQRTP (EN, s, d) ;

FBD/LD



■执行条件

指令	执行条件
BDSQRT	
BDSQRTP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行平方根运算的数据或存储了数据的起始软元件	0~99999999	BCD8位	ANY32
(d)	存储运算结果的起始软元件	—	BCD4位	ANY16_ARRAY (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 对(s)中指定的BCD8位数据的平方根进行运算后,将运算结果存储到(d)中指定的软元件中。

$$\sqrt{\left(\begin{array}{|c|} \hline (s)+1 \\ \hline \end{array} \begin{array}{|c|} \hline (s) \\ \hline \end{array} \right)} = \begin{array}{|c|} \hline (d) \\ \hline \end{array} . \begin{array}{|c|} \hline (d)+1 \\ \hline \end{array}$$

(s)+1、(s): 2字数据

(d): 整数部

(d)+1: 小数部

- (s)中指定的值是BCD值且最大8位(0~99999999)。
- (d)的运算结果将分别以BCD值存储0~9999.9999。
- 运算结果将变为舍去了小数部第5位数后的值。

出错

错误代码(SD0)	内容
3285H	(s)中指定的数据为BCD值以外时。

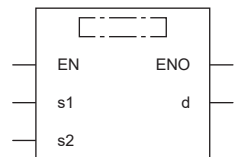
单精度实数幂运算

POW(P)

进行单精度实数的幂运算。

梯形图	ST
	ENO:=POW(EN, s1, s2, d); ENO:=POWP(EN, s1, s2, d);

FBD/LD



■执行条件

指令	执行条件
POW	
POWP	

设置数据

■内容、范围、数据类型

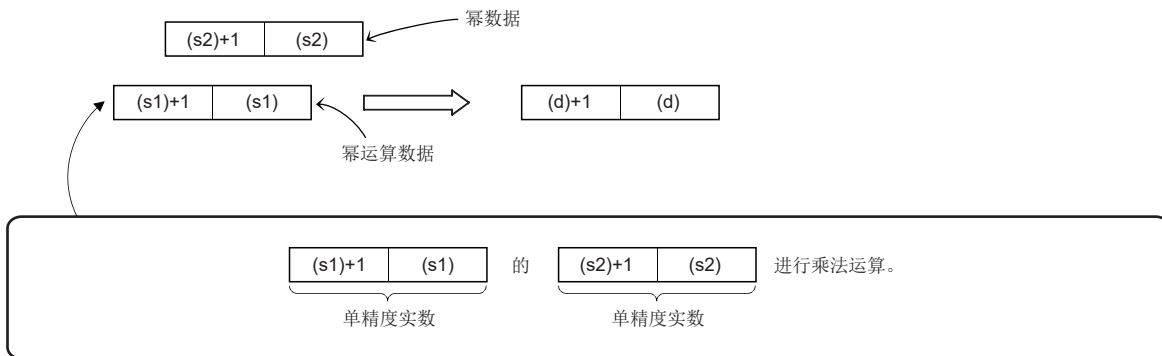
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	幂运算数据或存储了幂运算数据的起始软元件	$0, 2^{-126} \leq s1 < 2^{128}$	单精度实数	ANYREAL_32
(s2)	幂运算数据或存储了数据的起始软元件	$0, 2^{-126} \leq s2 < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储运算结果的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—

功能

- 对(s1)中指定的单精度实数以(s2)中指定的单精度实数进行幂运算后，将运算结果存储到(d)中指定的软元件中。



- (s1)、(s2)中可指定的值及可存储的值为 0 、 $2^{-126} \leq |\text{设定值(存储值)}| < 2^{128}$ 。
- 运算结果为 -0 或发生了下溢时，运算结果将变为 0 。
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s1)或(s2)中指定的值为 -0 、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d)超出下述范围时。(发生了上溢时) $ (d) < 2^{128}$

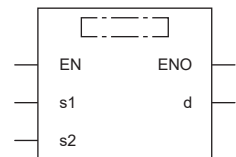
双精度实数幂运算

POWD(P)

进行双精度实数的幂运算。

梯形图	ST
	<pre>ENO:=POWD(EN, s1, s2, d); ENO:=POWDP(EN, s1, s2, d);</pre>

FBD/LD



■执行条件

指令	执行条件
POWD	
POWDP	

设置数据

■内容、范围、数据类型

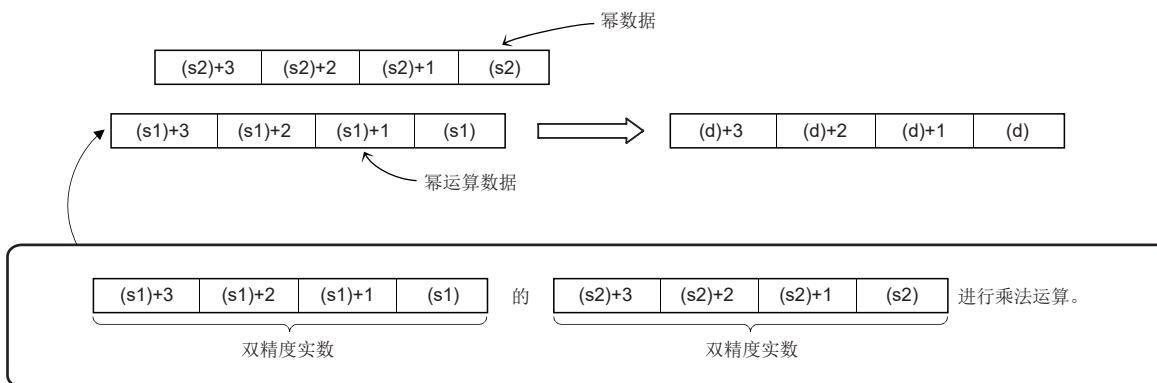
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	幂运算数据或存储了幂运算数据的起始软元件	$0, 2^{-1022} \leq s1 < 2^{1024}$	双精度实数	ANYREAL_64
(s2)	幂运算数据或存储了数据的起始软元件	$0, 2^{-1022} \leq s2 < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储运算结果的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—

功能

- 将(s1)中指定的双精度实数以(s2)中指定的双精度实数进行幂运算，将运算结果存储到(d)中指定的软元件中。



- (s1)、(s2)中可指定的值及可存储的值为 0 、 $2^{-1022} \leq |\text{设定值(存储值)}| < 2^{1024}$ 。
- 运算结果为 -0 或发生了下溢时，运算结果将变为 0 。
- 通过工程工具设置输入值的情况下，有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项，请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s1)或(s2)中指定的值为 -0 、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d)超出下述范围时。(发生了上溢时) $ (d) < 2^{1024}$

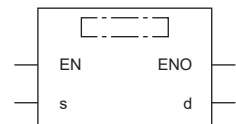
单精度实数常用对数运算

LOG10(P)

对以单精度实数指定的值的常用对数(以10为底的对数)进行运算。

梯形图	ST
	ENO:=LOG10(EN, s, d); ENO:=LOG10P(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
LOG10	
LOG10P	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行常用对数运算的数据或存储了数据的起始软元件	$0, 2^{-126} \leq (s) < 2^{128}$	单精度实数	ANYREAL_32
(d)	存储运算结果的起始软元件	—	单精度实数	ANYREAL_32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	○	○	○	○	○	—	—	—	—	

功能

- 对(s)中指定的值的常用对数(以10为底的对数)进行运算, 将运算结果存储到(d)中指定的软元件编号中。



- (s)中指定的值只能设置为正的数。(不能以负的数进行运算。)
- 运算结果为-0或发生了下溢时, 运算结果将变为0。
- 通过工程工具设置输入值的情况下, 有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项, 请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、 $\pm\infty$ 时。
3285H	(s)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> 指定的值为负的数 指定的值为0

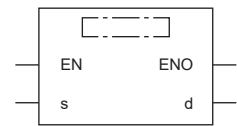
双精度实数常用对数运算

LOG10D(P)

对以双精度实数指定的值的常用对数(以10为底的对数)进行运算。

梯形图	ST
	ENO:=LOG10D(EN, s, d); ENO:=LOG10DP(EN, s, d);

FBD/LD



■执行条件

指令	执行条件
LOG10D	
LOG10DP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行常用对数运算的数据或存储了数据的起始软元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	双精度实数	ANYREAL_64
(d)	存储运算结果的起始软元件	—	双精度实数	ANYREAL_64
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	○	—	○	—	—	—	—	

功能

- 对(s)中指定的值的常用对数(以10为底的对数)进行运算, 将运算结果存储到(d)中指定的软元件编号中。

$$\log_{10} \left(\underbrace{((s)+3; (s)+2; (s)+1; (s))}_{\text{双精度实数}} \right) \implies \left(\underbrace{((d)+3; (d)+2; (d)+1; (d))}_{\text{双精度实数}} \right)$$

- (s)中指定的值只能设置为正的数。(不能以负的数进行运算。)
- 运算结果为-0或发生了下溢时, 运算结果将变为0。
- 通过工程工具设置输入值的情况下, 有可能产生化整误差。关于通过工程工具设置输入值情况下的注意事项, 请参阅下述内容。

☞ 38页 注意事项

出错

错误代码(SD0)	内容
3282H	(s)中指定的值为-0、非正规化数、非数、±∞时。
3285H	(s)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> 指定的值为负的数 指定的值为0

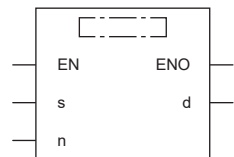
单精度实数最大值搜索

EMAX (P)

对单精度实数的块数据的最大值进行搜索。

梯形图	ST*1
	$ENO := EMAX(EN, s, n, d);$

FBD/LD*1



*1 在ST、FBD/LD中不支持EMAX指令。应使用通用函数的MAX。

☞ 1306页 MAX(_E)、MIN(_E)

■执行条件

指令	执行条件
EMAX	
EMAXP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	搜索最大值的单精度实数或存有单精度实数的起始软元件	$0, 2^{-126} \leq s < 2^{128}$	单精度实数	ANYREAL_32*1
(d)	存储搜索结果的起始软元件 (d)~(d)+1: 最大值 (d)+2: 位置 (d)+3: 个数	—	单精度实数	—*2 (ANY_REAL_32_ARRAY)
(n)	单精度实数块数据的点数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

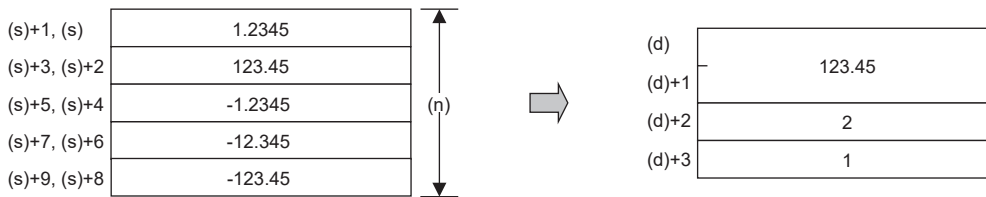
*2 不管使用的程序语言如何，通过软元件指定。请勿指定标签。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	○	○	—	—	—

功能

- 对从(s)开始的(n)点的单精度实数的块数据搜索最大值，存储到(d)中指定的软元件中。将最先检测到最大值的数据是从(s)开始的第几点存储到(d)+2中指定的软元件中，将与最大值相同值的个数存储到(d)+3中指定的软元件中。
- 关于搜索结果(位置)的值，将(s)中指定的块数据的起始作为第1点处理。



(d)、(d)+1: 最大值

(d)+2: 位置

(d)+3: 个数

- (n)中指定的值为0的情况下变为无处理。

出错

错误代码(SD0)	内容
3282H	(s)中指定的块数据中存在有单精度实数以外的值的情况下。

双精度实数最大值搜索

EDMAX (P)

对双精度实数的块数据的最大值进行搜索。

<p>梯形图</p>	<p>ST*1</p> <p>ENO:=EDMAXP (EN, s, n, d) ;</p>
-------------------	---

FBD/LD*1

*1 在ST、FBD/LD中不支持EDMAX指令。应使用通用函数的MAX。
 ↖ 1306页 MAX(_E)、MIN(_E)

■执行条件

指令	执行条件
EDMAX	
EDMAXP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	搜索最大值的双精度实数或存有双精度实数的起始软元件	$0, 2^{-1022} \leq s < 2^{1024}$	双精度实数	ANYREAL_64*1
(d)	存储搜索结果的起始软元件 (d)~(d)+3: 最大值 (d)+4: 位置 (d)+5: 个数	—	双精度实数	—*2 (ANY_REAL_64_ARRAY)
(n)	双精度实数块数据的点数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

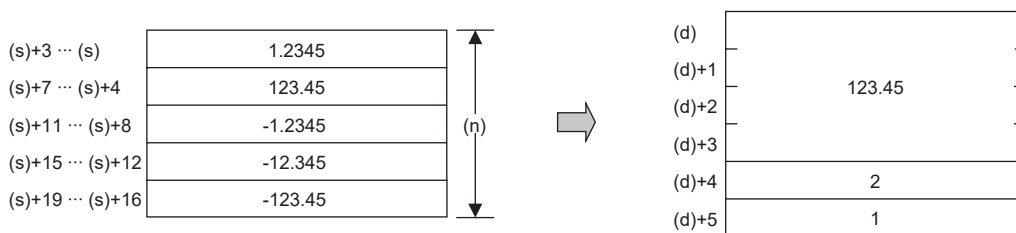
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。
 *2 不管使用的程序语言如何，通过软元件指定。请勿指定标签。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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	
(s)	—	—	○	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	○	—	—	—	—
(n)	○	○	○	○	○	—	○	○	—	—	—

功能

- 对从(s)开始的(n)点的双精度实数的块数据搜索最大值，存储到(d)中指定的软元件中。将最先检测到最大值的数据是从(s)开始的第几点存储到(d)+4中指定的软元件中，将与最大值相同值的个数存储到(d)+5中指定的软元件中。
- 关于搜索结果(位置)的值，将(s)中指定的块数据的起始作为第1点处理。



(d)、(d)+1、(d)+2、(d)+3: 最大值

(d)+4: 位置

(d)+5: 个数

- (n)中指定的值为0的情况下变为无处理。

出错

错误代码(SD0)	内容
3282H	(s)中指定的块数据中存在有双精度实数以外的值的情况下。

单精度实数最小值搜索

EMIN(P)

对单精度实数的块数据的最小值进行搜索。

梯形图	ST*1
	$ENO := EMINP(EN, s, n, d);$

FBD/LD*1

*1 在ST、FBD/LD中不支持EMIN指令。应使用通用函数的MIN。
 1306页 MAX(_E)、MIN(_E)

■执行条件

指令	执行条件
EMIN	
EMINP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行最小值搜索的单精度实数或存储了单精度实数的起始软元件	$0, 2^{-126} \leq (s) < 2^{128}$	单精度实数	ANYREAL_32*1
(d)	存储搜索结果的起始软元件 (d)~(d)+1: 最小值 (d)+2: 位置 (d)+3: 个数	—	单精度实数	—*2 (ANY_REAL_32_ARRAY)
(n)	单精度实数块数据的点数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

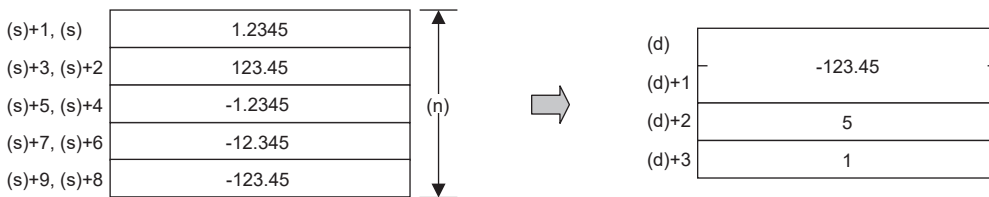
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。
 *2 不管使用的程序语言如何，通过软元件指定。请勿指定标签。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

- 对从(s)开始的(n)点的单精度实数的块数据搜索最小值，存储到(d)中指定的软元件中。将最先检测到最小值的数据是从(s)开始的第几点存储到(d)+2中指定的软元件中，将与最小值相同值的个数存储到(d)+3中指定的软元件中。
- 关于搜索结果(位置)的值，将(s)中指定的块数据的起始作为第1点处理。



(d)、(d)+1: 最小值

(d)+2: 位置

(d)+3: 个数

- (n)中指定的值为0的情况下变为无处理。

出错

错误代码(SD0)	内容
3282H	(s)中指定的块数据中存在有单精度实数以外的值的情况下。

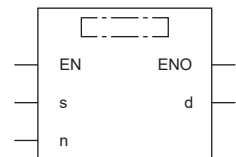
双精度实数最小值搜索

EDMIN(P)

对双精度实数的块数据的最小值进行搜索。

梯形图	ST*1
	$ENO := EDMINP(EN, s, n, d);$

FBD/LD*1



*1 在ST、FBD/LD中不支持EDMIN指令。应使用通用函数的MIN。
 1306页 MAX(_E)、MIN(_E)

■执行条件

指令	执行条件
EDMIN	
EDMINP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	进行最小值搜索的双精度实数或存储了双精度实数的起始软元件	$0, 2^{-1022} \leq (s) < 2^{1024}$	双精度实数	ANYREAL_64*1
(d)	存储搜索结果的起始软元件 (d)~(d)+3: 最小值 (d)+4: 位置 (d)+5: 个数	—	双精度实数	—*2 (ANY_REAL_64_ARRAY)
(n)	双精度实数块数据的点数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

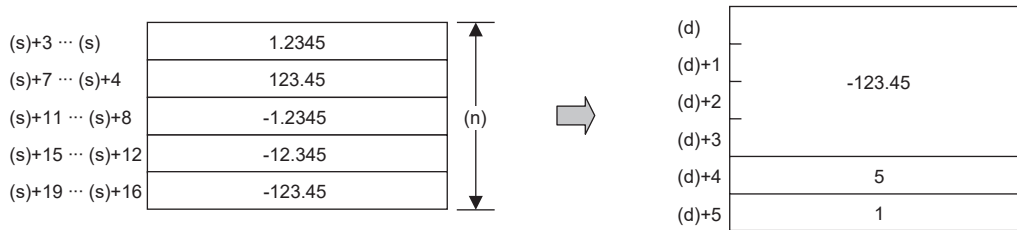
*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。
 *2 不管使用的程序语言如何，通过软元件指定。请勿指定标签。

■可用软元件

操作数	位		字		双字			间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

- 对从(s)开始的(n)点的双精度实数的块数据搜索最小值，存储到(d)中指定的软元件中。将最先检测到最小值的数据是从(s)开始的第几点存储到(d)+4中指定的软元件中，将与最小值相同值的个数存储到(d)+5中指定的软元件中。
- 关于搜索结果(位置)的值，将(s)中指定的块数据的起始作为第1点处理。



(d)、(d)+1、(d)+2、(d)+3: 最小值

(d)+4: 位置

(d)+5: 个数

- (n)中指定的值为0的情况下变为无处理。

出错

错误代码(SD0)	内容
3282H	(s)中指定的块数据中存在有双精度实数以外的值的情况下。

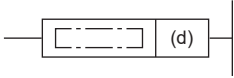
18 随机数

18.1 随机数指令

随机数发生

RND(P)

使之发生0到未滿32767的随机数，并存储到指定的软元件中。

梯形图	ST
	ENO:=RND(EN, d); ENO:=RNDP(EN, d);

FBD/LD



■执行条件

指令	执行条件
RND	
RNDP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	存储随机数的软元件	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	○	○	○	○	○	—	—	○	—	—	—	—

功能

发生0~32767的随机数，存储到(d)中指定的软元件中。RND(P)指令根据某个计算公式发生随机数。在计算公式中，将上次的计算结果作为系数使用。

出错

没有运算错误。

SRND (P)

按照指定的软元件中存储的BIN16位数据的内容更改随机数系列。

梯形图	ST
	ENO:=SRND (EN, s) ; ENO:=SRNDP (EN, s) ;

FBD/LD

■执行条件

指令	执行条件
SRND	
SRNDP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	随机数系列数据	-32768~32767	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

按照 (s) 中指定的软元件中存储的BIN16位数据的内容，更改随机数系列。SRND (P) 指令可以更改随机数的计算模式。

出错

没有运算错误。

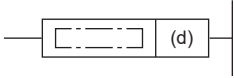
19 软元件操作

19.1 变址寄存器指令

变址寄存器的批量保存

ZPUSH(P)

将变址寄存器的内容保存到指定的区域中。

梯形图	ST
	ENO:=ZPUSH(EN, d); ENO:=ZPUSHP(EN, d);

FBD/LD



■执行条件

指令	执行条件
ZPUSH	
ZPUSHP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	保存变址寄存器的起始软元件	—	带符号BIN16位	ANY16* ¹
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	—	—	○	—	—	—	○	—	—	—	—	—

功能

- 将变址寄存器、长变址寄存器、局部变址寄存器、局部长变址寄存器的内容保存到(d)中指定的软元件或数组型标签的要素及其以后。保存的局部变址寄存器、局部长变址寄存器为执行了ZPUSH指令的程序局部变址寄存器、局部长变址寄存器。
- 保存变址寄存器的内容时，保存次数(d)将被+1。
- 与点数的分配无关，
- 保存变址寄存器、长变址寄存器、局部变址寄存器、局部长变址寄存器的48个字。
- 数据的恢复时，使用ZPOP(P)指令。通过成对使用ZPUSH(P)指令及ZPOP(P)指令，设置嵌套结构可以作为堆栈使用。

☞ 921页 ZPOP(P)

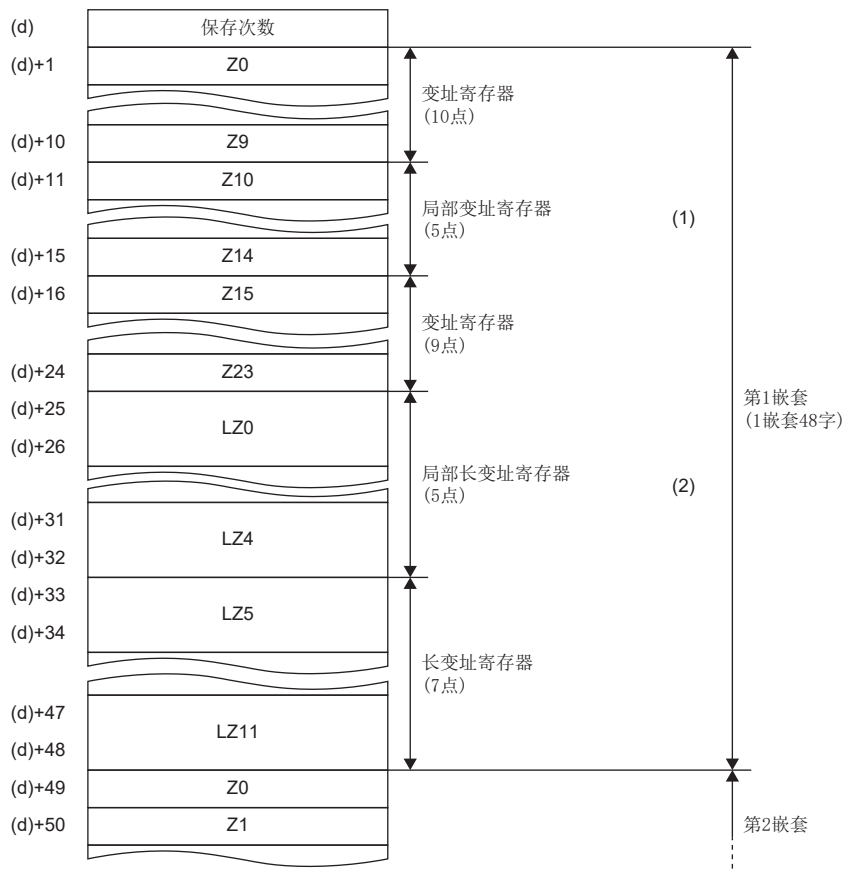
- 通过将ZPUSH(P)指令的(d)中指定的区域，再次指定到ZPOP(P)指令的(d)中，可以创建嵌套结构。每次执行ZPUSH(P)指令时，保存次数(d)将被+1。
- 嵌套结构时，将其它区域指定到ZPUSH(P)指令的(d)中的情况下，将变址寄存器的内容保存到指定的其它区域中。
- 设置了嵌套结构的情况下，每次执行ZPUSH(P)指令时将添加保存的数据，因此应预先预留出相当于使用次数的区域。

例

在工程工具中如下设置了变址寄存器设置的局部设置时，(d)及以后使用的区域配置如下所示。

☞ [CPU Parameter (CPU参数)] ⇒ [Memory/Device Setting (存储器/软元件设置)] ⇒ [Index Register Setting (变址寄存器设置)]

项目			设定值
局部设置	点数设置	局部变址寄存器 (Z)	5点
		局部长变址寄存器 (LZ)	5点
	起始	变址寄存器 (Z)	10
		长变址寄存器 (LZ)	0



- (1) 保存从Z10开始5点的局部变址寄存器。
- (2) 保存从LZ0开始5点的局部长变址寄存器。

出错

错误代码 (SD0)	内容
3285H	(d)+0的内容(保存次数)为FFFF时。

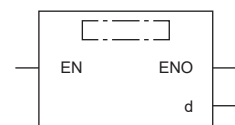
变址寄存器的批量恢复

ZPOP (P)

将指定区域中保存的数据读取到变址寄存器中。

梯形图	ST
	ENO:=ZPOP (EN, d) ; ENO:=ZPOPP (EN, d) ;

FBD/LD



■执行条件

指令	执行条件
ZPOP	
ZPOPP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	进行变址寄存器恢复的起始软元件	—	带符号BIN16位	ANY16*1
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	—	—	○	—	—	—	○	—	—	—	—

功能

- 将(d)中指定的软元件或数组型标签的要素及其以后保存的数据，读取到变址寄存器、长变址寄存器、局部变址寄存器、局部长变址寄存器中。
- 保存的变址寄存器的内容被读取时，保存次数(d)将被-1。
- 恢复的局部变址寄存器、局部长变址寄存器为执行了ZPOP指令的程序的局部变址寄存器、局部长变址寄存器。
- 关于(d)以后使用的区域构成，请参阅下述内容。

☞ 918页 ZPUSH(P)

出错

错误代码(SD0)	内容
3285H	(d)+0的内容(保存次数)为0时。

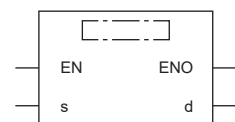
变址寄存器/长变址寄存器选择保存

ZPUSH(P)

将变址寄存器、长变址寄存器的内容保存到指定的区域中。

梯形图	ST
	ENO:=ZPUSH_2(EN, s, d); ENO:=ZPUSH_2(EN, s, d);

FBD/LD



(□中放入ZPUSH_2、ZPUSH_2。)

■执行条件

指令	执行条件
ZPUSH	
ZPUSHP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	保存的变址寄存器、长变址寄存器类型	1~3	无符号BIN16位	ANY16
(d)	保存变址寄存器、长变址寄存器的起始软元件	—	带符号BIN16位	ANY16*1
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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 只能使用U3E□\G□。

功能

- 将(s)中指定的要保存的变址寄存器、长变址寄存器、局部变址寄存器、局部长变址寄存器的内容保存到(d)中指定的软元件或数组型标签的要素及其以后。
- 当保存变址寄存器、长变址寄存器、局部变址寄存器、局部长变址寄存器的内容时，保存次数(d)将+1。
- 保存的局部变址寄存器、局部长变址寄存器为执行了ZPUSH指令的程序局部变址寄存器、局部长变址寄存器。
- 保存的数据的末尾处将存储保存的变址寄存器、长变址寄存器、局部变址寄存器、局部长变址寄存器的类型。
- (s)中指定的值及要保存的变址寄存器、长变址寄存器、局部变址寄存器、局部长变址寄存器如下所示。

(s)的值	保存的Z、LZ
1	Z、LZ全范围
2	Z全范围
3	LZ全范围

- 数据的恢复时，使用变址寄存器/长变址寄存器选择恢复指令(ZPOP(P)指令)。通过成对使用变址寄存器/长变址寄存器选择保存指令(ZPUSH(P)指令)与变址寄存器/长变址寄存器选择恢复指令(ZPOP(P)指令)，设置嵌套结构，可以作为堆栈使用。

☞ 926页 ZPOP(P)

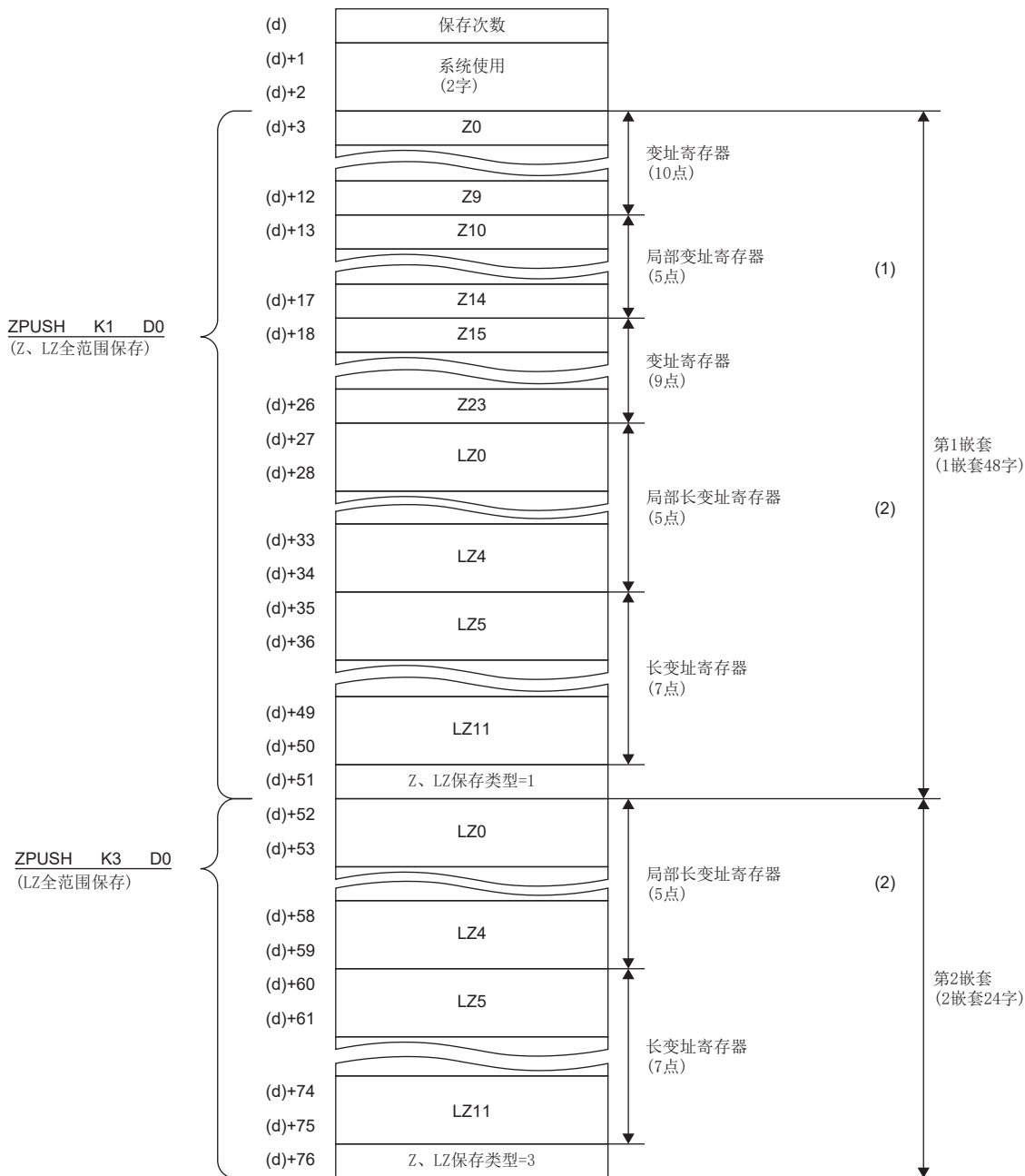
- 通过将变址寄存器/长变址寄存器选择保存指令(ZPUSH(P)指令)的(d)中指定的区域，再次指定到变址寄存器/长变址寄存器选择保存指令(ZPUSH(P)指令)的(d)中，可以创建嵌套结构。每次执行变址寄存器/长变址寄存器选择保存指令(ZPUSH(P)指令)时，保存次数(d)将被+1。
- 嵌套结构时，将其它区域指定到变址寄存器/长变址寄存器选择保存指令(ZPUSH(P)指令)的(d)中的情况下，将变址寄存器、长变址寄存器的内容保存到指定的其它区域中。
- 设置了嵌套结构的情况下，每次执行变址寄存器/长变址寄存器选择保存指令(ZPUSH(P)指令)时将添加保存的数据，因此应通过SD300、SD302确认变址寄存器、长变址寄存器点数，预先预留出相当于使用次数的区域。

例

在工程工具中如下设置了变址寄存器设置的局部设置时，(d)及以后使用的区域配置如下所示。

[CPU Parameter (CPU参数)] ⇒ [Memory/Device Setting (存储器/软元件设置)] ⇒ [Index Register Setting (变址寄存器设置)]

项目		设定值	
局部设置	点数设置	局部变址寄存器 (Z)	5点
		局部长变址寄存器 (LZ)	5点
	起始	变址寄存器 (Z)	10
		长变址寄存器 (LZ)	0



- (1) 保存从Z10开始5点的局部变址寄存器。
- (2) 保存从LZ0开始5点的局部长变址寄存器。

注意事项

ZPUSH(P) 指令的 (d)+1、(d)+2 为系统所用，因此请勿进行值的更改。
此外，(d) 中指定的区域内存储的 Z、LZ 保存类别也为系统所使用，因此请勿进行值的更改。
否则可能导致误动作。

出错

错误代码 (SD0)	内容
3285H	(s) 输入了超出允许指定范围的数据时。 指定的值为 1~3 以外
	(d) 中指定的区域内系统所用区域中存储的值被改写时。
	保存次数 (d) 为 FFFF 时。

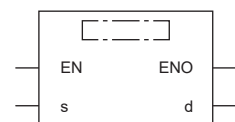
变址寄存器/长变址寄存器选择恢复

ZPOP (P)

将指定的区域中保存的数据，读取到变址寄存器、长变址寄存器中。

梯形图	ST
	ENO:=ZPOP_2(EN, s, d); ENO:=ZPOPP_2(EN, s, d);

FBD/LD



(□中放入ZPOP_2、ZPOPP_2。)

■执行条件

指令	执行条件
ZPOP	
ZPOPP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	虚拟	—	无符号BIN16位	ANY16
(d)	进行变址寄存器恢复的起始软元件	—	带符号BIN16位	ANY16*1
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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 只能使用U3E□\G□。

功能

- 将(d)中指定的软元件或数组型标签的要素及其以后保存的数据，读取到变址寄存器、长变址寄存器、局部变址寄存器、局部长变址寄存器中。
- 对保存的变址寄存器、长变址寄存器的内容进行读取时，保存次数(d)将被-1。
- 恢复的局部变址寄存器、局部长变址寄存器为执行了ZPOP指令的程序的局部变址寄存器、局部长变址寄存器。
- (s)中指定的数据被作为虚拟数据处理，忽略。
- 关于(d)以后使用的区域构成，请参阅下述内容。

☞ 922页 ZPUSH(P)

注意事项

(d)+1、(d)+2由系统使用，因此请勿进行值的更改。

此外，(d)中指定的区域中存储的Z、LZ保存类别也为系统所使用，因此请勿进行值的更改。否则可能导致误动作。

出错

错误代码 (SD0)	内容
3285H	(d)+0的保存次数为0时。
	(d)中指定的区域内系统所用区域中存储的值被改写时。
	Z、LZ保存类型中设置了1~3以外的值时。

19.2 文件寄存器操作指令

文件寄存器的块No. 切换

RSET (P)

更改程序中使用的文件寄存器的块No.。

梯形图	ST
	ENO:=RSET (EN, s) ; ENO:=RSETP (EN, s) ;

FBD/LD

■执行条件

指令	执行条件
RSET	
RSETP	

设置数据

■内容、范围、数据类型

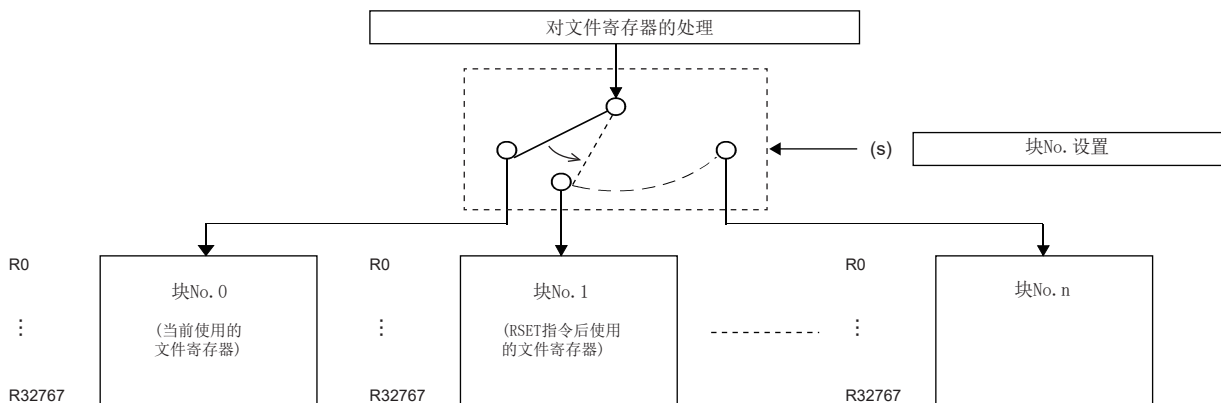
操作数	内容	范围	数据类型	数据类型(标签)
(s)	切换块No. 数据或存储了块No. 数据的软元件编号	0~32767	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	○	○	○	○	○	—	—	○	○	—	—	—

功能

将程序中使用的文件寄存器的块No. 更改为 (s) 中指定的软元件中存储的块No.。更改块No. 后，顺控程序中使用的所有文件寄存器将被处理为以更改后块No. 的文件寄存器为对象。



注意事项

通过工程工具将文件寄存器的“块号保存/回归设置”设置为“不保存/不回归”时，请勿在中断程序内通过本指令切换文件寄存器的块号。切换时，从中断程序恢复后不会立即反映，在下一个程序开始之前块号有可能不会切换。

关于文件寄存器的限制事项，请参阅下述内容。

☞ 57页 使用文件寄存器时的限制事项

出错

错误代码 (SD0)	内容
2820H	文件寄存器不存在时。
3285H	(s) 中指定的块No. 不存在时。

19.3 文件寄存器的1字节单位的读取/写入指令

文件寄存器的1字节数据读取

ZRRDB(P)

对指定的连号字节No. 的文件寄存器的内容进行读取。

梯形图	ST
	ENO:=ZRRDB(EN, s, d); ENO:=ZRRDBP(EN, s, d);

FBD/LD

■执行条件

指令	执行条件
ZRRDB	
ZRRDBP	

设置数据

■内容、范围、数据类型

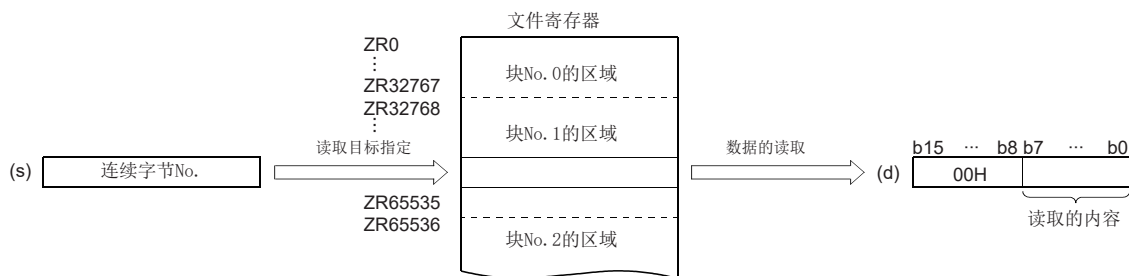
操作数	内容	范围	数据类型	数据类型(标签)
(s)	读取的文件寄存器的连号字节No.	0~4294967295	无符号BIN32位	ANY32
(d)	存储读取的数据的软元件编号	—	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

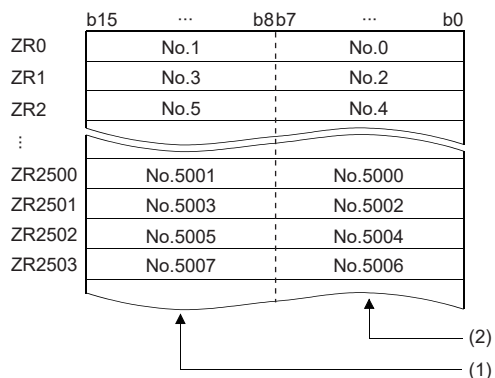
操作数	位		字		双字		间接指定	常数			其它	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K	H		E
(s)	○	○	○	○	○	○	○	○	—	—	—	—
(d)	○	○	○	○	○	—	—	○	—	—	—	—

功能

- 在不理会块No. 的状况下，读取 (s) 中指定的连号字节No. 的文件寄存器的内容，存储到 (d) 中指定的软元件的低位8位中。(d) 中指定的高位8位将变为00H。



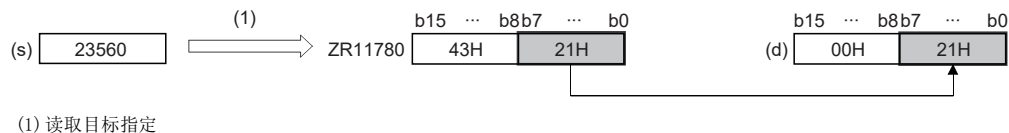
- 连号字节No. 对应的文件寄存器的编号如下所示。



- (1) 奇数编号指定时的数据
- (2) 偶数编号指定时的数据

例

指定了 (s)=23560 的情况下，读取 ZR11780 的低位8位的数据。



- (1) 读取目标指定

例

指定了 (s)=43257 的情况下，读取 ZR21628 的高位8位的数据。



- (1) 读取目标指定

出错

错误代码 (SD0)	内容
2820H	指定了超出允许指定范围的软元件编号 (连号字节No.) 时。

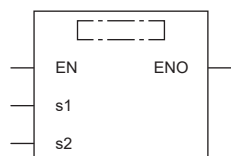
文件寄存器的1字节数据写入

ZRWRB(P)

将指定的软元件的低位的内容写入到连号字节No. 的文件寄存器中。

梯形图	ST
	ENO:=ZRWRB(EN, s1, s2); ENO:=ZRWRBP(EN, s1, s2);

FBD/LD



■执行条件

指令	执行条件
ZRWRB	
ZRWRBP	

设置数据

■内容、范围、数据类型

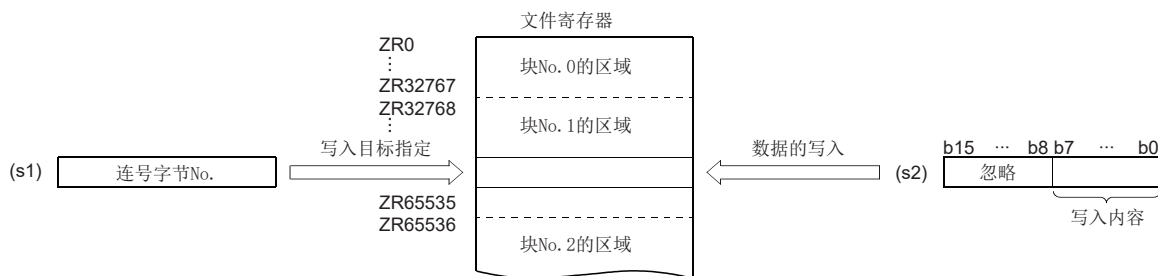
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	写入的文件寄存器的连号字节No.	0~4294967295	无符号BIN32位	ANY32
(s2)	存储了写入数据的软元件编号	-32768~32767	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

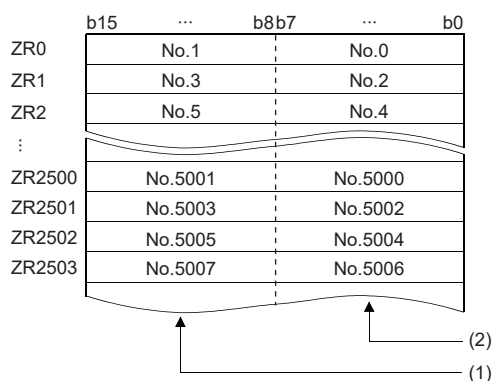
操作数	位		字		双字		间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC		LZ	K、H	E	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	—	—	○	—	—	—

功能

- 在不理会块No. 的状况下，将(s2)中指定的软元件中存储的低位8位的内容，写入到(s1)中指定的连号字节No. 的文件寄存器中。(s2)中指定的软元件的高位8位的数据将被忽略。



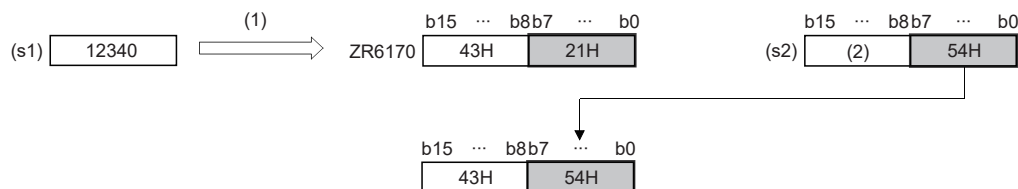
- 连号字节No. 对应的文件寄存器的编号如下所示。



- (1) 奇数编号指定时的数据
- (2) 偶数编号指定时的数据

例

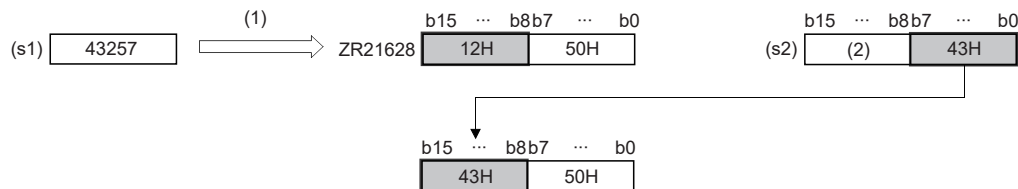
指定了(s1)=12340的情况下，写入到ZR11170的低位8位中。



- (1) 写入目标指定
- (2) 忽略

例

指定了(s1)=43257的情况下，写入到ZR21628的高位8位中。



- (1) 写入目标指定
- (2) 忽略

出错

错误代码(SD0)	内容
2820H	指定了超出允许指定范围的软元件编号(连号字节No.)时。

19.4 间接地址读取指令

间接地址读取

ADRSET (P)

读取指定的软元件的间接地址。

梯形图	ST
	<pre>ENO:=ADRSET (EN, s, d); ENO:=ADRSETP (EN, s, d); ENO:=_ADRSET (EN, s, d);</pre>

FBD/LD

■执行条件

指令	执行条件
ADRSET	
ADRSETP	

设置数据

■内容、范围、数据类型

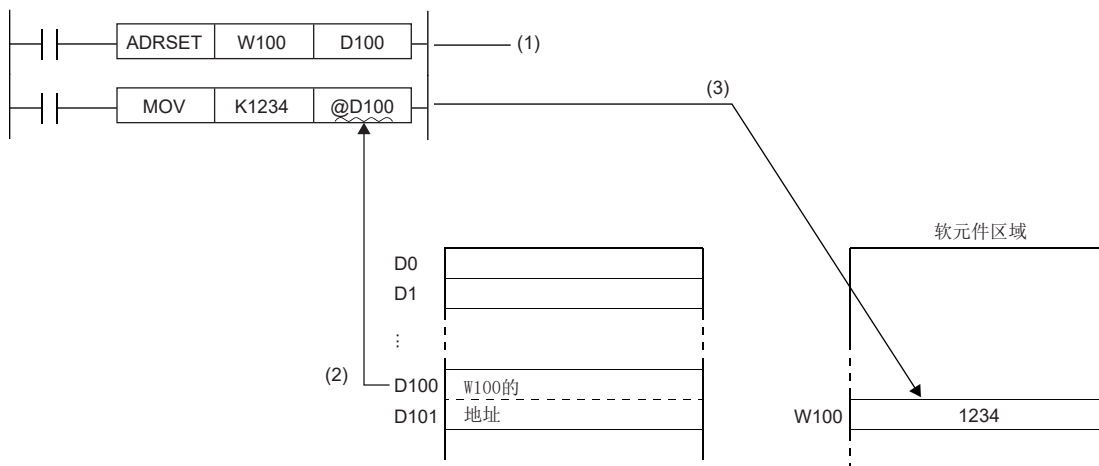
操作数	内容	范围	数据类型	数据类型(标签)
(s)	读取间接地址的软元件编号	—	软元件名	ANY_ELEMENTARY
(d)	存储(s)中指定的软元件的间接地址的软元件编号	—	带符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	○	—	○	—	—	—	—	○	—	—	—

功能

- 将(s)中指定的软元件的间接地址，存储到(d)中指定的软元件中。(d)+0、(d)+1的软元件中存储的地址，在通过程序进行软元件的间接地址指定的情况下使用。



- (1) 将W100的地址存储到D100、D101中。
- (2) 指定D100、D101的内容(W100的地址)。
- (3) 将1234写入到W100中。

- (s)中不能进行位软元件的位指定、字软元件的位指定。
- 关于软元件的间接指定，请参阅下述内容。

📖所使用的控制器的用户手册

- 不支持标签的间接指定。在(d)中指定标签使用的情况下，应通过以下方法使用。

标签	内容
指定了软元件的全局标签的情况下	作为间接地址使用的情况下，应使用全局标签中指定的软元件的间接指定。 ■ST的示例 <pre>ADRSET (TRUE, intVar, gvAddr); // 将intVar的间接地址读取到gvAddr。 INC (TRUE, @D0); // 通过指定至gvAddr的软元件D0的间接指定使用。</pre>
自动分配的全局标签/局部标签的情况下	将间接地址传送至软元件，使用传送目标软元件的间接指定。 ■ST的示例 <pre>ADRSET (TRUE, intVar, lvAddr); // 将intVar的间接地址读取到lvAddr。 DMOV (TRUE, lvAddr, D0); // 将读取到lvAddr中的间接地址传送至软元件。 INC (TRUE, @D0); // 通过传送的软元件的间接指定使用。</pre>

出错

没有运算错误。

20 定时器、计数器

20.1 特殊计数器指令

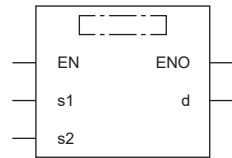
单相输入升值/降值计数器

UDCNT1

更新指定的计数器的当前值。

梯形图	ST
	ENO:=UDCNT1(EN, s1, s2, d);

FBD/LD



■执行条件

指令	执行条件
UDCNT1	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	(s1)+0: 计数输入用的输入编号 (s1)+1: 升值/降值设置用 OFF的情况下, 升值计数(对当前值进行递增计数) ON的情况下, 降值计数(向当前值减少方向计数)	—	位	ANYBIT_ARRAY*1 (要素数: 2)
(d)	通过UDCNT1指令进行计数的计数器编号(软元件名)	—	软元件名	ANY16*2
(s2)	设定值	-32768~32767	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能使用软元件(X)中分配的标签。

*2 只能使用软元件(C)中分配的标签。

■可用软元件

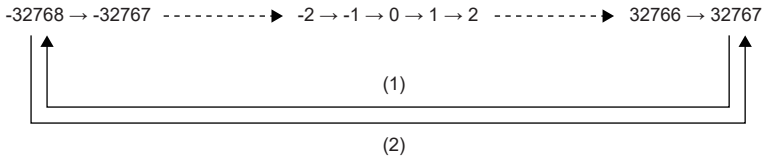
操作数	位		字		双字			间接指定	常数				其它	
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K	H	E	\$		
(s1)	○*1	—	—	—	—	—	—	—	—	—	—	—	—	—
(d)	—	—	○*2	—	—	—	—	—	—	—	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—	—	—

*1 只能使用X。但是, 只能在输入输出点数(可访问输入输出模块的点数)的范围内使用。

*2 只能使用C。

功能

- (s1)中指定的输入的OFF→ON时，对(d)中指定的计数器的当前值进行更新。
- 计数方向取决于(s1)+1中指定的输入的ON/OFF。
 - OFF：升值计数(对当前值进行递增计数)
 - ON：降值计数(向当前值减少方向计数)
- 计数处理按以下方式进行。
 - 升值计数中，当前值等于(s2)中指定的设定值时，(d)中指定的计数器的触点将变为ON。但是，即使(d)中指定的计数器的触点变为ON，当前值的计数仍将继续。
 - 降值计数中，当前值为设定值-1时，(d)中指定的计数器的触点将变为OFF。
 - (d)中指定的计数器是环形计数器。当前值为32767时如果进行升值计数，当前值将变为-32768。此外，当前值为-32768时如果进行降值计数，当前值将变为32767。当前值的计数处理内容如下所示。

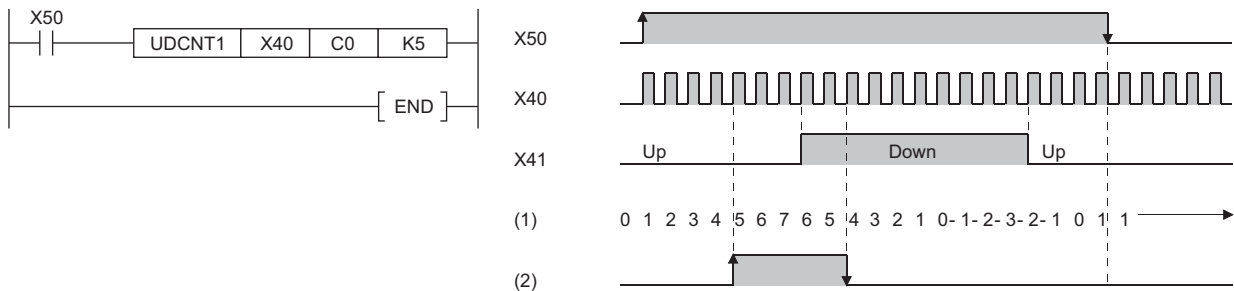


- (1) 升值计数时
(2) 降值计数时

- 使用了UDCNT1指令的计数处理的动作如下所示。

例

对X50为ON之后的X40的OFF→ON次数通过C0(升值/降值计数器)进行计数的程序及动作如下所示。



- (1) C0当前值
(2) C0触点

- 通过UDCNT1指令进行计数处理时，执行指令的OFF→ON时开始计数，ON→OFF时中止计数。再次将执行指令置为OFF→ON时，将从中止时的当前值开始再次开始计数。
- (d)中指定的计数器的当前值的清除及触点的OFF是通过RST指令执行。

要点

- UDCNT1指令将自变量的软元件数据登录到控制器的工作区域中后，通过系统中断处理实际的计数动作。(控制器的工作区域中登录的软元件数据，通过将执行指令置为OFF或，STOP→RUN进行清除。)因此，可计数的脉冲需要有比控制器的中断间隔长的ON及OFF时间。控制器的中断间隔为1ms。
- 通过UDCNT1指令进行的计数中(执行指令为ON中)，不能进行设定值更改。进行设定值更改的情况下，应将执行指令置为OFF之后再行更改。
- UDCNT1指令中指定的计数器不能用于其它指令。在其它指令中使用的情况下将无法计数。
- UDCNT1指令在正在执行的全部程序中最多可以使用6次。从第7次以后将变为无处理。

出错

没有运算错误。

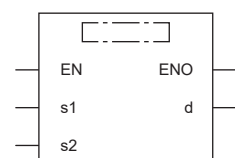
2相输入升值/降值计数器

UDCNT2

根据指定的A相脉冲及B相脉冲的状态，对计数器的当前值进行更新。

梯形图	ST
	ENO:=UDCNT2(EN, s1, s2, d);

FBD/LD



■执行条件

指令	执行条件
UDCNT2	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	(s1)+0: 计数输入用的输入编号(A相脉冲) (s1)+1: 计数输入用的输入编号(B相脉冲)	—	位	ANYBIT_ARRAY* ¹ (要素数: 2)
(d)	通过UDCNT2指令进行计数的计数器编号(软元件名)	—	软元件名	ANY16* ²
(s2)	设定值	-32768~32767	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能使用软元件(X)中分配的标签。

*2 只能使用软元件(C)中分配的标签。

■可用软元件

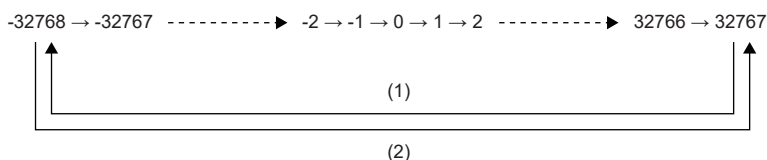
操作数	位		字			双字		间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K、H	E	\$	
(s1)	○* ¹	—	—	—	—	—	—	—	—	—	—	—
(d)	—	—	○* ²	—	—	—	—	—	—	—	—	—
(s2)	○	○	○	○	○	—	—	○	○	—	—	—

*1 只能使用X。但是，只能在输入输出点数(可访问输入输出模块的点数)的范围内使用。

*2 只能使用C。

功能

- 根据(s1)中指定的输入(A相脉冲)及(s1)+1中指定的输入(B相脉冲)的状态,对(d)中指定的计数器的当前值进行更新。
- 计数方向按下述方式确定。
 - (s1)为ON时(s1)+1由OFF→ON的情况下,进行升值计数(对当前值进行递增计数)。
 - (s1)为ON时(s1)+1由ON→OFF的情况下,进行降值计数(向当前值减少方向计数)。
 - (s1)为OFF时不进行计数。
- 计数处理按以下方式进行。
 - 升值计数中,当前值等于(s2)中指定的设定值时,(d)中指定的计数器的触点将变为ON。但是,即使(d)中指定的计数器的触点变为ON,当前值的计数仍将继续。
 - 降值计数中,当前值为设定值-1时,(d)中指定的计数器的触点将变为OFF。
 - (d)中指定的计数器是环形计数器。当前值为32767时如果进行升值计数,当前值将变为-32768。此外,当前值为-32768时如果进行降值计数,当前值将变为32767。当前值的计数处理内容如下所示。

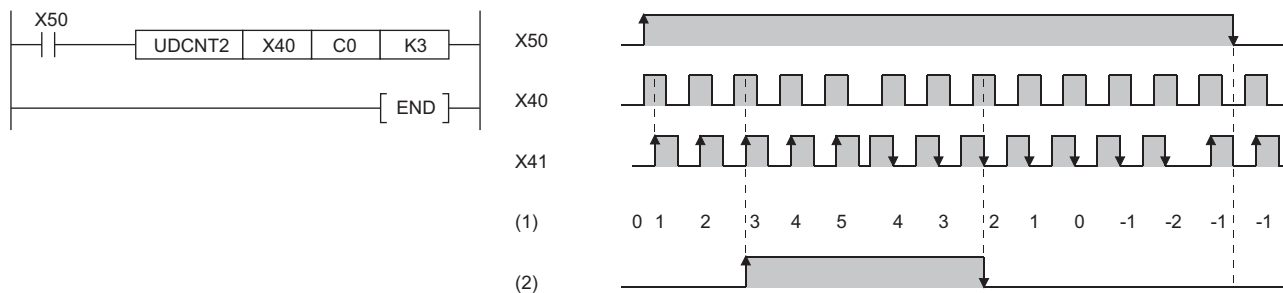


- (1) 升值计数时
(2) 降值计数时

- 使用了UDCNT2指令的计数处理的动作如下所示。

例

对X50为ON之后的X40、X41的状态通过C0(升值/降值计数器)进行计数的程序及动作如下所示。



- (1) C0当前值
(2) C0触点

- 通过UDCNT2指令进行计数处理时,执行指令的OFF→ON时开始计数,ON→OFF时中止计数。再次将执行指令置为OFF→ON时,将从中止时的当前值开始再次开始计数。
- (d)中指定的计数器的当前值的清除及触点的OFF是通过RST指令执行。

要点

- UDCNT2指令将自变量的软元件数据登录到控制器的工作区域中后,通过系统中断处理实际的计数动作。(控制器的工作区域中登录的软元件数据,通过将执行指令置为OFF或,STOP→RUN进行清除。)因此,可计数的脉冲需要有比控制器的中断间隔长的ON及OFF时间。控制器的中断间隔为1ms。
- 通过UDCNT2指令进行的计数中(执行指令为ON中),不能进行设定值的更改。进行设定值更改的情况下,应将执行指令置为OFF之后再行更改。
- UDCNT2指令中指定的计数器不能用于其它指令。在其它指令中使用的情况下将无法计数。
- UDCNT2指令在正在执行的全部程序中最多可以使用5次。从第6次以后将变为无处理。

出错

没有运算错误。

20.2 特殊定时器指令

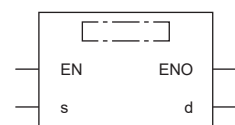
示教定时器

TMR

将计测指令的ON中的时间以秒单位进行测定，对乘以乘数后的值进行存储。

梯形图	ST
	ENO:=TMR (EN, s, d);

FBD/LD



■执行条件

指令	执行条件
TMR	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	(d)+0: 测定值存储软元件 (d)+1: 控制器的系统用	—	带符号BIN16位	ANY16_ARRAY (要素数: 2)
(s)	测定值的乘数	0~2	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	—	○	○	○	○	—	○	○	—	—	—

■控制数据

操作数: (d)			
软元件	内容	设置范围	设置方
+0	测定值存储软元件	—	系统
+1	控制器的系统用	—	系统

功能

- 将执行指令为ON的时间以秒单位进行测定，将乘以(s)中指定的乘数后的值存储到(d)中指定的软元件中。
- 执行指令由OFF→ON时，(d)+0、(d)+1中指定的软元件将被清除。
- (s)中可指定的乘数如下所示。

(s)	乘数
0	1
1	10
2	100

- (s)中指定的值为0~2以外的情况下，将变为无处理。

要点

- 执行TTMR指令时，进行时间计测。请勿通过JMP指令等跳过TTMR指令。跳过的情况下，将无法正确进行测定。
- TTMR指令执行中，请勿更改(s)中指定的乘数。更改了(s)中指定的乘数的情况下，将无法得出正确的值。
- (d)+1中指定的软元件是由控制器的系统所使用，因此请勿对其进行值的更改。进行了值的更改的情况下，(d)中指定的软元件中存储的值将不是正确的值。

出错

没有运算错误。

特殊功能定时器

STMR

进行下述4种类型的定时器输出。

- OFF延迟定时器输出
- OFF后单次触发定时器输出
- ON后单次触发定时器输出
- ON延迟+OFF延迟定时器输出

梯形图	ST
	ENO:=STMR (EN, s1, s2, d);

FBD/LD

■执行条件

指令	执行条件
STMR	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	定时器软元件或定时器型标签	—	软元件名	ANY16
(s2)	设定值	0~32767	带符号BIN16位	ANY16
(d)	(d)+0: OFF延迟定时器输出	—	位	ANYBIT_ARRAY (要素数: 4)
	(d)+1: OFF后单次触发定时器输出			
	(d)+2: ON后单次触发定时器输出			
	(d)+3: ON延迟+OFF延迟定时器输出			
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数			其它
	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)	○	—	—	—	—	—	—	—	—	—	—	—

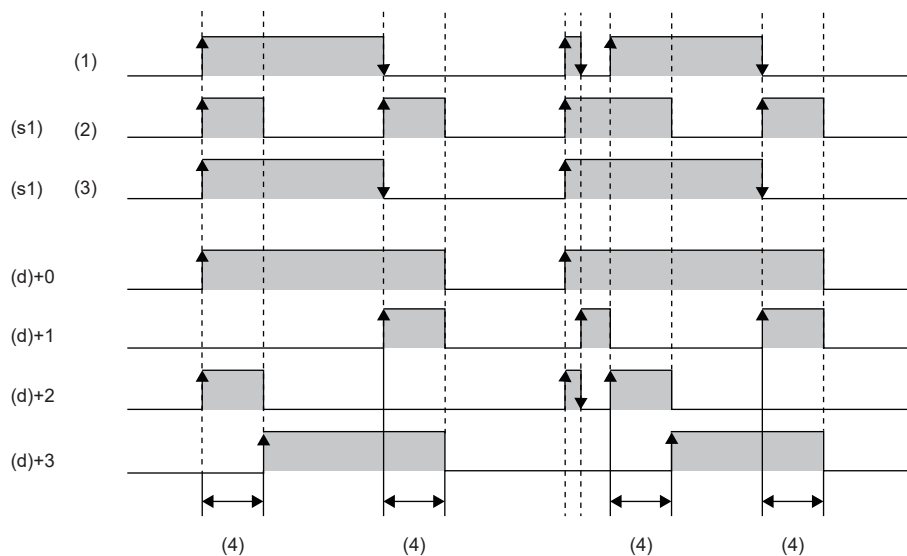
*1 只能使用T。

功能

- 使用(d)中指定的软元件开始的4点, 进行4种类型的定时器输出。

设置数据	内容	
(d)	+0	OFF延迟定时器输出: 通过STMR指令的指令的上升沿变为ON, 指令的下降沿后经过(s2)中指定的时间时将变为OFF。
	+1	OFF后单次触发定时器输出: 通过STMR指令的指令的下降沿变为ON, 经过(s2)中指定的时间时变为OFF。
	+2	ON后单次触发定时器输出: 通过软元件名STMR指令的指令的上升沿变为ON, 经过(s2)中指定的时间或STMR指令的指令为OFF时将变为OFF。
	+3	ON延迟+OFF延迟定时器输出: 通过软元件名定时器线圈的下降沿变为ON, STMR指令的指令的下降沿后, 经过(s2)中指定的时间时将变为OFF。

- (s1)中指定的定时器的线圈, 通过STMR指令的指令的上升沿及下降沿变为ON, 开始当前值的计测。
- 定时器的线圈继续计测, 直到(s2)中指定的设定值为止, 如果时限到则变为OFF。
- 如果时限到之前STMR指令的指令变为OFF, 则定时器的线圈将保持ON状态不变。此时定时器的计测将继续进行。STMR指令的指令再次变为ON时, 将当前值置为0后, 重启计测。
- 定时器的触点通过STMR指令的指令的上升沿变为ON, 定时器的线圈变为下降沿后, 通过STMR指令的指令的下降沿变为OFF。定时器的触点是系统所用, 因此不能使用。



- (1) STMR指令的指令
(2) 线圈
(3) 触点
(4) (s2)中指定的设定值

- 通过STMR指令指定的定时器执行当前值计测时, 与STMR指令的指令的ON/OFF状态无关。通过JMP指令等跳过了STMR指令的情况下, 将无法进行计测。
- (d)中指定的定时器的计测单位与低速定时器的相同。
- (s2)的设定值可在0~32767的范围内指定。0~32767以外的情况下将变为无处理。
- (s1)中指定的定时器, 请勿使用OUT指令。STMR指令与OUT指令中使用了同一定时器软元件或定时器型标签的情况下, 将无法正常工作。

注意事项

RUN中的梯形图块更改或RUN中的可编程控制器写入的写入范围内存在STMR指令的情况下, STMR指令将动作。

关于详细内容, 请参阅下述手册。

📖 所使用的控制器的用户手册

出错

没有运算错误。

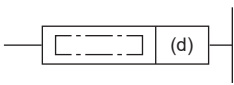
21 时钟

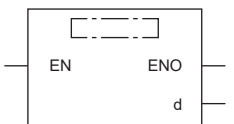
21.1 时钟用指令

时钟数据的读取


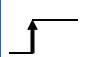
DATERD (P)

从控制器的时钟单元中读取“年、月、日、时、分、秒、星期”。

梯形图	ST
	ENO:=DATERD(EN, d); ENO:=DATERDP(EN, d);

FBD/LD


■执行条件

指令	执行条件
DATERD	
DATERDP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	存储读取的时钟数据的起始软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 7)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 从控制器的时钟单元中读取“年、月、日、时、分、秒、星期”，以BIN值存储到(d)中指定的软元件及其以后。

(数据)	(d)	(d)+1	(d)+2	(d)+3	(d)+4	(d)+5	(d)+6
(内容)	年(公历)	月	日	时(24小时计)	分	秒	星期
(设置范围)	1980~2079	1~12	1~31	0~23	0~59	0~59	0~6

- (d)的“年”以公历4位存储。
- (d)+6的“星期”，即“日~六”，以“0~6”存储。

星期	日	一	二	三	四	五	六
存储数据	0	1	2	3	4	5	6

- 闰年将被自动补偿。

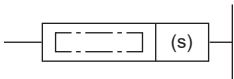
出错

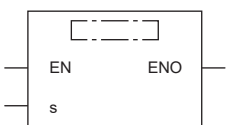
没有运算错误。

时钟数据的写入

DATEWR (P)

将指定的软元件及其以后存储的时钟数据，写入到控制器的时钟单元中。

梯形图	ST
	ENO:=DATEWR (EN, s) ; ENO:=DATEWRP (EN, s) ;

FBD/LD


■执行条件

指令	执行条件
DATEWR	
DATEWRP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s)	存储了写入时钟单元的时钟数据的起始软元件	功能参照	带符号BIN16位	ANY16_ARRAY (要素数: 7)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 将(s)中指定的软元件及其以后存储的时钟数据，写入到控制器的时钟单元中。

(数据)	(d)	(d)+1	(d)+2	(d)+3	(d)+4	(d)+5	(d)+6
(内容)	年(公历)	月	日	时(24小时计)	分	秒	星期
(设置范围)	1980~2079	1~12	1~31	0~23	0~59	0~59	0~6

- 各项目的设置以BIN值进行设置。
- (s)的“年”，以公历4位在1980~2079的范围内设置。
- (s)+1的“月”，在1~12(1月~12月)的范围内设置。
- (s)+2的“日”，在1~31(1日~31日)的范围内设置。
- (s)+3的“时”，在0~23(0时~23时)的范围内设置。(以24小时制进行设置。)
- (s)+4的“分”，在0~59(0分~59分)的范围内设置。
- (s)+5的“秒”，在0~59(0秒~59秒)的范围内设置。
- (s)+6的“星期”，即“日~六”，以“0~6”进行设置。

星期	日	一	二	三	四	五	六
存储数据	0	1	2	3	4	5	6

出错

错误代码(SD0)	内容
3285H	(s)~(s)+6的各项目中设置了超出设置范围的数据时。
3295H	(s)~(s)+6中设置了夏季时间开始时刻起不足1小时时。

要点

更改了时钟数据时，事件履历中将保存“时钟设置”（事件代码：24000）。因此，执行了本指令的情况下，事件履历中将保存“时钟设置”。

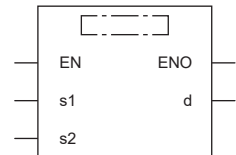
时钟数据的加法运算

DATE+ (P)

对时间数据进行加法运算。

梯形图	ST
	ENO:=DATEPLUS (EN, s1, s2, d) ; ENO:=DATEPLUSP (EN, s1, s2, d) ;

FBD/LD



(□中放入DATEPLUS、DATEPLUSP。)

■执行条件

指令	执行条件
DATE+	
DATE+P	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储了加法运算时间数据的起始软元件	功能参照	带符号BIN16位	ANY16_ARRAY (要素数: 3)
(s2)	存储了加法运算时间(时刻)数据的起始软元件	功能参照	带符号BIN16位	ANY16_ARRAY (要素数: 3)
(d)	存储加法运算结果时刻(时间)数据的起始软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 3)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 将(s1)中指定的时间数据与(s2)中指定的时间数据进行加法运算，将加法运算结果存储到(d)中指定的软元件编号及其以后。

(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

将6时32分40秒与7时48分10秒进行了加法运算的情况下

(s1)	6		(s2)	7		(d)	14
(s1)+1	32	+	(s2)+1	48		(d)+1	20
(s1)+2	40		(s2)+2	10		(d)+2	50

- 运算结果的时间超过了24时的情况下，减去了24小时后的值将成为运算结果。例如，将14时20分30秒与20时20分20秒进行了加法运算的情况下，其结果不是34时40分50秒，而是10时40分50秒。

(s1)	14		(s2)	20		(d)	10
(s1)+1	20	+	(s2)+1	20		(d)+1	40
(s1)+2	30		(s2)+2	20		(d)+2	50

出错

错误代码(SD0)	内容
3285H	(s1)、(s2)的数据超出范围时。

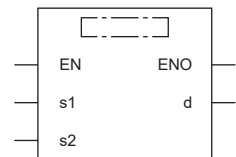
时钟数据的减法运算

DATE-(P)

对时间数据进行减法运算。

梯形图	ST
	ENO:=DATEMINUS(EN, s1, s2, d); ENO:=DATEMINUSP(EN, s1, s2, d);

FBD/LD



(□中放入DATEMINUS、DATEMINUSP。)

■执行条件

指令	执行条件
DATE-	
DATE-P	

设置数据

■内容、范围、数据类型

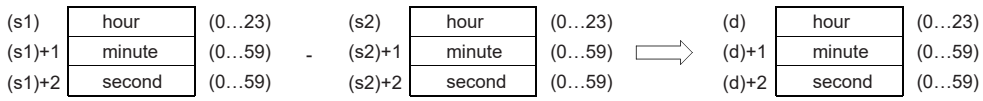
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储了被减的时间数据的起始软元件	功能参照	带符号BIN16位	ANY16_ARRAY (要素数: 3)
(s2)	存储了减法运算时间(时刻)数据的起始软元件	功能参照	带符号BIN16位	ANY16_ARRAY (要素数: 3)
(d)	存储减法运算结果时刻(时间)数据的起始软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 3)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—	—

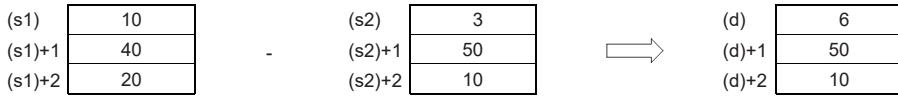
功能

- 将(s1)中指定的时间数据与(s2)中指定的时间数据进行减法运算，将减法运算结果存储到(d)中指定的软元件编号及其以后。

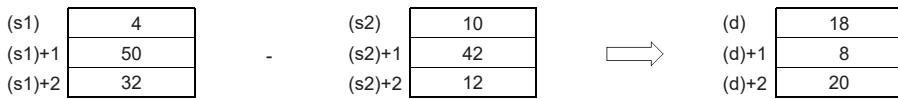


例

将10时40分20秒中与3时50分10秒进行减法运算的情况下



- 运算结果时间为负数的情况下，将该数据进行了+24的值将成为运算结果。例如，将4时50分32秒与10时42分12秒进行减法运算的情况下，其结果不是-6时8分20秒，而是18时8分20秒。



出错

错误代码(SD0)	内容
3285H	(s1)、(s2)的数据超出范围时。

时间数据的转换(时分秒→秒)

TIME2SEC(P)

将时间数据(时、分、秒)换算为秒数据。

梯形图	ST
	ENO:=TIME2SEC(EN, s, d); ENO:=TIME2SECP(EN, s, d);

FBD/LD

■执行条件

指令	执行条件
TIME2SEC	
TIME2SECP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储了转换前时钟数据的起始软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 3)
(d)	存储转换后时钟数据的起始软元件	—	带符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

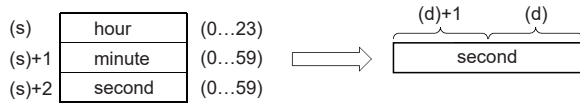
■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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不能使用。

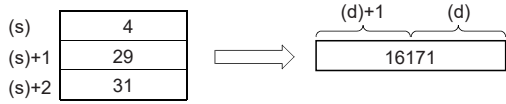
功能

将(s)中指定的软元件编号及其以后存储的时间数据，换算为秒并将换算结果存储到(d)中指定的软元件中。



例

(s)中指定了4小时29分31秒的情况下



出错

错误代码(SD0)	内容
3285H	(s)的数据超出范围时。

时间数据的转换(秒→时分秒)

SEC2TIME (P)

将秒数据换算为时间数据(时、分、秒)。

梯形图	ST
	ENO:=SEC2TIME (EN, s, d); ENO:=SEC2TIMEP (EN, s, d);

FBD/LD

■执行条件

指令	执行条件
SEC2TIME	
SEC2TIMEP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存储了转换前时钟数据的起始软元件	0~86399	带符号BIN32位	ANY32
(d)	存储转换后时钟数据的起始软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 3)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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不能使用。

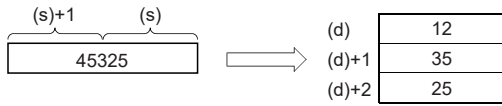
功能

将(s)中指定的软元件编号及其以后存储的秒数据，换算为时、分、秒，将换算结果存储到(d)中指定的软元件及其以后。



例

(s)中指定了45325秒的情况下



出错

错误代码(SD0)	内容
3285H	(s)的数据超出范围时。

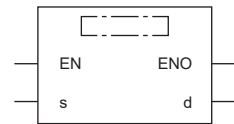
日期时间数据的转换(日期时间→秒)

DATE2SEC(P) (_U)

将日期时间数据换算为秒。

梯形图	ST
	<pre> ENO:=DATE2SEC(EN, s, d); ENO:=DATE2SECP(EN, s, d); ENO:=DATE2SEC_U(EN, s, d); ENO:=DATE2SECP_U(EN, s, d); </pre>

FBD/LD



■执行条件

指令	执行条件
DATE2SEC DATE2SEC_U	
DATE2SECP DATE2SECP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	存有更改前的日期时间数据的起始软元件	参阅功能详细内容。	带符号BIN16位	ANY16_ARRAY (要素数: 6)
(d)	DATE2SEC(P)	0~2145916799	带符号BIN32位	ANY32
	DATE2SEC(P)_U	0~3155759999	无符号BIN32位	ANY32_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

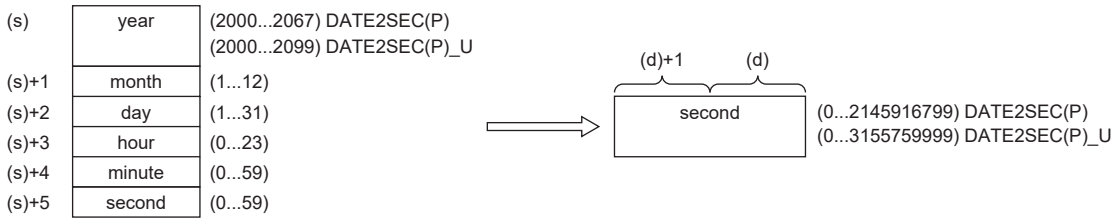
■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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不能使用。

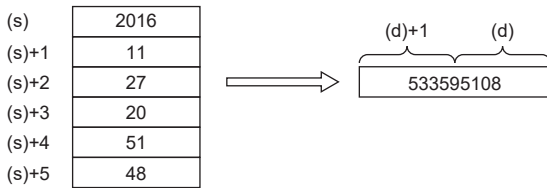
功能

- 将(s)中指定的软元件编号及其以后存储的日期时间数据换算为秒，并将换算结果存储到(d)中指定的软元件中。2000年1月1日0时0分0秒相当于起点(0秒)。



- 年的数据范围，在DATE2SEC(P)时为2000~2067，在DATE2SEC(P)_U时为2000~2099。

例如，指定了2016年11月27日20时51分48秒的情况下，将变为如下所示。



- 本指令的动作不受时钟数据的更改(包括时区、夏令时的时间校正)的影响。例如，夏令时的换算结果与夏令时外的换算结果相同。

出错

错误代码(SD0)	内容
3285H	(s)的数据超出范围时。 指定了实际上不存在的日期时间。(例：2016年2月30日)

日期时间数据的转换(秒→日期时间)

SEC2DATE(P) (_U)

将秒数据换算为日期时间。

梯形图	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

■执行条件

指令	执行条件
SEC2DATE SEC2DATE_U	
SEC2DATEP SEC2DATEP_U	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(s)	SEC2DATE(P)	存在更改前的秒数据的起始软元件	0~2145916799	带符号BIN32位	ANY32
	SEC2DATE(P)_U		0~3155759999	无符号BIN32位	ANY32_U
(d)		存储更改后的日期时间数据的起始软元件	参阅功能详细内容。	带符号BIN16位	ANY16_ARRAY (要素数: 7)
EN		执行条件	—	位	BOOL
ENO		执行结果	—	位	BOOL

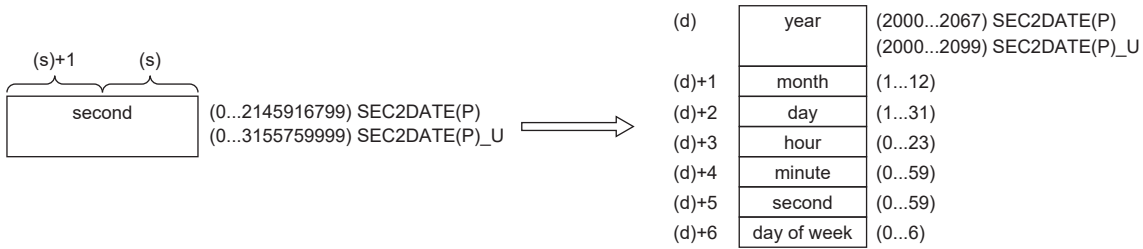
■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	—	—	○	—	—	—	—	○	—	—	—

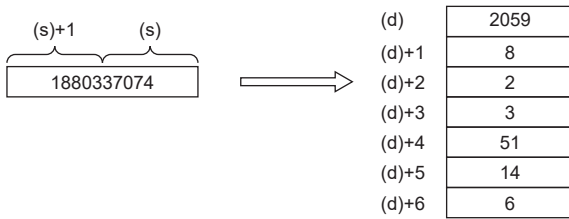
*1 FX、FY不能使用。

功能

- 将 (s) 中指定的软元件编号及其以后存储的秒数据换算为日期时间，并将换算结果存储到 (d) 中指定的软元件中。2000年1月1日0时0分0秒相当于起点(0秒)。



例如，指定了1880337074秒的情况下，将变为如下所示。



- (d)+6的星期，即“日~六”，以“0~6”存储。

星期	日	一	二	三	四	五	六
存储数据	0	1	2	3	4	5	6

- 本指令的动作不受时钟数据的更改(包括时区、夏令时的时间校正)的影响。例如，夏令时的换算结果与夏令时外的换算结果相同。

出错

错误代码(SD0)	内容
3285H	(s)的数据超出范围时。

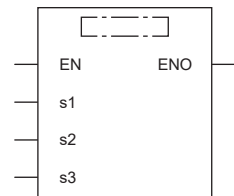
日期比较

LDDT□、ANDDT□、ORDT□

进行指定日期数据的比较，或将当前日期与日期数据进行比较。

梯形图	ST
<p>(□中放入DT=、DT<>、DT>、DT<=、DT<、DT>=。)</p>	<pre>ENO:=LDDT_□(EN, s1, s2, s3); ENO:=ANDDT_□(EN, s1, s2, s3); ENO:=ORDT_□(EN, s1, s2, s3); (□中为EQ、NE、GT、LE、LT、GE。)*1</pre>

FBD/LD



(□中放入LDDT_、ANDDT_、ORDT_与EQ、NE、GT、LE、LT、GE的组合。)*1

*1 EQ为=、NE为<>、GT为>、LE为<=、LT为<、GE为>=。

■执行条件

指令	执行条件
LDDT□、ANDDT□、ORDT□	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储了比较数据的起始软元件	—	带符号BIN16位	ANY_DT*1
(s2)	存储了比较数据的起始软元件	—	带符号BIN16位	ANY_DT*1
(s3)	表示比较对象的值或存储了比较对象的数据数	0001H~0007H、8001H~8007H	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

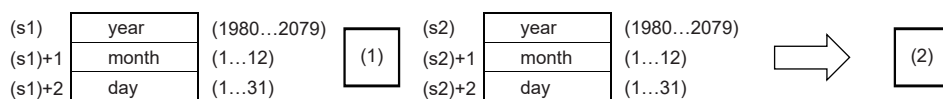
操作数	位		字		双字			间接指定	常数				其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	—
(s3)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

- 进行(s1)、(s2)中指定的日期数据的比较，或将当前的日期与(s1)中指定的日期数据进行比较。通过(s3)可以选择比较对象。

- 与任意日期数据的比较

将(s1)中指定的日期数据与(s2)中指定的日期数据的比较按照(s3)的条件进行常开触点处理。

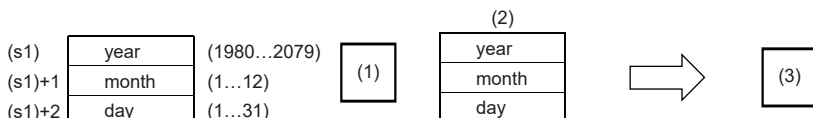


(1) 比较运算符

(2) 比较运算结果

- 与当前的日期数据的比较

将(s1)中指定的日期数据与当前的日期数据的比较按照(s3)的条件进行常开触点处理。(s2)中指定的日期数据被作为虚拟数据处理而忽略。



(1) 比较运算符

(2) 当前的日期数据

(3) 比较运算结果

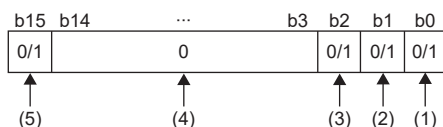
- 各项的设置以BIN值进行设置。

- (s1)、(s2)的“年”，以公历4位在1980~2079内设置。

- (s1)+1、(s2)+1的“月”，在1~12(1月~12月)内设置。

- (s1)+2、(s2)+2的“日”，在1~31(1日~31日)内设置。

- 通过在(s3)中指定下述值，可以对比较对象进行详细设置。(s3)的位构成如下所示。



(1) 在比较对象中添加“日”。

(2) 在比较对象中添加“月”。

(3) 在比较对象中添加“年”。

(4) 指定为0。指定为0以外时，与运算结果无关，将变为非导通。

(5) 将第15位为0N(1)时，将(s1)与当前的日期以0~第2位中指定的条件进行比较。

- 比较对象日期(0~第2位)为0的情况下，不进行比较对象的日期数据(年、月、日)的比较。为1的情况下，对比较对象的日期数据(年、月、日)进行比较。

- 比较运算对象(第15位)为0的情况下，进行(s1)中指定的日期数据与(s2)中指定的日期数据的比较。为1的情况下，进行(s1)中指定的日期数据与当前的日期数据的比较。(s2)中指定的日期数据将被忽略。

- 比较对象位的处理内容如下所示。

与任意日期数据比较时的(s3)值	与当前日期数据比较时的(s3)值	比较对象日期	处理内容
0001H	8001H	日	仅对(s1)+2进行比较。
0002H	8002H	月	仅对(s1)+1进行比较。
0003H	8003H	月、日	对(s1)+1、(s1)+2进行比较。
0004H	8004H	年	仅对(s1)进行比较。
0005H	8005H	年、日	对(s1)、(s1)+2进行比较。
0006H	8006H	年、月	对(s1)、(s1)+1进行比较。
0007H	8007H	年、月、日	对(s1)、(s1)+1、(s1)+2均进行比较。
0001H~0007H、8001H~8007H以外		无	对(s1)、(s1)+1、(s1)+2均不进行比较。(变为非导通。)

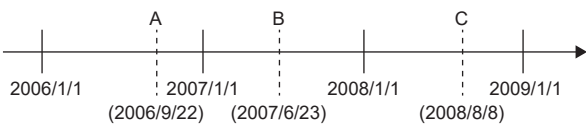
- 比较对象软元件中存储的数据不能被识别为日期数据的情况下，执行指令后SM709(DT、TM指令非法数据检测标志)将ON，变为非导通。即使不能被识别为日期数据的情况下，只要是在设置范围内，则SM709不变为ON。(s1)~(s1)+2或(s2)~(s2)+2超出相应软元件/标签的分配范围时，SM709也将为ON，变为非导通。一旦SM709变为ON，则在复位或电源OFF前，将一直保持ON状态，因此应根据需要将其置为OFF。

• 各指令的比较运算结果如下所示。

指令符号(梯形图、FBD/LD)	条件	比较运算结果
DT=、EQ	(s1)=(s2)	导通状态(ENO为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)	非导通状态(ENO为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各个日期的比较如下所示。



• 上述日期A、B、C的比较运算结果如下所示。即使以相同条件进行了比较的情况下，根据选择的比较对象其比较运算结果有所不同。

比较对象	比较条件*1		
	A<B	B<C	A<C
日	导通	非导通	非导通
月	非导通	导通	非导通
月、日	非导通	导通	非导通
年	导通	导通	导通
年、日	导通	导通	导通
年、月	导通	导通	导通
年、月、日	导通	导通	导通
无	非导通	非导通	非导通

*1 FBD/LD的情况下，导通时ENO显示为ON，非导通时ENO显示为OFF。

• 即使比较的日期不存在的情况下，只要是在设置范围内的日期，将按照下述条件进行比较运算。

- 日期A: 2006/02/30(不存在，但可进行日期的设置。)
- 日期B: 2007/03/29
- 日期C: 2008/02/31(不存在，但可进行日期的设置。)

比较对象	比较条件*2		
	A<B	B<C	A<C
日	非导通	导通	导通
月	导通	非导通	非导通
月、日	导通	非导通	导通
年	导通	导通	导通
年、日	导通	导通	导通
年、月	导通	导通	导通
年、月、日	导通	导通	导通
无	非导通	非导通	非导通

*2 FBD/LD的情况下，导通时ENO显示为ON，非导通时ENO显示为OFF。

- 通过FBD/LD使用了LDDT_□指令的情况下，应将EN指定为左母线，或始终ON的变量部件/常数部件。
- 将ORDT_□指令通过FBD/LD使用的情况下，将EN指定为TRUE时ENO将ON。EN不成为执行条件。

出错

没有运算错误。

时间比较

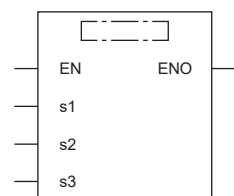
LDTM□、ANDTM□、ORTM□

进行指定的时间数据的比较，或进行指定的时间数据与当前时间的比较。

梯形图	ST
	<pre> ENO:=LDTM_□(EN, s1, s2, s3); ENO:=ANDTM_□(EN, s1, s2, s3); ENO:=ORTM_□(EN, s1, s2, s3); (□中为EQ、NE、GT、LE、LT、GE。)*1 </pre>

(□中放入TM=、TM<>、TM>、TM<=、TM<、TM>=。)

FBD/LD



(□中放入LDTM_、ANDTM_、ORTM_与EQ、NE、GT、LE、LT、GE的组合。)*1

*1 EQ为=、NE为<>、GT为>、LE为<=、LT为<、GE为>=。

■执行条件

指令	执行条件
LDTM□、ANDTM□、ORTM□	常时执行

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储了比较数据的起始软元件	—	带符号BIN16位	ANY_TM*1
(s2)	存储了比较数据的起始软元件	—	带符号BIN16位	ANY_TM*1
(s3)	表示比较对象的值或存储了比较对象的数据数	0001H~0007H、 8001H~8007H	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

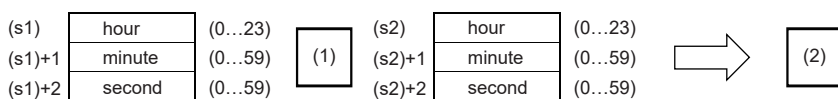
操作数	位		字		双字			间接指定	常数				其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	—	—	○	—	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	—
(s3)	○	○	○	○	○	—	—	○	○	—	—	—	—

功能

- 对(s1)、(s2)中指定的时间数据进行比较，或对(s1)中指定的时间数据与当前的时间进行比较。通过(s3)可以选择比较对象。

- 与任意时间数据的比较

将(s1)中指定的时间数据与(s2)中指定的时间数据的比较按照(s3)的条件进行常开触点处理。

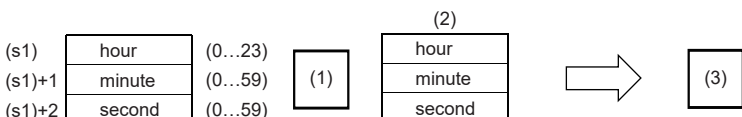


(1) 比较运算符

(2) 比较运算结果

- 与当前的时间数据的比较

将(s1)中指定的时间数据与当前的时间数据的比较按照(s3)的条件进行常开触点处理。(s2)中指定的时间数据将被作为虚拟数据处理而忽略。

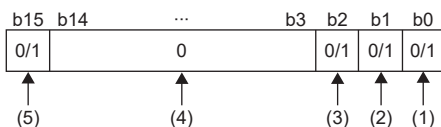


(1) 比较运算符

(2) 当前的时间数据

(3) 比较运算结果

- 各项的设置以BIN值进行设置。
- (s1)、(s2)的“时”，采24小时制，以0~23(0时~23时)进行设置。
- (s1)+1、(s2)+1的“分”，以0~59(0分~59分)进行设置。
- (s1)+2、(s2)+2的“秒”，以0~59(0秒~59秒)进行设置。
- 通过在(s3)中指定下述值，可以对比较对象进行详细设置。(s3)的位构成如下所示。



(1) 在比较对象中添加“秒”。

(2) 在比较对象中添加“分”。

(3) 在比较对象中添加“时”。

(4) 指定为0。指定为0以外时，与运算结果无关，将变为非导通。

(5) 将第15位为0N(1)时，将(s1)与当前的时间以0~第2位中指定的条件进行比较。

- 比较对象时间(0~第2位)为0的情况下，不进行比较对象的时间数据(时、分、秒)的比较。为1的情况下，进行比较对象的时间数据(时、分、秒)的比较。
- 比较运算对象(第15位)为0的情况下，对(s1)中指定的时间数据与(s2)中指定的时间数据进行比较。为1的情况下，对(s1)中指定的时间数据与当前的时间数据进行比较。(s2)中指定的时间数据将被忽略。
- 比较对象位的处理内容如下所示。

与任意时间数据比较时的(s3)值	与当前时间数据比较时的(s3)值	比较对象时间	处理内容
0001H	8001H	秒	仅对(s1)+2进行比较。
0002H	8002H	分	仅对(s1)+1进行比较。
0003H	8003H	分、秒	对(s1)+1、(s1)+2进行比较。
0004H	8004H	时	仅对(s1)进行比较。
0005H	8005H	时、秒	对(s1)、(s1)+2进行比较。
0006H	8006H	时、分	对(s1)、(s1)+1进行比较。
0007H	8007H	时、分、秒	对(s1)、(s1)+1、(s1)+2均进行比较。
0001H~0007H、8001H~8007H以外		无	对(s1)、(s1)+1、(s1)+2均不进行比较。(变为非导通。)

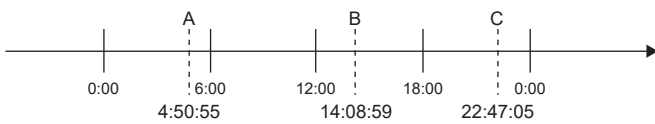
- 比较对象软元件中存储的数据不能识别为时间数据的情况下，执行指令后SM709(DT、TM指令非法数据检测标志)将ON，变为非导通(ENO为OFF)。(s1)~(s1)+2或(s2)~(s2)+2超出相应软元件/标签的分配范围时，SM709也将为ON，变为非导通(ENO为OFF)。一旦SM709变为ON，则在复位/电源OFF前，将一直保持ON状态，因此应根据需要将其置为OFF。

• 各指令的比较运算结果如下所示。

指令符号(梯形图、FBD/LD)	条件	比较运算结果
TM=、EQ	$(s1) = (s2)$	导通状态(ENO为ON)
TM<>、NE	$(s1) \neq (s2)$	
TM>、GT	$(s1) > (s2)$	
TM<=、LE	$(s1) \leq (s2)$	
TM<、LT	$(s1) < (s2)$	
TM>=、GE	$(s1) \geq (s2)$	
TM=、EQ	$(s1) \neq (s2)$	非导通状态(ENO为OFF)
TM<>、NE	$(s1) = (s2)$	
TM>、GT	$(s1) \leq (s2)$	
TM<=、LE	$(s1) > (s2)$	
TM<、LT	$(s1) \geq (s2)$	
TM>=、GE	$(s1) < (s2)$	

例

A、B、C各个时间的比较如下所示。



• 上述时间A、B、C的比较运算结果如下所示。即使以相同条件进行了比较的情况下，根据选择的比较对象其比较运算结果有所不同。

比较对象	比较条件*1		
	A<B	B<C	A<C
秒	导通	非导通	非导通
分	非导通	导通	非导通
分、秒	非导通	导通	非导通
时	导通	导通	导通
时、秒	导通	导通	导通
时、分	导通	导通	导通
时、分、秒	导通	导通	导通
无	非导通	非导通	非导通

*1 FBD/LD的情况下，导通时ENO显示为ON，非导通时ENO显示为OFF。

- 通过FBD/LD使用了LDTM_□指令的情况下，应将EN指定为左母线，或始终ON的变量部件/常数部件。
- 将ORTM_□指令通过FBD/LD使用的情况下，将EN指定为TRUE时ENO将ON。EN不成为执行条件。

出错

没有运算错误。

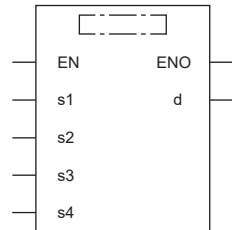
时间比较输出

TCMP (P)

对(s1)、(s2)、(s3)中指定的比较时间数据与(s4)中指定的时间数据进行比较, 根据其结果(小、一致、大)(d)、(d)+1、(d)+2中之一将变为ON。

梯形图	ST
	ENO:=TCMP (EN, s1, s2, s3, s4, d) ; ENO:=TCMPP (EN, s1, s2, s3, s4, d) ;

FBD/LD



■执行条件

指令	执行条件
TCMP	
TCMPP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储比较时间数据(时)的起始软元件	0~23	带符号BIN16位	ANY16
(s2)	存储比较时间数据(分)的起始软元件	0~59	带符号BIN16位	ANY16
(s3)	存储比较时间数据(秒)的起始软元件	0~59	带符号BIN16位	ANY16
(s4)	存储比较时间数据(时、分、秒)的起始软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 3)
(d)	存储比较结果的起始软元件	—	位	ANYBIT_ARRAY (要素数: 3)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

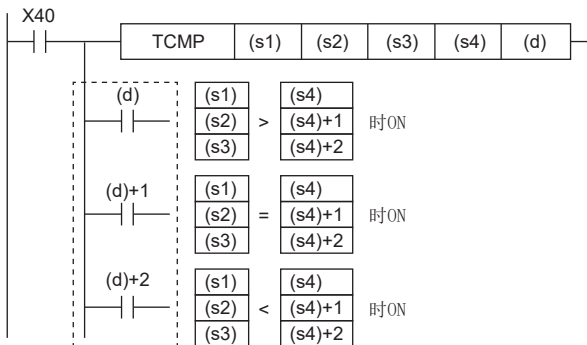
操作数	位		字		双字			间接指定	常数				其它
	X、Y、M、L、SM、F、B、SB、FX、FY	J□\□	T、ST、C、D、W、SD、SW、FD、R、ZR、RD	U□\G□、J□\□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	○*1	○	○	○	○	—	—	○	○	—	—	—	—
(s2)	○*1	○	○	○	○	—	—	○	○	—	—	—	—
(s3)	○*1	○	○	○	○	—	—	○	○	—	—	—	—
(s4)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	○	—	○*2	—	—	—	—	—	—	—	—	—	—

*1 FX、FY不能使用。

*2 T、ST、C不能使用。

功能

- 对(s1)、(s2)、(s3)中指定的比较时间数据与(s4)中指定的时间数据进行比较，根据其结果(小、一致、大)(d)、(d)+1、(d)+2中之一将变为ON。



- (s1): 时, (s2): 分, (s3): 秒
- (s4): 时, (s4)+1: 分, (s4)+2: 秒

出错

错误代码 (SD0)	内容
3285H	(s1)、(s4)中指定的值超出下述范围时。 0~23
	(s2)、(s3)、(s4)+1、(s4)+2中指定的值超出下述范围时。 0~59

扩展时钟数据的读取

S(P). DATERD

从控制器内部的时钟单元中读取包含有1/1000秒的时钟数据。

梯形图	ST
	ENO:=S_DATERD(EN, d); ENO:=SP_DATERD(EN, d);

FBD/LD

■执行条件

指令	执行条件
S. DATERD	
SP. DATERD	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d)	存储读取的时钟数据的起始软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 8)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 从控制器的时钟单元中读取“年、月、日、时、分、秒、星期、1/1000秒”，以BIN值存储到(d)中指定的软元件及其以后。

(数据)	(d)	(d)+1	(d)+2	(d)+3	(d)+4	(d)+5	(d)+6	(d)+7
(内容)	年(公历)	月	日	时(24小时计)	分	秒	星期	1/1000秒
(设置范围)	1980~2079	1~12	1~31	0~23	0~59	0~59	0~6	0~999

- (d)的“年”，以公历4位存储。
- (d)+6的“星期”，即“日~六”，以“0~6”存储。

星期	日	一	二	三	四	五	六
存储数据	0	1	2	3	4	5	6

- 闰年将被自动补偿。

注意事项

- S(P).DATERD指令，即使在控制器中设置了错误的时钟数据，也将读取时钟数据并存储到软元件中。(例：2月30日)通过DATEWR(P)指令及工程工具设置时钟数据时，应设置正确的时钟数据。
- 1/1000秒的时钟数据读取时的误差最大为2ms。(控制器内部时钟单元中存储的数据与通过S(P).DATERD指令读取的数据的误差。)

出错

没有运算错误。

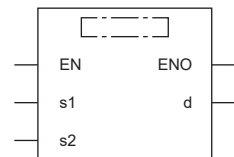
扩展时钟数据的加法运算

S(P). DATE+

对时间数据进行加法运算。

梯形图	ST
	<pre>ENO:=S_DATEPLUS(EN, s1, s2, d); ENO:=SP_DATEPLUS(EN, s1, s2, d);</pre>

FBD/LD



(□中放入S_DATEPLUS、SP_DATEPLUS。)

■执行条件

指令	执行条件
S. DATE+	
SP. DATE+	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储了加法运算时间数据的起始软元件	功能参照	带符号BIN16位	ANY16_ARRAY (要素数: 5)
(s2)	存储了加法运算时间(时刻)数据的起始软元件	功能参照	带符号BIN16位	ANY16_ARRAY (要素数: 5)
(d)	存储加法运算结果时刻(时间)数据的起始软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 5)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 将(s1)中指定的时间数据与(s2)中指定的时间数据进行加法运算，将加法运算结果存储到(d)中指定的软元件编号及其以后。

(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

将6时32分40秒875与7时48分10秒500进行加法运算的情况下

(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

- 运算结果的时间超过了24时的情况下，减去了24小时后的值将成为运算结果。例如，对14时20分30秒875与20时20分20秒500进行了加法运算的情况下，其结果不是34时40分51秒375，而是10时40分51秒375。

(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

要点

- (s1)+3、(s2)+3、(d)+3的软元件在运算中不使用。
- 可以将通过S(P). DATERD指令读取的时钟数据直接进行加法运算。

(d)	时
(d)+1	分
(d)+2	秒
(d)+3	星期
(d)+4	1/1000秒

通过S(P). DATERD指令读取的情况下，在“秒”与“1/1000秒”之间放入“星期”。
在S(P). DATE+指令中，不进行运算，因此可以直接进行加法运算。

出错

错误代码(SD0)	内容
3285H	(s1)、(s2)的设置数据超出范围时。

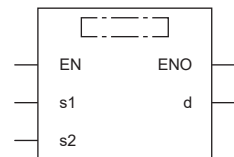
扩展时钟数据的减法运算

S(P). DATE-

对时间数据进行减法运算。

梯形图	ST
	ENO:=S_DATEMINUS(EN, s1, s2, d); ENO:=SP_DATEMINUS(EN, s1, s2, d);

FBD/LD



(□中放入S_DATEMINUS、SP_DATEMINUS。)

■执行条件

指令	执行条件
S. DATE-	
SP. DATE-	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储了被减的时间数据的起始软元件	功能参照	带符号BIN16位	ANY16_ARRAY (要素数: 5)
(s2)	存储了减法运算时间(时刻)数据的起始软元件	功能参照	带符号BIN16位	ANY16_ARRAY (要素数: 5)
(d)	存储减法运算结果时刻(时间)数据的起始软元件	—	带符号BIN16位	ANY16_ARRAY (要素数: 5)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	—	—	○	—	—	—	—	—

功能

- 将(s1)中指定的时间数据与(s2)中指定的时间数据进行减法运算，将减法运算结果存储到(d)中指定的软元件编号及其以后。

(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

将10时40分20秒875与3时50分10秒500进行减法运算的情况下

(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

- 运算结果时间为负数的情况下，将该数据进行了+24的值将成为运算结果。例如，将4时50分32秒875与10时42分12秒500进行减法运算的情况下，其结果不是-6时8分20秒375，而是18时8分20秒375。

(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

要点

- (s1)+3、(s2)+3、(d)+3的软元件在运算中不使用。
- 可以将通过S(P). DATERD指令读取的时钟数据直接进行减法运算。

(d)	时
(d)+1	分
(d)+2	秒
(d)+3	星期
(d)+4	1/1000秒

通过S(P). DATERD指令读取的情况下，在“秒”与“1/1000秒”之间放入“星期”。
在S(P). DATE-指令中不进行运算，因此可以直接进行减法运算。

出错

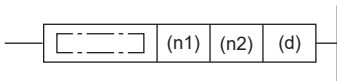
错误代码(SD0)	内容
3285H	(s1)、(s2)的设置数据超出范围时。

21.2 时机计测指令

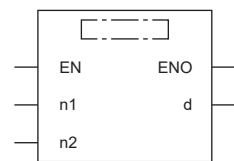
时机脉冲发生

DUTY


将用户用时机时钟按指定的扫描数置为ON，按指定的扫描数置为OFF。

梯形图	ST
	ENO:=DUTY (EN, n1, n2, d);

FBD/LD



■执行条件

指令	执行条件
DUTY	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(n1)	置为ON的扫描数	0~65535	无符号BIN16位	ANY16
(n2)	置为OFF的扫描数	0~65535	无符号BIN16位	ANY16
(d)	使其动作的用户用时机时钟特殊继电器软元件编号	SM420~SM424	位	ANY_BOOL*1
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能使用SM420~SM424中分配的标签。

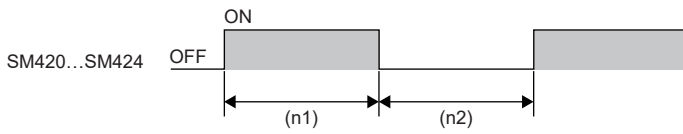
■可用软元件

操作数	位		字			双字		间接指定	常数			其它
	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 只能使用SM420~SM424。

功能

- 将(d)中指定的SM420~SM424(用户用时机时钟)按照(n1)中指定的扫描数为ON,按照(n2)中指定的扫描数为OFF。



(n1): (n1)扫描

(n2): (n2)扫描

- 在扫描执行类型程序中使用SM420~SM424。
- (n1)、(n2)被设置为0的情况下, (n1)=0、(n2)≥0时SM420~SM424将保持OFF不变, (n1)>0、(n2)=0时SM420~SM424将保持ON不变。
- 执行DUTY指令时将(n1)、(n2)、(d)中指定的数据登录到系统中, 时机脉冲的ON/OFF通过END处理进行。


出错

错误代码(SD0)	内容
2820H	(d)中指定的软元件为SM420~SM424以外时。

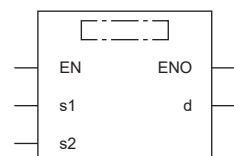
指定数据的时间计测

TIMCHK


对软元件的ON时间进行计测，当连续ON时间大于等于设置时间时，将指定软元件置为ON。

梯形图	ST
	ENO:=TIMCHK(EN, s1, s2, d);

FBD/LD



■执行条件

指令	执行条件
TIMCHK	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储计测的当前值的软元件(单位: 100ms)	—	带符号BIN16位	ANY16
(s2)	计测的设定值或存储了计测的设定值的软元件(单位: 100ms)	0~32767	带符号BIN16位	ANY16
(d)	时限到时置为ON的软元件	—	位	ANY_BOOL
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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	
(s1)	—	—	○	—	—	—	○	—	—	—	—
(s2)	○	○	○	○	○	—	○	○	—	—	—
(d)	○	—	○	—	—	—	○	—	—	—	—

功能

- 对条件软元件的ON时间进行计测，当连续ON时间大于等于(s2)中指定的软元件中设置的时间时，将(d)中指定的软元件置为ON。
- 如果对(s1)中指定的软元件的当前值执行清零，则(d)中指定的软元件在执行指令上升沿时变为OFF。(s1)中指定的软元件的当前值与(d)中指定的软元件的ON状态，即使执行指令变为OFF，其状态也将被保持。
- 计测的当前值以100ms单位被存储。此外，计测的设定值以100ms单位进行设置。
- (s2)中指定了0的情况下，如果对(s1)中指定的软元件的当前值执行清零，则(d)中指定的软元件在指令上升沿时变为OFF。
- (s2)中指定了0~32767以外的情况下，在执行指令ON的下一个扫描中(d)将变为ON。

出错

没有运算错误。

22 模块访问

22.1 模块访问指令

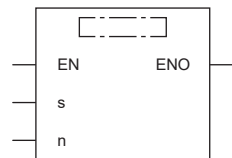
I/O刷新

RFS (P)

对指定的软元件开始的n点进行刷新，进行外部输入的获取或至输出模块的输出。

梯形图	ST
	ENO:=RFS (EN, s, n); ENO:=RFSP (EN, s, n);

FBD/LD



■执行条件

指令	执行条件
RFS	
RFSP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	刷新的起始软元件	—	位	ANY_BOOL*1
(n)	刷新点数	1~4096	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 只能使用软元件(X、Y)中分配的标签。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它
	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)	—	○	○	○	○	—	—	○	○	—	—

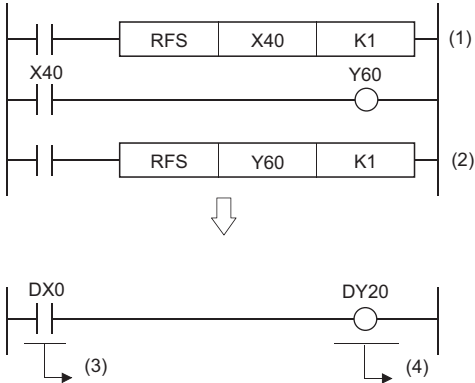
*1 只能使用X、Y。

功能

- 是1个扫描途中仅对相应的软元件进行刷新，进行外部输入的获取或至输出模块的输出的功能。
- 输入的获取及至外部的输出是仅在执行程序的END指令后批量地进行，因此1个扫描中不能向外部输出脉冲信号。执行I/O刷新指令时，程序执行途中相应的输入(X)或输出(Y)将被强制刷新，因此1个扫描中可以向外输出脉冲信号。
- 将输入(X)或输出(Y)以1点单位进行刷新的情况下，应使用直接访问输入(DX)、直接访问输出(DY)。

例

将通过RFS指令进行的程序设置为通过直接访问进行的程序的情况下



- (1) 进行X40的刷新。
- (2) 进行Y60的刷新。
- (3) 直接访问输入
- (4) 直接访问输出

出错

错误代码(SD0)	内容
2820H	(s)中指定的软元件开始的(n)点的范围超出了相邻I/O的范围时。

从模块中的1字/2字数据读取(16位指定)

FROM(P)、DFROM(P)

- FROM(P):

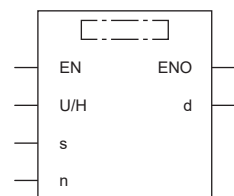
从指定的模块内的缓冲存储器地址开始读取n字的数据。

- DFROM(P):

从指定的模块内的缓冲存储器地址开始读取n×2字的数据。

梯形图	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



■执行条件

指令	执行条件
FROM DFROM	
FROMP DFROMP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(U/H)	智能模块No.	1H~10H、3E0H、40H、42H	无符号BIN16位	ANY16
(s)	存储了读取数据的缓冲存储器及CPU存储器的起始地址	0~65535	无符号BIN16位	ANY16
(d)	FROM(P)	—	带符号BIN16位	ANY16*1
	DFROM(P)		带符号BIN32位	ANY32*1
(n)	读取数据数	0~65535	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字			间接指定	常数				其它(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)	○	○	○	○	○	—	—	○	○	—	—	○	
(s)	○	○	○	○	○	—	—	○	○	—	—	—	
(d)	○	—	○	—	—	○*1	—	○	—	—	—	—	
(n)	○	○	○	○	○	—	—	○	○	—	—	—	

*1 只能使用DFROM(P)指令。

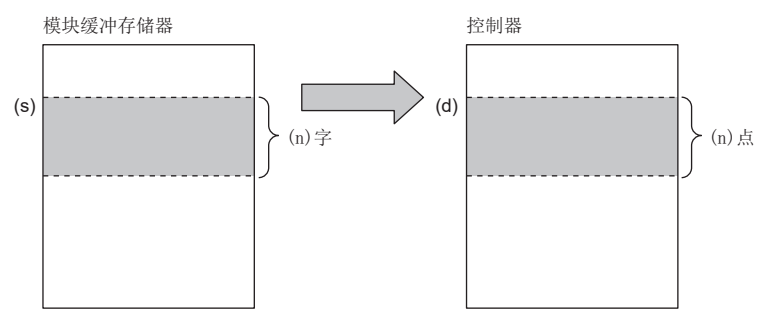
功能

按下表所示指定控制器的智能模块No.。

控制器	智能模块No.
CPU功能部	3E0H
内置网络功能部	40H
内置运动功能部	42H

FROM (P)

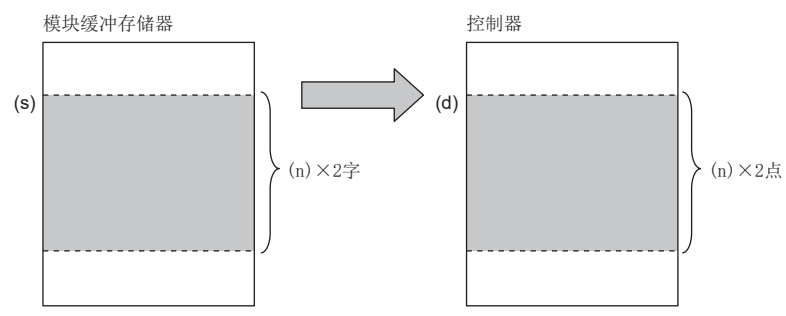
- 从(U/H)中指定的模块内的缓冲存储器的(s)中指定的地址开始，读取(n)字的数据，存储到(d)中指定的软元件以后。



- 读取数据(n)为0时，将变为无处理。
- 执行指令时，对象模块的异常及处理中等无法访问的情况下，将变为无处理。

DFROM (P)

- 从(U/H)中指定的模块内的缓冲存储器的(s)中指定的地址开始，读取(n)×2字的数据，存储到(d)中指定的软元件以后。



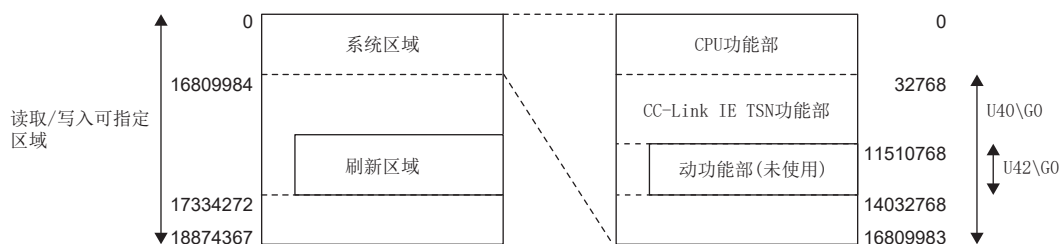
- 读取数据(n)为0时，将变为无处理。
- 执行指令时，对象模块的异常及处理中等无法访问的情况下，将变为无处理。

出错

错误代码 (SD0)	内容
2820H	(U) 中指定的输入输出编号是不具有缓冲存储器的模块时。
2823H	(H) 中指定的输入输出编号是不具有缓冲存储器的模块时。
	(s) 中指定的地址超出缓冲存储器的范围时。
	(s) 中指定的地址开始的 (n) 点超出缓冲存储器的范围时。(FROM(P) 指令的情况下)
	(s) 中指定的地址开始的 2×(n) 点超出缓冲存储器的范围时。(DFROM(P) 指令的情况下)
3602H	在优先级1~16的中断程序内对智能模块执行了本指令时。

要点

- 模块的数据读取也可使用模块访问软元件进行。(☞所使用的控制器的用户手册)
- CPU缓冲存储器的可读取/写入区域的刷新区域，在不进行刷新设置的情况下，可以作为可读取/写入指定的区域使用。此外，即使进行了刷新设置的情况下，刷新发送范围以后也可作为可读取/写入指定的区域使用。操作数(U/H)中指定了CC-Link IE TSN功能部或运动功能部的智能模块No. U40\G0、U42\G0时，可以访问下述范围。

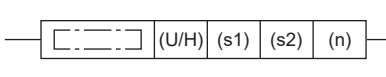


- 从CPU缓冲存储器读取数据时，可以使用CPU缓冲存储器访问软元件。(☞所使用的控制器的用户手册)
- 在FROM(P)指令、DFROM(P)指令中，可以从64K字或其以下的缓冲存储器地址中读取数据。从超过64K的缓冲存储器地址中读取数据的情况下，使用FROMD(P)指令、DFROMD(P)指令。(☞988页 FROMD(P)、DFROMD(P))

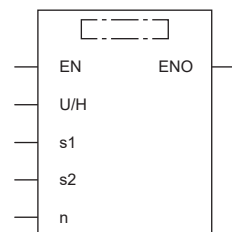
至模块的1字/2字数据写入(16位指定)

TO(P)、DTO(P)



- TO(P)：将指定的软元件开始的n点的数据写入到模块内的缓冲存储器中。
- DTO(P)：将指定的软元件开始的n×2点的数据写入到模块内的缓冲存储器中。

梯形图	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



■执行条件

指令	执行条件
TO DTO	
TOP DTOP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(U/H)	智能模块No.	1H~10H、3E0H、40H、42H	无符号BIN16位	ANY16	
(s1)	写入数据的缓冲存储器及CPU存储器的起始地址	0~65535	无符号BIN16位	ANY16	
(s2)	写入数据或存储了写入数据的起始软元件	TO(P)	-32768~32767	带符号BIN16位	ANY16*1
		DTO(P)	-2147483648~2147483647	带符号BIN32位	ANY32*1
(n)	写入数据数	0~65535	无符号BIN16位	ANY16	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字		间接指定	常数				其它(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 只能使用DTO(P)指令。

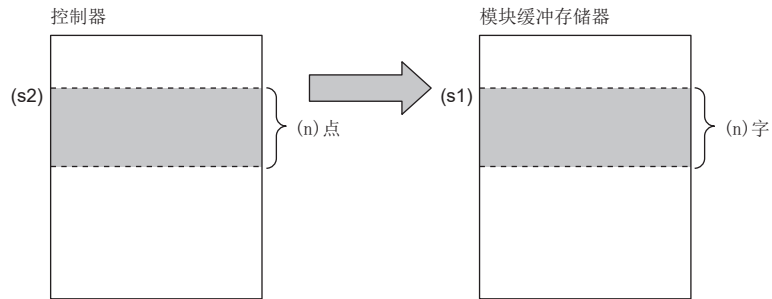
功能

按下表所示指定控制器的智能模块No.。

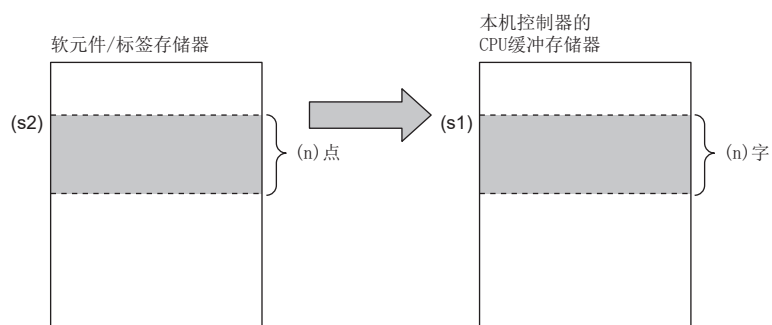
控制器	智能模块No.
CPU功能部	3E0H
内置网络功能部	40H
内置运动功能部	42H

■T0(P)

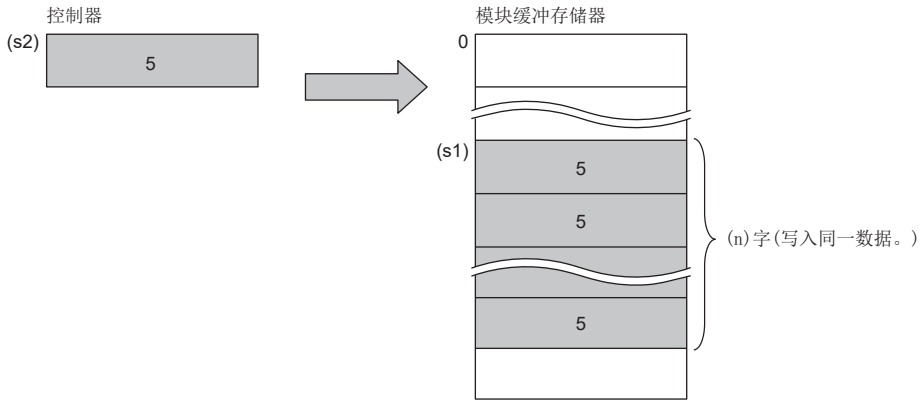
- 将 (s2) 中指定的软元件开始的 (n) 点的数据，写入到 (U/H) 中指定的模块或本机控制器内的缓冲存储器的 (s1) 中指定的地址及其以后。
- 至模块的字数据写入



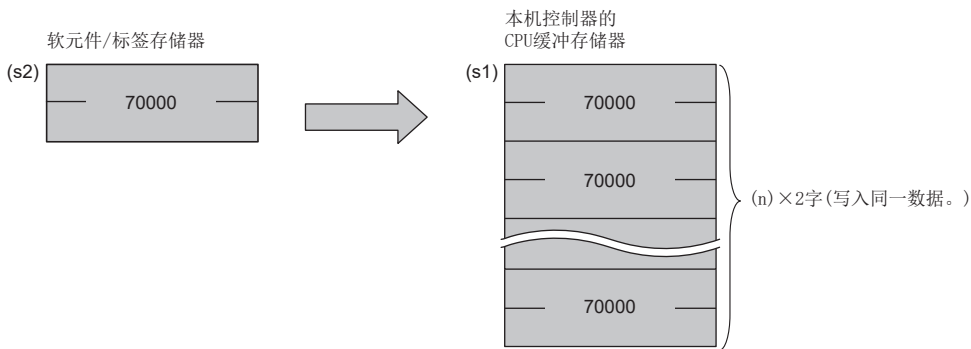
- 至本机控制器的字数据写入



- (s2)中指定了常数的情况下，将同一数据((s2)中指定的值)，写入到指定的缓冲存储器地址开始的(n)字中。
- 至模块的字数据写入



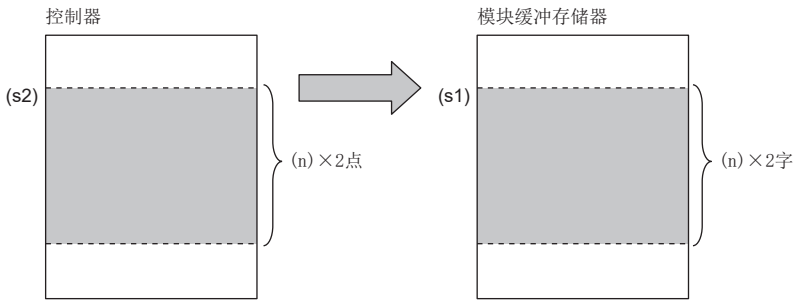
- 至本机控制器的字数据写入



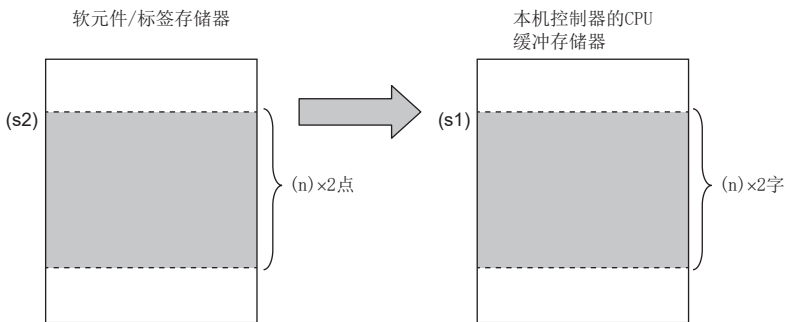
- 执行指令时，对象模块的异常及处理中等无法访问的情况下，将变为无处理。

■DT0(P)

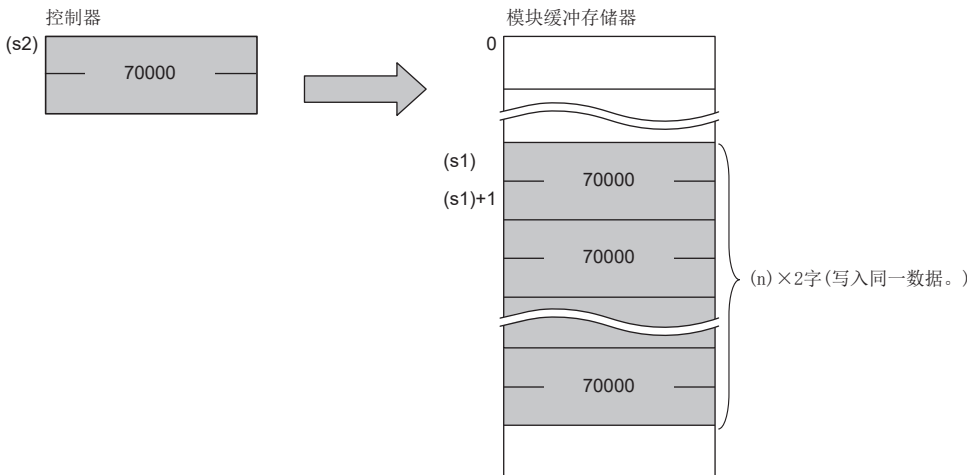
- 将(s2)中指定的软元件开始的(n)×2点的数据，写入到(U/H)中指定的模块或本机控制器内的缓冲存储器的(s1)中指定的地址及其以后。
- 至模块的双字数据写入



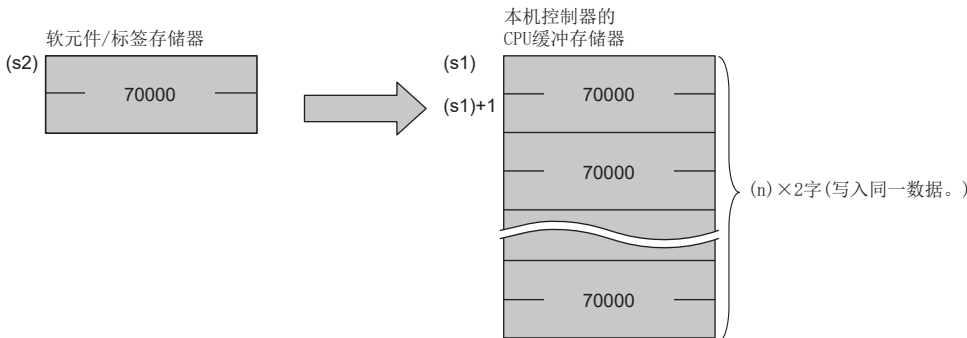
- 至本机控制器的双字数据写入



- (s2)中指定了常数的情况下，将同一数据((s2)中指定的值)，写入到指定的缓冲存储器地址开始的(n)×2字中。
- 至模块的双字数据写入



- 至本机控制器的双字数据写入



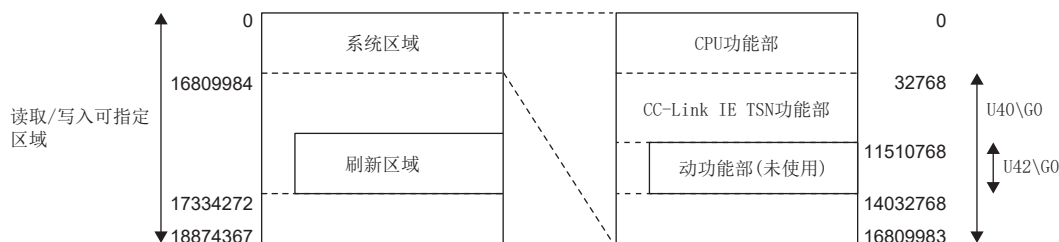
- 执行指令时，对象模块的异常及处理中等无法访问的情况下，将变为无处理。

出错

错误代码 (SD0)	内容
2820H	(U) 中指定的输入输出编号是不具有缓冲存储器的模块时。
2823H	(H) 中指定的输入输出编号是不具有缓冲存储器的模块时。
	(s1) 中指定的地址超出缓冲存储器的范围时。
	(s1) 中指定的地址开始的 (n) 点超出缓冲存储器的范围时。(TO(P) 指令的情况下)
	(s1) 中指定的地址开始的 $2 \times (n)$ 点超出缓冲存储器的范围时。(DT0(P) 指令的情况下)
3602H	在优先级1~16的中断程序内对智能模块执行了本指令时。

要点

- CPU缓冲存储器的可读取/写入区域的刷新区域，在不进行刷新设置的情况下，可以作为可读取/写入指定的区域使用。此外，即使进行了刷新设置的情况下，刷新发送范围以后也可作为可读取/写入指定的区域使用。操作数(U/H)中指定了CC-Link IE TSN功能部或运动功能部的智能模块No. U40\G0、U42\G0时，可以访问下述范围。



- 至CPU缓冲存储器的数据的写入时，可以使用CPU缓冲存储器访问软元件。(☞所使用的控制器的用户手册)
- 在TO(P)指令、DT0(P)指令中，可以将数据写入到64K字或其以下的缓冲存储器地址中。将数据写入超过64K字的缓冲存储器地址中的情况下，使用TOD(P)指令、DTOD(P)指令。(☞992页 TOD(P)、DTOD(P))

从模块中的1字/2字数据读取(32位指定)

FROMD(P)、DFROMD(P)

- FROMD(P) :

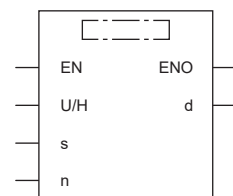
从指定的模块内的缓冲存储器地址开始读取n字的数据。

- DFROMD(P) :

从指定的模块内的缓冲存储器地址开始读取n×2字的数据。

梯形图	ST
	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);

FBD/LD



■执行条件

指令	执行条件
FROMD DFROMD	
FROMDP DFROMDP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(U/H)	智能模块No.	1H~10H、3E0H、40H、42H	无符号BIN16位	ANY16
(s)	存储了读取数据的缓冲存储器及存储CPU存储器的起始地址的起始软元件	0~4294967295	无符号BIN32位	ANY32
(d)	FROMD(P)	—	带符号BIN16位	ANY16*1
	DFROMD(P)		带符号BIN32位	ANY32*1
(n)	读取数据数	0~4294967295	无符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数				其它(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 只能使用DFROMD(P)指令。

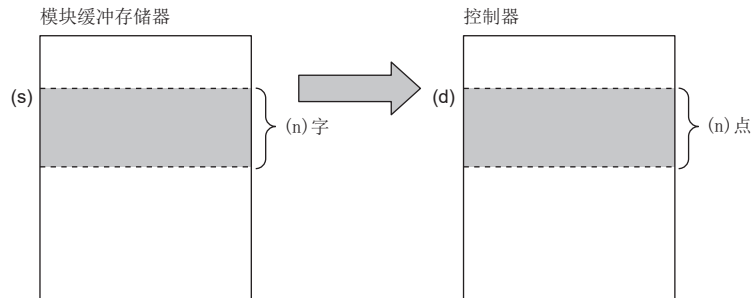
功能

按下表所示指定控制器的智能模块No.。

控制器	智能模块No.
CPU功能部	3E0H
内置网络功能部	40H
内置运动功能部	42H

■FROMD (P)

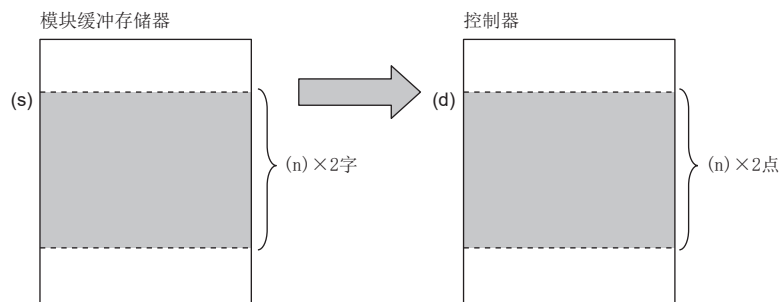
- 从(U/H)中指定的模块内的缓冲存储器的(s)中指定的地址开始，读取(n)字的数据，存储到(d)中指定的软元件以后。



- 读取数据(n)为0时，将变为无处理。
- 执行指令时，对象模块的异常及处理中等无法访问的情况下，将变为无处理。

■DFROMD (P)

- 从(U/H)中指定的模块内的缓冲存储器的(s)中指定的地址开始，读取(n)×2字的数据，存储到(d)中指定的软元件以后。



- 读取数据(n)为0时，将变为无处理。
- 执行指令时，对象模块的异常及处理中等无法访问的情况下，将变为无处理。

错误代码 (SD0)	内容
2820H	(U) 中指定的输入输出编号是不具有缓冲存储器的模块时。
	(s) 中指定的地址超出CPU缓冲存储器的范围时。
	(s) 中指定的地址开始的 (n) 点超出CPU缓冲存储器的范围时。(FROMD(P) 指令的情况下)
	(s) 中指定的地址开始的 2×(n) 点超出CPU缓冲存储器的范围时。(DFROMD(P) 指令的情况下)
2823H	(H) 中指定的输入输出编号是不具有缓冲存储器的模块时。
	(s) 中指定的地址超出缓冲存储器的范围时。
	(s) 中指定的地址开始的 (n) 点超出缓冲存储器的范围时。(FROMD(P) 指令的情况下)
	(s) 中指定的地址开始的 2×(n) 点超出缓冲存储器的范围时。(DFROMD(P) 指令的情况下)
3602H	在优先级1~16的中断程序内对智能模块执行了本指令时。

要点 

- CPU缓冲存储器的可读取/写入区域的刷新区域，在不进行刷新设置的情况下，可以作为可读取/写入指定的区域使用。此外，即使进行了刷新设置的情况下，刷新发送范围以后也可作为可读取/写入指定的区域使用。
- 从CPU缓冲存储器读取数据时，可以使用CPU缓冲存储器访问软元件。(📖所使用的控制器的用户手册)
- 在FROMD(P) 指令、DFROMD(P) 指令中，可以从超过64K的缓冲存储器地址中读取数据。

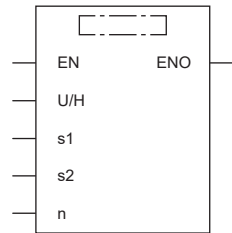
至模块的1字/2字数据写入 (32位指定)

TOD(P)、DTOD(P)

- TOD(P)：将指定的软元件开始的n点的数据写入到模块内的缓冲存储器中。
- DTOD(P)：将指定的软元件开始的n×2点的数据写入到模块内的缓冲存储器中。

梯形图	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



■执行条件

指令	执行条件
TOD DTOD	
TODP DTODP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(U/H)	智能模块No.	1H~10H、3E0H、40H、42H	无符号BIN16位	ANY16
(s1)	写入数据的缓冲存储器及存储CPU存储器的起始地址的起始软元件	0~4294967295	无符号BIN32位	ANY32
(s2)	TOD(P)	-32768~32767	带符号BIN16位	ANY16* ¹
	DTOD(P)	-2147483648~2147483647	带符号BIN32位	ANY32* ¹
(n)	写入数据数	0~4294967295	无符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字		间接指定	常数				其它(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 只能使用DTOD(P)指令。

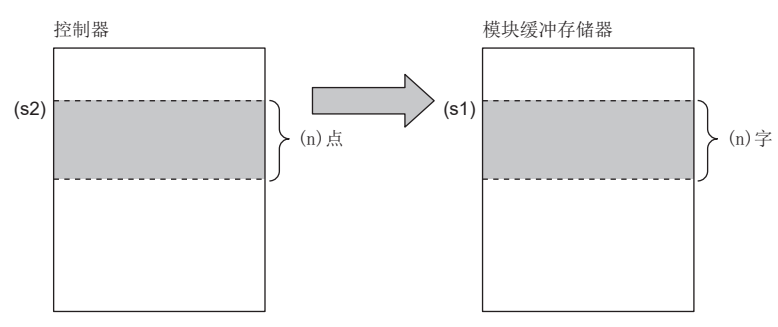
功能

按下表所示指定控制器的智能模块No.。

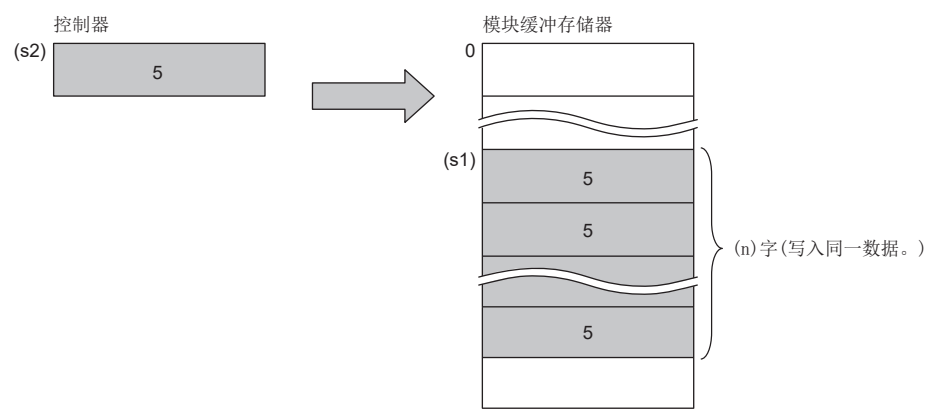
控制器	智能模块No.
CPU功能部	3E0H
内置网络功能部	40H
内置运动功能部	42H

■TOD (P)

- 将 (s2) 中指定的软元件开始的 (n) 点的数据，写入到 (U/H) 中指定的模块或本机控制器内的缓冲存储器的 (s1) 中指定的地址及其以后。



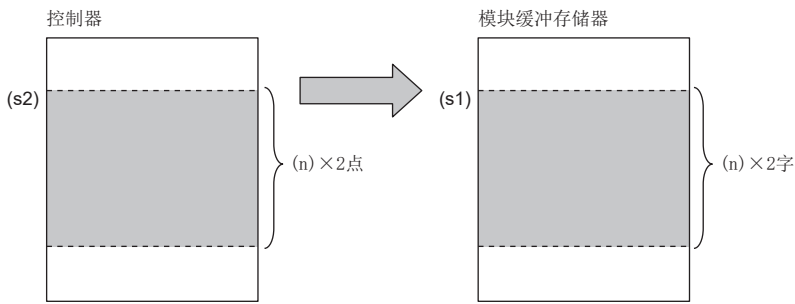
- (s2) 中指定了常数的情况下，将同一数据 ((s2) 中指定的值) 写入到指定的缓冲存储器地址开始的 (n) 字中



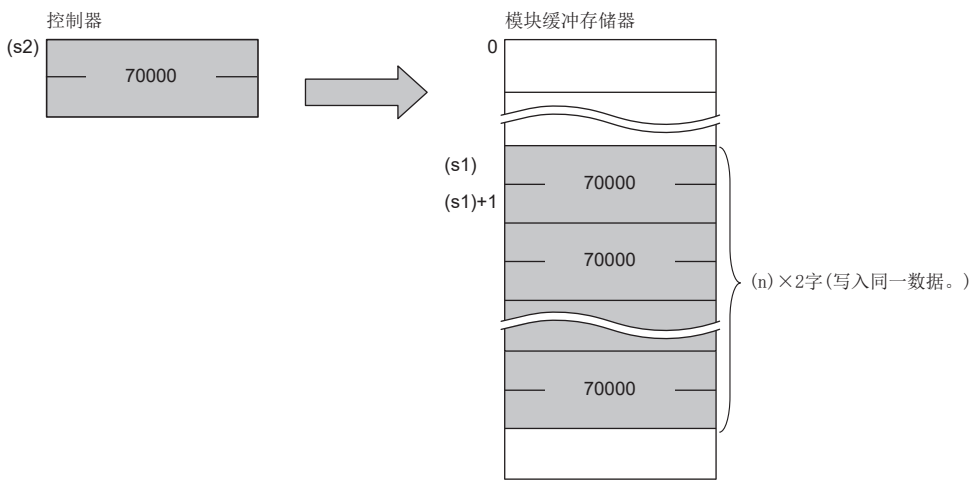
- 执行指令时，对象模块的异常及处理中等无法访问的情况下，将变为无处理。

■DTOD(P)

- 将(s2)中指定的软元件开始的 $(n) \times 2$ 点的数据，写入到(U/H)中指定的模块或本机控制器内的缓冲存储器的(s1)中指定的地址及其以后。



- (s2)中指定了常数的情况下，将同一数据((s2)中指定的值)写入到指定的缓冲存储器地址开始的 $(n) \times 2$ 字中。



- 执行指令时，对象模块的异常及处理中等无法访问的情况下，将变为无处理。

出错

错误代码 (SD0)	内容
2820H	(U) 中指定的输入输出编号是不具有缓冲存储器的模块时。
	(s1) 中指定的地址超出CPU缓冲存储器的范围时。
	(s1) 中指定的地址开始的 (n) 点超出CPU缓冲存储器的范围时。(TOD (P) 指令的情况下)
	(s1) 中指定的地址开始的 $2 \times (n)$ 点超出CPU缓冲存储器的范围时。(DTOD (P) 指令的情况下)
2823H	(H) 中指定的输入输出编号是不具有缓冲存储器的模块时。
	(s1) 中指定的地址超出缓冲存储器的范围时。
	(s1) 中指定的地址开始的 (n) 点超出缓冲存储器的范围时。(TOD (P) 指令的情况下)
	(s1) 中指定的地址开始的 $2 \times (n)$ 点超出缓冲存储器的范围时。(DTOD (P) 指令的情况下)
3602H	在优先级1~16的中断程序内对智能模块执行了本指令时。

要点 

- CPU缓冲存储器的可读取/写入区域的刷新区域，在不进行刷新设置的情况下，可以作为可读取/写入指定的区域使用。此外，即使进行了刷新设置的情况下，刷新发送范围以后也可作为可读取/写入指定的区域使用。
- 至CPU缓冲存储器的数据的写入时，可以使用CPU缓冲存储器访问软元件。(📖所使用的控制器的用户手册)
- 在TOD (P) 指令、DTOD (P) 指令中，可以将数据写入到超过64K的缓冲存储器地址中。

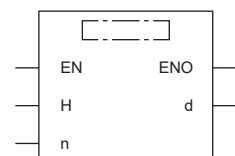
模块固有信息读取

UNIINFRD (P)

读取安装的所有模块的信息。

梯形图	ST
	ENO:=UNIINFRD (EN, H, n, d) ; ENO:=UNIINFRDP (EN, H, n, d) ;

FBD/LD



■执行条件

指令	执行条件
UNIINFRD	
UNIINFRDP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(H)	虚拟	—	无符号BIN16位	ANY16
(d)	存储模块信息的起始软元件 (应确保56字的区域)	—	字	ANY16*1
(n)	虚拟	28 (固定)	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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)	○	—	○	—	—	—	—	○	○	—	—	—	—

功能

将模块信息存储到(d)中指定的软元件及其以后。
数据结构体如下所示。

软元件	区域	内容
(d) (d)+1	区域0	CPU功能部的信息
(d)+2 (d)+3	区域1	内置网络功能部的信息
(d)+4 (d)+5	区域2	内置运动功能部的信息
(d)+6 (d)+7	区域3	内置I/O功能部的信息
(d)+8 (d)+9 ⋮	区域4 ⋮	扩展模块的信息
(d)+42 (d)+43	区域21	
(d)+44 (d)+45 ⋮	区域22 ⋮	
(d)+54 (d)+55	区域27	扩展适配器的信息

• 模块信息的详细内容如下所示。

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
(d)																
(d)+1																

软元件	位	项目名	内容
(d)	b0	输入点数	000: 0点
	b1		001: 8点
	b2		010: 16点 011: 24点
	b3	输出点数	100: 32点
	b4		101: 40点
	b5		110: 48点 111: 64点
	b6	模块类型	0000: 类型识别异常
	b7		0001: 输入
	b8		0010: 输出
	b9		0100: 高速输入输出 0101: 扩展电源 0110: 适配器(串行) 0111: 适配器(模拟) 1000: 智能、或CPU内置功能
	b10	系列类别	00: 不明
	b11		01: MX-F
	b12	模块安装状态	0: 模块未安装 1: 模块已安装
	b13	错误判定*1	00: 正常
	b14		01: 轻度异常 10: 中度异常 11: 重度异常
b15	模块准备完成状态	0: 准备未完成 1: 准备完成	
(d)+1	b0	起始XY、智能模块No.、串行通信通道、模拟ADP编号	<ul style="list-style-type: none"> • 模块类型为输入/输出/高速输入输出的情况下: 起始XY除以16的值 • 模块类型为智能功能模块的情况下: 智能模块No. • 模块类型为扩展适配器(串行)的情况下: 串行通信通道 • 模块类型为扩展适配器(模拟)的情况下: 模拟ADP编号
	b1		
	b2		
	b3		
	b4		
	b5		
	b6		
	b7		
	b8		
	b9		
	b10		
	b11		
	b12		
	b13		
	b14		
	b15		

*1 对于扩展适配器, 不存在各模块的错误状态, 因此始终存储“00: 正常”。

• 控制器内部的CPU功能部、内置网络功能部、内置运动功能部、内置I/O功能部的模块信息如下所示。

软元件	位	项目名	CPU功能部(区域0)	内置网络功能部(区域1)	内置运动功能部(区域2)	内置I/O功能部(区域3)
(d)	b0	输入点数	000 (0点)	000 (0点)	000 (0点)	010 (16点)
	b1					
	b2					
	b3	输出点数	000 (0点)	000 (0点)	000 (0点)	010 (16点)
	b4					
	b5					
	b6	模块类型	1000 (智能、或CPU内置功能)	0000	0000	0011
	b7					
	b8					
	b9					
	b10	系列类别	01 (MX-F)	00	00	01 (MX-F)
	b11					
	b12	模块安装状态	1	1	1	1
	b13	错误判定	(根据内容变化)	00	00	00
	b14					
b15	模块准备完成状态	1	0	0	1	
(d)+1	b15~b0	起始XY、智能模块No.、串行通信通道、模拟ADP编号	03E0H (智能模块No.)	0040H (智能模块No.)	0042H (智能模块No.)	0000H (起始XY除以16的值)

出错

错误代码(SD0)	内容
2820H	(d)中指定的范围超出相应软元件/标签的范围时。
3602H	在包含1台及以上智能模块的配置中，在优先级1~16的中断程序内执行了本指令时。
36C5H	执行指令时与模块的通信中发生了超时时。
36C7H	执行指令时在访问模块中检测到信号异常时。

23 PID运算指令

PID运算指令及PID控制指令的使用分类

用于进行PID控制的指令，有下述两种。

种类	使用	参照目标(概要)	参照目标(指令详细内容)
PID运算指令	实施使用了自动调谐的控制的情况下	1000页 概要	1012页 PID运算指令
PID控制指令	实施与MELSEC-Q系列及MELSEC-L系列相同的PID控制的情况下	1016页 概要	1025页 PID控制指令(不完全微分)、 1037页 PID控制指令(完全微分)

PID运算指令与PID控制指令的比较如下所示。

项目	PID运算指令	PID控制指令
PID运算方式	不完全微分	不完全微分/完全微分
采样周期/采样时间	1~32767ms	10~60000ms
控制环路数	1环路/指令	最大32环路
自动调谐	有(极限循环法/步响应法)	无

23.1 概要

以下介绍使用了PID运算指令的PID控制的概要。

PID运算指令

PID运算指令是指，为了趋近目标值(SV)，通过测定值(PV)组合P动作(比例动作)、I动作(积分动作)、D动作(微分动作)，对输出值(MV)进行运算的指令。

■报警输出功能

对于输入(测定值)变化量及输出(值)变化量，可以将报警输出置为ON。

■输出值的上下限设置

设置输出的上下限值，可以抑制PID控制的积分项的增大。

■自动调谐功能

可自动设置比例增益(K_p)、积分时间(T_I)、微分时间(T_D)。可以选择极限循环法或步响应法。

■PID运算指令的运算方式

通过速度型·测定值微分型进行运算。

PID运算指令的基本运算公式[参考]

本指令通过速度型・测定值微分型的运算公式进行PID运算。

(s3)+1(根据动作设置(ACT))的位0的内容, 执行正动作或逆动作的运算公式。

需要进行运算的各个值, 通过(s3)及其以后指定的控制用数据的内容进行运算。

• 运算公式

正动作/逆动作 (s3)+1的位0)	运算公式
正动作 (OFF)	$\Delta MV = K_p \{ (EV_n - EV_{n-1}) + \frac{T_s}{T_i} EV_n + D_n \}$ $EV_n = PV_{nf} - SV$ $D_n = \frac{T_D}{T_s + K_D \cdot T_D} (-2PV_{nf-1} + PV_{nf} + PV_{nf-2}) + \frac{K_D \cdot T_D}{T_s + K_D \cdot T_D} \cdot D_{n-1}$ $MV_n = \Sigma \Delta MV$
逆动作 (ON)	$\Delta MV = K_p \{ (EV_n - EV_{n-1}) + \frac{T_s}{T_i} EV_n + D_n \}$ $EV_n = SV - PV_{nf}$ $D_n = \frac{T_D}{T_s + K_D \cdot T_D} (2PV_{nf-1} - PV_{nf} - PV_{nf-2}) + \frac{K_D \cdot T_D}{T_s + K_D \cdot T_D} \cdot D_{n-1}$ $MV_n = \Sigma \Delta MV$

运算公式中的符号含义如下所示。

符号	含义
EV_n	本次样本时的偏差
EV_{n-1}	1周期前的偏差
SV	目标值
PV_{nf}	本次样本时的测定值(滤波器后)
PV_{nf-1}	1周期前的测定值(滤波器后)
PV_{nf-2}	2周期前的测定值(滤波器后)
ΔMV	输出变化量
MV_n	本次的操作量
D_n	本次的微分项
D_{n-1}	1周期前的微分项
K_p	比例增益
T_s	采样周期
T_i	积分常数
T_D	微分常数
K_D	微分增益

本次样本时的测定值(滤波器后)(PV_{nf})是将读入的测定值通过下式进行了运算后的值。未设置输入数据的滤波器系数的情况下, 将变为与输入数据的测定值(PV)相同的值。

$$PV_{nf} = PV_n + L(PV_{nf-1} - PV_n)$$

PV_{nf} : 本次样本时的测定值, L: 滤波器系数, PV_{nf-1} : 1周期前的测定值(滤波器后)

控制用数据

以下介绍PID运算指令的控制用数据的内容。

采样时间：(s3)

设置范围：1~32767[ms]

设置用于进行PID运算的周期(ms)。

- 自动调谐(极限循环法)时

应设置为(可编程控制器的运算周期)<(采样时间)。

- 自动调谐(步响应法)时

设置为1000ms或其以上。

■关于最大误差

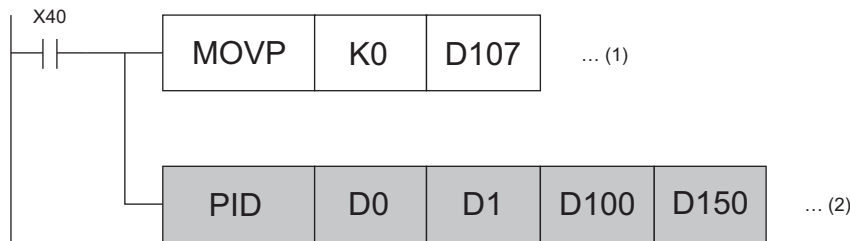
采样时间(T_s)的最大误差为 $-(1\text{运算周期}+1\text{ms}) \sim +(1\text{运算周期})$ 。

- 采样时间(T_s)为小值时

最大误差的变动有可能带来问题。应设置恒定扫描后执行，或在定时器中断程序内编程。

- 短于可编程控制器的1个运算周期的情况下

发生PID运算错误(11A6H)，但以采样时间(T_s)=运算周期执行PID运算。在此情况下，应在定时器中断内使用PID运算指令，在执行PID运算指令之前清除(s3)+7之后再使用。



(1) 对(s3)+7进行复位(首次执行中断程序时，应通过脉冲化指令清除内部处理用寄存器。)

(2) 执行PID运算。

动作设置：(s3)+1

■动作方向(正动作/逆动作)：(s3)+1 位0

设置范围：OFF=正动作/ON=逆动作

选择正动作或逆动作。

- 自动调谐(极限循环法)的情况下

自动调谐，根据需求，需要设置以正动作还是逆动作方向进行PID控制。

- 自动调谐(步响应法)的情况下

自动调谐，无论是以正动作还是逆动作执行，在完成时均将自动进行设置。

[正动作((s3)+1 位0=OFF)]

[逆动作((s3)+1 位0=ON)]

■报警设置(输入变化量、输出变化量)：(s3)+1 位1、2

设置范围：OFF=无报警/ON=报警有效

可以进行输入变化量、输出变化量的检查。检查结果，可通过(s3)+24确认。(☞ 1008页 报警输出的标志动作：(s3)+24)

- 输入变化量((s3)+1 位1)

使用输入变化量报警的情况下，需要将下述位置为ON，设置希望检查的值。

设置项目			内容	设置范围
动作设置(ACT)	(s3)+1	位1	输入变化量报警	ON：使用 OFF：不使用
输入变化量报警设置值	(s3)+20		输入变化量(增侧)报警设置值	0~32767
	(s3)+21		输入变化量(减侧)报警设置值	

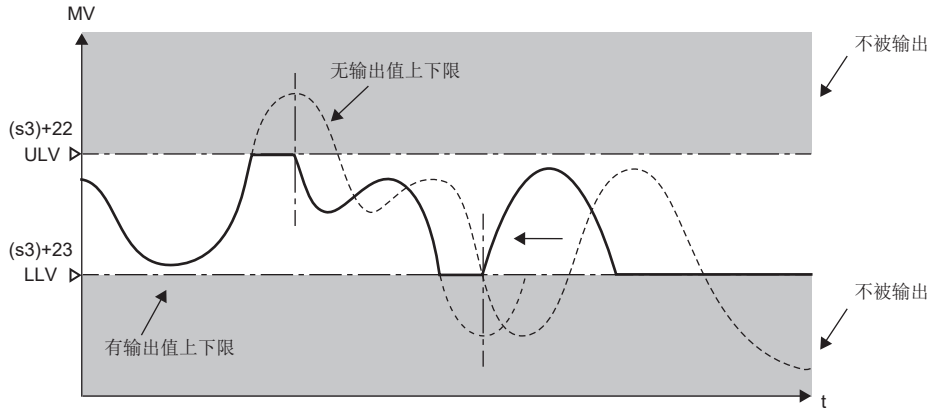
- 输出变化量((s3)+1 位2)

使用输出变化量报警的情况下，需要将下述位置为ON，设置希望检查的值。

设置项目			内容	设置范围
动作设置(ACT)	(s3)+1	位2	输出变化量报警	ON：使用 OFF：不使用
		位5	输出值上下限设置	必须设置为OFF
输出变化量报警设置值	(s3)+22		输出变化量(增侧)报警设置值	0~32767
	(s3)+23		输出变化量(减侧)报警设置值	

■输出值上下限设置：(s3)+1 位5

根据输出值上下限设置的输出值如下所示。



MV: 输出值

ULV: 输出上限值

LLV: 输出下限值

t: 时间

输出值上下限设置对PID控制的积分项的增大有抑制效果。使用本功能的情况下，必须将(s3)+1的位2设置为OFF。

设置项目			内容	设置范围
动作设置 (ACT)	(s3)+1	位2	输出变化量报警	必须设置为OFF
		位5	输出值上下限设置	ON: 使用 OFF: 不使用

输入滤波器：(s3)+2

设置范围：0~99[%]

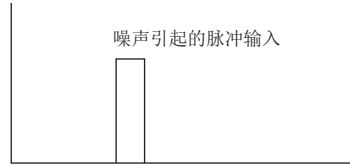
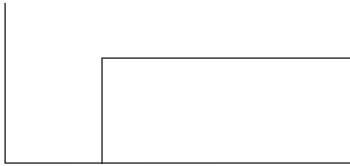
PID控制：比例动作、积分动作、微分动作

输入滤波器(α)是指，用于减少测定值(PV)的噪声引起的变动的软件滤波器。根据控制对象的特性及其噪声电平设置滤波器的输入滤波器(α)，可以抑制噪声的影响。

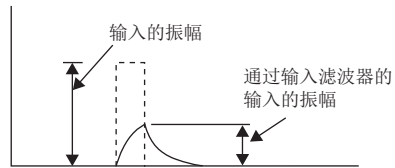
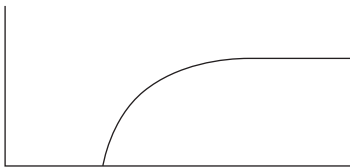
- 如果过小，滤波器的效果将变小。
- 如果过大，输入的反应将变差。

输入滤波器(α)对目标值(SV)起作用，因此比例动作、积分动作、微分动作将受到影响。

实际的测定值
(PV)



通过输入滤波器的
测定值 (PV)



比例增益: (s3)+3

设置范围: 1~32767 [%]

PID控制: 比例动作

通过比例动作, 输出值(MV)与偏差(目标值(SV)和测定值(PV)的差)成比例增加。该比例称为比例增益(K_p), 用下述关系式表示。

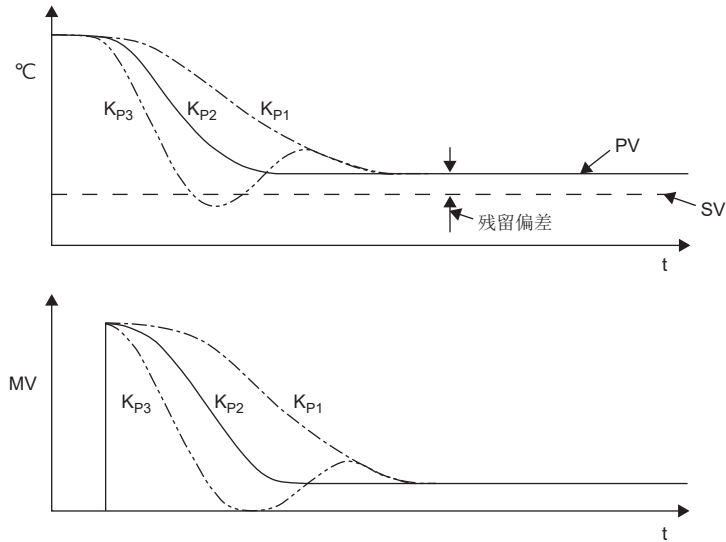
输出值(MV)=比例增益(K_p) \times 偏差(EV)

此外, 将比例增益(K_p)的倒数称为比例带。

随着比例增益(K_p)的增大, 使测定值(PV)接近目标值(SV)的动作变强。

例

制冷机(正动作)情况下的比例动作(P动作)



比例增益(K_p): $K_{p3} > K_{p2} > K_{p1}$

°C: 温度

SV: 目标值

PV: 测定值

MV: 输出

t: 时间

积分时间: (s3)+4

设置范围: 0~32767 [$\times 100\text{ms}$] (0的情况下, 作为 ∞ 处理。(无积分))

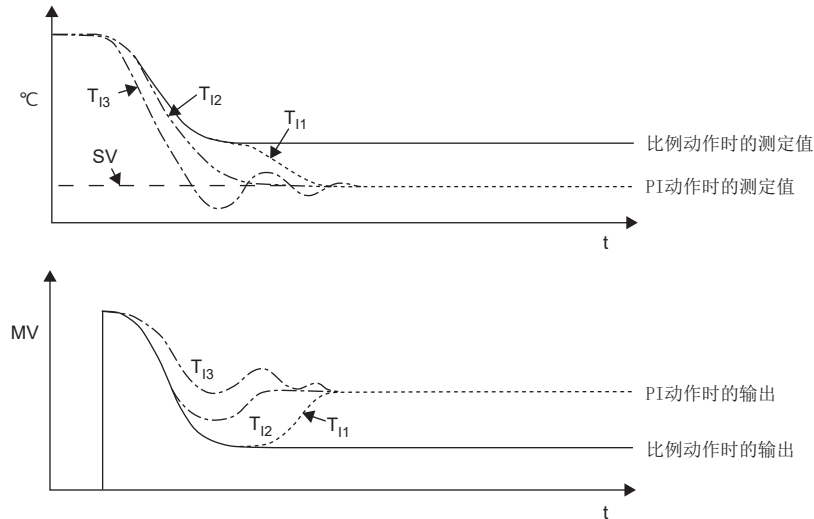
PID控制: 积分动作

积分动作中产生偏差后, 积分动作的输出变为比例动作的输出为止的时间称为积分时间, 以 T_I 表示。

减小 T_I 时, 积分动作变强。

例

制冷机(正动作)情况下的PI动作



积分时间(T_I): $0 < T_{I3} < T_{I2} < T_{I1}$

°C: 温度

SV: 目标值

MV: 输出

t: 时间

微分增益: (s3)+5

设置范围: 0~200 [%]

PID控制: 微分动作

对微分动作的输出施加滤波器。微分增益(K_D)仅影响微分动作。

- 减小微分增益(K_D)时, 对于因干扰等引起的测定值(PV)的变化, 会瞬间做出响应。
- 增大微分增益(K_D)时, 对于因干扰等引起的测定值(PV)的变化, 将长时间进行响应。

要点

最初应将微分增益(K_D)设置为0, 通过输入滤波器(α)进行调整。输出变化对干扰的响应过高时, 应增大微分增益(K_D)的值。

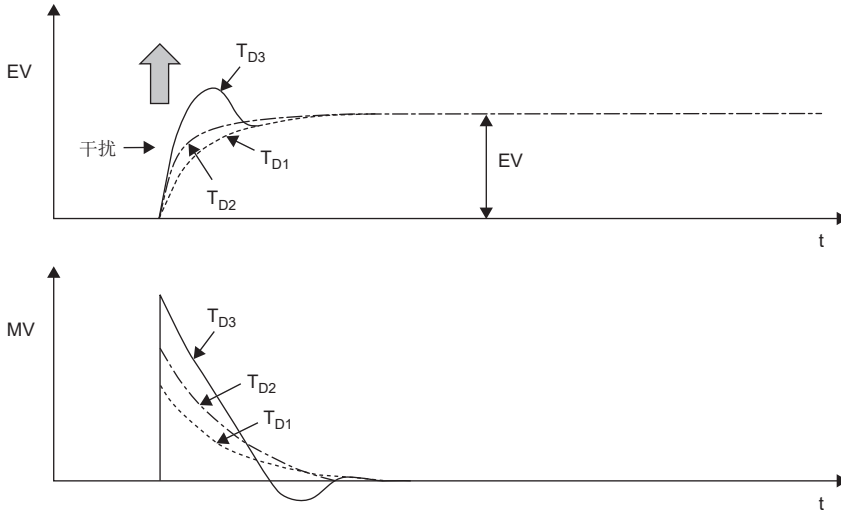
微分时间 (T_D): (s3)+6

设置范围: 0~32767 [$\times 10\text{ms}$] (0的情况下, 变为无微分。)

PID控制: 微分动作

用于对测定值(PV)的干扰等引起的波动敏感地反应, 将变动抑制至最小限度。

- 增大微分时间(T_D)时, 防止因干扰等导致控制对象发生较大变动的动作变强。



微分时间 (T_D): $T_{D3} > T_{D2} > T_{D1}$

EV: 偏差

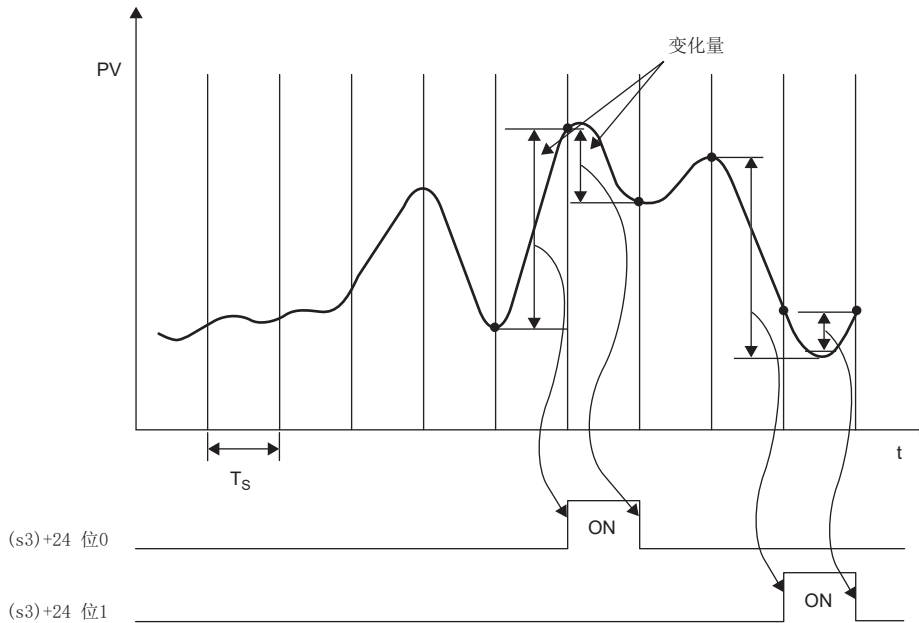
MV: 输出

t: 时间

报警输出的标志动作: (s3)+24

超过设置的输入输出变化量时, 作为报警标志(s3)+24的各个位在执行PID运算指令之后将变为ON。

- 输入变化量报警((s3)+1的位1为ON)的情况下



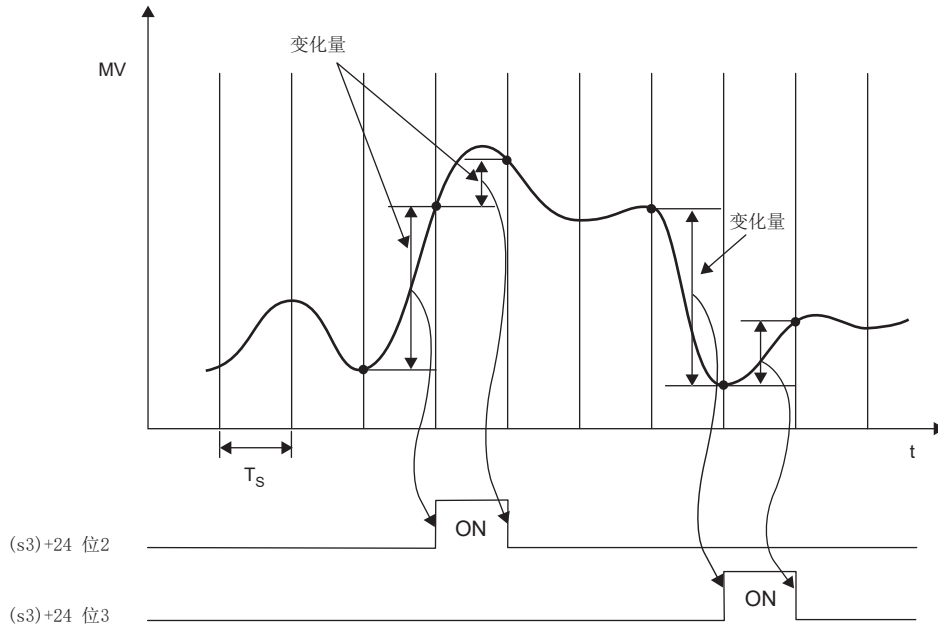
PV: 测定值

T_s : 采样时间

t: 时间

(s3)+24 位0、位1: 报警标志

• 输出变化量报警((s3)+1的位2为ON)的情况下



MV: 输出
 T_s : 采样时间
t: 时间
(s3)+24 位2、位3: 报警标志

自动调谐

为了在PID控制中获得良好的控制结果，需要根据控制对象求出各常数(控制用数据)的最佳值。自动调谐功能是指，为使PID控制达到最佳对重要常数比例增益、积分时间、微分时间进行自动设置的功能。

PID运算指令的自动调谐功能有极限循环法及步响应法这2种方法。以下介绍各个步骤。

要点

自动调谐时，应从系统稳定状态开始。如果从不稳定的状态开始，有可能无法正确调谐。

极限循环法

■自动调谐(极限循环法)中设置的控制用数据

- 比例增益(K_p): (s3)+3
- 积分时间(T_I): (s3)+4
- 微分时间(T_D): (s3)+6

■自动调谐步骤

1. 正动作/逆动作的设置

在(s3)+1(动作设置(ACT))的位0中，设置正动作或逆动作。

2. 极限循环法的选择

将(s3)+1(动作设置(ACT))的位6置为ON，选择极限循环法。
(OFF时，将以步响应法执行动作。)

3. 自动调谐执行标志的ON

将(s3)+1(动作设置(ACT))的位4置为ON。

4. 输入滤波器的设置

设置(s3)+2(输入滤波器(α))。

5. 采样时间的设置

设置(s3)(采样时间(T_S))。

6. 输出值上限(ULV)的设置

在(s3)+26(输出值上限(ULV))中，设置输出值(MV)的最大输出值。

7. 输出值下限(LLV)的设置

在(s3)+27(输出值下限(LLV))中，设置输出值(MV)的最小输出值。

8. PV值阈值(滞后)宽度(SHPV)的设置

设置(s3)+25(PV值阈值(滞后)宽度(SHPV))。

9. 目标值(SV)的设置

在PID运算指令的(s1)中设置目标值(SV)。

10. 自动调谐的开始

将PID运算指令的启动触点置为ON时，根据测定值(PV)开始自动调谐。

在调谐完成的时刻，(s3)+1(动作设置(ACT))的自动调谐标志(位4、位6)将变为OFF。

步响应法

■自动调谐(步响应法)中设置的控制用数据

- 动作方向(正动作/逆动作): (s3)+1 位0
- 比例增益(K_p): (s3)+3
- 积分时间(T_I): (s3)+4
- 微分时间(T_D): (s3)+6

■自动调谐步骤

1. 自动调谐用输出值的设置

将自动调谐用输出值传送到输出值(MV)中。

自动调谐用输出值, 应对输出设备设置可输出最大值 $\times 0.5\sim 1$ 的值。

2. 自动调谐中不能设置的数据的设置

对自动调谐中不能设置的下述项目根据系统进行设置。

设置项目	备注	
(s1)	目标值(SV)	应与测定值(PV)的差设置为150或其以上。 ^{*1}
(s3)	采样时间(T_S)	设置为1000ms或其以上。 ^{*2}
(s3)+2	输入滤波器(α)	—
(s3)+5	微分增益(K_D)	设置输入滤波器的情况下, 微分增益通常设置为“0”。
其它		根据需要进行设置。

*1 关于目标值(SV)的设置值与测定值(PV)的差

自动调谐开始时的测定值与目标值的差如果未达到150, 将无法正确进行自动调谐。因此, 不足150的情况下, 应设置自动调谐用目标值。

自动调谐完成后, 应重新设置目标值。

*2 关于采样时间(T_S)的设置时间

自动调谐时的采样时间(T_S)必须在1000ms或其以上。

此外, 应将该采样时间设置为足够大于输出变化周期的时间。

3. 自动调谐的开始

将(s3)+1(动作设置(ACT))的位4置为ON时, 自动调谐将开始。

从自动调谐开始时的测定值至目标值的变化量达到1/3以上时, 自动调谐完成, (s3)+1(动作设置(ACT))的位4将自动变为OFF。

自动调谐时的注意事项

- 测定值(PV)不变化时的程序对策

由于模拟输入断线等原因, 测定值(PV)不正常变化的情况下, 自动调谐将不结束。应通过导入输入值及自动调谐开始后的经过时间监视顺控程序进行检测, 避免此现象。

23.2 PID运算指令

PID

根据 (s1)、(s2)、(s3) 中设置的值执行PID运算，并将各采样时间的运算结果存储到 (d) 中。

限制事项

使用时，应确认控制器及工程工具的版本。

📖 所使用的控制器的用户手册

梯形图	ST
	ENO:=PID(EN, s1, s2, s3, d);

FBD/LD

■执行条件

指令	执行条件
PID	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储目标值(SV)的软元件	-32768~32767	带符号BIN16位	ANY16
(s2)	存储测定值(PV)的软元件	-32768~32767	带符号BIN16位	ANY16
(s3)	存储控制用数据的起始软元件	—	带符号BIN16位	ANY16
(d)	存储输出值(MV)的软元件	-32768~32767	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字			双字		间接指定	常数				其它
	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□\G□	Z	LT、LST、LC	LZ		K	H	E	\$	
(s1)	—	—	○	○	—	—	—	○	—	—	—	—	—
(s2)	—	—	○	○	—	—	—	○	—	—	—	—	—
(s3)	—	—	○	—	—	—	—	○	—	—	—	—	—
(d)	—	—	○	○	—	—	—	○	—	—	—	—	—

功能

设置目标值(s1)、测定值(s2)、控制用数据(s3)~(s3)+6后，执行程序时，在各采样时间(s3)将运算结果(MV)存储到输出值(d)中。关于详细情况，请参阅下述章节。

📖 1000页 概要

■各自变量的设置项目

设置项目	内容	占用点数
(s1)	目标值 (SV) 设置目标值 (SV)。 ■自动调谐：极限循环法的情况下 自动调谐用的目标值与进行PID控制时的目标值不相同的情况下，应设置加上偏置值后的值，在自动调谐标志变为OFF的时刻，设置实际的目标值。	1点
(s2)	测定值 (PV) 设置PID运算的输入值。	1点
(s3)	控制用数据*1 ■自动调谐：极限循环法的情况下 从(s3)中指定的起始软元件开始占用29点软元件。	29点
	■自动调谐：步响应法的情况下 从(s3)中指定的起始软元件开始占用25点软元件。	25点
(d)	输出值 (MV) ■PID控制(通常处理)的情况下 执行指令前由用户侧设置初始输出值。执行指令后存储运算结果。 ■自动调谐：极限循环法的情况下 自动调谐中自动地输出输出上限值(ULV)或输出下限值(LLV)，自动调谐结束后指定的输出值(MV)将被设置。 ■自动调谐：步响应法的情况下 执行指令前由用户侧设置步输出值。自动调谐中，输出值(MV)在PID指令侧不更改。	1点

*1 不使用自动调谐的情况下，将占用与使用步响应法时相同的点数。

■控制用数据的设置项目

操作数: (s3)				
软元件	项目	内容	备注	
+0	采样时间(T_S)	1~32767[ms]	不能以短于运算周期的值执行。	
+1	动作设置(ACT)	位0	0: 正动作 1: 逆动作	动作方向指定
		位1	0: 无输入变化量报警 1: 输入变化量报警有效	—
		位2	0: 无输出变化量报警 1: 输出变化量报警有效	请勿同时将位2与位5置为ON。
		位3	不能使用	—
		位4	0: 自动调谐不动作 1: 执行自动调谐	—
		位5	0: 无输出值上下限设置 1: 输出值上下限设置有效	请勿同时将位2与位5置为ON。
		位6	0: 步响应法 1: 极限循环法	自动调谐模式选择
	位7~位15	不能使用	—	
+2	输入滤波器常数(α)	0~99[%]	指定了0的情况下, 将变为无输入滤波器。	
+3	比例增益(K_P)	1~32767[%]	—	
+4	积分时间(T_I)	1~32767[$\times 100$ ms]	指定了0的情况下, 将作为 ∞ 处理。(无积分)	
+5	微分增益(K_D)	0~200[%]	指定了0的情况下, 将变为无微分增益。	
+6	微分时间(T_D)	1~32767[$\times 10$ ms]	指定了0的情况下, 将变为无微分。	
+7~+19	由于由PID运算的内部处理所占用, 禁止数据更改			
+20*1	输入变化量(增侧)报警设置值	0~32767	动作设置(ACT): (s3)+1 位1=1时有效	
+21*1	输入变化量(减侧)报警设置值	0~32767	动作设置(ACT): (s3)+1 位1=1时有效	
+22*1	输出变化量(增侧)报警设置值	0~32767	动作设置(ACT): (s3)+1 位2=1, 位5=0时有效	
	输出上限设置值	-32768~32767	动作设置(ACT): (s3)+1 位2=0, 位5=1时有效	
+23*1	输出变化量(减侧)报警设置值	0~32767	动作设置(ACT): (s3)+1 位2=1, 位5=0时有效	
	输出下限设置值	-32768~32767	动作设置(ACT): (s3)+1 位2=0, 位5=1时有效	
+24*1	报警输出	位0	0: 输入变化量(增侧)未溢出 1: 输入变化量(增侧)溢出	动作设置(ACT): (s3)+1 位1=1或位2=1时有效
		位1	0: 输入变化量(减侧)未溢出 1: 输入变化量(减侧)溢出	—
		位2	0: 输出变化量(增侧)未溢出 1: 输出变化量(增侧)溢出	—
		位3	0: 输出变化量(减侧)未溢出 1: 输出变化量(减侧)溢出	—
+25*2	PV值阈值(滞后)宽度(SHPV)	根据测定值(PV)的波动设置	动作设置(ACT): (s3)+1	
+26*2	输出值上限(ULV)	输出值(MV)的最大输出值(ULV)设置	位6=1(选择极限循环法)时占用	
+27*2	输出值下限(LLV)	输出值(MV)的最小输出值(LLV)设置		
+28*2	从调谐循环结束到PID控制开始的等待设置参数(K_W)	-50~32717[%]		

*1 (s3)+1的动作设置(ACT)的位1=1、位2=1或位5=1时占用。

*2 (s3)+1的动作设置(ACT)的位6=1时(选择极限循环法时)占用。

出错

错误代码 (SDO)	内容
11A0H	采样时间(T_S)中指定了超出范围($T_S \leq 0$)的值时。
11A1H	输入滤波器常数(α)中指定了超出范围($\alpha < 0$ 或 $100 \leq \alpha$)的值时。
11A2H	比例增益(K_P)中指定了超出范围($K_P < 0$)的值时。
11A3H	积分时间(T_I)中指定了超出范围($T_I < 0$)的值时。
11A4H	微分增益(K_D)中指定了超出范围($K_D < 0$ 或 $201 \leq K_D$)的值时。
11A5H	微分时间(T_D)中指定了超出范围($T_D < 0$)的值时。
11A6H	采样时间(T_S)中指定了运算周期以下的值时。
11A7H	测定值变化量(ΔPV)溢出时。
11A8H	偏差(EV)溢出时。
11A9H	积分计算值溢出时。
11AAH	微分增益(K_D)的值溢出时。
11ABH	微分计算值溢出时。
11ACH	PID运算结果溢出时。
11ADH	输出上限设置值中指定了小于输出下限设置值的值时。
11AEH	输入变化量报警设置值或输出变化量报警设置值中指定了小于0的值时。
11AFH	<p>■步响应法</p> 自动调谐结果不佳时。 <ul style="list-style-type: none"> 自动调谐(步响应法)开始时的偏差变为150以下。 在自动调谐(步响应法)结束时的偏差为开始时的偏差的1/3或其以上的情况下结束。
11B0H	<p>■步响应法</p> 自动调谐动作方向不一致时。 <ul style="list-style-type: none"> 通过自动调谐(步响应法)开始时的目标值与测定值的关系推测的动作方向与输出值的动作方向不一致。
11B1H	<p>■步响应法</p> 自动调谐动作不佳时。 <ul style="list-style-type: none"> 输入值(PV)不正常变化, 自动调谐(步响应法)未正确动作。
11B2H	<p>■极限循环法</p> 自动调谐(极限循环法)用的输出值上限(ULV)中指定了输出值下限(LLV)或其以下的值时。
11B3H	<p>■极限循环法</p> 自动调谐(极限循环法)用的PV值阈值(滞后)宽度(SHPV)中指定了超出(SHPV < 0)的值时。
11B4H	<p>■极限循环法</p> 自动调谐(极限循环法)中使用的系统区域被改写时。
11B5H	<p>■极限循环法</p> 由于自动调谐(极限循环法)的测定时间超时, τ 及 τ_{on} 的时间未能正常获取时。($\tau_{on} > \tau$ 、 $\tau_{on} < 0$ 、 $\tau < 0$)
11B6H	<p>■极限循环法</p> 通过自动调谐(极限循环法)计算出的比例增益(K_P)溢出时。
11B7H	<p>■极限循环法</p> 通过自动调谐(极限循环法)计算出的积分时间(T_I)为0~32767以外的结果时。
11B8H	<p>■极限循环法</p> 通过自动调谐(极限循环法)计算出的微分时间(T_D)为0~32767以外的结果时。

24 PID控制指令

用于进行PID控制的指令。

24.1 概要

以下介绍PID控制指令中的运算方式、PID控制步骤、其它功能有关内容。

要点

PID控制指令中，分别有不完全微分及完全微分的指令。

对于不完全微分，在微分项的输入放入了一次延迟滤波器的PID控制中，在下述情况下使用时将有效。

- 容易受到高频噪声影响的控制
- 完全微分方式中有步状态变化时，未被施加可使操作端动作的有效能源时

完全微分是原样不变地使用微分项的输入的PID控制。

运算方式

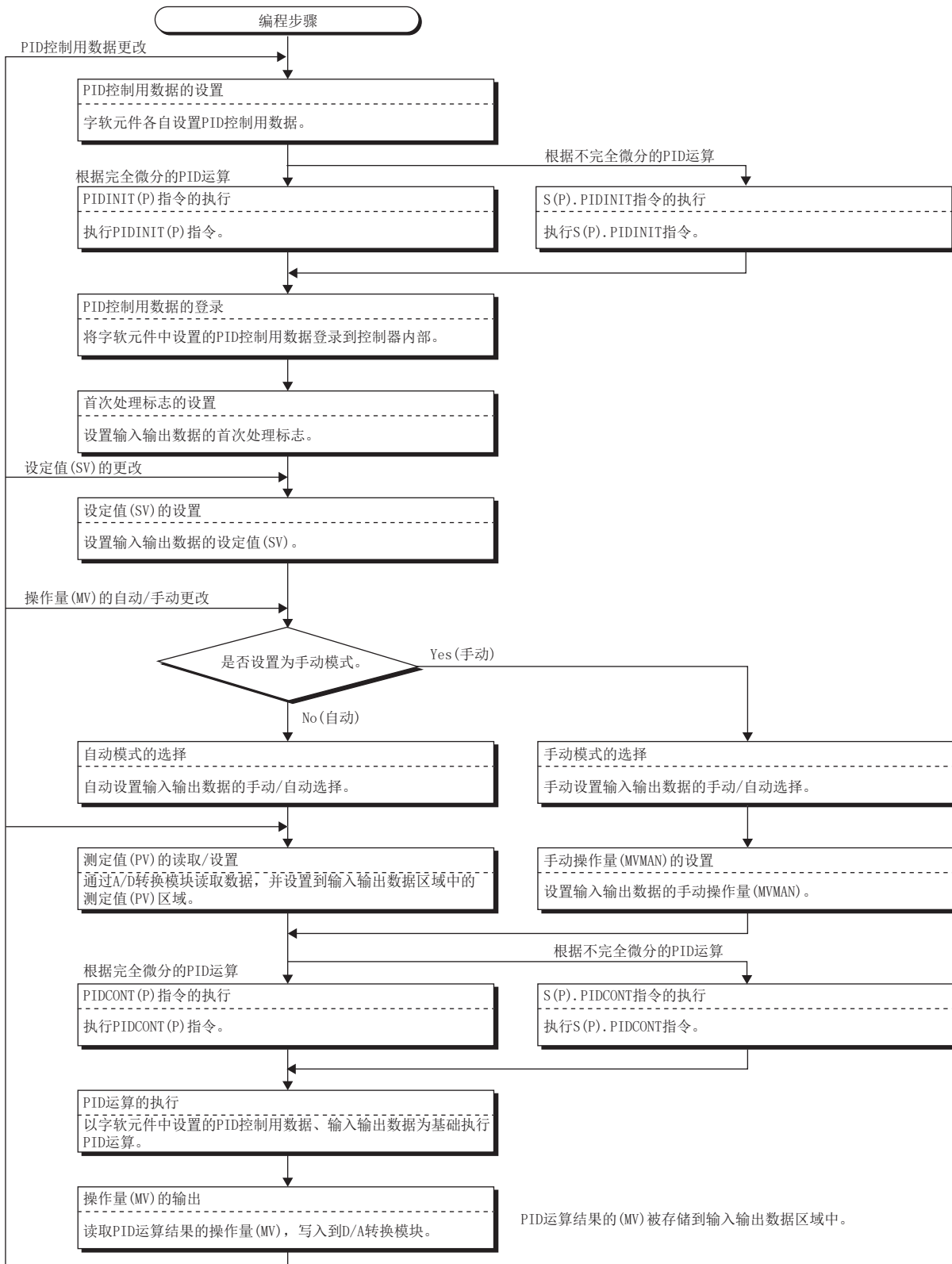
PID控制指令的PID控制中运算方式为速度型・测定值微分型。

在速度型・测定值微分型中进行以下控制。

运算方式	内容
速度型运算	是通过PID运算对操作量(MV)的变化进行计算的方式。实际的操作量为各采样周期中计算的操作量变化的累计值。
测定值微分型	是通过PID运算在微分项中使用测定值(PV)进行运算的方式。微分项中不使用偏差，因此可以减少设定值更改引起偏差变化时微分动作导致的输出骤变。

PID控制步骤

使用了用于进行PID控制的软件元件的编程步骤如下所示。



要点

- 对于PID控制用数据，可以在程序的各扫描中登录或更改。但是，进行了登录或更改的情况下，必须执行PIDINIT指令。如果未执行PIDINIT指令，在执行PIDCONT指令时登录或更改的数据将不被反映。
- 通过PIDPRMW指令在每个环路中更改PID控制用数据的情况下，无需执行PIDINIT指令。

PID控制用数据

是在PID运算中用于设置基准值的数据。通过PIDCONT指令开始PID运算之前，通过PIDINIT指令登录到控制器内部。此外，可以设置为字软元件的任意编号。但是，需要设置相当于使用环路的所有数据连续的软元件编号。

关于PID控制数据的分配，请参阅下述内容。

- 不完全微分：☞ 1027页 S(P).PIDINIT
- 完全微分：☞ 1039页 PIDINIT(P)

在PID控制用数据中，有全部环路通用设置的数据及在各环路进行设置的数据这2种。

数据的类型	项目	内容	设置范围	设置数据超出指定范围时的处理	
通用设置数据	使用环路数	是执行PID运算的环路数的设置。	1~32	变为错误状态，全部环路不执行PID运算。	
	1个扫描的执行环路数	是有多个达到采样周期的环路时，1次的PID运算中执行多少个环路的设置。	1~32		
各环路的设置数据	运算公式选择	是进行PID运算式的正动作还是逆动作的选择。	0: 正动作 1: 逆动作	变为错误状态，不执行相应环路的PID运算。	
	采样周期(T_S)	是进行PID运算的周期的设置。	1~6000(单位: 10ms)		
	比例常数(K_P)	是PID运算的比。	1~10000(单位: 0.01)		
	积分常数(T_I)	是表示积分动作(I动作)效果大小的常数。增大积分常数时，操作量的变化将变缓。	1~32767(单位: 100ms)		
	微分常数(T_D)	是表示微分动作(D动作)效果大小的常数。增大微分常数时，控制对象的轻微变化将引起操作量较大的变化。	0~30000(单位: 10ms)		
	滤波器系数(α)	是对测定值(来自于A/D转换模块的输入)施加多大程度滤波器的设置。越趋近于0则滤波器效果越小。	1~100		
	操作量下限值(MVLL)	是自动模式时，通过PID运算计算的操作量的下限值的设置。操作量低于操作量下限值(MVLL)时，操作量下限值(MVLL)将变为操作量。	-50~2050* ¹ -32768~32767* ²		在“有PID极限制”中，操作量下限值(MVLL)或操作量上限值(MVHL)的值超出指定范围时，将被转换为下述值。 • 小于-50时，变为-50。 • 超过2050时，变为2050。
	操作量上限值(MVHL)	是自动模式时，通过PID运算计算的操作量的上限值的设置。操作量超过操作量上限值(MVHL)时，操作量上限值(MVHL)将变为操作量。	-50~2050* ¹ -32768~32767* ²		
	操作量变化率限制值(ΔMVL)	是上次与本次的操作量中变化量的限制值的设置。操作量的变化量超过限制值时，报警用软元件的b1将变为1。 但是，操作量的变化量无限制。即使操作量的变化量超过限制值的情况下，也仍然原样不变地作为操作量的变化量使用，计算出操作量。	0~2000* ¹ 0~32767* ²		在“有PID极限制”中，操作量变化率限制值(ΔMVL)或测定值变化率限制值(ΔPVL)的值超出指定范围时，将被转换为下述值。 • 小于0时，变为0。 • 超过2000时，变为2000。
	测定值变化率限制值(ΔPVL)	是上次与本次的测定值中变化量的限制值的设置。测定值的变化量超过限制值时，报警用软元件的b0将变为1。 但是，测定值的变化量无限制。即使测定值的变化量超过限制值的情况下，也仍然原样不变地作为变化量使用，进行PID运算。	0~2000* ¹ 0~32767* ²		
微分增益(K_D)* ³	是对微分动作附加时间宽度(动作延迟)的设置。值越大则时间宽度越小，动作将趋近于完全微分。	0~32767(单位: 0.01)	变为错误状态，不执行相应环路的PID运算。		

*1 是在“有PID极限制”设置时。

*2 是在“无PID极限制”设置时。

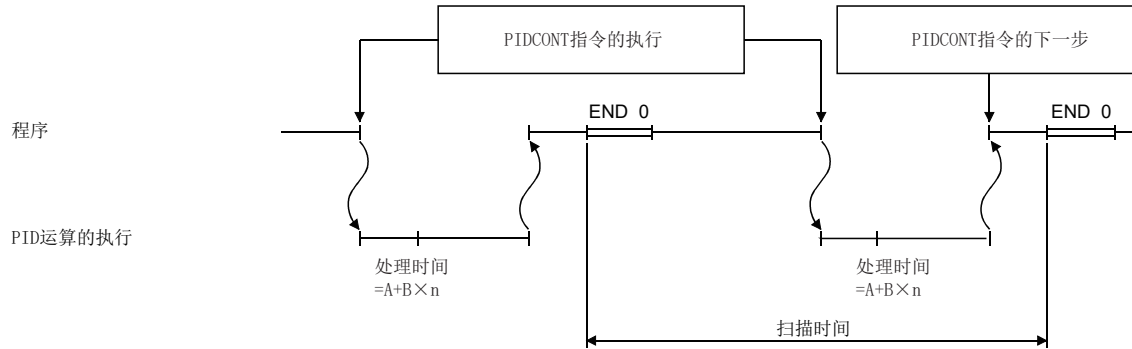
*3 仅为不完全微分时。

■使用环路数及1个扫描的执行环路数

使用环路数是使其执行PID运算的环路数。执行PIDCONT指令时进行设置的环路数的采样周期的计测，进行达到采样周期的环路的PID运算。

执行了PIDCONT指令情况下的处理时间与执行PID运算的环路数成正比。

1个扫描的执行环路数的含义是，有多个达到采样周期的环路时，1个扫描中进行多少个环路的PID运算的设置。如果预先指定1个扫描的执行环路数，即使执行PIDCONT指令时达到采样周期的环路数较多的情况下，也只执行设置的1个扫描的执行环路数的PID运算，剩余的将在下一个扫描中执行。



A: 采样周期的计测等的固定时间

B: 用于进行环路的PID运算的时间

n: 环路数

要点

达到采样周期的环路多于1个扫描的执行环路数情况下的优先顺序如下所示。

- 环路No. 较小编号的优先顺序越高。
- 有上一个扫描中未执行PID运算的环路及已执行过运算的环路的情况下，上一个扫描中未执行PID运算的环路将优先。

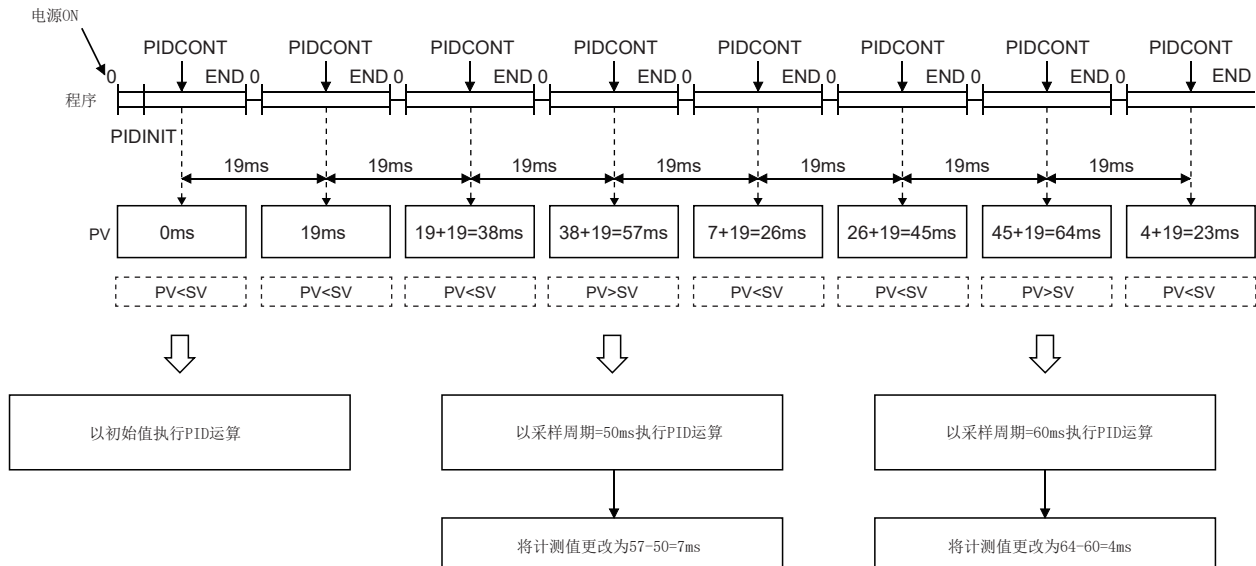
■采样周期

采样周期是进行PID运算的周期。每次执行PIDCONT指令时将1个扫描的计测时间加到上一个为止的计测时间中。相加后的值大于等于设置的采样周期时，进行相应环路的PID运算。

用于PID运算的采样周期使用10ms单位的执行值。

例

采样周期的设定值为50ms的情况下



PV: 计测值

SV: 设定值

限制事项

采样周期的计测是在执行PIDCONT指令时进行，因此采样周期中不能设置小于程序扫描时间的值。设置了小于扫描时间的值的情况下，将以扫描时间值进行PID运算。

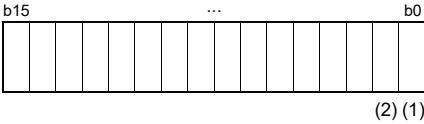
输入输出数据

输入输出数据是设定值(SV)、测定值(PV)等用于进行PID运算而设置的输入数据及运算结果等的输出数据。此外，可以指定字软元件的任意编号。但是，需要设置相当于使用环路的所有数据连续的软元件编号。

关于输入输出数据的分配，请参阅下述内容。

- 不完全微分：☞ 1030页 S(P).PIDCONT
- 完全微分：☞ 1041页 PIDCONT(P)

输入输出数据区域中，有分配至各环路中的以下项目及用于进行PID运算的系统所使用的工作区域。

项目	内容	设置范围	备注
设定值	SV 是PID控制的目标值。	0~2000* ¹ -32768~32767* ²	在“有PID极限制”中，设定值(SV)的值超出指定范围时，将被转换为下述值。 • 小于0时，变为0。 • 超过2000时，变为2000。
测定值	PV 是从控制对象反馈至A/D转换模块中的数据。	-50~2050* ¹ -32768~32767* ²	在“有PID极限制”中，测定值(PV)的值超出指定范围时，将被转换为下述值。 • 小于-50时，变为-50。 • 超过2050时，变为2050。
自动操作量	MV 是通过PID运算计算出的操作量。从D/A转换模块输出到控制对象。	-50~2050* ¹ -32768~32767* ²	—
滤波器后的测定值	PVf 是通过以下运算公式计算出的测定值。 $PV_{fn} = PV_n + \alpha(PV_{fn-1} - PV_n)$	-50~2050* ¹ -32768~32767* ²	—
手动操作量	MV _{MAN} 存储手动时从D/A转换模块输出的数据。	-50~2050* ¹ -32768~32767* ²	在“有PID极限制”中，手动操作量(MV _{MAN})的值超出设置范围时，将被转换为下述值。 • 小于-50时，变为-50。 • 超过2050时，变为2050。
手动/自动选择	MAN/AUTO 选择至D/A转换模块的输出数据是手动操作量还是自动操作量。手动时，自动操作量不变化。	0: 自动操作量 1: 手动操作量	设置为0、1以外时将变为错误，不执行相应环路的PID运算。
报警	ALARM 用于判别自动操作量(MV)、测定值(PV)的变化率是否超出极限值范围。进行了1次设置时，在复位之前将被保持。 <div style="text-align: center;">  <p>(2) (1)</p> </div> (1) 测定值(PV)超出极限范围时，b0将变为1。 (2) 自动操作量(MV)超出极限范围时，b1将变为1。	—	—

*1 是在“有PID极限制”设置时。

*2 是在“无PID极限制”设置时。

其它功能

在通过PID控制指令进行的PID控制中，自动进行如下所示的无冲击切换、操作量上下限制器控制。

无冲击切换

无冲击是模式切换(手动↔自动)时连续控制操作量(MV)的功能。进行模式切换时，按以下方式在自动模式的操作量区域与手动模式的操作量区域之间进行数据传送。

- 从手动模式切换为自动模式时：将手动模式的操作量传送至自动模式的操作量区域中。
- 从自动模式切换为手动模式时：将自动模式的操作量传送至手动模式的操作量区域中。

模式切换通过输入输出数据区域进行。(☞ 1022页 输入输出数据)

要点

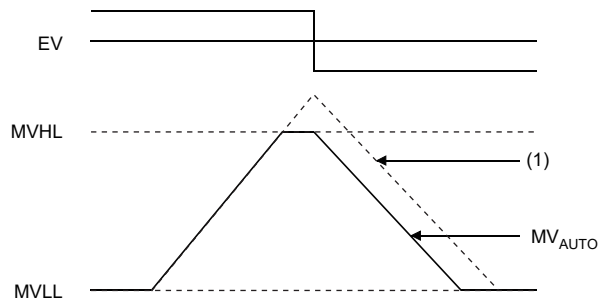
PID控制的自动模式、手动模式如下所示。

- 自动模式：是根据通过PID控制指令进行PID运算而计算出的操作量，对控制对象进行控制的模式。
- 手动模式：是在不通过PID控制指令进行PID运算的状况下根据计算出的操作量对控制对象进行控制的模式。设置为手动模式的环路在各采样周期将测定值(PV)存储到设定值区域中。

操作量上下限制器控制

操作量上下限制器控制是对通过PID运算计算出的操作量的上限或下限进行限制的功能。该功能仅在自动模式时才有效，手动模式时不执行。

通过设置操作量上限值(MVHL)、操作量下限值(MVLL)，可将PID运算计算出的操作量限制在下限值至上限值的范围内。在操作量上下限制器功能中，其动作如下所示。



(1) 未施加限制器控制情况下的 MV_{AUTO}

对于操作量上限值(MVHL)、操作量下限值(MVLL)，可在各环路中设置-50~2050或任意范围。默认值如下所示。

- 操作量上限值：2000
- 操作量下限值：0

对于操作量上限值(MVHL)、操作量下限值(MVLL)，如果设置为(上限值)<(下限值)将变为错误。

手动模式时将测定值变为设定值的传送功能

手动模式时也执行PIDCONT指令。手动模式时，通过SM792(PID无冲击处理(完全微分用))、SM794(PID无冲击处理(不完全微分用))的ON/OFF，可以选择执行PIDCONT指令时是否将从A/D转换模块获取的测定值传送至设定值。默认为OFF。此外，通过PIDCONT指令设定值及测定值将被存储到输入输出数据区域中指定的软元件或标签中。

SM792/SM794	动作内容
OFF	<ul style="list-style-type: none"> 执行PIDCONT指令时，将测定值传送至设定值。 手动模式→自动模式的切换时，可继续手动模式时的操作量输出。 如果在切换为自动模式后更改设定值，则可以执行从输出的操作量到设定值的控制。
ON	<ul style="list-style-type: none"> 执行PIDCONT指令时，不将测定值传送至设定值。 如果手动模式→自动模式的切换，则可以执行从手动模式时的操作量输出到设定值的控制。 至自动模式的切换前，应存储设定值。

要点

根据SM792(PID无冲击处理(完全微分用))、SM794(PID无冲击处理(不完全微分用))的ON/OFF，手动模式→自动模式切换时，有以下不同。

- OFF时，将测定值传送至设定值，因此模式切换时测定值与设定值无差异，不发生操作量的急剧变化。另一方面，由于切换后的设定值与自动模式中的目标值不同，因此应通过程序使设定值逐渐变为目标值。
- ON时，不将测定值传送至设定值，因此模式切换时测定值与设定值有差异。切换时该差异较大的情况下，有可能发生操作量的急剧变化，因此应在测定值充分趋近于设定值时进行切换的系统中使用。无需通过程序使设定值逐渐变化，可以立即通过自动模式实施。

PID控制用数据、输入输出数据的设置范围更改功能

可以将PID控制用数据、输入输出数据的设置范围通过设置更改为任意范围。

使设置生效的情况下，应将SD792~SD793(PID极限限制设置(完全微分用))、SD794~SD795(PID极限限制设置(不完全微分用))的相应环路的位设置为ON。

不完全微分	完全微分	动作内容																																
SD794	SD792	<table border="1"> <tr> <td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <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> <p>b0: 环路1 b1: 环路2 ⋮ b14: 环路15 b15: 环路16</p>	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	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
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																			
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	<table border="1"> <tr> <td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <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> <p>b0: 环路17 b1: 环路18 ⋮ b14: 环路31 b15: 环路32</p>	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	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
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																			
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																			

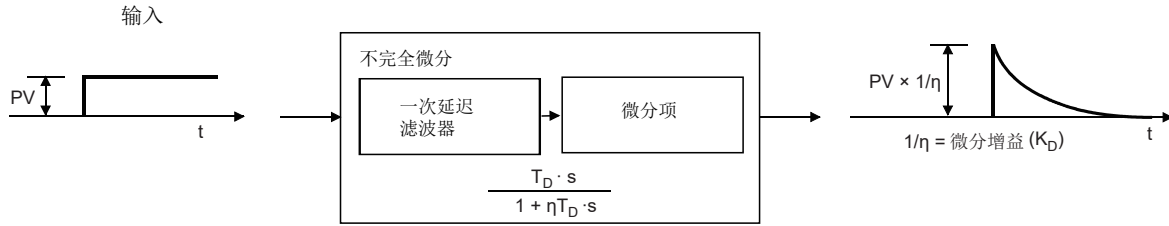
0: 有PID极限限制(默认)

1: 无PID极限限制

24.2 PID控制指令(不完全微分)

不完全微分是在微分项的输入中置入一次延迟滤波器的PID控制。不完全微分在以下情况下有效。

- 易于受到高频噪声影响的控制的情况下
- 完全微分方式中，有步状变化时无法赋予使操作端动作的有效能量的情况下

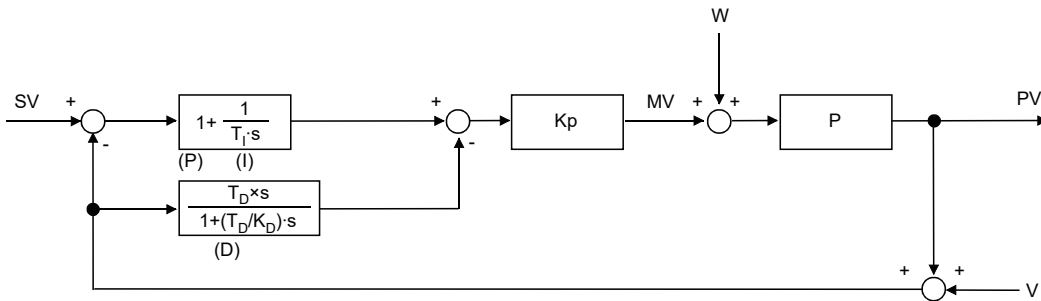


通过不完全微分进行的PID控制指令的性能规格如下所示。

项目		“有PID极限限制”时	“无PID极限限制”时
PID控制环路数	—	最大32环路	
采样周期	T_S	0.01~60.00s	
PID运算方式	—	测定值微分型不完全微分(正动作/逆动作)	
PID常数设置范围	比例常数	K_P	0.01~100.00
	积分常数	T_I	0.1~3000.0s
	微分常数	T_D	0.00~300.00s
	微分增益	K_D	0.00~300.00
设定值设置范围	SV	0~2000	-32768~32767
测定值设置范围	PV	-50~2050	-32768~32767
操作量输出范围	MV		

PID运算的块图及运算公式如下所示。

- 不完全微分的PID运算块



K_P : 增益
 W : 干扰
 P : 控制对象
 V : 检测杂音

• 运算公式

正动作/逆动作	运算公式
正动作	$EV_n = PV_{fn} - SV$ $\Delta MV = K_p \{ (EV_n - EV_{n-1}) + \frac{T_S}{T_I} \cdot EV_n + D_n \}$ $D_n = \frac{T_D}{T_S + \frac{T_D}{K_D}} (PV_{fn} - 2PV_{fn-1} + PV_{fn-2}) + \frac{\frac{T_D}{K_D}}{T_S + \frac{T_D}{K_D}} \cdot D_{n-1}$ $MV_n = \Sigma \Delta MV$
逆动作	$EV_n = SV - PV_{fn}$ $\Delta MV = K_p \{ (EV_n - EV_{n-1}) + \frac{T_S}{T_I} \cdot EV_n + D_n \}$ $D_n = \frac{T_D}{T_S + \frac{T_D}{K_D}} (-PV_{fn} + 2PV_{fn-1} - PV_{fn-2}) + \frac{\frac{T_D}{K_D}}{T_S + \frac{T_D}{K_D}} \cdot D_{n-1}$ $MV_n = \Sigma \Delta MV$

运算公式中的符号含义如下所示。

符号	含义
EV_n	本次样本时的偏差
EV_{n-1}	1周期前的偏差
SV	设定值
PV_{fn}	本次样本时的测定值(滤波器后)
PV_{fn-1}	1周期前的测定值(滤波器后)
PV_{fn-2}	2周期前的测定值(滤波器后)
ΔMV	输出变化量
MV_n	本次的操作量
D_n	本次的微分项
D_{n-1}	1周期前的微分项
T_S	采样周期
K_p	比例常数
T_I	积分常数
T_D	微分常数
K_D	微分增益

本次样本时的测定值(滤波器后) (PV_{fn}) 是将输入数据的测定值通过下式进行了运算后的值。未设置输入数据的滤波器系数的情况下, 将变为与输入数据的测定值(PV)相同的值。

$$PV_{fn} = PV_n + \alpha (PV_{fn-1} - PV_n)$$

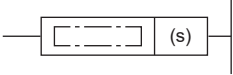
PV_n : 本次样本时的测定值, α : 滤波器系数, PV_{fn-1} : 1周期前的测定值(滤波器后)

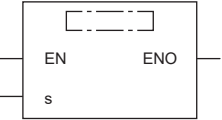
此外, PV_{fn} 将被存储到输入输出数据区域中。(☞ 1022页 输入输出数据)

PID控制用数据的设置

S(P).PIDINIT

将指定的软元件编号及其以后设置的使用环路数的PID控制数据批量登录到控制器内部。

梯形图	ST
	ENO:=S_PIDINIT(EN, s); ENO:=SP_PIDINIT(EN, s);

FBD/LD


■执行条件

指令	执行条件
S.PIDINIT	
SP.PIDINIT	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	设置了PID控制用数据的起始软元件	—	字	ANY16* ¹
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 将(s)中指定的软元件编号及其以后设置的使用环路数的PID控制用数据批量登录到控制器内部，置为可PID控制状态。
(☞ 1019页 PID控制用数据)
- PID控制用数据的分配如下所示。

项目	PID控制用数据	
全部环路通用	(s)+0	使用环路数
	(s)+1	1个扫描的执行环路数
No. 1环路用(14字)	(s)+2	运算公式选择
	(s)+3	采样周期(T_S)
	(s)+4	比例常数(K_P)
	(s)+5	积分常数(T_I)
	(s)+6	微分常数(T_D)
	(s)+7	滤波器系数(α)
	(s)+8	操作量下限值(MVLL)
	(s)+9	操作量上限值(MVHL)
	(s)+10	操作量变化率限制值(Δ MVL)
	(s)+11	测定值变化率限制值(Δ PVL)
	(s)+12	固定为0(指定了0以外的情况下将变为错误)
	(s)+13	微分增益(K_D)
	(s)+14	固定为0(指定了0以外的情况下将变为错误)
	(s)+15	固定为0(指定了0以外的情况下将变为错误)
⋮	⋮	⋮
No. n环路用(14字)*1	(s)+(m+0)	运算公式选择
	(s)+(m+1)	采样周期(T_S)
	(s)+(m+2)	比例常数(K_P)
	(s)+(m+3)	积分常数(T_I)
	(s)+(m+4)	微分常数(T_D)
	(s)+(m+5)	滤波器系数(α)
	(s)+(m+6)	操作量下限值(MVLL)
	(s)+(m+7)	操作量上限值(MVHL)
	(s)+(m+8)	操作量变化率限制值(Δ MVL)
	(s)+(m+9)	测定值变化率限制值(Δ PVL)
	(s)+(m+10)	固定为0(指定了0以外的情况下将变为错误)
	(s)+(m+11)	微分增益(K_D)
	(s)+(m+12)	固定为0(指定了0以外的情况下将变为错误)
(s)+(m+13)	固定为0(指定了0以外的情况下将变为错误)	

*1 $m=(n-1) \times 14+2$

(1) 固定为“0”。指定了“0”以外的情况下，将变为错误状态。

- PID控制用数据的设置中使用的软元件点数通过下式计算。

软元件点数=2+14×n (n: 使用环路数)

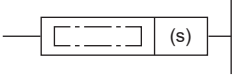
- 各数据应以BIN值进行设置。
- 使用环路的软元件点数超过指定软元件的最终软元件编号时将变为错误且不进行处理。
- 1个扫描中在多个位置执行了S(P).PIDINIT指令的情况下，最靠近S(P).PIDCONT指令执行的S(P).PIDINIT指令的设定值将有效。
- S(P).PIDINIT指令的执行必须在执行S(P).PIDCONT指令之前。未执行S(P).PIDINIT指令的情况下，不能进行PID控制。

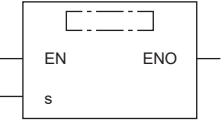
错误代码 (SD0)	内容
3285H	(s) 中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> • PID控制用数据中设置的值超出了允许设置范围时。 • (使用环路数) < (1个扫描的执行环路数) 时。 • (操作量上限值) < (操作量下限值) 时。 • PID控制用数据的固定为“0”的区域不是“0”时。

PID运算



S(P).PIDCONT

通过执行指令的ON, 进行采样周期的计测及PID运算。

梯形图	ST
	ENO:=S_PIDCONT(EN, s); ENO:=SP_PIDCONT(EN, s);

FBD/LD


■执行条件

指令	执行条件
S.PIDCONT	
SP.PIDCONT	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	分配到输入输出数据区域中的起始软元件	—	字	ANY16*1
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下, 应在确保动作所需区域的前提下定义数组, 指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 执行S(P).PIDCONT指令时，进行采样周期的计测及PID运算。
- S(P).PIDCONT指令，以(s)中指定的软元件编号及其以后设置的输入输出数据区域的设定值(SV)、测定值(PV)为基础进行PID运算，将运算结果存储到输入输出数据区域的自动操作量(MV)区域中。
- PID运算，是在经过采样周期的设置时间后的首个S(P).PIDCONT指令执行时进行。
- PID控制过程中必须将控制指令置为ON，并使S(P).PIDCONT指令在每个扫描中执行。未在每个扫描中执行的情况下，将无法以正常的采样周期进行PID运算。此外，1个扫描中不能多次执行S(P).PIDCONT指令。1个扫描中执行多次S(P).PIDCONT指令的情况下，将无法以正常的采样周期进行PID运算。
- S(P).PIDCONT指令不能记述到中断程序中使用。将S(P).PIDCONT指令记述到中断程序中的情况下，将无法以正常的采样周期进行PID运算。
- 在(s)中，将指定的软元件编号的起始指定到输入输出数据区域中。(☞ 1022页 输入输出数据)
- 将文件寄存器指定为输入输出数据区域的情况下，请勿对文件寄存器施加存储器保护。施加了存储器保护时不会变为错误，但无法进行正常的PID运算。
- 输入输出数据的分配如下所示。

项目		输入输出数据	
写入		(s)+0	首次处理标志
读取/写入禁止		(s)+1 ⋮ (s)+9	PID控制用工作区域 (用户不能使用)
No. 1环路用输入输出数据区域 (23字)	写入	(s)+10	设定值(SV)
		(s)+11	测定值(PV)
	读取	(s)+12	自动操作量(MV)
		(s)+13	滤波器后的测定值(PVf)
	写入	(s)+14	手动操作量(MV _{MAN})
		(s)+15	手动/自动选择(MAN/AUTO)
	读取/写入	(s)+16	报警(ALARM)
读取/写入禁止	(s)+17 ⋮ (s)+32	No. 1环路用工作区域 (用户不能使用)	
⋮	⋮	⋮	⋮
No. n环路用输入输出数据区域 (23字)*1	写入	(s)+(m)+0	设定值(SV)
		(s)+(m)+1	测定值(PV)
	读取	(s)+(m)+2	自动操作量(MV)
		(s)+(m)+3	滤波器后的测定值(PVf)
	写入	(s)+(m)+4	手动操作量(MV _{MAN})
		(s)+(m)+5	手动/自动选择(MAN/AUTO)
	读取/写入	(s)+(m)+6	报警(ALARM)
读取/写入禁止	(s)+(m)+7 ⋮ (s)+(m)+22	No. n环路用工作区域 (用户不能使用)	

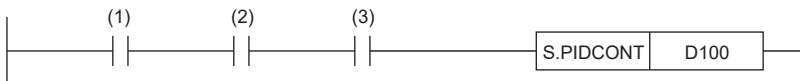
*1 $m=(n-1) \times 23+10$

- 输入输出数据设置中使用的软元件点数由下式计算。

软元件点数=10+23×n(n: 使用环路数)

- 各数据应以BIN值进行设置。
- 首次处理标志是PID运算开始时的处理方法的设置。
- 首次运算处理时，将视为到达设置的采样周期进行运算。
- 首次处理标志为0的情况下，将使用环路数的PID运算处理通过1个扫描进行批量处理。此外，0以外的情况下，将使用环路数的PID运算处理分割为数个扫描进行处理后，从首次处理完成的环路开始依次开始循环。每个扫描的处理环路数将变为设置的1个扫描的执行环路数。

- 输入输出数据区域中处于“写入”的数据，应由用户通过程序进行写入。处于“读取”的数据，由用户通过程序读取使用。对处于“读取/写入禁止”及“读取”的数据进行写入时，将无法正常进行运算，因此请勿进行。但是，从初始状态开始控制的情况下，需要通过程序进行数据清除。
- 使用环路的软元件点数超过指定软元件的最终软元件编号时将变为错误且不进行处理。
- 即使通过手动模式进行了手动操作量(MV_{MAN})的输出的情况下，也应在每个扫描执行S(P).PIDCONT指令。如果不执行S(P).PIDCONT指令，将无法执行无冲击功能。
- S(P).PIDCONT指令，应通过各模块的READY信号采取互锁，以便只有在用于获取测定值(PV)的A/D转换模块以及用于输出操作量(MV)的D/A转换模块正常时才执行指令。



- (1) 控制指令
- (2) A/D转换模块的READY信号
- (3) D/A转换模块的READY信号

如果在各模块异常时执行，将无法正常进行测定值(PV)的获取或正常进行操作量(MV)的输出，因此也无法正常进行PID运算。

出错

错误代码(SD0)	内容
3285H	(s)中指定的输入输出数据区域中设置的数据值超出允许设置范围时。
3292H	在执行S(P).PIDCONT指令之前，未执行S(P).PIDINIT指令时。

指定环路No. 的运算停止

S(P).PIDSTOP

停止指定环路No. 的PID运算。

梯形图	ST
	ENO:=S_PIDSTOP(EN, s); ENO:=SP_PIDSTOP(EN, s);

FBD/LD

■执行条件

指令	执行条件
S.PIDSTOP	
SP.PIDSTOP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	停止的环路No.	1~32	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	—	—	○	○	—	—	—	—

功能

- 停止(s)中指定的环路No. 的PID运算。通过S(P).PIDSTOP指令停止的环路，即使执行S(P).PIDINIT指令也不重启PID运算。
- 停止中运算数据将被保持。

出错

错误代码(SD0)	内容
3285H	(s)中指定了超出允许指定范围的数据时。 <ul style="list-style-type: none"> • 指定的环路No. 不存在 • 指定的值超出了1~32的范围
3292H	在执行S(P).PIDSTOP指令之前，未执行S(P).PIDINIT指令、S(P).PIDCONT指令时。

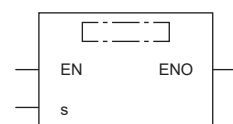
指定环路No. 的运算开始

S(P).PIDRUN

开始指定环路No. 的运算。

梯形图	ST
	ENO:=S_PIDRUN(EN, s); ENO:=SP_PIDRUN(EN, s);

FBD/LD



■执行条件

指令	执行条件
S.PIDRUN	
SP.PIDRUN	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	停止的环路No.	1~32	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	—	—	○	—	—	—	—	○	○	—	—	—	

功能

- 开始(s)中指定的环路No. 的PID运算。是用于对通过S(P).PIDSTOP指令停止PID运算的环路No. 再次执行运算的指令。
- 对正在执行PID运算的环路No. 执行了S(P).PIDRUN指令的情况下，将变为无处理。

出错

错误代码(SD0)	内容
3285H	(s)中指定的环路No. 不存在时。 (s)超出了1~32的范围时。
3292H	在执行S(P).PIDRUN指令之前，未执行S(P).PIDINIT指令、S(P).PIDCONT指令时。

指定环路No. 的参数更改

S(P).PIDPRMW

将指定环路No. 的运算参数，更改为指定的软元件编号及其以后存储的PID控制用数据。

梯形图	ST
	<pre>ENO:=S_PIDPRMW(EN, s1, s2); ENO:=SP_PIDPRMW(EN, s1, s2);</pre>

FBD/LD

■执行条件

指令	执行条件
S.PIDPRMW	
SP.PIDPRMW	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	更改的环路No.	1~32	无符号BIN16位	ANY16
(s2)	存储了更改的PID控制用数据的起始软元件	—	字	ANY16* ¹
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	—	—	○	—	—	—	○	—	—	—	—	—

功能

- 将(s1)中指定的环路No.的运算参数,更改为(s2)中指定的软元件编号及其以后存储的PID控制用数据。
- (s2)中指定的软元件编号及其以后的PID控制用数据的构成如下所示。

PID控制用数据	
(s2)+0	运算公式选择
(s2)+1	采样周期(T_s)
(s2)+2	比例常数(K_p)
(s2)+3	积分常数(T_I)
(s2)+4	微分常数(T_D)
(s2)+5	滤波器系数(α)
(s2)+6	操作量下限值(MVLL)
(s2)+7	操作量上限值(MVHL)
(s2)+8	操作量变化率限制值(ΔMVL)
(s2)+9	测定值变化率限制值(ΔPVL)
(s2)+10	0
(s2)+11	微分增益(K_D)
(s2)+12	0
(s2)+13	0

出错

错误代码(SD0)	内容
3285H	(s1)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none">• 指定的环路No.不存在• 指定的值超出了1~32的范围 (s2)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none">• PID控制用数据超出了允许设置范围• PID控制用数据的(s2)+10、(s2)+12、(s2)+13不为0
3292H	在执行S(P).PIDPRMW指令之前,未执行S(P).PIDINIT指令时。

24.3 PID控制指令(完全微分)

完全微分是原样不变地使用微分项的输入的PID控制。

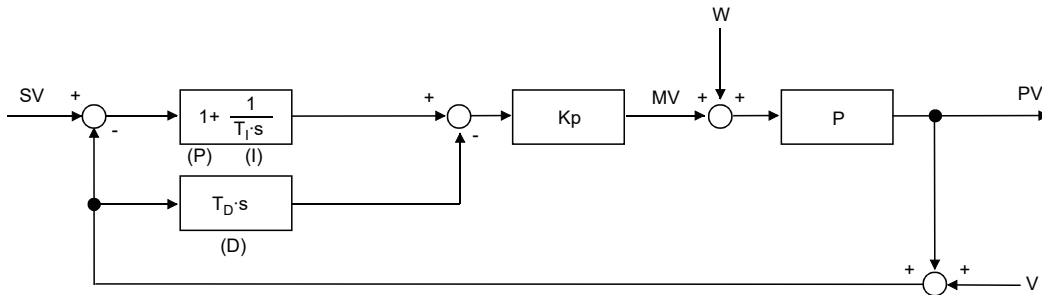


通过完全微分进行的PID控制指令的性能规格如下所示。

项目		“有PID极限限制”时	“无PID极限限制”时
PID控制环路数	—	最大32环路	
采样周期	T_S	0.01~60.00s	
PID运算方式	—	测定值微分型完全微分(正动作/逆动作)	
PID常数设置范围	比例常数	K_P	0.01~100.00
	积分常数	T_I	0.1~3000.0s
	微分常数	T_D	0.00~300.00s
设定值设置范围	SV	0~2000	-32768~32767
测定值设置范围	PV	-50~2050	-32768~32767
操作量输出范围	MV		

PID运算的块图及运算公式如下所示。

- 完全微分的PID运算块



K_P : 增益
 W : 干扰
 P : 控制对象
 V : 检测杂音

• 运算公式

正动作/逆动作	运算公式
正动作	$EV_n = PV_{fn} - SV$ $\Delta MV = K_p \left\{ (EV_n - EV_{n-1}) + \frac{T_S}{T_I} \cdot EV_n + D_n \right\}$ $D_n = \frac{T_D}{T_S} (PV_{fn} - 2PV_{fn-1} + PV_{fn-2})$ $MV_n = \Sigma \Delta MV$
逆动作	$EV_n = SV - PV_{fn}$ $\Delta MV = K_p \left\{ (EV_n - EV_{n-1}) + \frac{T_S}{T_I} \cdot EV_n + D_n \right\}$ $D_n = \frac{T_D}{T_S} (-PV_{fn} + 2PV_{fn-1} - PV_{fn-2})$ $MV_n = \Sigma \Delta MV$

运算公式中的符号含义如下所示。

符号	含义
EV_n	本次样本时的偏差
EV_{n-1}	1周期前的偏差
SV	设定值
PV_{fn}	本次样本时的测定值(滤波器后)
PV_{fn-1}	1周期前的测定值(滤波器后)
PV_{fn-2}	2周期前的测定值(滤波器后)
ΔMV	输出变化量
MV_n	本次的操作量
D_n	本次的微分项
T_S	采样周期
K_p	比例常数
T_I	积分常数
T_D	微分常数

本次样本时的测定值(滤波器后) (PV_{fn}) 是将输入数据的测定值通过下式进行了运算后的值。未设置输入数据的滤波器系数的情况下, 将变为与输入数据的测定值(PV)相同的值。

$$PV_{fn} = PV_n + \alpha (PV_{fn-1} - PV_n)$$

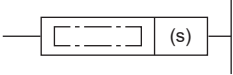
PV_n : 本次样本时的测定值, α : 滤波器系数, PV_{fn-1} : 1周期前的测定值(滤波器后)

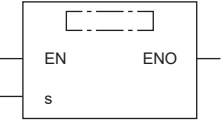
此外, PV_{fn} 将被存储到输入输出数据区域中。(☞ 1022页 输入输出数据)

PID控制用数据的设置


PIDINIT(P)

将指定的软元件编号及其以后设置的使用环路数的PID控制数据批量登录到控制器内部。

梯形图	ST
	ENO:=PIDINIT(EN, s); ENO:=PIDINITP(EN, s);

FBD/LD


■执行条件

指令	执行条件
PIDINIT	
PIDINITP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	设置了PID控制用数据的起始软元件	—	字	ANY16* ¹
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 将(s)中指定的软元件编号及其以后设置的使用环路数的PID控制用数据批量登录到控制器内部，置为可PID控制状态。
(☞ 1019页 PID控制用数据)
- PID控制用数据的分配如下所示。

项目	PID控制用数据	
全部环路通用	(s)+0	使用环路数
	(s)+1	1个扫描的执行环路数
No. 1环路用(10字)	(s)+2	运算公式选择
	(s)+3	采样周期(T_S)
	(s)+4	比例常数(K_P)
	(s)+5	积分常数(T_I)
	(s)+6	微分常数(T_D)
	(s)+7	滤波器系数(α)
	(s)+8	操作量下限值(MVLL)
	(s)+9	操作量上限值(MVHL)
	(s)+10	操作量变化率限制值(Δ MVL)
	(s)+11	测定值变化率限制值(Δ PVL)
	⋮	⋮
No. n环路用(10字)*1	(s)+(m+0)	运算公式选择
	(s)+(m+1)	采样周期(T_S)
	(s)+(m+2)	比例常数(K_P)
	(s)+(m+3)	积分常数(T_I)
	(s)+(m+4)	微分常数(T_D)
	(s)+(m+5)	滤波器系数(α)
	(s)+(m+6)	操作量下限值(MVLL)
	(s)+(m+7)	操作量上限值(MVHL)
	(s)+(m+8)	操作量变化率限制值(Δ MVL)
	(s)+(m+9)	测定值变化率限制值(Δ PVL)

*1 $m=(n-1) \times 10+2$

- PID控制用数据的设置中使用的软元件点数通过下式计算。

软元件点数=2+10×n(n: 使用环路数)

- 各数据应以BIN值进行设置。
- 使用环路的软元件点数超过指定软元件的最终软元件编号时将变为错误且不进行处理。
- 1个扫描中在多个位置执行了PIDINIT(P)指令的情况下，最靠近PIDCONT(P)指令执行的PIDINIT(P)指令的设定值将有效。
- PIDINIT(P)指令的执行必须在执行PIDCONT(P)指令之前。未执行PIDINIT(P)指令的情况下，不能进行PID控制。

出错

错误代码(SD0)	内容
3285H	(s)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> • PID控制用数据中设置的值超出了允许设置范围 • (使用环路数) < (1个扫描的执行环路数) • (操作量上限值) < (操作量下限值)

PID运算

PIDCONT (P)

通过执行指令的ON, 进行采样周期的计测及PID运算。

梯形图	ST
	ENO:=PIDCONT (EN, s) ; ENO:=PIDCONT (EN, s) ;

FBD/LD

■执行条件

指令	执行条件
PIDCONT	
PIDCONT (P)	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s)	分配到输入输出数据区域中的起始软元件	—	字	ANY16* ¹
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下, 应在确保动作所需区域的前提下定义数组, 指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它	
	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)	—	—	○	—	—	—	—	○	—	—	—	—

功能

- 执行PIDCONT (P) 指令时，进行采样周期的计测及PID运算。
- PIDCONT (P) 指令，以(s)中指定的软元件编号及其以后设置的输入输出数据区域的设定值(SV)、测定值(PV)为基础进行PID运算，将运算结果存储到输入输出数据区域的自动操作量(MV)区域中。
- PID运算，是在经过采样周期的设置时间后的首个PIDCONT (P) 指令执行时进行。
- PID控制过程中必须将控制指令置为ON，并使PIDCONT (P) 指令在每个扫描中执行。未在每个扫描中执行的情况下，将无法以正常的采样周期进行PID运算。此外，1个扫描中不能多次执行PIDCONT (P) 指令。1个扫描中执行多次PIDCONT (P) 指令的情况下，将无法以正常的采样周期进行PID运算。
- PIDCONT (P) 指令不能记述到中断程序中使用。将PIDCONT (P) 指令记述到中断程序中的情况下，将无法以正常的采样周期进行PID运算。
- 在(s)中，将指定的软元件编号的起始指定到输入输出数据区域中。(☞ 1022页 输入输出数据)
- 将文件寄存器指定为输入输出数据区域的情况下，请勿对文件寄存器施加存储器保护。施加了存储器保护时不会变为错误，但无法进行正常的PID运算。
- 输入输出数据的分配如下所示。

项目		输入输出数据	
写入		(s)+0	首次处理标志
读取/写入禁止		(s)+1 ⋮ (s)+9	PID控制用工作区域 (用户不能使用)
No. 1环路用输入输出数据区域 (18字)	写入	(s)+10	设定值(SV)
		(s)+11	测定值(PV)
	读取	(s)+12	自动操作量(MV)
		(s)+13	滤波器后的测定值(PVf)
	写入	(s)+14	手动操作量(MV _{MAN})
		(s)+15	手动/自动选择(MAN/AUTO)
	读取/写入	(s)+16	报警(ALARM)
读取/写入禁止	(s)+17 ⋮ (s)+27	No. 1环路用工作区域 (用户不能使用)	
⋮	⋮	⋮	⋮
No. n环路用输入输出数据区域 (18字)*1	写入	(s)+(m+0)	设定值(SV)
		(s)+(m+1)	测定值(PV)
	读取	(s)+(m+2)	自动操作量(MV)
		(s)+(m+3)	滤波器后的测定值(PVf)
	写入	(s)+(m+4)	手动操作量(MV _{MAN})
		(s)+(m+5)	手动/自动选择(MAN/AUTO)
	读取/写入	(s)+(m+6)	报警(ALARM)
读取/写入禁止	(s)+(m+7) ⋮ (s)+(m+17)	No. n环路用工作区域 (用户不能使用)	

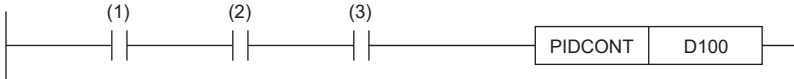
*1 $m=(n-1) \times 18+10$

- 输入输出数据设置中使用的软元件点数由下式计算。

软元件点数=10+18×n (n: 使用环路数)

- 各数据应以BIN值进行设置。
- 首次处理标志是PID运算开始时的处理方法的设置。
- 首次运算处理时，将视为到达设置的采样周期进行运算。
- 首次处理标志为0的情况下，将使用环路数的PID运算处理通过1个扫描进行批量处理。此外，0以外的情况下，将使用环路数的PID运算处理分割为数个扫描进行处理后，从首次处理完成的环路开始依次开始循环。每个扫描的处理环路数将变为设置的1个扫描的执行环路数。

- 输入输出数据区域中处于“写入”的数据，应由用户通过程序进行写入。处于“读取”的数据，由用户通过程序读取使用。对处于“读取/写入禁止”及“读取”的数据进行写入时，将无法正常进行运算，因此请勿进行。但是，从初始状态开始控制的情况下，需要通过程序进行数据清除。
- 即使通过手动模式进行了手动操作量(MV_{MAN})的输出的情况下，也应在每个扫描执行PIDCONT(P)指令。如果不执行PIDCONT(P)指令，将无法执行无冲击功能。
- PIDCONT(P)指令，应通过各模块的READY信号采取互锁，以便只有在用于获取测定值(PV)的A/D转换模块以及用于输出操作量(MV)的D/A转换模块正常时才执行指令。如果在各模块异常时执行，将无法正常进行测定值(PV)的获取或正常进行操作量(MV)的输出，因此也无法正常进行PID运算。



- (1) 控制指令
 (2) A/D转换模块的READY信号
 (3) D/A转换模块的READY信号

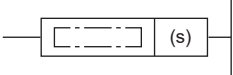
出错

错误代码(SD0)	内容
3285H	(s)中指定的输入输出数据区域中设置的数据值超出允许设置范围时。
3292H	执行PIDCONT(P)指令之前，未执行PIDINIT(P)指令时。

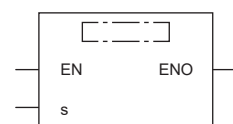
指定环路No. 的运算停止

PIDSTOP(P)


停止(s)中指定的环路No. 的PID运算。

梯形图	ST
	ENO:=PIDSTOP(EN, s); ENO:=PIDSTOPP(EN, s);

FBD/LD



■执行条件

指令	执行条件
PIDSTOP	
PIDSTOPP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	停止的环路No.	1~32	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	—	○	—	—	—	—	○	○	—	—	—	

功能

- 停止(s)中指定的环路No. 的PID运算。通过PIDSTOP(P)指令停止的环路，即使执行PIDINIT(P)指令，也不重启PID运算。
- 停止中运算数据将被保持。

出错

错误代码(SD0)	内容
3285H	(s)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> • 指定的环路No. 不存在 • 指定的值超出了1~32的范围
3292H	在执行PIDSTOP(P)指令之前，未执行PIDINIT(P)指令、PIDCONT(P)指令时。

指定环路No. 的运算开始

PIDRUN(P)

开始指定环路No. 的运算。

梯形图	ST
	ENO:=PIDRUN(EN, s); ENO:=PIDRUNP(EN, s);

FBD/LD

■执行条件

指令	执行条件
PIDRUN	
PIDRUNP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	停止的环路No.	1~32	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位		字		双字			间接指定	常数				其它
	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)	○	—	○	—	—	—	—	○	○	—	—	—	—

功能

- 开始(s)中指定的环路No. 的PID运算。是用于对通过PIDSTOP(P)指令停止了PID运算的环路No. 再次执行运算的指令。
- 对正在执行PID运算的环路No. 执行了PIDRUN(P)指令的情况下，将变为无处理。

出错

错误代码(SD0)	内容
3285H	(s)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> 指定的环路No. 不存在 指定的值超出了1~32的范围
3292H	在执行PIDRUN(P)指令之前，未执行PIDINIT(P)指令、PIDCONT(P)指令时。

指定环路No. 的参数更改

PIDPRMW (P)

将指定的环路No. 的运算参数，更改为指定的软元件编号及其以后存储的PID控制用数据。

梯形图	ST
	ENO:=PIDPRMW (EN, s1, s2); ENO:=PIDPRMWP (EN, s1, s2);

FBD/LD

■执行条件

指令	执行条件
PIDPRMW	
PIDPRMWP	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	更改的环路No.	1~32	无符号BIN16位	ANY16
(s2)	存储了更改的PID控制用数据的起始软元件	—	字	ANY16*1
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字		双字		间接指定	常数				其它
	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)	—	—	○	—	—	—	○	—	—	—	—	—

功能

- 将(s1)中指定的环路No.的运算参数,更改为(s2)中指定的软元件编号及其以后存储的PID控制用数据。
- (s2)中指定的软元件编号及其以后的PID控制用数据的构成如下所示。

PID控制用数据	
(s2)+0	运算公式选择
(s2)+1	采样周期(T_s)
(s2)+2	比例常数(K_p)
(s2)+3	积分常数(T_I)
(s2)+4	微分常数(T_D)
(s2)+5	滤波器系数(α)
(s2)+6	操作量下限值(MVLL)
(s2)+7	操作量上限值(MVHL)
(s2)+8	操作量变化率限制值(ΔMVL)
(s2)+9	测定值变化率限制值(ΔPVL)

出错

错误代码(SD0)	内容
3285H	(s1)中设置了超出允许指定范围的数据时。 指定的环路No.不存在 指定的值超出了1~32的范围
	(s2)中指定的PID控制用数据超出允许设置范围时。
3292H	在执行PIDPRMW(P)指令之前,未执行PIDINIT(P)指令时。

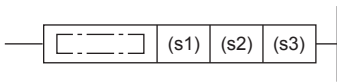
25 高速I/O控制指令

25.1 高速输入输出的开始/停止

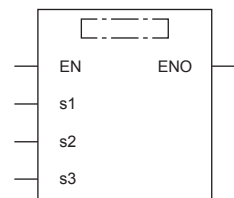
16位数据高速输入输出功能的开始/停止

HIOEN(P)

控制高速输入输出功能的开始/停止。

梯形图	ST
	ENO:=HIOEN(EN, s1, s2, s3); ENO:=HIOENP(EN, s1, s2, s3);

FBD/LD



设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	开始/停止的功能编号	请参阅功能编号(☞P 1049页 功能编号)	带符号BIN16位	ANY16
(s2)	设置将开始功能的通道编号的位	-32768~+32767	带符号BIN16位	ANY16
(s3)	设置已停止功能的通道编号的位	-32768~+32767	带符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可以使用的软元件

操作数	位	字	双字			间接指定	常数			其它	
	X、Y、M、L、SM、F、B、SB	T、ST、C、D、W、SD、SW、R、ZR、RD	U□\G□、U3E□\G□	Z	LT、LST、LC		LZ	K、H	E		\$
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(s3)	○	○	○	○	—	—	○	○	—	—	—

功能

在(s1)中指定要开始/停止的功能编号、在(s2)中指定要开始的通道的位、在(s3)中指定要停止的通道的位。
(s1)中可以指定的功能编号如下表所示。

■功能编号

功能编号	功能名称
K0	高速计数器
K10*1	脉冲密度/转速测定
K20*1	高速比较表(控制器)
K21*1	高速比较表(高速脉冲输入输出模块1)
K22*1	高速比较表(高速脉冲输入输出模块2)
K23*1	高速比较表(高速脉冲输入输出模块3)
K24*1	高速比较表(高速脉冲输入输出模块4)
K40	脉冲宽度测定
K50	PWM

*1 动作中使高速计数器(功能编号: K0)停止时, 虽然功能仍处于工作状态, 但因计数器停止, 因此将变为无处理。

在各功能编号中, (s2)、(s3)可以指定的值如下表所示。

• 功能编号为K0的情况

可对每个高速计数器的通道, 分别控制计数器的开始、停止。

CH1~CH8为控制器, CH9~CH16为高速脉冲输入输出模块。

位位置															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	通道7	通道6	通道5	通道4	通道3	通道2	通道1

例

要开始通道3时, 在(s2)中设置04H。要停止时, 在(s3)中设置04H。

要开始通道1、通道4、通道5时, 在(s2)中设置19H。要停止时, 在(s3)中设置19H。

要开始通道1、通道4、停止通道5时, 在(s2)中设置09H、在(s3)中设置10H。

• 功能编号为K10的情况

可对每个高速计数器的通道, 分别控制脉冲密度测定(转速测定)的开始、停止。应将b8~b15置为OFF。

位位置															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—								通道8	通道7	通道6	通道5	通道4	通道3	通道2	通道1

• 功能编号为K20~K24的情况

设置将开始、停止的高速比较表编号位设为ON的值。

控制器(K20)时, 可以使用b0~b15。

高速脉冲输入输出模块(K21~K24)时, 可以使用b0~b14。

位位置															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	通道7	通道6	通道5	通道4	通道3	通道2	通道1

• 功能编号为K40、K50的情况

可对每个通道, 分别控制脉冲宽度测定及PWM的开始、停止。应忽略空位。

CH1~CH4为控制器, CH5~CH12为高速脉冲输入输出模块。

位位置															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—				CH12	CH11	CH10	CH9	CH8	通道7	通道6	通道5	通道4	通道3	通道2	通道1

注意事项

- 请勿在中断优先度的设置为1的程序中指定高速脉冲输入输出模块的通道编号。
- 为开始与停止设置同一通道均为ON的值时，停止动作的优先级更高。
- 通过HIOEN指令执行脉冲密度(功能编号：K10)、高速比较表(功能编号：K20~K24)时，停止高速计数器(功能编号：K0)的情况下，虽然功能仍处于运行中，但不进行任何处理。
- 通过HIOEN指令执行高速计数器(功能编号：K0)时，如果存在发生错误(错误代码：3781H)的通道，则该通道之后不会进行计数器的开始/停止。
- 通过HIOEN指令使用高速比较表(功能编号：K20)时，高速比较同时使用次数(通过HIOEN指令指定的表格数+定位参数设置中选择了输入中断信号1的高速模式的通道数)应不超过32次。
- 通过HIOEN指令使用高速比较表(功能编号：使用K21~K24)时，高速比较同时使用次数(通过HIOEN指令指定的表数+定位参数设置中选择了输入中断信号1的高速模式的通道数+定位参数设置中选择了原点回归(零点信号)的通道数+定位参数设置中选择了原点回归(清除信号)的通道数)应不超过15次。
- 开始高速比较表(功能编号：K20~K24)时，需要事先通过HIOEN指令开始高速计数器。
- 使用高速计数器(功能编号：K0)时，应在HIOEN指令完成后进行至开始对象通道的脉冲输入。完成前进行了脉冲输入的情况下，可能会发生遗漏。
- 高速输入输出指令与下列参数设置联动动作。

功能编号	利用HIOEN指令指定的功能	参数设置
K0	高速计数器	高速计数器通道设置
K10	脉冲密度(转速测定)	脉冲密度/转速测定通道设置 高速计数器
K20	高速比较表(控制器)	高速计数器一致输出设置
K21	高速比较表(高速脉冲输入输出模块1)	
K22	高速比较表(高速脉冲输入输出模块2)	
K23	高速比较表(高速脉冲输入输出模块3)	
K24	高速比较表(高速脉冲输入输出模块4)	
K40	脉冲宽度测定	脉冲宽度测定通道设置
K50	PWM	PWM 通道设置

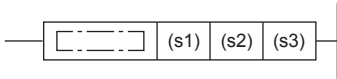
出错

错误代码(SD0)	内容
1810H	指定了其它指令已使用的通道编号时。
2801H	指定了不存在对象模块的通道编号时。
36C5H	执行指令时与对象模块的通信中发生了超时时。
36C7H	执行指令时在访问对象模块中检测到信号异常时。
3285H	为(s)指定了超出范围的功能编号时。
3600H	执行了参数设置中未选择的通道编号时。
3602H	在中断程序的中断优先度为1~16的情况下，通过HIOEN指令执行了高速输入输出模块的计数器计数开始/停止、脉冲宽度测定的开始/停止以及PWM的开始/停止时。
3781H	在环长设置及预设置输入值有效的通道中，指定了环长≤预设值并执行时。

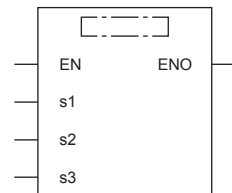
32位数据高速输入输出功能的开始/停止

DHIOEN(P)

控制高速输入输出功能的开始/停止。

梯形图	ST
	ENO:=DHIOEN(EN, s1, s2, s3); ENO:=DHIOENP(EN, s1, s2, s3);

FBD/LD



设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	开始/停止的功能编号	请参阅功能编号(☞ 1052页 功能编号)	带符号BIN16位	ANY16
(s2)	设置将开始功能的通道编号的位	-2147483648~+2147483647	带符号BIN32位	ANY32
(s3)	设置已停止功能的通道编号的位	-2147483648~+2147483647	带符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可以使用的软元件

操作数	位	字	双字			间接指定	常数			其它	
	X、Y、M、L、SM、F、B、SB	T、ST、C、D、W、SD、SW、R、ZR、RD	U□\G□、U3E□\G□	Z	LT、LST、LC		LZ	K、H	E		\$
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(s3)	○	○	○	○	○	○	○	○	—	—	—

功能

在(s1)中指定要开始/停止的功能编号、在(s2)中指定要开始的通道的位、在(s3)中指定要停止的通道的位。
(s1)中可以指定的功能编号如下表所示。

■功能编号

功能编号	功能名称
K0	高速计数器
K10*1	脉冲密度/转速测定
K20*1	高速比较表(控制器)
K21*1	高速比较表(高速脉冲输入输出模块1)
K22*1	高速比较表(高速脉冲输入输出模块2)
K23*1	高速比较表(高速脉冲输入输出模块3)
K24*1	高速比较表(高速脉冲输入输出模块4)
K40	脉冲宽度测定
K50	PWM

*1 动作中使高速计数器(功能编号: K0)停止时, 虽然功能仍处于工作状态, 但因计数器停止, 因此将变为无处理。
在各功能编号中, (s2)、(s3)可以指定的值如下表所示。

• 功能编号为K0的情况

可对每个高速计数器的通道, 分别控制计数器的开始、停止。

CH1~CH8为控制器, CH9~CH16为高速脉冲输入输出模块。

位位置															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	通道7	通道6	通道5	通道4	通道3	通道2	通道1

例

要开始通道3时, 在(s2)中设置04H。要停止时, 在(s3)中设置04H。

要开始通道1、通道4、通道5时, 在(s2)中设置19H。要停止时, 在(s3)中设置19H。

要开始通道1、通道4、停止通道5时, 在(s2)中设置09H、在(s3)中设置10H。

• 功能编号为K10的情况

可对每个高速计数器的通道, 分别控制脉冲密度测定(转速测定)的开始、停止。应将b8~b15置为OFF。

位位置															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—								通道8	通道7	通道6	通道5	通道4	通道3	通道2	通道1

• 功能编号为K20~K24的情况

设置将开始、停止的高速比较表编号位设为ON的值。

控制器(K20)时, 可以使用b0~b31。

高速脉冲输入输出模块(K21~K24)时, 可以使用b0~b14。

低位置															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	通道7	通道6	通道5	通道4	通道3	通道2	通道1

高位置															
b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CH32	CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17

• 功能编号为K40、K50的情况

可对每个通道, 分别控制脉冲宽度测定及PWM的开始、停止。应忽略空位。

CH1~CH4为控制器, CH5~CH12为高速脉冲输入输出模块。

位位置															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—				CH12	CH11	CH10	CH9	CH8	通道7	通道6	通道5	通道4	通道3	通道2	通道1

注意事项

- 请勿在中断优先度的设置为“1”的程序中，指定高速脉冲输入输出模块的通道编号。
- 为开始与停止设置同一通道均为ON的值时，停止动作的优先级更高。
- 通过HIOEN指令执行脉冲密度(功能编号：K10)、高速比较表(功能编号：K20~K24)时，停止高速计数器(功能编号：K0)的情况下，虽然功能仍处于运行中，但不进行任何处理。
- 通过DHIOEN指令执行高速计数器(功能编号：K0)时，如果存在发生错误(错误代码：3781H)的通道，则该通道之后不会进行计数器的开始/停止。
- 通过HIOEN指令使用高速比较表(功能编号：K20)时，高速比较同时使用次数(通过HIOEN指令指定的表格数+定位参数设置中选择了输入中断信号1的高速模式的通道数)应不超过32次。
- 通过HIOEN指令使用高速比较表(功能编号：使用K21~K24)时，高速比较同时使用次数(通过HIOEN指令指定的表数+定位参数设置中选择了输入中断信号1的高速模式的通道数+定位参数设置中选择了原点回归(零点信号)的通道数+定位参数设置中选择了原点回归(清除信号)的通道数)应不超过15次。
- 开始高速比较表(功能编号：K20~K24)时，需要事先通过HIOEN指令开始高速计数器。
- 使用高速计数器(功能编号：K0)时，应在DHIOEN指令完成后进行至开始对象通道的脉冲输入。完成前进行了脉冲输入的情况下，可能会发生遗漏。
- 高速输入输出指令与下列参数设置联动动作。

功能编号	利用HIOEN指令指定的功能	参数设置
K0	高速计数器	高速计数器通道设置
K10	脉冲密度(转速测定)	脉冲密度/转速测定通道设置 高速计数器
K20	高速比较表(控制器)	高速计数器一致输出设置
K21	高速比较表(高速脉冲输入输出模块1)	
K22	高速比较表(高速脉冲输入输出模块2)	
K23	高速比较表(高速脉冲输入输出模块3)	
K24	高速比较表(高速脉冲输入输出模块4)	
K40	脉冲宽度测定	脉冲宽度测定通道设置
K50	PWM	PWM 通道设置

出错

错误代码(SD0)	内容
1810H	指定了其它指令已使用的通道编号时。
2801H	指定了不存在对象模块的通道编号时。
36C5H	执行指令时与对象模块的通信中发生了超时时。
36C7H	执行指令时在访问对象模块中检测到信号异常时。
3285H	为(s)指定了超出范围的功能编号时。
3600H	执行了参数设置中未选择的通道编号时。
3602H	在中断程序的中断优先度为1~16的情况下，通过HIOEN指令执行了高速输入输出模块的计数器计数开始/停止、脉冲宽度测定的开始/停止以及PWM的开始/停止时。
3781H	在环长设置及预设置输入值有效的通道中，指定了环长≤预设值并执行时。

26 定位功能专用指令

26.1 机械原点复位

定位指令在发生正转脉冲或反转脉冲时，会增减当前地址的内容。

由于当前地址在控制器的电源为OFF时被清除，因此接通电源后，必须将机械位置与控制器内当前地址的位置进行统一。通过机械原点复位用的DSZR/DDSZR指令进行原点复位，使机械位置与控制器内的当前地址一致。

DSZR/DDSZR

使用该指令进行机械式原点复位。

- DSZR：在16位的数据中指定操作数。
- DDSZR：在32位的数据中指定操作数。

梯形图	ST	FBD/LD
	ENO:=DSZR(EN, s1, s2, d1, d2); ENO:=DDSZR(EN, s1, s2, d1, d2);	

设置数据

■内容、范围、数据类型

- DSZR

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	原点复位速度或存储了数据的字软元件编号	1~65535 (用户单位)	无符号BIN16位	ANY16
(s2)	蠕变速度或存储了数据的字软元件编号	1~65535 (用户单位)	无符号BIN16位	ANY16
(d1)	输出脉冲的轴编号	K1~K12	无符号BIN16位	ANY16
(d2)	原点复位结束，异常结束标志的位软元件编号	—	位	ANYBIT_ARRAY
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

- DDSZR

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	原点复位速度或存储了数据的字软元件编号	1~2147483647 (用户单位)	带符号BIN32位	ANY32
(s2)	蠕变速度或存储了数据的字软元件编号	1~2147483647 (用户单位)	带符号BIN32位	ANY32
(d1)	输出脉冲的轴编号	K1~K12	无符号BIN16位	ANY16
(d2)	原点复位结束，异常结束标志的位软元件编号	—	位	ANYBIT_ARRAY
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可以使用的软元件

• DSZR

操作数	位		字		双字			间接指定	常数			其它 (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)	○*2	—	○	○*4	○	—	—	○	○	—	—	—
(s2)	○*2	—	○	○*4	○	—	—	○	○	—	—	—
(d1)	—	—	○	○*4	○	—	—	○	○	—	—	—
(d2)*1	○*2	—	○*3	—	—	—	—	—	—	—	—	—

• DDSZR

操作数	位		字		双字			间接指定	常数			其它 (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)	○*2	—	○	○*4	○	○	○	○	○	—	—	—
(s2)	○*2	—	○	○*4	○	○	○	○	○	—	—	—
(d1)	—	—	○	○*4	○	—	—	○	○	—	—	—
(d2)*1	○*2	—	○*3	—	—	—	—	—	—	—	—	—

*1 从指定的软元件开始占用2点。

*2 FX、FY不能使用。

*3 T、ST、C、FD不能使用。

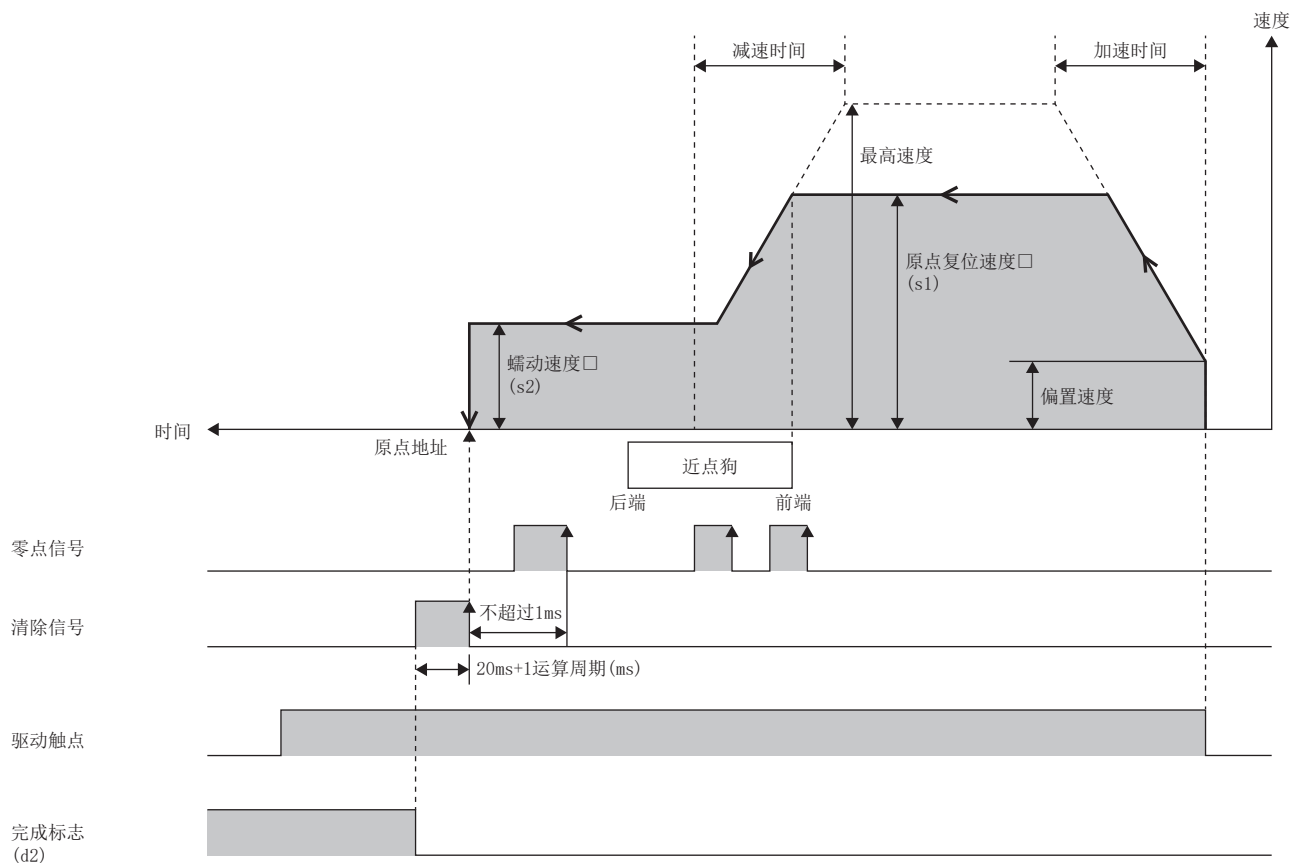
*4 不能使用J□\□。

功能

使用该指令进行机械式原点复位。

使用正转极限、反转极限，可以进行采用狗搜索功能的原点复位。关于狗搜索功能的详细内容，请参阅下述手册。

📖 所使用的控制器的用户手册



- (1) 驱动触点为ON时，输出脉冲，从偏置速度进行加速动作。
- (2) 到达原点复位速度后，以原点复位速度动作。
- (3) 检测出近点DOG时，进行减速动作。
- (4) 到达蠕变速度后，以蠕变速度动作。
- (5) 近点DOG由ON→OFF后，如果检测到零点信号，则停止脉冲输出。

- 在(s1)中指定原点复位速度。应通过脉冲换算置为1pps~200kpps的值。
- 在(s2)中指定蠕变速率。应通过脉冲换算置为1pps~200kpps的值。
- 在(d1)中指定输出脉冲的轴编号。应在高速I/O参数中指定设有定位参数的轴编号。指定了未进行设置的轴编号的情况下将不动作。
- 在(d2)中指定指令执行完成，在(d2)+1中指定异常完成标志的位软元件。
- 近点DOG信号和零点信号通过参数中设置的软元件工作。操作数中指定的原点复位速度、蠕变速度将覆盖特殊寄存器的原点复位速度、蠕变速度。间接指定了操作数的情况下，根据操作数的软元件值的更改，特殊寄存器的值也将被更改。

- 脉冲的输出方向通过原点复位方向和旋转方向设置进行确定。原点复位方向与旋转方向设置组合时的动作如下所示。

项目	旋转方向设置		
		通过正转脉冲输出增加当前地址	通过反转脉冲输出增加当前地址
原点复位方向	正方向(地址增加方向)	输出方向: 正转、地址: 增加	输出方向: 反转、地址: 增加
	负方向(地址减少方向)	输出方向: 反转、地址: 减少	输出方向: 正转、地址: 减少

- 原点复位速度(s1)和蠕变速度(s2)的定位动作中的指令速度更改的有效时机为到零点信号检测之前为止。检测出零点信号后进行了更改的情况下,在重新驱动DSZR/DDSZR指令时将反映。
- 完成标志的动作时机如下所示。指定了停留时间的情况下,在经过停留时间后变为ON。

项目	指令执行完成标志(d2)	指令执行异常完成标志(d2)+1
ON条件	从原点复位完成时开始到ON→OFF条件成立为止	在下述操作中或从功能的动作完成开始到ON→OFF条件成立为止 <ul style="list-style-type: none"> • 轴已使用 • 定位动作中驱动触点OFF • 脉冲停止指令 • 脉冲减速停止指令 • 所有的输出禁止(SM9904) • RUN中写入 • 原点地址异常 • 通过更改为原点复位速度、蠕变速度0进行减速停止 • 通过近点DOG检测后的极限检测进行减速停止
ON→OFF条件	在执行下述任意一项之前保持ON状态 <ul style="list-style-type: none"> • 用户设为OFF • 重启定位指令 	

■相关软元件

• 特殊继电器

控制器	高速脉冲输入输出模块				名称	高速I/O 参数	R/W
	第1台	第2台	第3台	第4台			
轴1~4	轴5~6	轴7~8	轴9~10	轴11~12			
SM5500 SM5501 SM5502 SM5503	SM5504 SM5505	SM5506 SM5507	SM5508 SM5509	SM5510 SM5511	定位指令驱动中	×	R
SM5516 SM5517 SM5518 SM5519	SM5520 SM5521	SM5522 SM5523	SM5524 SM5525	SM5526 SM5527	脉冲输出中监控	×	R
SM5532 SM5533 SM5534 SM5535	SM5536 SM5537	SM5538 SM5539	SM5540 SM5541	SM5542 SM5543	发生定位错误	×	R/W
SM5628 SM5629 SM5630 SM5631	SM5632 SM5633	SM5634 SM5635	SM5636 SM5637	SM5638 SM5639	脉冲停止指令	×	R/W
SM5644 SM5645 SM5646 SM5647	SM5648 SM5649	SM5650 SM5651	SM5652 SM5653	SM5654 SM5655	脉冲减速停止指令	×	R/W
SM5660 SM5661 SM5662 SM5663	SM5664 SM5665	SM5666 SM5667	SM5668 SM5669	SM5670 SM5671	正转极限	×	R/W
SM5676 SM5677 SM5678 SM5679	SM5680 SM5681	SM5682 SM5683	SM5684 SM5685	SM5686 SM5687	反转极限	×	R/W
SM5772 SM5773 SM5774 SM5775	SM5776 SM5777	SM5778 SM5779	SM5780 SM5781	SM5782 SM5783	旋转方向设置	○	R/W
SM5804 SM5805 SM5806 SM5807	SM5808 SM5809	SM5810 SM5811	SM5812 SM5813	SM5814 SM5815	原点复位方向指定	○	R/W
SM5820 SM5821 SM5822 SM5823	SM5824 SM5825	SM5826 SM5827	SM5828 SM5829	SM5830 SM5831	清除信号输出功能有效	○	R/W
SM5868 SM5869 SM5870 SM5871	SM5872 SM5873	SM5874 SM5875	SM5876 SM5877	SM5878 SM5879	零点信号计数开始时间	○	R/W

R: 读取专用、R/W: 读取/写入用、○: 支持、×: 不支持

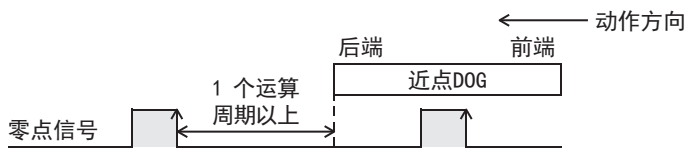
• 特殊寄存器

控制器	高速脉冲输入输出模块				名称	高速I/O 参数	R/W
	第1台	第2台	第3台	第4台			
轴1~4	轴5~6	轴7~8	轴9~10	轴11~12			
SD5500、SD5501 SD5540、SD5541 SD5580、SD5581 SD5620、SD5621	SD5660、SD5661 SD5700、SD5701	SD5740、SD5741 SD5780、SD5781	SD5820、SD5821 SD5860、SD5861	SD5900、SD5901 SD5940、SD5941	当前地址(用户单位)	×	R/W
SD5502、SD5503 SD5542、SD5543 SD5582、SD5583 SD5622、SD5623	SD5662、 SD5663SD5702、 SD5703	SD5742、SD5743 SD5782、SD5783	SD5822、SD5823 SD5862、SD5863	SD5902、SD5903 SD5942、SD5943	当前地址(脉冲单位)	×	R/W
SD5504、SD5505 SD5544、SD5545 SD5584、SD5585 SD5624、SD5625	SD5664、 SD5665SD5704、 SD5705	SD5744、SD5745 SD5784、SD5785	SD5824、SD5825 SD5864、SD5865	SD5904、SD5905 SD5944、SD5945	当前速度(用户单位)	×	R
SD5510 SD5550 SD5590 SD5630	SD5670 SD5710	SD5750 SD5790	SD5830 SD5870	SD5910 SD5950	定位错误 错误代码	×	R/W
SD5516、SD5517 SD5556、SD5557 SD5596、SD5597 SD5636、SD5637	SD5676、SD5677 SD5716、SD5717	SD5756、SD5757 SD5796、SD5797	SD5836、SD5837 SD5876、SD5877	SD5916、SD5917 SD5956、SD5957	最高速度	○	R/W
SD5518、SD5519 SD5558、SD5559 SD5598、SD5599 SD5638、SD5639	SD5678、SD5679 SD5718、SD5719	SD5758、SD5759 SD5798、SD5799	SD5838、SD5839 SD5878、SD5879	SD5918、SD5919 SD5958、SD5959	偏置速度	○	R/W
SD5520 SD5560 SD5600 SD5640	SD5680 SD5720	SD5760 SD5800	SD5840 SD5880	SD5920 SD5960	加速时间	○	R/W
SD5521 SD5561 SD5601 SD5641	SD5681 SD5721	SD5761 SD5801	SD5841 SD5881	SD5921 SD5961	减速时间	○	R/W
SD5526、SD5527 SD5566、SD5567 SD5606、SD5607 SD5646、SD5647	SD5686、SD5687 SD5726、SD5727	SD5766、SD5767 SD5806、SD5807	SD5846、SD5847 SD5886、SD5887	SD5926、SD5927 SD5966、SD5967	原点复位速度	○	R/W
SD5528、SD5529 SD5568、SD5569 SD5608、SD5609 SD5648、SD5649	SD5688、SD5689 SD5728、SD5729	SD5768、SD5769 SD5808、SD5809	SD5848、SD5849 SD5888、SD5889	SD5928、SD5929 SD5968、SD5969	蠕变速率	○	R/W
SD5530、SD5531 SD5570、SD5571 SD5610、SD5611 SD5650、SD5651	SD5690、 SD5691SD5730、 SD5731	SD5770、SD5771 SD5810、SD5811	SD5850、SD5851 SD5890、SD5891	SD5930、SD5931 SD5970、SD5971	原点地址	○	R/W
SD5532 SD5572 SD5612 SD5652	SD5692 SD5732	SD5772 SD5812	SD5852 SD5892	SD5932 SD5972	原点复位零点信号数	○	R/W
SD5533 SD5573 SD5613 SD5653	SD5693 SD5733	SD5773 SD5813	SD5853 SD5893	SD5933 SD5973	原点复位停留时间	○	R/W

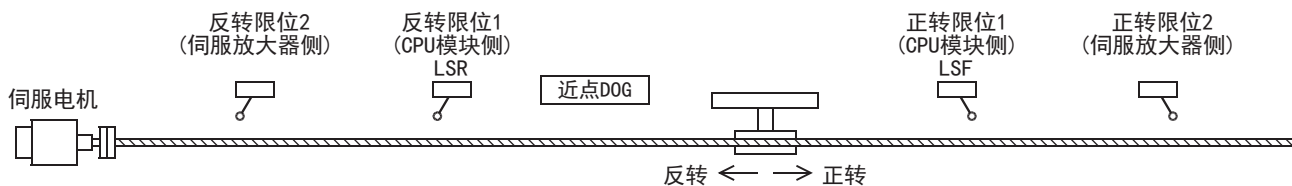
R: 读取专用、R/W: 读取/写入用、○: 支持、×: 不支持

注意事项

- 近点DOG的检测(前端/后端)受输入响应时间及顺序程序的运算时间的影响。从近点DOG后端开始至零点信号变为ON为止,应至少留出1个运算周期。
- 由于使用伺服电机的零点信号,因此应将近点DOG的后端与零点信号调整为如下关系。此外,原点位置需要微调时,应调整近点DOG的位置。



- 近点DOG应设计为在ON状态下有充分时间可减速至蠕动速度。在近点DOG的前端开始向蠕动速度减速,在“近点DOG的后端”或“近点DOG的后端开始检测出第一个零点信号”时停止,然后清除当前地址。到近点DOG的后端为止未减速至蠕动速度时,可能导致停止位置的偏差。
- 应将近点DOG设置在反转极限1(LSR)与正转极限1(LSF)之间。近点DOG、反转极限1(LSR)、正转极限1(LSF)未成为下图所示的关系时,可能不会进行预期的动作。



- 应充分降低蠕动速度。由于不进行减速停止,因此如果蠕动速度较快,惯性可能导致停止位置的偏差。

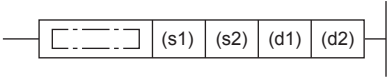
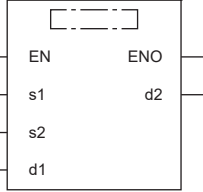
26.2 中断1速定位

在定位中，通过DVIT/DDVIT指令进行1速中断定长进给。
该指令可通过用户程序控制中断信号。

DVIT/DDVIT

该指令执行中断1速定长进给。

- DVIT：在16位的数据中指定操作数。
- DDVIT：在32位的数据中指定操作数。

梯形图	ST	FBD/LD
	ENO:=DVIT (EN, s1, s2, d1, d2); ENO:=DDVIT (EN, s1, s2, d1, d2);	

设置数据

■内容、范围、数据类型

- DVIT

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	定位地址或存储了数据的字软元件编号	-32768~+32767 (用户单位)	带符号BIN16位	ANY16
(s2)	指令速度或存储了数据的字软元件编号	1~65535 (用户单位)	无符号BIN16位	ANY16
(d1)	输出脉冲的轴编号	K1~K12	无符号BIN16位	ANY16
(d2)	指令执行完成、异常完成标志的位软元件编号	—	位	ANYBIT_ARRAY
EN	执行条件	-	位	BOOL
ENO	执行结果	-	位	BOOL

- DDVIT

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	定位地址或存储了数据的字软元件编号	-2147483648~+2147483647 (用户单位)	带符号BIN32位	ANY32
(s2)	指令速度或存储了数据的字软元件编号	1~2147483647 (用户单位)	带符号BIN32位	ANY32
(d1)	输出脉冲的轴编号	K1~K12	无符号BIN16位	ANY16
(d2)	指令执行完成、异常完成标志的位软元件编号	—	位	ANYBIT_ARRAY
EN	执行条件	-	位	BOOL
ENO	执行结果	-	位	BOOL

■可以使用的软元件

• DVIT

操作数	位		字		双字			间接指定	常数			其它 (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)	○*2	—	○	○*4	○	—	—	○	○	—	—	—
(s2)	○*2	—	○	○	○	—	—	○	○	—	—	—
(d1)	—	—	○	○	○	—	—	○	○	—	—	—
(d2)*1	○*2	—	○*3	—	—	—	—	—	—	—	—	—

• DDVIT

操作数	位		字		双字			间接指定	常数			其它 (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)	○*2	—	○	○*4	○	○	○	○	○	—	—	—
(s2)	○*2	—	○	○	○	○	○	○	○	—	—	—
(d1)	—	—	○	○	○	—	—	○	○	—	—	—
(d2)*1	○*2	—	○*3	—	—	—	—	—	—	—	—	—

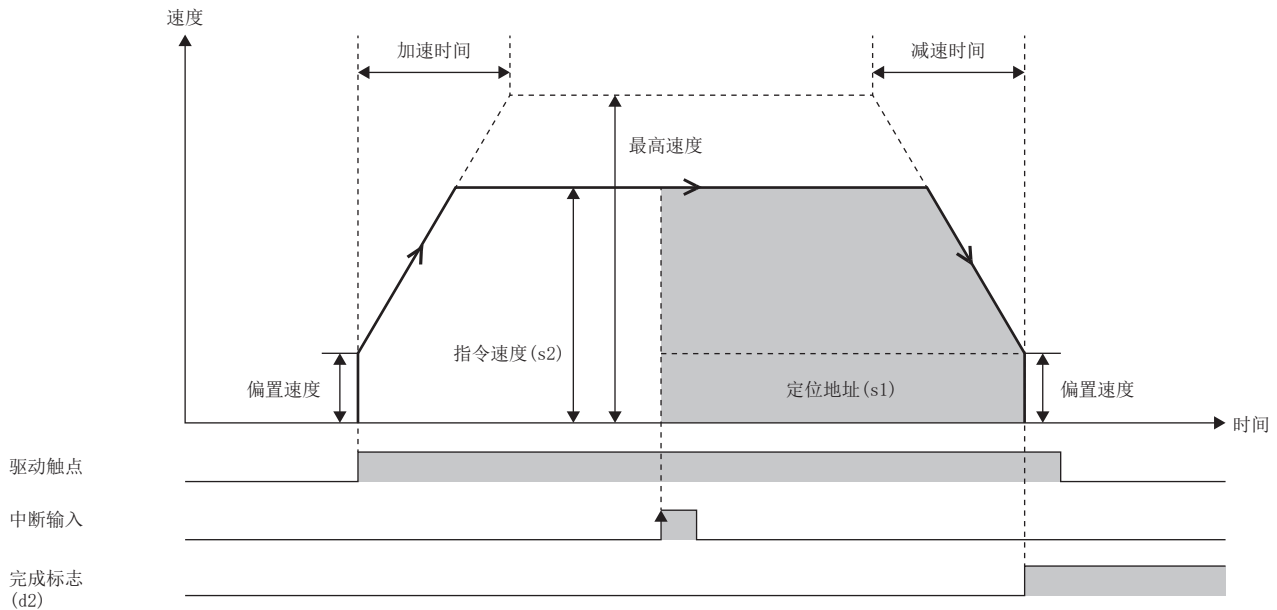
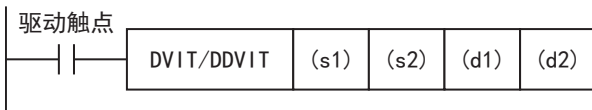
*1 从指定的软元件开始占用2点。

*2 FX、FY不能使用。

*3 T、ST、C、FD不能使用。

*4 不能使用J□\□。

该指令执行中断1速定长进给。从检测出中断输入的位置开始，以指定速度移动至指定定位地址。



(1) 驱动触点变为ON时，以(s2)中指定的指令速度向(s1)中指定的定位地址的方向移动。

(2) 从参数中设置的中断输入信号1变为ON的时刻开始，输出(s1)中指定的定位地址的脉冲并停止。

- 在(s1)中指定中断后输出的用户单位的移动量。应通过脉冲数换算置为-2147483648~2147483647的值。
- 在(s2)中指定用户单位的速度。应通过脉冲换算置为1pps~200kpps的值。
- 在(d1)中指定输出脉冲的轴编号。
- 在(d2)中指定指令执行完成，在(d2)+1中指定异常完成标志的位软元件。
- 如果检测出中断输入信号1，则从检测出中断输入信号1的位置开始，进行(s1)中指定的定位地址的脉冲输出。从可减速的位置开始进行减速停止。
- 中断输入信号1为无效的情况下，不能使用DVIT/DDVIT指令。未检测出中断输入信号1的情况下，则到检测出中断输入信号1为止，以(s2)的指定速度继续进行脉冲输出。
- 在指令驱动之前中断输入信号1为ON的情况下，即使驱动DVIT/DDVIT指令也不会检测出中断输入信号1。但是，使用外部开始信号时外部开始信号ON、中断输入信号1为ON的情况下，在驱动DVIT/DDVIT指令的同时将检测出中断输入信号1。
- 完成标志的动作时机如下所示。

项目	指令执行结束标志 (d2)	指令执行异常结束标志 (d2)+1
ON条件	从指定的定位地址的脉冲输出完成时开始到ON→OFF条件成立为止	在下述操作中或从功能的动作完成开始到ON→OFF条件成立为止 <ul style="list-style-type: none"> • 轴已使用 • 定位动作中驱动触点OFF • 脉冲停止指令 • 脉冲减速停止指令 • 行进方向的极限 • 所有的输出禁止(SM9904) • RUN中写入 • 定位地址异常 • 通过更改为指令速度0进行减速停止
ON→OFF条件	在执行下述任意一项之前保持ON状态 <ul style="list-style-type: none"> • 用户设为OFF • 重启定位指令 	

■相关软元件

• 特殊继电器

控制器	高速脉冲输入输出模块				名称	高速I/O 参数	R/W
	第1台	第2台	第3台	第4台			
轴1~4	轴5~6	轴7~8	轴9~10	轴11~12			
SM5500 SM5501 SM5502 SM5503	SM5504 SM5505	SM5506 SM5507	SM5508 SM5509	SM5510 SM5511	定位指令驱动中	×	R
SM5516 SM5517 SM5518 SM5519	SM5520 SM5521	SM5522 SM5523	SM5524 SM5525	SM5526 SM5527	脉冲输出中监控	×	R
SM5532 SM5533 SM5534 SM5535	SM5536 SM5537	SM5538 SM5539	SM5540 SM5541	SM5542 SM5543	发生定位错误	×	R/W
SM5628 SM5629 SM5630 SM5631	SM5632 SM5633	SM5634 SM5635	SM5636 SM5637	SM5638 SM5639	脉冲停止指令	×	R/W
SM5644 SM5645 SM5646 SM5647	SM5648 SM5649	SM5650 SM5651	SM5652 SM5653	SM5654 SM5655	脉冲减速停止指令	×	R/W
SM5660 SM5661 SM5662 SM5663	SM5664 SM5665	SM5666 SM5667	SM5668 SM5669	SM5670 SM5671	正转极限	×	R/W
SM5676 SM5677 SM5678 SM5679	SM5680 SM5681	SM5682 SM5683	SM5684 SM5685	SM5686 SM5687	反转极限	×	R/W
SM5772 SM5773 SM5774 SM5775	SM5776 SM5777	SM5778 SM5779	SM5780 SM5781	SM5782 SM5783	旋转方向设置	○	R/W

R: 读取专用、R/W: 读取/写入用、○: 支持、×: 不支持

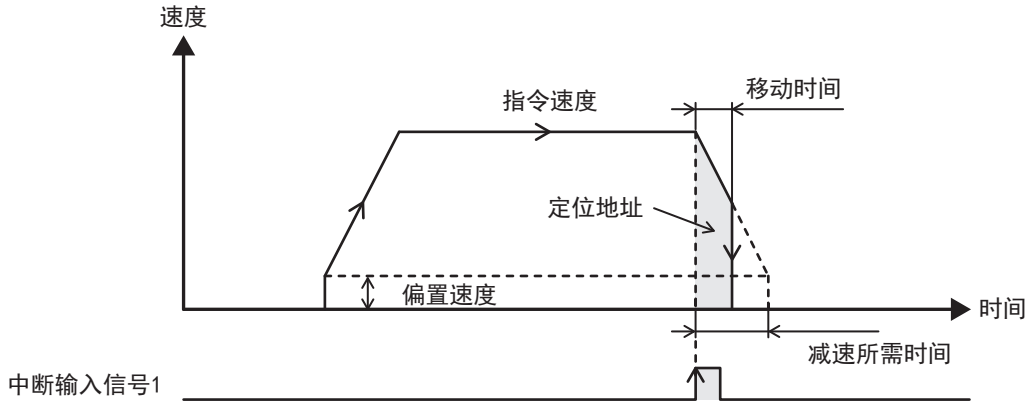
• 特殊寄存器

控制器	高速脉冲输入输出模块				名称	高速I/O 参数	R/W
	第1台	第2台	第3台	第4台			
轴1~4	轴5~6	轴7~8	轴9~10	轴11~12			
SD5500、SD5501 SD5540、SD5541 SD5580、SD5581 SD5620、SD5621	SD5660、SD5661 SD5700、SD5701	SD5740、SD5741 SD5780、SD5781	SD5820、SD5821 SD5860、SD5861	SD5900、SD5901 SD5940、SD5941	当前地址(用户单位)	×	R/W
SD5502、SD5503 SD5542、SD5543 SD5582、SD5583 SD5622、SD5623	SD5662、 SD5663SD5702、 SD5703	SD5742、SD5743 SD5782、SD5783	SD5822、SD5823 SD5862、SD5863	SD5902、SD5903 SD5942、SD5943	当前地址(脉冲单位)	×	R/W
SD5504、SD5505 SD5544、SD5545 SD5584、SD5585 SD5624、SD5625	SD5664、 SD5665SD5704、 SD5705	SD5744、SD5745 SD5784、SD5785	SD5824、SD5825 SD5864、SD5865	SD5904、SD5905 SD5944、SD5945	当前速度(用户单位)	×	R
SD5510 SD5550 SD5590 SD5630	SD5670 SD5710	SD5750 SD5790	SD5830 SD5870	SD5910 SD5950	定位错误 错误代码	×	R/W
SD5516、SD5517 SD5556、SD5557 SD5596、SD5597 SD5636、SD5637	SD5676、SD5677 SD5716、SD5717	SD5756、SD5757 SD5796、SD5797	SD5836、SD5837 SD5876、SD5877	SD5916、SD5917 SD5956、SD5957	最高速度	○	R/W
SD5518、SD5519 SD5558、SD5559 SD5598、SD5599 SD5638、SD5639	SD5678、SD5679 SD5718、SD5719	SD5758、SD5759 SD5798、SD5799	SD5838、SD5839 SD5878、SD5879	SD5918、SD5919 SD5958、SD5959	偏置速度	○	R/W
SD5520 SD5560 SD5600 SD5640	SD5680 SD5720	SD5760 SD5800	SD5840 SD5880	SD5920 SD5960	加速时间	○	R/W
SD5521 SD5561 SD5601 SD5641	SD5681 SD5721	SD5761 SD5801	SD5841 SD5881	SD5921 SD5961	减速时间	○	R/W

R: 读取专用、R/W: 读取/写入用、○: 支持、×: 不支持

注意事项

- 指令启动时定位地址(s1)为0的情况下，将异常完成。
- 在中断输入信号1检测前将定位地址(s1)更改为0的情况下，输入中断后将减速停止。减速停止后，将输出方向反转至接受了定位地址更改的地址，进行定位动作并正常完成。
- 至定位地址的移动时间少于减速停止所需的时间((s1)的设定值较少)的情况下，在到达定位地址之时立即停止。由于电机立即停止可能会损坏装置，应对立即停止加以注意。



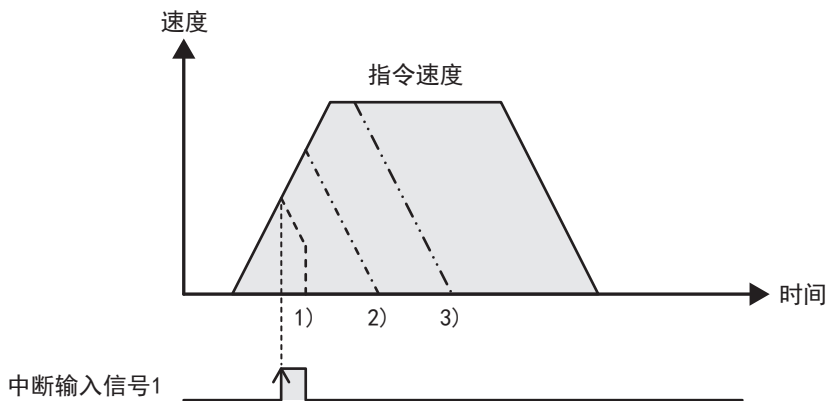
- 加速中检测出中断输入信号1的情况下，根据定位地址(s1)的值将变为下述动作。

1. 定位地址<从当前速度开始减速所需的脉冲数的情况下

中断输入信号1变为ON后，立即开始减速，在到达定位地址之时立即停止。由于电机立即停止可能会损坏装置，应对立即停止加以注意。

2. 从当前速度开始减速所需的脉冲数≤定位地址<从当前速度开始加减速所需的脉冲数的情况下
剩余脉冲数=加速至减速所需的脉冲数的位置，然后进行减速停止。

3. 从当前速度开始加减速所需的脉冲数≤定位地址的情况下
加速至指令速度(s2)，然后进行减速停止。



26.3 可变速度运行

在定位中，通过可变速脉冲输出指令进行可变速度运行。
该指令可以进行速度更改时的带加减速动作。

PLSV/DPLSV

输出带旋转方向输出的可变速脉冲。

- PLSV：在16位的数据中指定操作数。
- DPLSV：在32位的数据中指定操作数。

梯形图	ST	FBD/LD
	ENO:=PLSV (EN, s, d1, d2); ENO:=DPLSV (EN, s, d1, d2);	

设置数据

■内容、范围、数据类型

- PLSV

操作数	内容	范围	数据类型	数据类型(标签)
(s)	指令速度或存储了数据的字软元件编号	-32768~+32767 (用户单位)	带符号BIN16位	ANY16
(d1)	输出脉冲的轴编号	K1~K12	无符号BIN16位	ANY16
(d2)	指令执行完成、异常完成标志的位软元件编号	—	位	ANYBIT_ARRAY
EN	执行条件	-	位	BOOL
ENO	执行结果	-	位	BOOL

- DPLSV

操作数	内容	范围	数据类型	数据类型(标签)
(s)	指令速度或存储了数据的字软元件编号	-2147483648~+2147483647 (用户单位)	带符号BIN32位	ANY32
(d1)	输出脉冲的轴编号	K1~K12	无符号BIN16位	ANY16
(d2)	指令执行完成、异常完成标志的位软元件编号	—	位	ANYBIT_ARRAY
EN	执行条件	-	位	BOOL
ENO	执行结果	-	位	BOOL

■可以使用的软元件

• PLSV

操作数	位		字			双字		间接指定	常数			其它 (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	\$	
(s)	○*2	—	○	○*4	—	—	—	○	○	—	—	—
(d1)	—	—	○	○*4	—	—	—	○	○	—	—	—
(d2)*1	○*2	—	○*3	—	—	—	—	—	—	—	—	—

• DPLSV

操作数	位		字			双字		间接指定	常数			其它 (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	\$	
(s)	○*2	—	○	○*4	○	○	○	○	○	—	—	—
(d1)	—	—	○	○*4	○	—	—	○	○	—	—	—
(d2)*1	○*2	—	○*3	—	—	—	—	—	—	—	—	—

*1 从指定的软元件开始占用2点。

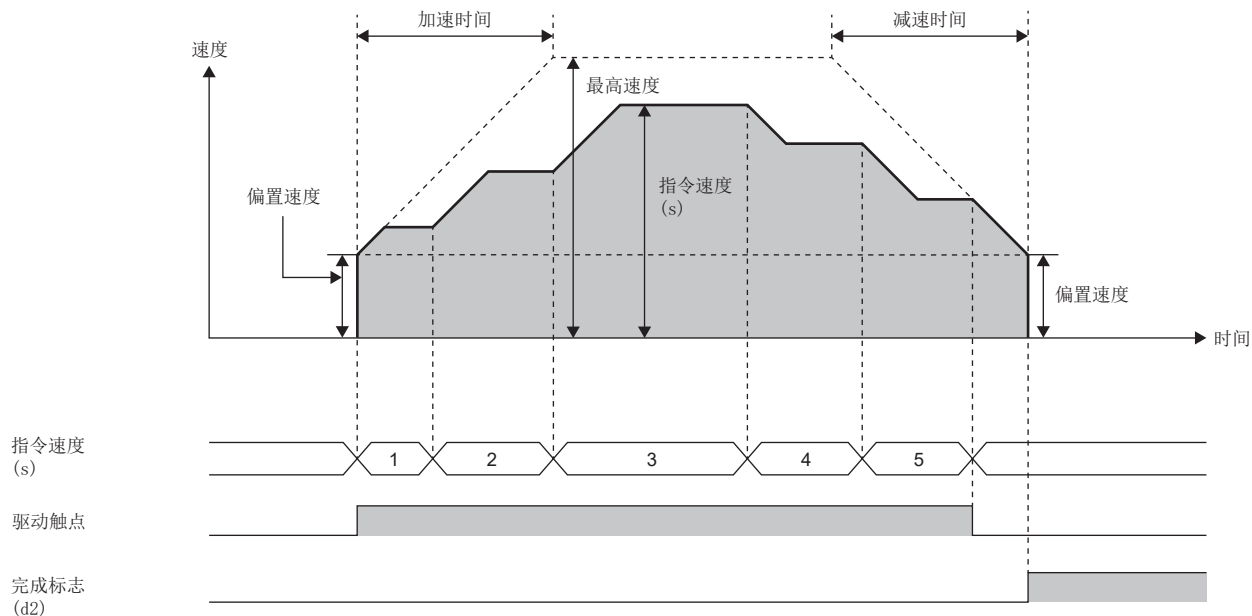
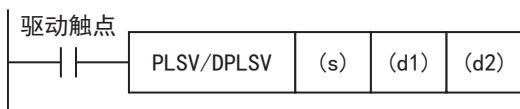
*2 FX、FY不能使用。

*3 T、ST、C、FD不能使用。

*4 不能使用J□\□。

功能

该指令用于输出带旋转方向输出的变速脉冲。



- (1) 驱动触点为ON时，输出脉冲，从偏置速度进行加速动作。
- (2) 到达指令速度后，以指令速度动作。
- (3) 动作中指令速度被更改的情况下，进行加减速动作，变速至指定的速度后进行动作。
- (4) 驱动触点变为OFF时，进行减速动作，停止脉冲输出。

- 在 (s) 中指定输出的指令速度。应通过脉冲换算置为1pps~200kpps的值。
- 在 (d1) 中指定输出脉冲的轴编号。
- 在 (d2) 中指定指令执行完成，在 (d2)+1 中指定异常完成标志的位元件。
- 动作中指令速度置为0的情况下，不变为异常结束而进行减速停止。此外，只要驱动触点未变为OFF，若更改指令速度，则脉冲输出会重新开始。
- 指令驱动时指令速度为0的情况下，则不动作。
- 加速时间为0的情况下，无加速动作而立即变速为指令速度。
- 减速时间为0的情况下，无减速动作，驱动触点变为OFF时立即停止。
- 完成标志的动作时机如下所示。

项目	指令执行结束标志 (d2)	指令执行异常结束标志 (d2)+1
ON条件	通过驱动触点OFF或脉冲减速停止指令从减速停止时开始到ON→OFF条件成立为止	在下述操作中或从功能的动作完成开始到ON→OFF条件成立为止 <ul style="list-style-type: none"> • 轴已使用 • 脉冲停止指令 • 行进方向的极限 • 所有的输出禁止 (SM9904) • RUN中写入 • 指令驱动时指令速度为0
ON→OFF条件	在执行下述任意一项之前保持ON状态 <ul style="list-style-type: none"> • 用户设为OFF • 重启定位指令 	

■相关软元件

• 特殊继电器

控制器	高速脉冲输入输出模块				名称	高速I/O 参数	R/W
	第1台	第2台	第3台	第4台			
轴1~4	轴5~6	轴7~8	轴9~10	轴11~12			
SM5500 SM5501 SM5502 SM5503	SM5504 SM5505	SM5506 SM5507	SM5508 SM5509	SM5510 SM5511	定位指令驱动中	×	R
SM5516 SM5517 SM5518 SM5519	SM5520 SM5521	SM5522 SM5523	SM5524 SM5525	SM5526 SM5527	脉冲输出中监控	×	R
SM5532 SM5533 SM5534 SM5535	SM5536 SM5537	SM5538 SM5539	SM5540 SM5541	SM5542 SM5543	发生定位错误	×	R/W
SM5628 SM5629 SM5630 SM5631	SM5632 SM5633	SM5634 SM5635	SM5636 SM5637	SM5638 SM5639	脉冲停止指令	×	R/W
SM5644 SM5645 SM5646 SM5647	SM5648 SM5649	SM5650 SM5651	SM5652 SM5653	SM5654 SM5655	脉冲减速停止指令	×	R/W
SM5660 SM5661 SM5662 SM5663	SM5664 SM5665	SM5666 SM5667	SM5668 SM5669	SM5670 SM5671	正转极限	×	R/W
SM5676 SM5677 SM5678 SM5679	SM5680 SM5681	SM5682 SM5683	SM5684 SM5685	SM5686 SM5687	反转极限	×	R/W
SM5772 SM5773 SM5774 SM5775	SM5776 SM5777	SM5778 SM5779	SM5780 SM5781	SM5782 SM5783	旋转方向设置	○	R/W

R: 读取专用、R/W: 读取/写入用、○: 支持、×: 不支持

• 特殊寄存器

控制器	高速脉冲输入输出模块				名称	高速I/O 参数	R/W
	第1台	第2台	第3台	第4台			
轴1~4	轴5~6	轴7~8	轴9~10	轴11~12			
SD5500、SD5501 SD5540、SD5541 SD5580、SD5581 SD5620、SD5621	SD5660、SD5661 SD5700、SD5701	SD5740、SD5741 SD5780、SD5781	SD5820、SD5821 SD5860、SD5861	SD5900、SD5901 SD5940、SD5941	当前地址(用户单位)	×	R/W
SD5502、SD5503 SD5542、SD5543 SD5582、SD5583 SD5622、SD5623	SD5662、 SD5663SD5702、 SD5703	SD5742、SD5743 SD5782、SD5783	SD5822、SD5823 SD5862、SD5863	SD5902、SD5903 SD5942、SD5943	当前地址(脉冲单位)	×	R/W
SD5504、SD5505 SD5544、SD5545 SD5584、SD5585 SD5624、SD5625	SD5664、 SD5665SD5704、 SD5705	SD5744、SD5745 SD5784、SD5785	SD5824、SD5825 SD5864、SD5865	SD5904、SD5905 SD5944、SD5945	当前速度(用户单位)	×	R
SD5510 SD5550 SD5590 SD5630	SD5670 SD5710	SD5750 SD5790	SD5830 SD5870	SD5910 SD5950	定位错误 错误代码	×	R/W
SD5516、SD5517 SD5556、SD5557 SD5596、SD5597 SD5636、SD5637	SD5676、SD5677 SD5716、SD5717	SD5756、SD5757 SD5796、SD5797	SD5836、SD5837 SD5876、SD5877	SD5916、SD5917 SD5956、SD5957	最高速度	○	R/W
SD5518、SD5519 SD5558、SD5559 SD5598、SD5599 SD5638、SD5639	SD5678、SD5679 SD5718、SD5719	SD5758、SD5759 SD5798、SD5799	SD5838、SD5839 SD5878、SD5879	SD5918、SD5919 SD5958、SD5959	偏置速度	○	R/W
SD5520 SD5560 SD5600 SD5640	SD5680 SD5720	SD5760 SD5800	SD5840 SD5880	SD5920 SD5960	加速时间	○	R/W
SD5521 SD5561 SD5601 SD5641	SD5681 SD5721	SD5761 SD5801	SD5841 SD5881	SD5921 SD5961	减速时间	○	R/W

R: 读取专用、R/W: 读取/写入用、○: 支持、×: 不支持

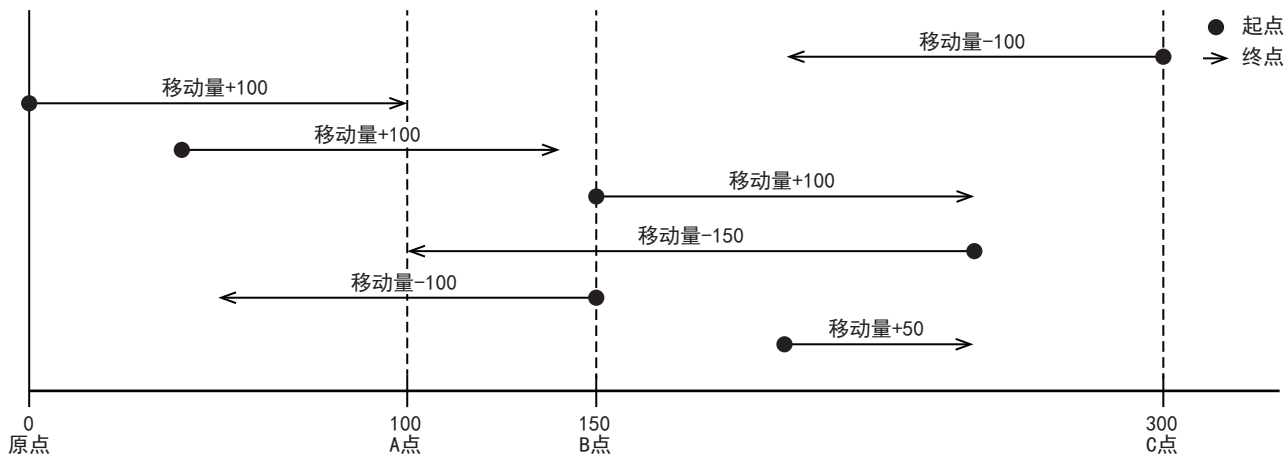
注意事项

- 动作中进行了指令速度符号变化的变速时，减速停止后从反方向开始进行脉冲输出的动作。从减速停止到再输出的等待时间为1ms+扫描时间。无法停止电机的等待时间时，切换方向应编程为先将指令速度设为0，并在减速停止后确保足够的等待时间后再次输出。
- 指令起动时指令速度(s)为0的情况下，则不动作而异常完成。

26.4 1速定位(相对)

该指令通过递增方式(根据相对地址指定位置)进行1速定位。

将当前停止的位置作为起点,指定移动方向和移动量(相对地址)进行定位动作。



DRVI/DDRVI

该指令通过相对地址进行1速定位。

- DRVI: 在16位的数据中指定操作数。
- DDRVI: 在32位的数据中指定操作数。

梯形图	ST	FBD/LD
	<pre>ENO:=DRVI (EN, s1, s2, d1, d2); ENO:=DDRVI (EN, s1, s2, d1, d2);</pre>	

设置数据

■内容、范围、数据类型

• DRVI

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	定位地址或存储了数据的字软元件编号	-32768~+32767 (用户单位)	带符号BIN16位	ANY16
(s2)	指令速度或存储了数据的字软元件编号	1~65535 (用户单位)	无符号BIN16位	ANY16
(d1)	输出脉冲的轴编号	K1~K12	无符号BIN16位	ANY16
(d2)	定位结束、异常结束标志的位软元件编号	—	位	ANYBIT_ARRAY
EN	执行条件	-	位	BOOL
ENO	执行结果	-	位	BOOL

• DDRVI

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	定位地址或存储了数据的字软元件编号	-2147483648~+2147483647 (用户单位)	带符号BIN32位	ANY32
(s2)	指令速度或存储了数据的字软元件编号	1~2147483647 (用户单位)	带符号BIN32位	ANY32
(d1)	输出脉冲的轴编号	K1~K12	无符号BIN16位	ANY16
(d2)	指令执行完成、异常完成标志的位软元件编号	—	位	ANYBIT_ARRAY
EN	执行条件	-	位	BOOL
ENO	执行结果	-	位	BOOL

■可以使用的软元件

• DRVI

操作数	位		字			双字		间接指定	常数				其它 (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)	○*2	—	○	○*4	○	—	—	○	○	—	—	—	—
(s2)	○*2	—	○	○*4	○	—	—	○	○	—	—	—	—
(d1)	—	—	○	○*4	○	—	—	○	○	—	—	—	—
(d2)*1	○*2	—	○*3	—	—	—	—	—	—	—	—	—	—

• DDRVI

操作数	位		字			双字		间接指定	常数				其它 (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)	○*2	—	○	○*4	○	○	○	○	○	—	—	—	—
(s2)	○*2	—	○	○*4	○	○	○	○	○	—	—	—	—
(d1)	—	—	○	○*4	○	—	—	○	○	—	—	—	—
(d2)*1	○*2	—	○	—	—	—	—	—	—	—	—	—	—

*1 从指定的软元件开始占用2点。

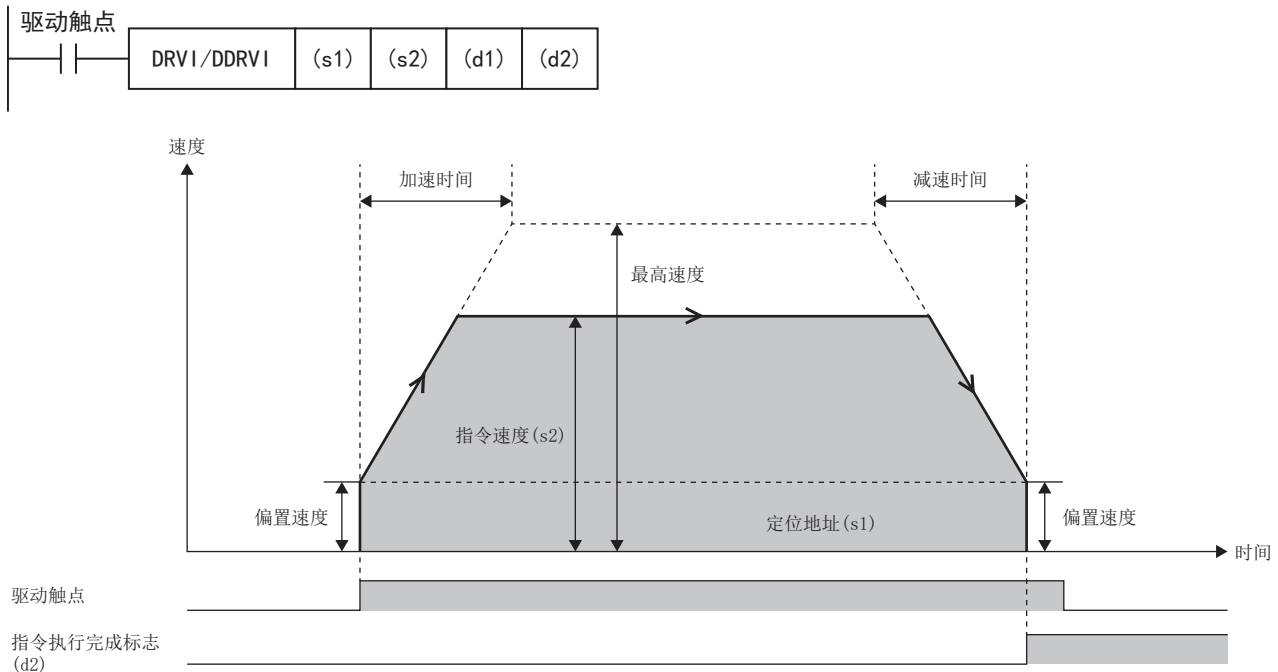
*2 FX、FY不能使用。

*3 T、ST、C、FD不能使用。

*4 不能使用J□\□。

功能

该指令通过相对地址进行1速定位。指定的定位地址采用递增方式，通过指定从当前位置开始的移动方向和移动量(相对地址)进行定位。



- (1) 驱动触点为ON时，输出脉冲，从偏置速度进行加速动作。
- (2) 到达指令速度后，以指令速度动作。
- (3) 从目标地点附近进行减速动作。
- (4) 在移动至指定的定位地址的地点，停止脉冲输出。

- 在(s1)中以相对地址指定用户单位的定位地址。应通过脉冲数换算置为-2147483647~2147483647的值。
- 在(s2)中指定用户单位的指令速度。应通过脉冲换算置为1pps~200kpps的值。
- 在(d1)中指定输出脉冲的轴编号。
- 在(d2)中指定指令执行完成，在(d2)+1中指定异常完成标志的位软元件。
- 完成标志的动作时机如下所示。

项目	指令执行结束标志(d2)	指令执行异常结束标志(d2)+1
ON条件	从指定的定位地址的脉冲输出完成时开始到ON→OFF条件成立为止	在下述操作中或从功能的动作完成开始到ON→OFF条件成立为止 <ul style="list-style-type: none"> • 轴已使用 • 定位动作中驱动触点OFF • 脉冲停止指令 • 脉冲减速停止指令*1 • 行进方向的极限 • 所有的输出禁止(SM9904) • RUN中写入 • 定位地址异常 • 通过更改为指令速度0进行减速停止
ON→OFF条件	在执行下述任意一项之前保持ON状态 <ul style="list-style-type: none"> • 用户设为OFF • 重启定位指令 	

*1 剩余距离运行有效为ON时，异常完成标志不变为ON。

■相关软元件

• 特殊继电器

控制器	高速脉冲输入输出模块				名称	高速I/O 参数	R/W
	第1台	第2台	第3台	第4台			
轴1~4	轴5~6	轴7~8	轴9~10	轴11~12			
SM5500 SM5501 SM5502 SM5503	SM5504 SM5505	SM5506 SM5507	SM5508 SM5509	SM5510 SM5511	定位指令驱动中	×	R
SM5516 SM5517 SM5518 SM5519	SM5520 SM5521	SM5522 SM5523	SM5524 SM5525	SM5526 SM5527	脉冲输出中监控	×	R
SM5532 SM5533 SM5534 SM5535	SM5536 SM5537	SM5538 SM5539	SM5540 SM5541	SM5542 SM5543	发生定位错误	×	R/W
SM5596 SM5597 SM5598 SM5599	SM5600 SM5601	SM5602 SM5603	SM5604 SM5605	SM5606 SM5607	剩余距离运行有效	×	R/W
SM5612 SM5613 SM5614 SM5615	SM5616 SM5617	SM5618 SM5619	SM5620 SM5621	SM5622 SM5623	剩余距离运行开始	×	R/W
SM5628 SM5629 SM5630 SM5631	SM5632 SM5633	SM5634 SM5635	SM5636 SM5637	SM5638 SM5639	脉冲停止指令	×	R/W
SM5644 SM5645 SM5646 SM5647	SM5648 SM5649	SM5650 SM5651	SM5652 SM5653	SM5654 SM5655	脉冲减速停止指令	×	R/W
SM5660 SM5661 SM5662 SM5663	SM5664 SM5665	SM5666 SM5667	SM5668 SM5669	SM5670 SM5671	正转极限	×	R/W
SM5676 SM5677 SM5678 SM5679	SM5680 SM5681	SM5682 SM5683	SM5684 SM5685	SM5686 SM5687	反转极限	×	R/W
SM5772 SM5773 SM5774 SM5775	SM5776 SM5777	SM5778 SM5779	SM5780 SM5781	SM5782 SM5783	旋转方向设置	○	R/W

R: 读取专用、R/W: 读取/写入用、○: 支持、×: 不支持

• 特殊寄存器

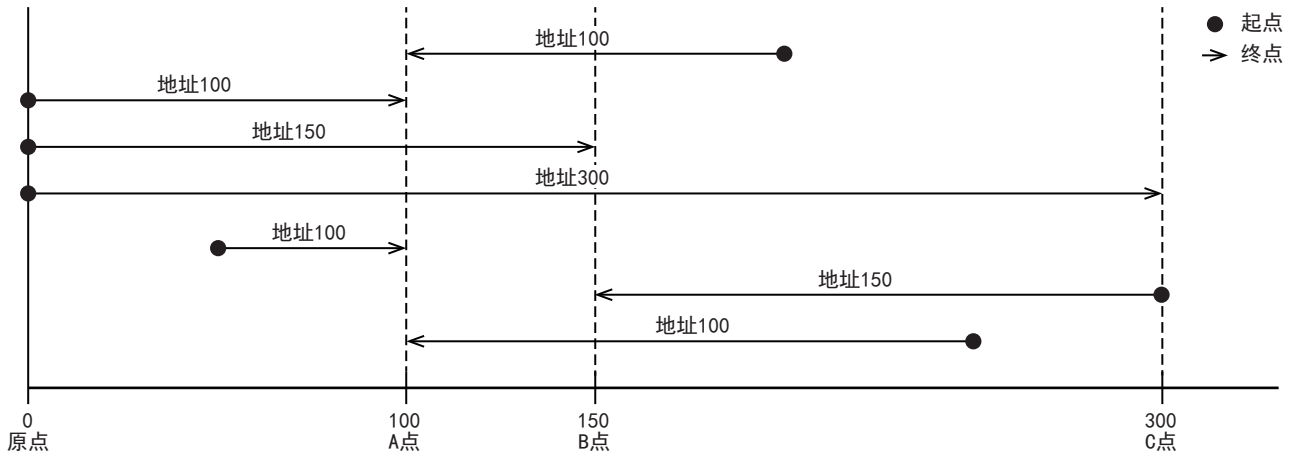
控制器	高速脉冲输入输出模块				名称	高速I/O 参数	R/W
	第1台	第2台	第3台	第4台			
轴1~4	轴5~6	轴7~8	轴9~10	轴11~12			
SD5500、SD5501 SD5540、SD5541 SD5580、SD5581 SD5620、SD5621	SD5660、SD5661 SD5700、SD5701	SD5740、SD5741 SD5780、SD5781	SD5820、SD5821 SD5860、SD5861	SD5900、SD5901 SD5940、SD5941	当前地址(用户单位)	×	R/W
SD5502、SD5503 SD5542、SD5543 SD5582、SD5583 SD5622、SD5623	SD5662、 SD5663SD5702、 SD5703	SD5742、SD5743 SD5782、SD5783	SD5822、SD5823 SD5862、SD5863	SD5902、SD5903 SD5942、SD5943	当前地址(脉冲单位)	×	R/W
SD5504、SD5505 SD5544、SD5545 SD5584、SD5585 SD5624、SD5625	SD5664、 SD5665SD5704、 SD5705	SD5744、SD5745 SD5784、SD5785	SD5824、SD5825 SD5864、SD5865	SD5904、SD5905 SD5944、SD5945	当前速度(用户单位)	×	R
SD5510 SD5550 SD5590 SD5630	SD5670 SD5710	SD5750 SD5790	SD5830 SD5870	SD5910 SD5950	定位错误 错误代码	×	R/W
SD5516、SD5517 SD5556、SD5557 SD5596、SD5597 SD5636、SD5637	SD5676、SD5677 SD5716、SD5717	SD5756、SD5757 SD5796、SD5797	SD5836、SD5837 SD5876、SD5877	SD5916、SD5917 SD5956、SD5957	最高速度	○	R/W
SD5518、SD5519 SD5558、SD5559 SD5598、SD5599 SD5638、SD5639	SD5678、SD5679 SD5718、SD5719	SD5758、SD5759 SD5798、SD5799	SD5838、SD5839 SD5878、SD5879	SD5918、SD5919 SD5958、SD5959	偏置速度	○	R/W
SD5520 SD5560 SD5600 SD5640	SD5680 SD5720	SD5760 SD5800	SD5840 SD5880	SD5920 SD5960	加速时间	○	R/W
SD5521 SD5561 SD5601 SD5641	SD5681 SD5721	SD5761 SD5801	SD5841 SD5881	SD5921 SD5961	减速时间	○	R/W

R: 读取专用、R/W: 读取/写入用、○: 支持、×: 不支持

26.5 1速定位(绝对)

该指令通过绝对方式(根据绝对地址指定位置)进行1速定位。

以原点为基准指定位置(绝对地址)进行定位动作。起点可以在任何位置。



DRVA/DDRVA

该指令通过绝对地址进行1速定位。

- DRVA: 在16位的数据中指定操作数。
- DDRVA: 在32位的数据中指定操作数。

梯形图	ST	FBD/LD
	<pre>ENO:=DRVA(EN, s1, s2, d1, d2); ENO:=DDRVA(EN, s1, s2, d1, d2);</pre>	

设置数据

■内容、范围、数据类型

• DRVA

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	定位地址或存储了数据的字软元件编号	-32768~+32767 (用户单位)	带符号BIN16位	ANY16
(s2)	指令速度或存储了数据的字软元件编号	1~65535 (用户单位)	无符号BIN16位	ANY16
(d1)	输出脉冲的轴编号	K1~K12	无符号BIN16位	ANY16
(d2)	指令执行完成、异常完成标志的位软元件编号	—	位	ANYBIT_ARRAY
EN	执行条件	-	位	BOOL
ENO	执行结果	-	位	BOOL

• DDRVA

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	定位地址或存储了数据的字软元件编号	-2147483648~+2147483647 (用户单位)	带符号BIN32位	ANY32
(s2)	指令速度或存储了数据的字软元件编号	1~2147483647 (用户单位)	带符号BIN32位	ANY32
(d1)	输出脉冲的轴编号	K1~K12	无符号BIN16位	ANY16
(d2)	指令执行完成、异常完成标志的位软元件编号	—	位	ANYBIT_ARRAY
EN	执行条件	-	位	BOOL
ENO	执行结果	-	位	BOOL

■可以使用的软元件

• DRVA

操作数	位		字			双字		间接指定	常数				其它 (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)	○*2	—	○	○*4	○	—	—	○	○	—	—	—	—
(s2)	○*2	—	○	○*4	○	—	—	○	○	—	—	—	—
(d1)	—	—	○	○*4	○	—	—	○	○	—	—	—	—
(d2)*1	○*2	—	○*3	—	—	—	—	—	—	—	—	—	—

• DDRVA

操作数	位		字			双字		间接指定	常数				其它 (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)	○*2	—	○	○*4	○	○	○	○	○	—	—	—	—
(s2)	○*2	—	○	○*4	○	○	○	○	○	—	—	—	—
(d1)	—	—	○	○*4	○	—	—	○	○	—	—	—	—
(d2)*1	○*2	—	○*3	—	—	—	—	—	—	—	—	—	—

*1 从指定的软元件开始占用2点。

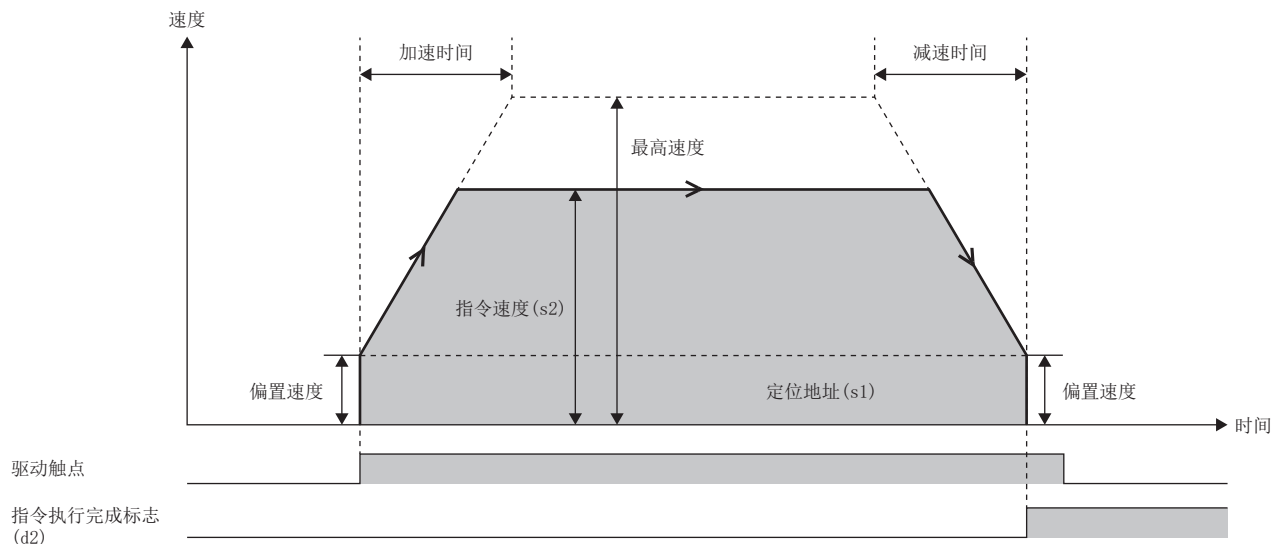
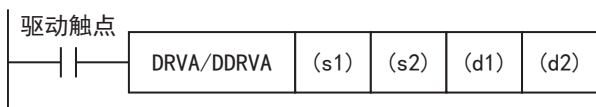
*2 FX、FY不能使用。

*3 T、ST、C、FD不能使用。

*4 不能使用J□\□。

功能

该指令通过绝对地址驱动进行1速定位。指定的定位地址采用绝对方式，以原点为基准，指定位置(绝对地址)进行定位。



(1) 驱动触点为ON时，输出脉冲，从偏置速度进行加速动作。

(2) 到达指令速度后，以指令速度动作。

(3) 从目标地点附近进行减速动作。

(4) 在指定的定位地址，停止脉冲输出。

- 在(s1)中以绝对地址指定用户单位的定位地址。应通过脉冲数换算置为-2147483647~2147483647的值。
- 在(s2)中指定用户单位的指令速度。应通过脉冲换算置为1pps~200kpps的值。
- 在(d1)中指定输出脉冲的轴编号。
- 在(d2)中指定指令执行完成，在(d2)+1中指定异常完成标志的位元件。
- 完成标志的动作时机如下所示。

项目	指令执行结束标志 (d2)	指令执行异常结束标志 (d2)+1
ON条件	从指定的定位地址的脉冲输出完成时开始到ON→OFF条件成立为止	在下述操作中或从功能的动作完成开始到ON→OFF条件成立为止 <ul style="list-style-type: none"> • 轴已使用 • 定位动作中驱动触点OFF • 脉冲停止指令 • 脉冲减速停止指令*1 • 行进方向的极限 • 所有的输出禁止 (SM9904) • RUN中写入 • 定位地址异常 • 通过更改为指令速度0进行减速停止
ON→OFF条件	在执行下述任意一项之前保持ON状态 <ul style="list-style-type: none"> • 用户设为OFF • 重启定位指令 	

*1 剩余距离运行有效为ON时，异常完成标志不变为ON。

■相关软元件

• 特殊继电器

控制器	高速脉冲输入输出模块				名称	高速I/O 参数	R/W
	第1台	第2台	第3台	第4台			
轴1~4	轴5~6	轴7~8	轴9~10	轴11~12			
SM5500 SM5501 SM5502 SM5503	SM5504 SM5505	SM5506 SM5507	SM5508 SM5509	SM5510 SM5511	定位指令驱动中	×	R
SM5516 SM5517 SM5518 SM5519	SM5520 SM5521	SM5522 SM5523	SM5524 SM5525	SM5526 SM5527	脉冲输出中监控	×	R
SM5532 SM5533 SM5534 SM5535	SM5536 SM5537	SM5538 SM5539	SM5540 SM5541	SM5542 SM5543	发生定位错误	×	R/W
SM5596 SM5597 SM5598 SM5599	SM5600 SM5601	SM5602 SM5603	SM5604 SM5605	SM5606 SM5607	剩余距离运行有效	×	R/W
SM5612 SM5613 SM5614 SM5615	SM5616 SM5617	SM5618 SM5619	SM5620 SM5621	SM5622 SM5623	剩余距离运行开始	×	R/W
SM5628 SM5629 SM5630 SM5631	SM5632 SM5633	SM5634 SM5635	SM5636 SM5637	SM5638 SM5639	脉冲停止指令	×	R/W
SM5644 SM5645 SM5646 SM5647	SM5648 SM5649	SM5650 SM5651	SM5652 SM5653	SM5654 SM5655	脉冲减速停止指令	×	R/W
SM5660 SM5661 SM5662 SM5663	SM5664 SM5665	SM5666 SM5667	SM5668 SM5669	SM5670 SM5671	正转极限	×	R/W
SM5676 SM5677 SM5678 SM5679	SM5680 SM5681	SM5682 SM5683	SM5684 SM5685	SM5686 SM5687	反转极限	×	R/W
SM5772 SM5773 SM5774 SM5775	SM5776 SM5777	SM5778 SM5779	SM5780 SM5781	SM5782 SM5783	旋转方向设置	○	R/W

R: 读取专用、R/W: 读取/写入用、○: 支持、×: 不支持

• 特殊寄存器

控制器	高速脉冲输入输出模块				名称	高速I/O 参数	R/W
	第1台	第2台	第3台	第4台			
轴1~4	轴5~6	轴7~8	轴9~10	轴11~12			
SD5500、SD5501 SD5540、SD5541 SD5580、SD5581 SD5620、SD5621	SD5660、SD5661 SD5700、SD5701	SD5740、SD5741 SD5780、SD5781	SD5820、SD5821 SD5860、SD5861	SD5900、SD5901 SD5940、SD5941	当前地址(用户单位)	×	R/W
SD5502、SD5503 SD5542、SD5543 SD5582、SD5583 SD5622、SD5623	SD5662、 SD5663SD5702、 SD5703	SD5742、SD5743 SD5782、SD5783	SD5822、SD5823 SD5862、SD5863	SD5902、SD5903 SD5942、SD5943	当前地址(脉冲单位)	×	R/W
SD5504、SD5505 SD5544、SD5545 SD5584、SD5585 SD5624、SD5625	SD5664、 SD5665SD5704、 SD5705	SD5744、SD5745 SD5784、SD5785	SD5824、SD5825 SD5864、SD5865	SD5904、SD5905 SD5944、SD5945	当前速度(用户单位)	×	R
SD5510 SD5550 SD5590 SD5630	SD5670 SD5710	SD5750 SD5790	SD5830 SD5870	SD5910 SD5950	定位错误 错误代码	×	R/W
SD5516、SD5517 SD5556、SD5557 SD5596、SD5597 SD5636、SD5637	SD5676、SD5677 SD5716、SD5717	SD5756、SD5757 SD5796、SD5797	SD5836、SD5837 SD5876、SD5877	SD5916、SD5917 SD5956、SD5957	最高速度	○	R/W
SD5518、SD5519 SD5558、SD5559 SD5598、SD5599 SD5638、SD5639	SD5678、SD5679 SD5718、SD5719	SD5758、SD5759 SD5798、SD5799	SD5838、SD5839 SD5878、SD5879	SD5918、SD5919 SD5958、SD5959	偏置速度	○	R/W
SD5520 SD5560 SD5600 SD5640	SD5680 SD5720	SD5760 SD5800	SD5840 SD5880	SD5920 SD5960	加速时间	○	R/W
SD5521 SD5561 SD5601 SD5641	SD5681 SD5721	SD5761 SD5801	SD5841 SD5881	SD5921 SD5961	减速时间	○	R/W

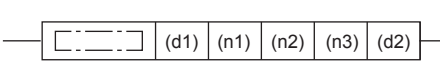
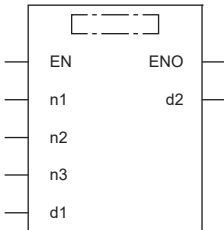
R: 读取专用、R/W: 读取/写入用、○: 支持、×: 不支持

26.6 表格运行(多个表格)

该指令可以在多个表格执行利用工程工具预先在数据表中设置的控制方式的动作。

DRVTBL

该指令通过工程工具设置的数据表格，以1个指令连续运行或步进运行多个表格。

梯形图	ST	FBD/LD
	<pre>ENO:=DRVTBL (EN, n1, n2, n3, d1, d2);</pre>	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(d1)	输出脉冲的轴编号	K1~K12	无符号BIN16位	ANY16
(n1)	执行的起始表编号	1~100	无符号BIN16位	ANY16_U
(n2)	执行的最终表编号	1~100	无符号BIN16位	ANY16_U
(n3)	表执行方法	0、1	无符号BIN16位	ANY16_U
(d2)	指令执行完成、异常完成标志的位软元件编号	—	位	ANYBIT_ARRAY (要素数: 2)
EN	执行条件	-	位	BOOL
ENO	执行结果	-	位	BOOL

■可以使用的软元件

操作数	位		字		双字		间接指定	常数			其它 (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	
(d1)	○*2	—	○	○*4	○	—	—	○	—	—	—
(n1)	○*2	—	○	○*4	○	—	—	○	—	—	—
(n2)	○*2	—	○	○*4	○	—	—	○	—	—	—
(n3)	○*2	—	○	○*4	○	—	—	○	—	—	—
(d2)*1	○*2	—	○*3	—	—	—	—	—	—	—	—

*1 从指定的软元件开始占用2点。

*2 FX、FY不能使用。

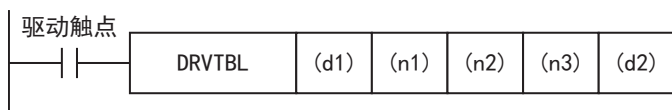
*3 T、ST、C、FD不能使用。

*4 不能使用J□\□。

功能

通过工程工具设置的数据表格，可以以1个指令连续运行或步进运行多个表格。关于表格的设置方法等的详细内容，请参阅下述手册。

📖所使用的控制器的用户手册



- 在(d1)中指定输出脉冲的轴编号。
- 在(n1)中指定，通过(d1)中指定的输出要执行的起始表。
- 在(n2)中指定最终的表格。
- 在(n3)中指定表格运行方式。以K0进行步进运行，以K1进行连续运行。
- 在(d2)中指定指令执行完成，在(d2)+1中指定异常完成标志的位软元件。
- (n1)=(n2)时，仅执行1表格。
- (n1)>(n2)时，执行最终表格(100个表格)后正常完成。
- (n3)为K0(步进运行)时，在1个表格结束时若检测出表格转移指令，则将转移到下一个表格中。此外，也可以通过外部开始信号转移表格。
- 完成标志的动作时机根据表格的控制方式而不同。

■相关软元件

- 特殊继电器

控制器	高速脉冲输入输出模块				名称	高速I/O参数	R/W
	第1台	第2台	第3台	第4台			
轴1~4	轴5~6	轴7~8	轴9~10	轴11~12			
SM5580 SM5581 SM5582 SM5583	SM5584 SM5585	SM5586 SM5587	SM5588 SM5589	SM5590 SM5591	表格转移指令	×	R/W
SM5916 SM5917 SM5918 SM5919	SM5920 SM5921	SM5922 SM5923	SM5924 SM5925	SM5926 SM5927	定位表格数据初始化无效	×	R/W

R/W: 读取/写入用、×: 不支持

- 特殊寄存器

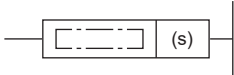
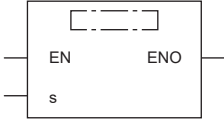
控制器	高速脉冲输入输出模块				名称	高速I/O参数	R/W
	第1台	第2台	第3台	第4台			
轴1~4	轴5~6	轴7~8	轴9~10	轴11~12			
SD5506 SD5546 SD5586 SD5626	SD5670 SD5706	SD5746 SD5786	SD5826 SD5866	SD5906 SD5946	定位执行中的表格编号	×	R
SD5511 SD5551 SD5591 SD5631	SD5671 SD5711	SD5751 SD5791	SD5831 SD5871	SD5911 SD5951	定位错误 发生错误表格编号	×	R/W

R: 读取专用、R/W: 读取/写入用、×: 不支持

26.7 中断输入1触发

该指令用于使与输入中断1信号同等的触发发生，不使用X软元件进行触发。

DITRG

梯形图	ST	FBD/LD
	ENO:=DITRG(EN, s);	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	发生输入中断1信号的轴编号	K1~K4	无符号BIN16位	ANY16
EN	执行条件	-	位	BOOL
ENO	执行结果	-	位	BOOL

■可以使用的软元件

操作数	位		字		双字		间接指定	常数			其它(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	
(s)	○*1	—	○	○*2	○	—	—	○	—	—	—

*1 FX、FY不能使用。

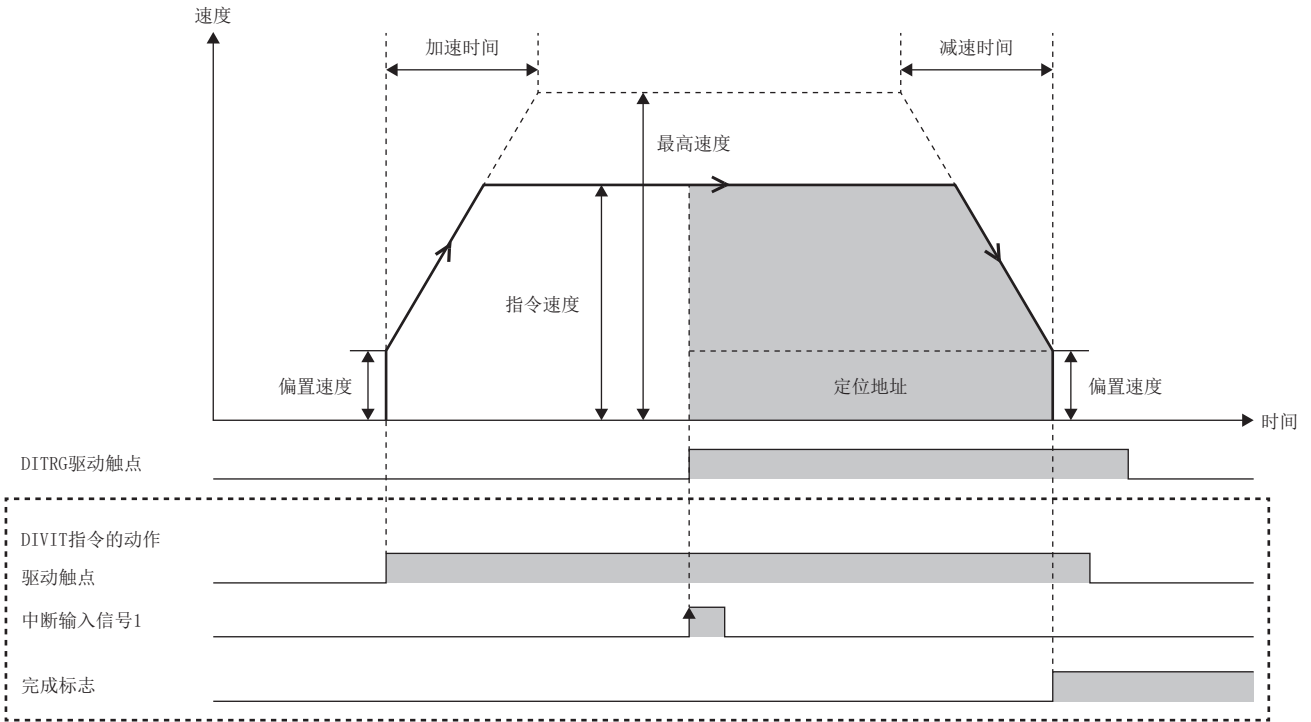
*2 不能使用J□\□。

功能

发生输入中断1信号。在(s)中指定输出输入中断1信号的轴编号。



- 本指令的特性上需要与其他指令组合使用。与DVIT指令组合使用的示例如下所示。



注意事项

- 为了提高从指令输入到触发开始的精度，应在中断程序内使用。
- 由于DITRG指令在上升沿动作，因此在中断程序内使用时，应编程为可检测驱动触点OFF的程序。

■关于与输入中断1信号的关系

DITRG指令与输入中断1信号将先到达的指令作为触发动作。动作的关系如下所示。

使用输入中断1信号的指令	触发		动作
	输入中断1信号	DITRG指令	
OFF	OFF	OFF→ON	不执行任何动作。
OFF	OFF→ON	OFF	不执行任何动作。
OFF	ON	OFF→ON	不执行任何动作。
OFF	OFF→ON	ON	不执行任何动作。
ON	OFF	OFF	不执行任何动作。
ON	OFF	OFF→ON	DITRG指令为触发。
OFF→ON	OFF	ON	执行指令后立即发生DITRG指令的触发。
ON	OFF→ON	OFF	输入中断1信号为触发。
OFF→ON	ON	OFF	不执行任何动作。 →即使在使用输入中断1信号的指令的驱动触点ON之前接收到输入中断1信号也不进行处理。之后，将输入中断1信号由OFF→ON，或接收到DITRG指令时将发生触发。
ON	ON	OFF→ON	不执行任何动作。 →输入中断1信号为触发，此后即使接收DITRG指令也不进行处理。此外，DITRG指令不会发生错误。
ON	OFF→ON	ON	不执行任何动作。 →DITRG指令为触发，此后即使接收输入中断1信号也不进行处理。
OFF→ON	ON	ON	不执行任何动作。

■关于与输入中断信号1的差异

中断输入信号1与DITRG指令的规格的差异如下所示。

有差异的项目	中断输入信号1	DITRG指令
参数：中断输入信号1	[0：无效]→不发生中断 [1：有效]→发生中断	[0：无效]→发生中断 [1：有效]→发生中断
操作数改写	支持	不支持

26.8 绝对位置恢复

与支持绝对位置恢复功能的伺服放大器连接，读取伺服放大器的绝对位置 (ABS) 数据。
读取数据时转换为脉冲值。

DABRST

梯形图	ST	FBD/LD
	ENO:=DABRST (EN, s, d1, d2, d3);	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)
(s)	输入伺服放大器发出的绝对值 (ABS) 数据用输出信号的软元件起始编号	—	位	ANYBIT_ARRAY (要素数: 3)
(d1)	向伺服放大器输出绝对值 (ABS) 数据用控制信号的软元件起始编号	—	位	ANYBIT_ARRAY (要素数: 3)
(d2)	绝对值 (ABS) 数据 (32位值) 的存储目标软元件编号	—	带符号BIN32位	ANY32
(d3)	绝对位置 (ABS) 数据读取完成、异常完成标志的位软元件编号	—	位	ANYBIT_ARRAY (要素数: 2)
EN	执行条件	-	位	BOOL
ENO	执行结果	-	位	BOOL

■可以使用的软元件

操作数	位		字		双字		间接指定	常数			其它 (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	
(s)	○*2	—	○	—	—	—	—	—	—	—	—
(d1)	○*2	—	○	—	—	—	—	—	—	—	—
(d2)	○*2	—	○	○*4	○	—	○	—	○	—	—
(d3)*1	○*2	—	○*3	—	—	—	—	—	—	—	—

*1 从指定的软元件开始占用2点。

*2 FX、FY不能使用。

*3 T、ST、C、FD不能使用。

*4 不能使用J□\□。

功能

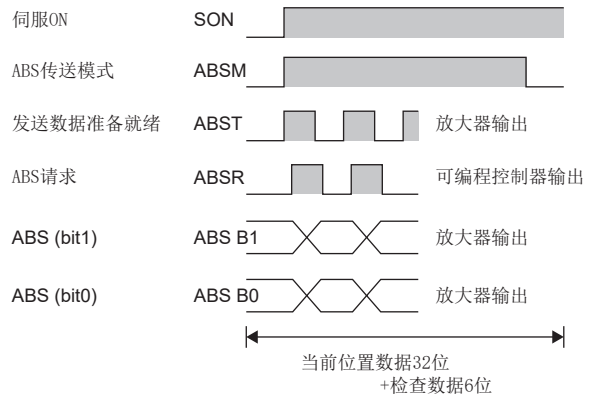
与伺服放大器连接，读取绝对位置(ABS)数据。读取数据时转换为脉冲值。



- 在(s)中指定输入伺服放大器发出的ABS数据用输出信号的软元件起始编号。以(s)为起始占用3点。在(s)中指定ABS(bit0)，在(s)+1中指定ABS(bit1)，在(s)+2中指定发送数据准备就绪信号。
- 在(d1)指定向伺服放大器输出ABS数据用控制信号的软元件起始编号。以(d1)为起始占用3点。在(d1)中指定伺服ON，在(d1)+1中指定ABS传送模式，在(d1)+2中指定ABS请求信号。
- 在(d2)中指定存储从伺服放大器读取的ABS数据的软元件编号。应通过脉冲数换算置为-2147483647~2147483647的值。
- 在(d3)中指定指令执行完成，在(d3)+1中指定异常完成标志的位软元件。

绝对位置的检测动作

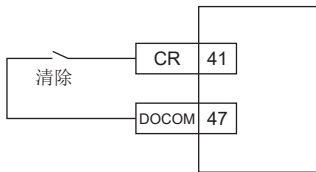
1. 可编程控制器驱动DABRST指令时，将驱动伺服ON输出和ABS传送模式输出。
2. 通过发送数据准备就绪信号和ABS请求信号确认相互发送接收的同时，进行32+6位的数据通信。
3. 数据使用ABS bit0、bit1的2位线路进行。
4. ABS数据读取完成时执行完成标志将动作。



关于初次原点定位

即使是带绝对位置检测功能的伺服电机，在装置制作时也至少需要进行1次原点定位，并对伺服电机提供清除信号。应通过以下任意一种方法进行初次原点定位。

- 应通过DSZR/DDSZR指令将清除信号功能设为有效，并进行原点复位。
- 通过JOG运行或通过手动进行位置调整等进行装置的原点定位后，应在伺服放大器中输入清除信号。清除信号的输入应使用可编程控制器的输出，或通过下图所示的外部开关进行。



注意事项

伺服放大器的详细内容，请参阅所使用的各伺服放大器的手册。

- 设计系统时应将控制器及伺服放大器的电源接通时机设计为同时接通或伺服放大器侧先接通。
- 驱动触点应在读取ABS值后仍保持ON状态。ABS值的读取完成后，如果将指令的驱动触点置为OFF，伺服ON(SON)信号将变为OFF，因此不动作。
- 读取中驱动触点置为了OFF的情况下，读取将中断。
- 即使无法与伺服放大器进行数据通信，也不会检测出错误，因此需要通过超时判定用定时器进行监视。
- 使用DABRST指令时，应将伺服电机的旋转方向设置为下述关系。其他设置的情况下，读取ABS值后，可编程控制器管理的当前值与伺服放大器内的符号(正负)可能不一致，应加以注意。

旋转方向	伺服放大器的设置
根据正转脉冲增加当前值	正转脉冲输入时正转 (CCW) 反转脉冲输入时反转 (CW)
根据反转脉冲减少当前值	正转脉冲输入时反转 (CW) 反转脉冲输入时正转 (CCW)

出错

错误代码	内容
1811H	同时执行了17个及以上的ABRST指令时。
1820H	对ABRST指令动作中的程序文件进行了RUN中写入时。
2820H	指令的操作数中设置的软元件超出范围时。
36F0H	来自伺服的ABS数据的和不一致时。

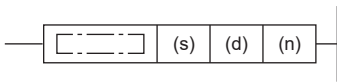
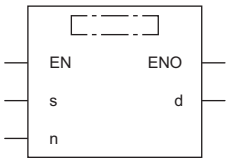
27 高速计数器功能专用指令

27.1 高速IO用SM/SD的读取/写入(更新)

16位数据读取/写入(更新)

HCMOV (P)

该指令用于进行高速计数器/脉冲宽度测定/PWM/定位用特殊继电器/特殊寄存器的读取、写入(更新)。

梯形图	ST
	ENO:=HCMOV (EN, s, n, d) ; ENO:=HCMOVP (EN, s, n, d) ;
FBD/LD	
	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	传送源的软元件编号	—	位/有符号BIN16位	ANY_ELEMENTARY
(d)	传送目标软元件编号	—	位/有符号BIN16位	ANY_ELEMENTARY
(n)	传送后, 显示的传送源软元件的清除提示	K0、K1	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可以使用的软元件

操作数	位	字			双字		间接指定	常数			其它
	X、Y、M、L、 SM、F、B、SB	T、ST、C、D、W、SD、 SW、R、ZR、RD	U□\G□、 U3E□\G□	Z	LT、LST、 LC	LZ		K、H	E	\$	
(s)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	○	—	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—	—

功能

将(s)中指定的软元件值传送到(d)中指定的软元件。此时, 如果(n)的值为K0, 则保持(s)的值。(n)的值为K1时, 传送后将(s)的值清零。

要点

- (s)为支持高速传送的软元件时
 - 执行HCMOV指令时, 获取高速计数器当前值等的最新值后, 传送到(d)。
- (d)为支持高速传送的软元件时
 - 执行HCMOV指令时, 可以更改高速计数器当前值等值。

■HCMOV指令的效果

- 同时使用输入中断与HCMOV指令的情况下, 可在外部输入处于上升沿或下降沿时, 获取高速计数器的当前值。
- 在比较指令(CMP指令/ZCP指令/比较触点指令)之前使用HCMOV指令时, 可用高速计数器的最新值进行比较。HCMOV指令的使用次数无限制, 因此可以多次使用。

注意事项

- 在高速计数器的当前值发生变化时进行比较，如需输出，应使用高速比较表。
- 使用HCMOV指令，对支持高速传送的BIN32位特殊软元件(高速计数器当前值等)执行读取或写入操作时，其动作方式与普通的MOV指令相似。不能更新为最新值或改写内部寄存器。
- 执行脉冲密度(转数计测)、SPD指令的过程中，请勿使用HCMOV指令改写高速计数器的当前值。
- 不能进行支持HCMOV的SM/SD之间的传送。

要点

HCMOV指令主要用于读取高速计数器/脉冲宽度测定的当前值，或更改定位的当前地址(用户单位)、当前地址(脉冲单位)。

出错

出错代码(SD0)	内容
2801H	指定了不存在对象模块的通道编号时。
2821H	指定在支持高速传送的SM和支持高速运转的SD间传送的操作数时。
36C5H	执行指令时与对象模块的通信中发生了超时时。
36C7H	执行指令时在访问对象模块中检测到信号异常时。
3285H	(n)中设置了超出数据范围的值时。
3602H	在中断程序的中断优先级为1~16的情况下，驱动了指定了不能作为HCMOV指令的操作数的软元件的HCMOV指令时。

32位数据读取/写入(更新)

DHCMOV (P)

该指令用于进行高速计数器/脉冲宽度测定/PWM/定位用特殊继电器/特殊寄存器的读取、写入(更新)。

梯形图	ST
	ENO:=DHCMOV (EN, s, n, d) ; ENO:=DHCMOVP (EN, s, n, d) ;

FBD/LD

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	传送源的软元件编号	—	位/有符号BIN32位	ANY_ELEMENTARY
(d)	传送目标软元件编号	—	位/有符号BIN32位	ANY_ELEMENTARY
(n)	传送后, 显示的传送源软元件的清除提示	K0、K1	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可以使用的软元件

操作数	位	字	双字			间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB	T、ST、C、D、W、SD、SW、R、ZR、RD	U□\G□、U3E□\G□	Z	LT、LST、LC		LZ	K、H	E	
(s)	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	○	○	○	—	—	—
(n)	○	○	○	○	—	—	○	○	—	—

功能

将(s)中指定的软元件值传送至(d)中指定的软元件。此时,如果(n)的值为K0,则保留(s)的值。(n)的值为K1时,传送后将(s)的值清零。

要点

- (s)为支持高速传送的软元件时
 - 执行DHCMOV指令时,获取高速计数器当前值等的最新值后,传送至(d)。
- (d)为支持高速传送的软元件时
 - 执行DHCMOV指令时,可以更改高速计数器当前值等值。

■DHCMOV指令的效果

- 同时使用输入中断与DHCMOV指令的情况下,可在外部输入处于上升沿或下降沿时,获取高速计数器的当前值。
- 在比较指令(DCMP指令/DZCP指令/比较触点指令)之前使用DHCMOV指令时,可用高速计数器的最新值进行比较。DHCMOV指令的使用次数无限制,因此可以多次使用。

注意事项

- 在高速计数器的当前值发生变化时进行比较,如需输出,应使用高速比较表。
- 执行脉冲密度(转数计数)、DSPD指令的过程中,请勿使用DHCMOV指令改写高速计数器的当前值。
- 请勿使用DHCMOV指令指定跨越BIN32位的SD边界的操作数。设定值可能会损坏。
- 不能在支持高速传送的SM和支持高速运转的SD间传送。
- 将高速计数器SD的预置值设置为大于环形长度的值时,如果在高速计数器停止中设置,则计数开始时会发生错误,但在高速计数器动作中设置的情况下不会发生错误,而是反映设定值。
- 分别读取高速计数器的SD软元件(当前值/最大值/最小值),仅更新读取的SD软元件。因此,高速计数器的SD软元件可能存在暂时不符合最小值 \leq 当前值 \leq 最大值关系的情况。

要点

DHCMOV指令主要用于读取高速计数器/脉冲宽度测定的当前值,或更改定位的当前地址(用户单位)、当前地址(脉冲单位)。

出错

出错代码 (SD0/SD8067)	内容
2801H	指定了不存在对象模块的通道编号时。
2821H	指定在支持高速传送的SM和支持高速运转的SD间传送的操作数时。
36C5H	执行指令时与对象模块的通信中发生了超时时。
36C7H	执行指令时在访问对象模块中检测到信号异常时。
3285H	(n)中设置了超出数据范围的值时。

28.1 脉冲宽度调制

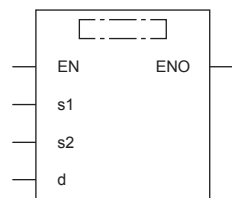
BIN16位脉冲宽度调制

PWMH

将(s1)中指定的ON时间(16位数据单位)及(s2)中指定的周期的脉冲(16位数据单位)输出到(d)中指定的输出目标中。

梯形图	ST
	ENO:=PWMH(EN, s1, s2, d);

FBD/LD



设置数据

■内容、范围、数据类型

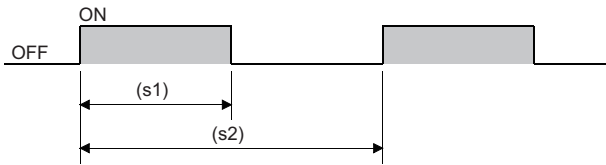
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储了ON时间(脉冲宽度数据)的软元件编号	1~65535	无符号BIN16位	ANY16
(s2)	存储了周期数据的软元件编号	1~65535	无符号BIN16位	ANY16
(d)	输出脉冲的通道编号	1~12	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可以使用的软元件

操作数	位	字	双字			间接指定	常数			其它	
	X、Y、M、L、SM、F、B、SB	T、ST、C、D、W、SD、SW、R、ZR、RD	U□\G□、U3E□\G□	Z	LT、LST、LC		LZ	K、H	E		\$
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(d)	○	○	○	○	—	—	—	○	—	—	—

功能

- 将(s1)中指定的ON时间及(s2)中指定的周期的脉冲输出到(d)中指定的输出目标中。



- 关于(s1)、(s2)中指定的时间，可指定为参数设置画面中选择的单位(μs或ms)的时间。
- 关于(d)中可指定的通道编号，可指定为参数设置画面中选择的输出目标。
- 从各通道输出的脉冲数、脉冲宽度、周期存储到SD软件元件中。脉冲宽度、周期以参数中设置的单位存储。脉冲输出指定了0的情况下，输出变为无限。

脉冲输出目标通道	脉冲输出数	R/W	初始值	反映到动作的时机	清除为初始值的时机
通道1	SD5301、SD5300	R/W	0	<ul style="list-style-type: none"> • 执行HCMOV指令时*1 • 执行PWMH指令时 • END处理 	STOP/PAUSE→RUN
通道2	SD5317、SD5316				
通道3	SD5333、SD5332				
通道4	SD5349、SD5348				
CH5	SD5364、SD5365				
CH6	SD5380、SD5381				
CH7	SD5396、SD5397				
CH8	SD5412、SD5413				
CH9	SD5428、SD5429				
CH10	SD5444、SD5445				
CH11	SD5460、SD5461				
CH12	SD5476、SD5477				

脉冲输出目标通道	ON时间	R/W	初始值	反映到动作的时机	清除为初始值的时机
通道1	SD5303、SD5302	R/W	0*2	<ul style="list-style-type: none"> • 执行HCMOV指令时*1 • 执行PWMH指令时*3 • END处理 	STOP/PAUSE→RUN
通道2	SD5319、SD5318				
通道3	SD5335、SD5334				
通道4	SD5351、SD5350				
CH5	SD5366、SD5367				
CH6	SD5382、SD5383				
CH7	SD5398、SD5399				
CH8	SD5414、SD5415				
CH9	SD5430、SD5431				
CH10	SD5446、SD5447				
CH11	SD5462、SD5463				
CH12	SD5478、SD5479				

脉冲输出目标通道	周期	R/W	初始值	反映到动作的时机	清除为初始值的时机
通道1	SD5305、SD5304	R/W	0*2	<ul style="list-style-type: none"> • 执行HCMOV指令时*1 • 执行PWMH指令时*3 • END处理 	STOP/PAUSE→RUN
通道2	SD5321、SD5320				
通道3	SD5337、SD5336				
通道4	SD5353、SD5352				
CH5	SD5368、SD5369				
CH6	SD5384、SD5385				
CH7	SD5400、SD5401				
CH8	SD5416、SD5417				
CH9	SD5432、SD5433				
CH10	SD5448、SD5449				
CH11	SD5464、SD5465				
CH12	SD5480、SD5481				

*1 使用DHCMOV指令时，可读取最新值。如果是可写入的软件元件，可立即更新。

- *2 在参数中设置的项目当STOP→RUN时，参数的设置值设置到SD软元件。
- *3 执行PWMH指令时，(s1)、(s2)中指定的脉冲宽度、周期设置到SD软元件。
- 如果开始从各通道输出脉冲，脉冲输出中监控置为ON。

脉冲输出目标通道	脉冲输出中监控	R/W	初始值	置为ON的时机	置为OFF的时机
通道1	SM5300	R	OFF	<ul style="list-style-type: none"> • 执行HIOEN指令 • 执行PWMH指令 	<ul style="list-style-type: none"> • 电源ON • 复位 • RUN→STOP/PAUSE • 指定的脉冲数输出结束 • 驱动触点置为OFF
通道2	SM5301				
通道3	SM5302				
通道4	SM5303				
CH5	SM5304				
CH6	SM5305				
CH7	SM5306				
CH8	SM5307				
CH9	SM5308				
CH10	SM5309				
CH11	SM5310				
CH12	SM5311				

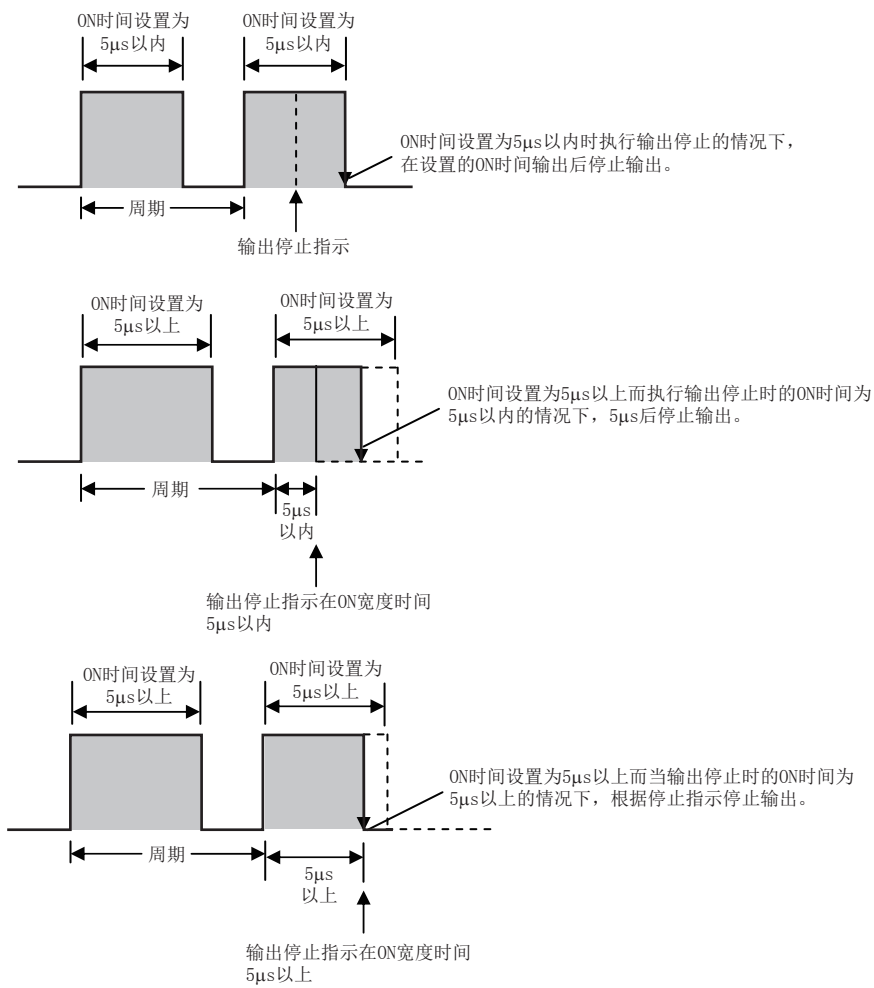
- 存储从各通道已输出的脉冲数。

脉冲输出目标通道	脉冲输出数当前值监控	R/W	初始值	反映到动作的时机	清除为初始值的时机
通道1	SD5307、SD5306	R/W	0	<ul style="list-style-type: none"> • 执行DHCMOV指令时 →更新SD软元件 • 执行PWMH指令 • END处理 	<ul style="list-style-type: none"> • 电源ON • 复位 • STOP/PAUSE→RUN
通道2	SD5323、SD5322				
通道3	SD5339、SD5338				
通道4	SD5355、SD5354				

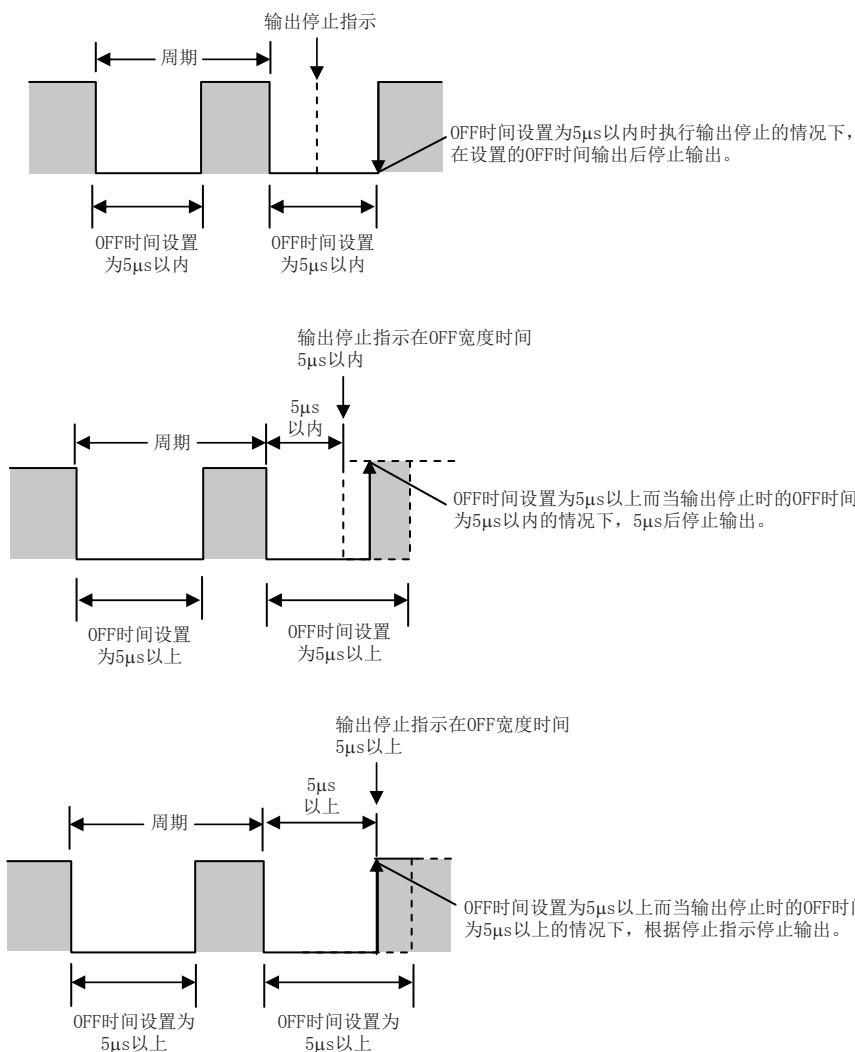
- SD软元件中设置的脉冲输出数的设置在PWMH指令中也有效，始终读取设置值并更新。
- 写入了脉冲输出数设置≤已输出脉冲数的情况下，输出中的脉冲输出后停止。
- 写入了脉冲输出数设置>已输出脉冲数的情况下，设置的脉冲输出后停止。
- 如果脉冲输出数设置是从无限制输出设置(脉冲输出数=0)中设置了脉冲输出数，则脉冲输出数不更新。(在无限制输出中有脉冲输出仍继续和停止的情况，因此不更新)
- 执行1次PWMH指令可输出的脉冲输出数(SD软元件中可设置的值)为2147483647。
- 脉冲输出中也可进行ON时间、周期设置。始终读取设置值并更新。
- 脉冲输出数为无限制输出设置(脉冲输出数=0)的情况下，脉冲输出数当前值监控变为0。
- 指定了脉冲输出数的情况下，对已输出的脉冲数进行监控。执行多次PWMH输出的情况下，脉冲输出数的监控将为累计的数值。
- 脉冲输出数当前值监控的更改也可在脉冲输出中进行。
- 脉冲输出数的当前值监控在正逻辑的情况下在脉冲的下降沿，负逻辑的情况下在脉冲的上升沿进行脉冲数的递增计数并更新。
- 输出始终为ON或OFF的情况下，脉冲输出数的当前值监控不变化。
- 脉冲输出数的当前值监控的最大值为FFFFFFFFH。脉冲输出数的当前值监控达到最大值后，从0再次计数。

注意事项

- (s1) 中指定的ON时间和 (s2) 中指定的周期的值应指定为 (s1): $2\mu\text{s}$ 及以上、(s2): $5\mu\text{s}$ 及以上。
- 在参数设置中将PWM输出未选择的通道编号指定到 (d) 中的情况下, PWMH指令不执行。变为运算错误。
- PWM输出停止时的动作 (输出脉冲为ON时)



• PWM输出停止时的动作 (输出脉冲为OFF时)



- 所有的输出禁止指令(SM9904)为ON时停止PWM输出，为OFF时开始PWM输出。
- 即使定位用的脉冲停止指令驱动，PWM输出也不停止。
- 指定脉冲输出数并执行PWMH指令，输出指定的脉冲数并停止后进行下一个脉冲输出的情况下，应将驱动PWMH指令的触点置为OFF。通过HIOEN指令使PWM输出动作的情况下，请先通过HIOEN指令将PWM输出OFF。
- 周期和ON时间设置相同的情况下，始终输出ON状态。在该状态下经过(周期)×(输出脉冲数)后，ON状态仍将继续。
- 即使通过RUN中写入添加PWMH指令，也不会开始脉冲输出。驱动触点为ON时，通过将驱动触点置为OFF→ON，可以开始脉冲输出。
- 发生RUN中写入时进行了PWMH指令的更改/删除的情况下，将停止脉冲输出。
- 在脉冲输出中存在PWMH指令的状态下发生了RUN中写入时，停止脉冲输出，并放弃对停止的通道控制权。

出错

错误代码(SD0)	内容
1810H	(d)中指定的输出目标已在其他指令(定位指令)中使用时。
1820H	对PWMH指令动作中的程序文件进行了RUN中写入时。
2221H	参数的设置值超出可使用范围。
3285H	(s1)、(s2)中设置了超出范围的数据时。
3600H	(d)中指定的输出目标指定了未在参数设置中选择的通道编号时。
3611H(通道1) 3612H(通道2) 3613H(通道3) 3614H(通道4)	以(s2) < (s1)的设置执行了本指令时。

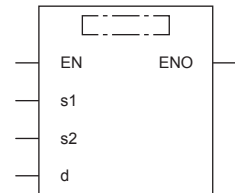
BIN32位脉冲宽度调制

DPWMH

将(s1)中指定的ON时间(32位数据单位)及(s2)中指定的周期的脉冲(32位数据单位)输出到(d)中指定的输出目标中。

<p>梯形图</p>	<p>ST</p> <p>ENO:=DPWM(EN, s1, s2, d);</p>
-------------------	---

FBD/LD



设置数据

■内容、范围、数据类型

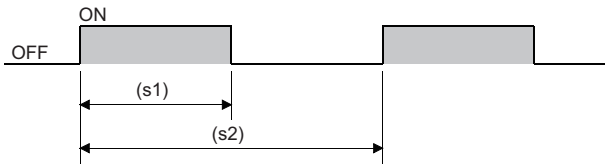
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	存储了ON时间(脉冲宽度数据)的软元件编号	1~2147483647	无符号BIN32位	ANY32
(s2)	存储了周期数据的软元件编号	1~2147483647	无符号BIN32位	ANY32
(d)	输出脉冲的通道编号	1~12	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可以使用的软元件

操作数	位	字	双字			间接指定	常数			其它
	X、Y、M、L、SM、F、B、SB	T、ST、C、D、W、SD、SW、R、ZR、RD	U□\G□、U3E□\G□	Z	LT、LST、LC		LZ	K、H	E	
(s1)	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	○	—	—	—	○	—	—

功能

- 将(s1)中指定的ON时间及(s2)中指定的周期的脉冲输出到(d)中指定的输出目标中。



- 关于(s1)、(s2)中指定的时间，可指定为参数设置画面中选择的单位(μs或ms)的时间。
- 关于(d)中可指定的通道编号，可指定为参数设置画面中选择的输出目标。
- 从各通道输出的脉冲数、脉冲宽度、周期存储到SD软件元件中。脉冲宽度、周期以参数中设置的单位存储。脉冲输出指定了0的情况下，输出变为无限。

脉冲输出目标通道	脉冲输出数	R/W	初始值	反映到动作的时机	清除为初始值的时机
通道1	SD5301、SD5300	R/W	0	<ul style="list-style-type: none"> • 执行DHCMOV指令时*1 • 执行DPWMH指令时 • END处理 	STOP/PAUSE→RUN
通道2	SD5317、SD5316				
通道3	SD5333、SD5332				
通道4	SD5349、SD5348				
CH5	SD5364、SD5365				
CH6	SD5380、SD5381				
CH7	SD5396、SD5397				
CH8	SD5412、SD5413				
CH9	SD5428、SD5429				
CH10	SD5444、SD5445				
CH11	SD5460、SD5461				
CH12	SD5476、SD5477				

脉冲输出目标通道	ON时间	R/W	初始值	反映到动作的时机	清除为初始值的时机
通道1	SD5303、SD5302	R/W	0*2	<ul style="list-style-type: none"> • 执行DHCMOV指令时*1 • 执行DPWMH指令时*3 • END处理 	STOP/PAUSE→RUN
通道2	SD5319、SD5318				
通道3	SD5335、SD5334				
通道4	SD5351、SD5350				
CH5	SD5366、SD5367				
CH6	SD5382、SD5383				
CH7	SD5398、SD5399				
CH8	SD5414、SD5415				
CH9	SD5430、SD5431				
CH10	SD5446、SD5447				
CH11	SD5462、SD5463				
CH12	SD5478、SD5479				

脉冲输出目标通道	周期	R/W	初始值	反映到动作的时机	清除为初始值的时机
通道1	SD5305、SD5304	R/W	0*2	<ul style="list-style-type: none"> • 执行DHCMOV指令时*1 • 执行DPWMH指令时*3 • END处理 	STOP/PAUSE→RUN
通道2	SD5321、SD5320				
通道3	SD5337、SD5336				
通道4	SD5353、SD5352				
CH5	SD5368、SD5369				
CH6	SD5384、SD5385				
CH7	SD5400、SD5401				
CH8	SD5416、SD5417				
CH9	SD5432、SD5433				
CH10	SD5448、SD5449				
CH11	SD5464、SD5465				
CH12	SD5480、SD5481				

*1 使用DHCMOV指令时，可读取最新值。如果是可写入的软件元件，可立即更新。

- *2 在参数中设置的项目当STOP→RUN时，参数的设置值设置到SD软元件。
- *3 执行DPWMH指令时，(s1)、(s2)中指定的脉冲宽度、周期设置到SD软元件。
- 如果开始从各通道输出脉冲，脉冲输出中监控置为ON。

脉冲输出目标通道	脉冲输出中监控	R/W	初始值	置为ON的时机	置为OFF的时机
通道1	SM5300	R	OFF	<ul style="list-style-type: none"> • 执行DHIOEN指令 • 执行DPWMH指令 	<ul style="list-style-type: none"> • 电源ON • 复位 • RUN→STOP/PAUSE • 指定的脉冲数输出结束 • 驱动触点置为OFF
通道2	SM5301				
通道3	SM5302				
通道4	SM5303				
CH5	SM5304				
CH6	SM5305				
CH7	SM5306				
CH8	SM5307				
CH9	SM5308				
CH10	SM5309				
CH11	SM5310				
CH12	SM5311				

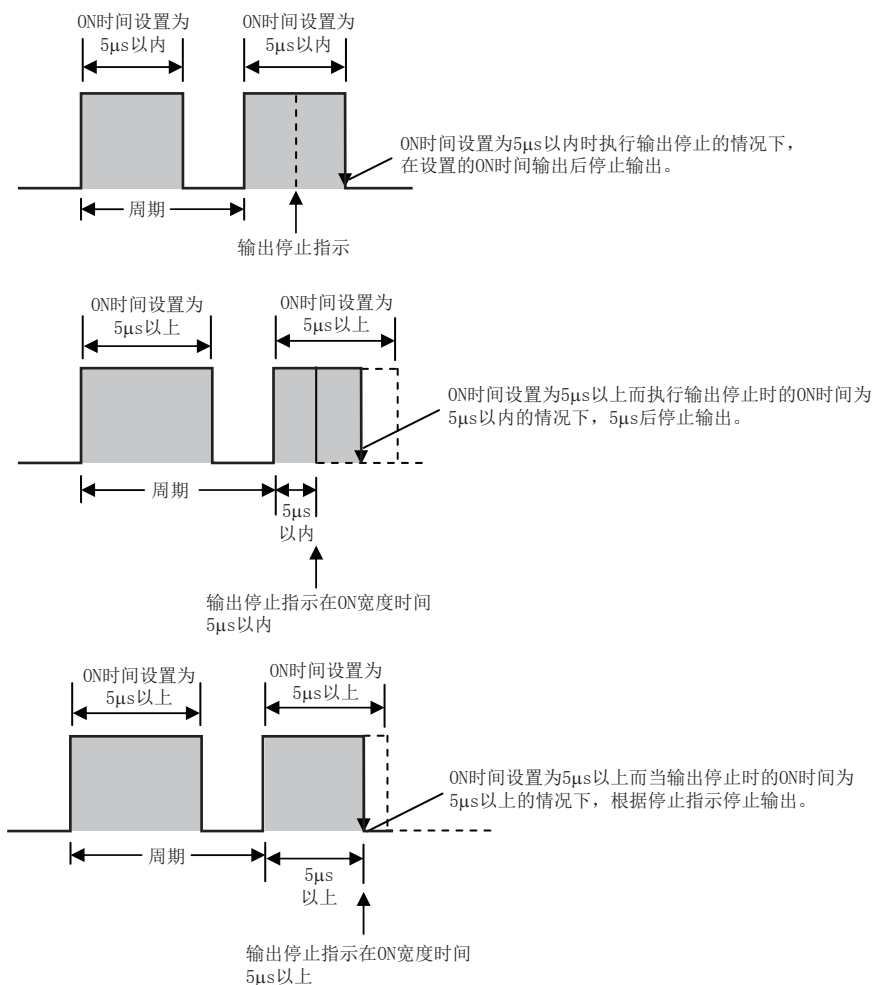
- 存储从各通道已输出的脉冲数。

脉冲输出目标通道	脉冲输出数当前值监控	R/W	初始值	反映到动作的时机	清除为初始值的时机
通道1	SD5307、SD5306	R/W	0	<ul style="list-style-type: none"> • 执行DHCMOV指令时 →更新SD软元件 • END处理 	<ul style="list-style-type: none"> • 电源ON • 复位 • STOP/PAUSE→RUN
通道2	SD5323、SD5322				
通道3	SD5339、SD5338				
通道4	SD5355、SD5354				

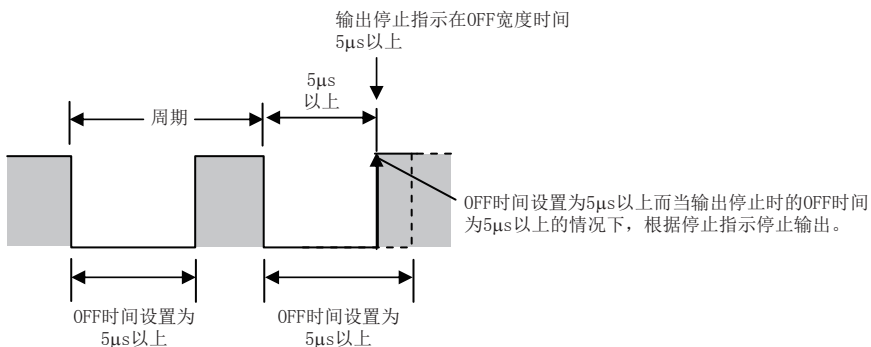
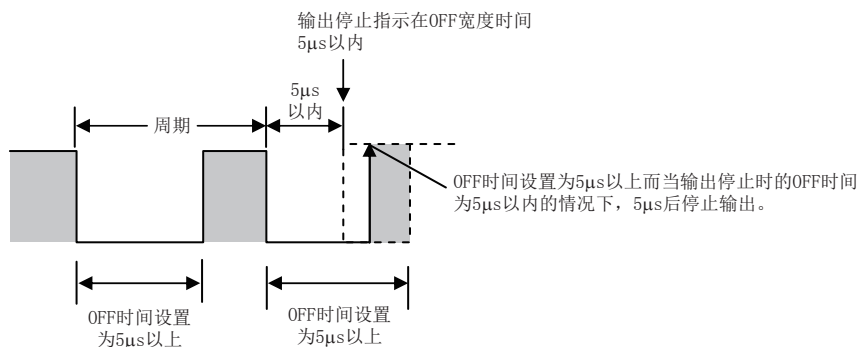
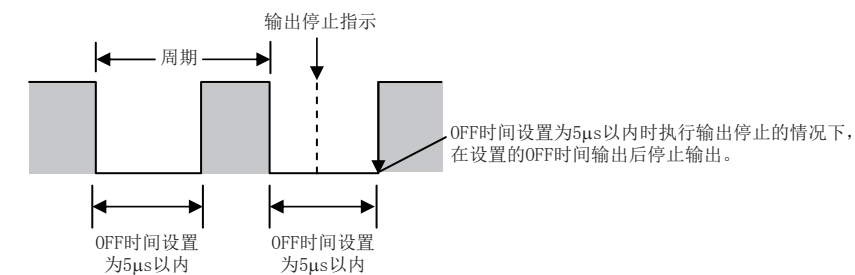
- SD软元件中设置的脉冲输出数的设置在DPWMH指令中也有效，始终读取设置值并更新。
- 写入了脉冲输出数设置≤已输出脉冲数的情况下，输出中的脉冲输出后停止。
- 写入了脉冲输出数设置>已输出脉冲数的情况下，设置的脉冲输出后停止。
- 如果脉冲输出数设置是从无限制输出设置(脉冲输出数=0)中设置了脉冲输出数，则脉冲输出数不更新。(在无限制输出中有脉冲输出仍继续和停止的情况，因此不更新)
- 执行1次DPWMH指令可输出的脉冲输出数(SD软元件中可设置的值)为2147483647。
- 脉冲输出中也可进行ON时间、周期设置。始终读取设置值并更新。
- 脉冲输出数为无限制输出设置(脉冲输出数=0)的情况下，脉冲输出数当前值监控变为0。
- 指定了脉冲输出数的情况下，对已输出的脉冲数进行监控。执行多次DPWMH输出的情况下，脉冲输出数的监控将为累计的数值。
- 脉冲输出数当前值监控的更改也可在脉冲输出中进行。
- 脉冲输出数的当前值监控在正逻辑的情况下在脉冲的下降沿，负逻辑的情况下在脉冲的上升沿进行脉冲数的递增计数并更新。
- 输出始终为ON或OFF的情况下，脉冲输出数的当前值监控不变化。
- 脉冲输出数的当前值监控的最大值为FFFFFFFFH。脉冲输出数的当前值监控达到最大值后，从0再次计数。

注意事项

- (s1)中指定的ON时间和(s2)中指定的周期的值应指定为(s1): $2\mu\text{s}$ 及以上、(s2): $5\mu\text{s}$ 及以上。
- (s1)中指定的ON时间和(s2)中指定的周期设置为负值的情况下, 变为运算出错。
- 在参数设置中将PWMH输出未选择的通道编号指定到(d)中的情况下, DPWMH指令不执行。变为运算错误。
- DPWMH输出停止时的动作(输出脉冲为ON时)



• DPWMH输出停止时的动作(输出脉冲为OFF时)



- 所有的输出禁止指令(SM9904)为ON时停止PWM输出, 为OFF时开始PWM输出。
- 即使定位用的脉冲停止指令驱动, PWM输出也不停止。
- 指定脉冲输出数并执行DPWMH指令, 输出指定的脉冲数并停止后进行下一个脉冲输出的情况下, 应将驱动DPWMH指令的触点置为OFF。通过DHIOEN指令使PWM输出动作的情况下, 请先通过DHIOEN指令将PWM输出OFF。
- 周期和ON时间设置相同的情况下, 始终输出ON状态。在该状态下经过(周期)×(输出脉冲数)后, ON状态仍将继续。

出错

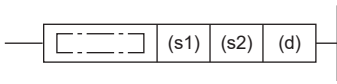
错误代码(SD0)	内容
1810H	(d)中指定的输出目标已在其他指令(定位指令)中使用时。
1820H	发生RUN中写入时本指令正在动作时。
2221H	参数的设置值超出可使用范围。
3285H	(s1)、(s2)中设置了超出范围的数据时。
3600H	(d)中指定的输出目标指定了未在参数设置中选择的通道编号时。
3611H(通道1) 3612H(通道2) 3613H(通道3) 3614H(通道4)	以(s2)<(s1)的设置执行了本指令时。

28.2 脉冲密度的测定

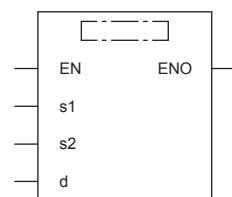
16位数据脉冲密度的测定

SPDH

通过中断输入测定指定时间的输入脉冲。

梯形图	ST
	ENO:=SPDH(EN, s1, s2, d);

FBD/LD



设置数据

■内容、范围、数据类型

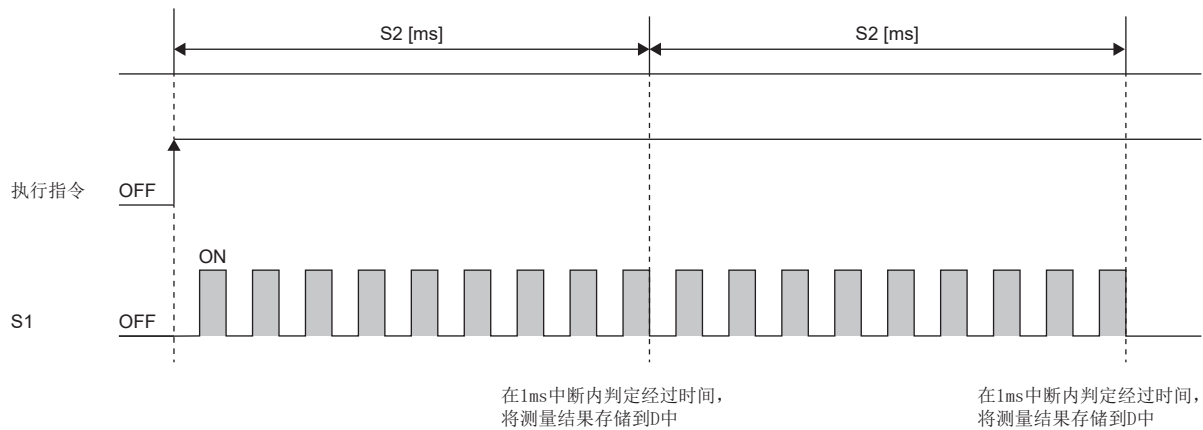
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	输入脉冲的高速计数器通道编号	1~8	无符号BIN16位	ANY16
(s2)	时间(ms)数据或存储了数据的字软元件	0~65535	无符号BIN16位	ANY16
(d)	脉冲密度数据存储目标的起始字软元件编号	—	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可以使用的软元件

操作数	位	字	双字		间接指定	常数			其它		
	X、Y、M、L、SM、F、B、SB	T、ST、C、D、W、SD、SW、R、ZR、RD	U□\G□、U3E□\G□	Z		LT、LST、LC	LZ	K、H		E	\$
(s1)	—	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(s3)	—	○	—	—	—	—	○	—	—	—	—

功能

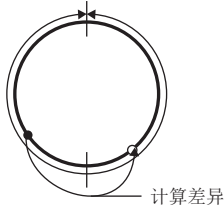
将(s1)中指定的软元件输入只按(s2)中指定的时间进行计数，将计数结果存储到(d)中指定的软元件中。



注意事项

- 关于输入X00~X0F的ON/OFF的最大频率，X00~X05：200kHz、X06~X07：10kHz。
- SPDH指令执行中更改操作数的测定时间的情况下，将在每次测定时间结束时反映更改后的测定时间。
- SPDH指令执行中，如果通过HCMOV指令改写高速计数器的当前值，或通过外部预置输入或自复位更改当前值，则无法正常测定脉冲密度。(继续动作。)
- 使用SPDH指令时，请勿在测定时间内输入超出高速计数器的环长/2的脉冲数。如果已输入，将无法计算正确的脉冲数。

环形计数器长/2



- SPDH指令的操作数中指定的测定时间将覆盖脉冲密度(旋转速度)测定功能中使用的测定单位时间的SD。
- 测定时间为1~2147483647以外的情况下，带符号四舍五入到1。
- SPDH指令只能指定设置了高速计数器参数的通道编号。在(s1)中指定通道编号或A相的X编号。指定了未设置参数的通道时，将发生运算错误H3600。
- SPDH指令执行时，UP/DOWN、预置输入、使能输入的设置以高速计数器的参数设置动作。
- 在SPDH指令的开始时、停止时，高速计数器通用功能也将与脉冲密度测定功能同时开始、停止。
- 通过HIOEN指令(K0：在高速计数器)驱动高速计数器的状态下，即使在SPDH指令中指定驱动中的通道并驱动，脉冲密度测定不会开始，也不会发生错误。

出错

错误代码(SD0)	内容
1810H	(s1)中指定了其他指令已使用的通道编号时。
3285H	(s1)中指定了对象范围外的位软元件时。 (s1)中指定了CH1~CH8以外的通道编号时。
3600H	(s1)中指定了参数设置中未选择的通道编号、字软元件时。

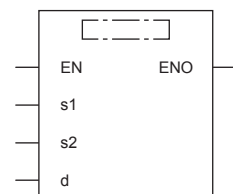
32位数据脉冲密度的测定

DSPDH

通过中断输入测定指定时间的输入脉冲。

梯形图	ST
	ENO:=DSPDH(EN, s1, s2, d);

FBD/LD



设置数据

■内容、范围、数据类型

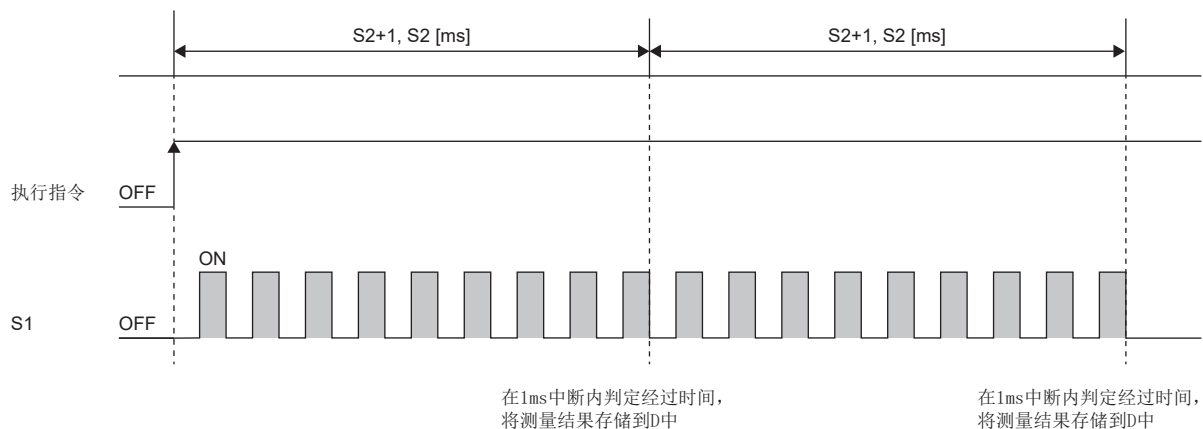
操作数	内容	范围	数据类型	数据类型(标签)
(s1)	输入脉冲的高速计数器通道编号	1~8	无符号BIN32位	ANY32
(s2)	时间(ms)数据或存储了数据的字软元件	1~2147483647	无符号BIN32位	ANY32
(d)	脉冲密度数据存储目标的起始字软元件编号	—	无符号BIN32位	ANY32
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可以使用的软元件

操作数	位	字	双字			间接指定	常数			其它	
	X、Y、M、L、SM、F、B、SB	T、ST、C、D、W、SD、SW、R、ZR、RD	U□\G□、U3E□\G□	Z	LT、LST、LC		LZ	K、H	E		\$
(s1)	—	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—
(s3)	—	○	—	—	—	—	○	—	—	—	—

功能

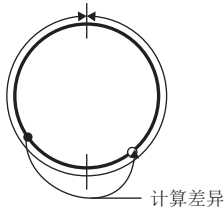
将(s1)中指定的软元件输入只按(s2)+1、(s2)中指定的时间进行计数，将计数结果存储到(d)+1、(d)中指定的软元件中。



注意事项

- 关于输入X00~X0F的ON/OFF的最大频率，X00~X05：200kHz、X06~X07：10kHz。
- SPDH指令执行中更改操作数的测定时间的情况下，将在每次测定时间结束时反映更改后的测定时间。
- SPDH指令执行中，如果通过HCMOV指令改写高速计数器的当前值，或通过外部预置输入或自复位更改当前值，则无法正常测定脉冲密度。(继续动作。)
- 使用SPDH指令时，请勿在测定时间内输入超出高速计数器的环长/2的脉冲数。如果已输入，将无法计算正确的脉冲数。

环形计数器长/2



- SPDH指令的操作数中指定的测定时间将覆盖脉冲密度(旋转速度)测定功能中使用的测定单位时间的SD。
- 测定时间为1~2147483647以外的情况下，带符号四舍五入到1。
- SPDH指令只能指定设置了高速计数器参数的通道编号。在(s1)中指定通道编号或A相的X编号。指定了未设置参数的通道时，将发生运算错误H3600。
- SPDH指令执行时，UP/DOWN、预置输入、使能输入的设置以高速计数器的参数设置动作。
- 在SPDH指令的开始时、停止时，高速计数器通用功能也将与脉冲密度测定功能同时开始、停止。
- 通过HIOEN指令(K0：在高速计数器)驱动高速计数器的状态下，即使在SPDH指令中指定驱动中的通道并驱动，脉冲密度测定不会开始，也不会发生错误。


出错

错误代码(SD0)	内容
1810H	(s1)中指定了其他指令已使用的通道编号时。
3285H	(s1)中指定了对象范围外的位软元件时。 (s1)中指定了CH1~CH8以外的通道编号时。
3600H	(s1)中指定了参数设置中未选择的通道编号、字软元件时。

第5部分 网络指令

本部分由下述章构成。

关于在MX-F型中使用智能功能模块时的专用指令的内容，请参阅下述手册。

 MELSEC iQ-F FX5编程手册(指令/通用FUN/FB篇)

29 以太网用指令

30 外部设备通信指令

29 以太网用指令

关于在MX-F型中使用智能功能模块时的专用指令的内容，请参阅下述手册。

📖 MELSEC iQ-F FX5编程手册(指令/通用FUN/FB篇)

29.1 打开/关闭处理指令

要点 🔍

执行以下指令中的连接，即使重复执行以下指令，请求也会被忽略。

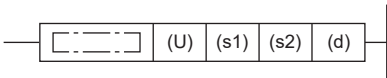
此时，在后执行的指令的执行条件成立的期间，先执行的指令完成后执行请求。此外，请求被忽略时，SM699(专用指令未执行标志)将变为ON。

- SOCOOPEN指令
- SOCCLOSE指令
- SOCSND指令
- SOCRCV指令
- ECPRTCL指令

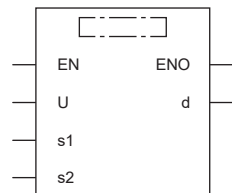
连接的建立

GP. SOCOOPEN、SP. SOCOOPEN


进行指定连接的打开处理。

梯形图	ST
	ENO:=GP_SOCOOPEN(EN, U, s1, s2, d); ENO:=SP_SOCOOPEN(EN, U, s1, s2, d);

FBD/LD



■ 执行条件

指令	执行条件
GP. SOCOOPEN SP. SOCOOPEN	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(U)	GP. SOCOOPEN	对象的起始输入输出编号 (指定起始输入输出编号的高3位)	3EOH	无符号BIN16位	ANY16
	SP. SOCOOPEN	虚拟	—	字符串	ANYSTRING_SINGLE
(s1)	连接编号	1~32	无符号BIN16位	ANY16	
(s2)	GP. SOCOOPEN[G]	存储控制数据的起始软元件	参阅控制数据	软元件	ANY16
	SP. SOCOOPEN			字	ANY16_ARRAY (要素数: 10)
(d)	指令完成时通过1个扫描变为ON的软元件 异常完成时(d)+1也将变为ON。	—	位	ANYBIT_ARRAY (要素数: 2)	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位	字				双字		间接指定	常数			其它 (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. SOCOOPEN	—	—	○	—	—	—	○	○	—	—	○
	SP. SOCOOPEN	—	—	○	—	—	—	○	—	—	○	○
(s1)	—	—	○	—	—	—	○	○	—	—	—	
(s2)	—	—	○	—	—	—	○	—	—	—	—	
(d)	○	—	○	—	—	—	○	—	—	—	—	

■控制数据

操作数: (s2)														
软元件	项目	内容	设置范围	设置方										
+0	执行/完成类型	指定连接的打开处理时, 是使用通过工程工具进行的参数设定值, 还是使用控制数据的(s2)+2~(s2)+9的设定值。 • 0000H: 以工程工具的“打开设置”中设置的内容进行打开处理。 • 8000H: 以控制数据的(s2)+2~(s2)+9中指定的内容进行打开处理。*3	0000H 8000H	用户										
+1	完成状态	指令完成时存储完成状态。 • 0000H: 正常完成 • 0000H以外: 异常完成(错误代码)	—	系统										
+2	使用用途设置区域	指定连接的使用用途。 <table border="1" style="margin-left: 40px;"> <tr> <td style="text-align: center;">b15 b14</td> <td style="text-align: center;">...</td> <td style="text-align: center;">b10 b9 b8 b7</td> <td style="text-align: center;">...</td> <td style="text-align: center;">b0</td> </tr> <tr> <td style="text-align: center;">(s2)+2 (3)</td> <td style="text-align: center;">0</td> <td style="text-align: center;">(4)(2)</td> <td style="text-align: center;">(1)</td> <td style="text-align: center;">0</td> </tr> </table> <p>(1) 通信方式(协议)(位8、7) • 00: TCP • 01: TLS • 10: UDP • 11: DTLS (2) 套接字通信功能的有序无序(位9) • 1: 无序(固定) (3) 打开方式(位15、14) • 00: Active打开或UDP • 10: Unpassive打开 • 11: Fullpassive打开 (4) 通信协议设置(位10) • 0: 不使用通信协议支持功能(使用套接字通信功能) • 1: 使用通信协议支持功能</p>	b15 b14	...	b10 b9 b8 b7	...	b0	(s2)+2 (3)	0	(4)(2)	(1)	0	—	用户
b15 b14	...	b10 b9 b8 b7	...	b0										
(s2)+2 (3)	0	(4)(2)	(1)	0										
+3	本站端口编号	指定本站的端口编号。	1~4999、5010~65534 (0001H~1387H、1392H~FFFEH)	用户										
+4	对象设备IP地址*1	指定对象设备的IP地址。	00000001H~FFFFFFFH	用户										
+5		进行广播轮询通信时指定FFFFFFFH。												
+6	对象设备端口编号*1	指定对象设备的端口编号。 全部端口编号作为对象进行接收时指定FFFFH。*2	1~65535 (0001H~FFFFH)	用户										
+7~+9	—	禁止使用	—	系统										

*1 Unpassive打开时对象设备IP地址、对象设备端口编号将被忽略。

*2 端口编号指定为65535(FFFFH)时, 无法发送数据。发送数据时应指定1~65534(1~FFFEH)。

*3 没有控制数据的设置项目如下所示。

生存确认: KeepAlive (TCP/TLS的情况下)、UDP (UDP/DTLS的情况下)

连接监视定时器的设置: 无效

要点

端口编号0001H~03FFH一般是保留的端口编号(WELL KNOWN PORT NUMBERS), F000H~FFFEH是其它功能变化使用的编号, 因此建议使用本站端口编号0400H~1387H、1392H~EFFFH。

功能

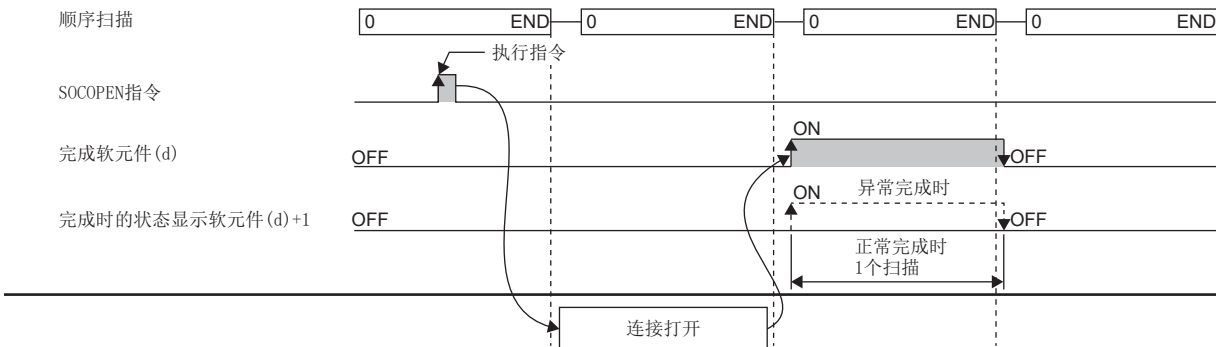
- 进行(s1)中指定的连接的打开处理。打开处理中使用的设定值是在(s2)+0中选择。
- 本指令的执行及正常/异常完成，可通过设置数据中指定的完成软元件(d)、完成时的状态显示软元件(d)+1进行确认。

• 完成软元件(d)
在本指令完成扫描的END处理中变为ON，在下一个END处理中变为OFF。

• 完成时的状态显示软元件(d)+1
根据本指令完成时的状态而进行ON/OFF。
正常完成时：保持OFF状态不变。

异常完成时：在本指令完成扫描的END处理中变为ON，在下一个END处理中变为OFF。

- 本指令的执行时机如下所示。



- 可以打开参数中未设置的连接(协议栏处于空栏状态的连接)使用。在此情况下应将(s2)+0设置为8000H，在控制数据的(s2)+1~(s2)+9中指定打开的内容。

要点

本指令支持加密通信协议(TLS和DTLS)。详细情况请确认所使用的控制器的用户手册。

注意事项

- 以以太网端口建立连接时的注意事项记载如下。连接状态的确认方法请确认所使用的控制器的用户手册。

项目	内容
在TCP/IP或者TLS协议，且为FullPassive模式下建立连接时	<p>执行本指令后，如果从对象设备连接配置设置中未设置的IP地址的对象设备接收到连接的请求，则在返回正常响应后变为切断连接的动作。从对象设备看，是先建立连接然后切断的动作。</p> <p>应确认下述指令的完成状态来判断连接建立是否成功。成功的情况下，存储0000H。失败的情况下，存储C1B1H。</p> <ul style="list-style-type: none"> • SOCSND指令 • SOCRCV指令 • SOCCSET指令 • ECPRTCL指令
在TCP/IP或者TLS协议，且为FullPassive模式/UnPassive模式下建立连接后，再次要建立连接时	<p>执行本指令后，在通过来自对象设备的连接请求建立连接的状态下，再次从对象设备接收了连接请求的情况下，不受理该请求，而切断连接。(例如，连接建立后在执行SOCLOSE指令之前对象设备可能会被复位，对象设备有可能再次请求打开连接。)</p> <p>由于控制器在可受理重新连接请求之前需要数百ms左右，因此对象设备确认断开后，应在间隔1~2秒后再请求连接。</p> <p>此外，应确认对象设备侧的连接状态是否处于打开状态，以确认对象设备中的重新连接请求是否成功。</p>

- 请勿同时执行以太网功能的插件及以太网指令。同时使用时，可能导致误动作。

出错

错误代码(SD0)	内容
2802H	以无以太网端口的设置执行了以太网功能用指令时。
3285H	(s1)中指定的连接编号超出范围时。

异常完成时，完成时的状态显示软元件(d)+1将变为ON，完成状态(s2)+1中将存储错误代码。

关于完成状态(s2)+1中存储的错误代码，请参阅下述手册。

所使用的控制器的用户手册

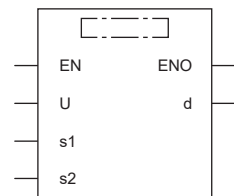
连接的断开

GP. SOCCLOSE、SP. SOCCLOSE

对指定的连接进行关闭处理。

梯形图	ST
	ENO:=GP_SOCCLOSE(EN, U, s1, s2, d); ENO:=SP_SOCCLOSE(EN, U, s1, s2, d);

FBD/LD



■执行条件

指令	执行条件
GP. SOCCLOSE SP. SOCCLOSE	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(U)	GP. SOCCLOSE	对象的起始输入输出编号 (指定起始输入输出编号的高3位)	3E0H	无符号BIN16位	ANY16
	SP. SOCCLOSE	虚拟	—	字符串	ANYSTRING_SINGLE
(s1)	连接编号	1~32	无符号BIN16位	ANY16	
(s2)	GP. SOCCLOSE	存储控制数据的起始软元件	参阅控制数据	软元件	ANY16
	SP. SOCCLOSE			字	ANY16_ARRAY (要素数: 2)
(d)	指令完成时通过1个扫描变为ON的软元件 异常完成时(d)+1也将变为ON。	—	位	ANYBIT_ARRAY (要素数: 2)	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

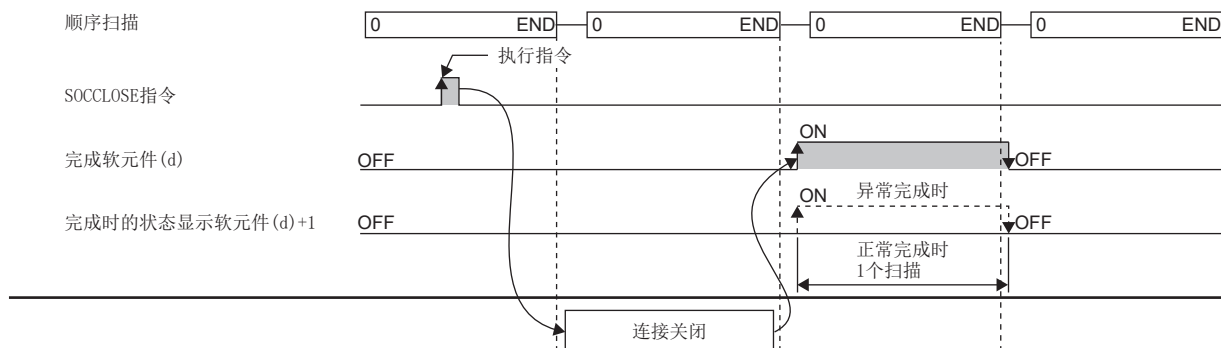
操作数	位 X、Y、M、L、 SM、F、B、 SB、FX、FY	字 J□\□	字 T、ST、C、D、 W、SD、SW、 FD、R、ZR、RD	字 U□\G□、J□\□	Z	双字		间接指定	常数				其它 (U)
						LT、 LST、LC	LZ		K、H	E	\$		
(U)	GP. SOCCLOSE	—	○	—	—	—	—	○	○	—	○	○	
	SP. SOCCLOSE	—	○	—	—	—	—	○	—	—	○	○	
(s1)	—	—	○	—	—	—	—	○	○	—	—	—	
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	○	—	○	—	—	—	—	○	—	—	—	—	

■控制数据

操作数: (s2)				
软元件	项目	内容	设置范围	设置方
+0	系统区域	—	—	—
+1	完成状态	指令完成时存储完成状态。 • 0000H: 正常完成 • 0000H以外: 异常完成(错误代码)	—	系统

功能

- 对(s1)中指定的连接进行关闭处理(连接的断开)。
- 本指令的执行及正常/异常完成, 可通过设置数据中指定的完成软元件(d)、完成时的状态显示软元件(d)+1进行确认。
 - 完成软元件(d)
 在本指令完成扫描的END处理中变为ON, 在下一个END处理中变为OFF。
 - 完成时的状态显示软元件(d)+1
 根据本指令完成时的状态而进行ON/OFF。
 正常完成时: 保持OFF状态不变。
 异常完成时: 在本指令完成扫描的END处理中变为ON, 在下一个END处理中变为OFF。
- 本指令的执行时机如下所示。



出错

错误代码(SD0)	内容
2802H	以无以太网端口的设置执行了以太网功能用指令时。
3285H	(s1)中指定的连接编号超出范围时。

异常完成时, 完成时的状态显示软元件(d)+1将变为ON, 完成状态(s2)+1中将存储错误代码。

关于完成状态(s2)+1中存储的错误代码, 请参阅下述手册。

📖 所使用的控制器的用户手册

要点

在TCP/TLS/DTLS Passive方式中, 将SOCCLOSE指令指定了打开等待状态的连接的情况下, SOCOPEN指令、SOCCLOSE指令均将正常完成, 连接将关闭。

29.2 套接字通信用指令

要点

执行以下指令中的连接，即使重复执行以下指令，请求也会被忽略。

此时，在后执行的指令的执行条件成立的期间，先执行的指令完成后执行请求。此外，请求被忽略时，SM699(专用指令未执行标志)将变为ON。

- SOCOPEIN指令
- SOCCLOSE指令
- SOCSND指令
- SOCRCV指令
- ECPRTCL指令

接收数据的END处理时读取

GP. SOCRCV、SP. SOCRCV

在执行指令后的END处理中，将指定的连接的接收数据从套接字通信接收数据区域读取。

梯形图	ST
	<pre>ENO:=GP_SOCRCV(EN, U, s1, s2, d1, d2); ENO:=SP_SOCRCV(EN, U, s1, s2, d1, d2);</pre>

FBD/LD

■执行条件

指令	执行条件
GP. SOCRCV SP. SOCRCV	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(U)	GP. SOCRCV	对象的起始输入输出编号 (指定起始输入输出编号的高3位)	3E0H	无符号BIN16位	ANY16
	SP. SOCRCV	虚拟	—	字符串	ANYSTRING_SINGLE
(s1)		连接编号	1~32	无符号BIN16位	ANY16
(s2)	GP. SOCRCV	存储控制数据的起始软元件	参阅控制数据	软元件	ANY16
	SP. SOCRCV			字	ANY16_ARRAY (要素数: 2)
(d1)	GP. SOCRCV	存储接收数据的起始软元件	—	软元件	ANY16*1
	SP. SOCRCV			字	
(d2)		指令完成时通过1个扫描变为ON的软元件 异常完成时(d2)+1也将变为ON。	—	位	ANYBIT_ARRAY (要素数: 2)
EN		执行条件	—	位	BOOL
ENO		执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位			字			双字		间接指定	常数				其它 (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. SOCRCV	—	—	○	—	—	—	—	○	○	—	○	○	
	SP. SOCRCV	—	—	○	—	—	—	—	○	—	—	○	○	
(s1)	GP. SOCRCV	○	—	○	—	—	—	—	○	○	—	—	—	
	SP. SOCRCV[S]	—	—	○	—	—	—	—	○	○	—	—	—	
(s2)		—	—	○	—	—	—	—	○	—	—	—	—	
(d1)		—	—	○	—	—	—	—	○	—	—	—	—	
(d2)		○	—	○	—	—	—	—	○	—	—	—	—	

■控制数据

操作数：(s2)				
软元件	项目	内容	设置范围	设置方
+0	系统区域	—	—	—
+1	完成状态	指令完成时存储完成状态。 • 0000H：正常完成 • 0000H以外：异常完成(错误代码)	—	系统

操作数：(d1)				
软元件	项目	内容	设置范围	设置方
+0	接收数据长	存储从套接字通信接收数据区域读取的数据的数据长。 (字节数)	0~10238	系统
+1~+□	接收数据	从套接字通信接收数据区域读取的数据，将从小编号开始依次被存储。*1	—	系统

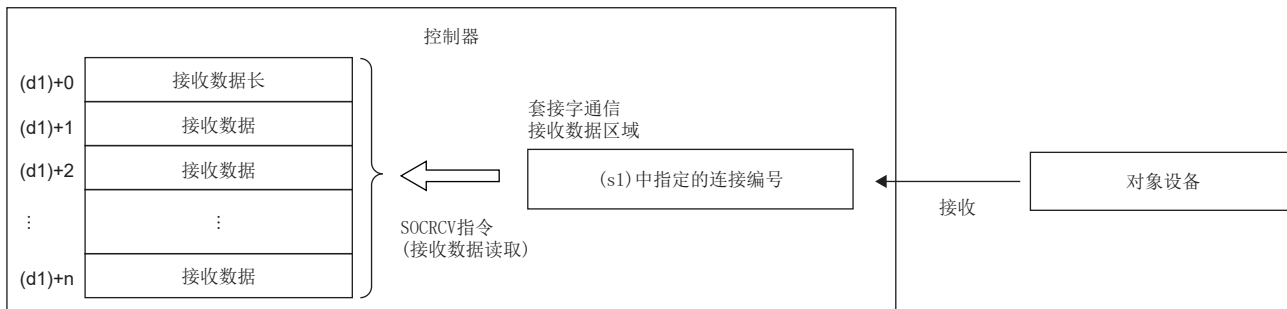
*1 接收的数据以字节单位从低位字节侧开始依次被存储。接收了奇数字节的数据时，最终数据存储区域的低位字节最后的接收数据将被存储。

要点

- 执行本指令时，从接收数据的套接字通信接收数据区域执行的读取，是通过END处理进行。因此，执行本指令时扫描时间将延长。
- 接收了奇数字节数据的情况下，存储了最后接收数据的软元件的高位字节中将放入无效的数据。

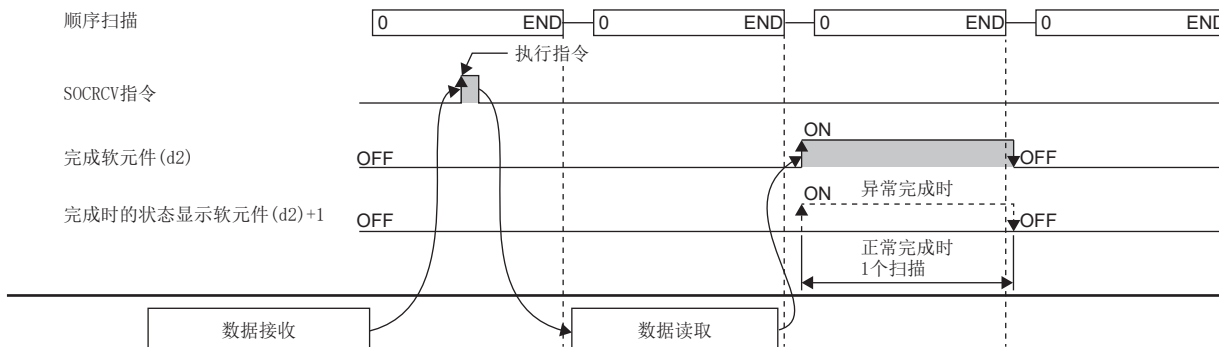
功能

- 执行本指令后的END处理中，将(s1)中指定连接的接收数据从套接字通信接收数据区域(存储从各连接对象设备接收的数据的区域)读取。



- 本指令的执行及正常/异常完成，可通过设置数据中指定的完成软元件(d2)、完成时的状态显示软元件(d2)+1进行确认。
 - 完成软元件(d2)
 在本指令完成扫描的END处理中变为ON，在下一个END处理中变为OFF。
 - 完成时的状态显示软元件(d2)+1
 根据本指令完成时的状态而进行ON/OFF。
 正常完成时：保持OFF状态不变。
 异常完成时：在本指令完成扫描的END处理中变为ON，在下一个END处理中变为OFF。

- 本指令的执行时机如下所示。



要点

本指令支持加密通信协议(TLS和DTLS)。详细情况请确认所使用的控制器的用户手册。

注意事项

对同一连接进行接收数据的读取的情况下，请勿同时使用SOCRCVS指令。

出错

错误代码(SD0)	内容
2802H	以无以太网端口的设置执行了以太网功能指令时。
3285H	(s1)中指定的连接编号超出范围时。

异常完成时，完成时的状态显示软元件(d2)+1将变为ON，完成状态(s2)+1中将存储错误代码。

关于完成状态(s2)+1中存储的错误代码，请参阅下述手册。

所使用的控制器的用户手册

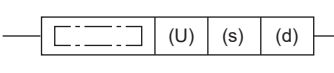
要点

- 为了防止接收过大的数据，通过SOCRCV指令设置接收数据容量，可以抑制接收的数据量。
- 通过将本指令的完成软元件使用常闭触点连接到执行指示，即使连续接收数据的情况下也可连续进行读取。

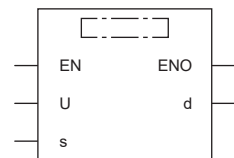
执行接收数据的指令时读取

G. SOCRCVS、S. SOCRCVS


将指定连接的接收数据从套接字通信接收数据区域读取。

梯形图	ST
	ENO:=G_SOCRCVS (EN, U, s, d); ENO:=S_SOCRCVS (EN, U, s, d);

FBD/LD



■执行条件

指令	执行条件
G. SOCRCVS S. SOCRCVS	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型 (标签)	
(U)	G. SOCRCVS	对象的起始输入输出编号 (指定起始输入输出编号的高3位)	3E0H	无符号BIN16位	ANY16
	S. SOCRCVS	虚拟	—	字符串	ANYSTRING_SINGLE
(s)	连接编号	1~32	无符号BIN16位	ANY16	
(d)	G. SOCRCVS	存储接收数据的起始软元件	—	软元件	ANY16* ¹
	S. SOCRCVS		—	字	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位 X、Y、M、L、 SM、F、B、 SB、FX、FY	J□\□	字 T、ST、C、D、 W、SD、SW、 FD、R、ZR、RD	U□\G□、J□\□	Z	双字		间接指定	常数			其它 (U)
						LT、 LST、LC	LZ		K、H	E	\$	
(U)	G. SOCRCVS	—	—	○	—	—	—	—	○	—	—	○
	S. SOCRCVS[S]	—	—	○	—	—	—	—	○	—	—	○
(s)	G. SOCRCVS	○	—	○	—	—	—	—	○	○	—	—
	S. SOCRCVS[S]	—	—	○	—	—	—	—	○	○	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

■控制数据

操作数: (d)				
软元件	项目	内容	设置范围	设置方
+0	接收数据长	存储从套接字通信接收数据区域读取的数据的数据长。 (字节数)	0~10238	系统
+1~+□	接收数据	从套接字通信接收数据区域读取的数据, 将从小编号开始依次被存储。*1	—	系统

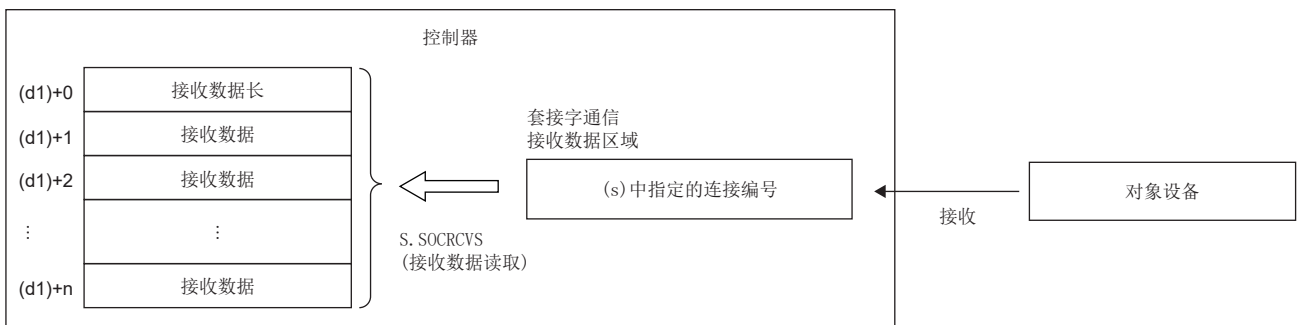
*1 接收的数据以字节单位从低位字节侧开始依次被存储。接收了奇数字节的数据时, 最终数据存储区域的低位字节最后的接收数据将被存储。

要点

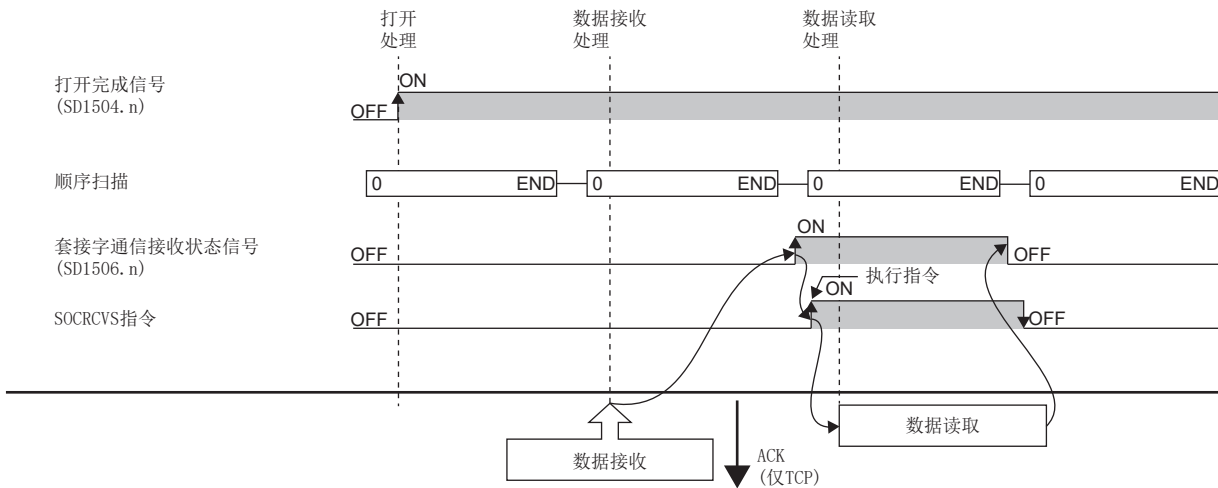
- 接收数据容量的默认值为2046字节。接收2047字节及其以上的数据的情况下, 应通过SOCRCV指令更改接收数据容量。
- 接收了奇数字节数据的情况下, 存储了最后接收数据的软元件的高位字节中将放入无效的数据。

功能

- 将(s)中指定连接的接收数据从套接字通信接收数据区域(存储从各连接的对象设备接收的数据的区域)读取。



- 使用了本指令的接收处理的时机如下所示。



要点

本指令支持加密通信协议(TLS和DTLS)。详细情况请确认所使用的控制器的用户手册。

注意事项

读取对于同一连接的接收数据时，请勿同时使用SOCRCV指令。

出错

错误代码 (SD0)	内容
2802H	以以太网端口的设置执行了以太网功能用指令时。
3285H	(s) 中指定的连接编号超出范围时。

要点

为了防止接收过大的数据，通过SOCRCV指令设置接收数据容量，可以抑制接收的数据量。

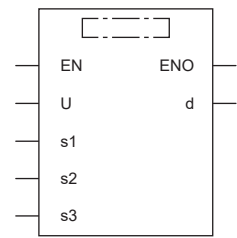
数据发送

GP. SOCSND、SP. SOCSND

将数据发送到指定连接的对象设备。

梯形图	ST
	<pre>ENO:=GP_SOCSND(EN, U, s1, s2, s3, d); ENO:=SP_SOCSND(EN, U, s1, s2, s3, d);</pre>

FBD/LD



■ 执行条件

指令	执行条件
GP. SOCSND SP. SOCSND	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(U)	GP. SOCSND	对象的起始输入输出编号 (指定起始输入输出编号的高3位)	3E0H	无符号BIN16位	ANY16
	SP. SOCSND	虚拟	—	字符串	ANYSTRING_SINGLE
(s1)	连接编号	1~32	无符号BIN16位	ANY16	
(s2)	GP. SOCSND	存储控制数据的起始软元件	参阅控制数据	软元件	ANY16
	SP. SOCSND			字	ANY16_ARRAY (要素数: 2)
(s3)	GP. SOCSND	存储发送数据的起始软元件	—	软元件	ANY16* ¹
	SP. SOCSND			字	
(d)	指令完成时通过1个扫描变为ON的软元件 异常完成时(d)+1也将变为ON。	—	位	ANYBIT_ARRAY (要素数: 2)	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

*1 通过标签进行设置的情况下,应在确保动作所需区域的前提下定义数组,指定该数组型标签的要素。

■可用软元件

操作数	位	字				双字		间接指定	常数				其它 (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[S]	—	—	○	—	—	—	—	○	—	—	○	○
(s1)	GP. SOCSND	○	—	○	—	—	—	—	○	○	—	—	—
	SP. SOCSND[S]	—	—	○	—	—	—	—	○	○	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—	
(s3)	—	—	○	—	—	—	—	○	—	—	—	—	
(d)	○	—	○	—	—	—	—	○	—	—	—	—	

■控制数据

操作数: (s2)				
软元件	项目	内容	设置范围	设置方
+0	系统区域	—	—	—
+1	完成状态	指令完成时存储完成状态。 • 0000H: 正常完成 • 0000H以外: 异常完成(错误代码)	—	系统

操作数: (s3)				
软元件	项目	内容	设置范围	设置方
+0	发送数据长	指定发送数据长。(字节数)	1~10238	用户
+1~+□	发送数据	指定发送数据。*1	—	用户

*1 发送数据以字节单位按照从低位字节侧开始的顺序被发送。发送了奇数字节的数据时,应将最后的发送数据存储至最终数据存储区域的低位字节。

要点

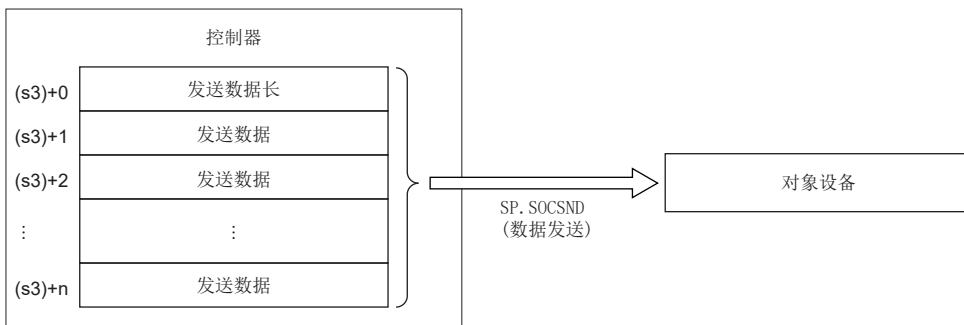
TCP的情况下,应将发送数据长设置为对象设备的最大窗口容量(接收缓冲)或其以下。

TLS的情况下,应将加密前的发送数据长设置为对象设备的最大窗口容量(接收缓冲)或其以下。

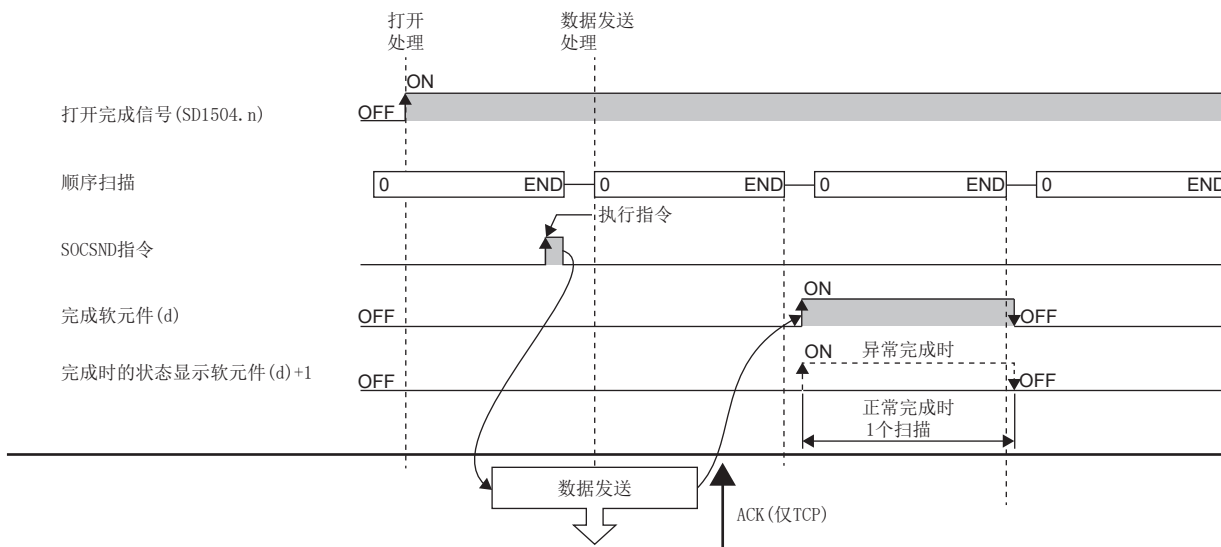
不能对超出对象设备的最大窗口容量的数据进行发送。

功能

- 将(s3)中设置的数据发送到(s1)中指定连接的对象设备。



- 本指令的执行及正常/异常完成，可通过设置数据中指定的完成软元件(d)、完成时的状态显示软元件(d)+1进行确认。
 - 完成软元件(d)
 在本指令完成扫描的END处理中变为ON，在下一个END处理中变为OFF。
 - 完成时的状态显示软元件(d)+1
 根据本指令完成时的状态而进行ON/OFF。
 正常完成时：保持OFF状态不变。
 异常完成时：在本指令完成扫描的END处理中变为ON，在下一个END处理中变为OFF。
- 使用了本指令的发送处理的时机如下所示。



完成软元件ON后，发送也可能继续进行。需要确认发送完成的情况下，应在接收侧进行完成确认等。

注意事项


执行本指令后，应在本站与对象设备双方创建到达确认手段，确认发送数据到达对象设备，数据发送正常完成。控制器通过TCP/IP或TLS协议与对象设备建立连接后，执行本指令并向对象设备发送数据的情况下，在不确认是否到达对象设备下完成本指令的执行，完成状态((S2)+1)将变为正常完成。
例如，通过通知对象设备已正常进行数据发送的程序，将到达通知对象设备。

出错

错误代码(SD0)	内容
2802H	以无以太网端口的设置执行了以太网功能指令时。
3285H	(s1)中指定的连接编号超出范围时。
3293H	(s3)中指定的发送数据长超出范围时。

异常完成时，完成时的状态显示软元件(d)+1将变为ON，完成状态(s2)+1中将存储错误代码。

关于完成状态(s2)+1中存储的错误代码，请参阅下述手册。

所使用的控制器的用户手册

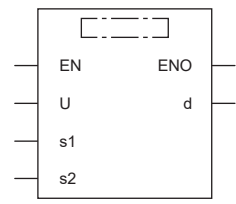
连接信息的读取

GP. SOCCINF、SP. SOCCINF

读取(s1)中指定连接的连接信息，存储到(d)中指定的软元件及其以后。

梯形图	ST
	<pre>ENO:=GP_SOCCINF(EN,U,s1,s2,d); ENO:=SP_SOCCINF(EN,U,s1,s2,d);</pre>

FBD/LD



■执行条件

指令	执行条件
GP. SOCCINF SP. SOCCINF	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(U)	GP. SOCCINF	对象的起始输入输出编号 (指定起始输入输出编号的高3位)	3E0H	无符号BIN16位	ANY16
	SP. SOCCINF	虚拟	—	字符串	ANYSTRING_SINGLE
(s1)	连接编号	1~32	无符号BIN16位	ANY16	
(s2)	GP. SOCCINF	存储控制数据的起始软元件	参阅控制数据	软元件	ANY16
	SP. SOCCINF			字	ANY16_ARRAY (要素数: 2)
(d)	存储连接信息的起始软元件	—	字	ANY16_ARRAY (要素数: 5)	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位	字		双字		间接指定	常数			其它(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. SOCCINF	—	—	○	—	—	—	—	○	—	—	○
	SP. SOCCINF	—	—	○	—	—	—	—	○	—	—	○
(s1)	GP. SOCCINF	○	—	○	—	—	—	○	○	—	—	—
	SP. SOCCINF	—	—	○	—	—	—	○	○	—	—	—
(s2)	—	—	○	—	—	—	—	○	—	—	—	—
(d)	—	—	○	—	—	—	—	○	—	—	—	—

■控制数据

操作数：(s2)																						
软元件	项目	内容	设置范围	设置方																		
+0	系统区域	—	—	—																		
+1	完成状态	指令完成时存储完成状态。 <ul style="list-style-type: none"> • 0000H：正常完成 • 0000H以外：异常完成(错误代码) 	—	系统																		
操作数：(d)																						
软元件	项目	内容	设置范围	设置方																		
+0 +1	对象设备IP地址	存储对象设备的IP地址。	00000001H~ FFFFFFFH 00000000H：无通信对象 (FFFFFFFH：广播轮询通信)	系统																		
+2	对象设备端口编号	存储对象设备的端口编号。	0001H~ FFFFH(FFFFH：广播轮询通信)	系统																		
+3	本站端口编号	存储本站端口编号。	0001H~1387H 1392H~FFFEH	系统																		
+4	连接使用用途	存储连接的使用用途。 <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td style="text-align: right;">b15</td> <td style="text-align: right;">b14</td> <td style="text-align: center;">…</td> <td style="text-align: right;">b10</td> <td style="text-align: right;">b9</td> <td style="text-align: right;">b8</td> <td style="text-align: right;">b7</td> <td style="text-align: center;">…</td> <td style="text-align: right;">b0</td> </tr> <tr> <td style="text-align: center;">(3)</td> <td style="text-align: center;">0</td> <td></td> <td style="text-align: center;">(4)</td> <td style="text-align: center;">(2)</td> <td style="text-align: center;">(1)</td> <td></td> <td style="text-align: center;">0</td> <td></td> </tr> </table> </div> <p>(1) 通信方式(协议)(位8、7)</p> <ul style="list-style-type: none"> • 00: TCP • 01: TLS • 10: UDP • 11: DTLS <p>(2) 套接字通信功能的有序无序(位9)</p> <ul style="list-style-type: none"> • 1: 无序(固定) <p>(3) 打开方式(位15、14)</p> <ul style="list-style-type: none"> • 00: Active打开或UDP • 10: Unpassive打开 • 11: Fullpassive打开 <p>(4) 通信协议设置(位10)</p> <ul style="list-style-type: none"> • 0: 不使用通信协议支持功能。(使用套接字通信功能。) • 1: 使用通信协议支持功能。 	b15	b14	…	b10	b9	b8	b7	…	b0	(3)	0		(4)	(2)	(1)		0		—	系统
b15	b14	…	b10	b9	b8	b7	…	b0														
(3)	0		(4)	(2)	(1)		0															

功能

读取 (s1) 中指定连接的连接信息。

要点

本指令支持加密通信协议 (TLS和DTLS)。详细情况请确认所使用的控制器的用户手册。

在TCP/TLS/DTLS的Passive方式中，如果在连接为打开状态且与对象设备未建立连接的状态下执行本指令，将按以下方式存储对象设备的信息以外的数据。此外，连接为非打开状态时将存储0H。


软元件	项目	存储的数据	
		连接为打开状态且与对象设备未建立连接的状态	连接为非打开状态
d+0 d+1	对象设备IP地址	00000000H	00000000H
d+2	对象设备端口编号	0000H	0000H
d+3	本站端口编号	打开时的数据	0000H
d+4	连接使用用途	打开时的数据	0000H

出错

错误代码 (SD0)	内容
2802H	以无以太网端口的设置执行了以太网功能指令时。
3285H	(s1) 中指定的连接编号超出范围时。

异常完成时，完成状态 (s2) +1 中将存储错误代码。

关于完成状态 (s2) +1 中存储的错误代码，请参阅下述手册。

 所使用的控制器的用户手册

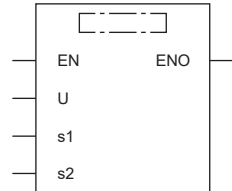
连接的通信目标更改(UDP/IP)

GP. SOCCSET、SP. SOCCSET

对指定连接的通信对象IP地址、通信对象端口编号进行更改。

梯形图	ST
	ENO:=GP_SOCCSET(EN, U, s1, s2); ENO:=SP_SOCCSET(EN, U, s1, s2);

FBD/LD



■执行条件

指令	执行条件
GP. SOCCSET SP. SOCCSET	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(U)	GP. SOCCSET	对象的起始输入输出编号 (指定起始输入输出编号的高3位)	3E0H	无符号BIN16位	ANY16
	SP. SOCCSET	虚拟	—	字符串	ANYSTRING_SINGLE
(s1)	连接编号	1~32	无符号BIN16位	ANY16	
(s2)	GP. SOCCSET	存储控制数据的起始软元件	参阅控制数据	软元件	ANY16
	SP. SOCCSET			字	ANY16_ARRAY (要素数: 5)
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位	字		双字		间接指定	常数			其它 (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)	—	—	—	○	—	—	—	—	○	—	—	—	—

■控制数据

操作数: (s2)				
软元件	项目	内容	设置范围	设置方
+0	系统区域	—	—	—
+1	完成状态	指令完成时存储完成状态。 • 0000H: 正常完成 • 0000H以外: 异常完成(错误代码)	—	系统
+2 +3	对象设备IP地址	存储对象设备的IP地址。	00000001H~ FFFFFFFFH (FFFFFFFFH: 广播 轮询通信)	用户
+4	对象设备端口编号	存储对象设备的端口编号。	0001H~FFFFH (FFFFH: 广播 轮询通信)	用户

功能

UDP通信时, 对(s1)中指定连接的通信对象IP地址、通信对象端口编号进行更改。

要点

- 使用本指令时, 即使未关闭连接也可更改通信对象。
- 本指令的执行时机为接收数据区域中有数据时, 执行本指令后, 执行SOCRCV指令/SOCRCVS指令之后将生效。此外, 接收数据区域中没有数据时, 执行本指令后将生效。

注意事项

SOCSND执行中请勿通过本指令更改通信对象。

出错

错误代码(SD0)	内容
2802H	以无以太网端口的设置执行了以太网功能用指令时。
3285H	(s1)中指定的连接编号超出范围时。

异常完成时, 完成状态(s2)+1中将存储错误代码。

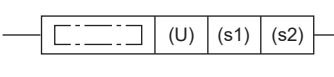
关于完成状态(s2)+1中存储的错误代码, 请参阅下述手册。

📖所使用的控制器的用户手册

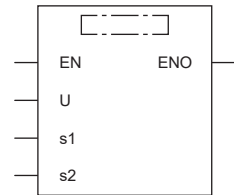
连接的接收模式更改

GP. SOCRMODE、SP. SOCRMODE


对指定的连接更改TCP/TLS接收模式(对UDP/DTLS通信的连接无效)及接收数据容量。

梯形图	ST
	<pre>ENO:=GP_SOCRMODE(EN, U, s1, s2); ENO:=SP_SOCRMODE(EN, U, s1, s2);</pre>

FBD/LD



■执行条件

指令	执行条件
GP. SOCRMODE SP. SOCRMODE	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(U)	GP. SOCRMODE	对象的起始输入输出编号 (指定起始输入输出编号的高3位)	3E0H	无符号BIN16位	ANY16
	SP. SOCRMODE	虚拟	—	字符串	ANYSTRING_SINGLE
(s1)	连接编号	1~32	无符号BIN16位	ANY16	
(s2)	GP. SOCRMODE	存储控制数据的起始软元件	参阅控制数据	软元件	ANY16
	SP. SOCRMODE			字	ANY16_ARRAY (要素数: 4)
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件

操作数	位	字	双字		间接指定	常数				其它 (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)	—	—	—	○	—	—	—	—	○	—	—	—	—

■控制数据

操作数: (s2)				
软元件	项目	内容	设置范围	设置方
+0	系统区域	—	—	—
+1	完成状态	指令完成时存储完成状态。 • 0000H: 正常完成 • 0000H以外: 异常完成(错误代码)	—	系统
+2	TCP /TLS接收模式*1	存储TCP/TLS接收模式。 • 0: TCP/TLS通常接收模式 • 1: TCP/TLS固定长接收模式	0、1	用户
+3	接收数据容量	存储套接字通信接收数据容量。(字节数)	1~10238	用户

*1 对于UDP/DTLS通信的连接无效。

功能

- 对于(s1)中指定的连接，更改UDP/DTLS通信的连接以外的TCP/TLS接收模式及接收数据容量。
- 对于TCP/TLS的连接将变为(s2)+2中指定的模式。

要点

本指令支持加密通信协议(TLS)。详细情况请确认所使用的控制器的用户手册。

■TCP/TLS通常接收模式

接收数据时，将数据存储到套接字通信接收数据区域中，将SD1506、SD2132~SD2135(套接字通信接收状态信号)置为ON。

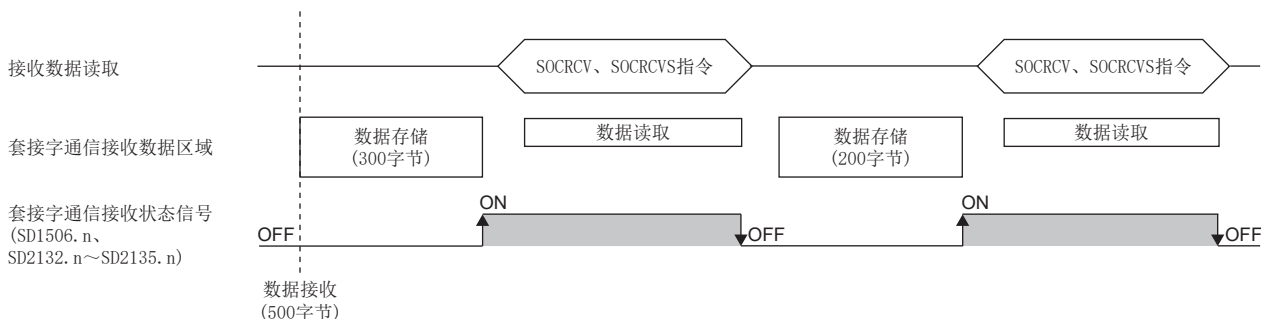
接收的数据超过了指定的接收数据容量的情况下，超出部分的接收数据将变为下一个接收数据。

此后，通过SOCRCV指令/SOCRCVS指令从套接字通信接收数据区域读取数据前接收了数据的情况下，将被存储到OS内接收数据中。

此外，通过SOCRCV指令/SOCRCVS指令从套接字通信接收数据区域获取数据时，如果OS内接收数据中存有数据，则将被存储到套接字通信接收数据区域中，将SD1506、SD2132~SD2135(套接字通信接收状态信号)置为ON。

例

将接收数据容量设置为300字节，但接收了500字节的情况下



■TCP/TLS固定长接收模式

接收数据时，将数据存储到套接字通信接收数据区域中。未达到指定的接收数据容量的情况下，SD1506、SD2132~SD2135(套接字通信接收状态信号)不变为ON。

在接收数据达到接收数据容量之前重复接收，在达到接收数据容量的时刻SD1506、SD2132~SD2135(套接字通信接收状态信号)将变为ON。

接收的数据超过了指定的接收数据容量的情况下，超过部分的接收数据将变为下一个接收数据。

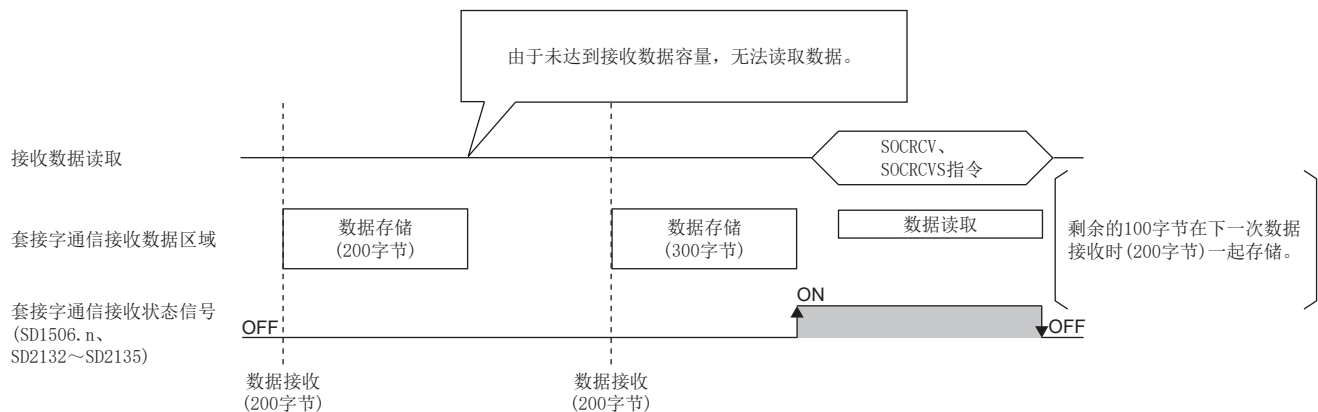
此后，通过SOCRCV指令/SOCRCVS指令从套接字通信接收数据区域读取数据前接收了数据的情况下，将被存储到OS内接收数据中。

通过SOCRCV指令/SOCRCVS指令从套接字通信接收数据区域获取数据时，如果OS内接收数据中存有数据，则将被存储到套接字通信接收数据区域中，但未达到指定的接收数据容量的情况下，SD1506、SD2132~SD2135(套接字通信接收状态信号)不变为ON。

之后，在接收数据达到接收数据容量之前重复接收，在达到接收数据容量的时刻SD1506、SD2132~SD2135(套接字通信接收状态信号)将变为ON。

例

将接收数据容量设置为300字节，但连续接收了200字节的情况下



要点

• 软件的有效利用

SOCRCV指令/SOCRCVS指令中使用的接收数据的存储用软件，虽然最多需要预留5119字(10238字节)，但通过抑制接收数据容量可以有效利用软件。

• 接收数据分离的防止

根据线路种类等，来自于对象设备的数据被分离到达的情况下，通过在TCP/TLS固定长接收模式中指定接收数据容量，可以防止数据分离。

• 接收数据合并的防止

由于顺控程序接收处理的延迟等，根据对象设备，分开发送的数据有可能被合并。


通过在TCP/TLS固定长接收模式中指定接收数据容量，可以正确分开接收。

- 在(s2)+3中，指定1次SOCRCV指令/SOCRCVS指令中读取的接收数据容量。UDP/DTLS的情况下，接收的数据超过了指定的接收数据容量的情况下，超过部分的接收数据将变为下一个接收数据。

出错

错误代码 (SD0)	内容
2802H	以无以太网端口的设置执行了以太网功能指令时。
3285H	(s1)中指定的连接编号超出范围时。

异常完成时，完成状态 (s2)+1 中将存储错误代码。
 关于完成状态 (s2)+1 中存储的错误代码，请参阅下述手册。

 所使用的控制器的用户手册

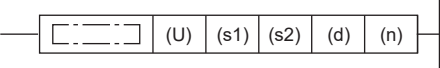
要点

在TCP/TLS固定长度接收模式下，接收状态信号不变为ON时，可以通过SOCRDATA指令读取当前接收的数据，并确认来自对象设备的发送数据是否不足。

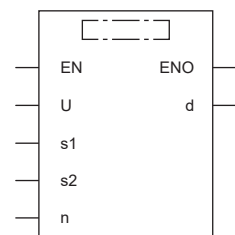
套接字通信接收数据读取

G(P).SOCRDATA、S(P).SOCRDATA


从(s1)中指定连接的套接字通信接收数据区域读取(n)中指定字数的数据，存储到(d)中指定的软元件及其以后。

梯形图	ST
	<pre> 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); </pre>

FBD/LD



■执行条件

指令	执行条件
G. SOCRDATA S. SOCRDATA	
GP. SOCRDATA SP. SOCRDATA	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(U)	G(P).SOCRDATA	对象的起始输入输出编号 (指定起始输入输出编号的高3位)	3E0H	无符号BIN16位	ANY16
	S(P).SOCRDATA	虚拟	—	字符串	ANYSTRING_SINGLE
(s1)	连接编号	1~32	无符号BIN16位	ANY16	
(s2)	G(P).SOCRDATA	存储控制数据的起始软元件 参阅控制数据	软元件	ANY16	
	S(P).SOCRDATA		字	ANY16_ARRAY (要素数: 2)	
(d)	存储读取的数据的起始软元件	—	字	ANY16*1	
(n)	读取的数据数	1~5120	无符号BIN16位	ANY16	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

■可用软元件

操作数	位		字			双字		间接指定	常数			其它 (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)		○	—	○	—	—	—	○	○	—	—	—

■控制数据

操作数：(s2)				
软元件	项目	内容	设置范围	设置方
+0	系统区域	—	—	—
+1	完成状态	指令完成时存储完成状态。 • 0000H：正常完成 • 0000H以外：异常完成(错误代码)	—	系统

功能

从(s1)中指定连接的套接字通信接收数据区域，读取(n)中指定字数的数据，存储到(d)中指定的软元件及其以后。读取数据(n)为0时将无处理。

要点

- 本指令支持加密通信协议(TLS和DTLS)。详细情况请确认所使用的控制器的用户手册。
- 如果将读取数据数设为1字，则可读取接收数据长。由此，可以更改执行SOCRCV指令/SOCRCVS指令时存储接收数据的软元件。
- 通过本指令对本次接收的数据进行检查，通过SOCRMODE指令指定下一次接收的数据容量后，可以通过SOCRCV指令/SOCRCVS指令读取本次的数据。由此，可以以本次接收的数据为基础，指定下一次接收的数据容量。

出错

错误代码(SD0)	内容
2802H	以无以太网端口的设置执行了以太网功能用指令时。
3285H	(s1)中指定的连接编号超出范围时。
3293H	(n)中指定的读取数据数超出范围的最大值时。

异常完成时，完成状态(s2)+1中将存储错误代码。

关于完成状态(s2)+1中存储的错误代码，请参阅下述手册。

📖所使用的控制器的用户手册

29.3 通信协议支持功能指令

要点

执行以下指令中的连接，即使重复执行以下指令，请求也会被忽略。

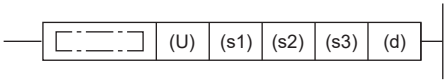
此时，在后执行的指令的执行条件成立的期间，先执行的指令完成后执行请求。此外，请求被忽略时，SM699(专用指令未执行标志)将变为ON。

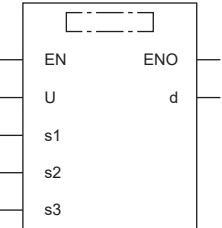
- SOCOPEIN指令
- SOCCLOSE指令
- SOCSND指令
- SOCRCV指令
- ECPRTCL指令

登录协议执行

GP. ECPRTCL、SP. ECPRTCL

执行工程工具中登录的协议。

梯形图	ST
	<pre>ENO:=GP_ECPRTCL(EN, U, s1, s2, s3, d); ENO:=SP_ECPRTCL(EN, U, s1, s2, s3, d);</pre>

FBD/LD


■执行条件

指令	执行条件
GP. ECPRTCL SP. ECPRTCL	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(U)	GP.ECPRTCL	对象的起始输入输出编号 (指定起始输入输出编号的高3位)	3E0H	无符号BIN16位	ANY16
	SP.ECPRTCL	虚拟	—	字符串	ANYSTRING_SINGLE
(s1)	连接编号	1~32	无符号BIN16位	ANY16	
(s2)	连续执行的协议数	1~8	无符号BIN16位	ANY16	
(s3)	GP.ECPRTCL	存储控制数据的起始软元件	参阅控制数据	软元件	ANY16_ARRAY (要素数: 18)
	SP.ECPRTCL			字	ANY16_ARRAY (要素数: 2)
(d)	指令完成时通过1个扫描变为ON的软元件 异常完成时(d)+1也将变为ON。	—	位	ANYBIT_ARRAY (要素数: 2)	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

■可用软元件


操作数	位	字			双字		间接指定	常数			其它 (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	○	—	○*1	—	—	○*1	—	—	—	—	
(d)	○*1	—	○	—	—	—	○*1	—	—	—	—	

*1 不能使用局部软元件及各程序中设置的文件寄存器。

■控制数据

操作数：(s3)				
软元件	项目	内容	设置范围	设置方
+0	执行数结果	存储通过本指令执行的协议数。发生错误的协议也包含在执行数中。设置数据、控制数据的设置内容有错误的情况下将存储“0”。	0、1~8	系统
+1	完成状态	指令完成时存储完成状态。 执行多个协议的情况下，将存储最后执行的协议的执行结果。 • 0：正常完成 • 0以外：异常完成(错误代码)	—	系统
+2	执行协议编号指定1	指定第1个执行的协议的协议编号。	1~128	用户
+3	执行协议编号指定2	指定第2个执行的协议的协议编号。	0、1~128	用户
+4	执行协议编号指定3	指定第3个执行的协议的协议编号。	0、1~128	用户
+5	执行协议编号指定4	指定第4个执行的协议的协议编号。	0、1~128	用户
+6	执行协议编号指定5	指定第5个执行的协议的协议编号。	0、1~128	用户
+7	执行协议编号指定6	指定第6个执行的协议的协议编号。	0、1~128	用户
+8	执行协议编号指定7	指定第7个执行的协议的协议编号。	0、1~128	用户
+9	执行协议编号指定8	指定第8个执行的协议的协议编号。	0、1~128	用户
+10	校验一致 接收数据包编号1	第1个执行的协议的通信类型中包含有接收的情况下，存储校验一致的接收数据包编号。通信类型为“仅发送”的情况下，将存储“0”。执行第1个协议时发生了错误的情况下将存储“0”。	0、1~16	系统
+11	校验一致 接收数据包编号2	第2个执行的协议的通信类型中包含有接收的情况下，存储校验一致的接收数据包编号。通信类型为“仅发送”的情况下，将存储“0”。执行第2个协议时发生了错误的情况下将存储“0”。执行的协议数不足2时将存储“0”。	0、1~16	系统
+12	校验一致 接收数据包编号3	第3个执行的协议的通信类型中包含有接收的情况下，存储校验一致的接收数据包编号。通信类型为“仅发送”的情况下，将存储“0”。执行第3个协议时发生了错误的情况下将存储“0”。执行的协议数不足3时将存储“0”。	0、1~16	系统
+13	校验一致 接收数据包编号4	第4个执行的协议的通信类型中包含有接收的情况下，存储校验一致的接收数据包编号。通信类型为“仅发送”的情况下，将存储“0”。执行第4个协议时发生了错误的情况下将存储“0”。执行的协议数不足4时将存储“0”。	0、1~16	系统
+14	校验一致 接收数据包编号5	第5个执行的协议的通信类型中包含有接收的情况下，存储校验一致的接收数据包编号。通信类型为“仅发送”的情况下，将存储“0”。执行第5个协议时发生了错误的情况下将存储“0”。执行的协议数不足5时将存储“0”。	0、1~16	系统
+15	校验一致 接收数据包编号6	第6个执行的协议的通信类型中包含有接收的情况下，存储校验一致的接收数据包编号。通信类型为“仅发送”的情况下，将存储“0”。执行第6个协议时发生了错误的情况下将存储“0”。执行的协议数不足6时将存储“0”。	0、1~16	系统
+16	校验一致 接收数据包编号7	第7个执行的协议的通信类型中包含有接收的情况下，存储校验一致的接收数据包编号。通信类型为“仅发送”的情况下，将存储“0”。执行第7个协议时发生了错误的情况下将存储“0”。执行的协议数不足7时将存储“0”。	0、1~16	系统
+17	校验一致 接收数据包编号8	第8个执行的协议的通信类型中包含有接收的情况下，存储校验一致的接收数据包编号。通信类型为“仅发送”的情况下，将存储“0”。执行第8个协议时发生了错误的情况下将存储“0”。执行的协议数不足8时将存储“0”。	0、1~16	系统

功能

- 执行工程工具中登录的协议。使用(s1)中指定的连接后，执行的协议取决于(s3)中指定的软元件及其以后的控制数据。
- 1次的指令执行中，连续执行(s2)中指定的数(最大8)的协议。
- 执行了协议的数将被存储到执行数结果(s3)+0中。
- 协议的执行状态，可通过通信协议支持功能执行状态确认用区域(Un\G350~Un\G669)进行确认。(《所使用的控制器的用户手册)

- 本指令的执行及正常/异常完成，可通过设置数据中指定的完成软元件(d)、完成时的状态显示软元件(d)+1进行确认。
- 完成软元件(d)

在本指令完成扫描的END处理中变为ON，在下一个END处理中变为OFF。

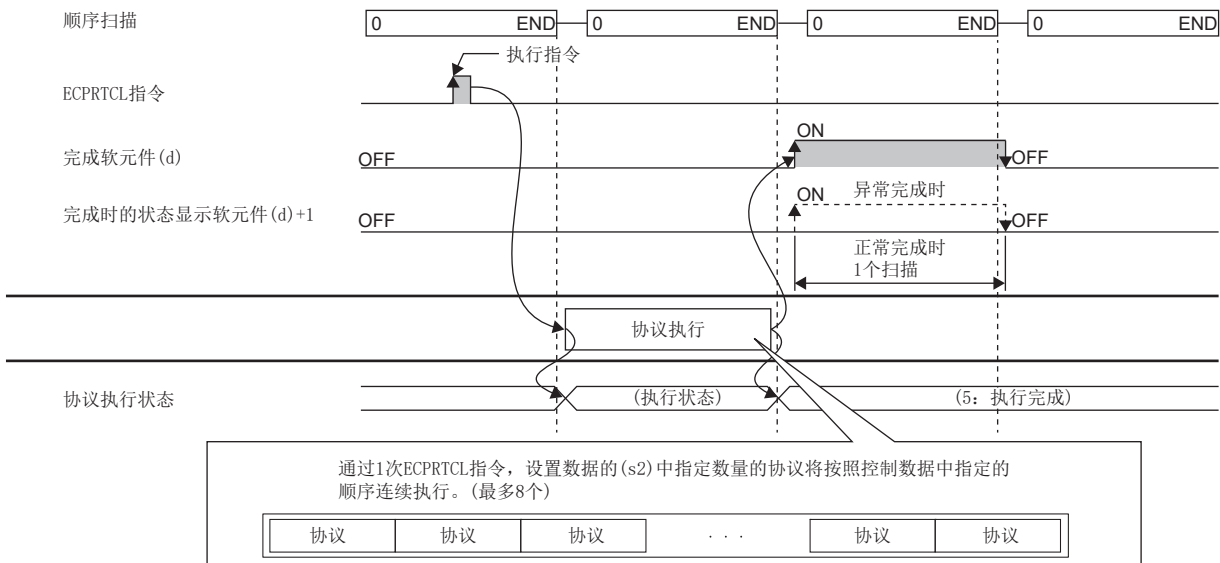
- 完成时的状态显示软元件(d)+1


根据本指令完成时的状态而进行ON/OFF。

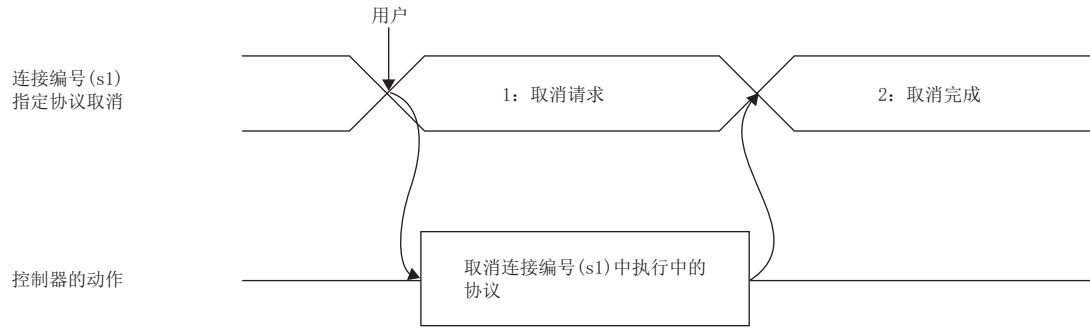
正常完成时：保持OFF状态不变。

异常完成时：在本指令完成扫描的END处理中变为ON，在下一个END处理中变为OFF。异常完成时，(s3)+1的完成状态中将存储错误代码。

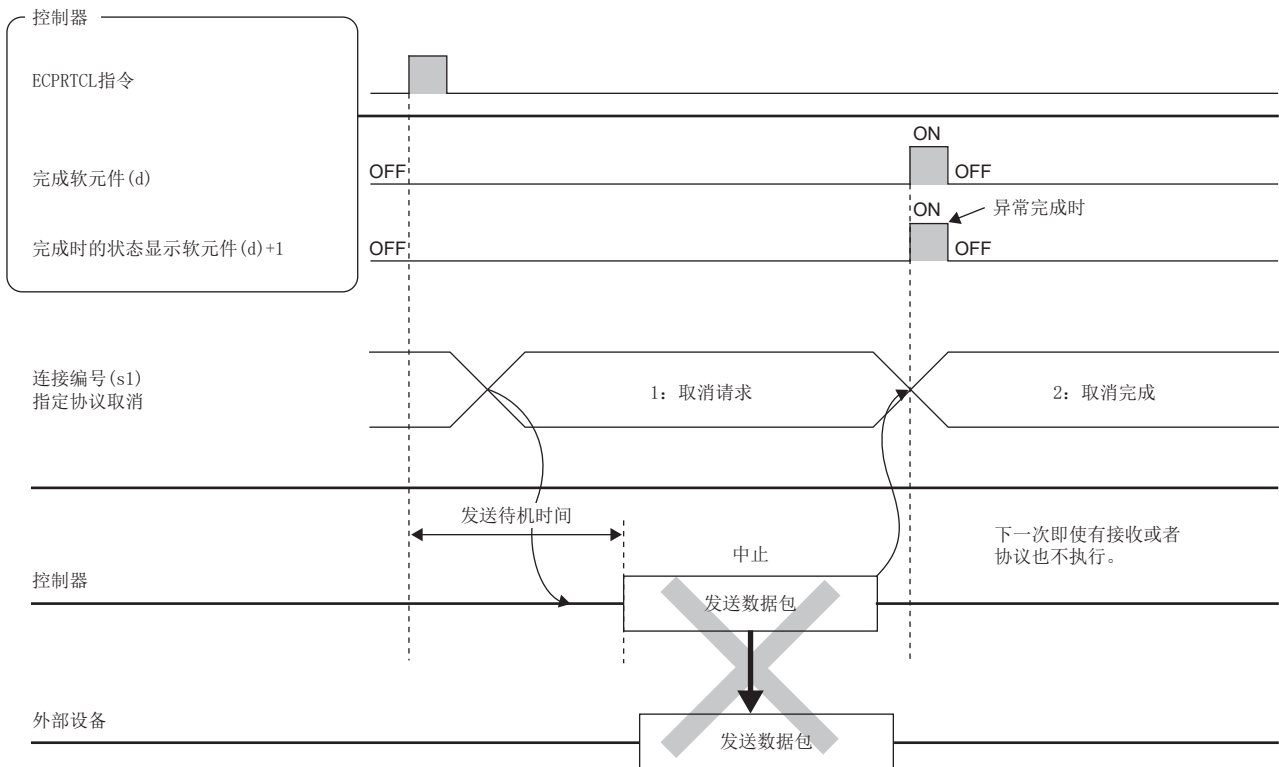
- 本指令的执行时机如下所示。



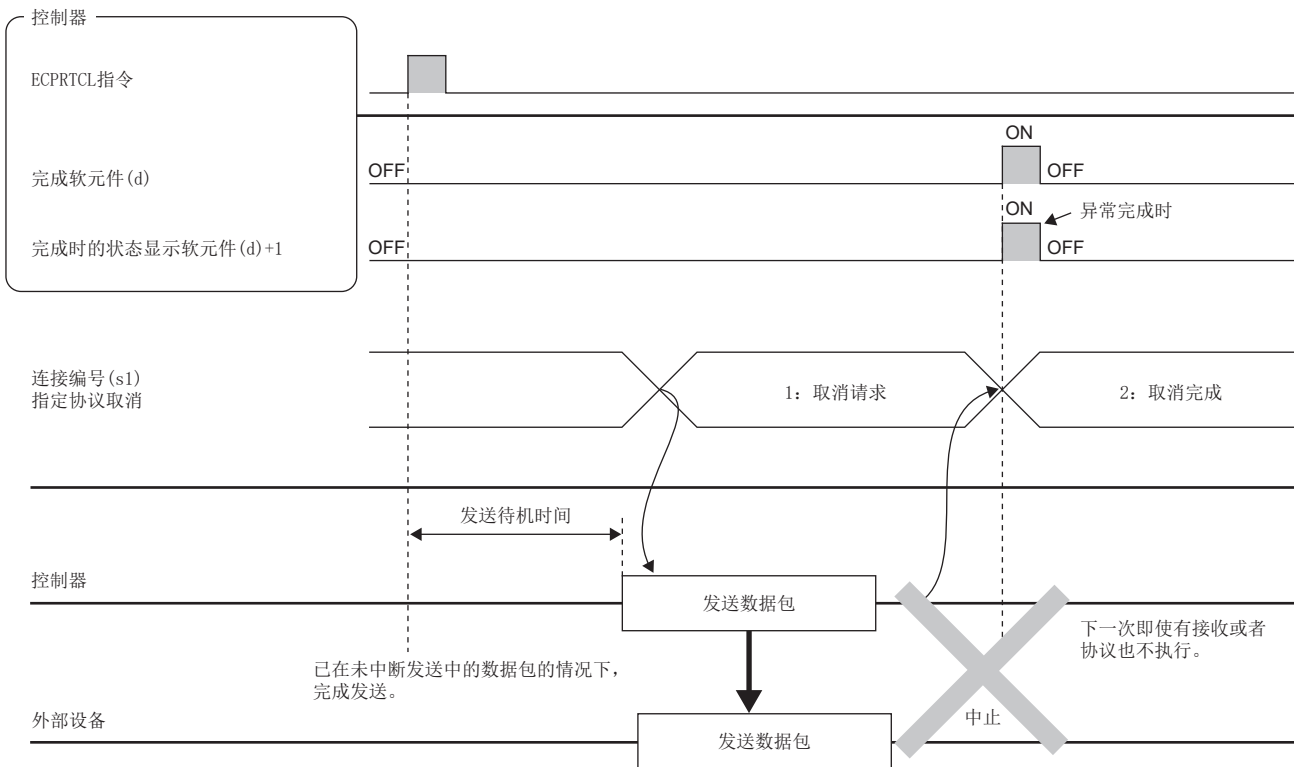
- 通过设置协议取消请求，可以取消协议的执行。协议取消指定时，使用通信协议支持功能执行状态确认用区域(Un\G350~Un\G669)。(《所使用的控制器的用户手册》)



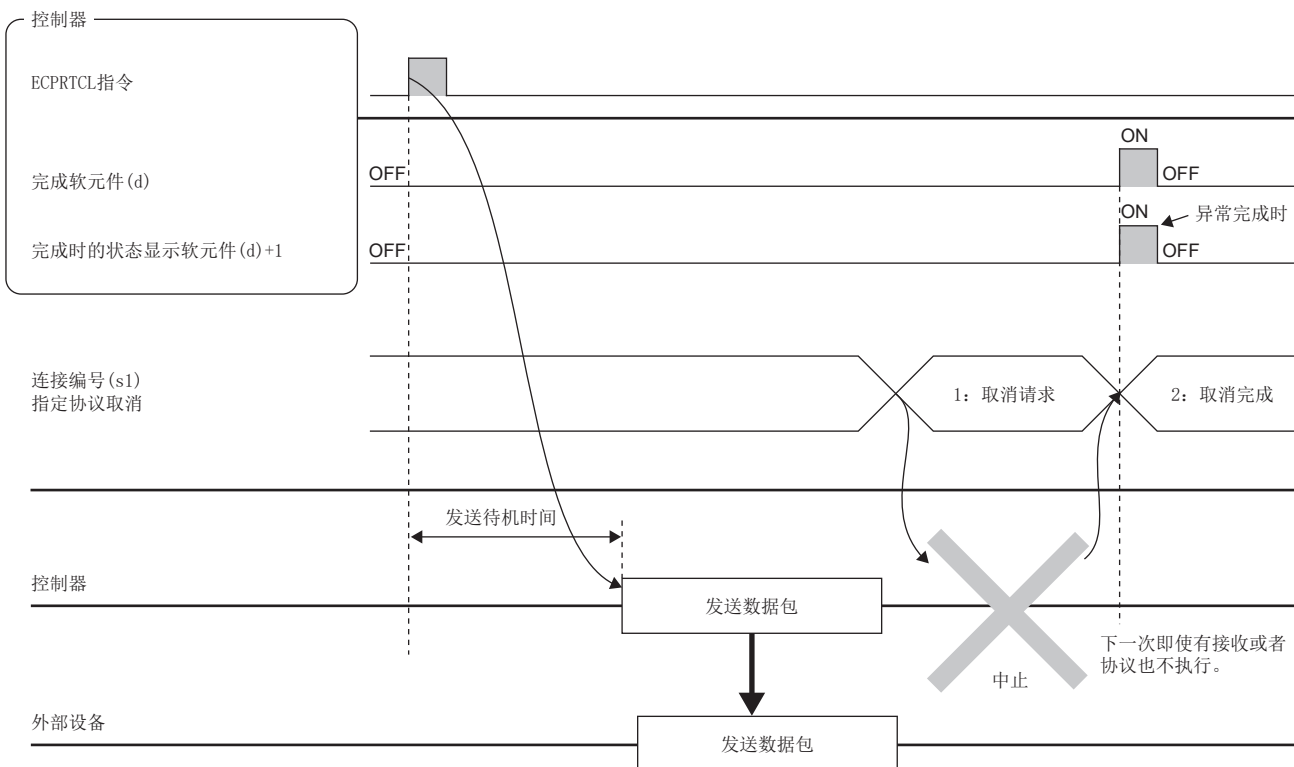
- 各时机中协议取消的动作如下所示。
- 发送之前有取消请求的情况下
- 协议执行状态为“1: 发送等待”情况下的动作如下所示。



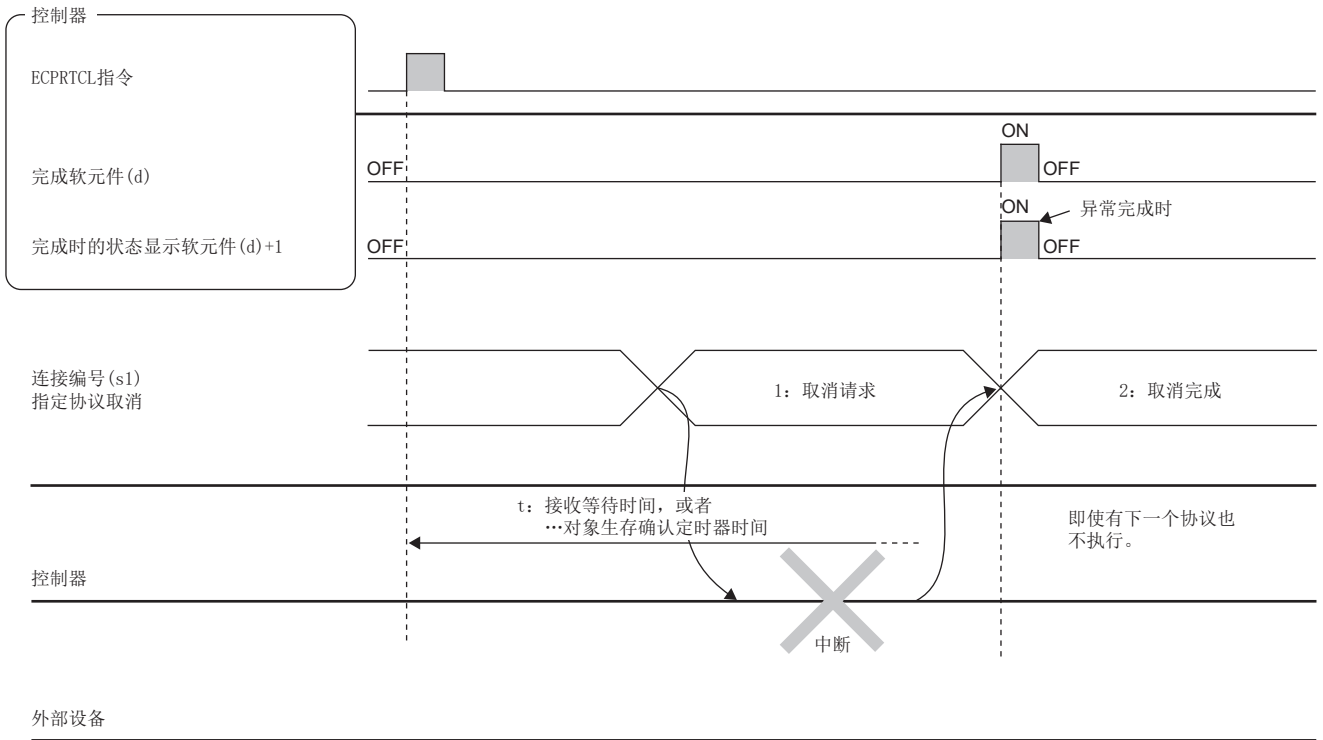
• 发送完成之前有取消请求的情况下
 协议执行状态为“2: 发送中”，未完成发送情况下的动作如下所示。



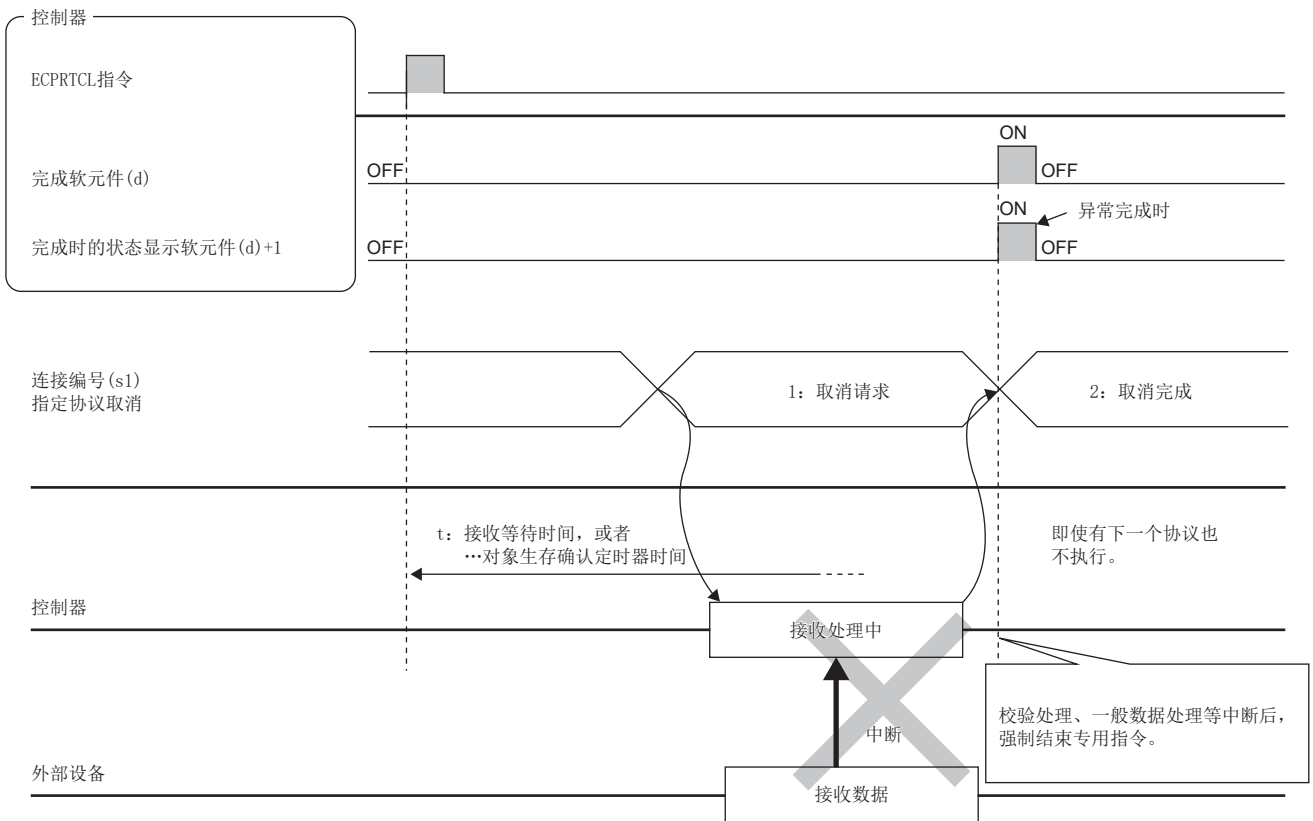
• 发送完成时有取消请求的情况下
 协议执行状态为“2: 发送中”，发送完成情况下的动作如下所示。



• 接收等待中有取消请求的情况下
 协议执行状态为“3: 接收数据等待”情况下的动作如下所示。



• 接收处理中有取消请求的情况下
 协议执行状态为“4: 接收处理中”情况下的动作如下所示。



出错

错误代码 (SD0)	内容
2802H	以无以太网端口的设置执行了以太网功能指令时。
3285H	(s1)超出范围时。

异常完成时，完成时的状态显示软元件(d)+1将变为0N，完成状态(s3)+1中将存储错误代码。

关于完成状态(s3)+1中存储的错误代码，请参阅下述手册。

📖 所使用的控制器的用户手册

注意事项

- 在执行多个协议的情况下，第m个协议中发生错误时，第m+1个及其以后的协议将不执行，指令异常完成。
- 执行包含有未转换转变量的协议的情况下，1个数据包中使用的变量的合计数据长超过了1920字节的情况下，有可能获取跨越了数个扫描的CPU软元件值，因此指令启动时开始至执行完成为止请勿对未转换变量中指定的CPU软元件值进行更改。
- 关于协议取消
 - 本指令异常完成，(s3)+1的完成状态中将存储协议取消请求错误(C404H)。
 - 在没有执行中的协议的状态下，设置了取消请求的情况下，控制器将以无处理进行取消完成。
 - 不使用通信协议的情况下，即使设置了取消请求也将被忽略。
 - 连续执行多个协议的情况下，第n个协议执行中如果进行了取消请求，控制器将强制结束第n个协议，此外，不执行第n+1个及其以后的协议。执行数结果((s3)+0)中将存储执行中的协议编号n，校验一致接收数据包编号1~(n-1)中将存储校验一致的接收数据包编号，完成状态((s3)+1)中将存储协议取消请求错误(C404H)。
 - 控制器通过定期处理确认有无取消请求。因此，从取消请求指示起至取消处理被进行为止有可能需要耗费时间。
- 在本指令单体中，不进行连接的打开/关闭，因此需要通过SOCOPEN指令/SOCCLOSE指令进行连接的打开/关闭处理。

📖 1110页 GP. SOCPEN、SP. SOCPEN

📖 1114页 GP. SOCCLOSE、SP. SOCCLOSE

- 对同一连接执行同一指令的情况下，在先执行中的指令完成之前，后一个指令将被忽略而不执行。
- 将接收等待时间设置为“0：无限等待”的情况下，在协议设置中指定的数据被接收之前本指令不完成。
- 选择通信类型“发送”，通过TCP/IP协议向对象设备发送数据的情况下，在不确认是否到达对象设备下完成本指令的执行，完成状态(S3)+1将变为正常完成。为了确认发送数据已到达对象设备，应选择通信类型“发送&接收”，等待来自对象设备的响应来指令完成。

29.4 SLMP帧发送指令

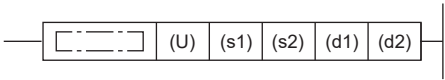
SLMP帧发送

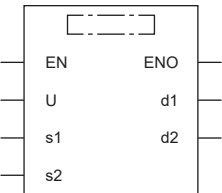
G(P).SLMPSND、J(P).SLMPSND、SP.SLMPSND

对SLMP对应设备发送SLMP的报文。



要点

- G(P).SLMPSND指令可用于以太网用端口、CC-Link IE TSN用端口及扩展模块。
- J(P).SLMPSND指令可用于CC-Link IE TSN用端口与扩展模块。
- SP.SLMPSND指令只能用于以太网用端口。

梯形图	ST
	<pre> 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) </pre>

FBD/LD


■执行条件

指令	执行条件
G.SLMPSND J.SLMPSND	
GP.SLMPSND JP.SLMPSND SP.SLMPSND	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(U)	G(P). SLMPSND	对象的起始输入输出编号 (指定起始输入输出编号的高3位)	以太网端口: 3E0H CC-Link IE TSN端口: 40H	无符号BIN16位	ANY16
	J(P). SLMPSND	本站网络No.	1~239	无符号BIN16位	ANY16
	SP. SLMPSND	虚拟	—	字符串	ANYSTRING_SINGLE
(s1)	G(P). SLMPSND	存储控制数据的起始软元件	☞ 1148页 控制数据	软元件	ANY16* ¹
	J(P). SLMPSND			字	
	SP. SLMPSND				
(s2)	存储请求帧的起始软元件	☞ 1149页 请求帧	字	ANY16* ¹	
(d1)	存储响应帧的起始软元件	☞ 1149页 响应帧	字	ANY16_ARRAY* ²	
(d2)	指令完成时通过1个扫描变为ON的软元件 异常完成时(d2)+1也将变为ON。	—	位	ANYBIT_ARRAY* ² (要素数: 2)	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

*1 通过标签进行设置的情况下,应在确保动作所需区域的前提下定义数组,指定该数组型标签的要素。

*2 通过标签进行设置的情况下,应在确保动作所需区域的前提下定义数组。

■可用软元件

操作数	位	字			双字		间接指定	常数			其它		
		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
(U)	G(P). SLMPSND	—	—	○	—	—	—	○	—	—	○	○	—
	SP. SLMPSND	—	—	—	—	—	—	—	—	—	—	—	○
(s1)	—	—	○	—	—	—	○	—	—	—	—	—	
(s2)	—	—	○	—	—	—	○	—	—	—	—	—	
(d1)	—	—	○	—	—	—	○	—	—	—	—	—	
(d2)	○	—	○* ¹	—	—	—	○	—	—	—	—	—	

*1 T、ST、C、FD不能使用。

■控制数据

操作数：(s1)				
软元件	项目	内容	设置范围	设置方
+0	执行・异常时完成类型	<div style="display: flex; justify-content: space-around; align-items: center;"> b15 b7 b0 </div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px;">(3)</div> <div style="border: 1px solid black; padding: 2px;">0</div> <div style="border: 1px solid black; padding: 2px;">(2)</div> <div style="border: 1px solid black; padding: 2px;">0</div> <div style="border: 1px solid black; padding: 2px;">(1)</div> </div> <p>(1) 执行类型(位0)</p> <ul style="list-style-type: none"> • 0: 无到达确认(通过从本站发送了请求报文而完成。)*¹ • 1: 有到达确认(通过对对象设备接收了响应报文而完成。) <p>(2) 异常时完成类型(位7)</p> <p>指定异常完成时的数据的设置状态。</p> <ul style="list-style-type: none"> • 0: 不将异常完成时的数据设置到(s1)+13及其以后。(清除(s1)+13及其以后。) • 1: 将异常完成时的数据设置到(s1)+13及其以后。 <p>(3) 指令类型(位15)</p> <ul style="list-style-type: none"> • 0: 操作数(U)为00H~FEH指定的情况下 • 1: 操作数(U)为3E0H指定的情况下 	—	用户
+1	完成状态	<p>村粗指令完成时的状态。</p> <p>0000H: 正常</p> <p>0000H以外: 异常(错误代码)</p>	—	系统
+2	本站使用通道	<p>指定本站使用的通道。根据通道决定是否在请求报文中附加序列号*⁴, 因此应根据用途指定通道。</p> <p>同时使用SLMPSEND指令的情况下, 附加序列号的通道为共享, 因此应避免重复设置。</p> <ul style="list-style-type: none"> • 1: 不附加序列号的通道 • 2~9: 附加序列号的通道 • 10~17: 站号扩展帧 	• 1~17* ²	用户
+3	对象设备IP地址(第3、4八位字节)	<p>指定对象设备IP地址(第3、4八位字节)。</p> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 5px;"> b15 b8 b7 b0 </div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> </div> <p>3、4: 表示IP地址的八位字节。</p>	根据(s1)+3、(s1)+4, 00000001H~FFFFFFFEH (1~4294967294)	用户
+4	对象设备IP地址(第1、2八位字节)	<p>指定对象设备IP地址(第1、2八位字节)。</p> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 5px;"> b15 b8 b7 b0 </div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> </div> <p>1、2: 表示IP地址的八位字节。</p>		用户
+5	对象设备端口编号	指定对象设备的端口编号。	1~65534 (1~FFFFEH)	用户
+6	请求目标网络编号	固定为0000H	0000H	用户
+7	请求目标站号	固定为00FFH	00FFH	用户
+8	请求目标模块I/O编号	<p>指定请求目标的模块I/O号。</p> <ul style="list-style-type: none"> • 03FFH: 本站/管理CPU • 03E0H: 多CPU1号机 • 03E1H: 多CPU2号机 • 03E2H: 多CPU3号机 • 03E3H: 多CPU4号机 	03FFH、03E0H~03E3H	用户
+9	请求目标多点站号	指定对象设备的多点编号。	0000H~FFFEH* ³	用户
+10	重新发送次数	<p>(s1)+0中指定的执行类型为“1: 有到达确认”时将生效。</p> <p>■执行指令时</p> <p>指定在(s1)+11中指定的监视时间内未完成的情况下, 重新发送的次数。</p> <ul style="list-style-type: none"> • 0~15(次) <p>■指令完成时</p> <p>存储重新发送的进行次数(结果)。</p> <ul style="list-style-type: none"> • 0~15(次) 	0~15	用户/系统
+11	到达监视时间	<p>指定处理完成为止的监视时间。在监视时间内未完成的情况下, 将重新发送直至达到(s1)+10中指定的次数为止。</p> <ul style="list-style-type: none"> • 0: 10秒 • 1~32767: 1~32767秒 	0~32767	用户
+12	时钟设置标志	<p>存储(s)+13及其以后的数据的有效/无效状态。此外, 当正常完成时, (s1)+13及其以后的数据将被清除。</p> <ul style="list-style-type: none"> • 0: 无效 • 1: 有效 	—	系统

操作数：(s1)						
软元件	项目	内容	设置范围	设置方		
+13	时钟数据	高位8位：月(01H~12H) 低位8位：年(00H~99H) 公历低2位	—	系统		
+14		高位8位：时(00H~23H) 低位8位：日(01H~31H)				
+15		高位8位：秒(00H~59H) 低位8位：分(00H~59H)				
+16		高位8位：年(00H~99H) 公历高2位 低位8位：星期(00H(日)~06H(六))				
+17	异常检测设备IP地址(第3、4八位字节)	存储检测出异常的设备的IP地址(第3、4八位字节)。 <div style="display: flex; justify-content: space-around; align-items: center;"> b15 b8 b7 b0 </div> <table border="1" style="margin: 5px auto; width: 80%;"> <tr> <td style="width: 50%; text-align: center;">3</td> <td style="width: 50%; text-align: center;">4</td> </tr> </table> 3、4：表示IP地址的八位字节。	3	4	—	系统
3	4					
+18	异常检测设备IP地址(第1、2八位字节)	存储检测出异常的设备的IP地址(第1、2八位字节)。 <div style="display: flex; justify-content: space-around; align-items: center;"> b15 b8 b7 b0 </div> <table border="1" style="margin: 5px auto; width: 80%;"> <tr> <td style="width: 50%; text-align: center;">1</td> <td style="width: 50%; text-align: center;">2</td> </tr> </table> 1、2：表示IP地址的八位字节。	1	2	—	系统
1	2					

*1 设置了(s1)+0的无到达确认的情况下，不能设置接收数据。在下述情况下应设置(s1)+0的无到达确认。

- 使用不返回响应报文的指令的情况下
- 不参照响应报文的情况下

*2 操作数(U)指定了3E0H的情况下，可以指定1~9。

*3 操作数(U)指定了3E0H的情况下，固定为0000H。

*4 向同一SLMP对应设备发送多个请求报文的情况下附加。附加的序列号由系统自动确定。

关于序列号，请参阅下述手册。

□□SLMP参考手册

■请求帧

操作数：(s2)				
软元件	项目	内容	设置范围	设置方
+0	请求数据长	指定从监视定时器至请求数据为止的数据长。(字节单位)	1~2000	用户
+1	监视定时器	是设置接收了请求报文的对象设备向访问目标请求处理后至返回响应为止的等待时间的定时器。(单位：250ms) <ul style="list-style-type: none"> • 0：无限等待 • 1~65535：1~65535×250ms 	0~65535	用户
+2~+□	请求数据	存储SLMP报文的请求数据。	—	用户

■响应帧

操作数：(d1)				
软元件	项目	内容	设置范围	设置方
+0	响应数据长	存储从结束代码起至响应数据为止的数据长。(字节单位)	2~2000	系统
+1	结束代码	存储指令处理结果。 正常结束时存储0。异常结束时存储对象设备中设置的错误代码。	—	系统
+2~+□	响应数据*1	设置对请求数据的执行结果。 关于请求数据的详细内容，请参阅下述手册。 □□SLMP参考手册	—	系统

*1 响应数据以字节单位从低位字节侧开始依次被存储。接收到奇数字节的响应数据时，最终数据存储区域的低位字节中将存储最后的响应数据。

功能

- 将(s2)中指定的软元件及其以后的请求帧，发送至控制数据的对象设备IP地址中指定的对象设备。通过对象设备接收响应报文时存储到(d1)中指定的软元件中。
- 本指令通过UDP进行通信。对象设备的设置也应设置为UDP。
- 本指令通过二进制代码进行通信。对象设备的设置也应根据二进制代码。
- 本指令的执行及正常/异常完成，可通过设置数据中指定的完成软元件(d2)+0、完成时的状态显示软元件(d2)+1进行确认。

• 完成软元件(d2)+0

在本指令完成扫描的END处理中变为ON，在下一个END处理中变为OFF。

• 完成时的状态显示软元件(d2)+1

根据本指令完成时的状态而进行ON/OFF。

正常完成时：保持OFF状态不变。

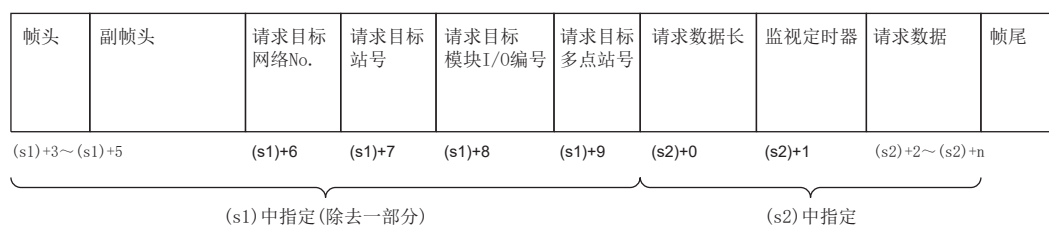
异常完成时：在本指令完成扫描的END处理中变为ON，在下一个END处理中变为OFF。

- 接收到奇数字节的响应数据的情况下，存储了最后的响应数据的软元件的高位字节中将放入无效的数据。

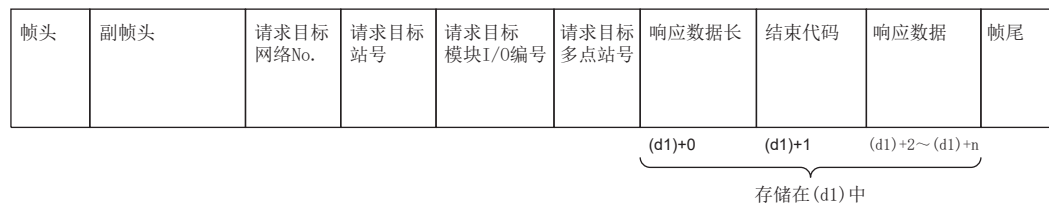
■报文格式(3E/4E帧)

3E/4E帧的请求报文和正常/异常结束时的响应报文如下所示。

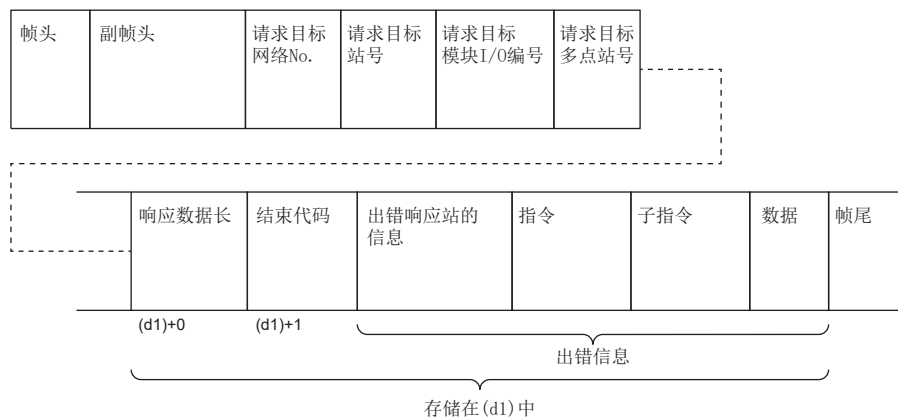
- 请求报文



- 响应报文(正常结束时)



- 响应报文(异常结束时)



将网络No.、站号、请求目标模块I/O编号、多点站号存储至错误响应站的信息。

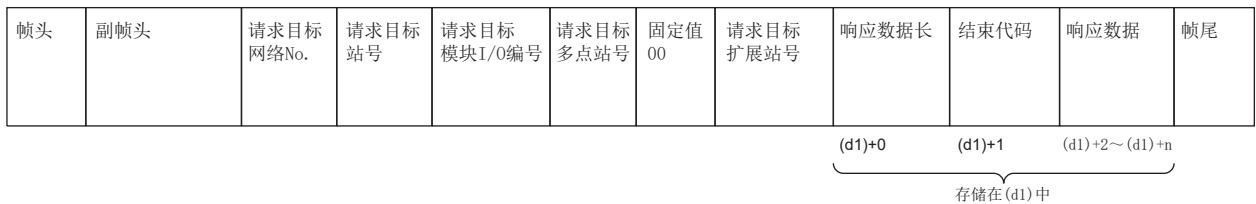
■报文格式(站号扩展帧)

站号扩展帧的请求报文和正常/异常结束时的响应报文如下所示。

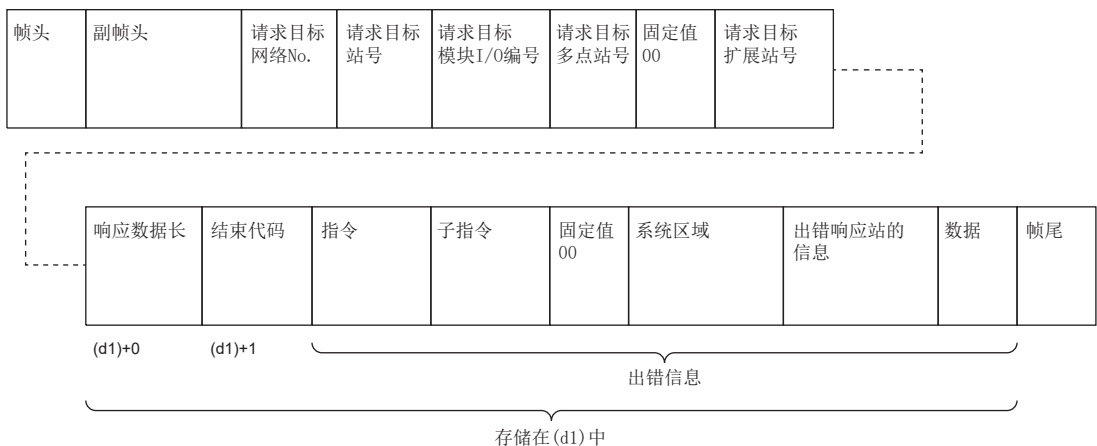
• 请求报文



• 响应报文(正常结束时)



• 响应报文(异常结束时)

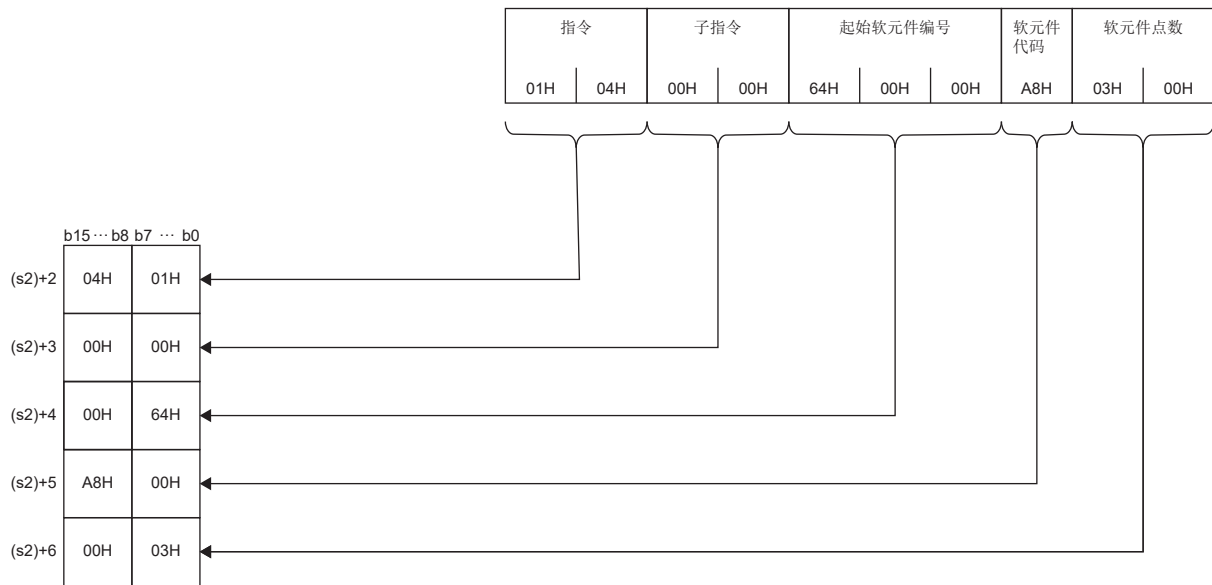


将网络No.、站号、请求目标模块I/O编号、多点站号、00(固定值)、扩展站号存储至错误响应站的信息。

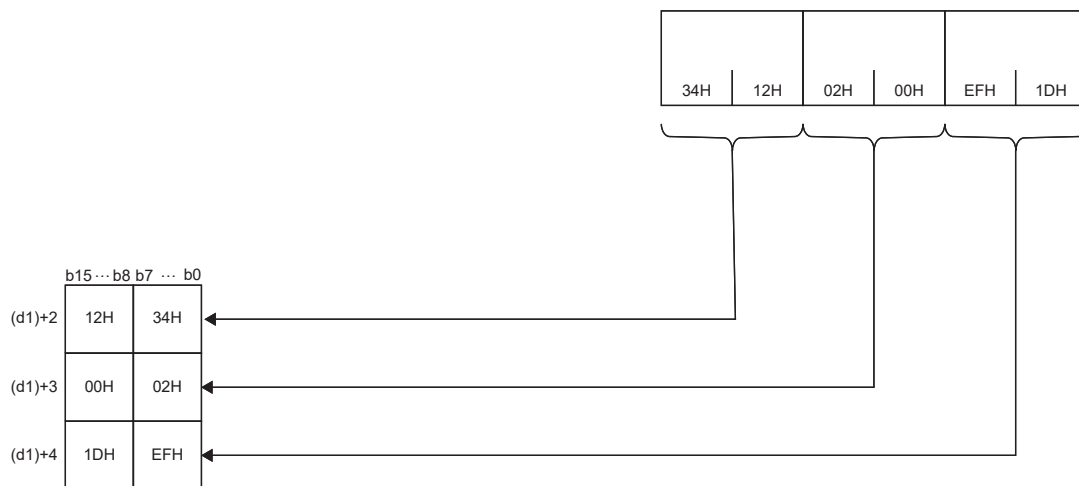
例

发送读取D100~D102的值“Read(指令: 0401H)”(字单位读取)的情况下

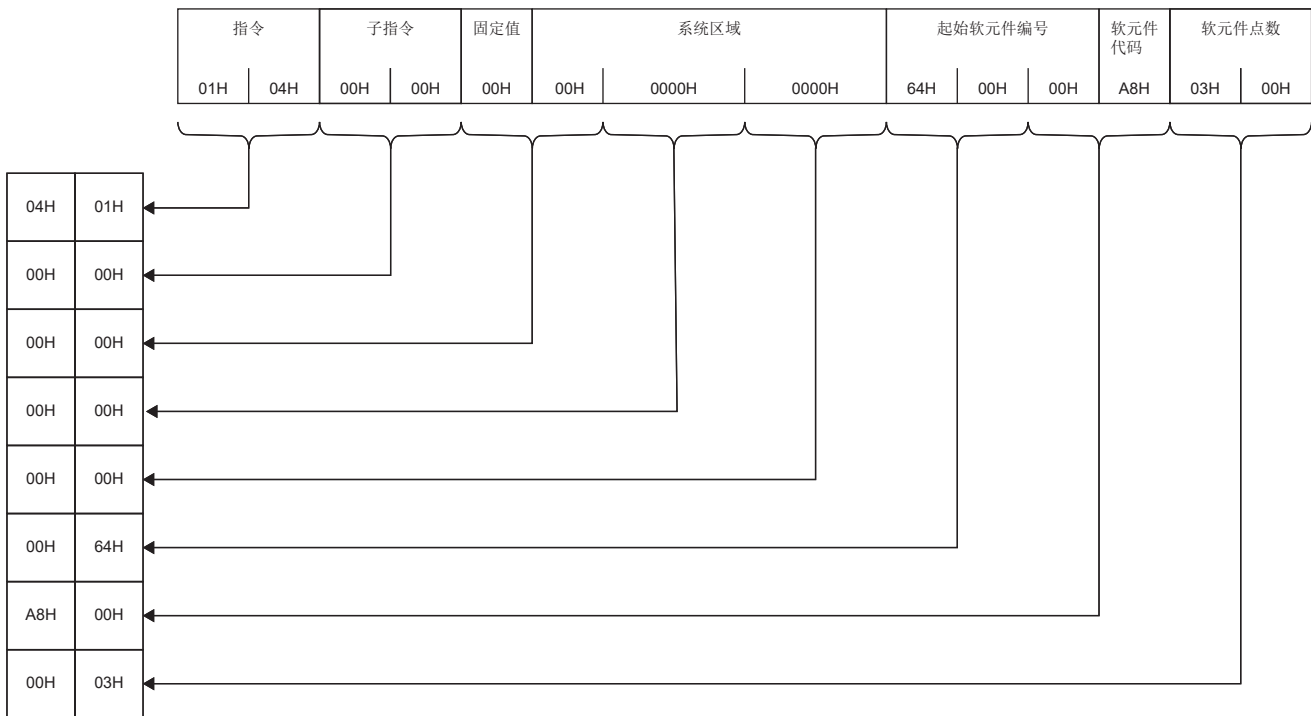
- 将请求数据存储在 (s2)+2 及以后的情况下 (3E/4E 帧)



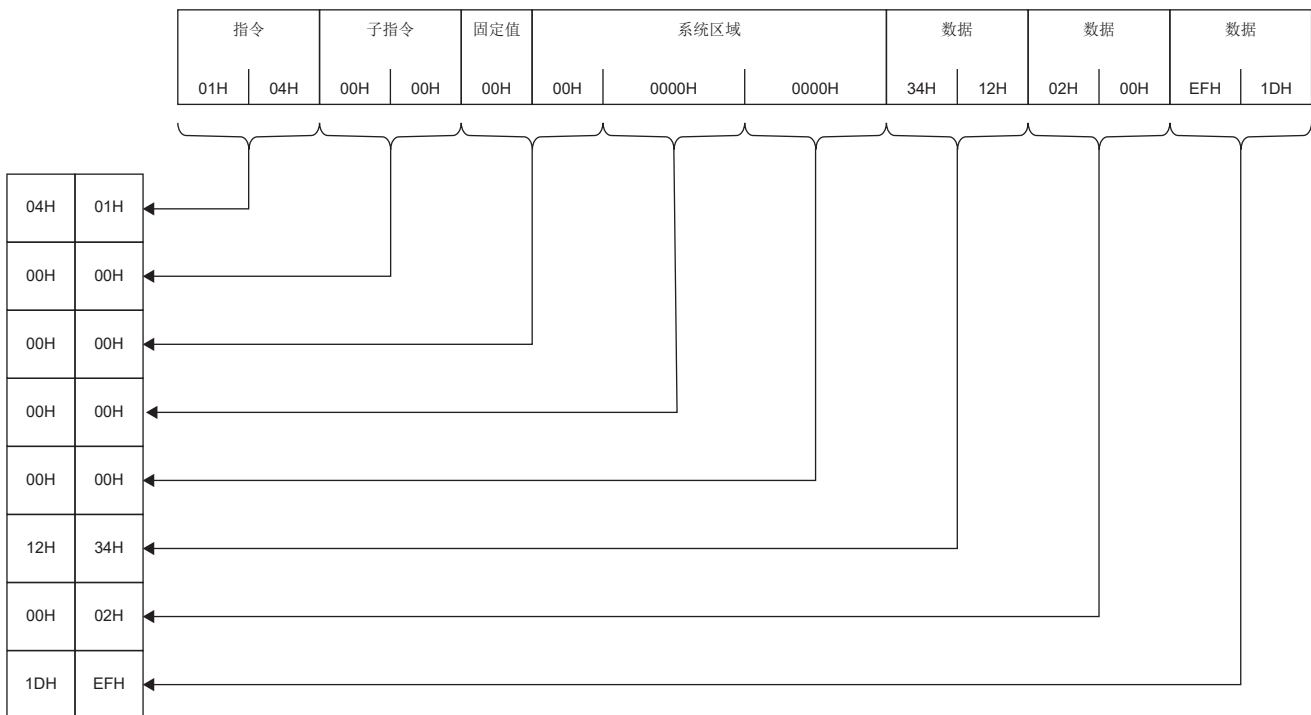
- 将响应数据存储在 (d1)+2 及以后的情况下 (3E/4E 帧)



• 将请求数据存储至 (s2)+2 及以后的示例 (站号扩展帧)

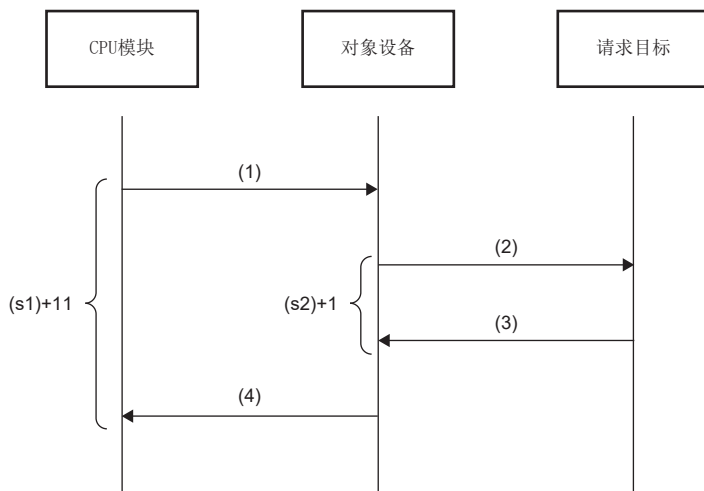


• 将响应数据存储至 (d1)+2 及以后的情况下 (站号扩展帧)



注意事项

- 执行SLMPSND指令的过程中执行了设置为同一通道的SLMPSND指令的情况下，之后执行的指令将被忽略。此时，在指令的执行条件成立的期间，先执行的指令完成后执行请求。请求被忽略期间，SM699(专用指令未执行标志)将为ON。
- 控制数据的到达监视时间((s1)+11)及请求帧的监视定时器((s2)+1)，应设置为到达监视时间≥监视定时器。



- (1) 请求报文
- (2) 从对象设备至请求目标的处理请求
- (3) 从请求目标至对象设备的处理响应
- (4) 响应报文

要点

本指令，即使对象设备返回了异常响应的情况下也将正常完成。本指令正常完成的情况下，根据响应帧的结束代码，判断是正常响应还是异常响应。异常响应的情况下，请通过所使用的SLMP对应设置的手册进行确认及处理。

出错

错误代码(SD0)	内容
2802H	以无以太网端口的设置执行了以太网功能指令时。
3285H	(s1)+2中指定的本站使用通道为范围外时。
	(s2)+0中指定的请求数据长为0或超过2000时。

异常完成时，完成时的状态显示软元件(d2)+1将变为ON，完成状态(s1)+1中将存储错误代码。关于完成状态(s1)+1中存储的错误代码，请参阅下述手册。

📖所使用的控制器的用户手册

加密的SLMP帧发送

G(P). SLMPSNDC

对SLMP对应设备发送加密的SLMP报文。

要点

本指令可用于以太网用端口与扩展模块。

梯形图	ST
	ENO:=G_SLMPSNDC(EN, U, s1, s2, d1, d2) ENO:=GP_SLMPSNDC(EN, U, s1, s2, d1, d2)

FBD/LD

■执行条件

指令	执行条件
G. SLMPSNDC	
GP. SLMPSNDC[G]	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(U)	对象的起始输入输出编号 (指定起始输入输出编号的高3位)	3E0H	无符号BIN16位	ANY16
(s1)	存储控制数据的起始软元件	1148页 控制数据	软元件名	ANY16* ¹
(s2)	存储请求帧的起始软元件	1149页 请求帧	软元件名	ANY16* ¹
(d1)	存储响应帧的起始软元件	1149页 响应帧	软元件名	ANY16_ARRAY* ²
(d2)	指令完成时通过1个扫描变为ON的软元件 异常完成时(d2)+1也将变为ON。	—	位	ANYBIT_ARRAY* ² (要素数: 2)
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组，指定该数组型标签的要素。

*2 通过标签进行设置的情况下，应在确保动作所需区域的前提下定义数组。

■可用软元件

操作数	位		字				双字		间接指定	常数			其它 (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不能使用。

■控制数据

操作数：(s1)									
软元件	项目	内容	设置范围	设置方					
+0	执行·异常时完成类型	b15 b7 b0 <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; width: 30px; text-align: center;">(3)</td> <td style="border: 1px solid black; width: 150px; text-align: center;">0</td> <td style="border: 1px solid black; width: 30px; text-align: center;">(2)</td> <td style="border: 1px solid black; width: 150px; text-align: center;">0</td> <td style="border: 1px solid black; width: 30px; text-align: center;">(1)</td> </tr> </table> <p>(1) 执行类型(位0) • 0: 无到达确认(通过从本站发送了请求报文而完成。)*1 • 1: 有到达确认(通过对对象设备接收了响应报文而完成。)</p> <p>(2) 异常时完成类型(位7) 指定异常完成时的数据的设置状态。 • 0: 不将异常完成时的数据设置到(s1)+13及其以后。(清除(s1)+13及其以后。) • 1: 将异常完成时的数据设置到(s1)+13及其以后。</p>	(3)	0	(2)	0	(1)	—	用户
(3)	0	(2)	0	(1)					
+1	完成状态	村粗指令完成时的状态。 0: 正常 0以外: 异常(错误代码)	—	系统					
+2	本站使用通道	指定本站使用的通道。根据通道决定是否在请求报文中附加序列号*2, 因此应根据用途指定通道。 • 1: 不附加序列号的通道 • 2~9: 附加序列号的通道	1~9	用户					
+3	本站端口编号	指定本站的端口编号。设置附加序列号的通道时, 应避免指定重复的本站端口编号。同时使用SLMPSND指令的情况下, 附加序列号的通道为共享, 因此应避免重复设置。	1~4999、5010~65534 (1H~1387H、1392H~FFFEH)	用户					
+4	对象设备IP地址(第3、4八位字节)	指定对象设备IP地址(第3、4八位字节)。 b15 b8 b7 b0 <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; width: 30px; text-align: center;">3</td> <td style="border: 1px solid black; width: 150px; text-align: center;">4</td> </tr> </table> 3、4: 表示IP地址的八位字节。	3	4	根据(s1)+3、(s1)+4, 00000001H~FFFFFFFEH (1~4294967294)	用户			
3	4								
+5	对象设备IP地址(第1、2八位字节)	指定对象设备IP地址(第1、2八位字节)。 b15 b8 b7 b0 <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; width: 30px; text-align: center;">1</td> <td style="border: 1px solid black; width: 150px; text-align: center;">2</td> </tr> </table> 1、2: 表示IP地址的八位字节。	1	2		用户			
1	2								
+6	对象设备端口编号	指定对象设备的端口编号。	1~65534 (1~FFFEH)	用户					
+7	请求目标网络编号	固定为0000H	0000H	用户					
+8	请求目标站号	固定为00FFH	00FFH	用户					
+9	请求目标模块I/O编号	指定请求目标的模块I/O号。 • 03FFH: 本站/管理CPU • 03E0H: 多CPU1号机 • 03E1H: 多CPU2号机 • 03E2H: 多CPU3号机 • 03E3H: 多CPU4号机	03FFH、03E0H~03E3H	用户					
+10	请求目标多点站号	固定为0000H	0000H	用户					

操作数: (s1)										
软元件	项目	内容	设置范围	设置方						
+11	重新发送次数	(s1)+0中指定的执行类型为“1: 有到达确认”时将生效。 ■执行指令时 指定在(s1)+11中指定的监视时间内未完成的情况下, 重新发送的次数。 • 0~15(次) ■指令完成时 存储重新发送的进行次数(结果)。 • 0~15(次)	0~15	用户/系统						
+12	到达监视时间	指定处理完成为止的监视时间。在监视时间内未完成的情况下, 将重新发送直至达到(s1)+10中指定的次数为止。 • 0: 10秒 • 1~32767: 1~32767秒	0~32767	用户						
+13	时钟设置标志	存储(s)+13及其以后的数据的有效/无效状态。此外, 当正常完成时, (s1)+13及其以后的数据将被清除。 • 0: 无效 • 1: 有效	—	系统						
+14	时钟数据	高位8位: 月(01H~12H) 低位8位: 年(00H~99H) 农历低2位	—	系统						
+15		高位8位: 时(00H~23H) 低位8位: 日(01H~31H)								
+16		高位8位: 秒(00H~59H) 低位8位: 分(00H~59H)								
+17		高位8位: 年(00H~99H) 农历高2位 低位8位: 星期(00H(日)~06H(六))								
+18	异常检测设备IP地址(第3、4八位字节)	存储检测出异常的设备的IP地址(第3、4八位字节)。 <table border="1" style="margin-left: 20px;"> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">b8 b7</td> <td style="text-align: center;">b0</td> </tr> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">4</td> <td></td> </tr> </table> 3、4: 表示IP地址的八位字节。	b15	b8 b7	b0	3	4		—	系统
b15	b8 b7	b0								
3	4									
+19	异常检测设备IP地址(第1、2八位字节)	存储检测出异常的设备的IP地址(第1、2八位字节)。 <table border="1" style="margin-left: 20px;"> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">b8 b7</td> <td style="text-align: center;">b0</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td></td> </tr> </table> 1、2: 表示IP地址的八位字节。	b15	b8 b7	b0	1	2		—	系统
b15	b8 b7	b0								
1	2									

*1 设置了(s1)+0的无到达确认的情况下, 不能设置接收数据。在下述情况下应设置(s1)+0的无到达确认。

- 使用不返回响应报文的指令的情况下
- 不参照响应报文的的情况下

*2 向同一SLMP对应设备发送多个请求报文的情况下附加。附加的序列号由系统自动确定。

关于序列号, 请参阅下述手册。

📖 SLMP参考手册

■请求帧

操作数: (s2)				
软元件	项目	内容	设置范围	设置方
+0	请求数据长	指定从监视定时器至请求数据为止的数据长。(字节单位)	1~2000	用户
+1	监视定时器	是设置接收了请求报文的对象设备向访问目标请求处理后至返回响应为止的等待时间的定时器。(单位: 250ms) • 0: 无限等待 • 1~65535: 1~65535×250ms	0~65535	用户
+2~+□	请求数据	存储SLMP报文的请求数据。	—	用户

■响应帧

操作数: (d1)				
软元件	项目	内容	设置范围	设置方
+0	响应数据长	存储从结束代码起至响应数据为止的数据长。(字节单位)	2~2000	系统
+1	结束代码	存储指令处理结果。 正常结束时存储0。异常结束时存储对象设备中设置的错误代码。	—	系统
+2~+□	响应数据*1	设置对请求数据的执行结果。 (根据指令, 有时不返回响应数据。)	—	系统

*1 响应数据以字节单位从低位字节侧开始依次被存储。接收到奇数字节的响应数据时, 最终数据存储区域的低位字节中将存储最后的响应数据。

功能

- 将(s2)中指定的软元件及其以后的请求帧，发送至控制数据的对象设备IP地址中指定的对象设备。通过对象设备接收响应报文时存储到(d1)中指定的软元件中。
- 本指令通过DTLS进行通信。对象设备的设置也应设置为DTLS。
- 本指令通过二进制代码进行通信。对象设备的设置也应根据二进制代码。
- 本指令的执行及正常/异常完成，可通过设置数据中指定的完成软元件(d2)+0、完成时的状态显示软元件(d2)+1进行确认。

• 完成软元件(d2)+0

在本指令完成扫描的END处理中变为ON，在下一个END处理中变为OFF。

• 完成时的状态显示软元件(d2)+1

根据本指令完成时的状态而进行ON/OFF。

正常完成时：保持OFF状态不变。

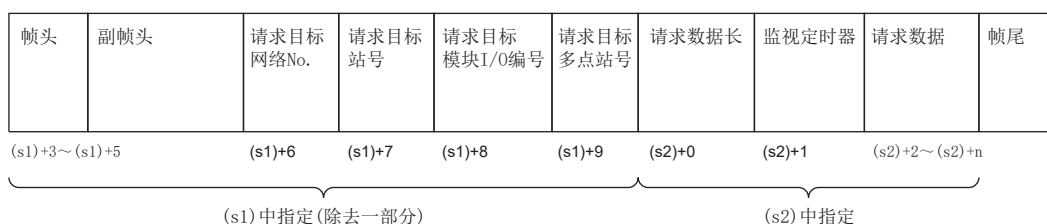
异常完成时：在本指令完成扫描的END处理中变为ON，在下一个END处理中变为OFF。

- 接收到奇数字节的响应数据的情况下，存储了最后的响应数据的软元件的高位字节中将放入无效的数据。

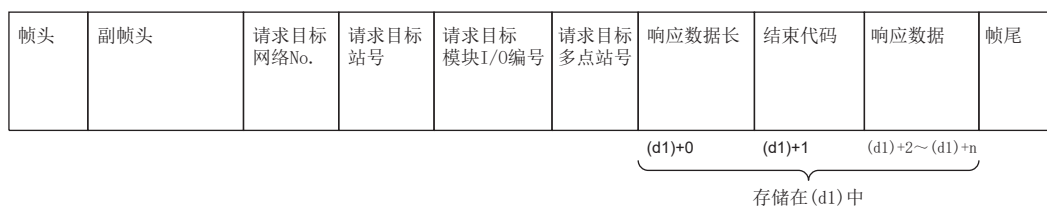
■报文格式(3E/4E帧)

3E/4E帧的请求报文和正常/异常结束时的响应报文如下所示。

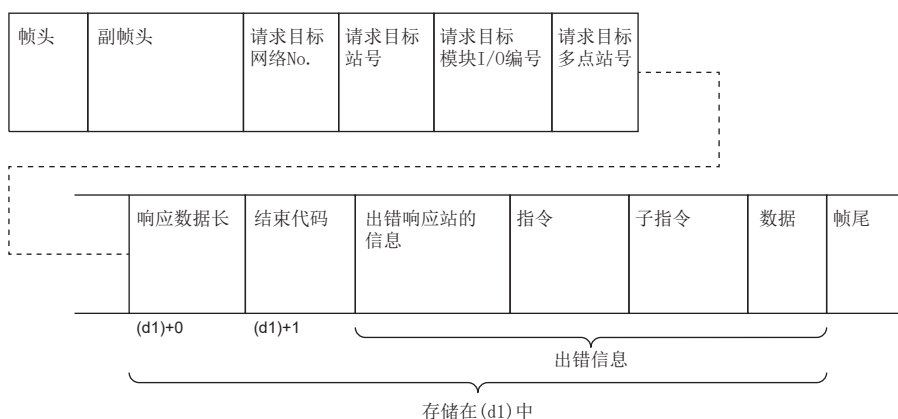
- 请求报文



- 响应报文(正常结束时)



- 响应报文(异常结束时)



将网络No.、站号、请求目标模块I/O编号、多点站号存储至错误响应站的信息。

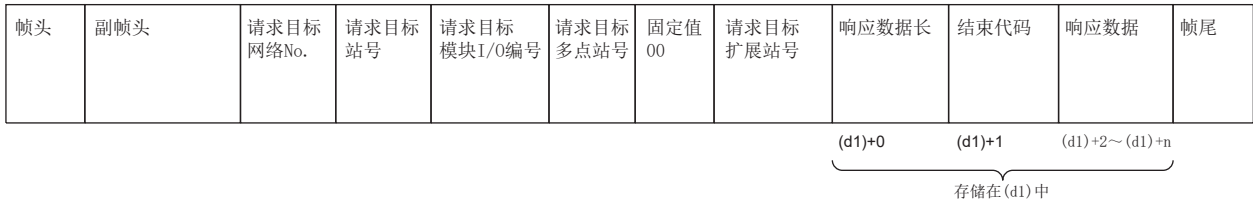
■报文格式(站号扩展帧)

站号扩展帧的请求报文和正常/异常结束时的响应报文如下所示。

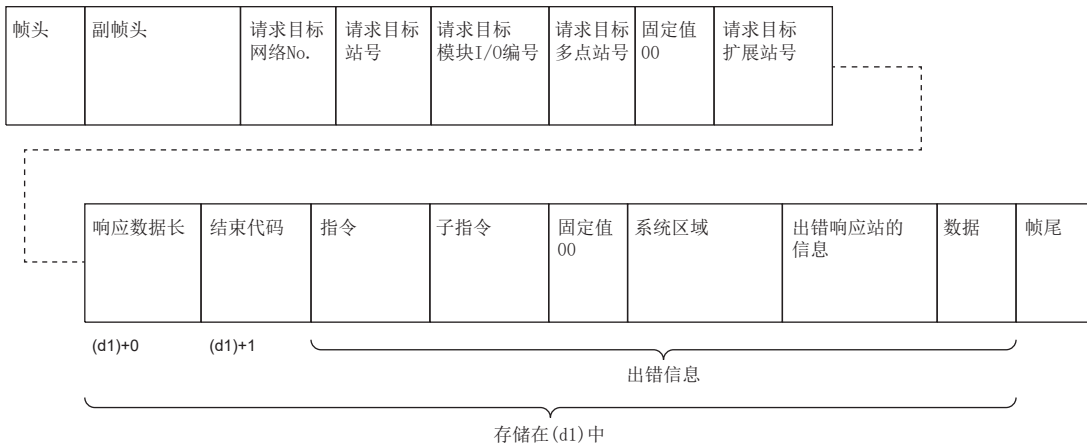
• 请求报文



• 响应报文(正常结束时)



• 响应报文(异常结束时)

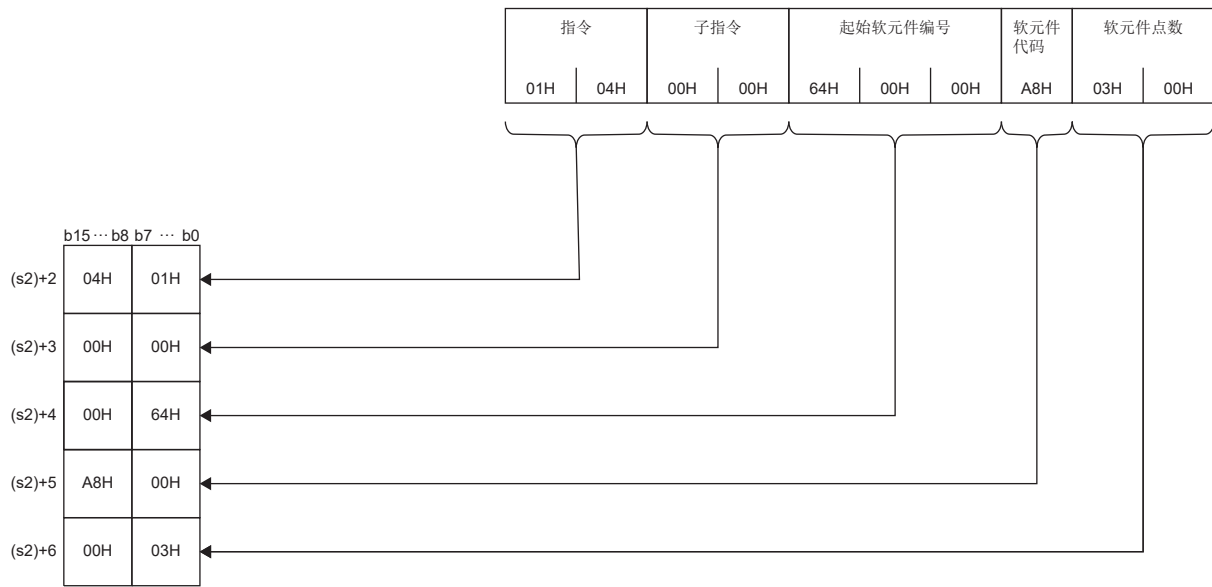


将网络No.、站号、请求目标模块I/O编号、多点站号、00(固定值)、扩展站号存储至错误响应站的信息。

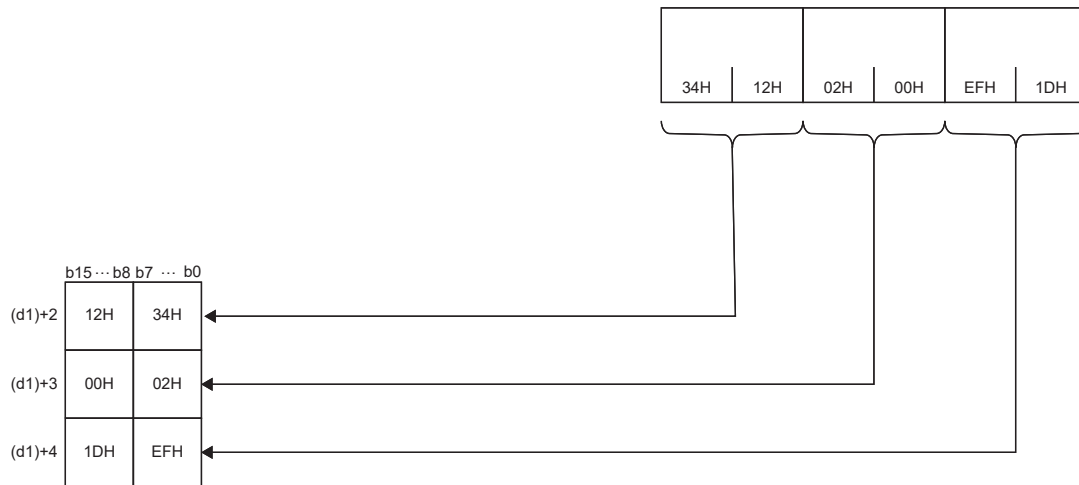
例

发送读取D100~D102的值“Read(指令: 0401H)”(字单位读取)的情况下

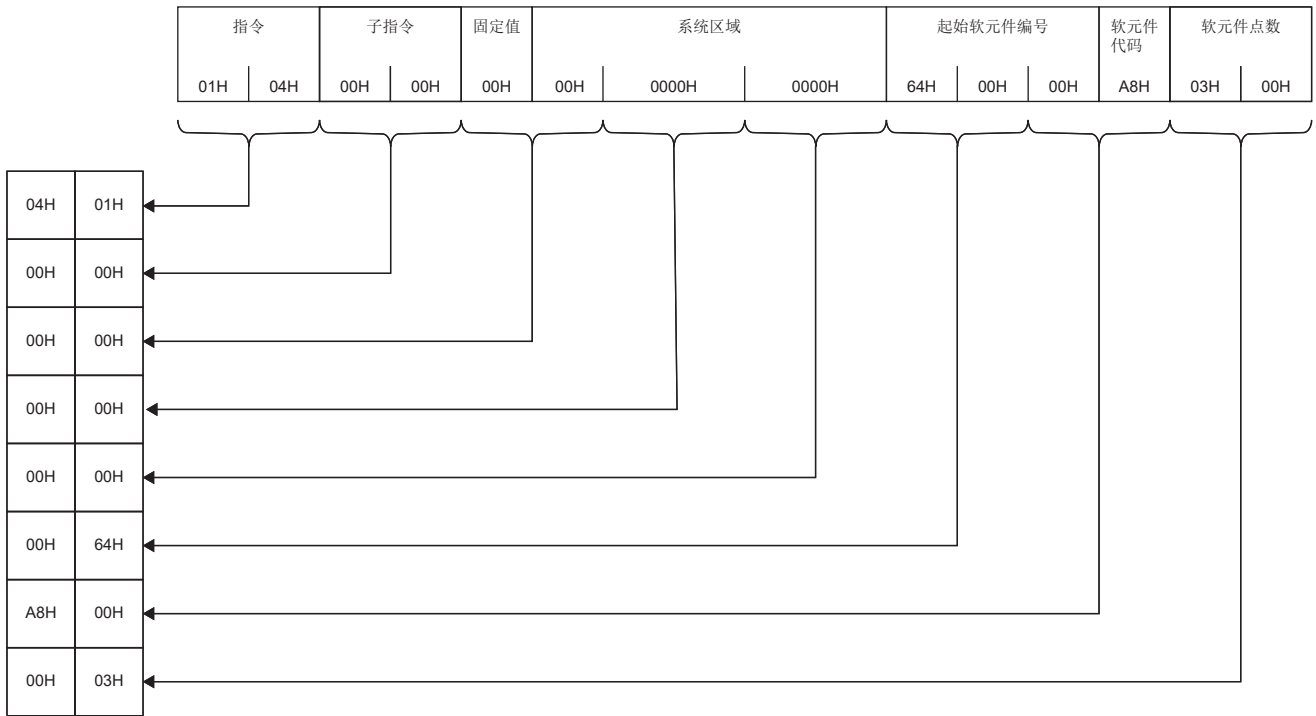
- 将请求数据存储至(s2)+2及以后的情况下(3E/4E帧)



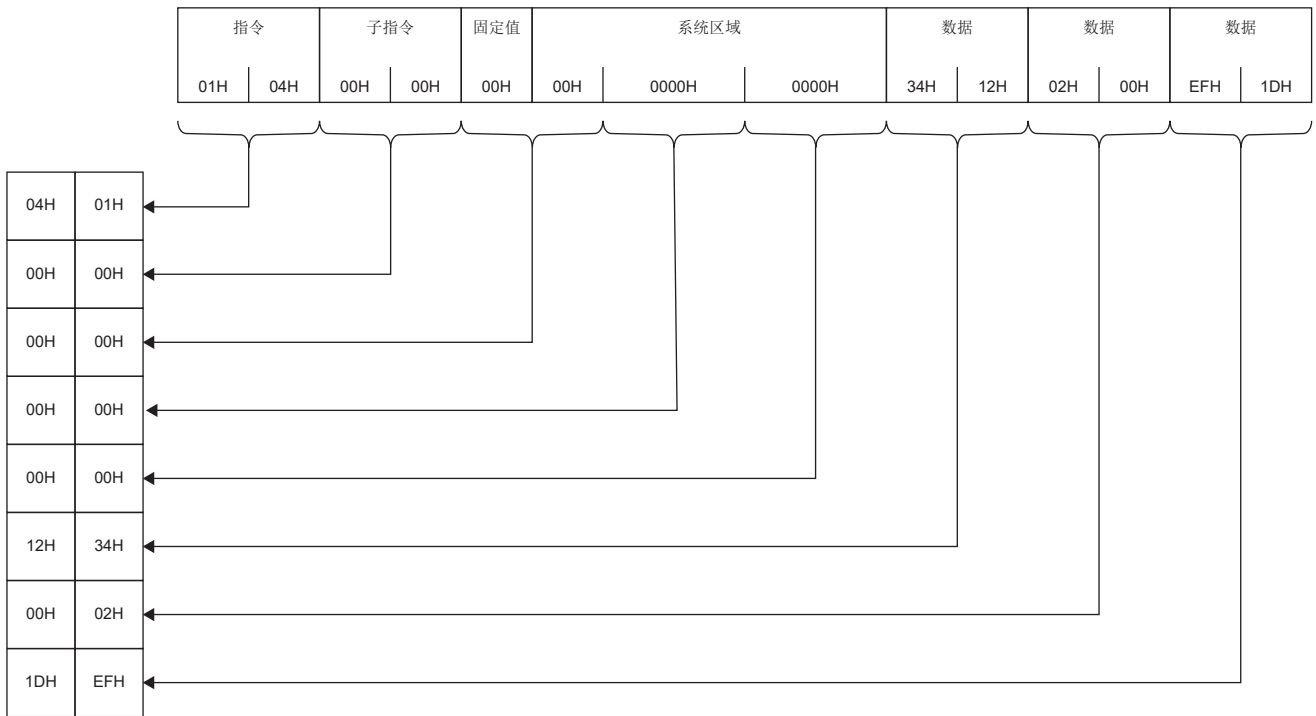
- 将响应数据存储至(d1)+2及以后的情况下(3E/4E帧)



• 将请求数据存储至 (s2)+2 及以后的示例 (站号扩展帧)

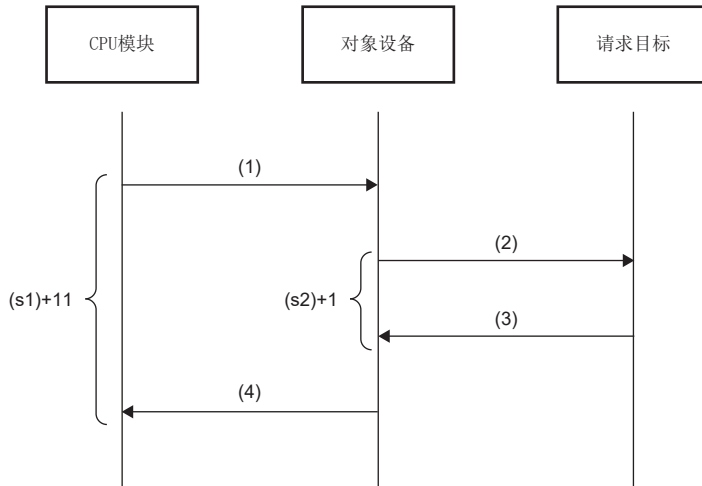


• 将响应数据存储至 (d1)+2 及以后的情况下 (站号扩展帧)



注意事项

- 同时执行多个本指令的情况下，设置时应避免本指令的通道重复。设置了同一通道的本指令不能同时使用。在同一顺控扫描内满足同一通道本指令的执行条件的情况下，仅执行最初已被执行的指令，不执行在那之后的指令。即使执行了执行中的指令与同一通道设置的指令，也不会执行之后执行的指令。在控制器未执行本指令的处理时，SM699(专用指令未执行标志)将变为ON。
- 控制数据的到达监视时间((s1)+11)及请求帧的监视定时器((s2)+1)，应设置为到达监视时间>监视定时器。



- (1) 请求报文
 (2) 从对象设备至请求目标的处理请求
 (3) 从请求目标至对象设备的处理响应
 (4) 响应报文

要点

本指令，即使对象设备返回了异常响应的情况下也将正常完成。本指令正常完成的情况下，根据响应帧的结束代码，判断是正常响应还是异常响应。异常响应的情况下，请通过所使用的SLMP对应设置的手册进行确认及处理。

出错

错误代码 (SD0)	内容
1193H	附加了序列号不同的通道中，本站端口编号相同的情况下
2802H	以无以太网端口的设置执行了以太网功能指令时。
3285H	(s1)+2中指定的本站使用通道为1~9以外时。
	(s2)+0中指定的请求数据长为0或超过2000时。

异常完成时，完成时的状态显示软元件(d2)+1将变为ON，完成状态(s1)+1中将存储错误代码。关于完成状态(s1)+1中存储的错误代码，请参阅下述手册。

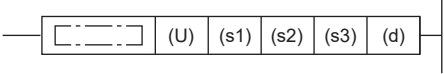
📖 所使用的控制器的用户手册

29.5 文件传送功能用指令

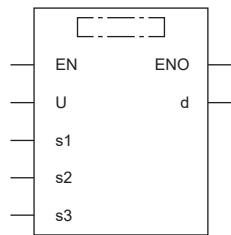
FTP客户端文件发送

GP. FTPCPUT、SP. FTTPUT


将(s2)中指定的控制器的文件发送至(s3)中指定的FTP服务器的文件夹路径。

梯形图	ST
	ENO:=GP_FTPCPUT(EN, U, s1, s2, s3, d); ENO:=SP_FTTPUT(EN, U, s1, s2, s3, d);

FBD/LD



■执行条件

指令	执行条件
GP. FTFCPUT SP. FTTPUT	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(U)	GP. FTFCPUT	对象的起始输入输出编号 (指定起始输入输出编号的高3位)	3EOH	无符号BIN16位	ANY16
	SP. FTTPUT	虚拟	—	字符串	ANYSTRING_SINGLE
(s1)	GP. FTFCPUT	存储控制数据的起始软元件	参阅控制数据	软元件	ANY16
	SP. FTTPUT			字	ANY16_ARRAY (要素数: 4)
(s2)	控制器中存储的文件名(传送源)*1	—	Unicode字符串*2	ANYSTRING_DOUBLE	
(s3)	FTP服务器的文件夹路径(传送目标)*1	—	Unicode字符串*2	ANYSTRING_DOUBLE	
(d)	指令完成时通过1个扫描变为ON的软元件 异常完成时(d)+1也将变为ON。	—	位	ANYBIT_ARRAY (要素数: 2)	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

*1 是Unicode字符串或存储了Unicode字符串的起始软元件。

*2 数据类型为Unicode字符串, 但只能使用半角英文数字、符号、假名字符、全角字符(移位JIS代码)。不支持的字符将被处理为“_”。

■可用软元件

操作数	位		字		双字		间接指定	常数			其它 (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)	—	—	○	—	—	—	—	○	—	—	○
(s3)	—	—	○	—	—	—	—	○	—	—	○
(d)	○	—	○*1	—	—	—	—	○	—	—	—

*1 T、ST、C、FD不能使用。

■控制数据

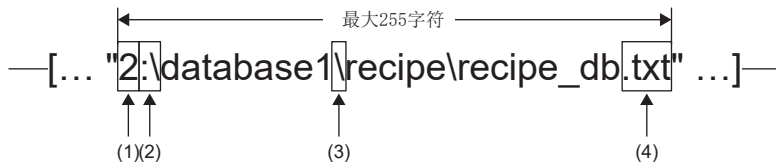
操作数：(s1)																
软元件	项目	内容	设置范围	设置方												
+0	使用用途设置区域	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">b15</td> <td style="width: 40%;">...</td> <td style="width: 10%;">b3</td> <td style="width: 10%;">b2</td> <td style="width: 10%;">b1</td> <td style="width: 15%;">b0</td> </tr> <tr> <td style="text-align: center;">0</td> <td></td> <td style="text-align: center;">(2)</td> <td style="text-align: center;">(1)</td> <td></td> <td style="text-align: center;">0</td> </tr> </table> <p>(1) 传送完成文件删除设置(位2) 指定传送后传送完成的文件的处理。 • 0: 不删除 • 1: 删除</p> <p>(2) 文件传送时的临时文件创建设置(位3) 指定是否创建文件传送时的临时文件(包括扩展名的传送源文件名.TMP)。 通过设置为“0: 创建”，即使文件传送中(文件改写中)发生了电缆断开或电源断开等，也可避免传送目标文件变为不正确的文件状态。 • 0: 创建 • 1: 不创建</p>	b15	...	b3	b2	b1	b0	0		(2)	(1)		0	如左所示	用户
b15	...	b3	b2	b1	b0											
0		(2)	(1)		0											
+1	完成状态	指令完成时存储完成状态。 • 0000H: 正常完成 • 0000H以外: 异常完成(错误代码)	—	系统												
+2	处理文件总数	存储本指令处理的文件的总数。	—	系统												
+3	处理完成文件数	存储处理完成的文件数。	—	系统												

功能

- 将(s2)中指定的控制器的文件发送至(s3)中指定的FTP服务器的文件夹路径。执行本指令时，连接“模块参数”的“FTP客户端设置”中指定的FTP服务器，文件发送后断开与FTP服务器的连接。关于“FTP客户端设置”的详细内容，请参阅下述手册。

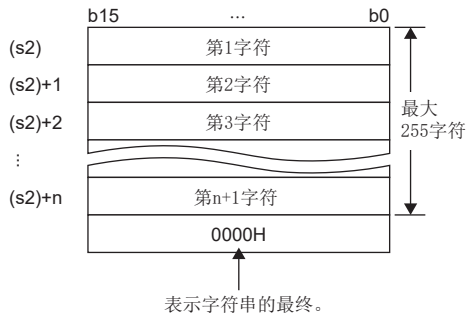
☞所使用的控制器的用户手册

- 本指令处理的文件总数被存储到(s1)+2的“处理文件总数”中，处理完成的文件数被存储到(s1)+3的“处理完成文件数”中。
- 在(s2)中，以Unicode字符串指定存储控制器传送源的驱动器No. 2~4、No. 6和文件的文件夹路径、文件名(包括点号、扩展名)不超过64字符的文件。文件路径的长度最大为255字符。文件名除外的路径长度，不包括文件分割符，最大为246字符。驱动器、文件夹路径、文件名的分割符使用半角的“\”或“/”。*1



- (1) 可指定的驱动器No. 为2~4、6。(☞所使用的控制器的用户手册)
- (2) 驱动器No. 的分割符使用“: \”或“:/”。
- (3) 文件夹路径、文件的分割符使用半角的“\”或“/”。
- (4) 指定的文件名包括扩展名。

*1 FTP客户端的路径中不能指定相对路径。



- (s2)中指定的文件名或扩展名中，不能使用通配符指定(*、?)。

符号	内容
*	从指定了“*”的位置开始以任意字符串(也包括无)的所有文件为对象。
?	指定了“?”的位置以任意字符(不包括无)的所有文件为对象。(可以使用多个“?”。)

不能对点号进行通配符指定。

下述指定方法将变为错误。

- 文件名(点号为止)或扩展名中使用了2个或其以上的“*”。(例：“*abc*.txt”)
- 文件名(点号为止)或扩展名中同时存在“*”及“?”。(例：“*ab?.txt”)

通配符指定中可传送文件数取决于指定的各文件的文件名容量的合计。可传送文件数以及各文件名容量满足下述公式的情况下，可以传送。不满足下述公式的情况下如果执行文件传送功能指令将变为异常完成。

$$(F_i + NM) + 1 < 65536 \text{ [字节]}$$

N: 符合通配符指定的文件总数
 F_i: 符合通配符指定的文件名的合计容量
 M: 指定信息容量(固定值: 6字节)

- 本指令执行过程中由于中途发送1个文件而发生错误的情况下，在错误发生时刻将中止传送，剩余的文件传送将不能执行。
- 不能传送的文件，即使符合通配符指定的条件也不为传送的对象。
- 包含文件名、扩展名的文件路径超过了255字符的情况下，即使符合通配符指定条件也不成为传送的对象。

- 在(s3)中，以Unicode字符串指定FTP服务器的传送目标的文件夹路径。通过FTP服务器的主目录开始的相对路径进行指定文件夹路径。文件夹路径的分割符中使用半角的“\”或“/”。*1文件夹路径的长度最大为255字符。但是，指定应避免文件夹路径(包括末尾的分割符)与(s2)的文件名部分的合计超过传送目标FTP服务器支持的最大路径长度。末尾的分割符可以省略，但省略的情况下，将被视为末尾处设置了“\”。此外，不存在的文件夹路径的情况下，执行指令时将由系统自动创建进行文件传送。*2



- (1) 文件夹路径的分隔符使用半角的“\”或“/”。*3
- (2) 末尾的分隔符可以省略。

*2 无法指定FTP服务器的驱动器。

*3 根据FTP服务器，有可能无法以“\”作为分隔符使用。

- (s3)中只指定了NULL或软元件中只指定了“0000H”的情况下，将访问FTP服务器的主目录以下。详细内容请参阅FTP服务器的规格。

• 传送目标中存在同名的文件的情况下，同名的文件将被覆盖。

• 每个文件的可发送文件容量最大为4G字节。

• 本指令的执行及正常/异常完成，可通过设置数据中指定的完成软元件(d)、完成时的状态显示软元件(d)+1进行确认。

• 完成软元件(d)

在本指令完成扫描的END处理中变为ON，在下一个END处理中变为OFF。

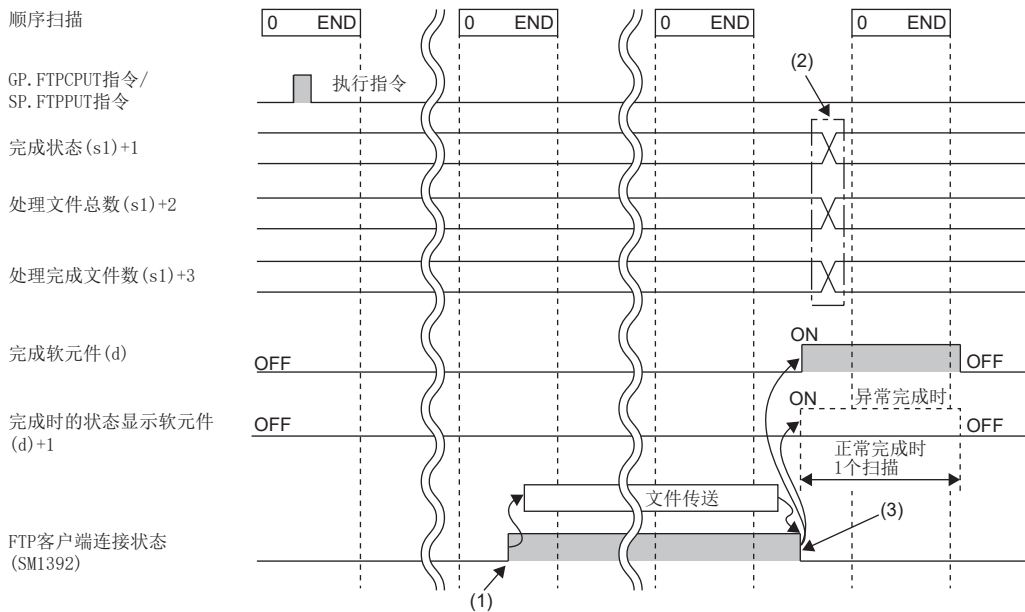
• 完成时的状态显示软元件(d)+1

根据本指令完成时的状态而进行ON/OFF。

正常完成时：保持OFF状态不变。

异常完成时：在本指令完成扫描的END处理中变为ON，在下一个END处理中变为OFF。异常完成时，在(s1)+1的完成状态中存储错误代码。

• 本指令的执行时机如下所示。



(1) 在FTP服务器连接后的END处理中变为ON。

(2) 指令完成时存储值。

(3) 全部文件的传送完成时OFF。

- FTP服务器连接中SM1392(FTP客户端连接状态)将ON, 如果断开则SM1392将OFF。
- 执行本指令的过程中, SM753(文件访问中)变为ON。SM753为ON时, 不能执行本指令。如果执行, 将变为无处理。
- 本指令或GP. FTPCGET指令/SP. FTPGET指令的执行过程中执行了本指令的情况下, 在先执行的指令完成之前, 后执行的指令将被忽略而无法执行。本指令被忽略的情况下, SM699(专用指令未执行标志)将ON。
- 传送目标中没有空余容量的情况下, 以及将(s1)+0的位3(文件传送时的临时文件创建设置)设置为0(创建)时传送目标中没有传送文件及临时文件(与传送文件相同的容量)的空余容量的情况下, 将异常完成。
- 文件传送执行过程中即使将控制器的状态置为RUN→STOP, 文件传送也将继续进行直至完成为止。

注意事项

- 文件传送过程中发生了电缆断线或控制器的电源断开·复位的情况下, 应根据需要删除FTP服务器内残留的无用文件(临时文件及不完整的文件)后, 再次执行。
- 对于传送完成后的传送源文件删除, 将(s1)+0的位2(传送完成文件删除设置)设置为1(删除)的情况下, 应注意下述几点。

项目	内容
传送控制器的文件的情况下	即使是控制器的动作所需文件也将从控制器中删除文件。不保证删除时的动作。
对传送文件名进行通配符指定后进行传送的情况下	必要文件有可能被意外删除。


- 通过文件口令功能对传送源文件进行了访问限制的情况下, 本指令将异常完成。但是, 将(s1)+0的位2(传送完成文件删除设置)设置为0(不删除), 文件口令的设置为“禁止写入”的情况下可以执行本指令。
- 将(s1)+0的位3(文件传送时的临时文件创建设置)设置为0(创建)时, 传送目标中将创建最多68个字符的临时文件(包括点号、扩展名的传送源文件名.TMP), 因此与传送目标文件夹路径匹配的路径长度不应超过传送目标FTP服务器支持的最大路径长度。
- 请勿使用不支持的字符。(s2)、(s3)中指定的文件名或文件夹名中包含有不支持的字符的情况下, 该字符将被转换处理为“_”。通配符指定时, 传送源控制器中存在的文件名中包含有不支持的字符的情况下, 该字符将被转换处理为“_”。因此, 文件传送中, 将进行与指定了转换为“_”后的文件名或文件夹名时相同的处理。

出错

错误代码(SD0)	内容
2802H	以以太网端口的设置执行了以太网功能指令时。
2820H	(s2)、(s3)中指定的软元件编号及其以后, 相应软元件/标签的分配范围中不存在0000H时。
3285H	(s2)中指定的软元件中存储的字符串超过了255字符时。
	(s2)中指定的文件名除外的路径(不包含文件的分割符)超过了246字符时。
	(s2)中指定的文件名(驱动器、文件夹路径除外)与(s3)中指定的字符串的合计超过了255字符时。
	(s2)中指定的驱动器No. 超出了可传送范围时。
	(s2)中指定的文件名超过了64字符时。
3296H	(s2)中未指定文件名时。
	(s2)中指定的文件名是不能传送的文件时。
	(s2)的驱动器No. 的分隔符为“: \”或“: /”以外时。
	(s2)中指定的文件名(点号之前)或扩展名中指定了2个或其以上的“*”时。
	(s2)中指定的文件名(点号之前)或扩展名中同时存在“*”及“?”时。
(s3)中包含有通配符指定字符(“*”、“?”)时。	
329BH	在未设置FTP客户端设置参数的状态下执行了本指令时。

异常完成时, 完成时的状态显示软元件(d)+1将变为ON, 完成状态(s1)+1中将存储错误代码。

关于完成状态(s1)+1中存储的错误代码, 请参阅下述手册。

所使用的控制器的用户手册

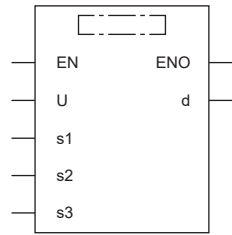
FTP客户端文件获取

GP. FTPCGET、SP. FTPGET

将(s2)中指定的FTP服务器文件获取到(s3)中指定的控制器的文件夹路径中。

梯形图	ST
	<pre>ENO:=GP_FTPCGET(EN, U, s1, s2, s3, d); ENO:=SP_FTPGET(EN, U, s1, s2, s3, d);</pre>

FBD/LD



■执行条件

指令	执行条件
GP. FTPCGET SP. FTPGET	

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)	
(U)	GP. FTPCGET	对象的起始输入输出编号 (指定起始输入输出编号的高3位)	3E0H	无符号BIN16位	ANY16
	SP. FTPGET	虚拟	—	字符串	ANYSTRING_SINGLE
(s1)	GP. FTPCGET	存储控制数据的起始软元件	参阅控制数据	软元件	ANY16
	SP. FTPGET			字	ANY16_ARRAY (要素数: 4)
(s2)	FTP服务器中存储的文件名(传送源)*1	—	Unicode字符串*2	ANYSTRING_DOUBLE	
(s3)	控制器的文件夹路径(传送目标)*1	—	Unicode字符串*2	ANYSTRING_DOUBLE	
(d)	指令完成时通过1个扫描变为ON的软元件 异常完成时(d)+1也将变为ON。	—	位	ANYBIT_ARRAY (要素数: 2)	
EN	执行条件	—	位	BOOL	
ENO	执行结果	—	位	BOOL	

*1 是Unicode字符串或存储了Unicode字符串的起始软元件。

*2 数据类型为Unicode字符串, 但只能使用半角英文数字、符号、假名字符、全角字符(移位JIS代码)。不支持的字符将被处理为“_”。

■可用软元件

操作数	位		字				双字		间接指定	常数			其它 (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)	—	—	○	—	—	—	—	○	—	—	○	—	
(s3)	—	—	○	—	—	—	—	○	—	—	○	—	
(d)	○	—	○*1	—	—	—	—	○	—	—	—	—	

*1 T、ST、C、FD不能使用。

■控制数据

操作数：(s1)								
软元件	项目	内容	设置范围	设置方				
+0	使用用途设置区域	b15 ... b3 b2 b1 b0 <table border="1" style="margin-left: 40px;"> <tr> <td style="width: 100px;">0</td> <td style="width: 20px;">(2)</td> <td style="width: 20px;">(1)</td> <td style="width: 20px;">0</td> </tr> </table> <p>(1) 传送完成文件删除设置(位2) 指定传送后传送完成的文件的处理。 • 0: 不删除 • 1: 删除</p> <p>(2) 文件传送时的临时文件创建设置(位3) 指定是否创建文件传送时的临时文件(包括扩展名的传送源文件名.TMP)。 通过设置为“0: 创建”，即使文件传送中(文件改写中)发生了电缆断开或电源断开等，也可避免传送目标文件变为不正确的文件状态。 • 0: 创建 • 1: 不创建</p>	0	(2)	(1)	0	如左所示	用户
0	(2)	(1)	0					
+1	完成状态	指令完成时存储完成状态。 • 0000H: 正常完成 • 0000H以外: 异常完成(错误代码)	—	系统				
+2	处理文件总数	存储本指令处理的文件的总数。	—	系统				
+3	处理完成文件数	存储处理完成的文件数。	—	系统				

功能

- 将(s2)中指定的FTP服务器文件获取到(s3)中指定的控制器的文件夹路径中。执行本指令时，连接“模块参数”的“FTP客户端设置”中指定的FTP服务器，文件获取后断开与FTP服务器的连接。关于“FTP客户端设置”的详细内容，请参阅下述手册。

📖 所使用的控制器的用户手册

- 本指令处理的文件总数被存储到(s1)+2的“处理文件总数”中，处理完成的文件数被存储到(s1)+3的“处理完成文件数”中。
- 在(s2)中，以Unicode字符串指定存储了FTP服务器传送源文件的文件夹路径、文件名(包括点号、扩展名)不超过64字符的文件。文件路径的长度最大为255字符。通过FTP服务器的主目录开始的相对路径指定文件路径。文件夹路径、文件名的分隔符中使用半角的“\”或“/”。*1*2

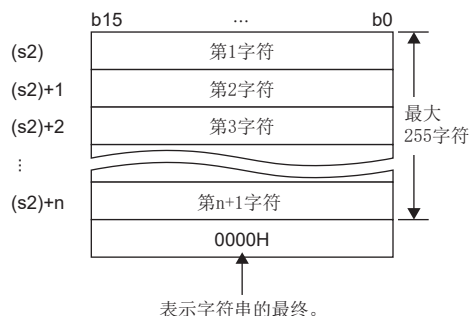


(1) 文件夹路径、文件的分隔符使用半角的“\”或“/”。*2

(2) 指定的文件名包括扩展名。

*1 无法指定FTP服务器的驱动器。

*2 根据FTP服务器，有可能无法以“\”作为分隔符使用。



- (s2)中指定的文件名或扩展名中，不能使用通配符指定(*、?)。

符号	内容
*	从指定了“*”的位置开始以任意字符串(也包括无)的所有文件为对象。
?	指定了“?”的位置以任意字符(不包括无)的所有文件为对象。(可以使用多个“?”。)

不能对点号进行通配符指定。

下述指定方法将变为错误。

- 文件名(点号为止)或扩展名中使用了2个或以上的“*”。(例：“*abc*.txt”)
- 文件名(点号为止)或扩展名中同时存在“*”及“?”。(例：“*ab?.txt”)

不能进行子文件夹内的文件传送。

指定了通配符的获取对象文件根据FTP服务器的环境而有所不同。请参阅FTP服务器的动作规格。

通配符指定中可传送文件数取决于指定的各文件的文件名容量的合计。可传送文件数以及各文件名容量满足下述公式的情况下，可以传送。不满足下述公式的情况下如果执行文件传送功能用指令将变为异常完成。

$$(Fi + NM) + 1 < 65536 \text{ [字节]}$$

N: 符合通配符指定的文件总数
 Fi: 符合通配符指定的文件名的合计容量
 M: 指定信息容量(固定值: 6字节)

- 本指令执行过程中由于中途获取1个文件而发生错误的情况下，在错误发生时刻将中止传送，剩余的文件传送将不能执行。
- 不能传送的文件，即使符合通配符指定的条件也不为传送的对象。
- 包含文件名、扩展名的文件路径超过了255字符的情况下，即使符合通配符指定条件也不成为传送的对象。
- (s2)中只指定了文件名的情况下，将访问FTP服务器的主目录以下。仅指定文件名的情况下，应通过分隔符进行指定。详细内容请参阅FTP服务器的规格。
- 在(s3)中，以Unicode字符串指定控制器的传送目标的文件夹路径。文件夹路径的分隔符中使用半角的“\”或“/”。文件夹路径的长度最大为247字符(省略末尾的分隔符的情况下，最大为246字符)。但是，指定时应避免文件夹路径(包含末尾的分隔符)与(s2)的文件名部分的合计超过控制器支持的最大路径长度(255字符)。末尾的分隔符可以省略，但省略的情况下，将被视为末尾处设置了“\”。此外，不存在的文件夹路径的情况下，执行指令时将由系统自动创建进行文件传送。*3

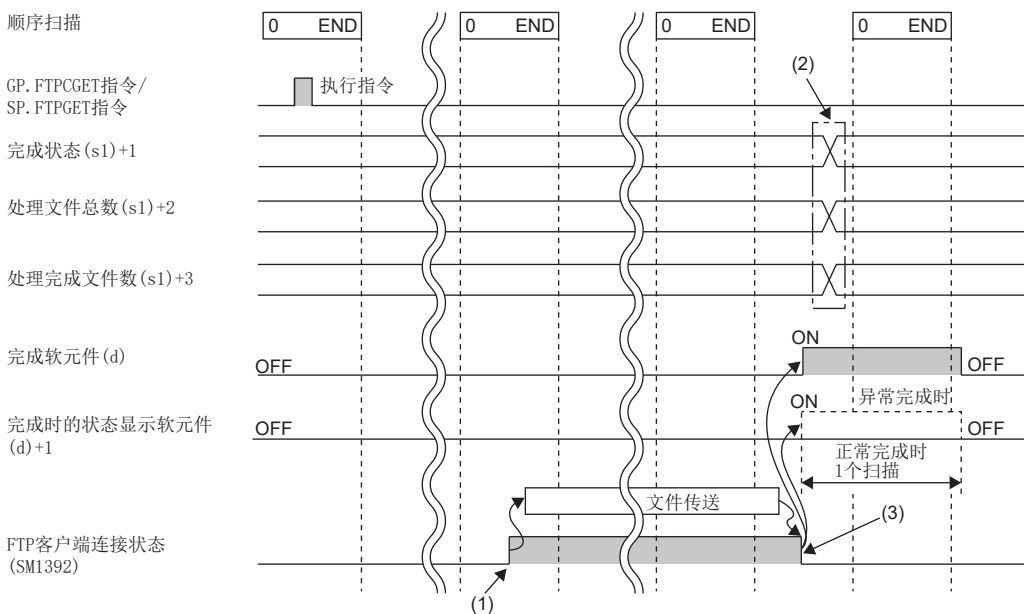
-[... "2:\database1\recipe\" ...]-

- (1) 可指定的驱动器No. 为2、4。()所使用的控制器的用户手册
- (2) 驱动器No. 的分隔符使用“: \”或“: /”。
- (3) 文件夹路径的分隔符使用半角的“\”或“/”。
- (4) 末尾的分隔符可以省略。

*3 FTP客户端的路径中不能指定相对路径。

- 传送目标中存在同名的文件的情况下，同名的文件将被覆盖。
 - 每个文件的可获取的文件容量最大为4G字节。
 - 本指令的执行及正常/异常完成，可通过设置数据中指定的完成软元件(d)、完成时的状态显示软元件(d)+1进行确认。
 - 完成软元件(d)
- 在本指令完成扫描的END处理中变为ON，在下一个END处理中变为OFF。
- 完成时的状态显示软元件(d)+1
- 根据本指令完成时的状态而进行ON/OFF。
- 正常完成时：保持OFF状态不变。
- 异常完成时：在本指令完成扫描的END处理中变为ON，在下一个END处理中变为OFF。异常完成时，在(s1)+1的完成状态中存储错误代码。

• 本指令的执行时机如下所示。



- (1) 在FTP服务器连接后的END处理中变为ON。
- (2) 指令完成时存储值。
- (3) 全部文件的传送完成时OFF。

- FTP服务器连接中SM1392(FTP客户端连接状态)将ON, 如果断开则SM1392将OFF。
- 执行本指令的过程中, SM753(文件访问中)变为ON。SM753为ON时, 不能执行本指令。如果执行, 将变为无处理。
- GP. FTPCPUT指令/SP. FTPPUT指令、或本指令的执行过程中执行本指令的情况下, 在先执行的指令完成之前, 后执行的指令将被忽略而无法执行。本指令被忽略的情况下, SM699(专用指令未执行标志)将ON。
- 传送目标中没有空余容量的情况下, 以及将(s1)+0的位3(文件传送时的临时文件创建设置)设置为0(创建)时传送目标中没有传送文件及临时文件(与传送文件相同的容量)的空余容量的情况下, 将异常完成。
- 文件传送执行过程中即使将控制器的状态置为RUN→STOP, 文件传送也将继续进行直至完成为止。

注意事项

- 文件传送过程中发生了电缆断线或控制器的电源断开・复位的情况下, 应再次执行。发生了电缆断线或电源断开・复位的情况下, 按下述方式删除无用文件。

无用文件	删除动作
临时文件(包括扩展名的传送源文件名.TMP)。	FTP客户端内残留的临时文件在下次对同一文件夹开始文件传送时将被自动删除。
不完整的传送对象文件	FTP客户端内残留的不完整的传送对象文件在下次对同一文件进行文件传送时将被覆盖。


- 对于传送源文件, 在传送目标中存在通过文件口令功能进行了访问限制的同名文件的情况下, 本指令将异常完成。
- 将(s1)+0的位3(文件传送时的临时文件创建设置)设置为0(创建)的情况下, 应将控制器的文件夹路径长度设置为最大243字符(省略了末尾分割符的情况下, 最大为242字符)。此外, 传送目标中将创建最多68个字符的临时文件(包括点号、扩展名的传送源文件名.TMP), 因此与传送目标文件夹路径匹配的路径长度不应超过控制器支持的最大路径长度(255字符)。
- 请勿使用不支持的字符。(s2)、(s3)中指定的文件名或文件夹名中包含有不支持的字符的情况下, 该字符将被转换处理为“_”。因此, 文件传送中, 将进行与指定了转换为“_”后的文件名或文件夹名时相同的处理。此外, 通配符指定时, 传送源FTP服务器中存在的文件名中包含有不支持的字符的情况下, 将无法正确获取文件。(关于如何处理不支持的字符, 请参阅FTP服务器的规格)

出错

错误代码(SD0)	内容
2802H	以以太网端口的设置执行了以太网功能指令时。
2820H	(s2)、(s3)中指定的软元件编号及其以后, 相应软元件/标签的分配范围中不存在0000H时。
3285H	(s2)中指定的软元件中存储的字符串超过了255字符时。
	(s2)中指定的文件名除外的路径(不包含文件的分割符)超过了246字符时。
	(s3)中指定的字符串与(s2)中指定的字符串的文件名部分的合计超过了255字符时。
	(s3)中指定的驱动器No. 超出了可传送范围时。
	(s2)中指定的文件名超过了64字符时。
3296H	(s2)中未指定文件名时。
	(s2)中指定的文件名是不能传送的文件时。
	(s3)的驱动器No. 的分割符为“: \”或“:/”以外时。
	(s2)中指定的文件名(点号之前)或扩展名中指定了2个或其以上的“*”时。
	(s2)中指定的文件名(点号之前)或扩展名中同时存在“*”及“?”时。
(s3)中包含有通配符指定字符(“*”、“?”)时。	
329BH	在未设置FTP客户端设置参数的状态下执行了本指令时。

异常完成时, 完成时的状态显示软元件(d)+1将变为ON, 完成状态(s1)+1中将存储错误代码。

关于完成状态(s1)+1中存储的错误代码, 请参阅下述手册。

 所使用的控制器的用户手册

30 外部设备通信指令

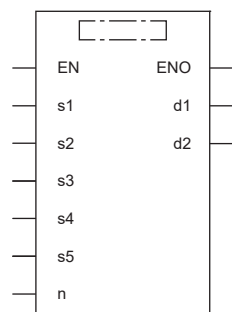
30.1 特殊适配器用指令

ADPRW

通过主站对应的功能代码，可以与从站进行通信(数据的读取/写入)。

梯形图	ST
	<pre>ENO:=ADPRW(EN, s1, s2, s3, s4, s5, n, d1, d2);</pre>

FBD/LD



设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s1)	从站站号	0~F7H	带符号BIN16位	ANY16
(s2)	功能代码 (☞ 1174页 功能代码和功能参数)	01H~06H、0FH、10H	带符号BIN16位	ANY16
(s3)	与功能代码对应的功能参数 (☞ 1174页 功能代码和功能参数)	0~FFFFH	带符号BIN16位	ANY16
(s4)	与功能代码对应的功能参数 (☞ 1174页 功能代码和功能参数)	1~2000	带符号BIN16位	ANY16
(s5)	与功能代码对应的功能参数 (☞ 1174页 功能代码和功能参数)	—	位/带符号BIN16位	ANY_ELEMENTARY
(d1)	与功能代码对应的功能参数 (☞ 1174页 功能代码和功能参数)	—	位/带符号BIN16位	ANY_ELEMENTARY
(d2)*1	输出通信执行状态的起始位软元件编号	—	位	ANYBIT_ARRAY (要素数: 3)
(n)	通信通道	1~2	无符号BIN16位	ANY16
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

*1 在起始处占用3点(d2)中指定的软元件。请注意不要与用于其他控制的软元件重复。

■可用软元件

操作数	位	字			双字		间接指定	常数			其他
		X、Y、M、B、F、SB、V、T、ST、LT、LST、C、LC、L、FX、FY、SM、J□\X□、J□\Y□、J□\B□、J□\SB□	T、ST、C、D、W、SW、FD、SD、R、ZR、RD	U□\G□、U3E□\G□、J□\W□、J□\SW□	Z	LT、LST、LC		LZ	K、H	E	
(s1)	—	○*1	○	○	—	—	○	○	—	—	—
(s2)	—	○*1	○	○	—	—	○	○	—	—	—
(s3)	—	○*1	○	○	—	—	○	○	—	—	—
(s4)	—	○*1	○	○	—	—	○	○	—	—	—
(s5)	○	○*1	○	○	—	—	○	○	—	—	—
(d1)	○	○*1	○	○	—	—	○	—	—	—	—
(d2)	○	○*1	—	—	—	—	—	—	—	—	—
(s5)	○*2	○	○	—	—	—	○	○	—	—	—

*1 不能使用T、ST、C。

*2 FX、FY不能使用。

功能

- 功能代码(s2)在从站站号(s1)上,将根据参数(s3)、(s4)、(s5)、(d1)执行动作。进行广播的情况下,请将从站站号(s1)设为0。(☞ 1174页 功能代码和功能参数)
- 通信执行状态(d2)依照ADPRW命令的通信执行中/正常结束/异常结束的各状态进行输出。(☞ 1175页 通信执行状态输出软元件)
- 使用通信通道(n)中指定的通道,执行ADPRW指令。

■关于常规执行以外的动作

在RUN中写入时,改写的程序文件与通信中有指令的程序文件是否相同,会决定动作的情况。

- 相同程序文件的改写:通过通信处理继续进行之后的动作。因此,MODBUS RTU通信中SM(SM8800、SM8810)将保持为1(通信中)。之后,如果通信处理中通信完成,则MODBUS RTU通信中SM将变为OFF,通信执行状态输出软元件的指令执行中为OFF,指令异常完成将变为ON。但是,读取指令时,读取的数据不存储到软元件中。在1个扫描中ADPRW指令的驱动触点变为OFF的情况下,即使在之后的扫描中变为ON,在前一次通信的通信处理结束之前也不会动作。
- 不同程序文件的改写:继续动作。

■功能代码和功能参数

通过各功能代码(s2)进行的(s3)、(s4)、(s5)、(d1)的参数分配如下所示。

(s2): 功能代码	(s3): MODBUS地址	(s4): 访问点数	(s5): 写入数据存储软元件起始 (d1): 读取数据存储软元件起始	
			对象软元件: ② (参阅下述对象软元件表)	占用点数
01H 线圈读取	MODBUS地址: 0000H~FFFFH	访问点数: 1~2000	读取数据存储软元件起始	
			对象软元件	字软元件 ① (参阅下述对象软元件表) 位软元件 ③ (参阅下述对象软元件表)
			占用点数	字软元件 $((s4)+15) \div 16 \text{点}^*1$ 位软元件 (s4)点
02H 输入读取	MODBUS地址: 0000H~FFFFH	访问点数: 1~2000	读取数据存储软元件起始	
			对象软元件	字软元件 ① (参阅下述对象软元件表) 位软元件 ③ (参阅下述对象软元件表)
			占用点数	字软元件 $((s4)+15) \div 16 \text{点}^*1$ 位软元件 (s4)点
03H 保持寄存器读取	MODBUS地址: 0000H~FFFFH	访问点数: 1~125	读取数据存储软元件起始	
			对象软元件	① (参阅下述对象软元件表)
			占用点数	(s4)点

(s2): 功能代码	(s3): MODBUS地址	(s4): 访问点数	(s5): 写入数据存储软元件起始	(d1): 读取数据存储软元件起始
	对象软元件: ② (参阅下述对象软元件表)			
04H 输入寄存器读取	MODBUS地址: 0000H~FFFFH	访问点数: 1~125	读取数据存储软元件起始	
			对象软元件	① (参阅下述对象软元件表)
			占用点数	(s4) 点
05H 线圈写入	MODBUS地址: 0000H~FFFFH	0 (固定)	写入数据存储软元件起始	
			对象软元件*2	字软元件 ② (参阅下述对象软元件表) 位软元件 ③ (参阅下述对象软元件表)
			占用点数	1点
06H 保持寄存器写入	MODBUS地址: 0000H~FFFFH	0 (固定)	写入数据存储软元件起始	
			对象软元件	② (参阅下述对象软元件表)
			占用点数	1点
0FH 多个点的线圈写入	MODBUS地址: 0000H~FFFFH	访问点数: 1~1968	写入数据存储软元件起始	
			对象软元件	字软元件 ② (参阅下述对象软元件表) 位软元件 ③ (参阅下述对象软元件表)
			占用点数	字软元件 $((s4)+15) \div 16$ 点*1 位软元件 (s4) 点
10H 多个点的保持寄存器写入	MODBUS地址: 0000H~FFFFH	访问点数: 1~123	写入数据存储软元件起始	
			对象软元件	② (参阅下述对象软元件表)
			占用点数	(s4) 点

*1 尾数被舍去。

*2 最低位为0时位OFF, 1时位ON。

▶对象软元件表

No.	对象软元件
①	T、ST、C、D、R、W、SW、SD、LT、LST、LC、Z、LZ、ZR、标签软元件
②	T、ST、C、D、R、W、SW、SD、LT、LST、LC、Z、LZ、ZR、标签软元件、K、H
③	X、Y、M、L、B、F、SB、SM、V、标签软元件

■通信执行状态输出软元件

通信执行状态输出软元件(d2)的根据各通信状态的动作时机与同时动作的特殊继电器如下所示。

操作数	动作的时机	同时动作的特殊继电器
(d2)	指令动作中为ON, 指令执行中以外为OFF	SM8800 (CH1)、SM8810 (CH2)
(d2)+1*1	指令正常完成时为ON, 通信开始时为OFF	—
(d2)+2*1	指令异常完成时为ON, 通信开始时为OFF	—

*1 (d2)+1在指令正常完成时变为ON, (d2)+2在指令异常完成时变为ON, 因此可以判断正常/异常。

注意事项

- 对于使用ADPRW指令的对象通道, 必须通过GX Works3进行MODBUS主站的设置。未设置的情况下, 即使执行ADPRW指令也不动作。运算错误(错误代码: 3600H)。
- 由于错误导致程序停止的情况下, 如果在通信状态输出软元件中指定了非锁存软元件, 软元件值将变为OFF。希望保留通信状态输出的情况下, 应指定锁存软元件。

30.2 串行数据传送2

RS2

经由安装在控制器上的通信适配器，通过无顺序通信进行数据的发送接收。

梯形图 	ST ENO:=RS2(EN, s, n1, n2, n3, d);
----------------	--

FBD/LD 	
-------------------	--

设置数据

■内容、范围、数据类型

操作数	内容	范围	数据类型	数据类型(标签)
(s)	发送数据的起始软元件	—	字	ANY16
(n1)	发送数据的点数	0~4096	无符号BIN16位	ANY16_U
(d)	保存接收数据的起始软元件	—	字	ANY16
(n2)	接收数据的点数	0~4096	无符号BIN16位	ANY16_U
(n3)	通信通道	K1、K2	无符号BIN16位	ANY16_U
EN	执行条件	—	位	BOOL
ENO	执行结果	—	位	BOOL

■可用软元件

操作数	位	字		双字		间接指定	常数			其他
	X、Y、M、B、F、SB、V、T、ST、LT、LST、C、LC、L、FX、FY、SM、J□\X□、J□\Y□、J□\B□、J□\SB□	T、ST、C、D、W、SW、FD、SD、R、ZR、RD	U□\G□、U3E□\G□、J□\W□、J□\SW□	Z	LT、LST、LC		LZ	K、H	E	
(s)	—	○*1	—	—	—	○	—	—	—	—
(n1)	○*2	○	○	○	—	○	○	—	—	—
(d)	—	○*1	—	—	—	○	—	—	—	—
(n2)	○*2	○	○	○	—	○	○	—	—	—
(n3)	—	—	—	—	—	—	○	—	—	—

*1 不能使用T、ST、C。

*2 FX、FY不能使用。

功能

经由安装在控制器上的通信适配器，通过无顺序通信进行数据的发送接收。用于指定从控制器发出的发送数据的起始元件和数据点数，以及保存接收数据的起始元件和可以接收的最大点数。

■关于常规执行以外的动作

在RUN中写入时，改写的程序文件与通信中有指令的程序文件是否相同，会决定动作的情况。

- 相同程序文件的改写：停止执行RS2指令。
- 不同程序文件的改写：继续动作。

注意事项

更改帧头、结束符、超时时间时，应在RS2指令驱动前(OFF中)进行。RS2指令驱动时有效。在RS2指令驱动中，请勿更改帧头、结束符、超时时间。

第6部分 通用FUN

本部分由下述章构成。

31 类型转换函数

32 单数值变量函数

33 算术运算函数

34 位移函数

35 位型布尔函数

36 选择函数

37 比较函数

38 字符串函数

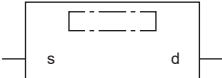
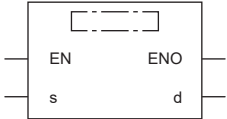
39 时间数据类型函数

31 类型转换函数

31.1 BOOL型→WORD型转换

BOOL_TO_WORD(_E)

将BOOL型数据转换为WORD型数据。

梯形图、FBD/LD	ST
[无EN/ENO] 	[无EN/ENO] d:=BOOL_TO_WORD(s);
[带EN/ENO] 	[带EN/ENO] d:=BOOL_TO_WORD_E(EN, ENO, s);

设置数据

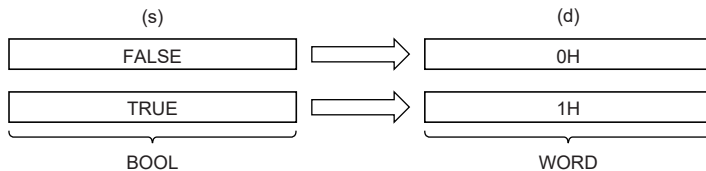
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	BOOL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	WORD

功能

■运算处理

- 将(s)中输入的BOOL型的数据, 转换为WORD型的数据后从(d)输出。
- 输入值为FALSE的情况下, 以WORD型的数据值输出0H。
- 输入值为TRUE的情况下, 以WORD型的数据值输出1H。



- 至(s)的输入值为BOOL型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

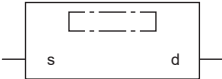
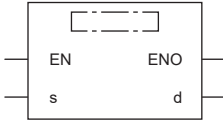
出错

没有运算错误。

31.2 BOOL型→DWORD型转换

BOOL_TO_DWORD(_E)

将BOOL型数据转换为DWORD型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=BOOL_TO_DWORD(s); [带EN/ENO] d:=BOOL_TO_DWORD_E(EN, ENO, s);

设置数据

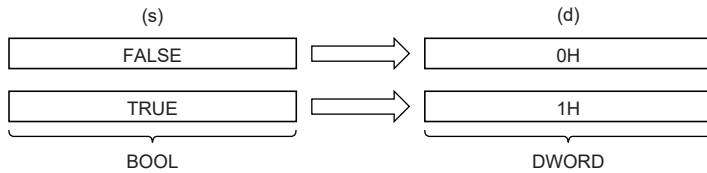
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	BOOL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	DWORD

功能

■运算处理

- 将(s)中输入的BOOL型的数据, 转换为DWORD型的数据后从(d)输出。
- 输入值为FALSE的情况下, 以DWORD型的数据值输出0H。
- 输入值为TRUE的情况下, 以DWORD型的数据值输出1H。



- 至(s)的输入值为BOOL型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

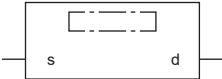
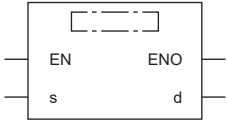
出错

没有运算错误。

31.3 BOOL型→INT型转换

BOOL_TO_INT(_E)

将BOOL型数据转换为INT型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=BOOL_TO_INT(s); [带EN/ENO] d:=BOOL_TO_INT_E(EN, ENO, s);

设置数据

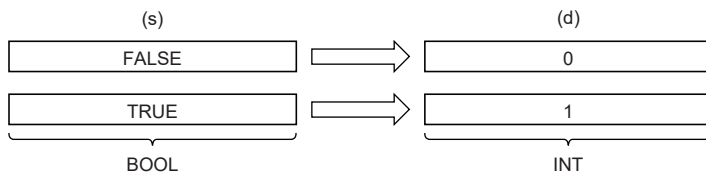
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	BOOL
ENO	输出状态(TRUE: 正常执行, FALSE: 异常或停止)	输出变量	BOOL
d	输出	输出变量	INT

功能

■运算处理

- 将(s)中输入的BOOL型的数据, 转换为INT型的数据后从(d)输出。
- 输入值为FALSE的情况下, 以INT型的数据值输出0。
- 输入值为TRUE的情况下, 以INT型的数据值输出1。



- 至(s)的输入值为BOOL型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


出错

没有运算错误。

31.4 BOOL型→DINT型转换

BOOL_TO_DINT(_E)

将BOOL型数据转换为DINT型数据。

梯形图、FBD/LD	ST
[无EN/ENO] 	[无EN/ENO] <code>d:=BOOL_TO_DINT(s);</code> [带EN/ENO] <code>d:=BOOL_TO_DINT_E(EN, ENO, s);</code>

设置数据

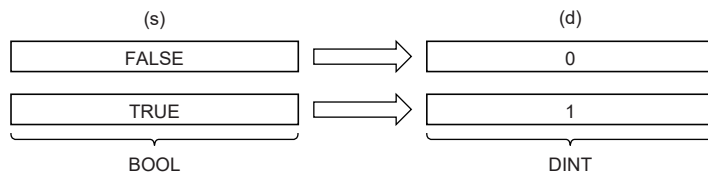
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	BOOL
ENO	输出状态(TRUE: 正常执行, FALSE: 异常或停止)	输出变量	BOOL
d	输出	输出变量	DINT

功能

■运算处理

- 将(s)中输入的BOOL型的数据, 转换为DINT型的数据后从(d)输出。
- 输入值为FALSE的情况下, 以DINT型的数据值输出0。
- 输入值为TRUE的情况下, 以DINT型的数据值输出1。



- 至(s)的输入值为BOOL型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

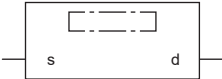
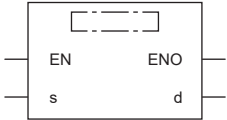
出错

没有运算错误。

31.5 BOOL型→TIME型转换

BOOL_TO_TIME(_E)

将BOOL型数据转换为TIME型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=BOOL_TO_TIME(s); [带EN/ENO] d:=BOOL_TO_TIME_E(EN, ENO, s);

设置数据

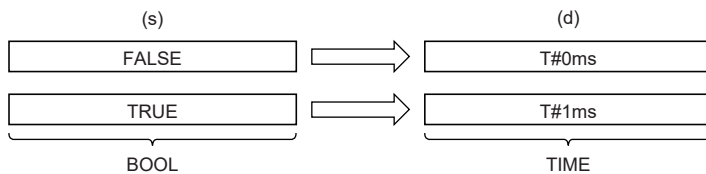
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	BOOL
ENO	输出状态(TRUE: 正常执行, FALSE: 异常或停止)	输出变量	BOOL
d	输出	输出变量	TIME

功能

■运算处理

- 将(s)中输入的BOOL型的数据, 转换为TIME型的数据后从(d)输出。
- 输入值为FALSE的情况下, 以TIME型的数据值输出0。
- 输入值为TRUE的情况下, 以TIME型的数据值输出1。



- 至(s)的输入值为BOOL型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

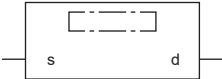
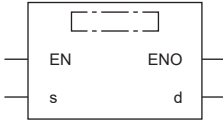
出错

没有运算错误。

31.6 BOOL型→STRING型转换

BOOL_TO_STRING(_E)

将BOOL型数据转换为STRING型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=BOOL_TO_STRING(s); [带EN/ENO] d:=BOOL_TO_STRING_E(EN, ENO, s);

设置数据

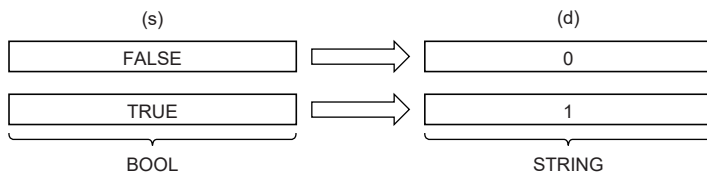
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	BOOL
ENO	输出状态(TRUE: 正常执行, FALSE: 异常或停止)	输出变量	BOOL
d	输出	输出变量	STRING

功能

■运算处理

- 将(s)中输入的BOOL型的数据, 转换为STRING型的数据后从(d)输出。
- 输入值为FALSE的情况下, 以STRING型的数据值输出0。
- 输入值为TRUE的情况下, 以STRING型的数据值输出1。



- 至(s)的输入值为BOOL型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

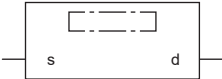
出错

没有运算错误。

31.7 WORD型→BOOL型转换

WORD_TO_BOOL(_E)

将WORD型数据转换为BOOL型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=WORD_TO_BOOL(s); [带EN/ENO] d:=WORD_TO_BOOL_E(EN, ENO, s);

设置数据

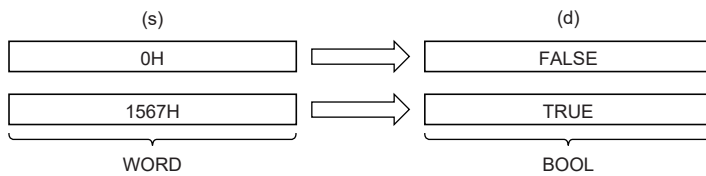
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	WORD
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	BOOL

功能

■运算处理

- 将(s)中输入的WORD型的数据, 转换为BOOL型的数据后从(d)输出。
- 输入值为0H的情况下, 输出FALSE。
- 输入值为0H以外的情况下, 输出TRUE。



- 至(s)的输入值为WORD型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

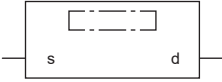
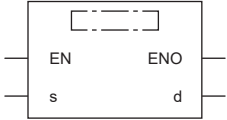
出错

没有运算错误。

31.8 WORD型→DWORD型转换

WORD_TO_DWORD(_E)

将WORD型数据转换为DWORD型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=WORD_TO_DWORD(s); [带EN/ENO] d:=WORD_TO_DWORD_E(EN, ENO, s);

设置数据

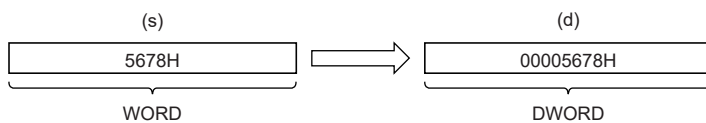
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	WORD
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	DWORD

功能

■运算处理

- 将(s)中输入的WORD型的数据, 转换为DWORD型的数据后从(d)输出。
- 数据转换后, 高位16位将变为0。



- 至(s)的输入值为WORD型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

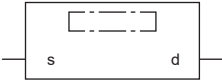
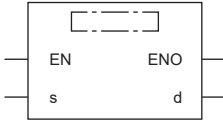
出错

没有运算错误。

31.9 WORD型→INT型转换

WORD_TO_INT(_E)

将WORD型数据转换为INT型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=WORD_TO_INT(s); [带EN/ENO] d:=WORD_TO_INT_E(EN, ENO, s);

设置数据

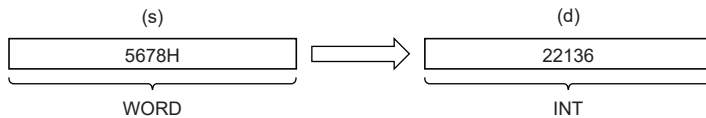
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	WORD
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	INT

功能

■运算处理

- 将(s)中输入的WORD型的数据, 转换为INT型的数据后从(d)输出。



- 至(s)的输入值为WORD型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


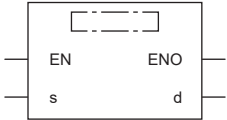
出错

没有运算错误。

31.10 WORD型→DINT型转换

WORD_TO_DINT(_E)

将WORD型数据转换为DINT型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=WORD_TO_DINT(s); [带EN/ENO] d:=WORD_TO_DINT_E(EN, ENO, s);

设置数据

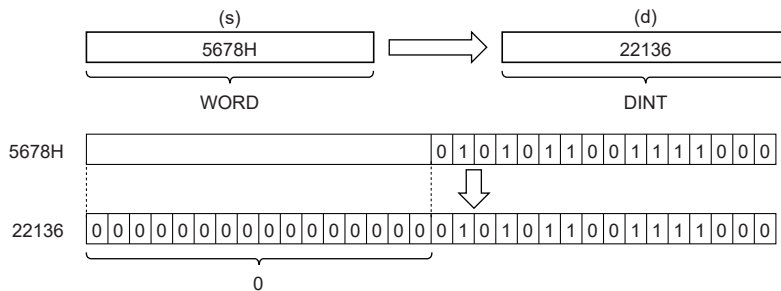
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	WORD
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	DINT

功能

■运算处理

- 将(s)中输入的WORD型的数据, 转换为DINT型的数据后从(d)输出。
- 数据转换后, 高位16位将变为0。



- 至(s)的输入值为WORD型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


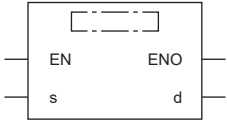
出错

没有运算错误。

31.11 WORD型→TIME型转换

WORD_TO_TIME(_E)

将WORD型数据转换为TIME型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=WORD_TO_TIME(s); [带EN/ENO] d:=WORD_TO_TIME_E(EN, ENO, s);

设置数据

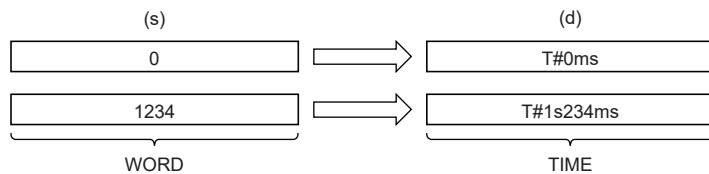
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	WORD
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	TIME

功能

■运算处理

- 将(s)中输入的WORD型的数据, 转换为TIME型的数据后从(d)输出。



- 至(s)的输入值为WORD型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


出错

没有运算错误。

31.12 WORD型→STRING型转换

WORD_TO_STRING(_E)

将WORD型数据转换为STRING型数据。

梯形图、FBD/LD	ST
[无EN/ENO] 	[无EN/ENO] <code>d:=WORD_TO_STRING(s);</code> [带EN/ENO] <code>d:=WORD_TO_STRING_E(EN, ENO, s);</code>

设置数据

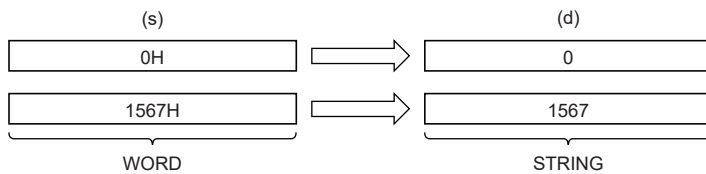
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	WORD
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	STRING(4)

功能

■运算处理

- 将(s)中输入的WORD型的数据, 转换为STRING型的数据后从(d)输出。



- 至(s)的输入值为WORD型的数据值。
- SM701(输出字数转换)为OFF的情况下, 字符串的最后将存储00H。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

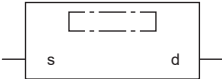
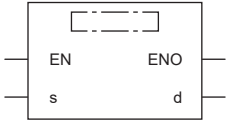
出错

没有运算错误。

31.13 DWORD型→BOOL型转换

DWORD_TO_BOOL(_E)

将DWORD型数据转换为BOOL型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=DWORD_TO_BOOL(s); [带EN/ENO] d:=DWORD_TO_BOOL_E(EN, ENO, s);

设置数据

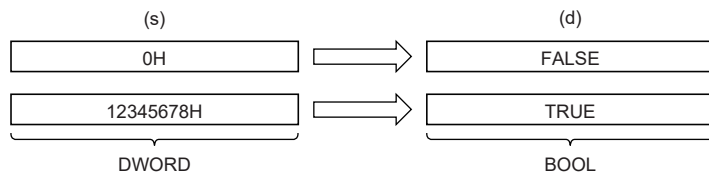
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	DWORD
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	BOOL

功能

■运算处理

- 将(s)中输入的DWORD型的数据, 转换为BOOL型的数据后从(d)输出。
- 输入值为0H的情况下, 输出FALSE。
- 输入值为0H以外的情况下, 输出TRUE。



- 至(s)的输入值为DWORD型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

出错

没有运算错误。

31.14 DWORD型→WORD型转换

DWORD_TO_WORD(_E)

将DWORD型数据转换为WORD型数据。

梯形图、FBD/LD	ST
[无EN/ENO] 	[无EN/ENO] $d := \text{DWORD_TO_WORD}(s);$ [带EN/ENO] $d := \text{DWORD_TO_WORD_E}(EN, ENO, s);$

设置数据

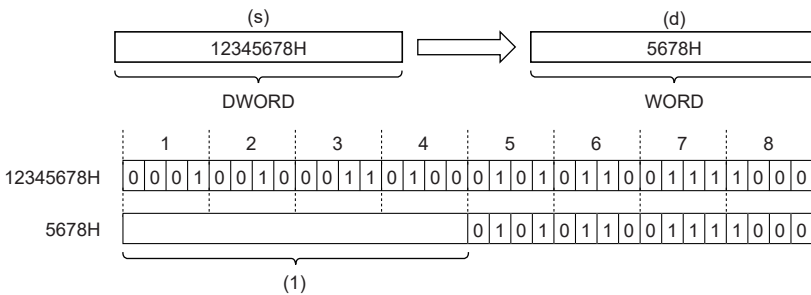
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	DWORD
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	WORD

功能

■运算处理

- 将(s)中输入的DWORD型的数据, 转换为WORD型的数据后从(d)输出。
- 输入值的高位16位的信息将被删除。(下述(1))



- 至(s)的输入值为DWORD型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

要点

执行DWORD_TO_WORD(_E)时, 从(s)输入的DWORD型数据值的高位16位的信息将被删除。

出错

没有运算错误。

31.15 DWORD型→INT型转换

DWORD_TO_INT(_E)

将DWORD型数据转换为INT型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=DWORD_TO_INT(s); [带EN/ENO] d:=DWORD_TO_INT_E(EN, ENO, s);

设置数据

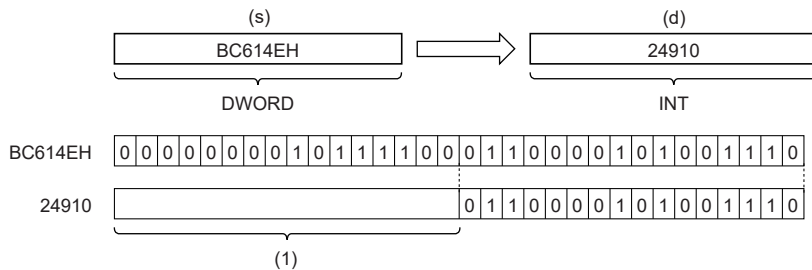
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	DWORD
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	INT

功能

■运算处理

- 将(s)中输入的DWORD型的数据, 转换为INT型的数据后从(d)输出。
- 输入值的高位16位的信息将被删除。(下述(1))



- 至(s)的输入值为DWORD型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

要点

执行DWORD_TO_INT(_E)时, 从(s)输入的DWORD型数据值的高位16位的信息将被删除。

出错

没有运算错误。

31.16 DWORD型→DINT型转换

DWORD_TO_DINT(_E)

将DWORD型数据转换为DINT型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=DWORD_TO_DINT(s); [带EN/ENO] d:=DWORD_TO_DINT_E(EN, ENO, s);

设置数据

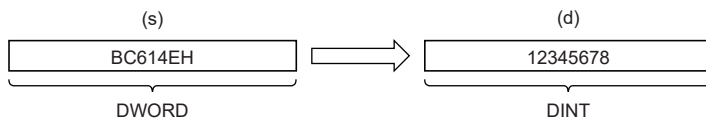
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	DWORD
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	DINT

功能

■运算处理

- 将(s)中输入的DWORD型的数据, 转换为DINT型的数据后从(d)输出。



- 至(s)的输入值为DWORD型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


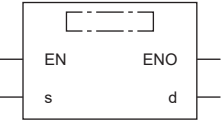
出错

没有运算错误。

31.17 DWORD型→TIME型转换

DWORD_TO_TIME(_E)

将DWORD型数据转换为TIME型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=DWORD_TO_TIME(s); [带EN/ENO] d:=DWORD_TO_TIME_E(EN, ENO, s);

设置数据

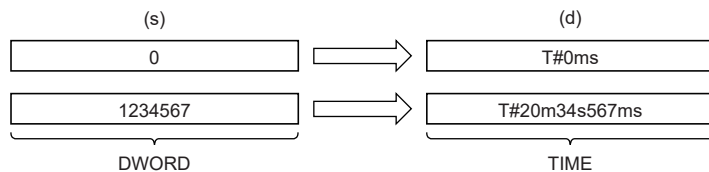
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	DWORD
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	TIME

功能

■运算处理

- 将(s)中输入的DWORD型的数据, 转换为TIME型的数据后从(d)输出。



- 至(s)的输入值为DWORD型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

出错

没有运算错误。

31.18 DWORD型→STRING型转换

DWORD_TO_STRING(_E)

将DWORD型数据转换为STRING型数据。

梯形图、FBD/LD	ST
[无EN/ENO] 	[无EN/ENO] <code>d:=DWORD_TO_STRING(s);</code> [带EN/ENO] <code>d:=DWORD_TO_STRING_E(EN, ENO, s);</code>

设置数据

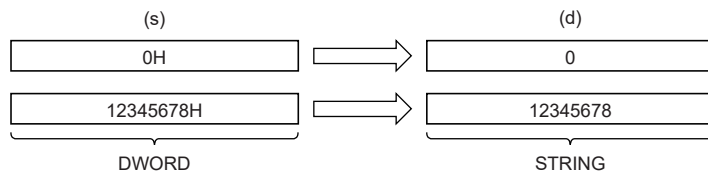
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	DWORD
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	STRING(8)

功能

■运算处理

- 将(s)中输入的DWORD型的数据, 转换为STRING型的数据后从(d)输出。



- 至(s)的输入值为DWORD型的数据值。
- SM701(输出字数转换)为OFF的情况下, 字符串的最后将存储00H。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


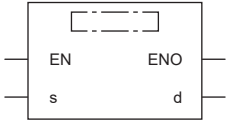
出错

没有运算错误。

31.19 INT型→BOOL型转换

INT_TO_BOOL(_E)

将INT型数据转换为BOOL型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=INT_TO_BOOL(s); [带EN/ENO] d:=INT_TO_BOOL_E(EN, ENO, s);

设置数据

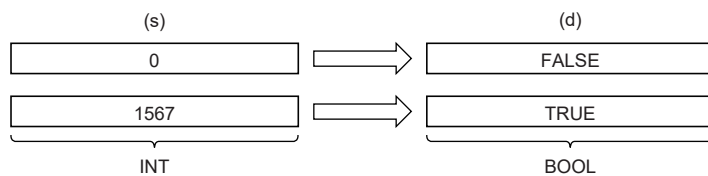
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	BOOL

功能

■运算处理

- 将(s)中输入的INT型的数据, 转换为BOOL型的数据后从(d)输出。
- 输入值为0的情况下, 输出FALSE。
- 输入值为0以外的情况下, 输出TRUE。



- 至(s)的输入值为INT型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

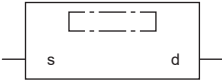
出错

没有运算错误。

31.20 INT型→WORD型转换

INT_TO_WORD(_E)

将INT型数据转换为WORD型数据。

梯形图、FBD/LD	ST
[无EN/ENO] 	[无EN/ENO] $d := \text{INT_TO_WORD}(s);$ [带EN/ENO] $d := \text{INT_TO_WORD_E}(\text{EN}, \text{ENO}, s);$

设置数据

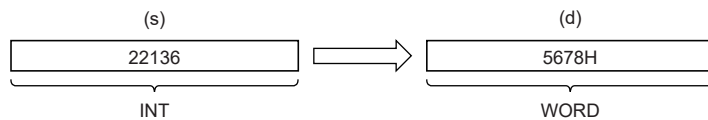
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	WORD

功能

■运算处理

- 将(s)中输入的INT型的数据, 转换为WORD型的数据后从(d)输出。



- 至(s)的输入值为INT型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

出错

没有运算错误。

31.21 INT型→DWORD型转换

INT_TO_DWORD(_E)

将INT型数据转换为DWORD型数据。

梯形图、FBD/LD	ST
<p>[无EN/ENO]</p>	<p>[无EN/ENO] d:=INT_TO_DWORD(s);</p> <p>[带EN/ENO] d:=INT_TO_DWORD_E(EN, ENO, s);</p>

设置数据

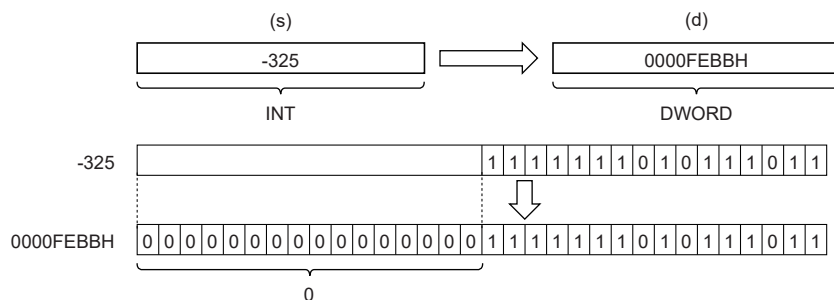
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	DWORD

功能

■运算处理

- 将(s)中输入的INT型的数据, 转换为DWORD型的数据后从(d)输出。
- 数据转换后, 高位16位将变为0。



- 至(s)的输入值为INT型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

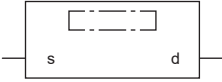
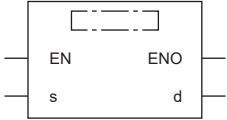
出错

没有运算错误。

31.22 INT型→DINT型转换

INT_TO_DINT(_E)

将INT型数据转换为DINT型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=INT_TO_DINT(s); [带EN/ENO] d:=INT_TO_DINT_E(EN, ENO, s);

设置数据

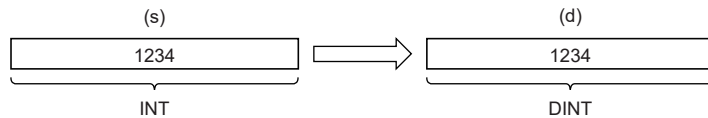
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	DINT

功能

■运算处理

- 将(s)中输入的INT型的数据, 转换为DINT型的数据后从(d)输出。



- 至(s)的输入值为INT型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

出错

没有运算错误。

31.23 INT型→BCD型转换

INT_TO_BCD(_E)

将INT型数据转换为BCD型数据。

梯形图、FBD/LD	ST
<p>[无EN/ENO]</p>	<p>[无EN/ENO] d:=INT_TO_BCD(s);</p> <p>[带EN/ENO] d:=INT_TO_BCD_E(EN, ENO, s);</p>

设置数据

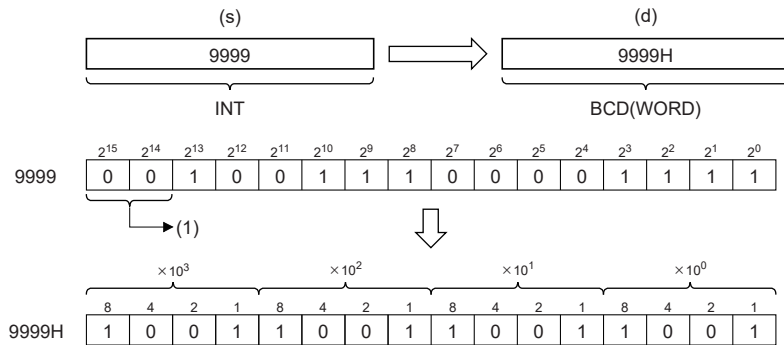
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	WORD

功能

■运算处理

- 将(s)中输入的INT型数据转换为BCD型(WORD型)数据后通过(d)进行输出。



(1) 必须设置为0。

- 至(s)的输入值是INT型的数据值且在0~9999的范围内。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


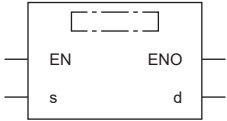
出错

错误代码(SD0)	内容
3281H	(s)的数据超出0~9999的范围时。

31.24 INT型→REAL型转换

INT_TO_REAL(_E)

将INT型数据转换为REAL型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=INT_TO_REAL(s); [带EN/ENO] d:=INT_TO_REAL_E(EN, ENO, s);

设置数据

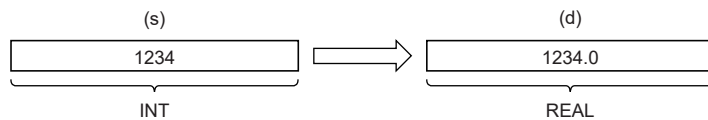
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	REAL

功能

■运算处理

- 将(s)中输入的INT型的数据, 转换为REAL型的数据后从(d)输出。



- 至(s)的输入值为INT型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

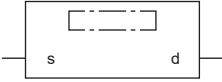
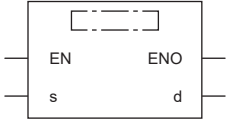
出错

没有运算错误。

31.25 INT型→LREAL型转换

INT_TO_LREAL(_E)

将INT型数据转换为LREAL型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=INT_TO_LREAL(s); [带EN/ENO] d:=INT_TO_LREAL_E(EN, ENO, s);

设置数据

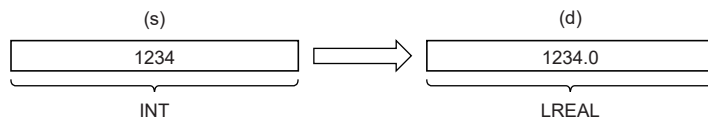
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	LREAL

功能

■运算处理

- 将(s)中输入的INT型的数据, 转换为LREAL型的数据后从(d)输出。



- 至(s)的输入值为INT型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

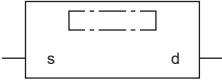
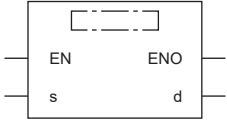
出错

没有运算错误。

31.26 INT型→TIME型转换

INT_TO_TIME(_E)

将INT型数据转换为TIME型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=INT_TO_TIME(s); [带EN/ENO] d:=INT_TO_TIME_E(EN, ENO, s);

设置数据

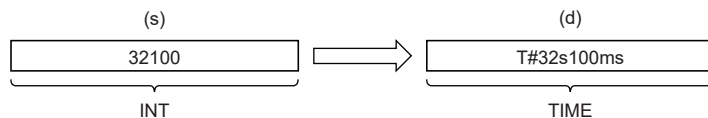
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	TIME

功能

■运算处理

- 将(s)中输入的INT型的数据, 转换为TIME型的数据后从(d)输出。



- 至(s)的输入值为INT型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

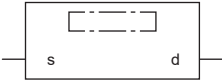
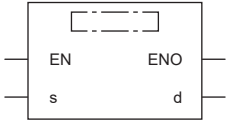
出错

没有运算错误。

31.27 INT型→STRING型转换

INT_TO_STRING(_E)

将INT型数据转换为STRING型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=INT_TO_STRING(s); [带EN/ENO] d:=INT_TO_STRING_E(EN, ENO, s);

设置数据

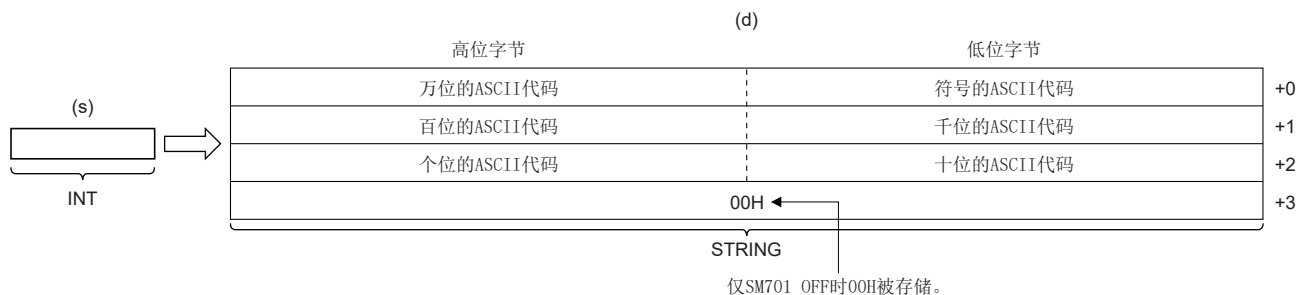
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	STRING(6)

功能

■运算处理

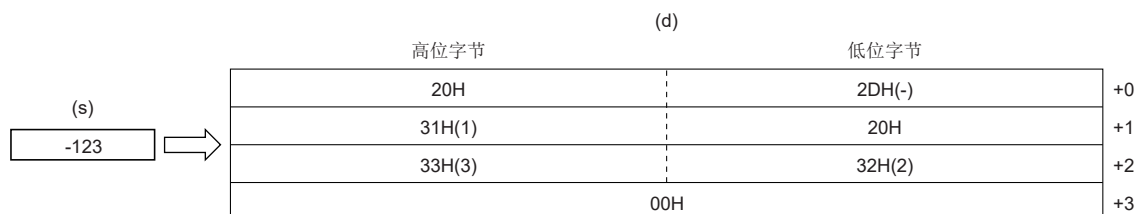
- 将(s)中输入的INT型的数据，转换为STRING型的数据后从(d)输出。



- 至(s)的输入值为INT型的数据值。
- 在符号数据中，输入的值为正时存储20H(空白)，为负时存储2DH(-)。
- 有效位数较少的情况下，高位的位中将存储20H(空白)。

例

输入了-123的情况下



- SM701 (输出字数转换)为OFF的情况下，字符串的最后(第4字)将存储00H。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。


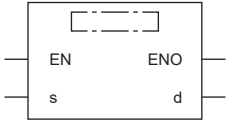
出错

没有运算错误。

31.28 DINT型→BOOL型转换

DINT_TO_BOOL(_E)

将DINT型数据转换为BOOL型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=DINT_TO_BOOL(s); [带EN/ENO] d:=DINT_TO_BOOL_E(EN, ENO, s);

设置数据

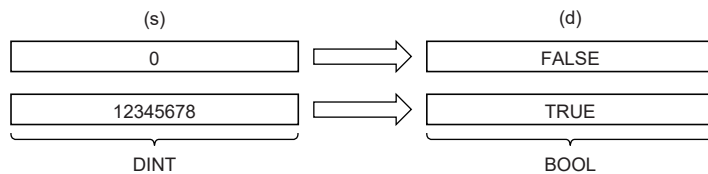
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	DINT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	BOOL

功能

■运算处理

- 将(s)中输入的DINT型的数据, 转换为BOOL型的数据后从(d)输出。
- 输入值为0的情况下, 输出FALSE。
- 输入值为0以外的情况下, 输出TRUE。



- 至(s)的输入值是DINT型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


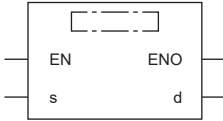
出错

没有运算错误。

31.29 DINT型→WORD型转换

DINT_TO_WORD(_E)

将DINT型数据转换为WORD型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=DINT_TO_WORD(s); [带EN/ENO] d:=DINT_TO_WORD_E(EN, ENO, s);

设置数据

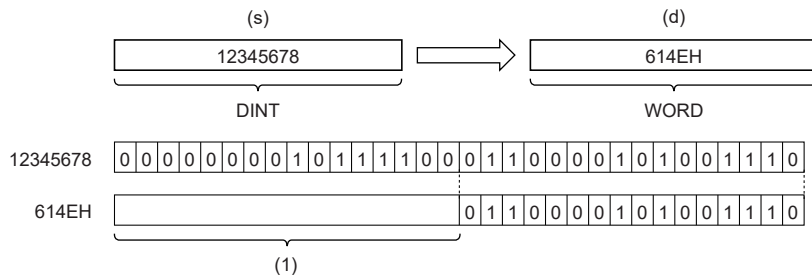
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	DINT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	WORD

功能

■运算处理

- 将(s)中输入的DINT型的数据, 转换为WORD型的数据后从(d)输出。
- 输入值的高位16位的信息将被删除。(下述(1))



- 至(s)的输入值是DINT型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

要点

执行DINT_TO_WORD(_E)时, 从(s)输入的DINT型数据值的高位16位的信息将被删除。


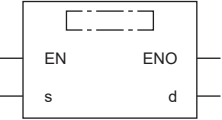
出错

没有运算错误。

31.30 DINT型→DWORD型转换

DINT_TO_DWORD(_E)

将DINT型数据转换为DWORD型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=DINT_TO_DWORD(s); [带EN/ENO] d:=DINT_TO_DWORD_E(EN, ENO, s);

设置数据

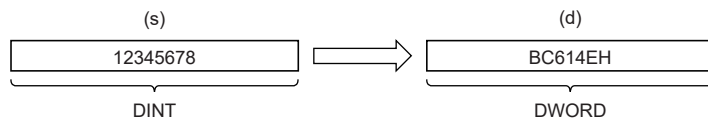
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	DINT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	DWORD

功能

■运算处理

- 将(s)中输入的DINT型的数据, 转换为DWORD型的数据后从(d)输出。



- 至(s)的输入值是DINT型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

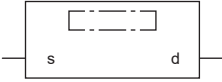
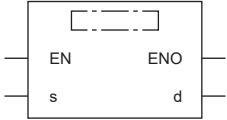
出错

没有运算错误。

31.31 DINT型→INT型转换

DINT_TO_INT(_E)

将DINT型数据转换为INT型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=DINT_TO_INT(s); [带EN/ENO] d:=DINT_TO_INT_E(EN, ENO, s);

设置数据

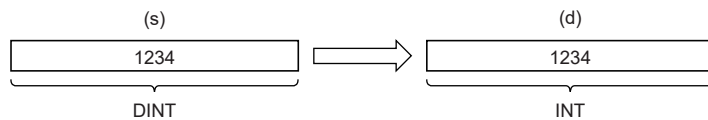
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	DINT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	INT

功能

■运算处理

- 将(s)中输入的DINT型的数据, 转换为INT型的数据后从(d)输出。



- 至(s)的输入值是DINT型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


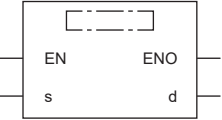
出错

错误代码(SD0)	内容
3281H	(s)中设置的带符号BIN32位数据超出-32768~32767的范围时。

31.32 DINT型→BCD型转换

DINT_TO_BCD(_E)

将DINT型数据转换为BCD型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=DINT_TO_BCD(s); [带EN/ENO] d:=DINT_TO_BCD_E(EN, ENO, s);

设置数据

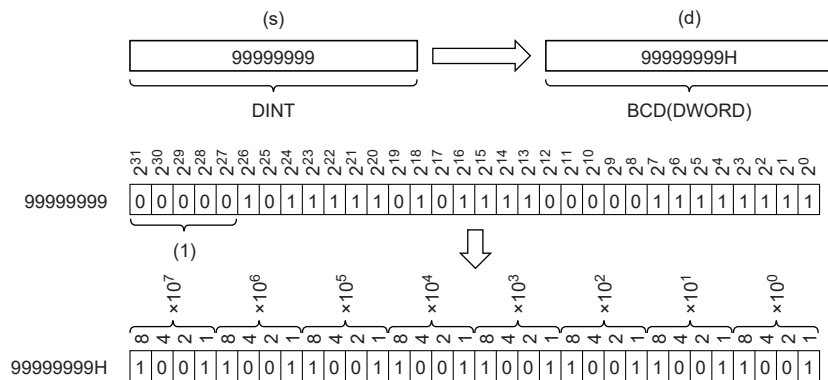
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	DINT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_BIT

功能

■运算处理

- 将(s)中输入的DINT型数据转换为BCD型(DWORD型)数据后通过(d)进行输出。



(1) 必须设置为0。

- 至(s)的输入值是DINT型的数据值。(d)为WORD的情况下, 输入值的范围为0~9999。(d)为DWORD型的情况下, 输入值的范围为0~99999999。
- (d)中可以指定WORD型、DWORD型。不能指定BOOL型。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

出错

- (d) 为WORD型的情况下

错误代码 (SD0)	内容
3281H	(s) 中设置的带符号BIN32位数据超出-32768~32767的范围时。
	(s) 的数据超出0~9999的范围时。

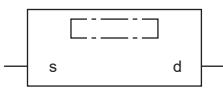
- (d) 为DWORD型的情况下

错误代码 (SD0)	内容
3281H	(s) 的数据超出0~99999999的范围时。

31.33 DINT型→REAL型转换

DINT_TO_REAL(_E)

将DINT型数据转换为REAL型数据。

梯形图、FBD/LD	ST
[无EN/ENO] 	[无EN/ENO] <code>d:=DINT_TO_REAL(s);</code> [带EN/ENO] <code>d:=DINT_TO_REAL_E(EN, ENO, s);</code>

设置数据

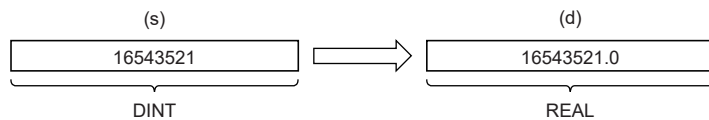
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	DINT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	REAL

功能

■运算处理

- 将(s)中输入的DINT型的数据, 转换为REAL型的数据后从(d)输出。



- 至(s)的输入值是DINT型的数据值。
- REAL型数据是以32位的单精度进行处理, 因此有效位数约为7位。
- 整数超出-16777216~16777215的范围的情况下, 转换后的值将产生误差。(化整误差)

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

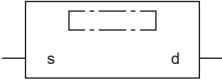
出错

没有运算错误。

31.34 DINT型→LREAL型转换

DINT_TO_LREAL(_E)

将DINT型数据转换为LREAL型数据。

梯形图、FBD/LD	ST
[无EN/ENO] 	[无EN/ENO] <code>d:=DINT_TO_LREAL(s);</code> [带EN/ENO] <code>d:=DINT_TO_LREAL_E(EN, ENO, s);</code>

设置数据

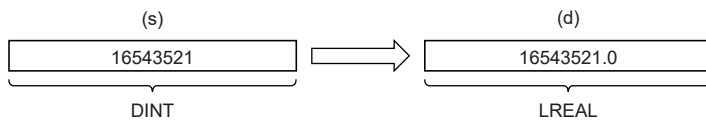
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	DINT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	LREAL

功能

■运算处理

- 将(s)中输入的DINT型的数据, 转换为LREAL型的数据后从(d)输出。



- 至(s)的输入值是DINT型的数据值。
- LREAL型数据是以64位的双精度进行处理, 因此有效位数约为15位。
- 整数超出-2147483648~2147483647的范围的情况下, 转换后的值将产生误差。(化整误差)

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

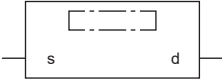
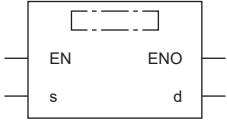
出错

没有运算错误。

31.35 DINT型→TIME型转换

DINT_TO_TIME(_E)

将DINT型数据转换为TIME型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=DINT_TO_TIME(s); [带EN/ENO] d:=DINT_TO_TIME_E(EN, ENO, s);

设置数据

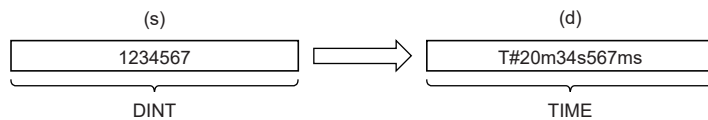
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	DINT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	TIME

功能

■运算处理

- 将(s)中输入的DINT型的数据, 转换为TIME型的数据后从(d)输出。



- 至(s)的输入值是DINT型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

出错

没有运算错误。

31.36 DINT型→STRING型转换

DINT_TO_STRING(_E)

将DINT型数据转换为STRING型数据。

梯形图、FBD/LD		ST
<p>[无EN/ENO]</p> 	<p>[带EN/ENO]</p> 	<p>[无EN/ENO] d:=DINT_TO_STRING(s);</p> <p>[带EN/ENO] d:=DINT_TO_STRING_E(EN, ENO, s);</p>

设置数据

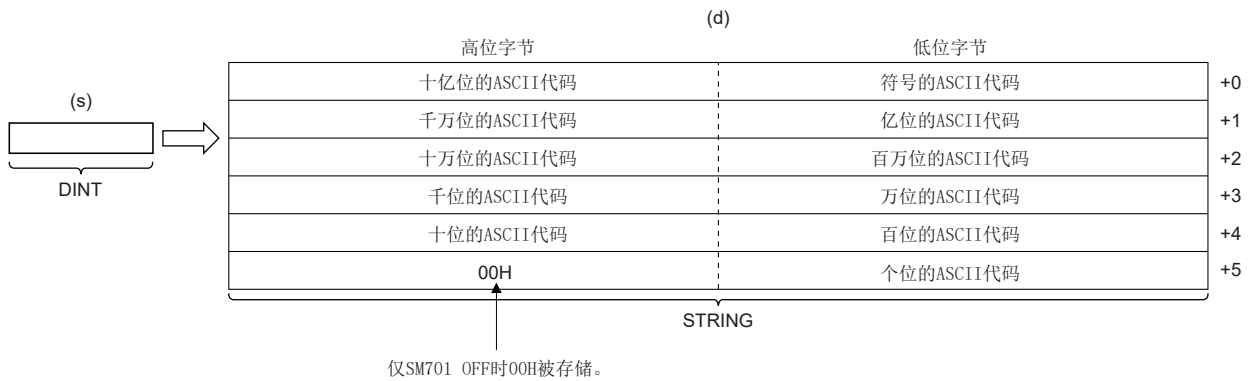
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	DINT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	STRING(11)

功能

■运算处理

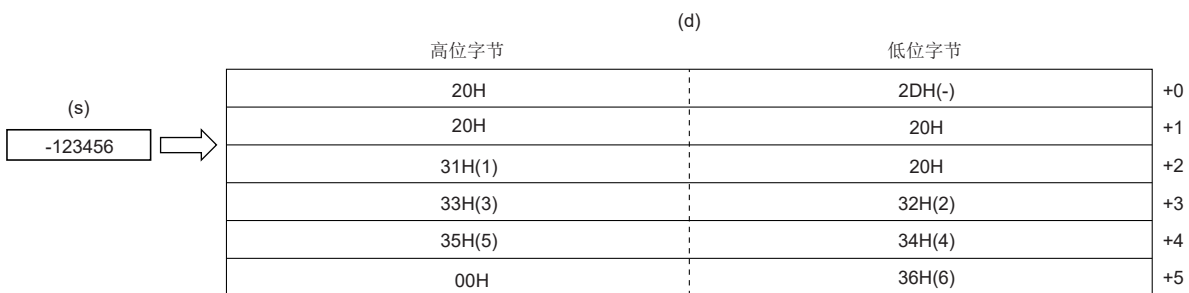
- 将(s)中输入的DINT型的数据，转换为STRING型的数据后从(d)输出。



- 至(s)的输入值是DINT型的数据值。
- 在符号数据中，输入的值为正时存储20H(空白)，为负时存储2DH(-)。
- 有效位数较少的情况下，高位的位中将存储20H(空白)。

例

输入了-123456的情况下



- SM701(输出字数转换)为OFF的情况下，字符串的最后(第6字高位)将存储00H。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

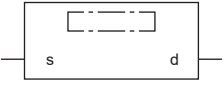
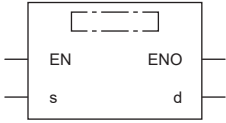
出错

没有运算错误。

31.37 BCD型→INT型转换

BCD_TO_INT(_E)

将BCD型数据转换为INT型数据。

梯形图、FBD/LD		ST
<p>[无EN/ENO]</p> 	<p>[带EN/ENO]</p> 	<p>[无EN/ENO] d:=BCD_TO_INT(s);</p> <p>[带EN/ENO] d:=BCD_TO_INT_E(EN, ENO, s);</p>

设置数据

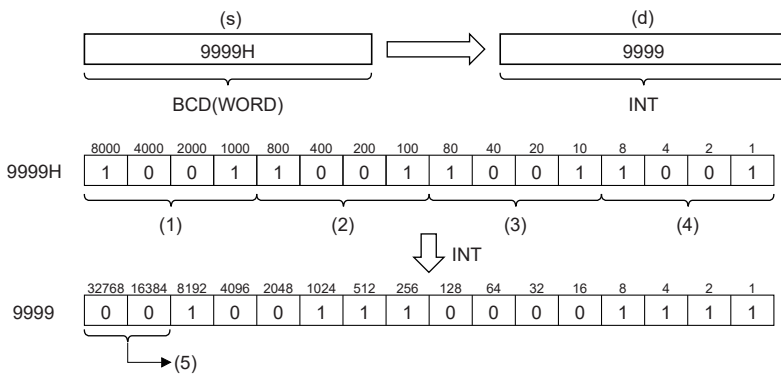
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	WORD
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	INT

功能

■运算处理

- 将(s)中输入的BCD型(WORD型)数据转换为INT型数据后通过(d)进行输出。



- (1) 千位
- (2) 百位
- (3) 十位
- (4) 个位
- (5) 必定变为0。

- 至(s)的输入值是WORD型的数据值且在0H~9999H(各位为0~9的范围)的范围内。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件		运算结果	
EN		ENO	(d)
TRUE (运算执行)		TRUE	运算输出值
FALSE (运算停止)		FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

出错

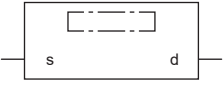
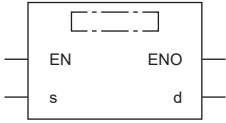
错误代码(SD0)	内容
3281H	(s)的各位中有0~9以外的值时。

通过预先将SM754置为ON，可以避免上述错误。设置了超出范围的数值的情况下，与SM754的ON/OFF无关，不执行指令。

31.38 BCD型→DINT型转换

BCD_TO_DINT(_E)

将BCD型数据转换为DINT型数据。

梯形图、FBD/LD		ST
<p>[无EN/ENO]</p> 	<p>[带EN/ENO]</p> 	<p>[无EN/ENO] d:=BCD_TO_DINT(s);</p> <p>[带EN/ENO] d:=BCD_TO_DINT_E(EN, ENO, s);</p>

设置数据

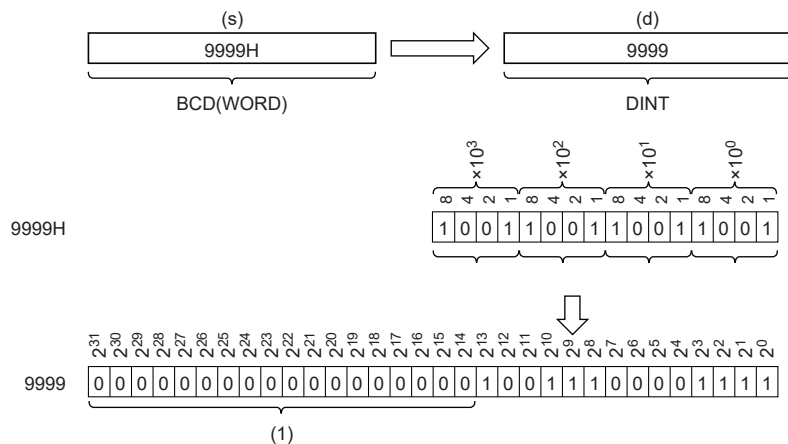
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	ANY_BIT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	DINT

功能

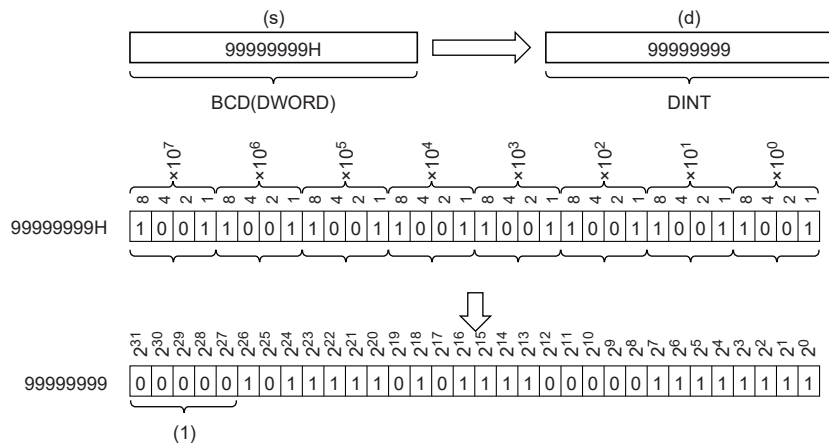
■运算处理

- 将(s)中输入的BCD型(WORD型或DWORD型)数据转换为DINT型数据后通过(d)进行输出。
- (s)中指定了WORD型的情况下



(1) 必定变为0。

- (s)中指定了DWORD型的情况下



(1) 必定变为0。

- 至(s)的输入值, 若为WORD型数据值, 在0H~9999H(各位为0~9的范围)的范围内, 若为DWORD型的数据值, 在0H~99999999H(各位为0~9的范围)的范围内。
- (s)中可以指定WORD型、DWORD型。不能指定BOOL型。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

出错

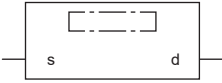
错误代码(SD0)	内容
3281H	(s)的各位中有0~9以外的值时。

通过预先将SM754置为ON, 可以避免上述错误。此外, 设置了超出范围的数值的情况下, 与SM754的ON/OFF无关, 不执行指令。

31.39 BCD型→STRING型转换

BCD_TO_STRING(_E)

将BCD型数据转换为STRING型数据。

梯形图、FBD/LD	ST
[无EN/ENO] 	[无EN/ENO] <code>d:=BCD_TO_STRING(s);</code> [带EN/ENO] <code>d:=BCD_TO_STRING_E(EN, ENO, s);</code>

设置数据

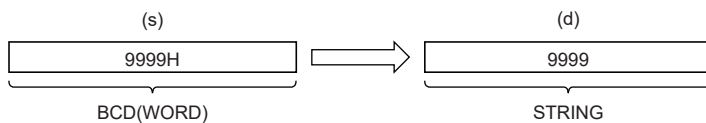
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	ANY_BIT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	STRING(8)

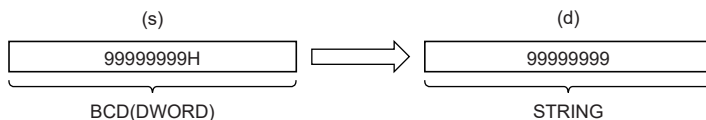
功能

■运算处理

- 将(s)中输入的BCD型(WORD型或DWORD型)数据转换为STRING型数据后通过(d)进行输出。
- (s)中指定了WORD型的情况下



- (s)中指定了DWORD型的情况下



- (s)中可以指定WORD型、DWORD型。不能指定BOOL型。
- SM701(输出字数转换)为OFF的情况下,字符串的最后将存储00H。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下,从(d)输出的数据将变为不定值。在此情况下,应进行不使用从(d)输出的数据的程序处理。

出错

- (s) 为WORD型的情况下

错误代码 (SD0)	内容
3281H	(s) 的数据超出0~9999的范围时。

- (s) 为DWORD型的情况下

错误代码 (SD0)	内容
3281H	(s) 的数据超出0~99999999的范围时。

31.40 REAL型→INT型转换

REAL_TO_INT(_E)

将REAL型数据转换为INT型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=REAL_TO_INT(s); [带EN/ENO] d:=REAL_TO_INT_E(EN, ENO, s);

设置数据

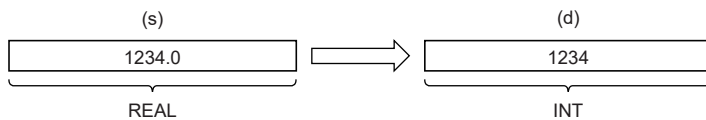
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	REAL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	INT

功能

■运算处理

- 将(s)中输入的REAL型的数据, 转换为INT型的数据后从(d)输出。



- 至(s)的输入值是REAL型的数据值且在-32768~32767的范围内。
- 转换后的数据将变为REAL型数据值的小数点以下第1位被四舍五入后的值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

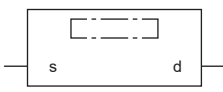
出错

错误代码(SD0)	内容
3281H	(s)中设置的单精度实数超出-32768~32767的范围时。
3282H	<ul style="list-style-type: none"> (s)中设置了特殊数时。 设置的单精度实数超出以下范围。 $-2^{128} < (s) \leq -2^{-126}$、$0$、$2^{-126} \leq (s) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38、0、E1.17549435-38~E3.40282347+38) 设置的软件、标签的内容为-0、非正规化数、非数、±∞时。

31.41 REAL型→DINT型转换

REAL_TO_DINT(_E)

将REAL型数据转换为DINT型数据。

梯形图、FBD/LD	ST
<p>[无EN/ENO]</p> 	<p>[无EN/ENO] d:=REAL_TO_DINT(s);</p> <p>[带EN/ENO] d:=REAL_TO_DINT_E(EN, ENO, s);</p>

设置数据

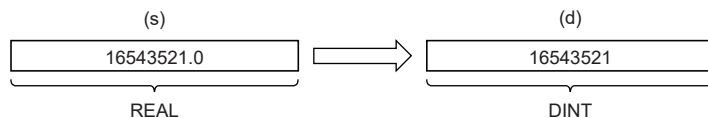
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	REAL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	DINT

功能

■运算处理

- 将(s)中输入的REAL型的数据, 转换为DINT型的数据后从(d)输出。



- 至(s)的输入值是REAL型的数据值且在-2147483648~2147483647的范围内。
- 转换后的数据将变为REAL型数据值的小数点以下第1位被四舍五入后的值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

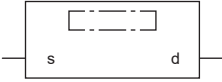
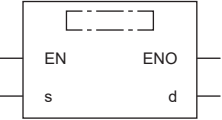
出错

错误代码(SD0)	内容
3281H	(s)中设置的单精度实数超出-2147483648~2147483647的范围时。
3282H	<ul style="list-style-type: none"> (s)中设置了特殊数时。 设置的单精度实数超出以下范围。 $-2^{128} < (s) \leq -2^{-126}$、$0$、$2^{-126} \leq (s) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38、0、E1.17549435-38~E3.40282347+38) 设置的软元件、标签的内容为-0、非正规化数、非数、±∞时。

31.42 REAL型→LREAL型转换

REAL_TO_LREAL(_E)

将REAL型数据转换为LREAL型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=REAL_TO_LREAL(s); [带EN/ENO] d:=REAL_TO_LREAL_E(EN, ENO, s);

设置数据

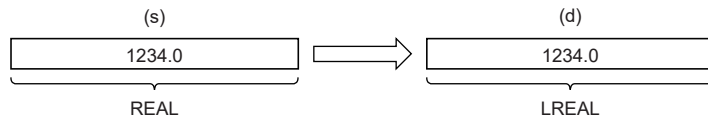
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	REAL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	LREAL

功能

■运算处理

- 将(s)中输入的REAL型的数据, 转换为LREAL型的数据后从(d)输出。



- 至(s)的输入值是REAL型的数据值。
- REAL型数据是以32位的单精度进行处理, 因此有效位数约为7位。
- 整数超出-16777216~16777215的范围的情况下, 转换后的值将产生误差。(化整误差)

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


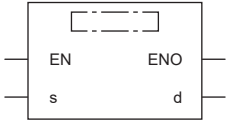
出错

错误代码(SD0)	内容
3282H	(s)的内容为-0或超出以下范围时。 $-2^{128} < (s) \leq -2^{-126}$ 、 0 、 $2^{-126} \leq (s) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38、0、E1.17549435-38~E3.40282347+38)

31.43 REAL型→STRING型转换

REAL_TO_STRING(_E)

将REAL型数据转换为STRING型(指数形式)数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=REAL_TO_STRING(s); [带EN/ENO] d:=REAL_TO_STRING_E(EN, ENO, s);

设置数据

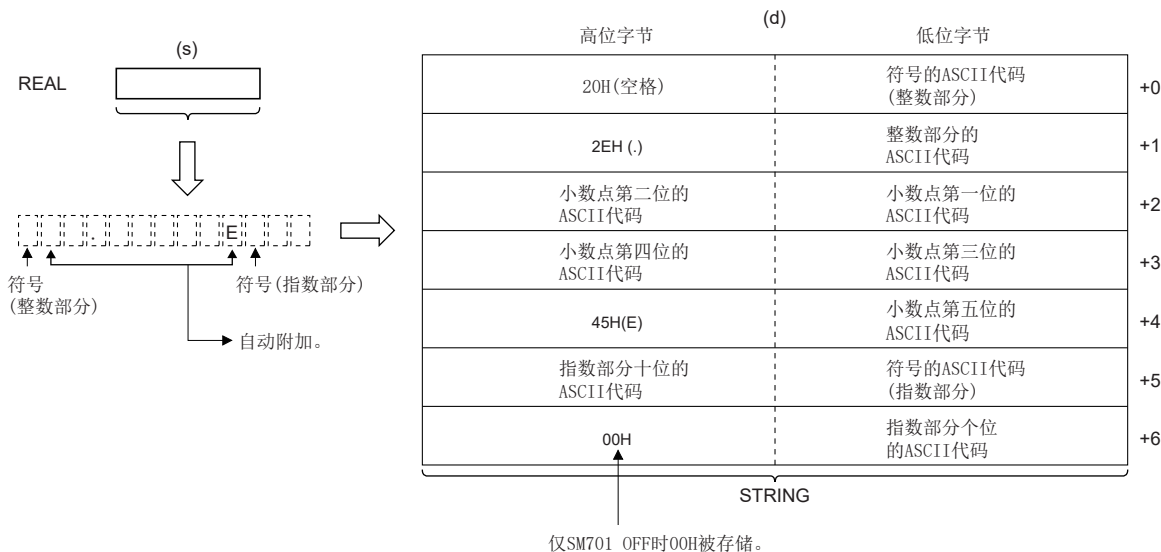
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	REAL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	STRING(13)

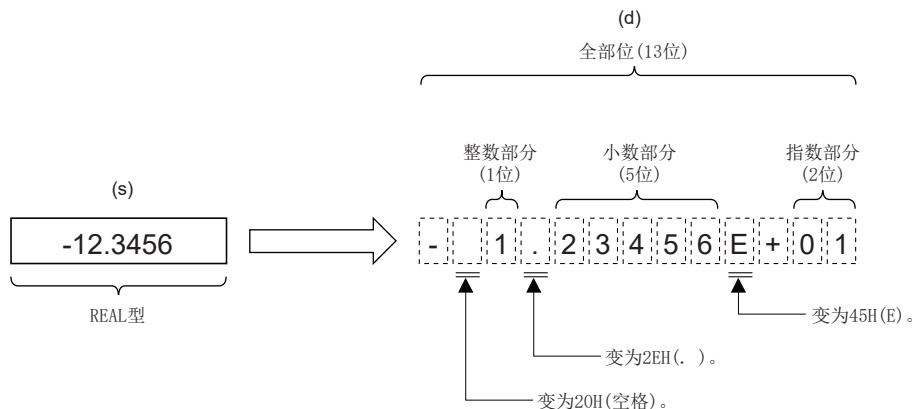
功能

■运算处理

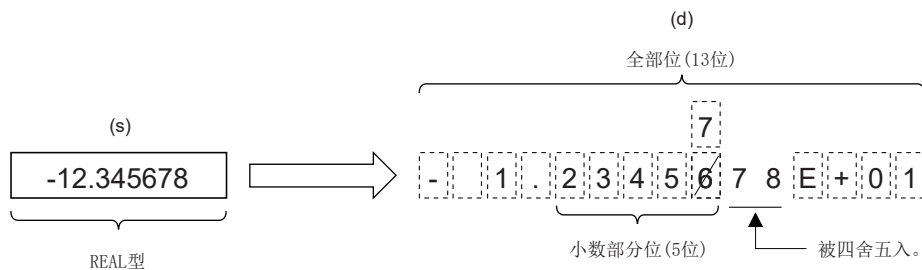
- 将(s)中输入的REAL型的数据，转换为STRING型(指数形式)的数据后从(d)输出。



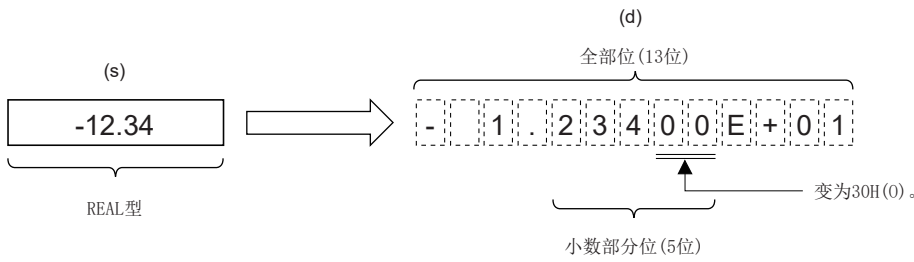
- 至(s)的输入值是REAL型的数据值。
- 转换后的字符串数据从(d)中按以下方式被输出。
- 整数部、小数部、指数部的位数是固定的。(整数部：1位，小数部：5位，指数部：2位)
- 第2字节、第4字节、第10字节中，将分别自动存储“20H”(空白)、“2EH”(.)、“45H”(E)。



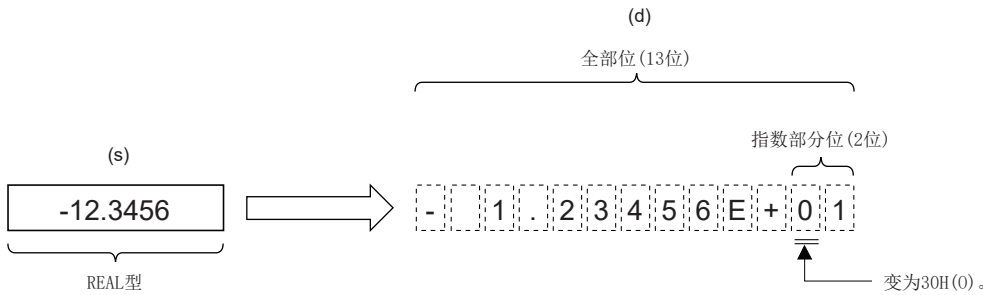
- 符号数据(整数部)中，输入的正时将存储“20H”(空白)，为负时将存储“2DH”(-)。
- 小数部的第6位及其以后将被四舍五入。



- 有效位数较少的情况下，小数部将存储“30H”(0)。



- 符号数据(指数部)中，指数为正时存储“2BH”(+)，为负时存储“2DH”(-)。
- 指数部为1位的情况下，在指数部十的位存储“30H”(0)。



- 字符串的最后(第7字)中将自动存储00H。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

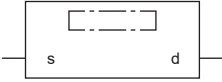
出错

错误代码(SD0)	内容
3282H	<ul style="list-style-type: none"> (s)的内容超出以下范围时。 $-2^{128} < (s) \leq -2^{-126}$、$0 < 2^{-126} \leq (s) < 2^{128}$ $(E-3.40282347+38 \sim E-1.17549435-38)$、$0$、$E1.17549435-38 \sim E3.40282347+38$ (s)的内容为-0、非正规化数、非数、$\pm\infty$时。
3286H	(d)中指定的标签或指定的软元件编号开始至相应软元件的最终软元件编号为止的点数中，无法完整存储转换后的字符串时。

31.44 LREAL型→INT型转换

LREAL_TO_INT(_E)

将LREAL型数据转换为INT型数据。

梯形图、FBD/LD	ST
[无EN/ENO] 	[无EN/ENO] $d := \text{LREAL_TO_INT}(s);$ [带EN/ENO] $d := \text{LREAL_TO_INT_E}(EN, ENO, s);$

设置数据

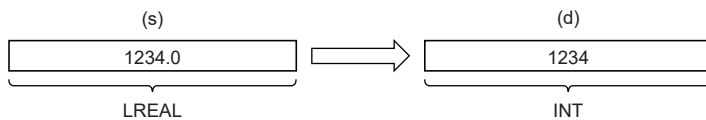
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	LREAL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	INT

功能

■运算处理

- 将(s)中输入的LREAL型的数据转换为INT型的数据后从(d)输出。



- 至(s)的输入值是LREAL型的数据值。
- 转换后的数据, 将变为对LREAL型数据值的小数点以下第1位进行了四舍五入后的值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


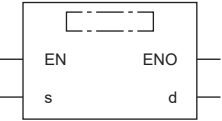
出错

错误代码(SD0)	内容
3281H	(s)的内容超出-32768~32767的范围时。
3282H	<ul style="list-style-type: none"> (s)的内容为-0或超出以下范围时。 $-2^{1024} < (s)$、$(d) \leq 2^{-1022}$、0、$2^{-1022} \leq (s)$、$(d) < 2^{1024}$ $(E-1.7976931348623157+308 \sim E-2.2250738585072014-308)$、$0$、$E2.2250738585072014-308 \sim E1.7976931348623157+308$) 设置的软元件、标签的内容为-0、非正规化数、非数、$\pm\infty$时。

31.45 LREAL型→DINT型转换

LREAL_TO_DINT(_E)

将LREAL型数据转换为DINT型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=LREAL_TO_DINT(s); [带EN/ENO] d:=LREAL_TO_DINT_E(EN, ENO, s);

设置数据

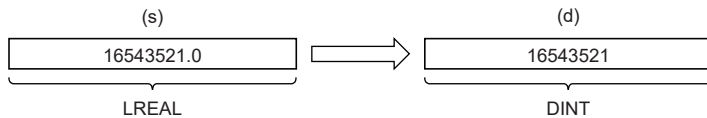
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	LREAL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	DINT

功能

■运算处理

- 将(s)中输入的LREAL型的数据, 转换为DINT型的数据后从(d)输出。



- 至(s)的输入值是LREAL型的数据值。
- 转换后的数据, 将变为对LREAL型数据值的小数点以下第1位进行了四舍五入后的值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

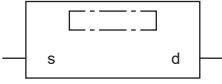
出错

错误代码(SD0)	内容
3281H	(s)的内容超出-2147483648~2147483647的范围时。
3282H	<ul style="list-style-type: none"> (S)的内容为-0或超出以下的范围时。 $-2^{1024} < (s)$、$(d) \leq 2^{-1022}$、0、$2^{-1022} \leq (s)$、$(d) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308、0、E2.2250738585072014-308~E1.7976931348623157+308) 设置的软元件、标签的内容为-0、非正规化数、非数、±∞时。

31.46 LREAL型→REAL型转换

LREAL_TO_REAL(_E)

将LREAL型数据转换为REAL型数据。

梯形图、FBD/LD	ST
[无EN/ENO] 	[无EN/ENO] $d := \text{LREAL_TO_REAL}(s);$ [带EN/ENO] $d := \text{LREAL_TO_REAL_E}(EN, ENO, s);$

设置数据

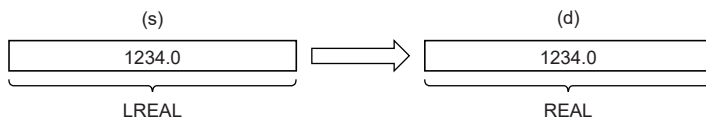
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	LREAL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	REAL

功能

■运算处理

- 将(s)中输入的LREAL型的数据, 转换为REAL型的数据后从(d)输出。



- 至(s)的输入值是LREAL型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


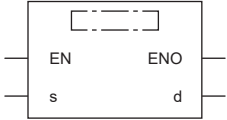
出错

错误代码(SD0)	内容
3282H	(s)的内容为-0或超出以下范围时。 $-2^{1024} < (s), (d) \leq -2^{1022}, 0, 2^{1022} \leq (s), (d) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308、0、E2.2250738585072014-308~E1.7976931348623157+308)
3283H	(d)超出下述范围时。(发生了溢出时) $ (d) < 2^{128}$

31.47 TIME型→BOOL型转换

TIME_TO_BOOL(_E)

将TIME型数据转换为BOOL型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=TIME_TO_BOOL(s); [带EN/ENO] d:=TIME_TO_BOOL_E(EN, ENO, s);

设置数据

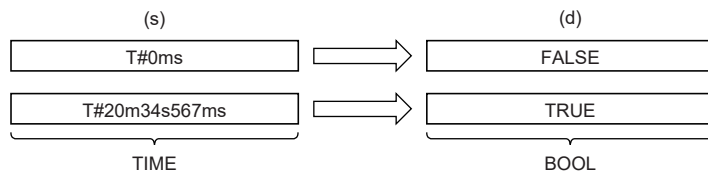
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	TIME
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	BOOL

功能

■运算处理

将(s)中输入的TIME型的数据, 转换为BOOL型的数据后从(d)输出。



■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


出错

没有运算错误。

31.48 TIME型→WORD型转换

TIME_TO_WORD(_E)

将TIME型数据转换为WORD型数据。

梯形图、FBD/LD	ST
[无EN/ENO] 	[无EN/ENO] $d := \text{TIME_TO_WORD}(s);$ [带EN/ENO] $d := \text{TIME_TO_WORD_E}(\text{EN}, \text{ENO}, s);$

设置数据

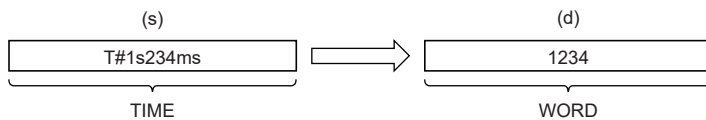
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	TIME
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	WORD

功能

■运算处理

- 将(s)中输入的TIME型的数据, 转换为WORD型的数据后从(d)输出。



- 至(s)的输入值是TIME型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


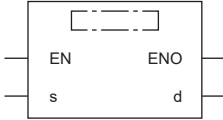
出错

没有运算错误。

31.49 TIME型→DWORD型转换

TIME_TO_DWORD(_E)

将TIME型数据转换为DWORD型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=TIME_TO_DWORD(s); [带EN/ENO] d:=TIME_TO_DWORD_E(EN, ENO, s);

设置数据

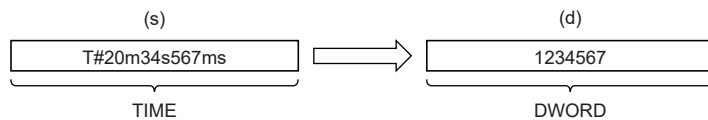
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	TIME
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	DWORD

功能

■运算处理

- 将(s)中输入的TIME型的数据, 转换为DWORD型的数据后从(d)输出。



- 至(s)的输入值是TIME型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


出错

没有运算错误。

31.50 TIME型→INT型转换

TIME_TO_INT(_E)

将TIME型数据转换为INT型数据。

梯形图、FBD/LD	ST
[无EN/ENO] 	[无EN/ENO] <code>d:=TIME_TO_INT(s);</code> [带EN/ENO] <code>d:=TIME_TO_INT_E(EN, ENO, s);</code>

设置数据

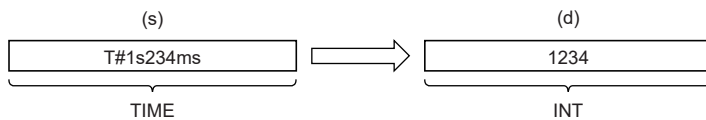
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	TIME
ENO	输出状态(TRUE: 正常执行, FALSE: 异常或停止)	输出变量	BOOL
d	输出	输出变量	INT

功能

■运算处理

- 将(s)中输入的TIME型的数据, 转换为INT型的数据后从(d)输出。



- 至(s)的输入值是TIME型的数据值。
- 转换为INT型时, TIME型的高位16位(1字)的数据将被舍去。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


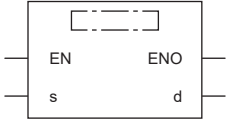
出错

没有运算错误。

31.51 TIME型→DINT型转换

TIME_TO_DINT(_E)

将TIME型数据转换为DINT型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=TIME_TO_DINT(s); [带EN/ENO] d:=TIME_TO_DINT_E(EN, ENO, s);

设置数据

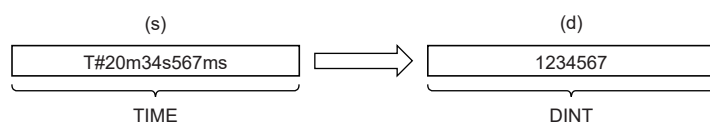
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	TIME
ENO	输出状态(TRUE: 正常执行, FALSE: 异常或停止)	输出变量	BOOL
d	输出	输出变量	DINT

功能

■运算处理

- 将(s)中输入的TIME型的数据, 转换为DINT型的数据后从(d)输出。



- 至(s)的输入值是TIME型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


出错

没有运算错误。

31.52 TIME型→STRING型转换

TIME_TO_STRING(_E)

将TIME型数据转换为STRING型数据。

梯形图、FBD/LD	ST
[无EN/ENO] 	[无EN/ENO] <code>d:=TIME_TO_STRING(s);</code> [带EN/ENO] <code>d:=TIME_TO_STRING_E(EN, ENO, s);</code>

设置数据

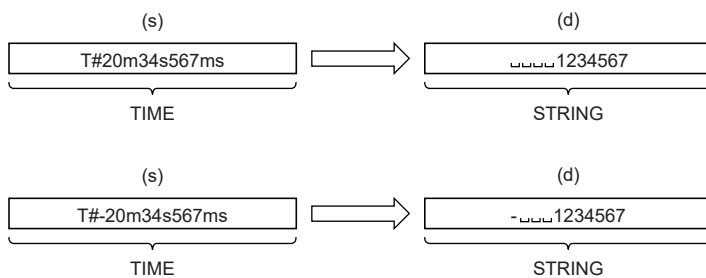
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	TIME
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	STRING STRING(11)

功能

■运算处理

- 将(s)中输入的TIME型的数据, 转换为STRING型的数据后从(d)输出。



- 至(s)的输入值是TIME型的数据值。
- SM701(输出字数转换)为OFF的情况下, 字符串的最后将存储00H。
- 输出中存储的运算结果如下所示。
 - 第1个字符中, BIN数据为正时存储20H(空白), 为负时存储2DH(-)。
 - 有效位数的左侧将存储20H(空白)。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


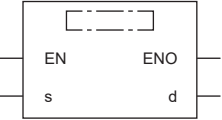
出错

没有运算错误。

31.53 STRING型→BOOL型转换

STRING_TO_BOOL(_E)

将STRING型数据转换为BOOL型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=STRING_TO_BOOL(s); [带EN/ENO] d:=STRING_TO_BOOL_E(EN, ENO, s);

设置数据

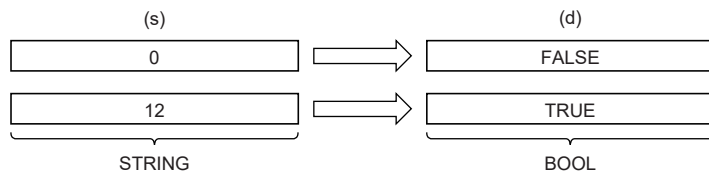
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	STRING(1)
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	BOOL

功能

■运算处理

将(s)中输入的STRING型(小数点形式/指数形式)的数据, 转换为BOOL型的数据后从(d)输出。



■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


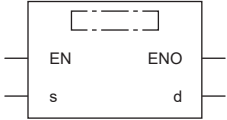
出错

没有运算错误。

31.54 STRING型→WORD型转换

STRING_TO_WORD(_E)

将STRING型数据转换为WORD型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=STRING_TO_WORD(s); [带EN/ENO] d:=STRING_TO_WORD_E(EN, ENO, s);

设置数据

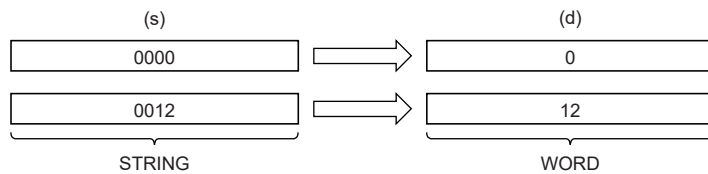
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	STRING (4)
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	WORD

功能

■运算处理

- 将(s)中输入的STRING型的数据, 转换为WORD型的数据后从(d)输出。



- 至(s)的输入值为STRING型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


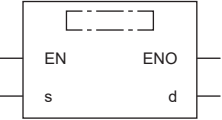
出错

错误代码(SD0)	内容
3281H	输入的各位的ASCII代码超出30H~39H、41H~46H的范围时。

31.55 STRING型→DWORD型转换

STRING_TO_DWORD(_E)

将STRING型数据转换为DWORD型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=STRING_TO_DWORD(s); [带EN/ENO] d:=STRING_TO_DWORD_E(EN, ENO, s);

设置数据

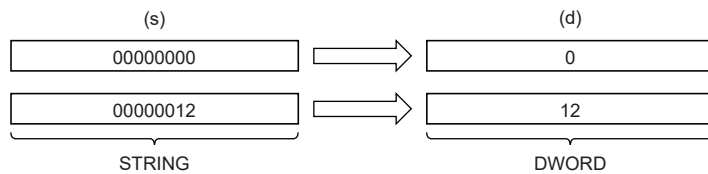
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	STRING (8)
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	DWORD

功能

■运算处理

- 将(s)中输入的STRING型的数据, 转换为DWORD型的数据后从(d)输出。



- 至(s)的输入值为STRING型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


出错

错误代码(SD0)	内容
3281H	输入的各位的ASCII代码超出30H~39H、41H~46H的范围时。

31.56 STRING型→INT型转换

STRING_TO_INT(_E)

将STRING型数据转换为INT型数据。

梯形图、FBD/LD	ST
[无EN/ENO] 	[无EN/ENO] <code>d:=STRING_TO_INT(s);</code> [带EN/ENO] <code>d:=STRING_TO_INT_E(EN, ENO, s);</code>

设置数据

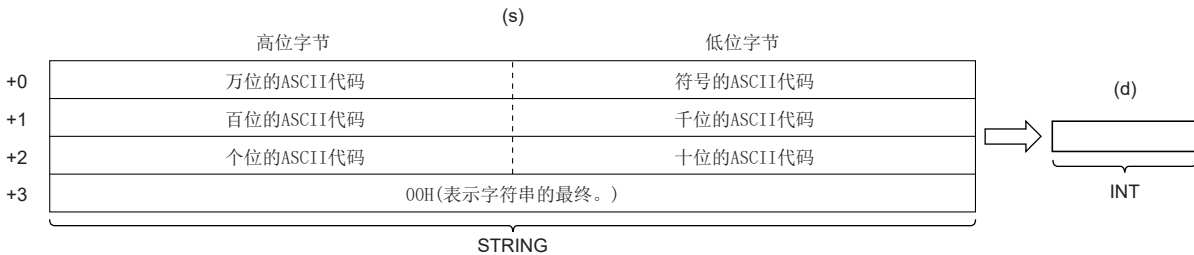
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	STRING(6)
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	INT

功能

■运算处理

- 将(s)中输入的STRING型的数据, 转换为INT型的数据后从(d)输出。



- 至(s)的输入值为STRING型的数据值且在如下所示的范围内。
- 以ASCII代码时30H~39H、20H、2DH、00H的范围
- 以STRING型的数据值时-32768~32767的范围

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

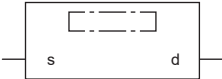
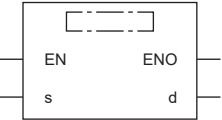
出错

错误代码(SD0)	内容
3281H	(s)中输入了不能转换的非法数据时。 <ul style="list-style-type: none"> 各位的ASCII代码超出30H~39H、20H、00H的范围。 ASCII数据超出-32768~32767的范围。

31.57 STRING型→DINT型转换

STRING_TO_DINT(_E)

将STRING型数据转换为DINT型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=STRING_TO_DINT(s); [带EN/ENO] d:=STRING_TO_DINT_E(EN, ENO, s);

设置数据

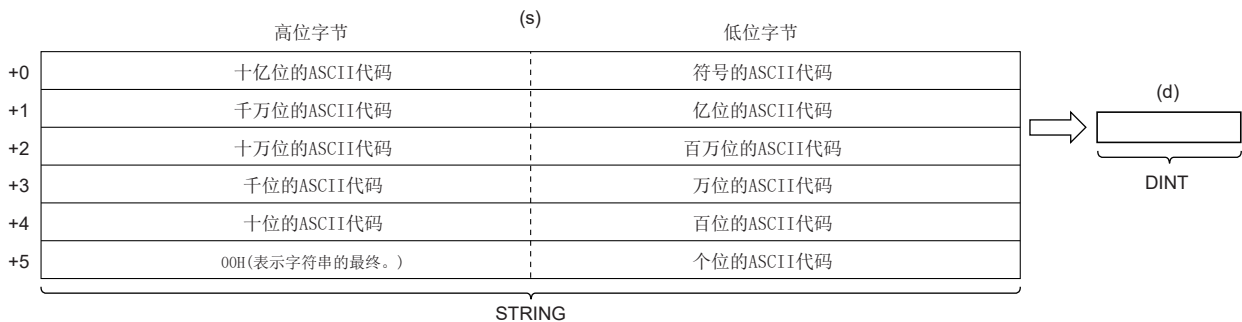
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	STRING(11)
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	DINT

功能

■运算处理

- 将(s)中输入的STRING型的数据, 转换为DINT型的数据后从(d)输出。



- 至(s)的输入值为STRING型的数据值且在如下所示的范围内。
- 以ASCII代码时30H~39H、20H、2DH、00H的范围
- 以STRING型的数据值时-2147483648~2147483647的范围

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值


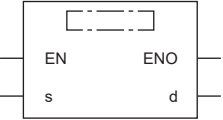
*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

错误代码 (SD0)	内容
3281H	(s) 中输入了不能转换的非法数据时。 • 各位的ASCII代码超出30H~39H、20H、00H的范围。 • ASCII数据超出-2147483648~2147483647的范围。

31.58 STRING型→BCD型转换

STRING_TO_BCD(_E)

将STRING型数据转换为BCD型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=STRING_TO_BCD(s); [带EN/ENO] d:=STRING_TO_BCD_E(EN, ENO, s);

设置数据

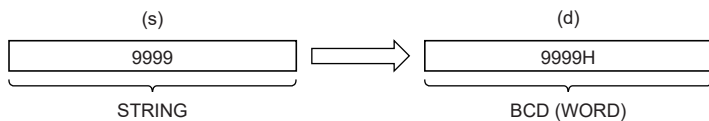
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	STRING(8)
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_BIT

功能

■运算处理

- 将(s)中输入的STRING型4字符数据转换为BCD型(WORD型)数据后通过(d)进行输出。



- 字符串中包含有20H(空白)的情况下，转换时将忽略20H。
- 字符串中的20H(空白)、30H(0)也作为1个字符计数。
- 至(s)的输入值为STRING型的数据值且在如下所示的范围内。
以ASCII代码时为30H~39H、20H、00H的范围
- 输入字符串不足4字符的情况下，字符串的后面将补充0后以4字符进行转换。因此，希望将不足4字符的字符串进行BCD转换的情况下，应输入填充了0的字符串（‘1’的情况下‘0001’）。
- 字符串超过4字符的情况下，从左开始的4字符将成为转换对象。

输入字符串	转换字符串	输出(BCD型)
'1'	'1000'	1000H(4096D)
'12'	'1200'	1200H(4608D)
'123'	'1230'	1230H(4656D)
'1234'	'1234'	1234H(4660D)
'12345'	'1234'	1234H(4660D)

- (d)中可以指定WORD型、DWORD型。不能指定BOOL型。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

- *1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

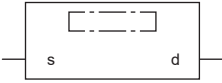
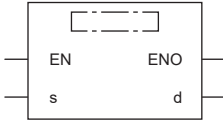
出错

错误代码(SD0)	内容
3281H	以ASCII代码输入了超出30H~39H、20H、00H的范围的字符串时。

31.59 STRING型→REAL型转换

STRING_TO_REAL(_E)

将STRING型数据转换为REAL型数据。

梯形图、FBD/LD		ST
<p>[无EN/ENO]</p> 	<p>[带EN/ENO]</p> 	<p>[无EN/ENO] d:=STRING_TO_REAL(s);</p> <p>[带EN/ENO] d:=STRING_TO_REAL_E(EN, ENO, s);</p>

设置数据

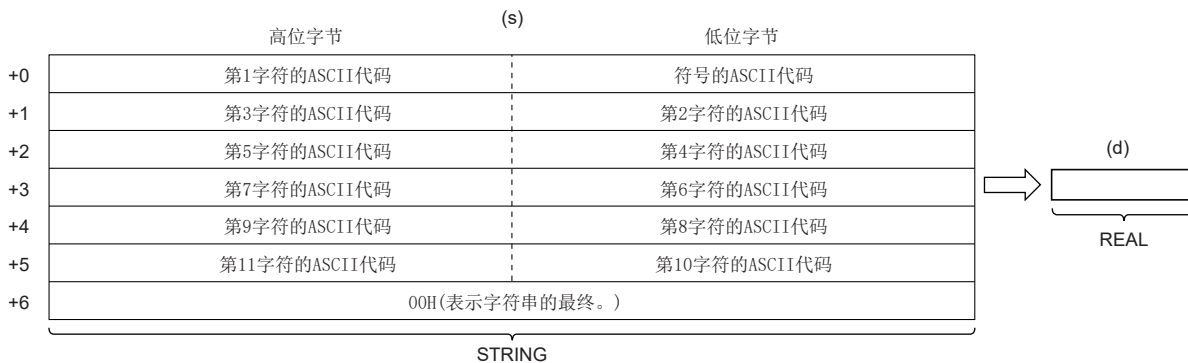
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	STRING(24)
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	REAL

功能

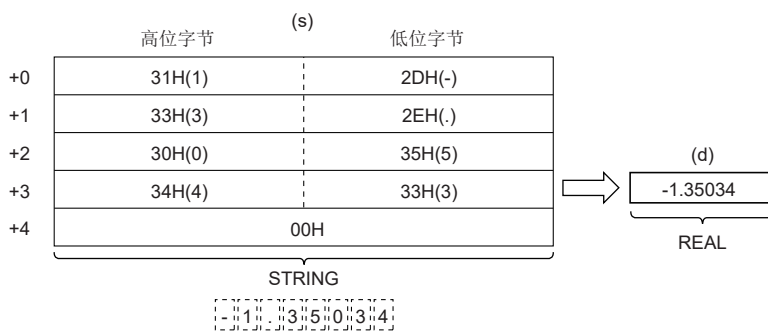
■运算处理

- 将(s)中输入的STRING型(小数点形式/指数形式)的数据, 转换为REAL型的数据后从(d)输出。

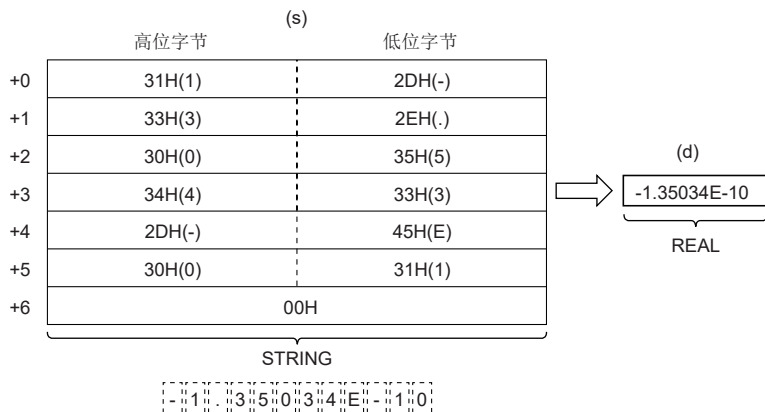


- STRING型数据可以转换为小数点形式、指数形式。

- 小数点形式的情况下

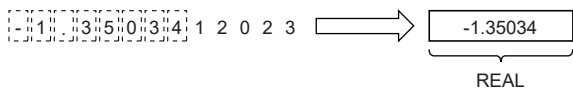


- 指数形式的情况下

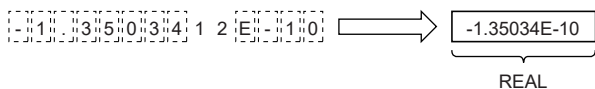


- STRING型数据, 如下所示, 在除去符号、小数点、指数部的6位有效, 转换时第7位及其以后将被舍去。

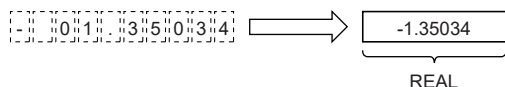
- 小数点形式的情况下



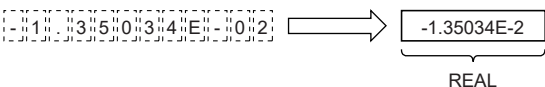
- 指数形式的情况下



- 在小数点形式中将符号指定为2BH(+), 或省略符号时将作为正值进行转换。此外, 将符号指定为2DH(-)时将作为负值进行转换。
- 在指数形式中将指数部的符号指定为2BH(+), 或省略符号时将作为正值进行转换。将指数部的符号指定为2DH(-)时将作为负值进行转换。
- 在STRING型数据中, 最初的0以外的数值之间存在20H(空白)或30H(0)的情况下, 转换时将按下述方式忽略20H、30H。



- 在STRING型数据(指数形式)中“E”与数值之间存在有30H(0)的情况下, 转换时将按下述方式忽略30H。



- 字符串中包含有20H(空白)的情况下, 转换时将忽略20H。
- STRING型数据最多可输入24字符。字符串中的20H(空白)、30H(0)也作为1个字符计数。
- 至(s)的输入值为STRING型的数据值且在如下所示的范围内。
- 以ASCII代码时30H~39H、45H、2BH、2DH、2EH、20H、00H的范围

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


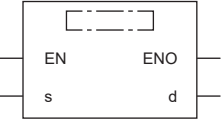
出错

错误代码(SD0)	内容
2820H	从(s)开始的相应软元件范围内没有00H时。
3281H	(s)中设置了不能转换的非法数据时。 <ul style="list-style-type: none"> • 整数部、小数部中, 有30H(0)~39H(9)以外的字符 • 指定的字符串中有2个或其以上的2EH(.) • 指定的字符串的指数部中, 有45H(E)、65H(e)、2BH(+)、2DH(-)以外的字符 • 指定的字符串中有多个45H(E)、65H(e)的指数部 • 指定的字符串中记述了3位或其以上的指数部的数值 • 指定的字符串中有多个2BH(+)、2DH(-)的指数部的符号 • 指定的字符串中, 2BH(+)、2DH(-)的符号为小数点形式的情况: 整数部, 为指数形式的情况: 尾部部中有多个 • (s)及其以后的字符数为0或超过了24字符时。
3283H	(d)超出下述范围时。(发生了溢出时) (d) < 2 ¹²⁸

31.60 STRING型→TIME型转换

STRING_TO_TIME(_E)

将STRING型数据转换为TIME型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=STRING_TO_TIME(s); [带EN/ENO] d:=STRING_TO_TIME_E(EN, ENO, s);

设置数据

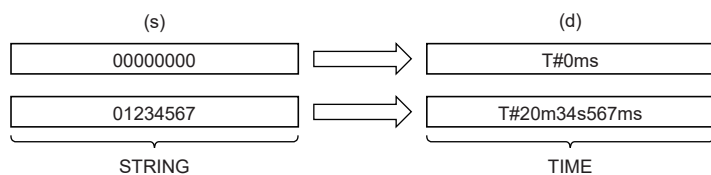
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	STRING(11)
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	TIME

功能

■运算处理

- 将(s)中输入的STRING型的数据, 转换为TIME型的数据后从(d)输出。



- 至(s)的输入值为STRING型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

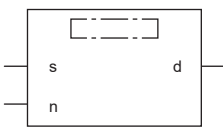
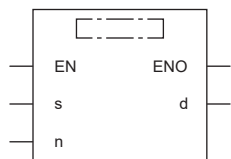
出错

错误代码(SD0)	内容
3281H	输入的各位的ASCII代码超出30H~39H、20H、00H的范围时。 输入的ASCII数据超出以下范围时。 -2147483648~2147483647

31.61 位数组→INT型转换

BITARR_TO_INT(_E)

将通过位数组指定的位数转换为INT型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=BITARR_TO_INT(s,n); [带EN/ENO] d:=BITARR_TO_INT_E(EN,ENO,s,n);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(BitArr)	输入(要素指定也可为变量)	输入变量	BOOL数组要素
n	只能指定4、8、12、16的常数	输入变量	INT
ENO	输出状态(TRUE: 正常执行, FALSE: 异常或停止)	输出变量	BOOL
d	输出	输出变量	ANY16

功能

■运算处理

- 将(s)中输入的位数组要素作为起始将(n)中指定的位数的数据转换为ANY16型的数据后从(d)输出。
- 指定的位数及其以上的输出位中将被设置0。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

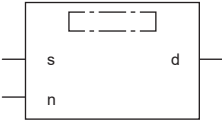
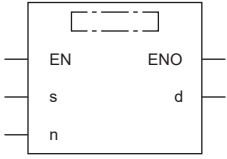
出错

没有运算错误。

31.62 位数组→DINT型转换

BITARR_TO_DINT(_E)

将通过位数组指定的位数转换为DINT型数据。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=BITARR_TO_DINT(s,n) [带EN/ENO] d:=BITARR_TO_DINT_E(EN, ENO, s, n);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(BitArr)	输入(要素指定也可为变量)	输入变量	BOOL数组要素
n	只能指定4、8、12、16、20、24、28、32的常数	输入变量	INT
ENO	输出状态(TRUE: 正常执行, FALSE: 异常或停止)	输出变量	BOOL
d	输出	输出变量	ANY32

功能

■运算处理

- 将(s)中输入的位数组要素作为起始将(n)中指定的位数的数据转换为ANY32型的数据后从(d)输出。
- 指定的位数及其以上的输出位中将被设置0。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

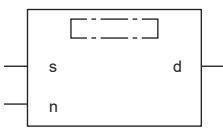
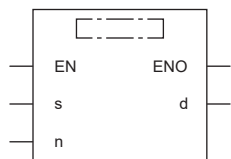
出错

没有运算错误。

31.63 INT型→位数组转换

INT_TO_BITARR(_E)

将INT型数据的低n位输出到位数组中。

梯形图、FBD/LD		ST
<p>[无EN/ENO]</p> 	<p>[带EN/ENO]</p> 	<p>[无EN/ENO] d:=INT_TO_BITARR(s, n);</p> <p>[带EN/ENO] d:=INT_TO_BITARR_E(EN, ENO, s, n);</p>

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s	输入	输入变量	ANY16
n	只能指定4、8、12、16的常数	输入变量	INT
ENO	输出状态(TRUE: 正常执行, FALSE: 异常或停止)	输出变量	BOOL
d	输出(要素指定也可作为变量)	输出变量	BOOL数组要素

功能

■运算处理

- 将(s)中指定的ANY16型的低(n)位输出到(d)中。
- 指定的位数及其以上的输出位不更改。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

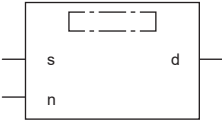
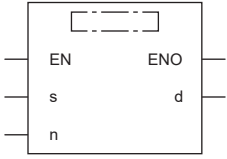
出错

没有运算错误。

31.64 DINT型→位数组转换

DINT_TO_BITARR(_E)

将DINT型数据的低n位输出到位数组中。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] <code>d:=DINT_TO_BITARR(s, n);</code> [带EN/ENO] <code>d:=DINT_TO_BITARR_E(EN, ENO, s, n);</code>

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s	输入	输入变量	ANY32
n	只能指定4、8、12、16、20、24、28、32的常数	输入变量	BOOL
ENO	输出状态(TRUE: 正常执行, FALSE: 异常或停止)	输出变量	BOOL
d	输出(要素指定也可为变量)	输出变量	BOOL数组要素

功能

■运算处理

- 将(s)中指定的ANY32型的低(n)位输出到(d)中。
- 指定的位数及其以上的输出位不更改。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

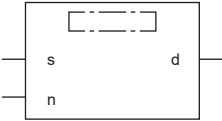
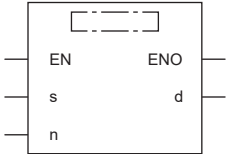
出错

没有运算错误。

31.65 位数组的复制

CPY_BITARR(_E)

将位数组按指定位容量进行复制。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=CPY_BITARR(s, n); [带EN/ENO] d:=CPY_BITARR_E(EN, ENO, s, n);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(BitArrIn)	输入	输入变量	BOOL数组要素
n	只能指定4、8、12、16、20、24、28、32的常数	输入变量	INT
ENO	输出状态(TRUE: 正常执行, FALSE: 异常或停止)	输出变量	BOOL
d	输出	输出变量	BOOL数组要素

功能

■运算处理

将(s)中指定的位数组的(n)位输出到(d)中。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

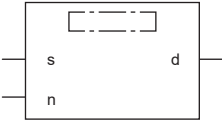
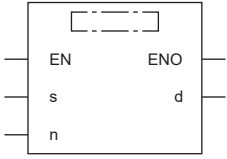
出错

没有运算错误。

31.66 字标签的指定位读取

GET_BIT_OF_INT(_E)

读取字标签的指定位。

梯形图*1		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=GET_BIT_OF_INT(s, n); [带EN/ENO] d:=GET_BIT_OF_INT_E(EN, ENO, s, n);

*1 在FBD/LD中不支持。FBD/LD的情况下，应使用标签的位指定。

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s	输入	输入变量	INT*1
n	只能指定从0至15的常数	输入变量	INT
ENO	输出状态(TRUE: 正常执行, FALSE: 异常或停止)	输出变量	BOOL
d	输出	输出变量	BOOL

*1 不能使用位指定的标签及软元件。

功能

■运算处理

输出(s)的第(n)位。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

出错

没有运算错误。

要点 

使用标签的位指定时，可以方便地创建与GET_BIT_OF_INT相同动作的程序。

以下示例与使用了GET_BIT_OF_INT的情况下一样，将g_int1的位5(b5)的值读取到g_bool1中。

梯形图的情况下



ST的情况下

```
g_bool1 := g_int1.5;
```

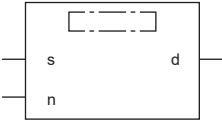
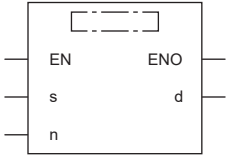
FBD/LD的情况下



31.67 字标签的指定位写入

SET_BIT_OF_INT(_E)

写入到字标签的指定位。

梯形图*1		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=SET_BIT_OF_INT(s, n); [带EN/ENO] d:=SET_BIT_OF_INT_E(EN, ENO, s, n);

*1 在FBD/LD中不支持。FBD/LD的情况下，应使用标签的位指定。

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s	输入	输入变量	BOOL
n	只能指定从0至15的常数	输入变量	INT
ENO	输出状态(TRUE: 正常执行, FALSE: 异常或停止)	输出变量	BOOL
d	输入输出	输入输出变量	INT*1

*1 不能使用位指定的标签及软元件。

功能

■运算处理

将(s)中指定的BOOL值写入到(d)的第(n)位中。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

注意事项

将SET_BIT_OF_INT(_E)通过ST使用的情况下，应创建将返回值代入变量中的程序。

```
//SET_BIT_OF_INT的返回值在代入变量之后使用。  
g_int1 := SET_BIT_OF_INT(TRUE, 0);  
g_bool1 := GET_BIT_OF_INT(g_int1, 0);
```

SET_BIT_OF_INT(_E)的返回值也被用于输入输出，因此请勿直接代入到其它指令、函数、功能块的输入自变量中。

```
//在以下程序中，GET_BIT_OF_INT的第一自变量的值将变为不定值。  
g_bool1 := GET_BIT_OF_INT( SET_BIT_OF_INT(TRUE, 0), 0);
```

出错

没有运算错误。

要点

使用标签的位指定时，可以方便地创建与SET_BIT_OF_INT相同动作的程序。

在以下示例中，与使用了SET_BIT_OF_INT的情况下一样，将g_int1的位5(b5)更改为g_bool1的值。

梯形图的情况下



ST的情况下

```
g_int1.5 := g_bool1;
```

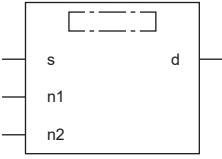
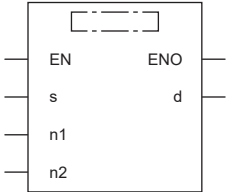
FBD/LD的情况下



31.68 字标签的指定位复制

CPY_BIT_OF_INT(_E)

将字标签的指定位复制到其它字标签的指定位中。

梯形图*1		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] <code>d:=CPY_BIT_OF_INT(s, n1, n2);</code> [带EN/ENO] <code>d:=CPY_BIT_OF_INT_E(EN, ENO, s, n1, n2);</code>

*1 在FBD/LD中不支持。FBD/LD的情况下，应使用标签的位指定。

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s	输入	输入变量	INT*1
n1	输入变量的位指定(只能指定从0至15的常数)	输入变量	INT
n2	输出变量的位指定(只能指定从0至15的常数)	输入变量	INT
ENO	输出状态(TRUE: 正常执行, FALSE: 异常或停止)	输出变量	BOOL
d	输入输出	输入输出变量	INT*1

*1 不能使用位指定的标签及软元件。

功能

■运算处理

将(s)中指定的字第(n1)位的值复制到(d)的第(n2)位中。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

注意事项

将CPY_BIT_OF_INT(_E)通过ST使用的情况下，应创建将返回值代入变量中的程序。

```
//CPY_BIT_OF_INT的返回值在代入变量之后使用。  
g_int2 := CPY_BIT_OF_INT(g_int1,5,3);  
g_bool1 := GET_BIT_OF_INT(g_int2,3);
```

CPY_BIT_OF_INT(_E)的返回值也被用于输入输出，因此请勿直接代入到其它指令、函数、功能块的输入自变量中。

```
//在以下程序中，GET_BIT_OF_INT的第一自变量的值将变为不定值。  
g_bool1 := GET_BIT_OF_INT(CPY_BIT_OF_INT(g_int1,5,3), 3);
```

出错

没有运算错误。

要点

使用标签的位指定时，可以方便地创建与CPY_BIT_OF_INT相同动作的程序。

以下示例中与使用了CPY_BIT_OF_INT的情况下一样，将g_int2的位3(b3)更改为g_int1的位5(b5)的值。

梯形图的情况下



ST的情况下

```
g_int2.3 := g_int1.5;
```

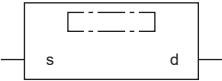
FBD/LD的情况下



31.69 起始数据的获取

GET_BOOL_ADDR、GET_INT_ADDR、GET_WORD_ADDR

将指定数组的起始数据作为BOOL型、INT型或WORD型数据进行输出。

梯形图、FBD/LD	ST
	<pre>d:=GET_BOOL_ADDR(s) d:=GET_INT_ADDR(s); d:=GET_WORD_ADDR(s);</pre>

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
s	输入	输入变量	ANY
d	输出	输出变量	BOOL/INT/WORD

功能

■运算处理

- GET_BOOL_ADDR从(s)中指定的数组开始，将起始数据作为BOOL型数据进行输出。
- GET_INT_ADDR从(s)中指定的数组开始，将起始数据作为INT型数据进行输出。对位长不低于32位的数据类型进行指定时，可获取低位16位。
- GET_WORD_ADDR从(s)中指定的数组开始，将起始数据作为WORD型数据进行输出。对位长不低于32位的数据类型进行指定时，可获取低位16位。

通用函数	输入数据类型	输出数据类型
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

■运算结果

执行运算处理。从(d)输出运算输出值。

出错

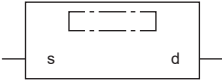
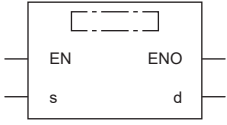
没有运算错误。

32 单数值变量函数

32.1 绝对值

ABS(_E)

输出输入值的绝对值。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=ABS(s); [带EN/ENO] d:=ABS_E(EN, ENO, s);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	ANY_NUM
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_NUM

功能

■运算处理

- 将(s)中输入的INT型/DINT型/REAL/LREAL型数据的绝对值, 从(d)以与(s)相同的数据类型输出。
- 如果将输入值置为A, 运算输出值置为B, 其情况如下所示。

$$B=|A|$$

- 至(s)的输入值为INT型/DINT型/REAL型/LREAL型的数据值。
- (s)的数据类型为INT型且输入了-32768的情况下, 从(d)输出-32768。
- (s)的数据类型为DINT型且输入了-2147483648的情况下, 从(d)输出-2147483648。(不变为运算错误。此外ABS_E时, 通过输出变量ENO输出TRUE。)

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

出错

- (s) 为REAL型的情况下

错误代码(SD0)	内容
3282H	(d) 中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。


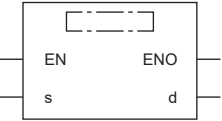
- (s) 为LREAL型的情况下

错误代码(SD0)	内容
3282H	(d) 中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。

32.2 平方根

SQRT(_E)

对输入值的平方根进行运算。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=SQRT(s); [带EN/ENO] d:=SQRT_E(EN, ENO, s);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	ANY_REAL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_REAL

功能

■运算处理

- 对(s)中输入的REAL型/LREAL型数据的平方根进行运算, 将运算结果存储到(d)中。
- 如果将输入值置为A, 运算输出值置为B, 其情况如下所示。

$$B = \sqrt{A}$$

- 至(s)的输入值为REAL型/LREAL型的数据值且在正数的范围内。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


出错

错误代码(SD0)	内容
3285H	输入的值为负数时。

32.3 自然对数运算

LN(_E)

输出输入值的自然对数运算结果。

梯形图、FBD/LD	ST
[无EN/ENO] 	[无EN/ENO] $d := \text{LN}(s);$ [带EN/ENO] $d := \text{LN_E}(\text{EN}, \text{ENO}, s);$

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	ANY_REAL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_REAL

功能

■运算处理

- 对将(s)中输入的REAL型/LREAL型数据的自然对数“e”作为底时的对数进行运算后, 通过(d)输出。
- 将输入值设置为A, 运算输出值设置为B时的情况如下所示。

$$B = \log_e A$$

- 自然对数运算中, 将底“e”作为2.71828进行运算。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


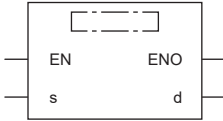
出错

错误代码(SD0)	内容
3285H	输入的值为0或负数时。

32.4 常用对数运算

LOG(_E)

输出输入值的常用对数(以10为底的对数)的运算结果。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=LOG(s); [带EN/ENO] d:=LOG_E(EN, ENO, s);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	ANY_REAL
ENO	输出条件(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_REAL

功能

■运算处理

- 对(s)中输入的REAL型/LREAL型数据的10为底时的对数进行运算后, 从(d)输出。
- 将输入值设置为A, 运算输出值设置为B时的情况如下所示。

$$B = \log_{10} A$$

- 至(s)的输入值为REAL型/LREAL型的数据值。
- (s)中指定的值, 仅正数可以设置。(负数不能进行运算。)
- 运算结果为-0或发生了下溢时, 将运算结果作为0进行输出。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

出错

- (s) 为REAL型的情况下

错误代码 (SD0)	内容
3282H	(s) 中指定的值为-0、非正规化数、非数、 $\pm\infty$ 时。
3285H	(s) 中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> • 指定的值为负的数 • 指定的值为“0”

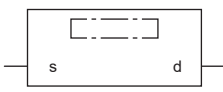
- (s) 为LREAL型的情况下

错误代码 (SD0)	内容
3282H	(s) 中指定的值为-0、非正规化数、非数、 $\pm\infty$ 时。
3285H	(s) 中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> • 指定的值为负的数 • 指定的值为“0”

32.5 指数运算

EXP(_E)

输出输入值的指数运算结果。

梯形图、FBD/LD	ST
<p>[无EN/ENO]</p> 	<p>[无EN/ENO] d:=EXP(s);</p> <p>[带EN/ENO] d:=EXP_E(EN, ENO, s);</p>

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	ANY_REAL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_REAL

功能

■运算处理

- 对(s)中输入的REAL型/LREAL型数据的指数进行运算, 通过(d)输出。
- 将输入值设置为A, 运算输出值设置为B时的情况如下所示。

$$B=e^A$$

- 指数运算中, 将底“e”作为2.71828进行运算。
- 至(s)的输入值为REAL型/LREAL型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


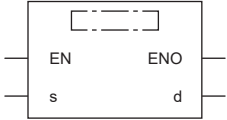
出错

错误代码(SD0)	内容
3282H	(s)中指定的数据为-0、非正规化数、非数、±∞时。
3283H	(d)超出下述范围时。(发生了上溢时) (d) < 2 ¹²⁸

32.6 SIN运算

SIN(_E)

输出输入值的SIN(正弦)值。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=SIN(s); [带EN/ENO] d:=SIN_E(EN, ENO, s);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	ANY_REAL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_REAL

功能

■运算处理

- 对(s)中输入的REAL型数据(角度)的SIN(正弦)值进行运算后, 从(d)输出。
- 将输入值设置为A, 运算输出值设置为B时的情况如下所示。

$B = \sin A$

- 至(s)的输入值(角度)为REAL型的数据值。输入值应以弧度单位(角度 $\times\pi/180$)进行输出。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


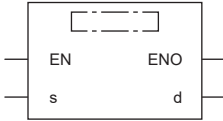
出错

错误代码(SD0)	内容
3282H	(s)中指定的数据为-0时。

32.7 COS运算

COS(_E)

输出输入值的COS(余弦)值。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=COS(s); [带EN/ENO] d:=COS_E(EN, ENO, s);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	ANY_REAL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_REAL

功能

■运算处理

- 对(s)中输入的REAL型数据(角度)的COS(余弦)值进行运算后, 从(d)输出。
- 将输入值设置为A, 运算输出值设置为B时的情况如下所示。

$$B = \cos A$$

- 至(s)的输入值(角度)为REAL型的数据值。输入值, 应以弧度单位(角度 $\times\pi/180$)输出。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

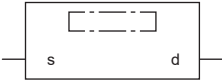
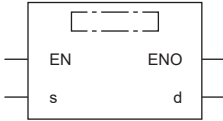
出错

错误代码(SD0)	内容
3282H	(s)中指定的数据为-0时。

32.8 TAN运算

TAN(_E)

输出输入值的TAN(正切)值。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=TAN(s); [带EN/ENO] d:=TAN_E(EN, ENO, s);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	ANY_REAL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_REAL

功能

■运算处理

- 对(s)中输入的REAL型数据(角度)的TAN(正切)值进行运算后,从(d)输出。
- 将输入值设置为A,运算输出值设置为B时的情况如下所示。

B=TAN A

- 输入的值为 $\pi/2$ 弧度、 $(3/2)\pi$ 弧度的情况下,弧度值中也将产生运算误差,不发生错误,因此应加以注意。
- 至(s)的输入值(角度)为REAL型的数据值。输入值,应以弧度单位(角度 $\times\pi/180$)输出。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下,从(d)输出的数据将变为不定值。在此情况下,应进行不使用从(d)输出的数据的程序处理。


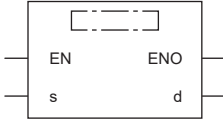
出错

错误代码(SD0)	内容
3282H	(s)中指定的数据为-0时。

32.9 SIN^{-1} 运算

ASIN(_E)

输出输入值的 SIN^{-1} (反正弦)值。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=ASIN(s); [带EN/ENO] d:=ASIN_E(EN, ENO, s);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	ANY_REAL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_REAL

功能

■运算处理

- 对(s)中输入的REAL型数据的 SIN^{-1} (反正弦)值进行运算, 通过(d)输出。
- 将输入值设置为A, 运算输出值设置为B时的情况如下所示。

$$B = \text{SIN}^{-1} A$$

- 至(s)的输入值为REAL型的数据值且在下述所示范围内。

$$\text{ASIN_E}: -1.0 \sim 1.0$$

- 来自于(d)的输出值(角度)为弧度单位(角度 $\times \pi/180$)。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


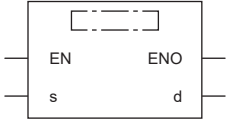
出错

错误代码(SD0)	内容
3282H	(s)中指定的数据为-0时。
3285H	ASIN(_E)中输入的值为-1.0~1.0以外时。

32.10 \cos^{-1} 运算

ACOS(_E)

输出输入值的 \cos^{-1} (反正弦)值。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] $d := \text{ACOS}(s);$ [带EN/ENO] $d := \text{ACOS_E}(EN, ENO, s);$

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	ANY_REAL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_REAL

功能

■运算处理

- 对(s)中输入的REAL型数据的 \cos^{-1} (反余弦)值进行运算, 通过(d)输出。
- 将输入值设置为A, 运算输出值设置为B时的情况如下所示。

$$B = \cos^{-1} A$$

- 至(s)的输入值为REAL型的数据值且在下述所示范围内。

$$\text{ACOS_E}: -1.0 \sim 1.0$$

- 来自于(d)的输出值(角度)为弧度单位(角度 $\times \pi/180$)。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。


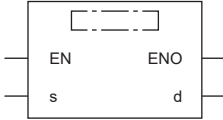
出错

错误代码(SD0)	内容
3282H	(s)中指定的数据为-0时。
3285H	ACOS(_E)中输入的值为-1.0~1.0以外时。

32.11 TAN^{-1} 运算

ATAN(_E)

输出输入值的 TAN^{-1} (反正弦)值。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=ATAN(s); [带EN/ENO] d:=ATAN_E(EN, ENO, s);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	ANY_REAL
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_REAL

功能

■运算处理

- 对(s)中输入的REAL型数据的 TAN^{-1} (反正切)值进行运算, 通过(d)输出。
- 将输入值设置为A, 运算输出值设置为B时的情况如下所示。

$$B = \text{TAN}^{-1} A$$

- 至(s)的输入值为REAL型的数据值且在下述所示范围内。

$$\text{ATAN_E}: \pm 1.17549 \times 10^{-38} \sim \pm 3.40282 \times 10^{38}$$

- 来自于(d)的输出值(角度)为弧度单位(角度 $\times \pi/180$)。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

出错



错误代码(SD0)	内容
3282H	(s)中指定的数据为-0时。

33 算术运算函数

33.1 加法运算

ADD(_E)

输出输入值的和((s1)+(s2)+...+(s28))。

梯形图、FBD/LD*1		ST*1
<p>[无EN/ENO]</p> 	<p>[带EN/ENO]</p> 	<p>[无EN/ENO] d:=ADD(s1, s2);</p> <p>[带EN/ENO] d:=ADD_E(EN, ENO, s1, s2);</p>

*1 输入变量s可以在2~28的范围内进行更改。

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (IN1) ~ s28 (IN28)	输入	输入变量	ANY_NUM
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_NUM

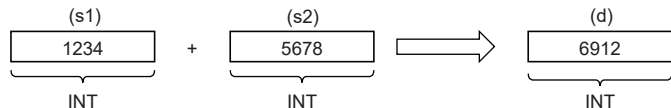
功能

■运算处理

- 进行(s1)~(s28)中输入的INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型数据的加法运算((s1)+(s2)+...+(s28))，将运算结果从(d)以与(s)相同的数据类型进行输出。

例

数据类型为INT型的情况下



- 至(s1)~(s28)的输入值为INT型/DINT型/WORD型/DWORD型//REAL型/LREAL型的数据值。
- 运算结果发生了下溢/上溢时，从(d)按下述方式被输出。

数据类型	内容
INT型	即使发生了下溢及上溢时，也不变为运算错误。此外，ADD_E的情况下，从ENO输出TRUE。 [例1] $32767+2=-32767$ $(7FFFH)+(0002H)=(8001H)$ 由于最高位的位变为1，因此将变为负值。 [例2] $-32767+(-2)=32766$ $(8000H)+(FFFEH)=(7FFEH)$ 由于最高位的位变为0，因此将变为正值。
DINT型	即使发生了下溢及上溢时，也不变为运算错误。此外，ADD_E的情况下，从ENO输出TRUE。 [例1] $2147483647+2=-2147483647$ $(7FFFFFFFH)+(00000002H)=(80000001H)$ 由于最高位的位变为1，因此将变为负值。 [例2] $-2147483648+(-2)=2147483646$ $(80000000H)+(FFFEH)=(7FFFFFFEH)$ 由于最高位的位变为0，因此将变为正值。
WORD型	即使发生了上溢，也不变为运算错误。此外，ADD_E的情况下，从ENO输出TRUE。 [例] $65535+1=0$ $(FFFFH)+(0001H)=(0000H)$
DWORD型	即使发生了上溢，也不变为运算错误。此外，ADD_E的情况下，从ENO输出TRUE。 [例] $4294967295+1=0$ $(FFFFFFFFH)+(00000001H)=(00000000H)$
REAL型	将变为运算错误，输出不定值。
LREAL型	

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件与运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

出错

• (s1) ~ (s28) 为REAL型的情况下

错误代码 (SD0)	内容
3282H	(s1) ~ (s28) 中指定的数据为-0、非正规化数、非数、±∞时。
	(d) 中指定的数据为-0、非正规化数、非数、±∞时。
3283H	(d) 超出下述范围时。(发生了上溢时) (d) < 2 ¹²⁸


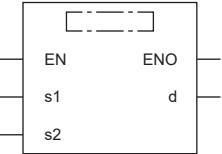
• (s1) ~ (s28) 为LREAL型的情况下

错误代码 (SD0)	内容
3282H	(s1) ~ (s28) 中指定的数据为-0、非正规化数、非数、±∞时。
	(d) 中指定的数据为-0、非正规化数、非数、±∞时。
3283H	(d) 超出下述范围时。(发生了上溢时) (d) < 2 ¹⁰²⁴

33.2 乘法运算

MUL(_E)

输出输入值的积 $((s1) \times (s2) \times \dots \times (s28))$ 。

梯形图、FBD/LD*1		ST*1
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=MUL(s1, s2); [带EN/ENO] d:=MUL_E(EN, ENO, s1, s2);

*1 输入变量s可以在2~28的范围内进行更改。

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (IN1) ~ s28 (IN28)	输入	输入变量	ANY_NUM
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_NUM

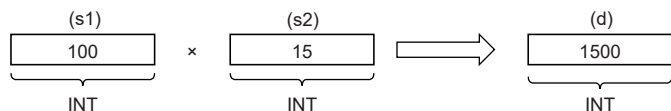
功能

■运算处理

- 进行(s1)~(s28)中输入的INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型数据的乘法运算((s1)×(s2)×...×(s28))，将运算结果以与(s)相同的数据类型从(d)输出。

例

数据类型为INT型的情况下



- 至(s1)~(s28)的输入值为INT型/DINT型/WORD型/DWORD型//REAL型/LREAL型的数据值。
- 运算结果发生了下溢/上溢时，从(d)按下述方式被输出。

数据类型	内容
INT型 WORD型	<ul style="list-style-type: none"> 即使发生了下溢及上溢时，也不变为运算错误。此外，MUL_E的情况下，从ENO输出TRUE。 即使运算结果超过了INT型/WORD型数据范围，也输出INT型/WORD型的数据。(运算结果变为DINT型/DWORD型，但是输出时以删除了高位16位的INT型/WORD型的数据被输出。) 运算结果超过了INT型/WORD型数据范围的情况下，应根据INT_TO_DINT/WORD_TO_DWORD将输入值转换至DINT型/DWORD型数据后，再进行运算。
DINT型 DWORD型	<ul style="list-style-type: none"> 即使发生了下溢及上溢时，也不变为运算错误。此外，MUL_E的情况下，从ENO输出TRUE。 即使运算结果超过了DINT型/DWORD型数据范围，也输出DINT型/DWORD型的数据。(运算结果变为64位数据，但是输出时以删除了高位32位的DINT型/DWORD型的数据被输出。) 运算结果超过了DINT型/DWORD型数据范围的情况下，应根据DINT_TO_REAL将输入值转换至REAL型数据后，再进行运算。
REAL型 LREAL型	将变为运算错误，输出不定值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件与运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

要点

运算结果超过了数据类型范围的情况下，应在转换输入值的数据类型之后再执行运算。

出错

- (s1)~(s28)为REAL型的情况下

错误代码(SD0)	内容
3282H	(s1)~(s28)中指定的数据为-0、非正规化数、非数、±∞时。
3283H	(d)超出下述范围时。(发生了上溢时) $ d < 2^{128}$


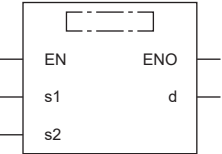
- (s1)~(s28)为LREAL型的情况下

错误代码(SD0)	内容
3282H	(s1)~(s28)中指定的数据为-0、非正规化数、非数、±∞时。
3283H	(d)超出下述范围时。(发生了上溢时) $ d < 2^{1024}$

33.3 减法运算

SUB(_E)

输出输入值的差((s1)-(s2))。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] $d := \text{SUB}(s1, s2);$ [带EN/ENO] $d := \text{SUB_E}(\text{EN}, \text{ENO}, s1, s2);$

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1(IN1)、s2(IN2)	输入	输入变量	ANY_NUM
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_NUM

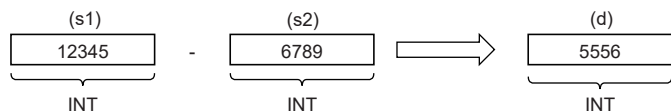
功能

■运算处理

- 进行(s1)、(s2)中输入的INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型数据的减法运算((s1)-(s2))，将运算结果从(d)以与(s)相同的数据类型进行输出。

例

数据类型为INT型的情况下



- 至(s1)、(s2)的输入值为INT型/DINT型/WORD型/DWORD型//REAL型/LREAL型的数据值。
- 运算结果发生了下溢/上溢时，从(d)按下述方式被输出。

数据类型	内容
INT型	即使发生了下溢及上溢时，也不变为运算错误。此外，SUB_E的情况下，从ENO输出TRUE。 [例1] $32767 - (-2) = -32767$ $(7FFFH) - (FFFEH) = (8001H)$ 由于最高位的位变为1，因此将变为负值。 [例2] $-32767 - 2 = 32766$ $(8000H) - (0002H) = (7FFE H)$ 由于最高位的位变为0，因此将变为正值。
DINT型	即使发生了下溢及上溢时，也不变为运算错误。此外，SUB_E的情况下，从ENO输出TRUE。 [例1] $2147483647 - (-2) = -2147483647$ $(7FFFFFFFH) - (0000FFFEH) = (80000001H)$ 由于最高位的位变为1，因此将变为负值。 [例2] $-2147483648 - 2 = 2147483646$ $(80000000H) - (00000002H) = (7FFFFFFEH)$ 由于最高位的位变为0，因此将变为正值。
WORD型	即使发生了下溢，也不变为运算错误。此外，ADD_E的情况下，从ENO输出TRUE。 [例] $0 - 1 = 65535$ $(0000H) - (0001H) = (FFFFH)$
DWORD型	即使发生了下溢，也不变为运算错误。此外，ADD_E的情况下，从ENO输出TRUE。 [例] $0 - 1 = 4294967295$ $(00000000H) - (00000001H) = (FFFFFFFFH)$
REAL型	将变为运算错误，输出不定值。
LREAL型	

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件与运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

出错

- (s1)、(s2)为REAL型的情况下

错误代码(SD0)	内容
3282H	(s1)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
	(s2)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
	(d)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d)超出下述范围时。(发生了上溢时) $ (d) < 2^{128}$

- (s1)、(s2)为LREAL型的情况下

错误代码(SD0)	内容
3282H	(s1)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
	(s2)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
	(d)中指定的数据为-0、非正规化数、非数、 $\pm\infty$ 时。
3283H	(d)超出下述范围时。(发生了上溢时) $ (d) < 2^{1024}$

33.4 除法运算

DIV(_E)

输出输入值的商((s1)÷(s2))。

梯形图、FBD/LD		ST
<p>[无EN/ENO]</p>	<p>[带EN/ENO]</p>	<p>[无EN/ENO] d:=DIV (s1, s2) ;</p> <p>[带EN/ENO] d:=DIV_E (EN, ENO, s1, s2) ;</p>

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (IN1)	被除数	输入变量	ANY_NUM
s2 (IN2)	除数	输入变量	ANY_NUM
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_NUM

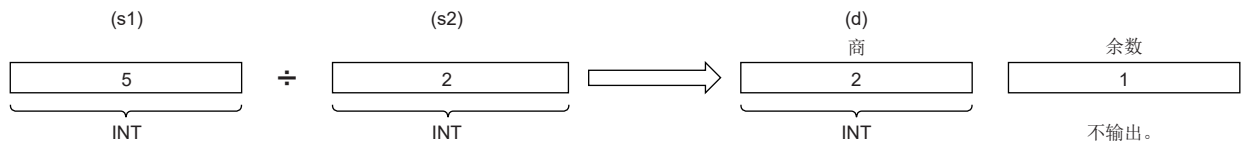
功能

■运算处理

- 进行(s1)、(s2)中输入的INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型数据的除法运算((s1)÷(s2))，将运算结果的商从(d)以及与(s)相同的数据类型进行输出。

例

数据类型为INT型的情况下



- 至(s1)、(s2)的输入值为INT型/DINT型/WORD型/DWORD型//REAL型/LREAL型的数据值。(但是，输入至(s2)的值为0以外。)

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件与运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

出错

- (s1)、(s2)为INT型/WORD型的情况下

错误代码(SD0)	内容
3280H	(s2)中指定的值(除数)为0时。

- (s1)、(s2)为DINT型/DWORD型的情况下

错误代码(SD0)	内容
3280H	(s2)中指定的值(除数)为0时。

- (s1)、(s2)为REAL型的情况下

错误代码(SD0)	内容
3280H	(s2)中指定的值(除数)为0时。
3282H	(s1)中指定的数据为-0、非正规化数、非数、±∞时。 (s2)中指定的数据为-0、非正规化数、非数、±∞时。
3283H	(d)超出下述范围时。(发生了上溢时) $ (d) < 2^{128}$


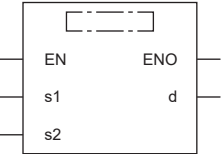
- (s1)、(s2)为LREAL型的情况下

错误代码(SD0)	内容
3280H	(s2)中指定的值(除数)为0时。
3282H	(s1)中指定的数据为-0、非正规化数、非数、±∞时。 (s2)中指定的数据为-0、非正规化数、非数、±∞时。
3283H	(d)超出下述范围时。(发生了上溢时) $ (d) < 2^{1024}$

33.5 余数

MOD(_E)

输出输入值的余数((s1)÷(s2))。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] 作为运算符记述。(☞ 92页 运算符) [带EN/ENO] $d := MOD_E(EN, ENO, s1, s2);$

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (IN1)	被除数	输入变量	ANY_INT
s2 (IN2)	除数	输入变量	ANY_INT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_INT

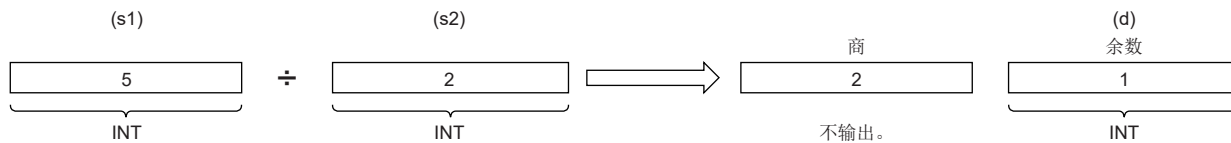
功能

■运算处理

- 进行(s1)、(s2)中输入的INT型/DINT型/WORD型/DWORD型数据的除法运算 $((s1) \div (s2))$ ，将运算结果的余数从(d) 以与(s)相同的数据类型进行输出。

例

数据类型为INT型的情况下



- 至(s1)、(s2)的输入值为INT型/DINT型/WORD型/DWORD型的数据值。(但是，输入至(s2)的值为0以外。)

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件与运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

出错

- (s1)、(s2)为INT型/WORD型的情况下

错误代码(SD0)	内容
3280H	(s2)中指定的值(除数)为0时。

- (s1)、(s2)为DINT型/DWORD型的情况下

错误代码(SD0)	内容
3280H	(s2)中指定的值(除数)为0时。

33.6 幂

EXPT(_E)

输出输入值的幂。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=EXPT(s1, s2); [带EN/ENO] d:=EXPT_E(EN, ENO, s1, s2);

设置数据

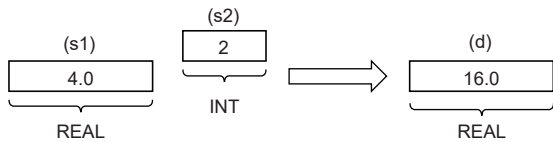
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (IN1)	基数	输入变量	ANY_REAL
s2 (IN2)	指数	输入变量	ANY_NUM
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_REAL

功能

■运算处理

将(s1)中输入的REAL型/LREAL型数据以(s2)中指定的INT型/DINT型/REAL型/LREAL型进行幂运算，从(d)输出运算结果。



■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件与运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

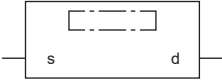
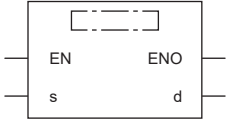
出错

错误代码(SD0)	内容
3282H	(s1)或(s2)中指定的数据为-0、非正规化数、非数、±∞时。

33.7 代入

MOVE(_E)

输出输入值的代入。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] <code>d:=MOVE(s);</code> [带EN/ENO] <code>d:=MOVE_E(EN, ENO, s);</code>

设置数据

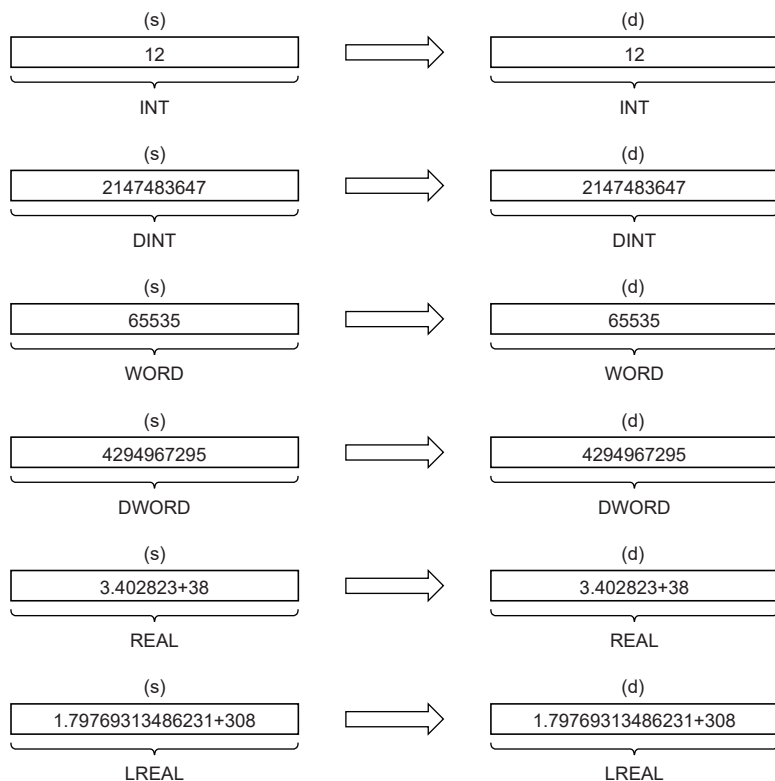
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	ANY
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY

功能

■运算处理

- 将(s)中指定的变量的值代入至(d)中指定的变量中。
- 可以对(s)、(d)指定BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型/结构体型/数组型。此外，(s)与(d)仅可以指定相同的数据类型。



■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件与运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

出错

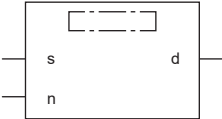
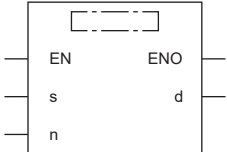
错误代码(SD0)	内容
2820H	(s)中指定的软元件编号及其以后，软元件/标签存储器的各设置区域之间不存在00H时。
3285H	(s)的字符串超过了16383字符时。
3286H	(d)中指定的软元件编号及其以后，相应软元件/标签存储器的各设置区域的最终编号为止的点数中，无法存储指定的全部字符串时。

34 位移函数

34.1 n位左移

SHL(_E)

将输入值左移(n)位数后输出。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=SHL(s,n); [带EN/ENO] d:=SHL_E(EN,ENO,s,n);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	ANY_BIT
n(N)	移位位数指定	输入变量	ANY_BIT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_BIT

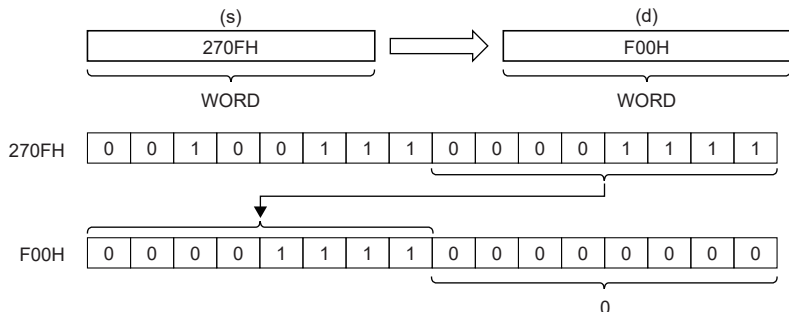
功能

■运算处理

- 将(s)中输入的16位数据/32位数据(WORD型/DWORD型)左移(n)位后,从(d)以与(s)相同的数据类型进行输出。
- 左移的(n)位数为输入至(n)的值。

例

(s)的数据为16位数据(WORD型),至(n)的输入值为8的情况下



- 从最低位开始的(n)位将变为0。
- 至(s)的输入值为16位数据/32位数据(WORD型/DWORD型)的数据。
- 至(n)的输入值(移位位数指定)在下列范围内。

(s)的数据为16位数据(WORD型)的情况下	(s)的数据为32位数据(DWORD型)的情况下
(n)的输入值为0~15的范围内。 使用至(n)的输入值的低4位的数据。 [例] 输入值为6的情况下: 6 输入值为22的情况下: 6	(n)的输入值为0~31的范围内。 使用至(n)的输入值的低5位的数据。 [例] 输入值为6的情况下: 6 输入值为22的情况下: 22

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下,从(d)输出的数据将变为不定值。在此情况下,应进行不使用从(d)输出的数据的程序处理。

出错

没有运算错误。

34.2 n位右移

SHR(_E)

将输入值右移(n)位数后输出。

梯形图、FBD/LD		ST
[无EN/ENO]		[无EN/ENO] $d := \text{SHR}(s, n);$ [带EN/ENO] $d := \text{SHR_E}(\text{EN}, \text{ENO}, s, n);$

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	ANY_BIT
n(N)	移位位数指定	输入变量	ANY_BIT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_BIT

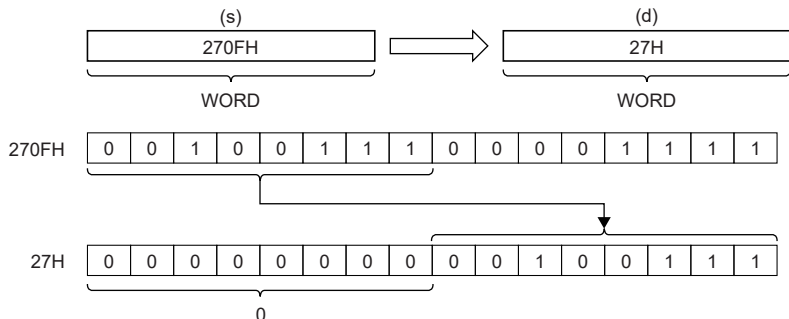
功能

■运算处理

- 将(s)中输入的16位数据/32位数据(WORD型/DWORD型)右移(n)位后,从(d)以与(s)相同的数据类型进行输出。
- 右移的(n)位数时输入至(n)的值。

例

(s)的数据为16位数据(WORD型),至(n)的输入值为8的情况下



- 从最高位开始的(n)位将变为0。
- 至(s)的输入值为16位数据/32位数据(WORD型/DWORD型)的数据。
- 至(n)的输入值(移位位数指定)在下述范围内。

(s)的数据为16位数据(WORD型)的情况下	(s)的数据为32位数据(DWORD型)的情况下
(n)的输入值为0~15的范围内。 使用至(n)的输入值的低4位的数据。 [例] 输入值为6的情况下: 6 输入值为22的情况下: 6	(n)的输入值为0~31的范围内。 使用至(n)的输入值的低5位的数据。 [例] 输入值为6的情况下: 6 输入值为22的情况下: 22

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下,从(d)输出的数据将变为不定值。在此情况下,应进行不使用从(d)输出的数据的程序处理。

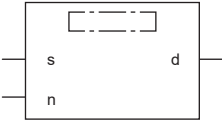
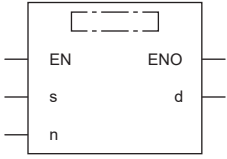
出错

没有运算错误。

34.3 n位左旋

ROL(_E)

将输入值左旋(n)位数后输出。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] <code>d:=ROL(s,n);</code> [带EN/ENO] <code>d:=ROL_E(EN,ENO,s,n);</code>

设置数据

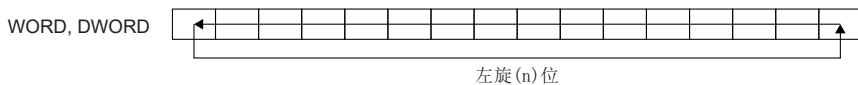
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	ANY_BIT
n(N)	移位位数指定	输入变量	ANY_BIT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_BIT

功能

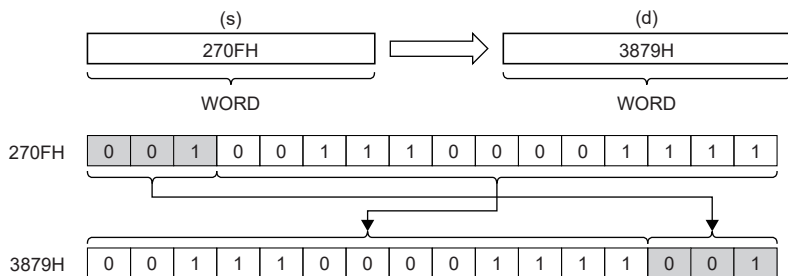
■运算处理

- 对(s)中输入的16位数据/32位数据(WORD型/DWORD型)向左(n)位回转(旋转)后,从(d)以与(s)相同的数据类型进行输出。
- 向左回转的(n)位数为输入至(n)的值。



例

(s)的数据类型为16位数据(WORD型),至(n)的输入值为3的情况下。(向左回转3位。)



- 至(s)的输入值为16位数据/32位数据(WORD型/DWORD型)的数据。
- 至(n)的输入值(移位位数指定)在下列范围内。

(s)的数据为16位数据(WORD型)的情况下	(s)的数据为32位数据(DWORD型)的情况下
(n)的输入值为0~15的范围内。 使用至(n)的输入值的低4位的数据。 [例] 输入值为6的情况下: 6 输入值为22的情况下: 6	(n)的输入值为0~31的范围内。 使用至(n)的输入值的低5位的数据。 [例] 输入值为6的情况下: 6 输入值为22的情况下: 22

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下,从(d)输出的数据将变为不定值。在此情况下,应进行不使用从(d)输出的数据的程序处理。

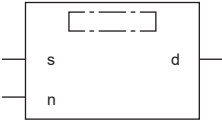
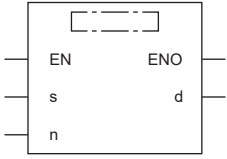
出错

没有运算错误。

34.4 n位右旋

ROR(_E)

将输入值右旋(n)位数后输出。

梯形图、FBD/LD		ST
[无EN/ENO]	[带EN/ENO]	[无EN/ENO] d:=ROR(s,n); [带EN/ENO] d:=ROR_E(EN,ENO,s,n);
		

设置数据

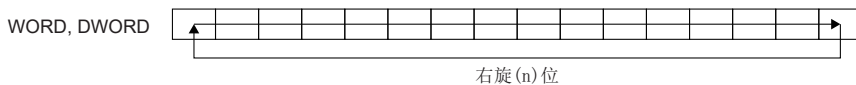
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	ANY_BIT
n(N)	移位位数指定	输入变量	ANY_BIT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_BIT

功能

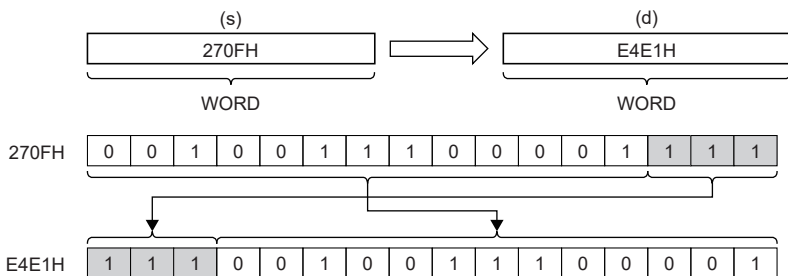
■运算处理

- 对(s)中输入的16位数据/32位数据(WORD型/DWORD型)向右(n)位回转(旋转)后,从(d)以与(s)相同的数据类型进行输出。
- 向右旋转的(n)位数为输入至(n)的值。



例

(s)的数据为16位数据(WORD型),至(n)的输入值为3的情况下。(向右回转3位。)



- 至(s)的输入值为16位数据/32位数据(WORD型/DWORD型)的数据。
- 至(n)的输入值(移位位数指定)在下述范围内。

(s)的数据为16位数据(WORD型)的情况下	(s)的数据为32位数据(DWORD型)的情况下
(n)的输入值为0~15的范围内。 使用至(n)的输入值的低4位的数据。 [例] 输入值为6的情况下: 6 输入值为22的情况下: 6	(n)的输入值为0~31的范围内。 使用至(n)的输入值的低5位的数据。 [例] 输入值为6的情况下: 6 输入值为22的情况下: 22

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下,从(d)输出的数据将变为不定值。在此情况下,应进行不使用从(d)输出的数据的程序处理。

出错

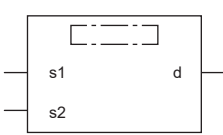
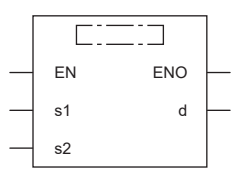
没有运算错误。

35 位型布尔函数

35.1 逻辑与、逻辑或、异或

AND(_E)、OR(_E)、XOR(_E)

- AND(_E)：输出输入值的逻辑与。
- OR(_E)：输出输入值的逻辑或。
- XOR(_E)：输出输入值的异或。

梯形图、FBD/LD*1		ST*1
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] 作为运算符记述。(☞ 92页 运算符) [带EN/ENO] $d := \text{AND_E}(EN, ENO, s1, s2);$ $d := \text{OR_E}(EN, ENO, s1, s2);$ $d := \text{XOR_E}(EN, ENO, s1, s2);$

*1 输入变量s，可以在2~28的范围内进行更改。

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1(IN1)~s28(IN28)	输入	输入变量	ANY_BIT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_BIT

功能

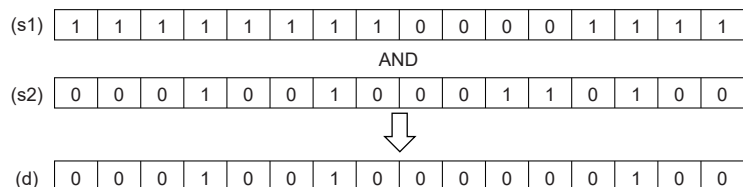
■运算处理

1. AND(_E)

对(s1)~(s28)中输入的BOOL型/WORD型/DWORD型数据按各个位进行逻辑与运算，将运算结果从(d)以与(s)相同的数据类型进行输出。

例

数据类型为WORD型的情况下

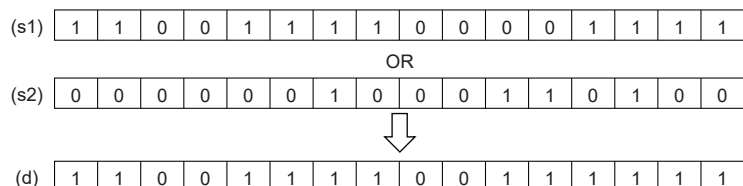


2. OR(_E)

对(s1)~(s28)中输入的BOOL型/WORD型/DWORD型数据按各个位进行逻辑或运算，将运算结果从(d)以与(s)相同的数据类型进行输出。

例

数据类型为WORD型的情况下

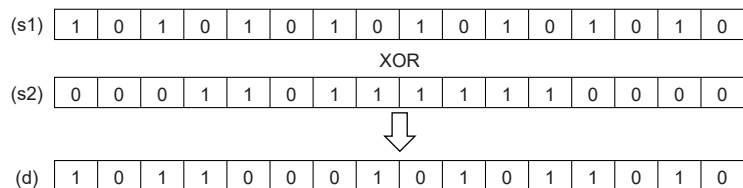


3. XOR(_E)

对(s1)~(s28)中输入的BOOL型/WORD型/DWORD型数据按各个位进行异或运算，将运算结果从(d)以与(s)相同的数据类型进行输出。

例

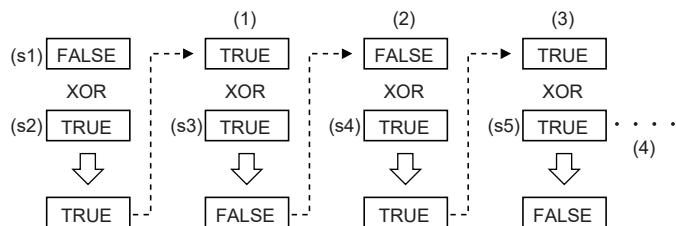
数据类型为WORD型的情况下



有3个及以上(s)的情况下，对于将(s1)与(s2)进行了XOR的结果，对(s3)进行XOR。此外，有(s4)的情况下，对于将(s3)进行了XOR的结果，对(s4)进行XOR。以后仅(s5)、(s6)与(s)的数进行XOR。

例

数据类型为BOOL型的情况下



(1) s的数为3个的情况下

(2) s的数为4个的情况下

(3) s的数为5个的情况下

(4) 以后仅s的数进行XOR。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件与运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

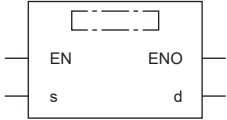
出错

没有运算错误。

35.2 逻辑否

NOT (_E)

输出输入值的逻辑否。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] 作为运算符记述。(☞ 92页 运算符) [带EN/ENO] d:=NOT_E(EN, ENO, s);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)*1	输入	输入变量	ANY_BIT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_BIT

*1 不能使用DX。

功能

■运算处理

- 对(s)中输入的BOOL型/WORD型/DWORD型数据按各个位进行逻辑否运算，将运算结果从(d)以与(s)相同的数据类型进行输出。

例

数据类型为WORD型的情况下

(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

- 至(s)的输入值为BOOL型/WORD型/DWORD型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件与运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

出错

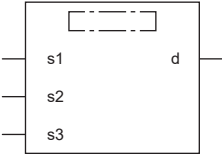
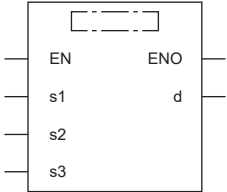
没有运算错误。

36 选择函数

36.1 选择值

SEL(_E)

输出选择的输入值。

梯形图、FBD/LD		ST
<p>[无EN/ENO]</p> 	<p>[带EN/ENO]</p> 	<p>[无EN/ENO] <code>d:=SEL(s1, s2, s3);</code> [带EN/ENO] <code>d:=SEL_E(EN, ENO, s1, s2, s3);</code></p>

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (G)	输出条件(TRUE: s3输出, FALSE: s2输出)	输入变量	BOOL
s2 (IN0)	输入	输入变量	ANY
s3 (IN1)			
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY

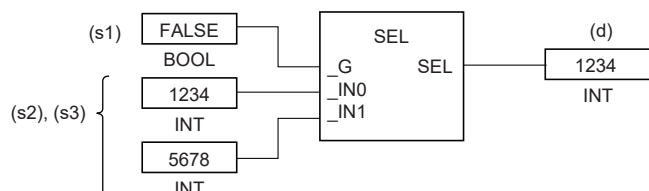
功能

■运算处理

- 按照输入至(s1)的输入值，将输入至(s2)、(s3)的值的某个，从(d)以与(s2)、(s3)相同的数据类型进行输出。
- (s1)的输入值为FALSE(=0)的情况下，从(d)输出(s2)的输入值。
- (s1)的输入值为TRUE(=1)的情况下，从(d)输出(s3)的输入值。

例

(s2)、(s3)的数据类型为INT型的情况下(自变量名的(s2)、(s3)对应于(s1)的位值(0或1)。)



- (s1)的输入值为BOOL型的数据值。
- 至(s2)、(s3)的输入值为BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型/结构体型/数组型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

出错

- (s2)、(s3)为STRING型的情况下

错误代码(SD0)	内容
2820H	在(s2)中指定的标签或软元件编号开始，至相应软元件的最终软元件编号为止之间，未设置00H时。 在(s3)中指定的标签或软元件编号开始，至相应软元件的最终软元件编号为止之间，未设置00H时。
3286H	在(d)中指定的标签或软元件编号开始至相应软元件的最终软元件编号为止的点数中，指定的字符串无法全部存储时。

36.2 最大值、最小值选择

MAX(_E)、MIN(_E)

- MAX(_E)：输出输入值的最大值。
- MIN(_E)：输出输入值的最小值。

梯形图、FBD/LD*1		ST
[无EN/ENO]	[带EN/ENO]	[无EN/ENO] d:=MAX(s1, s2); d:=MIN(s1, s2); [带EN/ENO] d:=MAX_E(EN, ENO, s1, s2); d:=MIN_E(EN, ENO, s1, s2);

*1 输入变量s可以在2~28的范围内进行更改。

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (IN1) ~ s28 (IN28)	输入	输入变量	ANY_ELEMENTARY
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_ELEMENTARY

功能

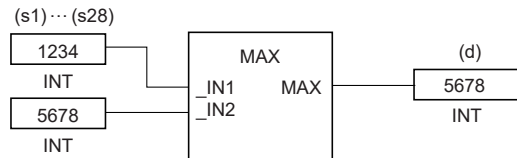
■运算处理

• MAX(_E)

将(s1)~(s28)中输入的BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型数据的最大值从(d)以与(s)相同的数据类型进行输出。

例

数据类型为INT型的情况下

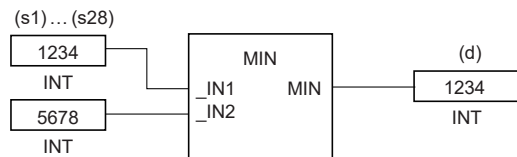


• MIN(_E)

将(s1)~(s28)中输入的BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型数据的最小值从(d)以与(s)相同的数据类型进行输出。

例

数据类型为INT型的情况下



• 至(s1)~(s28)的输入值为BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型的数据值。

• 进行字符串比较的情况下，比较条件如下所示。

- | | |
|-----------|---------------------------|
| 一致的条件： | • 所有的字符串一致的情况下 |
| 较大字符串的条件： | • 不同字符串的情况下，字符代码较大的字符串 |
| | • 字符串的长度不同的情况下，字符串较长的字符串 |
| 较小字符串的条件： | • 不同字符串的情况下，字符代码较小的字符串 |
| | • 字符串的长度不相同的情况下，字符串较短的字符串 |

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

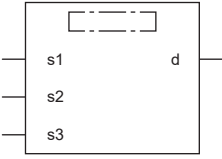
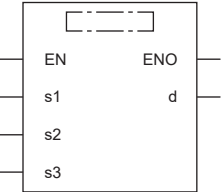
出错

错误代码(SD0)	内容
2820H	(s1)~(s28)中指定的软元件编号及其以后，软元件/标签存储器的各设置区域之间不存在00H时。
3285H	(s1)~(s28)的字符串超过了16383字符时。
3286H	(d)中指定的软元件编号及其以后，相应软元件/标签存储器的各设置区域的最终编号为止的点数中无法存储指定的全部字符串时。

36.3 上下限限位控制

LIMIT(_E)

输出上下限限位控制的输入值。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] $d := \text{LIMIT}(s1, s2, s3);$ [带EN/ENO] $d := \text{LIMIT_E}(\text{EN}, \text{ENO}, s1, s2, s3);$

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (MN)*1	下限限值(最小输出极限值)	输入变量	ANY_ELEMENTARY
s2 (IN)*1	通过上下限限位控制控制的输入值	输入变量	ANY_ELEMENTARY
s3 (MX)*1	上限限值(最大输出极限值)	输入变量	ANY_ELEMENTARY
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY_ELEMENTARY

*1 不能使用DX。

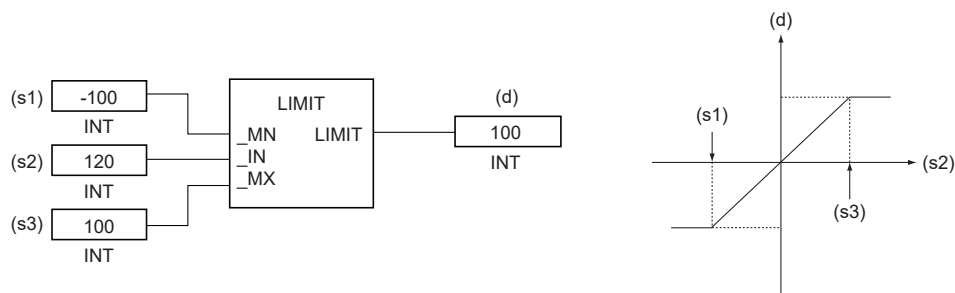
功能

■运算处理

- 按照输入至(s1)、(s2)、(s3)的BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型数据，从(d)以与(s1)、(s2)、(s3)相同的数据类型进行输出。
- (s2)的输入值>(s3)的输入值的情况下，从(d)输出(s3)的输入值。
- (s2)的输入值<(s1)的输入值的情况下，从(d)输出(s1)的输入值。
- (s1)的输入值≤(s2)的输入值≤(s3)的输入值的情况下，从(d)输出(s2)的输入值。

例

数据类型为INT型的情况下



- 至(s1)、(s2)、(s3)的输入值为BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型的数据值。(但是，(s1)的输入值<(s3)的输入值)
- 进行字符串比较的情况下，比较条件如下所示。
 - 一致的条件：
 - 所有的字符串一致的情况下
 - 较大字符串的条件：
 - 不同字符串的情况下，字符代码较大的字符串
 - 字符串的长度不同的情况下，字符串较长的字符串
 - 较小字符串的条件：
 - 不同字符串的情况下，字符代码较小的字符串
 - 字符串的长度不相同的情况下，字符串较短的字符串

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

出错

- (s1)、(s2)、(s3)为INT型/WORD型的情况下

错误代码(SD0)	内容
3285H	(s1)中指定的下限限值大于(s2)中指定的上限限值时。

- (s1)、(s2)、(s3)为DINT型/DWORD型/TIME型的情况下

错误代码(SD0)	内容
3285H	(s1)中指定的下限限值大于(s2)中指定的上限限值时。

- (s1)、(s2)、(s3)为BOOL型的情况下

错误代码(SD0)	内容
3285H	(s1)中指定的下限限值大于(s3)中指定的上限限值时。

- (s1)、(s2)、(s3)为REAL型的情况下

错误代码(SD0)	内容
3282H	(s1)内容为下述范围以外时。 $-2^{128} < (s1) \leq -2^{-126}$ 、 0 、 $2^{-126} \leq (s1) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38、0、E1.17549435-38~E3.40282347+38) (s1)中指定的数据为-0、非正规化数、非数、±∞时。
	(s2)内容为下述范围以外时。 $-2^{128} < (s2) \leq -2^{-126}$ 、 0 、 $2^{-126} \leq (s2) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38、0、E1.17549435-38~E3.40282347+38) (s2)中指定的数据为-0、非正规化数、非数、±∞时。
	(s3)的内容为下述范围外时。 $-2^{128} < (s3) \leq -2^{-126}$ 、 0 、 $2^{-126} \leq (s3) < 2^{128}$ (E-3.40282347+38~E-1.17549435-38、0、E1.17549435-38~E3.40282347+38) (s3)中指定的数据为-0、非正规化数、非数、±∞时。
3285H	(s1)中指定的下限限值大于(s3)中指定的上限限值时。

- (s1)、(s2)、(s3)为LREAL型的情况下

错误代码(SD0)	内容
3282H	(s1)内容为下述范围以外时。 $-2^{1024} < (s1) \leq -2^{-1022}$ 、 0 、 $2^{-1022} \leq (s1) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308、0、E2.2250738585072014-308~E1.7976931348623157+308) (s1)中指定的数据为-0、非正规化数、非数、±∞时。
	(s2)内容为下述范围以外时。 $-2^{1024} < (s2) \leq -2^{-1022}$ 、 0 、 $2^{-1022} \leq (s2) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308、0、E2.2250738585072014-308~E1.7976931348623157+308) (s2)中指定的数据为-0、非正规化数、非数、±∞时。
	(s3)的内容为下述范围外时。 $-2^{1024} < (s3) \leq -2^{-1022}$ 、 0 、 $2^{-1022} \leq (s3) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308、0、E2.2250738585072014-308~E1.7976931348623157+308) (s3)中指定的数据为-0、非正规化数、非数、±∞时。
3285H	(s1)中指定的下限限值大于(s3)中指定的上限限值时。

- (s1)、(s2)、(s3)为STRING型的情况下

错误代码(SD0)	内容
2820H	在(s1)、(s2)、(s3)中指定标签或指定的软件编号开始，至相应元件的最终软件编号为止之间，未设置00H时。
3285H	(s1)中指定的下限限值大于(s3)中指定的上限限值时。 (s1)、(s2)、(s3)中指定的字符串的字符数超过了16383字符时。
3286H	在(d)中指定的标签或指定的软件编号开始至相应元件的最终软件编号为止的点数中，指定的字符串无法全部存储时。

36.4 多路复用器

MUX(_E)

输出多个输入值中之一。

梯形图、FBD/LD*1		ST*1
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=MUX(n, s1, s2); [带EN/ENO] d:=MUX_E(EN, ENO, n, s1, s2);

*1 输入变量s可以在2~28的范围内进行更改。

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
n(K)	输出值选择	输入变量	ANY_INT
s1(IN0)~s28(IN27)	输入	输入变量	ANY
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANY

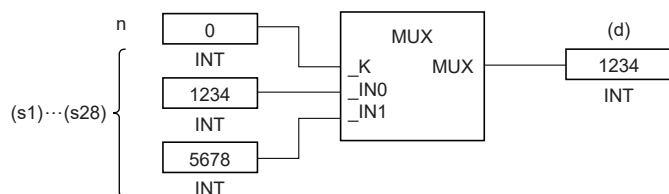
功能

■运算处理

- 按照输入至(n)的输入值，将输入至(s1)~(s28)的值的某个从(d)以与(s)相同的数据类型进行输出。
- (n)的输入值为0的情况下，从(d)输出输入至(s1)的值。
- (n)的输入值为(n)-1的情况下，从(d)输出输入至(sn)的值。

例

数据类型为INT型的情况下



- (n)中输入了(s)的针数范围外的情况下，从(d)输出不定值。(不变为运算错误。此外MUX_EN时，通过(ENO)输出FALSE。)
- 至(n)的输入值为INT型/DINT型/WORD型/DWORD型的数据值且在0~27的范围内。(但是，在(s)的针数范围内。)
- 至(s)的输入值为BOOL型/INT型/DINT型/WORD型/DWORD型/REAL型/LREAL型/STRING型/TIME型/结构体型/数组型的数据值。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

出错

错误代码(SD0)	内容
2820H	(s1)~(s28)中指定的软件元件编号及其以后，软元件/标签存储器的各设置区域之间不存在00H时。
3285H	(s1)~(s28)的字符串超过了16383字符时。
3286H	(d)中指定的软件元件编号及其以后，相应软元件/标签存储器的各设置区域的最终编号为止的点数中无法存储指定的全部字符串时。

37 比较函数

37.1 比较

GT(_E)、GE(_E)、EQ(_E)、LE(_E)、LT(_E)

输出输入值的数据比较结果。

梯形图、FBD/LD*1		ST*1
<p>[无EN/ENO]</p>	<p>[带EN/ENO]</p>	<p>[无EN/ENO]</p> <pre>d:=GT(s1, s2); d:=GE(s1, s2); d:=EQ(s1, s2); d:=LE(s1, s2); d:=LT(s1, s2);</pre> <p>[带EN/ENO]</p> <pre>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);</pre>

*1 输入变量s可以在2~28的范围内进行更改。

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (IN1) ~ s28 (IN28)*1	输入	输入变量	ANY_ELEMENTARY
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出(TRUE: 真值, FALSE: 假值)	输出变量	BOOL

*1 不能使用DX。

功能

■运算处理

- 进行输入至(s)的输入值的比较运算后,将运算结果从(d)以BOOL型进行输出。
- GT(E):进行[(s1)>(s2)]&[(s2)>(s3)]&…&[(s)_(n-1)>(s)_(n)]的比较。
 - 所有的(s)_(n-1)>(s)_(n)时,输出TRUE。
 - 某一个为(s)_(n-1)≤(s)_(n)时,输出FALSE。
- GE(E):进行[(s1)≥(s2)]&[(s2)≥(s3)]&…&[(s)_(n-1)≥(s)_(n)]的比较。
 - 所有的(s)_(n-1)≥(s)_(n)时,输出TRUE。
 - 某一个为(s)_(n-1)<(s)_(n)时,输出FALSE。
- EQ(E):进行[(s1)=(s2)]&[(s2)=(s3)]&…&[(s)_(n-1)=(s)_(n)]的比较。
 - 所有的(s)_(n-1)=(s)_(n)时,输出TRUE。
 - 某一个为(s)_(n-1)≠(s)_(n)时,输出FALSE。
- LE(E):进行[(s1)≤(s2)]&[(s2)≤(s3)]&…&[(s)_(n-1)≤(s)_(n)]的比较。
 - 所有的(s)_(n-1)≤(s)_(n)时,输出TRUE。
 - 某一个为(s)_(n-1)>(s)_(n)时,输出FALSE。
- LT(E):进行[(s1)<(s2)]&[(s2)<(s3)]&…&[(s)_(n-1)<(s)_(n)]的比较。
 - 所有的(s)_(n-1)<(s)_(n)时,输出TRUE。
 - 某一个为(s)_(n-1)≥(s)_(n)时,输出FALSE。
- 至(s)的输入值为INT型/DINT型/REAL型/LREAL型/BOOL型/WORD型/DWORD型/TIME型/STRING型的数据值。不能指定WSTRING型的Unicode字符串。
- 进行字符串比较的情况下,比较条件如下所示。
 - 一致的条件:
 - 所有的字符串一致的情况下
 - 较大字符串的条件:
 - 不同字符串的情况下,字符代码较大的字符串
 - 字符串的长度不同的情况下,字符串较长的字符串
 - 较小字符串的条件:
 - 不同字符串的情况下,字符代码较小的字符串
 - 字符串的长度不相同的情况下,字符串较短的字符串

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件与运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下,从(d)输出的数据将变为不定值。在此情况下,应进行不使用从(d)输出的数据的程序处理。

出错

错误代码(SD0)	内容
2820H	(s1)~(s28)中指定的软元件编号及其以后,软元件/标签存储器的各设置区域之间不存在00H时。
3285H	(s1)~(s28)的字符串超过了16383字符时。
3286H	(d)中指定的软元件编号及其以后,相应软元件/标签存储器的各设置区域的最终编号为止的点数中无法存储指定的全部字符串时。

37.2 比较

NE(_E)

输出输入值的数据比较结果。

梯形图、FBD/LD		ST
[无EN/ENO]	[带EN/ENO]	[无EN/ENO] d:=NE(s1, s2); [带EN/ENO] d:=NE_E(EN, ENO, s1, s2);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1、s2	输入	输入变量	ANY_ELEMENTARY
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出(TRUE: 真值, FALSE: 假值)	输出变量	BOOL

功能

■运算处理

- 进行输入至(s)的输入值的比较运算后, 将运算结果从(d)以BOOL型进行输出。
 - NE(_E): 进行[(s1)≠(s2)]的比较。
 - (s1)≠(s2)时输出TRUE。
 - (s1)=(s2)时输出FALSE。
- 至(s)的输入值为INT型/DINT型/REAL型/LREAL型/BOOL型/WORD型/DWORD型/TIME型/STRING型的数据值。不能指定WSTRING型的Unicode字符串。
- 进行字符串比较的情况下, 比较条件如下所示。
 - 一致的条件:
 - 所有的字符串一致的情况下
 - 较大字符串的条件:
 - 不同字符串的情况下, 字符代码较大的字符串
 - 字符串的长度不同的情况下, 字符串较长的字符串
 - 较小字符串的条件:
 - 不同字符串的情况下, 字符代码较小的字符串
 - 字符串的长度不相同的情况下, 字符串较短的字符串

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件与运算结果如下所示。

执行条件	运算结果	
	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

出错

错误代码 (SD0)	内容
2820H	(s) 中指定的软元件编号及其以后, 软元件/标签存储器的各设置区域之间不存在00H时。
3285H	(s) 的字符串超过了16383字符时。
3286H	(d) 中指定的软元件编号及其以后, 相应软元件/标签存储器的各设置区域的最终编号为止的点数中无法存储指定的全部字符串时。

38.1 字符串的长度检测

LEN(_E)

检测并输出输入的字符串的长度。

梯形图、FBD/LD		ST
[无EN/ENO]	[带EN/ENO]	[无EN/ENO] d:=LEN(s); [带EN/ENO] d:=LEN_E(EN, ENO, s);

设置数据

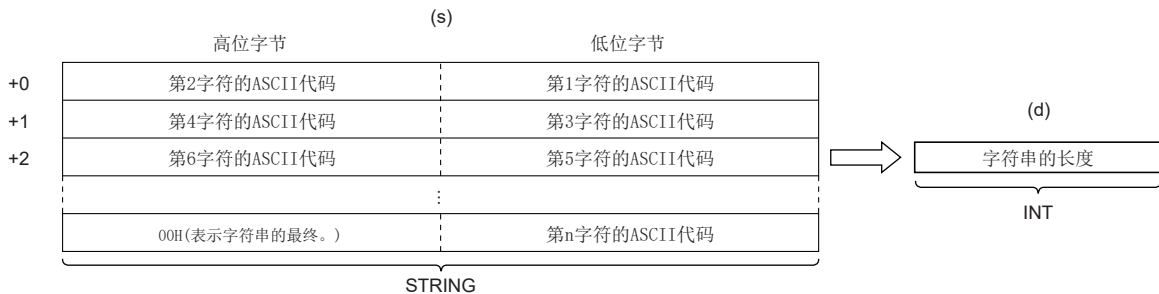
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	ANYSTRING_SINGLE
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	INT

功能

■运算处理

- 检测(s)中输入的字符串的长度后, 从(d)进行输出。



- 至(s)的输入值为STRING型的数据值且在0~255字节的范围内。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

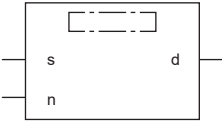
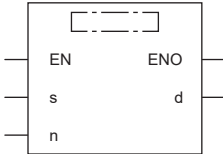
出错

错误代码 (SD0)	内容
2820H	(s) 中指定的软元件编号及其以后，软元件/标签存储器的各设置区域中不存在00H时。
3285H	(s) 的字符串超过了16383字符时。

38.2 从字符串的左侧、右侧提取

LEFT(_E)、RIGHT(_E)

- LEFT(_E)：从输入的字符串数据的左侧开始输出指定字符。
- RIGHT(_E)：从输入的字符串数据的右侧开始输出指定字符。

梯形图、FBD/LD		ST
[无EN/ENO]		[无EN/ENO] d:=LEFT(s, n); d:=RIGHT(s, n);
[带EN/ENO]		[带EN/ENO] d:=LEFT_E(EN, ENO, s, n); d:=RIGHT_E(EN, ENO, s, n);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	ANYSTRING_SINGLE
n(L)	提取字符数指定	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANYSTRING_SINGLE

功能

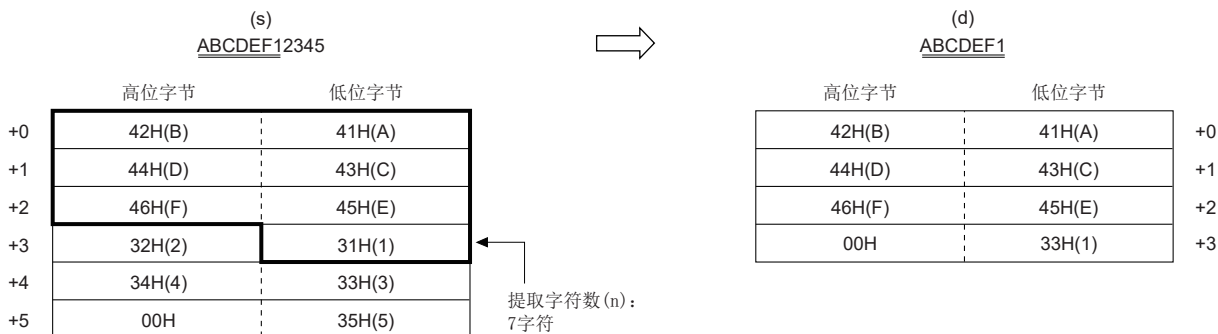
■运算处理

• LEFT(_E)

从输入至(s)的字符串的左侧开始从(d)输出指定字符数的数据。
提取的字符数根据至(n)的输入值进行指定。

例

至(n)的输入值为7的情况下

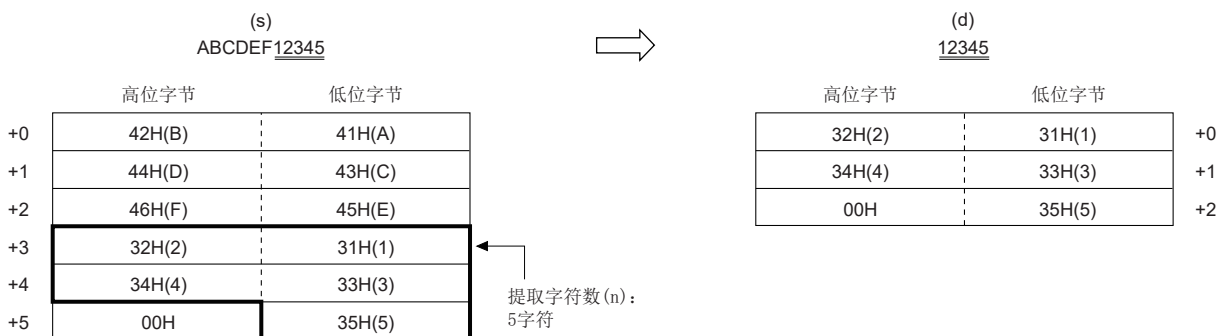


• RIGHT(_E)

从输入至(s)的字符串的右侧开始从(d)输出指定字符数的数据。
提取的字符数根据至(n)的输入值进行指定。

例

至(n)的输入值为5的情况下



- 至(s)的输入值为STRING型的数据值且在0~255字节的范围内。
- 至(n)的输入值为INT型的数据值且在0~255的范围内。(但是, 限制在被输入至(s)的字符串的字符数以内。)

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

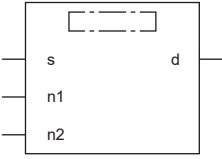
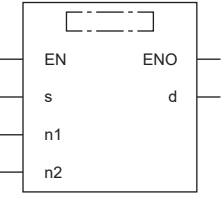
*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

错误代码 (SD0)	内容
2820H	(s) 中指定的软元件编号及其以后，软元件/标签存储器的各设置区域中不存在00H时。
3285H	(s) 的字符串超过了16383字符时。
	(s) 的字符串为0字符时。
	(n) 超过了 (s) 中指定的字符串的字符数时。

38.3 字符串的提取

MID(_E)

从输入的字符串的任意位置开始输出指定字符。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] <code>d:=MID(s, n1, n2);</code> [带EN/ENO] <code>d:=MID_E(EN, ENO, s, n1, n2);</code>

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输入	输入变量	ANYSTRING_SINGLE
n1 (L)	提取字符数指定	输入变量	INT
n2 (P)	提取起始位置指定	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANYSTRING_SINGLE

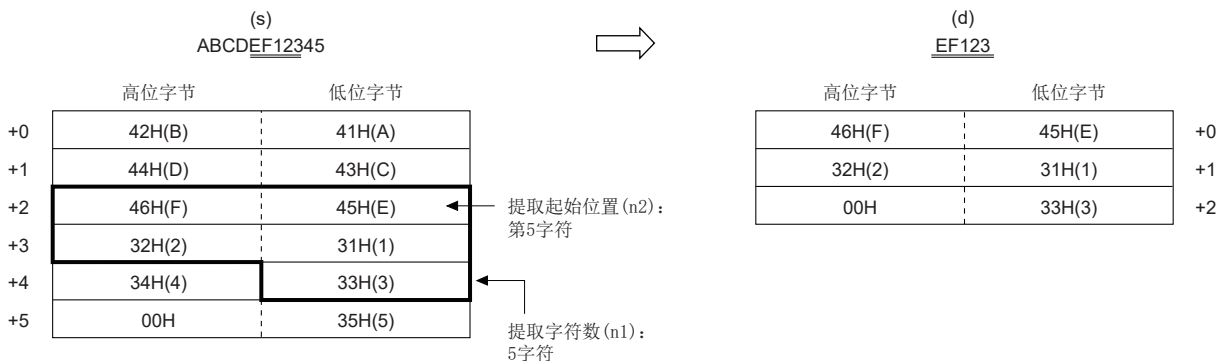
功能

■运算处理

- 从输入至(s)的字符串的任意位置开始从(d)输出指定字符数的数据。
- 提取的字符数根据至(n1)的输入值进行指定。
- 提取的字符串的起始位置根据至(n2)的输入值进行指定。

例

至(n1)、(n2)的输入值为5的情况下



- 至(s)的输入值为STRING型的数据值且在0~255字节的范围内。
- 至(n1)的输入值为INT型的数据值且在0~255的范围内。(但是, 限制在被输入至(s)的字符串的字符数以内。)
- 至(n2)的输入值为INT型的数据值且在1~255的范围内。(但是, 限制在被输入至(s)的字符串的字符数以内。)

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

出错

错误代码(SD0)	内容
2820H	(s)中指定的软元件编号及其以后, 软元件/标签存储器的各设置区域中不存在00H时。
3285H	(s)的字符串超过了16383字符时。 (n1)、(n2)中设置了超出允许指定范围的数据时。 <ul style="list-style-type: none"> • (n1)、(n2)的值为0或其以下 • (n2)的值为有效值(-1、0、1或其以上)以外 • (n1)的值超过了(s)的字符数 • (n1)与(n2)的加法运算后的值超过了(s)的字符数

38.4 字符串的合并

CONCAT(_E)

合并字符串后输出。

梯形图、FBD/LD*1		ST*1
[无EN/ENO]	[带EN/ENO]	[无EN/ENO] d:=CONCAT(s1, s2); [带EN/ENO] d:=CONCAT_E(EN, ENO, s1, s2);

*1 输入变量s可以在2~28的范围内进行更改。

设置数据

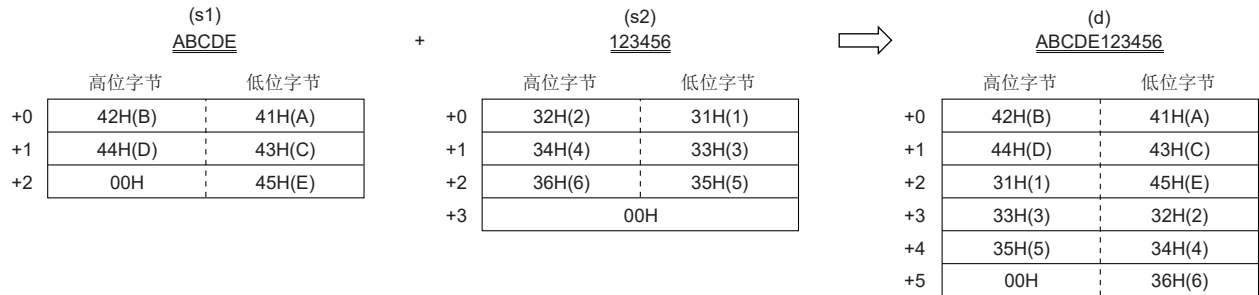
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (IN1)~s28 (IN28)	输入	输入变量	ANYSTRING_SINGLE
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANYSTRING_SINGLE

功能

■运算处理

- 在输入至(s1)的字符串后面, 合并输入至输入变量(s2)~(s28)的字符串后, 从(d)进行输出。
- 字符串合并时, 将忽略表示(s1)的字符串的结束的00H, 连接(s2)~(s28)的字符串进行合并。
- 合并的字符串超过了255字节的情况下, 输出直到255字节为止的字符串。



- 至输入变量(s1)、(s2)~(s28)的输入值为STRING型的数据值且在0~255字节的范围内。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

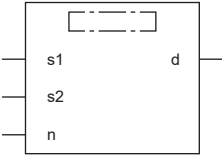
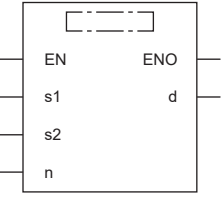
*1 从ENO输出了FALSE的情况下, 从(d)输出的数据将变为不定值。在此情况下, 应进行不使用从(d)输出的数据的程序处理。

错误代码 (SD0)	内容
2820H	(s1) ~ (s28) 中指定的软元件编号及其以后，软元件/标签存储器的各设置区域之间不存在00H时。
	(d) 中指定的软元件编号及其以后，软元件/标签存储器的各设置区域之间不存在00H时。
2821H	(s1) ~ (s28) 中 (d) 中指定的字符串的存储软元件编号重复时。
3285H	(s1) ~ (s28) 的字符串超过了16383字符时。
	(s1) ~ (s28) 的字符串为0字符时。
	(d) 的字符串超过16383字符时。
3286H	(d) 中指定的软元件编号及其以后，相应软元件/标签存储器的各设置区域的最终编号为止的点数中无法存储合并的全部字符串时。

38.5 字符串的插入

INSERT(_E)

在字符串之间插入字符串后输出。

梯形图、FBD/LD		ST
<p>[无EN/ENO]</p> 	<p>[带EN/ENO]</p> 	<p>[无EN/ENO] <code>d:=INSERT(s1, s2, n);</code> [带EN/ENO] <code>d:=INSERT_E(EN, ENO, s1, s2, n);</code></p>

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1(IN1)、s2(IN2)	输入	输入变量	ANYSTRING_SINGLE
n(P)	插入起始位置指定	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANYSTRING_SINGLE

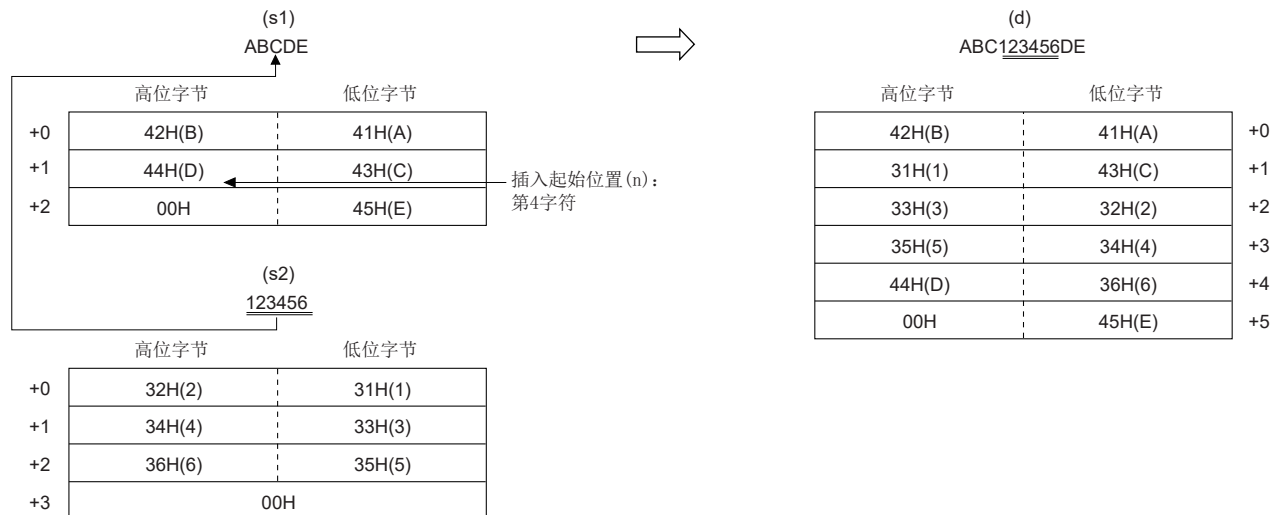
功能

■运算处理

- 将输入至(s2)的字符串插入到从输入至(s1)的字符串的起始开始的第n字符(插入起始位置)处后,从(d)进行输出。
- 将(s2)的字符串插入(s1)的字符串后,表示(s2)的字符串的结束的00H将被忽略。
- 插入的字符串超过了255字节的情况下,输出直到255字节为止的字符串。

例

至(n)的输入值为4的情况下



- 至(s1)、(s2)的输入值为STRING型的数据值且在0~255字节的范围内。
- 至(n)的输入值为INT型的数据值且在1~255的范围内。(但是,限制在输入至(s1)的字符串的字符数以内。)

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下,从(d)输出的数据将变为不定值。在此情况下,应进行不使用从(d)输出的数据的程序处理。

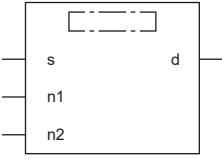
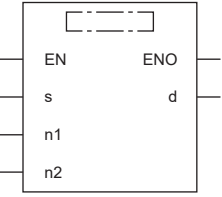
出错

错误代码(SD0)	内容
2820H	(s1)~(s28)中指定的软元件编号及其以后,软元件/标签存储器的各设置区域之间不存在00H时。 (d)中指定的软元件编号及其以后,软元件/标签存储器的各设置区域之间不存在00H时。
2821H	(s1)~(s28)中(d)中指定的字符串的存储软元件编号重复时。
3285H	(s1)~(s28)的字符串超过了16383字符时。 (s1)~(s28)的字符串为0字符时。 (d)的字符串超过16383字符时。
3286H	(d)中指定的软元件编号及其以后,相应软元件/标签存储器的各设置区域的最终编号为止的点数中无法存储合并的全部字符串时。

38.6 字符串的删除

DELETE (_E)

删除字符串的任意范围后输出。

梯形图、FBD/LD		ST
<p>[无EN/ENO]</p> 	<p>[带EN/ENO]</p> 	<p>[无EN/ENO] d:=DELETE(s, n1, n2);</p> <p>[带EN/ENO] d:=DELETE_E(EN, ENO, s, n1, n2);</p>

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s(IN)	输入	输入变量	ANYSTRING_SINGLE
n1(L)	删除字符数指定	输入变量	INT
n2(P)	删除起始位置指定	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANYSTRING_SINGLE

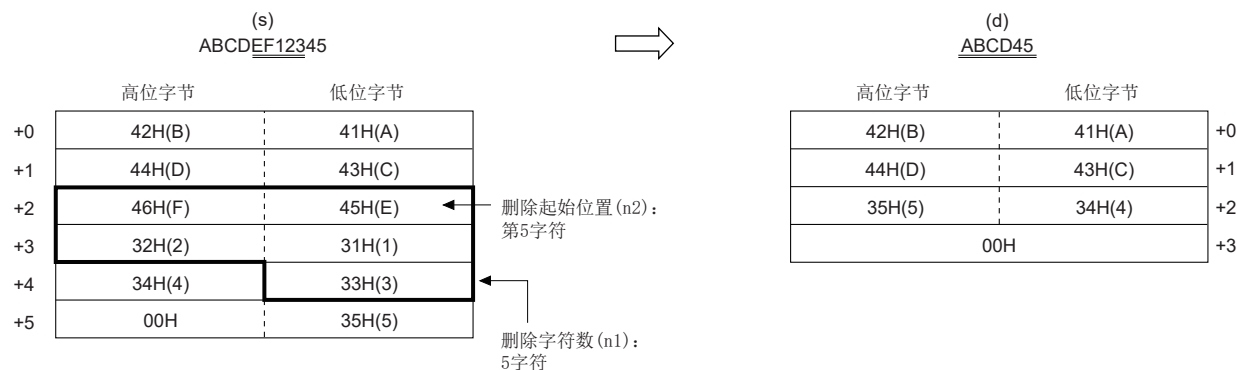
功能

■运算处理

- 从输入至(s)的字符串的任意位置开始删除指定字符数的数据，从(d)输出剩余的字符数。
- 删除的字符数根据至(n1)的输入值进行指定。
- 删除的字符串的起始位置根据至(n2)的输入值进行指定。

例

至(n1)、(n2)的输入值为5的情况下



- 至(s)的输入值为STRING型的数据值且在0~255字节的范围内。
- 至(n1)的输入值为INT型的数据值且在0~255的范围内。(但是，限制在被输入至(s)的字符串的字符数以内。)
- 至(n2)的输入值为INT型的数据值且在1~255的范围内。(但是，限制在被输入至(s)的字符串的字符数以内。)

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

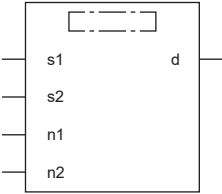
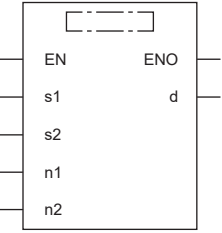
出错

错误代码 (SD0)	内容
2820H	在(s)中指定的标签或指定的软元件编号开始，至相应软元件的最终软元件编号为止之间，未设置00H时。
3285H	(s)中指定的字符串的字符数超过了255字符时。
	(n1)中指定的内容为0~255的范围外时。
	(n2)中指定的内容为1~255的范围外时。
	(n1)超过了(s)中指定的字符串的字符数时。
	(n2)超过了(s)中指定的字符串的字符数时。
3286H	在(d)中指定的标签或指定的软元件编号开始至相应软元件的最终软元件编号为止的点数中，删除后的字符串无法全部存储时。

38.7 字符串的替换

REPLACE(_E)

替换字符串的任意范围后输出。

梯形图、FBD/LD		ST
<p>[无EN/ENO]</p> 	<p>[带EN/ENO]</p> 	<p>[无EN/ENO] d:=REPLACE(s1, s2, n1, n2);</p> <p>[带EN/ENO] d:=REPLACE_E(EN, ENO, s1, s2, n1, n2);</p>

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1(IN1)、s2(IN2)	输入	输入变量	ANYSTRING_SINGLE
n1(L)	替换字符数指定	输入变量	INT
n2(P)	替换起始位置指定	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	ANYSTRING_SINGLE

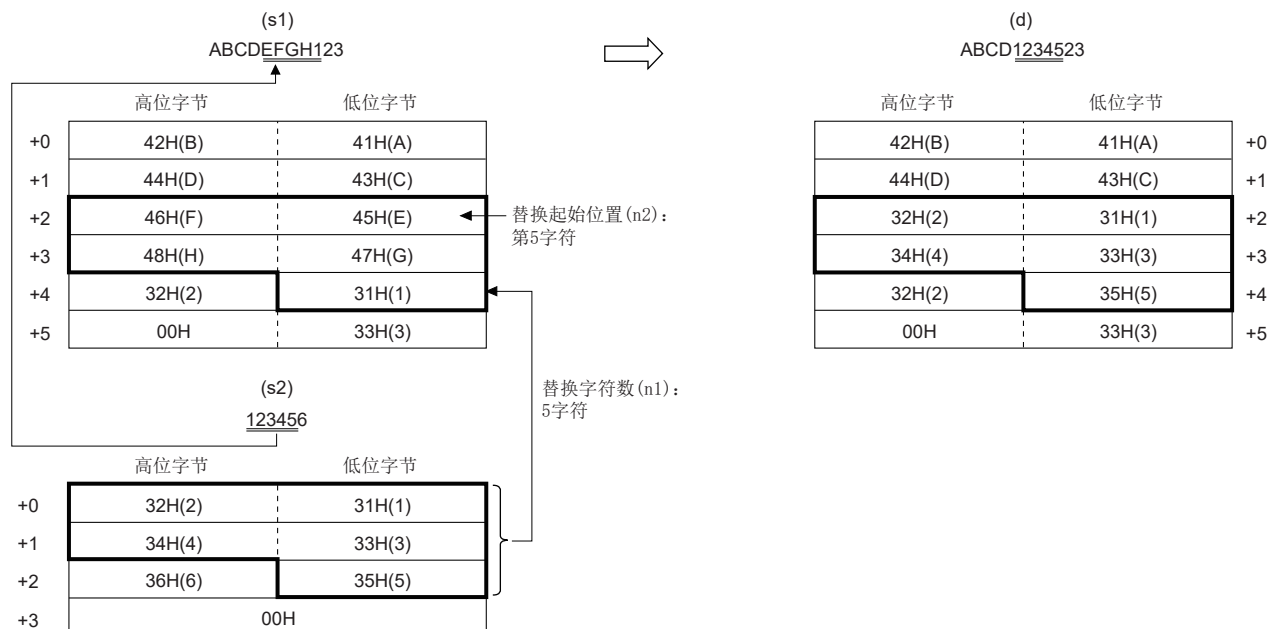
功能

■运算处理

- 从输入至(s1)的字符串的任意位置开始将指定字符数的数据替换为输入至(s2)的字符串，从(d)进行输出。
- 替换的字符数根据至(n1)的输入值进行指定。
- 替换的字符串的起始位置根据至(n2)的输入值进行指定。

例

至(n1)、(n2)的输入值为5的情况下



- 至(s1)、(s2)的输入值为STRING型的数据值且在0~255字节的范围内。
- 至(n1)的输入值为INT型的数据值且在0~255的范围内。(但是，限制在输入至(s1)的字符串的字符数以内。)
- 至(n2)的输入值为INT型的数据值且在1~255的范围内。(但是，限制在输入至(s1)的字符串的字符数以内。)

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

出错

错误代码 (SD0)	内容
2820H	在 (s1) 中指定标签或指定的软元件编号开始, 至相应软元件的最终软元件编号为止之间, 未设置00H时。
	在 (s2) 中指定标签或指定的软元件编号开始, 至相应软元件的最终软元件编号为止之间, 未设置00H时。
3285H	(s1) 中指定的字符串的字符数超过了255字符时。
	(s2) 中指定的字符串的字符数超过了255字符时。
	(n1) 中指定的内容为0~255的范围外时。
	(n2) 中指定的内容为1~255的范围外时。
	(n1) 超过了 (s2) 中指定的字符串的字符数时。
	(n2) 超过了 (s1) 中指定的字符串的字符数时。
3286H	在 (d) 中指定的标签或指定的软元件编号开始至相应软元件的最终软元件编号为止的点数中删除后的字符串无法全部存储时。

38.8 字符串的搜索

FIND(_E)

搜索字符串并输出搜索结果。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=FIND(s1,s2); [带EN/ENO] d:=FIND_E(EN,ENO,s1,s2);

设置数据

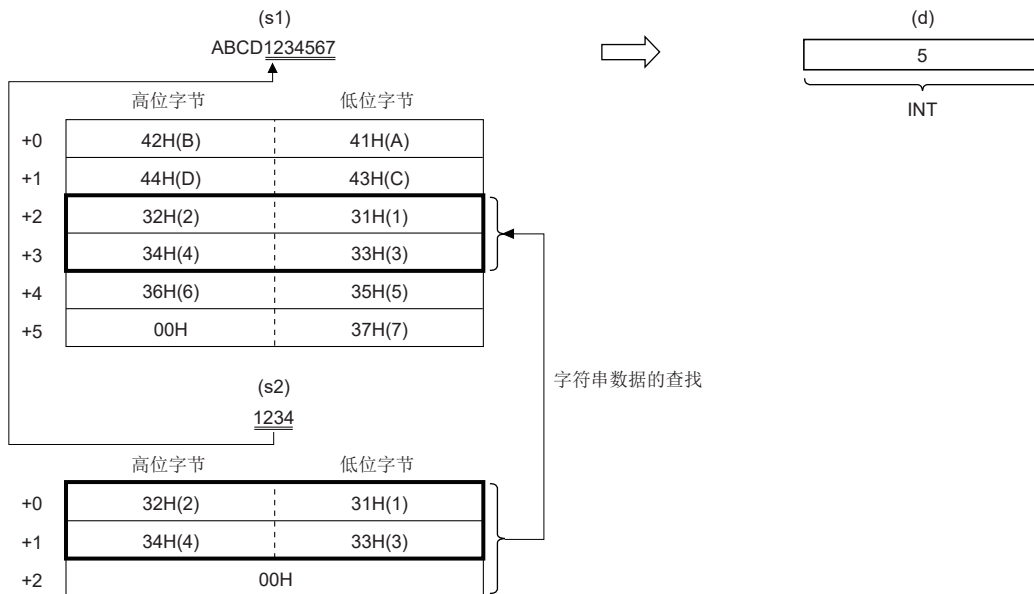
■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1(IN1)、s2(IN2)	输入	输入变量	ANYSTRING_SINGLE
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	INT

功能

■运算处理

- 从输入至(s1)的字符串的最初开始，搜索输入至(s2)的字符串，从(d)输出搜索结果。
- 搜索结果输出最初搜索的字符串的起始字符位置。
- 从(s1)的字符串开始，无法搜索(s2)的字符串的情况下，输出0。



- 至(s1)、(s2)的输入值为STRING型的数据值且在0~255字节的范围内。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下，从(d)输出的数据将变为不定值。在此情况下，应进行不使用从(d)输出的数据的程序处理。

出错


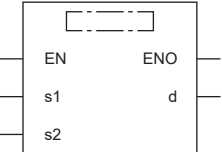
错误代码 (SD0)	内容
2820H	(s1)中指定的软元件编号及其以后，软元件/标签存储器的各设置区域中不存在00H时。
	(s2)中指定的软元件编号及其以后，软元件/标签存储器的各设置区域中不存在00H时。
3285H	(s1)的字符串超过了16383字符时。
	(s1)的字符串为0字符时。
	(s2)的字符串超过了16383字符时。
	(s2)的字符串为0字符时。

39 时间数据类型函数

39.1 加法运算

ADD_TIME(_E)

输出输入值(TIME型)的和((s1)+(s2))。

梯形图、FBD/LD		ST
<p>[无EN/ENO]</p> 	<p>[带EN/ENO]</p> 	<p>[无EN/ENO] d:=ADD_TIME(s1, s2);</p> <p>[带EN/ENO] d:=ADD_TIME_E(EN, ENO, s1, s2);</p>

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1(IN1)、s2(IN2)	输入	输入变量	TIME
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	TIME

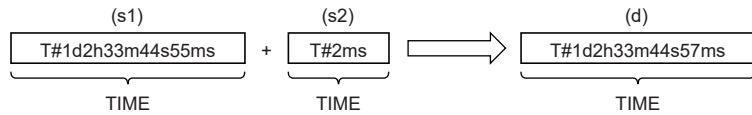
功能

■运算处理

- 进行(s1)、(s2)中输入的TIME型数据的加法运算((s1)+(s2))后,将运算结果从(d)以TIME型进行输出。

例

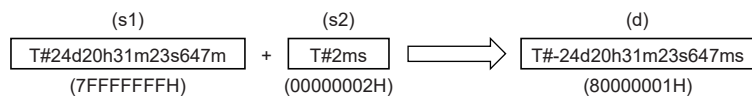
至(s1)、(s2)的输入值为T#1d2h33m44s55ms(1日2小时33分44秒55毫秒)与T#2ms(2毫秒)的情况下



- 至(s1)、(s2)的输入值为TIME型的数据值。
- 即使运算结果中发生了下溢/上溢,也不会变为运算错误。按下述方式输出至(d)中。此外,ADD_TIME_E的情况下,从输出变量ENO输出TRUE。

例

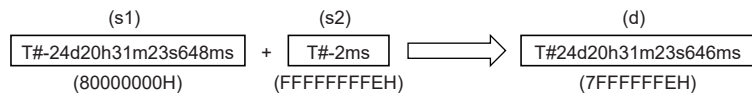
上溢



由于最高位的位变为1,因此变为负的时间。

例

下溢



由于最高位的位变为0,因此变为正的时间。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下,从(d)输出的数据将变为不定值。在此情况下,应进行不使用从(d)输出的数据的程序处理。


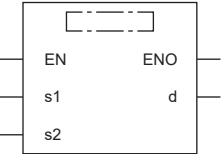
出错

没有运算错误。

39.2 减法运算

SUB_TIME(_E)

输出输入值 (TIME型) 的差 ((s1)-(s2))。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] d:=SUB_TIME(s1, s2); [带EN/ENO] d:=SUB_TIME_E(EN, ENO, s1, s2);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (IN1)、s2 (IN2)	输入	输入变量	TIME
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	TIME

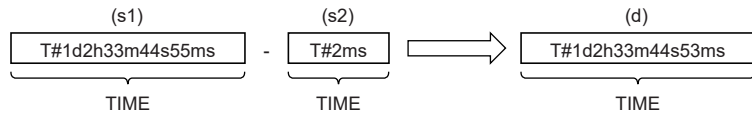
功能

■运算处理

- 进行(s1)、(s2)中输入的TIME型数据的减法运算((s1)-(s2))后,将运算结果从(d)以TIME型进行输出。

例

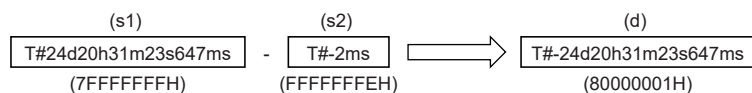
至(s1)、(s2)的输入值为T#1d2h33m44s55ms(1日2小时33分44秒55毫秒)与T#2ms(2毫秒)的情况下



- 至(s1)、(s2)的输入值为TIME型的数据值。
- 即使运算结果中发生了下溢/上溢,也不会变为运算错误。按下述方式输出至(d)中。此外, SUB_TIME_E的情况下,从输出变量ENO输出TRUE。

例

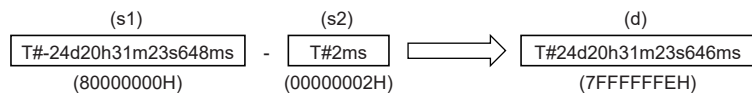
上溢



由于最高位的位变为1,因此变为负的时间。

例

下溢



由于最高位的位变为0,因此变为正的时间。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下,从(d)输出的数据将变为不定值。在此情况下,应进行不使用从(d)输出的数据的程序处理。

出错

没有运算错误。

39.3 乘法运算

MUL_TIME(_E)

输出输入值 (TIME型) 的积 ((s1) × (s2))。

梯形图、FBD/LD		ST
[无EN/ENO]		[无EN/ENO] $d := \text{MUL_TIME}(s1, s2);$ [带EN/ENO] $d := \text{MUL_TIME_E}(\text{EN}, \text{ENO}, s1, s2);$

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件 (TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (IN1)	输入	输入变量	TIME
s2 (IN2)	输入	输入变量	ANY_NUM
ENO	输出状态 (TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	TIME

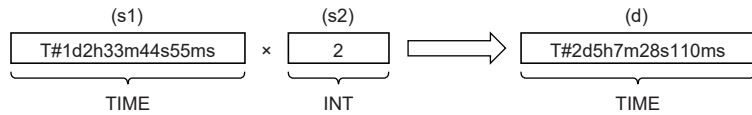
功能

■运算处理

- 进行(s1)、(s2)中输入的TIME型数据的乘法运算((s1)×(s2))后,将运算结果从(d)以TIME型进行输出。

例

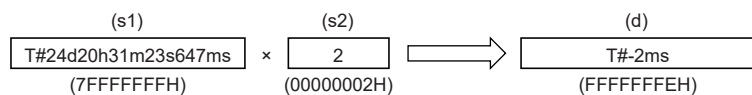
至(s1)、(s2)的输入值为T#1d2h33m44s55ms(1日2小时33分44秒55毫秒)与2的情况下



- 至(s1)的输入值为TIME型的数据值。
- 至(s2)的输入值为INT型/DINT型/REAL型/LREAL型。
- 即使运算结果中发生了下溢/上溢,也不会变为运算错误。按下述方式输出至(d)中。此外,MUL_TIME_E的情况下,从输出变量ENO输出TRUE。(运算结果变为64位数据,但是输出通过删除了高位32位的数据类型数据被输出。)

例

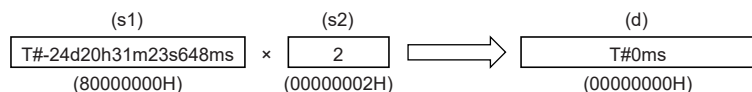
上溢



由于最高位的位变为1,因此变为负的时间。

例

下溢



由于最高位的位变为0,因此变为正的时间。

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下,从(d)输出的数据将变为不定值。在此情况下,应进行不使用从(d)输出的数据的程序处理。

出错

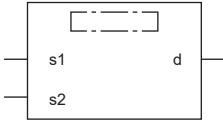
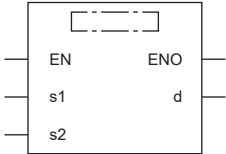
- (s2)为LREAL型的情况下

错误代码(SD0)	内容
3282H	(s2)的内容为下述范围外时。 $-2^{1024} < (s2) \leq -2^{-1022}$ 、 0 、 $2^{-1022} \leq (s2) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308、0、E2.2250738585072014-308~E1.7976931348623157+308) (s2)中指定的数据为-0、非正规化数、非数、±∞时。
3285H	(s2)中设置的单精度实数数据超出-2147483648~2147483647的范围时。

39.4 除法运算

DIV_TIME(_E)

输出输入值 (TIME型) 的商 ((s1)÷(s2))。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] <code>d:=DIV_TIME(s1, s2);</code> [带EN/ENO] <code>d:=DIV_TIME_E(EN, ENO, s1, s2);</code>

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (IN1)	输入	输入变量	TIME
s2 (IN2)	输入	输入变量	ANY_NUM
ENO	输出状态(TRUE: 正常, FALSE: 异常)	输出变量	BOOL
d	输出	输出变量	TIME

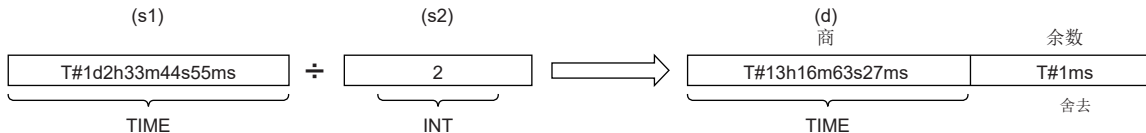
功能

■运算处理

- 进行(s1)、(s2)中输入的TIME型数据的除法运算((s1)÷(s2))后,将运算结果从(d)以TIME型进行输出。舍去余数。

例

至(s1)、(s2)的输入值为T#1d2h33m44s55ms(1日2小时33分44秒55毫秒)与2的情况下



- 至(s1)的输入值为TIME型的数据值。
- 至(s2)的输入值为INT型/DINT型/REAL型/LREAL型。(但是,至(s2)的输入值为0以外。)

■运算结果

1. 无EN/ENO函数

执行运算处理。从(d)输出运算输出值。

2. 带EN/ENO函数

执行条件及运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE*1	不定值

*1 从ENO输出了FALSE的情况下,从(d)输出的数据将变为不定值。在此情况下,应进行不使用从(d)输出的数据的程序处理。

出错

错误代码(SD0)	内容
3280H	至(s2)的输入值为0时。(除0)

- (s2)为LREAL型的情况下

错误代码(SD0)	内容
3282H	(s2)的内容为下述范围外时。 $-2^{1024} < (s2) \leq -2^{-1022}$ 、 0 、 $2^{-1022} \leq (s2) < 2^{1024}$ (E-1.7976931348623157+308~E-2.2250738585072014-308、0、E2.2250738585072014-308~E1.7976931348623157+308)
3285H	(s2)的内容超出-2147483648~2147483647的范围时。

第7部分 通用FB

本部分由下述章构成。

40 双稳态FB

41 边缘检测FB


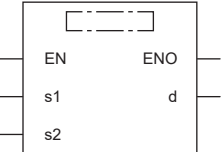
42 计数器/定时器FB

40 双稳态FB

40.1 双稳态FB(设置优先)

SR(_E)

判别2个输入值，输出1 (TRUE) 或0 (FALSE)。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] 实例名 (S1:=s1, R:=s2, Q1:=d); [带EN/ENO] 实例名 (EN:=en, ENO:=eno, S1:=s1, R:=s2, Q1:=d);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件 (TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (S1)	设置指令	输入变量	BOOL
s2 (R)	复位指令	输入变量	BOOL
ENO	输出状态 (TRUE: 正常, FALSE: 异常或停止)	输出变量	BOOL
d (Q1)	输出	输出变量	BOOL

功能

■运算处理

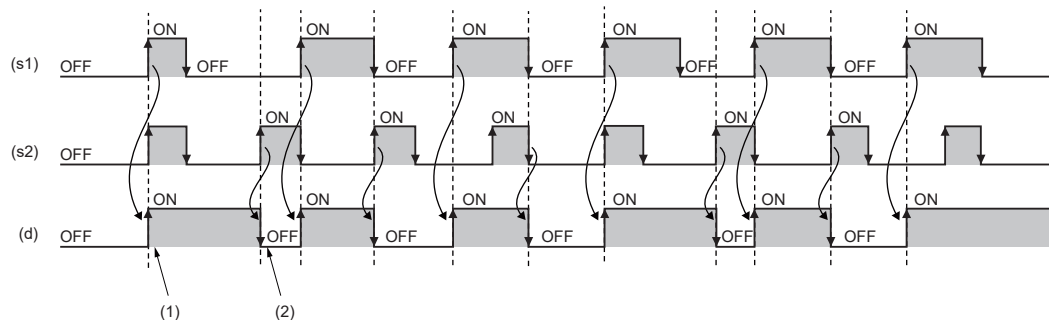
- (s1)变为ON时，对(d)进行SET。(s1)为OFF时，如果将(s2)置为ON，可以对(d)进行RESET。
- (s1)为ON时，即使将(s2)置为ON，(d)也不被RESET。

■运算结果

1. 无EN/ENO的FB

执行运算处理。从(d)输出运算输出值。

- 时序图



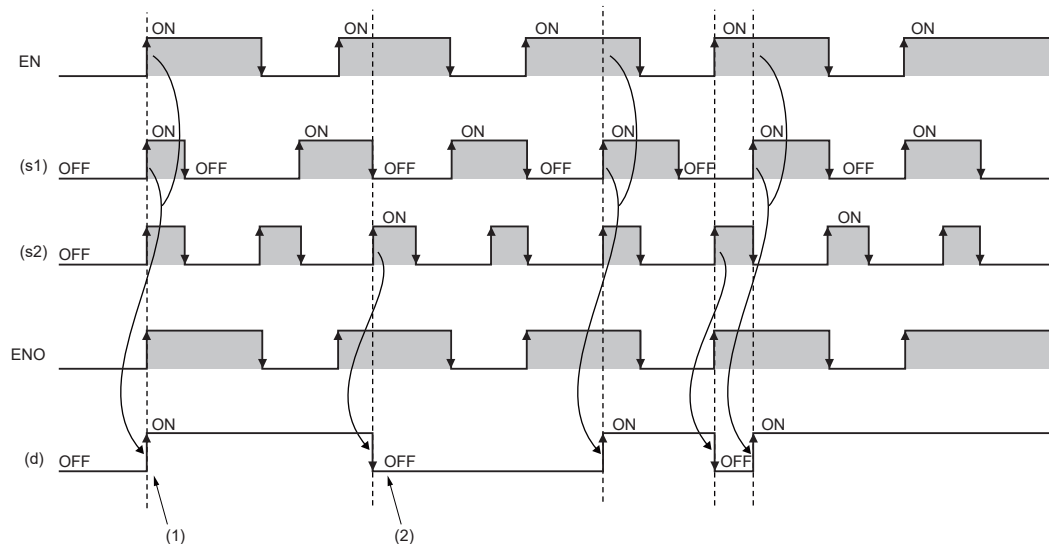
- (1) (s1)的OFF→ON时，将(d)置为ON。
 (2) (s1)=OFF且(s2)的OFF→ON时，将(d)置为OFF。

2. 带EN/ENO的FB

执行条件与运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE	上次输出值

- 时序图



- (1) EN=ON且(s1)的OFF→ON时，将(d)置为ON。
 (2) EN=ON、(s1)=OFF且(s2)OFF→ON时，将(d)置为OFF。

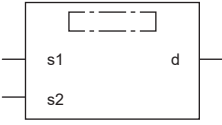
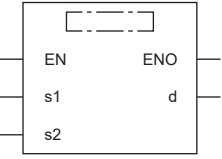
出错

没有运算错误。

40.2 双稳态FB(复位优先)

RS(_E)

判别2个输入值，输出1(TRUE)或0(FALSE)。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] 实例名(S:=s1, R1:=s2, Q1:=d); [带EN/ENO] 实例名(EN:=en, ENO:=eno, S:=s1, R1:=s2, Q1:=d);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1(S)	设置指令	输入变量	BOOL
s2(R1)	复位指令	输入变量	BOOL
ENO	输出状态(TRUE: 正常, FALSE: 异常或停止)	输出变量	BOOL
d(Q1)	输出	输出变量	BOOL

功能

■运算处理

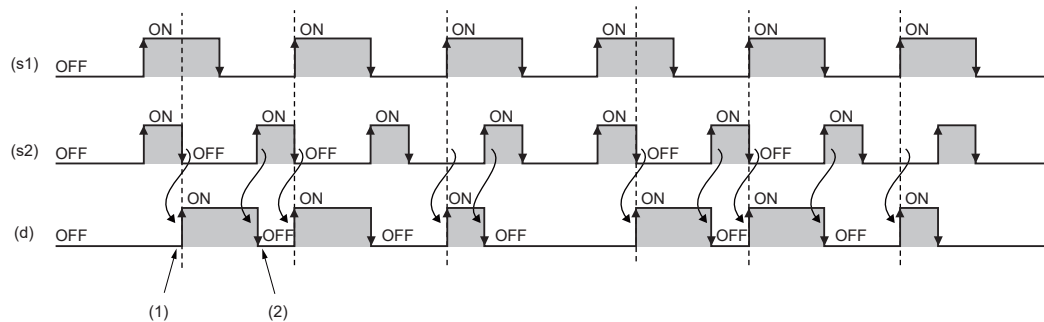
- (s1)变为ON时，对(d)进行SET。如果将(s2)置为ON，则对(d)进行RESET。
- (s2)为ON时，即使将(s1)置为ON，(d)也不被SET。

■运算结果

1. 无EN/ENO的FB

执行运算处理。从(d)输出运算输出值。

- 时序图



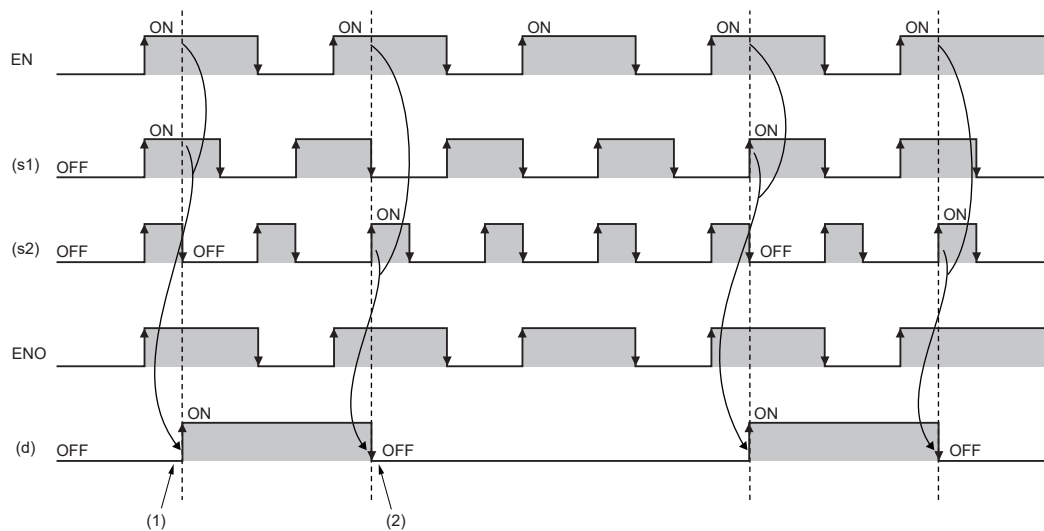
- (1) (s1)=ON且(s2)的ON→OFF时，将(d)置为ON。
 (2) (s2)的OFF→ON时，将(d)置为OFF。

2. 带EN/ENO的FB

执行条件与运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE	上次输出值

- 时序图



- (1) EN=ON、(s1)=ON且(s2)ON→OFF时，将(d)置为ON。
 (2) EN=ON且(s2)的OFF→ON时，将(d)置为OFF。

出错

没有运算错误。

41 边缘检测FB

41.1 上升沿边缘检测

R_TRIG(_E)

检测信号的上升沿后输出脉冲信号。

梯形图、FBD/LD	ST
<p>[无EN/ENO]</p> 	<p>[带EN/ENO]</p> 
	<p>[无EN/ENO] 实例名 (CLK:=s, Q:=d);</p> <p>[带EN/ENO] 实例名 (EN:=en, ENO:=eno, CLK:=s, Q:=d);</p>

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (CLK)	上升沿边缘检测输入	输入变量	BOOL
ENO	输出状态(TRUE: 正常, FALSE: 异常或停止)	输出变量	BOOL
d(Q)	输出	输出变量	BOOL

功能

■运算处理

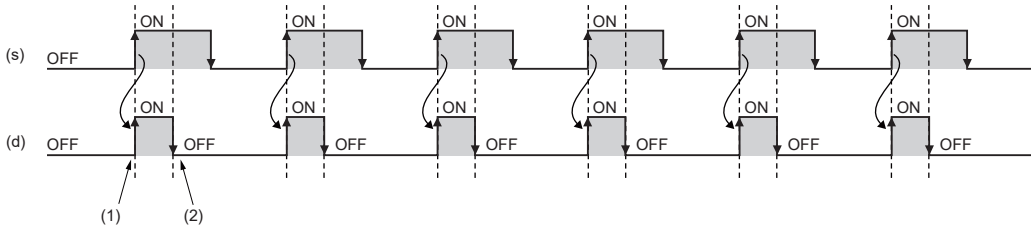
(s)变为ON时，仅1个扫描将(d)置为ON。

■运算结果

1. 无EN/ENO的FB

执行运算处理。从(d)输出运算输出值。

• 时序图



(1) (s)的上升沿时将(d)置为ON。

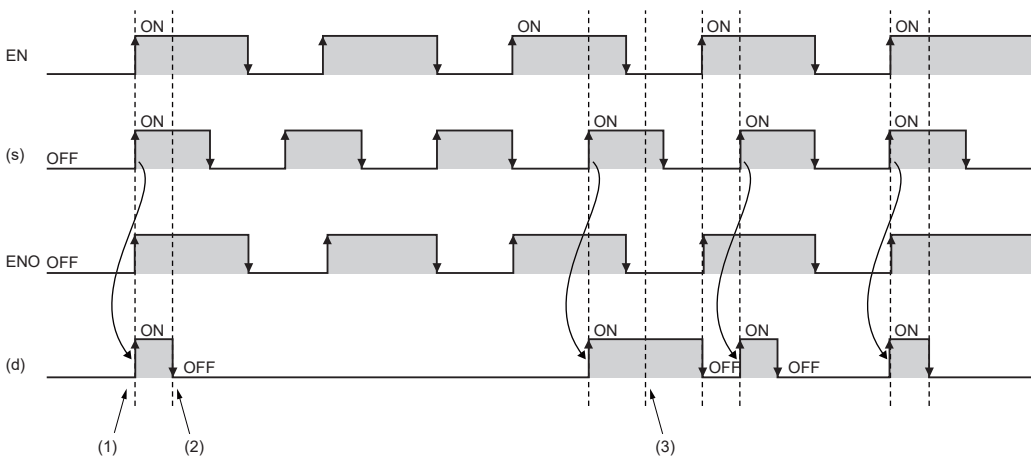
(2) 下一个扫描时将(d)置为OFF。

2. 带EN/ENO的FB

执行条件与运算结果如下所示。

执行条件		运算结果
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE	上次输出值

• 时序图



(1) EN=ON且(s)的上升沿时将(d)置为ON。

(2) 下一个扫描时将(d)置为OFF。

(3) EN=OFF的情况下，(d)保持上次扫描的输出。

出错

没有运算错误。

41.2 下降沿边缘检测

F_TRIG(_E)

检测信号的下降沿后输出脉冲信号。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] 实例名 (CLK:=s, Q:=d); [带EN/ENO] 实例名 (EN:=en, ENO:=eno, CLK:=s, Q:=d);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (CLK)	下降沿边缘检测输入	输入变量	BOOL
ENO	输出状态(TRUE: 正常, FALSE: 异常或停止)	输出变量	BOOL
d (Q)	输出	输出变量	BOOL

功能

■运算处理

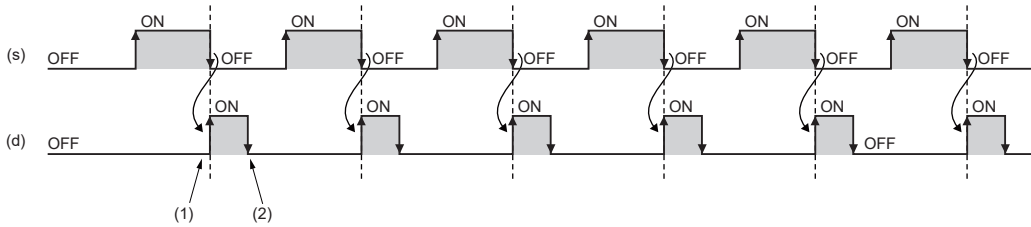
(s)变为OFF时，仅1个扫描将(d)置为ON。

■运算结果

1. 无EN/ENO的FB

执行运算处理。从(d)输出运算输出值。

• 时序图



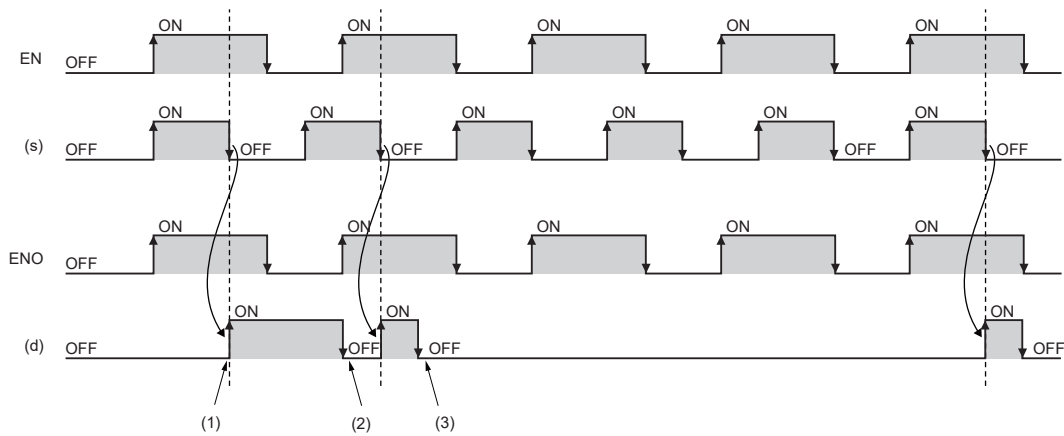
- (1) (s)下降沿时将(d)置为ON。
 (2) 下一个扫描时将(d)置为OFF。

2. 带EN/ENO的FB

执行条件与运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE	上次输出值

• 时序图



- (1) EN=ON且(s)的下降沿时将(d)置为ON。
 (2) 下一个扫描时将(d)置为OFF。
 (3) EN=OFF的情况下，(d)保持上次扫描的输出。

出错

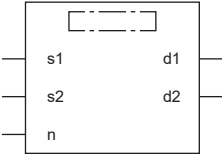
没有运算错误。

42 计数器/定时器FB

42.1 升值计数器

CTU(_E)

对信号的上升沿次数进行递增计数。

梯形图、FBD/LD	ST
<p>[无EN/ENO]</p> 	<p>[无EN/ENO] 实例名 (CU:=s1, R:=s2, PV:=n, Q:=d1, CV:=d2) ;</p> <p>[带EN/ENO] 实例名 (EN:=en, ENO:=eno, CU:=s1, R:=s2, PV:=n, Q:=d1, CV:=d2) ;</p>

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (CU)	计数信号输入	输入变量	BOOL
s2 (R)	计数值复位	输入变量	BOOL
n (PV)	计数最大值	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常或停止)	输出变量	BOOL
d1 (Q)	计数完成	输出变量	BOOL
d2 (CV)	计数值	输出变量	INT

功能

■运算处理

1. 递增计数

- 如果(s1)变为OFF→ON, 对(d2)进行加法计数(+1)。
- 如果(d2)到达计数器的(n), 则(d1)变为ON, 加法计数停止。
- (n)设置计数器的最大值。如果将(s2)置为ON, 则(d1)变为OFF, (d2)被设置为0。

2. 计数最大值

(n)的有效设置范围为0~32767。

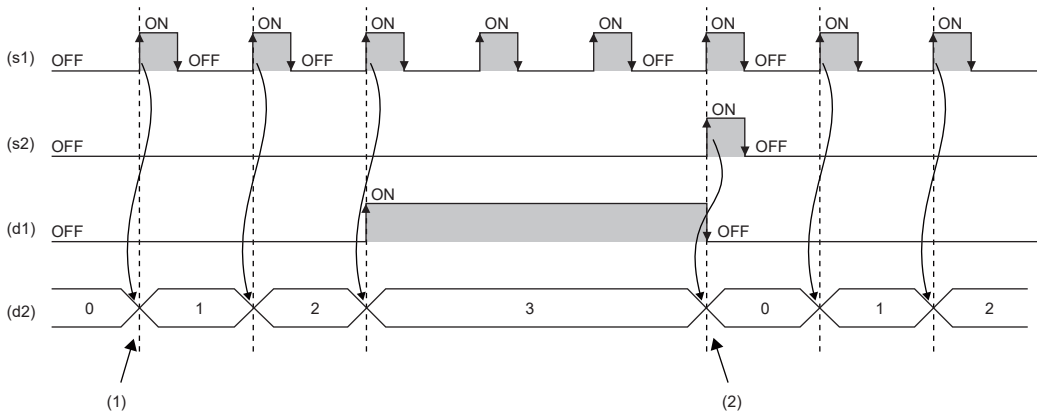
■运算结果

1. 无EN/ENO的FB

执行运算处理。从(d1)、(d2)输出运算输出值。

• 时序图

n=3的情况下



(1) (s1)的OFF→ON时(d2)递增计数。

(2) (s2)的OFF→ON时对(d2)进行初始化。

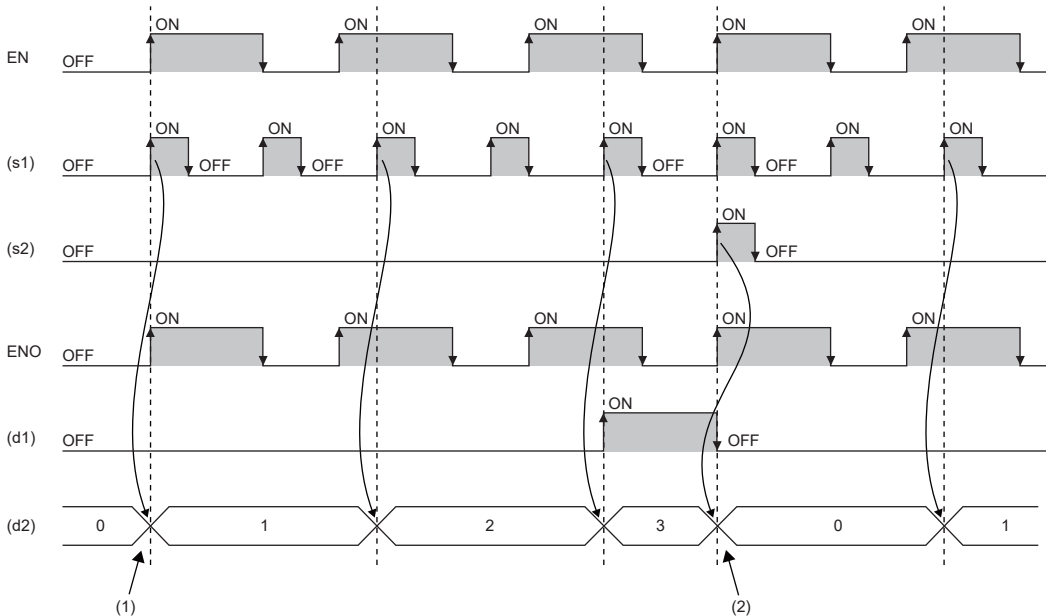
2. 带EN/ENO的FB

执行条件与运算结果如下所示。

执行条件		运算结果
EN	ENO	(d1)、(d2)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE	上次输出值

• 时序图

n=3的情况下



(1) EN=ON且(s1)的OFF→ON时(d2)递增计数。

(2) EN=ON且(s2)的OFF→ON时对(d2)进行初始化。

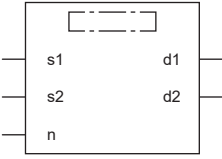
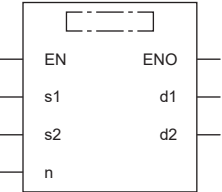
出错

没有运算错误。

42.2 降值计数器

CTD(_E)

对信号的上升沿次数进行递减计数。

梯形图、FBD/LD		ST
[无EN/ENO] 	[带EN/ENO] 	[无EN/ENO] 实例名 (CD:=s1, LD:=s2, PV:=n, Q:=d1, CV:=d2); [带EN/ENO] 实例名 (EN:=en, ENO:=eno, CD:=s1, LD:=s2, PV:=n, Q:=d1, CV:=d2);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (CD)	计数信号输入	输入变量	BOOL
s2 (LD)	计数值设置	输入变量	BOOL
n (PV)	计数开始值	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常或停止)	输出变量	BOOL
d1 (Q)	计数完成	输出变量	BOOL
d2 (CV)	计数值	输出变量	INT

功能

■运算处理

1. 递减计数

- 如果(s1)变为OFF→ON, 对(d2)进行减法计数(-1)。
- (d2)为0的情况下, (d1)变为ON, 减法计数停止。
- (n)设置为计数开始值。如果将(s2)置为ON, (d1)变为OFF, (n)被设置为(d2)。

2. 计数开始值

(n)的有效设置范围为0~32767。

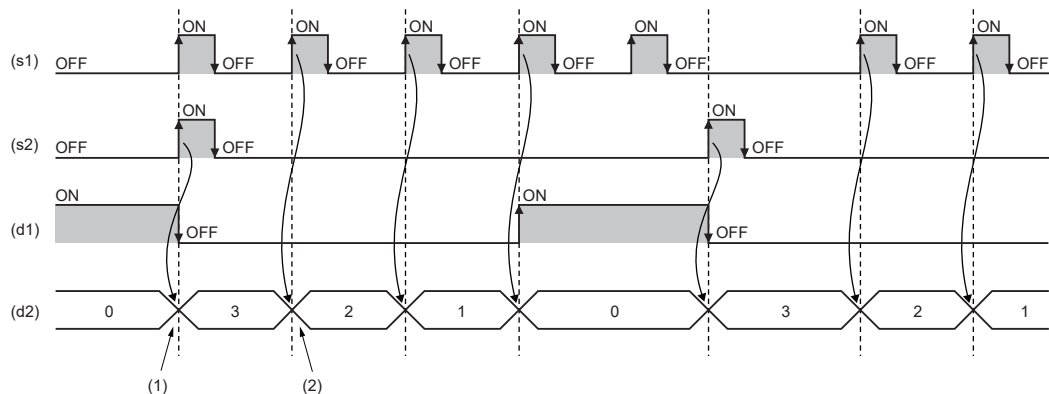
■运算结果

1. 无EN/ENO的FB

执行运算处理。从(d1)、(d2)输出运算输出值。

• 时序图

n=3的情况下



(1) (s2)的OFF→ON时对(d2)进行初始化。

(2) (s1)的OFF→ON时(d2)递减计数。

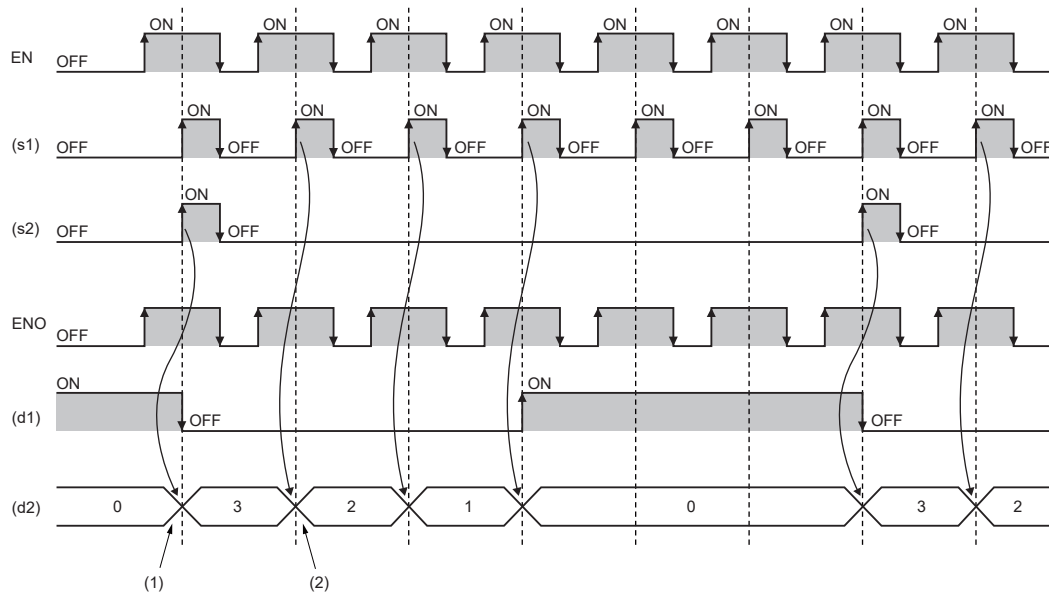
2. 带EN/ENO的FB

执行条件与运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d1)、(d2)
TRUE(运算执行)	TRUE	运算输出值
FALSE(运算停止)	FALSE	上次输出值

• 时序图

n=3的情况下



(1) EN=ON且(s2)的OFF→ON时对(d2)进行初始化。

(2) EN=ON且(s1)OFF→ON时(d2)递减计数。

出错

没有运算错误。

42.3 升值贬值计数器

CTUD(_E)

对信号的上升沿次数进行递增/递减计数。

梯形图、FBD/LD		ST
<p>[无EN/ENO]</p>	<p>[带EN/ENO]</p>	<p>[无EN/ENO] 实例名 (CU:=s1, CD:=s2, R:=s3, LD:=s4, PV:=n, QU:=d1, QD:=d2, CV:=d3) ;</p> <p>[带EN/ENO] 实例名 (EN:=en, ENO:=eno, CU:=s1, CD:=s2, R:=s3, LD:=s4, PV:=n, QU:=d1, QD:=d2, CV:=d3) ;</p>

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s1 (CU)	递增计数信号输入	输入变量	BOOL
s2 (CD)	递减计数信号输入	输入变量	BOOL
s3 (R)	计数值复位	输入变量	BOOL
s4 (LD)	计数值设置	输入变量	BOOL
n (PV)	计数最大值/开始值	输入变量	INT
ENO	输出状态(TRUE: 正常, FALSE: 异常或停止)	输出变量	BOOL
d1 (QU)	递增计数完成	输出变量	BOOL
d2 (QD)	递减计数完成	输出变量	BOOL
d3 (CV)	当前计数值	输出变量	INT

功能

■运算处理

加法计数时在(n)中设置计数最大值，减法计数时设置计数开始值。(n)的有效范围为0~32767。

1. 递增计数

- 如果(s1变为OFF→ON, 对(d3)进行加法计数(+1)。
- 如果(d3)到达(n), 则(d1)变为ON, 加法计数停止。
- 如果将(s3)置为ON, 则(d1)变为OFF, (d3)被设置为0。

2. 递减计数

- 如果(s2)变为OFF→ON, 对(d3)进行减法计数(-1)。
- (d3)为0的情况下, (d2)变为ON, 减法计数停止。
- 如果将(s4)置为ON, 则(d2)变为OFF, (n)被设置为(d3)。

3. 其它

- 如果(s1)、(s2)同时变为OFF→ON, (s1)优先对(d3)进行加法计数(+1)。
- 如果将(s3)、(s4)同时置为ON, s3优先将(d3)设置为0。

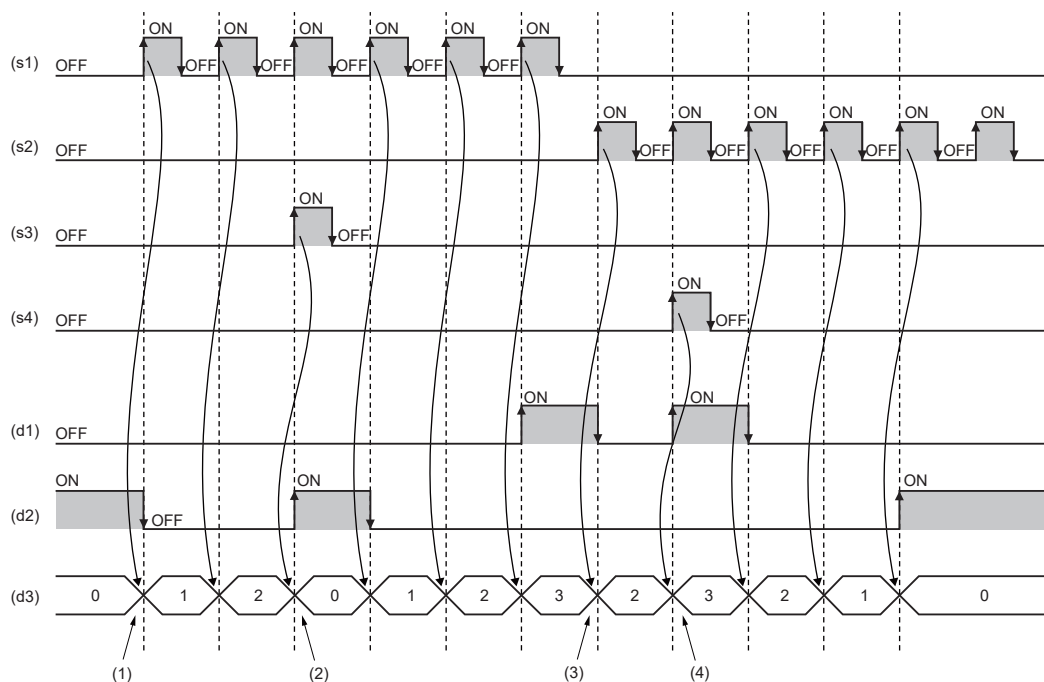
■运算结果

1. 无EN/ENO的FB

执行运算处理。从(d1)、(d2)、(d3)输出运算输出值。

- 时序图

n=3的情况下



- (1) (s1)的OFF→ON时(d3)递增计数。
- (2) (s3)的OFF→ON时对(d3)进行初始化。
- (3) (s2)的OFF→ON时(d3)递减计数。
- (4) (s4)的OFF→ON时对(d3)进行初始化。

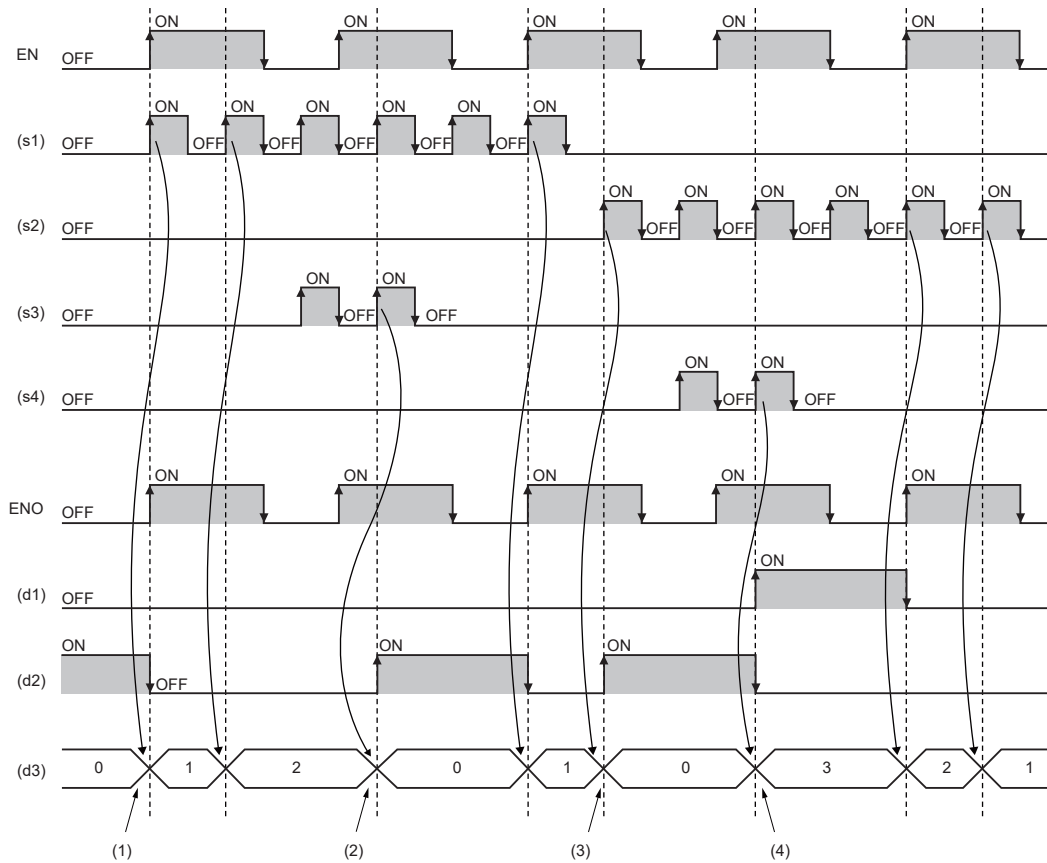
2. 带EN/ENO的FB

执行条件与运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d1)、(d2)、(d3)
TRUE (运算执行)	TRUE	运算输出值
FALSE (运算停止)	FALSE	上次输出值

• 时序图

n=3的情况下



- (1) EN=ON且(s1)的OFF→ON时(d3)递增计数。
- (2) EN=ON且(s3)的OFF→ON时对(d3)进行初始化。
- (3) EN=ON且(s2)的OFF→ON时(d3)递减计数。
- (4) EN=ON且(s4)的OFF→ON时对(d3)进行初始化。

出错

没有运算错误。

42.4 计数器FB

COUNTER_FB_M

执行条件成立时，执行递增计数。

梯形图、FBD/LD	ST
	实例名 (Coil:=s1, Preset:=s2, ValueIn:=s3, ValueOut:=d1, Status:=d2);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
s1 (Coil)	执行条件 (TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s2 (Preset)	计数器设定值	输入变量	INT
s3 (ValueIn)	计数器初始值	输入变量	INT
d1 (ValueOut)	计数器当前值	输出变量	ANY16
d2 (Status)	输出	输出变量	BOOL

功能

■运算处理

- 检测(s1)的上升沿(OFF→ON)后进行计数。(s1)为ON不变的状况下不进行计数。计数从(s3)的值开始,如果变为(s2)的值,则(d2)变为ON。当前的计数值被存储打破(d1)中。
- (s2)中可以指定0~32767的值。
- (s3)中可以指定-32768~32767的值。但是,指定了负值的情况下初始值为0。
- 希望复位计数器当前值(d1)的情况下,应直接复位FB的(s1)。

例

标签名: COUNTER_FB_M_1的情况

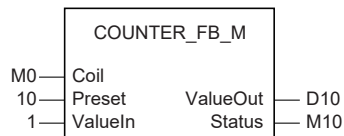
[梯形图程序]



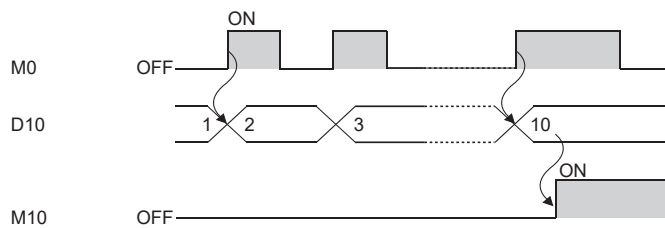
[ST程序]

```
RST(M0, COUNTER_FB_M_1.Coil)
```

[梯形图示例]



[时序图]



出错

没有运算错误。

42.5 脉冲定时器

TP(_E)

在指定时间期间将信号置为ON。

梯形图、FBD/LD		ST
[无EN/ENO]	[带EN/ENO]	[无EN/ENO] 实例名 (IN:=s, PT:=n, Q:=d1, ET:=d2); [带EN/ENO] 实例名 (EN:=en, ENO:=eno, IN:=s, PT:=n, Q:=d1, ET:=d2);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	输出开始	输入变量	BOOL
n (PT)	输出时间设定值	输入变量	TIME
ENO	输出状态(TRUE: 正常, FALSE: 异常或停止)	输出变量	BOOL
d1 (Q)	输出	输出变量	BOOL
d2 (ET)	经过时间	输出变量	TIME

功能

■运算处理

1. 输出

- 如果(s)变为ON, 则(n)中设置的时间内将(d1)置为ON。(d2)设置(d1)变为ON后的经过时间。
- 经过时间的计数使用长定时器。

2. 输出结束

- 如果经过时间达到了设置时间则将(d1)置为OFF。
- (d1)变为OFF后, (s)为OFF的情况下复位经过时间。
- 即使(d1)为ON时(s)变为OFF, (d1)也不被置为OFF。

3. 输出时间的设置

(n)的有效范围为T#1ms~T#2147483ms。但是, 由于更改工程工具的定时器时限设置, 有效设置范围如下所示。

最小值	最大值
与定时器时限的长定时器设定值[ms]相同。 但是, 长定时器设定值不满1ms的情况下将变为1ms。	为满足下述条件的值。 但是, 由于输出时间设定值为时间型(32位值), 最大值为包含在时间型范围中的值。 • 输出时间设定值[ms]≤2147483647[ms]×定时器时限的长定时器设定值[ms] [例] • 定时器时限的长定时器设定值为0.001ms的情况下: T#1ms~T#2147483ms • 定时器时限的长定时器设定值为1000ms的情况下: T#1000ms~T#2147483000ms

(n)的设定值, 在(d1)变为OFF→ON(上升沿)时的值将被使用。(d1)为ON时更改了(n)的值的的情况下, 更改的值在下次输出开始时有效。

■运算结果

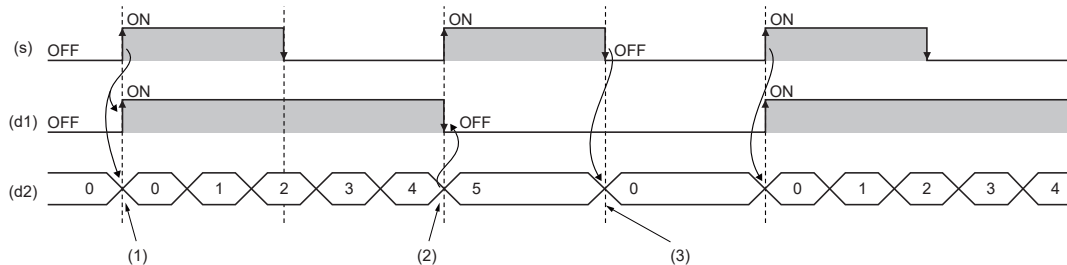
1. 无EN/ENO的FB

运算结果如下所示。

运算结果	(d1)、(d2)
无运算错误	运算输出值
有运算错误	不定值

• 时序图

n=T#5s (5秒) 的情况下



- (1) (s)的OFF→ON时将(d1)置为ON。(s)的OFF→ON时开始(d2)的时间计测。
- (2) (d2)到达(n)中指定的时间时,将(d1)置为OFF。
- (3) (s)=OFF且(d1)=OFF时对(d2)进行初始化。

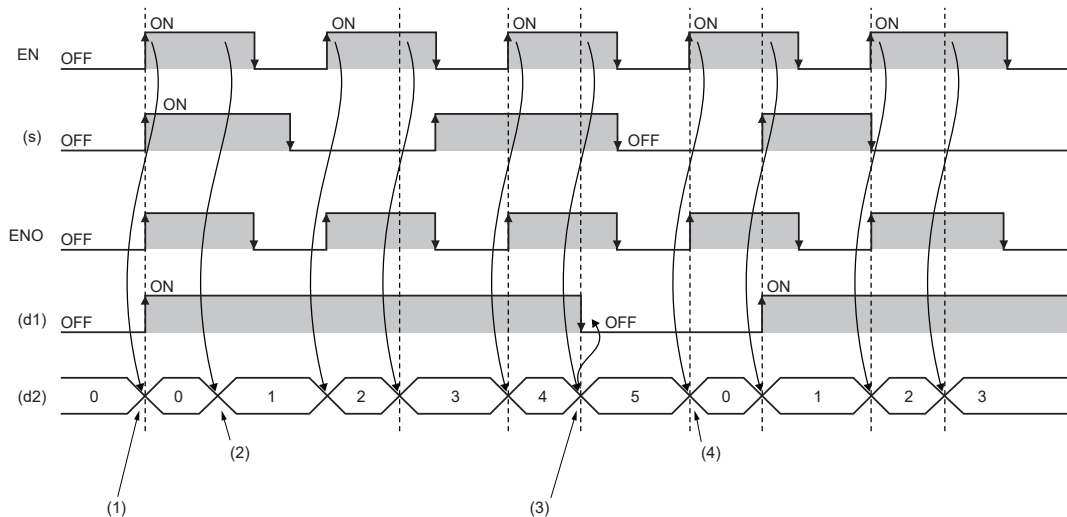
2. 带EN/ENO的FB

执行条件与运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d1)、(d2)
TRUE (运算执行)	TRUE (无运算错误)	运算输出值
	FALSE (有运算错误)	不定值
FALSE (运算停止)	FALSE	上次输出值

• 时序图

n=T#5s (5秒) 的情况下



- (1) EN=ON且(s)的OFF→ON时将(d1)置为ON。EN=ON且(s)的OFF→ON时开始(d2)的时间计测。
- (2) 开始计测后,EN=ON时计测时间到。
- (3) (d2)到达(n)中指定的时间时,将(d1)置为OFF。
- (4) EN=ON且(s)=OFF且(d1)=OFF时对(d2)进行初始化。

出错

错误代码(SD0)	内容
3281H	输出时间设定值超出有效范围时。

42.6 ON延迟定时器

TON(_E)

在指定时间后将信号置为ON。

梯形图、FBD/LD		ST
[无EN/ENO]	[带EN/ENO]	[无EN/ENO] 实例名 (IN:=s, PT:=n, Q:=d1, ET:=d2); [带EN/ENO] 实例名 (EN:=en, ENO:=eno, IN:=s, PT:=n, Q:=d1, ET:=d2);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	时间计测	输入变量	BOOL
n (PT)	延迟时间设定值	输入变量	TIME
ENO	输出状态(TRUE: 正常, FALSE: 异常或停止)	输出变量	BOOL
d1 (Q)	输出	输出变量	BOOL
d2 (ET)	经过时间	输出变量	TIME

功能

■运算处理

1. 输出

- 如果(s)变为ON, 则(n)中设置的时间后将(d1)置为ON。(d2)设置(d1)变为ON后的延迟经过时间。
- 如果(s)变为OFF则将(d1)置为OFF并复位延迟经过时间。
- 经过时间的计数使用长定时器。

2. 延迟时间的设置

(n)的有效范围为T#1ms~T#2147483ms。但是, 由于更改工程工具的定时器时限设置, 有效设置范围如下所示。

最小值	最大值
与定时器时限的长定时器设定值[ms]相同。 但是, 长定时器设定值不满1ms的情况下将变为1ms。	为满足下述条件的值。 但是, 由于延迟时间设定值为时间型(32位值), 最大值为包含在时间型范围中的值。 • 延迟时间设定值[ms]≤2147483647[ms]×定时器时限的长定时器设定值[ms] [例] • 定时器时限的长定时器设定值为0.001ms的情况下: T#1ms~T#2147483ms • 定时器时限的长定时器设定值为1000ms的情况下: T#1000ms~T#2147483000ms

(n)的设定值使用(d)变为了OFF→ON(上升沿)时的值。(s)为ON时更改了(n)的值的的情况下, 更改的值在(s)的下次上升沿时有效。

■运算结果

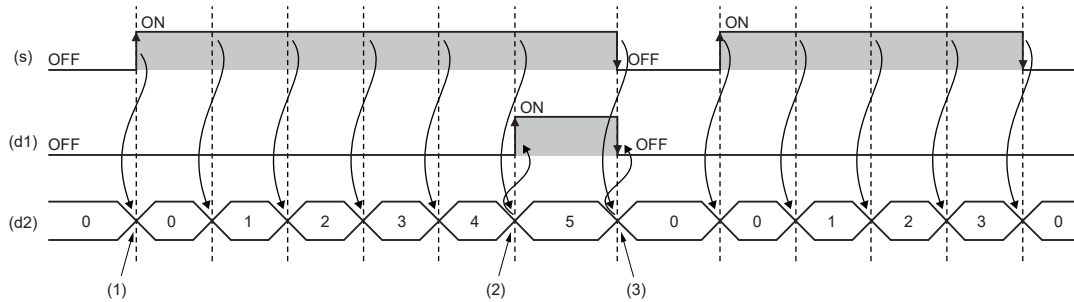
1. 无EN/ENO的FB

运算结果如下所示。

运算结果	(d1)、(d2)
无运算错误	运算输出值
有运算错误	不定值

• 时序图

n=T#5s (5秒) 的情况下



- (1) (s)的OFF→ON时开始(d2)的时间计测。
- (2) (d2)到达(n)中指定的时间时,将(d1)置为ON。
- (3) (s)的ON→OFF且(d1)的ON→OFF时对(d2)进行初始化。

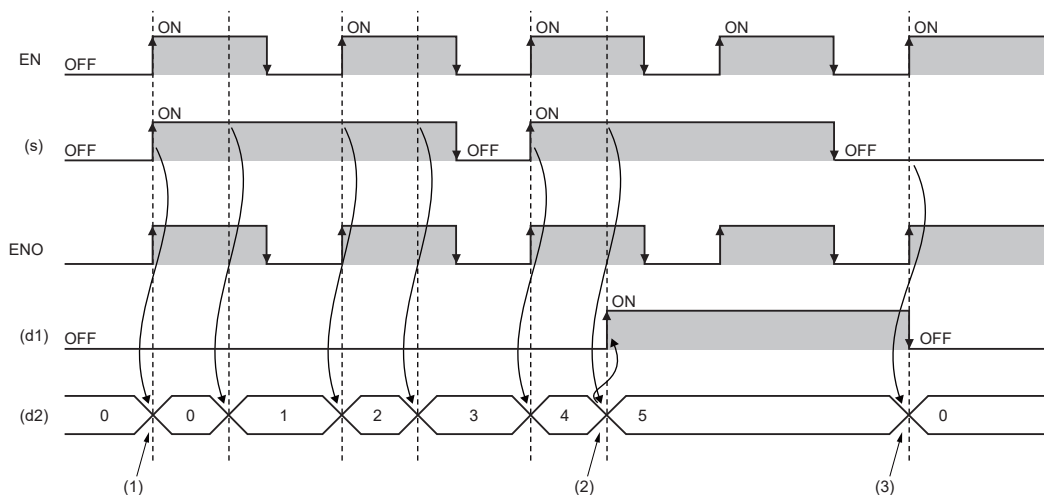
2. 带EN/ENO的FB

执行条件与运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d1)、(d2)
TRUE (运算执行)	TRUE (无运算错误)	运算输出值
	FALSE (有运算错误)	上次输出值
FALSE (运算停止)	FALSE	上次输出值

• 时序图

n=T#5s (5秒) 的情况下



- (1) EN=ON且(s)的OFF→ON时开始(d2)的时间计测。
- (2) (d2)到达(n)中指定的时间时,将(d1)置为ON。
- (3) EN=ON且(s)的ON→OFF且(d1)的ON→OFF时对(d2)进行初始化。

出错

错误代码(SD0)	内容
3281H	输出时间设定值超出有效范围时。

42.7 OFF延迟定时器

TOF (_E)

在指定时间后将信号置为OFF。

梯形图、FBD/LD		ST
[无EN/ENO]	[带EN/ENO]	[无EN/ENO] 实例名 (IN:=s, PT:=n, Q:=d1, ET:=d2); [带EN/ENO] 实例名 (EN:=en, ENO:=eno, IN:=s, PT:=n, Q:=d1, ET:=d2);

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
EN	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s (IN)	时间计测	输入变量	BOOL
n (PT)	延迟时间设定值	输入变量	TIME
ENO	输出状态(TRUE: 正常, FALSE: 异常或停止)	输出变量	BOOL
d1 (Q)	输出	输出变量	BOOL
d2 (ET)	经过时间	输出变量	TIME

功能

■运算处理

1. 输出

- 如果(s)变为ON, 则将(d1)置为ON。
- 如果(s)变为ON→OFF, 则(n)中设置的时间后将(d1)置为OFF。(d2)设置(d1)变为OFF之前的经过时间。
- 经过时间的计数使用长定时器。

2. 延迟时间的设置

(n)的有效范围为T#1ms~T#2147483ms。但是, 由于更改工程工具的定时器时限设置, 有效设置范围如下所示。

最小值	最大值
与定时器时限的长定时器设定值[ms]相同。 但是, 长定时器设定值不满1ms的情况下将变为1ms。	为满足下述条件的值。 但是, 由于延迟时间设定值为时间型(32位值), 最大值为包含在时间型范围中的值。 • 延迟时间设定值[ms]≤2147483647[ms]×定时器时限的长定时器设定值[ms] [例] • 定时器时限的长定时器设定值为0.001ms的情况下: T#1ms~T#2147483ms • 定时器时限的长定时器设定值为1000ms的情况下: T#1000ms~T#2147483000ms

(n)的设定值使用(s)由ON→OFF(下降沿)时的值。(s)为OFF时更改了(n)的值的的情况下, 更改的值在(s)的下次下降沿时有效。

■运算结果

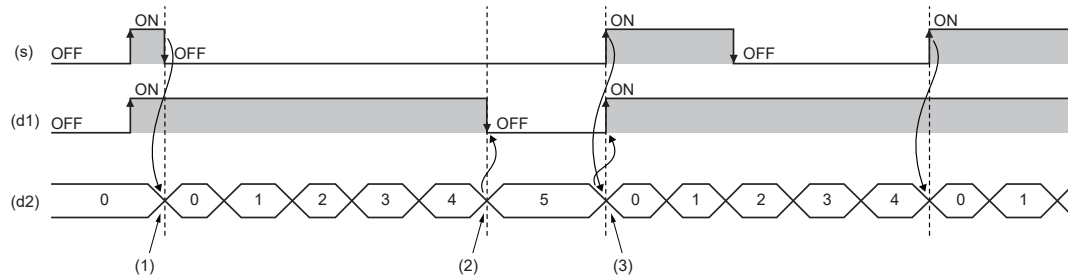
1. 无EN/ENO的FB

运算结果如下所示。

运算结果	(d1)、(d2)
无运算错误	运算输出值
有运算错误	不定值

• 时序图

n=T#5s (5秒) 的情况下



- (1) (s) 的ON→OFF时开始 (d2) 的时间计测。
- (2) (d2) 到达 (n) 中指定的时间时, 将 (d1) 置为ON。
- (3) (s) 的OFF→ON时对 (d2) 进行初始化。

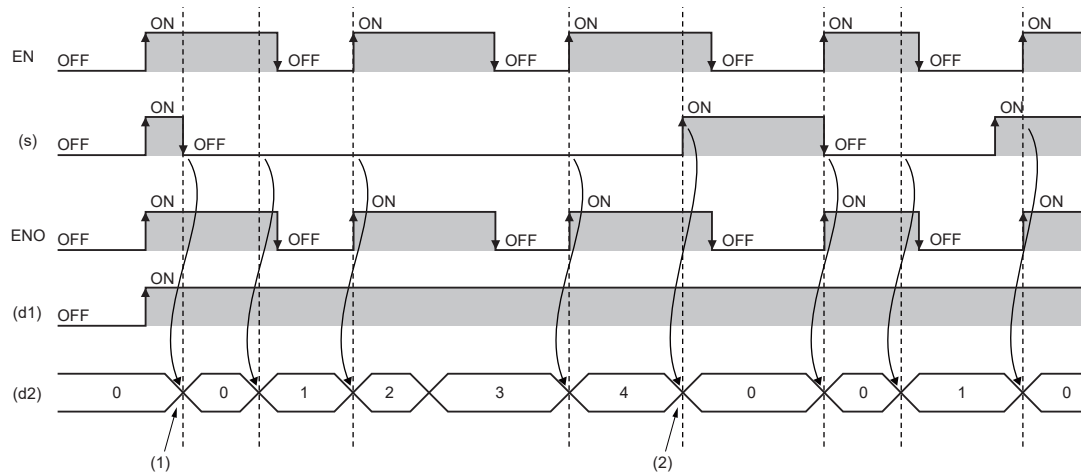
2. 带EN/ENO的FB

执行条件与运算结果如下所示。

执行条件	运算结果	
EN	ENO	(d)
TRUE (运算执行)	TRUE (无运算错误)	运算输出值
	FALSE (有运算错误)	上次输出值
FALSE (运算停止)	FALSE	上次输出值

• 时序图

n=T#5s (5秒) 的情况下



- (1) EN=ON且 (s) 的ON→OFF时开始 (d2) 的时间计测。
- (2) EN=ON且 (s) 的OFF→ON时对 (d2) 进行初始化。

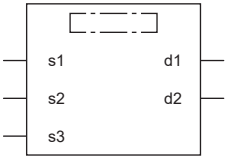
出错

错误代码 (SD0)	内容
3281H	输出时间设定值超出有效范围时。

42.8 定时器FB

TIMER_□_M

执行条件成立时，至设置的时间为止执行定时器计数。

梯形图、FBD/LD	ST
 <p>(□中放入TIMER_10_FB_M、TIMER_100_FB_M、TIMER_HIGH_FB_M、TIMER_LOW_FB_M、TIMER_CONT_FB_M、TIMER_CONTHFB_M。)</p>	<p>实例名 (Coil:=s1, Preset:=s2, ValueIn:=s3, ValueOut:=d1, Status:=d2);</p>

设置数据

■内容、类型、数据类型

自变量	内容	类型	数据类型
s1 (Coil)	执行条件(TRUE: 执行, FALSE: 停止)	输入变量	BOOL
s2 (Preset)	定时器设定值	输入变量	INT
s3 (ValueIn)	定时器初始值	输入变量	INT
d1 (ValueOut)	定时器当前值	输出变量	ANY16
d2 (Status)	输出	输出变量	BOOL

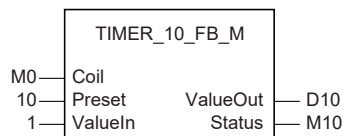
功能

■TIMER_10_FB_M

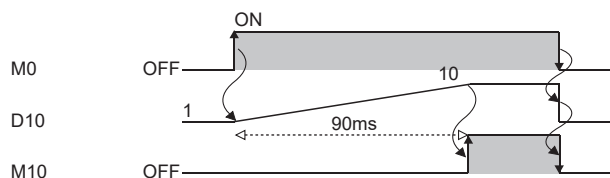
- (s1)的执行条件变为ON时，开始当前值的计测。从(s3)×10ms开始计测，直到(s2)×10ms为止到达计测值时(d2)变为ON。当前计测值被输出到(d1)中。
- 如果(s1)的执行条件变为OFF，则当前值(d1)变为0，(d2)也变为OFF。
- 使用工程工具将高速定时器的计测单位(定时器时限设置)从默认值更改为其它值的情况下，转换/全部转换时将发生报警。
- (s2)中可以指定0~32767的值。
- (s3)中可以指定-32768~32767的值。但是，指定了负值的情况下，初始值为0。

例

[梯形图示例]



[时序图]

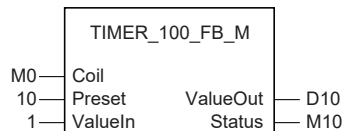


■TIMER_100_FB_M

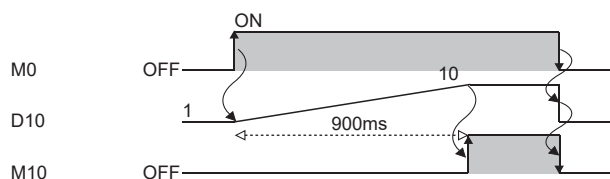
- (s1)的执行条件变为ON时，开始当前值的计测。从(s3)×100ms开始计测，直到(s2)×100ms为止到达计测值时(d2)变为ON。当前计测值被输出到(d1)中。
- 如果(s1)的执行条件变为OFF，则当前值(d1)变为0，(d2)也变为OFF。
- 使用工程工具将低速定时器的计测单位(定时器时限设置)从默认值更改为其它值的情况下，转换/全部转换时将发生报警。
- (s2)中可以指定0~32767的值。
- (s3)中可以指定-32768~32767的值。但是，指定了负值的情况下，初始值为0。

例

[梯形图示例]



[时序图]

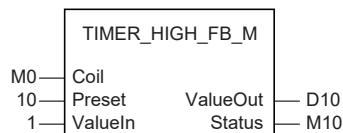


■TIMER_HIGH_FB_M

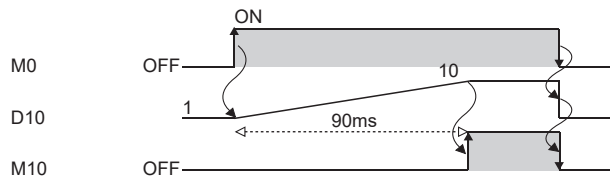
- 是计测单位为0.1~100ms的高速定时器。(s1)的执行条件变为ON时，开始当前值的计测。从(s3)×0.1~100ms(可变。通过参数设置)开始计测，直到(s2)×0.1~100ms为止达到计测值时(d2)变为ON。当前计测值被输出到(d1)中。
- 如果(s1)的执行条件变为OFF，则当前值(d1)变为0，(d2)也变为OFF。
- 高速定时器的计测单位(定时器时限设置)的默认值为10ms。计测单位可以在0.01ms~100ms的范围内更改。
- (s2)中可以指定0~32767的值。
- (s3)中可以指定-32768~32767的值。但是，指定了负值的情况下，初始值为0。

例

[梯形图示例]



[时序图]

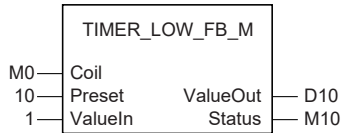


■TIMER_LOW_FB_M

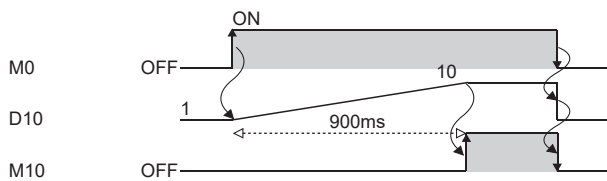
- 是计测单位为1~1000ms的低速定时器。(s1)的执行条件变为ON时，开始当前值的计测。从(s3)×1~1000ms(可变。通过参数设置)开始计测，直到(s2)×1~1000ms为止达到计测值时(d2)变为ON。当前计测值被输出到(d1)中。
- 如果(s1)的执行条件变为OFF，则当前值(d1)变为0，(d2)也变为OFF。
- 低速定时器的计测单位(定时器时限设置)的默认值为100ms。计测单位可以在1~1000ms内以1ms单位更改。
- (s2)中可以指定0~32767的值。
- (s3)中可以指定-32768~32767的值。但是，指定了负值的情况下，初始值为0。

例

[梯形图示例]



[时序图]



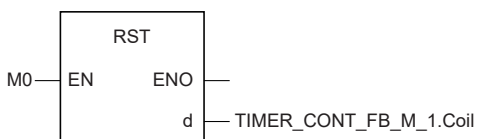
■TIMER_CONT_FB_M/TIMER_CONTHFB_M

- 是计测变量处于ON的时间的累计定时器。(s1)的执行条件变为ON时,开始当前值的计测。累计定时器有低速累计定时器(TIMER_CONT_FB_M)与高速累计定时器(TIMER_CONTHFB_M)两种类型。
- 从(s3)×1~1000ms(高速累计定时器时为0.1~100ms)(可变。通过参数设置)开始计测,直到(s2)×1~1000ms(高速累计定时器时为0.1~100ms)达到计测值时(d2)变为ON。当前计测值被输出到(d1)中。
- 即使(s1)的执行条件变为OFF,仍保持(d1)、(d2)的ON/OFF状态。(s1)的执行条件再次变为ON时,从保持的计测值重新开始计测。
- 低速定时器(TIMER_LOW_FB_M)与高速定时器(TIMER_HIGH_FB_M)的累计定时器的计测单位(时限)相同。
 - 低速累计定时器:低速定时器
 - 高速累计定时器:高速定时器
- (s2)中可以指定0~32767的值。
- (s3)中可以指定-32768~32767的值。但是,指定了负值的情况下,初始值为0。
- 希望复位累计定时器的(d1)的情况下,应直接复位FB的(s1)。

例

标签名: TIMER_CONT_FB_M_1的情况下

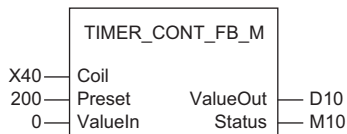
[梯形图程序]



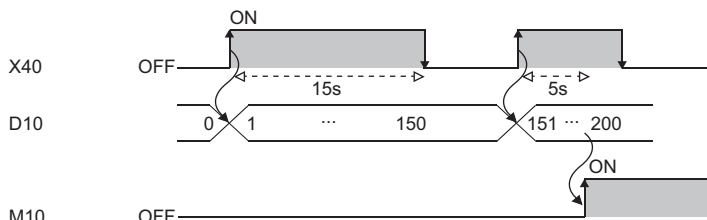
[ST程序]

RST(M0, TIMER_CONT_FB_M_1.Coil)

[梯形图示例]



[时序图]



出错

没有运算错误。

第8部分 运动控制FB

本部分由下述章构成。

43 运动控制FB的概要

44 运动控制FB变量一览

45 管理系统的FB

46 运行系统的FB

47 一般FB

43 运动控制FB的概要

以下对运动控制FB的使用方法、规格有关内容进行说明。

运动控制FB中包含PLCopen[®]中规定的FB。输入输出信号的基本规格基于PLCopen[®]的运动控制FB。

有关FB的通用规格请参阅下述内容。

📖 64页 功能块 (FB)

43.1 运动控制FB

以下对运动控制FB有关内容进行说明。

运动控制FB的使用方法

运动控制FB的使用步骤如下所示。

1. 通过工程工具创建运动控制FB的实例(全局标签或局部标签)。
2. 对于创建的运动控制FB的实例，创建设置输入输出参数的程序。
3. 将运动控制FB的执行指令 (Execute) 或有效 (Enable) 置为TRUE时，执行控制。

要点

- 1个运动控制FB实例的执行中(包括缓冲中)不能更改轴输入 (AXIS_REF输入或AXES_GROUP_REF输入)并在多个轴中反复使用。运动控制FB的轴输入只能在未执行时更改。执行中如果更改轴输入将发生警告，更改将被忽略。因此，仅需要根据同时控制的轴数创建运动控制FB的实例。
- 请勿将1个运动控制FB实例同时从多个程序(包含中断程序的并联执行)中调用。不保证同时调用时的动作。

各系统状态的运动控制FB的动作

运动控制FB只能在RUN中执行。

○：可以， ×：不能

系统的状态	动作可否
STOP中	×
RUN中	○
中度异常中	×
重度异常中	×

运动控制FB的类型

运动控制FB根据动作内容及执行方法进行分类。

管理系统FB/运行系统FB/常规FB

运动控制FB根据动作内容有下述类型。

类型	动作内容
管理系统FB	<ul style="list-style-type: none"> 是将轴或轴组作为参数，轴状态或轴组状态不根据执行而变化的运动控制FB。(有部分例外) 基本上对同一轴或轴组可以同时执行多个实例。 对同一轴或轴组同时执行了多个实例的情况下，从运动控制FB的执行优先级较高的一方开始优先控制。
运行系统FB	<ul style="list-style-type: none"> 是将轴或轴组作为参数，轴状态或轴组状态根据执行而变化的运动控制FB。 基本上只能对同一轴或轴组执行1个。但是，根据运动控制FB有的也可以并联执行。 运行系统FB执行中即使执行管理系统FB，轴状态或轴组状态也基本不变化。但是，根据运动控制FB有可能引起特定的状态转换。 对同一轴或轴组同时执行了多个实例的情况下，从运动控制FB的执行优先级较高的一方开始优先控制。
常规FB	<ul style="list-style-type: none"> 是不将轴或轴组作为参数的运动控制FB。 可以同时执行多个实例。由于与轴不相关，因此运行系统FB及管理系统FB相互无影响。

运动控制FB的执行优先级如下所示。

一：无优先度的分配

优先级	运动控制FB	
高 ↑ ↓ 低	1	<ul style="list-style-type: none"> • MC_Stop(强制停止) • MC_GroupStop(组强制停止) • MCv_MoveWait(轴控制开始等待) • MCv_GroupMoveWait(轴组控制开始等待)
	2	<ul style="list-style-type: none"> • MC_Home(原点复位) • MC_MoveAbsolute(绝对值定位) • MC_MoveRelative(相对值定位) • MCv_Jog(JOG运行) • MC_MoveVelocity(速度控制) • MC_TorqueControl(转矩控制) • MCv_SpeedControl(速度控制(包含位置循环)) • MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制) • MCv_MoveLinearInterpolateRelative(相对值直线插补控制) • MCv_MoveCircularInterpolateAbsolute(绝对值圆弧插补控制) • MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制) • MC_SetPosition(当前位置更改) • MCv_SetTorqueLimit(转矩限制值) • MC_CamIn(凸轮动作开始) • MC_GearIn(齿轮动作开始) • MC_CombineAxes(加减法定位) • MCv_BacklashCompensationFilter(齿隙补偿滤波器) • MCv_SmoothingFilter(平滑滤波器) • MCv_DirectionFilter(移动方向限制滤波器) • MCv_SpeedLimitFilter(速度限制滤波器) • MCv_AdvancedSync(高级同步控制) • MCv_MovePositioningData(多轴定位数据运行) • MCv_PositionControl(绝对位置跟踪控制) • MCv_CyclicPosition(运动循环位置控制) • MCv_CyclicVelocity(运动循环速度控制) • MCv_CyclicTorque(运动循环转矩控制)
	3	<ul style="list-style-type: none"> • MC_TouchProbe(触摸探针有效) • MCv_ChangeCycle(1周期当前值更改)
	—	<ul style="list-style-type: none"> • MC_GroupEnable(轴组有效) • MC_GroupDisable(轴组无效) • MC_Power(允许运行) • MC_SetOverride(倍率修调值设置) • MC_ReadParameter(参数读取) • MC_WriteParameter(参数写入) • MC_Reset(轴错误复位) • MC_GroupReset(轴组错误复位) • MC_AbortTrigger(触摸探针无效) • MC_CamTableSelect(凸轮表选择) • MCv_ReadProfileData(配置文件读取) • MCv_WriteProfileData(配置文件写入) • MCv_AllPower(所有轴允许运行) • MC_GroupSetOverride(轴组倍率修调值设置) • MCv_MotionErrorReset(运动错误复位) • MCv_SyncOperationCycles(控制运算周期同步) • MCv_AdvPositionPerCycleCalc(高级同步控制1周期当前位置计算) • MCv_AdvCamSetPositionCalc(高级同步控制凸轮指令当前位置计算) • MC_DigitalCamSwitch(数字凸轮开关输出)

执行指令(Execute)类型/有效(Enable)类型

运动控制FB有通过执行指令(Execute)执行及通过有效(Enable)执行的类型。

类型	执行指令(Execute)类型	有效(Enable)类型	其它类型
管理系统 FB	<ul style="list-style-type: none"> • MC_GroupEnable(轴组有效) • MC_GroupDisable(轴组无效) • MC_SetPosition(当前位置更改) • MCv_SetTorqueLimit(转矩限制值) • MC_WriteParameter(参数写入) • MC_Reset(轴错误复位) • MC_GroupReset(轴组错误复位) • MC_TouchProbe(触摸探针有效) • MC_AbortTrigger(触摸探针无效) • MC_CamTableSelect(凸轮表选择) • MCv_ChangeCycle(1周期当前值更改) • MCv_MotionErrorReset(运动错误复位) • MCv_AdvPositionPerCycleCalc(高级同步控制1周期当前位置计算) • MCv_AdvCamSetPositionCalc(高级同步控制凸轮指令当前位置计算) 	<ul style="list-style-type: none"> • MC_Power(允许运行) • MC_SetOverride(倍率修调值设置) • MC_ReadParameter(参数读取) • MCv_AllPower(所有轴允许运行) • MC_GroupSetOverride(轴组倍率修调值设置) • MCv_SyncOperationCycles(控制运算周期同步) • MC_DigitalCamSwitch(数字凸轮开关输出) 	—
运行系统 FB	<ul style="list-style-type: none"> • MC_Home(原点复位) • MC_Stop(强制停止) • MC_GroupStop(组强制停止) • MC_MoveAbsolute(绝对值定位) • MC_MoveRelative(相对值定位) • MC_MoveVelocity(速度控制) • MC_TorqueControl(转矩控制) • MCv_SpeedControl(速度控制(包含位置循环)) • MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制) • MCv_MoveLinearInterpolateRelative(相对值直线插补控制) • MCv_MoveCircularInterpolateAbsolute(绝对值圆弧插补控制) • MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制) • MC_CamIn(凸轮动作开始) • MC_GearIn(齿轮动作开始) • MC_CombineAxes(加减法定位) • MCv_MoveWait(轴控制开始等待) • MCv_GroupMoveWait(轴组控制开始等待) • MCv_MovePositioningData(多轴定位数据运行) • MCv_PositionControl(绝对位置跟踪控制) • MCv_CyclicPosition(运动循环位置控制) • MCv_CyclicVelocity(运动循环速度控制) • MCv_CyclicTorque(运动循环转矩控制) 	<ul style="list-style-type: none"> • MCv_BacklashCompensationFilter(齿隙补偿滤波器) • MCv_SmoothingFilter(平滑滤波器) • MCv_DirectionFilter(移动方向限制滤波器) • MCv_SpeedLimitFilter(速度限制滤波器) • MCv_AdvancedSync(高级同步控制) 	• MCv_Jog(JOG运行)
常规FB	<ul style="list-style-type: none"> • MCv_ReadProfileData(配置文件读取) • MCv_WriteProfileData(配置文件写入) 	—	—

通过执行指令(Execute)及有效(Enable)的运动控制FB的基本动作如下所示。但是,根据运动控制FB,规格有可能不同。

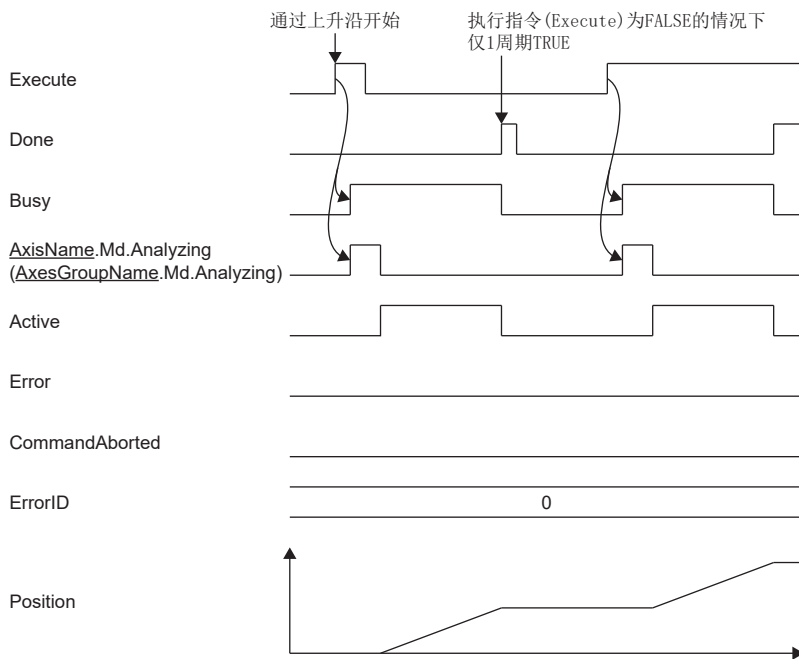
■通过执行指令(Execute)类型的运动控制FB的基本动作

- 执行指令(Execute)类型的运动控制FB在执行指令(Execute)的上升沿读取输入参数,开始动作。一旦开始动作,即使将执行指令(Execute)置为FALSE也将继续动作直至完成为止。
- 开始动作时,执行中(Busy)、执行完成(Done)、错误(Error)、取消受理(CommandAborted)的输出变量中只有1个变为TRUE。
- 执行完成(Done)、错误(Error)、错误代码(ErrorID)、取消受理(CommandAborted)在执行指令(Execute)的下降沿复位。执行中(Busy)、控制中(Active)不受影响。
- 在动作中更改了输入参数的情况下,通过执行指令(Execute)的重新启动(重启)或使用了连续更新(ContinuousUpdate)的连续更新,更改将被反映。
- 执行中(Busy)的上升沿起解析中(AxisName.Md.Analyzing/AxesGroupName.Md.Analyzing)变为TRUE,动作开始时解析中(AxisName.Md.Analyzing/AxesGroupName.Md.Analyzing)变为FALSE。
- 在脉冲中使用执行指令(Execute)的情况下,执行完成(Done)仅在1周期内变为TRUE。

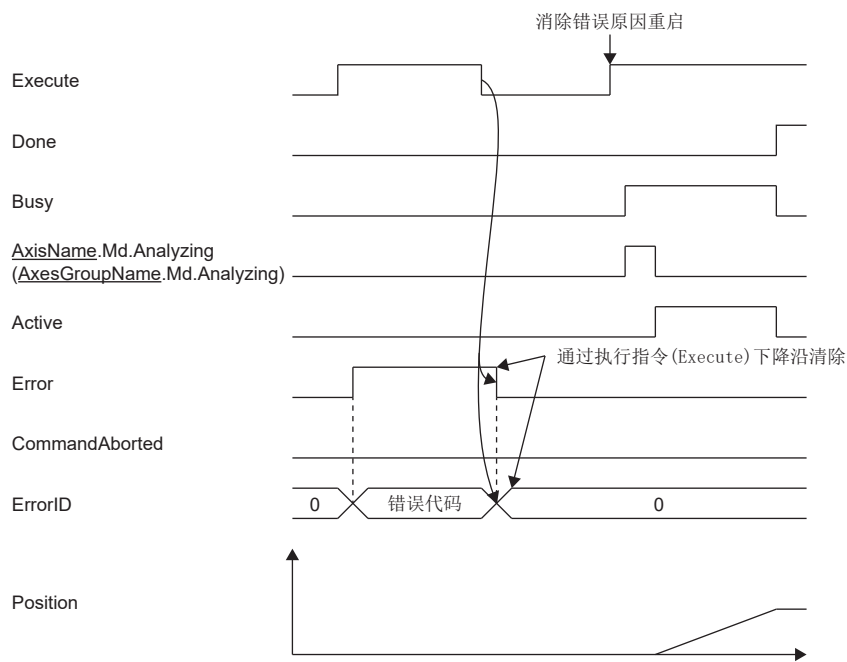
例

通过执行指令(Execute)类型的运动控制FB的时序图如下所示。

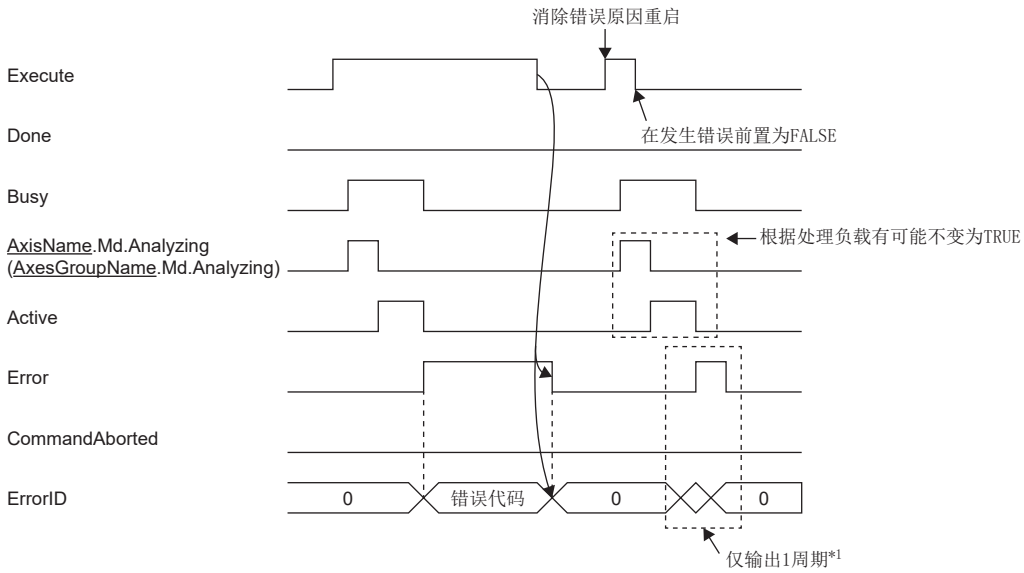
• 正常时



• 输入输出变量异常时



• 输入变量异常时



*1 运动控制FB的结束条件(执行指令(Execute)为FALSE)成立，因此变为上述动作。与轴无关的运动控制FB或不减速停止的运动控制FB的情况下，仅1个周期错误(Error)变为TRUE，输出错误代码(ErrorID)。需要进行减速停止的运动控制FB的情况下，在轴减速停止之前错误(Error)将变为TRUE，保持错误代码(ErrorID)，轴停止完成时错误(Error)将变为FALSE，并清除错误代码(ErrorID)。

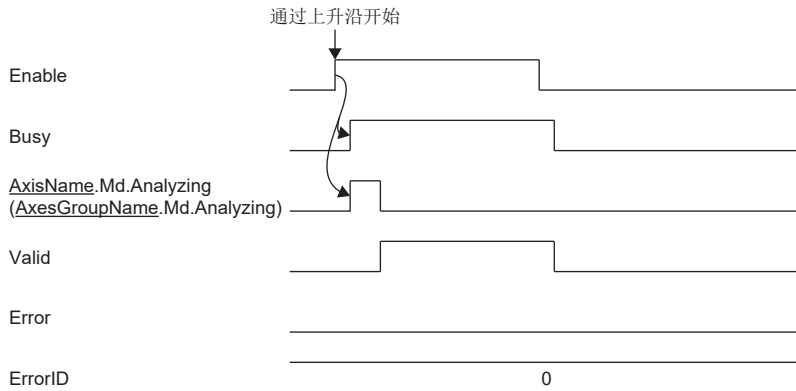
■通过有效 (Enable) 类型的运动控制FB的基本动作

- 有效 (Enable) 类型的运动控制FB在有效 (Enable) 为TRUE期间，将继续重复执行。
- 输出值有效 (Valid) 表示输出为有效值。输出值有效 (Valid) 变为FALSE以后，各输出不变化。
- 输出值有效 (Valid)/有效中 (Enabled)/执行中 (Busy)、错误 (Error)、执行中断 (CommandAborted) 的输出变量中只有1个变为TRUE。
- 执行中 (Busy) 的上升沿起解析中 (AxisName.Md.Analyzing/AxesGroupName.Md.Analyzing) 变为TRUE，动作开始时解析中 (AxisName.Md.Analyzing/AxesGroupName.Md.Analyzing) 变为FALSE。
- 上述未包含的输出变量在有效 (Enable) 的下降沿复位。(复位的时机与输出值有效 (Valid)/有效中 (Enabled)/执行中 (Busy)、错误 (Error)、执行中断 (CommandAborted) 相同。)

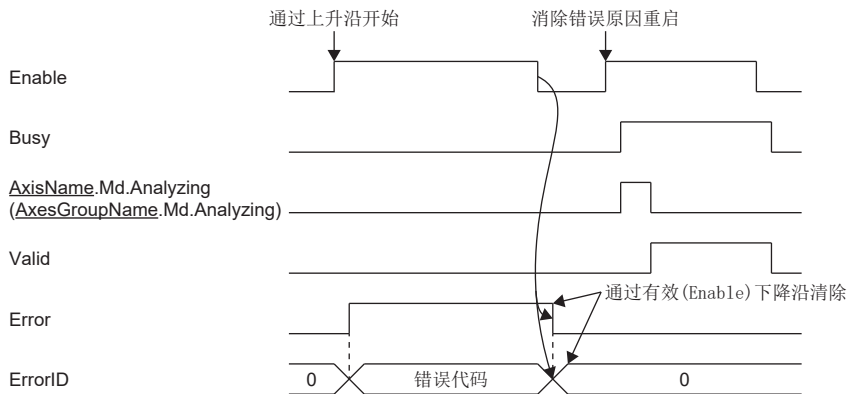
例

通过有效 (Enable) 类型的运动控制FB的时序图如下所示。

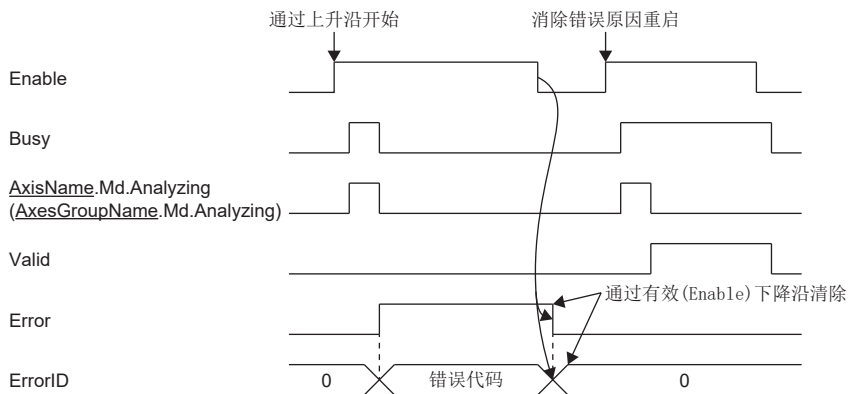
- 正常时



- 输入输出变量异常时



- 输入变量异常时



运动控制的类型

根据运行系统运动控制FB，可执行的轴、轴组的控制有下述类型。

大分类	中分类	小分类	内容
轴控制	单轴控制	定位控制	轴状态 (<u>AxisName</u> . Md. AxisStatus) 置为“5: 定位运行中 (DiscreteMotion)”，以移动到目标位置的控制。
		连续控制	轴状态 (<u>AxisName</u> . Md. AxisStatus) 置为“6: 连续动作运行中 (ContinuousMotion)”，以进行连续控制的控制。
		同步控制	主轴 (Master) 和从轴 (Slave) 作为输入输出变量，从轴 (Slave) 的轴状态 (<u>AxisName</u> . Md. AxisStatus) 置为“7: 同步运行中 (SynchronizedMotion)”，以进行同步控制的控制。
		手动控制	有效 (Enable) 类型中具有执行完成 (Done) 输出变量的定位控制FB、连续控制FB、同步控制FB。
		原点复位控制	轴状态 (<u>AxisName</u> . Md. AxisStatus) 置为“3: 原点复位中 (Homing)”，以进行连续控制的控制。
轴组控制	多轴控制	定位控制	轴组状态 (<u>AxesGroupName</u> . Md. GroupStatus) 置为“5: 动作中 (GroupMoving)”，以移动到目标位置的控制。


错误处理

运动控制FB的执行时发生错误时错误 (Error) 将变为TRUE，并将错误代码输出到错误代码 (ErrorID) 中。此时，轴的情况下轴状态 (AxisName. Md. AxisStatus) 转变为“1: 错误停止中 (ErrorStop)”，轴组的情况下轴组状态 (AxesGroupName. Md. GroupStatus) 转变为“1: 错误停止中 (GroupErrorStop)”。

可动作的轴的状态转变为“1: 错误停止中 (ErrorStop)”时，所有的缓冲FB均将中断。中断的运动控制FB的错误 (Error) 变为TRUE。

此后，轴或轴组启动时，需要执行错误复位。

关于轴或轴组的错误确认方法及错误复位方法，请参阅下述手册。

 所使用的控制器的用户手册

要点

在错误代码 (ErrorID) 中，输出与下述某个相同的值 (根据控制而不同)。但是，不输出事件代码。


- 轴错误代码 (AxisName. Md. ErrorID)
- 轴组错误代码 (AxesGroupName. Md. ErrorID)
- 运动部最新系统错误代码 (MotionSystem. Md. ErrorID)

控制中使用的单位

对于运动系统内的处理位置、速度、加减速速度、Jerk的单位，按照使用的轴的单位系统。

运动系统内的处理位置及速度等的控制值有下述类型。

关于单位系统及位置、速度相关的控制值的详细内容，请参阅下述手册。


 所使用的控制器的用户手册

类型	内容
指定值	是基于至运动控制FB的输入的值 (目标值)。 (指定位置、指定速度等)
指令当前值	是通过运动运算创建的当前的控制值。 (指令当前位置、指令当前速度等)
反馈值	是实轴中将来自于设备的反馈值转换为轴的单位系统的值。 (反馈位置、反馈速度等)

运动控制FB的输入输出变量

以下对运动控制FB的输入输出变量有关内容进行说明。
运动控制FB需要定义输入输出变量、输入变量、输出变量。

输入输出变量

设置轴信息(Axis)、轴组信息(AxesGroup)等要控制的驱动器模块的轴、轴组。
轴、轴组通过工程工具的轴设置、轴组设置生成轴，并进行初始化。
设置的轴、轴组作为轴变量、轴组变量被分配到全局标签数据中。
关于轴、轴组的详细内容，请参阅下述手册。
所使用的控制器的用户手册

输入变量

设置目标位置及指令速度等的动作条件。

输出变量


输出运动控制FB的状态、驱动器模块的状态、错误的有无等。

分类

输入输出变量、输入变量、输出变量的分类如下所示。

变量	分类
输入输出变量	VAR_IN_OUT
输入变量	VAR_INPUT
输出变量	VAR_OUTPUT

数据类型

变量根据位长、处理方法、值的范围等对变量的类型进行分类。
关于数据类型，请参阅下述章节。
 28页 数据的指定方法

输入参数的省略

省略了运动控制FB的输入时，各运动控制FB中定义的初始值将适用。关于初始值的详细内容，请参阅使用的运动控制FB。
在多重启动的运动控制FB中省略了速度等输入的情况下，将继承之前控制的运动控制FB的输入值。

输入输出的刷新时机

运动控制FB的参数在运动控制FB的调用时机进行刷新。要与运算周期同步控制运动控制FB的输入输出，应从与运算周期相同周期的恒定周期程序调用运动控制FB。

输入输出编号的指定

使用运动控制FB时，需要在使用的输入输出变量(例：输入输出No. (AxisName. AxisRef. StartIO))中设置对象的输入输出编号。(运动控制FB中使用的输入输出变量(例：输入输出No. (AxisName. AxisRef. StartIO)))只能使用默认值。)

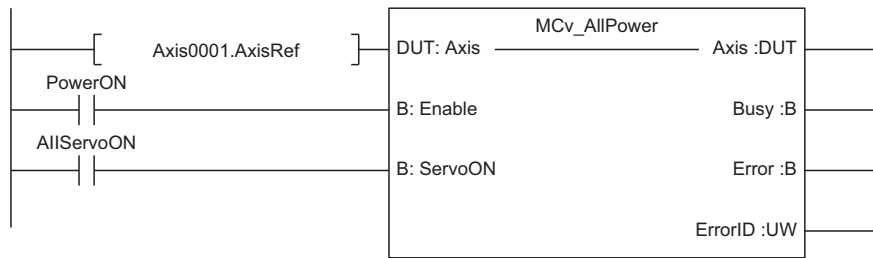
设置输入输出编号时，使用轴变量或轴组变量的AXIS_REF(轴变量为AxisName. AxisRef，轴组为AxesGroupName. AxesGroupRef)。

轴变量或轴组变量的AXIS_REF的构件(轴No. (AxisNo)、轴组No. (GroupNo)、输入输出No. (StartIO))中，由于输入输出编号已被设置为结构体，无需注意也可以通过设置轴的标签来设置输入输出编号。

设置输入输出编号时，推荐使用轴信息(AxisName. AxisRef)的标签进行设置。此外，哪个轴信息(AxisName. AxisRef)的标签都可以设置。

例

使用轴1(Axis0001)的标签，执行MCv_AllPower(全部轴可运行时)



*1 轴信息(AxisName. AxisRef)的标签中，包含下述构件信息。

名称	设定值	标签名
轴No.	1	Axis0001.AxisRef.AxisNo
输入输出No.	3E00H	Axis0001.AxisRef.StartIO

运动控制FB的注意事项

- 运动控制FB的输入输出参数的刷新是在运动控制FB的调用时机进行的，但实际执行控制的周期根据运动控制FB而有所不同。例如，运行系统FB与运动控制FB调用任务(正常/恒定周期)无关，以运算周期进行控制。详细内容，请参阅所使用的运动控制FB。
- 运动控制FB的输出参数刷新的时机与轴变量等相关标签的刷新时机不同，因此各功能中记载的时序图与实际信号的变化可能不一致。应考虑各信号的刷新周期创建程序。
- 运动控制FB的动作中(执行中(Busy)为TRUE状态)，应调用运动控制FB。如果不在程序内调用，将继续进行控制，但即使更改输入变量的值也不会反映到控制中，执行完成或错误等输出变量的更新将无法进行。
- 在初始执行型程序中，运动控制FB的动作中FB调用可能会停止，因此请勿使用运动控制FB。
- 与FOR~NEXT指令之间及控制语法内(IF语句、FOR语句、CASE语句等)组合使用的情况下，应确认执行中(Busy)及执行完成(Done)等是运动控制FB在运动控制FB的执行完成之前被调用。
- 运动控制FB在将执行指令(Execute)或有效(Enable)置为TRUE时获取输入值。因此，应在将执行指令(Execute)或有效(Enable)置为TRUE之前设置输入值。此外，通过运动控制FB的重启或连续更新更改多个输入参数时，为了使各参数的获取时机一致，应在与运动控制FB调用任务相同的任务中进行更改。
- 将运动控制FB的执行指令(Execute)置为TRUE后，将执行指令(Execute)置为FALSE的情况下，应在执行中(Busy)变为TRUE状态后之后再置为FALSE。
- 在运动控制功能中，轴的当前位置及运动控制FB的目标位置等使用了实数数据(浮点数据)。因此，有可能包含运算误差。例如，重复执行MC_MoveRelative(相对值定位)之类的相对位置指定的定位的情况下，指令当前位置与设置的移动量之间有可能累计运算误差。运算误差的累计会带来问题的情况下，应执行MC_MoveAbsolute(绝对值定位)之类的绝对位置指定的定位。此外，通过设置为运动控制功能关联的实数型的参数中不包含小数点以下，可以抑制功能内部实数数据处理导致发生的运算误差。关于运动控制FB使用的实数数据(浮点数据)的详细内容，请参阅下述章节。

☞ 37页 实数数据(浮点数据)

- 请勿将1个运动控制FB实例同时从多个程序(包含中断程序的并联执行)中调用。不保证同时调用时的动作。

43.2 运动控制FB的配置

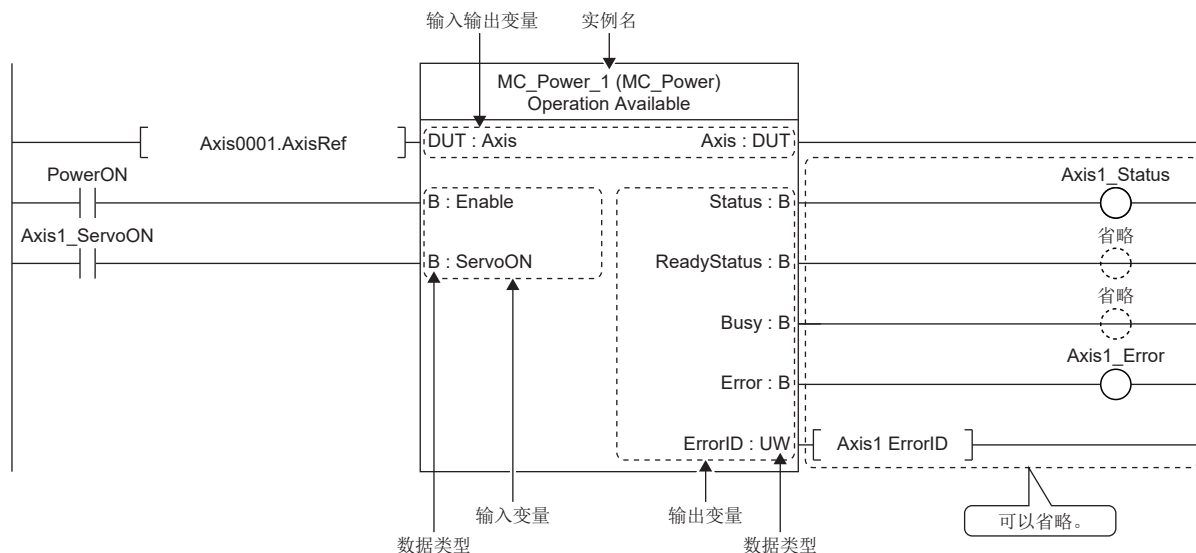
以下对运动控制FB的各程序语言的记述方法进行说明。

梯形图

在梯形图语言中，按下述方式配置运动控制FB。

例

MC_Power (允许运行)的情况下



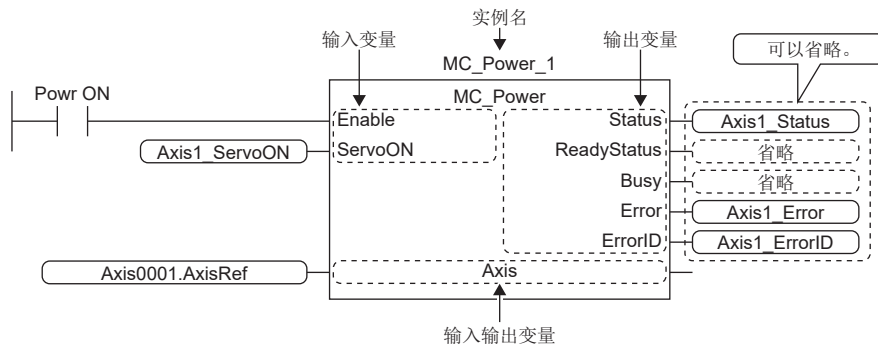
名称	内容
实例名	是对每个FB添加的实例名。 可以更改实例名。
输入输出变量	设置轴信息 (Axis)、轴组信息 (AxesGroup) 等要控制的驱动器模块等的轴变量名。
输入变量	设置目标位置及指令速度等的动作条件。 可以省略设置。省略的情况下，输入变量变为初始值。
输出变量	输出FB的状态及驱动器模块的状态。
数据类型	表示输入输出变量、输入变量、输出变量的数据类型。 关于输入输出变量、输入变量、输出变量的数据类型，请参阅下述章节。 ☞ 28页 数据的指定方法

FBD/LD(功能块图表/梯形图)

在FBD/LD语言中，按下述方式配置运动控制FB。

例

MC_Power (允许运行)的情况下



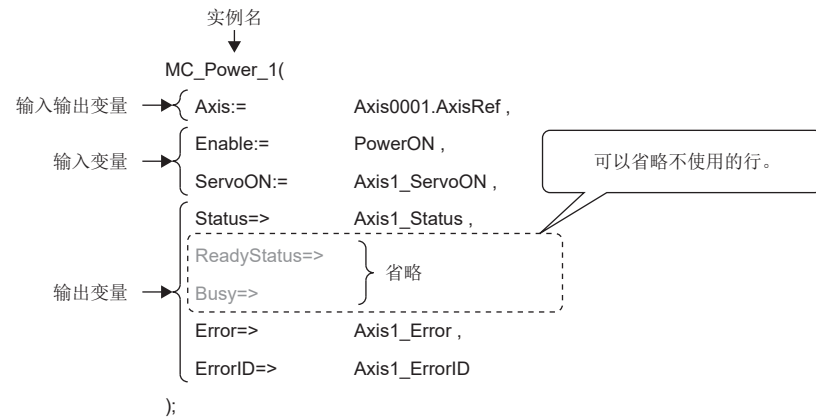
名称	内容
实例名	是对每个FB添加的实例名。 可以更改实例名。
输入输出变量	设置轴信息(Axis)、轴组信息(AxesGroup)等要控制的驱动器模块等的轴变量名。
输入变量	设置目标位置及指令速度等的动作条件。 可以省略设置。省略的情况下，输入变量变为初始值。
输出变量	输出FB的状态及驱动器模块的状态。

ST(结构化文本)

在ST语言中，按下述方式配置运动控制FB。

例

MC_Power (允许运行)的情况下



名称	内容
实例名	是对每个FB添加的实例名。 可以更改实例名。
输入输出变量	设置轴信息(Axis)、轴组信息(AxesGroup)等要控制的驱动器模块等的轴变量名。
输入变量	设置目标位置及指令速度等的动作条件。 可以省略设置。省略的情况下，输入变量变为初始值。
输出变量	输出FB的状态及驱动器模块的状态。

44 运动控制FB变量一览

以下对运动控制FB使用的变量的一览进行说明。

44.1 变量一览

运动控制中处理的变量有下述类型。

变量	内容	参阅	
轴变量	是由轴类型等的参数数据及轴的当前位置等的监视数据组成的与轴相关的变量。	1387页 轴变量	
轴组变量	是由配置轴等的参数数据及轴组状态等的监视数据组成的与轴组相关的变量。	1408页 轴组变量	
系统变量	是由与插件相关的参数数据及运算周期等的监视数据组成的与系统相关的变量。	1414页 系统变量	
高级同步控制变量	输入轴变量	是由高级同步控制的输入轴的参数数据及轴的当前位置等的监视数据组成的与输入轴相关的变量。	1418页 输入轴变量
	输出轴变量	是由高级同步控制的输出轴的参数数据及轴的当前位置等的监视数据组成的与输出轴相关的变量。	1420页 输出轴变量
	系统变量	是由与高级同步控制插件相关的参数数据及监视数据组成的与系统相关的变量。	1425页 系统变量(高级同步控制)
	其他结构体的变量	是高级同步控制中使用的除上述以外分类的结构体的变量。	1445页 其它结构体的变量(高级同步控制)
其他结构体的变量	是除上述以外分类的结构体的变量。	1426页 其他结构体的变量	

要点

本手册中记载的变量的带下划线的变量部分表示由用户定义变量名。

AxisName.AxisRef.AxisNo

↓
用户定义的变量

变量一览表的阅读方法

一览表的各项目的阅读方法如下所示。

■获取

项目	内容
运算周期	各运算周期的执行时机
正常	运动系统的正常任务的执行时机
服务	运动服务处理控制的执行时机
即时	发生了事件时
启动时	FB启动时
停止时	FB停止时
系统启动时	运动系统启动时
设备连接时	设备连接时
就绪ON	将“可编程控制器就绪(MotionSystem.Cd.SequenceReady)”置为ON时
系统基本周期	系统基本周期处理的执行时机
运算周期(GroupEnable中)	各运算周期的执行时机(GroupEnable中)
轴组有效时	轴组变为有效的时机
运算周期(FB启动中)	执行运算周期处理时机(FB启动中)
凸轮数据第0点通过时	1周期当前位置通过了当前执行中的凸轮数据的第0点时 通过凸轮分辨率512时的第0点位置的时机

■属性

显示	内容
LIST_WRITE_BACK	标签列表(初始值可回写)
LIST_READ_ONLY	标签列表(初始值不可回写)

轴变量

对于轴变量，根据轴类型将为下述数据类型。

轴类型	数据类型
实际驱动轴	AXIS_REAL
实际编码器轴	AXIS_ENCODER
虚拟驱动轴	AXIS_VIRTUAL
虚拟编码器轴	AXIS_VIRTUAL_ENCODER
虚拟连接轴	AXIS_VIRTUAL_LINK

要点

根据使用的轴类型，可设置的轴变量有所不同。(☞ 1403页 轴变量中可对各轴类型设置的变量)

各轴的数据类型具有下述构件。

构件名	数据类型*1*2	内容	参照目标
AxisRef	AXIS_REF	是运动控制用功能块的输入/输出用的数据结构。 与轴类型无关，为固定的类型。	1387页 AxisName.AxisRef . (轴信息)
PrConst	AXIS_□_PRM_CONST	存储轴的参数数据(常数)。 轴变量初始化时展开设定值。 轴变量初始化后不实施至控制的重新获取。	1388页 AxisName.PrConst . (轴参数常数)
Pr	AXIS_□_PRM	存储轴的参数数据。 轴变量初始化时展开初始值。 轴变量初始化后也实施至控制的重新获取。 根据参数至控制的获取时机有所不同。	1392页 AxisName.Pr . (轴参数)
Md	AXIS_□_MONI	存储轴的监视数据。 以各监视数据中确定的周期实施刷新。	1395页 AxisName.Md . (轴监视数据)
Cd	AXIS_□_CMD	存储轴控制用指令数据。 各控制运算周期中获取最新的值用于控制。	1402页 AxisName.Cd . (轴控制数据)

*1 □: 各轴类型的数据类型

*2 根据轴类型，数据类型构件有所不同。

AxisName.AxisRef. (轴信息)

变量名	名称	获取	数据类型	属性	内容
AxisNo	轴No.	—	WORD(UINT)	LIST_WRITE_BACK	设置轴No.。 • 0: 未设置 • 1~10000: 设置轴No.
StartIO	输入输出No.	—	WORD(HEX)	LIST_WRITE_BACK	设置输入输出No.。 默认情况下，将自动输入3E0H。 • 000H~0FFH: 起始输入输出编号(以16进制数4位数表示时的前3位数)

AxisName.PrConst. (轴参数常数)

变量名	名称	获取	数据类型	属性	内容
AddressOfStation	站地址设置	系统启动时	WSTRING(63)	LIST_WRITE _BACK	以字符串设置设备站的网络地址(IP地址)。类似于多轴驱动器模块,1个站中包含多个逻辑轴的情况下,设置用于识别逻辑轴的多点编号。 ■单轴设备的情况下 192.168.3.1 └── IP地址 ■多轴设备的情况下 <例> MR-J5W-G的C轴的情况下 192.168.3.1#2 └── 多点编号 └── IP地址 * 以“#+编号(10进制数)”设置多点编号。 • #0: A轴 • #1: B轴 • #2: C轴 * 多点编号可以省略。省略的情况下,将视为“#0”。
AxisType	轴类型设置	系统启动时	INT (MC_AXIS_TYPE) ☞ 1458页 MC_AXIS_TYPE[M]	LIST_WRITE _BACK	设置轴类型。 • 0: 实际驱动轴(DriveAxis) • 2: 实际编码器轴(EncoderAxis) • 3: 虚拟驱动轴(VirtualDriveAxis) • 4: 虚拟编码器轴(VirtualEncoderAxis) • 5: 虚拟连接轴(VirtualLinkAxis)
Encoder_AxisType	实际编码器轴类型设置	系统启动时	INT (MC_ENCODER_AXIS_TYPE) ☞ 1458页 MC_ENCODER_AXIS_TYPE[M]	LIST_WRITE _BACK	设置实际编码器轴的类型。 • 1: 经由驱动器模块(Drive)
Encoder_CounterDisableSignal	计数器禁用信号	系统启动时	SIGNAL_SELECT ☞ 1431页 SIGNAL_SELECT(信号选择)	LIST_WRITE _BACK	设置用于切换为计数器禁用的信号。计数器禁用信号(Encoder_CounterDisableSignal)固有的设置及动作如下所示。 ■对象(Target) • 数据类型中只能设置[VAR]、[DEV]、[CONST]。 • [VAR]、[DEV]指定时参照的设备站的网络同步通信设置,只能指定“同步”。 ■信号检测方法(Detection) 只允许设置下述等级检测。 • 0: TRUE时检测(HighLevel) • 1: FALSE时检测(LowLevel) ■补偿时间(CompensationTime) 只能设置“0.0[s]”。 ■筛选时间(FilterTime) 设置范围为“0.0~5.0[s]”。 * 设置了超出范围的值的情况下,将变为超出参数范围(轴)(错误代码:1A60H)。
Encoder_RingCout_LowerValue	编码器环形计数器下限值	系统启动时	DINT	LIST_WRITE _BACK	设置编码器环形计数器的下限值。 ■PosActualValue为1字 • -32768~32767 * 设置了超出1字的范围的值的情况下,将变为超出参数范围(轴)(错误代码:1A60H)。 ■PosActualValue为2字或省略 • -2147483648~2147483647
Encoder_RingCout_UpperValue	编码器环形计数器上限值	系统启动时	DINT	LIST_WRITE _BACK	设置编码器环形计数器的上限值。 ■PosActualValue为1字 • -32768~32767 * 设置了超出1字的范围的值的情况下,将变为超出参数范围(轴)(错误代码:1A60H)。 ■PosActualValue为2字或省略 • -2147483648~2147483647

变量名	名称	获取	数据类型	属性	内容
HwStrokeLimit_FlsSignal	上限限位信号	系统启动时	SIGNAL_SELECT ☞ 1431页 SIGNAL_SELECT(信号选择)	LIST_WRITE _BACK	<p>设置使用上限限位信号(FLS)的信号。根据作为外部信号使用的变量及控制器管理的设备的状态,有可能作为信号检测处理。上限限位信号(HwStrokeLimit_FlsSignal)固有的设置及动作如下所示。</p> <p>■输入输出编号(StartIO) 忽略输入值。</p> <p>■对象(Target) 无指定时,将判断为信号无效且置为始终信号非检测状态。此外,不获取硬件行程限位倍率修调(AxisName.Md.HwStrokeLimit_Override)。数据类型仅可指定[VAR]、[DEV]、[CONST]。</p> <p>* 设置了不能使用的值的情况下,为超出参数范围(轴)(错误代码:1A60H)。可通过下述详细代码确认原因。</p> <ul style="list-style-type: none"> • 设置了不正确的字符串(详细代码:0001H) • 设置了不能使用的数据类型(详细代码:0002H) • 指定了不能使用的类型(详细代码:0003H) • 标签指定时指定了数值型以外的类型(详细代码:0004H) • 设置了不存在的型号(详细代码:0005H) • 设置了外部设备访问无效的全局标签(详细代码:0005H) • 超出了字符数的上限(详细代码:0007H) • 在参照的设备站的网络同步通信设置中设置了“Do not Synchronize(不同步)”(详细代码:8002H) • 设置了不支持的软元件(详细代码:8002H) • 设置了超出范围的软元件编号(详细代码:8002H) <p>■信号检测方法(Detection) 只允许指定下述等级检测。</p> <ul style="list-style-type: none"> • 0: TRUE时检测(HighLevel) • 1: FALSE时检测(LowLevel) <p>* 设置了超出范围的值的情况下,将变为超出参数范围(轴)(错误代码:1A60H(详细代码:8003H))。</p> <p>■补偿时间(CompensationTime) 忽略输入值。</p> <p>■筛选时间(FilterTime) 设置范围为0.0~+5.0[s]。</p> <p>* 指定了超出范围的值的情况下,为超出各轴信号的筛选时间设置范围警告(事件代码:00D24H),且筛选时间将以“0.0”执行动作。</p>

变量名	名称	获取	数据类型	属性	内容
HwStrokeLimit_RlsSignal	下限限位信号	系统启动时	SIGNAL_SELECT ☞ 1431页 SIGNAL_SELECT(信号选择)	LIST_WRITE _BACK	<p>设置使用下限限位信号(RLS)的信号。</p> <p>根据作为外部信号使用的变量及控制器管理的设备的状态,有可能作为信号检测处理。</p> <p>下限限位信号(HwStrokeLimit_RlsSignal)固有的设置及动作如下所示。</p> <p>■输入输出编号(StartIO) 忽略输入值。</p> <p>■对象(Target) 无指定时,将判断为信号无效且置为始终信号非检测状态。此外不获取控制数据。</p> <p>数据类型仅可指定[VAR]、[DEV]、[CONST]。</p> <p>* 设置了不能使用的值的情况下,为超出参数范围(轴)(错误代码:1A60H)。可通过下述详细代码确认原因。</p> <ul style="list-style-type: none"> • 设置了不正确的字符串(详细代码:0001H) • 设置了不能使用的数据类型(详细代码:0002H) • 指定了不能使用的类型(详细代码:0003H) • 标签指定时指定了数值型以外的类型(详细代码:0004H) • 设置了不存在的型号(详细代码:0005H) • 设置了外部设备访问无效的全局标签(详细代码:0005H) • 超出了字符数的上限(详细代码:0007H) • 在参照的设备站的网络同步通信设置中设置了“Do not Synchronize(不同步)”(详细代码:8002H) • 设置了不支持的软元件(详细代码:8002H) • 设置了超出范围的软元件编号(详细代码:8002H) <p>■信号检测方法(Detection) 只允许指定下述等级检测。</p> <ul style="list-style-type: none"> • 0: TRUE时检测(HighLevel) • 1: FALSE时检测(LowLevel) <p>* 设置了超出范围的值的情况下,将变为超出参数范围(轴)(错误代码:1A60H(详细代码:8003H))。</p> <p>■补偿时间(CompensationTime) 忽略输入值。</p> <p>■筛选时间(FilterTime) 设置范围为0.0~+5.0[s]。</p> <p>* 指定了超出范围的值的情况下,为超出各轴信号的筛选时间设置范围警告(事件代码:00D24H),且筛选时间将以“0.0”执行动作。</p>
OperationCycle	控制周期设置	系统启动时	INT	LIST_WRITE _BACK	<p>设置轴的控制运算周期。</p> <p>■实际轴 自动选择与分配给轴的设备站的通信周期相同的运算周期。</p> <p>没有相应的运算周期的情况下,将发生周期分配不正确(轴)(错误代码:1B2BH)。</p> <ul style="list-style-type: none"> • 0: 自动设置 <p>■虚拟轴、或者站地址为未设置的实际轴</p> <ul style="list-style-type: none"> • 0: 以第1运算周期运行 • 1: 以第1运算周期运行 • 2: 以第2运算周期运行 • 3: 以第3运算周期运行
PosRestoration_AbsPosEnable	绝对位置管理设置	系统启动时	INT (MC_ABS_SYSTEM) ☞ 1458页 MC_ABS_SYSTEM[M]	LIST_WRITE _BACK	<p>设置绝对位置管理。</p> <ul style="list-style-type: none"> • 0: 不使用绝对位置系统(ABSDisabled) • 1: 使用绝对位置系统(Enabled) • -1: 自动设置(从连接设备获取)(Auto)
PosRestoration_RestorationBase	当前位置恢复基准位置设置	系统启动时	INT (MC_POS_RESTORATION_BASE) ☞ 1460页 MC_POS_RESTORATION_BASE	LIST_WRITE _BACK	<p>使用绝对位置系统的情况下,设置当前位置恢复的基准位置。</p> <ul style="list-style-type: none"> • 0: 备份位置 • 1: 原点 • -1: 自动判定
RingCount_Enable	环形计数器有效选择	系统启动时	BOOL	LIST_WRITE _BACK	<p>设置环形计数器有效/无效。</p> <ul style="list-style-type: none"> • FALSE: 无效 • TRUE: 有效
RingCount_LowerValue	环形计数器下限值	系统启动时	LREAL	LIST_WRITE _BACK	<p>设置环形计数器下限值。</p> <ul style="list-style-type: none"> • -10000000000.0~10000000000.0 <p>* 环形计数器无效时不获取</p>

变量名	名称	获取	数据类型	属性	内容
RingCount_UpperValue	环形计数器上限值	系统启动时	LREAL	LIST_WRITE _BACK	设置环形计数器上限值。 • -10000000000.0~10000000000.0 * 环形计数器无效时不获取
SlaveEmulate_Enable	轴模拟有效	系统启动时	BOOL	LIST_WRITE _BACK	设置是否作为模拟轴使用。 • FALSE: 无效 • TRUE: 有效
SlaveObject(系统用)	对象数据	系统启动时	SLAVE_OBJECT_REAL ☞ 1443页 SLAVE_OBJECT_REAL(系统用)	LIST_WRITE _BACK	在实际驱动轴上设置对象数据。 * 本变量为系统用变量, 因此请勿使用
SlaveObject	对象数据	系统启动时	SLAVE_OBJECT_VIRTUAL_ENCODER ☞ 1444页 SLAVE_OBJECT_VIRTUAL_ENCODER	LIST_WRITE _BACK	在虚拟编码器轴上设置对象数据。
TorqueLimit_Max	转矩限制最大值	系统启动时	LREAL	LIST_WRITE _BACK	设置可作为转矩限制值指定的最大值。(正方向/负方向相同。) • 0.0~1000.0[%]
TorqueLimit_NegativeInitial	负方向转矩限制初始值	系统启动时	LREAL	LIST_WRITE _BACK	设置负方向转矩限制值 (AxisName.Cd.TorqueLimit_Negative)的初始值。 • 0.0~1000.0[%]
TorqueLimit_PositiveInitial	正方向转矩限制初始值	系统启动时	LREAL	LIST_WRITE _BACK	设置正方向转矩限制值 (AxisName.Cd.TorqueLimit_Positive)的初始值。 • 0.0~1000.0[%]

AxisName.Pr. (轴参数)

变量名	名称	获取	数据类型	属性	内容
AccelerationLimit	加速度限制值	就绪ON	LREAL	LIST_WRITE _BACK	<p>设置加速度限制值。</p> <ul style="list-style-type: none"> • 0.0000、0.0001~2147483647.0[U/s²] * 加减速方式为“1: 加减速时间恒定方式 (mcFixedTime)”的情况下, 不进行加速度的限制。 * “0.0000”的情况下, 不进行加速度的限制。 * 设置了小于0.0001的正数的情况下, 将其作为“0.0000”获取。
AccelerationZeroBehavior	启动时加减速速度0指定时动作选择	就绪ON	INT (MC_ACC_ZERO_MODE) ☞ 1459页 MC_ACC_ZERO_MODE[M]	LIST_WRITE _BACK	<p>设置启动时, 在加速度、减速度或加减速时间中设置了“0.0”时的动作。</p> <ul style="list-style-type: none"> • -1: 错误(不启动)(ACCErrror) • 1: 最大加减速(MaximumAcceleration)
CmdInPos_Width	指令到位宽度	就绪ON	LREAL	LIST_WRITE _BACK	<p>设置指令到位宽度。</p> <ul style="list-style-type: none"> • 0.0: 功能无效 • 0.000000001~10000000000.0: 功能有效 * 设置了小于0.000000001的正数的情况下, 将其作为“0.0”获取。
DecelerationLimit	减速度限制值	就绪ON	LREAL	LIST_WRITE _BACK	<p>设置减速度限制值。</p> <ul style="list-style-type: none"> • 0.0000、0.0001~2147483647.0[U/s²] * 加减速方式为“1: 加减速时间恒定方式 (mcFixedTime)”的情况下, 不进行加速度的限制。 * “0.0000”的情况下, 不进行减速度的限制。 * 设置了小于0.0001的正数的情况下, 将其作为“0.0000”获取。
Drive_UnitConvRatioNumerator	驱动器单位转换分子	就绪ON	DWORD(UDINT)	LIST_WRITE _BACK	<p>设置用于将运动系统的指令单位转换为驱动器模块的指令单位的分子。</p> <ul style="list-style-type: none"> • 1~2147483647
Drive_UnitConvRatioDenominator	驱动器单位转换分母	就绪ON	DWORD(UDINT)	LIST_WRITE _BACK	<p>设置用于将运动系统的指令单位转换为驱动器模块的指令单位的分母。</p> <ul style="list-style-type: none"> • 1~2147483647
ForcedStop_Signal	紧急停止信号	就绪ON	SIGNAL_SELECT ☞ 1431页 SIGNAL_SELECT(信号选择)	LIST_WRITE _BACK	<p>设置使用紧急停止的信号。</p> <p>根据作为外部信号使用的变量及控制器管理的设备的状态, 有可能作为信号检测处理。关于详细情况, 请参阅下述章节。</p> <p>紧急停止信号(ForcedStop_Signal)固有的设置及动作如下所示。</p> <p>■输入输出编号(StartIO) 忽略输入值。</p> <p>■对象(Target) 无指定时, 将判断为信号无效且置为始终信号非检测状态。 数据类型仅可指定[VAR]、[DEV]、[CONST]。</p> <ul style="list-style-type: none"> * 设置了不能使用的值的情况下, 为超出参数范围(轴)(错误代码: 1A60H)。可通过下述详细代码确认原因。 • 设置了不正确的字符串(详细代码: 0001H) • 设置了不能使用的数据类型(详细代码: 0002H) • 指定了不能使用的类型(详细代码: 0003H) • 标签指定时指定了数值型以外的类型(详细代码: 0004H) • 设置了不存在的型号(详细代码: 0005H) • 设置了外部设备访问无效的全局标签(详细代码: 0005H) • 超出了字符数的上限(详细代码: 0007H) • 在参照的设备站的网络同步通信设置中设置了“Do not Synchronize(不同步)”(详细代码: 8002H) • 设置了不支持的软件(详细代码: 8002H) • 设置了超出范围的软件编号(详细代码: 8002H) <p>■信号检测方法(Detection) 只允许指定下述的等级检测。</p> <ul style="list-style-type: none"> • 0: TRUE时检测(HighLevel) • 1: FALSE时检测(LowLevel) <ul style="list-style-type: none"> * 指定了边缘检测的情况下, 为超出参数范围(轴)(错误代码: 1A60H)(详细代码: 8003H)。 <p>■补偿时间(CompensationTime) 忽略输入值。</p> <p>■筛选时间(FilterTime) 设置范围为“0.0~5.0[s]”。</p> <ul style="list-style-type: none"> * 指定了超出范围的值的情况下, 为超出各轴信号的筛选时间设置范围警告(事件代码: 00D24H), 且筛选时间将以“0.0”执行动作。

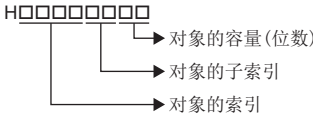
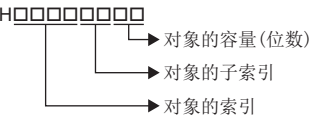
变量名	名称	获取	数据类型	属性	内容
Homing_Required	原点复位要否设置	就绪ON	BOOL	LIST_WRITE _BACK	设置原点复位的要否。 不需要原点复位的情况下，将设置为FALSE。 设置为FALSE时，当前位置恢复时原点复位请求不变为TRUE。 • FALSE：不需要原点复位 • TRUE：需要原点复位
JerkLimit	Jerk限制值	就绪ON	LREAL	LIST_WRITE _BACK	设置Jerk限制值。 • 0.0000、0.0001~2147483647.0[U/s ³] * “0.0000”的情况下，不进行Jerk限制。 * 设置了小于0.0001的正数的情况下，将其作为“0.0000”获取。
OverrunOperation	越程时动作设置	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_WRITE _BACK	设置定位动作中超出了目标位置时的动作。 • 1：立即停止(ImmediateStop) • 2：继续执行当前的加减速速度(KeepCurrentAcc)
StartableAtUnhomed	原点复位未完时启动允许	就绪ON	BOOL	LIST_WRITE _BACK	设置原点复位未完时是否允许轴的启动。 • FALSE：不允许 • TRUE：允许
StopMode_Deceleration	停止时减速度	就绪ON	LREAL	LIST_WRITE _BACK	设置由于发生停止原因而减速停止时的减速度/减速时间。 ■指定加速度/减速度的加减速方式(加减速速度指定方式)的情况下 • 0.0000、0.0001~2147483647.0[U/s ²] ■指定加减速时间的加减速方式(加减速时间恒定方式)的情况下 • 0.000000、0.000001~8400.0[s] * 与启动时加减速速度0指定时动作选择(AccelerationZeroBehavior)无关，“0.0”的情况下将立即停止。 * 发生停止原因时上限值超出范围的情况下，将被固定为上限值，下限值超出范围的情况下，将作为“0(立即停止)”处理。
StopMode_DecelerationCurve	减速停止时停止处理选择	就绪ON	INT (MC_STOP_CURVE_MODE) ☞ 1458页 MC_STOP_CURVE_MODE[M]	LIST_WRITE _BACK	设置减速中(包含停止原因、自动减速)发生了停止原因时的动作。 • 1：重新创建减速曲线(OverrideCurve)
StopMode_General	发生停止原因时停止选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_WRITE _BACK	设置发生了停止原因时的动作。 • 1：立即停止(ImmediateStop) • 2：继续执行当前的加减速速度(KeepCurrentAcc) • 3：替代加减速速度(AlternativeAcc) * 执行无减速度指定的FB时，将立即停止。
StopMode_HwStrokeLimit	发生硬件行程限位错误时停止选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_WRITE _BACK	设置发生了硬件行程限位的错误时的动作。 • 1：立即停止(ImmediateStop) • 2：继续执行当前的加减速速度(KeepCurrentAcc) • 3：替代加减速速度(AlternativeAcc) * 执行无减速度指定的FB时，将立即停止。
StopMode_ServoOff	运行中伺服OFF指令时处理选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_WRITE _BACK	设置运行中变为了伺服OFF时的动作。 • 0：忽略(Ignore) • 4：立即停止后伺服OFF(ServoOffAfterImmediateStop) • 5：减速停止后伺服OFF(ServoOffAfterDecelStop)
StopMode_SwStrokeLimit	发生软件行程限位错误时停止选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_WRITE _BACK	设置发生了软件行程限位的错误时的动作。 • 1：立即停止(ImmediateStop) • 2：继续执行当前的加减速速度(KeepCurrentAcc) • 3：替代加减速速度(AlternativeAcc) * 执行无减速度指定的FB时，将立即停止。
StopOption_DriverTargetIgnored	驱动器指令删除检测设置	就绪ON	BOOL	LIST_WRITE _BACK	设置是否在轴动作中检测出驱动器模块的指令删除状态并且因错误而停止。 • FALSE：检测无效 • TRUE：检测有效

变量名	名称	获取	数据类型	属性	内容
StopSignal	停止信号	就绪ON	SIGNAL_SELECT 1431页 SIGNAL_SELECT (信号选择)	LIST_WRITE _BACK	<p>设置使用外部输入信号的停止信号(STOP)的信号。根据作为外部信号使用的变量及控制器管理的设备的状态,有可能作为信号检测处理。</p> <p>停止信号(StopSignal)固有的设置及动作如下所示。</p> <p>■输入输出编号(StartIO) 忽略输入值。</p> <p>■对象(Target) 无指定时,将判断为信号无效且置为始终信号非检测状态。数据类型仅可指定[VAR]、[DEV]、[CONST]。</p> <p>* 设置了不能使用的值的情况下,为超出参数范围(轴)(错误代码:1A60H)。可通过下述详细代码确认原因。</p> <ul style="list-style-type: none"> • 设置了不正确的字符串(详细代码:0001H) • 设置了不能使用的数据类型(详细代码:0002H) • 指定了不能使用的类型(详细代码:0003H) • 标签指定时指定了数值型以外的类型(详细代码:0004H) • 设置了不存在的型号(详细代码:0005H) • 设置了外部设备访问无效的全局标签(详细代码:0005H) • 超出了字符数的上限(详细代码:0007H) • 在参照的设备站的网络同步通信设置中设置了“Do not Synchronize(不同步)”(详细代码:8002H) • 设置了不支持的软件(详细代码:8002H) • 设置了超出范围的软件编号(详细代码:8002H) <p>■信号检测方法(Detection) 只允许指定下述的等级检测。</p> <ul style="list-style-type: none"> · 0: TRUE时检测(HighLevel) · 1: FALSE时检测(LowLevel) <p>* 指定了边缘检测的情况下,为超出参数范围(轴)(错误代码:1A60H)(详细代码:8003H)。</p> <p>■补偿时间(CompensationTime) 忽略输入值。</p> <p>■筛选时间(FilterTime) 设置范围为“0.0~5.0[s]”。</p> <p>* 指定了超出范围的值的情况下,为超出各轴信号的筛选时间设置范围警告(事件代码:00D24H),且筛选时间将以“0.0”执行动作。</p>
SwStrokeLimit_Lower	软件行程限位下限值	就绪ON	LREAL	LIST_WRITE _BACK	<p>设置软件行程限位下限值。</p> <ul style="list-style-type: none"> • -10000000000.0~10000000000.0
SwStrokeLimit_Target	软件行程限位对象	就绪ON	INT (MC_POS_SOURCE) 1459页 MC_POS_SOURCE[M]	LIST_WRITE _BACK	<p>设置软件行程限位的对象。</p> <ul style="list-style-type: none"> • -1: 无效(Invalid) • 1: 指令当前位置(SetPosition) • 3: 进给机械位置(FeedMachinePosition)
SwStrokeLimit_Upper	软件行程限位上限值	就绪ON	LREAL	LIST_WRITE _BACK	<p>设置软件行程限位上限值。</p> <ul style="list-style-type: none"> • -10000000000.0~10000000000.0
Unit_Position	位置指令单位	就绪ON	DWORD (HEX)	LIST_WRITE _BACK	<p>设置运动控制中使用的位置指令单位。</p> <ul style="list-style-type: none"> • 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: 任意单位
Unit_PositionString	位置指令单位字符串	就绪ON	WSTRING (31)	LIST_WRITE _BACK	<p>以字符串设置运动控制中使用的指令单位。位置指令单位(AxisName.Pr.Unit_Position)为“00FF0000H: 任意单位”的情况下设置此项。</p>
Unit_Velocity	速度指令单位	就绪ON	DWORD (HEX)	LIST_WRITE _BACK	<p>设置运动控制中使用的速度指令单位。</p> <ul style="list-style-type: none"> • 00000300H: U/s • 00004700H: U/min • FD000300H: U/ms • FA000300H: U/μs • F7000300H: U/ns
VelocityBias	速度偏置值	就绪ON	LREAL	LIST_WRITE _BACK	<p>设置各轴的速度偏置值。</p> <ul style="list-style-type: none"> • 0.0000、0.0001~2500000000.0 <p>“0.0000”的情况下,速度偏置将变为无效。</p>

变量名	名称	获取	数据类型	属性	内容
VelocityLimit_Negative	负方向速度限制值	就绪ON	LREAL	LIST_WRITE_BACK	设置地址减少方向速度限制值。 • 0.0001~2500000000.0
VelocityLimit_OverOperation	速度限制值溢出时动作设置	就绪ON	INT (MC_VELOCITY_LIMIT_MODE) ☞ 1457页 MC_VELOCITY_LIMIT_MODE[M]	LIST_WRITE_BACK	设置控制中的速度限制值溢出时的动作。 • 0: 忽略(Ignore) • 3: 立即停止(ImmediateStop) * 设置了“0: 忽略(Ignore)”的情况下, 检测出超出速度限制值的指令速度时, 将变为速度限制值溢出警告(事件代码: 00D03H), 并原样不变地输出超出速度限制值的指令。 * 设置了“3: 立即停止(ImmediateStop)”的情况下, 检测出超出速度限制值的指令速度时, 将变为控制中速度限制值溢出(错误代码: 1AA4H), 且执行中的定位控制将停止。
VelocityLimit_Positive	正方向速度限制值	就绪ON	LREAL	LIST_WRITE_BACK	设置地址增加方向速度限制值。 • 0.0001~2500000000.0

AxisName.Md. (轴监视数据)

变量名	名称	获取	数据类型	属性	内容
AccelerationLimit	加速度限制值	就绪ON	LREAL	LIST_READ_ONLY	存储加速度限制值。
AccelerationOverride	加速度倍率修调系数	运算周期	LREAL	LIST_READ_ONLY	存储加速度倍率修调系数。
AccelerationZeroBehavior	启动时加减速速度0指定时动作选择	就绪ON	INT (MC_ACC_ZERO_MODE) ☞ 1459页 MC_ACC_ZERO_MODE[M]	LIST_READ_ONLY	显示在加速度、减速度或加减速时间中设置了“0.0”时的动作。 • -1: 错误(不启动)(ACCError) • 1: 最大加减速(MaximumAcceleration)
ActualPosition	反馈位置	运算周期	LREAL	LIST_READ_ONLY	存储实际的当前位置。 将为“反馈位置 = 当前位置 × 驱动器单位转换分母 / 驱动器单位转换分子”的值。 存储以环形计数器范围化整后的值。
ActualVelocity	反馈速度	运算周期	LREAL	LIST_READ_ONLY	存储反馈速度。 是根据反馈位置的差异计算出的速度。 向正方向(地址增加方向)移动的情况下将为正的值, 向负方向(地址减少方向)移动的情况下将为负的值。
Analyzing	分析中	运算周期	BOOL	LIST_READ_ONLY	显示定位分析中的状态。 • FALSE: 不处于分析中 • TRUE: 分析中
AutoDeceleration	自动减速中	运算周期	BOOL	LIST_READ_ONLY	显示自动减速处理的状态。 进行了自动减速处理的期间, 将存储TRUE。 进行了多重启动的情况下, 最终定位点执行中进行了自动减速处理的期间, 将变为TRUE。 进行了控制更改的情况下, 将变为FALSE。 轴组运行中, 将FALSE存储至配置轴的自动减速中(AxisName.Md.AutoDeceleration)。 • FALSE: 不处于自动减速中 • TRUE: 自动减速中
AxisName	轴名称	系统启动时	WSTRING(127)	LIST_READ_ONLY	显示轴名称。
AxisStatus	轴状态	运算周期	INT (MC_AXIS_STATUS) ☞ 1456页 MC_AXIS_STATUS[M]	LIST_READ_ONLY	显示轴的状态。 • -1: 轴变量未初始化/轴参数异常(Invalid) • 0: 轴无效(Disabled) • 1: 错误停止中(ErrorStop) • 2: 减速停止中(Stopping) • 3: 原点复位中(Homing) • 4: 待机中(Standstill) • 5: 定位运行中(DiscreteMotion) • 6: 连续动作运行中(ContinuousMotion) • 7: 同步运行中(SynchronizedMotion)
BufferingFBs	缓冲FB数	启动时/停止时	INT	LIST_READ_ONLY	显示缓冲FB的数(0~2)。
CmdInPos	指令到位	运算周期	BOOL	LIST_READ_ONLY	至目标位置的剩余距离小于或等于指令到位宽度(AxisName.Md.CmdInPos_Width)时变为ON的信号。 • TRUE: 小于或等于指令到位宽度 • FALSE: 大于指令到位宽度

变量名	名称	获取	数据类型	属性	内容
CmdInPos_Width	指令到位宽度	就绪ON	LREAL	LIST_READ_ONLY	显示指令到位宽度。 变为获取了指令到位宽度(AxisName.Pr.CmdInPos_Width)的设定值的值。 指令到位宽度(AxisName.Md.CmdInPos_Width)为“0.0”的情况下,此轴的指令到位宽度为无效。
CommandedAcceleration	指定加速度	运算周期	LREAL	LIST_READ_ONLY	存储用户设置的加速度。(单位: U/s ²) 加减速方式为“1: 加减速时间恒定方式(mcFixedTime)”的情况下,将存储指定的加减速时间。(单位: s) 轴组运行中,将“0.0”存储至配置轴的指定加速度(AxisName.Md.CommandedAcceleration)中。
CommandedDeceleration	指定减速度	运算周期	LREAL	LIST_READ_ONLY	存储用户设置的减速度。(单位: U/s ²)加减速方式为“1: 加减速时间恒定方式(mcFixedTime)”的情况下,将为不正确的值。轴组运行中,将“0.0”存储至配置轴的指定减速度(AxisName.Md.CommandedDeceleration)中。
CommandedJerk	指定Jerk	运算周期	LREAL	LIST_READ_ONLY	存储用户设置的Jerk。(单位: U/s ³)轴组运行中,将“0.0”存储至配置轴的指定Jerk(AxisName.Md.CommandedJerk)中。
CommandedPosition	指定位置	运算周期	LREAL	LIST_READ_ONLY	存储控制中的定位系统的运动控制FB中设置的目标位置(Position)、移动量(Distance)。 定位完成时将存储“0.0”。
CommandedVelocity	指定速度	运算周期	LREAL	LIST_READ_ONLY	存储控制中的运行系统的运动控制FB中获取的指定速度。 控制完成时将存储“0.0”。 轴组运行中,将“0.0”存储至配置轴的指定速度(AxisName.Md.CommandedVelocity)中。
Cst_SetTorque	cst/ct时指令当前转矩	运算周期	LREAL	LIST_READ_ONLY	存储循环转矩模式(cst)/挡块控制模式(ct)时的指令当前转矩。
Cst_TargetTorque	cst/ct时的目标转矩	运算周期	LREAL	LIST_READ_ONLY	存储循环转矩模式(cst)/挡块控制模式(ct)时的目标转矩。
CumulativePosition	累计当前位置	运算周期	LREAL	LIST_READ_ONLY	存储当前位置的地址。 进行当前位置更改时,地址将被更改为当前位置更改值。 将为“-1000000000.0 ≤ 累计当前位置 < 1000000000.0”的环形地址。
DecelerationLimit	减速度限制值	就绪ON	LREAL	LIST_READ_ONLY	存储减速度限制值。
Drive_RPDO[1..64]	RPDO(运动系统→设备)映射	设备连接时	DWORD (HEX)	LIST_READ_ONLY	显示在网络参数的RPDO映射设置中,第n个指定的对象索引。  <ul style="list-style-type: none">• 对于对象的容量,以位数显示1字节、2字节、4字节、8字节。• 第n个的对象容量为1字节的情况下,第n+1个中将输入填充对象。
Drive_TPDO[1..64]	TPDO(设备→运动系统)映射	设备连接时	DWORD (HEX)	LIST_READ_ONLY	显示在网络参数的TPDO映射设置中,第n个指定的对象索引。  <ul style="list-style-type: none">• 对于对象的容量,以位数显示1字节、2字节、4字节、8字节。• 第n个的对象容量为1字节的情况下,第n+1个中将输入填充对象。
Drive_UnitConvRatioNumerator	驱动器单位转换分子	就绪ON	DWORD (UDINT)	LIST_READ_ONLY	存储用于转换为驱动器模块的指令单位的分子。
Drive_UnitConvRatioDenominator	驱动器单位转换分母	就绪ON	DWORD (UDINT)	LIST_READ_ONLY	存储用于转换为驱动器模块的指令单位的分母。
Driver_Mode	驱动器控制模式	运算周期	INT (MC_DRIVE_MODE) 1457页 MC_DRIVE_MODE [M]	LIST_READ_ONLY	显示驱动器模块的控制模式的状态。 <ul style="list-style-type: none">• 6: 原点复位模式(hm)• 8: 循环位置模式(csp)• 9: 循环速度模式(csv)• 10: 循环转矩模式(cst)• -104: 挡块控制模式(ct)

变量名	名称	获取	数据类型	属性	内容
Driver_ReadyOn	驱动器就绪ON状态	运算周期	BOOL	LIST_READ_ONLY	显示连接的驱动器模块的状态。 驱动器状态(AxisName.Md.Driver_State)为以下时, 变为TRUE。 • 2: Fault Reaction Active • 5: Switched On • 6: Operation Enable • 7: Quick Stop Active
Driver_ServoOn	驱动器伺服ON状态	运算周期	BOOL	LIST_READ_ONLY	显示连接的驱动器模块的状态。 驱动器状态(AxisName.Md.Driver_State)为以下时, 变为TRUE。 • 6: Operation Enable • 7: Quick Stop Active
Driver_State	驱动器状态	运算周期	INT (MC_DRIVE_STATE) 1458页 MC_DRIVE_STATE[M]	LIST_READ_ONLY	显示驱动器模块的状态。 • 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	驱动器模块错误检测	即时	BOOL	LIST_READ_ONLY	显示驱动器模块错误的状态。 • FALSE: 无驱动器模块错误 • TRUE: 有驱动器模块错误
DriverErrorID	驱动器模块错误代码	即时	WORD(HEX)	LIST_READ_ONLY	发生驱动器模块错误时, 显示对象数据的“Current alarm(2A41H)”的高位16位。
DriverErrorDetailID	驱动器模块错误详细代码	即时	WORD(HEX)	LIST_READ_ONLY	发生驱动器模块错误时, 显示对象数据的“Current alarm(2A41H)”的低位16位。
Encoder_Connected	连接状态	运算周期	BOOL	LIST_READ_ONLY	显示虚拟编码器轴的连接状态。 • FALSE: 未连接 • TRUE: 连接中
Encoder_CounterDisable	计数器无效中	运算周期	BOOL	LIST_READ_ONLY	显示从编码器的输入状态。 • FALSE: 计数器启用 • TRUE: 计数器禁用
Error	轴错误检测	即时	BOOL	LIST_READ_ONLY	显示轴错误的状态。 • FALSE: 无轴错误 • TRUE: 有轴错误
ErrorID	轴错误代码	即时	WORD(HEX)	LIST_READ_ONLY	显示发生的轴错误的错误代码。
FeedMachinePosition	进给机械位置	运算周期	LREAL	LIST_READ_ONLY	存储根据机械坐标的当前位置的地址。 将为从原点复位完成位置开始的累计值。 即使进行当前位置更改, 地址也不被更改。 将为“-10000000000.0 ≤ 进给机械位置 < 10000000000.0”的环形地址。
FollowupDisable	跟踪无效中	运算周期	BOOL	LIST_READ_ONLY	显示跟踪的禁用状态。 禁用的情况下, 不进行跟踪。 • FALSE: 跟踪有效 • TRUE: 跟踪无效
ForcedStop_Released	紧急停止解除中	运算周期	BOOL	LIST_READ_ONLY	显示紧急停止的解除状态。 • FALSE: 紧急停止 • TRUE: 紧急停止解除
ForcedStop_Signal	紧急停止信号	就绪ON	SIGNAL_SELECT 1431页 SIGNAL_SELECT(信号选择)	LIST_READ_ONLY	显示紧急停止信号的输入状态。 紧急停止信号(ForcedStop_Signal)固有的输出如下所示。 ■输入输出编号(StartIO) 始终显示“0”。 ■对象(Target) 显示参数的获取结果。 ■信号检测方法(Detection) 显示参数的获取结果。 ■补偿时间(CompensationTime) 始终显示“0.0”。 ■筛选时间(FilterTime) 显示参数的获取结果。

变量名	名称	获取	数据类型	属性	内容
Homing_Complete	原点复位完成	即时	BOOL	LIST_READ_ONLY	显示原点复位是否正常完成。 运行开始时, 或需要原点复位时, 将变为FALSE。 • FALSE: 原点复位未完 • TRUE: 原点复位完成
Homing_Request	原点复位请求	即时	BOOL	LIST_READ_ONLY	显示是否需要原点复位。 原点复位完成时将变为FALSE。 • FALSE: 无原点复位请求 • TRUE: 有原点复位请求
Homing_Required	原点复位要否设置	就绪ON	BOOL	LIST_READ_ONLY	显示是否需要轴的原点复位。 FALSE的情况下, 当前位置恢复时原点复位请求不变为TRUE。 • FALSE: 不需要原点复位 • TRUE: 需要原点复位
Homing_Status	原点复位动作状态	运算周期	WORD (HEX)	LIST_READ_ONLY	显示驱动器模块的原点复位状态。 • 0000H: 原点复位启动中 • 0001H: 原点复位启动待机状态或原点复位中断 • 0002H: 原点复位完成(未到达目标位置) • 0003H: 原点复位正常完成 • 0004H: 发生原点复位错误(速度为0以外) • 0005H: 发生原点复位错误(速度为0) • FFFFH: 原点复位中以外时
HwStrokeLimit_FlsStatus	上限限位信号状态	运算周期	BOOL	LIST_READ_ONLY	显示上限限位信号(FLS)的检测状态。 • FALSE: 上限限位信号(FLS)检测中 • TRUE: 上限限位信号(FLS)未检测 * 不是信号的输入状态, 而是显示反映了信号检测方法、筛选时间的检测状态。
HwStrokeLimit_Override	硬件行程限位倍率修调	启动时	STRING (15)	LIST_READ_ONLY	显示硬件行程限位检查的有效/无效切换指令的受理状态。 • DISABLE: 检查禁用 • ONLY_INSIDE: 仅至范围内方向检查无效 • 上述以外: 无禁用请求
HwStrokeLimit_RlsStatus	下限限位信号状态	运算周期	BOOL	LIST_READ_ONLY	显示下限限位信号(RLS)的检测状态。 • FALSE: 下限限位信号(RLS)检测中 • TRUE: 下限限位信号(RLS)未检测 * 不是信号的输入状态, 而是显示反映了信号检测方法、筛选时间的检测状态。
InVelocity	目标速度到达	运算周期	BOOL	LIST_READ_ONLY	显示指令当前速度是否到达目标速度。 轴组运行中, 配置轴的目标速度到达(AxisName, Md. InVelocity)中将显示FALSE。 • FALSE: 未到达 • TRUE: 到达
Io_PosActualValue	对象数据_PosActualValue	运算周期	DINT	LIST_READ_ONLY	显示对象数据Position actual value(6064H)的值。(从设备接收的位置反馈) 实际编码器轴、虚拟编码器轴的情况下, 将显示从编码器获取的输入脉冲的值[编码器脉冲单位]。
Io_PosEncoderResolution	对象数据_PosEncoderResolution	运算周期	DWORD (UDINT)	LIST_READ_ONLY	显示对象数据Position encoder resolution(608FH)的值。 实际编码器轴、虚拟编码器轴的情况下, 将显示同步编码器的分辨率。
Io_Statusword	对象数据_Statusword	运算周期	WORD (HEX)	LIST_READ_ONLY	显示对象数据Statusword(6041H)的值。 实际编码器轴的情况下, 在运动系统侧模拟并显示编码器的状态。
Io_TargetPos	对象数据_TargetPos	运算周期	DINT	LIST_READ_ONLY	显示对象数据Target position(607AH)的值。(发送至设备站的位置指令) 虚拟驱动轴的情况下, 将显示将累计当前位置转换为驱动器单位的指令值后的值。
Io_TargetVelocity	对象数据_TargetVelocity	运算周期	DINT	LIST_READ_ONLY	显示对象数据Target Velocity(60FFH)的值。(发送至设备站的速度指令)
Io_TorqueActualValue	对象数据_TorqueActualValue	运算周期	INT	LIST_READ_ONLY	显示对象数据Torque actual value(6077H)的值。(从设备站接收的反馈转矩)
Io_VelActualValue	对象数据_VelActualValue	运算周期	DINT	LIST_READ_ONLY	显示对象数据Velocity actual value(606CH)的值。(从设备站接收的速度反馈)
JerkLimit	Jerk限制值	就绪ON	LREAL	LIST_READ_ONLY	存储Jerk限制值。

变量名	名称	获取	数据类型	属性	内容
OperationCycle	控制周期	系统启动时	INT	LIST_READ_ONLY	存储轴的控制运算周期。 • 0: 轴未初始化/控制周期设置范围外 • 1: 以第1运算周期运行 • 2: 以第2运算周期运行 • 3: 以第3运算周期运行
OverrunOperation	越程时动作设置	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_READ_ONLY	显示越程动作设置的输入状态。 • 1: 立即停止(ImmediateStop) • 2: 继续执行当前的加减速速度(KeepCurrentAcc)
PosRestoration_Status	当前位置恢复状态	服务	INT (MC_POS_RESTORATION_STATUS) ☞ 1459页 MC_POS_RESTORATION_STATUS[M]	LIST_READ_ONLY	显示当前位置恢复状态。 • 0: 未实施(NotExecute) • 1: 恢复请求等待(WaitingRequest) • 2: 增量系统中恢复完成(RestoredInIncSystem) • 3: 绝对位置系统中恢复完成(原点复位未完)(RestoredInAbsSystemUnHomed) • 4: 绝对位置系统中恢复完成(RestoredInAbsSystem) * 设备断开时将变为“0: 未实施(NotExecute)”。
ProfileID	执行配置文件ID编号	运算周期	WORD(UINT)	LIST_READ_ONLY	存储当前正在执行的配置文件ID。
SetAcceleration	指令当前加速度	运算周期	LREAL	LIST_READ_ONLY	存储指令输出加速度(根据指令当前速度的差异计算出的加减速速度)。*1(单位: U/s ²) 根据移动方向及指令当前加速度(AxisName, Md.SetAcceleration)的值, 加减速的状态有所不同。 ■移动方向为正方向(地址增加方向) • 0.0: 停止中或定速中 • 正值: 加速中 • 负值: 减速中 ■移动方向为负方向(地址减少方向) • 0.0: 停止中或定速中 • 正值: 减速中 • 负值: 加速中 到达目标速度时存储“0.0”。 轴组动作中, 存储配置轴的指令当前加速度。
SetPosition	指令当前位置	运算周期	LREAL	LIST_READ_ONLY	存储当前位置的地址。*1 指令当前位置为将累计当前位置以环形计数器范围化整后的值。
SetVelocity	指令当前速度	运算周期	LREAL	LIST_READ_ONLY	存储指令输出速度。*1 是根据指令当前位置的差异计算出的速度。 向正方向(地址增加方向)移动的情况下将为正值, 向负方向(地址减少方向)移动的情况下将为负值。 轴组动作中, 存储配置轴的指令当前速度。
SlaveEmulate_Enable	轴模拟中	服务	BOOL	LIST_READ_ONLY	显示驱动器模拟运行是有效还是无效。 • FALSE: 无效 • TRUE: 有效
StartableAtUnhomed	原点复位未完时启动允许	就绪ON	BOOL	LIST_READ_ONLY	显示原点复位未完时的启动状态。 • FALSE: 不允许 • TRUE: 允许
StopMode_Deceleration	停止时减速度	就绪ON	LREAL	LIST_READ_ONLY	显示停止时减速度的输入状态。
StopMode_DecelerationCurve	减速停止时停止处理选择	就绪ON	INT (MC_STOP_CURVE_MODE) ☞ 1458页 MC_STOP_CURVE_MODE[M]	LIST_READ_ONLY	显示减速停止时停止处理选择的输入状态。 • 1: 重新创建减速曲线(OverrideCurve)
StopMode_General	发生停止原因时停止选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_READ_ONLY	显示发生停止原因时停止选择的输入状态。 • 1: 立即停止(ImmediateStop) • 2: 继续执行当前的加减速速度(KeepCurrentAcc) • 3: 替代加减速速度(AlternativeAcc)
StopMode_HwStrokeLimit	发生硬件行程限位错误时停止选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_READ_ONLY	显示发生硬件行程限位错误时停止选择的输入状态。 • 1: 立即停止(ImmediateStop) • 2: 继续执行当前的加减速速度(KeepCurrentAcc) • 3: 替代加减速速度(AlternativeAcc)
StopMode_ServoOff	运行中伺服OFF指令时处理选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_READ_ONLY	显示运行中伺服OFF指令时处理选择的输入状态。 • 0: 忽略(Ignore) • 4: 立即停止后伺服OFF(ServoOffAfterImmediateStop) • 5: 减速停止后伺服OFF(ServoOffAfterDecelStop)

变量名	名称	获取	数据类型	属性	内容
StopMode_SwStrokeLimit	发生软件行程限位错误时停止选择	就绪ON	INT (MC_DECEL_STOP_MODE) 1458页 MC_DECEL_STOP_MODE[M]	LIST_READ_ONLY	显示发生软件行程限位错误时停止选择的输入状态。 • 1: 立即停止(ImmediateStop) • 2: 继续执行当前的加减速度(KeepCurrentAcc) • 3: 替代加减速度(AlternativeAcc)
StopOption_DriverTargetIgnored	驱动器指令删除检测设置	就绪ON	BOOL	LIST_READ_ONLY	显示驱动器指令删除检测设置的状态。 • FALSE: 检测无效 • TRUE: 检测有效
StopSignal	停止信号	就绪ON	SIGNAL_SELECT 1431页 SIGNAL_SELECT(信号选择)	LIST_READ_ONLY	显示外部输入信号的停止信号(STOP)的输入状态。 停止信号(StopSignal)固有的输出如下所示。 ■输入输出编号(StartIO) 始终显示“0”。 ■对象(Target) 显示参数的获取结果。 ■信号检测方法(Detection) 显示参数的获取结果。 ■补偿时间(CompensationTime) 始终显示“0.0”。 ■筛选时间(FilterTime) 显示参数的获取结果。
StopStatus	停止状态	运算周期	BOOL	LIST_READ_ONLY	显示外部输入信号的停止信号(STOP)的输入状态。 • FALSE: 停止处理解除 • TRUE: 停止处理中
SwStrokeLimit_Lower	软件行程限位下限值	就绪ON	LREAL	LIST_READ_ONLY	显示软件行程限位下限值。
SwStrokeLimit_Override	软件行程限位倍率修调	启动时	STRING(15)	LIST_READ_ONLY	显示软件行程限位有效/无效状态。 • DISABLE: 检查无效 • ONLY_INSIDE: 仅至范围内方向检查无效 • 上述以外: 检查有效(无禁用请求)
SwStrokeLimit_Target	软件行程限位对象	就绪ON	INT (MC_POS_SOURCE) 1459页 MC_POS_SOURCE[M]	LIST_READ_ONLY	显示软件行程限位的对象。 • -1: 无效(Invalid) • 1: 指令当前位置(SetPosition) • 3: 进给机械位置(FeedMachinePosition)
SwStrokeLimit_Upper	软件行程限位上限值	就绪ON	LREAL	LIST_READ_ONLY	显示软件行程限位上限值。
TargetAcceleration	目标加速度	运算周期	LREAL	LIST_READ_ONLY	存储加速中最大的加速度。(单位: U/s ²) 轴组运行中, 将“0.0”存储至配置轴的目标加速度(AxisName.Md.TargetAcceleration)中。
TargetDeceleration	目标减速度	运算周期	LREAL	LIST_READ_ONLY	存储减速中最大的减速度。(单位: U/s ²) 轴组运行中, 将“0.0”存储至配置轴的目标减速度(AxisName.Md.TargetDeceleration)中。
TargetVelocity	目标速度	运算周期	LREAL	LIST_READ_ONLY	存储考虑了倍率修调、速度限制值的实际的目标速度。 控制完成时将存储“0.0”。 轴组运行中, 将“0.0”存储至配置轴的目标速度(AxisName.Md.TargetVelocity)中。 ■定位控制的情况下 与移动方向无关, 将为大于“0.0”的值。 ■定位控制以外的情况下 向正方向(地址增加方向)移动的情况下将为正的值, 向负方向(地址减少方向)移动的情况下将为负的值。
TorqueLimit_Negative	负方向转矩限制值	运算周期	LREAL	LIST_READ_ONLY	显示当前有效的负方向转矩限制值。
TorqueLimit_Positive	正方向转矩限制值	运算周期	LREAL	LIST_READ_ONLY	显示当前有效的正方向转矩限制值。
Unit_Position	位置指令单位	就绪ON	DWORD(HEX)	LIST_READ_ONLY	显示运动控制中使用的位置指令单位。 • 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: 任意单位

变量名	名称	获取	数据类型	属性	内容
Unit_PositionDisplay	位置指令单位显示	就绪ON	WSTRING(31)	LIST_READ_ONLY	以字符串显示当前控制中的位置指令单位。
Unit_PositionString	位置指令单位字符串	就绪ON	WSTRING(31)	LIST_READ_ONLY	以位置指令单位 (<u>AxisName</u> , Pr. Unit_Position) 设置了“00FF0000H: 任意单位”的情况下, 以字符串显示设置的位置指令单位字符串 (<u>AxisName</u> , Pr. Unit_PositionString) 的控制中的位置指令单位。
Unit_Velocity	速度指令单位	就绪ON	DWORD(HEX)	LIST_READ_ONLY	显示运动控制中使用的速度指令单位。 <ul style="list-style-type: none"> • 00000300H: U/s • 00004700H: U/min • FD000300H: U/ms • FA000300H: U/μs • F7000300H: U/ns
Unit_VelocityDisplay	速度指令单位显示	就绪ON	WSTRING(31)	LIST_READ_ONLY	以字符串输出当前控制中的速度指令单位。
UseInGroup	轴组使用中	运算周期	BOOL	LIST_READ_ONLY	显示指定的轴在轴组中是正在使用还是未使用。 <ul style="list-style-type: none"> • FALSE: 未使用 • TRUE: 使用中
VelocityBias	速度偏置值	就绪ON	LREAL	LIST_READ_ONLY	存储就绪ON时获取的速度偏置值 (<u>AxisName</u> , Pr. VelocityBias) 的设定值。
VelocityLimit_Negative	负方向速度限制值	就绪ON	LREAL	LIST_READ_ONLY	存储地址减少方向速度限制值。
VelocityLimit_OverOperation	速度限制值溢出时动作设置	就绪ON	INT (MC_VELOCITY_LIMIT_MODE) ☞ 1457页 MC_VELOCITY_LIMIT_MODE[M]	LIST_READ_ONLY	显示控制中的速度限制值溢出时的动作。 <ul style="list-style-type: none"> • 0: 忽略(Ignore) • 3: 立即停止(ImmediateStop)
VelocityLimit_Positive	正方向速度限制值	就绪ON	LREAL	LIST_READ_ONLY	存储地址增加方向速度限制值。
VelocityOverride	速度倍率修调系数	运算周期	LREAL	LIST_READ_ONLY	显示速度倍率修调系数。
Warning	轴警告检测	即时	BOOL	LIST_READ_ONLY	显示轴警告的状态。 <ul style="list-style-type: none"> • FALSE: 无轴警告 • TRUE: 有轴警告
WarningID	轴警告代码	即时	WORD(HEX)	LIST_READ_ONLY	存储发生了轴警告时的事件代码。

*1 由于发生浮点误差, 因此存储的值包含有误差。

AxisName.Cd. (轴控制数据)

变量名	名称	获取	类型	属性	范围
AccelerationOverride	加速度倍率修调系数	运算周期	LREAL	LIST_READ_ONLY	设置加速度倍率修调系数。 • 0.01~10.00
Encoder_Connect	连接指令	运算周期	BOOL	LIST_READ_ONLY	切换虚拟编码器轴输入的连接状态的有效/无效。 • FALSE: 无效 • TRUE: 有效
Encoder_CounterDisable	计数器无效中	运算周期	BOOL	LIST_READ_ONLY	切换为计数器禁用。 • FALSE: 无执行 • TRUE: 执行计数器禁用请求
Encoder_InputValue	编码器输入值	运算周期	DINT	LIST_READ_ONLY	逐次设置作为虚拟编码器轴的输入值使用的值。
ErrorReset	轴错误复位	服务	BOOL	LIST_READ_ONLY	对轴的错误、警告、驱动器模块错误进行复位。 错误复位完成时将变为FALSE。 • FALSE: 无执行 • TRUE: 执行错误复位
FollowupDisable	跟踪无效中	运算周期	BOOL	LIST_READ_ONLY	禁用跟踪。 • FALSE: 无执行 • TRUE: 执行跟踪禁用请求
Homing_ClearRequest	原点复位请求清除	运算周期	BOOL	LIST_READ_ONLY	清除原点复位请求。 将原点复位请求(Homing_Request)置为了FALSE后, 原点复位请求清除(Homing_ClearRequest)将自动变为FALSE。 • FALSE: 无执行 • TRUE: 执行原点复位请求清除
HwStrokeLimit_Override	硬件行程限位倍率修调	启动时	STRING(15)	LIST_READ_ONLY	暂时切换硬件行程限位检查的有效/无效。 • DISABLE: 检查禁用 • ONLY_INSIDE: 仅至范围内方向检查无效 • 上述以外: 无禁用请求
SwStrokeLimit_Override	软件行程限位倍率修调	启动时	STRING(15)	LIST_READ_ONLY	暂时切换软件行程限位的检查有效/无效。 • DISABLE: 检查无效 • ONLY_INSIDE: 仅至范围内方向检查无效 • 上述以外: 检查有效(无禁用请求)
TorqueLimit_Negative	负方向转矩限制值	运算周期	LREAL	LIST_READ_ONLY	设置负方向转矩限制值。 • 0.0~1000.0[%]
TorqueLimit_Positive	正方向转矩限制值	运算周期	LREAL	LIST_READ_ONLY	设置正方向转矩限制值。 • 0.0~1000.0[%]
VelocityOverride	速度倍率修调系数	运算周期	LREAL	LIST_READ_ONLY	设置速度倍率修调系数。 • 0.00~10.00

轴变量中可对各轴类型设置的变量

对于轴变量中使用的变量，可对各轴类型设置的变量有所不同。

各轴类型中可设置的变量如下所示。

■AxisName, AxisRef. (轴信息)

○：可设置，×：不可设置

变量名	名称	轴类型				
		实际驱动轴	实际编码器轴	虚拟驱动轴	虚拟编码器轴	虚拟连接轴
AxisNo	轴No.	○	○	○	○	○
StartIO	输入输出No.	○	○	○	○	○

■AxisName, PrConst. (轴参数常数)

○：可设置，×：不可设置

变量名	名称	轴类型				
		实际驱动轴	实际编码器轴	虚拟驱动轴	虚拟编码器轴	虚拟连接轴
AddressOfStation	站地址设置	○	○	×	×	×
AxisType	轴类型设置	○	○	○	○	○
Encoder_AxisType	实际编码器轴类型设置	×	○	×	×	×
Encoder_CounterDisableSignal	计数器禁用信号	×	○	×	○	×
Encoder_RingCout_LowerValue	编码器环形计数器下限值	×	×	×	○	×
Encoder_RingCout_UpperValue	编码器环形计数器上限值	×	×	×	○	×
HwStrokeLimit_FlsSignal	上限限位信号	○	×	○	×	×
HwStrokeLimit_RlsSignal	下限限位信号	○	×	○	×	×
OperationCycle	控制周期设置	○	○	○	○	○
PosRestoration_AbsPosEnable	绝对位置管理设置	○	○	○	○	○
RingCount_Enable	环形计数器有效选择	○	○	○	○	○
RingCount_LowerValue	环形计数器下限值	○	○	○	○	○
RingCount_UpperValue	环形计数器上限值	○	○	○	○	○
SlaveEmulate_Enable	轴模拟有效	○	×	×	×	×
SlaveObject	对象数据	×	×	×	○	×
TorqueLimit_Max	转矩限制最大值	○	×	×	×	×
TorqueLimit_NegativeInitial	负方向转矩限制初始值	○	×	×	×	×
TorqueLimit_PositiveInitial	正方向转矩限制初始值	○	×	×	×	×

■AxisName.Pr. (轴参数)

○：可设置，×：不可设置

变量名	名称	轴类型				
		实际驱动轴	实际编码器轴	虚拟驱动轴	虚拟编码器轴	虚拟连接轴
AccelerationLimit	加速度限制值	○	×	○	×	×
AccelerationZeroBehavior	启动时加减速速度0指定时动作选择	○	×	○	×	×
CmdInPos_Width	指令到位宽度	○	×	○	×	×
DecelerationLimit	减速度限制值	○	×	○	×	×
Drive_UnitConvRatioNum	驱动器单位转换分子	○	○	×	○	×
Drive_UnitConvRatioDen	驱动器单位转换分母	○	○	×	○	×
ForcedStop_Signal	紧急停止信号	○	×	○	×	×
Homing_Required	原点复位要否设置	○	○	○	○	○
JerkLimit	Jerk限制值	○	×	○	×	×
OverrunOperation	越程时动作设置	○	×	○	×	×
StartableAtUnhomed	原点复位未完时启动允许	○	○	○	○	○
StopMode_Deceleration	停止时减速度	○	×	○	×	×
StopMode_DecelerationCurve	减速停止时停止处理选择	○	×	○	×	×
StopMode_General	发生停止原因时停止选择	○	×	○	×	×
StopMode_HwStrokeLimit	发生硬件行程限位错误时停止选择	○	×	○	×	×
StopMode_ServoOff	运行中伺服OFF指令时处理选择	○	×	○	×	×
StopMode_SwStrokeLimit	发生软件行程限位错误时停止选择	○	×	○	×	×
StopOption_DriverTargetIgnored	驱动器指令删除检测设置	○	×	×	×	×
StopSignal	停止信号	○	×	○	×	×
SwStrokeLimit_Lower	软件行程限位下限值	○	×	○	×	×
SwStrokeLimit_Target	软件行程限位对象	○	×	○	×	×
SwStrokeLimit_Upper	软件行程限位上限值	○	×	○	×	×
Unit_Position	位置指令单位	○	○	○	○	○
Unit_PositionString	位置指令单位字符串	○	○	○	○	○
Unit_Velocity	速度指令单位	○	○	○	○	○
VelocityBias	速度偏置值	○	×	○	×	○
VelocityLimit_Negative	负方向速度限制值	○	×	○	×	×
VelocityLimit_OverOperation	速度限制值溢出时动作设置	○	×	○	×	×
VelocityLimit_Positive	正方向速度限制值	○	×	○	×	×

■AxisName. Md. (轴监视数据)

○：可设置，×：不可设置

变量名	名称	轴类型				
		实际驱动轴	实际编码器轴	虚拟驱动轴	虚拟编码器轴	虚拟连接轴
AccelerationLimit	加速度限制值	○	×	○	×	×
AccelerationOverride	加速度倍率修调系数	○	×	○	×	×
AccelerationZeroBehavior	启动时加减速速度0指定时动作选择	○	×	○	×	×
ActualPosition	反馈位置	○	○	×	×	×
ActualVelocity	反馈速度	○	○	×	×	×
Analyzing	分析中	○	○	○	○	○
AutoDeceleration	自动减速中	○	×	○	×	×
AxisName	轴名称	○	○	○	○	○
AxisStatus	轴状态	○	○	○	○	○
BufferingFBs	缓冲FB数	○	×	○	×	×
CmdInPos	指令到位	○	×	○	×	×
CmdInPos_Width	指令到位宽度	○	×	○	×	×
CommandedAcceleration	指定加速度	○	×	○	×	×
CommandedDeceleration	指定减速度	○	×	○	×	×
CommandedJerk	指定Jerk	○	×	○	×	×
CommandedPosition	指定位置	○	×	○	×	×
CommandedVelocity	指定速度	○	×	○	×	×
Cst_SetTorque	cst/ct时指令当前转矩	○	×	×	×	×
Cst_TargetTorque	cst/ct时的目标转矩	○	×	×	×	×
CumulativePosition	累计当前位置	○	○	○	○	○
DecelerationLimit	减速度限制值	○	×	○	×	×
Drive_RPDO[1..64]	RPDO(运动系统→设备)映射	○	×	×	×	×
Drive_TPDO[1..64]	TPDO(设备→运动系统)映射	○	×	×	×	×
Drive_UnitConvRatioNum	驱动器单位转换分子	○	○	×	×	×
Drive_UnitConvRatioDen	驱动器单位转换分母	○	○	×	×	×
Driver_Mode	驱动器控制模式	○	×	×	×	×
Driver_ReadyOn	驱动器就绪ON状态	○	×	×	×	×
Driver_ServoOn	驱动器伺服ON状态	○	×	×	×	×
Driver_State	驱动器状态	○	×	×	×	×
DriverError	驱动器模块错误检测	○	×	×	×	×
DriverErrorID	驱动器模块错误代码	○	×	×	×	×
DriverErrorDetailID	驱动器模块错误详细代码	○	×	×	×	×
Encoder_Connected	连接状态	×	×	×	○	×
Encoder_CounterDisable	计数器无效中	×	○	×	○	×
Error	轴错误检测	○	○	○	○	○
ErrorID	轴错误代码	○	○	○	○	○
FeedMachinePosition	进给机械位置	○	×	×	×	×
FollowupDisable	跟踪无效中	○	×	×	×	×
ForcedStop_Released	紧急停止解除中	○	×	○	×	×
ForcedStop_Signal	紧急停止信号	○	×	○	×	×
Homing_Complete	原点复位完成	○	○	○	○	○
Homing_Request	原点复位请求	○	○	○	○	○
Homing_Required	原点复位要否设置	○	○	○	○	○
Homing_Status	原点复位动作状态	○	×	×	×	×
HwStrokeLimit_FlsStatus	上限限位信号状态	○	×	○	×	×
HwStrokeLimit_Override	硬件行程限位倍率修调	○	×	○	×	×
HwStrokeLimit_RlsStatus	下限限位信号状态	○	×	○	×	×
InVelocity	目标速度到达	○	×	○	×	×
Io_PosActualValue	对象数据_PosActualValue	○	○	×	○	×

变量名	名称	轴类型				
		实际驱动轴	实际编码器轴	虚拟驱动轴	虚拟编码器轴	虚拟连接轴
Io_PosEncoderResolution	对象数据 _PosEncoderResolution	×	○	×	○	×
Io_Statusword	对象数据_Statusword	○	○	×	×	×
Io_TargetPos	对象数据_TargetPos	○	×	○	×	×
Io_TargetVelocity	对象数据_TargetVelocity	○	×	×	×	×
Io_TorqueActualValue	对象数据_TorqueActualValue	○	×	×	×	×
Io_VelActualValue	对象数据_VelActualValue	○	×	×	×	×
JerkLimit	Jerk限制值	○	×	○	×	×
OperationCycle	控制周期	○	○	○	○	○
OverrunOperation	越程时动作设置	○	×	○	×	×
PosRestoration_Status	当前位置恢复状态	○	○	○	○	○
ProfileID	执行配置文件ID编号	○	×	○	×	○
SetAcceleration	指令当前加速度	○	×	○	×	×
SetPosition	指令当前位置	○	○	○	○	○
SetVelocity	指令当前速度	○	×	○	○	×
SlaveEmulate_Enable	轴模拟中	○	×	×	×	×
StartableAtUnhomed	原点复位未完时启动允许	○	○	○	○	○
StopMode_Deceleration	停止时减速度	○	×	○	×	×
StopMode_DecelerationCurve	减速停止时停止处理选择	○	×	○	×	×
StopMode_General	发生停止原因时停止选择	○	×	○	×	×
StopMode_HwStrokeLimit	发生硬件行程限位错误时停止选择	○	×	○	×	×
StopMode_ServoOff	运行中伺服OFF指令时处理选择	○	×	○	×	×
StopMode_SwStrokeLimit	发生软件行程限位错误时停止选择	○	×	○	×	×
StopOption_DriverTargetIgnored	驱动器指令删除检测设置	○	×	×	×	×
StopSignal	停止信号	○	×	○	×	×
StopStatus	停止状态	○	×	○	×	×
SwStrokeLimit_Lower	软件行程限位下限值	○	×	○	×	×
SwStrokeLimit_Override	软件行程限位倍率修调	○	×	○	×	×
SwStrokeLimit_Target	软件行程限位对象	○	×	○	×	×
SwStrokeLimit_Upper	软件行程限位上限值	○	×	○	×	×
TargetAcceleration	目标加速度	○	×	○	×	×
TargetDeceleration	目标减速度	○	×	○	×	×
TargetVelocity	目标速度	○	×	○	×	×
TorqueLimit_Negative	负方向转矩限制值	○	×	×	×	×
TorqueLimit_Positive	正方向转矩限制值	○	×	×	×	×
Unit_Position	位置指令单位	○	○	○	○	○
Unit_PositionDisplay	位置指令单位显示	○	○	○	○	○
Unit_PositionString	位置指令单位字符串	○	○	○	○	○
Unit_Velocity	速度指令单位	○	○	○	○	○
Unit_VelocityDisplay	速度指令单位显示	○	○	○	○	○
UseInGroup	轴组使用中	○	○	○	○	○
VelocityBias	速度偏置值	○	×	○	×	○
VelocityLimit_Negative	负方向速度限制值	○	×	○	×	×
VelocityLimit_OverOperation	速度限制值溢出时动作设置	○	×	○	×	×
VelocityLimit_Positive	正方向速度限制值	○	×	○	×	×
VelocityOverride	速度倍率修调系数	○	×	○	×	×
Warning	轴警告检测	○	○	○	○	○
WarningID	轴警告代码	○	○	○	○	○

■AxisName.Cd. (轴控制数据)

○：可设置，×：不可设置

变量名	名称	轴类型				
		实际驱动轴	实际编码器轴	虚拟驱动轴	虚拟编码器轴	虚拟连接轴
AccelerationOverride	加速度倍率修调系数	○	×	○	×	×
Encoder_Connect	连接指令	×	×	×	○	×
Encoder_CounterDisable	计数器无效中	×	○	×	○	×
Encoder_InputValue	编码器输入值	×	×	×	○	×
ErrorReset	轴错误复位	○	○	○	○	○
FollowupDisable	跟踪无效中	○	×	×	×	×
Homing_ClearRequest	原点复位请求清除	○	○	○	○	○
HwStrokeLimit_Override	硬件行程限位倍率修调	○	×	○	×	×
SwStrokeLimit_Override	软件行程限位倍率修调	○	×	○	×	×
TorqueLimit_Negative	负方向转矩限制值	○	×	×	×	×
TorqueLimit_Positive	正方向转矩限制值	○	×	×	×	×
VelocityOverride	速度倍率修调系数	○	×	○	×	×

轴组变量

轴组变量的数据类型以AXES_GROUP型表示。

各轴组的数据类型具有下述构件。

构件名	数据类型	内容	参照目标
AxesGroupRef	AXES_GROUP_REF	是运动控制用功能块的输入/输出用的数据结构。	1408页 AxesGroupName.AxesGroupRef. (轴组信息)
PrConst	AXES_GROUP_PRM_CONST	存储轴组的参数数据(常数)。 轴组变量初始化时展开设定值。 轴组变量初始化后不实施至控制的重新获取。	1408页 AxesGroupName.PrConst. (轴组 参数常数)
Pr	AXES_GROUP_PRM	存储轴组的参数数据。 轴组变量创建时展开初始值。 轴组变量初始化后也实施至控制的重新获取。 根据参数至控制的获取时机有所不同。	1409页 AxesGroupName.Pr. (轴 组参数)
Md	AXES_GROUP_MONI	存储轴组的监视数据。 以各监视数据中确定的周期实施刷新。	1411页 AxesGroupName.Md. (轴 组监视数据)
Cd	AXES_GROUP_CMD	存储轴组控制用指令数据。 各控制运算周期中获取最新的值用于控制。	1413页 AxesGroupName.Cd. (轴 组控制数据)

[AxesGroupName.AxesGroupRef.](#) (轴组信息)

变量名	名称	获取	数据类型	属性	内容
GroupNo	轴组No.	—	WORD(UINT)	LIST_WRITE_ BACK	设置轴组No.。 • 0: 未设置 • 1~10000: 设置轴组No.
StartIO	输入输出No.	—	WORD(HEX)	LIST_WRITE_ BACK	设置输入输出No.。

[AxesGroupName.PrConst.](#) (轴组参数常数)

变量名	名称	获取	数据类型	属性	内容
AxesGroupType	轴组类型设置	系统启动时	INT (MC_AXES_GROUP_TYPE)  1459页 MC_AXES_GROUP_TYPE	LIST_WRITE_ BACK	设置轴组类型。 • 0: 标准(Standard)

AxesGroupName.Pr. (轴组参数)

变量名	名称	获取	数据类型	属性	内容
AccelerationLimit	加速度限制值	就绪ON	LREAL	LIST_WRITE_ BACK	<p>设置加速度限制值。</p> <ul style="list-style-type: none"> • 0.0000、0.0001~2147483647.0 <p>* 加减速方式为“1: 加减速时间恒定方式 (mcFixedTime)”的情况下, 将对根据指定的加减速时间计算出的加速度进行限制。</p> <p>* “0.0000”的情况下, 不进行加速度的限制。</p> <p>* 设置了小于0.0001的正数的情况下, 将其作为“0.0000”获取。</p>
AccelerationZeroBehavior	启动时加减速速度0指定时动作选择	就绪ON	INT (MC_ACC_ZERO_MODE) ☞ 1459页 MC_ACC_ZERO_MODE[M]	LIST_WRITE_ BACK	<p>设置启动时, 在加速度、减速度或加减速时间中设置了“0.0”时的动作。</p> <ul style="list-style-type: none"> • -1: 错误(不启动) (ACError) • 1: 最大加减速 (MaximumAcceleration)
Axis[1..16]	配置轴	就绪ON	AXIS_REF ☞ 1387页 AxisName.AxisRef. (轴信息)	LIST_WRITE_ BACK	<p>设置配置轴组的轴信息(AxisName.AxisRef)的轴No. (AxisNo)。</p>
CmdInPos_Width	指令到位宽度	就绪ON	LREAL	LIST_WRITE_ BACK	<p>设置轴组的合成轴上的指令到位宽度。</p> <ul style="list-style-type: none"> • 0.0: 功能无效 • 0.000000001~10000000000.0: 功能有效 <p>* 设置了小于0.000000001的正数的情况下, 将其作为“0.0”获取。</p>
DecelerationLimit	减速度限制值	就绪ON	LREAL	LIST_WRITE_ BACK	<p>设置减速度限制值。</p> <ul style="list-style-type: none"> • 0.0000、0.0001~2147483647.0 <p>* 加减速方式为“1: 加减速时间恒定方式 (mcFixedTime)”的情况下, 将对根据指定的加减速时间计算出的减速度进行限制。</p> <p>* “0.0000”的情况下, 不进行减速度的限制。</p> <p>* 设置了小于0.0001的正数的情况下, 将其作为“0.0000”获取。</p>
JerkLimit	Jerk限制值	就绪ON	LREAL	LIST_WRITE_ BACK	<p>设置Jerk限制值。</p> <ul style="list-style-type: none"> • 0.0000、0.0001~2147483647.0 <p>* “0.0000”的情况下, 不进行Jerk限制。</p> <p>* 设置了小于0.0001的正数的情况下, 将其作为“0.0000”获取。</p>
OverrunOperation	越程时动作设置	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_WRITE_ BACK	<p>设置动作中超出了目标位置时的动作。</p> <ul style="list-style-type: none"> • 1: 立即停止 (ImmediateStop)
StopMode_Deceleration	停止时减速度	就绪ON	LREAL	LIST_WRITE_ BACK	<p>设置由于发生停止原因而减速停止时的减速度/减速时间。</p> <p>■指定加速度/减速度的加减速方式(加减速速度指定方式)的情况下</p> <ul style="list-style-type: none"> • 0.0000、0.0001~2147483647.0 <p>■指定加减速时间的加减速方式(加减速时间恒定方式)的情况下</p> <ul style="list-style-type: none"> • 0.000000、0.000001~8400.0 <p>* 与启动时加减速速度0指定时动作选择 (AccelerationZeroBehavior)无关, “0.0”的情况下将立即停止。</p> <p>* 发生停止原因时上限值超出范围的情况下, 将被固定为上限值, 下限值超出范围的情况下, 将作为“0(立即停止)”处理。</p>
StopMode_DecelerationCurve	减速停止时停止处理选择	就绪ON	INT (MC_STOP_CURVE_MODE) ☞ 1458页 MC_STOP_CURVE_MODE[M]	LIST_WRITE_ BACK	<p>设置减速中(包含停止原因、自动减速)发生了停止原因时的动作。</p> <ul style="list-style-type: none"> • 1: 重新创建减速曲线(OverrideCurve)
StopMode_ErrorInGroup	发生轴停止原因时配置轴动作选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_WRITE_ BACK	<p>设置以轴组运行中, 配置轴的驱动器模块中变为伺服OFF, 且发生了立即停止的轴错误的情况下, 未发生轴错误的轴的动作。</p> <ul style="list-style-type: none"> • 1: 立即停止 (ImmediateStop)
StopMode_General	发生停止原因时停止选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_WRITE_ BACK	<p>设置发生了轴组的停止原因时的动作。</p> <ul style="list-style-type: none"> • 1: 立即停止 (ImmediateStop) • 2: 继续执行当前的加减速度 (KeepCurrentAcc) • 3: 替代加减速度 (AlternativeAcc) <p>* 执行无减速度指定的FB时, 将立即停止。</p>

变量名	名称	获取	数据类型	属性	内容
Unit_Position	位置指令单位	轴组有效时	DWORD (HEX)	LIST_WRITE_BACK	设置运动控制中使用的位置指令单位。 <ul style="list-style-type: none"> • 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: 任意单位
Unit_PositionString	位置指令单位字符串	轴组有效时	WSTRING (31)	LIST_WRITE_BACK	以字符串设置运动控制中使用的指令单位。 位置指令单位(Unit_Position)为“00FF0000H: 任意单位”的情况下设置此项。
Unit_Velocity	速度指令单位	轴组有效时	DWORD (HEX)	LIST_WRITE_BACK	设置运动控制中使用的速度指令单位。 <ul style="list-style-type: none"> • 00000300H: U/s • 00004700H: U/min • FD000300H: U/ms • FA000300H: U/μs • F7000300H: U/ns
VelocityBias	速度偏置值	就绪ON	LREAL	LIST_WRITE_BACK	设置各轴组的速度偏置值。 “0.0000”的情况下，速度偏置将变为无效。 <ul style="list-style-type: none"> • 0.0000、0.0001~2500000000.0
VelocityLimit	速度限制值	就绪ON	LREAL	LIST_WRITE_BACK	设置轴组的速度限制值。 <ul style="list-style-type: none"> • 0.0001~2500000000.0

AxesGroupName.Md. (轴组监视数据)

变量名	名称	获取	数据类型	属性	内容
AccelerationLimit	加速度限制值	就绪ON	LREAL	LIST_READ_ONLY	存储加速度限制值。
AccelerationOverride	加速度倍率修调系数	运算周期	LREAL	LIST_READ_ONLY	存储加速度倍率修调系数。
AccelerationZeroBehavior	启动时加减速速度0指定时动作选择	就绪ON	INT (MC_ACC_ZERO_MODE) ☞ 1459页 MC_ACC_ZERO_MODE[M]	LIST_READ_ONLY	显示在加速度、减速度或加减速时间中设置了“0.0”时的动作。 • -1: 错误(不启动)(ACCError) • 1: 最大加减速(MaximumAcceleration)
ActualVelocity	反馈速度	运算周期 (GroupEnable中)	LREAL	LIST_READ_ONLY	存储反馈速度。 是配置轴的反馈速度的合成速度。与配置轴的移动方向无关, 将为大于“0.0”的值。 轴组无效的情况下, 将存储“0.0”。
Analyzing	分析中	运算周期	BOOL	LIST_READ_ONLY	显示轴组的定位分析中的状态。 • FALSE: 不处于分析中 • TRUE: 分析中
AutoDeceleration	自动减速中	运算周期 (GroupEnable中)	BOOL	LIST_READ_ONLY	显示自动减速处理的状态。 进行了自动减速处理的期间, 将存储TRUE。 进行了多重启动的情况下, 最终定位点执行中进行了自动减速处理的期间, 将变为TRUE。 进行了控制更改的情况下, 将变为FALSE。 • FALSE: 不处于自动减速中 • TRUE: 自动减速中
Axis[1..16]	配置轴	即时	AXIS_REF ☞ 1387页 AxisName, AxisRef. (轴信息)	LIST_READ_ONLY	存储配置轴组的轴信息(AxisName, AxisRef)的轴No. (AxisNo)。
BufferingFBs	缓冲FB数	运算周期 (GroupEnable中)	INT	LIST_READ_ONLY	显示缓冲FB的数(0~2)。
CmdInPos	指令到位	运算周期	BOOL	LIST_READ_ONLY	显示至目标位置的合成轴上的剩余距离是否小于或等于指令到位宽度(AxesGroupName.Pr.CmdInPos_Width)。 • FALSE: 大于指令到位宽度 • TRUE: 小于或等于指令到位宽度
CmdInPos_Width	指令到位宽度	就绪ON	LREAL	LIST_READ_ONLY	显示轴组的合成轴上的指令到位宽度。 变为获取了指令到位宽度(AxesGroupName.Pr.CmdInPos_Width)的设定值的值。 指令到位宽度(AxesGroupName.Md.CmdInPos_Width)为“0.0”的情况下, 此轴组的指令到位功能为无效。
CommandedAcceleration	指定加速度	运算周期 (GroupEnable中)	LREAL	LIST_READ_ONLY	存储用户设置的加速度。(单位: U/s ²)加减速方式为“1: 加减速时间恒定方式(mcFixedTime)”的情况下, 将存储指定的加减速时间。(单位: s)
CommandedDeceleration	指定减速度	运算周期 (GroupEnable中)	LREAL	LIST_READ_ONLY	存储用户设置的减速度。(单位: U/s ²)加减速方式为“1: 加减速时间恒定方式(mcFixedTime)”的情况下, 将为不正确的值。
CommandedJerk	指定Jerk	运算周期 (GroupEnable中)	LREAL	LIST_READ_ONLY	存储用户设置的Jerk。(单位: U/s ³)
CommandedVelocity	指定速度	运算周期 (GroupEnable中)	LREAL	LIST_READ_ONLY	存储控制中的运行系统的运动控制FB中获取的指定速度。 控制完成时将存储“0.0”。 * 直线插补控制中, 在速度模式(VelocityMode)中设置了基准轴速度(ReferenceAxisSpeed)、长轴速度(LongAxisSpeed)的情况下, 将存储相应轴的指定速度。
DecelerationLimit	减速度限制值	就绪ON	LREAL	LIST_READ_ONLY	存储减速度限制值。
Error	轴组错误检测	即时	BOOL	LIST_READ_ONLY	显示轴组错误发生的状态。 • FALSE: 无轴组错误 • TRUE: 有轴组错误
ErrorID	轴组错误代码	即时	WORD (HEX)	LIST_READ_ONLY	显示发生的轴组错误的错误代码。

变量名	名称	获取	数据类型	属性	内容
GroupName	轴组名称	即时	WSTRING(127)	LIST_READ_ONLY	存储轴组的名称。
GroupStatus	轴组状态	运算周期	INT (MC_AXES_GROUP_STATUS) ☞ 1456页 MC_AXES_GROUP_STATUS[M]	LIST_READ_ONLY	显示轴组的状态。 • -1: 轴组变量未初始化/轴组参数异常(Invalid) • 0: 轴组无效(GroupDisabled) • 1: 错误停止中(GroupErrorStop) • 2: 减速停止中(GroupStopping) • 4: 待机中(GroupStandby) • 5: 动作中(GroupMoving)
InterpolationAxes	插补轴	运算周期 (GroupEnable中)	DWORD(HEX)	LIST_READ_ONLY	以位显示轴组中插补控制执行中的配置轴。
InVelocity	目标速度到达	运算周期 (GroupEnable中)	BOOL	LIST_READ_ONLY	显示指令当前速度是否到达目标速度。 • FALSE: 未到达 • TRUE: 到达
JerkLimit	Jerk限制值	就绪ON	LREAL	LIST_READ_ONLY	存储Jerk限制值。
NumberOfAxes	配置轴数	即时	INT	LIST_READ_ONLY	显示轴组的配置轴数。
OverrunOperation	越程时动作设置	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_READ_ONLY	显示越程动作设置的输入状态。 • 1: 立即停止(ImmediateStop)
SetAcceleration	指令当前加速度	运算周期 (GroupEnable中)	LREAL	LIST_READ_ONLY	存储指令输出加速度(根据指令当前速度的差异计算出的加减速度)。(单位: U/s ²) 向正方向(地址增加方向)的加速度发生变化的情况下将为正的值, 向负方向(地址减少方向)的加速度发生变化的情况下将为负的值。 由于发生浮点误差, 因此存储的值包含有误差。
SetVelocity	指令当前速度	运算周期 (GroupEnable中)	LREAL	LIST_READ_ONLY	轴组动作中, 存储指令输出速度。 与配置轴的移动方向无关, 将为大于“0.0”的值。 控制完成时将存储“0.0”。 * 直线插补控制中, 在速度模式(VelocityMode)中设置了基准轴速度(ReferenceAxisSpeed)、长轴速度(LongAxisSpeed)的情况下, 将存储相应轴的指令当前速度的绝对值。
StopMode_Deceleration	停止时减速度	就绪ON	LREAL	LIST_READ_ONLY	显示停止时减速度的输入状态。
StopMode_DecelerationCurve	减速停止时停止处理选择	就绪ON	INT (MC_STOP_CURVE_MODE) ☞ 1458页 MC_STOP_CURVE_MODE[M]	LIST_READ_ONLY	显示减速停止时停止处理选择的输入状态。 • 1: 重新创建减速曲线(OverrideCurve)
StopMode_ErrorInGroup	发生轴停止原因时配置轴动作选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_READ_ONLY	显示发生轴停止原因时配置轴动作选择的输入状态。 • 1: 立即停止(ImmediateStop)
StopMode_General	发生停止原因时停止选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_READ_ONLY	显示发生停止原因时停止选择的输入状态。 • 1: 立即停止(ImmediateStop) • 2: 继续执行当前的加加速度(KeepCurrentAcc) • 3: 替代加加速度(AlternativeAcc)
TargetAcceleration	目标加速度	运算周期 (GroupEnable中)	LREAL	LIST_READ_ONLY	存储加速中最大的加速度。(单位: U/s ²)
TargetDeceleration	目标减速度	运算周期 (GroupEnable中)	LREAL	LIST_READ_ONLY	存储减速中最大的减速度。(单位: U/s ²)
TargetVelocity	目标速度	运算周期 (GroupEnable中)	LREAL	LIST_READ_ONLY	存储考虑了倍率修调、速度限制值的实际的目标速度。 与配置轴的移动方向无关, 将为大于“0.0”的值。 控制完成时将存储“0.0”。 * 直线插补控制中, 在速度模式(VelocityMode)中设置了基准轴速度(ReferenceAxisSpeed)、长轴速度(LongAxisSpeed)的情况下, 将存储相应轴的目标速度。

变量名	名称	获取	数据类型	属性	内容
Unit_Position	位置指令单位	轴组有效时	DWORD (HEX)	LIST_READ_ONLY	显示运动控制中使用的位置指令单位。 <ul style="list-style-type: none"> • 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: 任意单位
Unit_PositionDisplay	位置指令单位显示	轴组有效时	WSTRING (31)	LIST_READ_ONLY	以字符串输出当前控制中的位置指令单位。
Unit_PositionString	位置指令单位字符串	轴组有效时	WSTRING (31)	LIST_READ_ONLY	以位置指令单位 (<u>AxisGroupName.Pr.Unit_Position</u>) 设置了“00FF0000H: 任意单位”的情况下, 以字符串显示设置的位置指令单位字符串 (<u>AxisGroupName.Pr.Unit_PositionString</u>) 的控制中的位置指令单位。
Unit_Velocity	速度指令单位	轴组有效时	DWORD (HEX)	LIST_READ_ONLY	显示运动控制中使用速度指令单位。 <ul style="list-style-type: none"> • 00000300H: U/s • 00004700H: U/min • FD000300H: U/ms • FA000300H: U/μs • F7000300H: U/ns
Unit_VelocityDisplay	速度指令单位显示	轴组有效时	WSTRING (31)	LIST_READ_ONLY	以字符串输出当前控制中的速度指令单位。
VelocityBias	速度偏置值	就绪ON	LREAL	LIST_READ_ONLY	存储就绪ON时获取的速度偏置值 (<u>AxisGroupName.Pr.VelocityBias</u>) 的设定值。
VelocityLimit	速度限制值	就绪ON	LREAL	LIST_READ_ONLY	存储轴组的速度限制值。
VelocityOverride	速度倍率修调系数	运算周期	LREAL	LIST_READ_ONLY	显示速度倍率修调系数。
Warning	轴组警告检测	即时	BOOL	LIST_READ_ONLY	显示轴组警告的状态。 <ul style="list-style-type: none"> • FALSE: 无轴组警告 • TRUE: 有轴组警告
WarningID	轴组警告代码	即时	WORD (HEX)	LIST_READ_ONLY	存储发生的轴组警告的事件代码。

AxisGroupName.Cd. (轴组控制数据)

变量名	名称	获取	数据类型	属性	内容
AccelerationOverride	加速度倍率修调系数	运算周期	LREAL	LIST_READ_ONLY	设置加速度倍率修调系数。 <ul style="list-style-type: none"> • 0.01~10.00
ErrorReset	轴组错误复位	服务	BOOL	LIST_READ_ONLY	对轴组的错误、警告进行复位。 <ul style="list-style-type: none"> • FALSE: 无执行 • TRUE: 执行错误复位
VelocityOverride	速度倍率修调系数	运算周期	LREAL	LIST_READ_ONLY	设置速度倍率修调系数。 <ul style="list-style-type: none"> • 0.00~10.00

系统变量

系统变量的数据类型以MT_SYSTEM类型表示。

系统的数据类型具有下述构件。

构件名	数据类型	内容	参照目标
PrConst	SYS_CONST	存储系统的参数数据(常数)。 系统变量初始化时展开设定值。 系统变量初始化后不实施至控制的重新获取。	1414页 MotionSystem.PrConst. (运动系统参数常数)
Pr	SYS_PRM	存储系统的参数数据。 系统变量初始化时展开初始值。 根据参数至控制的获取时机有所不同。	1415页 MotionSystem.Pr. (运动系统参数)
Md	SYS_MONI	存储系统的监视数据。 以各监视数据中确定的周期实施刷新。	1416页 MotionSystem.Md. (运动系统监视数据)
Cd	SYS_CMD	存储系统用指令数据。各运动服务处理控制中获取最新的值用于控制。	1417页 MotionSystem.Cd. (运动系统控制数据)

MotionSystem.PrConst. (运动系统参数常数)

变量名	名称	获取	数据类型	属性	内容
ExcludeWarning	除外报警	系统启动时	STRING(255)	LIST_WRITE_BACK	设置不检测的警告。 <ul style="list-style-type: none"> • "": 未设置 • "0x1000, 0x1001": 不检测0x1000、0x1001的警告 • "0x1000-0x1010": 不检测0x1000~0x1010的警告
OperationCycle[1]	运算周期设置	系统启动时	CYCLE_PARAM <small>☞ 1432页</small> CYCLE_PARAM	LIST_WRITE_BACK	设置第1运算周期的各设定值。 运算周期设置(OperationCycle[1])固有的设置及动作如下所示。 ■ 周期设置(Cycle) <ul style="list-style-type: none"> • 0: 与网络的基本周期同步
OperationCycle[2]	运算周期设置	系统启动时	CYCLE_PARAM <small>☞ 1432页</small> CYCLE_PARAM	LIST_WRITE_BACK	设置第2运算周期的各设定值。 运算周期设置(OperationCycle[2])固有的设置及动作如下所示。 ■ 周期设置(Cycle) <ul style="list-style-type: none"> • 0: 与网络的中速周期同步
OperationCycle[3]	运算周期设置	系统启动时	CYCLE_PARAM <small>☞ 1432页</small> CYCLE_PARAM	LIST_WRITE_BACK	设置第3运算周期的各设定值。 运算周期设置(OperationCycle[3])固有的设置及动作如下所示。 ■ 周期设置(Cycle) <ul style="list-style-type: none"> • 0: 与网络的低速周期同步
ProfileRootDrive	运算配置文件根驱动器	系统启动时	WSTRING(7)	LIST_WRITE_BACK	设置运算配置文件的保存目标驱动器。 ■ 设置范围 <ul style="list-style-type: none"> • 数据存储器(驱动器4): /rom • SD存储卡(驱动器2): /sdc
SequenceReadyInterlock	可编程控制器就绪联动选择	系统启动时	BOOL	LIST_WRITE_BACK	选择将“可编程控制器就绪(MotionSystem.Cd.SequenceReady)”置为ON的方式。 <small>☞ 1417页 MotionSystem.Cd. (运动系统控制数据)</small> <ul style="list-style-type: none"> • TRUE: 与CPU功能部联动。 • FALSE: 不与CPU功能部联动。

变量名	名称	获取	数据类型	属性	内容
ForcedStop_Signal	所有轴紧急停止信号	就绪ON	SIGNAL_SELECT ☞ 1431页 SIGNAL_SELECT (信号选择)	LIST_WRITE_BACK	<p>设置使用所有轴紧急停止的信号。根据作为外部信号使用的变量及控制器管理的设备的状态，有可能作为信号检测处理。</p> <p>所有轴紧急停止信号(ForcedStop_Signal)固有的设置及动作如下所示。</p> <p>■输入输出编号(StartIO) 忽略输入值。</p> <p>■对象(Target) 无指定时，将判断为信号无效且置为始终信号非检测状态。</p> <p>数据类型仅可指定[VAR]、[DEV]、[CONST]。</p> <p>* 指定了不能使用的数据类型的环境下，将变为超出参数范围(系统)(错误代码：1A62H)。</p> <p>指定[VAR]、[DEV]时，在参照的设备站的网络同步通信设置中设置了“Do not Synchronize(不同步)”的情况下，超出参数范围(系统)(错误代码：1A62H)。</p> <p>■信号检测方法(Detection) 只允许指定下述等级检测。指定了边缘检测的情况下，为超出参数范围(系统)(错误代码：1A62H)。</p> <ul style="list-style-type: none"> • 0: TRUE时检测(HighLevel) • 1: FALSE时检测(LowLevel) <p>■补偿时间(CompensationTime) 忽略输入值。</p> <p>■筛选时间(FilterTime) 设置范围为“0.0~5.0[s]”。</p> <p>* 指定了超出范围的值的环境下，为超出系统信号的筛选时间设置范围警告(事件代码：00F0FH)，且筛选时间将以“0.0”执行动作。</p>
StopMode_All	发生所有轴停止原因时停止选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_WRITE_BACK	<p>设置发生了所有轴停止原因的情况下，是立即停止还是减速停止。</p> <ul style="list-style-type: none"> • 1: 立即停止(ImmediateStop) • 2: 继续执行当前的加减速速度(KeepCurrentAcc) • 3: 替代加减速速度(AlternativeAcc) <p>* 执行无减速度指定的FB时，将立即停止。</p>
StopMode_AllDeceleration	所有轴停止时减速度	就绪ON	LREAL	LIST_WRITE_BACK	<p>指定发生了所有轴停止原因的情况下减速停止时的减速度。</p> <p>设置了0.0000的情况下立即停止。</p> <p>根据加减速方式设置范围有所不同。</p> <p>加减速方式可以通过动作系统FB的选项(Options)的“位0~2”进行选择，并通过位指定进行设置。</p> <p>[设置范围]</p> <ul style="list-style-type: none"> • 加减速速度指定方式: 0.0000、0.0001~2147483647.0 • 加减速时间恒定方式: 0.0000、0.000001~8400.0

MotionSystem.Md. (运动系统监视数据)

变量名	名称	获取	数据类型	属性	内容
Error	运动部系统错误检测	即时	BOOL	LIST_READ _ONLY	显示错误的状态。 • FALSE: 无错误 • TRUE: 有错误
ErrorID	运动部最新系统错误代码	即时	WORD(HEX)	LIST_READ _ONLY	显示最新的错误代码。
ForcedStop_Released	紧急停止解除中	运算周期	BOOL	LIST_READ _ONLY	显示紧急停止解除状态。 • FALSE: 所有轴紧急停止 • TRUE: 所有轴紧急停止解除
ForcedStop_Signal	所有轴紧急停止信号	就绪ON	SIGNAL_SELECT ☞ 1431页 SIGNAL_SELECT (信号选择)	LIST_READ _ONLY	显示所有轴紧急停止信号的输入状态。 所有轴紧急停止信号(ForcedStop_Signal)固有的输出如下所示。 ■输入输出编号(StartIO) 始终显示“0”。 ■对象(Target) 显示参数的获取结果。 ■信号检测方法(Detection) 显示参数的获取结果。 ■补偿时间(CompensationTime) 始终显示“0.0”。 ■筛选时间(FilterTime) 显示参数的获取结果。
MotionService	运动服务处理监视	服务	CYCLE_MONI ☞ 1432页 CYCLE_MONI	LIST_READ _ONLY	存储运动服务处理的监视值。
OperationCycle[1..3]	运算周期监视	即时	CYCLE_MONI ☞ 1432页 CYCLE_MONI	LIST_READ _ONLY	存储第1运算周期与第2运算周期、第3运算周期的监视值。
Ready	准备完成	服务	BOOL	LIST_READ _ONLY	显示内置运动准备完成(X420)的ON/OFF状态。 • FALSE: OFF(准备未就绪) • TRUE: ON(准备就绪)
StopMode_All	发生所有轴停止原因时停止选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 1458页 MC_DECEL_STOP_MODE[M]	LIST_READ _ONLY	显示发生所有轴停止原因时停止选择的输入状态。 • 1: 立即停止(ImmediateStop) • 2: 继续执行当前的加减速速度(KeepCurrentAcc) • 3: 替代加减速速度(AlternativeAcc)
StopMode_AllDeceleration	所有轴停止时减速度	就绪ON	LREAL	LIST_READ _ONLY	显示所有轴停止时减速度的输入状态。
Sync	同步用标志	服务	BOOL	LIST_READ _ONLY	显示同步用标志(X421)的ON/OFF状态。 • FALSE: OFF(模块访问禁止) • TRUE: ON(模块访问允许)
Warning	运动部系统警告检测	即时	BOOL	LIST_READ _ONLY	显示警告的状态。 • FALSE: 无警告 • TRUE: 有警告
WarningID	运动部最新系统警告代码	即时	WORD(HEX)	LIST_READ _ONLY	存储最新的事件代码。

MotionSystem.Cd. (运动系统控制数据)

变量名	名称	获取	数据类型	属性	内容
ErrorReset	系统错误复位	服务	BOOL	LIST_READ _ONLY	<p>对所有的错误、警告进行复位。</p> <ul style="list-style-type: none"> • FALSE: 无执行 • TRUE: 执行系统错误复位
SequenceReady	可编程控制器就绪	服务	BOOL	LIST_READ _ONLY	<p>确认CPU功能部正常，并显示通知运动功能部的信号状态。</p> <ul style="list-style-type: none"> • FALSE: 可编程控制器就绪OFF • TRUE: 可编程控制器就绪ON <p>■ “可编程控制器就绪联动选择 (MotionSystem.PrConst.SequenceReadyInterlock)” 为 TRUE的情况下</p> <p>下述条件全部成立时，本变量将变为TRUE。</p> <ul style="list-style-type: none"> • 同步用标志 (X421) 进行OFF→ON • CPU功能部进行STOP→PAUSE/RUN <p>■ “可编程控制器就绪联动选择 (MotionSystem.PrConst.SequenceReadyInterlock)” 为 FALSE的情况下</p> <p>下述条件全部成立时，本变量将变为TRUE。</p> <ul style="list-style-type: none"> • 同步用标志 (X421) 进行OFF→ON • CPU功能部进行STOP→PAUSE/RUN • 可编程控制器就绪 (Y420) 进行OFF→ON


输入轴变量

高级同步控制中使用的输入轴变量的数据类型以ADV_INPUT型表示。

各输入轴的数据类型具有下述构件。

构件名	数据类型	内容	参阅
LabelID	WORD (UINT)	设置输入轴的标签ID。	1418页 AdvInputName . (输入轴设置)
Axis	AXIS_REF	设置作为高级同步控制输入轴的轴。 多个输入轴信息中不能指定同一轴。	
PrConst	ADV_INPUT_PRM_CONST	存储作为输入轴的轴的参数数据(常数)。 输入轴变量初始化时展开设定值。 输入轴变量初始化后, 控制时不需要再次获取。	1419页 AdvInputName . PrConst. (输入轴参数常数)
Pr	ADV_INPUT_PRM	存储作为输入轴的轴的参数数据。 输入轴变量初始化时展开初始值。 输入轴变量初始化后, 控制时也需要再次获取。 根据参数至控制的获取时机有所不同。	1419页 AdvInputName . Pr. (输入轴参数)
Md	ADV_INPUT_MONI	存储作为输入轴的轴的监视数据。 以各监视数据中确定的周期实施刷新。	1419页 AdvInputName . Md. (输入轴监视数据)

AdvInputName. (输入轴设置)

变量名	名称	获取	数据类型	属性	内容
LabelID	轴标签ID	系统启动时	WORD (UINT)	LIST_WRITE _BACK	设置输入轴的标签ID。 • 1~256
Axis	轴信息	系统启动时	AXIS_REF  1387页 AxisName . AxisRef . (轴信息)	LIST_WRITE _BACK	设置作为高级同步控制的输入轴的轴信息(AxisName . AxisRef)。 可创建的输入轴最大为256个。不同输入轴的轴信息中不能设置同一轴。设置了同一轴的情况下, 将变为高级同步控制的输入轴不正确(错误代码: 1A0EH)。 请勿设置局部标签中已定义的轴的轴信息(AxisName . AxisRef)。已设置的情况下, 将不能创建输入轴。

AdvInputName.PrConst. (输入轴参数常数)

变量名	名称	获取	数据类型	属性	内容
SourceValue	数据源选择	系统启动时	INT (MC_SOURCE) ☞ 1455页 MC_SOURCE[M]	LIST_WRITE _BACK	设置高级同步控制中参照的输入轴的数据源。 • 1: 指令当前值(mcSetValue) • 2: 反馈值(mcActualValue) • 101: 最新指令当前值(mcLatestSetValue) • 102: 最新反馈值(mcLatestActualValue)
SmoothingTimeConstant	平滑时间常数	系统启动时	WORD(UINT)	LIST_WRITE _BACK	设置平滑处理输入轴的输入移动量时的平均时间。 • 0~5000[ms]
PhaseCompensationTimeConstant	相位补偿时间常数	系统启动时	WORD(UINT)	LIST_WRITE _BACK	设置将相位补偿时的相位补偿量通过一次延迟反映时的时间常数。 通过设置的时间常数相位补偿量的63%将被反映。 • 0~65535[ms]
DirectionRestriction	移动方向限制	系统启动时	INT (MC_INPUT_DIRECTION) ☞ 1461页 MC_INPUT_DIRECTION	LIST_WRITE _BACK	将输入轴中的输入移动量限制为一个方向时进行此设置。 • 0: 无移动方向限制(NoDirectionRestriction) • 1: 当前位置仅允许增加方向(mcPositiveDirection) • 2: 当前位置仅允许减少方向(mcNegativeDirection)

AdvInputName.Pr. (输入轴参数)

变量名	名称	获取	数据类型	属性	内容
PhaseCompensationAdvanceTime	相位补偿超前时间	运算周期	DINT	LIST_WRITE _BACK	在超前或延迟输入轴的相位(输入响应)时进行此设置。 • -100000000~100000000[μs]

AdvInputName.Md. (输入轴监视数据)

变量名	名称	获取	数据类型	属性	内容
ActiveStatus	有效状态	即时	INT	LIST_READ ONLY	存储输入轴的状态。 • -1: 无效 • 0: 未连接 • 1: 输入轴有效
CumulativePosition	累计当前位置	运算周期	LREAL	LIST_READ ONLY	存储将输入轴的累计当前位置设置为输入轴的轴信息(AdvInputName.Axis)的轴的位置单位。 输入轴的当前位置是平滑处理、相位补偿处理、移动方向限制处理后的值。
SetVelocity	指令当前速度	运算周期	LREAL	LIST_READ ONLY	存储将输入轴的速度设置为输入轴的轴信息(AdvInputName.Axis)的轴的速度单位。 输入轴的速度是平滑处理、相位补偿处理、移动方向限制处理后的值。
PhaseCompensationAmount	相位补偿量	运算周期	LREAL	LIST_READ ONLY	存储将输入轴的相位补偿量设置为输入轴的轴信息(AdvInputName.Axis)的轴的位置指令单位。 输入轴的相位补偿量是平滑处理、相位补偿处理后的值。
DirectionRestrictionAmount	移动方向限制量	运算周期	LREAL	LIST_READ ONLY	输入轴的移动方向限制时, 存储将允许方向与反方向的输入移动量的累计值设置为输入轴的轴的位置单位。

输出轴变量

高级同步控制中使用的输出轴变量的数据类型以ADV_OUTPUT型表示。

各输出轴的数据类型具有下述构件。

构件名	数据类型	内容	参阅
LabelID	WORD (UINT)	设置输出轴的标签ID。	1420页 AdvOutputName . (输出轴设置)
AxisRef	AXIS_REF	设置作为高级同步控制输出轴的轴。 多个输出轴信息中不能指定同一轴。	
PrConst	ADV_OUTPUT_PRM_CONST	存储作为输出轴的轴的参数数据(常数)。 输出轴变量初始化时展开设定值。 输出轴变量初始化后, 控制时不需要再次获取。	1421页 AdvOutputName . PrConst. (输出轴参数常数)
Pr	ADV_OUTPUT_PRM	存储作为输出轴的轴的参数数据。 输出轴变量初始化时展开初始值。 输出轴变量初始化后, 控制时也需要再次获取。 根据参数至控制的获取时机有所不同。	1422页 AdvOutputName . Pr. (输出轴参数)
Md	ADV_OUTPUT_MONI	存储作为输出轴的轴的监视数据。 以各监视数据中确定的周期实施刷新。	1424页 AdvOutputName . Md. (输出轴监视数据)
Cd	ADV_OUTPUT_CMD	存储输出轴控制用指令数据。 各控制运算周期中获取最新的值用于控制。	1424页 AdvOutputName . Cd. (输出轴控制数据)

AdvOutputName. (输出轴设置)

变量名	名称	获取	数据类型	属性	内容
LabelID	轴标签ID	系统启动时	WORD (UINT)	LIST_WRITE _BACK	设置输出轴的标签ID。 • 1~256
Axis	轴信息	系统启动时	AXIS_REF 1387页 AxisName . AxisRef. (轴信息)	LIST_WRITE _BACK	设置作为高级同步控制的输出轴的轴信息 (AxisName . AxisRef)。 可创建的输出轴最大为256个。不同的输出轴的轴信息中不能设置同一轴。已设置的情况下, 将变为高级同步控制输出轴不正确(错误代码: 1A0FH)。 请勿设置局部标签中已定义的轴的轴信息 (AxisName . AxisRef)。即使设置也不能创建输出轴。 输出轴中可设置的轴类型为实际驱动轴、虚拟驱动轴、虚拟连接轴。设置了不能使用的轴的情况下, 将变为高级同步控制输出轴不正确(错误代码: 1A0FH)。

AdvOutputName.PrConst. (输出轴参数常数)

变量名	名称	获取	数据类型	属性	内容
SmoothingTimeConstant	输出轴平滑时间常数	系统启动时	WORD (UINT)	LIST_WRITE _BACK	设置平滑处理输出轴的输出移动量时的平均时间[ms]。 • 0~5000[ms]
MasterClutchSmoothingTimeConstant	主轴离合器平滑时间常数	系统启动时	WORD (UINT)	LIST_WRITE _BACK	主轴离合器平滑方式 (AdvOutputName.Pr.Clutch.MasterSmoothingMethod) 设置为“1: 时间常数方式(指数)(TimeConstantExponent)”或“2: 时间常数方式(直线)(TimeConstantLinear)”时, 设置时间常数。 设置离合器ON/OFF通用的时间常数。 • 0~5000[ms]
AuxClutchSmoothingTimeConstant	辅助轴离合器平滑时间常数	系统启动时	WORD (UINT)	LIST_WRITE _BACK	辅助轴离合器平滑方式 (AdvOutputName.Pr.Clutch.AuxSmoothingMethod) 设置为“1: 时间常数方式(指数)(TimeConstantExponent)”或“2: 时间常数方式(直线)(TimeConstantLinear)”时, 设置时间常数。 设置离合器ON/OFF通用的时间常数。 • 0~5000[ms]
MasterSpeedChangeGearSmoothingTimeConstant	主轴变速器平滑时间常数	系统启动时	WORD (UINT)	LIST_WRITE _BACK	设置平滑处理主轴变速处理时的速度变化时的平均时间。 此外, 通过平滑处理, 仅设置时间的输入值传达延迟。 设定值为“0”时, 速度将直接改变。 • 0~5000[ms]
AuxSpeedChangeGearSmoothingTimeConstant	辅助轴变速器平滑时间常数	系统启动时	WORD (UINT)	LIST_WRITE _BACK	设置平滑处理辅助轴变速处理时的速度变化时的平均时间。 此外, 通过平滑处理, 仅设置时间的输入值传达延迟。 设定值为“0”时, 速度将直接改变。 • 0~5000[ms]
OutSpeedChangeGearSmoothingTimeConstant	输出轴变速器平滑时间常数	系统启动时	WORD (UINT)	LIST_WRITE _BACK	设置平滑处理输出轴变速处理时的速度变化时的平均时间。 此外, 通过平滑处理, 仅设置时间的输入值传达延迟。 设定值为“0”时, 速度将直接改变。 • 0~5000[ms]

AdvOutputName.Pr. (输出轴参数)

变量名	名称	获取	数据类型	属性	内容
SubAxis	子输入轴对象	启动时	AXIS_REF ☞ 1387页 AxisName, AxisRef. (轴信息)	LIST_WRITE _BACK	设置子输入轴的对象轴。 设置了“0”的情况下无效。 设置了输入设置中未分配的轴的情况下, 将变为高级同步控制轴未设置(错误代码: 1B22H)。
AuxAxis	辅助轴对象	启动时	AXIS_REF ☞ 1387页 AxisName, AxisRef. (轴信息)	LIST_WRITE _BACK	设置辅助轴的对象轴。 设置了“0”的情况下无效。 设置了输入设置中未分配的轴的情况下, 将变为高级同步控制轴未设置(错误代码: 1B22H)。
Gear	齿轮参数	启动时	ADV_GEAR_PARAM ☞ 1445页 ADV_GEAR_PARAM(齿轮参数)	LIST_WRITE _BACK	设置齿轮功能的参数。
MasterOnClutchSignal	主轴ON离合器信号设置	启动时	SIGNAL_SELECT ☞ 1431页 SIGNAL_SELECT(信号选择)	LIST_WRITE _BACK	<p>主轴ON离合器控制设置(AdvOutputName.Pr.Clutch.MasterOnControl)为“15: 输入输出数据指定(ClutchSignal)”的情况下, 设置离合器的外部信号。 主轴ON离合器信号设置(MasterOnClutchSignal)固有的设置及动作如下所示。</p> <p>■对象(Target) 数据类型中只能设置[OBJ]、[VAR]、[DEV]、[LINK]。 ■信号检测方法(Detection) 只允许设置下述等级检测。 • 2: FALSE→TRUE(上升沿)时检测(RisingEdge) • 3: TRUE→FALSE(下降沿)时检测(FallingEdge) • 4: 上升沿/下降沿时检测(BothEdges) ■补偿时间(CompensationTime) 设置范围为“-5.0~5.0[s]”。 ■筛选时间(FilterTime) 设置范围为“0.0”。</p>
MasterOffClutchSignal	主轴OFF离合器信号设置	启动时	SIGNAL_SELECT ☞ 1431页 SIGNAL_SELECT(信号选择)	LIST_WRITE _BACK	<p>主轴OFF离合器控制设置(AdvOutputName.Pr.Clutch.MasterOffControl)为“15: 输入输出数据指定(ClutchSignal)”的情况下, 设置离合器的外部信号。 主轴OFF离合器信号设置(MasterOffClutchSignal)固有的设置及动作如下所示。</p> <p>■对象(Target) 数据类型中只能设置[OBJ]、[VAR]、[DEV]、[LINK]。 ■信号检测方法(Detection) 只允许设置下述等级检测。 • 2: FALSE→TRUE(上升沿)时检测(RisingEdge) • 3: TRUE→FALSE(下降沿)时检测(FallingEdge) • 4: 上升沿/下降沿时检测(BothEdges) ■补偿时间(CompensationTime) 设置范围为“-5.0~5.0[s]”。 ■筛选时间(FilterTime) 设置范围为“0.0”。</p>
AuxOnClutchSignal	辅助轴ON离合器信号设置	启动时	SIGNAL_SELECT ☞ 1431页 SIGNAL_SELECT(信号选择)	LIST_WRITE _BACK	<p>辅助轴ON离合器控制设置(AdvOutputName.Pr.Clutch.AuxOnControl)为“15: 输入输出数据指定(ClutchSignal)”的情况下, 设置离合器的外部信号。 辅助轴ON离合器信号设置(AuxOnClutchSignal)固有的设置及动作如下所示。</p> <p>■对象(Target) 数据类型中只能设置[OBJ]、[VAR]、[DEV]、[LINK]。 ■信号检测方法(Detection) 只允许设置下述等级检测。 • 2: FALSE→TRUE(上升沿)时检测(RisingEdge) • 3: TRUE→FALSE(下降沿)时检测(FallingEdge) • 4: 上升沿/下降沿时检测(BothEdges) ■补偿时间(CompensationTime) 设置范围为“-5.0~5.0[s]”。 ■筛选时间(FilterTime) 设置范围为“0.0”。</p>

变量名	名称	获取	数据类型	属性	内容
AuxOffClutchSignal	辅助轴OFF离合器信号设置	启动时	SIGNAL_SELECT ☞ 1431页 SIGNAL_SELECT (信号选择)	LIST_WRITE _BACK	<p>辅助轴OFF离合器控制设置 (AdvOutputName.Pr.Clutch.AuxOffControl) 为“15: 输入输出数据指定 (ClutchSignal)”的情况下, 设置离合器的外部信号。</p> <p>辅助轴OFF离合器信号设置 (AuxOffClutchSignal) 固有的设置及动作如下所示。</p> <p>■对象 (Target) 数据类型中只能设置 [OBJ]、[VAR]、[DEV]、[LINK]。</p> <p>■信号检测方法 (Detection) 只允许设置下述等级检测。</p> <ul style="list-style-type: none"> • 2: FALSE→TRUE (上升沿) 时检测 (RisingEdge) • 3: TRUE→FALSE (下降沿) 时检测 (FallingEdge) • 4: 上升沿/下降沿时检测 (BothEdges) <p>■补偿时间 (CompensationTime) 设置范围为“-5.0~5.0[s]”。</p> <p>■筛选时间 (FilterTime) 设置范围为“0.0”。</p>
Clutch	离合器参数	—	ADV_CLUTCH_PARAM ☞ 1445页 ADV_CLUTCH_PARAM (离合器参数)	LIST_WRITE _BACK	设置离合器功能的参数。
SpeedChangeGear	变速器参数	—	ADV_SPEEDCHANGE_GEAR_PARAM ☞ 1450页 ADV_SPEEDCHANGE_GEAR_PARAM (变速器参数)	LIST_WRITE _BACK	设置变速器功能的参数。
Cam	凸轮参数	—	ADV_CAM_PARAM ☞ 1451页 ADV_CAM_PARAM (凸轮参数)	LIST_WRITE _BACK	设置凸轮功能的参数。
Restore	同步控制初始位置参数	—	ADV_RESTORE_PARAM ☞ 1453页 ADV_RESTORE_PARAM (同步控制初始位置参数)	LIST_WRITE _BACK	设置同步控制初始位置的参数。

AdvOutputName. Md. (输出轴监视数据)

变量名	名称	获取	数据类型	属性	内容
SyncStatus	同步状态	即时	INT	LIST_READ_ONLY	存储输出轴的同步状态。 • -1: 无效 • 0: 未同步 • 1: 同步中
MasterAxisNo	主轴	启动时	WORD(UINT)	LIST_READ_ONLY	存储输出轴与同步中的输入轴的轴信息 (AdvInputName.Axis) 中设置的主轴的轴编号。未连接的情况下, 将存储为“0”。
SubAxisNo	子输入轴	启动时	WORD(UINT)	LIST_READ_ONLY	存储输出轴与同步中的输入轴的轴信息 (AdvInputName.Axis) 中设置的子输入轴的轴编号。未连接的情况下, 将存储为“0”。
AuxAxisNo	辅助轴	启动时	WORD(UINT)	LIST_READ_ONLY	存储输出轴与同步中的输入轴的轴信息 (AdvInputName.Axis) 中设置的辅助轴的轴编号。未连接的情况下, 将存储为“0”。
Clutch	离合器监视	—	ADV_CLUTCH_MONI ☞ 1448页 ADV_CLUTCH_MONI (离合器监视)	LIST_READ_ONLY	存储离合器功能的监视数据。
Cam	凸轮监视	—	ADV_CAM_MONI ☞ 1452页 ADV_CAM_MONI (凸轮监视)	LIST_READ_ONLY	存储凸轮功能的监视数据。

AdvOutputName. Cd. (输出轴控制数据)

变量名	名称	获取	数据类型	属性	内容
Clutch	离合器控制数据	—	ADV_CLUTCH_CMD ☞ 1449页 ADV_CLUTCH_CMD (离合器控制数据)	LIST_READ_ONLY	设置离合器功能的控制数据。
SpeedChangeGear	变速机控制数据	—	ADV_SPEEDCHANGEGEAR_CMD ☞ 1450页 ADV_SPEEDCHANGEGEAR_CMD (变速机控制数据)	LIST_WRITE_BACK	设置变速机功能的控制数据。
Cam	凸轮控制数据	—	ADV_CAM_CMD ☞ 1453页 ADV_CAM_CMD (凸轮控制数据)	LIST_READ_ONLY	设置凸轮功能的控制数据。

系统变量(高级同步控制)

系统变量的数据类型以SYSTEM型表示。

系统的数据类型具有下述构件。

构件名	数据类型	内容	参阅
PrConst	SYS_CONST	存储系统的参数数据(常数)。 系统变量初始化时展开设定值。 系统变量初始化后不实施至控制的重新获取。	1425页 AddonSystem.PrConst.(插件系统参数常数)
Md	SYS_MONI	存储系统的监视数据。 以各监视数据中确定的周期实施刷新。	1425页 AddonSystem.Md.(插件系统监视数据)

AddonSystem.PrConst.(插件系统参数常数)

变量名	名称	获取	数据类型	属性	内容
MT_MotionControl_AdvancedSync	插件 MotionControl_AdvancedSync参数	系统启动时	ADDON_PARAM 1432页 ADDON_PARAM	LIST_WRITE_BACK	设置插件MotionControl_AdvancedSync参数中使用的存储器的最大使用量。

AddonSystem.Md.(插件系统监视数据)

变量名	名称	获取	数据类型	属性	内容
MT_MotionControl_AdvancedSync	插件 MotionControl_AdvancedSync监视	服务	ADDON_MONI 1432页 ADDON_MONI	LIST_READ_ONLY	存储插件MotionControl_AdvancedSync参数中使用的存储器的可用空间。

其他结构体的变量

MC_CAM_REF

变量名	名称	数据类型	内容
ProfileData	配置文件	PROFILE_DATA ☞ 1431页 PROFILE_DATA	设置凸轮数据的配置文件。
CamID	凸轮ID	MC_CAM_ID ☞ 1426页 MC_CAM_ID	设置凸轮数据的配置文件ID。

MC_POSITIONING_DATA_REF

变量名	名称	数据类型	内容
ProfileData	配置文件	PROFILE_DATA ☞ 1431页 PROFILE_DATA	设置定位数据的配置文件。
PositioningData	定位数据	MC_POSITIONING_DATA_ID ☞ 1426页 MC_POSITIONING_DATA_ID	设置定位数据的ID编号。

MC_CAM_ID

变量名	名称	数据类型	内容
ProfileID	配置文件ID	PROFILE_ID ☞ 1431页 PROFILE_ID	设置凸轮数据的配置文件ID。

MC_POSITIONING_DATA_ID

变量名	名称	数据类型	内容
ProfileData	配置文件ID	PROFILE_ID ☞ 1431页 PROFILE_ID	设置定位数据的运算配置文件ID。

MC_TRIGGER_REF[M]

变量名	名称	数据类型	内容
Signal	触发信号	SIGNAL_SELECT ☞ 1431页 SIGNAL_SELECT(信号选择)	设置触发信号。

MC_INPUT_REF[M]

变量名	名称	数据类型	内容
Signal	输入信号	SIGNAL_SELECT ☞ 1431页 SIGNAL_SELECT(信号选择)	设置要输入的信号。

MC_OUTPUT_REF[M]

变量名	名称	数据类型	内容
Signal	输出信号	SIGNAL_SELECT ☞ 1431页 SIGNAL_SELECT(信号选择)	设置要输出的信号。

MC_CAMSWITCH_REF

变量名	名称	数据类型	内容
ProfileData	配置文件	PROFILE_DATA ☞ 1431页 PROFILE_DATA	设置数字凸轮开关的配置文件。
CamSwitchData	凸轮开关数据	MC_CAMSWITCH_DATA_REF ☞ 1427页 MC_CAMSWITCH_DATA_REF	设置数字凸轮开关的ID编号。

MC_TRACK_REF

变量名	名称	数据类型	内容
OnCompensation	ON时间补偿	LREAL	补偿输出信号变为ON的时机。
OffCompensation	OFF时间补偿	LREAL	补偿输出信号变为OFF的时机。
Hysteresis	转换禁止距离	LREAL	设置禁止数据源转换的距离。 数据源在转换点附近增减时，防止输出信号的振荡

MC_CAMSWITCH_DATA_REF

变量名	名称	数据类型	内容
ProfileData	配置文件ID	PROFILE_ID 📄 1431页 PROFILE_ID	设置数字凸轮开关的运算配置文件ID。

MC_CAMSWITCH_SOURCE_REF

变量名	名称	数据类型	内容
Source	数据源	TARGET_REF 📄 1428页 TARGET_REF (输入信号)	指定数据源。
RingCount_LowerValue	数据源环形计数器下限值	LREAL	设置数据源的环形计数器下限值。
RingCount_UpperValue	数据源环形计数器上限值	LREAL	设置数据源的环形计数器上限值。
SpeedAverageNum	速度平均次数	WORD (UINT)	为了计算出开关的ON/OFF时机，设置是否取得到前几个运算周期为止的速度平均。

TARGET_REF (输入信号)

变量名	名称	数据类型	内容
StartIO	IO编号	WORD (HEX)	设置IO编号。
Target	对象	WSTRING (63)	以下述字符串格式设置控制中使用的信号。 * 不区分大小写。空格将被忽略。 ■ [类型] (类型) 数据名. 位位置@对象修饰 • [类型] ^{*1} • (类型) ^{*2} • 数据名 ^{*1} • . 位位置 ^{*3} • @对象修饰 ^{*4} ■ [类型] (型) WSTRING型标签 [^] • [类型] ^{*1} • (类型) ^{*2} • WSTRING型标签 ^{^*8}

*1 在“[类型]”中设置数据类型，在“数据名”中设置对象数据。根据设置的类型，数据名的设置方法有所不同。

类型	对象	数据名		设置示例
		格式	内容	
[OBJ]	设备的CANopen对象(已映射到循环数据中)	[OBJ]0x□□□□**# #	设置对象的索引、子索引、容量。 • □□□□: 索引 • **: 子索引 • ##: 容量(位数) * 应将(类型)的设置与对象的容量匹配。 * 一部分对象可作为外部信号高精度输入使用。 * 无法使用容量为“1字节”的对象。	[OBJ]0x607A0020 * MR-J5(W)-G的Target position(Obj. 607Ah: 00h) 4字节对象(20H)的情况下
[VAR]	运动系统的标签	[VAR]标签名	设置运动系统内的标签。 * 设置局部标签的情况下，在“@对象修饰”中设置POU名。	[VAR]ADunit10.OutputEnable
[AXIS]	轴数据	[AXIS]MC_SOURCE	设置MC_SOURCE列举型的列举符。 在数据的(类型)中，只能设置“LREAL”。 • 1: 指令当前值(mcSetValue) • 2: 反馈值(mcActualValue) • 101: 最新指令当前值(mcLatestSetValue) • 102: 最新反馈值(mcLatestActualValue) * 不能对本数据进行写入。	[AXIS]mcSetValue
[DEV]	运动系统内的软元件(包括本模块链接软元件)	[DEV]软元件名	设置本模块内的软元件。关于可设置的软元件，请参阅下述章节。 1430页 [DEV]类型支持的软元件一览	[DEV]X10 [DEV]U1\G11500000.1
[CONST]	常数	[CONST]常数 [CONST]0x常数	以浮点数(也可使用E格式)、10进制整数、16进制整数进行设置。 * 不能对本数据进行写入。	[CONST]1000 [CONST]0x100

*2 明确设置数据的类型的情况下记述。可进行设置的数据类型如下所示。

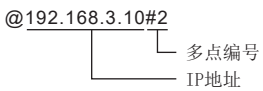
对类型中设置的数据本身的类型为BOOL类型以外的数据设置了“(BOOL)”的情况下，将作为指定的位No. 指定了0的数据处理。

类型
(BOOL)
(INT)
(DINT)
(WORD)
(DWORD)
(REAL)
(LREAL)

- *3 对于[类型]为下述的数据，附加位位置时，将作为BOOL型的数据处理。
同时设置了(类型)与位位置的情况下，将忽略(类型)的设置，且变为BOOL型的数据。

类型	位位置设置范围
[OBJ]	0~F
[VAR] (仅WORD型、DWORD型、INT型、DINT型)	
[DEV]	

- *4 设置用于特定数据的辅助信息。根据设置的[类型]，内容有所不同。
必须根据使用信号的功能进行设置。在无需对象修饰的数据类型及功能中设置了对象修饰的情况下将被忽略。

类型	对象修饰		设置示例
	格式	内容	
[OBJ]	@站地址	设置站地址(用于识别设备的IP地址)。 类似于多轴驱动器模块，1个站中包含多个逻辑轴的情况下，设置用于识别逻辑轴的多点编号。  * 以“#+编号(10进制数)”设置多点编号。 • #0: A轴 • #1: B轴 • #2: C轴 * 多点编号可以省略。省略的情况下，将视为“#0”。	@192.168.3.10 @192.168.3.10#2
[VAR]	@POU名	设置局部标签的情况下设置POU名。 与在全局标签与局部标签中使用同一标签名的设置无关，无@POU的情况下，将作为全局标签处理。	
[AXIS]	@Position@CumulativePos	设置对象数据(位置)。 根据“数据名”中设置的MC_SOURCE型的列举符设置@Position或@CumulativePos。*5	@Position @CumulativePos
[DEV]	—	不能设置字符修饰。即使设置也将被忽略。	
[CONST]	@+加法运算值 @-减法运算值	每当参照值，更改读取值时使用。 设置对象修饰时可以在各周期中进行值的更改。 数据类型为REAL或LREAL的情况下，本设置将无效。	[CONST] (INT) 0@-10*6 [CONST] (BOOL) 0x00.3@+1*7

- *5 可设置的列举符如下所示。
○：可以设置，×：不能设置

MC_SOURCE型的列举符	内容	修饰	
		@Position	@CumulativePos
1: 指令当前值(mcSetValue)	上次的运算周期的指令值	○	○
2: 反馈值(mcActualValue)	上次的运算周期的反馈值	○	×
101: 最新指令当前值(mcLatestSetValue)	本次的运算周期的指令值	○	○
102: 最新反馈值(mcLatestActualValue)	本次的运算周期的反馈值	○	×

- *6 参照值将为“0、-10、-20、-30、…”。
*7 参照值每8个周期(2³)切换一次TRUE和FALSE。(将1字节的值每周期“+1”以获取位3的状态。)
[CONST]的位指定可以使用“0~7”。
*8 设置超出63字符的格式的情况下，应使用WSTRING型标签，并通过下述方法进行设置。
(1) 在运动系统内宣言WSTRING型全局标签，并存储数据名.位位置@对象修饰的部分。
(2) 作为TARGET_REF的WSTRING型标签，将设置(1)的标签名。(在末尾处附加表示参照的“^”。)

■[DEV]类型支持的软元件一览


[DEV]类型的支持软元件一览如下所示。

R: 仅可读取, RW: 可读取/写入, —: 不支持

软元件类型	软元件名	符号	支持
用户软元件	输入	X	RW*1*2
	直接访问输入	DX	—
	输出	Y	RW*1*2
	直接访问输出	DY	—
	内部继电器	M	RW*2
	锁存继电器	L	RW*2
	链接继电器	B	RW*2
	链接特殊继电器	SB	RW*2
	报警器	F	R
	变址继电器	V	—
	定时器	T	R*3
	累计定时器	ST	R*3
	长定时器	LT	R*3
	长累计定时器	LST	R*3
	计数器	C	R*3
	长计数器	LC	R*3
	数据寄存器	D	RW*2
	链接寄存器	W	RW*2
	链接特殊寄存器	SW	RW*2
	系统软元件	特殊继电器	SM
特殊寄存器		SD	RW
功能输入		FX	—
功能输出		FY	—
功能寄存器		FD	—
文件寄存器	文件寄存器	R	—
		ZR	—
变址寄存器	变址寄存器	Z	—
	(32位变址修饰)	(ZZ)	—
	长变址寄存器	LZ	—
指针	指针	P	—
	中断指针	I	—
模块访问软元件	模块访问软元件	Un\G	RW*4
CPU缓冲存储器访问软元件	CPU缓冲存储器访问软元件	U3E0\G	RW*5
刷新数据寄存器	刷新数据寄存器	RD	—
嵌套	嵌套	N	—
其他	网络No. 指定软元件	J	—
	智能模块No. 指定软元件	U	—

*1 仅可使用未分配的软元件。

*2 在刷新设置中设置了刷新目标的情况下, 监视设备站是否已连接。在断开设备站的情况下的动作, 请参阅下述手册。

所使用的控制器的用户手册

在“刷新设置”中设置了刷新目标的情况下, 只能使用“网络同步通信设置”中设置了“同步”的设备的信号。

*3 定时器、累计定时器、长定时器、长累计定时器、计数器、长计数器具有触点(S)、线圈(C)、当前值(N)。指定为数据名时, 按触点: “软元件符号S”、线圈: “软元件符号C”、当前值: 指定为“软元件符号N”。

数据名中指定累计定时器ST10的触点时: [DEV]STS10

数据名中指定长计数器LC10的当前值时: [DEV]LCN10

使用定时器、累计定时器、长定时器、长累计定时器、计数器、长计数器时, 应根据软元件的容量指定类型。

*4 仅在指定了控制器的运动功能部或CC-Link IE TSN功能部的缓冲存储器时支持。

*5 仅在指定了控制器的CPU缓冲存储器时支持。

SIGNAL_SELECT(信号选择)

变量名	名称	数据类型	内容
Source	信号	TARGET_REF ☞ 1428页 TARGET_REF(输入信号)	设置控制中使用的BOOL型的信号。 在Source(信号)中,只能设置下述数据类型。 • [OBJ](已映射到循环数据中) • [VAR] • [DEV] • [CONST]
Detection	信号检测方法	INT (MC_SIGNAL_LOGIC) ☞ 1459页 MC_SIGNAL_LOGIC[M]	设置信号的逻辑。 • 0: TRUE时检测(HighLevel) • 1: FALSE时检测(LowLevel) • 2: FALSE→TRUE(上升沿)时检测(RisingEdge) • 3: TRUE→FALSE(下降沿)时检测(FallingEdge) • 4: 上升沿/下降沿时检测(BothEdges)
CompensationTime	补偿时间	LREAL	以秒单位设置输入输出信号的补偿时间。 * 各功能中设置范围有所不同。
FilterTime	筛选时间	LREAL	以秒单位设置用于去除输入输出信号的振荡的筛选时间。 * 各功能中设置范围有所不同。

PROFILE_DATA

变量名	名称	数据类型	内容
Location	运算配置文件存储位置	FILE_LOCATION ☞ 1431页 FILE_LOCATION	设置运算配置文件的文件名称、存储位置。
ID	配置文件ID	PROFILE_ID ☞ 1431页 PROFILE_ID	设置运算配置文件的配置文件ID的编号。

FILE_LOCATION

变量名	名称	数据类型	内容
FileName	文件名称	WSTRING(63)	设置文件名称。 (可以设置包含扩展名的63个字符。)* ¹
Path	文件夹指定	WSTRING(63)	设置存储文件的文件夹路径。 (最多可设置63字符。)* ¹

*¹ 文件名称(FileName)与文件夹指定(Path)的字符数应设置为合计最多127字符(包含null字符)。

PROFILE_ID

变量名	名称	数据类型	内容
Number	配置文件ID编号	WORD(UINT)	设置配置文件ID编号。 • 0~60000
StartIO	输入输出编号	WORD	设置输入输出No.。

CYCLE_PARAM

变量名	名称	数据类型	内容
NumOfCycleOverWngDetectTimes	周期溢出警告检测次数	WORD (UINT)	可设置检测出各种周期的周期溢出警告为止的周期数。设置了0的情况下不检测警告。 设置了超出范围的值的的情况下，将以默认值执行动作。 • 0~5次
NumOfCycleOverErrDetectTimes	周期溢出错误检测次数	WORD (UINT)	可设置检测出各种周期的周期溢出错误为止的周期数。设置了超出范围的值的的情况下，将以默认值执行动作。 • 1~5次
CycleOverErrorType	周期溢出错误选择	INT (MC_ERROR_CLASS) ☞ 1459页 MC_ERROR_CLASS[M]	设置各周期的溢出错误的分配。 • 2: 轻度异常 (MinorError) • 3: 中度异常 (ModerateError)

CYCLE_MONI

变量名	名称	数据类型	内容
ProcessingTime	处理时间	DWORD (UDINT)	存储各周期的处理时间[ns]。
MaximumProcessingTime	最大处理时间	DWORD (UDINT)	存储各周期的最大处理时间[ns]。
Cycle	设置周期	DWORD (UDINT)	存储各周期的周期设置[ns]。
CycleOver	周期溢出	BOOL	检测出各周期的周期溢出时将变为TRUE。

ADDON_PARAM

变量名	名称	数据类型	内容
RamSizeMax	RAM最大容量	DWORD (UDINT)	设置插件中使用的系统存储器 (RAM) 的最大使用量[k字节]。
BackupRamSizeMax	备份RAM最大容量	DWORD (UDINT)	设置插件中使用的系统存储器 (备份RAM) 的最大使用量[k字节]。

ADDON_MONI

变量名	名称	数据类型	内容
RamUsage	RAM使用量	DWORD (UDINT)	存储插件中使用的系统存储器 (RAM) 的当前的使用量[k字节]。
RamMaxUsage	RAM使用量最大	DWORD (UDINT)	存储插件中使用的系统存储器 (RAM) 的最大使用量[k字节]。
BackupRamUsage	备份RAM使用量	DWORD (UDINT)	存储插件中使用的系统存储器 (备份RAM) 的当前的使用量[k字节]。
BackupRamMaxUsage	备份RAM使用量最大	DWORD (UDINT)	存储插件中使用的系统存储器 (备份RAM) 的最大使用量[k字节]。
Version	版本	STRING (15)	存储插件的版本信息。

INSTANCE_ID[I]

变量名	名称	数据类型	内容
StartIO	IO编号	WORD (HEX)	设置IO编号。
Number	实例ID编号	WORD (UINT)	设置实例ID的编号。

PROFILE_CAM_DATA[P]

变量名	名称 (软元件偏置)	数据类型	内容
Interpolate	插补方法指定 (+0)	INT	设置凸轮数据的插补方法指定。 • 0: 直线插补 • 1: 各区间中指定 • 2: 样条插补
Resolution	分辨率 (+2)	DWORD (UDINT)	设置1周期的凸轮的分辨率。 • 8~65535
InputUnitString	输入单位字符串 (+4)	WSTRING (31)	以字符串设置输入数据的单位。 省略的情况下, 将作为“pulse”处理。
OutputUnitString	输出单位字符串 (+36)	WSTRING (31)	以字符串设置输出数据的单位。 省略的情况下, 将作为“pulse”处理。
StartPoint	开始点*1 (+68)	LREAL	设置开始点。 应设置“0.0”。 * 设置了“0.0”以外的情况下, 将变为运算配置文件不正确(错误代码: 1B8FH)。
StartStroke	初始行程量*1 (+72)	LREAL	设置对开始点的行程量。 (-行程量(Stroke) ≤ 初始行程量(StartStroke) ≤ 行程量(Stroke)) • -100.00000000~100.00000000
StartVelocity	初始速度*2 (+76)	LREAL	设置对开始点的速度。 • -2500000000.0 < 初始速度(StartVelocity) ≤ 2500000000.0
StartAcceleration	初始加速度*2 (+80)	LREAL	设置对开始点的加速度。 • -2147483647.0 < 初始加速度(StartAcceleration) ≤ 2147483647.0
CycleLength	1周期长*1 (+84)	LREAL	设置1周期所需的输入量。 (0.00000000000001 ≤ 1周期长(CycleLength) ≤ 10000000000.0) • 0.00000000000001~10000000000.0
CycleMin	1周期最小值 (+88)	LREAL	设置1周期的最小值。 应设置“0.0”。 * 设置了“0.0”以外的情况下, 将变为运算配置文件不正确(错误代码: 1B8FH)。
CycleMax	1周期最大值 (+92)	LREAL	设置1周期的最大值。 (1周期最小值(CycleMin) < 1周期最大值(CycleMax) ≤ 10000000000.0) • -10000000000.0~10000000000.0
CycleTime	1周期时间*2 (+96)	LREAL	设置1周期的时间。 (0.001 < 1周期时间(CycleTime) ≤ 100000.0) • 0.001~100000.000
Stroke	行程量 (+100)	LREAL	设置行程量。 (0.00000000000001 ≤ 行程量(Stroke) ≤ 10000000000.0) • 0.00000000000001~10000000000.0
NumberOfSections	区间数*1 (+104)	DWORD (UDINT)	设置凸轮数据的区间数。 ■设置了“1: 各区间中指定”的情况下 • 1~360 ■设置了“2: 样条插补”的情况下 • 3~360 * 设置了“0: 直线插补”的情况下不使用。即使设置也将被忽略。
Options	选项 (+106)	DWORD (HEX)	以位指定设置选项。 ■重复动作(位0) • 0: 单发动作 • 1: 重复动作 ■主轴绝对位置(位1) • 0: 相对位置 ■从轴绝对位置(位2) • 0: 相对位置 • 1: 绝对位置 *在位3~31中应设置“0”。

*1 在插补方法指定(Interpolate)中设置了“0: 直线插补”的情况下不使用。即使设置也将被忽略。

*2 仅在凸轮曲线为“12: 5次(调整)(FifthCurve_SpeedDesignation)”的情况下使用。其它曲线的情况下忽略。

PROFILE_CAM_ELEMENT[P]

变量名	名称 (软元件偏置)	数据类型	内容
CurveType	凸轮曲线类型 (+0)	INT (MC_CAM_CURVE_TYPE) ☞ 1457页 MC_CAM_CURVE_TYPE[M]	设置凸轮曲线。 <ul style="list-style-type: none"> • 0: 匀速 (ConstantSpeed) • 1: 恒定加速度 (ConstantAcceleration) • 2: 变形梯形 (DistortedTrapezoid) • 3: 变形正弦 (DistortedSine) • 4: 变形匀速 (DistortedConstantSpeed) • 5: 摆线 (Cycloid) • 6: 5次 (FifthCurve) • 7: 梯弦型 (Trapecloid) • 8: 反梯弦型 (ReverseTrapecloid) • 9: 双弦 (DoubleHypotenuse) • 10: 反双弦 (ReverseDoubleHypotenuse) • 11: 单弦 (SingleHypotenuse) • 12: 5次 (调整) (FifthCurve_SpeedDesignation)
EndPoint	结束点*1 (+4)	LREAL	设置对1周期长的位置(1周期当前值)。 <ul style="list-style-type: none"> • $0.0 < \text{结束点(EndPoint)} \leq 1\text{周期长(CycleLength)}$
Stroke	行程 (+8)	LREAL	设置行程位置。 <ul style="list-style-type: none"> • 行程(Stroke)的绝对值 \leq 行程量(Stroke)
RangeP1	曲线适用范围(P1) (+12)	LREAL	设置凸轮曲线的曲线适用范围(始点: P1, 终点: P2)。 设置时应满足“ $P1 < P2$ ”的条件。但是, “ $P1 = P2 = 0$ ”时, 将适用“ $P1 = 0$ ”、“ $P2 = 1$ ”。 <ul style="list-style-type: none"> • $0.0 \sim 1.0$
RangeP2	曲线适用范围(P2) (+16)	LREAL	
RangeL1	加减速范围补偿(范围L1) (+20)	LREAL	设置凸轮曲线中加减速的范围(L1、L2)。 根据凸轮曲线可设置的范围有所不同。 “ $L1 = L2 = 0.0000$ ”时, 适用各凸轮曲线的默认值。 在不使用L1、L2的曲线中, 忽略设定值。 <ul style="list-style-type: none"> • $0.0001 < \text{范围(L1、L2)} < 1.0000$
RangeL2	加减速范围补偿(范围L2) (+24)	LREAL	
EndVelocity	终点速度*2 (+28)	LREAL	设置凸轮曲线的终点速度。 <ul style="list-style-type: none"> • $-2500000000.0 < \text{终点速度(EndVelocity)} < 2500000000.0$
EndAcceleration	终点加速度*2 (+32)	LREAL	设置凸轮曲线的终点加速度。 <ul style="list-style-type: none"> • $-2147483647.0 < \text{终点加速度(EndAcceleration)} < 2147483647.0$

*1 在到达最终区间之前结束点超过了1周期长的情况下, 将该时刻的区间编号作为最终区间编号, 并以1周期长覆盖结束点。

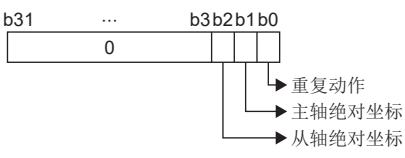
*2 根据运算配置文件的插补方法指定, 读取或写入中使用的设置项目有所不同。

PROFILE_ROTARY_CUTTER[P]

变量名	名称 (软元件偏置)	数据类型	内容
Resolution	分辨率 (+0)	DWORD (UDINT)	设置要创建的凸轮的分辨率。 • 8~32768
InputUnitString	输入单位字符串 (+2)	WSTRING (31)	设置输入单位字符串。
Options	选项 (+34)	DWORD (HEX)	<p>以位指定设置选项。</p> <p> ■重复动作(位0) • 0: 单发动作 • 1: 重复动作 ■主轴绝对位置(位1) • 0: 相对位置 ■从轴绝对位置(位2) • 0: 相对位置 • 1: 绝对位置 * 在位3~31中应设置“0”。 </p>
AutoGenerationOptions	自动创建选项 (+36)	WORD (HEX)	<p>以位指定设置自动创建选项。</p> <p> ■加减速方式(位0) • 0: 梯形加减速 • 1: S字加减速 ■同步轴长设置(位1) • 0: 直径 • 1: 周长 * 在位2~15中应设置“0”。 </p>
NumberOfCutter	刀具数 (+37)	WORD (UINT)	设置刀具数。 • 设置范围: 1~256
SyncSectionAccRatio	同步区间增速率 (+40)	LREAL	对同步区间的同步速度进行微调整时进行此设置。 变为“同步区间速度 = 同步速度 × (100% + 增速率)”。 • -50.0~50.0
SheetLength	表单长 (+44)	LREAL	设置表单长。 (0.0 < 表单长(SheetLength) ≤ 10000000000.0) • 0.00000000000001~10000000000.0
SheetSyncWidth	表单同步宽度 (+48)	LREAL	设置表单同步宽度(密封宽度)。 表单同步宽度前后需要保存动作作用的同步速度区间的情况下, 应按保存宽度添加。 (0.0 < 表单同步宽度(SheetSyncWidth) < 表单长(SheetLength)) (0.0 < 表单同步宽度(SheetSyncWidth) < 周长 / 刀具数(NumberOfCutter)) • 0.00000000000001~10000000000.0
SyncAxisLength	同步轴长 (+52)	LREAL	设置旋转刀具轴长。 ■自动创建选项为“0: 直径”的情况下 作为“周长 = 设定值 × π”进行计算。 ■自动创建选项为“1: 周长”的情况下 作为“周长 = 设定值”进行计算。 (0.0 < 周长 < 10000000000.0) • 0.00000000000001~10000000000.0
SyncPositionAdjustment	同步位置调整 (+56)	LREAL	设置同步区间的位置调整。 负的值: 将同步区间调整到表单起始侧 0: 表单中央为同步区间 正的值: 将同步区间调整到表单终端侧 (同步位置调整(SyncPositionAdjustment)的绝对值 < 表单长(SheetLength) / 2) • -10000000000.0~10000000000.0

变量名	名称 (软元件偏置)	数据类型	内容
AccDecWidth	加减速宽度 (+60)	LREAL	设置加减速区域的表单宽度(单侧)。 设置了负的值的情况下, 进行计算以确保加减速宽度为最大。 (2 × 加减速宽度(AccDecWidth) ≤ 表单长(SheetLength) - 表单同步宽度(SheetSyncWidth)) • -10000000000.0~10000000000.0
AsyncSpd	非同步速度结果 (+64)	LREAL	自动创建正常完成时, 非同步速度将作为同步速度的比率存储。

PROFILE_CAMSWITCH_DATA

变量名	名称 (软元件偏置)	数据类型	内容
InputUnitString	输入单位字符串 (+0)	WSTRING(31)	以字符串设置输入数据的单位。
NumberOfSwitches	开关数 (+32)	DWORD(UDINT)	指定开关总数。 • 1~10000
Options	选项 (+34)	DWORD(HEX)	以位指定设置选项。 在位3~31中应设置“0”。  ■重复动作(位0)*1 • 0: 单发动作 • 1: 重复动作 ■主轴绝对坐标(位1)*1 • 0: 相对坐标 • 1: 绝对坐标 ■从轴绝对位置(位2)*1 • 0: 相对坐标 • 1: 绝对坐标
ReferenceSpeed	基准速度 (+36)	LREAL	不使用本设置。应设置0.0。
ReferenceSpeedUnitString	基准速度单位字符串 (+40)	WSTRING(31)	不使用本设置。应设置空白。
RingCount_LowerValue	环形计数器下限值 (+72)	LREAL	设置环形计数器的下限值。
RingCount_UpperValue	环形计数器上限值 (+76)	LREAL	设置环形计数器的上限值。

*1 应指定“1”。

PROFILE_CAMSWITCH_ELEMENT

变量名	名称 (软元件偏置)	数据类型	内容
FirstOnPosition	ON开始位置 (+0)	LREAL	设置开关为ON的下限位置或开关为ON的开始位置。 • -10000000000.0~10000000000.0
LastOnPosition	ON结束位置 (+4)	LREAL	设置开关为ON的上限位置。 • -10000000000.0~10000000000.0
Direction	方向选择 (+8)	INT	设置开关为ON的输入方向。 • 0: 两方向 • 1: 正方向 • 2: 负方向
CamSwitchMode	开关模式选择 (+9)	INT	设置ON条件的指定方法。 • 0: 开关无效 • 1: 位置基准
Duration	ON时间 (+12)	LREAL	不使用本设置。设定值将被忽略。

PROFILE_POSITIONING_DATA

变量名	名称 (软元件偏置)	数据类型	内容
NumberOfDataSettings	定位数据设置数 (+0)	WORD(UINT)	设置定位数据数。 将定位数据设置数(NumberOfDataSettings)设置为不超过定位总数(TotalNumberOfData)。 • 0~5000 设置了超出范围的值,或者设置了超出定位数据总数的值的情况下,展开时将变为运算配置文件不正确(错误代码:1B8FH)。
TotalNumberOfData	定位数据总数 (+1)	WORD(UINT)	设置定位数据的总数。 设置展开或写入时要设置的定位数据号的最大值。 • 1~5000 设置了超出范围的值的情况下,展开时将变为运算配置文件不正确(错误代码:1B8FH)。
Options	选项 (+2)	DWORD(HEX)	设置选项。 应设置“0000000H”。

PROFILE_POSITIONING_DATA_ELEMENT

变量名	名称 (软元件偏置)	数据类型	内容
DataNo	定位数据号 (+0)	WORD(UINT)	设置定位数据的索引编号。 按照定位数据号(DataNo)的顺序执行定位数据。按照定位数据号(DataNo)的升序设置定位数据。 • 1~定位数据总数(5000)
OperationPattern	运行类型 (+1)	INT (MC_OPERATION_PAT TERN) ☞ 1460页 MC_OPERATION_PAT TERN	设置定位数据的运行类型。 设置继续执行还是结束定位数据号。 • 0: 定位结束(PositioningComplete) • 1: 连续定位(ContinuousPositioning) • 2: 连续轨迹(BlendingLow)(ContinuousBlendingLow) • 3: 连续轨迹(BlendingPrevious)(ContinuousBlendingPrevious) • 4: 连续轨迹(BlendingNext)(ContinuousBlendingNext) • 5: 连续轨迹(BlendingHigh)(ContinuousBlendingHigh)
ControlMethod	控制方式 (+2)	INT (MC_CONTROL_METHO D) ☞ 1460页 MC_CONTROL_METHOD	设置定位数据号的控制方式。 • 0080H: NOP(NOP) • 0082H: JUMP(JUMP) • 0083H: LOOP(LOOP) • 0084H: LEND(LEND) • 0101H: 绝对值直线插补(LinearAbsolute) • 0102H: 相对值直线插补(LinearRelative) • 0103H: 绝对值圆弧插补(CircularAbsolute) • 0104H: 相对值圆弧插补(CircularRelative) • 0105H: 绝对值螺旋插补(HelicalAbsolute) • 0106H: 相对值螺旋插补(HelicalRelative)

变量名	名称 (软件元件偏置)	数据类型	内容
InterpolationAxis1	插补轴1 (+3)	WORD (UINT)	从轴组的配置轴中，设置执行插补控制的插补轴。 通过控制方式(ControlMethod)的设置，设置直线插补的直线插补轴或圆弧插补的圆弧插补轴。
InterpolationAxis2	插补轴2 (+4)	WORD (UINT)	作为插补轴指定的配置轴，可以对各定位数据设置不同的配置轴和插补轴数。
InterpolationAxis3	插补轴3 (+5)	WORD (UINT)	但是，在连续轨迹数据与下一个定位数据之间，指定的配置轴或插补轴数不同时，在定位数据切换时各轴速度可能会急剧变化，应加以注意。
InterpolationAxis4	插补轴4 (+6)	WORD (UINT)	• 1~16
CircMode	圆弧插补模式 (+7)	INT (MC_CIRC_MODE) ☞ 1456页 MC_CIRC_MODE[M]	设置圆弧插补控制的指定方法。 • 0: 边界点指定 (mcBorder) • 1: 中心点指定 (mcCenter) • 2: 半径指定 (mcRadius)
CircPathChoice	路径选择 (+8)	INT (MC_CIRC_PATHCHOICE) ☞ 1456页 MC_CIRC_PATHCHOICE[M]	设置圆弧插补控制的旋转方向。 • 0: CW (mcCW) • 1: CCW (mcCCW) • 2: 就近 (mcShortWay) • 3: 就远 (mcLongWay) • 4: CW就远 (mcCWLongWay) • 5: CCW就远 (mcCCWLongWay)
Reserve1[0..2]	保留1 (+9)	WORD (UINT)	不使用。(偏置调整用)
Position1	目标位置/移动量/终点1 (+12)	LREAL	根据控制方式(ControlMethod)的设置如下所示。 <直线插补>
Position2	目标位置/移动量/终点2 (+16)	LREAL	目标位置/移动量/终点1~4(Position1~Position4)设置插补轴1~4(InterpolationAxis1~InterpolationAxis4)中设置的配置轴的目标位置/移动量。
Position3	目标位置/移动量/终点3 (+20)	LREAL	• -10000000000.0~10000000000.0 可以省略与“0”或省略的插补轴1~4(InterpolationAxis1~InterpolationAxis4)对应的目标位置/移动量/终点1~4(Position1~Position4)。
Position4	目标位置/移动量/终点4 (+24)	LREAL	■“0101H: 绝对值直线插补(LinearAbsolute)”的情况下 设置绝对位置的目标位置。 ■“0102H: 相对值直线插补(LinearRelative)”的情况下 设置从定位数据执行时的当前位置开始的移动量。 <圆弧插补> 目标位置/移动量/终点1、2(Position1、Position2)是设置插补轴1、2(InterpolationAxis1、InterpolationAxis2)中设置的配置轴的目标位置/移动量。 • -10000000000.0~10000000000.0 可以省略目标位置/移动量/终点3、4(Position3、Position4)。(忽略设置。) ■“0103H: 绝对值圆弧插补(CircularAbsolute)”的情况下 设置绝对位置的终点位置。 ■“0104H: 相对值圆弧插补(CircularRelative)”的情况下 设置从定位数据执行时的当前位置到终点为止的移动量。
Direction1	方向选择1 (+28)	INT (MC_DIRECTION) ☞ 1455页 MC_DIRECTION[M]	通过直线插补控制设置从当前位置向目标位置移动的方向。 在控制方式(ControlMethod)中设置了“0101H: 绝对值直线插补(LinearAbsolute)”的情况下进行设置。 方向选择1~4(Direction1~Direction4)是设置插补轴1~4(InterpolationAxis1~InterpolationAxis4)中设置的配置轴的方向。
Direction2	方向选择2 (+29)	INT (MC_DIRECTION) ☞ 1455页 MC_DIRECTION[M]	• 1: 正方向 (mcPositiveDirection) • 2: 负方向 (mcNegativeDirection) • 3: 最短路径 (mcShortestWay)
Direction3	方向选择3 (+30)	INT (MC_DIRECTION) ☞ 1455页 MC_DIRECTION[M]	根据设置，情况如下所示。 • 对于软件行程限位有效的插补轴，将忽略设置。
Direction4	方向选择4 (+31)	INT (MC_DIRECTION) ☞ 1455页 MC_DIRECTION[M]	• 可以省略与“0”或省略的插补轴1~4(InterpolationAxis1~InterpolationAxis4)对应的方向选择1~4(Direction1~Direction4)的数组元素。

变量名	名称 (软件元件偏置)	数据类型	内容
CircAuxPoint1	辅助点1 (+32)	LREAL	设置进行圆弧插补控制的辅助点(边界点、中心点、半径)的位置。 根据圆弧插补模式(CircMode)的设置, 设置范围如下所示。
CircAuxPoint2	辅助点2 (+36)	LREAL	■ “0: 边界点指定(mcBorder)”、“1: 中心点指定(mcCenter)”的情况下 根据控制方式(ControlMethod), 设置如下所示。
CircAuxPoint3	辅助点3 (+40)	LREAL	• “0103H: 绝对值圆弧插补(CircularAbsolute)”: -10000000000.0 ≤ 设定值 < 10000000000.0
CircAuxPoint4	辅助点4 (+44)	LREAL	环形计数器有效的情况下, 将变为环形计数器范围。 • “0104H: 相对值圆弧插补(CircularRelative)”: -10000000000.0~10000000000.0 ■ “2: 半径指定(mcRadius)”的情况下 • 0.000001~2147483647.0
VelocityMode	速度模式 (+48)	INT (MC_INTERPOLATE_S PEED_MODE) 1457页 MC_INTERPOLATE_S PEED_MODE[M]	设置直线插补控制的速度模式。 • 0: 合成速度(VectorSpeed) • 1: 长轴速度(LongAxisSpeed) • 2: 基准轴速度(ReferenceAxisSpeed)
Reserve2[0..2]	保留2 (+49)	WORD(UINT)	不使用。(偏置调整用)
Velocity	速度 (+52)	LREAL	设置多轴插补的定位速度。 • 0.0000、0.0001~2500000000.0
Acceleration	加速度 (+56)	LREAL	设置多轴插补的加速度。 根据定位数据选项(DataOptions)的加减速方式设置(位0~2)的设置, 设置范围如下所示。 ■ “0: 加减速速度指定方式(mcAccDec)”的情况下 • 0.0000、0.0001~2147483647.0[U/s ²] ■ “1: 加减速时间恒定方式(mcFixedTime)”的情况下 • 0.000000、0.000001~8400.0[s]
Deceleration	减速度 (+60)	LREAL	设置多轴插补的减速度。 ■ “0: 加减速速度指定方式(mcAccDec)”的情况下 • 0.0000、0.0001~2147483647.0[U/s ²] ■ “1: 加减速时间恒定方式(mcFixedTime)”的情况下 • 不使用减速度(Deceleration)
Jerk	Jerk (+64)	LREAL	设置多轴定位控制的Jerk。 ■ “0: 加减速速度指定方式(mcAccDec)”的情况下 • 0.0000、0.0001~2147483647.0[U/s ³] ■ “1: 加减速时间恒定方式(mcFixedTime)”的情况下 • 不使用Jerk
CircErrorTolerance	圆弧插补误差允许值 (+68)	LREAL	执行中心指定的圆弧插补的情况下, 设置圆弧插补误差允许值。 • 0.000001~100000.0

变量名	名称 (软元件偏置)	数据类型	内容
DataOptions	定位数据选项 (+72)	DWORD (HEX)	<p>以位指定设置选项。</p> <p> ■加减速方式设置(位0~2) • 0: 加减速速度指定方式 (mcAccDec) • 1: 加减速时间恒定方式 (mcFixedTime) ■缓冲模式时位置选择(位3) 在控制方式(ControlMethod)中设置了“0102H: 相对值直线插补 (LinearRelative)”、“0104H: 相对值圆弧插补 (CircularRelative)”的情况下进行设置。 • 0: 指令当前位置 • 1: 反馈位置 “0101H: 绝对值直线插补 (LinearAbsolute)”、“0103H: 绝对值圆弧插补 (CircularAbsolute)”的情况下, 应设置“0”。 ■超出环形计数器的目标位置指定(位16) 在控制方式(ControlMethod)中设置了“0101H: 绝对值直线插补 (LinearAbsolute)”的情况下进行设置。 • 0: 不允许 • 1: 允许 “0102H: 相对值直线插补 (LinearRelative)”、“0103H: 绝对值圆弧插补 (CircularAbsolute)”、“0104H: 相对值圆弧插补 (CircularRelative)”的情况下, 应设置“0”。 </p>
Reserve3[0..1]	保留3 (+74)	WORD (UINT)	不使用。(偏置调整用)
DwellTime	停留时间 (+76)	LREAL	设置停留时间。 • 0.0[s]: 停留功能无效 • 0.000001~8400.0[s]: 停留功能有效 设置了小于0.000001的正数的情况下, 将其作为“0.0[s]”获取。
Mcode	M代码 (+80)	WORD (UINT)	设置各定位数据的M代码。 不输出M代码的情况下, 设置“0(初始值)”。 • 1~65535
McodeOutput_Override	M代码输出时序超驰 (+81)	INT (MC_MCODE_OUTPUT_OVERRIDE) 1460页 MC_MCODE_OUTPUT_OVERRIDE	设置各定位数据的M代码输出时序。 • 0: 使用FB选项 (UseFbOptions) • 1: WITH模式 (WithMode) • 2: AFTER模式 (AfterMode)
JumpDestinationDataNo	跳转目标定位数据号 (+82)	WORD (UINT)	设置JUMP中指定的跳转目标的定位数据号。 设置控制方式(ControlMethod)为“0082H: JUMP (JUMP)”以外的定位数据的定位数据号(DataNo)。 • 1~定位数据总数(5000)
ConditionSignalNo	条件信号No. (+83)	WORD (UINT)	设置条件信号No.。 不使用条件信号的情况下, 设置“0(初始值)”。 • 1~10
LoopCount	LOOP~LEND环路次数 (+84)	WORD (UINT)	设置环路控制的重复次数。 • 1~65535
SkipSignalNo	跳过信号No. (+85)	WORD (UINT)	设置跳过信号No.。 • 1~10 不使用跳过信号的情况下, 应设置“0”。
Pitch	螺距 (+86)	WORD (UINT)	设置螺距。 应设置“0”。
Reserve4	保留4 (+87)	WORD (UINT)	不使用。(偏置调整用)

PROFILE_POSITIONING_DATA_EXTENDED

变量名	名称 (软元件偏置)	数据类型	内容
ConditionSignal	条件信号 (+0)	WSTRING(63)	通过TARGET_REF型的字符串设置条件信号。关于详细情况，请参阅下述章节。 ☞ 1428页 TARGET_REF(输入信号) 字符串的设置范围如下所示。 ^{*1} ■数据类型 • [VAR] • [DEV] ■类型 • PROFILE_SIGNAL_SELECT_ELEMENT(☞ 1442页 PROFILE_SIGNAL_SELECT_ELEMENT) ""(不设置)的情况下，不执行读取/写入。
SkipSignal	跳过信号 (+64)	WSTRING(63)	通过TARGET_REF型的字符串设置跳过信号。关于详细情况，请参阅下述章节。 ☞ 1428页 TARGET_REF(输入信号) 字符串的设置范围如下所示。 ^{*1} ■数据类型 • [VAR] • [DEV] ■类型 • PROFILE_SIGNAL_SELECT_ELEMENT(☞ 1442页 PROFILE_SIGNAL_SELECT_ELEMENT) ""(不设置)的情况下，不执行读取/写入。
PositioningData	定位数据 (+128)	WSTRING(63)	通过TARGET_REF型的字符串设置定位数据。关于详细情况，请参阅下述章节。 ☞ 1428页 TARGET_REF(输入信号) 字符串的设置范围如下所示。 ■数据类型 • [VAR] • [DEV] ■类型 • PROFILE_POSITIONING_DATA_ELEMENT(☞ 1437页 PROFILE_POSITIONING_DATA_ELEMENT) ""(不设置)的情况下，将变为“读取/写入数据类型不正确”(错误代码： 1BAAH)。

*1 指定的类型不满足各结构体构件请求的情况下，将变为读取/写入数据类型不正确(错误代码：1BAAH)。此外，指定的数据类型不满足各结构体构件请求的情况下，将变为读取/写入数据类型不正确(错误代码：1BA8H)。

PROFILE_SIGNAL_SELECT_ELEMENT

变量名	名称 (软元件偏置)	数据类型	内容*1
StartIO	输入输出编号 (+0)	WORD(HEX)	不使用。 忽略设置。
Reserve1	保留1 (+1)	WORD(UINT)	不使用。(偏置调整用)
Target	对象 (+2)	WSTRING(63)	以字符串设置要使用的信号。关于详细情况，请参阅下述章节。 <small>☞ 1428页 TARGET_REF(输入信号)</small> 字符串的设置范围如下所示。 • "" (长度为0的字符串)*2 ■数据类型 • [OBJ]*3 • [VAR] • [DEV] ■类型 • BOOL
Detection	信号检测方法 (+66)	INT (MC_SIGNAL_LOGIC) <small>☞ 1459页</small> MC_SIGNAL_LOGIC[M]	指定信号的逻辑。 • 0: TRUE时检测(HighLevel) • 1: FALSE时检测(LowLevel)
Reserve2	保留2 (+67)	WORD(UINT)	不使用。(偏置调整用)
CompensationTime	补偿时间 (+68)	LREAL	设置“0.0”。
FilterTime	筛选时间 (+72)	LREAL	以秒单位设置用于去除输入输出信号的振荡的筛选时间。 • 0.0~5.0[s]

*1 写入时，不发生由于设定值导致的错误。写入的展开区域在运算配置文件控制FB中使用时，超出设置范围的情况下将发生错误。

*2 对象(Target)为“” (长度为0的字符串)的情况下，相应跳过信号将作为无设置进行处理。

*3 设置了[OBJ]的情况下，不能省略对象修饰。设置了省略时，将变为运算配置文件控制配置文件ID设置不正确警告(事件代码: 00D48H)。

SLAVE_OBJECT_REAL(系统用)

本变量为系统用变量，因此请勿使用

变量名	名称	数据类型	内容
ControlWord	ControlWord	WSTRING (63)	对驱动器模块进行状态切换请求。
EncoderIncrements	EncoderIncrements	WSTRING (63)	获取驱动器模块的编码器分辨率。
FollowingErrActualValue	FollowingErrActualValue	WSTRING (63)	是驱动器模块的滞留脉冲。
GearRatioMotorRevolutions	GearRatioMotorRevolutions	WSTRING (63)	对驱动器模块设置伺服电机轴旋转数(分子)。
HomeOffset	HomeOffset	WSTRING (63)	对驱动器模块设置机械坐标系统的零位置及原点复位位置的差。
MaxMotorSpeed	MaxMotorSpeed	WSTRING (63)	从驱动器模块中获取伺服电机的最大速度。
MaxTorque	MaxTorque	WSTRING (63)	从驱动器模块中获取伺服电机的最大转矩。
ModesOfOp	ModesOfOp	WSTRING (63)	对驱动器模块进行控制模式的切换请求。
ModesOfOpDisp	ModesOfOpDisp	WSTRING (63)	获取驱动器模块的控制模式。
NegativeTorqueLimitValue	NegativeTorqueLimitValue	WSTRING (63)	对驱动器模块设置负方向转矩限制值。
Polarity	Polarity	WSTRING (63)	设置驱动器模块的旋转方向选择。
PosActualValue	PosActualValue	WSTRING (63)	是驱动器模块的当前位置。
PosEncoderResolution	PosEncoderResolution	WSTRING (63)	返回Sub Index的总数(=2)。
PosEncoderResolutionMotorRevolutions	PosEncoderResolutionMotorRevolutions	WSTRING (63)	是驱动器模块的旋转数。
PositiveTorqueLimitValue	PositiveTorqueLimitValue	WSTRING (63)	对驱动器模块设置正方向转矩限制值。
ShaftRevolutions	ShaftRevolutions	WSTRING (63)	设置驱动器模块的驱动轴旋转数(分母)。
SiUnitPos	SiUnitPos	WSTRING (63)	获取驱动器模块的SI单位位置。
SiUnitVel	SiUnitVel	WSTRING (63)	获取驱动器模块的SI单位速度。
StatusWord	StatusWord	WSTRING (63)	获取驱动器模块的状态。
SupportedDriveModes	SupportedDriveModes	WSTRING (63)	获取驱动器模块的对应控制模式。
TargetPos	TargetPos	WSTRING (63)	是输出至驱动器模块的指令位置。
TargetTorque	TargetTorque	WSTRING (63)	是输出至驱动器模块的指令转矩。
TargetVelocity	TargetVelocity	WSTRING (63)	是输出至驱动器模块的指令速度。
TargetActualValue	TorqueActualValue	WSTRING (63)	是驱动器模块的当前转矩。
VelActualValue	VelActualValue	WSTRING (63)	是驱动器模块的当前速度。
vControlDi1	ControlDi1	WSTRING (63)	对驱动器模块进行输入软元件的设置。
vControlDi2	ControlDi2	WSTRING (63)	对驱动器模块进行输入软元件的设置。
vControlDi3	ControlDi3	WSTRING (63)	对驱动器模块进行输入软元件的设置。
vControlDi4	ControlDi4	WSTRING (63)	对驱动器模块进行输入软元件的设置。
vControlDi5	ControlDi5	WSTRING (63)	对驱动器模块进行输入软元件的设置。
vControlDi6	ControlDi6	WSTRING (63)	对驱动器模块进行输入软元件的设置。
vControlDi7	ControlDi7	WSTRING (63)	对驱动器模块进行输入软元件的设置。
vCurrentAlarm	CurrentAlarm	WSTRING (63)	从驱动器模块中获取发生中的报警。
vEncoderStatus1	EncoderStatus1	WSTRING (63)	从驱动器模块中获取编码器状态。
vHomeAbsCounter	HomeAbsCounter	WSTRING (63)	从驱动器模块中获取作为原点保存的编码器多旋转计数器。
vHomeCycleCounter	HomeCycleCounter	WSTRING (63)	从驱动器模块中获取电源接通时的编码器1旋转内位置。
vInitialAbsCounter	InitialAbsCounter	WSTRING (63)	从驱动器模块中获取电源接通时的编码器多旋转计数器。
vInitialCycleCounter	InitialCycleCounter	WSTRING (63)	从驱动器模块中获取电源接通时的编码器1旋转内位置。
vInitialPos	InitialPos	WSTRING (63)	从驱动器模块中获取电源接通时的当前位置。
vMaxAbsCounter	MaxAbsCounter	WSTRING (63)	从驱动器模块中获取编码器多旋转计数器的最大值。
vStatusDo1	StatusDo1	WSTRING (63)	获取驱动器模块的输出软元件状态。
vStatusDo2	StatusDo2	WSTRING (63)	获取驱动器模块的输出软元件状态。
vStatusDo3	StatusDo3	WSTRING (63)	获取驱动器模块的输出软元件状态。
vStatusDo4	StatusDo4	WSTRING (63)	获取驱动器模块的输出软元件状态。
vStatusDo5	StatusDo5	WSTRING (63)	获取驱动器模块的输出软元件状态。
vStatusDo6	StatusDo6	WSTRING (63)	获取驱动器模块的输出软元件状态。
vStatusDo7	StatusDo7	WSTRING (63)	获取驱动器模块的输出软元件状态。
vSupportedControlDi1	SupportedControlDi1	WSTRING (63)	获取驱动器模块支持的输入软元件。
vSupportedControlDi2	SupportedControlDi2	WSTRING (63)	获取驱动器模块支持的输入软元件。

变量名	名称	数据类型	内容
vSupportedControlDi3	SupportedControlDi3	WSTRING (63)	获取驱动器模块支持的输入软元件。
vSupportedControlDi4	SupportedControlDi4	WSTRING (63)	获取驱动器模块支持的输入软元件。
vSupportedControlDi5	SupportedControlDi5	WSTRING (63)	获取驱动器模块支持的输入软元件。
vSupportedControlDi6	SupportedControlDi6	WSTRING (63)	获取驱动器模块支持的输入软元件。
vSupportedControlDi7	SupportedControlDi7	WSTRING (63)	获取驱动器模块支持的输入软元件。
vSupportedStatusDo1	SupportedStatusDo1	WSTRING (63)	获取驱动器模块支持的输出软元件。
vSupportedStatusDo2	SupportedStatusDo2	WSTRING (63)	获取驱动器模块支持的输出软元件。
vSupportedStatusDo3	SupportedStatusDo3	WSTRING (63)	获取驱动器模块支持的输出软元件。
vSupportedStatusDo4	SupportedStatusDo4	WSTRING (63)	获取驱动器模块支持的输出软元件。
vSupportedStatusDo5	SupportedStatusDo5	WSTRING (63)	获取驱动器模块支持的输出软元件。
vSupportedStatusDo6	SupportedStatusDo6	WSTRING (63)	获取驱动器模块支持的输出软元件。
vSupportedStatusDo7	SupportedStatusDo7	WSTRING (63)	获取驱动器模块支持的输出软元件。
vSyncAbsCounter	SyncAbsCounter	WSTRING (63)	从驱动器模块中获取标尺测量编码器的多旋转计数器。
vSyncCycleCounter	SyncCycleCounter	WSTRING (63)	从驱动器模块中获取标尺测量编码器的1旋转内位置。
vVelLimitValue	VelLimitValue	WSTRING (63)	对驱动器模块设置速度限制值。
vWatchDogCounterD1	WatchDogCounterD1	WSTRING (63)	向驱动器模块通知看门狗计数器值。
vWatchDogCounterU1	WatchDogCounterU1	WSTRING (63)	将看门狗计数器值获取到驱动器模块中。
vEncoderStatus2	EncoderStatus2	WSTRING (63)	从驱动器模块中获取标尺测量编码器状态。
vScaleCycleCounter	ScaleCycleCounter	WSTRING (63)	从驱动器模块中获取标尺测量编码器的1旋转内位置。
vScaleAbsCounter	ScaleAbsCounter	WSTRING (63)	从驱动器模块中获取标尺测量编码器的多旋转计数器。
vScaleMeasurementEncoderResolution	ScaleMeasurementEncoderResolution	WSTRING (63)	从驱动器模块中获取标尺测量编码器的分辨率。
vScaleMeasurementEncoderReceptionStatus	ScaleMeasurementEncoderReceptionStatus	WSTRING (63)	从驱动器模块中获取编码器1旋转内位置。

SLAVE_OBJECT_ENCODER[S]

变量名	名称	数据类型	内容
PosActualValue	PosActualValue	WSTRING (63)	设置存储同步编码器输入的数据。 I/O设备处于连接中，将从设置目标中获取值。

SLAVE_OBJECT_VIRTUAL_ENCODER

变量名	名称	数据类型	内容
PosActualValue	PosActualValue	WSTRING (63)	<p>将作为编码器输入值使用的数据以字符串格式进行设置。类型中可设置[VAR]、[DEV]、[CONST]。</p> <p>■设置了[VAR]的情况下</p> <ul style="list-style-type: none"> 设置数据类型为(INT)、(DINT)、(WORD)、(DWORD)的数据。 使用[VAR]的情况下，不能设置局部标签。 <p>■设置了[DEV]、[CONST]的情况下</p> <ul style="list-style-type: none"> 在(INT)、(DINT)、(WORD)、(DWORD)中设置数据类型。 <p>* 省略设置时，使用编码器输入值 (AxisName.Cd.Encoder_InputValue)。</p>

其它结构体的变量(高级同步控制)

ADV_GEAR_PARAM(齿轮参数)

变量名	名称	获取	数据类型	属性	内容
MasterCompositeGear	主轴合成齿轮	运算周期 (FB启动中)	WORD (HEX)	LIST_WRITE _BACK	设置主轴与子输入轴的当前位置的合成方法。 <ul style="list-style-type: none"> • 0000H: 无主轴输入/无子输入轴输入 • 0001H: 主轴输入+/无子输入轴输入 • 0002H: 主轴输入-/无子输入轴输入 • 0010H: 无主轴输入/子输入轴输入+ • 0011H: 主轴输入+/子输入轴输入+ • 0012H: 主轴输入-/子输入轴输入+ • 0020H: 无主轴输入/子输入轴输入- • 0021H: 主轴输入+/子输入轴输入- • 0022H: 主轴输入-/子输入轴输入-
MasterNumerator	主轴齿轮分子	启动时	DINT	LIST_WRITE _BACK	设置主轴齿轮的分子。 <ul style="list-style-type: none"> • -2147483647~2147483647
MasterDenominator	主轴齿轮分母	启动时	DWORD (UDINT)	LIST_WRITE _BACK	设置主轴齿轮的分母。 <ul style="list-style-type: none"> • 1~2147483647
AuxCompositeGear	辅助轴合成齿轮	运算周期 (FB启动中)	WORD (HEX)	LIST_WRITE _BACK	设置主轴与子输入轴合成后的当前位置与辅助轴的当前位置的合成方法。 <ul style="list-style-type: none"> • 0000H: 无主轴输入/无辅助轴输入 • 0001H: 主轴输入+/无辅助轴输入 • 0002H: 主轴输入-/无辅助轴输入 • 0010H: 无主轴输入/辅助轴输入+ • 0011H: 主轴输入+/辅助轴输入+ • 0012H: 主轴输入-/辅助轴输入+ • 0020H: 无主轴输入/辅助轴输入- • 0021H: 主轴输入+/辅助轴输入- • 0022H: 主轴输入-/辅助轴输入-
AuxNumerator	辅助轴齿轮分子	启动时	DINT	LIST_WRITE _BACK	设置辅助轴齿轮的分子。 <ul style="list-style-type: none"> • -2147483647~2147483647
AuxDenominator	辅助轴齿轮分母	启动时	DWORD (UDINT)	LIST_WRITE _BACK	设置辅助轴齿轮的分母。 <ul style="list-style-type: none"> • 1~2147483647

ADV_CLUTCH_PARAM(离合器参数)

变量名	名称	获取	数据类型	属性	内容
MasterOnControl	主轴ON离合器控制设置	运算周期 (FB启动中)	INT (MC_CLUTCH_METHOD) ☞ 1461页 MC_CLUTCH_METHOD	LIST_WRITE _BACK	设置主轴离合器的ON控制方法。 <ul style="list-style-type: none"> • 0: 无效(NoClutch) (直接连接动作) • 1: 离合器指令 (ClutchCommand) • 2: 离合器指令上升沿 (ClutchLeadingEdge) • 3: 离合器指令下降沿 (ClutchTrailingEdge) • 4: 地址模式 (ClutchAddress) • 15: 输入输出数据指定 (ClutchSignal) *: 即使在同步控制中也可以更改, 但是不能从“0: 无效(NoClutch)”以外的设置更改为“0: 无效(NoClutch)”。
MasterOffControl	主轴OFF离合器控制设置	运算周期 (FB启动中)	INT (MC_CLUTCH_METHOD) ☞ 1461页 MC_CLUTCH_METHOD	LIST_WRITE _BACK	设置主轴离合器的OFF控制方法。 <ul style="list-style-type: none"> • 0: 无效(NoClutch) (OFF控制无效) • 1: 离合器指令 (ClutchCommand) (快捷动作) • 2: 离合器指令上升沿 (ClutchLeadingEdge) • 3: 离合器指令下降沿 (ClutchTrailingEdge) • 4: 地址模式 (ClutchAddress) • 15: 输入输出数据指定 (ClutchSignal)
MasterReference	主轴离合器参照地址设置	启动时	INT (MC_CLUTCH_REFERENCE) ☞ 1461页 MC_CLUTCH_REFERENCE	LIST_WRITE _BACK	设置主轴离合器控制时参照的地址。 <ul style="list-style-type: none"> • 0: 齿轮前当前位置 (GearFrontPosition) • 1: 齿轮后1周期当前位置 (GearPositionPerCycle) *: 应注意由于参照地址的设置, 齿轮与离合器的处理顺序可能会发生变化。

变量名	名称	获取	数据类型	属性	内容
MasterOnAddress	主轴离合器ON地址	运算周期 (FB启动中)	LREAL	LIST_WRITE _BACK	<p>主轴ON离合器控制设置 (AdvOutputName.Pr.Clutch.MasterOnControl) 设置为“4: 地址模式(ClutchAddress)”时, 设置使离合器ON的地址。 主轴离合器参照地址设置 (AdvOutputName.Pr.Clutch.MasterReference) 为“1: 齿轮后1周期当前位置(GearPositionPerCycle)”时, 将设置的值转换为“0.0~(1周期长度 - 0.00001)”的范围内后进行控制。 • -10000000000.0~10000000000.0</p>
MasterMovementAmountBeforeOn	主轴离合器ON前移动量	离合器ON条件成立时	LREAL	LIST_WRITE _BACK	<p>以带符号的值设置主轴离合器ON控制中从ON条件成立到离合器实际变为ON的参照地址的移动量。 • -2147483648.0~2147483647.0</p>
MasterOffAddress	主轴离合器OFF地址	运算周期 (FB启动中)	LREAL	LIST_WRITE _BACK	<p>主轴OFF离合器控制设置 (AdvOutputName.Pr.Clutch.MasterOffControl) 设置为“4: 地址模式(ClutchAddress)”时, 设置使离合器OFF的地址。 主轴离合器参照地址设置 (AdvOutputName.Pr.Clutch.MasterReference) 为“1: 齿轮后1周期当前位置(GearPositionPerCycle)”时, 将设置的值转换为“0.0~(1周期长度 - 0.00001)”的范围内后进行控制。 • -10000000000.0~10000000000.0</p>
MasterMovementAmountBeforeOff	主轴离合器OFF前移动量	离合器OFF条件成立时*1	LREAL	LIST_WRITE _BACK	<p>以带符号的值设置主轴离合器OFF控制中从OFF条件成立到离合器实际变为OFF的参照地址的移动量。 • -2147483648.0~2147483647.0</p>
MasterSmoothingMethod	主轴离合器平滑方式	启动时	INT (MC_CLUTCH_SMOOTHING_METHOD) ☞ 1461页 MC_CLUTCH_SMOOTHING_METHOD	LIST_WRITE _BACK	<p>设置主轴离合器ON/OFF时的平滑方式。 • 0: 直接(ClutchSmoothingDisabled) • 1: 时间常数方式(指数)(TimeConstantExponent) • 2: 时间常数方式(直线)(TimeConstantLinear) • 3: 滑存量方式(指数)(SlippageExponent) • 4: 滑存量方式(直线)(SlippageLinear) • 5: 滑动量方式(直线: 输入量跟踪)(SlippageLinearFollow)</p>
MasterOnSlippageAmount	主轴离合器ON时滑存量	离合器ON开始时	LREAL	LIST_WRITE _BACK	<p>主轴离合器平滑方式 (AdvOutputName.Pr.Clutch.MasterSmoothingMethod) 设置为“3: 滑存量方式(指数)(SlippageExponent)”、“4: 滑存量方式(直线)(SlippageLinear)”、“5: 滑动量方式(直线: 输入量跟踪)(SlippageLinearFollow)”时, 设置离合器ON时的滑存量。 • 0.0~2147483647.0 *: 设定值为负值时, 主轴离合器ON时滑存量 (AdvOutputName.Pr.Clutch.MasterOnSlippageAmount) 将作为“0.0(直接)”进行控制。</p>
MasterOffSlippageAmount	主轴离合器OFF时滑存量	离合器OFF开始时	LREAL	LIST_WRITE _BACK	<p>主轴离合器平滑方式 (AdvOutputName.Pr.Clutch.MasterSmoothingMethod) 设置为“3: 滑存量方式(指数)(SlippageExponent)”、“4: 滑存量方式(直线)(SlippageLinear)”、“5: 滑动量方式(直线: 输入量跟踪)(SlippageLinearFollow)”时, 设置离合器OFF时的滑存量。 • 0.0~2147483647.0 *: 设定值为负值时, 主轴离合器OFF时滑存量 (AdvOutputName.Pr.Clutch.MasterOffSlippageAmount) 将作为“0.0(直接)”进行控制。</p>
AuxOnControl	辅助轴ON离合器控制设置	运算周期 (FB启动中)	INT (MC_CLUTCH_METHOD) ☞ 1461页 MC_CLUTCH_METHOD	LIST_WRITE _BACK	<p>设置辅助轴离合器的ON控制方法。 • 0: 无效(NoClutch)(直接连接动作) • 1: 离合器指令(ClutchCommand) • 2: 离合器指令上升沿(ClutchLeadingEdge) • 3: 离合器指令下降沿(ClutchTrailingEdge) • 4: 地址模式(ClutchAddress) • 15: 输入输出数据指定(ClutchSignal) *: 即使在同步控制中也可以更改, 但是不能从“0: 无效(直接连接动作)(NoClutch)”以外的设置更改为“0: 无效(直接连接动作)(NoClutch)”。</p>
AuxOffControl	辅助轴OFF离合器控制设置	运算周期 (FB启动中)	INT (MC_CLUTCH_METHOD) ☞ 1461页 MC_CLUTCH_METHOD	LIST_WRITE _BACK	<p>设置辅助轴离合器的OFF控制方法。 • 0: 无效(NoClutch)(OFF控制无效) • 1: 离合器指令(ClutchCommand)(快捷动作) • 2: 离合器指令上升沿(ClutchLeadingEdge) • 3: 离合器指令下降沿(ClutchTrailingEdge) • 4: 地址模式(ClutchAddress) • 15: 输入输出数据指定(ClutchSignal)</p>

变量名	名称	获取	数据类型	属性	内容
AuxReference	辅助轴离合器参照地址设置	启动时	INT (MC_CLUTCH_REFERENCE) ☞ 1461页 MC_CLUTCH_REFERENCE	LIST_WRITE _BACK	设置辅助轴离合器控制时参照的地址。 • 0: 齿轮前当前位置(GearFrontPosition) • 1: 齿轮后1周期当前位置(GearPositionPerCycle) *: 应注意由于参照地址的设置, 齿轮与离合器的处理顺序可能会发生变化。
AuxOnAddress	辅助轴离合器ON地址	运算周期 (FB启动中)	LREAL	LIST_WRITE _BACK	辅助轴ON离合器控制设置 (AdvOutputName.Pr.Clutch.AuxOnControl) 设置为“4: 地址模式(ClutchAddress)”时, 设置使离合器ON的地址。 辅助轴离合器参照地址设置 (AdvOutputName.Pr.Clutch.AuxReference) 为“1: 齿轮后1周期当前位置(GearPositionPerCycle)”时, 将设置的值转换为“0.0~(1周期长度 - 0.00001)”的范围内后进行控制。 • -1000000000.0~1000000000.0
AuxMovementAmountBeforeOn	辅助轴离合器ON前移动量	离合器ON条件成立时	LREAL	LIST_WRITE _BACK	以带符号的值设置辅助轴离合器ON控制中从ON条件成立到离合器实际变为ON的参照地址的移动量。 • -2147483648.0~2147483647.0
AuxOffAddress	辅助轴离合器OFF地址	运算周期 (FB启动中)	LREAL	LIST_WRITE _BACK	辅助轴OFF离合器控制设置 (AdvOutputName.Pr.Clutch.AuxOffControl) 设置为“4: 地址模式(ClutchAddress)”时, 设置使离合器OFF的地址。 辅助轴离合器参照地址设置 (AdvOutputName.Pr.Clutch.AuxReference) 为“1: 齿轮后1周期当前位置(GearPositionPerCycle)”时, 将设置的值转换为“0.0~(1周期长度 - 0.00001)”的范围内后进行控制。 • -1000000000.0~1000000000.0
AuxMovementAmountBeforeOff	辅助轴离合器OFF前移动量	离合器OFF条件成立时*1	LREAL	LIST_WRITE _BACK	以带符号的值设置辅助轴离合器OFF控制中从OFF条件成立到离合器实际变为OFF的参照地址的移动量。 • -2147483648.0~2147483647.0
AuxSmoothingMethod	辅助轴离合器平滑方式	启动时	INT (MC_CLUTCH_SMOOTHING_METHOD) ☞ 1461页 MC_CLUTCH_SMOOTHING_METHOD	LIST_WRITE _BACK	设置辅助轴离合器ON/OFF时的平滑方式。 • 0: 直接(ClutchSmoothingDisabled) • 1: 时间常数方式(指数)(TimeConstantExponent) • 2: 时间常数方式(直线)(TimeConstantLinear) • 3: 滑存量方式(指数)(SlippageExponent) • 4: 滑存量方式(直线)(SlippageLinear) • 5: 滑动量方式(直线: 输入量跟踪)(SlippageLinearFollow)
AuxOnSlippageAmount	辅助轴离合器ON时滑存量	离合器ON开始时	LREAL	LIST_WRITE _BACK	辅助轴离合器平滑方式 (AdvOutputName.Pr.Clutch.AuxSmoothingMethod) 设置为“3: 滑存量方式(指数)(SlippageExponent)”、“4: 滑存量方式(直线)(SlippageLinear)”、“5: 滑动量方式(直线: 输入量跟踪)(SlippageLinearFollow)”时, 设置离合器ON时的滑存量。 • 0.0~2147483647.0 *: 设定值为负值时, 辅助轴离合器ON时滑存量 (AdvOutputName.Pr.Clutch.AuxOnSlippageAmount) 将作为“0.0(直接)”进行控制。
AuxOffSlippageAmount	辅助轴离合器OFF时滑存量	离合器OFF开始时	LREAL	LIST_WRITE _BACK	辅助轴离合器平滑方式 (AdvOutputName.Pr.Clutch.AuxSmoothingMethod) 设置为“3: 滑存量方式(指数)(SlippageExponent)”、“4: 滑存量方式(直线)(SlippageLinear)”、“5: 滑动量方式(直线: 输入量跟踪)(SlippageLinearFollow)”时, 设置离合器OFF时的滑存量。 • 0.0~2147483647.0 *: 设定值为负值时, 辅助轴离合器OFF时滑存量 (AdvOutputName.Pr.Clutch.AuxOffSlippageAmount) 将作为“0.0(直接)”进行控制。

*1 快捷动作时, 变为“离合器ON条件成立时”。

ADV_CLUTCH_MONI (离合器监视)

变量名	名称	获取	数据类型	属性	内容
MasterOnOffStatus	主轴离合器ON/OFF状态	运算周期(FB启动中)	BOOL	LIST_READ _ONLY	存储主轴离合器的ON/OFF状态。 • FALSE: 离合器OFF状态 • TRUE: 离合器ON状态
MasterSmoothingStatus	主轴离合器平滑状态	运算周期(FB启动中)	BOOL	LIST_READ _ONLY	存储主轴离合器的平滑状态。 • FALSE: 无离合器平滑 • TRUE: 离合器平滑中
MasterCumulativeSlippage	主轴离合器滑行程量累计值	运算周期(FB启动中)	LREAL	LIST_READ _ONLY	以带符号的值存储滑行程量方式的离合器平滑时的滑行程量累计值。
AuxOnOffStatus	辅助轴离合器ON/OFF状态	运算周期(FB启动中)	BOOL	LIST_READ _ONLY	存储辅助轴离合器的ON/OFF状态。 • FALSE: 离合器OFF状态 • TRUE: 离合器ON状态
AuxSmoothingStatus	辅助轴离合器平滑状态	运算周期(FB启动中)	BOOL	LIST_READ _ONLY	存储辅助轴离合器的平滑状态。 • FALSE: 无离合器平滑 • TRUE: 离合器平滑中
AuxCumulativeSlippage	辅助轴离合器滑行程量累计值	运算周期(FB启动中)	LREAL	LIST_READ _ONLY	以带符号的值存储滑行程量方式的离合器平滑时的滑行程量累计值。

ADV_CLUTCH_CMD(离合器控制数据)

变量名	名称	获取	数据类型	属性	内容
MasterCommand	主轴离合器指令	运算周期 (FB启动中)	BOOL	LIST_READ _ONLY	设置主轴离合器指令的ON/OFF。 主轴离合器控制设置 (AdvOutputName.Pr.Clutch.MasterOnControl)为“1: 离合器指令(ClutchCommand)”、“2: 离合器指令上升沿(ClutchLeadingEdge)”、“3: 离合器指令下降沿(ClutchTrailingEdge)”时使用。 • FALSE: 离合器指令OFF • TRUE: 离合器指令ON
MasterInvalidCommand	主轴离合器控制无效指令	运算周期 (FB启动中)	BOOL	LIST_READ _ONLY	切换主轴离合器控制的有效/无效。 离合器ON/OFF状态为保持离合器控制变为无效之前的状态。 离合器ON前移动中或离合器OFF前移动中, 离合器控制不变为无效。离合器ON前移动或离合器OFF前移动完成后, 离合器控制变为无效。 • FALSE: 离合器控制有效 • TRUE: 离合器控制无效
MasterForcedOff	主轴离合器强制OFF指令	运算周期 (FB启动中)	BOOL	LIST_READ _ONLY	使主轴离合器强制OFF。 • FALSE: 离合器通常控制 • TRUE: 离合器强制OFF *: 离合器平滑执行中, 离合器的输出将立即变为“0”。 *: 执行了滑行道方式的平滑时, 累计滑行道将被清除为0。 *: 离合器强制OFF后, 将设定值置为“FALSE”时, 将从离合器OFF状态开始重新开启通常的离合器控制。
AuxCommand	辅助轴离合器指令	运算周期 (FB启动中)	BOOL	LIST_READ _ONLY	设置辅助轴离合器指令的ON/OFF。 辅助轴离合器控制设置 (AdvOutputName.Pr.Clutch.AuxOnControl)设置为“1: 离合器指令(ClutchCommand)”、“2: 离合器指令上升沿(ClutchLeadingEdge)”、“3: 离合器指令下降沿(ClutchTrailingEdge)”时使用。 • FALSE: 离合器指令OFF • TRUE: 离合器指令ON
AuxInvalidCommand	辅助轴离合器控制无效指令	运算周期 (FB启动中)	BOOL	LIST_READ _ONLY	切换辅助轴离合器控制的有效/无效。 离合器ON/OFF状态为保持离合器控制变为无效之前的状态。 离合器ON前移动中或离合器OFF前移动中, 离合器控制不变为无效。离合器ON前移动或离合器OFF前移动完成后, 离合器控制变为无效。 • FALSE: 离合器控制有效 • TRUE: 离合器控制无效
AuxForcedOff	辅助轴离合器强制OFF指令	运算周期 (FB启动中)	BOOL	LIST_READ _ONLY	使辅助轴离合器强制OFF。 • FALSE: 离合器通常控制 • TRUE: 离合器强制OFF *: 离合器平滑执行中, 离合器的输出将立即变为“0”。 *: 执行了滑行道方式的平滑时, 累计滑行道将被清除为0。 *: 离合器强制OFF后, 将设定值置为“FALSE”时, 将从离合器OFF状态开始重新开启通常的离合器控制。
MasterClutchSmoothingTimeConstant	主轴离合器平滑时间常数更改值	启动时	INT	LIST_READ _ONLY	更改主轴离合器平滑时间常数 (AdvOutputName.Cd.Clutch.MasterClutchSmoothingTimeConstant)。 • 0~5000[ms] *: 设置了负值时, 不执行平滑。 *: 设置了“0”时, 以主轴离合器平滑时间常数 (AdvOutputName.PrConst.MasterClutchSmoothingTimeConstant)的设定值进行平滑。
AuxClutchSmoothingTimeConstant	辅助轴离合器平滑时间常数更改值	启动时	INT	LIST_READ _ONLY	更改辅助轴离合器平滑时间常数 (AdvOutputName.Cd.Clutch.AuxClutchSmoothingTimeConstant)。 • 0~5000[ms] *: 设置了负值时, 不执行平滑。 *: 设置了“0”时, 以辅助轴离合器平滑时间常数 (AdvOutputName.PrConst.AuxClutchSmoothingTimeConstant)的设定值进行平滑。

ADV_SPEEDCHANGE_GEAR_PARAM(变速器参数)

变量名	名称	获取	数据类型	属性	内容
MasterGearIn	主轴变速器启用设置	启动时	BOOL	LIST_WRITE _BACK	设置是否使用主轴的变速器。 • FALSE: 无变速器 • TRUE: 有变速器
MasterRatioNumerator	主轴变速比分子	运算周期 (FB启动中)	DINT	LIST_WRITE _BACK	设置主轴变速比的分子。 • -2147483648~2147483647
MasterRatioDenominator	主轴变速比分母	运算周期 (FB启动中)	DWORD(UDINT)	LIST_WRITE _BACK	设置主轴变速比的分子。 • 1~2147483647
AuxGearIn	辅助轴变速器启用设置	启动时	BOOL	LIST_WRITE _BACK	设置是否使用辅助轴的变速器。 • FALSE: 无变速器 • TRUE: 有变速器
AuxRatioNumerator	辅助轴变速比分子	运算周期 (FB启动中)	DINT	LIST_WRITE _BACK	设置辅助轴变速比的分子。 • -2147483648~2147483647
AuxRatioDenominator	辅助轴变速比分母	运算周期 (FB启动中)	DWORD(UDINT)	LIST_WRITE _BACK	设置辅助轴变速比的分子。 • 1~2147483647
OutGearIn	输出轴变速器启用设置	启动时	BOOL	LIST_WRITE _BACK	设置是否使用输出轴的变速器。 • FALSE: 无变速器 • TRUE: 有变速器
OutRatioNumerator	输出轴变速比分子	运算周期 (FB启动中)	DINT	LIST_WRITE _BACK	设置输出轴变速比的分子。 • -2147483648~2147483647
OutRatioDenominator	输出轴变速比分母	运算周期 (FB启动中)	DWORD(UDINT)	LIST_WRITE _BACK	设置输出轴变速比的分子。 • 1~2147483647

ADV_SPEEDCHANGE_GEAR_CMD(变速器控制数据)

变量名	名称	获取	数据类型	属性	内容
MasterSmoothingTimeConstant	主轴变速器平滑时间常数更改值	启动时	INT	LIST_WRITE _BACK	更改主轴变速器平滑时间常数。 • 0~5000[ms] *: 设置了负值时, 不执行平滑。 *: 设置了“0”时, 以主轴变速器平滑时间常数 (AdvOutPutName.PrConst.MasterSpeedChangeGearSmoothingTimeConstant)的设定值进行平滑。
AuxSmoothingTimeConstant	辅助轴变速器平滑时间常数更改值	启动时	INT	LIST_WRITE _BACK	更改辅助轴的变速器平滑时间常数。 • 0~5000[ms] *: 设置了负值时, 不执行平滑。 *: 设置了“0”时, 以辅助轴变速器平滑时间常数 (AdvOutPutName.PrConst.AuxSpeedChangeGearSmoothingTimeConstant)的设定值进行平滑。
OutSmoothingTimeConstant	输出轴变速器平滑时间常数更改值	启动时	INT	LIST_WRITE _BACK	更改输出轴的变速器平滑时间常数。 • 0~5000[ms] *: 设置了负值时, 不执行平滑。 *: 设置了“0”时, 以输出轴变速器平滑时间常数 (AdvOutPutName.PrConst.OutSpeedChangeGearSmoothingTimeConstant)的设定值进行平滑。

ADV_CAM_PARAM(凸轮参数)

变量名	名称	获取	数据类型	属性	内容
LengthPerCycle	1周期长度	启动时/凸轮数据第0点通过时	LREAL	LIST_WRITE _BACK	设置凸轮轴的1周期长以生成1周期当前位置。为MCv_AdvancedSync(高级同步控制)的主轴(Master)中设置的输入轴的单位。 • 1.0~2147483647.0
StrokeAmount	凸轮行程量	启动时/凸轮数据第0点通过时	LREAL	LIST_WRITE _BACK	运算配置文件的插补方法指定(Interpolate)为“1:对区间分别指定”或“2:样条插补”时,或数据格式的类型中设置为“旋转刀具”时,以输出轴位置单位设置行程比100%对应的凸轮行程量。 • -2147483648.0~2147483647.0
CamNo	凸轮No.	启动时/凸轮数据第0点通过时	WORD(UINT)	LIST_WRITE _BACK	设置凸轮控制中使用的凸轮No.。 • 0~60000
StartingPoint	凸轮开始点	启动时/凸轮数据第0点通过时	DWORD(UDINT)	LIST_WRITE _BACK	以分辨率设置凸轮数据的开始位置。运算配置文件的插补方法指定(Interpolate)为“1:对区间分别指定”或“2:样条插补”时,或数据格式的类型中设置为“旋转刀具”时有效。 凸轮开始点 (AdvOutputName.Pr.Cam.StrokeAmount)的初始值为“0”。(从凸轮数据第0点(行程比0%)开始控制凸轮轴。)本参数设置为“0”以外时,可以从0%以外的行程比开始控制凸轮。 • 0~65535
LengthPerCycleChange	1周期更改设置	启动时	INT (MC_LENGTH_PER_CYCLE_CHANGE) ☞ 1461页 MC_LENGTH_PER_CYCLE_CHANGE	LIST_WRITE _BACK	同步控制中更改1周期长度 (AdvOutputName.Pr.Cam.LengthPerCycle)时进行此设置。 • 0:无效(LengthChangeInvalid) • 1:有效(LengthChangeValid)
PhaseCompensationAdvancedTime	相位补偿超前时间	启动时/运算周期(FB启动中)	DINT	LIST_WRITE _BACK	凸轮控制时,在超前或延迟1周期当前位置的相位时进行此设置。 • -100000000~100000000[μs]
PhaseCompensationTimeConstant	相位补偿时间常数	启动时	WORD(UINT)	LIST_WRITE _BACK	设置将相位补偿时的相位补偿量通过一次延迟反映时的时间常数。 通过设置的时间常数相位补偿量的63%将被反映。 • 0~65535[ms]

ADV_CAM_MONI (凸轮监视)

变量名	名称	获取	数据类型	属性	内容
PhaseCompensationAmount	相位补偿量	运算周期 (FB启动中)	LREAL	LIST_READ_ONLY	以MCv_AdvancedSync (高级同步控制) 的主轴 (Master) 中设置的输入轴的单位存储凸轮轴的相位补偿量。 存储通过相位补偿时间常数 (AdvOutputName.Pr.Cam.PhaseCompensationTimeConstant) 进行平滑处理后的相位补偿量。
MasterCompositeGearSetPosition	主轴合成齿轮后当前位置	运算周期 (FB启动中)	LREAL	LIST_READ_ONLY	将通过主轴合成齿轮合成主轴与子输入轴后的当前位置作为累计值存储。 以MCv_AdvancedSync (高级同步控制) 的主轴 (Master) 中设置的输入轴的单位进行存储。
MasterGearPositionPerCycle	主轴齿轮后1周期当前位置	运算周期 (FB启动中)	LREAL	LIST_READ_ONLY	在“0.0~(1周期长度 - 0.00001)”的范围内存储主轴齿轮后的输入移动量。 为MCv_AdvancedSync (高级同步控制) 的主轴 (Master) 中设置的输入轴的单位。
AuxGearPositionPerCycle	辅助轴齿轮后1周期当前位置	运算周期 (FB启动中)	LREAL	LIST_READ_ONLY	在“0.0~(1周期长度 - 0.00001)”的范围内存储辅助轴齿轮后的输入移动量。 为MCv_AdvancedSync (高级同步控制) 的主轴 (Master) 中设置的输入轴的单位。
PositionPerCycle	1周期当前位置	运算周期 (FB启动中)	LREAL	LIST_READ_ONLY	在“0.0~(1周期长度 - 0.00001)”的范围内存储1周期当前位置。可以监视相位补偿处理后的当前位置。 为MCv_AdvancedSync (高级同步控制) 的主轴 (Master) 中设置的输入轴的单位。
ReferenceSetPosition	凸轮基准位置	运算周期 (FB启动中)	LREAL	LIST_READ_ONLY	存储凸轮动作的凸轮基准位置的指令当前位置。 为输出轴的轴信息 (AdvOutputName.Axis) 中设置的轴的位置指令单位。
SetPosition	凸轮指令当前位置	运算周期 (FB启动中)	LREAL	LIST_READ_ONLY	存储凸轮轴的指令当前位置。 为输出轴的轴信息 (AdvOutputName.Axis) 中设置的轴的位置指令单位。
CamNo	执行配置文件No.	运算周期 (FB启动中)	WORD (UINT)	LIST_READ_ONLY	存储执行中的凸轮No.。同步控制中更改了凸轮No. (AdvOutputName.Pr.Cam.CamNo) 时, 实际上控制中的凸轮No. 将在切换时更新。也在执行配置文件ID编号 (AxisName.Md.ProfileID) 中存储相同的值。
StrokeAmount	执行凸轮行程量	运算周期 (FB启动中)	LREAL	LIST_READ_ONLY	存储执行中的凸轮行程量。 同步控制中更改了凸轮行程量 (AdvOutputName.Pr.Cam.StrokeAmount) 时, 实际上控制中的凸轮行程量将在切换时更新。
LengthPerCycle	执行1周期长度	运算周期 (FB启动中)	LREAL	LIST_READ_ONLY	存储执行中的凸轮1周期长度。同步控制中更改了凸轮1周期长度 (AdvOutputName.Pr.Cam.LengthPerCycle) 时, 实际上控制中的凸轮1周期长度将在切换时更新。
SyncControlChangeStatus	同步控制更改状态	运算周期 (FB启动中)	INT	LIST_READ_ONLY	存储通过同步控制更改请求 (AdvOutputName.Cd.Cam.SyncControlChangeRequest) 执行了同步控制更改的状态。 <ul style="list-style-type: none"> • -1: 同步控制更改失败 • 0: 无同步控制更改请求 • 1: 同步控制更改中 • 2: 同步控制更改完成
StartingPoint	执行凸轮开始点	运算周期 (FB启动中)	DWORD (UDINT)	LIST_READ_ONLY	存储执行中的凸轮开始点。同步控制中更改了凸轮开始点 (AdvOutputName.Pr.Cam.StartingPoint) 时, 凸轮将在切换时更新。

ADV_CAM_CMD(凸轮控制数据)

变量名	名称	获取	数据类型	属性	内容
SyncControlChangeRequest	同步控制更改请求	运算周期 (FB启动中)	BOOL	LIST_READ_ONLY	执行同步控制更改请求。 设置TRUE(同步控制更改请求)时,执行同步控制更改指令 (AdvOutputName, Cd. Cam. SyncControlChangeCommand)。 • FALSE: 无同步控制更改请求 • TRUE: 同步控制更改请求
SyncControlChangeCommand	同步控制更改指令	运算周期 (FB启动中)	INT (MC_SYNC_CHANGE_COMMAND) ☞ 1461页 MC_SYNC_CHANGE_COMMAND	LIST_READ_ONLY	设置同步控制更改指令。 • 0: 凸轮基准位置移动 (ReferenceSetPositionMovement) • 1: 1周期当前位置更改 (ChangeCurrentPositionPerCycle) • 2: 主轴齿轮后1周期当前位置更改 (ChangeMasterGearPositionPerCycle) • 3: 辅助轴齿轮后1周期当前位置更改 (ChangeAuxGearPositionPerCycle) • 4: 1周期当前位置移动 (PositionPerCycleMovement)
SyncControlChange	同步控制更改值	运算周期 (FB启动中)	LREAL	LIST_READ_ONLY	设置同步控制更改处理的更改值。 • -2147483648.0~2147483647.0
SyncControlReflectionTime	同步控制更改反映时间	运算周期 (FB启动中)	WORD (UINT)	LIST_READ_ONLY	设置同步控制更改处理的反映时间。 • 0~65535[ms]
OutSmoothingTimeConstant	输出轴平滑时间常数更改值	启动时	INT	LIST_WRITE_BACK	更改输出轴平滑时间常数。 • 0~5000[ms] *: 设置了负值时,不执行平滑。 *: 设置了“0”时,以输出轴平滑时间常数 (AdvOutputName, PrConst, SmoothingTimeConstant)的设定值进行平滑。

ADV_RESTORE_PARAM(同步控制初始位置参数)

变量名	名称	获取	数据类型	属性	内容
MasterGearPositionPerCycleMethod	主轴齿轮后1周期当前位置设置方法	启动时	INT (MC_GEAR_RESTORE_METHOD) ☞ 1461页 MC_GEAR_RESTORE_METHOD	LIST_WRITE_BACK	MCv_AdvancedSync(高级同步控制)启动时设置主轴齿轮后1周期当前位置 (AdvOutputName, Md. Gear. MasterGearPositionPerCycle)的设置方法。 • 0: 上次值(PreviousPosition) • 1: 主轴齿轮后1周期当前位置初始设定值 (InitialGearPositionPerCycle) • 2: 通过输入轴计算 (CalculateFromInputAxis)
MasterGearInitialPositionPerCycle	主轴齿轮后1周期当前位置初始设定值	启动时	LREAL	LIST_WRITE_BACK	主轴齿轮后1周期当前位置设置方法 (AdvOutputName, Md. Gear. MasterGearPositionPerCycleMethod)设置为“1: 主轴齿轮后1周期当前位置初始设定值 (InitialGearPositionPerCycle)”时,设置主轴齿轮后1周期当前位置的初始设定值。 在“0.0~(1周期长度 - 0.00001)”的范围内设置。 • 0.0~2147483647.0
AuxGearPositionPerCycleMethod	辅助轴齿轮后1周期当前位置设置方法	启动时	INT (MC_GEAR_RESTORE_METHOD) ☞ 1461页 MC_GEAR_RESTORE_METHOD	LIST_WRITE_BACK	MCv_AdvancedSync(高级同步控制)启动时设置辅助轴齿轮后1周期当前位置 (AdvOutputName, Md. Gear. AuxGearPositionPerCycle)的设置方法。 • 0: 上次值(PreviousPosition) • 1: 辅助轴齿轮后1周期当前位置初始设定值 (InitialGearPositionPerCycle) • 2: 通过输入轴计算 (CalculateFromInputAxis)

变量名	名称	获取	数据类型	属性	内容
AuxGearInitialPositionPerCycle	辅助轴齿轮后1周期当前位置初始设定值	启动时	LREAL	LIST_WRITE _BACK	辅助轴齿轮后1周期当前位置设置方法 (AdvOutputName, Md. Gear. AuxGearPositionPerCycle) 设置为“1: 辅助轴齿轮后1周期当前位置初始设定值(InitialGearPositionPerCycle)”时, 设置辅助轴齿轮后1周期当前位置的初始设定值。 在“0.0~(1周期长度 - 0.00001)”的范围内设置。 • 0.0~2147483647.0
PositionRestorationObject	凸轮轴位置恢复对象	启动时	INT (MC_CAM_RESTORE_METHOD) ☞ 1462页 MC_CAM_RESTORE_METHOD	LIST_WRITE _BACK	同步控制启动时设置要恢复的对象。 • 0: 1周期当前位置恢复 (RestorePositionPerCycle) • 1: 凸轮基准位置恢复 (RestoreReferenceSetPosition) • 2: 凸轮指令当前位置恢复 (RestoreCamSetPosition)
ReferenceSetPositionMethod	凸轮基准位置设置方法	启动时	INT (MC_CAM_REFERENCE_METHOD) ☞ 1462页 MC_CAM_REFERENCE_METHOD	LIST_WRITE _BACK	凸轮轴位置恢复对象 (AdvOutputName, Pr. Restore. PositionRestorationObject) 设置为“0: 1周期当前位置恢复 (RestorePositionPerCycle)”、“2: 凸轮指令当前位置恢复 (RestoreCamSetPosition)”时, 设置用于恢复的凸轮基准位置的设置方法。 • 0: 上次值(PreviousPosition) • 1: 凸轮基准位置初始设定值 (InitialReferenceSetPosition) • 2: 凸轮指令当前位置 (CamSetPosition)
InitialReferenceSetPosition	凸轮基准位置初始设定值	启动时	LREAL	LIST_WRITE _BACK	凸轮基准位置设置方法 (AdvOutputName, Pr. Restore. PositionRestorationMethod) 设置为“1: 凸轮基准位置初始设定值 (InitialReferenceSetPosition)”时, 以输出轴的轴信息 (AdvOutputName, Axis) 中设置的轴的单位设置凸轮基准位置的初始设定值。 • -10000000000.0~10000000000.0
PositionPerCycleMethod	1周期当前位置设置方法	启动时	INT (MC_CAM_CYCLE_METHOD) ☞ 1462页 MC_CAM_CYCLE_METHOD	LIST_WRITE _BACK	凸轮轴位置恢复对象 (AdvOutputName, Pr. Restore. PositionRestorationObject) 设置为“1: 凸轮基准位置恢复 (RestorePositionPerCycle)”、“2: 凸轮指令当前位置恢复 (RestoreCamSetPosition)”时, 设置用于恢复的1周期当前位置的设置方法。 • 0: 上次值(PreviousPosition) • 1: 1周期当前位置初始设定值 (InitialPositionPerCycle) • 2: 主轴齿轮后1周期当前位置 (MasterGearPositionPerCycle) • 3: 辅助轴齿轮后1周期当前位置 (AuxGearPositionPerCycle)
InitialPositionPerCycle	1周期当前位置初始设定值	启动时	LREAL	LIST_WRITE _BACK	根据凸轮轴位置恢复对象 (AdvOutputName, Pr. Restore. PositionRestorationObject) 的设置, 设置值。 为MCv_AdvancedSync(高级同步控制)的主轴 (Master) 中设置的输入轴的单位。 在“0.0~(1周期长度 - 0.00001)”的范围内设置。 • 0.0~2147483647.0
RestorationAllowablePosition	凸轮指令当前位置恢复允许移动量设定值	启动时	DWORD(UDINT)	LIST_WRITE _BACK	凸轮轴位置恢复对象 (AdvOutputName, Pr. Restore. PositionRestorationObject) 设置为“2: 凸轮指令当前位置恢复”时, 以输出轴的位置指令值的单位设置恢复的凸轮指令当前位置和指令当前位置的差的允许值。 若本设定值过大, 同步启动时可能会引起急剧运行。 • 0.0~2147483647.0[pulse]

44.2 枚举符一览

各种参数及监视数据、运动控制FB中使用的枚举符一览如下所示。

ENUM枚举符

对于各种参数及监视数据、运动控制FB中使用的枚举型的常数，实际上使用INT型的值。
在工程工具中，可以使用“枚举型名称_枚举符名称”的INT型全局标签。

例

在工程工具中使用MC_BUFFER_MODE型的枚举符“mcBuffered”的情况下

- 以INT型全局标签使用时：设置“MC_BUFFER_MODE_mcBuffered”。

MC_BUFFER_MODE[M]

枚举符	设定值	内容
mcAborting	0	Aborting
mcBuffered	1	Buffered
mcBlendingLow	2	BlendingLow
mcBlendingPrevious	3	BlendingPrevious
mcBlendingNext	4	BlendingNext
mcBlendingHigh	5	BlendingHigh

MC_EXECUTION_MODE[M]

枚举符	设定值	内容
mcImmediately	0	立即执行
mcQueued	1	等待完成后执行
mcSpeculatively	3	推测执行

MC_COMBINE_MODE[M]

枚举符	设定值	内容
mcAddAxes	0	对2个输入轴的位置进行加法运算
mcSubAxes	1	对2个输入轴的位置进行减法运算

MC_DIRECTION[M]

枚举符	设定值	内容
mcPositiveDirection	1	正方向
mcNegativeDirection	2	负方向
mcShortestWay	3	最短路径
mcCurrentDirection	4	当前方向

MC_SOURCE[M]

枚举符	设定值	内容
mcSetValue	1	指令当前值
mcActualValue	2	反馈值
mcLatestSetValue	101	最新指令当前值
mcLatestActualValue	102	最新反馈值

MC_CIRC_MODE [M]

列举符	设定值	内容
mcBorder	0	边界点指定
mcCenter	1	中心点指定
mcRadius	2	半径指定

MC_CIRC_PATHCHOICE [M]

列举符	设定值	内容
mcCW	0	CW
mcCCW	1	CCW
mcShortWay	2	就近
mcLongWay	3	就远
mcCWLongWay	4	CW就远
mcCCWLongWay	5	CCW就远

MC_START_MODE [M]

列举符	设定值	内容
mcImmediate	0	即时
mcAbsolute	1	绝对

MC_AXIS_STATUS [M]

列举符	设定值	内容
Invalid	-1	轴变量未初始化/轴参数异常
Disabled	0	轴无效
ErrorStop	1	错误停止中
Stopping	2	减速停止中
Homing	3	原点复位中
Standstill	4	待机中
DiscreteMotion	5	定位运行中
ContinuousMotion	6	连续动作运行中
SynchronizedMotion	7	同步运行中

MC_AXES_GROUP_STATUS [M]

列举符	设定值	内容
Invalid	-1	轴组变量未初始化/轴组参数异常
GroupDisabled	0	轴组无效
GroupErrorStop	1	错误停止中
GroupStopping	2	减速停止中
GroupStandby	4	待机中
GroupMoving	5	动作中

MC_DRIVE_MODE[M]

列举符	设定值	内容
NoModeChange	0	不更改
hm	6	hm(原点复位模式)
csp	8	csp(循环位置模式)
csv	9	csv(循环速度模式)
cst	10	cst(循环转矩模式)
ct	-104	ct(挡块控制模式)

MC_VELOCITY_LIMIT_MODE[M]

列举符	设定值	内容
Ignore	0	忽略
Truncate	2	舍去
ImmediateStop	3	立即停止
ClampWithoutRamp	4	限定(无减速时斜率)

MC_INTERPOLATE_SPEED_MODE[M]

列举符	设定值	内容
VectorSpeed	0	合成速度
LongAxisSpeed	1	长轴速度
ReferenceAxisSpeed	2	基准轴速度

MC_RECORD_MODE[M]

列举符	设定值	内容
OneShot	0	单发模式
RecordCount	1	指定次数模式
RingBuffer	2	环形缓冲模式

MC_CAM_CURVE_TYPE[M]

列举符	设定值	内容
ConstantSpeed	0	匀速
ConstantAcceleration	1	恒定加速度
DistortedTrapezoid	2	变形梯形
DistortedSine	3	变形正弦
DistortedConstantSpeed	4	变形匀速
Cycloid	5	摆线
FifthCurve	6	5次
Trapeclloid	7	梯弦型
ReverseTrapeclloid	8	反梯弦型
DoubleHypotenuse	9	双弦
ReverseDoubleHypotenuse	10	反双弦
SingleHypotenuse	11	单弦
FifthCurve_SpeedDesignation	12	5次(调整)

MC_AXIS_TYPE[M]

列举符	设定值	内容
DriveAxis	0	实际驱动轴
EncoderAxis	2	实际编码器轴
VirtualDriveAxis	3	虚拟驱动轴
VirtualEncoderAxis	4	虚拟编码器轴
VirtualLinkAxis	5	虚拟连接轴

MC_ENCODER_AXIS_TYPE[M]

列举符	设定值	内容
Drive	1	经由驱动器模块

MC_ABS_SYSTEM[M]

列举符	设定值	内容
ABSDisabled	0	不使用绝对位置系统
Enabled	1	使用绝对位置系统
Auto	-1	自动设置(从连接设备获取)

MC_DRIVE_STATE[M]

列举符	设定值	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[M]

列举符	设定值	内容
Ignore	0	忽略
ImmediateStop	1	立即停止
KeepCurrentAcc	2	继续执行当前的加减速速度
AlternativeAcc	3	替代加减速速度
ServoOffAfterImmediateStop	4	立即停止后伺服OFF
ServoOffAfterDecelStop	5	减速停止后伺服OFF

MC_STOP_CURVE_MODE[M]

列举符	设定值	内容
OverrideCurve	1	重新创建减速曲线

MC_POS_SOURCE[M]

列举符	设定值	内容
Invalid	-1	无效
SetPosition	1	指令当前位置
FeedMachinePosition	3	进给机械位置

MC_ACC_ZERO_MODE[M]

列举符	设定值	内容
ACCError	-1	错误(不启动)
MaximumAcceleration	1	最大加减速

MC_SIGNAL_LOGIC[M]

列举符	设定值	内容
HighLevel	0	TRUE时检测
LowLevel	1	FALSE时检测
RisingEdge	2	FALSE→TRUE(上升沿)时检测
FallingEdge	3	TRUE→FALSE(下降沿)时检测
BothEdges	4	上升沿/下降沿时检测

MC_ERROR_CLASS[M]

列举符	设定值	内容
_None	0	无
Warning	1	警告
MinorError	2	轻度异常
ModerateError	3	中度异常
MajorError	4	重度异常

MC_EXECUTION_STATE[M]

列举符	设定值	内容
Ready	0	执行请求等待
Executing	1	执行中
Done	2	执行完成
Error_	-1	错误发生

MC_POS_RESTITUTION_STATUS[M]

列举符	设定值	内容
NotExecute	0	未实施
WaitingRequest	1	恢复请求等待
RestoredInIncSystem	2	增量系统中恢复完成
RestoredInAbsSystemUnHomed	3	绝对位置系统中恢复完成(原点复位未完)
RestoredInAbsSystem	4	绝对位置系统中恢复完成

MC_AXES_GROUP_TYPE

列举符	设定值	内容
Standard	0	标准

MC_OPERATION_PATTERN

列举符	设定值	内容
PositioningComplete	0	定位结束
ContinuousPositioning	1	连续定位
ContinuousBlendingLow	2	连续轨迹 (BlendingLow)
ContinuousBlendingPrevious	3	连续轨迹 (BlendingPrevious)
ContinuousBlendingNext	4	连续轨迹 (BlendingNext)
ContinuousBlendingHigh	5	连续轨迹 (BlendingHigh)

MC_CONTROL_METHOD

列举符	设定值	内容
NOP	128	NOP
JUMP	130	JUMP
LOOP	131	LOOP
LEND	132	LEND
LinearAbsolute	257	绝对值直线插补
LinearRelative	258	相对值直线插补
CircularAbsolute	259	绝对值圆弧插补
CircularRelative	260	相对值圆弧插补
HelicalAbsolute	261	绝对值螺旋插补
HelicalRelative	262	相对值螺旋插补

MC_MCODE_OUTPUT_OVERRIDE

列举符	设定值	内容
UseFbOptions	0	用FB选项
WithMode	1	WITH模式
AfterMode	2	AFTER模式

MC_STEP_MODE

列举符	设定值	内容
Ignored	0	忽略
StepPerDec	1	减速单位步
StepPerPositioningData	2	定位数据单位步

MC_POS_RESTORATION_BASE

列举符	设定值	内容
BackupPosition	0	备份位置
HomePosition	1	原点
Auto	-1	自动判定

MC_POS_RESTORATION_REQUEST

列举符	设定值	内容
NoRequest	0	无处理
RestorationRequest	1	当前位置恢复执行
RestorationRequest_EraseAbsPos	2	绝对位置丢失情况下当前位置恢复执行

MC_INPUT_DIRECTION

列举符	设定值	内容
NoDirectionRestriction	0	无移动方向限制
mcPositiveDirection	1	当前位置仅允许增加方向
mcNegativeDirection	2	当前位置仅允许减少方向

MC_CLUTCH_METHOD

列举符	设定值	内容
NoClutch	0	无效
ClutchCommand	1	离合器指令
ClutchLeadingEdge	2	离合器指令上升沿
ClutchTrailingEdge	3	离合器指令下降沿
ClutchAddress	4	地址模式
ClutchSignal	15	输入输出数据指定

MC_CLUTCH_REFERENCE

列举符	设定值	内容
GearFrontPosition	0	齿轮前当前位置
GearPositionPerCycle	1	齿轮后1周期当前位置

MC_CLUTCH_SMOOTHING_METHOD

列举符	设定值	内容
ClutchSmoothingDisabled	0	直接
TimeConstantExponent	1	时间常数方式(指数)
TimeConstantLinear	2	时间常数方式(直线)
SlippageExponent	3	滑动量方式(指数)
SlippageLinear	4	滑动量方式(直线)
SlippageLinearFollow	5	滑动量方式(直线: 输入量跟踪)

MC_LENGTH_PER_CYCLE_CHANGE

列举符	设定值	内容
LengthChangeInvalid	0	无效
LengthChangeValid	1	有效

MC_SYNC_CHANGE_COMMAND

列举符	设定值	内容
ReferenceSetPositionMovement	0	凸轮基准位置移动
ChangeCurrentPositionPerCycle	1	1周期当前位置更改
ChangeMasterGearPositionPerCycle	2	主轴齿轮后1周期当前位置更改
ChangeAuxGearPositionPerCycle	3	辅助轴齿轮后1周期当前位置更改
PositionPerCycleMovement	4	1周期当前位置移动

MC_GEAR_RESTORE_METHOD

列举符	设定值	内容
PreviousPosition	0	上次值
InitialGearPositionPerCycle	1	齿轮后1周期当前位置初始设定值
CalculateFromInputAxis	2	通过输入轴计算

MC_CAM_RESTORE_METHOD

列举符	设定值	内容
RestorePositionPerCycle	0	1周期当前位置恢复
RestoreReferenceSetPosition	1	凸轮基准位置恢复
RestoreCamSetPosition	2	凸轮指令当前位置恢复

MC_CAM_REFERENCE_METHOD

列举符	设定值	内容
PreviousPosition	0	上次值
InitialReferenceSetPosition	1	凸轮基准位置初始设定值
CamSetPosition	2	指令当前位置

MC_CAM_CYCLE_METHOD

列举符	设定值	内容
PreviousPosition	0	上次值
InitialPositionPerCycle	1	1周期当前位置初始设定值
MasterGearPositionPerCycle	2	主轴齿轮后1周期当前位置
AuxGearPositionPerCycle	3	辅助轴齿轮后1周期当前位置

45.1 轴组有效

MC_GroupEnable

指定的轴组状态(AxesGroupName.Md.GroupStatus)从“0: 轴组无效(GroupDisabled)”转变为“4: 待机中(GroupStandby)”，并将轴组置为有效。

梯形图	FBD/LD	ST
		<pre>MC_GroupEnable(AxesGroup:= ?AXES_GROUP_REF?, Execute:= ?BOOL?, Done=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
AxesGroup	轴组信息	AXES_GROUP_REF	启动时	—	不能省略	设置轴组。 关于使用的变量(AxesGroupName.AxesGroupRef.)，请参阅下述内容。 ☞ 1408页 AxesGroupName.AxesGroupRef. (轴组信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_GroupEnable(轴组有效)。

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	转变为“4: 待机中(GroupStandby)”状态时，将变为TRUE。
Busy	执行中	BOOL	FALSE	MC_GroupEnable(轴组有效)执行中将变为TRUE。 轴组状态(AxesGroupName.Md.GroupStatus)转换为“4: 待机中(GroupStandby)”状态时，将变为FALSE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 ☞ 所使用的控制器的用户手册

功能

- AXES_GROUP_REF结构体的轴组状态 (AxesGroupName.Md.GroupStatus)从“0: 轴组无效(GroupDisabled)”转变为“4: 待机中(GroupStandby)”。
- 执行指令(Execute)变为TRUE时将执行MC_GroupEnable(轴组有效), 正常开始处理时执行中(Busy)将变为TRUE。
- 处理正常完成时, 执行完成(Done)将变为TRUE, 执行中(Busy)将变为FALSE。
- 在MC_GroupEnable(轴组有效)内发生了异常的情况下, 错误(Error)将变为TRUE, 并将错误代码存储到错误代码(ErrorID)中。关于错误代码的详细内容, 请参阅下述手册。

📖所使用的控制器的用户手册

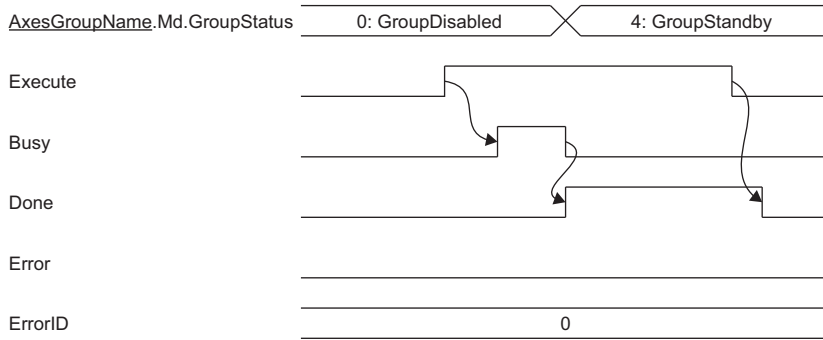
- 关于轴组的状态转换的详细内容, 请参阅下述手册。

📖所使用的控制器的用户手册

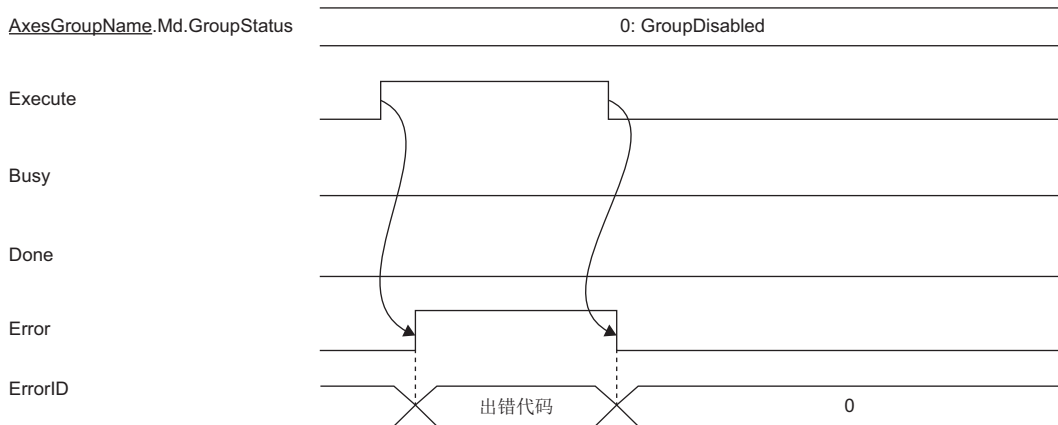
- MC_GroupEnable(轴组有效)在所有的配置轴的轴状态 (AxisName.Md.AxisStatus)为“4: 待机中(Standstill)”或“0: 轴无效(Disabled)”状态的情况下才能执行。
- 轴组状态 (AxesGroupName.Md.GroupStatus)变为“4: 待机中(GroupStandby)”时, 配置轴的轴组使用中 (AxisName.Md.UseInGroup)变为TRUE。
- 轴组使用中 (AxisName.Md.UseInGroup)设置了包含TRUE的配置轴的其他轴组并执行了MC_GroupEnable(轴组有效)的情况下, 将发生轴组配置轴使用中(错误代码: 1BB8H)。
- 配置轴中的任意一个发生了错误的情况下, 轴组状态 (AxesGroupName.Md.GroupStatus)变为“1: 错误停止中(GroupErrorStop)”。
- 对已经处于轴组有效的轴组执行了MC_GroupEnable(轴组有效)的情况下, 执行完成(Done)将变为TRUE并结束。

■时序图

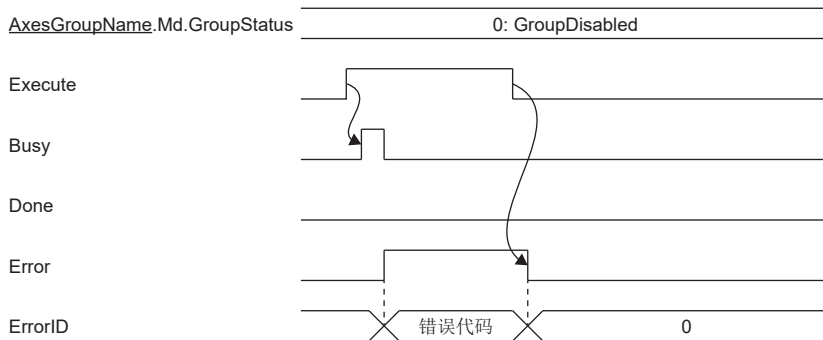
- 正常完成的情况下



- 输入输出变量异常的情况下



- 异常完成的情况下



注意事项

- 在执行MC_GroupEnable (轴组有效)之前，需要预先设置AXES_GROUP_REF结构体的轴组No. (GroupNo)。
- 轴组中设置的所有的配置轴的轴状态 (AxisName.Md.AxisStatus)为“4: 待机中(Standstill)”或“0: 轴无效(Disabled)”的情况下才能执行。
- 轴组变量初始化时发生了参数异常的情况下，将变为超出参数范围(轴组) (错误代码: 1A61H)。此时，轴组状态 (AxesGroupName.Md.GroupStatus)变为“-1: 轴组变量未初始化/轴组参数异常(Invalid)”。对于变为了“-1: 轴组变量未初始化/轴组参数异常(Invalid)”的轴组，将不执行监视数据等的刷新，通过用户程序进行指定时将变为超出轴组编号范围 (错误代码: 1B84H)。

程序示例

将轴组有效指令(bGroupEnable)置为TRUE, 并将轴组1(AxesGroup001)置为有效的程序示例如下所示。

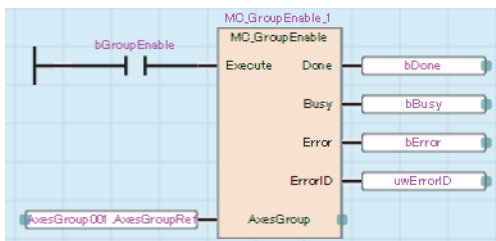
■轴组

轴组No.	标签名	数据类型	注释
1	AxesGroup001	AXES_GROUP_REF	轴组1

■使用的标签

标签名	数据类型	注释
MC_GroupEnable_1	MC_GroupEnable	轴组有效FB
bGroupEnable	位	轴组有效指令
bDone	位	执行完成
bBusy	位	执行中
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序



■ST程序

```
MC_GroupEnable_1(  
    AxesGroup:= AxesGroup001.AxesGroupRef ,  
    Execute:= bGroupEnable ,  
    Done=> bDone ,  
    Busy=> bBusy ,  
    Error=> bError ,  
    ErrorID=> uwErrorID  
);
```

要点

MC_GroupEnable(轴组有效)与进行插补控制的运动控制FB组合使用。关于与插补控制的程序示例, 请参阅下述章节。

- 1656页 绝对值直线插补控制
- 1670页 相对值直线插补控制
- 1683页 绝对值圆弧插补控制
- 1699页 相对值圆弧插补控制

45.2 轴组无效

MC_GroupDisable

指定的轴组状态 (AxesGroupName.Md.GroupStatus) 转换为“0: 轴组无效(GroupDisabled)”，并将轴组置为无效。

梯形图	FBD/LD	ST
		<pre>MC_GroupDisable(AxesGroup:= ?AXES_GROUP_REF? , Execute:= ?BOOL? , Done=> ?BOOL? , Busy=> ?BOOL? , Error=> ?BOOL? , ErrorID=> ?WORD?);</pre>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
AxesGroup	轴组信息	AXES_GROUP_REF	启动时	—	不能省略	设置轴组。 关于使用的变量 (<u>AxesGroupName</u> .AxesGroupRef.)，请参阅下述内容。 ☞ 1408页 <u>AxesGroupName</u> .AxesGroupRef. (轴组信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_GroupDisable (轴组无效)。

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	转变为“0: 轴组无效(GroupDisabled)”状态时，将变为TRUE。
Busy	执行中	BOOL	FALSE	MC_GroupDisable (轴组无效) 执行中将变为TRUE。 轴组状态 (<u>AxesGroupName</u> .Md.GroupStatus) 转换为“0: 轴组无效(GroupDisabled)”状态时，将变为FALSE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 ☞ 所使用的控制器的用户手册

功能

- AXES_GROUP_REF结构体的轴组状态(AxesGroupName. Md. GroupStatus)转换为“0: 轴组无效(GroupDisabled)”。
- 在执行指令(Execute)为TRUE时执行MC_GroupDisable(轴组无效), 正常开始处理时执行中(Busy)将变为TRUE。
- 处理正常完成时, 执行完成(Done)将变为TRUE, 执行中(Busy)将变为FALSE。
- 在MC_GroupDisable(轴组无效)内发生了异常的情况下, 错误(Error)将变为TRUE, 并将错误代码存储到错误代码(ErrorID)中。关于错误代码的详细内容, 请参阅下述手册。

📖所使用的控制器的用户手册

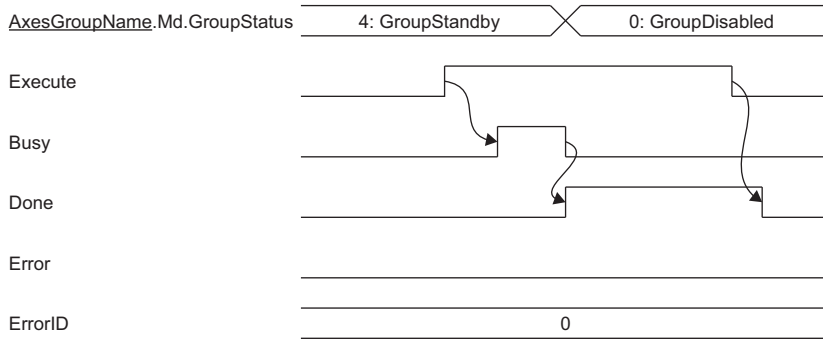
- 关于轴组的状态转换的详细内容, 请参阅下述手册。

📖所使用的控制器的用户手册

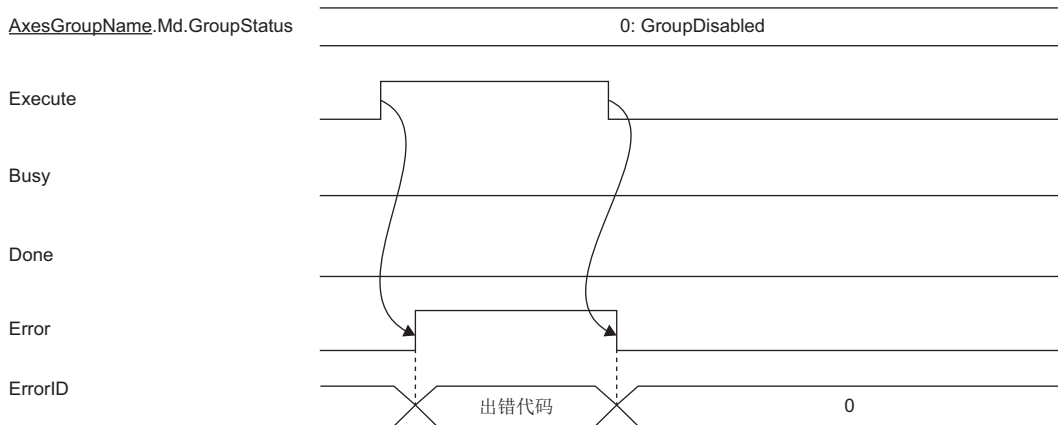
- MC_GroupDisable(轴组无效)在轴组状态(AxesGroupName. Md. GroupStatus)为“4: 待机中(GroupStandby)”或“1: 错误停止中(GroupErrorStop)”的情况下才能执行。轴组状态为“1: 错误停止中(GroupErrorStop)”, 且在轴组动作中执行了MC_GroupDisable(轴组无效)的情况下, 停止完成后将转变至“0: 轴组无效(GroupDisabled)”。轴组状态(AxesGroupName. Md. GroupStatus)为“4: 待机中(GroupStandby)”或“1: 错误停止中(GroupErrorStop)”以外时执行了MC_GroupDisable(轴组无效)的情况下, 将变为轴组状态不正确(轴组无效时)(错误代码: 1AEFH), 且执行中的FB将停止。
- 轴组状态(AxesGroupName. Md. GroupStatus)为“0: 轴组无效(GroupDisabled)”时, 配置轴的轴组使用中(AxisName. Md. UseInGroup)变为FALSE。
- 对处于轴组无效的轴组执行了MC_GroupDisable(轴组无效)的情况下, 执行完成(Done)将变为TRUE并结束。

■时序图

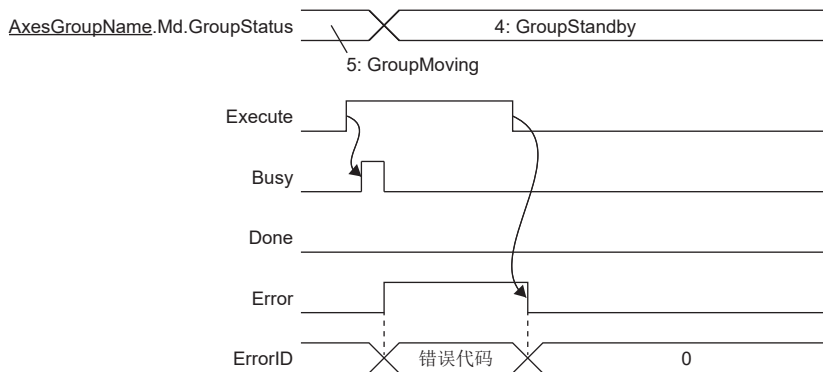
- 正常完成的情况下



- 输入输出变量异常的情况下



- 异常完成的情况下



注意事项

- 在执行MC_GroupDisable(轴组无效)之前, 需要预先设置AXES_GROUP_REF结构体的轴组No. (GroupNo)。
- 轴组变量初始化时发生了参数异常的情况下, 将变为超出参数范围(轴组) (错误代码: 1A61H)。此时, 轴组状态 (AxesGroupName.Md.GroupStatus)变为“-1: 轴组变量未初始化/轴组参数异常(Invalid)”。对于变为了“-1: 轴组变量未初始化/轴组参数异常(Invalid)”的轴组, 将不执行监视数据等的刷新, 通过用户程序进行指定时将变为超出轴组编号范围 (错误代码: 1B84H)。

程序示例

将轴组无效指令(bGroupDisable)置为TRUE，并将轴组1(AxesGroup001)置为无效的程序示例如下所示。

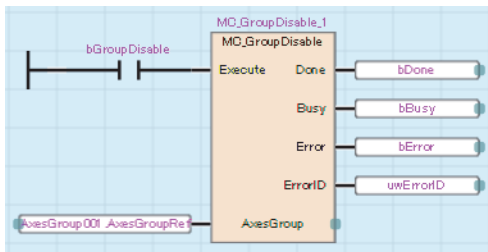
■轴组

轴组No.	标签名	数据类型	注释
1	AxesGroup001	AXES_GROUP_REF	轴组1

■使用的标签

标签名	数据类型	注释
MC_GroupDisable_1	MC_GroupDisable	轴组无效FB
bGroupDisable	位	轴组无效指令
bDone	位	执行完成
bBusy	位	执行中
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序



■ST程序

```
MC_GroupDisable_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bGroupDisable ,
    Done=> bDone ,
    Busy=> bBusy ,
    Error=> bError ,
    ErrorID=> uwErrorID
);
```

要点

MC_GroupDisable(轴组无效)与进行插补控制的运动控制FB组合使用。关于与插补控制的程序示例，请参阅下述章节。

- ☞ 1656页 绝对值直线插补控制
- ☞ 1670页 相对值直线插补控制
- ☞ 1683页 绝对值圆弧插补控制
- ☞ 1699页 相对值圆弧插补控制

45.3 允许运行

MC_Power

将指定的轴切换为允许运行状态。

梯形图	FBD/LD	ST
		<pre>MC_Power(Axis:= ?AXIS_REF?, Enable:= ?BOOL?, ServoON:= ?BOOL?, Status=> ?BOOL?, ReadyStatus=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	始终	—	不能省略	设置轴。 关于使用的变量 (<u>AxisName.AxisRef.</u>)，请参阅下述内容。 1387页 <u>AxisName.AxisRef.</u> (轴信息) * 将“可编程控制器就绪 (MotionSystem.Cd.SequenceReady)”切换为ON后，首次调用FB时轴信息将确定。

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Enable	有效	BOOL	始终	TRUE、FALSE	FALSE	设置为TRUE时轴控制将变为有效，且变为允许运行状态。 设置为FALSE时轴控制将变为无效，且解除允许运行状态。
ServoON	伺服ON请求	BOOL	始终	TRUE、FALSE	FALSE	TRUE时请求轴的伺服ON。

■输出变量

输出变量	名称	数据类型	初始值	内容
Status	允许运行	BOOL	FALSE	变为了允许运行状态时，将变为TRUE。
ReadyStatus	就绪ON状态	BOOL	FALSE	驱动器就绪变为ON时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_Power (允许运行)时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 所使用的控制器的用户手册

功能

- 对指定的轴的信息进行初始化，并切换为允许运行状态。
- 将有效 (Enable) 及伺服ON请求 (ServoON) 的输入置为TRUE时，可以将指定的轴切换为允许运行状态。
- 开始处理时执行中 (Busy) 将变为TRUE。
- 驱动器模块变为允许运行状态时允许运行 (Status) 将变为TRUE。
- 在MC_Power (允许运行) 内发生了异常的情况下，错误 (Error) 将变为TRUE，并将错误代码存储到错误代码 (ErrorID) 中。关于错误代码的详细内容，请参阅下述手册。

📖 所使用的控制器的用户手册

- AXIS_MONI 结构体的轴状态 (AxisName.Md.AxisStatus) 从 “0: 轴无效 (Disabled)” 转变为 “4: 待机中 (Standstill)” 的状态。关于轴的状态转变，请参阅下述手册。

📖 所使用的控制器的用户手册

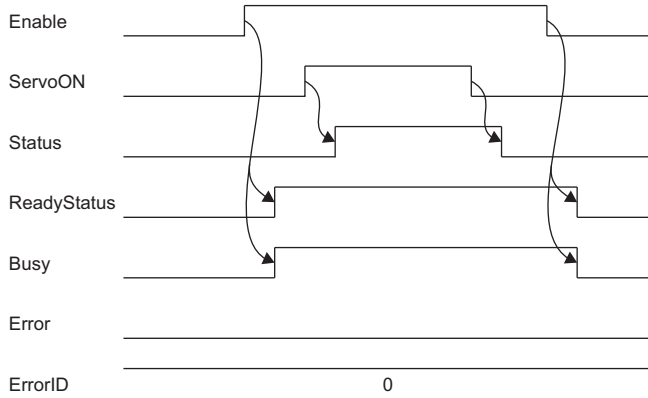
- 根据有效 (Enable) 及伺服ON请求 (ServoON) 的输入，将按下述方式切换轴的伺服ON/OFF状态、驱动器模块的状态。

输入变量		输出变量		伺服ON/OFF状态	驱动器状态 (AxisName.Md.Driver_State)
有效 (Enable)	伺服ON请求 (ServoON)	就绪ON状态 (ReadyStatus)	允许运行 (Status)		
TRUE	TRUE	TRUE	TRUE	伺服ON	6: Operation Enable (OperationEnable)
	FALSE	TRUE	FALSE	伺服OFF	5: Switched On (SwitchedOn)
FALSE	TRUE	FALSE	FALSE	伺服OFF	3: Switched On Disabled (SwitchOnDisabled)
	FALSE	FALSE	FALSE	伺服OFF	3: Switched On Disabled (SwitchOnDisabled)

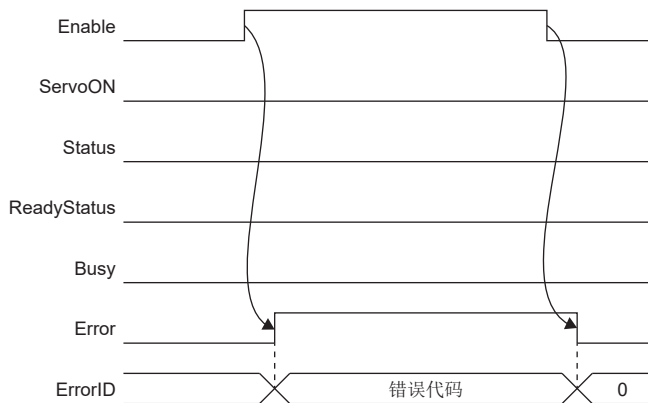
- 伺服OFF中实轴受外力而旋转的情况下，将进行跟踪处理。
- 伺服ON/OFF的控制操作与控制模式无关。伺服OFF时的控制模式取决于驱动器模块的规格。
- 驱动器模块错误发生中，MC_Power (允许运行) 将发送到驱动器模块中，因此无需再次将有效 (Enable) 及伺服ON请求 (ServoON) 从FALSE置为TRUE。

■时序图

- 正常完成的情况下



- 异常完成的情况下



注意事项

- 在执行MC_Power(允许运行)之前，需要预先设置AXIS_REF结构体的轴No. (AxisNo)。
- MC_Power(允许运行)将“可编程控制器就绪(MotionSystem.Cd.SequenceReady)”切换为ON后，首次调用FB时轴信息将确定。有效(Enable)为FALSE时即使更改轴信息也不会反映。
- 请勿对同一轴配置2个及以上的MC_Power(允许运行)。配置了2个及以上的情况下将无法保证动作。
- 内置运动准备完成(X420)处于OFF中时，保持程序停止时的伺服ON/OFF状态。

程序示例

将伺服ON请求指令 (bServoON) 置为TRUE，并将轴1 (Axis0001) 置为允许运行状态的程序示例如下所示。

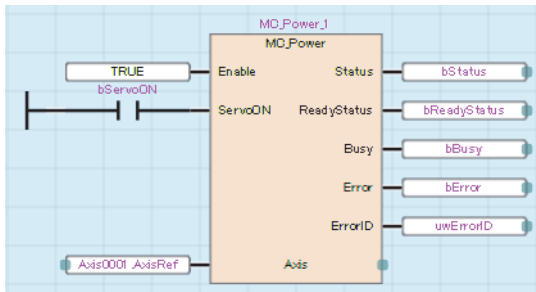
■轴

轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

标签名	数据类型	注释
MC_Power_1	MC_Power	允许运行FB
bServoON	位	伺服ON请求指令
bStatus	位	允许运行
bReadyStatus	位	就绪ON状态
bBusy	位	执行中
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序



■ST程序

```

MC_Power_1(
    Axis:= Axis0001.AxisRef ,
    Enable:= TRUE ,
    ServoON:= bServoON ,
    Status=> bStatus ,
    ReadyStatus=> bReadyStatus ,
    Busy=> bBusy ,
    Error=> bError ,
    ErrorID=> uwErrorID
);
    
```

45.4 当前位置更改

MC_SetPosition

更改指定的轴的当前位置(指令位置、反馈位置)。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName, AxisRef.), 请参阅下述内容。 ☞ 1387页 AxisName, AxisRef. (轴信息)


■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_SetPosition(当前位置更改)。
Position	目标位置	LREAL	启动时	-10000000000.0~10000000000.0	0.0	设置要更改的目标位置的值。 根据相对位置选择(Relative)的设置, 设定值有所不同。 关于详细内容, 请参阅下述章节。 ☞ 1476页 目标位置(Position)
Relative	相对位置选择	BOOL	启动时	TRUE、FALSE	FALSE	设置是通过相对距离更改当前位置, 还是通过绝对位置更改当前位置。 • FALSE: 通过绝对位置进行当前位置更改 • TRUE: 通过相对距离进行当前位置更改
ExecutionMode	启动模式	INT (MC_EXECUTION_MODE)	启动时	1、3	3	设置用于执行MC_SetPosition(当前位置更改)的执行方法。 • 1: 等待完成后执行(mcQueued)*1 • 3: 推测执行(mcSpeculatively)*2 关于详细内容, 请参阅下述章节。 ☞ 1477页 启动模式(ExecutionMode)
Options	选项	DWORD (HEX)	启动时	00000000H~00010000H	00000000H	将MC_SetPosition(当前位置更改)的功能选项以位指定进行设置。 关于详细内容, 请参阅下述章节。 ☞ 1478页 选项(Options)

*1 前一个FB动作完成之后再执行。


*2 仅在指定轴中没有执行中的FB的情况下才能执行。

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	当前位置更改完成时, 将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_SetPosition(当前位置更改)时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MC_SetPosition(当前位置更改)的执行中断时, 将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时, 将返回错误代码。 关于错误代码的详细内容, 请参阅下述手册。  所使用的控制器的用户手册
CancelAccepted	取消受理	BOOL	FALSE	MC_SetPosition(当前位置更改)受理了取消时将变为TRUE。

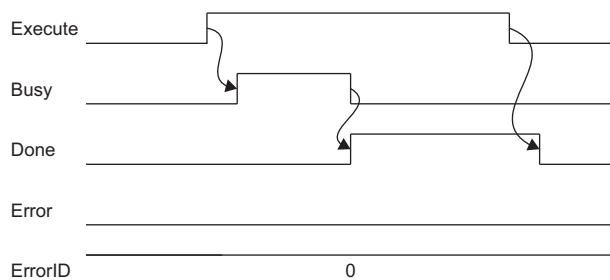
功能

- 更改指定的轴的当前位置。
- 相对位置选择(Relative)为TRUE的情况下, 更改为在当前位置加上目标位置(相对距离)后的位置。
- 相对位置选择(Relative)为FALSE的情况下, 更改为目标位置(绝对位置)。
- 执行指令(Execute)变为TRUE时将执行MC_SetPosition(当前位置更改), 正常开始处理时执行中(Busy)将变为TRUE。
- 处理完成, 更改当前位置时执行完成(Done)将变为TRUE。
- 在启动模式(ExecutionMode)中设置“1: 等待完成后执行(mcQueued)”, 启动MC_SetPosition(当前位置更改)后, 处理完成前轴状态(AxisName.Md.AxisStatus)变为“0: 轴无效(Disabled)”或“1: 错误停止中(ErrorStop)”的情况下, 将变为MC_SetPosition指令异常(错误代码: 1B9CH)。
- 在MC_SetPosition(当前位置更改)内发生了异常的情况下, 错误(Error)将变为TRUE, 并将错误代码存储到错误代码(ErrorID)中。关于错误代码的详细内容, 请参阅下述手册。

所使用的控制器的用户手册


■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图, 请参阅下述章节。

 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■目标位置(Position)

目标位置(Position)中可设置的值根据相对位置选择(Relative)为FALSE还是为TRUE而有所不同。

目标位置(Position)应在下述范围内进行设置。

但是, 即使在指定范围内, 当前位置更改后的指令当前位置超出软件行程限位范围的情况下, 将发生错误且不执行。

相对位置选择(Relative)	目标位置(Position)的设置范围
FALSE(通过绝对位置进行当前位置更改)	<ul style="list-style-type: none"> ■环形计数器无效时 -10000000000.0 ≤ 设定值 < 10000000000.0 ■环形计数器有效时 环形计数器下限值 ≤ 设定值 < 环形计数器上限值
TRUE(通过相对距离进行当前位置更改)	<ul style="list-style-type: none"> ■环形计数器无效时 -10000000000.0 ≤ 设定值 ≤ 10000000000.0 ■环形计数器有效时 -(环形计数器上限值 - 环形计数器下限值) / 2 ≤ 设定值 ≤ (环形计数器上限值 - 环形计数器下限值) / 2

■启动模式 (ExecutionMode)

轴动作中执行了MC_SetPosition(当前位置更改)的情况下,更改的时机根据设置的启动模式(ExecutionMode)而有所不同。启动模式(ExecutionMode)的时机如下所示。

设定值	内容
1: 等待完成后执行 (mcQueued)	<p>执行中的FB结束后执行。 通过执行指令(Execute)的上升沿检测,执行中(Busy)变为TRUE,轴状态(AxisName.Md.AxisStatus)变为“4: 待机中(Standstill)”之前待机。 轴状态(AxisName.Md.AxisStatus)变为“4: 待机中(Standstill)”时,开始当前位置更改动作。</p>
3: 推测执行 (mcSpeculatively)	<p>执行指令(Execute)的上升沿检测时,相应轴的轴状态(AxisName.Md.AxisStatus)为“4: 待机中(Standstill)”的情况下,可以执行当前位置更改。 相应轴的轴状态(AxisName.Md.AxisStatus)不为“4: 待机中(Standstill)”的情况下,或存在有执行中的FB的情况下,将变为MC_SetPosition指令异常(错误代码: 1B9CH),且不执行当前位置更改。</p> <p>AxisName.Md.AxisStatus 4: Standstill</p>

■选项 (Options)

将MC_SetPosition(当前位置更改)中使用的功能选项以位指定进行设置。

以位指定设置的内容如下所示。



*1 空余中应设置“0”。设置了“0”以外的情况下，将变为超出Options范围(错误代码: 1ABBH)。

位	名称	内容
16	FB启动后取消允许	设置MC_SetPosition(当前位置更改)启动后是否允许取消。 • 0: 不允许 • 1: 允许

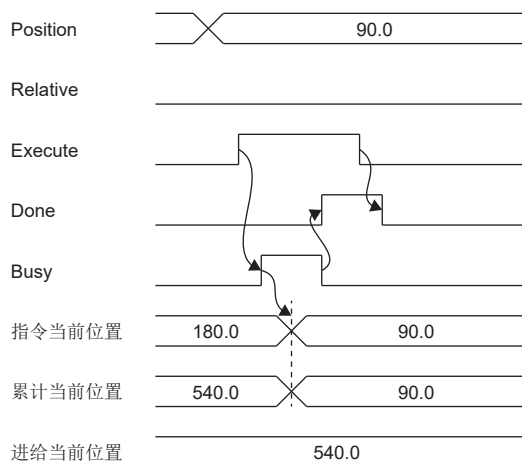
■动作概要

在轴处于停止状态下执行了MC_SetPosition(当前位置更改)的情况下，通过执行指令(Execute)的上升沿检测，开始当前位置更改动作。在当前位置更改中，将指令当前位置、累计当前位置更改为目标位置(Position)中设置的位置。对于目标位置(Position)，相对位置选择(Relative)为TRUE的情况下将变为“通过相对距离进行当前位置更改”，FALSE的情况下将变为“通过绝对位置进行当前位置更改”。

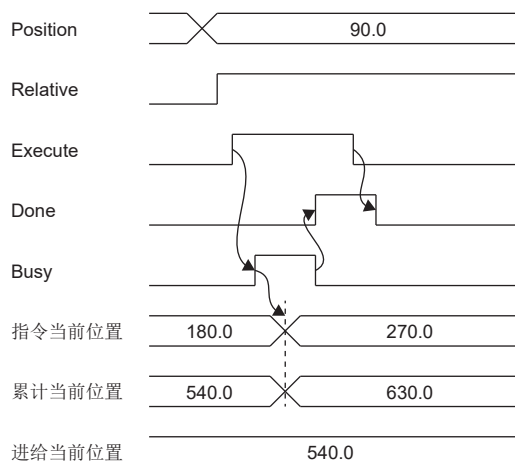
例

以环形计数器下限值“0.0”、环形计数器上限值“360.0”、指令当前位置“180.0”(累计当前位置“540.0”)停止时，执行了至“90.0”的当前位置更改的情况下

<相对位置选择(Relative)为FALSE(绝对位置)时>



<相对位置选择(Relative)为TRUE(相对距离)时>



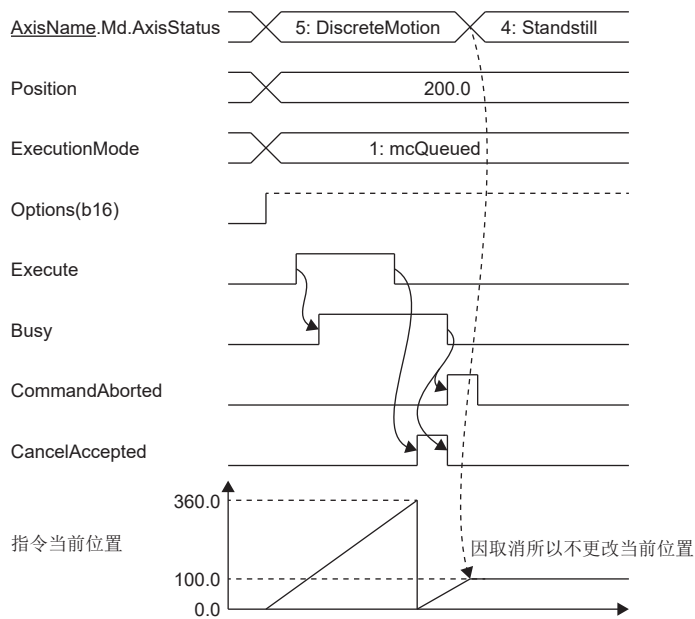
取消

MC_SetPosition(当前位置更改)执行后可以取消处于待机状态的当前位置更改。

- 将取消置为有效时，在将“FB启动后取消允许(选项(Options): 位16)”设置为“1: 允许”的状态下执行MC_SetPosition(当前位置更改)。
- 通过执行指令(Execute)的下降沿检测，开始取消。
- 取消的受理只有在输出引脚的执行中(Busy)为TRUE时才进行。
- MC_SetPosition(当前位置更改)受理取消时输出引脚的取消受理(CancelAccepted)将变为TRUE。
- 取消完成时输出引脚的执行中断(CommandAborted)将变为TRUE。
- 执行了取消的情况下，不更改指令当前位置。

例

将启动模式(ExecutionMode)以“1: 等待完成后执行(mcQueued)”启动后，轴状态(AxisName.Md.AxisStatus)将变为“4: 待机中(Standstill)”的切换待机中取消的情况下



注意事项

轴动作中更改了当前位置的情况下，虽然执行中的MC_SetPosition(当前位置更改)的目标位置不变化，但是由于当前位置被更改，因此将更改至目标位置为止的动作。

程序示例

将当前值更改指令 (bSetPosition) 置为TRUE，并将轴1 (Axis0001) 的当前值更改为“100.0”的程序示例如下所示。

■轴

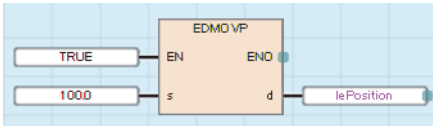
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

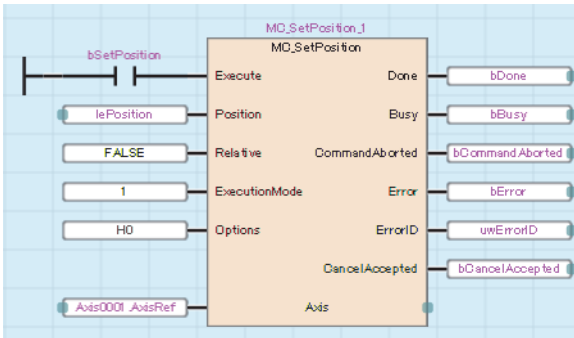
标签名	数据类型	注释
MC_SetPosition_1	MC_SetPosition	当前位置更改FB
bSetPosition	位	当前位置更改指令
lePosition	双精度实数	目标位置
bDone	位	准备就绪
bBusy	位	执行中
bCommandAborted	位	执行中断
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码
bCancelAccepted	位	取消受理

■FBD/LD程序

- 当前位置更改用数据设置



- 当前位置更改



■ST程序

//-----当前位置更改用数据设置-----

lePosition:= 100.0;

//-----当前位置更改-----

```
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
);
```

45.5 转矩限制值

MCv_SetTorqueLimit

执行转矩限制值的更改。

梯形图	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>

设置数据


■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName.AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName.AxisRef. (轴信息)

■输入变量


输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MCv_SetTorqueLimit (转矩限制值)。
PositiveValid	正方向转矩限制有效	BOOL	启动时/可重启	TRUE、FALSE	FALSE	设置将正方向的转矩限制设定值的更改是置为有效还是置为无效。 • FALSE: 将正方向的转矩限制置为无效。 • TRUE: 将正方向的转矩限制置为有效。
PositiveValue	正方向转矩限制值	LREAL	启动时/可重启	0.0~1000.0[%]	0.0	设置至正方向的转矩限制值的值。 正方向转矩限制有效(PositiveValid)为TRUE的情况下获取。
NegativeValid	负方向转矩限制有效	BOOL	启动时/可重启	TRUE、FALSE	FALSE	设置将负方向的转矩限制设定值的更改是置为有效还是置为无效。 • FALSE: 将负方向的转矩限制置为无效。 • TRUE: 将负方向的转矩限制置为有效。
NegativeValue	负方向转矩限制值	LREAL	启动时/可重启	0.0~1000.0[%]	0.0	设置至负方向的转矩限制值的值。 负方向转矩限制有效(NegativeValid)为TRUE的情况下获取。
ExecutionMode	启动模式	INT (MC_EXECUTION_MODE)	启动时	0、1、3	0	设置执行MCv_SetTorqueLimit (转矩限制值)的时机。 • 0: 立即执行 (mcImmediately) • 1: 等待完成后执行 (mcQueued) • 3: 推测执行 (mcSpeculatively) 关于详细内容，请参阅下述章节。 ☞ 1485页 启动模式 (ExecutionMode)
Options	选项	DWORD (HEX)	启动时	00000000H~00010000H	00000000H	将MCv_SetTorqueLimit (转矩限制值)的功能选项以位指定进行设置。 关于详细内容，请参阅下述章节。 ☞ 1485页 选项 (Options)

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	转矩限制值更改正常完成时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_SetTorqueLimit(转矩限制值)时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MCv_SetTorqueLimit(转矩限制值)的执行中断时，将变为TRUE。
CancelAccepted	取消受理	BOOL	FALSE	MCv_SetTorqueLimit(转矩限制值)受理了取消时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。  所使用的控制器的用户手册

功能

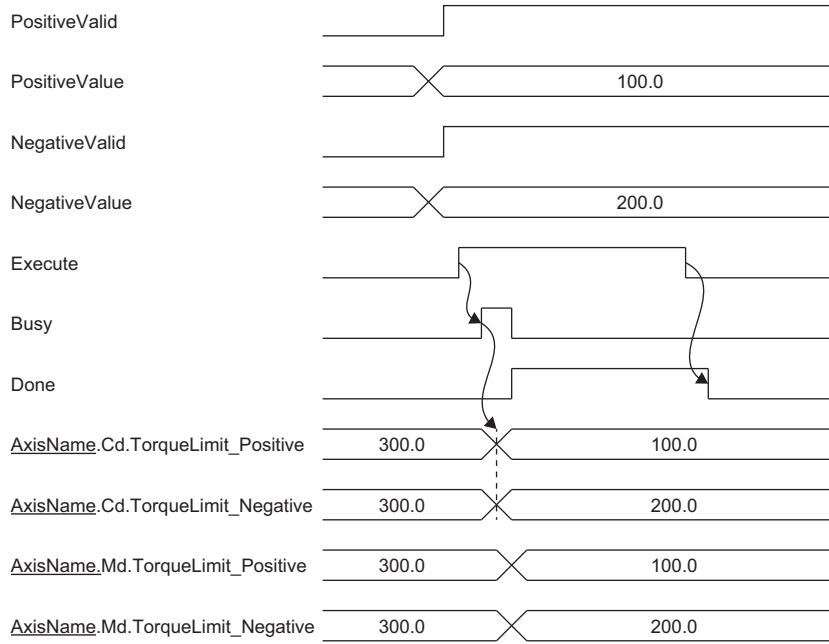
- 更改正方向转矩限制值或负方向转矩限制值。
- 正方向转矩限制有效(PositiveValid)为TRUE的情况下，将正方向转矩限制值更改为正方向转矩限制值(PositiveValue)的值。
- 负方向转矩限制有效(NegativeValid)为TRUE的情况下，将负方向转矩限制值更改为负方向转矩限制值(NegativeValue)的值。
- 在执行指令(Execute)为TRUE时执行MCv_SetTorqueLimit(转矩限制值)，正常开始处理时执行中(Busy)将变为TRUE。
- 对于更改转矩限制值的时机，通过启动模式(ExecutionMode)进行设置。
- 处理完成，更改转矩限制值时执行完成(Done)将变为TRUE。
- 在启动模式(ExecutionMode)中设置“1: 等待完成后执行(mcQueued)”，启动MCv_SetTorqueLimit(转矩限制值)后，处理完成前轴状态(AxisName.Md.AxisStatus)变为“0: 轴无效(Disabled)”或“1: 错误停止中(ErrorStop)”的情况下，将中断处理且执行中断(CommandAborted)将变为TRUE。
- 在MCv_SetTorqueLimit(转矩限制值)内发生了异常的情况下，错误(Error)将变为TRUE，并将错误代码存储到错误代码(ErrorID)中。关于错误代码的详细内容，请参阅下述手册。

所使用的控制器的用户手册

- 正方向转矩限制有效(PositiveValid)与负方向转矩限制有效(NegativeValid)均为FALSE的情况下，不进行转矩限制值的更改，执行完成(Done)将变为TRUE。

■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■启动模式 (ExecutionMode)

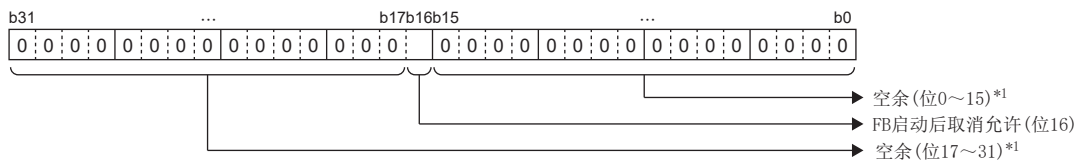
轴动作中执行了MCv_SetTorqueLimit (转矩限制值) 的情况下, 更改的时机根据设置的启动模式 (ExecutionMode) 而有所不同。启动模式 (ExecutionMode) 的时机如下所示。

设定值	内容
0: 立即执行 (mcImmediately)	与轴状态 (AxisName.Md.AxisStatus) 无关, 立即执行。 通过执行指令 (Execute) 的上升沿检测, 更改正方向转矩限制值 (AxisName.Cd.TorqueLimit_Positive)、负方向转矩限制值 (AxisName.Cd.TorqueLimit_Negative)。
3: 推测执行 (mcSpeculatively)	<p>执行指令 (Execute) 的上升沿检测时, 相应轴的轴状态 (AxisName.Md.AxisStatus) 为 “4: 待机中 (Standstill)” 的情况下, 执行转矩限制值更改。 相应轴的轴状态 (AxisName.Md.AxisStatus) 不为 “4: 待机中 (Standstill)” 的情况下, 或存在有执行中的FB的情况下, 将变为MCv_SetTorqueLimit指令异常 (错误代码: 1B9EH), 且不执行转矩限制值更改。</p> <p>The diagram illustrates the timing for mode 3. It shows that the torque limit registers are updated only when the axis is in the '4: Standstill' state and the 'MCv_SetTorqueLimit' function block is in the 'Execute' phase. The 'Busy' phase occurs after 'Execute' and before 'Done'. The torque limit registers are updated at the rising edge of the 'Execute' signal.</p>

■选项 (Options)

将MCv_SetTorqueLimit (转矩限制值) 中使用的功能选项以位指定进行设置。

以位指定设置的内容如下所示。

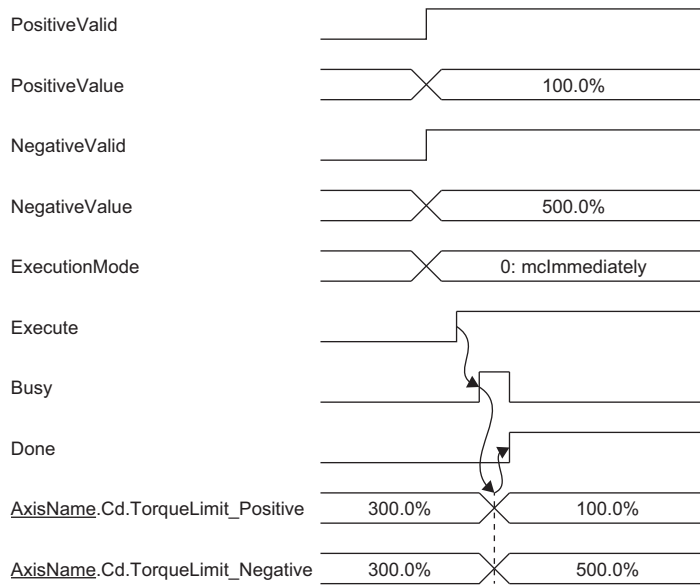


*1 空余中应设置 “0”。设置了 “0” 以外的情况下, 将变为超出Options范围 (错误代码: 1ABBH)。

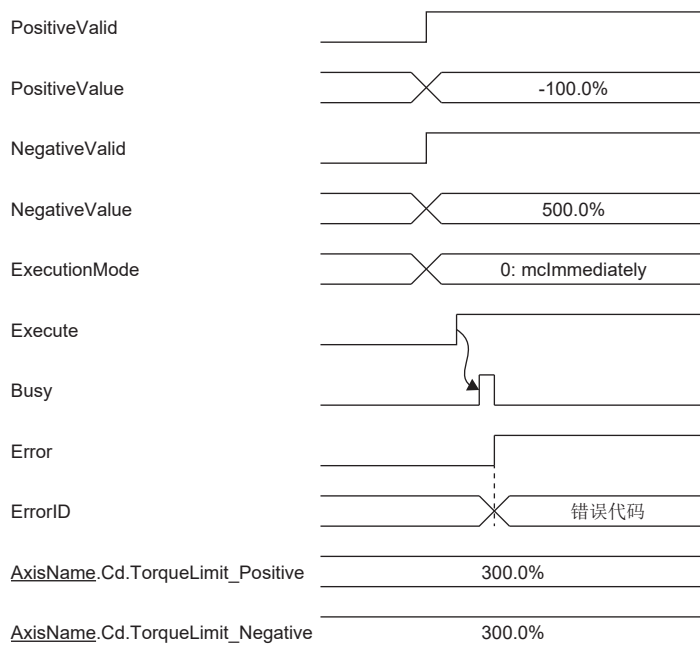
位	名称	内容
16	FB启动后取消允许	设置MCv_SetTorqueLimit (转矩限制值) 启动后是否允许取消。 • 0: 不允许 • 1: 允许

■动作概要

- 在MCv_SetTorqueLimit(转矩限制值)中,更改正方向转矩限制值(AxisName.Cd.TorqueLimit_Positive)、负方向转矩限制值(AxisName.Cd.TorqueLimit_Negative)。



- 在正方向转矩限制值(PositiveValue)或负方向转矩限制值(NegativeValue)中输入超出范围的值,将执行指令(Execute)置为TRUE时,MCv_SetTorqueLimit(转矩限制值)将发生错误,正方向转矩限制值(AxisName.Cd.TorqueLimit_Positive)、负方向转矩限制值(AxisName.Cd.TorqueLimit_Negative)不更改。



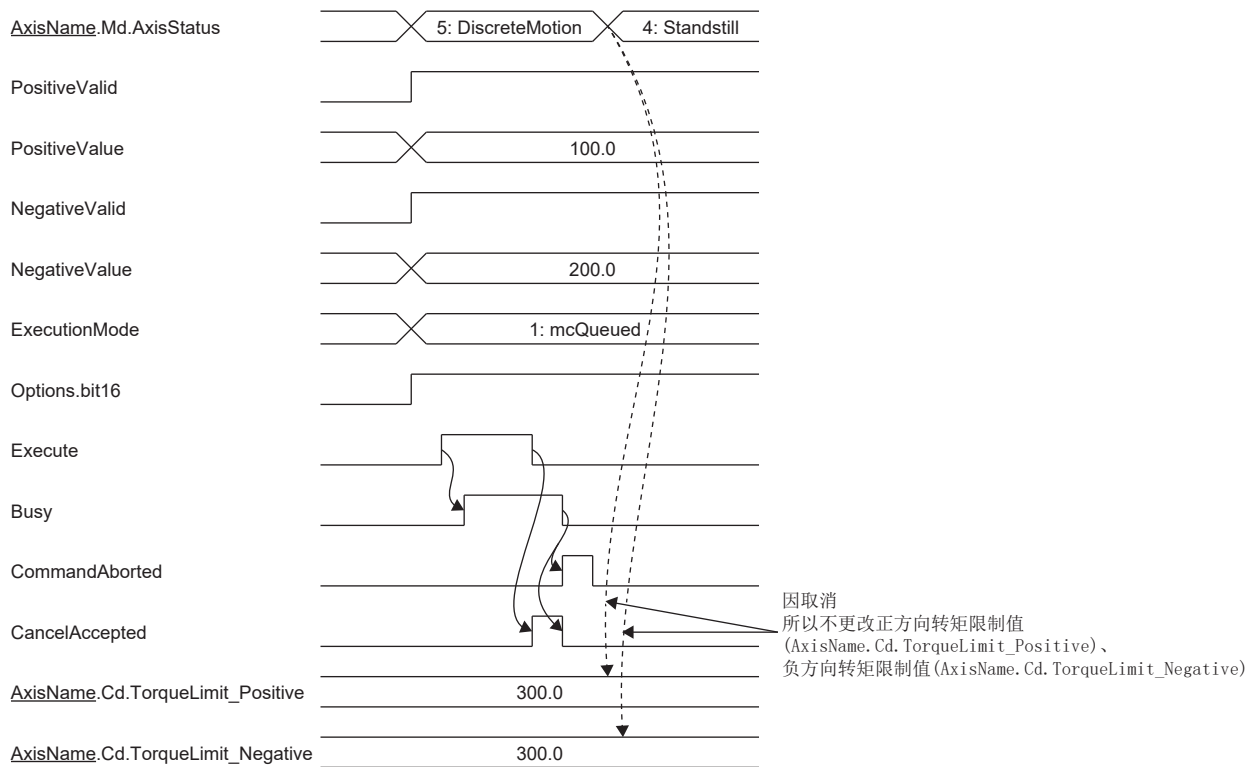
■取消

MCv_SetTorqueLimit (转矩限制值) 执行后可以取消处于待机状态的转矩限制值更改。

- 将取消置为有效时，在将“FB启动后取消允许(选项(Options): 位16)”设置为“1: 允许”的状态下，执行MCv_SetTorqueLimit (转矩限制值)。
- 通过执行指令(Execute)的下降沿检测，开始取消。
- 取消的受理只有在输出引脚的执行中(Busy)为TRUE时才进行。
- MCv_SetTorqueLimit (转矩限制值) 受理取消时输出引脚的取消受理(CancelAccepted)将变为TRUE。
- 取消完成时输出引脚的执行中断(CommandAborted)将变为TRUE。
- 取消的情况下，不更改正方向转矩限制值(AxisName.Cd.TorqueLimit_Positive)、负方向转矩限制值(AxisName.Cd.TorqueLimit_Negative)。

例

启动模式(ExecutionMode)以“1: 等待完成后执行(mcQueued)”启动后，在轴状态(AxisName.Md.AxisStatus)至“4: 待机中(Standstill)”的切换待机中取消的情况下



注意事项

在MCv_SetTorqueLimit (转矩限制值) 中，更改正方向转矩限制值(AxisName.Cd.TorqueLimit_Positive)、负方向转矩限制值(AxisName.Cd.TorqueLimit_Negative)。

程序示例

将转矩限制值更改指令(bSetTorqueLimit)置为TRUE，并将轴1(Axis0001)的正方向转矩限制值更改为“100.0”，将负方向转矩限制值更改为“200.0”的程序示例如下所示。

■轴

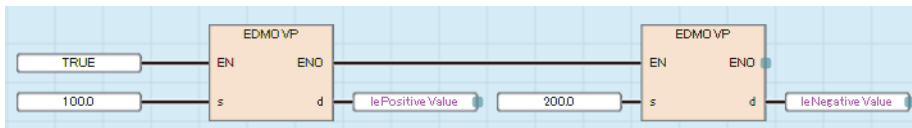
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

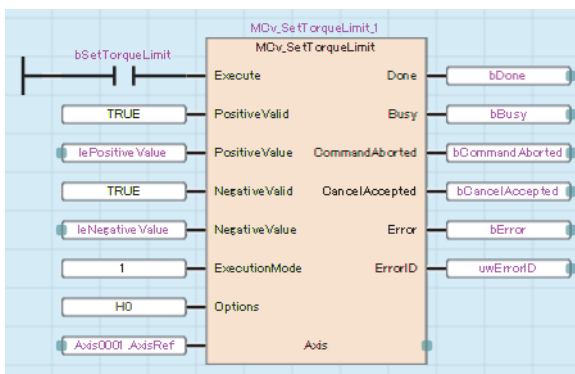
标签名	数据类型	注释
MCv_SetTorqueLimit_1	MCv_SetTorqueLimit	转矩限制值FB
bSetTorqueLimit	位	转矩限制值更改指令
lePositiveValue	双精度实数	正方向转矩限制值
leNegativeValue	双精度实数	负方向转矩限制值
bDone	位	准备就绪
bBusy	位	执行中
bCommandAborted	位	执行中断
bCancelAccepted	位	取消受理
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

- 转矩限制值更改用数据设置



- 转矩限制值更改



■ST程序

```
//-----转矩限制值更改用数据设置-----
lePositiveValue:= 100.0;
leNegativeValue:= 200.0;

//-----转矩限制值更改-----
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
);
```

45.6 倍率修调值设置

MC_SetOverride

执行指定的轴的目标速度、目标加速度、目标减速度的更改。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName.AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName.AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	输入获取	有效范围	初始值	内容
Enable	有效	BOOL	始终	TRUE、FALSE	FALSE	TRUE时执行MC_SetOverride(倍率修调值设置)。
VelFactor	速度倍率修调系数	LREAL	始终	0.00~10.00	0.00	设置速度的倍率修调系数。 有效(Enable)为TRUE时，将始终获取。
AccFactor	加速度倍率修调系数	LREAL	始终	0.00、0.01~10.00	0.00	设置加速度的倍率修调系数。 有效(Enable)为TRUE时，将始终获取。 设置了“0.00”的情况下，不进行加速度倍率修调系数的更改，维持上次值进行控制。
JerkFactor	Jerk倍率修调系数	LREAL	始终	0.0	0.0	应设置“0.0”。 * 设置了“0.0”以外的情况下，将变为超出Jerk倍率修调系数(JerkFactor)范围(错误代码：1BBEH)。

■输出变量

输出变量	名称	数据类型	初始值	内容
Enabled	有效中	BOOL	FALSE	倍率修调值正常设置的情况下，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_SetOverride(倍率修调值设置)时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 ☞ 所使用的控制器的用户手册

功能

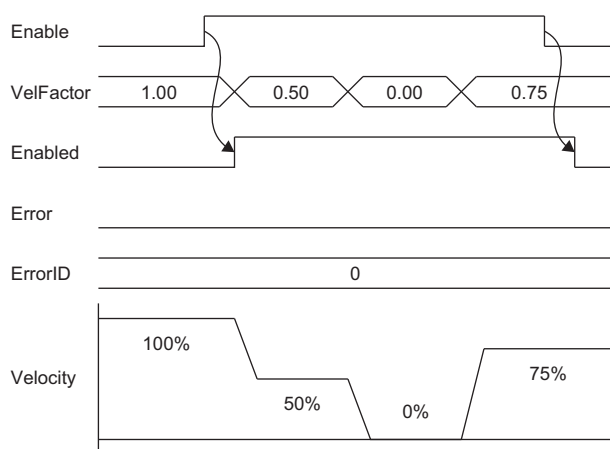
- 更改指定的轴的目标速度、目标加速度、目标减速度。
- 更改为当前动作中的目标速度、目标加速度、目标减速度乘以倍率修调系数后的值。
- 有效(Enable)变为TRUE时将执行MC_SetOverride(倍率修调值设置)。倍率修调系数为有效中时，有效中(Enabled)将变为TRUE。
- 有效(Enable)为TRUE时，如果更改倍率修调系数的值，则将反映新的倍率修调系数。
- 在MC_SetOverride(倍率修调值设置)内发生了异常的情况下，错误(Error)将变为TRUE，并将错误代码存储到错误代码(ErrorID)中。关于错误代码的详细内容，请参阅下述手册。

📖 所使用的控制器的用户手册

- 如果在速度倍率修调系数(VelFactor)中设置“0.00”的值，轴状态(AxisName.Md.AxisStatus)不转移至“4: 待机中(Standstill)”而停止轴。
- 在加速度倍率修调系数(AccFactor)中设置“0.00”的值时，不更改加速度倍率修调系数，维持上次的加速度倍率修调系数。

■ 时序图

- 正常完成的情况下



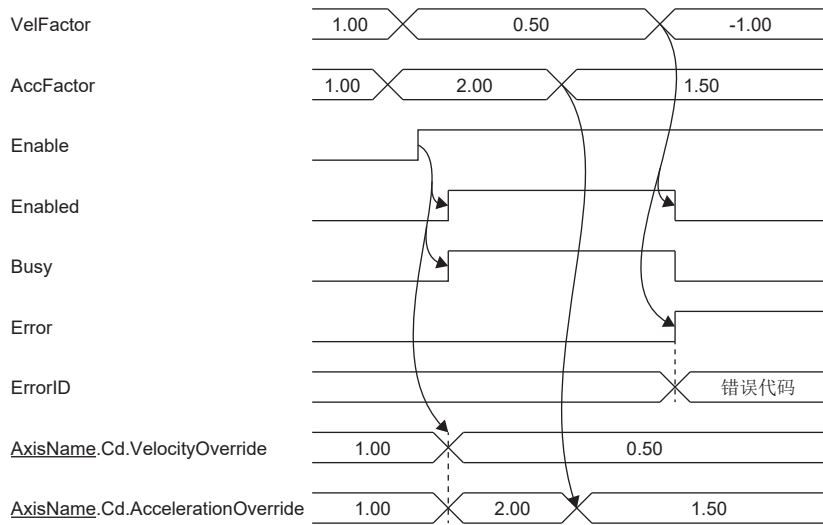
- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

📖 1378页 通过有效(Enable)类型的运动控制FB的基本动作

■动作概要

- 在MC_SetOverride(倍率修调值设置)中,更改速度倍率修调系数(AxisName.Cd.VelocityOverride)、加速度倍率修调系数(AxisName.Cd.AccelerationOverride)。



- 在倍率修调系数中设置了超出范围的值的的情况下,MC_SetOverride(倍率修调值设置)将发生错误并停止以后的获取。再次执行获取的情况下,应再次启动有效(Enable)。

注意事项

- 请勿对同一轴配置2个及以上的MC_SetOverride(倍率修调值设置)。配置了2个及以上的情况下将无法保证动作。
- 在单轴控制中,仅轴控制数据中设置的倍率修调系数有影响。
- MC_SetOverride(倍率修调值设置)执行中,请勿直接进行速度倍率修调系数(AxisName.Cd.VelocityOverride)、加速度倍率修调系数(AxisName.Cd.AccelerationOverride)的操作。
- 速度倍率修调后的速度超出范围的情况下,请参阅下述手册的“速度范围”。

📖所使用的控制器的用户手册

- 加速度倍率修调后的加速度、减速度、加速时间、减速时间超出范围的情况下,请参阅下述手册的“加减速处理功能”。

📖所使用的控制器的用户手册

程序示例

将倍率修调值更改指令(bSetOverride)置为TRUE, 并将其更改为轴1(Axis0001)的目标速度、目标加速度、目标减速度乘以速度倍率修调系数“1.0”、加减速速度倍率修调系数“2.0”后的值的程序示例如下所示。

■轴

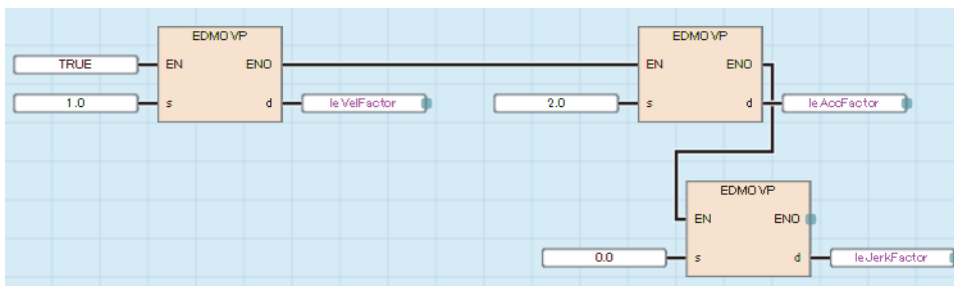
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

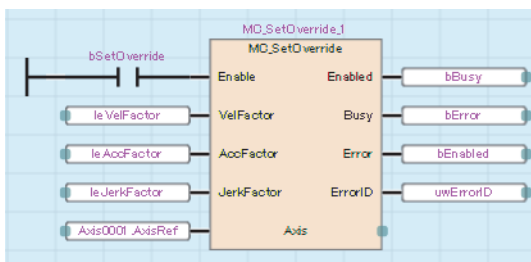
标签名	数据类型	注释
MC_SetOverride_1	MC_SetOverride	倍率修调值设置FB
bSetOverride	位	倍率修调值更改指令
leVelFactor	双精度实数	速度倍率修调系数
leAccFactor	双精度实数	加减速速度倍率修调系数
leJerkFactor	双精度实数	Jerk倍率修调系数
bEnabled	位	有效中
bBusy	位	执行中
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

- 倍率修调值更改用数据设置



- 倍率修调值更改



■ST程序

```
//-----倍率修调值更改用数据设置-----
```

```
leVelFactor:= 1.00;
```

```
leAccFactor:= 2.00;
```

```
leJerkFactor:= 0.0;
```

```
//-----倍率修调值更改-----
```

```
MC_SetOverride_1(
```

```
    Axis:= Axis0001.AxisRef ,
```

```
    Enable:= bSetOverride ,
```

```
    VelFactor:= leVelFactor ,
```

```
    AccFactor:= leAccFactor ,
```

```
    JerkFactor:= leJerkFactor ,
```

```
    Enabled=> bEnabled ,
```

```
    Busy=> bBusy ,
```

```
    Error=> bError ,
```

```
    ErrorID=> uwErrorID
```

```
);
```

45.7 参数读取

MC_ReadParameter

进行设备的对象的读取。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName, AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	输入获取	有效范围	初始值	内容
Enable	有效	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_ReadParameter(参数读取)。
ParameterNumber	参数编号	DWORD (HEX)	始终	00010000H~ FFFFFFFFH	00000000H	设置设备的对象。 关于详细内容，请参阅下述章节。 ☞ 1498页 参数编号(ParameterNumber)
ReadCount	读取次数	WORD (UNIT)	始终	0~65535	0	设置参数的读取次数。 设置了“0”的情况下，在有效(Enable)变为FALSE之前将进行连续读取。 重新获取在更改了参数编号的时机进行。
Options	选项	DWORD (HEX)	始终	00000000H~ 00010000H	00000000H	将MC_ReadParameter(参数读取)的功能选项以位指定进行设置。 重新获取在更改了参数编号的时机进行。 关于详细内容，请参阅下述章节。 ☞ 1498页 选项(Options)

■输出变量

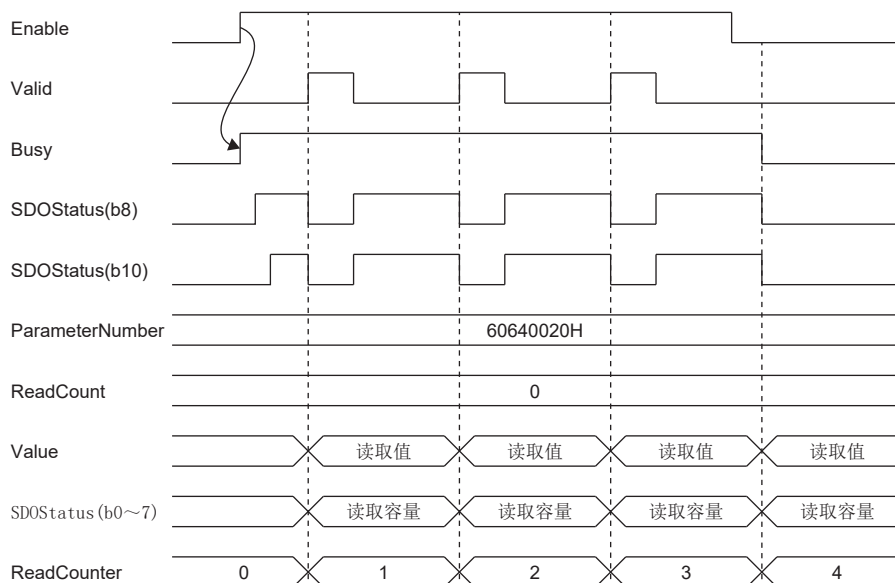
输出变量	名称	数据类型	初始值	内容
Valid	输出值有效	BOOL	FALSE	输出值为有效时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_ReadParameter (参数读取)时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 □□所使用的控制器的用户手册
SDOErrorID	瞬时错误代码	DWORD (HEX)	0000000H	在SDO通信中发生了异常时，将返回响应代码(SDO Abort Code)。
Value	读取值	LREAL	0.0	输出指定的参数的读取值。 成为对象的参数为整数数据的情况下也作为LREAL型存储。
SDOStatus	SDO传送状态	WORD (UINT)	0	存储瞬时请求的处理状态。 关于详细内容，请参阅下述章节。 ☞ 1498页 SDO传送状态(SDOStatus)
ReadCounter	读取次数	WORD (UINT)	0	存储参数的读取次数。 在读取次数(ReadCount)中设置了“0”的情况下，将变为“0~65535”的环形计数器。

功能

- 通过在参数编号(ParameterNumber)中设置00010000H及以后来读取对象数据。在此情况下，使用瞬时传送功能发送接收对象数据。
- 从软元件设备接收了对瞬时请求的错误(响应代码(SDO Abort Code))的情况下，错误(Error)将变为TRUE，并在错误代码(ErrorID)中将存储SDO通信异常(错误代码：1A48H)，在瞬时错误代码(SDOErrorID)中将存储设备的响应代码(SDO Abort Code)。

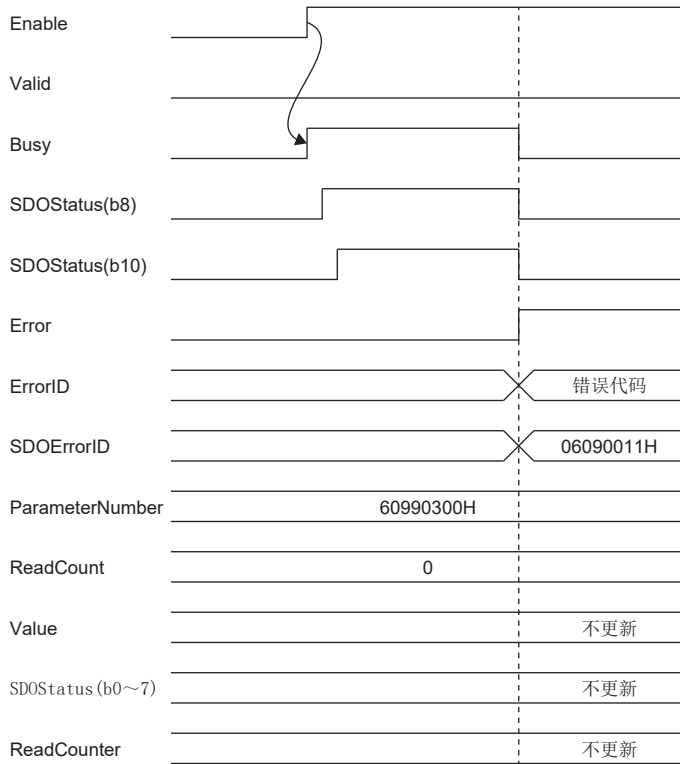
■时序图

- 正常完成的情况下

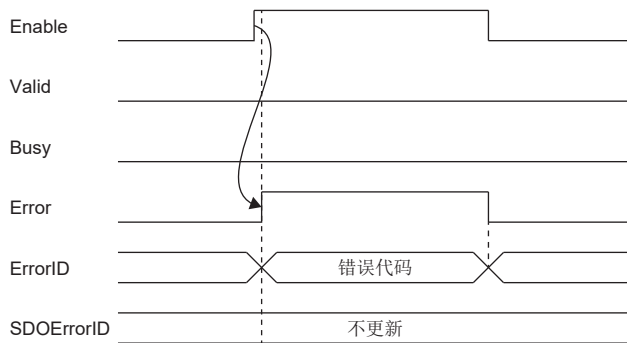


在有效(Enable)变为FALSE之前，按照读取次数(ReadCount)中设置的次数执行读取
(读取次数(ReadCount)为“0”的情况下，连续执行直至有效(Enable)变为FALSE为止)

- 异常完成的情况下
 - 发生SDO通信错误时



- 输入输出变量、输入变量异常

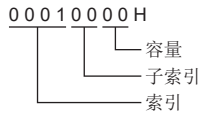


■参数编号 (ParameterNumber)

- 参数读取时设置的参数编号 (ParameterNumber) 的区分如下所示。

参数编号	内容
00010000H~FFFFFFFH	设备的对象

- 设置设备的对象的参数编号 (ParameterNumber) 的内容如下所示。



- 对于设备的对象设置时的容量，应将1字节 (08H) / 2字节 (10H) / 4字节 (20H) / 8字节 (40H) 替换为位数后进行设置。设置了“00H”的情况下，在参数读取中将以默认容量读取。设置了设定值以外的值的情况下，将变为超出参数编号范围 (错误代码: 1BC2H)。
- 参数编号 (ParameterNumber) 中设置的容量与对象数据的默认容量不相同时的动作取决于驱动器模块的规格。

例

设置对象索引“6099H”、子索引“02H”的UNSIGNED32对象、容量“00H”的情况下设置“60990200H”。

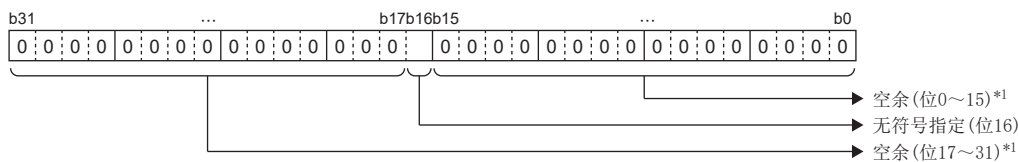
要点

关于各软元件设备的对象，请参阅软元件设备的手册。

■选项 (Options)

将MC_ReadParameter (参数读取) 中使用的功能选项以位指定进行设置。

以位指定设置的内容如下所示。

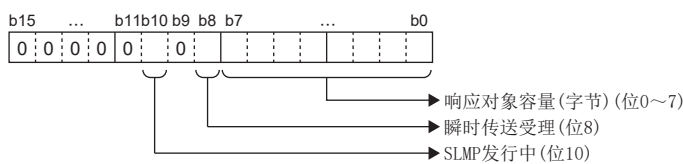


*1 空余中应设置“0”。设置了“0”以外的情况下，将变为超出Options范围 (错误代码: 1ABBH)。

位	名称	内容
16	无符号指定	设置将整数数据的读取值是作为有符号存储还是作为无符号存储。 • 0: 有符号 • 1: 无符号

■SDO传送状态 (SDOStatus)

存储瞬时请求的处理状态。



位	名称	内容
0~7	响应对象容量 (字节)	处理完成时，存储设备响应的对象容量。
8	瞬时传送受理	存储瞬时传送的状态。 • 0: 瞬时传送未执行 • 1: 瞬时传送受理中
10	SLMP发行中	存储SLMP的状态。 • 0: SLMP未执行 • 1: SLMP发行中

注意事项

- 对同一轴最多可执行4个瞬时传送。对已经正在执行4个瞬时传送的轴执行将对象数据作为对象的MC_ReadParameter(参数读取)时, 将变为参数读写FB执行不可错误(错误代码: 1BC1H), 且不执行后续的FB。但是, 执行中的FB将继续。
- 在驱动器式原点复位中, 使用瞬时传送功能进行驱动器模块的原点数据的获取。因此, 执行MC_Home(原点复位)时正在执行将对象数据作为对象的参数读取的情况下, MC_Home(原点复位)中有可能变为ABS基准点读取错误(错误代码: 1AF9H)。
- 对整数型、BOOL型的参数及对象数据进行读取的情况下, 通过MC_ReadParameter(参数读取)读取的值将转换为LREAL型、BOOL型后输出。

例

使用MC_ReadParameter(参数读取)通过2字节容量指定读取参数

参数值	读取值(Value)(数据类型: LREAL)	
	无符号指定(选项(Options): 位16) “0: 有符号”	无符号指定(选项(Options): 位16) “1: 无符号”
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

- 通过MC_ReadParameter(参数读取)更改的值不会保存。在下次接通电源时使用更改的值的的情况下, 应进行参数的保存。设备对象的情况下, 关于参数的保存方法, 请参阅设备的手册。
- 瞬时传送中的软元件设备解除连接的情况下, 将解除连接的软元件设备作为对象的瞬时请求将全部错误, 并发生SDO通信异常(错误代码: 1A48H)。
- MC_ReadParameter(参数读取)应在确认驱动器状态(AxisName.Md.Driver_State)是否为“-1: 轴变量未初始化/轴参数异常(Invalid)”或“0: Not Ready To Switch On(NotReadyToSwitchOn)”后再启动。在上述以外的情况下执行时, 将发生参数读写FB不可执行错误(错误代码: 1BC1H)。

程序示例

将参数读取指令(bReadParameter)置为TRUE, 并读取轴1(Axis0001)的索引“6099H”、子索引“02H”、容量“00H”的对象的值的程序示例如下所示。

■轴

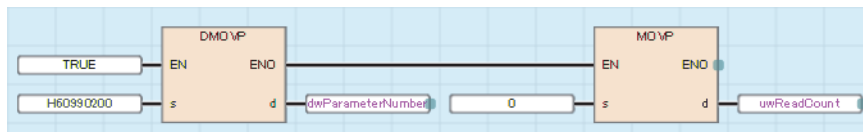
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

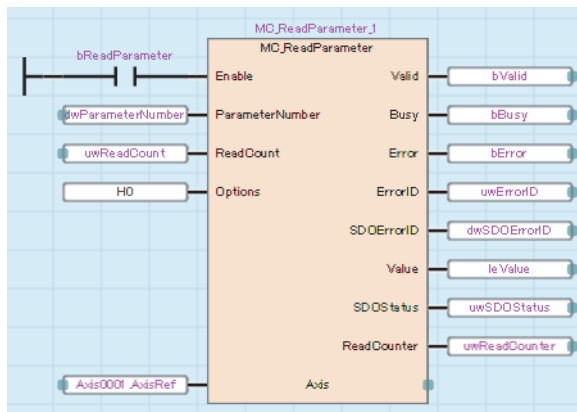
标签名	数据类型	注释
MC_ReadParameter_1	MC_ReadParameter	参数读取FB
bReadParameter	位	参数读取指令
dwParameterNumber	双字[无符号]/位串[32位](16进制数)	参数编号
uwReadCount	字[无符号]/位串[16位]	读取次数
bValid	位	输出值有效
bBusy	位	执行中
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码
dwSDOErrorID	双字[无符号]/位串[32位](16进制数)	瞬时错误代码
leValue	双精度实数	读取值
uwSDOStatus	字[无符号]/位串[16位]	SDO传送状态
uwReadCounter	字[无符号]/位串[16位]	读取次数

■FBD/LD程序

- 参数读取用数据设置



- 参数读取



■ST程序

```
//-----参数读取用数据设置-----  
dwParameterNumber:= H60990200;  
uwReadCount:= 0;  
  
//-----参数读取-----  
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 ,  
    SDOSStatus=> uwSDOSStatus ,  
    ReadCounter=> uwReadCounter  
);
```

45.8 参数写入

MC_WriteParameter

进行设备的对象的写入。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName, AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_WriteParameter(参数写入)。
ParameterNumber	参数编号	DWORD (HEX)	启动时	00010000H~ FFFFFFFH	00000000H	设置设备的对象。 关于详细内容，请参阅下述章节。 ☞ 1504页 参数编号(ParameterNumber)
Value	设定值	LREAL	启动时	—	0.0	设置已设置的参数的设定值。
ExecutionMode	启动模式	INT (MC_EXECUTION_MODE)	启动时	0、1	0	设置已设置的参数的写入方法。 • 0: 立即写入(mcImmediately) • 1: 等待轴的停止之后再写入(mcQueued) 关于详细内容，请参阅下述章节。 ☞ 1505页 启动模式(ExecutionMode)
Options	选项	DWORD (HEX)	启动时	00000000H~ 00010000H	00000000H	将MC_WriteParameter(参数写入)的功能选项以位指定进行设置。 关于详细内容，请参阅下述章节。 ☞ 1505页 选项(Options)

■输出变量

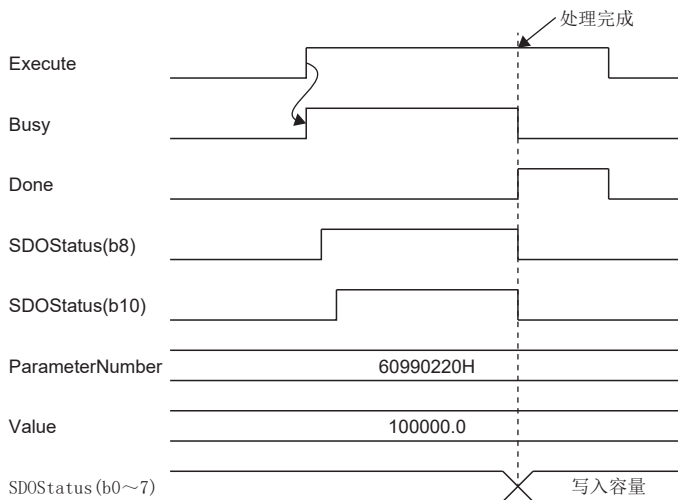
输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	参数的写入完成时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_WriteParameter(参数写入)时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 ☞所使用的控制器的用户手册
SDOErrorID	瞬时错误代码	DWORD (HEX)	0000000H	在SDO通信中发生了异常时，将返回响应代码(SDO Abort Code)。
SDOStatus	SDO传送状态	WORD (UINT)	0	存储瞬时请求的处理状态。 关于详细内容，请参阅下述章节。 ☞1505页 SDO传送状态(SDOStatus)

功能

- 通过在参数编号(ParameterNumber)中设置00010000H及以后来写入对象数据。在此情况下，使用瞬时传送功能发送接收对象数据。
- 从软元件设备接收了对瞬时请求的错误(响应代码(SDO Abort Code))的情况下，错误(Error)将变为TRUE，并在错误代码(ErrorID)中将存储SDO通信异常(错误代码：1A48H)，在瞬时错误代码(SDOErrorID)中将存储设备的响应代码(SDO Abort Code)。

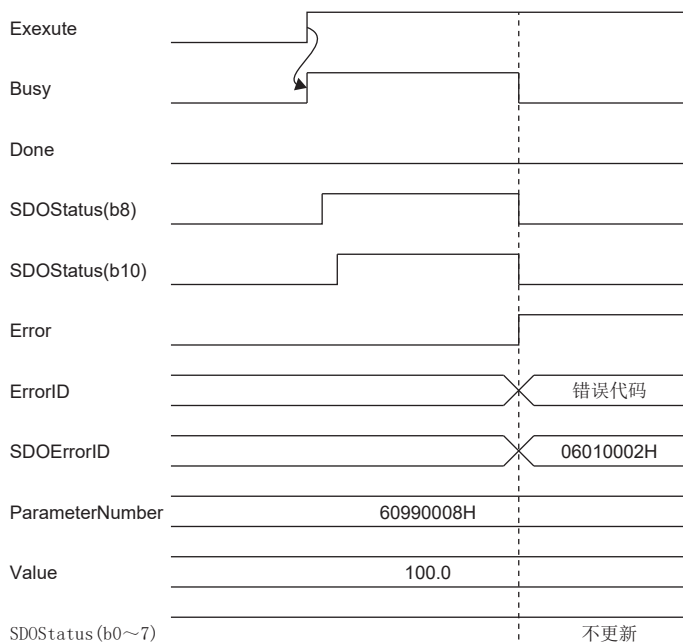
■时序图

- 正常完成的情况下

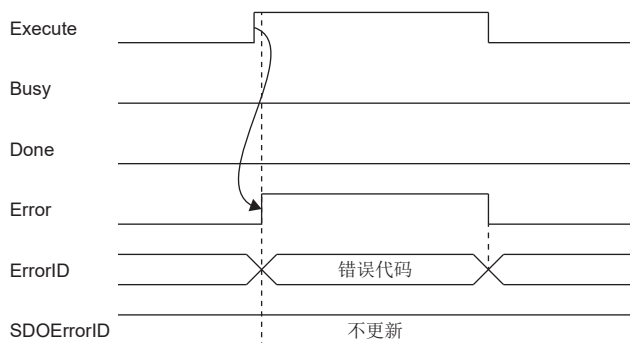


• 异常完成的情况下

- 发生SDO通信错误时



- 输入输出变量、输入变量异常

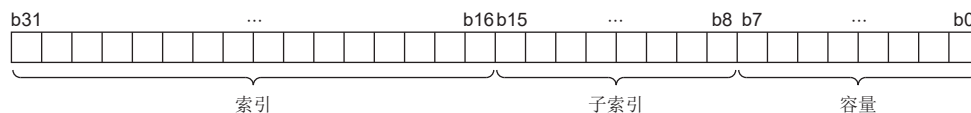


■ 参数编号 (ParameterNumber)

- 参数写入时设置的参数编号 (ParameterNumber) 的区分如下所示。

参数编号	内容
00010000H~FFFFFFFH	设备的对象

- 设置设备的对象的参数编号 (ParameterNumber) 的内容如下所示。



- 对于设备的对象设置时的容量，应将1字节 (08H)/2字节 (10H)/4字节 (20H)/8字节 (40H) 替换为位数后进行设置。设置了“00H”的情况下，在参数写入中将以4字节写入。设置了设定值以外的值的情况下，将变为超出参数编号范围 (错误代码: 1BC2H)。
- 参数编号 (ParameterNumber) 中设置的容量与对象数据的默认容量不相同时的动作取决于驱动器模块的规格。

例

设置对象索引“6099H”、子索引“02H”的UNSIGNED32对象、容量“00H”的情况下设置“60990200H”。

■启动模式 (ExecutionMode)

参数写入执行时，将写入的时机通过启动模式(ExecutionMode)进行设置。

设定值	内容
0: 立即写入 (mcImmediately)	立即执行写入。 反映时机取决于各参数，因此有可能影响轴动作。
1: 等待轴的停止之后再写入 (mcQueued)	等待轴的停止 (轴状态 (AxisName.Md.AxisStatus) 为 “0: 轴无效 (Disabled)” 或 “4: 待机中 (Standstill)”) 之后再执行写入。

■选项 (Options)

将MC_WriteParameter (参数写入) 中使用的功能选项以位指定进行设置。

以位指定设置的内容如下所示。

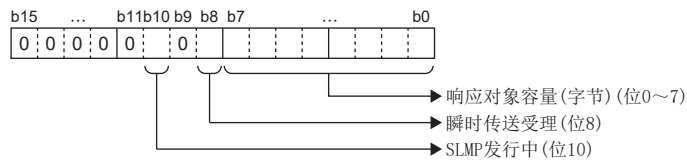


*1 空余中应设置“0”。设置了“0”以外的情况下，将变为超出Options范围 (错误代码: 1ABBH)。

位	名称	内容
16	无符号指定	设置将整数数据的设定值是作为有符号存储还是作为无符号存储。 • 0: 有符号 • 1: 无符号

■SDO传送状态 (SDOStatus)

存储瞬时请求的处理状态。



位	名称	内容
0~7	响应对象容量 (字节)	处理完成时，存储设备响应的对象容量。
8	瞬时传送受理	存储瞬时传送的状态。 • 0: 瞬时传送未执行 • 1: 瞬时传送受理中
10	SLMP发行中	存储SLMP的状态。 • 0: SLMP未执行 • 1: SLMP发行中

注意事项

- 对同一轴最多可执行4个瞬时传送。对已经正在执行4个瞬时传送的轴执行将对象数据作为对象的MC_WriteParameter(参数写入)时, 将变为参数读写FB执行不可错误(错误代码: 1BC1H), 且不执行后续的FB。(执行中的FB将继续。)
- 在驱动器式原点复位中, 使用瞬时传送功能进行驱动器模块的原点数据的获取。因此, 执行MC_Home(原点复位)时正在执行将对象数据作为对象的参数写入的情况下, MC_Home(原点复位)中有可能变为ABS基准点读取错误(错误代码: 1AF9H)。
- 对整数型、BOOL型的参数及对象数据进行写入的情况下, MC_WriteParameter(参数写入)的设定值将转换为整数型、BOOL型后获取。

例

使用MC_WriteParameter(参数写入)通过2字节容量指定写入参数

设定值(Value) (数据类型: LREAL)	写入值(2字节数据)	
	无符号指定(选项(Options): 位16) “0: 有符号”	无符号指定(选项(Options): 位16) “1: 无符号”
10000.5	10000	10000
-10000.4	-10000	55536
65535.0	-1	65535
-65535.0	1	1
63356.0	0	0
131701.0	-1	65535

- 通过MC_WriteParameter(参数写入)更改的值不会保存。在下次接通电源时使用更改的值的的情况下, 应进行参数的保存。设备对象的情况下, 关于参数的保存方法, 请参阅设备的手册。
- 瞬时传送中的软元件设备解除连接的情况下, 将解除连接的软元件设备作为对象的瞬时请求将全部错误, 并发生SDO通信异常(错误代码: 1A48H)。
- MC_WriteParameter(参数写入)应在确认驱动器状态(AxisName.Md.Driver_State)是否为“-1: 轴变量未初始化/轴参数异常(Invalid)”或“0: Not Ready To Switch On(NotReadyToSwitchOn)”后再启动。在上述以外的情况下执行时, 将发生参数读写FB不可执行错误(错误代码: 1BC1H)。

程序示例

将参数写入指令(bWriteParameter)置为TRUE, 并在轴1(Axis0001)的索引“6099H”、子索引“02H”、容量“00H”的对象中写入“100.0”的程序示例如下所示。

■轴

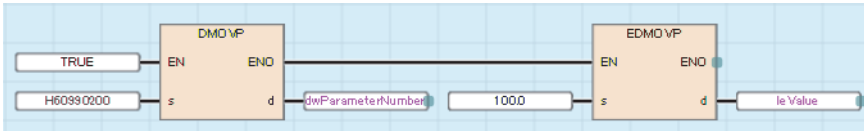
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

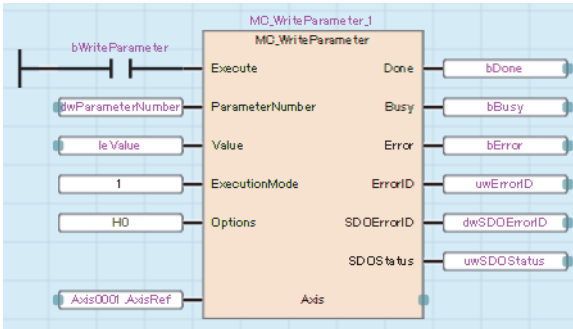
标签名	数据类型	注释
MC_WriteParameter_1	MC_WriteParameter	参数写入FB
bWriteParameter	位	参数写入指令
dwParameterNumber	双字[无符号]/位串[32位](16进制数)	参数编号
leValue	双精度实数	设定值
bDone	位	执行完成
bBusy	位	执行中
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码
dwSDOErrorID	双字[无符号]/位串[32位](16进制数)	瞬时错误代码
uwSDOStatus	字[无符号]/位串[16位]	SDO传送状态

■FBD/LD程序

- 参数写入用数据设置



- 参数写入



■ST程序

//-----参数写入用数据设置-----

dwParameterNumber:= H60990200;

leValid:= 100.0;

//-----参数写入-----

```
MC_WriteParameter_1(
    Axis:= Axis0001.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
);
```

45.9 轴错误复位

MC_Reset

对轴的错误、警告进行复位。

梯形图	FBD/LD	ST
		<pre>MC_Reset(Axis:= ?AXIS_REF?, Execute:= ?BOOL?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName.AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName.AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_Reset (轴错误复位)。
Options	选项	DWORD (HEX)	启动时	00000000H	00000000H	应设置“00000000H”。 * 设置了“00000000H”以外的情况下，将变为超出Options范围(错误代码: 1BC3H)。

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	复位完成时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_Reset (轴错误复位)时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	由于超时，MC_Reset (轴错误复位)中断时将变为TRUE。 执行指令(Execute)变为FALSE时，将变为FALSE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 ☞ 所使用的控制器的用户手册

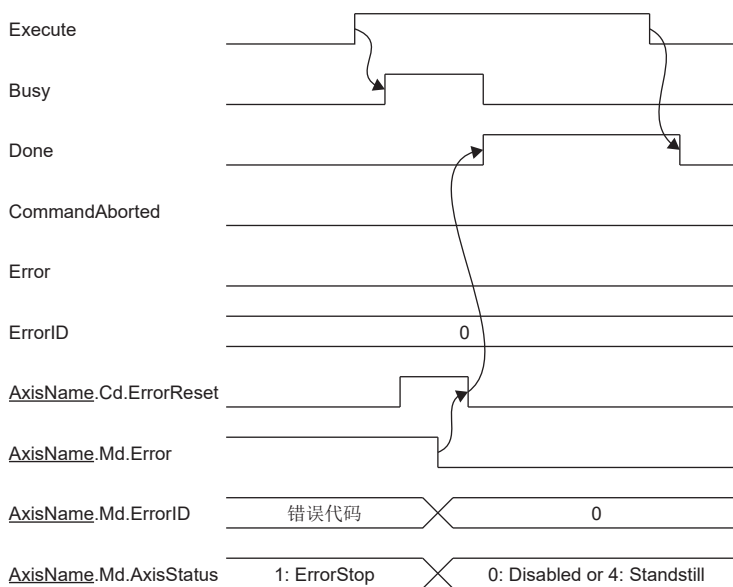
功能

- 执行指令(Execute)为TRUE时执行MC_Reset(轴错误复位), 开始处理时执行中(Busy)将变为TRUE, 将对象轴的轴错误复位(AxisName.Cd.ErrorReset)置为TRUE。
- 轴的错误、警告的解除完成时执行中(Busy)将变为FALSE, 执行完成(Done)将变为TRUE。
- 轴状态(AxisName.Md.AxisStatus)置为“1: 错误停止中(ErrorStop)”以外时执行, 解除警告后, 执行中(Busy)将变为FALSE, 执行完成(Done)将变为TRUE。
- 即使在残留了轴的错误、警告的原因的状态下将执行指令(Execute)置为TRUE, 也不解除错误、警告。执行指令后1秒以内未解除错误的情况下, 执行中断(CommandAborted)将变为TRUE。应将执行指令(Execute)置为FALSE一次, 消除了错误、警告的原因后, 再次将执行指令(Execute)置为TRUE。
- 在MC_Reset(轴错误复位)内发生了异常的情况下, 错误(Error)将变为TRUE, 并将错误代码存储到错误代码(ErrorID)中。关于错误代码的详细内容, 请参阅下述手册。

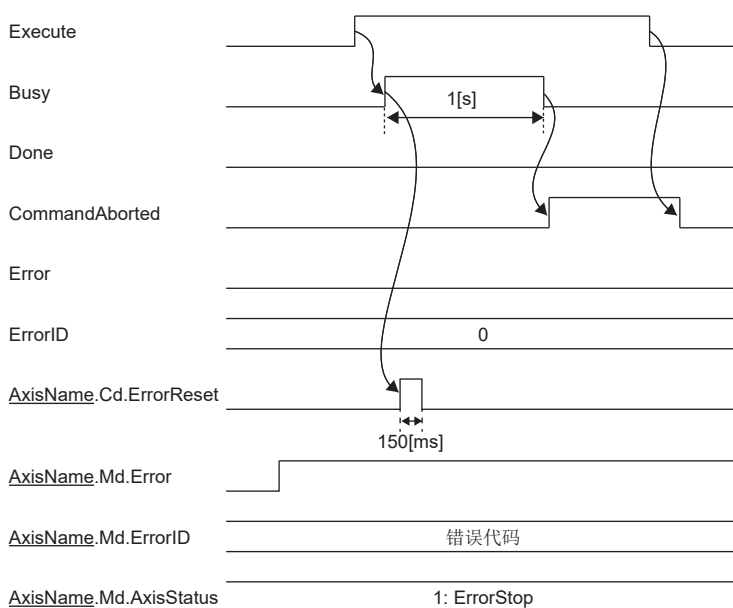
📖 所使用的控制器的用户手册

■ 时序图

- 正常完成的情况下



- 超时的情况下



注意事项

在执行MC_Reset (轴错误复位)的过程中, 请勿进行轴错误复位 (AxisName.Cd. ErrorReset) 的直接操作。

程序示例

将错误复位指令(bErrorReset)置为TRUE, 并对轴1 (Axis0001)的错误、警告进行复位的程序示例如下所示。

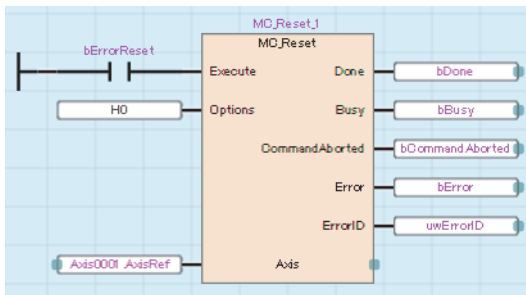
■轴

轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

标签名	数据类型	注释
MC_Reset_1	MC_Reset	轴错误复位FB
bErrorReset	位	错误复位指令
bDone	位	执行完成
bBusy	位	执行中
bCommandAborted	位	执行中断
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序



■ST程序

```
MC_Reset_1(  
  Axis:= Axis0001.AxisRef ,  
  Execute:= bErrorReset ,  
  Options:= H00000000 ,  
  Done=> bDone ,  
  Busy=> bBusy ,  
  CommandAborted=> bCommandAborted ,  
  Error=> bError ,  
  ErrorID=> uwErrorID  
);
```

45.10 轴组错误复位

MC_GroupReset

对轴组及轴组中所属的各轴的错误、警告进行复位。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
AxesGroup	轴组信息	AXES_GROUP_REF	启动时	—	不能省略	设置轴组。 关于使用的变量(AxesGroupName.AxesGroupRef.)，请参阅下述内容。 ☞ 1408页 AxesGroupName.AxesGroupRef. (轴组信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_GroupReset(轴组错误复位)。
Options	选项	DWORD(HEX)	启动时	00000000H	00000000H	应设置“00000000H”。 * 设置了“00000000H”以外的情况下，将变为超出Options范围(错误代码：1BC3H)。

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	轴组的所有轴的复位完成时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_GroupReset(轴组错误复位)时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	由于超时，MC_GroupReset(轴组错误复位)中断时将变为TRUE。 执行指令(Execute)变为FALSE时，将变为FALSE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 ☞ 所使用的控制器的用户手册

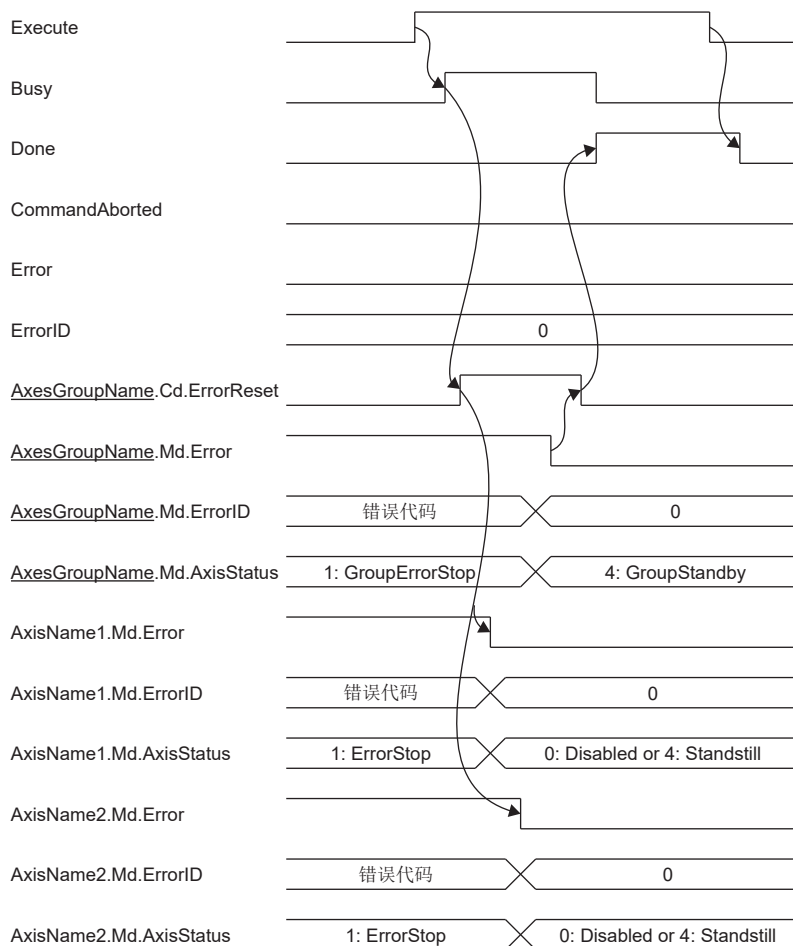
功能

- 在执行指令(Execute)的上升沿检测中执行MC_GroupReset(轴组错误复位), 开始处理时执行中(Busy)将变为TRUE, 将对象轴组的轴组错误复位(AxesGroupName.Cd.ErrorReset)置为TRUE。
- 轴组的错误或警告及单轴的错误或警告的解除完成时执行中(Busy)将变为FALSE, 执行完成(Done)将变为TRUE。
- 轴组状态(AxesGroupName.Md.GroupStatus)为“1: 错误停止中(GroupErrorStop)”以外时执行, 解除警告后, 执行中(Busy)将变为FALSE, 执行完成(Done)将变为TRUE。
- 即使在残留了轴的错误或警告的原因的状态下将执行指令(Execute)置为TRUE, 也不解除错误、警告。在这种情况下, 如果在执行FB后的1秒内未解除错误, 执行中断(CommandAborted)将变为TRUE, 将对象轴组的轴组错误复位(AxesGroupName.Cd.ErrorReset)置为FALSE。应将执行指令(Execute)置为FALSE一次, 消除了错误或警告的原因后, 再次将执行指令(Execute)置为TRUE。
- 在MC_GroupReset(轴组错误复位)内发生了异常的情况下, 错误(Error)将变为TRUE, 并将错误代码存储到错误代码(ErrorID)中。关于错误代码的详细内容, 请参阅下述手册。

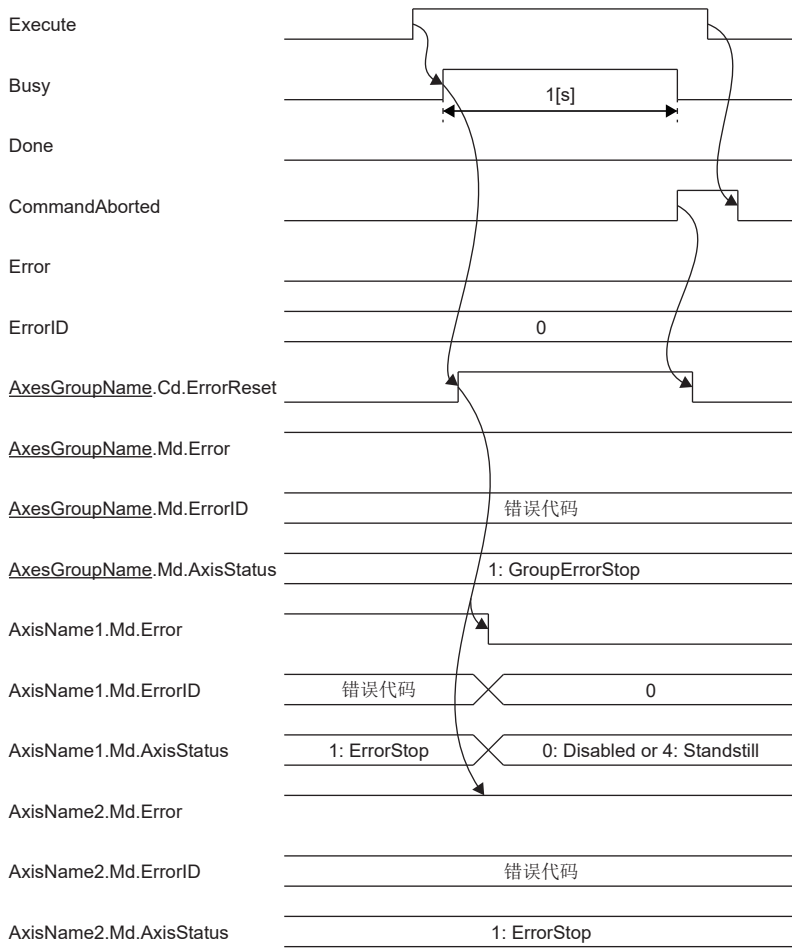
📖 所使用的控制器的用户手册

■ 时序图

- 正常完成的情况下



• 超时的情况下



注意事项

在执行MC_GroupReset (轴组错误复位) 的过程中, 请勿进行轴组错误复位 (AxesGroupName.Cd.ErrorReset) 的直接操作。

程序示例

将轴组错误复位指令(bGroupErrorReset)置为TRUE，并对轴组1(AxesGroup001)的错误、警告进行复位的程序示例如下所示。

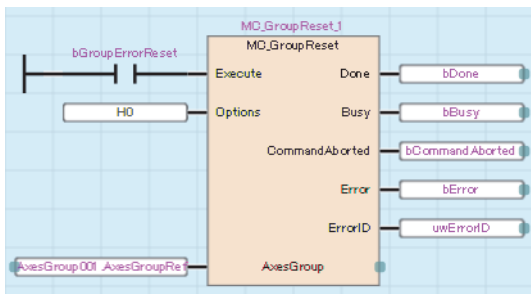
■轴组

轴组No.	标签名	数据类型	注释
1	AxesGroup001	AXES_GROUP_REF	轴组1

■使用的标签

标签名	数据类型	注释
MC_GroupReset_1	MC_GroupReset	轴组错误复位FB
bGroupErrorReset	位	轴组错误复位指令
bDone	位	执行完成
bBusy	位	执行中
bCommandAborted	位	执行中断
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序



■ST程序

```

MC_GroupReset_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bGroupErrorReset ,
    Options:= H00000000 ,
    Done=> bDone ,
    Busy=> bBusy ,
    CommandAborted=> bCommandAborted ,
    Error=> bError ,
    ErrorID=> uwErrorID
);
    
```

45.11 触摸探针有效

MC_TouchProbe

通过触发事件发生记录任意数据。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	可省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName, AxisRef. (轴信息)
TriggerInput	触发输入信号	MC_TRIGGER_REF	启动时	—	不能省略	设置触发输入信号。 关于详细内容，请参阅下述章节。 ☞ 1520页 触发输入信号(TriggerInput)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_TouchProbe(触摸探针有效)。
ContinuousUpdate	连续更新	BOOL	启动时	TRUE、FALSE	FALSE	设置将窗口有效(WindowOnly)、下限位置(FirstPosition)、上限位置(LastPosition)的连续更改是置为有效还是置为无效。 • FALSE: 无效 • TRUE: 有效
WindowOnly	窗口有效	BOOL	启动时/可重启/可连续更新	TRUE、FALSE	FALSE	设置仅在下限位置与上限位置中设置的有效范围内发生了触发事件时，是否进行检测。 • FALSE: 在全范围内检测 • TRUE: 仅在设置的范围内检测 关于详细内容，请参阅下述章节。 ☞ 1520页 窗口有效(WindowOnly)/下限位置(FirstPosition)/上限位置(LastPosition)
FirstPosition	下限位置	LREAL	启动时/可重启/可连续更新	—	0.0	设置触发事件的有效范围的下限位置。 有效区域包含下限位置。 关于详细内容，请参阅下述章节。 ☞ 1520页 窗口有效(WindowOnly)/下限位置(FirstPosition)/上限位置(LastPosition)

输入变量	名称	数据类型	获取	有效范围	初始值	内容
LastPosition	上限位置	LREAL	启动时/可重启/可连续更新	—	0.0	设置触发事件的有效范围的上限位置。有效区域包含上限位置。关于详细内容，请参阅下述章节。 ☞ 1520页 窗口有效(WindowOnly)/下限位置(FirstPosition)/上限位置(LastPosition)
ProbeData	探针数据	TARGET_REF	启动时	—	—	设置探针数据。关于详细内容，请参阅下述章节。 ☞ 1521页 探针数据(ProbeData)
CompensationTime	补偿时间	LREAL	启动时	-5.0~5.0 [s]	0.0	设置对触摸探针处理的延迟时间进行补偿的值。对延迟进行补偿的情况下，设置正的值。关于详细内容，请参阅下述章节。 ☞ 1522页 补偿时间(CompensationTime)
RecordMode	锁存模式	INT (MC_RECORD_MODE)	启动时	0~2	0	设置数据的锁存方法。 • 0: 单发模式(OneShot) • 1: 指定次数模式(RecordCount) • 2: 环形缓冲模式(RingBuffer) 关于详细内容，请参阅下述章节。 ☞ 1523页 锁存模式(RecordMode)
RecordCount	锁存次数	WORD (UINT)	启动时	1~65535	1	设置数据的锁存次数。关于详细内容，请参阅下述章节。 ☞ 1523页 锁存次数(RecordCount)
OutputBuffer	锁存数据存储目标	TARGET_REF	启动时	—	—	设置锁存数据的存储目标。省略的情况下，仅向锁存位置(RecordedPosition)输出锁存数据。关于详细内容，请参阅下述章节。 ☞ 1524页 锁存数据存储目标(OutputBuffer)

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	完成	BOOL	FALSE	控制完成时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_TouchProbe(触摸探针有效)时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	由于错误及多重启动等，MC_TouchProbe(触摸探针有效)的执行中断时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。关于错误代码的详细内容，请参阅下述手册。 ☞ 所使用的控制器的用户手册
PositionData	当前值监视数据	LREAL	0.0	存储锁存数据的当前值。
RecordedPosition	锁存位置	LREAL	0.0	存储发生触发事件时的当前值。多次锁存的情况下，将覆盖为最新数据。
RecordedCounter	锁存次数	WORD (UINT)	0	存储锁存的次数。
TouchProbeID	探针ID	WORD (UINT)	0	存储MC_TouchProbe(触摸探针有效)固有的ID。

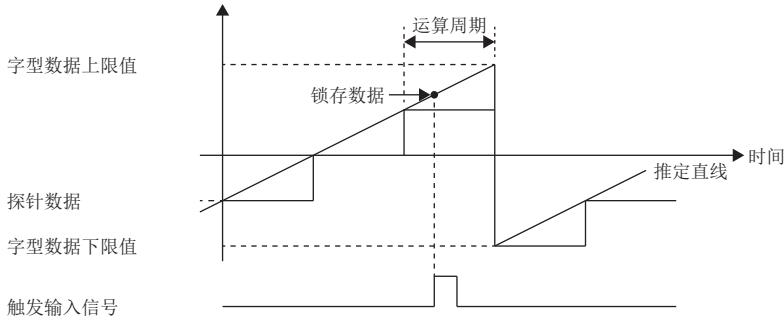
功能

- 在触发输入信号(TriggerInput)中指定的信号的上升沿、下降沿或两方向对锁存数据进行推定计算。
- 窗口有效(WindowOnly)为TRUE的情况下, 确认发生触发事件时的锁存数据位于从下限位置(FirstPosition)到上限位置(LastPosition)的范围内。超出范围的情况下, 数据不锁存。
- 根据锁存模式(RecordMode), 将锁存数据存储到锁存位置(RecordedPosition)及锁存数据存储目标(OutputBuffer)中设置的存储目标中, 并更新锁存次数(RecordedCounter)。
- 对运算周期期间的探针数据进行推定计算。将在输入了触发输入信号(TriggerInput)的时机进行了推定计算的值作为锁存数据。如下图所示计算值。

■探针数据为WORD型的情况下

- 使用外部信号高精度输入的情况下

触发输入时机将变为设备站检测出信号的时间。



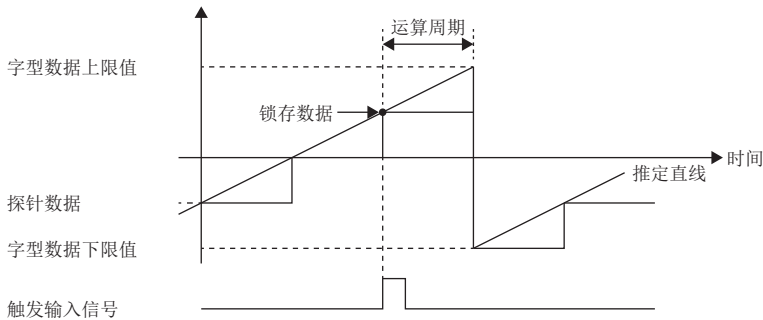
* 关于外部信号选择的高精度输入的设置方法, 请参阅下述手册。

☞所使用的控制器的用户手册

* 关于信号检测时间的精度, 请参阅所使用的设备的手册。

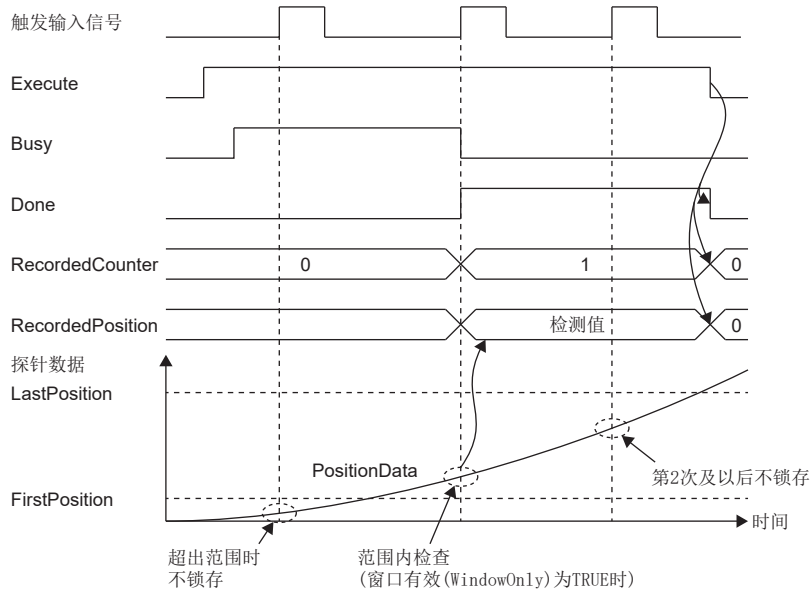
- 不使用外部信号高精度输入的情况下

触发输入时机将变为运算周期。

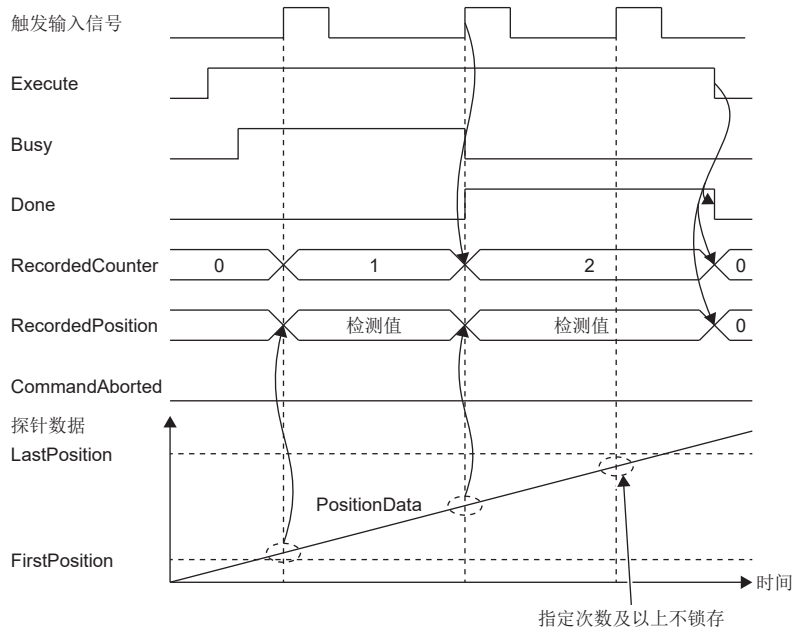


■时序图

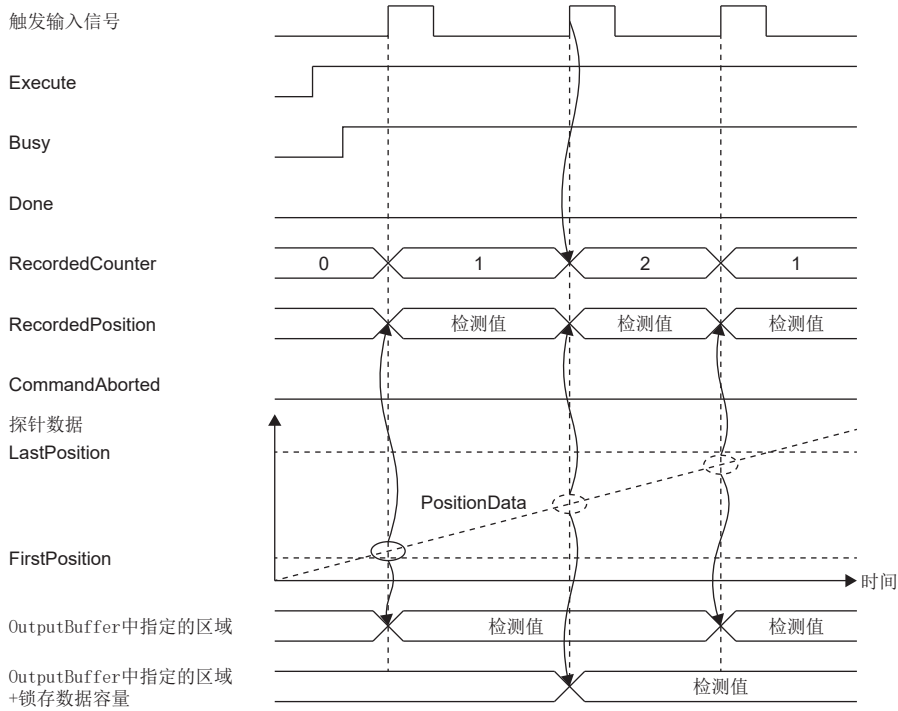
- 正常完成的情况下
- 锁存模式(RecordMode)为“0: 单发模式(OneShot)”的情况下



- 锁存模式(RecordMode)为“1: 指定次数模式(RecordCount)”的情况下
- 锁存次数(RecordCount)为2次的情况下



• 锁存模式(RecordMode)为“2: 环形缓冲模式(RingBuffer)”的情况下
锁存次数(RecordCount)为2次的情况下



• 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■触发输入信号(TriggerInput)

在MC_TRIGGER_REF结构体中设置触发输入信号。关于MC_TRIGGER_REF结构体，请参阅下述章节。

☞ 1426页 MC_TRIGGER_REF[M]

触发输入信号(TriggerInput)中设置的MC_TRIGGER_REF结构体的触发信号(Signal)的设置范围如下所示。

触发信号(Signal)的类型将为SIGNAL_SELECT结构体。关于SIGNAL_SELECT结构体，请参阅下述章节。

☞ 1431页 SIGNAL_SELECT(信号选择)

结构体	变量名	类型	设置范围
SIGNAL_SELECT(信号选择)	Source(信号)	TARGET_REF ☞ 1428页 TARGET_REF(输入信号)	<ul style="list-style-type: none"> ■类型 • BOOL ■数据类型 • [OBJ]*1*2 • [LINK] • [VAR] • [DEV] • [CONST]
	Detection(信号检测方法)	INT (MC_SIGNAL_LOGIC) ☞ 1459页 MC_SIGNAL_LOGIC[M]	<ul style="list-style-type: none"> • 2: FALSE→TRUE(上升沿)时检测(RisingEdge) • 3: TRUE→FALSE(下降沿)时检测(FallingEdge) • 4: 上升沿/下降沿时检测(BothEdges)
	CompensationTime(补偿时间)	LREAL	-5.0~5.0[s]
	FilterTime(筛选时间)	LREAL	0.0~5.0[s]

*1 在省略了对象修饰的情况下，参照轴信息(Axis)的轴的对象。但是，轴信息(Axis)的轴不具有站地址，或站地址未设置的情况下，将变为外部信号选择不正确(错误代码：1B31H、1B32H、1BD6H、1BD7H)。在对象修饰中指定了站地址的情况下，轴信息(Axis)将被忽略。

要参照的站轴模拟功能有效且已设置站地址的情况下，将参照正在进行模拟的对象。

使用外部信号高精度输入时，应将Detection(信号检测方法)设置为4：上升沿/下降沿时检测(BothEdges)。

*2 可以使用外部信号高精度输入。关于详细内容，请参阅下述手册。

☞ 所使用的控制器的用户手册

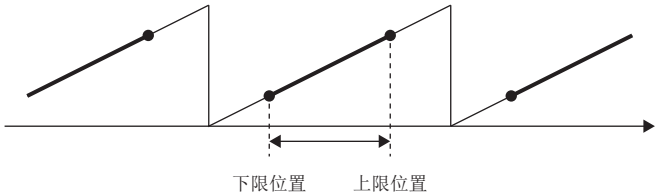
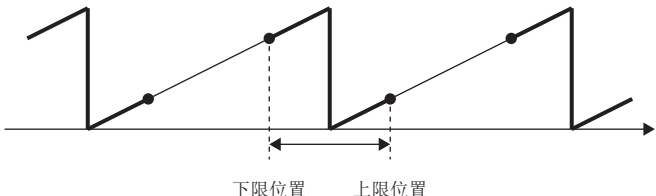
信号名(TriggerInput.Signal.Source.Target)中使用了不能使用的信号的情况下，将输出外部信号选择字符串不正确(错误代码：1B05H)、外部信号选择不正确(错误代码：1B31H、1B32H、1BD6H、1BD7H)、插件SignalIO内部异常(错误代码：1B33H、1BD8H)的任意一个。

要点

使用外部信号高精度输入时，应将筛选时间(TriggerInput.Signal.FilterTime)设置为“0”。设置了“0”以外的情况下，将无法正确获取信号的检测时间。在设备中存在输入滤波器的情况下，应使用设备的滤波器。在设备中不存在输入滤波器的情况下，应同时使用触摸探针有效(MC_TouchProbe)的窗口有效(WindowOnly)、下限位置(FirstPosition)、上限位置>LastPosition)以防止噪声导致的触发信号的误检测。

■窗口有效(WindowOnly)/下限位置(FirstPosition)/上限位置>LastPosition)

窗口有效(WindowOnly)为有效(TRUE)的情况下，只有探针数据位于从下限位置(FirstPosition)到上限位置>LastPosition)的范围内才检测出触发。无效(FALSE)的情况下对全范围进行触摸探针处理。

下限位置(FirstPosition)/上限位置>LastPosition)的关系	触摸探针处理
下限位置(FirstPosition) ≤ 上限位置>LastPosition)	<p>探针数据在下限位置(FirstPosition)及以上且上限位置>LastPosition)及以下时，进行触摸探针处理。</p> 
下限位置(FirstPosition) > 上限位置>LastPosition)	<p>探针数据小于上限位置>LastPosition)或大于下限位置(FirstPosition)时，进行触摸探针处理。</p> 

■探针数据 (ProbeData)

在TARGET_REF结构体中设置要锁存的数据。关于TARGET_REF结构体，请参阅下述章节。

☞ 1428页 TARGET_REF (输入信号)

可使用的数据类型如下所示。

数据类型	内容
[AXIS]	参照轴信息 (Axis) 的轴数据。 需要设置对象修饰 (@Position等)。
[VAR]	轴信息 (Axis) 将被忽略。
[DEV]	
[OBJ] (已映射到循环数据中)	<ul style="list-style-type: none"> 在对象修饰中设置了站地址的情况下，轴信息 (Axis) 将被忽略。 在省略了对象修饰的情况下，参照轴信息 (Axis) 的轴的对象。但是，轴信息 (Axis) 的轴不具有站地址，或站地址未设置的情况下，将发生错误。 要参照的站轴模拟功能有效且已设置站地址的情况下，将参照正在进行模拟的对象。

信号名 (TARGET_REF.Target) 中使用了不能使用的信号的情况下，将输出外部信号选择字符串不正确 (错误代码: 1B05H)、外部信号选择不正确 (错误代码: 1B31H、1B32H、1BD6H、1BD7H)、插件SignalIO内部异常 (错误代码: 1B33H、1BD8H) 的任意一个。

• 数据将更新为环形计数器。环形计数器的上限值、下限值根据探针数据 (ProbeData) 的数据类型、数据类型而有所不同。

数据类型	内容
[AXIS]	<ul style="list-style-type: none"> 在对象修饰中设置了“@Position”的情况下 <ul style="list-style-type: none"> 将变为轴的环形计数器上限值、环形计数器下限值。 在对象修饰中设置了“@CumulativePos”的情况下 <ul style="list-style-type: none"> 将变为轴的定位范围上限值、定位范围下限值。
[VAR]	将变为数据类型的最大值、最小值。
[DEV]	<例>
[OBJ]	在 [VAR] 中设置了字型标签的情况下。 <ul style="list-style-type: none"> 环形计数器上限值: 65535 环形计数器下限值: 0

限制事项

设置时应使每1运算周期的变化量满足下述公式。未满足时探针数据的实际变化量与锁存的变化量可能会不一致。

$$1\text{运算周期的变化量} < \frac{|(\text{环形计数器最大值}) - (\text{环形计数器最小值}) + 1|}{2}$$

■补偿时间(CompensationTime)

补偿触摸探针处理的延迟时间。除触发输入信号的补偿时间以外，对触摸探针处理固有的延迟等进行补偿的情况下进行此设置。补偿延迟的情况下应设置正的值，补偿提前的情况下应设置负的值。

合计的补偿时间 = 补偿时间(CompensationTime) - 补偿时间(TriggerInput.Signal.CompensationTime)

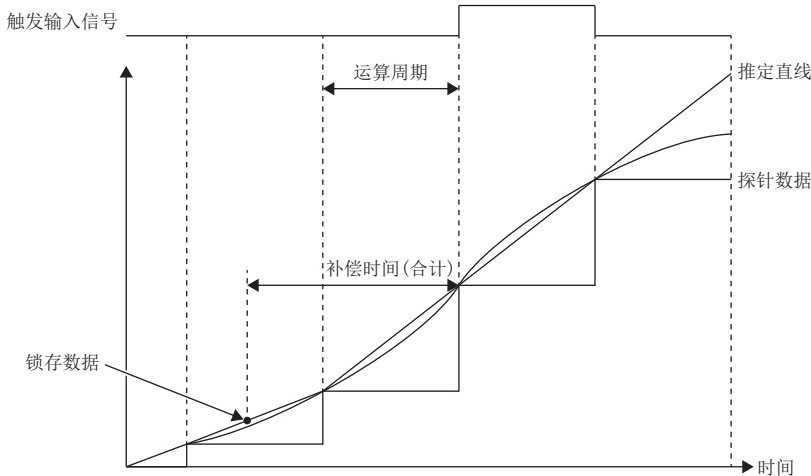
要点

对延迟时间进行补偿时，锁存数据的推定计算中使用过去值。因此，在从执行中(Busy)为TRUE的时机开始经过了补偿时间的时刻，精度变为最大。此外，根据补偿时间与运算周期，将消耗下述存储器。

• 存储器消耗量 = 24字节 × (合计的补偿时间 ÷ 运算周期)

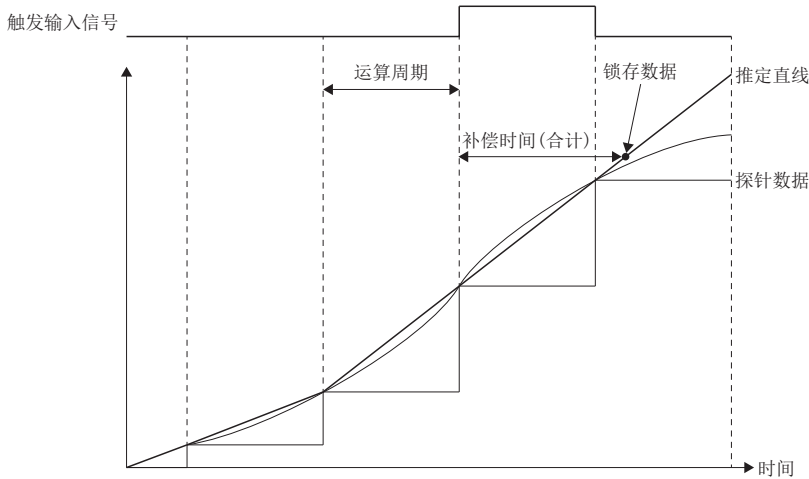
- 对延迟进行补偿的情况下(合计的补偿时间为正值的情况下)

根据从锁存的时机到补偿时间前的运算周期期间的数据进行推定计算。



- 对提前进行补偿的情况下(合计的补偿时间为负值的情况下)

根据锁存的时机的运算周期期间的数据进行推定计算。



■锁存模式 (RecordMode)

设置数据的锁存方法。锁存方法如下所示。

设定值	内容
0: 单发模式 (OneShot)	仅存储1次锁存数据。
1: 指定次数模式 (RecordCount)	<ul style="list-style-type: none"> 按锁存次数 (RecordCount) 的设定值存储锁存数据。 向锁存数据存储目标 (OutputBuffer) 中指定的存储目标, 以指定的存储目标为起始, 将锁存数据存储到每次锁存时按锁存数据容量移位的区域中。 在锁存次数 (RecordCount) 的锁存完成前停止触摸探针功能的情况下, 应执行MC_AbortTrigger (触摸探针无效)。
2: 环形缓冲模式 (RingBuffer)	<ul style="list-style-type: none"> 将锁存数据存储到相当于锁存次数 (RecordCount) 的设定值的环形缓冲中。始终进行锁存动作。 向锁存数据存储目标 (OutputBuffer) 中指定的存储目标, 以指定的存储目标为起始, 将锁存数据存储到每次锁存时按锁存数据容量移位, 且相当于锁存次数 (RecordCount) 的环形缓冲中。 在锁存次数 (RecordCount) 的锁存完成前停止触摸探针功能的情况下, 应执行MC_AbortTrigger (触摸探针无效)。

- 根据锁存模式 (RecordMode) 的设置, 在锁存次数 (RecordedCounter) 及锁存位置 (RecordedPosition) 中存储下述值。

锁存模式 (RecordMode)	锁存次数 (RecordedCounter)	锁存位置 (RecordedPosition)
0: 单发模式 (OneShot)	根据锁存动作变为“1”。	仅存储1次锁存数据。
1: 指定次数模式 (RecordCount)	每次锁存动作时将“+1”。	每次锁存动作时存储最新的锁存数据。
2: 环形缓冲模式 (RingBuffer)	每次锁存动作时将“+1”。 大于锁存次数 (RecordCount) 的情况下, 将返回到“0”。	每次锁存动作时存储最新的锁存数据。

■锁存次数 (RecordCount)

根据锁存模式 (RecordMode) 的设置, 在锁存次数 (RecordCount) 中设置下述值。

锁存模式 (RecordMode)	锁存次数 (RecordCount)
0: 单发模式 (OneShot)	忽略。
1: 指定次数模式 (RecordCount)	设置数据的锁存次数。设置为超出范围的情况下, 将发生锁存次数不正确 (错误代码: 1B8AH)。
2: 环形缓冲模式 (RingBuffer)	设置环形缓冲的次数。设置为超出范围的情况下, 将发生锁存次数不正确 (错误代码: 1B8AH)。

■锁存数据存储目标(OutputBuffer)

在TARGET_REF结构体中设置锁存的数据的存储目标。关于TARGET_REF结构体，请参阅下述章节。

☞ 1428页 TARGET_REF(输入信号)

- 可使用的数据类型为[VAR]及[DEV]。
- 由于锁存的数据将作为LREAL型存储，因此将LREAL型以外的数据设置为TARGET_REF结构体的对象(Target)的情况下，应明确在数据的类型中设置“(LREAL)”。
- 锁存的数据的存储目标如下所示。

锁存模式(RecordMode)	存储目标
0: 单发模式(OneShot)	锁存数据存储目标(OutputBuffer)中指定的区域
1: 指定次数模式(RecordCount)	下述的区域
2: 环形缓冲模式(RingBuffer)	锁存数据存储目标(OutputBuffer)中设置的区域 + 锁存次数(RecordedCounter) × 锁存数据容量

- 锁存数据存储目标(OutputBuffer)可以省略。省略的情况下，仅在锁存位置(RecordedPosition)中存储锁存数据。
- 将锁存数据存储目标(OutputBuffer)设置为TARGET_REF结构体的对象(Target)的示例如下所示。

例

下述设置的情况下

例	内容	设置
1	<ul style="list-style-type: none"> • 锁存模式(RecordMode)为“0: 单发模式(OneShot)”时。 • 锁存模式(RecordMode)为“1: 指定次数模式(RecordCount)”或“2: 环形缓冲模式(RingBuffer)”且锁存次数(RecordCount)为“1”时。 	<ul style="list-style-type: none"> • [VAR]OutputBufferData1*1
2	<ul style="list-style-type: none"> • 锁存模式(RecordMode)为“1: 指定次数模式(RecordCount)”或“2: 环形缓冲模式(RingBuffer)”且锁存次数(RecordCount)为“100”时。 	<ul style="list-style-type: none"> • [VAR]OutputBufferData2*2 • [VAR](LREAL)OutputBufferData3*3

*1 “OutputBufferData1”为LREAL型的实例

*2 “OutputBufferData2”为LREAL[0..99]型的实例

*3 “OutputBufferData3”为WORD[0..399]型的实例

- 锁存数据存储目标(OutputBuffer)中设置的元素数少于锁存次数(RecordCount)的情况下，将变为锁存数据存储目标不足(错误代码: 1BBFH)。此外，即使锁存动作时对软元件及标签的范围外进行了访问的情况下也将变为锁存数据存储目标不足”(错误代码: 1BBFH)。
- 各项目的设定值与动作的关系如下所示。

锁存模式(RecordMode)	锁存次数(RecordCount)	锁存数据存储目标(OutputBuffer)	检测动作	数据存储目标
0: 单发模式(OneShot)	设置范围内/设置范围外	省略	单发	锁存位置(RecordedPosition)
		设置		锁存位置(RecordedPosition)/缓冲
1: 指定次数模式(RecordCount)	设置范围外	省略/设置	错误	—
	设置范围内	省略	设置次数	锁存位置(RecordedPosition)
		设置		锁存位置(RecordedPosition)/缓冲
2: 环形缓冲模式(RingBuffer)	设置范围外	省略/设置	错误	—
	设置范围内	省略	始终	锁存位置(RecordedPosition)
		设置		锁存位置(RecordedPosition)/缓冲

注意事项

- 触摸探头的控制(触发输入信号的检测、锁存)与FB调用任务(扫描/恒定周期/事件)无关,而是以下述运算周期进行控制。触发输入信号的检测周期根据TriggerInput中设置的数据类型而有所不同。

类型	周期
[VAR]	第1运算周期
[DEV]	
[CONST]	
[OBJ]	对象站的通信周期
[LINK]	
[VAR]网络标签指定时	

探头数据(ProbeData)中设置的数据通过下述周期进行锁存。

类型	周期
[VAR]	第1运算周期
[DEV]	
[CONST]	
[OBJ]	对象站的通信周期
[LINK]	
[VAR]网络标签指定时	
[AXIS]	对象轴的控制运算周期

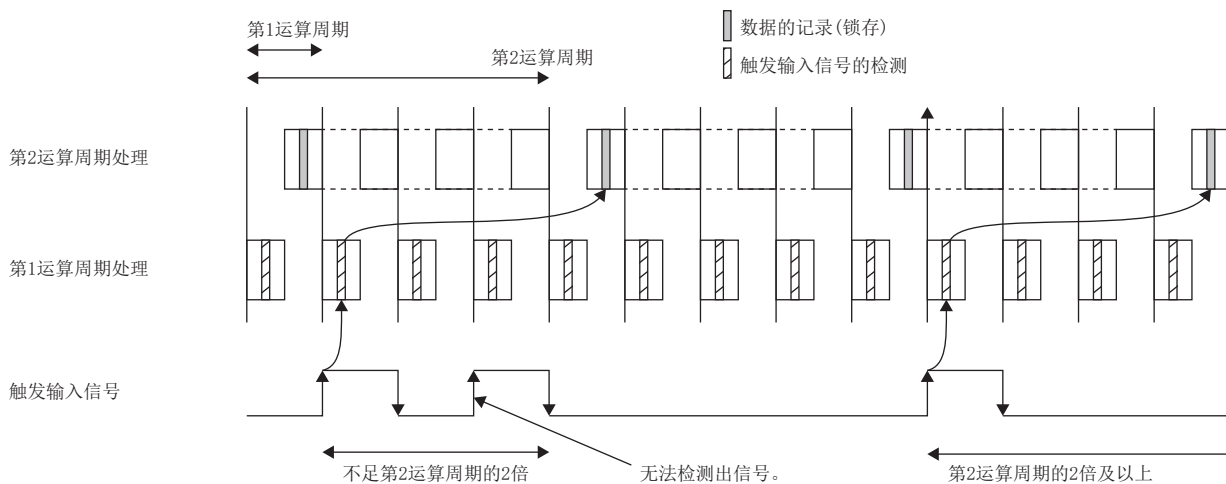
- 即使对象轴未连接的情况下及错误的情况下,也执行MC_TouchProbe(触摸探针有效)。
- 触发输入信号的TRUE、FALSE的各时间应长于下述时间。短时间情况下,无法正确检测信号。

外部信号高精度输入	触发输入信号的TRUE、FALSE的各时间
使用	<ul style="list-style-type: none"> • 设备站: 设备站的筛选时间 • 输入模块: 输入模块的输入响应时间
不使用	触发输入信号的检测周期

- 触发输入信号的间隔应至少为触发输入信号的检测周期与锁存周期中较长的一方的2倍。以比此时间短的间隔输入的情况下,无法正确检测信号。锁存前输入了多个触发输入信号的情况下,使用最初输入的信号。

例

触发输入信号的检测周期为第1运算周期,锁存周期为第2运算周期的情况下



程序示例

将触摸探针启动(bTouchProbe)置为TRUE, 每当运动系统内的触摸探针信号(G_bTouchProbeSignal)的上升沿时对轴1(Axis0001)的反馈位置进行推定计算, 并将其存储到锁存位置(leRecordedPosition)后, 将触摸探针无效启动(bAbortTrigger)置为TRUE, 停止锁存动作的程序示例如下所示。

■轴

轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■输入输出数据

数据名	数据类型	数据类型	源类型	源	信号检测方法
SignalData0001	触发信号	MC_TRIGGER_REF	全局标签	G_bTouchProbeSignal	FALSE→TRUE(上升沿)时检测(RisingEdge)

■使用的全局标签

标签名	数据类型	注释
G_bTouchProbeSignal	位	触摸探针信号

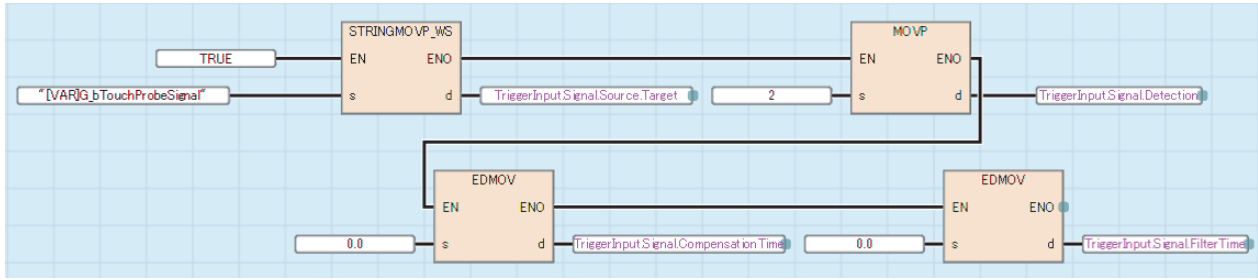
■使用的标签

标签名	数据类型	注释
MC_TouchProbe_1	MC_TouchProbe	触摸探针有效FB
bTouchProbe	位	触摸探针启动
leFirstPosition	双精度实数	下限位置
leLastPosition	双精度实数	上限位置
leCompensationTime	双精度实数	补偿时间
ProbeData	TARGET_REF	探针数据
uwRecordCount	字[无符号]/位串[16位]	锁存次数
OutputBuffer	TARGET_REF	锁存数据存储目标
bTPDone	位	触摸探针有效完成
bTPBusy	位	触摸探针有效执行中
bTPCommandAborted	位	触摸探针有效执行中断
bTPError	位	触摸探针有效错误
uwTPErrorID	字[无符号]/位串[16位]	触摸探针有效错误代码
lePositionData	双精度实数	当前值监视数据
leRecordedPosition	双精度实数	锁存位置
uwRecordedCounter	字[无符号]/位串[16位]	锁存次数
uwTouchProbeID	字[无符号]/位串[16位]	探针ID
leLatchPos_array	双精度实数(0..1)	锁存位置存储用变量
MC_AbortTrigger_1	MC_AbortTrigger	触摸探针无效FB
bAbortTrigger	位	触摸探针无效启动
bATDone	位	触摸探针无效完成
bATBusy	位	触摸探针无效处理中
bATCommandAborted	位	触摸探针无效执行中断
bATError	位	触摸探针无效错误
uwATErrorID	字[无符号]/位串[16位]	触摸探针无效错误代码
TriggerInput*1	MC_TRIGGER_REF	触发输入信号

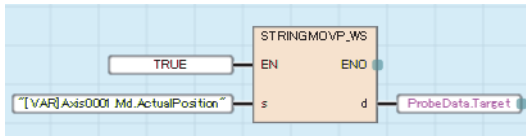
*1 仅FBD/LD的程序使用(为了在FBD/LD的程序中使用输入输出数据需要此项)

■FBD/LD程序

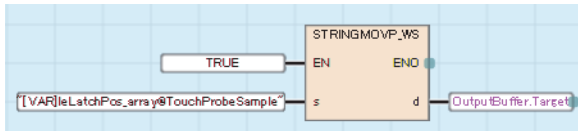
- 触发输入信号(输入输出数据)



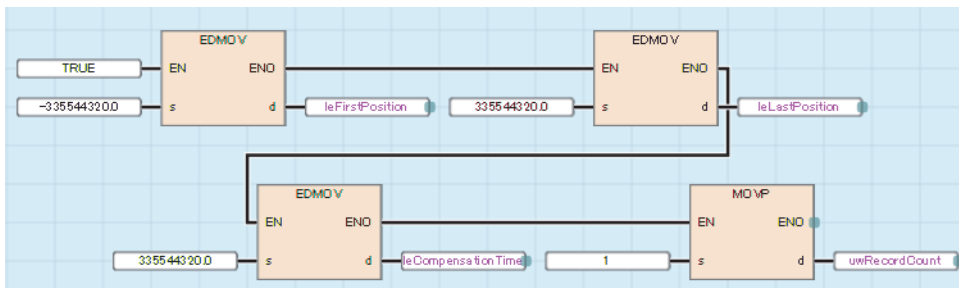
- 锁存对象用数据



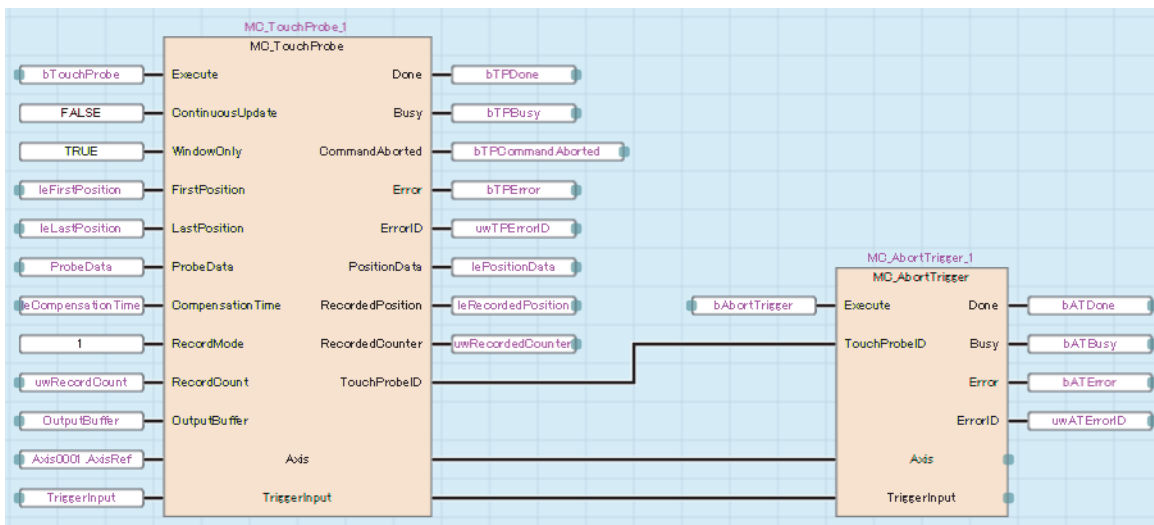
- 触摸探针处理



- 触摸探针用数据设置



- 触摸探针有效/无效



■ST程序

```
//-----锁存对象用数据-----
ProbeData.Target:= "[VAR]Axis0001.Md.ActualPosition";

//-----触摸探针处理-----
OutputBuffer.Target:= "[VAR]leLatchPos_array@TouchProbeSample";

//-----触摸探针用数据设置-----
leFirstPosition:= -335544320.0;
leLastPosition:= 335544320.0;
leCompensationTime:= 0.0;
uwRecordCount:= 1;

//-----触摸探针有效-----
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
);

//-----触摸探针无效-----
MC_AbortTrigger_1(
  Axis:= Axis0001.AxisRef ,
  TriggerInput:= SignalData0001 ,
  Execute:= bAbortTrigger ,
  TouchProbeID:= MC_TouchProbe_1.TouchProbeID ,
  Done=> bATDone ,
  Busy=> bATBusy ,
  Error=> bAError ,
  ErrorID=> uwAErrorID
);
```

45.12 触摸探针无效

MC_AbortTrigger

将执行中的锁存置为无效。

梯形图	FBD/LD	ST
		<pre> MC_AbortTrigger(Axis:= ?AXIS_REF?, TriggerInput:= ?MC_TRIGGER_REF?, Execute:= ?BOOL?, TouchProbeID:= ?WORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?); </pre>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	可省略	设置轴。 关于使用的变量(AxisName.AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName.AxisRef. (轴信息)
TriggerInput	触发输入信号	MC_TRIGGER_REF	启动时	—	可省略	设置触发输入信号。 关于详细内容，请参阅下述章节。 ☞ 1530页 触发输入信号(TriggerInput)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	启动	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_AbortTrigger(触摸探针无效)。
TouchProbeID	触摸探针ID	WORD(UINT)	启动时	0~65535	0	设置无效的触摸探针的固有ID。 设置无效的MC_TouchProbe(触摸探针有效)的执行指令(Execute)为上升沿后的探针ID(TouchProbeID)中存储的触摸探针ID。

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	控制完成时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_AbortTrigger(触摸探针无效)时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 ☞ 所使用的控制器的用户手册

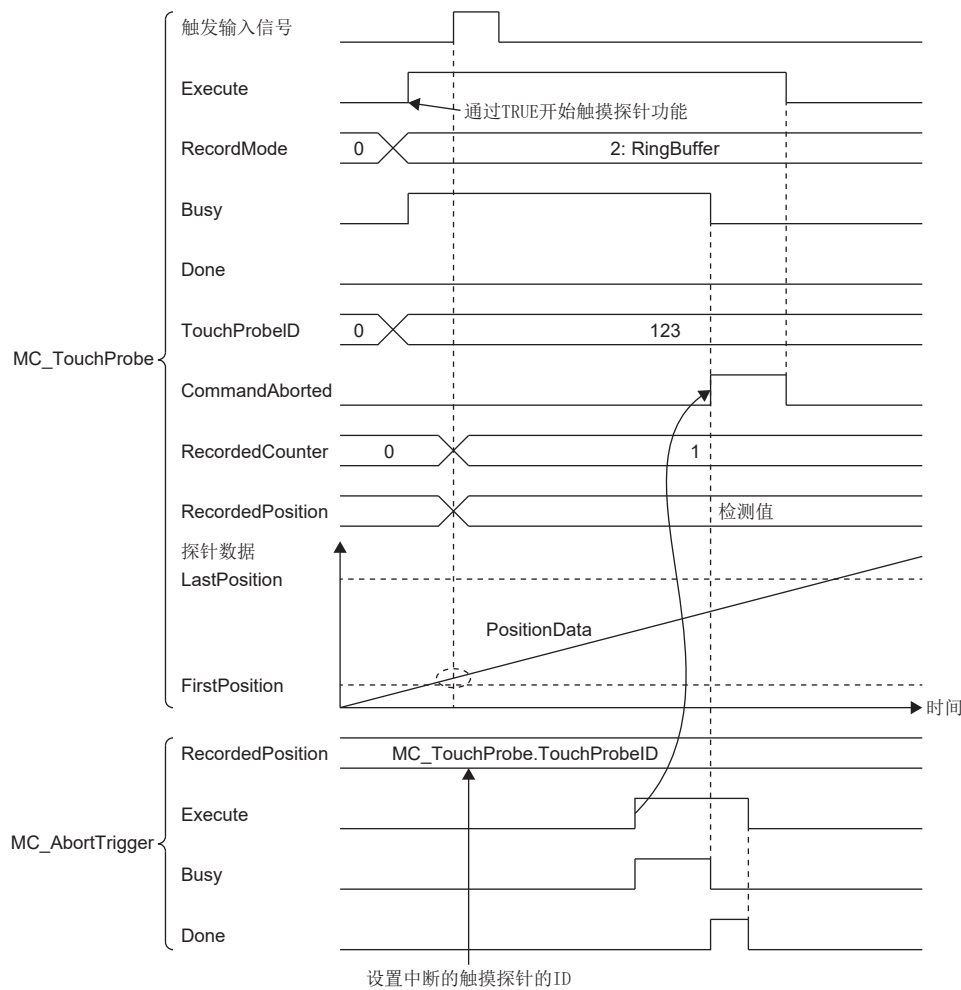
功能

- 将触摸探针ID(TouchProbeID)中设置的MC_TouchProbe(触摸探针有效)置为无效。
- 执行MC_AbortTrigger(触摸探针无效)时，触摸探针ID(TouchProbeID)中设置的MC_TouchProbe(触摸探针有效)未执行动作的情况下，执行完成(Done)将立即变为TRUE。
- 由于忽略轴信息(Axis)、触发输入信号(TriggerInput)，因此可以省略。

■时序图

- 正常完成的情况下

锁存模式 (RecordMode) 为 “2: 环形缓冲模式(RingBuffer)” 的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■触发输入信号(TriggerInput)

在MC_TRIGGER_REF结构体中设置触发输入信号。关于MC_TARGET_REF结构体，请参阅下述章节。

☞ 1426页 MC_TRIGGER_REF[M]

触发输入信号(TriggerInput)中设置的MC_TRIGGER_REF结构体的触发信号(Signal)的设置范围如下所示。

触发信号(Signal)的类型将为SIGNAL_SELECT结构体。关于SIGNAL_SELECT结构体，请参阅下述章节。

☞ 1431页 SIGNAL_SELECT(信号选择)

结构体	变量名	类型	设置范围
SIGNAL_SELECT(信号选择)	Source(信号)	TARGET_REF ☞ 1428页 TARGET_REF(输入信号)	<ul style="list-style-type: none"> ■类型 • BOOL ■数据类型 • [VAR] • [DEV] • [CONST]
	Detection(信号检测方法)	INT (MC_SIGNAL_LOGIC) ☞ 1459页 MC_SIGNAL_LOGIC[M]	<ul style="list-style-type: none"> • 2: FALSE→TRUE(上升沿)时检测(RisingEdge) • 3: TRUE→FALSE(下降沿)时检测(FallingEdge) • 4: 上升沿/下降沿时检测(BothEdges)
	CompensationTime(补偿时间)	LREAL	-5.0~5.0[s]
	FilterTime(筛选时间)	LREAL	0.0~5.0[s]

程序示例

将触摸探针启动(bTouchProbe)置为TRUE，每当运动系统内的触摸探针信号(G_bTouchProbeSignal)的上升沿时对轴1(Axis0001)的反馈位置进行推定计算，并将其存储到锁存位置(leRecordedPosition)后，将触摸探针无效启动(bAbortTrigger)置为TRUE，停止锁存动作的程序示例如下所示。

■轴

轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■输入输出数据

数据名	数据类型	数据类型	源类型	源	信号检测方法
SignalData0001	触发信号	MC_TRIGGER_REF	全局标签	G_bTouchProbeSignal	FALSE→TRUE(上升沿)时检测(RisingEdge)

■使用的全局标签

标签名	数据类型	注释
G_bTouchProbeSignal	位	触摸探针信号

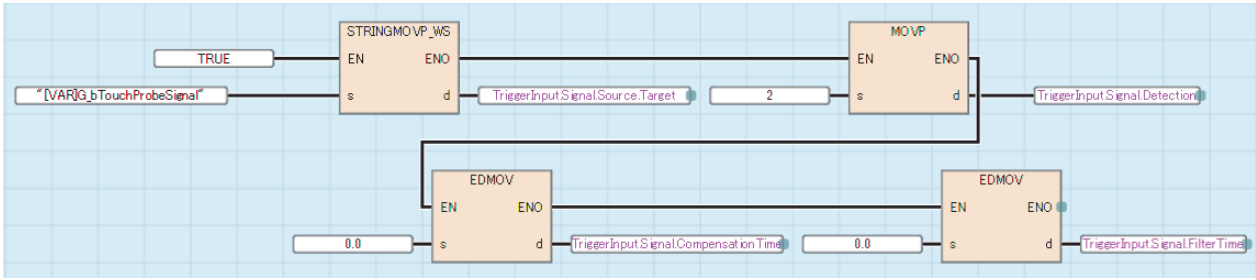
■使用的标签

标签名	数据类型	注释
MC_TouchProbe_1	MC_TouchProbe	触摸探针有效FB
bTouchProbe	位	触摸探针启动
leFirstPosition	双精度实数	下限位置
leLastPosition	双精度实数	上限位置
leCompensationTime	双精度实数	补偿时间
ProbeData	TARGET_REF	探针数据
uwRecordCount	字[无符号]/位串[16位]	锁存次数
OutputBuffer	TARGET_REF	锁存数据存储目标
bTPDone	位	触摸探针有效完成
bTPBusy	位	触摸探针有效执行中
bTPCommandAborted	位	触摸探针有效执行中断
bTPError	位	触摸探针有效错误
uwTPErrorID	字[无符号]/位串[16位]	触摸探针有效错误代码
lePositionData	双精度实数	当前值监视数据
leRecordedPosition	双精度实数	锁存位置
uwRecordedCounter	字[无符号]/位串[16位]	锁存次数
uwTouchProbeID	字[无符号]/位串[16位]	探针ID
leLatchPos_array	双精度实数(0..1)	锁存位置存储用变量
MC_AbortTrigger_1	MC_AbortTrigger	触摸探针无效FB
bAbortTrigger	位	触摸探针无效启动
bATDone	位	触摸探针无效完成
bATBusy	位	触摸探针无效处理中
bATCommandAborted	位	触摸探针无效执行中断
bATError	位	触摸探针无效错误
uwATErrorID	字[无符号]/位串[16位]	触摸探针无效错误代码
TriggerInput*1	MC_TRIGGER_REF	触发输入信号

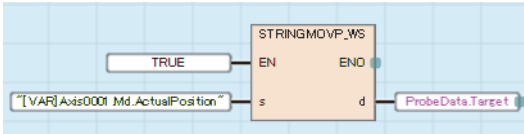
*1 仅FBD/LD的程序使用(为了在FBD/LD的程序中使用输入输出数据需要此项)

■FBD/LD程序

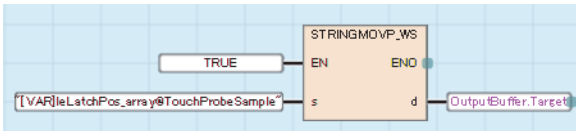
- 触发输入信号(输入输出数据)



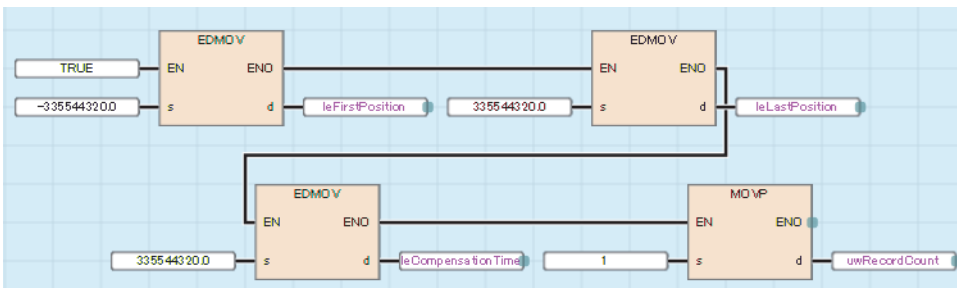
- 锁存对象用数据



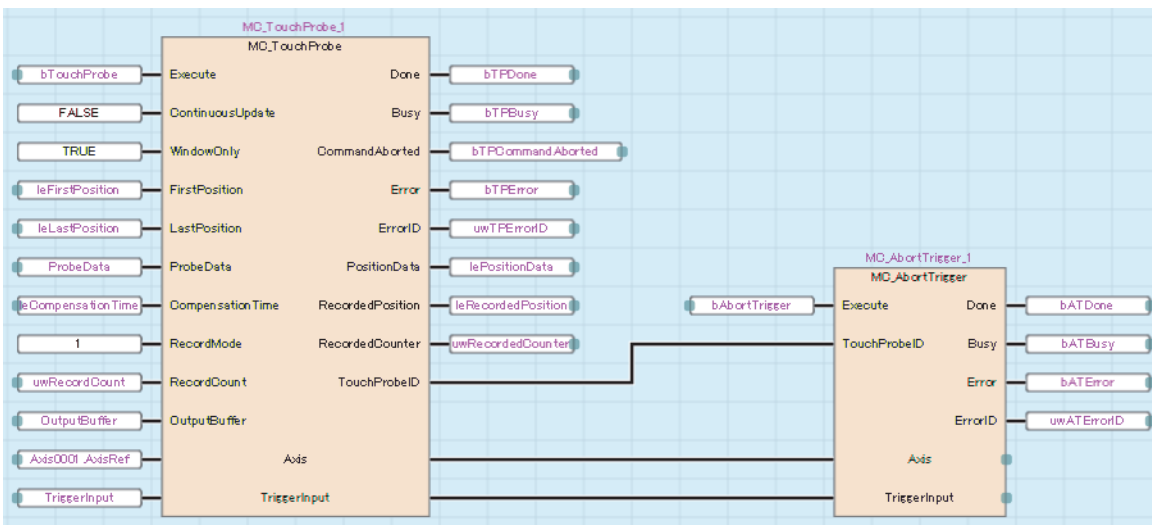
- 触摸探针处理



- 触摸探针用数据设置



- 触摸探针有效/无效



■ST程序

```
//-----锁存对象用数据-----
ProbeData.Target:= "[VAR]Axis0001.Md.ActualPosition";

//-----触摸探针处理-----
OutputBuffer.Target:= "[VAR]leLatchPos_array@TouchProbeSample";

//-----触摸探针用数据设置-----
leFirstPosition:= -335544320.0;
leLastPosition:= 335544320.0;
leCompensationTime:= 0.0;
uwRecordCount:= 1;

//-----触摸探针有效-----
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
);

//-----触摸探针无效-----
MC_AbortTrigger_1(
  Axis:= Axis0001.AxisRef ,
  TriggerInput:= SignalData0001 ,
  Execute:= bAbortTrigger ,
  TouchProbeID:= MC_TouchProbe_1.TouchProbeID ,
  Done=> bATDone ,
  Busy=> bATBusy ,
  Error=> bAError ,
  ErrorID=> uwAErrorID
);
```

45.13 凸轮表选择

MC_CamTableSelect

将指定的运算配置文件(凸轮数据)存储到展开区域中。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Master	主轴	AXIS_REF	启动时	—	可省略	使用MC_CamTableSelect(凸轮表选择)时, 设置输入输出No. (Master.StartIO)。轴No. (Master.AxisNo)将被忽略。 ☞ 1387页 AxisName, AxisRef. (轴信息)
Slave	从轴	AXIS_REF	启动时	—	可省略	使用MC_CamTableSelect(凸轮表选择)时, 无需设置输入输出No. (Slave.StartIO)。参阅输入输出No. (Master.StartIO)中设置的输入输出No.。轴No. (Slave.AxisNo)将被忽略。 ☞ 1387页 AxisName, AxisRef. (轴信息)
CamTable	凸轮表	MC_CAM_REF	启动时	—	不能省略	设置运算配置文件(凸轮数据)。执行MC_CamTableSelect(凸轮表选择)时, 无需设置配置文件ID(MC_CAM_ID.ProfileID)。参照 PROFILE_DATA.ID(MC_CAM_REF.ProfileData.ID)的设定值。但是, 无需设置输入输出No. (CamTable.ProfileData.ID.StartIO)。关于详细内容, 请参阅下述章节。 ☞ 1537页 凸轮表(CamTable)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	启动	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_CamTableSelect (凸轮表选择)。
Periodic	重复动作	BOOL	启动时	TRUE、FALSE	FALSE	设置执行运算配置文件(凸轮数据)的动作。 <ul style="list-style-type: none"> • FALSE: 单发动作 • TRUE: 重复动作 关于详细内容, 请参阅下述章节。 ☞ 1538页 重复动作(Periodic)
MasterAbsolute	主轴绝对坐标	BOOL	启动时	FALSE	FALSE	设置主轴的坐标。 <ul style="list-style-type: none"> • FALSE: 相对坐标 • TRUE: 不能设置 关于详细内容, 请参阅下述章节。 ☞ 1538页 主轴绝对坐标(MasterAbsolute) * 设置“TRUE”时将变为超出, MasterAbsolute 范围(错误代码: 1B94H)。
SlaveAbsolute	从轴绝对坐标	BOOL	启动时	TRUE、FALSE	FALSE	设置从轴的坐标。 <ul style="list-style-type: none"> • FALSE: 相对坐标 • TRUE: 绝对坐标 关于详细内容, 请参阅下述章节。 ☞ 1539页 从轴绝对坐标(SlaveAbsolute)
ExecutionMode	启动模式	INT (MC_EXECUTION_MODE)	启动时	0、1、3	0	设置执行MC_CamTableSelect (凸轮表选择)的时机。 <ul style="list-style-type: none"> • 0: 立即执行(mcImmediately) • 1: 等待完成后执行(mcQueued) • 3: 推测执行(mcSpeculatively) 关于详细内容, 请参阅下述章节。 ☞ 1539页 启动模式(ExecutionMode)

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	完成	BOOL	FALSE	控制完成时, 将变为TRUE。 动作完成时根据启动(Execute)的状态将变为如下所示。 ■启动(Execute)为TRUE的情况下 在将启动(Execute)置为FALSE之前将保持为TRUE不变。 ■启动(Execute)为FALSE的情况下 仅1周期为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_CamTableSelect (凸轮表选择)时, 将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时, 将返回错误代码。 关于错误代码的详细内容, 请参阅下述手册。 ☞所使用的控制器的用户手册
CamTableID	凸轮表ID	MC_CAM_ID	0	输出配置文件ID。

功能

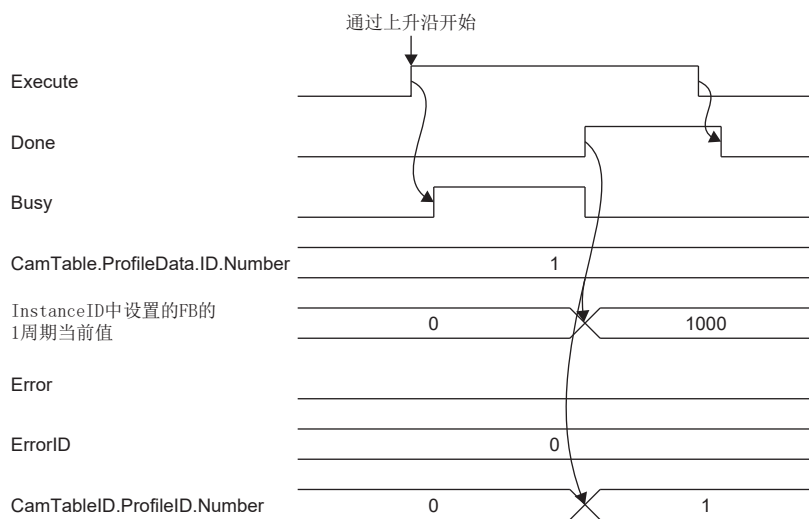
- 使用MC_CamTableSelect(凸轮表选择), 将PROFILE_DATA结构体的运算配置文件存储位置(Location)中设置的运算配置文件(凸轮数据)的文件展开到展开区域中。
- 展开到展开区域中的数据中, 将分配PROFILE_ID结构体的配置文件ID(Number)。

要点

- 关于运算配置文件(凸轮数据)的创建的详细内容, 请参阅下述内容。
📖 所使用的控制器的用户手册
- MC_CamTableSelect(凸轮表选择)通过运动服务处理进行。根据处理内容及运算配置文件, 展开有可能需要耗费一定时间。
- 作为MC_CamTableSelect(凸轮表选择)的输入变量, 存在有PROFILE_DATA结构体的配置文件ID(ID)、重复动作(Periodic)、主轴绝对坐标(MasterAbsolute)、从轴绝对坐标(SlaveAbsolute)的情况下, 忽略文件的设定值, 并参照输入变量的设置进行展开。(不覆盖文件的设定值。)

■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图, 请参阅下述章节。

📖 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■凸轮表(CamTable)

在MC_CAM_REF型的PROFILE_DATA结构体中设置运算配置文件。关于PROFILE_DATA结构体，请参阅下述章节。

☞ 1431页 PROFILE_DATA

参照PROFILE_DATA结构体的运算配置文件存储位置(Location)，对文件名称(FileName)及文件夹指定(Path)表示的文件的运算配置文件进行展开。

展开区域中，将分配以PROFILE_DATA结构体的配置文件ID(ID)表示的PROFILE_ID结构体的配置文件ID编号(Number)。

根据配置文件ID编号(Number)中设置的值，动作如下所示。

配置文件ID编号(Number)	动作
设置了“0”的情况下	自动分配空余ID，并将ID存储到配置文件ID编号(Number)中。 对于同一运算配置文件，多次执行配置文件ID编号(Number)为“0”的MC_CamTableSelect(凸轮表选择)时，将被展开到多个不同的展开区域中。 没有空余ID的情况下，将变为运算配置文件ID不足(错误代码：1BA0H)，不进行展开处理。
设置了“1~60000”的情况下	在不改变设置的ID的状况下进行展开。 指定了已展开的ID的情况下，将覆盖展开数据。

展开时的配置文件ID编号(Number)的值与展开目标区域的关系如下所示。

配置文件ID编号(Number)的值		展开目标区域
MC_CamTableSelect(凸轮表选择)执行前	MC_CamTableSelect(凸轮表选择)执行后	
0	存储1~60000	新建
1~60000		已经存在的情况下，将覆盖已展开的数据。 不存在的情况下，将分配新的展开区域。

- 功能固有型的情况下，在功能固有型的配置文件ID(MC_CAM_ID.ProfileID等)中存储与PROFILE_DATA.ID(MC_CAM_REF.ProfileData.ID等)相同的值。应在运算配置文件控制FB(MC_CamIn等)的输入变量中设置功能固有型的配置文件ID。

配置文件ID型的详细内容，请参阅下述手册。

☞ 所使用的控制器的用户手册

- 执行展开FB时，无需设置MC_CAM_REF.CamID.ProfileID。参照MC_CAM_REF.ProfileData.ID的设定值。

展开FB的一览，请参阅下述手册。

☞ 所使用的控制器的用户手册

- 运算配置文件存储位置(Location)中设置的运算配置文件不存在的情况下，将变为无运算配置文件(错误代码：1B8EH)，不进行展开处理。
- 运算配置文件存储位置(Location)中设置的运算配置文件已损坏，或格式有问题的情况下，将变为运算配置文件异常(错误代码：1B91H)，不进行展开处理。
- 运算配置文件存储位置(Location)中指定的运算配置文件的设定值超出范围的情况下，将发生运算配置文件不正确(错误代码：1B8FH)，不进行展开处理。

例

分辨率(第4行、第1列)超出范围的情况下，将文件内的行编号存储到数据No，列编号存储到项目No中。

详细信息1(程序错误信息)

实例名：(FB的实例名)

详细信息2(数据错误信息)

详细代码：1004H 运算配置文件的设定值超出范围

数据No：4

项目No：1

数据名：calc_profile/PD00/ProfileData0001.CSV

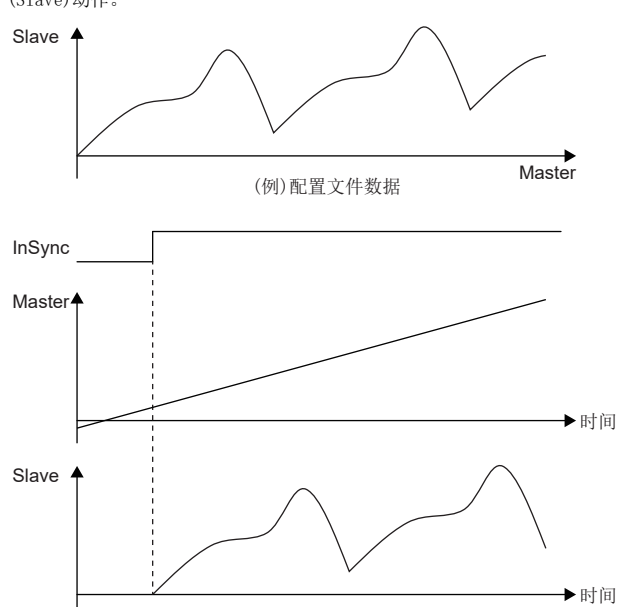
■重复动作(Periodic)

设置通过重复动作(Periodic)的运算配置文件控制执行时的动作。

设定值	内容
FALSE(单发动作)	执行至运算配置文件的终点为止时，结束控制。
TRUE(重复动作)	连续重复运算配置文件的执行。

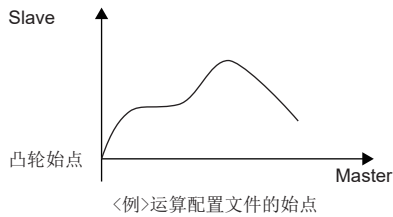
■主轴绝对坐标(MasterAbsolute)

设置通过主轴绝对坐标(MasterAbsolute)的运算配置文件控制执行时的动作。(输入单位为时间指定的情况下忽略。)

设定值	内容
FALSE(相对坐标)	<p>当前的主轴(Master)的值将成为运算配置文件的始点。同步中(InSync)变为TRUE时，将根据主轴(Master)的相对移动量使从轴(Slave)动作。</p>  <p>(例)配置文件数据</p>
TRUE	<p>无法设置TRUE。 设置TRUE时将变为超出MasterAbsolute范围(错误代码：1B94H)。</p>

■从轴绝对坐标(SlaveAbsolute)

设置通过从轴绝对坐标(SlaveAbsolute)的运算配置文件控制执行时的动作。



设定值	内容
FALSE(相对坐标)	<p>同步中(InSync)变为TRUE时,将以当前的值为基准使从轴(Slave)动作。</p> <p>运算配置文件的始点</p>
TRUE(绝对坐标)	<p>执行动作,以使同步中(InSync)变为TRUE的时刻的从轴(Slave)在运算配置文件的1周期开始时始终成为始点。 凸轮数据的1周期长等运算配置文件的主轴(Master)的值有限制,重复动作(Periodic)为“TRUE:重复动作”且运算配置文件的始点与终点中不同的情况下,将以1运算周期输出指令以确保在下一个1周期开始时刻返回到最初的从轴(Slave)。</p> <p>运算配置文件的始点</p>

■启动模式(ExecutionMode)

设置执行MC_CamTableSelect(凸轮表选择)的时机。

设定值	内容
0: 立即执行(mcImmediately)	<p>立即写入展开区域的内容。 有可能对执行中的控制带来影响。 但是,FB的执行中进行展开的情况下,运算配置文件的格式、分辨率不一致时,将变为运算配置文件操作中(错误代码:1B90H)。</p>
1: 等待完成后执行(mcQueued)	<p>等待执行中的FB的执行完成后展开。 多个FB处于等待状态的情况下,下一个要执行的FB将成为在优先级较高的任务中执行的FB。 相同的优先度的情况下,将按启动顺序执行。</p>
3: 推测执行(mcSpeculatively)	<p>将变为运算配置文件操作中(错误代码:1B90H),不更改展开区域。</p>

要点

如果没有正在执行的FB,则无论启动模式(ExecutionMode)的设置如何,都将立即执行。

访问展开区域的各FB执行中,各FB的执行中(Busy)将变为TRUE。应根据需要将启动模式(ExecutionMode)用于互锁。

注意事项

控制中改写展开区域的情况下，根据波形模式，有可能导致行程值急剧变化，给机械带来冲击。应创建运算配置文件，避免更改前后的波形不连续。

程序示例

将凸轮表选择指令(bCamTableSelect)置为TRUE，并将运算配置文件(凸轮数据1)展开到展开区域中，以分配配置文件ID的程序示例如下所示。

■轴

轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■运算配置文件

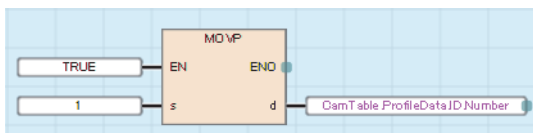
配置文件ID	标签名	数据类型	注释
1	ProfileData0001	MC_CAM_REF	凸轮数据1

■使用的标签

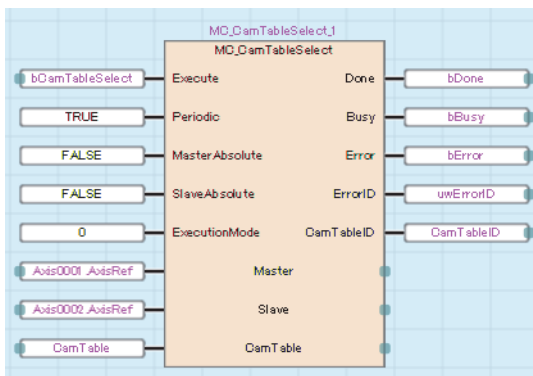
标签名	数据类型	注释
MC_CamTableSelect_1	MC_CamTableSelect	凸轮表选择FB
CamTable	MC_CAM_REF	凸轮表
bCamTableSelect	位	凸轮表选择指令
bDone	位	执行完成
bBusy	位	执行中
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码
CamTableID	MC_CAM_ID	凸轮表ID

■FBD/LD程序

- 运算配置文件(凸轮数据)设置



- 凸轮表选择



■ST程序

```
//-----运算配置文件(凸轮数据)设置-----  
CamTable.ProfileData:= ProfileData0001.ProfileData;  
  
//-----凸轮表选择-----  
MC_CamTableSelect_1(  
    CamTable:= CamTable ,  
    Execute:= bCamTableSelect ,  
    Periodic:= TRUE ,  
    MasterAbsolute:= FALSE ,  
    SlaveAbsolute:= FALSE ,  
    ExecutionMode:= MC_EXECUTION_MODE__mcImmediately ,  
    Done=> bDone ,  
    Busy=> bBusy ,  
    Error=> bError ,  
    ErrorID=> uwErrorID ,  
    CamTableID=> CamTableID  
);
```

45.14 1周期当前值更改

MCv_ChangeCycle

更改指定的运算配置文件控制FB的1周期当前值。

梯形图	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>

设置数据

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	启动	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MCv_ChangeCycle(1周期当前值更改)。
InstanceID	实例ID	INSTANCE_ID	启动时	1~65535	0	设置更改1周期当前值的运算配置文件控制FB的实例ID。 实例ID在“可编程控制器就绪(MotionSystem.Cd.SequenceReady)”由ON→OFF之前有效。 使用MCv_ChangeCycle(1周期当前值更改)时，设置输入输出编号(StartIO)。
Cycle	1周期当前值	LREAL[0..2]	启动时	—	0.0	设置更改的1周期当前值。 • 相对选择(Relative)为FALSE时，设置绝对位置。 • 相对选择(Relative)为TRUE时，设置相对距离。 关于详细内容，请参阅下述章节。 ☞ 1544页 1周期当前值(Cycle)
Relative	相对选择	BOOL	启动时	TRUE、FALSE	FALSE	设置更改的1周期当前值的方法。 • FALSE：绝对位置 • TRUE：相对距离 关于详细内容，请参阅下述章节。 ☞ 1545页 相对选择(Relative)
ExecutionMode	启动模式	INT (MC_EXECUTION_MODE)	启动时	0、1、3	0	设置执行MCv_ChangeCycle(1周期当前值更改)的时机。 • 0：立即执行(mcImmediately) 关于详细内容，请参阅下述章节。 ☞ 1546页 启动模式(ExecutionMode)
Options	选项	DWORD (HEX)	启动时	00000000H~00010000H	00000000H	将MCv_ChangeCycle(1周期当前值更改)的功能选项以位指定进行设置。 关于详细内容，请参阅下述章节。 ☞ 1547页 选项(Options)

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	完成	BOOL	FALSE	控制完成时，将变为TRUE。 动作完成时根据启动(Execute)的状态将变为如下所示。 ■启动(Execute)为TRUE的情况下 在将启动(Execute)置为FALSE之前将保持为TRUE不变。 ■启动(Execute)为FALSE的情况下 仅1周期为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_ChangeCycle(1周期当前值更改)时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	由于错误及多重启动等，MCv_ChangeCycle(1周期当前值更改)的执行中断时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 □□所使用的控制器的用户手册
CancelAccepted	取消受理	BOOL	FALSE	MCv_ChangeCycle(1周期当前值更改)受理了取消时将变为TRUE。

功能

- 将运算配置文件控制FB的1周期当前值更改为控制执行时设置的值。
- 在实例ID(InstanceID)中，设置运算配置文件控制FB(MC_CamIn等)的公开变量的实例ID(InstanceID)。
- 进行1周期当前值更改时运算配置文件控制FB(MC_CamIn等)的公开变量如下所示。

公开变量	更新值	备注
1周期当前值(InputPerCycle)	1周期当前值(Cycle)中设置的值	—
基准值(Reference)	与1周期当前值(Cycle)相当的行程值	为了固定输出值(OutputData)，更新基准值(Reference)。
输出值(OutputData)	不更新	—

要点

即使在MC_CamTableSelect(凸轮表选择)的从轴绝对坐标(SlaveAbsolute)为TRUE(绝对坐标)的情况下，也进行了1周期当前值更改的情况下，为了固定输出值(OutputData)将更新基准值(Reference)。

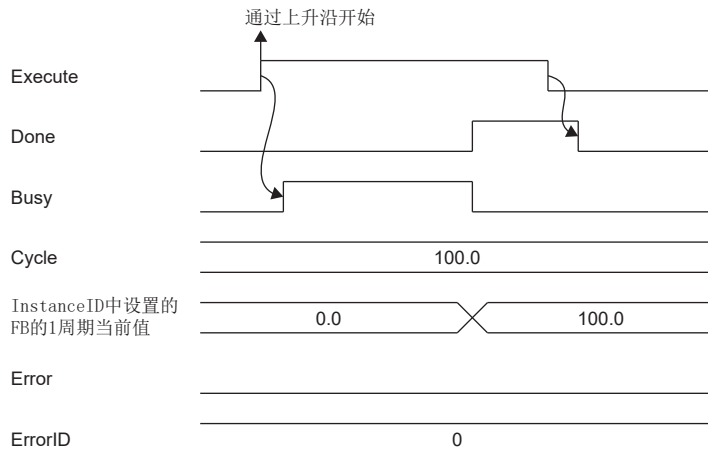
- 1周期当前值更改的方法如下。

方法	说明
MCv_ChangeCycle(1周期当前值更改)	更改为控制执行时指定的值。

- 对于更改的1周期当前值的指定，可以通过相对选择(Relative)选择“绝对指定”或“相对指定”。
- 如果对MC_CamIn(凸轮动作开始)以外的运算配置文件控制FB进行1周期当前值更改，则将发生不支持1周期当前值更改(错误代码：1BAFH)。

■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■1周期当前值(Cycle)

设置更改的1周期当前值。

1周期当前值(Cycle)中可设置的值根据相对选择(Relative)的设置而有所不同。应在下述设置范围内进行设置。设置了超出范围的值的的情况下，将变为MCv_ChangeCycle指令异常(错误代码：1BB9H)(详细代码：0001H)。

1周期当前值(Cycle)	相对选择(Relative)的设定值	1周期当前值(Cycle)的设置范围
Cycle[0]	FALSE(绝对位置)	$0.0 \leq \text{设定值} < 1\text{周期当前值}$
	TRUE(相对距离)	$-(1\text{周期长}) / 2 \leq \text{设定值} \leq (1\text{周期长}) / 2$
Cycle[1]、Cycle[2]	FALSE(绝对位置)	0.0
	TRUE(相对距离)	

■相对选择 (Relative)

根据相对选择 (Relative) 的设置，按下述方式更改1周期当前值 (Cycle) 的值。

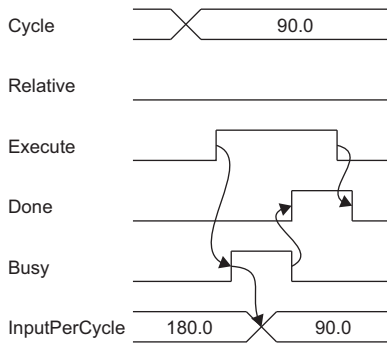
相对选择 (Relative) 的设定值	内容
FALSE (绝对位置)	将1周期当前值 (InputPerCycle)*1更改为在当前的1周期当前值 (InputPerCycle)*1中加上1周期当前值 (Cycle) 的值。
TRUE (相对距离)	将1周期当前值 (InputPerCycle)*1更改为1周期当前值 (Cycle) 的值。

*1 运算配置文件控制FB (MC_CamIn等) 的公开变量

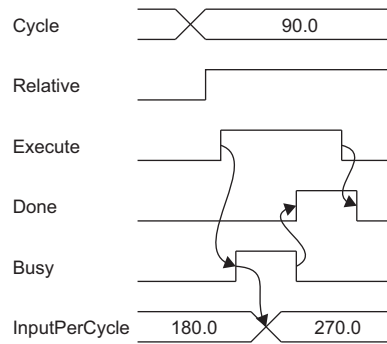
例

在公开变量的1周期当前值 (InputPerCycle) 为“180.0”处停止时，在1周期当前值 (Cycle) 中设置了“90.0”的1周期当前值更改的情况下

• 相对选择 (Relative) 为FALSE (绝对位置) 时



• 相对选择 (Relative) 为TRUE (相对距离) 时



通过1周期当前值更改的1周期当前值的过峰可否如下所示。

○：可以过峰， ×：不能过峰

相对选择 (Relative) 的设定值	1周期长与1周期当前值的移动量的关系	过峰可否
FALSE (绝对位置)	1周期长 > 1周期当前值的移动量	×
TRUE (相对距离)	1周期长 > 1周期当前值的移动量	○
	1周期长 ≤ 1周期当前值的移动量	○*1

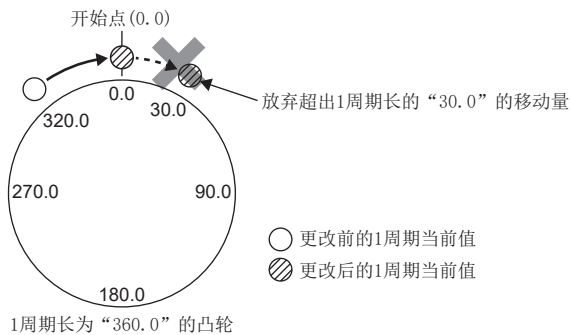
*1 根据MC_CamTableSelect (凸轮表选择) 中设置的重复动作 (Periodic) 的设定值，过峰时的动作有所不同。

重复动作 (Periodic) 的设定值	内容
FALSE (单发动作)	删除超出1周期长的移动量。
TRUE (重复动作)	超过1周期长更改1周期当前值。

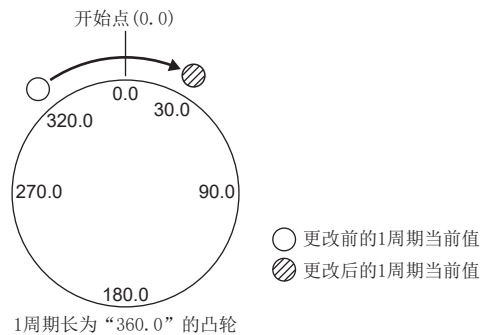
例

将1周期当前值 (Cycle) 从“320.0”更改为“30.0”的情况下

• 重复动作 (Periodic) 为FALSE (单发动作) 时



• 重复动作 (Periodic) 为TRUE (重复动作) 时



■启动模式 (ExecutionMode)

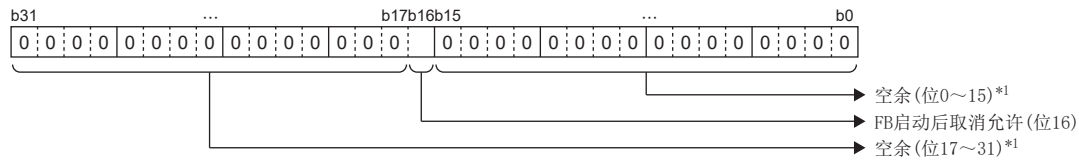
执行了MCv_ChangeCycle(1周期当前值更改)的情况下，更改1周期当前值的时机根据设置的启动模式 (ExecutionMode) 而有所不同。启动模式 (ExecutionMode) 的时机如下所示。

设定值	内容
0: 立即执行 (mcImmediately)	<p>即使运算配置文件控制FB(MC_CamIn等)的执行中(Busy)为TRUE，也通过启动(Execute)的上升沿检测立即执行。</p> <p><例></p> <p>MC_CamIn(凸轮动作开始)的执行中(Busy)为TRUE时，在1周期当前值(Cycle)中设置“100.0”，执行了1周期当前值更改的情况下(MC_CamIn(凸轮动作开始)的主轴(Master)的当前值不变化。)</p> <p>MC_CamIn</p> <ul style="list-style-type: none"> Execute Busy Active InSync EndOfProfile InputPerCycle: 0.0^{*1} → 100.0 Reference: 0.0^{*1} → -100.0 OutputData: 0.0^{*1} → (不变化) InstanceID: 0 → 1000 从轴的位置: 0.0^{*1} → (不变化) 从轴的状态 (AxisName.Md.AxisStatus): 4: Standstill → 7: SynchronizedMotion <p>MCv_ChangeCycle</p> <ul style="list-style-type: none"> Execute Done Busy CancelAccepted CommandAborted Cycle: 100.0 InstanceID: 0 → 1000^{*3} Error ErrorID: 0 <p>*1 由于主轴不动作，因此不变化。 *2 1周期当前值更改完成时完成(Done)将变为TRUE，执行中(Busy)将变为FALSE。 *3 手动设置执行1周期当前值更改的FB的实例ID。</p>

■选项(Options)

将MCv_ChangeCycle(1周期当前值更改)中使用的功能选项以位指定进行设置。

以位指定设置的内容如下所示。



*1 空余中应设置“0”。设置了“0”以外的情况下，将变为超出Options范围(错误代码: 1ABBH)。

位	名称	内容
16	FB启动后取消允许	设置MCv_ChangeCycle(1周期当前值更改)启动后是否允许1周期当前值更改的取消。 • 0: 不允许 • 1: 允许

- 根据FB启动后取消允许的设置，可以取消执行FB后处于待机状态的1周期当前值更改。
 - 通过启动(Execute)的下降沿检测，开始取消。
 - 取消的受理只有在执行中(Busy)为TRUE时才进行。
 - FB受理取消时，取消受理(CancelAccepted)将变为TRUE。
 - 取消完成时，执行中断(CommandAborted)将变为TRUE。
 - 执行了取消的情况下，不更改1周期当前值(InputPerCycle)。

程序示例

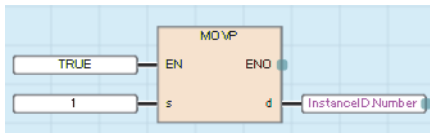
将1周期当前值更改指令 (bChangeCycle) 置为TRUE，并将实例ID (InstanceID) 为“1”的1周期当前值更改为“90.0”的程序示例如下所示。

■使用的标签

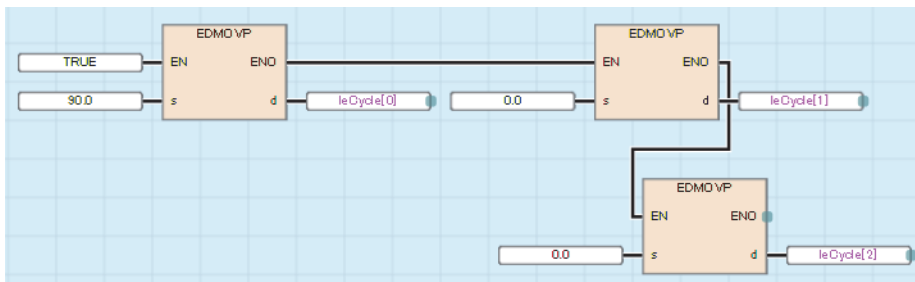
标签名	数据类型	注释
MCv_ChangeCycle_1	MCv_ChangeCycle	1周期当前值更改FB
bChangeCycle	位	1周期当前值更改指令
InstanceID	INSTANCE_ID	实例ID
leCycle	双精度实数(0..2)	1周期当前值
bDone	位	执行完成
bBusy	位	执行中
bCommandAborted	位	执行中断
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码
bCancelAccepted	位	取消受理

■FBD/LD程序

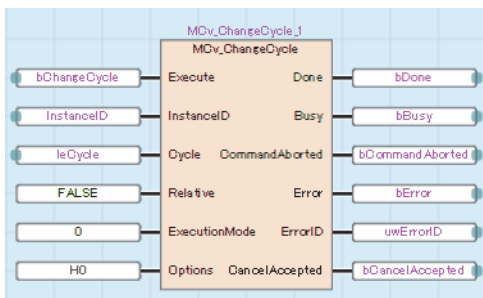
- 运算配置文件设置



- 1周期当前值更改用数据设置



- 1周期当前值更改



■ST程序

//-----运算配置文件设置-----

InstanceID.Number:= 1;

//-----1周期当前值更改用数据设置-----

leCycle[0]:= 90.0;

leCycle[1]:= 0.0;

leCycle[2]:= 0.0;

//-----1周期当前值更改-----

MCv_ChangeCycle_1(

Execute:= bChangeCycle ,

InstanceID:= InstanceID ,

Cycle:= leCycle ,

Relative:= FALSE ,

ExecutionMode:= MC_EXECUTION_MODE__mcImmediately ,

Options:= H00000000 ,

Done=> bDone ,

Busy=> bBusy ,

CommandAborted=> bCommandAborted ,

Error=> bError ,

ErrorID=> uwErrorID ,

CancelAccepted=> bCancelAccepted

);

45.15 全部轴可运行

MCv_AllPower

将所有轴切换为允许运行状态。

梯形图	FBD/LD	ST
		<pre>MCv_AllPower(Axis:= ?AXIS_REF?, Enable:= ?BOOL?, ServoON:= ?BOOL?, Busy=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	可省略	忽略设定值。

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Enable	有效	BOOL	始终	TRUE、FALSE	FALSE	设置为TRUE时轴控制将变为有效，且变为允许运行状态。 设置为FALSE时轴控制将变为无效，且解除允许运行状态。
ServoON	伺服ON请求	BOOL	始终	TRUE、FALSE	FALSE	TRUE时请求轴的伺服ON。

■输出变量

输出变量	名称	数据类型	初始值	内容
Busy	执行中	BOOL	FALSE	执行了MCv_AllPower(所有轴允许运行)时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 所使用的控制器的用户手册

功能

- 对所有轴的信息进行初始化，并切换为允许运行状态。
- 将有效 (Enable) 及伺服ON请求 (ServoON) 的输入置为TRUE时，将所有的轴切换为允许运行状态。
- 开始处理时执行中 (Busy) 将变为TRUE。
- 在MCv_AllPower (所有轴允许运行) 内发生了异常的情况下，错误 (Error) 将变为TRUE，并将错误代码存储到错误代码 (ErrorID) 中。关于错误代码的详细内容，请参阅下述手册。

📖 所使用的控制器的用户手册

- 根据有效 (Enable) 及伺服ON请求 (ServoON) 的输入，将按下述方式切换所有实轴的伺服ON/OFF状态、驱动器模块的状态。

输入变量		伺服ON/OFF状态	驱动器状态 (AxisName.Md.Driver_State)
有效 (Enable)	伺服ON请求 (ServoON)		
TRUE	TRUE	伺服ON	6: Operation Enable (OperationEnable)
	FALSE	伺服OFF	5: Switched On (SwitchedOn)
FALSE	TRUE	伺服OFF	3: Switched On Disabled (SwitchOnDisabled)
	FALSE	伺服OFF	3: Switched On Disabled (SwitchOnDisabled)

- 伺服OFF中实轴受外力而旋转的情况下，将进行跟踪处理。
- 伺服ON/OFF的控制操作与控制模式无关。伺服OFF时的控制模式取决于驱动器模块的规格。
- 驱动器模块错误发生中，MCv_AllPower (所有轴允许运行) 将发送到驱动器模块中，因此无需再次将有效 (Enable) 及伺服ON请求 (ServoON) 从FALSE置为TRUE。

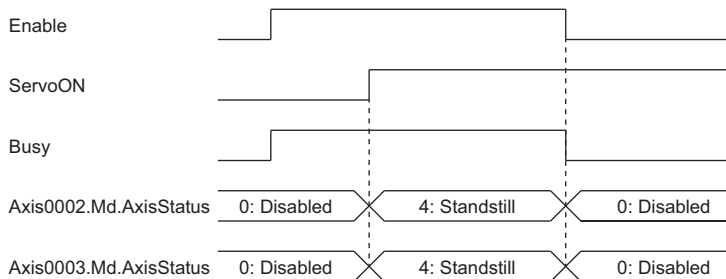
要点

使用MCv_AllPower (所有轴允许运行) 时，个别执行伺服OFF的情况下，应同时使用MC_Power (允许运行)。同时使用了MCv_AllPower (所有轴允许运行) 与MC_Power (允许运行) 的情况下，MC_Power (允许运行) 的指令将优先。

■时序图

例

将轴2 (Axis0002)、轴3 (Axis0003) 设置为实际驱动轴的情况下



注意事项

- 设置了MC_Power (允许运行) 的轴的情况下，切换为运行状态时，MC_Power (允许运行) 的指令将优先。
- 内置运动准备完成 (X420) 处于OFF中时，保持程序停止时的伺服ON/OFF状态。

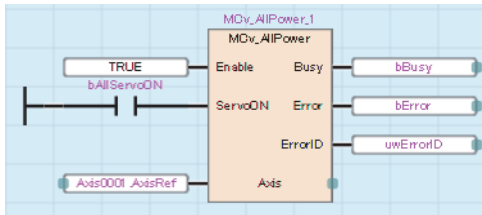
程序示例

将所有轴伺服ON/OFF (bAllServoON) 置为TRUE, 并将所有的轴置为允许运行状态的程序示例如下所示。

■使用的标签

标签名	数据类型	注释
MCv_AllPower_1	MCv_AllPower	所有轴伺服ONFB
bAllServoON	位	所有轴伺服ON/OFF
bBusy	位	执行中
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序



■ST程序

```
MCv_AllPower_1(  
    Enable:= TRUE ,  
    ServoON:= bAllServoON ,  
    Busy=> bBusy ,  
    Error=> bError ,  
    ErrorID=> uwErrorID  
);
```

45.16 轴组倍率修调值设置

MC_GroupSetOverride

执行指定的轴组的目标速度、目标加速度、目标减速度的更改。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
AxesGroup	轴组信息	AXES_GROUP_REF	启动时	—	不能省略	设置轴组。 使用MC_GroupSetOverride(轴组倍率修调值设置)时, 设置输入输出No. (StartIO)。 关于使用的变量(AxesGroupName, AxesGroupRef.), 请参阅下述内容。 ☞ 1408页 AxesGroupName, AxesGroupRef. (轴组信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Enable	有效	BOOL	始终	TRUE、FALSE	FALSE	TRUE时执行MC_GroupSetOverride(轴组倍率修调值设置)。
VelFactor	速度倍率修调系数	LREAL	始终	0.00~10.00	0.00	设置速度的倍率修调系数。 有效(Enable)为TRUE时, 将始终获取。
AccFactor	加速度倍率修调系数	LREAL	始终	0.00、0.01~10.00	0.00	设置加速度的倍率修调系数。 有效(Enable)为TRUE时, 将始终获取。 设置了“0.00”的情况下, 不进行加速度倍率修调系数的更改, 维持上次值进行控制。
JerkFactor	Jerk倍率修调系数	LREAL	始终	0.0	0.0	应设置“0.0”。 * 设置了“0.0”以外的情况下, 将变为超出Jerk倍率修调系数(JerkFactor)范围(错误代码: 1BBEH)。

■输出变量

输出变量	名称	数据类型	初始值	内容
Enabled	有效中	BOOL	FALSE	倍率修调值正常设置的情况下, 将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_GroupSetOverride(轴组倍率修调值设置)时, 将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时, 将返回错误代码。 关于错误代码的详细内容, 请参阅下述手册。 ☞ 所使用的控制器的用户手册

功能

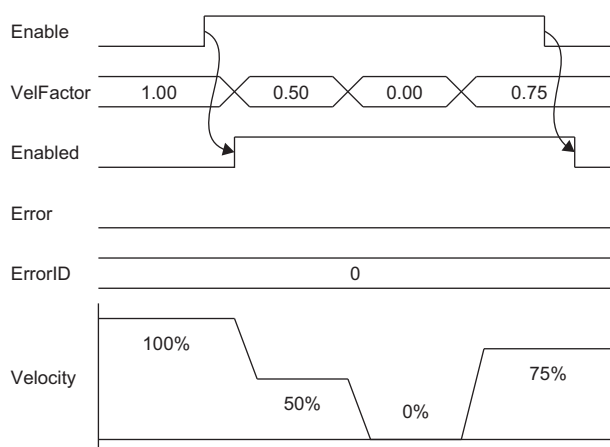
- 更改指定的轴组的目标速度、目标加速度、目标减速度。
- 更改为当前动作中的目标速度、目标加速度、目标减速度乘以倍率修调系数后的值。
- 有效(Enable)变为TRUE时将执行MC_GroupSetOverride(轴组倍率修调值设置)。倍率修调系数为有效中时，有效中(Enabled)将变为TRUE。
- 有效(Enable)为TRUE时，如果更改倍率修调系数的值，则将反映新的倍率修调系数。
- 在MC_GroupSetOverride(轴组倍率修调值设置)内发生了异常的情况下，错误(Error)将变为TRUE，并将错误代码存储到错误代码(ErrorID)中。关于错误代码的详细内容，请参阅下述手册。

📖 所使用的控制器的用户手册

- 如果在速度倍率修调系数(VelFactor)中设置“0.00”的值，轴状态(AxisName.Md.AxisStatus)不转移至“4: 待机中(Standstill)”而停止轴。
- 在加速度倍率修调系数(AccFactor)中设置“0.00”的值时，不更改加速度倍率修调系数，维持上次的加速度倍率修调系数。

■ 时序图

- 正常完成的情况下



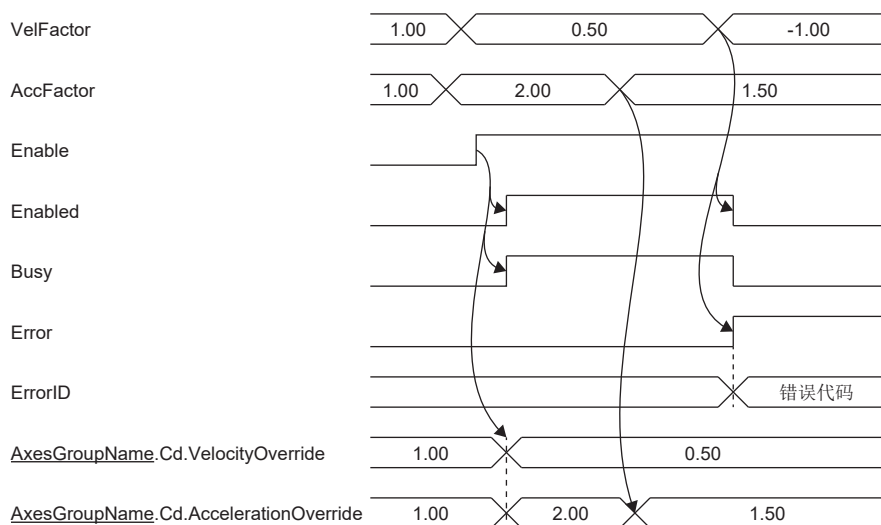
- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

📖 1378页 通过有效(Enable)类型的运动控制FB的基本动作

■动作概要

- 在MC_GroupSetOverride(轴组倍率修调值设置)中, 更改速度倍率修调系数(AxisGroupName.Cd.VelocityOverride)、加速度倍率修调系数(AxisGroupName.Cd.AccelerationOverride)。



- 在倍率修调系数中设置了超出范围的值的的情况下, MC_GroupSetOverride(轴组倍率修调值设置)将发生错误并停止以后的获取。再次执行获取的情况下, 应再次启动有效(Enable)。

■注意事项

- 请勿对同一轴组配置2个及以上的MC_GroupSetOverride(轴组倍率修调值设置)。配置了2个及以上的情况下将无法保证动作。
- 在轴组控制中, 仅轴组控制数据中设置的倍率修调系数有影响。
- MC_GroupSetOverride(轴组倍率修调值设置)执行中, 请勿直接进行速度倍率修调系数(AxisGroupName.Cd.VelocityOverride)、加速度倍率修调系数(AxisGroupName.Cd.AccelerationOverride)的操作。
- 速度倍率修调后的速度超出范围的情况下, 请参阅下述手册的“速度范围”。

📖所使用的控制器的用户手册

- 加速度倍率修调后的加速度、减速度、加速时间、减速时间超出范围的情况下, 请参阅下述手册的“加减速处理功能”。

📖所使用的控制器的用户手册

程序示例

将轴组倍率修调值更改指令(bGroupSetOverride)置为TRUE，并将其更改为轴组1(AxesGroup001)的目标速度、目标加速度、目标减速度乘以速度倍率修调系数“1.0”、加减速度倍率修调系数“2.0”后的值的程序示例如下所示。

■轴组

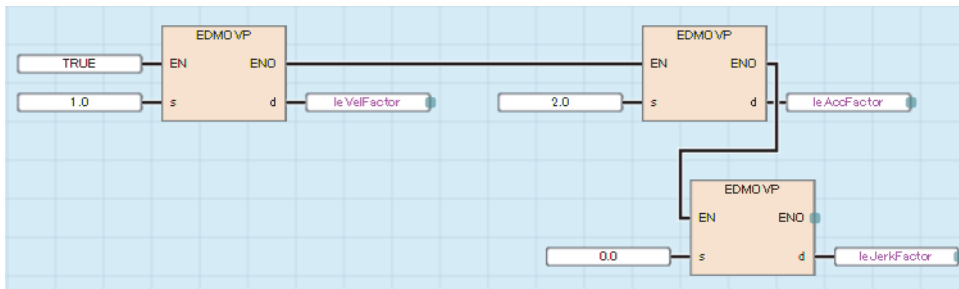
轴组No.	标签名	数据类型	注释
1	AxesGroup001	AXES_GROUP_REF	轴组1

■使用的标签

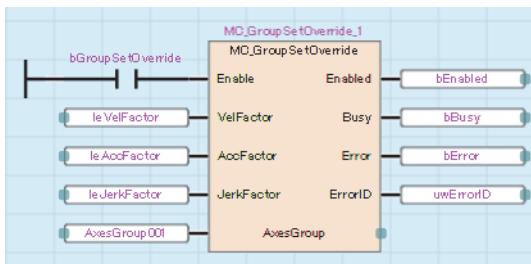
标签名	数据类型	注释
MC_GroupSetOverride_1	MC_GroupSetOverride	轴组倍率修调值设置FB
bGroupSetOverride	位	轴组倍率修调值更改指令
leVelFactor	双精度实数	速度倍率修调系数
leAccFactor	双精度实数	加减速度倍率修调系数
leJerkFactor	双精度实数	Jerk倍率修调系数
bEnabled	位	有效中
bBusy	位	执行中
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

- 轴组倍率修调值更改用数据设置



- 轴组倍率修调值更改



■ST程序

```
//-----轴组倍率修调值更改用数据设置-----  
leVelFactor:= 1.00;  
leAccFactor:= 2.00;  
leJerkFactor:= 0.0;  
  
//-----轴组倍率修调值更改-----  
MC_GroupSetOverride_1(  
    AxesGroup:= AxesGroup001.AxesGroupRef ,  
    Enable:= bGroupSetOverride ,  
    VelFactor:= leVelFactor ,  
    AccFactor:= leAccFactor ,  
    JerkFactor:= leJerkFactor ,  
    Enabled=> bEnabled ,  
    Busy=> bBusy ,  
    Error=> bError ,  
    ErrorID=> uwErrorID  
);
```

45.17 运动错误复位

MCv_MotionErrorReset

对运动系统的所有的错误、警告进行复位。

梯形图	FBD/LD	ST
		<pre>MCv_MotionErrorReset(StartIO:= ?WORD?, Execute:= ?BOOL?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?);</pre>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
StartIO	输入输出No.	WORD (HEX)	启动时	000H~0FEH	可省略	设置起始输入输出编号(以16进制数4位数表示时的前3位数)。

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MCv_MotionErrorReset(运动错误复位)。
Options	选项	DWORD (HEX)	启动时	00000000H	00000000H	应设置“00000000H”。 * 设置了“00000000H”以外的情况下，将变为超出Options范围(错误代码:1BC3H)。

■输出变量

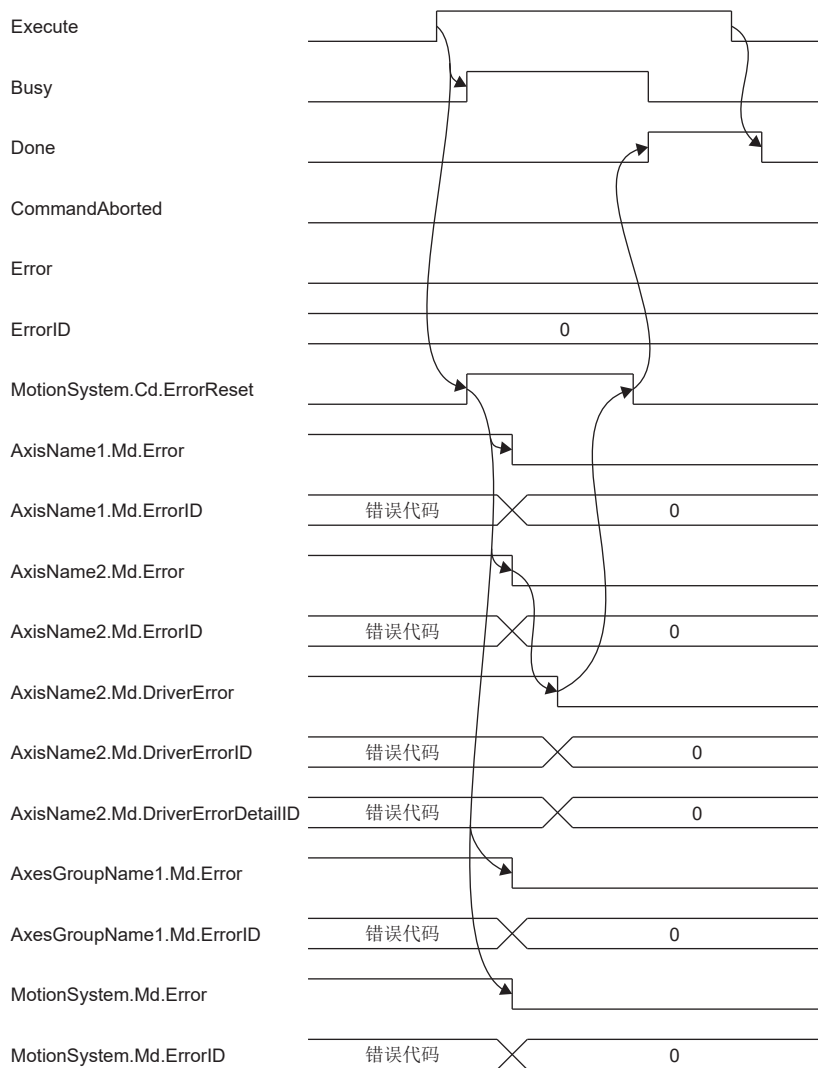
输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	复位完成时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_MotionErrorReset(运动错误复位)时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	由于超时，MCv_MotionErrorReset(运动错误复位)中断时将变为TRUE。 执行指令(Execute)变为FALSE时，将变为FALSE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 □□所使用的控制器的用户手册

功能

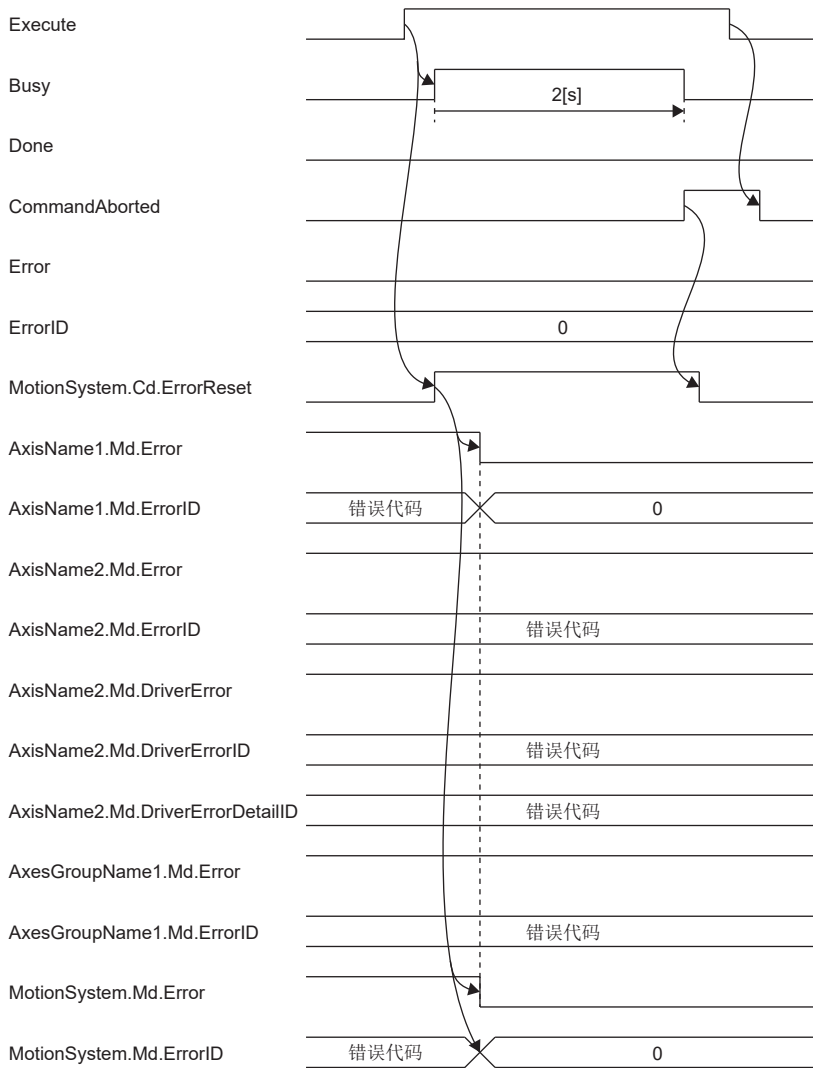
- 执行指令(Execute)为TRUE时执行MCv_MotionErrorReset(运动错误复位),开始处理时执行中(Busy)将变为TRUE,系统错误复位(MotionSystem.Cd.ErrorReset)置为TRUE。
- 运动系统的错误、警告的解除完成时执行中(Busy)将变为FALSE,执行完成(Done)将变为TRUE。
- 即使在残留了错误、警告的原因的状态下将执行指令(Execute)置为TRUE,也不解除错误、警告。在这种情况下,如果在执行FB后的2秒内未解除错误,执行中断(CommandAborted)将变为TRUE,系统错误复位(MotionSystem.Cd.ErrorReset)置为FALSE。应将执行指令(Execute)置为FALSE一次,消除了错误、警告的原因后,再次将执行指令(Execute)置为TRUE。

■时序图

- 正常完成的情况下



• 超时的情况下



注意事项

- 在MCv_MotionErrorReset (运动错误复位) 中，不执行特殊继电器 (SM0)、特殊寄存器 (SD0) 的清除及模块的错误LED的熄灯。这些操作的执行方法，请确认所使用的控制器的用户手册。
- 在执行MCv_MotionErrorReset (运动错误复位) 的过程中，请勿直接操作系统错误复位 (MotionSystem.Cd.ErrorReset)。

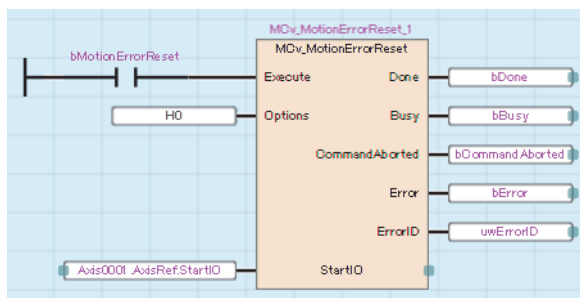
程序示例

将错误复位(bMotionErrorReset)置为TRUE，并对运动系统的所有错误、警告进行复位的程序示例如下所示。

■使用的标签

标签名	数据类型	注释
MCv_MotionErrorReset_1	MCv_MotionErrorReset	运动错误复位FB
bMotionErrorReset	位	错误复位
bDone	位	执行完成
bBusy	位	执行中
bCommandAborted	位	执行中断
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序



■ST程序

```

MCv_MotionErrorReset_1(
    Execute:= bMotionErrorReset ,
    Options:= H00000000 ,
    Done=> bDone ,
    Busy=> bBusy ,
    CommandAborted=> bCommandAborted ,
    Error=> bError ,
    ErrorID=> uwErrorID
);

```

45.18 控制运算周期同步

MCv_SyncOperationCycles

与设置了轴的控制运算周期的周期同步。

限制事项

使用时，应确认控制器及工程工具的版本。

所使用的控制器的用户手册

梯形图	FBD/LD	ST
		<pre> MCv_SyncOperationCycles (Axis:= ?AXIS_REF? , ValueSource:= ?SIGNAL_SELECT? , Enable:= ?BOOL? , OperationCycle:= ?INT? , Options:= ?DWORD? , Enabled=> ?BOOL? , Busy=> ?BOOL? , Error=> ?BOOL? , ErrorID=> ?WORD?); </pre>

设置数据

输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 1387页 AxisName, AxisRef. (轴信息)
ValueSource	数据源选择	SIGNAL_SELECT	启动时	—	可省略	输入将被忽略。

输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Enable	有效	BOOL	始终	TRUE、FALSE	FALSE	在TRUE期间，控制运算周期同步将变为有效。
OperationCycle	控制运算周期	INT	启动时	1~3	1	设置同步目标的控制运算周期。 • 1: 第1运算周期 • 2: 第2运算周期 • 3: 第3运算周期
Options	选项	DWORD (HEX)	启动时	0000000H	0000000H	应设置“0000000H”。 设置“0000000H”以外的情况下，将变为错误。

输出变量

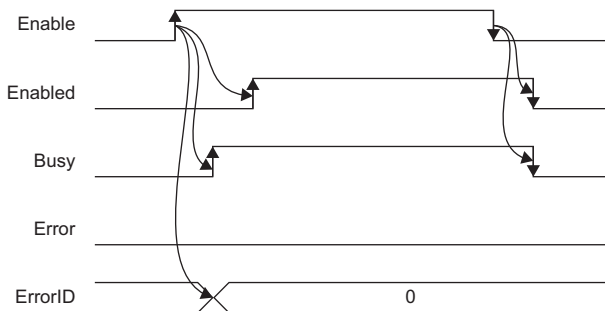
输出变量	名称	数据类型	初始值	内容
Enabled	有效中	BOOL	FALSE	控制运算周期同步正常动作的情况下，将变为TRUE。
Busy	执行中	BOOL	FALSE	MCv_SyncOperationCycles(控制运算周期同步)执行中时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 所使用的控制器的用户手册

功能

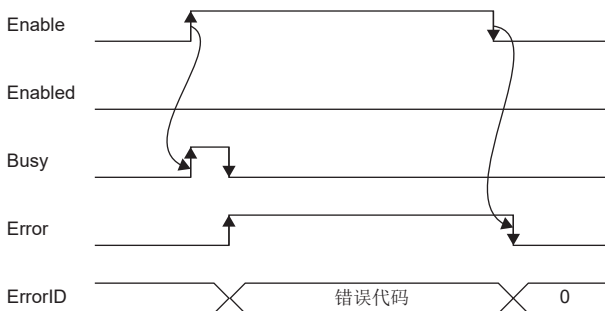
- MCv_SyncOperationCycles (控制运算周期同步) 与设置了所选数据 (轴或数据源) 的控制运算周期的值的周期同步。
- 单轴同步控制FB中，主轴与从轴的控制运算周期不同时使用。
- 在有效 (Enable) 为TRUE时，控制运算周期同步将变为有效。控制运算周期同步处于有效中时，有效中 (Enabled) 将变为TRUE。

■时序图

- 正常完成的情况下



- 异常完成的情况下



注意事项

- 本FB将选择的数据与指定的控制时运算周期同步，但不更改控制运算周期设置。
- 输入与输出 (同步控制中输入为主轴，输出为从轴) 的控制运算周期不同的情况下，应启动本FB，然后启动同步控制。未同步启动同步控制的情况下，将发生运算周期不一致 (错误代码：1B29H)。
- 如果执行本FB，运算时间将增加。
- 运动控制FB的输入变量的刷新是在FB的读取时机进行的，但本FB与FB调用任务 (扫描/恒定周期/事件) 无关，而是以运算周期进行刷新。

程序示例

使用控制运算周期同步(MCv_SyncOperationCycles)对控制周期不同的轴进行同步控制的程序示例如下所示。
在本程序示例中,将第1运算周期中控制的Axis0001作为主轴,将第2运算周期中控制的Axis0002作为从轴启动MC_GearIn。
省略将MR-J5-G置为伺服ON的程序。

■CC-Link IE TSN配置

站号	型号	IP地址	网络周期通信设置	通信周期设置
1	MR-J5-G	192.168.4.1	同步	基本周期
2	MR-J5-G	192.168.4.2	同步	中速

■轴

轴No.	标签名	站地址设置	轴类型	控制周期设置
1	Axis0001	192.168.4.1	实际驱动轴	以第1运算周期运行
2	Axis0002	192.168.4.2	实际驱动轴	以第2运算周期运行

■使用的标签

标签名	数据类型	注释
bSyncStart	位	同步控制开始
bMCGearInExecute	位	齿轮动作开始
MCv_SyncOperationCycles_1	MCv_SyncOperationCycles	控制运算周期同步FB
MC_GearIn_1	MC_GearIn	齿轮动作开始FB

■ST程序

```
SET(MCv_SyncOperationCycles_1.Enabled, bMCGearInExecute);  
//-----控制运算周期同步-----  
MCv_SyncOperationCycles_1(  
    Axis:= Axis0001.AxisRef ,  
    //ValueSource:= ?SIGNAL_SELECT? ,//省略  
    Enable:= bSyncStart ,  
    OperationCycle:= 2 ,//2:第2运算周期  
    Options:= 0  
);  
  
//-----齿轮动作开始-----  
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  
);
```

45.19 高级同步控制1周期当前位置计算

MCv_AdvPositionPerCycleCalc

以指定的凸轮数据为基准，计算1周期当前位置。

限制事项

使用时，应确认控制器及工程工具的版本。

所使用的控制器的用户手册

梯形图	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>

设置数据

输入输出变量


输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	可省略	设置将被忽略。

输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行本FB。
CamTableID	凸轮表ID	MC_CAM_ID	启动时	0~6000	0	设置计算对象的运算凸轮表ID。 在PROFILE_ID结构体的配置文件ID编号(Number)中设置了“0”的情况下，作为直线凸轮进行计算。
RingCountUpperValue	环形计数器上限	LREAL	启动时	0.0、2.0~2147483647.0	0.0	在凸轮位置计算中考虑环形计数器上限的情况下，设置计算对象的输出轴的环形计数器上限值。 设置了“0.0”的情况下，将环形计数器作为“-10000000000.0~10000000000.0”进行计算。
LengthPerCycle	凸轮位置1周期长度	LREAL	启动时	1.0~2147483647.0	1.0	设置凸轮计算所需的1周期长度。
PositionPerCycle	凸轮位置计算1周期当前位置	LREAL	启动时	0.0~2147483647.0	0.0	设置凸轮计算所需的1周期当前位置。 设置的值超出了凸轮位置1周期长度(LengthPerCycle)时，将转换为“0.0~(凸轮位置1周期长度(LengthPerCycle)-0.00001)”的范围内后进行计算。
StrokeAmount	凸轮位置计算行程量	LREAL	启动时	-2147483648.0~2147483647.0	0.0	设置凸轮计算所需的凸轮行程量。
ReferenceSetPosition	凸轮位置计算凸轮基准位置	LREAL	启动时	-10000000000.0~10000000000.0	0.0	设置凸轮计算所需的凸轮基准位置。
SetPosition	凸轮位置计算凸轮指令当前位置	LREAL	启动时	-10000000000.0~10000000000.0	0.0	设置凸轮计算所需的输出轴的当前位置。

输入变量	名称	数据类型	获取	有效范围	初始值	内容
StartingPoint	凸轮位置计算 凸轮开始点	DWORD	启动时	0~65535	0	凸轮计算中考虑凸轮开始点的情况下进行设置。
Options	选项	DWORD (HEX)	启动时	固定为0	0	在本FB中不能设置。应设置为0。设置了0以外的情况下，将变为超出Options范围(错误代码：1ABBH)。

■输出变量


输出变量	名称	数据类型	初始值	内容
CalculationResult	计算结果	LREAL	0.0	存储1周期当前位置的计算结果。
Done	执行完成	BOOL	FALSE	计算完成时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了本FB时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。关于错误代码的详细内容，请参阅下述手册。  所使用的控制器的用户手册

■公开变量

公开变量	名称	数据类型	初始值	内容
OutputData	输出数据	LREAL	0.0	<ul style="list-style-type: none"> 直线凸轮，或运算配置文件的插补方法指定(Interpolate)为“0：直线插补”时，计算结果所对应的行程量将被存储。 运算配置文件的插补方法指定为“1：对区间分别指定”或“2：样条插补”时，或数据格式的类型中设置为“旋转刀具”时，以%单位存储计算结果(1周期当前位置)所对应的凸轮数据的行程比。 输出数据(OutputData)在执行完成(Done)变为TRUE时存储值。
InstanceID	实例ID	INSTANCE_ID	0	是实例ID。 实例创建时由系统自动设置。

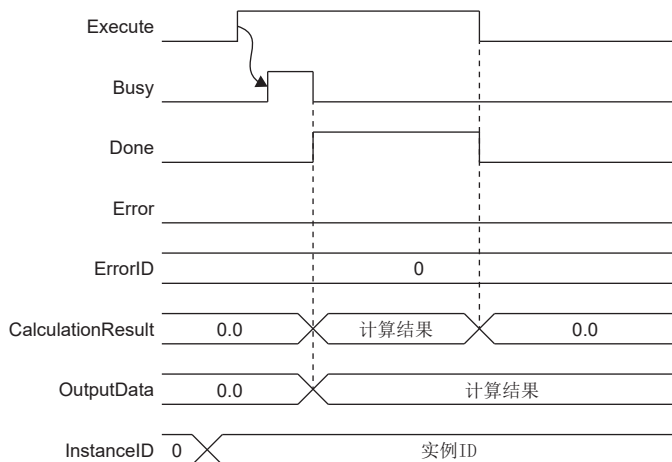
功能

- 从凸轮数据开始，在凸轮位置计算凸轮基准位置(ReferenceSetPosition)与凸轮位置计算凸轮指令当前位置(SetPosition)的基础上计算1周期当前位置。计算结果输出到计算结果(CalculationResult)。
- 关于1周期当前位置计算的详细内容，请参阅下述手册。

所使用的控制器的用户手册


■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作


注意事项

- 对于凸轮数据，应事先通过展开设置自动展开，或通过MC_CamTableSelect(凸轮表选择)展开到展开区域后使用。
- 1周期当前位置计算(MCv_AdvPositionPerCycleCalc)的计算结果应仅应用于MCv_AdvancedSync(高级同步控制)中。应用于MCv_AdvancedSync(高级同步控制)以外的同步控制FB(MC_CamIn等)的情况下，计算结果与实际的动作将不一致。
- 坐标数据的1周期当前位置计算中，凸轮数据的结束点根据1周期长可能有下述计算结果。使用坐标数据的情况下，建议凸轮数据的结束点与1周期长保持一致。

凸轮数据的结束点	内容
大于1周期长时	由于不根据1周期长度的设定值进行凸轮数据的全范围查找，因此可能出现与同步控制的轨迹不同的计算结果。
小于1周期长时	根据凸轮指令位置与凸轮基准位置的关系，1周期当前值可能大于1周期长。

程序示例

关于本FB的程序示例，请参阅下述手册。

所使用的控制器的用户手册

45.20 高级同步控制凸轮指令当前位置计算

MCv_AdvCamSetPositionCalc

以指定的凸轮数据为基准，计算凸轮指令当前位置。

限制事项

使用时，应确认控制器及工程工具的版本。

所使用的控制器的用户手册

梯形图	FBD/LD	ST
		<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>

设置数据


输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	可省略	设置将被忽略。

输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行本FB。
CamTableID	凸轮表ID	MC_CAM_ID	启动时	0~6000	0	设置计算对象的运算凸轮表ID。 在PROFILE_ID结构体的配置文件ID编号(Number)中设置了“0”的情况下，作为直线凸轮进行计算。
RingCountUpperValue	环形计数器上限	LREAL	启动时	0.0、2.0~2147483647.0	0.0	在凸轮位置计算中考虑环形计数器上限的情况下，设置计算对象的输出轴的环形计数器上限值。 设置了“0.0”的情况下，将环形计数器作为“-10000000000.0~10000000000.0”进行计算。
LengthPerCycle	凸轮位置1周期长度	LREAL	启动时	1.0~2147483647.0	1.0	设置凸轮计算所需的1周期长度。
StrokeAmount	凸轮位置计算行程量	LREAL	启动时	-2147483648.0~2147483647.0	0.0	设置凸轮计算所需的凸轮行程量。
ReferenceSetPosition	凸轮位置计算凸轮基准位置	LREAL	启动时	-10000000000.0~10000000000.0	0.0	设置凸轮计算所需的凸轮基准位置。
PositionPerCycle	凸轮位置计算1周期当前位置	LREAL	启动时	0.0~2147483647.0	0.0	设置凸轮计算所需的1周期当前位置。 应在下述范围内进行设置。 0.0~凸轮位置1周期长度(LengthPerCycle)的设定值
StartingPoint	凸轮位置计算凸轮开始点	DWORD	启动时	0~65535	0	凸轮计算中考虑凸轮开始点的情况下进行设置。
Options	选项	DWORD (HEX)	启动时	固定为0	0	在本FB中不能设置。应设置为0。设置了0以外的情况下，将变为超出Options范围(错误代码：1ABBH)。

■输出变量


输出变量	名称	数据类型	初始值	内容
CalculationResult	计算结果	LREAL	0.0	存储凸轮指令当前位置的计算结果。
Done	执行完成	BOOL	FALSE	计算完成时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了本FB时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。关于错误代码的详细内容，请参阅下述手册。  所使用的控制器的用户手册

■公开变量

公开变量	名称	数据类型	初始值	内容
InputData	输入数据	LREAL	0.0	<ul style="list-style-type: none"> 直线凸轮时，存储计算结果所对应的1周期当前位置。 运算配置文件的插补方法指定 (Interpolate) 为“0：直线插补”时，存储计算结果 (凸轮指令当前位置) 所对应的1周期当前位置。 运算配置文件的插补方法指定为“1：对区间分别指定”或“2：样条插补”时，或数据格式的类型中设置为“旋转刀具”时，存储计算结果 (凸轮指令当前位置) 所对应的凸轮数据的分辨率。 输入数据 (InputData) 在执行完成 (Done) 变为TRUE时存储值。
InstanceID	实例ID	INSTANCE_ID	0	是实例ID。 实例创建时由系统自动设置。

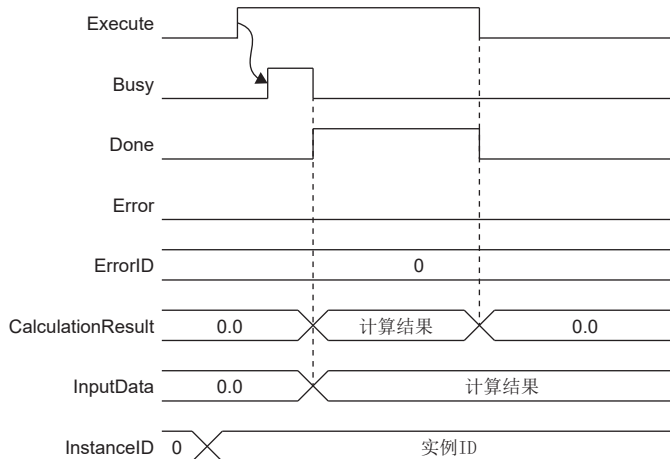
功能

- 在凸轮位置计算1周期当前位置 (PositionPerCycle) 与凸轮位置计算凸轮基准位置 (ReferenceSetPosition) 的基础上计算凸轮指令当前位置。计算结果输出到计算结果 (CalculationResult)。
- 关于凸轮指令当前位置计算的详细内容，请参阅下述手册。

所使用的控制器的用户手册


■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。


 1375页 通过执行指令 (Execute) 类型的运动控制FB的基本动作

注意事项

- 对于凸轮数据，应事先通过展开设置自动展开，或通过MC_CamTableSelect(凸轮表选择)展开到展开区域后使用。
- 凸轮指令当前位置计算(MCv_AdvCamSetPositionCalc)的计算结果应仅应用于MCv_AdvancedSync(高级同步控制)中。应用于MCv_AdvancedSync(高级同步控制)以外的同步控制FB(MC_CamIn等)的情况下，计算结果与实际的动作将不一致。

程序示例

关于本FB的程序示例，请参阅下述手册。

所使用的控制器的用户手册

45.21 数字凸轮开关输出

MC_DigitalCamSwitch

根据任意数据的值输出ON/OFF信号。

限制事项

使用时，应确认控制器及工程工具的版本。

所使用的控制器的用户手册

梯形图	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>

设置数据


输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴	AXIS_REF	启动时	—	可省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName, AxisRef. (轴信息)
Switches	开关	MC_CAMSWITCH_DATA_REF ☞ 1426页 MC_CAMSWITCH_REF	始终	1~60000	不能省略	使用运算配置文件并设置开关设置。 ☞ 1572页 开关(Switches)
Outputs	输出信号	MC_OUTPUT_REF	启动时	—	不能省略	设置输出信号。 关于详细内容，请参阅下述章节。 ☞ 1573页 输出信号(Outputs)
TrackOptions	跟踪选项	MC_TRACK_REF	始终	—	—	设置跟踪选项。 关于详细内容，请参阅下述章节。 ☞ 1574页 跟踪选项(TrackOptions)

输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Enable	有效	BOOL	始终	TRUE、FALSE	FALSE	在TRUE期间，执行本FB。
ValueSource	数据源选择	MC_CAMSWITCH_SOURCE_REF	启动时	—	—	设置数据源。 关于详细内容，请参阅下述章节。 ☞ 1575页 数据源选择(ValueSource)
ForcedOff	强制OFF	BOOL	始终	TRUE、FALSE	FALSE	在TRUE期间，输出信号强制变为OFF。 关于详细内容，请参阅下述章节。 ☞ 1576页 强制OFF(ForcedOff)
ForcedOn	强制ON	BOOL	始终	TRUE、FALSE	FALSE	在TRUE期间，输出信号强制变为ON。 关于详细内容，请参阅下述章节。 ☞ 1576页 强制ON(ForcedOn)
Options	选项	DWORD (HEX)	启动时	—	0000000H	设置功能选项。 关于详细内容，请参阅下述章节。 ☞ 1576页 选项(Options)

■输出变量


输出变量	名称	数据类型	初始值	内容
InOperation	运行中	BOOL	FALSE	输出信号为有效时，将变为TRUE。
Busy	执行中	BOOL	FALSE	本FB执行中时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。  所使用的控制器的用户手册

■公开变量

公开变量	名称	数据类型	初始值	内容
InputsPerCycle	1周期当前位置	LREAL	0	存储数据源的值。
OutputData	输出值	BOOL	0	存储输出信号的输出值。
SwitchNo	开关编号	WORD (UINT)	0	存储ON中的开关编号。
CurrentProfileNo	执行中配置文件No.	WORD (UINT)	0	存储正在执行的配置文件ID。
InstanceID	实例ID	INSTANCE_ID	0	是实例ID。 实例创建时由系统自动设置。


功能

判定数据源选择(ValueSource)中设置的数据源是否满足开关设置的条件后输出输出信号。关于数字凸轮开关功能的详细内容，请参阅下述手册。

所使用的控制器的用户手册

■时序图


关于本FB的时序图请参阅下述章节。

 1378页 通过有效(Enable)类型的运动控制FB的基本动作

■开关(Switches)

在MC_CAMSWITCH_DATA_REF结构体中指定运算配置文件ID。

关于设置方法请参阅下述手册。

所使用的控制器的用户手册

执行中将开关(Switches)更改为超出范围或没有展开区域时，为程序警告(事件代码：00F19H)，并以更改前的值继续动作。

■输出信号(Outputs)

在MC_OUTPUT_REF结构体中设置输出信号。关于MC_OUTPUT_REF结构体的详细内容，请参阅下述章节。

☞ 1426页 MC_OUTPUT_REF[M]

输出信号(Outputs)中设置的MC_OUTPUT_REF结构体的输出信号(Outputs)的设置范围如下所示。

输出信号(Outputs)的类型为SIGNAL_SELECT结构体。关于SIGNAL_SELECT结构体，请参阅下述章节。

☞ 1431页 SIGNAL_SELECT(信号选择)

结构体	变量名	类型	设置范围
SIGNAL_SELECT(信号选择)	Source(信号)	TARGET_REF ☞ 1428页 TARGET_REF(输入信号)	<ul style="list-style-type: none"> ■类型 • BOOL ■数据类型 • [LINK]*1
	Detection(信号检测方法)	INT (MC_SIGNAL_LOGIC) ☞ 1459页 MC_SIGNAL_LOGIC[M]	无法选择位输出信号的信号逻辑。
	CompensationTime(补偿时间)*2	LREAL	-5.0~5.0[s]
	FilterTime(筛选时间)	LREAL	0.0[s]*3

*1 指定了[LINK]的情况下，对应的网络标签的写入将被忽略。仅在使用外部信号高精度输出时可以设置。详细情况请确认所使用的控制器的用户手册。

*2 对设备的响应时间导致的延迟等进行补偿的情况下进行设置。时间补偿的详细情况请确认所使用的控制器的用户手册。

*3 设置了0.0以外的情况下，将变为超出信号的筛选时间设置范围(错误代码：1BC8H)。去除振荡的情况下，应设置跟踪选项(TrackOptions)的Hysteresis(转换禁止距离)。

多个FB实例中设置了同一输出信号的情况下，输出输出结果的逻辑或。

输出信号的刷新与使用的程序执行周期无关，而是以下述时机进行刷新。

数据类型	进行刷新的时机
[LINK]	对象站的通信周期

■跟踪选项(TrackOptions)

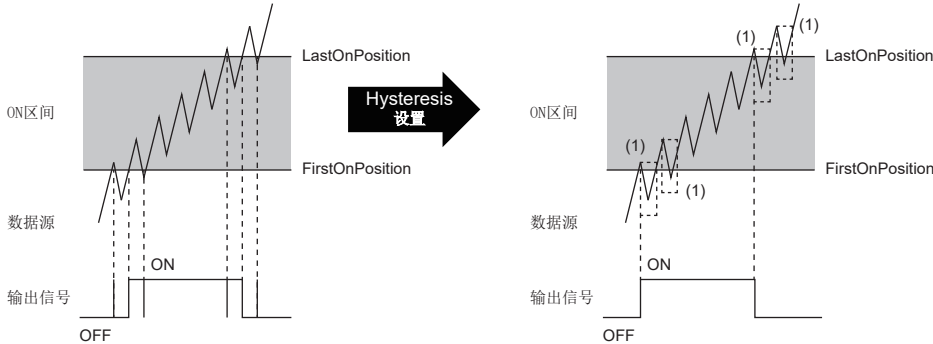
在MC_TRACK_REF结构体中设置跟踪选项。关于MC_TRACK_REF结构体的详细内容，请参阅下述章节。

☞ 1427页 MC_TRACK_REF

跟踪选项(TrackOptions)中设置的MC_TRACK_REF结构体的跟踪选项(TrackOptions)的设置范围如下所示。

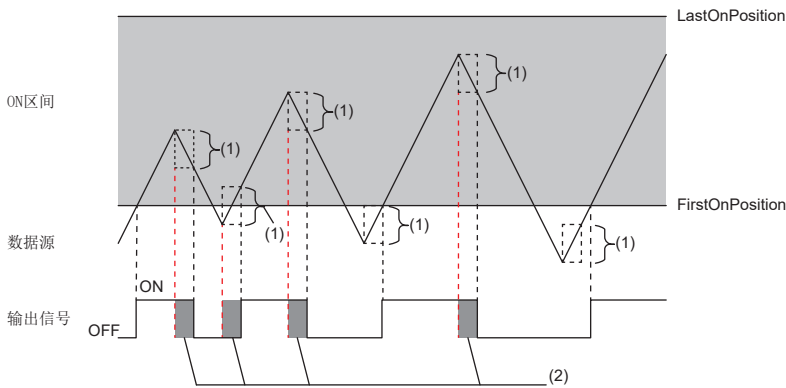
项目	设置范围	内容
Hysteresis (转换禁止距离)	0.0~10000000000.0	<p>设置数据源的速度不稳定时，防止输出信号振荡的死区。^{*1}</p> <p>应设置为小于数据源环形计数器范围的一半。设置为数据源环形计数器范围的一半及以上时，将变为MC_DigitalCamSwitch指令异常(错误代码：1BCAH)。</p> <p>执行中更改为超出范围的设定值时，为程序警告(事件代码：00F19H)，并以更改前的值继续动作。</p> <p>从数据源的移动方向反转的位置开始至位于Hysteresis内期间，即使更改Hysteresis也不会反映到控制中。在超出Hysteresis时将反映到控制中。</p>

*1 详细内容如下所示。



(1) Hysteresis (转换禁止距离)

- 开关设置的开关模式选择(CamSwitchMode)为“1: 位置基准”的情况下，数据源的变化方向位于反转的位置与Hysteresis(转换禁止距离)之间时，不进行输出信号的ON/OFF切换。开关设置的方向选择(Direction)为“1: 正方向”的情况下，按下述方式动作。



(1) Hysteresis (转换禁止距离)

(2) 不进行输出信号的ON/OFF切换。

■数据源选择(ValueSource)

在MC_CAMSWITCH_SOURCE_REF结构体中指定下述内容。

- 数据源
- 数据源的上限值及下限值

关于MC_CAMSWITCH_SOURCE_REF结构体的详细内容，请参阅下述章节。

☞ 1427页 MC_CAMSWITCH_SOURCE_REF

变量名	类型	设置范围	内容
Source(数据源)	TARGET_REF ☞ 1428页 TARGET_REF(输入信号)	■类型 • LREAL*1 ■数据类型 • [AXIS] • [VAR] • [DEV] • [CONST]	数据类型中指定了[AXIS]的情况下，参照轴(Axis)中指定的轴数据。即使对象轴错误或被切断的情况下，本FB也将继续执行控制。指定[AXIS]，且在本FB执行中对对象轴执行当前位置更改或原点复位的情况下，输出信号的ON/OFF可能会出现预期外的结果。应在本FB停止中或强制OFF、强制ON中进行。 数据类型中指定了[VAR]、[DEV]、[CONST]的情况下，忽略轴(Axis)中指定的轴数据。
RingCount_LowerValue(数据源环形计数器下限值)*2	LREAL	-10000000000.0 ~10000000000.0	将数据源的环形计数器范围设置在下述公式的范围内。 $RingCount_LowerValue \leq 数据源 < RingCount_UpperValue$ 应结合数据类型中指定(LREAL)前的数据源范围进行设置。 ■数据源中使用INT型数据情况下的设置示例 RingCount_LowerValue: -32768.0 RingCount_UpperValue: 32768.0 ■使用取得0~99范围整数值的用户定义数据情况下的设置示例 RingCount_LowerValue: 0.0 RingCount_UpperValue: 100.0 RingCount_LowerValue(数据源环形计数器下限值)为RingCount_UpperValue(数据源环形计数器上限值)及以上的情况下，将变为MC_DigitalCamSwitch指令异常(错误代码: 1BCAH)。 RingCount_UpperValue(数据源环形计数器上限值)-RingCount_LowerValue(数据源环形计数器下限值)的绝对值小于2时，将变为MC_DigitalCamSwitch指令异常(错误代码: 1BCAH)。 应设置每个刷新周期的数据源的移动量小于数据源环形计数器范围的一半。每个刷新周期的数据源的移动量为数据源环形计数器范围的一半及以上时，不能正常进行方向判定，且不能在正确的时机输出输出信号。
RingCount_UpperValue(数据源环形计数器上限值)*2			

*1 ON区间判定作为LREAL型执行动作。指定LREAL型以外的数据的情况下，应明确在数据的类型中指定(LREAL)。数据的类型中不指定(LREAL)时，将变为外部信号选择字符串不正确(错误代码: 1B05H)。

*2 Source(数据源)中指定了[Axis]的情况下，由于参照轴(Axis)中指定的轴数据，因此设定值将被忽略。

对于1个数据源可以同时执行多个FB实例。

输出信号的刷新与使用的程序执行周期无关，而是以下述时机进行刷新。

数据类型	进行刷新的时机
[VAR]	网络标签: 对象站的通信周期 网络标签以外: 第1运算周期
[DEV]	第1运算周期
[CONST]	
[AXIS]	对象站的通信周期

■强制OFF (ForcedOff)

执行中将输出信号强制置为OFF。

优先于跟踪选项(TrackOptions)的Hysteresis(转换禁止距离)进行控制。

强制OFF (ForcedOff) 状态	动作
FALSE	根据开关设置执行输出信号的ON/OFF。 ☞所使用的控制器的用户手册
TRUE	将输出信号强制置为OFF。

■强制ON (ForcedOn)

执行中将输出信号强制置为ON。

强制OFF (ForcedOff) 为TRUE时，将强制ON (ForcedOn) 置为TRUE时将强制变为ON。

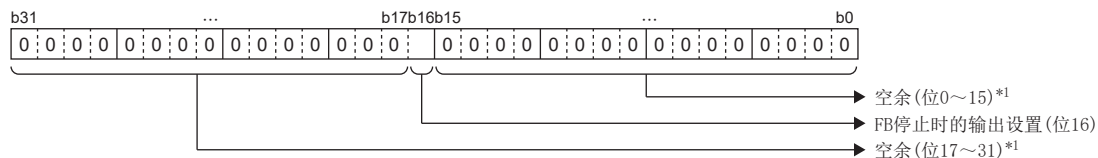
优先于跟踪选项(TrackOptions)的Hysteresis(转换禁止距离)进行控制。

强制ON (ForcedOn) 状态	动作
FALSE	根据开关设置执行输出信号的ON/OFF。 ☞所使用的控制器的用户手册
TRUE	将输出信号强制置为ON。

■选项 (Options)

以位指定设置本FB中使用的功能选项。

以位指定设置的内容如下所示。



*1 空余中应设置“0”。设置了“0”以外的情况下，将变为MC_DigitalCamSwitch指令异常(错误代码: 1BCAH)且不启动。

位	名称	内容
16	FB停止时输出设置	<p>设置在有效(Enable)为TRUE→FALSE、“可编程控制器就绪(MotionSystem.Cd.SequenceReady)”为ON→OFF或发生错误的情况下，当FB停止时是保持输出信号，还是置为OFF。</p> <ul style="list-style-type: none"> • 0: 保持 • 1: 置为OFF <p>执行中切换“可编程控制器就绪(MotionSystem.Cd.SequenceReady)”时的动作如下所示。</p> <p>可编程控制器就绪</p> <p>ON区间</p> <p>输出信号 (FB停止时输出设置: 为0时)</p> <p>输出信号 (FB停止时输出设置: 为1时)</p> <p>LastOnPosition</p> <p>FirstOnPosition</p> <p>■Outputs: [LINK]指定时 关于控制器STOP、出错时的输出，请参阅下述手册。 ☞所使用的控制器的用户手册 由于至链接软元件RWw的输出将保持，因此远程I/O的输出HOLD/CLEAR设置功能优先。关于详细内容，请参阅下述手册。 ☞所使用的控制器的用户手册</p>

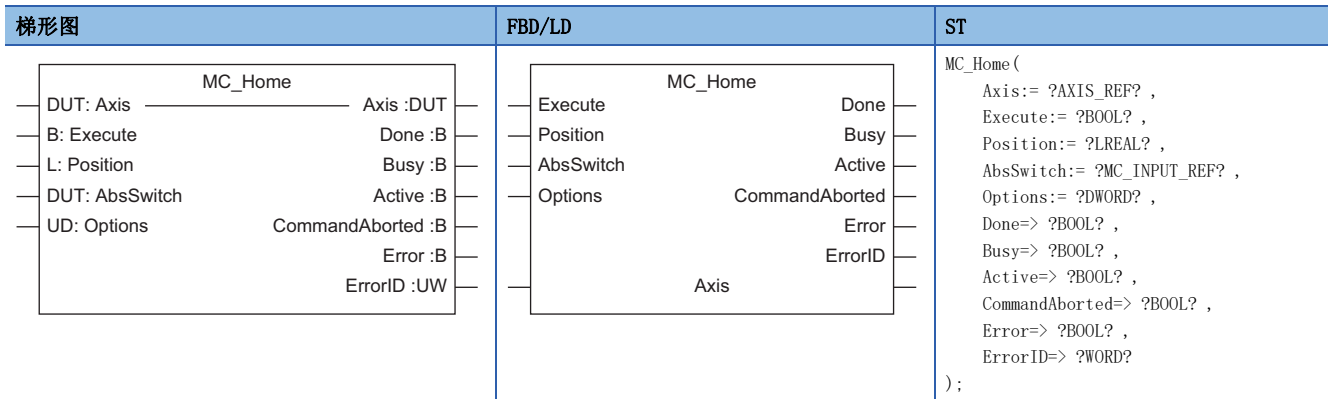
注意事项

输出信号 (Outputs) 中设置的信号与输出值 (OutputData) 不同时，应确认其他功能中是否操作输出信号 (Outputs) 中设置的信号。

46.1 原点复位

MC_Home

进行指定的轴的原点复位。



设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName, AxisRef. (轴信息)

■输入变量

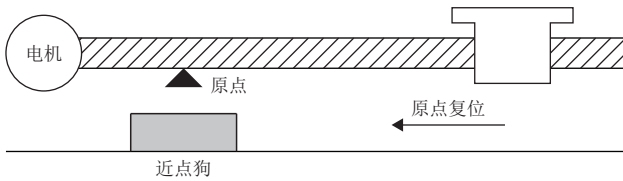
输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_Home(原点复位)。
Position	目标位置	LREAL	启动时	-10000000000.0~10000000000.0	0.0	设置原点地址。 应在下述范围内进行设置。 • -10000000000.0 ≤ 设定值 < 10000000000.0 * 环形计数器有效的情况下，将变为环形计数器范围。
AbsSwitch	原点开关	MC_INPUT_REF	启动时	—	—	设置驱动器式原点复位中传输至设备的近点狗信号。 关于详细内容，请参阅下述章节。 ☞ 1582页 原点开关(AbsSwitch)
Options	选项	DWORD (HEX)	启动时	00000000H~00010000H	00000000H	将MC_Home(原点复位)的功能选项以位指定进行设置。 关于详细内容，请参阅下述章节。 ☞ 1582页 选项(Options)

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	原点复位完成时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_Home(原点复位)时，将变为TRUE。
Active	控制中	BOOL	FALSE	MC_Home(原点复位)正在控制轴时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MC_Home(原点复位)的执行因错误或多重启动等而中断时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 ☞ 所使用的控制器的用户手册

功能

- 在零点复位控制中，进行机械原点的确定。此时，不使用运动系统及驱动器模块中存储的地址信息。零点复位后，将机械确定的位置作为定位控制的起点“原点”。



- 零点复位方式有“驱动器式零点复位”、“数据集式零点复位”。
根据下述条件，将变为零点复位控制启动时的零点复位方式。

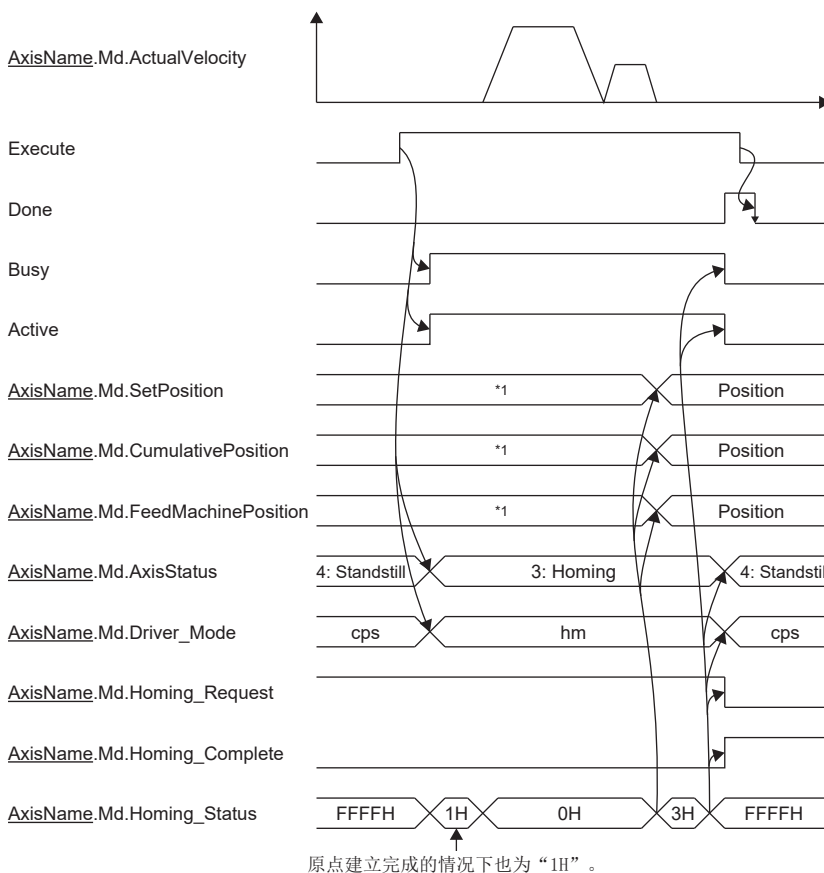
零点复位方式	零点复位方式的条件
驱动器式零点复位	满足下述所有条件的情况下，将变为驱动器式零点复位。 <ul style="list-style-type: none"> • 轴类型为实际驱动轴时。 • 驱动器模块支持Homing模式时。 • 将“Home offset(607CH)”设置为对象数据时。
数据集式零点复位	除上述以外的情况下，将变为数据集式零点复位。

■驱动器式零点复位

将驱动器模块切换为Homing模式，根据驱动器模块侧设置的定位模式进行零点复位。

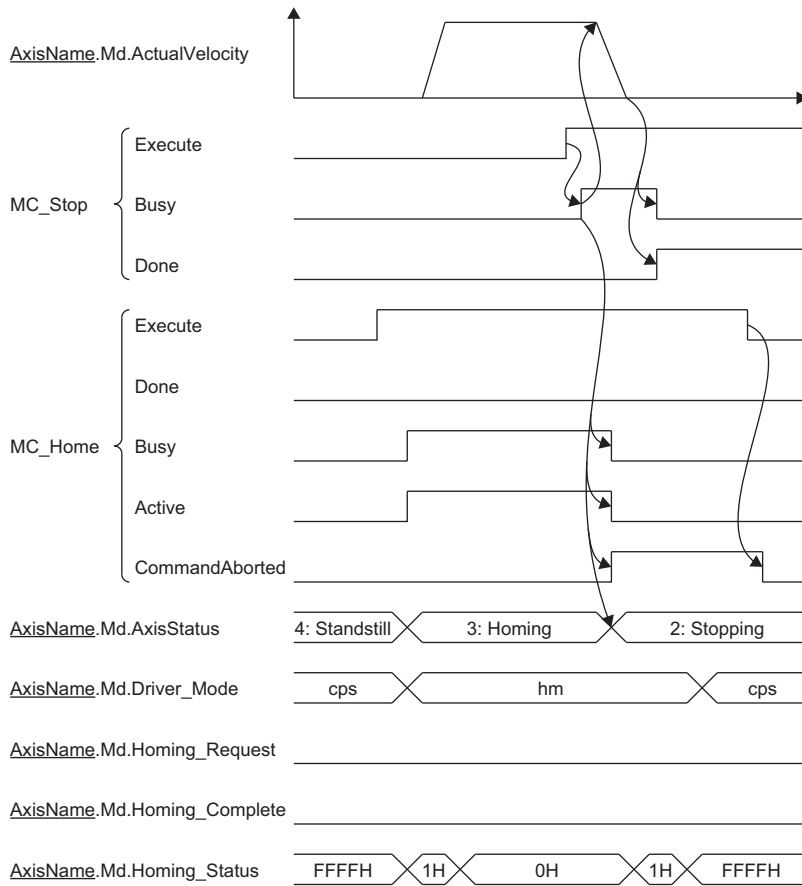
更改零点复位方式及各种参数的情况下，应通过MC_WriteParameter(参数写入)更改驱动器模块的零点复位数据。零点复位的动作及可设置的参数取决于驱动器模块的规格。关于驱动器模块的详细内容，请参阅驱动器模块的手册。

- 时序图
 - 通常运行时



*1 跟踪有效/无效选择(选项(Options))：位16为“0：跟踪无效”的情况下，零点复位中不进行更新。“1：跟踪有效”的情况下，零点复位中根据当前位置的变动进行更新。

- 异常完成时
- 关于异常完成时的时序图，请参阅下述章节。
- ☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作
- 执行MC_Stop(强制停止)时



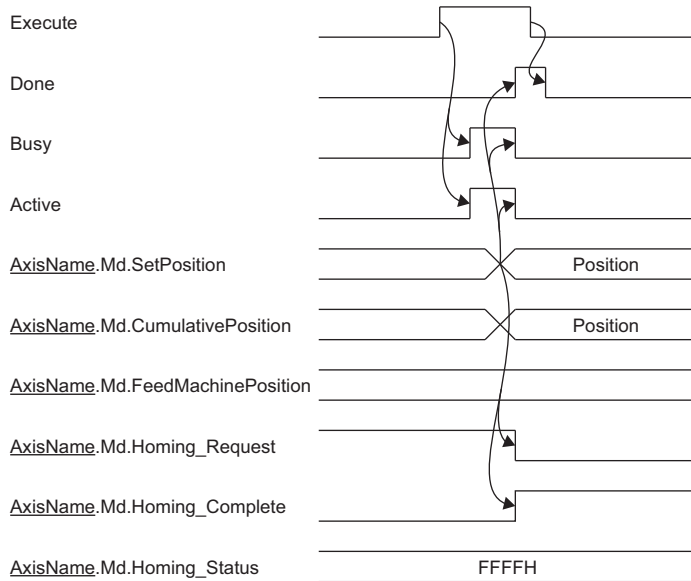
- 原点复位时将MC_Stop(强制停止)置为了TRUE的情况下，将向驱动器模块发送“HALT”信号。使用不支持HALT的驱动器模块的情况下，由于不通过MC_Stop(强制停止)停止，因此应使用紧急停止。
- 原点复位时的停止处理按照驱动器模块的规格。因此，将忽略MC_Stop(强制停止)的减速度(Deceleration)及Jerk。

■数据集式原点复位

对虚拟轴及设备侧不具有原点信息的实轴进行原点复位。在运动系统内部完成，不使用外部信号等。将进行了原点复位时的目标位置(Position)作为原点登录到运动系统中，并将指令当前位置(AxisName.Md.SetPosition)与累计当前位置(AxisName.Md.CumulativePosition)改写为目标位置(Position)。

• 时序图

• 通常运行时



• 异常完成时

关于异常完成时的时序图，请参阅下述章节。

☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■关于原点复位请求

- 需要执行原点复位的情况下，运动系统将原点复位请求 (AxisName.Md.Homing_Request) 置为TRUE。原点复位请求 (AxisName.Md.Homing_Request) 为以下时，变为TRUE。
 - 接通了系统电源，或对其进行了复位时。^{*1}
 - 更改了轴类型时。
 - 启动了原点复位时。(除非原点复位正常完成，否则原点复位请求不会变为FALSE。)
 - 更改了轴的驱动器单位转换(分子/分母)时。
 - 执行了当前值恢复时。
 - 运动系统内的绝对位置数据由于存储器异常等原因而丢失时。
 - 接通了驱动器模块的电源时。^{*1*2}
 - 在驱动器模块侧检测出“绝对位置丢失”时。^{*2}
 - 更改了对象数据的“Polarity(607EH) (b7: position polarity)”时。^{*2}
 - 更改了驱动器模块侧的电子齿轮时。^{*2}
 - 检测出驱动器模块及电机编码器的更改时。
 - 在标尺测量编码器中检测出绝对位置丢失时。^{*3}
 - 更改了标尺测量编码器的编码器分辨率时。^{*3}
 - 更改了编码器环形计数器上限值/下限值时。^{*4}

*1 绝对位置系统的情况下，系统/驱动器模块电源断开、复位时完成了原点复位的情况下不变为TRUE。

*2 仅实际驱动轴。

*3 仅实际编码器轴(经由驱动器模块)。

*4 仅虚拟编码器轴。

要点

原点复位请求 (AxisName.Md.Homing_Request) 变为TRUE时，将记录到事件履历中。
可以在事件履历中确认内容。

- 在不需要进行原点复位的系统中，应进行原点复位请求清除。通过将原点复位请求清除 (AxisName.Cd.Homing_ClearRequest) 置为TRUE，原点复位请求 (AxisName.Md.Homing_Request) 将变为FALSE。

要点

通过程序等将原点复位请求 (AxisName.Md.Homing_Request) 直接改写为FALSE的情况下，将不清除运动系统内部的原点复位请求。必须使用原点复位请求清除 (AxisName.Cd.Homing_ClearRequest) 将原点复位请求置为FALSE。

■原点开关(AbsSwitch)

设置驱动器式原点复位中传输至设备的近点狗信号。

原点开关(AbsSwitch)在MC_INPUT_REF结构体中进行设置。关于MC_INPUT_REF结构体，请参阅下述章节。

☞ 1426页 MC_INPUT_REF[M]

原点开关(AbsSwitch)中设置的输入信号(AbsSwitch.Signal)的设置范围如下所示。

触发信号(TriggerInput.Signal)的类型将为SIGNAL_SELECT结构体。

结构体	变量名	类型	设置范围
SIGNAL_SELECT(信号选择)	Source(信号)	TARGET_REF ☞ 1428页 TARGET_REF(输入信号)	<ul style="list-style-type: none"> ■类型 • BOOL ■数据类型 • [OBJ] • [VAR] • [DEV] • [CONST]
	Detection(信号检测方法)	INT (MC_SIGNAL_LOGIC) ☞ 1459页 MC_SIGNAL_LOGIC[M]	<ul style="list-style-type: none"> • 0: TRUE时检测(HighLevel) • 1: FALSE时检测(LowLevel)
	CompensationTime(补偿时间)	LREAL	0.0[s]
	FilterTime(筛选时间)	LREAL	0.0~5.0[s]

- [OBJ]指定时省略了对象修饰的情况下，参照轴(Axis)的轴的对象。但是，轴(Axis)的轴不具有站地址，或站地址未设置的情况下，输出外部信号选择不正确(错误代码：1B31H、1B32H、1BD6H、1BD7H)。
- [OBJ]指定时参照的站为轴模拟功能有效且站地址已设置的情况下，参照正在模拟的对象。
- 信号名(AbsSwitch.Signal.Source.Target)中使用了不能使用的信号的情况下，输出外部信号选择字符串不正确(错误代码：1B05H)、外部信号选择不正确(错误代码：1B31H、1B32H、1BD6H、1BD7H)、插件SignalIO内部异常(错误代码：1B33H、1BD8H)的任意一个。
- 轴中指定了MELSERVO以外的驱动器模块的情况下，无法将原点开关中指定的信号作为近点狗信号传输至驱动器模块。

■选项(Options)

将MC_Home(原点复位)中使用的功能选项以位指定进行设置。

以位指定设置的内容如下所示。



*1 空余中应设置“0”。设置“0”以外的情况下，将变为错误。

位	名称	内容
16	跟踪有效/无效选择	设置驱动器式原点复位中，将指令当前位置的跟踪是置为有效，还是置为无效。 <ul style="list-style-type: none"> • 0: 跟踪无效 • 1: 跟踪有效

注意事项

- 原点复位中不能启动MC_Stop(强制停止)以外的FB。
- 对原点复位启动时的目标位置(Position)的软件行程限位检查如下所示。

原点复位方式	内容
驱动器式原点复位	不进行软件行程限位检查。
数据集式原点复位	进行软件行程限位检查。

- 原点复位启动时不进行启动位置的软件行程限位检查。
- 原点复位中，不进行软件行程限位的检查。
- 原点复位完成时的移动方向为正方向。

■驱动器式原点复位的注意事项

- 伺服OFF中，不能启动原点复位。因此，在伺服OFF中不能执行驱动器模块的原点复位方式(Homing method35、37(数据集式))。
- 在运动系统中进行了外部信号(轴变量中设置的硬件行程限位及原点开关(AbsSwitch)中设置的近点狗)的分配的情况下，将外部信号传输至驱动器模块。但是，将近点狗信号作为基准的(不是Z相基准)方式中需要位置精度的情况下，建议使用驱动器模块内置的DI。
- 在跟踪有效/无效选择(选项(Options):位16)中设置了“0:跟踪无效”的情况下，在驱动器式原点复位中，不进行指令当前位置的跟踪。此时，请勿将原点复位中的轴设置为主轴后执行同步控制。从轴中可能会发生控制中速度范围溢出(错误代码:1B1AH)。
- 在跟踪有效/无效选择(选项(Options):位16)中设置了“1:跟踪有效”，并将原点复位中的轴设置为主轴后执行了同步控制的情况下，从轴根据主轴数据源选择将变为下述的动作。

主轴数据源选择(MasterValueSource)	同步控制中的从轴动作
1: 指令当前(mcSetValue)101: 最新指令当前值(mcLatestSetValue)	原点复位结束后也继续进行同步控制。
2: 反馈值(mcActualValue) 102: 最新反馈值(mcLatestActualValue)	与主轴同步直到原点复位完成，但可能会因在结束时刻不进行同步控制且电机急加速或输入了过大指令，导致在驱动器模块侧发生错误或报警。关于详细内容，请参阅各驱动器模块的手册。 MR-J5的情况下，可能会发生伺服报警[AL. 031.1_伺服电机速度异常]或[AL. 035.1_指令频率异常]。

- 原点复位动作状态(AxisName.Md.Homing_Status)为“0002H:原点复位完成(未到达目标位置)”时发生了停止原因的情况下，跟踪的指令当前位置将大幅变动。因此，即使在主轴数据源选择(MasterValueSource)中，设置了“1:指令当前(mcSetValue)”，或“101:最新指令当前值(mcLatestSetValue)”的情况下，在驱动器模块侧也可能发生错误或伺服报警。关于详细内容，请参阅各驱动器模块的手册。

例:MR-J5的情况下，可能会发生伺服报警([AL.031.1(伺服电机速度异常)],或[AL.035.1(指令频率异常)])。

- 未对下述对象数据进行映射的情况下，将无法保证备份。

Home cycle counter(2D3DH)

Home ABS counter(2D3EH)

- MR-J5(W)-G连接时，可以将输入到伺服放大器中的DOG信号设置为原点开关信号。关于设置方法的详细内容，请参阅下述手册。

📖所使用的控制器的用户手册

- 原点复位中，进行硬件行程限位信号的检测，检测时向驱动器模块发送“HALT”信号。使用驱动器模块侧的限位开关信号进行停止的情况下，应将硬件行程限位倍率(AxisName.Cd.HwStrokeLimit_Override)置为“DISABLE(检查无效)”，将硬件行程限位检查置为暂时无效。
- 原点复位启动时，硬件行程限位倍率修调(AxisName.Cd.HwStrokeLimit_Override)中设置了“ONLY_INSIDE(仅向范围内方向检查无效)”的情况下，将发生启动不可(错误代码:1AFCH)，且不进行原点复位启动。

程序示例

将原点复位指令(bHomingCMD)置为TRUE，通过原点开关“[DEV](BOOL)D0”进行轴1(Axis0001)的原点复位的程序示例如下所示。

■轴

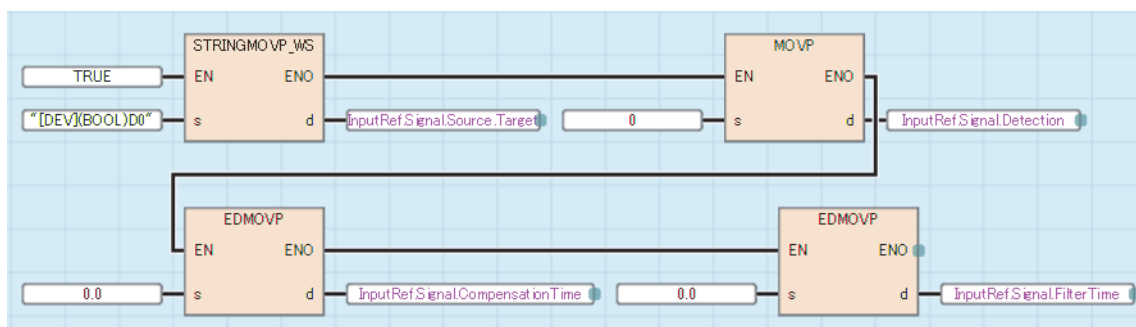
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

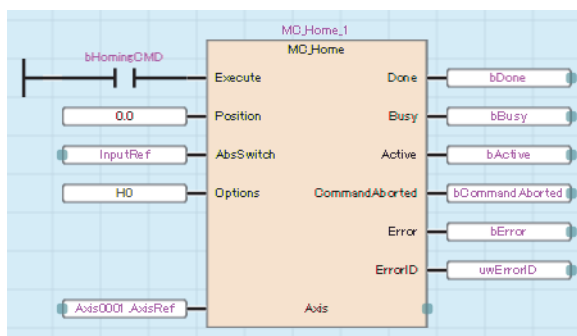
标签名	数据类型	注释
MC_Home_1	MC_Home	原点复位FB
bHomingCMD	位	原点复位指令
InputRef	MC_INPUT_REF	原点开关
bDone	位	执行完成
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

- 原点开关数据设置



- 原点复位



■ST程序

```
//-----原点开关数据设置-----
InputRef.Signal.Source.Target:= "[DEV] (BOOL) D0";
InputRef.Signal.Detection:= MC_SIGNAL_LOGIC__HighLevel;
InputRef.Signal.CompensationTime:= 0.0;
InputRef.Signal.FilterTime:= 0.0;

//-----原点复位-----
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
);
```

46.2 强制停止

MC_Stop

对指定的轴进行减速停止。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName, AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_Stop(强制停止)。
Deceleration	减速度	LREAL	启动时/可重启	0.0000、0.0001~2147483647.0	0.0	设置减速度。 根据加减速方式，设置内容有所不同。 关于详细内容，请参阅下述手册的“加减速处理功能”。 ☞ 所使用的控制器的用户手册 * 加减速方式将继承执行中的控制中指定的方式。
Jerk	Jerk	LREAL	启动时/可重启	0.0	0.0	应设置“0.0”。 * 设置了“0.0”以外的情况下，将变为超出Jerk范围(错误代码：1A8FH)。
Options	选项	DWORD(HEX)	启动时	00000000H	00000000H	应设置“00000000H”。 * 设置了“00000000H”以外的情况下，将变为超出Options范围(错误代码：1ABBH)。

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	到达了速度0时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_Stop(强制停止)时，将变为TRUE。
Active	控制中	BOOL	FALSE	MC_Stop(强制停止)正在控制轴时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MC_Stop(强制停止)的执行中断时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 ☞ 所使用的控制器的用户手册

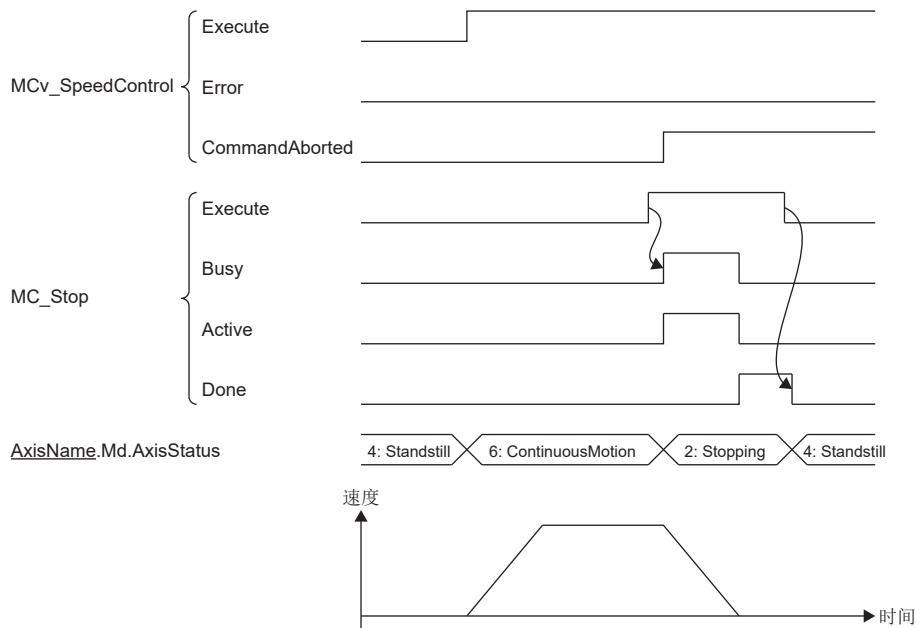
功能

- 对于MC_Stop(强制停止), 设置减速度(Deceleration), 并对当前控制中的FB进行减速停止。
- 执行MC_Stop(强制停止)时, 当前执行中的FB的执行中断(CommandAborted)将变为TRUE, 轴状态(AxisName.Md.AxisStatus)将转换为“2: 减速停止中(Stopping)”。执行指令(Execute)为TRUE时, 或速度未达到“0.0”时将保持“2: 减速停止中(Stopping)”。停止完成时执行完成(Done)为TRUE, 且执行指令(Execute)变为FALSE时将转换为“4: 待机中(Standstill)”。

■时序图

- 正常完成的情况下

MCv_SpeedControl(速度控制(包含位置循环))中执行了MC_Stop(强制停止)的情况下



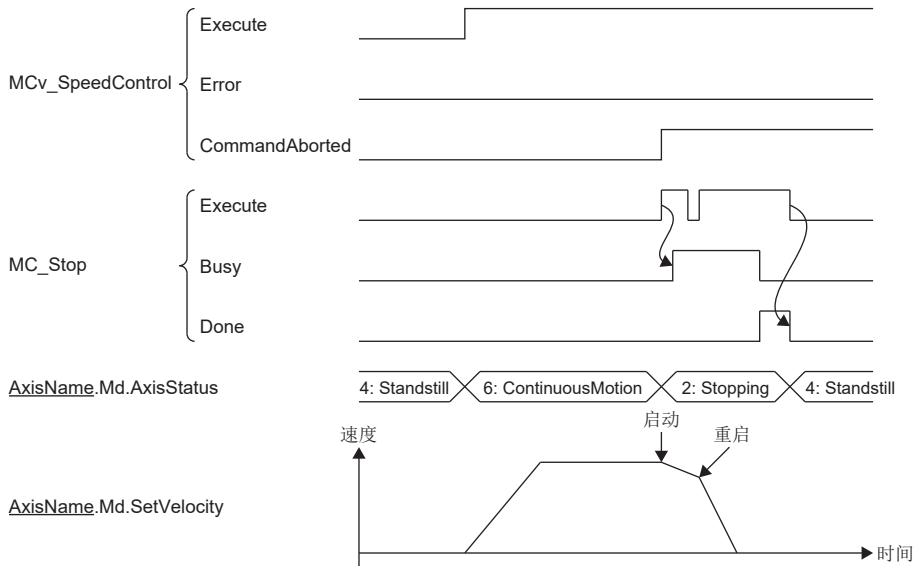
- 异常完成的情况下

关于异常完成时的时序图, 请参阅下述章节。

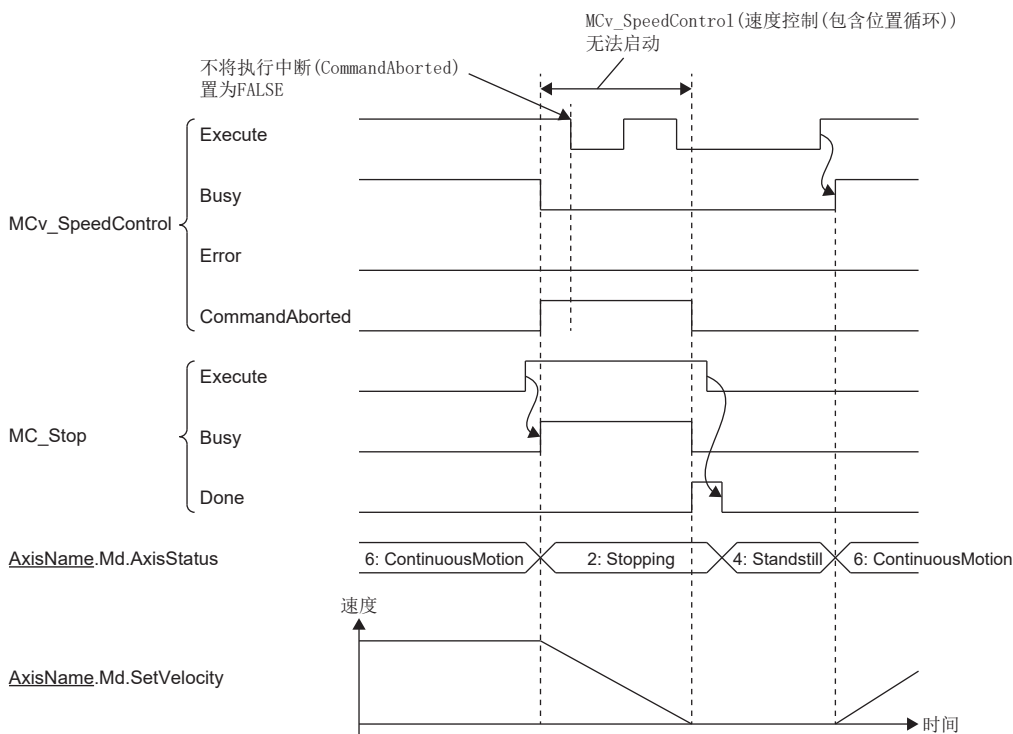
☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■动作概要

- 通过MC_Stop(强制停止)的减速停止中及输入变量的执行指令(Execute)为TRUE时,不受理MC_Stop(强制停止)以外的动作指令。
- 将减速度(Deceleration)设置为“0.0”或省略时将立即停止。
- 加减速方式及Jerk将沿用执行中的控制中设置的方式进行减速。
- 单轴同步控制中执行了MC_GroupStop(组强制停止)的情况下,至主轴的同步将中止。
- 更改减速度(Deceleration)的设定值重启了MC_Stop(强制停止)的情况下,从重启的时刻开始基于MC_Stop(强制停止)中设置的减速度(Deceleration)进行减速停止。



- 轴状态(AxisName.Md.AxisStatus)在“2: 减速停止中(Stopping)”中执行了MC_Stop(强制停止)以外的轴的运行FB的情况下,将变为不可启动(错误代码:1AFCH),MC_Stop(强制停止)的错误(Error)变为TRUE,轴状态(AxisName.Md.AxisStatus)为“1: 错误停止中(ErrorStop)”。
- 轴停止后,通过执行轴错误复位(AxisName.Cd.ErrorReset),将转变至“4: 待机中(Standstill)”。
- 即使将通过MC_Stop(强制停止)而执行中断的FB的执行指令(Execute)置为FALSE,一旦停止动作开始时在停止完成之前执行中断(CommandAborted)也将继续为TRUE。停止完成后,执行中断的FB的执行指令(Execute)为FALSE的情况下,执行中断(CommandAborted)将变为FALSE。



程序示例

将轴停止请求(bStop)置为TRUE, 并以减速度“100000.0”使轴1(Axis0001)减速停止的程序示例如下所示。

■轴

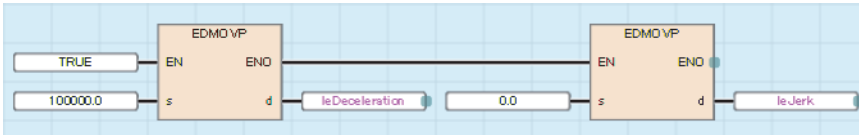
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

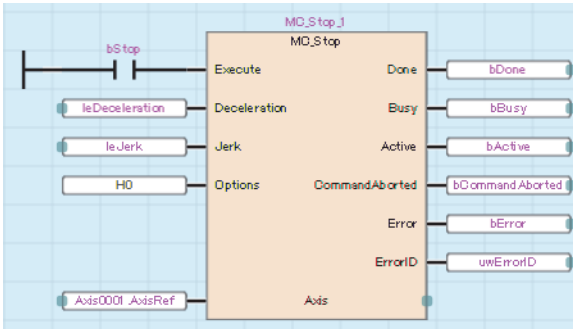
标签名	数据类型	注释
MC_Stop_1	MC_Stop	轴停止FB
bStop	位	轴停止请求
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bDone	位	完成
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

- 轴停止用数据设置



- 轴停止



■ST程序

//-----轴停止用数据设置-----

```
leDeceleration:= 100000.0;
```

```
leJerk:= 0.0;
```

//-----轴停止-----

```
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  
);
```

46.3 组强制停止

MC_GroupStop

减速停止指定的轴组。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
AxesGroup	轴组信息	AXES_GROUP_REF	启动时	—	不能省略	设置轴组。 关于使用的变量 (AxesGroupName, AxesGroupRef.)，请参阅下述内容。 ☞ 1408页 AxesGroupName, AxesGroupRef. (轴组信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_GroupStop(组强制停止)。
Deceleration	减速度	LREAL	启动时/可重启	0.0000、0.0001~2147483647.0	0.0	设置减速度。 根据加减速方式，设置内容有所不同。 关于详细内容，请参阅下述手册的“加减速处理功能”。 ☞ 所使用的控制器的用户手册 * 加减速方式将继承执行中的控制中指定的方式。
Jerk	Jerk	LREAL	启动时	0.0	0.0	应设置“0.0”。 * 设置了“0.0”以外的情况下，将变为超出Jerk范围(错误代码：1A8FH)。
Options	选项	DWORD (HEX)	启动时	0000000H	0000000H	应设置“0000000H”。 * 设置了“0000000H”以外的情况下，将变为超出Options范围(错误代码：1ABBH)。

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	到达了速度0时, 将变为TRUE。
Busy	执行中	BOOL	FALSE	至速度0的动作中时, 将变为TRUE。
Active	控制中	BOOL	FALSE	MC_GroupStop(组强制停止)正在控制轴时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MC_GroupStop(组强制停止)的执行中断时, 将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时, 将返回错误代码。 关于错误代码的详细内容, 请参阅下述手册。 📖所使用的控制器的用户手册

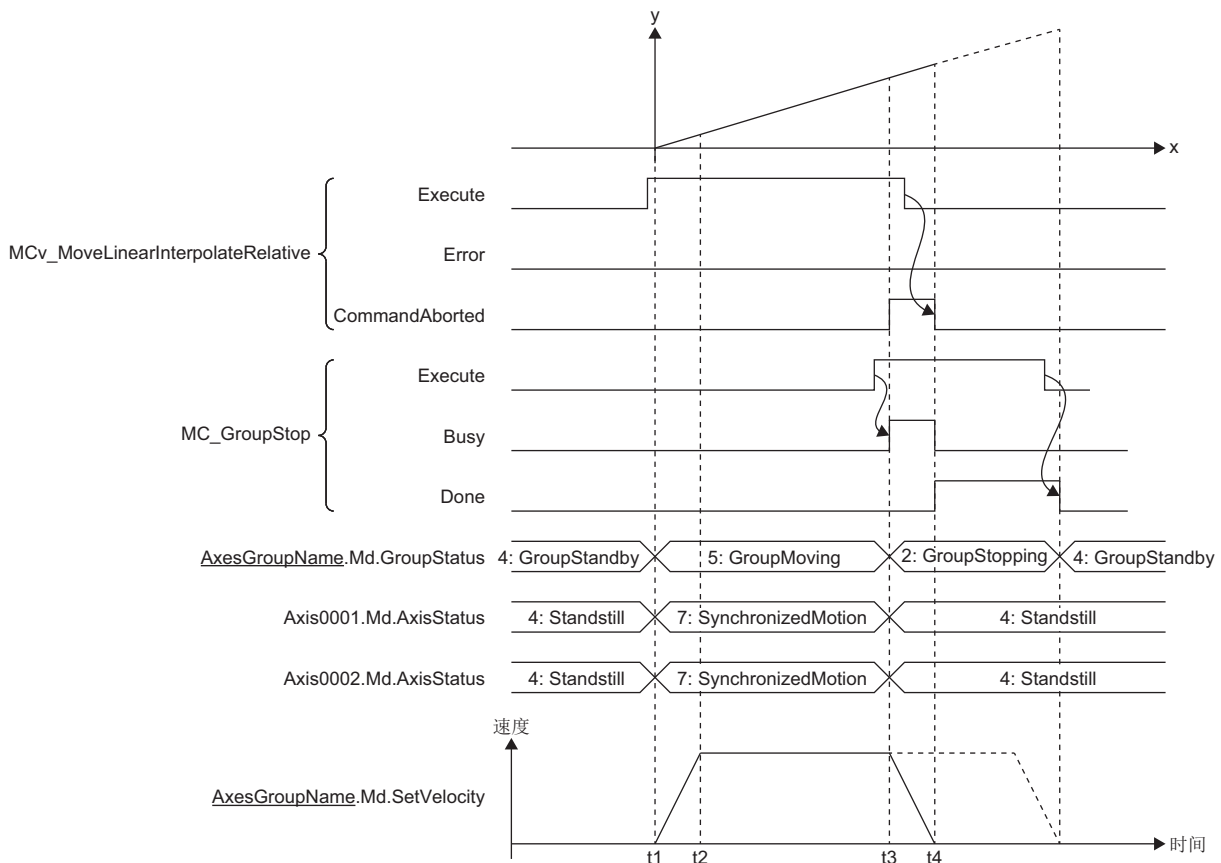
功能

- 对于MC_GroupStop(组强制停止), 设置减速度(Deceleration), 并对当前控制中的FB进行减速停止。轴组经过之前的动作的轨迹并进行减速停止。
- 执行MC_GroupStop(组强制停止)时, 当前执行中的FB的执行中断(CommandAborted)将变为TRUE, 轴组状态(AxesGroupName.Md.GroupStatus)将转换为“2: 减速停止中(GroupStopping)”。执行指令(Execute)为TRUE时, 或速度未达到“0.0”时将保持“2: 减速停止中(GroupStopping)”。停止完成时执行完成(Done)为TRUE, 且执行指令(Execute)变为FALSE时将转变为“4: 待机中(GroupStandby)”。

■时序图

- 正常完成的情况下

MCv_MoveLinearInterpolateRelative中(相对值直线插补控制)执行了MC_GroupStop(组强制停止)的情况下



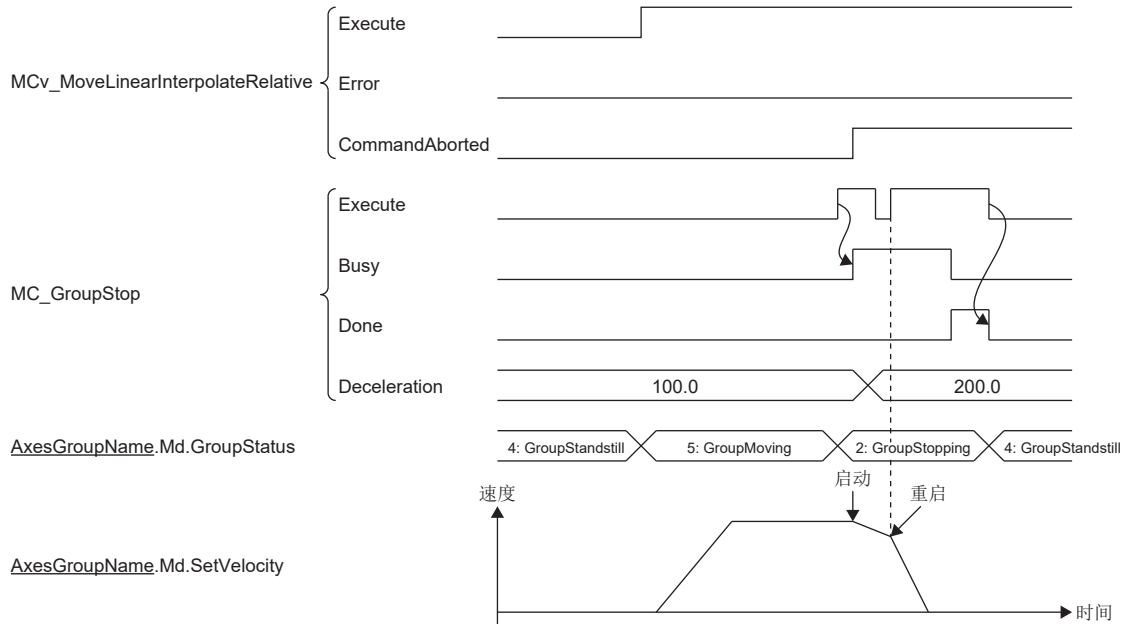
- 异常完成的情况下

关于异常完成时的时序图, 请参阅下述章节。

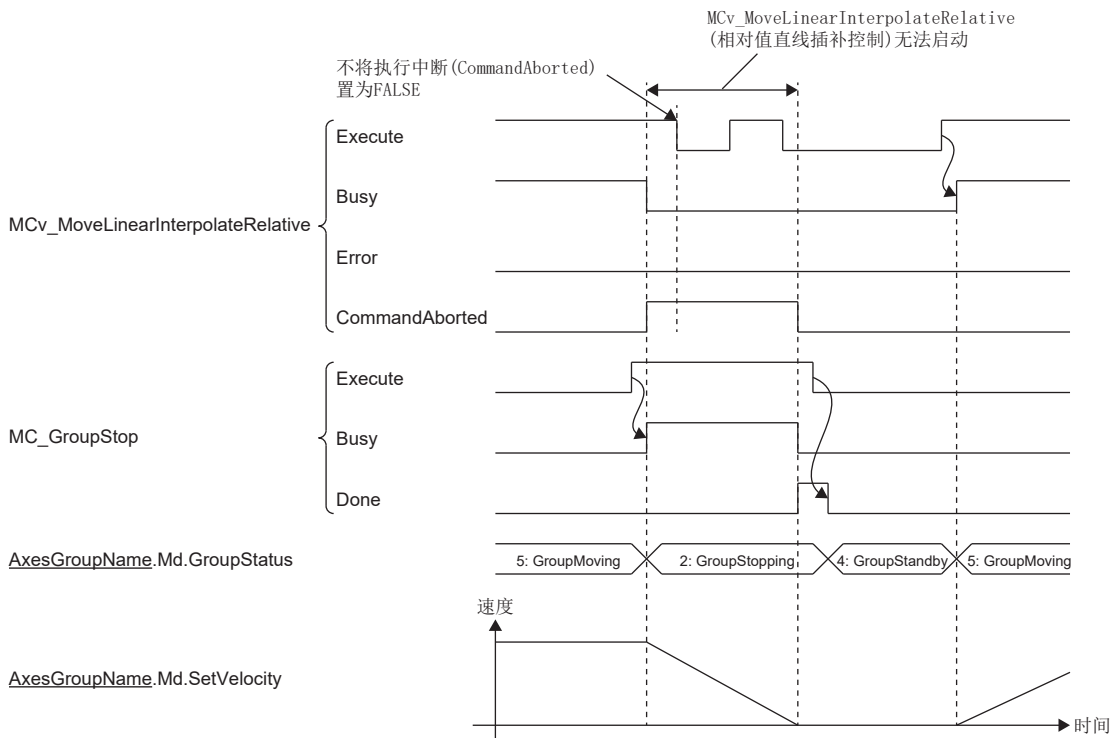
📖 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■动作概要

- 通过MC_GroupStop(组强制停止)的减速停止中及输入变量的执行指令(Execute)为TRUE时, 不受理MC_GroupStop(组强制停止)以外的动作指令。
- 将减速度(Deceleration)设置为“0.0”或省略时将立即停止。
- 加减速方式及Jerk将沿用执行中的控制中指定的方式进行减速。
- 更改减速度(Deceleration)的设定值重启了MC_GroupStop(组强制停止)的情况下, 从重启的时刻开始基于MC_GroupStop(组强制停止)中设置的减速度(Deceleration)、控制中的FB中使用的Jerk进行减速停止。



- 轴组状态 (AxesGroupName.Md.GroupStatus)为“2: 减速停止中(GroupStopping)”中执行了MC_GroupStop(组强制停止)以外的轴的运行FB的情况下, 将变为不可启动(错误代码: 1AFCH), MC_GroupStop(组强制停止)的错误(Error)变为TRUE, 轴组状态 (AxesGroupName.Md.GroupStatus)变为“1: 错误停止中(GroupErrorStop)”。
- 即使将通过MC_GroupStop(组强制停止)而执行中断的FB的执行指令(Execute)置为FALSE, 一旦停止动作开始时在停止完成之前执行中断(CommandAborted)也将继续为TRUE。停止完成后, 执行中断的FB的执行指令(Execute)为FALSE的情况下, 执行中断(CommandAborted)将变为FALSE。



程序示例

将轴组停止请求(bGroupStop)置为TRUE,并以减速度“100000.0”使轴组1(AxesGroup001)减速停止的程序示例如下所示。

■轴组

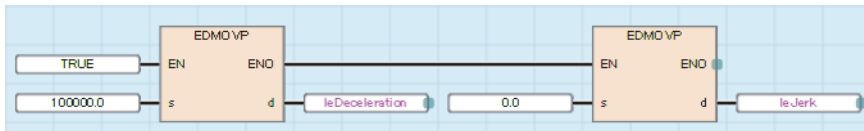
轴组No.	标签名	数据类型	注释
1	AxesGroup001	AXES_GROUP_REF	轴组1

■使用的标签

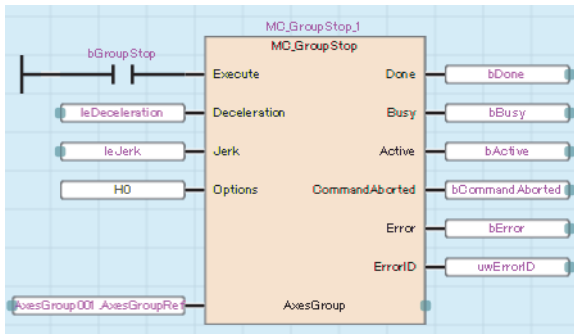
标签名	数据类型	注释
MC_GroupStop_1	MC_GroupStop	轴组停止FB
bGroupStop	位	轴组停止请求
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bDone	位	完成
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

- 轴组停止用数据设置



- 轴组停止



■ST程序

```
//-----轴组停止用数据设置-----  
leDeceleration:= 100000.0;  
leJerk:= 0.0;  
  
//-----轴组停止-----  
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  
);
```

46.4 绝对值定位

MC_MoveAbsolute

设置绝对位置的目标位置，执行定位。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName, AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_MoveAbsolute(绝对值定位)。
ContinuousUpdate	连续更新	BOOL	启动时	TRUE、FALSE	FALSE	设置将目标位置(Position)、速度(Velocity)、加速度(Acceleration)、减速度(Deceleration)的连续更改是置为有效，还是置为无效。 • FALSE: 无效 • TRUE: 有效
Position	目标位置	LREAL	启动时/可重启/可连续更新	-10000000000.0~10000000000.0	0.0	设置绝对位置的目标位置。 可设置的范围根据各设置而有所不同。 关于详细内容，请参阅下述章节。 ☞ 1600页 目标位置(Position)
Velocity	速度	LREAL	启动时/可重启/可连续更新	0.0、0.0001~2500000000.0	0.0	设置速度。 关于详细内容，请参阅下述章节。 ☞ 1601页 速度(Velocity)
Acceleration	加速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置加速度。 关于详细内容，请参阅下述章节。 ☞ 1601页 加速度(Acceleration)
Deceleration	减速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置减速度。 关于详细内容，请参阅下述章节。 ☞ 1601页 减速度(Deceleration)
Jerk	Jerk	LREAL	启动时	0.0000、0.0001~2147483647.0	0.0	设置Jerk。 关于详细内容，请参阅下述章节。 ☞ 1601页 Jerk

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Direction	方向选择	INT (MC_DIRECTION)	启动时	1~3	0	软件行程限位无效时，设置从当前位置向目标位置移动的方向。 <ul style="list-style-type: none"> • 1: 正方向 (mcPositiveDirection) • 2: 负方向 (mcNegativeDirection) • 3: 最短路径 (mcShortestWay) * 省略了设置的情况下，将变为超出方向选择范围 (错误代码: 1AA5H)。 关于详细内容，请参阅下述章节。 ☞ 1602页 方向选择 (Direction)
BufferMode	缓冲模式	INT (MC_BUFFER_MODE)	启动时	0~5	0	设置缓冲模式。 <ul style="list-style-type: none"> • 0: Aborting (mcAborting) • 1: Buffered (mcBuffered) • 2: BlendingLow (mcBlendingLow) • 3: BlendingPrevious (mcBlendingPrevious) • 4: BlendingNext (mcBlendingNext) • 5: BlendingHigh (mcBlendingHigh) 关于详细内容，请参阅下述章节。 ☞ 1605页 缓冲模式 (BufferMode)
Options	选项	DWORD (HEX)	启动时	00000000H~00010021H	00000000H	设置MC_MoveAbsolute (绝对值定位) 的功能选项。 关于详细内容，请参阅下述章节。 ☞ 1606页 选项 (Options)

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	到达了目标位置时，将变为TRUE。
Busy	执行中	BOOL	FALSE	正在执行MC_MoveAbsolute (绝对值定位) 时，将变为TRUE。 到达了目标位置后，将变为FALSE。
Active	控制中	BOOL	FALSE	MC_MoveAbsolute (绝对值定位) 正在控制轴时，将变为TRUE。 到达了目标位置后，将变为FALSE。
CommandAborted	执行中断	BOOL	FALSE	MC_MoveAbsolute (绝对值定位) 的执行中断时，将变为TRUE。 执行指令 (Execute) 变为FALSE时，将变为FALSE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 ☞ 所使用的控制器的用户手册

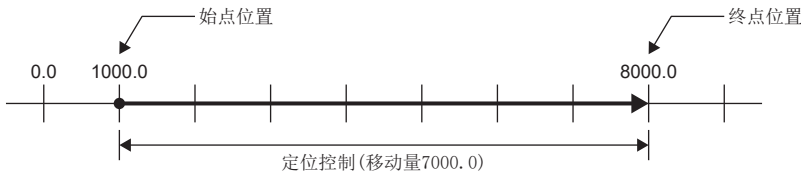
功能

- 设置目标位置(Position)、速度(Velocity)、加速度(Acceleration)、减速度(Deceleration)、Jerk、方向选择(Direction)、缓冲模式(BufferMode)、选项(Options)，并从启动时的当前位置(始点位置)向目标位置(Position)中设置的指定位置(终点位置)进行定位。

例

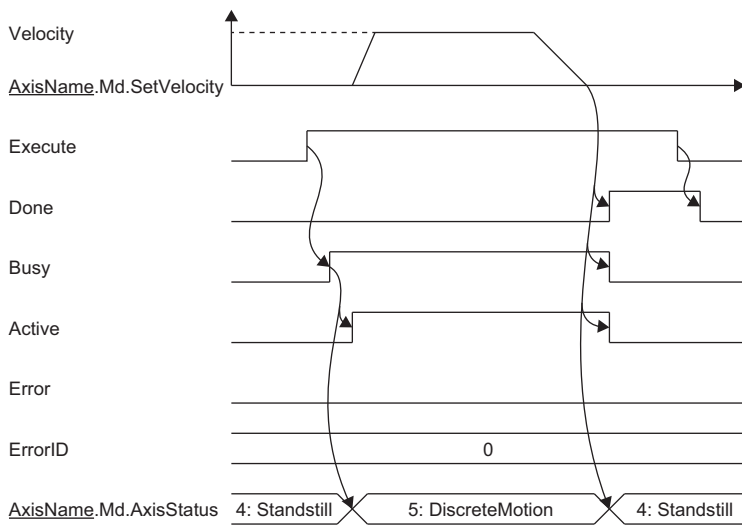
始点位置(当前的停止位置)为“1000.0”时，在目标位置(Position)中设置了“8000.0”的情况下

- 向正方向进行移动量为“7000.0(8000.0 - 1000.0)”的定位。



■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■目标位置(Position)

设置绝对位置的目标位置。有效范围根据软件行程限位有效/无效、方向选择(Direction)、超出环形计数器的目标位置指定(选项(Options):位16)而有所不同。

- 软件行程限位有效时(软件行程限位对象(AxisName.Md.SwStrokeLimit_Target)为“-1:无效(Invalid)”以外时)与方向选择(Direction)、超出环形计数器的目标位置指定(选项(Options):位16)的设置无关,有效范围如下所示。

有效范围
环形计数器下限值 ≤ 目标位置 < 环形计数器上限值

设置了超出范围的值的情况下,将变为超出目标位置范围(错误代码:1A85H),不启动。此外,即使在有效范围内将软件行程限位溢出的位置设置为目标位置的情况下,也将变为软件行程限位溢出(目标位置)(错误代码:1A80H)且不启动。

- 软件行程限位无效时(软件行程限位对象(AxisName.Md.SwStrokeLimit_Target)为“-1:无效(Invalid)”时)通过将“超出环形计数器的目标位置指定(选项(Options):位16)”设置为“1:允许”,可以进行环形计数器范围1旋转及以下的定位。

设置了超出范围的值的情况下,将变为超出目标位置范围(错误代码:1A85H),不启动。

方向选择(Direction)	超出环形计数器的目标位置指定(选项(Options):位16)	有效范围
1: 正方向(mcPositiveDirection)	0: 不允许	环形计数器下限值 ≤ 目标位置 < 环形计数器上限值
	1: 允许	环形计数器下限值 ≤ 目标位置 < 定位范围上限值
2: 负方向(mcNegativeDirection)	0: 不允许	环形计数器下限值 ≤ 目标位置 < 环形计数器上限值
	1: 允许	定位范围下限值 ≤ 目标位置 < 环形计数器上限值
3: 最短路径(mcShortestWay)	0: 不允许	环形计数器下限值 ≤ 目标位置 < 环形计数器上限值
	1: 允许	环形计数器下限值 ≤ 目标位置 < 环形计数器上限值

- 关于向目标位置移动的方向,请参阅下述章节。

☞ 1602页 方向选择(Direction)

■速度 (Velocity)

设置MC_MoveAbsolute(绝对值定位)中的指令速度。

设置范围
0.0、0.0001~2500000000.0 ^{*1*2}

- *1 由于进行浮点运算，因此指令速度的下限值中将产生下述限制。
对指令速度进行了运算周期换算的速度小于“0.00001”的情况下，将变为超出运算周期换算速度范围(错误代码：1B16H)(速度更改时为超出运算周期换算速度范围警告(事件代码：00D2FH))。为了提高浮点运算的精度，应通过更改位置指令单位(AxisName.Pr.Unit_Position)或速度指令单位(AxisName.Pr.Unit_Velocity)设置运算周期换算后的速度不低于“0.00001”。
- *2 在进行多重启动的运动控制FB中指定速度为“0.0”的情况下，将变为之前的运动控制FB的指定速度。

■加速度 (Acceleration)

设置MC_MoveAbsolute(绝对值定位)中的加速度。

根据加减速方式设置(选项(Options)：位0~2)的设置，设置范围有所不同。

加减速方式设置(选项(Options)：位0~2)	设置范围
0：加减速速度指定方式(mcAccDec)	0.0000 ^{*1} 、0.0001~2147483647.0[U/s ²]的正数
1：加减速时间恒定方式(mcFixedTime)	0.000000 ^{*1} 、0.000001~8400.0[s]的正数

- *1 启动时，根据启动时加减速速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior)，动作将发生变化。加速度更改、减速度更改时，不受理更改。

■减速度 (Deceleration)

设置MC_MoveAbsolute(绝对值定位)中的减速度。

设置了加减速方式设置(选项(Options)：位0~2)的“0：加减速速度指定方式(mcAccDec)”的情况下，对设置范围进行设置。设置了“1：加减速时间恒定方式(mcFixedTime)”的情况下，不使用减速度(Deceleration)。

加减速方式设置(选项(Options)：位0~2)	设置范围
0：加减速速度指定方式(mcAccDec)	0.0000 ^{*1} 、0.0001~2147483647.0[U/s ²]的正数
1：加减速时间恒定方式(mcFixedTime)	不使用

- *1 启动时，根据启动时加减速速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior)，动作将发生变化。加速度更改、减速度更改时，不受理更改。

■Jerk

设置MC_MoveAbsolute(绝对值定位)中的Jerk。

设置了加减速方式设置(选项(Options)：位0~2)的“0：加减速速度指定方式(mcAccDec)”的情况下，对设置范围进行设置。设置了“1：加减速时间恒定方式(mcFixedTime)”的情况下，不使用Jerk。

加减速方式设置(选项(Options)：位0~2)	设置范围
0：加减速速度指定方式(mcAccDec)	0.0000、0.0001~2147483647.0[U/s ³]的正数
1：加减速时间恒定方式(mcFixedTime)	不使用

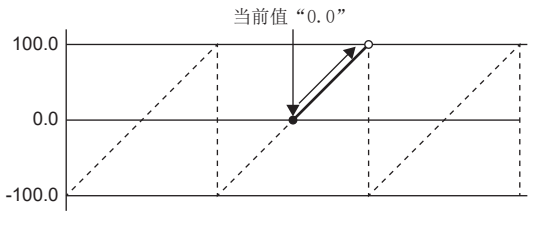
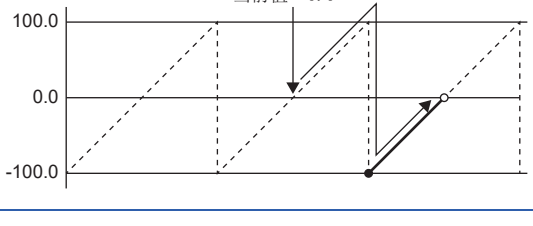
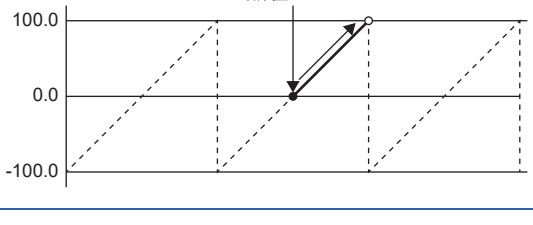
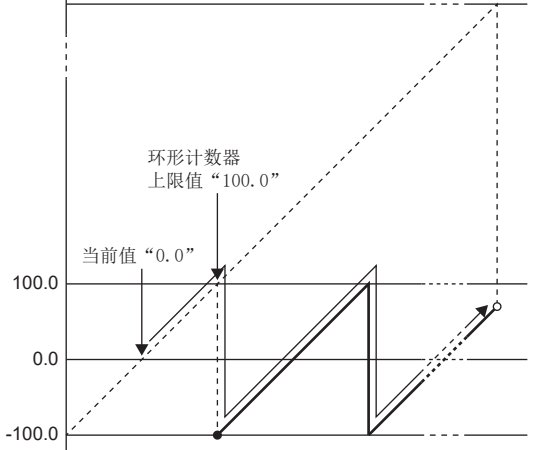
■方向选择(Direction)

软件行程限位有效时，忽略方向选择(Direction)的设置。以不超出软件行程限位范围的方向进行定位控制。但是，正方向、负方向均不超出软件行程限位范围的情况下，以当前位置为基准，以趋近目标位置的方向(移动量的绝对值较短的一方)进行定位控制。正方向、负方向距离相同的情况下以当前方向执行动作。

软件行程限位无效时，可以从方向选择(Direction)的“1: 正方向(mcPositiveDirection)”、“2: 负方向(mcNegativeDirection)”、“3: 最短路径(mcShortestWay)”中选择，并设置从当前位置向目标位置移动的方向。

• 1: 正方向(mcPositiveDirection)

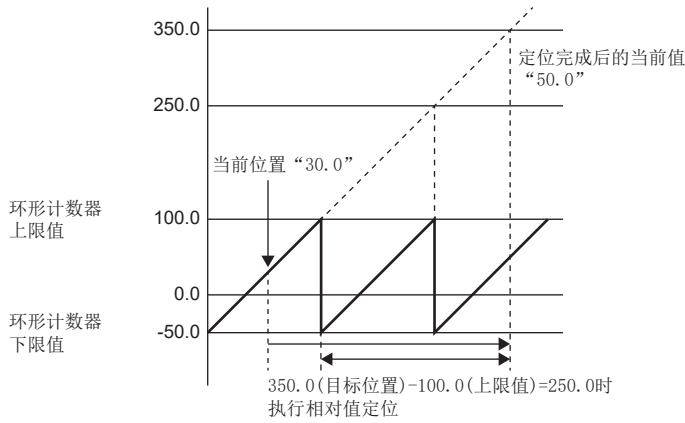
从当前位置向目标位置进行正方向(地址增加)的定位。

超出环形计数器的目标位置指定(选项(Options): 位16)	设置范围	动作
<p>■0: 不允许 目标位置的设置范围为“环形计数器下限值 ≤ 目标位置 < 环形计数器上限值”。</p>	当前值 ≤ 目标位置 < 环形计数器上限值	<p>环形计数器上限值</p>  <p>当前值“0.0”</p> <p>环形计数器下限值</p>
	环形计数器下限值 ≤ 目标位置 < 当前值	<p>环形计数器上限值</p>  <p>当前值“0.0”</p> <p>环形计数器下限值</p>
<p>■1: 允许 可以进行超出环形计数器上限值的定位运行。在此情况下，以环形计数器上限值为基准，将超出部分的移动量作为相对移动量进行定位。设置了小于环形计数器下限值的目标位置的情况下，将变为超出目标位置范围(错误代码: 1A85H)，不启动。</p>	当前值 ≤ 目标位置 < 环形计数器上限值 环形计数器下限值 ≤ 目标位置 < 当前值	<p>环形计数器上限值</p>  <p>当前值“0.0”</p> <p>环形计数器下限值</p>
	环形计数器上限值 ≤ 目标位置 < 定位范围上限值	<p>定位范围上限值</p>  <p>环形计数器上限值“100.0”</p> <p>当前值“0.0”</p> <p>环形计数器上限值</p> <p>环形计数器下限值</p>

例

设置了下述值时的动作(进行超出环形计数器上限值后的相对值定位。)

- 环形计数器上限值: 100.0
- 环形计数器下限值: -50.0
- 当前位置: 30.0
- 目标位置: 350.0



- 2: 负方向(mcNegativeDirection)

从当前位置向负方向(地址减少)的目标位置进行定位。

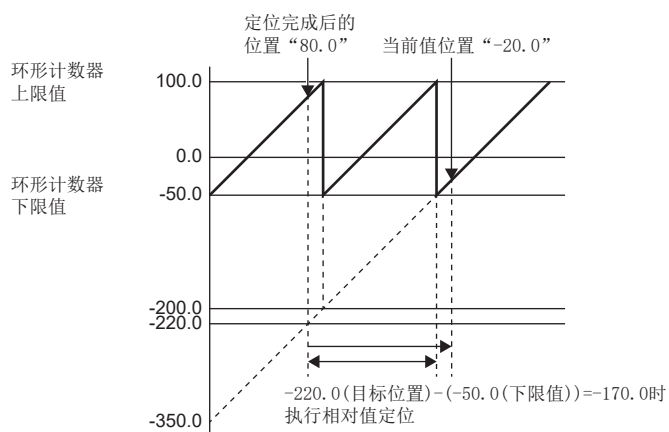
超出环形计数器的目标位置指定(选项(Options): 位16)	设置范围	动作
■0: 不允许 目标位置的设置范围为“环形计数器下限值 ≤ 目标位置 < 环形计数器上限值”。	环形计数器下限值 ≤ 目标位置 ≤ 当前值	环形计数器上限值 环形计数器下限值
	当前值 < 目标位置 < 环形计数器上限值	环形计数器上限值 环形计数器下限值

超出环形计数器的目标位置指定(选项(Options): 位16)	设置范围	动作
<p>■1: 允许</p> <p>可以进行低于环形计数器下限值的定位运行。在此情况下, 以环形计数器下限值为基准, 将超出部分的移动量作为相对移动量进行定位。设置了大于环形计数器上限值的目标位置的情况下, 将变为超出目标位置范围(错误代码: 1A85H), 不启动。</p>	<ul style="list-style-type: none"> • 环形计数器下限值 ≤ 目标位置 ≤ 当前值 • 当前值 < 目标位置 < 环形计数器上限值 	<p>环形计数器上限值</p> <p>当前值 “0.0”</p> <p>环形计数器下限值</p>
	<p>定位范围下限值 ≤ 目标位置 < 环形计数器下限值</p>	<p>环形计数器上限值</p> <p>环形计数器下限值 “-100.0”</p> <p>当前值 “0.0”</p> <p>定位范围下限值</p>

例

设置了下述值时的动作

- 设置了下述值时的动作
- 环形计数器上限值: 100.0
- 环形计数器下限值: -50.0
- 当前位置: -20.0
- 目标位置: -220.0



• 3: 最短路径 (mcShortestWay)

以当前位置为基准，以趋近目标位置的方向（移动量的绝对值较短的一方）进行定位控制。正方向、负方向距离相同的情况下以当前方向执行动作。与超出环形计数器的目标位置指定（选项（Options）：位16）的设置无关，目标位置的设置范围为“环形计数器下限值 ≤ 目标位置 < 环形计数器上限值”。设置了超出范围的值的情况下，将变为超出目标位置范围（错误代码：1A85H），不启动。

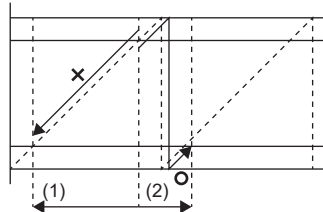
例

设置了下述值时的动作

- 环形计数器上限值：100.0
- 环形计数器下限值：-100.0
- 当前位置：80.0
- 目标位置：-80.0

环形计数器上限值：100.0
当前值：80.0

目标位置：-80.0
环形计数器下限值：-100.0



*1 (1) 的情况下移动量变为“160.0”，(2) 的情况下移动量变为“40.0”，因此以(2)的正方向移动。

■缓冲模式 (BufferMode)

设置用于进行多重启动（缓冲模式）的动作。

MC_MoveAbsolute（绝对值定位）中可设置的缓冲模式如下所示。

设定值	内容
0: Aborting (mcAborting)	中断（取消）控制中的FB并立即执行下一个FB。
1: Buffered (mcBuffered)	在控制中的FB中缓冲下一个FB。 控制中的FB已经被缓冲的情况下，对之前的FB的下一个进行缓冲。（最多2个） 控制中的FB完成时，依次执行缓冲FB。
2: BlendingLow (mcBlendingLow)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下，对之前的FB的下一个进行缓冲。（最多2个） 控制中的FB到达目标位置后，依次执行缓冲FB。 控制中的FB到达了目标位置时，将控制中的FB与缓冲FB的目标速度中较低一方的速度作为切换速度。
3: BlendingPrevious (mcBlendingPrevious)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下，对之前的FB的下一个进行缓冲。（最多2个） 控制中的FB到达目标位置后，依次执行缓冲FB。 控制中的FB到达了目标位置时，将控制中的FB的目标速度作为切换速度。
4: BlendingNext (mcBlendingNext)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下，对之前的FB的下一个进行缓冲。（最多2个） 控制中的FB到达目标位置后，依次执行缓冲FB。 控制中的FB到达了目标位置时，将缓冲FB的目标速度作为切换速度。
5: BlendingHigh (mcBlendingHigh)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下，对之前的FB的下一个进行缓冲。（最多2个） 控制中的FB到达目标位置后，依次执行缓冲FB。 控制中的FB到达了目标位置时，将控制中的FB与缓冲FB的目标速度中较高一方的速度作为切换速度。

*1 在控制中的FB与缓冲FB之间不执行停止。

要点

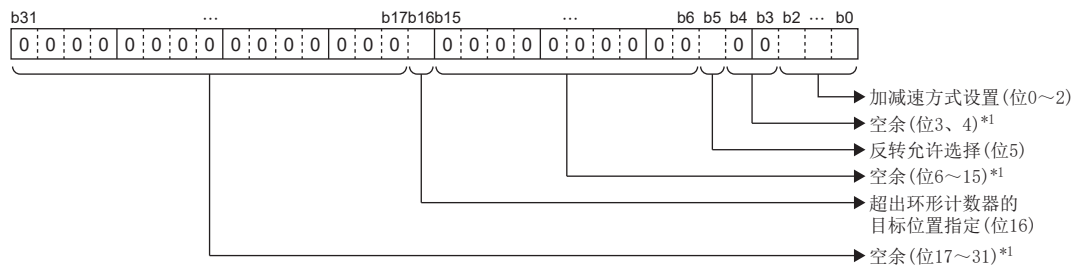
关于多重启动（缓冲模式）的详细内容，请参阅下述手册。

所使用的控制器的用户手册

■选项 (Options)

将MC_MoveAbsolute (绝对值定位) 中使用的功能选项以位指定进行设置。

以位指定设置的内容如下所示。



*1 空余中应设置“0”。设置了“0”以外的情况下，将变为超出Options范围(错误代码：1ABBH)。

位	名称	内容
0~2	加减速方式设置	设置用于进行控制的加减速方式。 <ul style="list-style-type: none"> 0: 加减速速度指定方式 (mcAccDec) 1: 加减速时间恒定方式 (mcFixedTime)
5	反转允许选择	设置执行中的FB与要缓冲的FB中动作方向不相同的情况下是否允许反转。 <ul style="list-style-type: none"> 0: 允许 1: 不允许 *: 轴动作中切换为后续缓冲FB的情况下，或目标位置更改时有效。
16	超出环形计数器的目标位置指定	设置软件行程限位无效时，是否允许超出环形计数器上限值、环形计数器下限值的目标位置。 <ul style="list-style-type: none"> 0: 不允许 1: 允许

• 加减速方式设置 (位0~2)

设定值	内容
0: 加减速速度指定方式 (mcAccDec)	是使用MC_MoveAbsolute (绝对值定位) 中设置的加速度 (Acceleration)、减速度 (Deceleration)、Jerk进行加速/减速的方式。
1: 加减速时间恒定方式 (mcFixedTime)	是与速度无关，使用MC_MoveAbsolute (绝对值定位) 中设置的加减速时间进行加速/减速的方式。对于加减速时间，设置为加速度 (Acceleration)，不使用减速度 (Deceleration)、Jerk。

• 反转允许选择 (位5)

设定值	内容
0: 允许	允许反转。 进行减速停止一次，减速停止完成后，向更改的方向开始动作。
1: 不允许	不允许反转。 切换时将变为超载运行错误 (错误代码：1AE3H) 且进行减速停止。

■必要对象数据

使用MC_MoveAbsolute (绝对值定位) 的情况下，应在轴中设置下述对象数据。

- Target position (607AH)

未设置对象数据的情况下，将变为必须从站对象未设置 (错误代码：1AF7H) 且不启动。

关于对象数据设置的详细内容，请参阅下述手册。

📖 所使用的控制器的用户手册

程序示例

将单轴绝对值定位启动(bMoveAbsolute)置为TRUE，并根据下述设置进行轴1(Axis0001)的绝对值定位控制的程序示例如下所示。

• 动作



• 设置

项目	设定值
目标位置	80000.0
速度	50000.0
加速度	100000.0
减速度	100000.0
Jerk	0.0

■ 轴

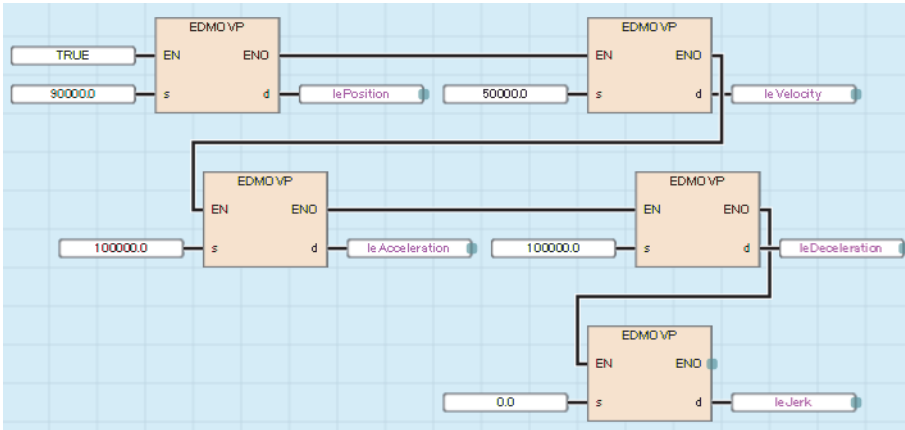
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■ 使用的标签

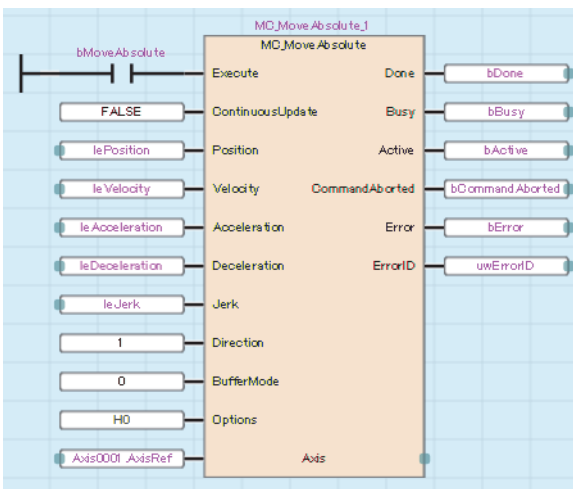
标签名	数据类型	注释
MC_MoveAbsolute_1	MC_MoveAbsolute	绝对值定位FB
bMoveAbsolute	位	单轴绝对值定位启动
lePosition	双精度实数	目标位置
leVelocity	双精度实数	速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bDone	位	执行完成
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

- 单轴定位控制用数据的设置



- 绝对值定位



■ST程序

```
//-----单轴定位控制用数据的设置-----
lePosition:= 80000.0;
leVelocity:= 50000.0;
leAcceleration:= 100000.0;
leDeceleration:= 100000.0;
leJerk:= 0.0;

//-----绝对值定位-----
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
);
```

46.5 相对值定位

MC_MoveRelative

设置相对位置的移动量，执行定位。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName, AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_MoveRelative(相对值定位)。
ContinuousUpdate	连续更新	BOOL	启动时	TRUE、FALSE	FALSE	设置将移动量(Distance)、速度(Velocity)、加速度(Acceleration)、减速度(Deceleration)的连续更改是置为有效，还是置为无效。 • FALSE: 无效 • TRUE: 有效
Distance	移动量	LREAL	启动时/可重启/可连续更新	-10000000000.0~10000000000.0	0.0	设置从启动时的当前位置至终点的相对位置。 关于详细内容，请参阅下述章节。 ☞ 1612页 移动量(Distance)
Velocity	速度	LREAL	启动时/可重启/可连续更新	0.0、0.0001~2500000000.0	0.0	设置速度。 关于详细内容，请参阅下述章节。 ☞ 1612页 速度(Velocity)
Acceleration	加速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置加速度。 关于详细内容，请参阅下述章节。 ☞ 1613页 加速度(Acceleration)
Deceleration	减速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置减速度。 关于详细内容，请参阅下述章节。 ☞ 1613页 减速度(Deceleration)
Jerk	Jerk	LREAL	启动时	0.0000、0.0001~2147483647.0	0.0	设置Jerk。 关于详细内容，请参阅下述章节。 ☞ 1613页 Jerk

输入变量	名称	数据类型	获取	有效范围	初始值	内容
BufferMode	缓冲模式	INT (MC_BUFFER_MODE)	启动时	0~5	0	设置缓冲模式。 <ul style="list-style-type: none"> • 0: Aborting (mcAborting) • 1: Buffered (mcBuffered) • 2: BlendingLow (mcBlendingLow) • 3: BlendingPrevious (mcBlendingPrevious) • 4: BlendingNext (mcBlendingNext) • 5: BlendingHigh (mcBlendingHigh) 关于详细内容, 请参阅下述章节。 1613页 缓冲模式(BufferMode)
Options	选项	DWORD (HEX)	启动时	00000000H~00000021H	00000000H	设置MC_MoveRelative(相对值定位)的功能选项。 关于详细内容, 请参阅下述章节。 1614页 选项(Options)

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	到达了相对位置时, 将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_MoveRelative(相对值定位)时, 将变为TRUE。 到达了相对位置后, 将变为FALSE。
Active	控制中	BOOL	FALSE	MC_MoveRelative(相对值定位)正在控制轴时, 将变为TRUE。 到达了相对位置后, 将变为FALSE。
CommandAborted	执行中断	BOOL	FALSE	MC_MoveRelative(相对值定位)的执行中断时, 将变为TRUE。 执行指令(Execute)变为FALSE时, 将变为FALSE。
Error	错误	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时, 将返回错误代码。 关于错误代码的详细内容, 请参阅下述手册。 所使用的控制器的用户手册

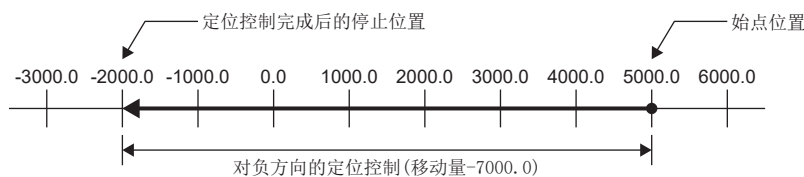
功能

- 设置移动量(Distance)、速度(Velocity)、加速度(Acceleration)、减速度(Deceleration)、Jerk、缓冲模式(BufferMode), 并从启动时的当前位置(始点位置)进行移动量(Distance)中设置的移动量的定位。运行中的方向取决于移动量的符号。轴状态(AxisName.Md.AxisStatus)变为“5: 定位运行中(DiscreteMotion)”。

例

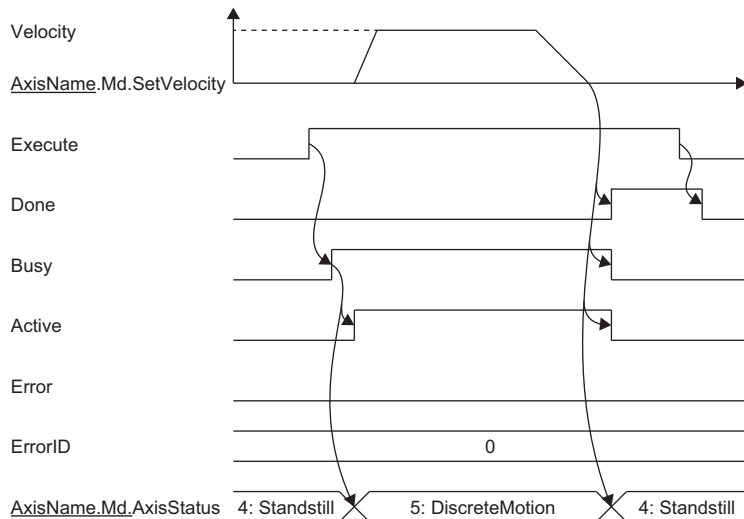
始点位置(当前的停止位置)为“5000.0”时, 在移动量中设置了“-7000.0”的情况下

- 进行至“-2000.0”的定位。



■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■移动量(Distance)

设置从启动时的当前位置到终点为止的移动量。

移动方向取决于移动量的符号(+/-)。

移动量	内容
正(+)的情况下	向正方向(地址增加方向)移动。
负(-)的情况下	向负方向(地址减少方向)移动。

移动量为“0”的情况下，轴不动作但执行完成(Done)将变为TRUE。

移动量设置为超出范围的值的情况下，将变为超出目标位置范围(错误代码：1A85H)且不启动。

软件行程限位有效时，“当前值+移动量”超出软件行程限位上限值、软件行程限位下限值的情况下，将变为软件行程限位溢出(目标位置)(错误代码：1A80H)且不启动。

■速度(Velocity)

设置MC_MoveRelative(相对值定位)中的指令速度。

设置范围
0.0、0.0001~2500000000.0*1*2

- *1 由于进行浮点运算，因此指令速度的下限值中将产生下述限制。
对指令速度进行了运算周期换算的速度小于“0.00001”的情况下，将变为超出运算周期换算速度范围(错误代码：1B16H)(速度更改时为超出运算周期换算速度范围警告(事件代码：00D2FH))。为了提高浮点运算的精度，应通过更改位置指令单位(AxisName.Pr.Unit_Position)或速度指令单位(AxisName.Pr.Unit_Velocity)设置运算周期换算后的速度不低于“0.00001”。
- *2 在进行多重启动的运动控制FB中指定速度为“0.0”的情况下，将变为之前的运动控制FB的指定速度。

■加速度 (Acceleration)

设置MC_MoveRelative (相对值定位)中的加速度。

根据加减速方式设置 (选项 (Options): 位0~2) 的设置, 设置范围有所不同。

加减速方式设置 (选项 (Options): 位0~2)	设置范围
0: 加减速速度指定方式 (mcAccDec)	0.0000* ¹ 、0.0001~2147483647.0[U/s ²]的正数
1: 加减速时间恒定方式 (mcFixedTime)	0.000000* ¹ 、0.000001~8400.0[s]的正数

*1 启动时, 根据启动时加减速速度0指定时动作选择 (AxisName.Pr.AccelerationZeroBehavior), 动作将发生变化。加速度更改、减速度更改时, 不受理更改。

■减速度 (Deceleration)

设置MC_MoveRelative (相对值定位)中的减速度。

设置了加减速方式设置 (选项 (Options): 位0~2) 的“0: 加减速速度指定方式 (mcAccDec)”的情况下, 对设置范围进行设置。设置了“1: 加减速时间恒定方式 (mcFixedTime)”的情况下, 不使用减速度 (Deceleration)。

加减速方式设置 (选项 (Options): 位0~2)	设置范围
0: 加减速速度指定方式 (mcAccDec)	0.0000* ¹ 、0.0001~2147483647.0[U/s ²]的正数
1: 加减速时间恒定方式 (mcFixedTime)	不使用

*1 启动时, 根据启动时加减速速度0指定时动作选择 (AxisName.Pr.AccelerationZeroBehavior), 动作将发生变化。加速度更改、减速度更改时, 不受理更改。

■Jerk

设置MC_MoveRelative (相对值定位)中的Jerk。

设置了加减速方式设置 (选项 (Options): 位0~2) 的“0: 加减速速度指定方式 (mcAccDec)”的情况下, 对设置范围进行设置。设置了“1: 加减速时间恒定方式 (mcFixedTime)”的情况下, 不使用Jerk。

加减速方式设置 (选项 (Options): 位0~2)	设置范围
0: 加减速速度指定方式 (mcAccDec)	0.0000、0.0001~2147483647.0[U/s ²]的正数
1: 加减速时间恒定方式 (mcFixedTime)	不使用

■缓冲模式 (BufferMode)

设置用于进行多重启动 (缓冲模式) 的动作。


MC_MoveRelative (相对值定位) 中可设置的缓冲模式如下所示。

设定值	内容
0: Aborting (mcAborting)	中断 (取消) 控制中的FB并立即执行下一个FB。
1: Buffered (mcBuffered)	在控制中的FB中缓冲下一个FB。 控制中的FB已经被缓冲的情况下, 对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB完成时, 依次执行缓冲FB。
2: BlendingLow (mcBlendingLow)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下, 对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后, 依次执行缓冲FB。 控制中的FB到达了目标位置时, 将控制中的FB与缓冲FB的目标速度中较低一方的速度作为切换速度。
3: BlendingPrevious (mcBlendingPrevious)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下, 对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后, 依次执行缓冲FB。 控制中的FB到达了目标位置时, 将控制中的FB的目标速度作为切换速度。
4: BlendingNext (mcBlendingNext)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下, 对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后, 依次执行缓冲FB。 控制中的FB到达了目标位置时, 将缓冲FB的目标速度作为切换速度。
5: BlendingHigh (mcBlendingHigh)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下, 对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后, 依次执行缓冲FB。 控制中的FB到达了目标位置时, 将控制中的FB与缓冲FB的目标速度中较高一方的速度作为切换速度。

*1 在控制中的FB与缓冲FB之间不执行停止。

要点

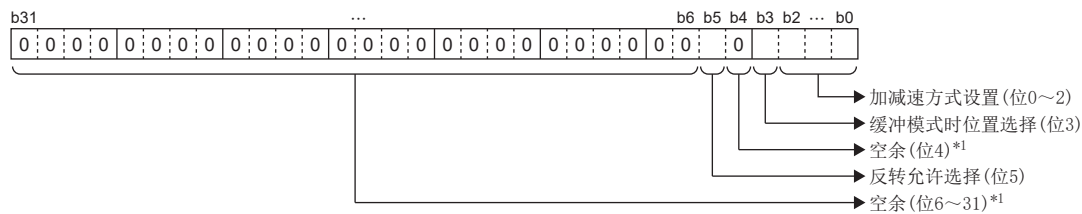
关于多重启动 (缓冲模式) 的详细内容, 请参阅下述手册。

 所使用的控制器的用户手册

■选项 (Options)

将MC_MoveRelative(相对值定位)中使用的功能选项以位指定进行设置。

以位指定设置的内容如下所示。



*1 空余中应设置“0”。设置了“0”以外的情况下，将变为超出Options范围(错误代码: 1ABBH)。

位	名称	内容
0~2	加减速方式设置	设置用于进行控制的加减速方式。 • 0: 加减速速度指定方式 (mcAccDec) • 1: 加减速时间恒定方式 (mcFixedTime)
3	缓冲模式时位置选择	设置对相对值定位控制进行多重启动时的位置。 • 0: 指令当前位置 • 1: 反馈位置 * 缓冲模式(BufferMode)的“0: Aborting(mcAborting)”设置时将有效。
5	反转允许选择	设置执行中的FB与要缓冲的FB中动作方向不相同的情况下是否允许反转。 • 0: 允许 • 1: 不允许 *: 轴动作中切换为后续缓冲FB的情况下, 或目标位置更改时有效。

• 加减速方式设置 (位0~2)

设定值	内容
0: 加减速速度指定方式 (mcAccDec)	是使用MC_MoveRelative(相对值定位)中设置的加速度(Acceleration)、减速度(Deceleration)、Jerk进行加速/减速的方式。
1: 加减速时间恒定方式 (mcFixedTime)	是与速度无关, 使用MC_MoveRelative(相对值定位)中设置的加减速时间进行加速/减速的方式。 对于加减速时间, 设置为加速度(Acceleration), 不使用减速度(Deceleration)、Jerk。

• 缓冲模式时位置选择 (位3)

设定值	内容
0: 指令当前位置	<p>是从指令当前位置开始的相对位置控制。</p> <p><例> 以移动量(Distance)为“5000.0”, 选项(Options)为“00000000H(位3为0: 指令当前位置)”的设置进行了多重启动的情况下</p> <p>按照指令当前位置基准 仅移动移动量(Distance)</p>
1: 反馈位置	<p>是从反馈位置开始的相对位置控制。</p> <p><例> 以移动量(Distance)为“5000.0”, 选项(Options)为“00000008H(位3为1: 反馈位置)”的设置进行了多重启动的情况下</p> <p>按照反馈位置基准 仅移动移动量(Distance)</p>

- 反转允许选择(位5)

设定值	内容
0: 允许	允许反转。 进行减速停止一次，减速停止完成后，向更改的方向开始动作。
1: 不允许	不允许反转。 切换时将变为超载运行错误(错误代码: 1AE3H)且进行减速停止。


■必要对象数据

使用MC_MoveRelative(相对值定位)的情况下，应在轴中设置下述对象数据。

- Target position(607AH)

未设置对象数据的情况下，将变为必须从站对象未设置(错误代码: 1AF7H)且不启动。

关于对象数据设置的详细内容，请参阅下述手册。

所使用的控制器的用户手册

■注意事项

- 即使进行重启/连续更新，最初启动了MC_MoveRelative(相对值定位)的当前位置也将为启动地址。
- 由于以浮点型进行处理，因此重复执行相对值定位控制时由于运算误差可能不能达到设置的移动量。

程序示例

将单轴相对值定位启动(bMoveRelative)置为TRUE，并根据下述设置进行轴1(Axis0001)的相对值定位控制的程序示例如下所示。

- 动作



- 设置

项目	设定值
目标位置	80000.0
速度	50000.0
加速度	100000.0
减速度	100000.0
Jerk	0.0

■轴

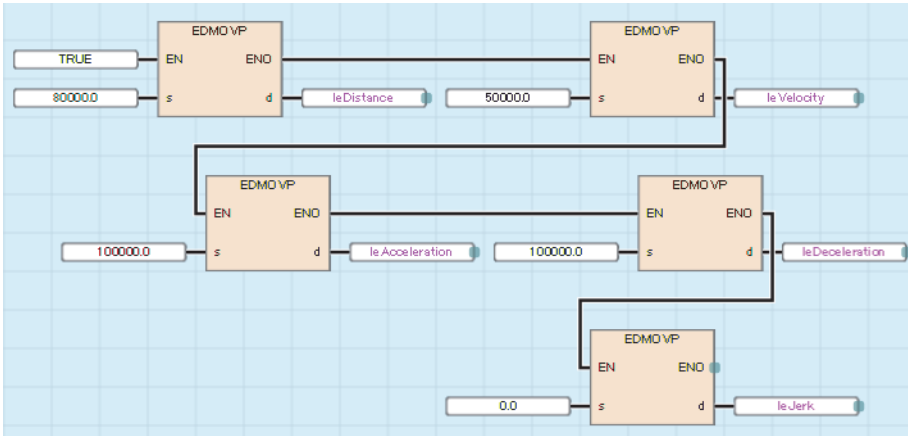
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

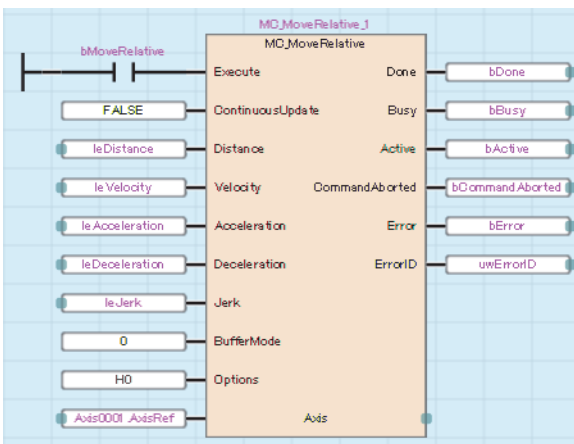
标签名	数据类型	注释
MC_MoveRelative_1	MC_MoveRelative	相对值定位FB
bMoveRelative	位	单轴相对值定位启动
leDistance	双精度实数	移动量
leVelocity	双精度实数	速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bDone	位	执行完成
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

- 单轴定位控制用数据的设置



- 相对值定位



■ST程序

```
//-----单轴定位控制用数据的设置-----  
leDistance:= 80000.0;  
leVelocity:= 50000.0;  
leAcceleration:= 100000.0;  
leDeceleration:= 100000.0;  
leJerk:= 0.0;  
  
//-----相对值定位-----  
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  
);
```

46.6 JOG运行

MCv_Jog

按照指令速度执行JOG运行。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName.AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName.AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
JogForward	正转JOG指令	BOOL	始终	TRUE、FALSE	FALSE	TRUE时执行至正方向的MCv_Jog(JOG运行)。 关于详细内容，请参阅下述章节。 ☞ 1622页 正转JOG指令(JogForward)/反转JOG指令(JogBackward)
JogBackward	反转JOG指令	BOOL	始终	TRUE、FALSE	FALSE	TRUE时执行至反方向的MCv_Jog(JOG运行)。 关于详细内容，请参阅下述章节。 ☞ 1622页 正转JOG指令(JogForward)/反转JOG指令(JogBackward)
Velocity	速度	LREAL	启动时	0.0、0.0001~2500000000.0	0.0	设置指令速度。 关于详细内容，请参阅下述章节。 ☞ 1623页 速度(Velocity)
Acceleration	加速度	LREAL	启动时	0.0000、0.0001~2147483647.0	0.0	设置加速度。 关于详细内容，请参阅下述章节。 ☞ 1623页 加速度(Acceleration)
Deceleration	减速度	LREAL	启动时	0.0000、0.0001~2147483647.0	0.0	设置减速度。 关于详细内容，请参阅下述章节。 ☞ 1624页 减速度(Deceleration)
Jerk	Jerk	LREAL	启动时	0.0000、0.0001~2147483647.0	0.0	设置Jerk。 关于详细内容，请参阅下述章节。 ☞ 1624页 Jerk
Options	选项	DWORD (HEX)	启动时	00000000H~00000001H	00000000H	将MCv_Jog(JOG运行)的功能选项以位指定进行设置。 关于详细内容，请参阅下述章节。 ☞ 1624页 选项(Options)

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	通过JOG指令的OFF，减速停止完成时，仅1个扫描变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_Jog(JOG运行)时，将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_Jog(JOG运行)正在控制轴时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MCv_Jog(JOG运行)的执行中断时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。  所使用的控制器的用户手册

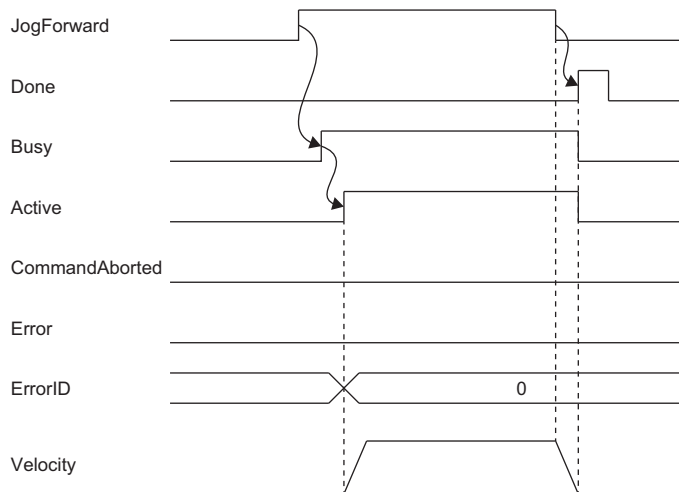
功能

- 将正转JOG指令(JogForward)或反转JOG指令(JogBackward)置为TRUE时，对象轴将向指定的方向移动。
- JOG运行中的轴状态(AxisName.Md.AxisStatus)变为“6: 连续动作运行中(ContinuousMotion)”。
- 将正转JOG指令(JogForward)或反转JOG指令(JogBackward)置为FALSE时，将进行减速停止。
- 减速停止完成时，轴状态(AxisName.Md.AxisStatus)为“4: 待机中(Standstill)”。
- 通过正转JOG指令(JogForward)或反转JOG指令(JogBackward)的FALSE的减速中，错误(Error)变为了TRUE的情况下，在将正转JOG指令(JogForward)或反转JOG指令(JogBackward)置为TRUE之前，错误(Error)将变为TRUE。
- JOG运行动作中启动了其它动作FB的情况下，将按照启动的动作FB的缓冲模式(BufferMode)的设置执行动作。
- 其他动作FB中启动了JOG运行的情况下，将忽略启动请求，且变为运行中启动警告(事件代码: 00D01H)。应使轴状态(AxisName.Md.AxisStatus)在“4: 待机中(Standstill)”时启动。
- 在JOG运行中更改速度的情况下，应使用通过倍率修调功能进行的速度更改。关于倍率修调功能的详细内容，请参阅下述内容。

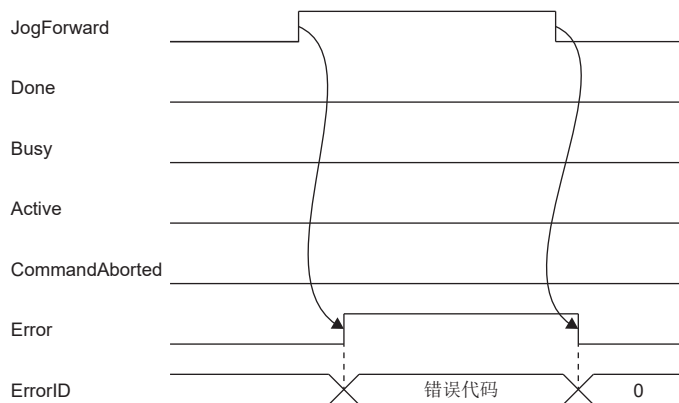
📖 所使用的控制器的用户手册

■ 时序图

- 正常完成的情况下

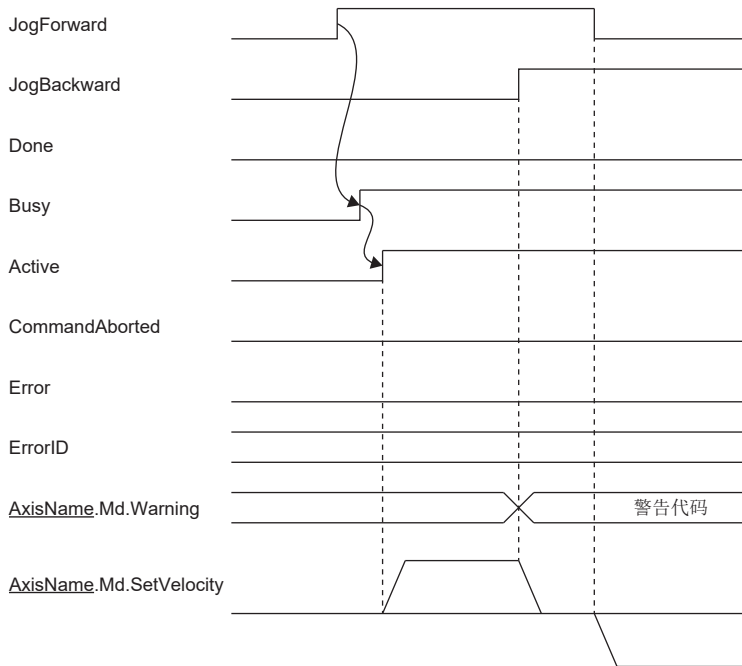


- 异常完成的情况下

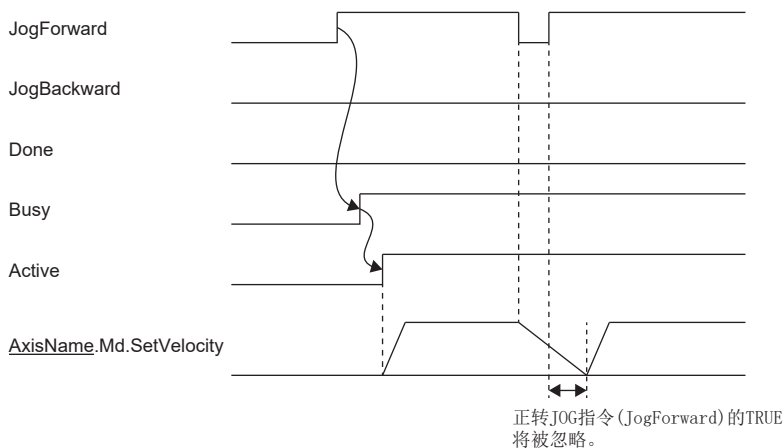


■正转JOG指令(JogForward)/反转JOG指令(JogBackward)

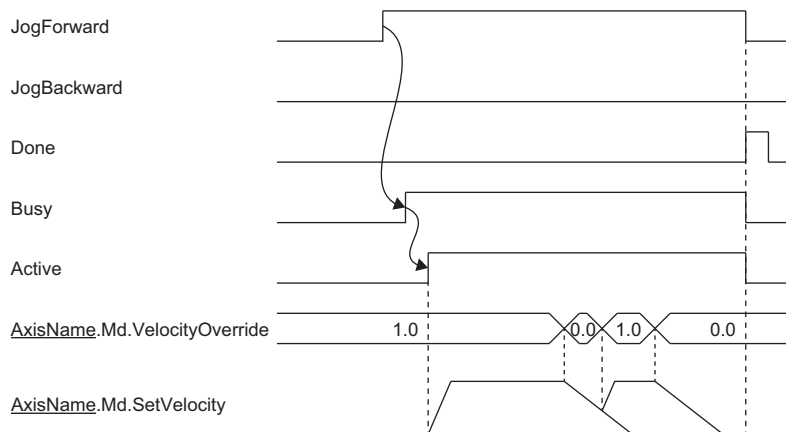
- 将正转JOG指令(JogForward)置为TRUE时将启动正转JOG运行，将反转JOG指令(JogBackward)置为TRUE时将启动反转JOG运行，且对象轴将向指定的方向移动。
- 同时将正转JOG指令(JogForward)与反转JOG指令(JogBackward)置为了TRUE的情况下，将变为两方向JOG指令输入警告(事件代码: 00D02H)，执行中断(CommandAborted)将变为TRUE，且不进行JOG运行。之后，将两方向的JOG指令置为FALSE，将执行中断(CommandAborted)置为了FALSE的状态后，将正转JOG指令(JogForward)或反转JOG指令(JogBackward)的其中任何一个置为TRUE时，将进行JOG运行。
- JOG运行中，MCv_Jog(JOG运行)的另一个(正转JOG指令(JogForward)或反转JOG指令(JogBackward))的JOG指令变为了TRUE的情况下，将变为两方向JOG指令输入警告(事件代码: 00D02H)，且从检测出TRUE的时刻开始进行减速停止。减速停止完成后，仅将某个的JOG指令(正转JOG指令(JogForward)或反转JOG指令(JogBackward))置为TRUE时，将重新开始移动。进行JOG运行的情况下，应仅将正转JOG指令(JogForward)或反转JOG指令(JogBackward)的其中任何一个置为TRUE。



- 通过正转JOG指令(JogForward)或反转JOG指令(JogBackward)的FALSE的减速停止中，即使将JOG指令置为TRUE也不进行再加速。减速停止完成后，进行加速。



- 通过JOG运行进行的移动中，进行重复减速停止及加速的动作的情况下，应使用通过倍率修调功能进行的速度更改。



■速度 (Velocity)

设置MCv_Jog(JOG运行)中的指令速度。

通过正转JOG指令(JogForward)或反转JOG指令(JogBackward)的TRUE的启动时进行获取。不受理JOG运行中的更改。

设置范围

0.0、0.0001~2500000000.0^{*1*2}

- *1 由于进行浮点运算，因此指令速度的下限值中将产生下述限制。
对指令速度进行了运算周期换算的速度小于“0.00001”的情况下，将变为超出运算周期换算速度范围(错误代码：1B16H)(速度更改时为超出运算周期换算速度范围警告(事件代码：00D2FH))。为了提高浮点运算的精度，应通过更改位置指令单位(AxisName.Pr.Unit_Position)或速度指令单位(AxisName.Pr.Unit_Velocity)设置运算周期换算后的速度不低于“0.00001”。
- *2 在进行多重启动的运动控制FB中指定速度为“0.0”的情况下，将变为之前的运动控制FB的指定速度。

■加速度 (Acceleration)

设置MCv_Jog(JOG运行)中的加速度。

通过正转JOG指令(JogForward)或反转JOG指令(JogBackward)的TRUE的启动时进行获取。不受理JOG运行中的更改。

根据加减速方式设置(选项(Options)：位0~2)的设置，设置范围有所不同。

加减速方式设置(选项(Options)：位0~2)	设置范围
0：加减速速度指定方式(mcAccDec)	0.0000 ^{*1} 、0.0001~2147483647.0[U/s ²]的正数
1：加减速时间恒定方式(mcFixedTime)	0.000000 ^{*1} 、0.000001~8400.0[s]的正数

- *1 启动时，根据启动时加减速速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior)，动作将发生变化。加速度更改、减速度更改时，不受理更改。

■减速度 (Deceleration)

设置MCv_Jog (JOG运行) 中的减速度。

通过正转JOG指令 (JogForward) 或反转JOG指令 (JogBackward) 的TRUE的启动时进行获取。不受理JOG运行中的更改。

设置了加减速方式设置 (选项 (Options): 位0~2) 的“0: 加减速速度指定方式 (mcAccDec)”的情况下, 对设置范围进行设置。设置了“1: 加减速时间恒定方式 (mcFixedTime)”的情况下, 不使用减速度 (Deceleration)。

加减速方式设置 (选项 (Options): 位0~2)	设置范围
0: 加减速速度指定方式 (mcAccDec)	0.0000* ¹ 、0.0001~2147483647.0[U/s ²]的正数
1: 加减速时间恒定方式 (mcFixedTime)	不使用

*1 启动时, 根据启动时加减速速度0指定时动作选择 (AxisName.Pr.AccelerationZeroBehavior), 动作将发生变化。加速度更改、减速度更改时, 不受理更改。

■Jerk

设置MCv_Jog (JOG运行) 中的Jerk。

通过正转JOG指令 (JogForward) 或反转JOG指令 (JogBackward) 的TRUE的启动时进行获取。不受理JOG运行中的更改。

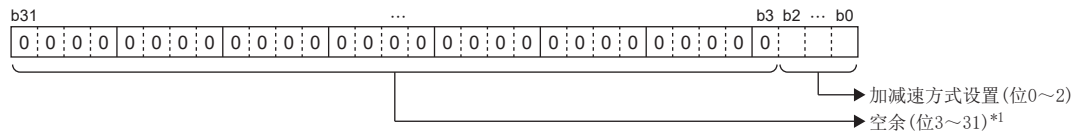
设置了加减速方式设置 (选项 (Options): 位0~2) 的“0: 加减速速度指定方式 (mcAccDec)”的情况下, 对设置范围进行设置。设置了“1: 加减速时间恒定方式 (mcFixedTime)”的情况下, 不使用Jerk。

加减速方式设置 (选项 (Options): 位0~2)	设置范围
0: 加减速速度指定方式 (mcAccDec)	0.0000、0.0001~2147483647.0[U/s ³]的正数
1: 加减速时间恒定方式 (mcFixedTime)	不使用

■选项 (Options)

将MCv_Jog (JOG运行) 中使用的功能选项以位指定进行设置。

以位指定设置的内容如下所示。



*1 空余中应设置“0”。设置了“0”以外的情况下, 将变为超出Options范围 (错误代码: 1ABBH)。

位	名称	内容
0~2	加减速方式设置	设置用于进行控制的加减速方式。 • 0: 加减速速度指定方式 (mcAccDec) • 1: 加减速时间恒定方式 (mcFixedTime)

• 加减速方式设置 (位0~2)

设定值	内容
0: 加减速速度指定方式 (mcAccDec)	是使用MCv_Jog (JOG运行) 中设置的加速度 (Acceleration)、减速度 (Deceleration)、Jerk进行加速/减速的方式。
1: 加减速时间恒定方式 (mcFixedTime)	是与速度无关, 使用MCv_Jog (JOG运行) 中设置的加减速时间进行加速/减速的方式。 对于加减速时间, 设置为加速度 (Acceleration), 不使用减速度 (Deceleration)、Jerk。

■必要对象数据

使用MCv_Jog (JOG运行) 的情况下, 应在轴中设置下述对象数据。

- Target position (607AH)

未设置对象数据的情况下, 将变为必须从站对象未设置 (错误代码: 1AF7H) 且不启动。

关于对象数据设置的详细内容, 请参阅下述手册。

所使用的控制器的用户手册

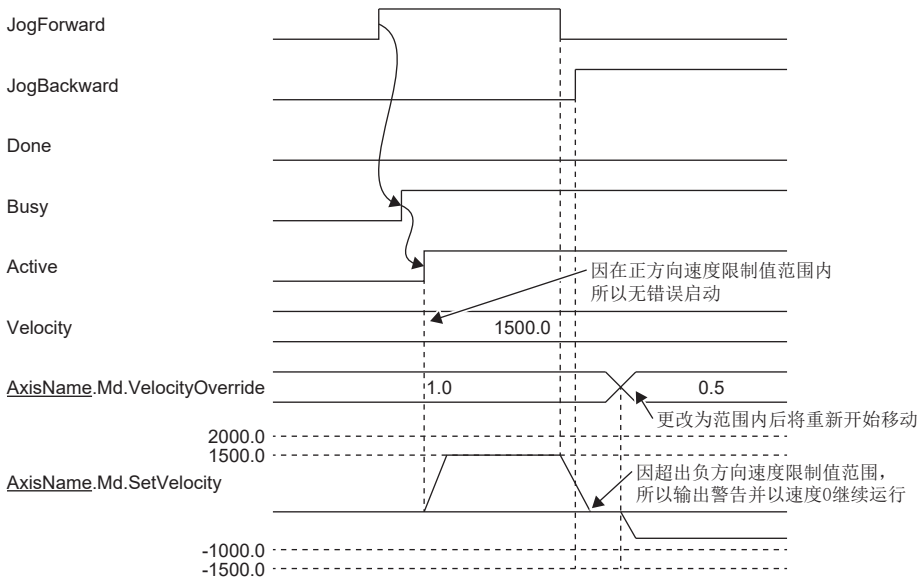
注意事项

- 为了安全起见，对于速度(Velocity)中设置的值，最初应先设置较小的值并确认动作，然后逐渐增大值。
- 在上限位、下限限位的附近进行JOG运行的情况下，应使用硬件行程限位功能。不使用硬件行程限位功能时，工件有可能超出移动范围，导致发生事故。
- 软件行程限位功能有效，软件行程限位倍率修调(AxisName.Cd.SwStrokeLimit_Override)为DISABLE(检查无效)及ONLY_INSIDE(仅向范围内方向检查无效)以外时，执行MCv_Jog(JOG运行)时，将软件行程限位倍率修调(AxisName.Cd.SwStrokeLimit_Override)改写为ONLY_INSIDE(仅向范围内方向检查无效)，JOG运行完成时将其改写为空白。
- 硬件行程限位功能有效，硬件行程限位倍率修调(AxisName.Cd.HwStrokeLimit_Override)为DISABLE(检查无效)及ONLY_INSIDE(仅至范围内方向检查无效)以外时，执行MCv_Jog(JOG运行)时，将硬件行程限位倍率修调(AxisName.Cd.HwStrokeLimit_Override)更改为ONLY_INSIDE(范围内方向检查无效)，在JOG运行完成时将其改写为空白。
- MCv_Jog(JOG运行)执行中，请勿更改软件行程限位倍率修调(AxisName.Cd.SwStrokeLimit_Override)及硬件行程限位倍率修调(AxisName.Cd.HwStrokeLimit_Override)。
- JOG运行中多重启动的情况下，由于在运行中进行下一个FB的分析，因此多重启动时的软件行程限位倍率修调(AxisName.Cd.SwStrokeLimit_Override)及硬件行程限位倍率修调(AxisName.Cd.HwStrokeLimit_Override)的值将被沿用到下一个FB中，请注意。
- JOG运行中进行减速停止并向与启动时相反方向进行移动时，指令速度超出速度限制值的情况下，将变为方向反转时的速度限制值溢出警告(事件代码：00D20H)。此外，加速时间超出8400秒的情况下，将变为方向反转时的加速时间溢出警告(事件代码：00D32H)。(以速度0继续运行。)通过进行控制更改以消除原因，开始移动。

例

设置了下述值的情况下

- 正方向速度限制值(AxisName.Pr.VelocityLimit_Positive)：2000.0
- 负方向速度限制值(AxisName.Pr.VelocityLimit_Negative)：1000.0



程序示例

进行正转JOG运行的情况下将JOG正转指令(bJogForward)，进行反转JOG运行的情况下将JOG反转指令(bJogBackward)置为TRUE，并根据下述设置进行轴1(Axis0001)的JOG运行的程序示例如下所示。

- 设置

项目	设定值
速度	1500.0
加速度	2000.0
减速度	2000.0
Jerk	0.0

■轴

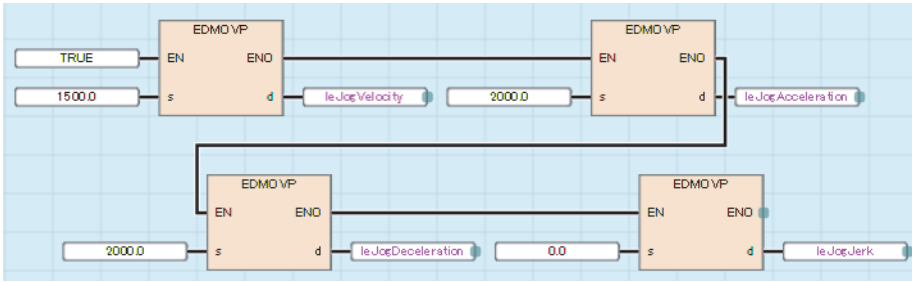
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

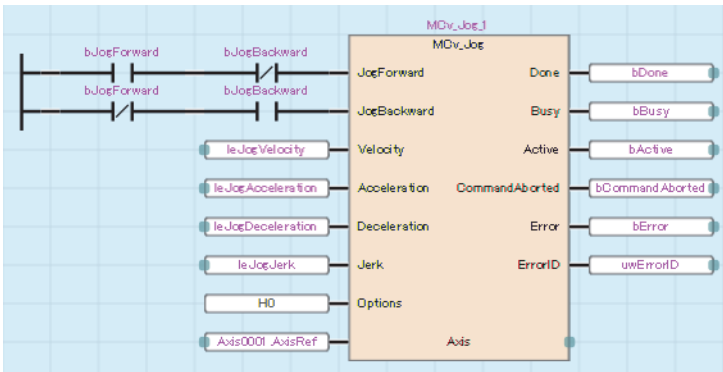
标签名	数据类型	注释
MCv_Jog_1	MCv_Jog	JOG运行FB
bJogForward	位	JOG正转指令
bJogBackward	位	JOG反转指令
leJogVelocity	双精度实数	JOG速度
leJogAcceleration	双精度实数	JOG加速度
leJogDeceleration	双精度实数	JOG减速度
leJogJerk	双精度实数	JOG Jerk
bJogF	位	JOG正转
bJogB	位	JOG反转
bDone	位	执行完成
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

- JOG运行用数据的设置



- Axis0001 JOG运行



■ST程序

```
//-----JOG运行用数据的设置-----
leJogVelocity:= 1500.0;
leJogAcceleration:= 2000.0;
leJogDeceleration:= 2000.0;
leJogJerk:= 0.0;

//-----Axis0001 JOG运行-----
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
);
```

46.7 速度控制

MC_MoveVelocity

将驱动器模块切换为CSV，按照指定的速度进行速度控制。

梯形图	FBD/LD	ST
		<pre> 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?); </pre>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName, AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_MoveVelocity(速度控制)。
ContinuousUpdate	连续更新	BOOL	启动时	TRUE、FALSE	FALSE	设置将速度(Velocity)、加速度(Acceleration)、减速度(Deceleration)的连续更改是置为有效，还是置为无效。 • FALSE: 无效 • TRUE: 有效
Velocity	速度	LREAL	启动时/可重启/可连续更新	0.0、±0.0001~±2500000000.0	0.0	设置指令速度。 速度为负的情况下，向负方向移动。设置了“0.0”时，轴不动作但轴状态(AxisName, Md, AxisStatus)变为“6: 连续动作运行中(ContinuousMotion)”。 关于详细内容，请参阅下述章节。 ☞ 1631页 速度(Velocity)
Acceleration	加速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置加速度。 关于详细内容，请参阅下述章节。 ☞ 1632页 加速度(Acceleration)
Deceleration	减速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置减速度。 关于详细内容，请参阅下述章节。 ☞ 1632页 减速度(Deceleration)
Jerk	Jerk	LREAL	启动时	0.0000、0.0001~2147483647.0	0.0	设置Jerk。 关于详细内容，请参阅下述章节。 ☞ 1632页 Jerk

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Direction	方向选择	INT (MC_DIRECTION)	启动时	1、2	0	设置方向选择。 <ul style="list-style-type: none"> • 1: 正方向(mcPositiveDirection) • 2: 负方向(mcNegativeDirection) * 设置了“2: 负方向(mcNegativeDirection)”, 且速度(Velocity)为负的情况下, 电机的移动方向将变为正方向。 * 省略了设置的情况下, 将变为超出方向选择范围(错误代码: 1AA5H)。
BufferMode	缓冲模式	INT (MC_BUFFER_MODE)	启动时	0、1	0	设置缓冲模式。 <ul style="list-style-type: none"> • 0: Aborting(mcAborting) • 1: Buffered(mcBuffered) 关于详细内容, 请参阅下述章节。 1632页 缓冲模式(BufferMode)
Options	选项	DWORD (HEX)	启动时	00000000H~0002001H	00000000H	将MC_MoveVelocity(速度控制)的功能选项以位指定进行设置。 关于详细内容, 请参阅下述章节。 1633页 选项(Options)

■输出变量

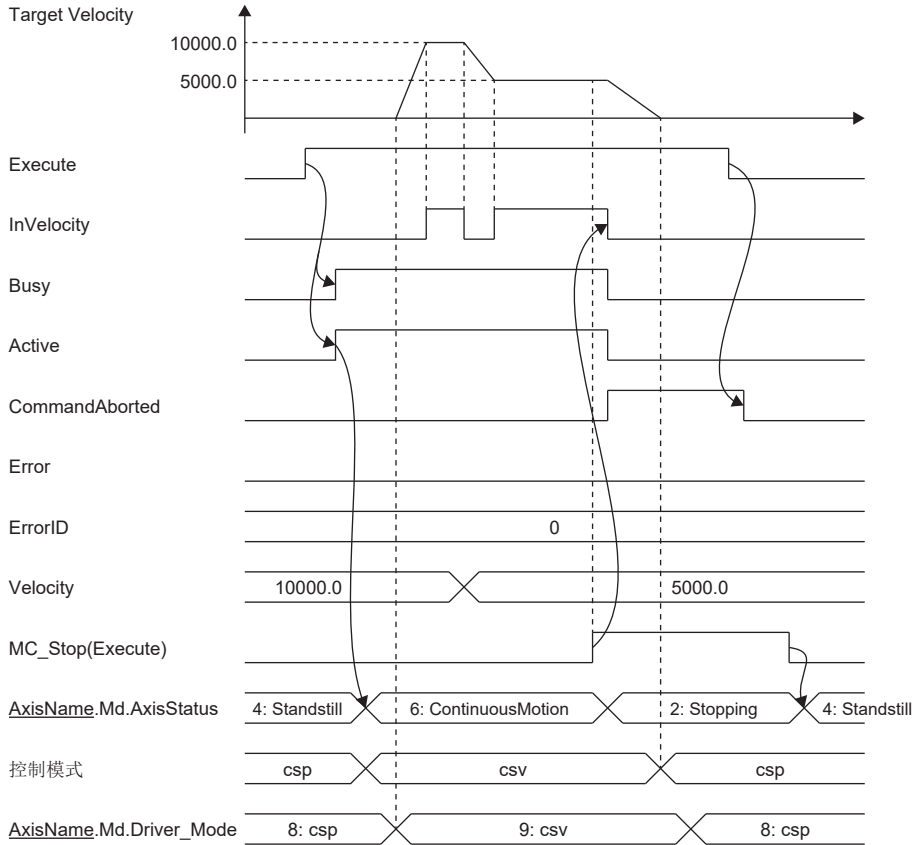
输出变量	名称	数据类型	初始值	内容
InVelocity	目标速度到达	BOOL	FALSE	运动系统中计算的指令速度到达了目标速度时, 将变为TRUE。 通过连接更新(ContinuousUpdate)为有效(TRUE)时的更改更改了目标速度的情况下, 在到达更改后的目标速度之前将变为FALSE。
Busy	执行中	BOOL	FALSE	执行了MC_MoveVelocity(速度控制)时, 将变为TRUE。
Active	控制中	BOOL	FALSE	MC_MoveVelocity(速度控制)正在控制轴时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MC_MoveVelocity(速度控制)的执行中断时, 将变为TRUE。 执行指令(Execute)变为FALSE时, 将变为FALSE。
Error	错误	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时, 将返回错误代码。 关于错误代码的详细内容, 请参阅下述手册。 1633页 所使用的控制器的用户手册

功能

对于MC_MoveVelocity(速度控制)，将驱动器的控制模式切换为csv并进行控制。根据设置的加速度(Acceleration)、减速度(Deceleration)、Jerk，对指令速度进行控制。结束MC_MoveVelocity(速度控制)时，应启动MC_Stop(强制停止)。

■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■速度(Velocity)

设置MC_MoveVelocity(速度控制)中的指令速度。

设置范围

0.0、±0.0001~±2500000000.0*1*2

- *1 由于进行浮点运算，因此指令速度的下限值中将产生下述限制。
对指令速度进行了运算周期换算的速度小于“0.00001”的情况下，将变为超出运算周期换算速度范围(错误代码：1B16H)(速度更改时为超出运算周期换算速度范围警告(事件代码：00D2FH))。为了提高浮点运算的精度，应通过更改位置指令单位(AxisName.Pr.Unit_Position)或速度指令单位(AxisName.Pr.Unit_Velocity)设置运算周期换算后的速度不低于“0.00001”。
- *2 在进行多重启动的运动控制FB中指定速度为“0.0”的情况下，将变为之前的运动控制FB的指定速度。

■加速度 (Acceleration)

设置MC_MoveVelocity(速度控制)中的加速度。

根据加减速方式设置(选项(Options): 位0~2)的设置, 设置范围有所不同。

加减速方式设置(选项(Options): 位0~2)	设置范围
0: 加减速速度指定方式(mcAccDec)	0.0000* ¹ 、0.0001~2147483647.0[U/s ²]的正数
1: 加减速时间恒定方式(mcFixedTime)	0.000000* ¹ 、0.000001~8400.0[s]的正数

*1 启动时, 根据启动时加减速速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior), 动作将发生变化。加速度更改、减速度更改时, 不受理更改。

■减速度 (Deceleration)

设置MC_MoveVelocity(速度控制)中的减速度。

设置了加减速方式设置(选项(Options): 位0~2)的“0: 加减速速度指定方式(mcAccDec)”的情况下, 对设置范围进行设置。设置了“1: 加减速时间恒定方式(mcFixedTime)”的情况下, 不使用减速度(Deceleration)。

加减速方式设置(选项(Options): 位0~2)	设置范围
0: 加减速速度指定方式(mcAccDec)	0.0000* ¹ 、0.0001~2147483647.0[U/s ²]的正数
1: 加减速时间恒定方式(mcFixedTime)	不使用

*1 启动时, 根据启动时加减速速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior), 动作将发生变化。加速度更改、减速度更改时, 不受理更改。

■Jerk

设置MC_MoveVelocity(速度控制)中的Jerk。

设置了加减速方式设置(选项(Options): 位0~2)的“0: 加减速速度指定方式(mcAccDec)”的情况下, 对设置范围进行设置。设置了“1: 加减速时间恒定方式(mcFixedTime)”的情况下, 不使用Jerk。

加减速方式设置(选项(Options): 位0~2)	设置范围
0: 加减速速度指定方式(mcAccDec)	0.0000、0.0001~2147483647.0[U/s ²]的正数
1: 加减速时间恒定方式(mcFixedTime)	不使用

■缓冲模式 (BufferMode)


设置用于进行多重启动(缓冲模式)的动作。

MC_MoveVelocity(速度控制)中可设置的缓冲模式如下所示。

设定值	内容
0: Aborting(mcAborting)	中断(取消)控制中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在控制中的FB中缓冲下一个FB。 控制中的FB已经被缓冲的情况下, 对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB完成时, 依次执行缓冲FB。

要点

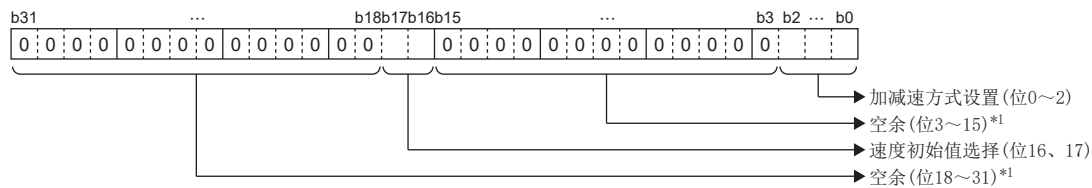
关于多重启动(缓冲模式)的详细内容, 请参阅下述手册。

所使用的控制器的用户手册

■选项 (Options)

将MC_MoveVelocity(速度控制)中使用的功能选项以位指定进行设置。

以位指定设置的内容如下所示。



*1 空余中应设置“0”。设置了“0”以外的情况下，将变为超出Options范围(错误代码: 1ABBH)。

位	名称	内容
0~2	加减速方式设置	设置用于进行控制的加减速方式。 <ul style="list-style-type: none"> • 0: 加减速速度指定方式 (mcAccDec) • 1: 加减速时间恒定方式 (mcFixedTime)
16、17	速度初始值选择	设置从循环位置模式 (csp) 至循环速度模式 (csv) 的控制模式的切换时的速度初始值。 <ul style="list-style-type: none"> • 0: 指令速度 • 1: 反馈速度 • 2: 自动选择

• 加减速方式设置 (位0~2)

设定值	内容
0: 加减速速度指定方式 (mcAccDec)	是使用MC_MoveVelocity(速度控制)中设置的加速度(Acceleration)、减速度(Deceleration)、Jerk进行加速/减速的方式。
1: 加减速时间恒定方式 (mcFixedTime)	是与速度无关, 使用MC_MoveVelocity(速度控制)中设置的加减速时间进行加速/减速的方式。 对于加减速时间, 设置为加速度(Acceleration), 不使用减速度(Deceleration)、Jerk。

• 速度初始值选择 (位16、17)

设定值	内容
0: 指令速度	切换之后至驱动器模块的指令速度将变为指令中的速度。
1: 反馈速度	将变为切换时从驱动器模块接收的电机旋转数。 ^{*1}
2: 自动选择	切换之后至驱动器模块的指令速度将变为“0: 指令速度”与“1: 反馈速度”中较低一方的速度。 ^{*2}

*1 未映射对象数据的“Velocity actual value (606CH)”的情况下, 不变为切换时从驱动器模块接收的电机旋转数。(速度初始值将变为“0”。)

*2 未映射对象数据的“Velocity actual value (606CH)”的情况下, 切换之后至驱动器模块的指令速度将变为“0: 指令速度”。

■关联对象数据

未设置对象数据、未进行PDO映射或设备站中无对象时, 错误及规格限制的功能如下所示。

关于对象数据设置的详细内容, 请参阅下述手册。

☞ 所使用的控制器的用户手册

◎: 必须, ○: 推荐

指令・功能	所需的对象数据	PDO映射	未设置对象时的动作
MC_MoveVelocity(速度控制)	Target velocity	◎	必要对象数据未设置(错误代码: 1AF7H)。
	Velocity actual value	○	将为速度初始值选择(选项(Options)位: 16、17)中设置的动作。 ☞ 1633页 选项(Options)

注意事项

- 速度倍率修调系数 (AxisName. Cd.VelocityOverride)、加速度倍率修调系数 (AxisName. Cd.AccelerationOverride) 为有效。
 - 通过跟踪更新指令当前位置、进给机械位置。
 - 控制模式切换为止的时间取决于驱动器模块的规格。
 - 控制模式切换中发生了停止原因的情况下，将立即停止。
 - 请勿在控制模式切换中启动定位控制用的FB。驱动器控制模式 (AxisName. Md.Driver_Mode) 为“9：速度控制(csv)”后，应启动定位控制用的FB。
 - 将控制模式从循环位置模式(csp)切换为循环速度模式(csv)的情况下，如果在控制模式切换之前发生停止原因(检测出轴错误)，则有可能保持循环位置模式(csp)，进行跟踪直至变为零速度状态。
 - 使用MR-J5(W)-G，在不等待电机的停止的状况下从csp切换为csv的情况下，或从csv切换为csp的情况下，应注意以下几点。
 - 应将伺服参数(扩展设置)的“控制切换时ZSP无效选择(PC76.1)”设置为“1：无效(与ZSP范围无关进行控制切换)”，并将零速度状态的监视设置为无效。但是，切换控制模式时可能会发生振动或冲击。
 - 关于伺服参数(基本设置)的“电子齿轮分子(PA06)”、“电子齿轮分母(PA07)”的设定值，请参阅下述手册。
- 所使用的控制器的用户手册

功能

将速度控制指令(bMoveVelocity)置为TRUE，切换至速度控制，并根据下述设置进行轴1(Axis0001)的速度控制的程序示例如下所示。

- 设置

项目	设定值
速度	100000.0
加速度	50000.0
减速度	50000.0
Jerk	0.0

■轴

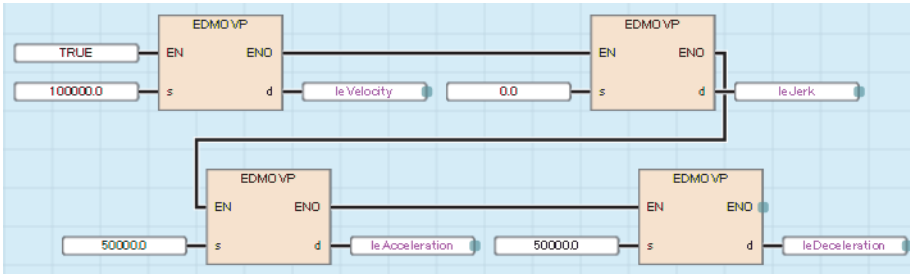
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

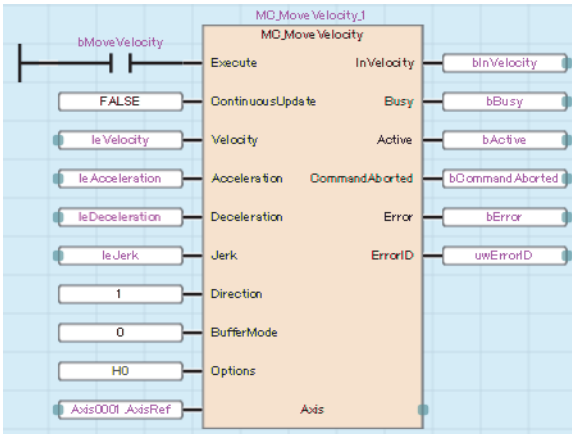
标签名	数据类型	注释
MC_MoveVelocity_1	MC_MoveVelocity	速度控制FB
bMoveVelocity	位	速度控制指令
leVelocity	双精度实数	指令速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bInVelocity	位	目标速度到达
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

- 速度控制用数据的设置



- 速度控制



■ST程序

```
//-----速度控制用数据的设置-----
leVelocity:= 100000.0;
leAcceleration:= 50000.0;
leDeceleration:= 50000.0;
leJerk:= 0.0;

//-----速度控制-----
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
);
```

46.8 转矩控制

MC_TorqueControl

将驱动器模块切换为cst，按照指定的目标转矩进行转矩控制。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName.AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName.AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_TorqueControl (转矩控制)。
ContinuousUpdate	连续更新	BOOL	启动时	TRUE、FALSE	FALSE	设置将目标转矩(Torque)、转矩正方向斜率(TorquePositiveRamp)、转矩负方向斜率(TorqueNegativeRamp)、速度限制(LimitVelocity)、加速度(Acceleration)、减速度(Deceleration)的连续更改是置为有效，还是置为无效。 • FALSE: 无效 • TRUE: 有效
Torque	目标转矩	LREAL	启动时/可重启/可连续更新	-1000.0~1000.0[%]	0.0	设置指令转矩。 对使用的伺服电机的额定转矩的比率以%单位进行设置。 小数点以下的有效位数，根据对象数据图中分配的ID而有所不同。 有效位数以下的值将被舍去。 将指令转矩向正方向(地址增加方向)输出的情况下应设置正的值，向负方向(地址减少方向)输出的情况下应设置负的值。
TorquePositiveRamp	转矩正方向斜率	LREAL	启动时/可重启/可连续更新	■“0: 斜率方式”的情况下 0.0~1000.0[%/s] ■“1: 时间常数方式”、“2: 时间恒定方式”的情况下 0.0~8400.0[s]	0.0	设置正方向的指令转矩。 关于详细内容，请参阅下述章节。 ☞ 1640页 转矩正方向斜率(TorquePositiveRamp)/转矩负方向斜率(TorqueNegativeRamp)
TorqueNegativeRamp	转矩负方向斜率	LREAL	启动时/可重启/可连续更新		0.0	设置负方向的指令转矩。 ☞ 1640页 转矩正方向斜率(TorquePositiveRamp)/转矩负方向斜率(TorqueNegativeRamp)

输入变量	名称	数据类型	获取	有效范围	初始值	内容
LimitVelocity	限制速度	LREAL	启动时/可重启/可连续更新	0.0001~250000000.0	0.0	设置循环转矩模式时的速度限制值。 *：关于速度限制值的值，请参阅驱动器模块的手册。 ■使用MR-J5(W)-G的情况下 •速度限制值使用对象数据的“Velocity limit value(2D20H)”。 •未映射对象数据的“Velocity limit value(2D20H)”的情况下，控制器的速度指令不变为有效。(伺服参数的“速度限制(PT67)”的设定值将变为有效。) * 关于对象数据的详细内容，请参阅下述手册。 □□所使用的控制器的用户手册
Acceleration	加速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置达到限制速度之前的加速度/加减速时间。 关于详细内容，请参阅下述章节。 ☞ 1641页 加速度(Acceleration)
Deceleration	减速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置达到限制速度之前的减速度。 关于详细内容，请参阅下述章节。 ☞ 1641页 减速度(Deceleration)
Jerk	Jerk	LREAL	启动时	0.0000、0.0001~2147483647.0	0.0	设置达到限制速度之前的Jerk。 关于详细内容，请参阅下述章节。 ☞ 1641页 Jerk * “0.0”或省略了设置的情况下，不适用Jerk。
Direction	方向选择	INT (MC_DIRECT ION)	启动时	0	0	应设置“0”。 * 设置了“0”以外的情况下，将变为超出方向选择范围(错误代码：1AA5H)。
BufferMode	缓冲模式	INT (MC_BUFFER _MODE)	启动时	0、1	0	设置缓冲模式。 •0: Aborting(mcAborting) •1: Buffered(mcBuffered) 关于详细内容，请参阅下述章节。 ☞ 1641页 缓冲模式(BufferMode)
Options	选项	DWORD(HEX)	启动时	00000000H~00060001H	00000000H	将MC_TorqueControl(转矩控制)的功能选项以位指定进行设置。 关于详细内容，请参阅下述章节。 ☞ 1642页 选项(Options)

■输出变量

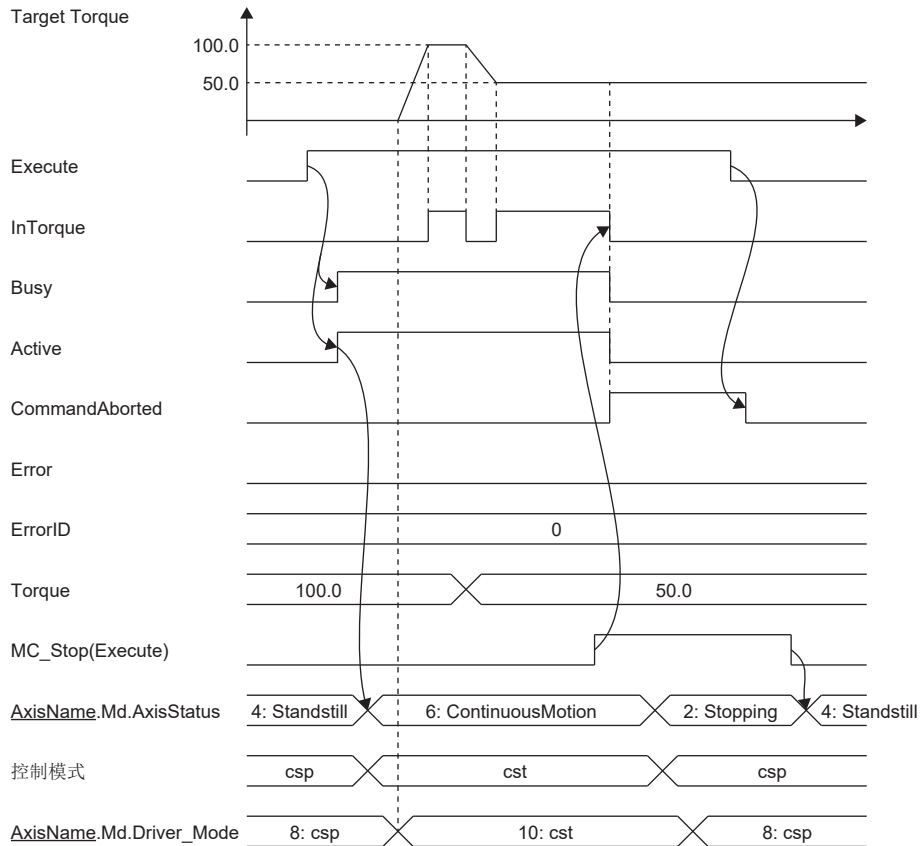
输出变量	名称	数据类型	初始值	内容
InTorque	目标转矩到达	BOOL	FALSE	指令转矩到达了目标转矩时，将变为TRUE。 通过连续更新(ContinuousUpdate)为TRUE时的更改更改了目标转矩的情况下，在到达更改后的目标转矩之前将变为FALSE。
Busy	执行中	BOOL	FALSE	执行了MC_TorqueControl(转矩控制)时，将变为TRUE。
Active	控制中	BOOL	FALSE	MC_TorqueControl(转矩控制)正在控制轴时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MC_TorqueControl(转矩控制)的执行中断时，将变为TRUE。 执行指令(Execute)变为FALSE时，将变为FALSE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 □□所使用的控制器的用户手册

功能

对于MC_TorqueControl (转矩控制)，将驱动器模块的控制模式切换为cst (循环转矩模式)，并进行控制。根据设置的转矩正方向斜率(TorquePositiveRamp)、转矩负方向斜率(TorqueNegativeRamp)对指令转矩进行控制。结束MC_TorqueControl (转矩控制)时，应启动MC_Stop (强制停止)。

■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

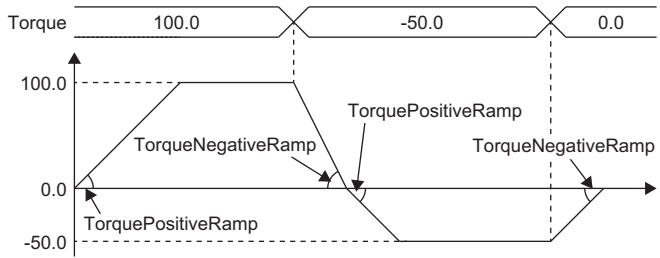
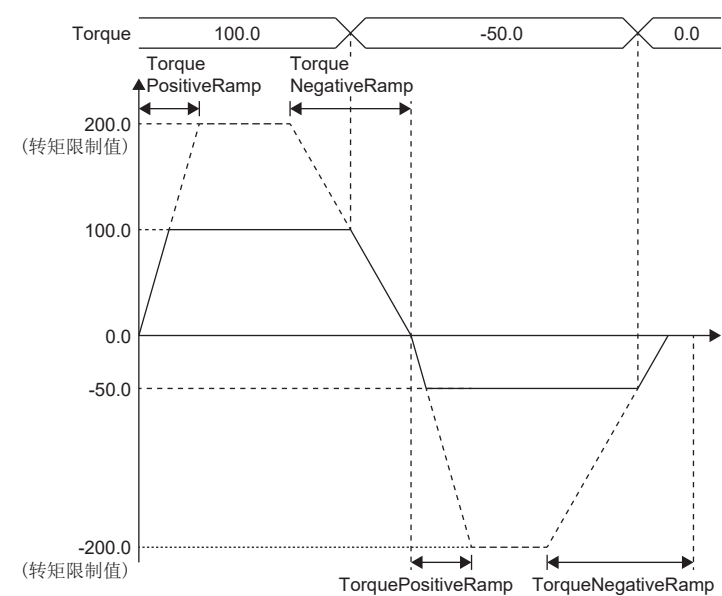
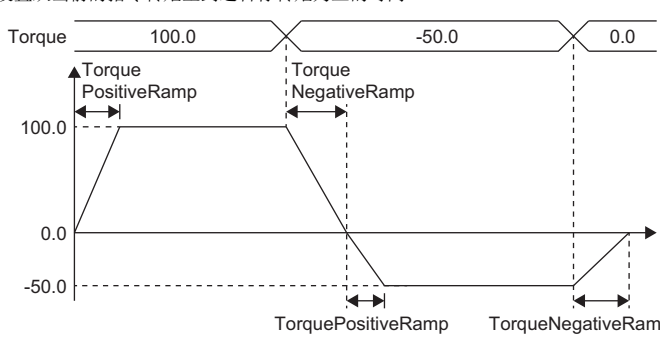
☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■转矩正方向斜率(TorquePositiveRamp)/转矩负方向斜率(TorqueNegativeRamp)

设置要指定的指令转矩。

设置从当前的指令转矩至到达目标转矩为止的设定值。

到达目标转矩为止的设定值根据转矩斜率功能选择(选项(Option): 位16、17)的设置而有所不同。

转矩斜率功能选择(选项(Option): 位16、17)的设定值	内容	设置范围
0: 斜率方式	<p>设置从当前的指令转矩至到达目标转矩为止的斜率。</p>  <p>The graph shows a torque profile with three target values: 100.0, -50.0, and 0.0. The torque starts at 0.0, ramps up to 100.0 (labeled TorquePositiveRamp), stays constant, then ramps down to -50.0 (labeled TorqueNegativeRamp), stays constant, then ramps up to 0.0 (labeled TorquePositiveRamp). The ramping slopes are indicated by dashed lines and arrows.</p>	0.0~1000.0[%/s]
1: 时间常数方式	<p>设置指令转矩从“0”至到达正方向转矩限制值、负方向转矩限制值为止的时间。</p>  <p>The graph shows a torque profile with target values 100.0, -50.0, and 0.0. It also shows torque limits at 200.0 and -200.0. The torque starts at 0.0, ramps up to 200.0 (labeled TorquePositiveRamp), stays constant, then ramps down to -200.0 (labeled TorqueNegativeRamp), stays constant, then ramps up to 0.0 (labeled TorquePositiveRamp). The ramping slopes are indicated by dashed lines and arrows.</p>	0.0~8400.0[s]
2: 时间恒定方式	<p>设置从当前的指令转矩至到达目标转矩为止的时间。</p>  <p>The graph shows a torque profile with target values 100.0, -50.0, and 0.0. The torque starts at 0.0, ramps up to 100.0 (labeled TorquePositiveRamp), stays constant, then ramps down to -50.0 (labeled TorqueNegativeRamp), stays constant, then ramps up to 0.0 (labeled TorquePositiveRamp). The ramping slopes are indicated by dashed lines and arrows.</p>	0.0~8400.0[s]

由于目标转矩的更改输出转矩的方向发生变化的情况下，根据转矩负方向斜率(TorqueNegativeRamp)的设定值指令转矩将变为“0.0”，之后，根据转矩正方向斜率(TorquePositiveRamp)的设定值将变为目标转矩。

在转矩正方向斜率(TorquePositiveRamp)、转矩负方向斜率(TorqueNegativeRamp)中设置了“0.0”的情况下，将在1运算周期内到达目标转矩。

■加速度 (Acceleration)

设置MC_TorqueControl (转矩控制) 中的加速度。

根据加减速方式设置 (选项 (Options): 位0~2) 的设置, 设置范围有所不同。

加减速方式设置 (选项 (Options): 位0~2)	设置范围
0: 加减速速度指定方式 (mcAccDec)	0.0000*1、0.0001~2147483647.0[U/s ²]的正数
1: 加减速时间恒定方式 (mcFixedTime)	0.000000*1、0.000001~8400.0[s]的正数

*1 启动时, 根据启动时加减速速度0指定时动作选择 (AxisName.Pr.AccelerationZeroBehavior), 动作将发生变化。加速度更改、减速度更改时, 不受理更改。

■减速度 (Deceleration)

设置MC_TorqueControl (转矩控制) 中的减速度。

设置了加减速方式设置 (选项 (Options): 位0~2) 的“0: 加减速速度指定方式 (mcAccDec)”的情况下, 对设置范围进行设置。设置了“1: 加减速时间恒定方式 (mcFixedTime)”的情况下, 不使用减速度 (Deceleration)。

加减速方式设置 (选项 (Options): 位0~2)	设置范围
0: 加减速速度指定方式 (mcAccDec)	0.0000*1、0.0001~2147483647.0[U/s ²]的正数
1: 加减速时间恒定方式 (mcFixedTime)	不使用

*1 启动时, 根据启动时加减速速度0指定时动作选择 (AxisName.Pr.AccelerationZeroBehavior), 动作将发生变化。加速度更改、减速度更改时, 不受理更改。

■Jerk

设置MC_TorqueControl (转矩控制) 中的Jerk。

设置了加减速方式设置 (选项 (Options): 位0~2) 的“0: 加减速速度指定方式 (mcAccDec)”的情况下, 对设置范围进行设置。设置了“1: 加减速时间恒定方式 (mcFixedTime)”的情况下, 不使用Jerk。

加减速方式设置 (选项 (Options): 位0~2)	设置范围
0: 加减速速度指定方式 (mcAccDec)	0.0000、0.0001~2147483647.0[U/s ²]的正数
1: 加减速时间恒定方式 (mcFixedTime)	不使用

■缓冲模式 (BufferMode)


设置用于进行多重启动 (缓冲模式) 的动作。

MC_TorqueControl (转矩控制) 中可设置的缓冲模式如下所示。

设定值	内容
0: Aborting (mcAborting)	中断 (取消) 控制中的FB并立即执行下一个FB。 立即切换为转矩控制。
1: Buffered (mcBuffered)	在控制中的FB中缓冲下一个FB。 控制中的FB已经被缓冲的情况下, 对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB完成时, 依次执行缓冲FB。 前一个FB完成后, 切换为转矩控制。 前一个FB为MC_TorqueControl (转矩控制) 的情况下, 目标转矩到达 (InTorque) 为TRUE时将切换。

要点

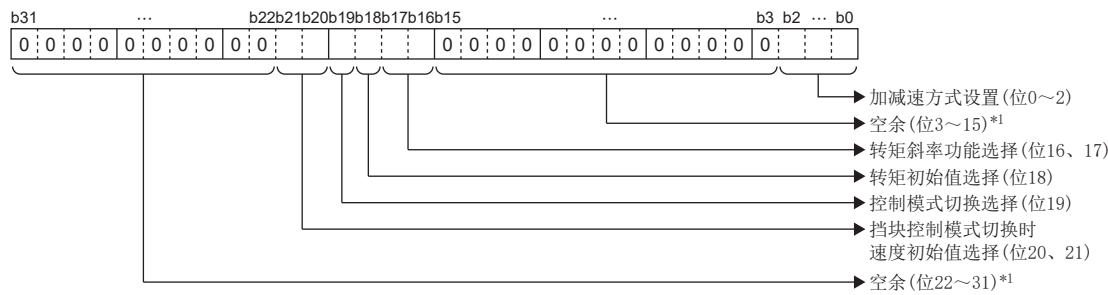
关于多重启动 (缓冲模式) 的详细内容, 请参阅下述手册。

 所使用的控制器的用户手册

■选项(Options)

将MC_TorqueControl(转矩控制)中使用的功能选项以位指定进行设置。

以位指定设置的内容如下所示。



*1 空余中应设置“0”。设置了“0”以外的情况下，将变为超出Options范围(错误代码：1ABBH)。

位	名称	内容
0~2	加减速方式设置	设置用于进行控制的加减速方式。 • 0: 加减速速度指定方式(mcAccDec) • 1: 加减速时间恒定方式(mcFixedTime)
16、17	转矩斜率功能选择	设置从指令转矩至到达目标转矩为止的方式。 • 0: 斜率方式 • 1: 时间常数方式 • 2: 时间恒定方式 * 关于详细内容请参阅下述章节。 ☞ 1640页 转矩正方向斜率(TorquePositiveRamp)/转矩负方向斜率(TorqueNegativeRamp)
18	转矩初始值选择	设置切换至循环转矩模式(cst)时的转矩初始值。 • 0: 目标转矩 • 1: 反馈转矩
19	控制模式切换选择*2	设置要切换的控制模式。 • 0: cst(循环转矩模式) • 1: ct(挡块控制模式)
20、21	挡块控制模式切换时速度初始值选择*3	设置从循环位置模式(csp)向挡块控制模式(ct)切换控制模式时的速度初始值。 • 0: 指令速度 • 1: 反馈速度 • 2: 自动选择

*2 对象数据的“Supported drive modes(6502Hh)(位20)”为TRUE的情况下可以切换至挡块控制。在FALSE的状态下设置了“1: ct(挡块控制模式)”的情况下，将变为驱动器控制模式不支持(错误代码：1B1BH)。

关于挡块控制模式的详细内容，请参阅下述手册。

☞ 所使用的控制器的用户手册

*3 只有在控制模式切换选择(Options: 位19)中设置了“1: ct(挡块控制模式)”的情况下才有效。

• 加减速方式设置(位0~2)

设定值	内容
0: 加减速速度指定方式(mcAccDec)	是使用MC_TorqueControl(转矩控制)中设置的加速度(Acceleration)、减速度(Deceleration)、Jerk进行加速/减速的方式。
1: 加减速时间恒定方式(mcFixedTime)	是与速度无关，使用MC_TorqueControl(转矩控制)中设置的加减速时间进行加速/减速的方式。对于加减速时间，设置为加速度(Acceleration)，不使用减速度(Deceleration)、Jerk。

• 转矩初始值选择(位18)

设定值	内容
0: 目标转矩	控制模式切换之后，与转矩正方向斜率(TorquePositiveRamp)、转矩负方向斜率(TorqueNegativeRamp)的值无关，启动时的目标转矩(Torque)的值将原样不变地变为指令转矩。
1: 反馈转矩	切换时的Torque actual value的值变为指令转矩。*1

*1 未将分配至对象数据的Torque actual value的对象映射到PDO中的情况下，切换时不会变为从驱动器模块接收的反馈转矩。初始值为“0”。

要点

通常应设置为“0: 目标转矩”。只有在至电机的指令完成之后，不等待伺服电机的停止而转移控制模式的情况下才应设置“1: 反馈转矩”。

- 挡块控制模式切换时速度初始值选择(位20、21)

设定值	内容
0: 指令速度	切换之后至驱动器模块的指令速度将变为指令中的速度。
1: 反馈速度	将变为切换时从驱动器模块接收的电机旋转数。 ^{*1}
2: 自动选择	切换之后至驱动器模块的指令速度将变为“0: 指令速度”与“1: 反馈速度”中较低一方的速度。 ^{*2}


*1 未映射对象数据的“Velocity actual value(606CH)”的情况下，不变为切换时从驱动器模块接收的电机旋转数。(速度初始值将变为“0”。)

*2 未映射对象数据的“Velocity actual value(606CH)”的情况下，切换之后至驱动器模块的指令速度将变为“0: 指令速度”。


■关联对象数据

未设置对象数据、未进行PDO映射或设备站中无对象时，错误及规格限制的功能如下所示。

关于对象数据设置的详细内容，请参阅下述手册。


所使用的控制器的用户手册

◎：必须，○：推荐

指令·功能	所需的对象数据	PDO映射	未设置对象时的动作
MC_TorqueControl(转矩控制) (控制模式切换 ^{*1} : cst)	Target torque	◎	必要对象数据未设置(错误代码: 1AF7H)。
	Velocity limit value	○	cst期间基于LimitVelocity的限制速度无效。
	Torque actual value	○	将为转矩初始值选择(选项(Options)位: 18)中设置的动作。  1642页 选项(Options)
MC_TorqueControl(转矩控制) (控制模式切换 ^{*1} : ct)	Target torque	◎	必要对象数据未设置(错误代码: 1AF7H)。
	Velocity limit value	◎	必要对象数据未设置(错误代码: 1AF7H)。
	Torque actual value	○	将为转矩初始值选择(选项(Options)位: 18)中设置的动作。  1642页 选项(Options)
	Velocity actual value	○	将为速度初始值选择(选项(Options)位: 16、17)中设置的动作。  1633页 选项(Options)

*1 可以在控制模式切换选择(Options bit19)中进行设置。

注意事项

- 速度倍率修调系数(AxisName. Cd.VelocityOverride)、加速度倍率修调系数(AxisName. Cd.AccelerationOverride)为有效。
- 通过跟踪更新指令当前位置、进给机械位置。
- 控制模式切换为止的时间取决于驱动器模块的规格。
- 控制模式切换中发生了停止原因的情况下，将立即停止。
- 请勿在控制模式切换中启动定位控制用的FB。驱动器控制模式(AxisName. Md.Driver_Mode)为“10: 转矩控制(cst)”后，应启动定位控制用的FB。
- 将控制模式从循环位置模式(csp)切换为循环速度模式(csv)的情况下，如果在控制模式切换之前发生停止原因(检测出轴错误)，则有可能保持循环位置模式(csp)，进行跟踪直至变为零速度状态。
- 使用MR-J5(W)-G，在不等待电机的停止的状况下从csp切换为cst的情况下，或从cst切换为csp的情况下，应注意以下几点。
 - 应将伺服参数(扩展设置)的“控制切换时ZSP无效选择(PC76.1)”设置为“1: 无效(与ZSP范围无关进行控制切换)”，并将零速度状态的监视设置为无效。但是，切换控制模式时可能会发生振动或冲击。
 - 关于伺服参数(基本设置)的“电子齿轮分子(PA06)”、“电子齿轮分母(PA07)”的设定值，请参阅下述手册。
所使用的控制器的用户手册
- 未映射Velocity limit value的情况下，发生停止原因时将指令转矩更改为0。

■关于循环转矩模式(cst)中的转矩限制更改

- 通过重启或连续更新，将目标转矩更改为大于转矩限制值(正方向转矩限制值(AxisName. Md.TorqueLimit_Positive)、负方向转矩限制值(AxisName. Md.TorqueLimit_Negative))的值时，将发生转矩限制值溢出警告(事件代码: 00D12H)，并以更改前的值执行动作。
- 转矩控制中，如果更改为小于目标转矩的正方向转矩限制值(AxisName. Cd.TorqueLimit_Positive)、负方向转矩限制值(AxisName. Cd.TorqueLimit_Negative)，则将目标转矩更改为转矩限制值。此时，指令转矩以1周期更改。

程序示例

将转矩控制指令(bTorqueControl)置为TRUE，切换至转矩控制，并根据下述设置进行轴1(Axis0001)的转矩控制的程序示例如下所示。

- 设置

项目	设定值
速度	100000.0
加速度	50000.0
减速度	50000.0
Jerk	0.0

■轴

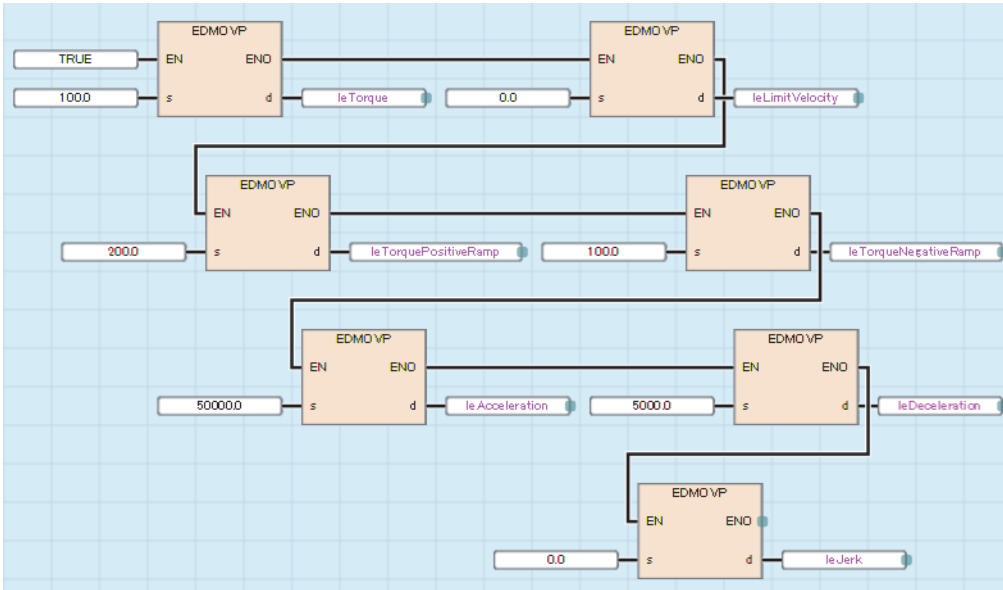
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

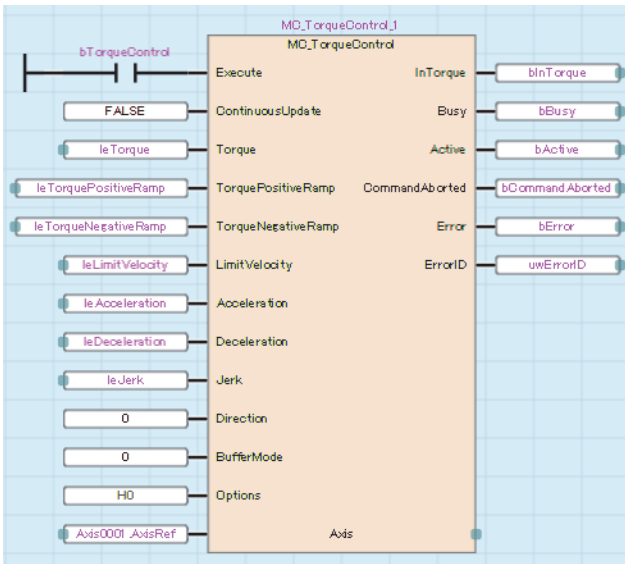
标签名	数据类型	注释
MC_TorqueControl_1	MC_TorqueControl	转矩控制FB
bTorqueControl	位	转矩控制指令
leTorque	双精度实数	目标转矩
leTorquePositiveRamp	双精度实数	转矩正方向斜率
leTorqueNegativeRamp	双精度实数	转矩负方向斜率
leLimitVelocity	双精度实数	限制速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bInTorque	位	目标转矩到达
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

- 转矩控制用数据设置



- 转矩控制



■ST程序

```
//-----转矩控制用数据设置-----
leTorque:= 100.0;
leTorquePositiveRamp:= 200.0;
leTorqueNegativeRamp:= 100.0;
leLimitVelocity:= 100000.0;
leAcceleration:= 50000.0;
leDeceleration:= 50000.0;
leJerk:= 0.0;

//-----转矩控制-----
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
);
```

46.9 速度控制(包含位置循环)

MCv_SpeedControl

执行包含位置循环的速度控制。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName, AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MCv_SpeedControl(速度控制(包含位置循环))。
ContinuousUpdate	连续更新	BOOL	启动时	TRUE、FALSE	FALSE	设置将速度(Velocity)、加速度(Acceleration)、减速度(Deceleration)的连续更改是置为有效，还是置为无效。 • FALSE: 无效 • TRUE: 有效
Velocity	速度	LREAL	启动时/可重启/可连续更新	0.0、±0.0001~±2500000000.0	0.0	设置指令速度。 速度为负的情况下，向反转方向移动。 省略设置时，轴不动作但轴状态(AxisName, Md, AxisStatus)变为“6: 连续动作运行中(ContinuousMotion)”。 关于详细内容，请参阅下述章节。 ☞ 1649页 速度(Velocity)
Acceleration	加速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置加速度。 关于详细内容，请参阅下述章节。 ☞ 1649页 加速度(Acceleration)
Deceleration	减速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置减速度。 关于详细内容，请参阅下述章节。 ☞ 1650页 减速度(Deceleration)
Jerk	Jerk	LREAL	启动时	0.0000、0.0001~2147483647.0	0.0	设置Jerk。 关于详细内容，请参阅下述章节。 ☞ 1650页 Jerk

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Direction	方向选择	INT (MC_DIRECTION)	启动时	1、2	0	设置方向选择。 • 1: 正方向(mcPositiveDirection) • 2: 负方向(mcNegativeDirection) * 设置了“2: 负方向(mcNegativeDirection)”, 且速度(Velocity)为负的情况下, 电机的移动方向 将变为正方向。 * 省略了设置的情况下, 将变为超出方向选择范围 (错误代码: 1AA5H)。
BufferMode	缓冲模式	INT (MC_BUFFER_MODE)	启动时	0~5	0	设置缓冲模式。 • 0: Aborting(mcAborting) • 1: Buffered(mcBuffered) • 2: BlendingLow(mcBlendingLow) • 3: BlendingPrevious(mcBlendingPrevious) • 4: BlendingNext(mcBlendingNext) • 5: BlendingHigh(mcBlendingHigh) 关于详细内容, 请参阅下述章节。 1650页 缓冲模式(BufferMode)
Options	选项	DWORD (HEX)	启动时	00000000H~00000001H	00000000H	将MCv_SpeedControl(速度控制(包含位置循环))的 功能选项以位指定进行设置。 关于详细内容, 请参阅下述章节。 1651页 选项(Options)

■输出变量

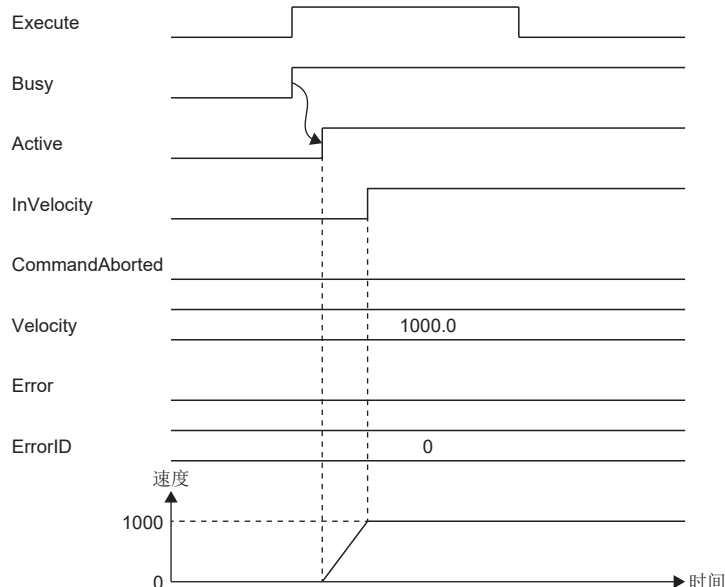
输出变量	名称	数据类型	初始值	内容
InVelocity	目标速度到达	BOOL	FALSE	指令速度到达了目标速度时, 将变为TRUE。 通过连接更新(ContinuousUpdate)为有效(TRUE)时的更改更改了目标速度的情况下, 在到达更改后的目标速度之前将变为FALSE。
Busy	执行中	BOOL	FALSE	执行了MCv_SpeedControl(速度控制(包含位置循环))时, 将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_SpeedControl(速度控制(包含位置循环))正在控制轴时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MCv_SpeedControl(速度控制(包含位置循环))的执行中断时, 将变为TRUE。 执行指令(Execute)变为FALSE时, 将变为FALSE。
Error	错误	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时, 将返回错误代码。 关于错误代码的详细内容, 请参阅下述手册。 所使用的控制器的用户手册

功能

将驱动器模块侧的控制模式设置为csp(循环位置模式)，并根据对设置的轴设置的速度执行速度控制。停止轴时，使用MC_Stop(强制停止)，或启动其它的运行系统FB。将驱动器模块侧的控制模式设置为csv(循环速度模式)进行速度控制的情况下，应使用MC_MoveVelocity(速度控制)。

■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■速度(Velocity)

设置MCv_SpeedControl(速度控制(包含位置循环))中的指令速度。

设置范围

0.0、 $\pm 0.0001 \sim \pm 2500000000.0^{*1*2}$

- *1 由于进行浮点运算，因此指令速度的下限值中将产生下述限制。

对指令速度进行了运算周期换算的速度小于“0.00001”的情况下，将变为超出运算周期换算速度范围(错误代码：1B16H)(速度更改时为超出运算周期换算速度范围警告(事件代码：00D2FH))。为了提高浮点运算的精度，应通过更改位置指令单位(AxisName.Pr.Unit_Position)或速度指令单位(AxisName.Pr.Unit_Velocity)设置运算周期换算后的速度不低于“0.00001”。

- *2 在进行多重启动的运动控制FB中指定速度为“0.0”的情况下，将变为之前的运动控制FB的指定速度。

■加速度(Acceleration)

设置MCv_SpeedControl(速度控制(包含位置循环))中的加速度。

根据加减速方式设置(选项(Options):位0~2)的设置，设置范围有所不同。

加减速方式设置(选项(Options):位0~2)	设置范围
0: 加减速速度指定方式(mcAccDec)	0.0000^{*1} 、 $0.0001 \sim 2147483647.0 [U/s^2]$ 的正数
1: 加减速时间恒定方式(mcFixedTime)	0.000000^{*1} 、 $0.000001 \sim 8400.0 [s]$ 的正数

- *1 启动时，根据启动时加减速速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior)，动作将发生变化。加速度更改、减速度更改时，不受理更改。

■减速度(Deceleration)

设置MCv_SpeedControl(速度控制(包含位置循环))中的减速度。

设置了加减速方式设置(选项(Options):位0~2)的“0:加减速速度指定方式(mcAccDec)”的情况下,对设置范围进行设置。设置了“1:加减速时间恒定方式(mcFixedTime)”的情况下,不使用减速度(Deceleration)。

加减速方式设置(选项(Options):位0~2)	设置范围
0:加减速速度指定方式(mcAccDec)	0.0000*1、0.0001~2147483647.0[U/s ²]的正数
1:加减速时间恒定方式(mcFixedTime)	不使用

*1 启动时,根据启动时加减速速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior),动作将发生变化。加速度更改、减速度更改时,不受理更改。

■Jerk

设置MCv_SpeedControl(速度控制(包含位置循环))中的Jerk。

设置了加减速方式设置(选项(Options):位0~2)的“0:加减速速度指定方式(mcAccDec)”的情况下,对设置范围进行设置。设置了“1:加减速时间恒定方式(mcFixedTime)”的情况下,不使用Jerk。

加减速方式设置(选项(Options):位0~2)	设置范围
0:加减速速度指定方式(mcAccDec)	0.0000、0.0001~2147483647.0[U/s ²]的正数
1:加减速时间恒定方式(mcFixedTime)	不使用

■缓冲模式(BufferMode)

设置用于进行多重启动(缓冲模式)的动作。


MCv_SpeedControl(速度控制(包含位置循环))中可设置的缓冲模式如下所示。

设定值	内容
0: Aborting(mcAborting)	中断(取消)控制中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在控制中的FB中缓冲下一个FB。 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB完成时,依次执行缓冲FB。
2: BlendingLow(mcBlendingLow)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将控制中的FB与缓冲FB的目标速度中较低一方的速度作为切换速度。
3: BlendingPrevious(mcBlendingPrevious)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将控制中的FB的目标速度作为切换速度。
4: BlendingNext(mcBlendingNext)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将缓冲FB的目标速度作为切换速度。
5: BlendingHigh(mcBlendingHigh)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将控制中的FB与缓冲FB的目标速度中较高一方的速度作为切换速度。

*1 在控制中的FB与缓冲FB之间不执行停止。

要点

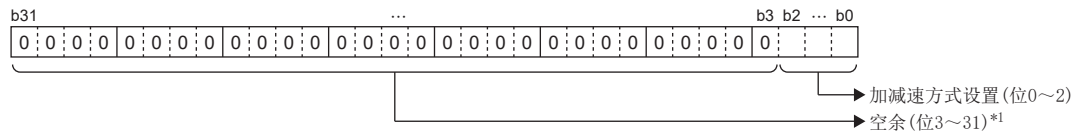
关于多重启动(缓冲模式)的详细内容,请参阅下述手册。

所使用的控制器的用户手册

■选项(Options)

将MCv_SpeedControl(速度控制(包含位置循环))中使用的功能选项以位指定进行设置。

以位指定设置的内容如下所示。



*1 空余中应设置“0”。设置了“0”以外的情况下，将变为超出Options范围(错误代码: 1ABBH)。

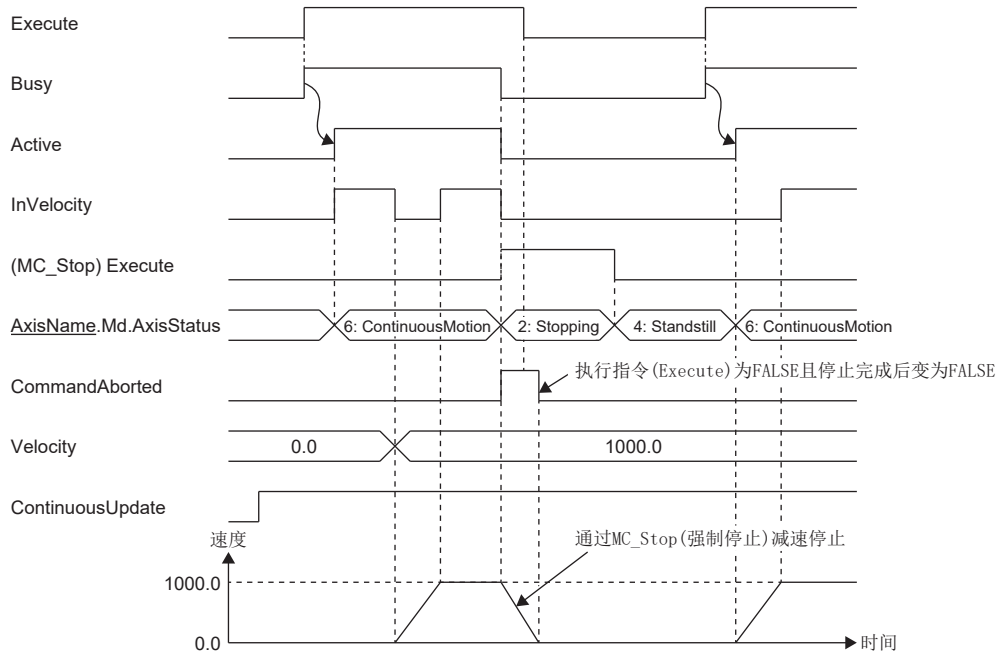
位	名称	内容
0~2	加减速方式设置	设置用于进行控制的加减速方式。 <ul style="list-style-type: none"> • 0: 加减速速度指定方式(mcAccDec) • 1: 加减速时间恒定方式(mcFixedTime)

• 加减速方式设置(位0~2)

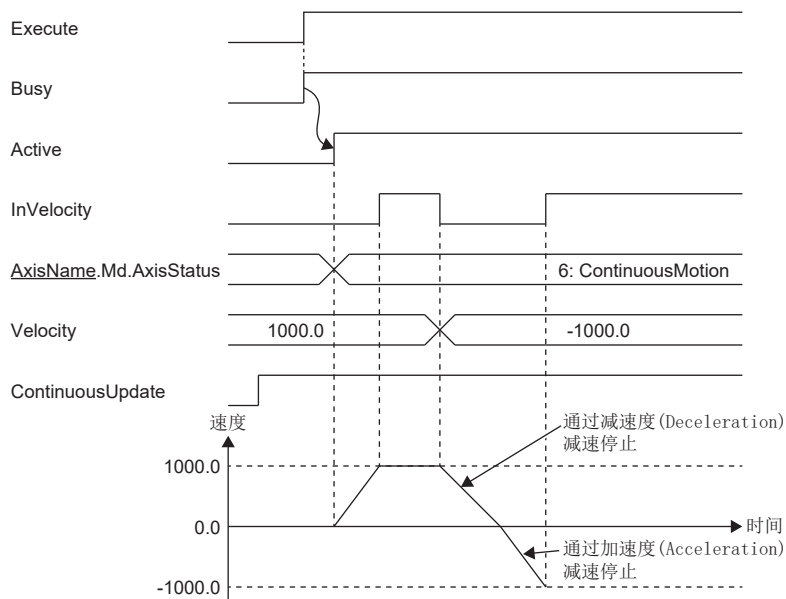
设定值	内容
0: 加减速速度指定方式(mcAccDec)	是使用MCv_SpeedControl(速度控制(包含位置循环))中设置的加速度(Acceleration)、减速度(Deceleration)、Jerk进行加速/减速的方式。
1: 加减速时间恒定方式(mcFixedTime)	是与速度无关, 使用MCv_SpeedControl(速度控制(包含位置循环))中设置的加减速时间进行加速/减速的方式。对于加减速时间, 设置为加速度(Acceleration), 不使用减速度(Deceleration)、Jerk。

■动作概要

- 启动时及发生停止原因时的动作如下所示。



- 连续更新(ContinuousUpdate)为TRUE时，速度(Velocity)的符号反转，运行的方向发生变化的情况下，一旦减速停止后将向目标速度进行加速度。



■必要对象数据

使用单轴的MCv_SpeedControl(速度控制(包含位置循环))的情况下，应在轴中设置下述对象数据。

- Target position(607AH)

未设置对象数据的情况下，将变为必须从站对象未设置(错误代码：1AF7H)且不启动。

关于对象数据设置的详细内容，请参阅下述手册。

📖所使用的控制器的用户手册

程序示例

将速度控制指令(bSpeedControl)置为TRUE, 切换至包含位置循环的速度控制, 并根据下述设置进行轴1(Axis0001)的包含位置循环的速度控制的程序示例如下所示。

- 设置

项目	设定值
速度	100000.0
加速度	50000.0
减速度	50000.0
Jerk	0.0

■轴

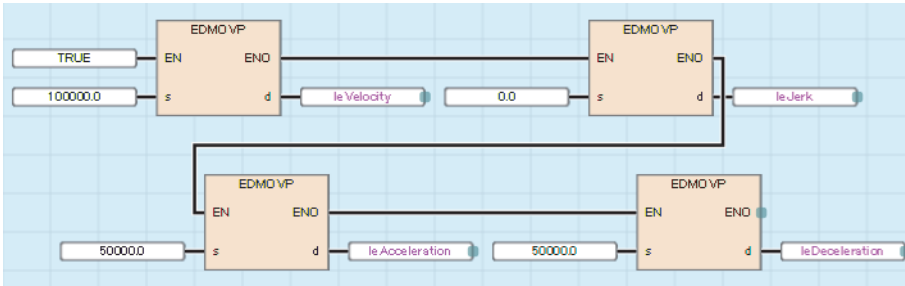
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

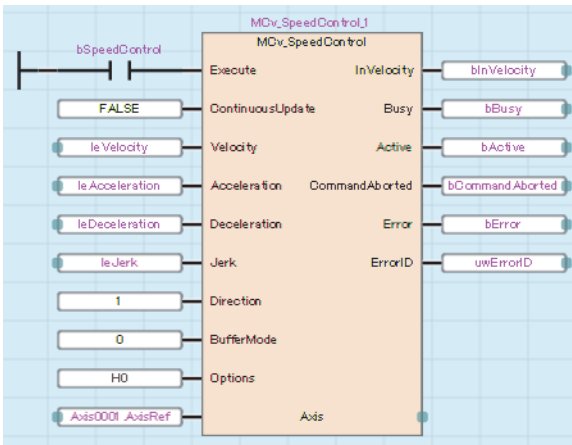
标签名	数据类型	注释
MCv_SpeedControl_1	MCv_SpeedControl	速度控制FB
bSpeedControl	位	速度控制指令
leVelocity	双精度实数	指令速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bInVelocity	位	目标速度到达
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	错误
uwErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

- 速度控制用数据的设置



- 速度控制(包含位置循环)



■ST程序

```
//-----速度控制用数据的设置-----
leVelocity:= 100000.0;
leAcceleration:= 50000.0;
leDeceleration:= 50000.0;
leJerk:= 0.0;

//-----速度控制(包含位置循环)-----
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
);
```

46.10 绝对值直线插补控制

MCv_MoveLinearInterpolateAbsolute

对设置的轴组的绝对位置的目标位置进行设置，并通过直线插补控制执行定位。

梯形图	FBD/LD	ST
		<pre> MCv_MoveLinearInterpolateAbsolute(AxesGroup:= ?AXES_GROUP_REF?, Execute:= ?BOOL?, ContinuousUpdate:= ?BOOL?, LinearAxes:= ?INT(0..15)?, Position:= ?LREAL(0..15)?, Velocity:= ?LREAL?, Acceleration:= ?LREAL?, Deceleration:= ?LREAL?, Jerk:= ?LREAL?, VelocityMode:= ?INT?, Direction:= ?INT(0..15)?, BufferMode:= ?INT?, Options:= ?DWORD?, Done=> ?BOOL?, Busy=> ?BOOL?, Active=> ?BOOL?, CommandAborted=> ?BOOL?, Error=> ?BOOL?, ErrorID=> ?WORD?); </pre>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
AxesGroup	轴组信息	AXES_GROUP_REF	启动时	—	不能省略	设置轴组。 关于使用的变量 (<u>AxesGroupName.AxesGroupRef.</u>)，请参阅下述内容。 ☞ 1408页 <u>AxesGroupName.AxesGroupRef.</u> (轴组信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	启动	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行 MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)。
ContinuousUpdate	连续更新	BOOL	启动时	TRUE、FALSE	FALSE	设置将速度(Velocity)、加速度(Acceleration)、减速度(Deceleration)的连续更改是置为有效，还是置为无效。 通过连续更新，对执行中的实例在不中断动作的状况下进行输入变量的重新获取。 • FALSE: 无效 • TRUE: 有效
LinearAxes	直线插补轴	INT[0..15]	启动时	1~16	0	从配置轴中设置直线插补控制中使用的轴。 以数组设置配置轴的索引编号(1~16)。 * 在速度模式(VelocityMode)中设置了“2: 基准轴速度(ReferenceAxisSpeed)”的情况下，将数组的第1元素视为基准轴。 关于详细内容，请参阅下述章节。 ☞ 1659页 直线插补轴(LinearAxes)
Position	目标位置	LREAL[0..15]	启动时	-10000000000.0~10000000000.0	0.0	设置直线插补的目标位置。 是1维的数组数据。作为配置轴1~16的绝对位置处理。 关于详细内容，请参阅下述章节。 ☞ 1659页 目标位置(Position)

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Velocity	速度	LREAL	启动时/可重启/可连续更新	0.0、0.0001~250000000.0	0.0	设置速度指令值。 关于详细内容，请参阅下述章节。 ☞ 1660页 速度(Velocity)
Acceleration	加速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置加速度。 关于详细内容，请参阅下述章节。 ☞ 1660页 加速度(Acceleration)
Deceleration	减速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置减速度。 关于详细内容，请参阅下述章节。 ☞ 1660页 减速度(Deceleration)
Jerk	Jerk	LREAL	启动时	0.0000、0.0001~2147483647.0	0.0	设置Jerk。 关于详细内容，请参阅下述章节。 ☞ 1660页 Jerk
VelocityMode	速度模式	INT (MC_INTERPOLATE_SPEED_MODE)	启动时	0~2	0	设置插补控制的速度模式。 • 0: 合成速度(VectorSpeed) • 1: 长轴速度(LongAxisSpeed) • 2: 基准轴速度(ReferenceAxisSpeed) 关于详细内容，请参阅下述章节。 ☞ 1661页 速度模式(VelocityMode)
Direction	方向选择	INT (MC_DIRECTION[0..15])	启动时	1~3	0	设置方向选择。 是1维的数组数据。作为配置轴1~16的方向选择处理。 • 1: 正方向(mcPositiveDirection) • 2: 负方向(mcNegativeDirection) • 3: 最短路径(mcShortestWay) * 省略了设置的情况下，将变为超出方向选择范围(错误代码: 1AA5H)。 关于详细内容，请参阅下述章节。 ☞ 1663页 方向选择(Direction)
BufferMode	缓冲模式	INT (MC_BUFFER_MODE)	启动时	0~5	0	设置缓冲模式。 • 0: Aborting(mcAborting) • 1: Buffered(mcBuffered) • 2: BlendingLow(mcBlendingLow) • 3: BlendingPrevious(mcBlendingPrevious) • 4: BlendingNext(mcBlendingNext) • 5: BlendingHigh(mcBlendingHigh) 关于详细内容，请参阅下述章节。 ☞ 1663页 缓冲模式(BufferMode)
Options	选项	DWORD (HEX)	启动时	00000000H~00010001H	00000000H	将MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)的功能选项以位指定进行设置。 关于详细内容，请参阅下述章节。 ☞ 1664页 选项(Options)

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	完成	BOOL	FALSE	控制完成时，将变为TRUE。 动作完成时根据启动(Execute)的状态将变为如下所示。 ■启动(Execute)为TRUE的情况下 在将启动(Execute)置为FALSE之前将保持为TRUE不变。 ■启动(Execute)为FALSE的情况下 仅1周期为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)时，将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)正在控制轴时，将变为TRUE。 对同一轴组执行了多个MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)的情况下，控制中(Active)变为TRUE的只有1个MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)。
CommandAborted	执行中断	BOOL	FALSE	由于错误及多重启动等，MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)的执行中断时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 ☞ 所使用的控制器的用户手册

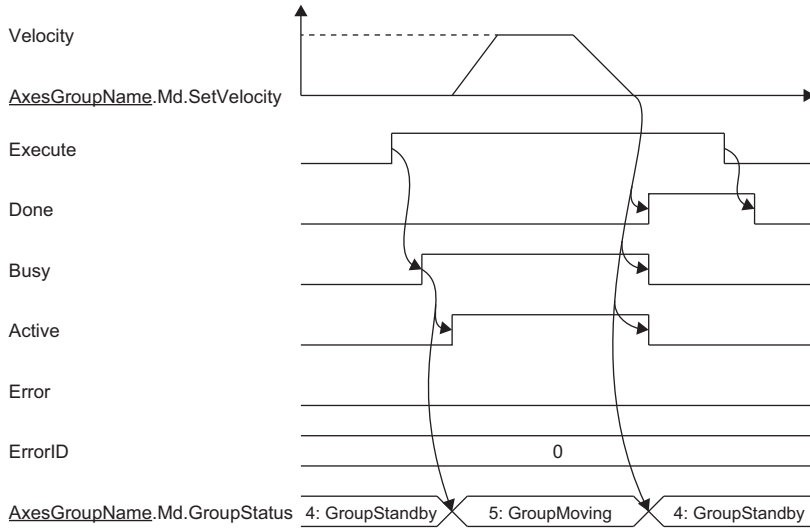
功能

- 直线插补控制指定轴组，进行插补控制，使从始点(移动开始点)到终点的轨迹为直线。在直线插补控制中，进行最大使用4轴的插补控制。
- 在MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)中，指定绝对位置的目标位置进行绝对值直线插补控制。
- 关于1轴~3轴的绝对值直线插补控制的动作，请参阅下述手册的“直线插补控制”。

📖 所使用的控制器的用户手册

■ 时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

📖 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■直线插补轴 (LinearAxes)

在直线插补控制中，从轴组中设置的配置轴中，使用任意的轴进行直线插补。

从轴组的配置轴中，通过直线插补轴 (LinearAxes) 设置进行直线插补的配置轴。直线插补轴 (LinearAxes) 具有16个数组元素。在数组中，设置插补控制中使用的配置轴的索引编号 (1~16: 配置轴1~16) 仅以插补控制中使用的配置轴向前填充对齐的方式进行设置，并在剩余的数组中设置“0”。也可以省略设置了“0”的数组。

- 设置时应使直线插补轴的设置数不大于直线插补控制的最大插补轴数的4轴。
- 设置时应使插补控制中使用的配置轴数不大于配置轴中登录的轴数。
- 速度模式 (VelocityMode) 为“2: 基准轴速度 (ReferenceAxisSpeed)”的情况下，直线插补轴 (LinearAxes) 的第1元素的配置轴将成为基准轴。
- 进行了下述设置的情况下，将变为直线插补轴设置不正确 (错误代码: 1B08H) 且不启动。
 - 设置了轴未设置的配置轴的情况下
 - 直线插补轴 (LinearAxes) 的第1元素为“0”的情况下
 - 直线插补轴 (LinearAxes) 中重复设置了相同的索引编号的情况下
 - 直线插补轴 (LinearAxes) 的设置轴数超出了最大插补轴数的情况下

例

将配置轴2、3、4设置为直线插补轴的情况下

```
LinearAxes[0]:= 2;
LinearAxes[1]:= 3;
LinearAxes[2]:= 4;
LinearAxes[3]:= 0;*1
:
LinearAxes[15]:= 0;*1
```

*1 可以省略“LinearAxes[3]:= 0;~LinearAxes[15]:= 0;”。

■目标位置 (Position)

设置直线插补的目标位置。

目标位置 (Position) 具有16个数组元素。

- 可设置的有效范围根据软件行程限位有效/无效、方向选择 (Direction)、超出环形计数器的目标位置指定 (选项 (Options): 位16) 而有所不同。关于详细内容，请参阅下述章节。

☞ 1600页 目标位置 (Position)

- Position[0..15] 表示配置轴1~16的目标位置。对直线插补轴 (LinearAxes) 中设置的配置轴设置位置。
- 未设置为直线插补轴的配置轴的目标位置 (Position) 将被忽略。
- 设置为直线插补轴的所有配置轴后面的目标位置 (Position) 可以省略。

例

将配置轴2、3、4设置为直线插补轴，并在配置轴2、3、4的目标位置中设置2000.0、3000.0、4000.0的情况下

```
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 可以省略“LinearAxes[3]:= 0;~LinearAxes[15]:= 0;”和“Position[4]:= 0.0;~Position[15]:= 0.0;”。

■速度 (Velocity)

设置MCv_MoveLinearInterpolateAbsolute (绝对值直线插补控制) 中的路径的最大速度。

设置范围
0.0、0.0001~2500000000.0*1*2

- *1 由于进行浮点运算，因此指令速度的下限值中将产生下述限制。
对指令速度进行了运算周期换算的速度小于“0.00001”的情况下，将变为超出运算周期换算速度范围(错误代码：1B16H)(速度更改时为超出运算周期换算速度范围警告(事件代码：00D2FH))。为了提高浮点运算的精度，应通过更改位置指令单位(AxisName.Pr.Unit_Position)或速度指令单位(AxisName.Pr.Unit_Velocity)设置运算周期换算后的速度不低于“0.00001”。
- *2 在进行多重启动的运动控制FB中指定速度为“0.0”的情况下，将变为之前的运动控制FB的指定速度。

■加速度 (Acceleration)

设置MCv_MoveLinearInterpolateAbsolute (绝对值直线插补控制) 中的加速度。

根据加减速方式设置(选项(Options)：位0~2)的设置，设置范围有所不同。

加减速方式设置(选项(Options)：位0~2)	设置范围
0：加减速速度指定方式(mcAccDec)	0.0000*1、0.0001~2147483647.0[U/s ²]的正数
1：加减速时间恒定方式(mcFixedTime)	0.000000*1、0.000001~8400.0[s]的正数

- *1 启动时，根据启动时加减速速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior)，动作将发生变化。加速度更改、减速度更改时，不受理更改。

■减速度 (Deceleration)

设置MCv_MoveLinearInterpolateAbsolute (绝对值直线插补控制) 中的减速度。

设置了加减速方式设置(选项(Options)：位0~2)的“0：加减速速度指定方式(mcAccDec)”的情况下，对设置范围进行设置。设置了“1：加减速时间恒定方式(mcFixedTime)”的情况下，不使用减速度(Deceleration)。

加减速方式设置(选项(Options)：位0~2)	设置范围
0：加减速速度指定方式(mcAccDec)	0.0000*1、0.0001~2147483647.0[U/s ²]的正数
1：加减速时间恒定方式(mcFixedTime)	不使用

- *1 启动时，根据启动时加减速速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior)，动作将发生变化。加速度更改、减速度更改时，不受理更改。

■Jerk

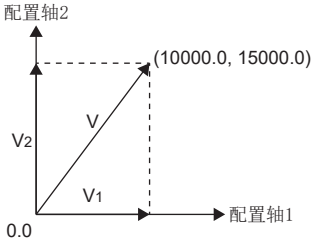
设置MCv_MoveLinearInterpolateAbsolute (绝对值直线插补控制) 中的Jerk。

设置了加减速方式设置(选项(Options)：位0~2)的“0：加减速速度指定方式(mcAccDec)”的情况下，对设置范围进行设置。设置了“1：加减速时间恒定方式(mcFixedTime)”的情况下，不使用Jerk。

加减速方式设置(选项(Options)：位0~2)	设置范围
0：加减速速度指定方式(mcAccDec)	0.0000、0.0001~2147483647.0[U/s ²]的正数
1：加减速时间恒定方式(mcFixedTime)	不使用

■速度模式 (VelocityMode)

设置直线插补控制中的速度模式。在速度模式 (VelocityMode) 中设置了“2: 基准轴速度 (ReferenceAxisSpeed)”的情况下, 基准轴将变为直线插补轴 (LinearAxes) 的第1元素中设置的配置轴。

设定值	内容
0: 合成速度 (VectorSpeed)	<p>速度 (Velocity) 设置为合成速度。 各轴的定位速度 (V_n) 是根据设置的控制对象的定位速度 (V), 由运动系统根据各轴的移动量 (D_n) 计算的。 <例> 2轴的直线插补控制的情况下</p>  <ul style="list-style-type: none"> 配置轴1的移动量 (D_1): 10000.0[pulse] 配置轴2的移动量 (D_2): 15000.0[pulse] 轴组的速度单位: [s] 合成速度 (V): 7000.0[pulse/s] <p>上述的情况下, 运动系统通过下述计算公式计算各轴的定位速度。</p> <ul style="list-style-type: none"> 配置轴1的定位速度: $V_1 = V \times D_1 / \sqrt{D_1^2 + D_2^2}$ 配置轴2的定位速度: $V_2 = V \times D_2 / \sqrt{D_1^2 + D_2^2}$ <p>■要点</p> <ul style="list-style-type: none"> 合成速度的情况下, 设置的速度限制值将对作为合成速度的速度 (Velocity) 有效。
1: 长轴速度 (LongAxisSpeed)	<p>速度 (Velocity) 设置为长轴的速度。 各插补轴中设置的指定位置中, 根据移动量最大的插补轴的定位速度 (长轴速度: V) 进行控制。 其他插补轴的定位速度 (V_n) 是由运动系统根据各插补轴的移动量 (D_n) 计算的。 <例> 4轴的直线插补控制的情况下</p> <ul style="list-style-type: none"> 配置轴1的移动量 (D_1): 10000.0[pulse] 配置轴2的移动量 (D_2): 15000.0[pulse] 配置轴3的移动量 (D_3): 5000.0[pulse] 配置轴4的移动量 (D_4): 20000.0[pulse] 配置轴4的速度单位: [s] 长轴速度 (V): 7000.0[pulse/s] <p>上述的情况下, 长轴将为移动量最大的配置轴4, 并以长轴速度控制配置轴4。 对于其它配置轴的定位速度, 运动系统通过下述计算公式计算。</p> <ul style="list-style-type: none"> 配置轴1的定位速度: $V_1 = D_1 / D_4 \times V$ 配置轴2的定位速度: $V_2 = D_2 / D_4 \times V$ 配置轴3的定位速度: $V_3 = D_3 / D_4 \times V$ <p>■要点</p> <ul style="list-style-type: none"> 长轴速度的情况下, 设置的速度限制值对作为长轴速度的速度 (Velocity) 有效。 应注意长轴速度指定时的合成速度有可能大于速度限制值。2轴的直线插补中设置了下述值的情况下, 合成速度超出速度限制值。 <p><例> 设置项目与设定值为下述的情况下</p> <ul style="list-style-type: none"> 直线插补轴: 配置轴1、配置轴2 配置轴1的移动量: 100[pulse] 配置轴2的移动量: 200[pulse] 长轴速度: 50[pulse/s] 配置轴2的速度限制值: 55[pulse/s] <p>上述的情况下, 基准轴为移动量最大的配置轴2, 并以配置轴2中设置的速度限制值控制。此外, 各轴的定位速度及合成速度如下所示。</p> <ul style="list-style-type: none"> 配置轴1的定位速度: $100 / 200 \times 50 = 25$[pulse/s] 配置轴2的定位速度: 50[pulse/s] 合成速度: $\sqrt{25^2 + 50^2} = 55.9$[pulse/s] <p>合成速度为超过配置轴2的速度限制值“55”的值。</p>

设定值	内容
2: 基准轴速度(ReferenceAxisSpeed)	<p>速度(Velocity)设置为基准轴的速度。</p> <p>根据设置的基准轴的定位速度(基准轴速度: V), 运动系统通过各插补轴的移动量(Dn)计算其它插补轴的定位速度(Vn)并进行控制。</p> <p>设置为基准轴的配置轴移动量为“0.0”的情况下, 将变为基准轴移动量0(错误代码: 1AFAH)。</p> <p><例></p> <p>4轴的直线插补控制的情况下</p> <ul style="list-style-type: none"> 配置轴1的移动量(D₁): 10000.0[pulse] 配置轴2的移动量(D₂): 15000.0[pulse] 配置轴3的移动量(D₃): 5000.0[pulse] 配置轴4的移动量(D₄): 20000.0[pulse] 基准轴速度(V): 7000.0[pulse/s] <p>上述的情况下, 基准轴将为配置轴4, 并以配置轴4中设置的定位速度进行控制。</p> <p>对于其它轴的定位速度, 运动系统通过下述计算公式计算。</p> <ul style="list-style-type: none"> 配置轴1的定位速度: $V_1 = D_1 / D_4 \times V$ 配置轴2的定位速度: $V_2 = D_2 / D_4 \times V$ 配置轴3的定位速度: $V_3 = D_3 / D_4 \times V$ <p>■要点</p> <ul style="list-style-type: none"> 基准轴速度的情况下, 设置的速度限制值对作为基准轴速度的速度(Velocity)有效。 应注意比基准轴移动量大的轴的定位速度, 将大于设置的基准轴速度。2轴的直线插补中设置了下述值的情况下, 配置轴2的定位速度与合成速度超出速度限制值。 <p><例></p> <p>设置项目与设定值为下述的情况下</p> <ul style="list-style-type: none"> 直线插补轴: 配置轴1、配置轴2 配置轴1的移动量: 100[pulse] 配置轴2的移动量: 200[pulse] 基准轴速度: 50[pulse/s] 配置轴1的速度限制值: 55[pulse/s] <p>上述的情况下, 基准轴为配置轴1, 并以配置轴1中设置的速度限制值进行控制。此外, 各插补轴的定位速度及合成速度如下所示。</p> <ul style="list-style-type: none"> 配置轴1的定位速度: 50[pulse/s] 配置轴2的定位速度: $200 / 100 \times 50 = 100$[pulse/s] 合成速度: $\sqrt{50^2 + 100^2} = 111.8$[pulse/s] <p>配置轴2的定位速度与合成速度为超过配置轴1的速度限制值“55”的值。</p>

限制事项

- 在速度模式(VelocityMode)中设置了“0: 合成速度(VectorSpeed)”时, 如果各轴的移动量超过“4, 294, 967, 296.0(=2³²)”, 将发生线性移动量超出范围(错误代码: 1B10H)且不启动。
- 将缓冲模式(BufferMode)设置为“0: Aborting(mcAborting)”, 对MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)进行多重启动的情况下, 速度模式(VelocityMode)中应设置“0: 合成速度(VectorSpeed)”。设置了“0: 合成速度(VectorSpeed)”以外的情况下, 将变为超出速度模式范围(错误代码: 1AC9H)。

■方向选择(Direction)

可以从正方向、负方向、最短路径中选择，并设置从当前位置向目标位置移动的方向。方向选择(Direction)具有16个数组元素。

- Direction[0..15]表示配置轴1~16的方向选择。对直线插补轴(LinearAxes)中设置的配置轴设定值。
- 未设置为直线插补轴的配置轴的方向选择(Direction)将被忽略。
- 设置为直线插补轴的所有配置轴后面的方向选择(Direction)可以省略。
- 设置了设定值以外的值的情况下，将变为超出方向选择范围(错误代码：1AA5H)且不启动。
- 对于软件行程限位为有效的插补轴，将忽略方向选择(Direction)的设置。关于方向选择的动作详细内容，请参阅下述章节。

☞ 1602页 方向选择(Direction)

设定值	内容
1: 正方向(mcPositiveDirection)	从当前位置向正方向(地址增加)的目标位置进行定位。
2: 负方向(mcNegativeDirection)	从当前位置向负方向(地址减少)的目标位置进行定位。
3: 最短路径(mcShortestWay)	向当前位置与目标位置的距离较近的方向进行定位。

■缓冲模式(BufferMode)

设置用于进行多重启动(缓冲模式)的动作。

MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)中可设置的缓冲模式如下所示。

设定值	内容
0: Aborting(mcAborting)	中断(取消)控制中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在控制中的FB中缓冲下一个FB。 控制中的FB已经被缓冲的情况下，对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB完成时，依次执行缓冲FB。
2: BlendingLow(mcBlendingLow)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下，对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后，依次执行缓冲FB。 控制中的FB到达了目标位置时，将控制中的FB与缓冲FB的目标速度中较低一方的速度作为切换速度。
3: BlendingPrevious(mcBlendingPrevious)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下，对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后，依次执行缓冲FB。 控制中的FB到达了目标位置时，将控制中的FB的目标速度作为切换速度。
4: BlendingNext(mcBlendingNext)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下，对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后，依次执行缓冲FB。 控制中的FB到达了目标位置时，将缓冲FB的目标速度作为切换速度。
5: BlendingHigh(mcBlendingHigh)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下，对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后，依次执行缓冲FB。 控制中的FB到达了目标位置时，将控制中的FB与缓冲FB的目标速度中较高一方的速度作为切换速度。

*1 在控制中的FB与缓冲FB之间不执行停止。

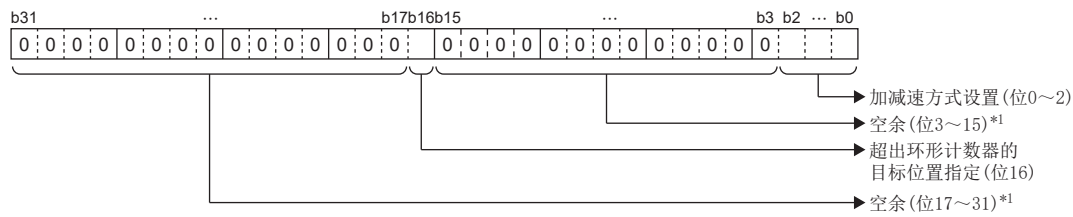
要点

关于多重启动(缓冲模式)的详细内容，请参阅下述手册。

☞ 所使用的控制器的用户手册

■选项(Options)

将MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)中使用的功能选项以位指定进行设置。
以位指定设置的内容如下所示。



*1 空余中应设置“0”。设置了“0”以外的情况下，将变为超出Options范围(错误代码：1ABBH)。

位	名称	内容
0~2	加减速方式设置	设置用于进行控制的加减速方式。 • 0: 加减速速度指定方式 (mcAccDec) • 1: 加减速时间恒定方式 (mcFixedTime)
16	超出环形计数器的目标位置指定	设置软件行程限位无效时，是否允许超出环形计数器上限值、环形计数器下限值的目标位置。 • 0: 不允许 • 1: 允许

• 加减速方式设置(位0~2)

设定值	内容
0: 加减速速度指定方式 (mcAccDec)	是使用MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)中设置的加速度(Acceleration)、减速度(Deceleration)、Jerk进行加速/减速的方式。
1: 加减速时间恒定方式 (mcFixedTime)	是与速度无关，使用MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)中设置的加减速时间进行加速/减速的方式。 对于加减速时间，设置为加速度(Acceleration)，不使用减速度(Deceleration)、Jerk。

■必要对象数据

使用MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)的情况下，应对指定的轴组的所有配置轴设置下述对象数据。

- Target position (607AH)

有未设置对象数据的配置轴的情况下，将变为必要对象数据未设置(错误代码：1AF7H)且不启动。

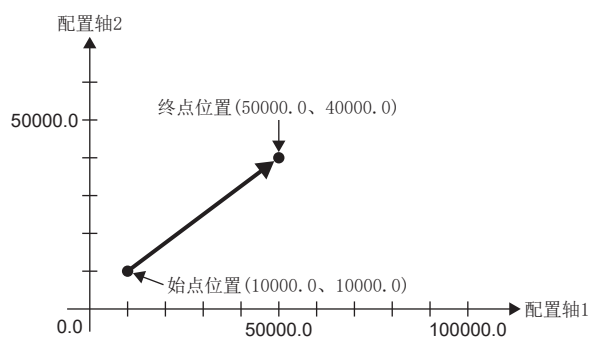
关于对象数据设置的详细内容，请参阅下述手册。

📖所使用的控制器的用户手册

程序示例

将直线插补控制启动(bLinearInterpolateCMD)置为TRUE, 并将轴组1(AxesGroup001)置为了有效后, 根据下述设置进行轴组1(AxesGroup001)的2轴直线插补的绝对值定位控制的程序示例如下所示。

• 动作



• 轴组

项目	设定值
配置轴[1]	Axis0001
配置轴[2]	Axis0002

• 设置

项目	设定值	
	配置轴1	配置轴2
目标位置	50000.0	40000.0
速度	50000.0	
加速度	50000.0	
减速度	50000.0	
Jerk	0.0	
方向选择	正方向	正方向

■ 轴

轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1
2	Axis0002	AXIS_REF	轴2

■ 轴组

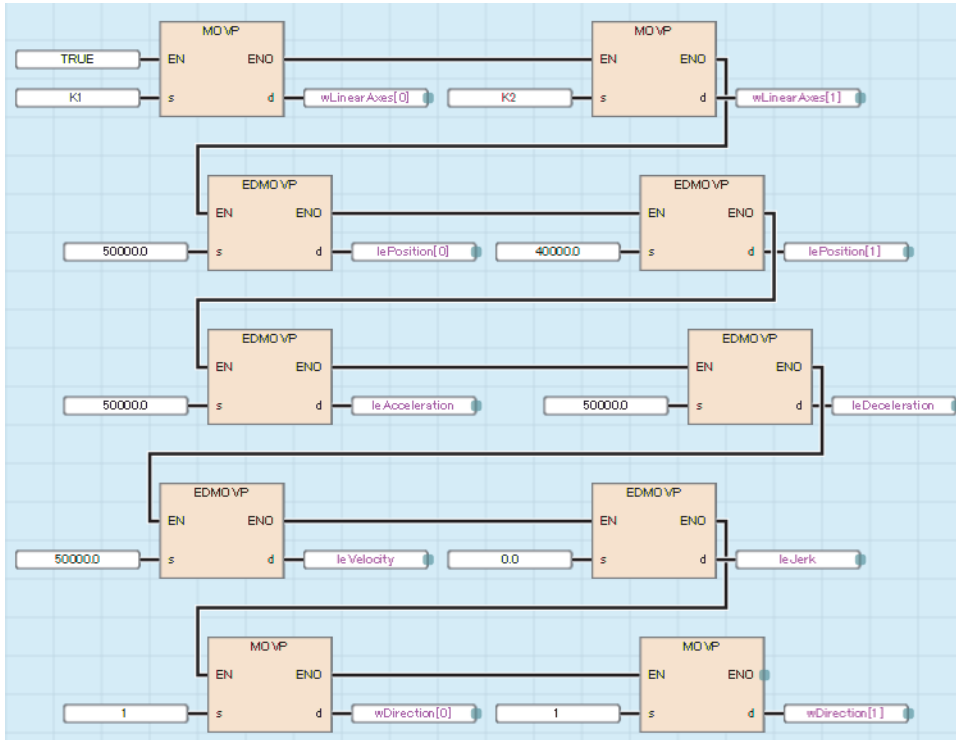
轴组No.	标签名	数据类型	注释
1	AxesGroup001	AXES_GROUP_REF	轴组1

■使用的标签

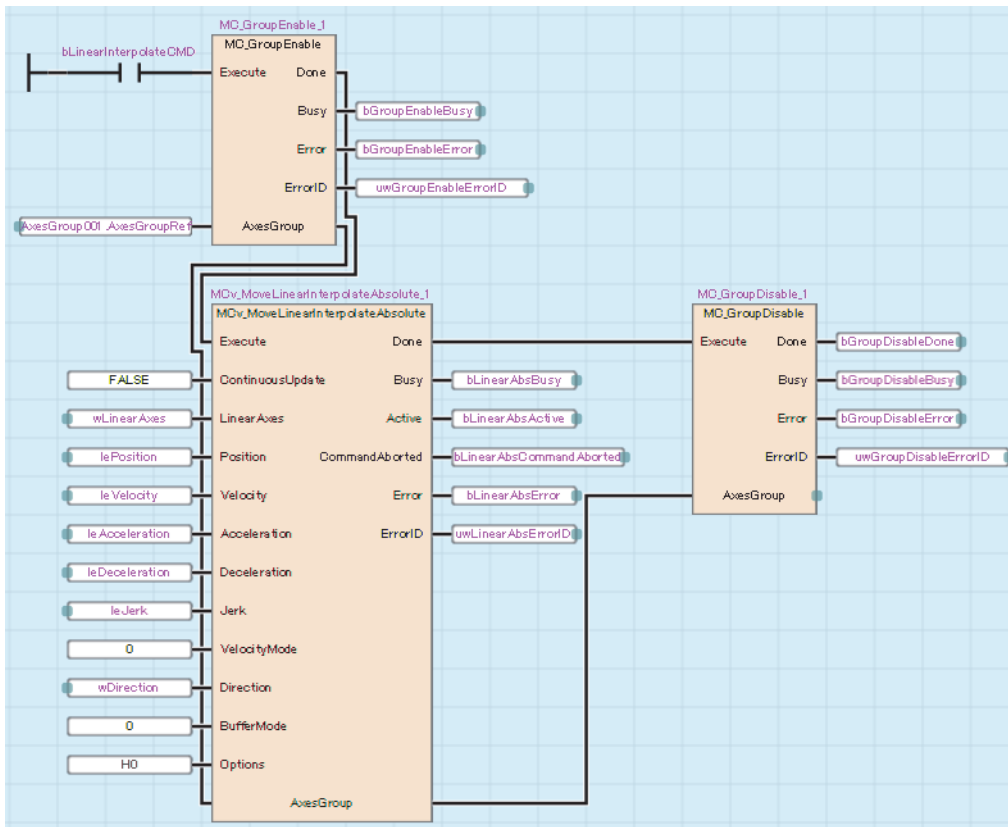
标签名	数据类型	注释
MCv_MoveLinearInterpolateAbsolute_1	MCv_MoveLinearInterpolateAbsolute	绝对值直线插补控制FB
wLinearAxes	字[有符号](0..15)	直线插补轴
lePosition	双精度实数(0..15)	目标位置
leVelocity	双精度实数	速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
wDirection	字[有符号](0..15)	方向选择
bLinearAbsDone	位	执行完成
bLinearAbsBusy	位	执行中
bLinearAbsActive	位	控制中
bLinearAbsCommandAborted	位	执行中断
bLinearAbsError	位	错误
uwLinearAbsErrorID	字[无符号]/位串[16位]	错误代码
MC_GroupEnable_1	MC_GroupEnable	轴组有效FB
bLinearInterpolateCMD	位	直线插补控制启动
bGroupEnableDone	位	轴组有效完成
bGroupEnableBusy	位	执行中
bGroupEnableError	位	错误
uwGroupEnableErrorID	字[无符号]/位串[16位]	错误代码
MC_GroupDisable_1	MC_GroupDisable	轴组无效FB
bGroupDisableDone	位	轴组无效完成
bGroupDisableBusy	位	执行中
bGroupDisableError	位	错误
uwGroupDisableErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

• 2轴直线插补控制用数据的设置



• 轴组有效/2轴直线插补控制/轴组无效



■ST程序

```
//-----2轴直线插补控制用数据的设置-----  
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;  
  
//-----轴组有效-----  
MC_GroupEnable_1(  
    AxesGroup:= AxesGroup001.AxesGroupRef ,  
    Execute:= bLinearInterpolateCMD ,  
    Done=> bGroupEnableDone ,  
    Busy=> bGroupEnableBusy ,  
    Error=> bGroupEnableError ,  
    ErrorID=> uwGroupEnableErrorID  
);  
  
//-----2轴直线插补控制-----  
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  
);  
  
//-----轴组无效-----  
MC_GroupDisable_1(  
    AxesGroup:= AxesGroup001.AxesGroupRef ,  
    Execute:= bLinearAbsDone ,  
    Done=> bGroupDisableDone ,  
    Busy=> bGroupDisableBusy ,
```

```
Error=> bGroupDisableError ,  
ErrorID=> uwGroupDisableErrorID  
);
```

46.11 相对值直线插补控制

MCv_MoveLinearInterpolateRelative

对设置的轴组的相对位置的移动量进行设置，并通过直线插补控制执行定位。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
AxesGroup	轴组信息	AXES_GROUP_REF	启动时	—	不能省略	设置轴组。 关于使用的变量 (<u>AxesGroupName</u> . AxesGroupRef.)，请参阅下述内容。 ☞ 1408页 <u>AxesGroupName</u> . AxesGroupRef. (轴组信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	启动	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行 MCv_MoveLinearInterpolateRelative(相对值直线插补控制)。
ContinuousUpdate	连续更新	BOOL	启动时	TRUE、FALSE	FALSE	设置将速度(Velocity)、加速度(Acceleration)、减速度(Deceleration)的连续更改是置为有效，还是置为无效。 通过连续更新，对执行中的实例在不中断动作的状况下进行输入变量的重新获取。 • FALSE: 无效 • TRUE: 有效
LinearAxes	直线插补轴	INT[0..15]	启动时	1~16	0	从配置轴中指定直线插补控制中使用的轴。 以数组指定配置轴的索引编号(1~16)。 * 在速度模式(VelocityMode)中设置了“2: 基准轴速度(ReferenceAxisSpeed)”的情况下，将数组的第1元素视为基准轴。 关于详细内容，请参阅下述章节。 ☞ 1673页 直线插补轴(LinearAxes)
Distance	移动量	LREAL[0..15]	启动时	-10000000000.0~10000000000.0	0.0	设置从启动时的当前位置到终点为止的移动量。 是1维的数组数据。作为配置轴1~16的相对位置处理。 关于详细内容，请参阅下述章节。 ☞ 1673页 移动量(Distance)

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Velocity	速度	LREAL	启动时/可重启/可连续更新	0.0、0.0001~250000000.0	0.0	设置速度指令值。 关于详细内容，请参阅下述章节。 ☞ 1674页 速度(Velocity)
Acceleration	加速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置加速度。 关于详细内容，请参阅下述章节。 ☞ 1674页 加速度(Acceleration)
Deceleration	减速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置减速度。 关于详细内容，请参阅下述章节。 ☞ 1674页 减速度(Deceleration)
Jerk	Jerk	LREAL	启动时	0.0000、0.0001~2147483647.0	0.0	设置Jerk。 关于详细内容，请参阅下述章节。 ☞ 1674页 Jerk
VelocityMode	速度模式	INT (MC_INTERPOLATE_SPEED_MODE)	启动时	0~2	0	设置插补控制的速度模式。 • 0: 合成速度(VectorSpeed) • 1: 长轴速度(LongAxisSpeed) • 2: 基准轴速度(ReferenceAxisSpeed) 关于详细内容，请参阅下述章节。 ☞ 1675页 速度模式(VelocityMode)
BufferMode	缓冲模式	INT (MC_BUFFER_MODE)	启动时	0~5	0	设置缓冲模式。 • 0: Aborting(mcAborting) • 1: Buffered(mcBuffered) • 2: BlendingLow(mcBlendingLow) • 3: BlendingPrevious(mcBlendingPrevious) • 4: BlendingNext(mcBlendingNext) • 5: BlendingHigh(mcBlendingHigh) 关于详细内容，请参阅下述章节。 ☞ 1677页 缓冲模式(BufferMode)
Options	选项	DWORD(HEX)	启动时	00000000H~00000005H	00000000H	将MCv_MoveLinearInterpolateRelative(相对值直线插补控制)的功能选项以位指定进行设置。 关于详细内容，请参阅下述章节。 ☞ 1678页 选项(Options)

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	完成	BOOL	FALSE	控制完成时，将变为TRUE。 动作完成时根据启动(Execute)的状态将变为如下所示。 ■启动(Execute)为TRUE的情况下 在将启动(Execute)置为FALSE之前将保持为TRUE不变。 ■启动(Execute)为FALSE的情况下 仅1周期为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_MoveLinearInterpolateRelative(相对值直线插补控制)时，将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_MoveLinearInterpolateRelative(相对值直线插补控制)正在控制轴时，将变为TRUE。对同一轴组执行了多个MCv_MoveLinearInterpolateRelative(相对值直线插补控制)的情况下，控制中(Active)变为TRUE的只有1个MCv_MoveLinearInterpolateRelative(相对值直线插补控制)。
CommandAborted	执行中断	BOOL	FALSE	由于错误及多重启动等，MCv_MoveLinearInterpolateRelative(相对值直线插补控制)的执行中断时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 ☞ 所使用的控制器的用户手册

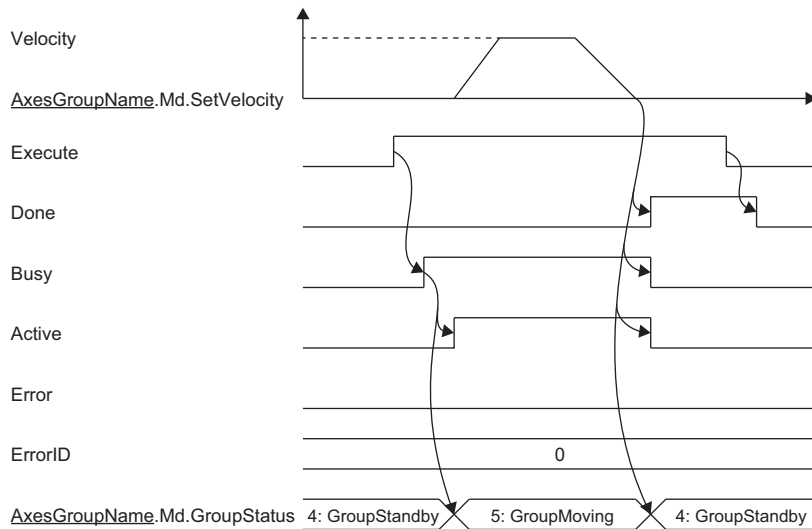
功能

- 直线插补控制指定轴组，进行插补控制，使从始点(移动开始点)到终点的轨迹为直线。在直线插补控制中，进行最大使用4轴的插补控制。
- 在MCv_MoveLinearInterpolateRelative(相对值直线插补控制)中，指定从当前位置开始的相对移动量进行直线插补控制。
- 关于1轴~3轴的相对值直线插补控制的动作，请参阅下述手册的“直线插补控制”。

📖 所使用的控制器的用户手册

■ 时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

📖 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■直线插补轴 (LinearAxes)

在直线插补控制中，从轴组中设置的配置轴中，使用任意的轴进行直线插补。

从轴组的配置轴中，通过直线插补轴 (LinearAxes) 设置进行直线插补的配置轴。直线插补轴 (LinearAxes) 具有16个数组元素。在数组中，设置插补控制中使用的配置轴的索引编号 (1~16: 配置轴1~16) 仅以插补控制中使用的配置轴向前填充对齐的方式进行设置，并在剩余的数组中设置“0”。也可以省略设置了“0”的数组。

- 设置时应使直线插补轴的设置数不大于直线插补控制的最大插补轴数的4轴。
- 设置时应使插补控制中使用的配置轴数不大于配置轴中登录的轴数。
- 速度模式 (VelocityMode) 为“2: 基准轴速度 (ReferenceAxisSpeed)”的情况下，直线插补轴 (LinearAxes) 的第1元素的配置轴将成为基准轴。
- 进行了下述设置的情况下，将变为直线插补轴设置不正确 (错误代码: 1B08H) 且不启动。
 - 设置了轴未设置的配置轴的情况下
 - 直线插补轴 (LinearAxes) 的第1元素为“0”的情况下
 - 直线插补轴 (LinearAxes) 中重复设置了相同的索引编号的情况下
 - 直线插补轴 (LinearAxes) 的设置轴数超出了最大插补轴数的情况下

例

将配置轴2、3、4设置为直线插补轴的情况下

```
LinearAxes[0]:= 2;
LinearAxes[1]:= 3;
LinearAxes[2]:= 4;
LinearAxes[3]:= 0;*1
:
LinearAxes[15]:= 0;*1
```

*1 可以省略“LinearAxes[3]:= 0;~LinearAxes[15]:= 0;”。

■移动量 (Distance)

设置从启动时的当前位置到终点为止的移动量。移动量 (Distance) 具有16个数组元素。

- 移动量 (Distance) 的输入范围为-10000000000.0~10000000000.0。
- Distance[0..15]表示配置轴1~16的移动量。对直线插补轴 (LinearAxes) 中设置的配置轴设置位置。
- 未设置为直线插补轴的配置轴的移动量 (Distance) 将被忽略。
- 设置为直线插补轴的所有配置轴后面的移动量 (Distance) 可以省略。

例

将配置轴2、3、4设置为直线插补轴，并在配置轴2、3、4的目标位置中设置2000.0、3000.0、4000.0的情况下

```
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 可以省略“LinearAxes[3]:= 0;~LinearAxes[15]:= 0;”和“Distance[4]:= 0.0;~Distance[15]:= 0.0;”。

■速度 (Velocity)

设置MCv_MoveLinearInterpolateRelative(相对值直线插补控制)中的路径的最大速度。

设置范围
0.0、0.0001~2500000000.0*1*2

- *1 由于进行浮点运算，因此指令速度的下限值中将产生下述限制。
对指令速度进行了运算周期换算的速度小于“0.00001”的情况下，将变为超出运算周期换算速度范围(错误代码：1B16H)(速度更改时为超出运算周期换算速度范围警告(事件代码：00D2FH))。为了提高浮点运算的精度，应通过更改位置指令单位(AxisName.Pr.Unit_Position)或速度指令单位(AxisName.Pr.Unit_Velocity)设置运算周期换算后的速度不低于“0.00001”。
- *2 在进行多重启动的运动控制FB中指定速度为“0.0”的情况下，将变为之前的运动控制FB的指定速度。

■加速度 (Acceleration)

设置MCv_MoveLinearInterpolateRelative(相对值直线插补控制)中的加速度。

根据加减速方式设置(选项(Options)：位0~2)的设置，设置范围有所不同。

加减速方式设置(选项(Options)：位0~2)	设置范围
0：加减速速度指定方式(mcAccDec)	0.0000*1、0.0001~2147483647.0[U/s ²]的正数
1：加减速时间恒定方式(mcFixedTime)	0.000000*1、0.000001~8400.0[s]的正数

- *1 启动时，根据启动时加减速速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior)，动作将发生变化。加速度更改、减速度更改时，不受理更改。

■减速度 (Deceleration)

设置MCv_MoveLinearInterpolateRelative(相对值直线插补控制)中的减速度。

设置了加减速方式设置(选项(Options)：位0~2)的“0：加减速速度指定方式(mcAccDec)”的情况下，对设置范围进行设置。设置了“1：加减速时间恒定方式(mcFixedTime)”的情况下，不使用减速度(Deceleration)。

加减速方式设置(选项(Options)：位0~2)	设置范围
0：加减速速度指定方式(mcAccDec)	0.0000*1、0.0001~2147483647.0[U/s ²]的正数
1：加减速时间恒定方式(mcFixedTime)	不使用

- *1 启动时，根据启动时加减速速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior)，动作将发生变化。加速度更改、减速度更改时，不受理更改。

■Jerk

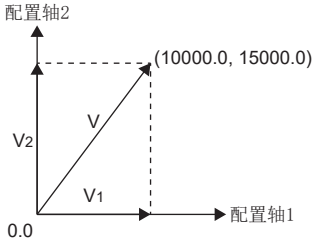
设置MCv_MoveLinearInterpolateRelative(相对值直线插补控制)中的Jerk。

设置了加减速方式设置(选项(Options)：位0~2)的“0：加减速速度指定方式(mcAccDec)”的情况下，对设置范围进行设置。设置了“1：加减速时间恒定方式(mcFixedTime)”的情况下，不使用Jerk。

加减速方式设置(选项(Options)：位0~2)	设置范围
0：加减速速度指定方式(mcAccDec)	0.0000、0.0001~2147483647.0[U/s ²]的正数
1：加减速时间恒定方式(mcFixedTime)	不使用

■速度模式 (VelocityMode)

设置直线插补控制中的速度模式。在速度模式 (VelocityMode) 中设置了“2: 基准轴速度 (ReferenceAxisSpeed)”的情况下, 基准轴将变为直线插补轴 (LinearAxes) 的第1元素中设置的配置轴。

设定值	内容
0: 合成速度 (VectorSpeed)	<p>速度 (Velocity) 设置为合成速度。 各轴的定位速度 (V_n) 是根据设置的控制对象的定位速度 (V), 由运动系统根据各轴的移动量 (D_n) 计算的。 <例> 2轴的直线插补控制的情况下</p>  <ul style="list-style-type: none"> 配置轴1的移动量 (D_1): 10000.0 [pulse] 配置轴2的移动量 (D_2): 15000.0 [pulse] 轴组的速度单位: [s] 合成速度 (V): 7000.0 [pulse/s] <p>上述的情况下, 运动系统通过下述计算公式计算各轴的定位速度。</p> <ul style="list-style-type: none"> 配置轴1的定位速度: $V_1 = V \times D_1 / \sqrt{D_1^2 + D_2^2}$ 配置轴2的定位速度: $V_2 = V \times D_2 / \sqrt{D_1^2 + D_2^2}$ <p>■要点</p> <ul style="list-style-type: none"> 合成速度的情况下, 设置的速度限制值将对作为合成速度的速度 (Velocity) 有效。
1: 长轴速度 (LongAxisSpeed)	<p>速度 (Velocity) 设置为长轴的速度。 各插补轴中设置的指定位置中, 根据移动量最大的插补轴的定位速度 (长轴速度: V) 进行控制。其他插补轴的定位速度 (V_n) 是由运动系统根据各插补轴的移动量 (D_n) 计算的。 <例> 4轴的直线插补控制的情况下</p> <ul style="list-style-type: none"> 配置轴1的移动量 (D_1): 10000.0 [pulse] 配置轴2的移动量 (D_2): 15000.0 [pulse] 配置轴3的移动量 (D_3): 5000.0 [pulse] 配置轴4的移动量 (D_4): 20000.0 [pulse] 配置轴4的速度单位: [s] 长轴速度 (V): 7000.0 [pulse/s] <p>上述的情况下, 长轴将为移动量最大的配置轴4, 并以长轴速度控制配置轴4。对于其它配置轴的定位速度, 运动系统通过下述计算公式计算。</p> <ul style="list-style-type: none"> 配置轴1的定位速度: $V_1 = D_1 / D_4 \times V$ 配置轴2的定位速度: $V_2 = D_2 / D_4 \times V$ 配置轴3的定位速度: $V_3 = D_3 / D_4 \times V$ <p>■要点</p> <ul style="list-style-type: none"> 长轴速度的情况下, 设置的速度限制值对作为长轴速度的速度 (Velocity) 有效。 应注意长轴速度指定时的合成速度有可能大于速度限制值。2轴的直线插补中设置了下述值的情况下, 合成速度超出速度限制值。 <p><例> 设置项目与设定值为下述的情况下</p> <ul style="list-style-type: none"> 直线插补轴: 配置轴1、配置轴2 配置轴1的移动量: 100 [pulse] 配置轴2的移动量: 200 [pulse] 长轴速度: 50 [pulse/s] 配置轴2的速度限制值: 55 [pulse/s] <p>上述的情况下, 基准轴为移动量最大的配置轴2, 并以配置轴2中设置的速度限制值进行控制。此外, 各轴的定位速度及合成速度如下所示。</p> <ul style="list-style-type: none"> 配置轴1的定位速度: $100 / 200 \times 50 = 25$ [pulse/s] 配置轴2的定位速度: 50 [pulse/s] 合成速度: $\sqrt{25^2 + 50^2} = 55.9$ [pulse/s] <p>合成速度为超过配置轴2的速度限制值“55”的值。</p>

设定值	内容
2: 基准轴速度(ReferenceAxisSpeed)	<p>速度(Velocity)设置为基准轴的速度。</p> <p>根据设置的基准轴的定位速度(基准轴速度: V), 运动系统通过各插补轴的移动量(Dn)计算其它插补轴的定位速度(Vn)并进行控制。</p> <p>设置为基准轴的配置轴移动量为“0.0”的情况下, 将变为基准轴移动量0(错误代码: 1AFAH)。</p> <p><例></p> <p>4轴的直线插补控制的情况下</p> <ul style="list-style-type: none"> 配置轴1的移动量(D₁): 10000.0[pulse] 配置轴2的移动量(D₂): 15000.0[pulse] 配置轴3的移动量(D₃): 5000.0[pulse] 配置轴4的移动量(D₄): 20000.0[pulse] 基准轴速度(V): 7000.0[pulse/s] <p>上述的情况下, 基准轴将为配置轴4, 并以配置轴4中设置的定位速度进行控制。</p> <p>对于其它轴的定位速度, 运动系统通过下述计算公式计算。</p> <ul style="list-style-type: none"> 配置轴1的定位速度: $V_1 = D_1 / D_4 \times V$ 配置轴2的定位速度: $V_2 = D_2 / D_4 \times V$ 配置轴3的定位速度: $V_3 = D_3 / D_4 \times V$ <p>■要点</p> <ul style="list-style-type: none"> 基准轴速度的情况下, 设置的速度限制值对作为基准轴速度的速度(Velocity)有效。 应注意比基准轴移动量大的轴的定位速度, 将大于设置的基准轴速度。2轴的直线插补中设置了下述值的情况下, 配置轴2的定位速度与合成速度超出速度限制值。 <p><例></p> <p>设置项目与设定值为下述的情况下</p> <ul style="list-style-type: none"> 直线插补轴: 配置轴1、配置轴2 配置轴1的移动量: 100[pulse] 配置轴2的移动量: 200[pulse] 基准轴速度: 50[pulse/s] 配置轴1的速度限制值: 55[pulse/s] <p>上述的情况下, 基准轴为配置轴1, 并以配置轴1中设置的速度限制值进行控制。此外, 各插补轴的定位速度及合成速度如下所示。</p> <ul style="list-style-type: none"> 配置轴1的定位速度: 50[pulse/s] 配置轴2的定位速度: $200 / 100 \times 50 = 100$[pulse/s] 合成速度: $\sqrt{50^2 + 100^2} = 111.8$[pulse/s] <p>配置轴2的定位速度与合成速度为超过配置轴1的速度限制值“55”的值。</p>

限制事项

- 在速度模式(VelocityMode)中设置了“0: 合成速度(VectorSpeed)”时, 如果各轴的移动量超过“4, 294, 967, 296.0(=2³²)”, 将发生线性移动量超出范围(错误代码: 1B10H)且不启动。
- 将缓冲模式(BufferMode)设置为“0: Aborting(mcAborting)”, 对MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)进行多重启动的情况下, 速度模式(VelocityMode)中应设置“0: 合成速度(VectorSpeed)”。设置了“0: 合成速度(VectorSpeed)”以外的情况下, 将变为超出速度模式范围(错误代码: 1AC9H)。

■缓冲模式(BufferMode)

设置用于进行多重启动(缓冲模式)的动作。


MCv_MoveLinearInterpolateRelative(相对值直线插补控制)中可设置的缓冲模式如下所示。

设定值	内容
0: Aborting(mcAborting)	中断(取消)控制中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在控制中的FB中缓冲下一个FB。 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB完成时,依次执行缓冲FB。
2: BlendingLow(mcBlendingLow)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将控制中的FB与缓冲FB的目标速度中较低一方的速度作为切换速度。
3: BlendingPrevious(mcBlendingPrevious)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将控制中的FB的目标速度作为切换速度。
4: BlendingNext(mcBlendingNext)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将缓冲FB的目标速度作为切换速度。
5: BlendingHigh(mcBlendingHigh)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将控制中的FB与缓冲FB的目标速度中较高一方的速度作为切换速度。

*1 在控制中的FB与缓冲FB之间不执行停止。

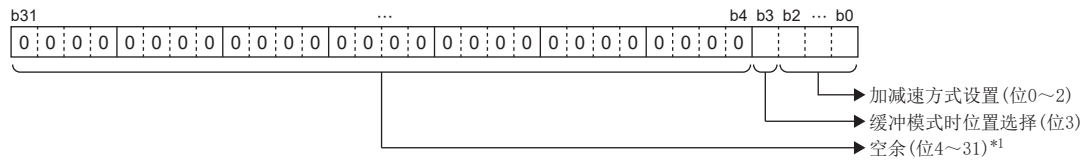
要点

关于多重启动(缓冲模式)的详细内容,请参阅下述手册。

所使用的控制器的用户手册

■选项(Options)

将MCv_MoveLinearInterpolateRelative(相对值直线插补控制)中使用的功能选项以位指定进行设置。
以位指定设置的内容如下所示。



*1 空余中应设置“0”。设置了“0”以外的情况下，将变为超出Options范围(错误代码: 1ABBH)。

位	名称	内容
0~2	加减速方式设置	设置用于进行控制的加减速方式。 • 0: 加减速速度指定方式(mcAccDec) • 1: 加减速时间恒定方式(mcFixedTime)
3	缓冲模式时位置选择	设置对相对值定位控制进行多重启动时的位置。 • 0: 指令当前位置 • 1: 反馈位置 * 缓冲模式(BufferMode)的“0: Aborting(mcAborting)”设置时将有效。

• 加减速方式设置(位0~2)

设定值	内容
0: 加减速速度指定方式(mcAccDec)	是使用MCv_MoveLinearInterpolateRelative(相对值直线插补控制)中设置的加速度(Acceleration)、减速度(Deceleration)、Jerk进行加速/减速的方式。
1: 加减速时间恒定方式(mcFixedTime)	是与速度无关, 使用MCv_MoveLinearInterpolateRelative(相对值直线插补控制)中设置的加减速时间进行加速/减速的方式。对于加减速时间, 设置为加速度(Acceleration), 不使用减速度(Deceleration)、Jerk。

• 缓冲模式时位置选择(位3)

设定值	内容
0: 指令当前位置	<p>是从指令当前位置开始的相对位置控制。 <例> 以移动量(Distance)为“5000.0”, 选项(Options)为“00000000H(位3为0: 指令当前位置)”的设置进行了多重启动的情况下</p>
1: 反馈位置	<p>是从反馈位置开始的相对位置控制。 <例> 以移动量(Distance)为“5000.0”, 选项(Options)为“00000008H(位3为1: 反馈位置)”的设置进行了多重启动的情况下</p>

■必要对象数据

使用MCv_MoveLinearInterpolateRelative(相对值直线插补控制)的情况下, 应对指定的轴组的所有配置轴设置下述对象数据。

- Target position(607AH)

存在有未设置对象数据的配置轴的情况下, 将变为必须从站对象未设置(错误代码: 1AF7H)且不启动。

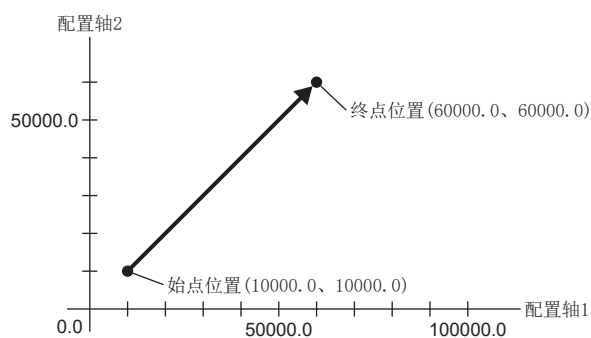
关于对象数据设置的详细内容, 请参阅下述手册。

📖所使用的控制器的用户手册

程序示例

将直线插补控制启动(bLinearInterpolateCMD)置为TRUE, 并将轴组1(AxesGroup001)置为了有效后, 根据下述设置进行轴组1(AxesGroup001)的2轴直线插补的相对值定位控制的程序示例如下所示。

• 动作



• 轴组

项目	设定值
配置轴[1]	Axis0001
配置轴[2]	Axis0002

• 设置

项目	设定值	
	配置轴1	配置轴2
目标位置	50000.0	50000.0
速度	50000.0	
加速度	50000.0	
减速度	50000.0	
Jerk	0.0	

■ 轴

轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1
2	Axis0002	AXIS_REF	轴2

■ 轴组

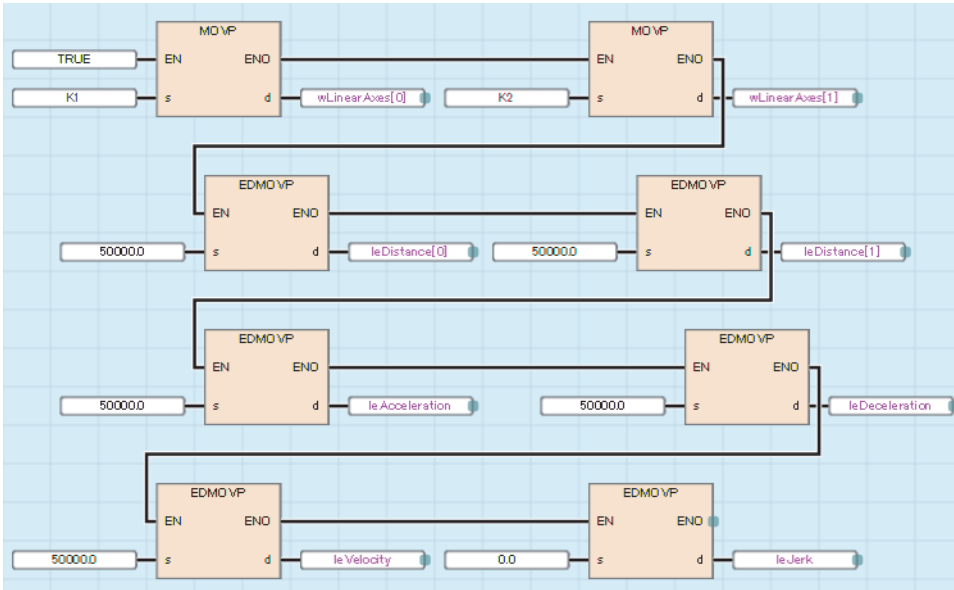
轴组No.	标签名	数据类型	注释
1	AxesGroup001	AXES_GROUP_REF	轴组1

■使用的标签

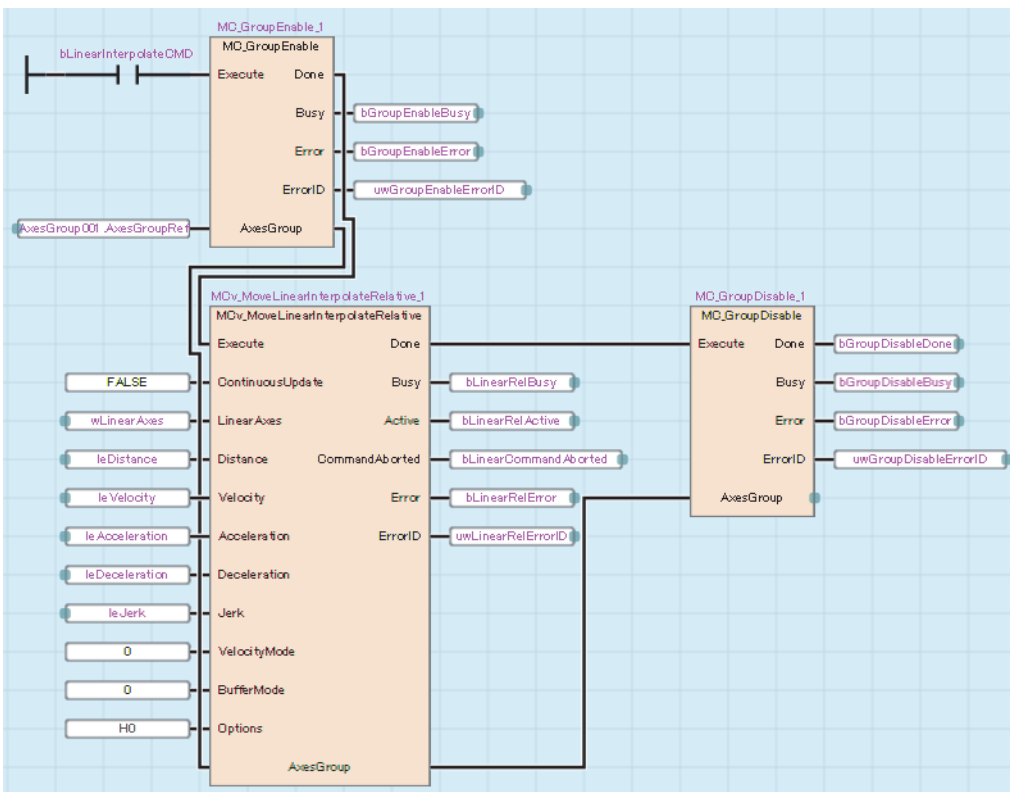
标签名	数据类型	注释
MCv_MoveLinearInterpolateRelative_1	MCv_MoveLinearInterpolateRelative	相对值直线插补控制FB
wLinearAxes	字[有符号](0..15)	直线插补轴
leDistance	双精度实数(0..15)	移动量
leVelocity	双精度实数	速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bLinearRelDone	位	执行完成
bLinearRelBusy	位	执行中
bLinearRelActive	位	控制中
bLinearCommandAborted	位	执行中断
bLinearRelError	位	错误
uwLinearRelErrorID	字[无符号]/位串[16位]	错误代码
MC_GroupEnable_1	MC_GroupEnable	轴组有效FB
bLinearInterpolateCMD	位	插补控制控制启动
bGroupEnableDone	位	轴组有效完成
bGroupEnableBusy	位	执行中
bGroupEnableError	位	错误
uwGroupEnableErrorID	字[无符号]/位串[16位]	错误代码
MC_GroupDisable_1	MC_GroupDisable	轴组无效FB
bGroupDisableDone	位	轴组无效完成
bGroupDisableBusy	位	执行中
bGroupDisableError	位	错误
uwGroupDisableErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

- 2轴直线插补控制用数据的设置



- 轴组有效/2轴直线插补控制/轴组无效



■ST程序

```
//-----2轴直线插补控制用数据的设置-----  
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;  
  
//-----轴组有效-----  
MC_GroupEnable_1(  
    AxesGroup:= AxesGroup001.AxesGroupRef ,  
    Execute:= bLinearInterpolateCMD ,  
    Done=> bGroupEnableDone ,  
    Busy=> bGroupEnableBusy ,  
    Error=> bGroupEnableError ,  
    ErrorID=> uwGroupEnableErrorID  
);  
  
//-----2轴直线插补控制-----  
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  
);  
  
//-----轴组无效-----  
MC_GroupDisable_1(  
    AxesGroup:= AxesGroup001.AxesGroupRef ,  
    Execute:= bLinearRelDone ,  
    Done=> bGroupDisableDone ,  
    Busy=> bGroupDisableBusy ,  
    Error=> bGroupDisableError ,  
    ErrorID=> uwGroupDisableErrorID  
);
```

46.12 绝对值圆弧插补控制

MCv_MoveCircularInterpolateAbsolute

使用设置的轴组的配置轴，设置绝对位置的终点及辅助点，通过2轴的圆弧插补执行定位。

梯形图	FBD/LD	ST
		<pre> 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?); </pre>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
AxesGroup	轴组信息	AXES_GROUP_REF	启动时	—	不能省略	设置轴组。 关于使用的变量 (<u>AxesGroupName</u> . <u>AxesGroupRef</u> .), 请参阅下述内容。 📖 1408页 <u>AxesGroupName</u> . <u>AxesGroupRef</u> . (轴组信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	启动	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行 MCv_MoveCircularInterpolateAbsolute(绝对值圆弧插补控制)。
ContinuousUpdate	连续更新	BOOL	启动时	TRUE、FALSE	FALSE	设置将速度(Velocity)、加速度(Acceleration)、减速度(Deceleration)的连续更改是置为有效,还是置为无效。 通过连续更新,对执行中的实例在不中断动作的状况下进行输入变量的重新获取。 • FALSE: 无效 • TRUE: 有效
CircAxes	圆弧插补轴	INT[0..1]	启动时	1~16	0	从配置轴中设置圆弧插补控制中使用的轴。 以数组设置配置轴的索引编号(1~16)。数组的第1元素将变为基准轴。 关于详细内容,请参阅下述章节。 ☞ 1686页 圆弧插补轴(CircAxes)
CircMode	圆弧插补模式	INT(MC_CIRC_MODE)	启动时	0~2	0	设置圆弧插补的指定方法。 • 0: 边界点指定(mcBorder) • 1: 中心点指定(mcCenter) • 2: 半径指定(mcRadius) 关于详细内容,请参阅下述章节。 ☞ 1687页 圆弧插补模式(CircMode)
AuxPoint	辅助点	LREAL[0..15]	启动时	■ “0: 边界点指定(mcBorder)”、“1: 中心点指定(mcCenter)”的情况下: -10000000000.0~10000000000.0 ■ “2: 半径指定(mcRadius)”的情况下: 0.000001~2147483647.0	0.0	根据轴的单位设置辅助点(边界点、中心点、半径)的绝对位置。 是1维的数组数据。根据圆弧插补模式(CircMode),将变为如下所示。 ■ “0: 边界点指定(mcBorder)”、“1: 中心点指定(mcCenter)”的情况下: 作为配置轴1~16的绝对位置处理。设置边界点、中心点的绝对位置。 ■ “2: 半径指定(mcRadius)”的情况下: 将第1元素作为半径处理,并忽略第2元素及以后。 关于详细内容,请参阅下述章节。 ☞ 1687页 辅助点(AuxPoint)
EndPoint	终点	LREAL[0..15]	启动时	-10000000000.0~10000000000.0	0.0	设置终点的位置。 是1维的数组数据。作为配置轴1~16的绝对位置处理。 关于详细内容,请参阅下述章节。 ☞ 1687页 终点(EndPoint)
PathChoice	路径选择	INT(MC_CIRC_PATHCHOICE)	启动时	0~5	0	设置圆弧插补的旋转方向。 • 0: CW(mcCW) • 1: CCW(mcCCW) • 2: 就近(mcShortWay) • 3: 就远(mcLongWay) • 4: CW就远(mcCWLongWay) • 5: CCW就远(mcCCWLongWay) * 在圆弧插补模式(CircMode)中设置了“0: 边界点指定(mcBorder)”的情况下,将忽略输入。 * 在圆弧插补模式(CircMode)中设置了“2: 半径指定(mcRadius)”的情况下,“0: CW(mcCW)”表示CW就近,“1: CCW(mcCCW)”表示CCW就近的含义。 关于详细内容,请参阅下述章节。 ☞ 1688页 路径选择(PathChoice)
Velocity	速度	LREAL	启动时/可重启/可连续更新	0.0、0.0001~2500000000.0	0.0	设置速度指令值。 关于详细内容,请参阅下述章节。 ☞ 1690页 速度(Velocity)
Acceleration	加速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置加速度。 关于详细内容,请参阅下述章节。 ☞ 1690页 加速度(Acceleration)
Deceleration	减速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置减速度。 关于详细内容,请参阅下述章节。 ☞ 1690页 减速度(Deceleration)

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Jerk	Jerk	LREAL	启动时	0.0000、0.0001~2147483647.0	0.0	设置Jerk。 关于详细内容，请参阅下述章节。 ☞ 1690页 Jerk
CircularError Tolerance	圆弧插补误差允许值	LREAL	启动时	0.000001~100000.0	100.0	设置圆弧插补误差的允许范围。 关于详细内容，请参阅下述章节。 ☞ 1690页 圆弧插补误差允许值 (CircularErrorTolerance)
BufferMode	缓冲模式	INT (MC_BUFFER_MODE)	启动时	0~5	0	设置缓冲模式。 <ul style="list-style-type: none"> • 0: Aborting (mcAborting) • 1: Buffered (mcBuffered) • 2: BlendingLow (mcBlendingLow) • 3: BlendingPrevious (mcBlendingPrevious) • 4: BlendingNext (mcBlendingNext) • 5: BlendingHigh (mcBlendingHigh) 关于详细内容，请参阅下述章节。 ☞ 1691页 缓冲模式 (BufferMode)
Options	选项	DWORD (HEX)	启动时	00000000H~00000001H	00000000H	将MCv_MoveCircularInterpolateAbsolute (绝对值圆弧插补控制) 的功能选项以位指定进行设置。 关于详细内容，请参阅下述章节。 ☞ 1691页 选项 (Options)

■输出变量

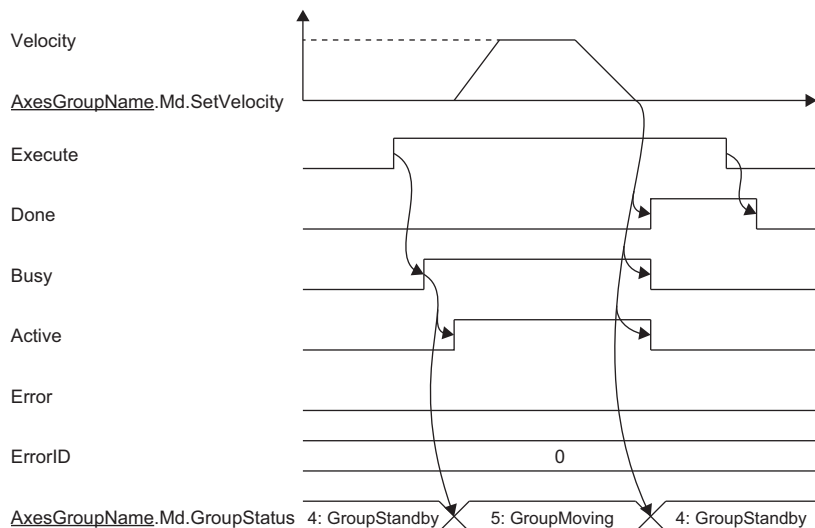
输出变量	名称	数据类型	初始值	内容
Done	完成	BOOL	FALSE	控制完成时，将变为TRUE。 动作完成时根据启动 (Execute) 的状态将变为如下所示。 <ul style="list-style-type: none"> ■启动 (Execute) 为TRUE的情况下 在将启动 (Execute) 置为FALSE之前将保持为TRUE不变。 ■启动 (Execute) 为FALSE的情况下 仅1周期为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_MoveCircularInterpolateAbsolute (绝对值圆弧插补控制) 时，将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_MoveCircularInterpolateAbsolute (绝对值圆弧插补控制) 正在控制轴时，将变为TRUE。 对同一轴组执行了多个MCv_MoveCircularInterpolateAbsolute (绝对值圆弧插补控制) 的情况下，控制中 (Active) 变为TRUE的只有1个MCv_MoveCircularInterpolateAbsolute (绝对值圆弧插补控制)。
CommandAborted	执行中断	BOOL	FALSE	由于错误及多重启动等，MCv_MoveCircularInterpolateAbsolute (绝对值圆弧插补控制) 的执行中断时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 ☞ 所使用的控制器的用户手册

功能

- 圆弧插补控制指定轴组，对机械直交配置的直线轴进行插补控制，以确保从起点(移动开始点)到终点的轨迹为圆弧。
- 在圆弧插补控制中，可以从轴组中设置的配置轴中使用任意的2轴进行插补控制。
- 在圆弧插补控制中，可以设置边界点指定、中心点指定、半径指定的圆弧插补方式。
- 在MCv_MoveCircularInterpolateAbsolute(绝对值圆弧插补控制)中，指定绝对位置的终点及辅助点进行2轴的圆弧插补控制。

■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■圆弧插补轴(CircAxes)

在圆弧插补控制中，从轴组中设置的配置轴中，使用任意的2轴进行圆弧插补。

从轴组的配置轴中，通过圆弧插补轴(CircAxes)设置进行圆弧插补的轴。圆弧插补轴(CircAxes)具有2个数组元素。在数组中，设置插补控制中使用的配置轴的索引编号(1~16: 配置轴1~16)。

数组的第1元素将变为基准轴。

指定了未设置轴的配置轴的情况下，将变为圆弧插补轴未设置(错误代码: 1ACAH)。

例

将配置轴2、3设置为圆弧插补轴的情况下

```
CircAxes[0]:= 2;
```

```
CircAxes[1]:= 3;
```

■圆弧插补模式(CircMode)

设置用于进行圆弧插补控制的圆弧插补方式(边界点指定、中心点指定、半径指定)。

根据圆弧插补模式(CircMode)，辅助点(AuxPoint)的设置内容有所不同。

设定值	辅助点(AuxPoint)
0: 边界点指定(mcBorder)	辅助点(AuxPoint)设置连接始点与终点的圆弧上的边界点。 使用圆弧插补轴(CircAxes)中设置的2轴的配置轴，以通过设置的边界点的圆弧的轨迹进行定位控制。
1: 中心点指定(mcCenter)	辅助点(AuxPoint)设置圆弧的中心点。 使用圆弧插补轴(CircAxes)中设置的2轴的配置轴，以将设置的中心点作为中心的圆弧的轨迹进行定位控制。
2: 半径指定(mcRadius)	辅助点(AuxPoint)设置圆弧的半径。 使用圆弧插补轴(CircAxes)中设置的2轴的配置轴，以具有设置的半径的圆弧的轨迹进行定位控制。

关于通过设置的圆弧插补模式(CircMode)进行的圆弧插补控制的动作，请参阅下述手册。

📖所使用的控制器的用户手册

■辅助点(AuxPoint)

设置进行圆弧插补的辅助点(边界点、中心点、半径)的设置。

辅助点(AuxPoint)具有16个数组元素。对于辅助点(AuxPoint)，根据圆弧插补模式(CircMode)设置内容有所不同。

圆弧插补模式(CircMode)	设置范围	辅助点(AuxPoint)的设置
0: 边界点指定(mcBorder)	$-10000000000.0 \leq \text{设定值} < 10000000000.0^{*1}$	辅助点(AuxPoint)设置连接始点与终点的圆弧上的边界点。 AuxPoint[0..15]表示配置轴1~16的辅助点位置。 • 未设置为圆弧插补轴的配置轴的辅助点(AuxPoint)将被忽略。 • 设置为圆弧插补轴的所有配置轴后面的辅助点(AuxPoint)可以省略。
1: 中心点指定(mcCenter)		辅助点(AuxPoint)设置圆弧的中心点。 AuxPoint[0..15]表示配置轴1~16的辅助点位置。 • 未设置为圆弧插补轴的配置轴的辅助点(AuxPoint)将被忽略。 • 设置为圆弧插补轴的所有配置轴后面的辅助点(AuxPoint)可以省略。
2: 半径指定(mcRadius)	0.000001~2147483647.0	辅助点(AuxPoint)设置圆弧的半径。 AuxPoint[0]表示圆弧的半径。 • AuxPoint[1]及以后的输入将被忽略，因此可以省略。

*1 环形计数器有效的情况下，将变为环形计数器范围。

例

将配置轴2、3设置为圆弧插补轴，并在配置轴2、3的辅助点的绝对位置中设置2000.0、3000.0的情况下

```
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 可以省略“AuxPoint[3]:= 0.0;~AuxPoint[15]:= 0.0;”。

■终点(EndPoint)

设置通过绝对位置的终点的位置。

终点(EndPoint)具有16个数组元素。EndPoint[0..15]表示配置轴1~16的终点位置。

- 未设置为圆弧插补轴的配置轴的终点(EndPoint)将被忽略。
- 设置为圆弧插补轴的所有配置轴后面的终点(EndPoint)可以省略。

终点(EndPoint)应在下述范围内进行设置。

设置范围
$-10000000000.0 \leq \text{设定值} < 10000000000.0^{*1}$

*1 环形计数器有效的情况下，将变为环形计数器范围。

■路径选择 (PathChoice)

设置圆弧插补中的旋转方向。将圆弧插补轴(CircAxes)中设置的数组的第1元素作为基准轴考虑旋转方向。

对于路径选择(PathChoice)，在圆弧插补模式(CircMode)中设置了“1：中心点指定(mcCenter)”或“2：半径指定(mcRadius)”的情况下进行设置。设置了“0：边界点指定(mcBorder)”的情况下，将忽略输入。

通过设置的圆弧插补模式(CircMode)的路径选择(PathChoice)的设定值、设置内容、可控制的圆弧的中心角、路径如下所示。

- “1：中心点指定(mcCenter)”的路径

设定值	设置内容	可控制的圆弧的中心角	路径
0: CW(mcCW)	CW	$0^\circ < \theta \leq 360^\circ$	
1: CCW(mcCCW)	CCW	$0^\circ < \theta \leq 360^\circ$	
2: 就近(mcShortWay)	就近 <ul style="list-style-type: none"> • 始点、中心点、终点位于一直线上的情况下，将变为CW。(向CW方向绘制半圆。) • 始点=终点的情况下，将变为路径选择设置不正确(错误代码：1B06H)。 	$0^\circ < \theta \leq 180^\circ$	
3: 就远(mcLongWay)	就远 <ul style="list-style-type: none"> • 始点、中心点、终点位于一直线上的情况下，将变为CW。(向CW方向绘制半圆。) • 始点=终点的情况下，将变为CW。(向CW方向绘制正圆。) 	$180^\circ \leq \theta \leq 360^\circ$	

• “2: 半径指定 (mcRadius)” 的路径

设定值	设置内容	可控制的圆弧的中心角	路径
0: CW (mcCW)	CW (CW就近) • 始点、中心点、终点位于一直线上的情况下，将变为路径选择设置不正确(错误代码: 1B06H)。	$0^\circ < \theta < 180^\circ$	<p>始点 中心点 终点 半径 $\theta < 180^\circ$ 定位路径</p>
1: CCW (mcCCW)	CCW (CCW就近) • 始点、中心点、终点位于一直线上的情况下，将变为路径选择设置不正确(错误代码: 1B06H)。	$0^\circ < \theta < 180^\circ$	<p>始点 中心点 终点 半径 $\theta < 180^\circ$ 定位路径</p>
4: CW就远 (mcCWLongWay)	CW就远 • 始点、中心点、终点位于一直线上的情况下，圆弧将绘制半圆。	$180^\circ \leq \theta < 360^\circ$	<p>始点 中心点 终点 半径 $180^\circ \leq \theta < 360^\circ$ 定位路径</p>
5: CCW就远 (mcCCWLongWay)	CCW就远 • 始点、中心点、终点位于一直线上的情况下，圆弧将绘制半圆。	$180^\circ \leq \theta < 360^\circ$	<p>始点 中心点 终点 半径 $180^\circ \leq \theta < 360^\circ$ 定位路径</p>

■速度 (Velocity)

设置MCv_MoveCircularInterpolateAbsolute(绝对值圆弧插补控制)中的路径的最大速度。

设置范围
0.0、0.0001~2500000000.0*1*2

- *1 由于进行浮点运算，因此指令速度的下限值中将产生下述限制。
对指令速度进行了运算周期换算的速度小于“0.00001”的情况下，将变为超出运算周期换算速度范围(错误代码：1B16H)(速度更改时为超出运算周期换算速度范围警告(事件代码：00D2FH))。为了提高浮点运算的精度，应通过更改位置指令单位(AxisName.Pr.Unit_Position)或速度指令单位(AxisName.Pr.Unit_Velocity)设置运算周期换算后的速度不低于“0.00001”。
- *2 在进行多重启动的运动控制FB中指定速度为“0.0”的情况下，将变为之前的运动控制FB的指定速度。

■加速度 (Acceleration)

设置MCv_MoveCircularInterpolateAbsolute(绝对值圆弧插补控制)中的加速度。

根据加减速方式设置(选项(Options)：位0~2)的设置，设置范围有所不同。

加减速方式设置(选项(Options)：位0~2)	设置范围
0：加减速速度指定方式(mcAccDec)	0.0000*1、0.0001~2147483647.0[U/s ²]的正数
1：加减速时间恒定方式(mcFixedTime)	0.000000*1、0.000001~8400.0[s]的正数

- *1 启动时，根据启动时加减速速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior)，动作将发生变化。加速度更改、减速度更改时，不受理更改。

■减速度 (Deceleration)

设置MCv_MoveCircularInterpolateAbsolute(绝对值圆弧插补控制)中的减速度。

设置了加减速方式设置(选项(Options)：位0~2)的“0：加减速速度指定方式(mcAccDec)”的情况下，对设置范围进行设置。设置了“1：加减速时间恒定方式(mcFixedTime)”的情况下，不使用减速度(Deceleration)。

加减速方式设置(选项(Options)：位0~2)	设置范围
0：加减速速度指定方式(mcAccDec)	0.0000*1、0.0001~2147483647.0[U/s ²]的正数
1：加减速时间恒定方式(mcFixedTime)	不使用

- *1 启动时，根据启动时加减速速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior)，动作将发生变化。加速度更改、减速度更改时，不受理更改。

■Jerk

设置MCv_MoveCircularInterpolateAbsolute(绝对值圆弧插补控制)中的Jerk。

设置了加减速方式设置(选项(Options)：位0~2)的“0：加减速速度指定方式(mcAccDec)”的情况下，对设置范围进行设置。设置了“1：加减速时间恒定方式(mcFixedTime)”的情况下，不使用Jerk。

加减速方式设置(选项(Options)：位0~2)	设置范围
0：加减速速度指定方式(mcAccDec)	0.0000、0.0001~2147483647.0[U/s ²]的正数
1：加减速时间恒定方式(mcFixedTime)	不使用

■圆弧插补误差允许值 (CircularErrorTolerance)

在中心点指定的圆弧插补控制中，通过始点位置与中心点位置计算出的圆弧的轨迹与终点(EndPoint)中设置的终点位置有可能有偏差。

圆弧插补误差允许值(CircularErrorTolerance)设置计算出的圆弧的轨迹与终点位置的误差的允许范围。

允许范围	内容
计算出的误差 ≤ 圆弧插补误差允许值 (CircularErrorTolerance)	<p>在通过螺旋插补进行误差补偿的同时，向设置的终点地址进行圆弧插补。</p>
计算出的误差 > 圆弧插补误差允许值 (CircularErrorTolerance)	<p>定位启动时将变为圆弧插补误差允许值溢出(错误代码：1AD9H)且不启动。定位控制中的情况下，检测到错误时将立即停止。</p>

■缓冲模式(BufferMode)

设置用于进行多重启动(缓冲模式)的动作。


MCv_MoveCircularInterpolateAbsolute(绝对值圆弧插补控制)中可设置的缓冲模式如下所示。

设定值	内容
0: Aborting(mcAborting)	中断(取消)控制中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在控制中的FB中缓冲下一个FB。 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB完成时,依次执行缓冲FB。
2: BlendingLow(mcBlendingLow)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将控制中的FB与缓冲FB的目标速度中较低一方的速度作为切换速度。
3: BlendingPrevious(mcBlendingPrevious)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将控制中的FB的目标速度作为切换速度。
4: BlendingNext(mcBlendingNext)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将缓冲FB的目标速度作为切换速度。
5: BlendingHigh(mcBlendingHigh)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将控制中的FB与缓冲FB的目标速度中较高一方的速度作为切换速度。

*1 在控制中的FB与缓冲FB之间不执行停止。

要点

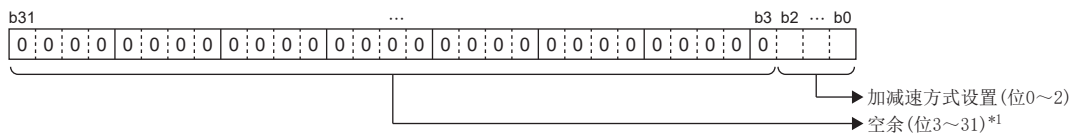
关于多重启动(缓冲模式)的详细内容,请参阅下述手册。

所使用的控制器的用户手册

■选项(Options)

将MCv_MoveCircularInterpolateAbsolute(绝对值圆弧插补控制)中使用的功能选项以位指定进行设置。

以位指定设置的内容如下所示。



*1 空余中应设置“0”。设置了“0”以外的情况下,将变为超出Options范围(错误代码:1ABBH)。

位	名称	内容
0~2	加减速方式设置	设置用于进行控制的加减速方式。 • 0: 加减速速度指定方式(mcAccDec) • 1: 加减速时间恒定方式(mcFixedTime)

• 加减速方式设置(位0~2)

设定值	内容
0: 加减速速度指定方式(mcAccDec)	是使用MCv_MoveCircularInterpolateAbsolute(绝对值圆弧插补控制)中设置的加速度(Acceleration)、减速度(Deceleration)、Jerk进行加速/减速的方式。
1: 加减速时间恒定方式(mcFixedTime)	是与速度无关,使用MCv_MoveCircularInterpolateAbsolute(绝对值圆弧插补控制)中设置的加减速时间进行加速/减速的方式。 对于加减速时间,设置为加速度(Acceleration),不使用减速度(Deceleration)、Jerk。

■必要对象数据

使用MCv_MoveCircularInterpolateAbsolute(绝对值圆弧插补控制)的情况下,应对指定的轴组的所有配置轴设置下述对象数据。

- Target position(607AH)

存在有未设置对象数据的配置轴的情况下,将变为必须从站对象未设置(错误代码:1AF7H)且不启动。

关于对象数据设置的详细内容,请参阅下述手册。

📖所使用的控制器的用户手册

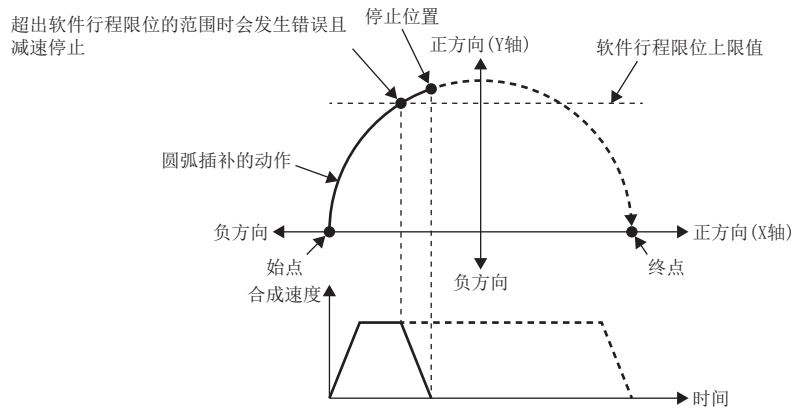
注意事项

- 无法进行包含将行程限位设置为无效的轴的圆弧插补。否则将变为圆弧插补时软件行程限位无效(错误代码:1ADAH)且不启动。
- 插补动作中插补路径超出行程限位范围的情况下,将变为软件行程限位溢出(正方向)(错误代码:1A83H)或软件行程限位溢出(负方向)(错误代码:1A84H)且停止运行。

例

在Y轴的正方向上超出了软件行程限位的上限的情况下立即停止的情况下,在发生错误的同时执行停止。

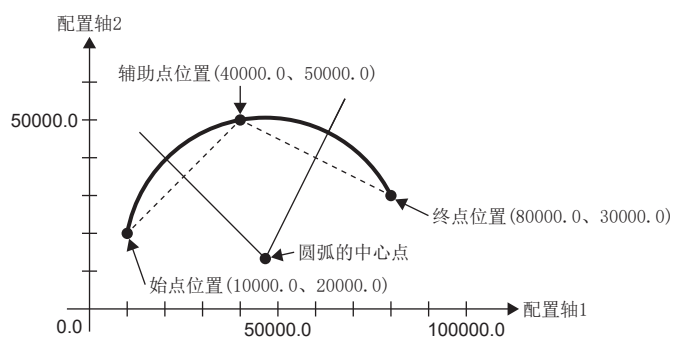
减速停止的情况下,将如下图所示沿着圆弧的轨迹进行减速,并停止。



程序示例

将圆弧插补控制启动(bCircularInterpolateCMD)置为TRUE，并将轴组1(AxesGroup001)置为了有效后，根据下述设置进行轴组1(AxesGroup001)的2轴圆弧插补(辅助点指定)的绝对值定位控制的程序示例如下所示。

• 动作



• 轴组

项目	设定值
配置轴[1]	Axis0001
配置轴[2]	Axis0002

• 设置

项目	设定值	
	配置轴1	配置轴2
辅助点	40000.0	50000.0
终点	80000.0	30000.0
速度	50000.0	
加速度	50000.0	
减速度	50000.0	
Jerk	0.0	
圆弧插补误差允许值	100.0	

■ 轴

轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1
2	Axis0002	AXIS_REF	轴2

■ 轴组

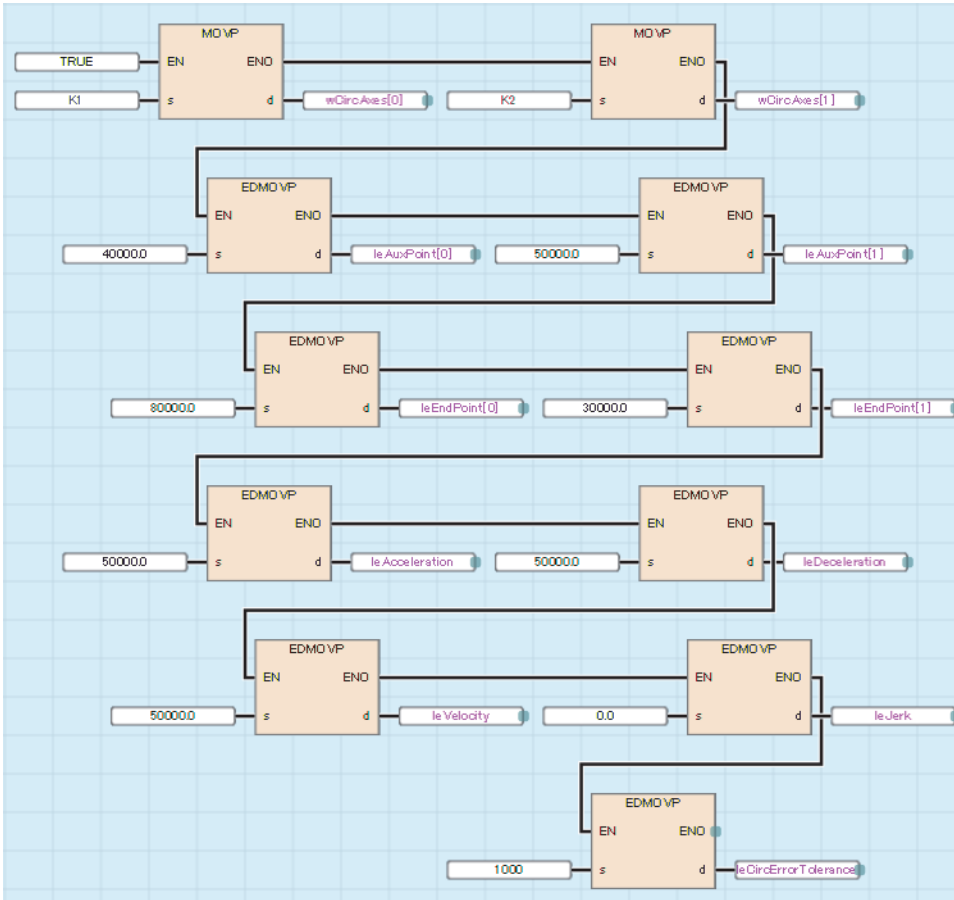
轴组No.	标签名	数据类型	注释
1	AxesGroup001	AXES_GROUP_REF	轴组1

■使用的标签

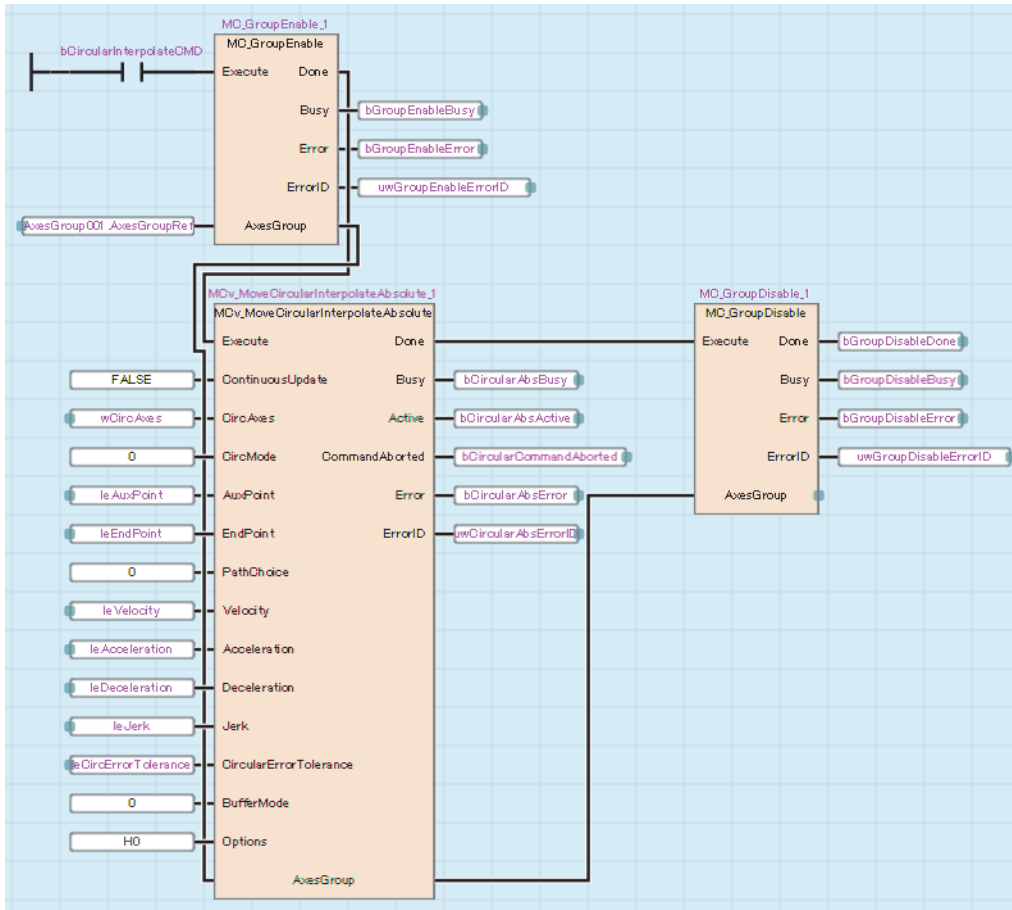
标签名	数据类型	注释
MCv_MoveCircularInterpolateAbsolute_1	MCv_MoveCircularInterpolateAbsolute	绝对值圆弧插补控制FB
wCircAxes	字[有符号](0..1)	圆弧插补轴
leAuxPoint	双精度实数(0..15)	辅助点
leEndPoint	双精度实数(0..15)	终点
leVelocity	双精度实数	速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
leCircErrorTolerance	双精度实数	圆弧插补误差允许值
bCircularAbsDone	位	执行完成
bCircularAbsBusy	位	执行中
bCircularAbsActive	位	控制中
bCircularCommandAborted	位	执行中断
bCircularAbsError	位	错误
uwCircularAbsErrorID	字[无符号]/位串[16位]	错误代码
MC_GroupEnable_1	MC_GroupEnable	轴组有效FB
bCircularInterpolateCMD	位	圆弧插补控制启动
bGroupEnableDone	位	轴组有效完成
bGroupEnableBusy	位	执行中
bGroupEnableError	位	错误
uwGroupEnableErrorID	字[无符号]/位串[16位]	错误代码
MC_GroupDisable_1	MC_GroupDisable	轴组无效FB
bGroupDisableDone	位	轴组无效完成
bGroupDisableBusy	位	执行中
bGroupDisableError	位	错误
uwGroupDisableErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

- 2轴圆弧插补控制用数据的设置



• 轴组有效/圆弧插补控制(边界点指定)/轴组无效



■ST程序

//-----2轴圆弧插补控制用数据的设置-----

```
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;
```

//-----轴组有效-----

```
MC_GroupEnable_1(
  AxesGroup:= AxesGroup001.AxesGroupRef ,
  Execute:= bCircularInterpolateCMD ,
  Done=> bGroupEnableDone ,
  Busy=> bGroupEnableBusy ,
  Error=> bGroupEnableError ,
  ErrorID=> uwGroupEnableErrorID
);
```

//-----圆弧插补控制(边界点指定)-----

```
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
);
```

```
//-----轴组无效-----  
MC_GroupDisable_1(  
    AxesGroup:= AxesGroup001.AxesGroupRef ,  
    Execute:= bCircularAbsDone ,  
    Done=> bGroupDisableDone ,  
    Busy=> bGroupDisableBusy ,  
    Error=> bGroupDisableError ,  
    ErrorID=> uwGroupDisableErrorID  
);
```

46.13 相对值圆弧插补控制

MCv_MoveCircularInterpolateRelative

使用设置的轴组的配置轴，从启动时的当前位置开始向终点及辅助点设置相对位置，通过2轴的圆弧插补执行定位。

梯形图	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>

设置数据

■输入输出变量


输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
AxesGroup	轴组信息	AXES_GROUP_REF	启动时	—	不能省略	设置轴组。 关于使用的变量 (AxesGroupName, AxesGroupRef.)，请参阅下述内容。 ☞ 1408页 AxesGroupName, AxesGroupRef. (轴组信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	启动	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行 MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)。
ContinuousUpdate	连续更新	BOOL	启动时	TRUE、FALSE	FALSE	设置将速度(Velocity)、加速度(Acceleration)、减速度(Deceleration)的连续更改是置为有效，还是置为无效。 通过连续更新，对执行中的实例在不中断动作的状况下进行输入变量的重新获取。 • FALSE: 无效 • TRUE: 有效
CircAxes	圆弧插补轴	INT[0..1]	启动时	1~16	0	从配置轴中设置圆弧插补控制中使用的轴。 以数组设置配置轴的索引编号(1~16)。数组的第1元素将变为基准轴。 关于详细内容，请参阅下述章节。 ☞ 1702页 圆弧插补轴(CircAxes)
CircMode	圆弧插补模式	INT(MC_CIRC_MODE)	启动时	0~2	0	设置圆弧插补的指定方法。 • 0: 边界点指定(mcBorder) • 1: 中心点指定(mcCenter) • 2: 半径指定(mcRadius) 关于详细内容，请参阅下述章节。 ☞ 1703页 圆弧插补模式(CircMode)

输入变量	名称	数据类型	获取	有效范围	初始值	内容
AuxPoint	辅助点	LREAL[0..15]	启动时	<ul style="list-style-type: none"> ■ “0: 边界点指定 (mcBorder)”、“1: 中心点指定 (mcCenter)”的情况下: -10000000000.0~10000000000.0 ■ “2: 半径指定 (mcRadius)”的情况下: 0.000001~2147483647.0 	0.0	<p>从启动时的当前位置设置辅助点(边界点、中心点、半径)的相对位置。</p> <p>是1维的数组数据。根据圆弧插补模式(CircMode), 将变为如下所示。</p> <ul style="list-style-type: none"> ■ “0: 边界点指定 (mcBorder)”、“1: 中心点指定 (mcCenter)”的情况下: 作为配置轴1~16的相对位置处理。设置边界点、中心点的绝对位置。 ■ “2: 半径指定 (mcRadius)”的情况下: 将第1元素作为半径处理, 并忽略第2元素及以后。 <p>关于详细内容, 请参阅下述章节。 ☞ 1703页 辅助点(AuxPoint)</p>
EndPoint	终点	LREAL[0..15]	启动时	-10000000000.0~10000000000.0	0.0	<p>设置从启动时的当前位置到终点为止的移动量。</p> <p>是1维的数组数据。作为配置轴1~16的绝对位置处理。</p> <p>关于详细内容, 请参阅下述章节。 ☞ 1703页 终点(EndPoint)</p>
PathChoice	路径选择	INT (MC_CIRC_PATHCHOICE)	启动时	0~5	0	<p>设置圆弧插补的旋转方向。</p> <ul style="list-style-type: none"> • 0: CW (mcCW) • 1: CCW (mcCCW) • 2: 就近 (mcShortWay) • 3: 就远 (mcLongWay) • 4: CW就远 (mcCWLongWay) • 5: CCW就远 (mcCCWLongWay) <p>* 在圆弧插补模式(CircMode)中设置了“0: 边界点指定 (mcBorder)”的情况下, 将忽略输入。</p> <p>* 在圆弧插补模式(CircMode)中设置了“2: 半径指定 (mcRadius)”的情况下, “0: CW (mcCW)”表示CW就近, “1: CCW (mcCCW)”表示CCW就近的含义。</p> <p>关于详细内容, 请参阅下述章节。 ☞ 1704页 路径选择(PathChoice)</p>
Velocity	速度	LREAL	启动时/可重启/可连续更新	0.0、0.0001~2500000000.0	0.0	<p>设置速度指令值。</p> <p>关于详细内容, 请参阅下述章节。 ☞ 1706页 速度(Velocity)</p>
Acceleration	加速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	<p>设置加速度。</p> <p>关于详细内容, 请参阅下述章节。 ☞ 1706页 加速度(Acceleration)</p>
Deceleration	减速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	<p>设置减速度。</p> <p>关于详细内容, 请参阅下述章节。 ☞ 1706页 减速度(Deceleration)</p>
Jerk	Jerk	LREAL	启动时	0.0000、0.0001~2147483647.0	0.0	<p>设置Jerk。</p> <p>关于详细内容, 请参阅下述章节。 ☞ 1706页 Jerk</p>
CircularErrorTolerance	圆弧插补误差允许值	LREAL	启动时	0.000001~100000.0	100.0	<p>设置圆弧插补误差的允许范围。</p> <p>关于详细内容, 请参阅下述章节。 ☞ 1706页 圆弧插补误差允许值(CircularErrorTolerance)</p>
BufferMode	缓冲模式	INT (MC_BUFFER_MODE)	启动时	0~5	0	<p>设置缓冲模式。</p> <ul style="list-style-type: none"> • 0: Aborting (mcAborting) • 1: Buffered (mcBuffered) • 2: BlendingLow (mcBlendingLow) • 3: BlendingPrevious (mcBlendingPrevious) • 4: BlendingNext (mcBlendingNext) • 5: BlendingHigh (mcBlendingHigh) <p>关于详细内容, 请参阅下述章节。 ☞ 1707页 缓冲模式(BufferMode)</p>
Options	选项	DWORD (HEX)	启动时	00000000H~00000005H	00000000H	<p>将McV_MoveCircularInterpolateRelative(相对值圆弧插补控制)的功能选项以位指定进行设置。</p> <p>关于详细内容, 请参阅下述章节。 ☞ 1707页 选项(Options)</p>

■输出变量

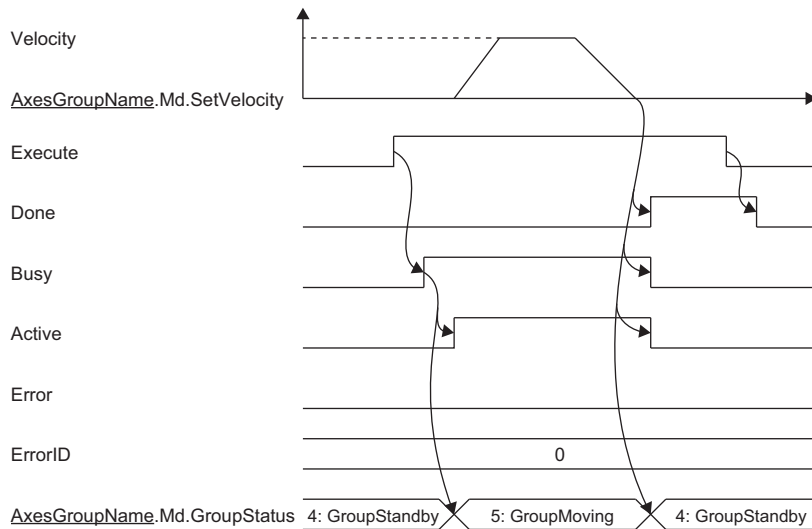
输出变量	名称	数据类型	初始值	内容
Done	完成	BOOL	FALSE	控制完成时，将变为TRUE。 动作完成时根据启动(Execute)的状态将变为如下所示。 ■启动(Execute)为TRUE的情况下 在将启动(Execute)置为FALSE之前将保持为TRUE不变。 ■启动(Execute)为FALSE的情况下 仅1周期为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)时，将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)正在控制轴时，将变为TRUE。 对同一轴组执行了多个MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)的情况下，控制中(Active)变为TRUE的只有1个MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)。
CommandAborted	执行中断	BOOL	FALSE	由于错误及多重启动等，MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)的执行中断时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。  所使用的控制器的用户手册

功能

- 圆弧插补控制指定轴组，对机械直交配置的直线轴进行插补控制，以确保从起点(移动开始点)到终点的轨迹为圆弧。
- 在圆弧插补控制中，可以从轴组中设置的配置轴中使用任意的2轴进行插补控制。
- 在圆弧插补控制中，可以设置边界点指定、中心点指定、半径指定的圆弧插补方式。
- 在MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)中，从当前位置指定终点及辅助点的移动量进行2轴的圆弧插补控制。

■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■圆弧插补轴(CircAxes)

在圆弧插补控制中，从轴组中设置的配置轴中，使用任意的2轴进行圆弧插补。

从轴组的配置轴中，通过圆弧插补轴(CircAxes)设置进行圆弧插补的轴。圆弧插补轴(CircAxes)具有2个数组元素。在数组中，设置插补控制中使用的配置轴的索引编号(1~16: 配置轴1~16)。

数组的第1元素将变为基准轴。

指定了未设置轴的配置轴的情况下，将变为圆弧插补轴未设置(错误代码: 1ACAH)。

例

将配置轴2、3设置为圆弧插补轴的情况下

```
CircAxes[0]:= 2;
```

```
CircAxes[1]:= 3;
```


■圆弧插补模式(CircMode)

设置用于进行圆弧插补控制的圆弧插补方式(边界点指定、中心点指定、半径指定)。

根据圆弧插补模式(CircMode)，辅助点(AuxPoint)的设置内容有所不同。

设定值	辅助点(AuxPoint)
0: 边界点指定(mcBorder)	辅助点(AuxPoint)设置连接始点与终点的圆弧上的边界点。 使用圆弧插补轴(CircAxes)中设置的2轴的配置轴，以通过设置的边界点的圆弧的轨迹进行定位控制。
1: 中心点指定(mcCenter)	辅助点(AuxPoint)设置圆弧的中心点。 使用圆弧插补轴(CircAxes)中设置的2轴的配置轴，以将设置的中心点作为中心的圆弧的轨迹进行定位控制。
2: 半径指定(mcRadius)	辅助点(AuxPoint)设置圆弧的半径。 使用圆弧插补轴(CircAxes)中设置的2轴的配置轴，以具有设置的半径的圆弧的轨迹进行定位控制。

关于通过设置的圆弧插补模式(CircMode)进行的圆弧插补控制的动作，请参阅下述手册。

所使用的控制器的用户手册

■辅助点(AuxPoint)

设置进行圆弧插补的辅助点(边界点、中心点、半径)的设置。

辅助点(AuxPoint)具有16个数组元素。对于辅助点(AuxPoint)，根据圆弧插补模式(CircMode)设置内容有所不同。

圆弧插补模式(CircMode)	设置范围	辅助点(AuxPoint)的设置
0: 边界点指定(mcBorder)	-10000000000.0~10000000000.0	辅助点(AuxPoint)设置连接始点与终点的圆弧上的边界点。 AuxPoint[0..15]表示配置轴1~16的辅助点位置。 • 未设置为圆弧插补轴的配置轴的辅助点(AuxPoint)将被忽略。 • 设置为圆弧插补轴的所有配置轴后面的辅助点(AuxPoint)可以省略。
1: 中心点指定(mcCenter)		辅助点(AuxPoint)设置圆弧的中心点。 AuxPoint[0..15]表示配置轴1~16的辅助点位置。 • 未设置为圆弧插补轴的配置轴的辅助点(AuxPoint)将被忽略。 • 设置为圆弧插补轴的所有配置轴后面的辅助点(AuxPoint)可以省略。
2: 半径指定(mcRadius)	0.000001~2147483647.0	辅助点(AuxPoint)设置圆弧的半径。 AuxPoint[0]表示圆弧的半径。 • AuxPoint[1]及以后的输入将被忽略，因此可以省略。

例

将配置轴2、3设置为圆弧插补轴，并在配置轴2、3的辅助点的相对位置中设置2000.0、3000.0的情况下

```
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 可以省略“AuxPoint[3]:= 0.0;~AuxPoint[15]:= 0.0;”。

■终点(EndPoint)

设置从启动时的当前位置到终点为止的移动量。

终点(EndPoint)具有16个数组元素。EndPoint[0..15]表示配置轴1~16的终点位置。

- 未设置为圆弧插补轴的配置轴的终点(EndPoint)将被忽略。
- 设置为圆弧插补轴的所有配置轴后面的终点(EndPoint)可以省略。

■路径选择 (PathChoice)

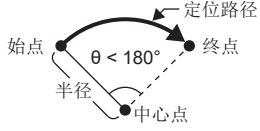
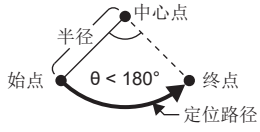
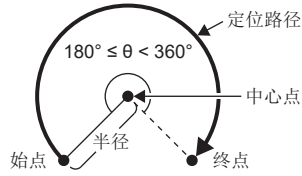
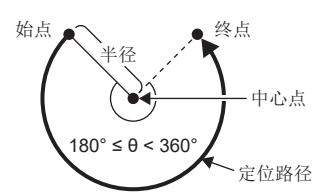
设置圆弧插补中的旋转方向。将CircAxes (圆弧插补轴) 中设置的数组的第1元素作为基准轴考虑旋转方向。对于路径选择 (PathChoice)，在圆弧插补模式 (CircMode) 中设置了“1: 中心点指定 (mcCenter)”或“2: 半径指定 (mcRadius)”的情况下进行设置。设置了“0: 边界点指定 (mcBorder)”的情况下，将忽略输入。

通过设置的圆弧插补模式 (CircMode) 的路径选择 (PathChoice) 的设定值、设置内容、可控制的圆弧的中心角、路径如下所示。

- “1: 中心点指定 (mcCenter)” 的路径

设定值	设置内容	可控制的圆弧的中心角	路径
0: CW (mcCW)	CW	$0^\circ < \theta \leq 360^\circ$	
1: CCW (mcCCW)	CCW	$0^\circ < \theta \leq 360^\circ$	
2: 就近 (mcShortWay)	就近 <ul style="list-style-type: none"> • 始点、中心点、终点位于一直线上的情况下，将变为CW。(向CW方向绘制半圆。) • 始点=终点的情况下，将变为路径选择设置不正确 (错误代码: 1B06H)。 	$0^\circ < \theta \leq 180^\circ$	
3: 就远 (mcLongWay)	就远 <ul style="list-style-type: none"> • 始点、中心点、终点位于一直线上的情况下，将变为CW。(向CW方向绘制半圆。) • 始点 = 终点的情况下，将变为CW。(向CW方向绘制正圆。) 	$180^\circ \leq \theta \leq 360^\circ$	

• “2: 半径指定 (mcRadius)” 的路径

设定值	设置内容	可控制的圆弧的中心角	路径
0: CW (mcCW)	CW <ul style="list-style-type: none"> 始点、中心点、终点位于一直线上的情况下，将变为路径选择设置不正确(错误代码: 1B06H)。 	$0^\circ < \theta < 180^\circ$	
1: CCW (mcCCW)	CCW <ul style="list-style-type: none"> 始点、中心点、终点位于一直线上的情况下，将变为路径选择设置不正确(错误代码: 1B06H)。 	$0^\circ < \theta < 180^\circ$	
4: CW就远 (mcCWLongWay)	CW就远 <ul style="list-style-type: none"> 始点、中心点、终点位于一直线上的情况下，圆弧将绘制半圆。 	$180^\circ \leq \theta < 360^\circ$	
5: CCW就远 (mcCCWLongWay)	CCW就远 <ul style="list-style-type: none"> 始点、中心点、终点位于一直线上的情况下，圆弧将绘制半圆。 	$180^\circ \leq \theta < 360^\circ$	

■速度 (Velocity)

设置MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)中的路径的最大速度。

设置范围
0.0、0.0001~2500000000.0*1*2

- *1 由于进行浮点运算，因此指令速度的下限值中将产生下述限制。
对指令速度进行了运算周期换算的速度小于“0.00001”的情况下，将变为超出运算周期换算速度范围(错误代码：1B16H)(速度更改时为超出运算周期换算速度范围警告(事件代码：00D2FH))。为了提高浮点运算的精度，应通过更改位置指令单位(AxisName.Pr.Unit_Position)或速度指令单位(AxisName.Pr.Unit_Velocity)设置运算周期换算后的速度不低于“0.00001”。
- *2 在进行多重启动的运动控制FB中指定速度为“0.0”的情况下，将变为之前的运动控制FB的指定速度。

■加速度 (Acceleration)

设置MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)中的加速度。

根据加减速方式设置(选项(Options)：位0~2)的设置，设置范围有所不同。

加减速方式设置(选项(Options)：位0~2)	设置范围
0：加减速速度指定方式(mcAccDec)	0.0000*1、0.0001~2147483647.0[U/s ²]的正数
1：加减速时间恒定方式(mcFixedTime)	0.000000*1、0.000001~8400.0[s]的正数

- *1 启动时，根据启动时加减速速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior)，动作将发生变化。加速度更改、减速度更改时，不受理更改。

■减速度 (Deceleration)

设置MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)中的减速度。

设置了加减速方式设置(选项(Options)：位0~2)的“0：加减速速度指定方式(mcAccDec)”的情况下，对设置范围进行设置。设置了“1：加减速时间恒定方式(mcFixedTime)”的情况下，不使用减速度(Deceleration)。

加减速方式设置(选项(Options)：位0~2)	设置范围
0：加减速速度指定方式(mcAccDec)	0.0000*1、0.0001~2147483647.0[U/s ²]的正数
1：加减速时间恒定方式(mcFixedTime)	不使用

- *1 启动时，根据启动时加减速速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior)，动作将发生变化。加速度更改、减速度更改时，不受理更改。

■Jerk

设置MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)中的Jerk。

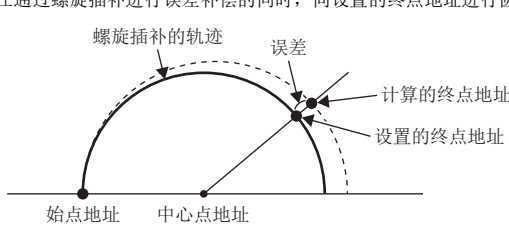
设置了加减速方式设置(选项(Options)：位0~2)的“0：加减速速度指定方式(mcAccDec)”的情况下，对设置范围进行设置。设置了“1：加减速时间恒定方式(mcFixedTime)”的情况下，不使用Jerk。

加减速方式设置(选项(Options)：位0~2)	设置范围
0：加减速速度指定方式(mcAccDec)	0.0000、0.0001~2147483647.0[U/s ²]的正数
1：加减速时间恒定方式(mcFixedTime)	不使用

■圆弧插补误差允许值 (CircularErrorTolerance)

在中心点指定的圆弧插补控制中，通过始点位置与中心点位置计算出的圆弧的轨迹与终点(EndPoint)中设置的终点位置有可能有偏差。

圆弧插补误差允许值(CircularErrorTolerance)设置计算出的圆弧的轨迹与终点位置的误差的允许范围。

允许范围	内容
计算出的误差 ≤ 圆弧插补误差允许值 (CircularErrorTolerance)	<p>在通过螺旋插补进行误差补偿的同时，向设置的终点地址进行圆弧插补。</p> 
计算出的误差 > 圆弧插补误差允许值 (CircularErrorTolerance)	<p>定位启动时将变为圆弧插补误差允许值溢出(错误代码：1AD9H)且不启动。 定位控制中的情况下，检测到错误时将立即停止。</p>

■缓冲模式(BufferMode)

设置用于进行多重启动(缓冲模式)的动作。

MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)中可设置的缓冲模式如下所示。

设定值	内容
0: Aborting(mcAborting)	中断(取消)控制中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在控制中的FB中缓冲下一个FB。 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB完成时,依次执行缓冲FB。
2: BlendingLow(mcBlendingLow)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将控制中的FB与缓冲FB的目标速度中较低一方的速度作为切换速度。
3: BlendingPrevious(mcBlendingPrevious)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将控制中的FB的目标速度作为切换速度。
4: BlendingNext(mcBlendingNext)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将缓冲FB的目标速度作为切换速度。
5: BlendingHigh(mcBlendingHigh)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将控制中的FB与缓冲FB的目标速度中较高一方的速度作为切换速度。

^{*1} 在控制中的FB与缓冲FB之间不执行停止。

要点

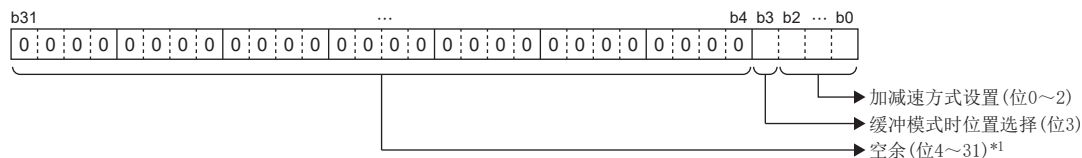
关于多重启动(缓冲模式)的详细内容,请参阅下述手册。

📖所使用的控制器的用户手册

■选项(Options)

将MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)中使用的功能选项以位指定进行设置。

以位指定设置的内容如下所示。



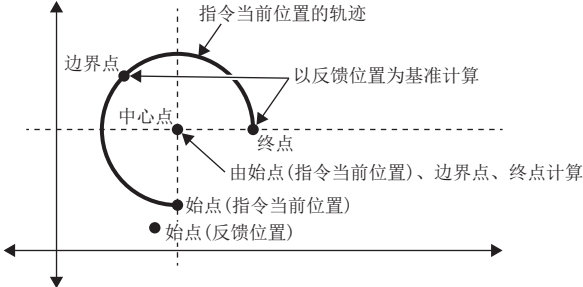
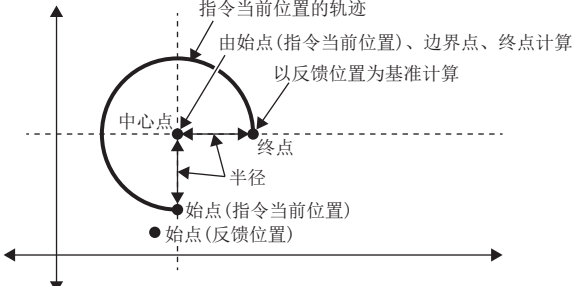
^{*1} 空余中应设置“0”。设置了“0”以外的情况下,将变为超出Options范围(错误代码:1ABBH)。

位	名称	内容
0~2	加减速方式设置	设置用于进行控制的加减速方式。 • 0: 加减速速度指定方式(mcAccDec) • 1: 加减速时间恒定方式(mcFixedTime)
3	缓冲模式时位置选择	设置对相对值定位控制进行多重启动时的位置。 • 0: 指令当前位置 • 1: 反馈位置 * 缓冲模式(BufferMode)的“0: Aborting(mcAborting)”设置时将有效。

• 加减速方式设置(位0~2)

设定值	内容
0: 加减速速度指定方式(mcAccDec)	是使用MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)中设置的加速度(Acceleration)、减速度(Deceleration)、Jerk进行加速/减速的方式。
1: 加减速时间恒定方式(mcFixedTime)	是与速度无关,使用MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)中设置的加减速时间进行加速/减速的方式。 对于加减速时间,设置为加速度(Acceleration),不使用减速度(Deceleration)、Jerk。

• 缓冲模式时位置选择(位3)

设定值	内容
0: 指令当前位置	是从指令当前位置开始的相对位置控制。
1: 反馈位置	<p>是从反馈位置开始的相对位置控制。 辅助点(边界点、中心点、半径)、终点以反馈位置为基准计算,且始点使用指令当前位置。 各圆弧插补模式(CircMode)的动作如下所示。</p> <p>■“0: 边界点指定(mcBorder)”的情况下 以反馈位置为基准计算边界点、终点。中心点由始点(指令当前位置)、边界点、终点确定。</p>  <p>■“1: 中心点指定(mcCenter)”的情况下 以反馈位置为基准计算中心点、终点。圆弧的轨迹可能会变为椭圆的圆弧。</p>  <p>■“2: 半径指定(mcRadius)”的情况下 以反馈位置为基准计算终点。中心点由始点(指令当前位置)、半径、终点确定。</p> 

■必要对象数据

使用MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)的情况下,应对指定的轴组的所有配置轴设置下述对象数据。

- Target position(607AH)

存在有未设置对象数据的配置轴的情况下,将变为必须从站对象未设置(错误代码:1AF7H)且不启动。

关于对象数据设置的详细内容,请参阅下述手册。

📖所使用的控制器的用户手册

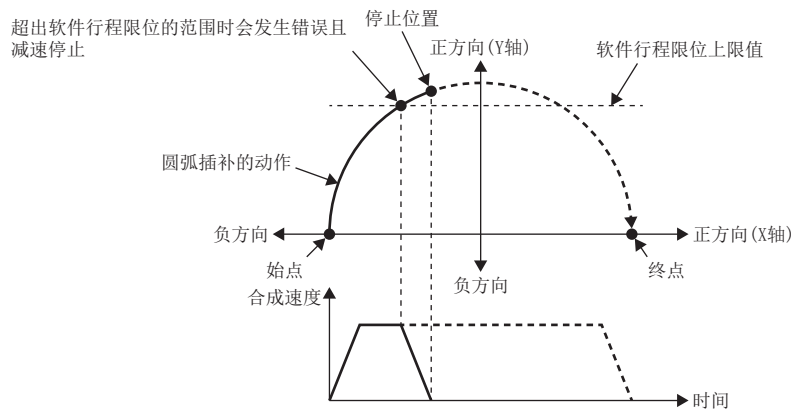
注意事项

- 无法进行包含将行程限位设置为无效的轴的圆弧插补。否则将变为圆弧插补时软件行程限位无效(错误代码: 1ADAH)且不启动。
- 插补动作中插补路径超出行程限位范围的情况下, 将变为软件行程限位溢出(正方向)(错误代码: 1A83H)或软件行程限位溢出(负方向)(错误代码: 1A84H)且停止运行。

例

在Y轴的正方向上超出了软件行程限位的上限的情况下立即停止的情况下, 在发生错误的同时执行停止。

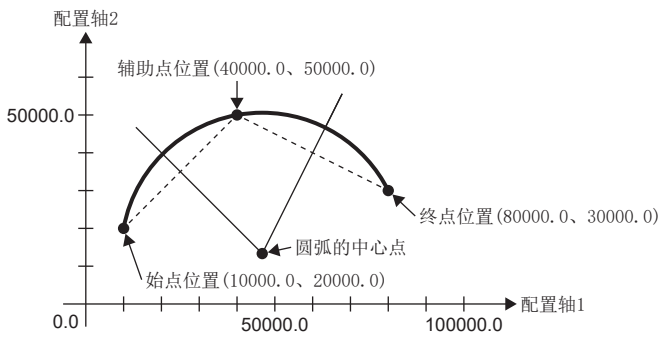
减速停止的情况下, 将如下图所示沿着圆弧的轨迹进行减速, 并停止。



程序示例

将圆弧插补控制启动(bCircularInterpolateCMD)置为TRUE，并将轴组1(AxesGroup001)置为了有效后，根据下述设置进行轴组1(AxesGroup001)的2轴圆弧插补(辅助点指定)的相对值定位控制的程序示例如下所示。

• 动作



• 轴组

项目	设定值
配置轴[1]	Axis0001
配置轴[2]	Axis0002

• 设置

项目	设定值	
	配置轴1	配置轴2
辅助点	30000.0	30000.0
终点	70000.0	10000.0
速度	50000.0	
加速度	50000.0	
减速度	50000.0	
Jerk	0.0	
圆弧插补误差允许值	100.0	

■轴

轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1
2	Axis0002	AXIS_REF	轴2

■轴组

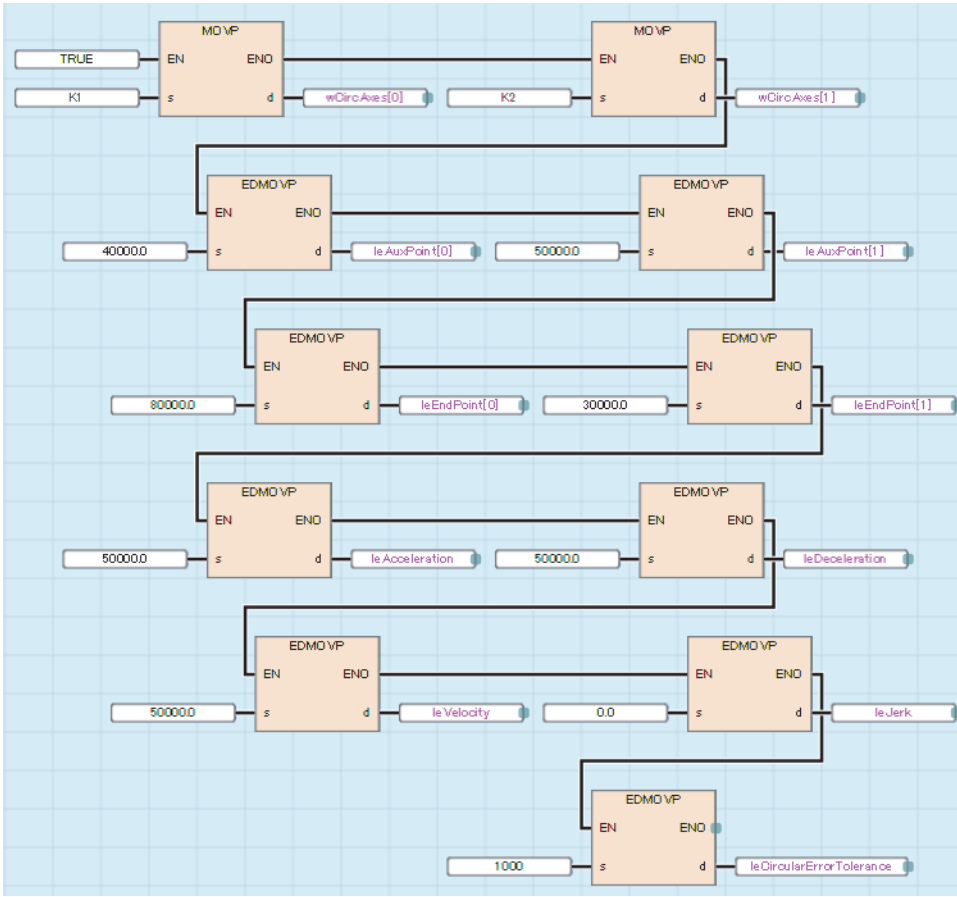
轴组No.	标签名	数据类型	注释
1	AxesGroup001	AXES_GROUP_REF	轴组1

■使用的标签

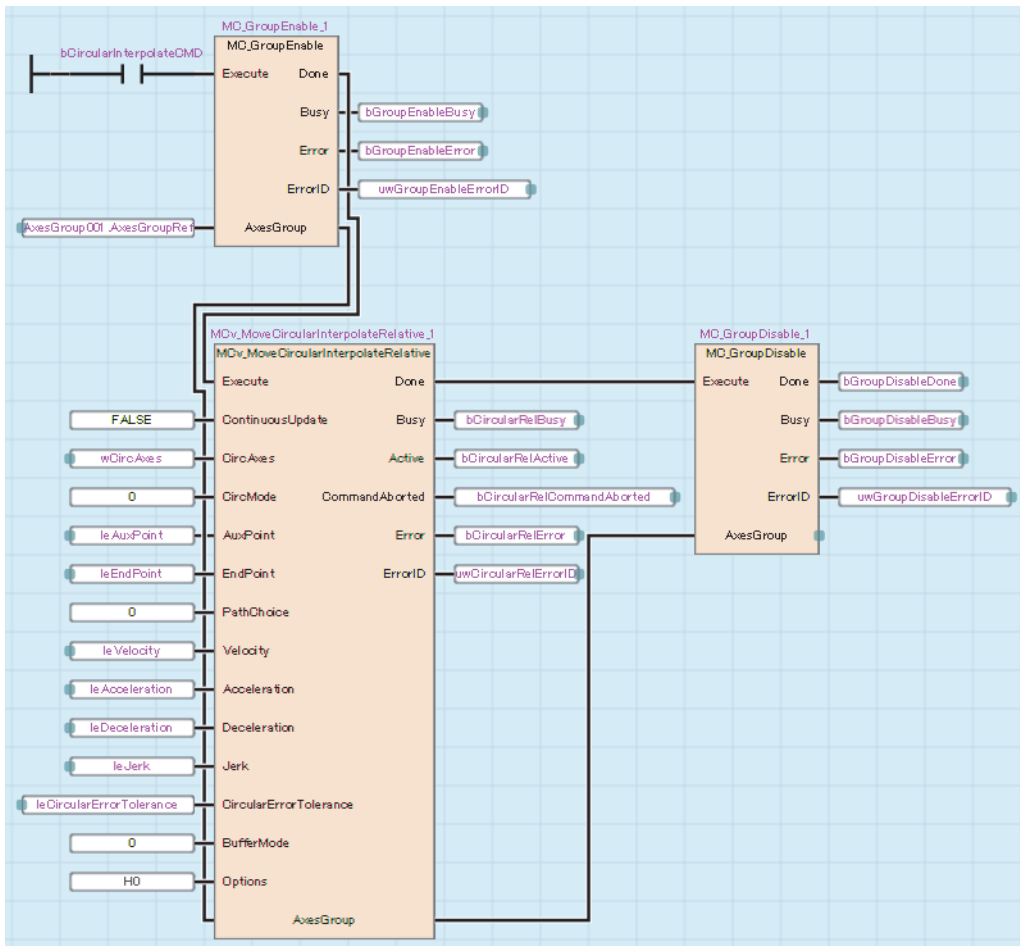
标签名	数据类型	注释
MCv_MoveCircularInterpolateRelative_1	MCv_MoveCircularInterpolateRelative	相对值圆弧插补控制FB
wCircAxes	字[有符号](0..1)	圆弧插补轴
leAuxPoint	双精度实数(0..15)	辅助点
leEndPoint	双精度实数(0..15)	终点
leVelocity	双精度实数	速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
leCircularErrorTolerance	双精度实数	圆弧插补误差允许值
bCircularRelDone	位	执行完成
bCircularRelBusy	位	执行中
bCircularRelActive	位	控制中
bCircularRelCommandAborted	位	执行中断
bCircularRelError	位	错误
uwCircularRelErrorID	字[无符号]/位串[16位]	错误代码
MC_GroupEnable_1	MC_GroupEnable	轴组有效FB
bCircularInterpolateCMD	位	圆弧插补控制启动
bGroupEnableDone	位	轴组有效完成
bGroupEnableBusy	位	执行中
bGroupEnableError	位	错误
uwGroupEnableErrorID	字[无符号]/位串[16位]	错误代码
MC_GroupDisable_1	MC_GroupDisable	轴组无效FB
bGroupDisableDone	位	轴组无效完成
bGroupDisableBusy	位	执行中
bGroupDisableError	位	错误
uwGroupDisableErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

- 2轴圆弧插补控制用数据的设置



• 轴组有效/圆弧插补控制(边界点指定)/轴组无效



■ST程序

```
//-----2轴圆弧插补控制用数据的设置-----
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;

//-----轴组有效-----
MC_GroupEnable_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bCircularInterpolateCMD ,
    Done=> bGroupEnableDone ,
    Busy=> bGroupEnableBusy ,
    Error=> bGroupEnableError ,
    ErrorID=> uwGroupEnableErrorID
);

//-----圆弧插补控制(边界点指定)-----
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
);

//-----轴组无效-----
MC_GroupDisable_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
```

```
Execute:= bCircularReIDone ,  
Done=> bGroupDisableDone ,  
Busy=> bGroupDisableBusy ,  
Error=> bGroupDisableError ,  
ErrorID=> uwGroupDisableErrorID  
);
```

46.14 凸轮动作开始

MC_CamIn

按照指定的凸轮数据开始凸轮动作。

梯形图	FBD/LD	ST
		<pre> 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?); </pre>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Master	主轴	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName.AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName.AxisRef. (轴信息)
Slave	从轴	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName.AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName.AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_CamIn(凸轮动作开始)。
ContinuousUpdate	连续更新	BOOL	启动时	TRUE、FALSE	FALSE	设置将主轴偏置(MasterOffset)、从轴偏置(SlaveOffset)、主轴系数(MasterScaling)、从轴系数(SlaveScaling)、凸轮表ID(CamTableID)的连续更改是置为有效，还是置为无效。 • FALSE: 无效 • TRUE: 有效
MasterOffset	主轴偏置	LREAL	启动时/可重启/可连续更新	-10000000000.0~10000000000.0	0.0	将主轴(Master)的相位按偏置量移动。 关于详细内容，请参阅下述章节。 ☞ 1727页 主轴偏置(MasterOffset)
SlaveOffset	从轴偏置	LREAL	启动时/可重启/可连续更新	-10000000000.0~10000000000.0	0.0	将从轴(Slave)的位移按偏置量移动。 关于详细内容，请参阅下述章节。 ☞ 1728页 从轴偏置(SlaveOffset)
MasterScaling	主轴系数	LREAL	启动时/可重启/可连续更新	0.01~10.0	1.0	对凸轮表的1周期长进行放大、缩小。 关于详细内容，请参阅下述章节。 ☞ 1729页 主轴系数(MasterScaling)

输入变量	名称	数据类型	获取	有效范围	初始值	内容
SlaveScaling	从轴系数	LREAL	启动时/可重启/可连续更新	0.01~10.0	1.0	对凸轮表的行程量进行放大、缩小。 关于详细内容, 请参阅下述章节。 ☞ 1729页 从轴系数(SlaveScaling)
MasterStartDistance	主轴跟踪距离	LREAL	启动时	-10000000000.0~10000000000.0	0.0	忽略设定值。
MasterSyncPosition	主轴同步开始位置	LREAL	启动时	-10000000000.0~10000000000.0	0.0	设置1周期当前值(InputPerCycle)开始同步的主轴(Master)的位置。 * 主轴同步开始位置的通过检查对象指定(选项(Options): 位21)为“1: 主轴的指令当前位置或反馈位置”的情况下, 有效范围在环形计数器范围内。
StartMode	开始模式	INT (MC_START_MODE)	启动时	0、1	0	设置开始凸轮动作的时机。 • 0: 即时(mcImmediate) • 1: 绝对(mcAbsolute) 关于详细内容, 请参阅下述章节。 ☞ 1730页 开始模式(StartMode)
MasterValueSource	主轴数据源选择	INT (MC_SOURCE)	启动时	1、2、101、102	1	设置主轴(Master)的数据源。 • 1: 指令当前值(mcSetValue) • 2: 反馈值(mcActualValue) • 101: 最新指令当前值(mcLatestSetValue) • 102: 最新反馈值(mcLatestActualValue) 关于详细内容, 请参阅下述章节。 ☞ 1731页 主轴数据源选择(MasterValueSource)
CamTableID	凸轮表ID	MC_CAM_ID	启动时/可重启/可连续更新	1~60000	0	设置凸轮的ID。 对于凸轮的ID, 通过MC_CamTableSelect(凸轮表选择)事先展开到展开区域后使用。
BufferMode	缓冲模式	INT (MC_BUFFER_MODE)	启动时	0、1	0	设置缓冲模式。 • 0: Aborting(mcAborting) • 1: Buffered(mcBuffered) 关于详细内容, 请参阅下述章节。 ☞ 1732页 缓冲模式(BufferMode)
Options	选项	DWORD(HEX)	启动时	00000000H~00210000H	00000000H	将MC_CamIn(凸轮动作开始)的功能选项以位指定进行设置。 关于详细内容, 请参阅下述章节。 ☞ 1736页 选项(Options)

■输出变量

输出变量	名称	数据类型	初始值	内容
InSync	同步中	BOOL	FALSE	输出值(OutputData)开始了同步时, 将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_CamIn(凸轮动作开始)时, 将变为TRUE。
Active	控制中	BOOL	FALSE	1周期当前值(InputPerCycle)开始了同步时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MC_CamIn(凸轮动作开始)的执行中断时, 将变为TRUE。 执行指令(Execute)变为FALSE时, 将变为FALSE。
Error	错误	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时, 将返回错误代码。 关于错误代码的详细内容, 请参阅下述手册。 ☞ 所使用的控制器的用户手册
EndOfProfile	凸轮循环完成	BOOL	FALSE	控制中(Active)变为TRUE后, 每次移动1周期长时, 仅调用FB的POU(程序部件)的执行周期的1个周期, 将变为TRUE。 关于详细内容, 请参阅下述章节。 ☞ 1736页 凸轮循环完成(EndOfProfile)

■公开变量

公开变量	名称	数据类型	初始值	内容
InputPerCycle	1周期当前值	LREAL	0.0	存储1周期当前值。 关于详细内容, 请参阅下述章节。 ☞ 1736页 1周期当前值 (InputPerCycle)
Reference	基准值	LREAL	0.0	存储基准值。 关于详细内容, 请参阅下述章节。 ☞ 1737页 基准值 (Reference)
OutputData	输出值	LREAL	0.0	存储输出值。 关于详细内容, 请参阅下述章节。 ☞ 1738页 输出值 (OutputData)
InstanceID	实例ID	WORD (UINT)	0	是实例ID。 实例创建时由系统自动设置。 在FB的输入等中使用。

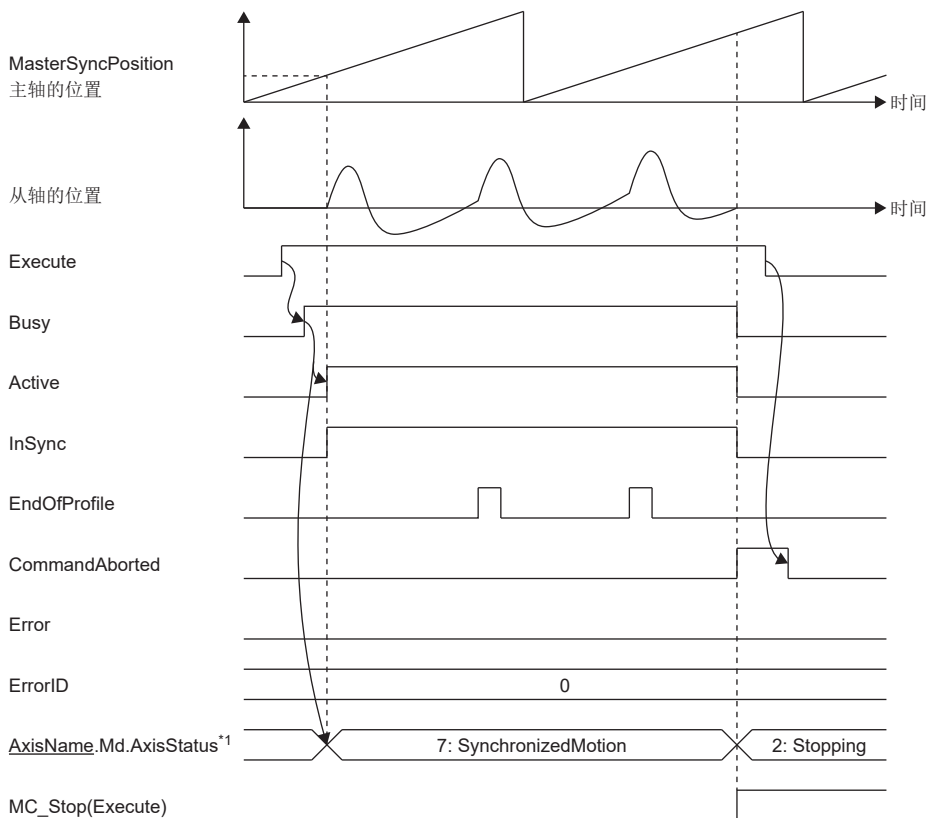
功能

- 对于MC_CamIn(凸轮动作开始), 设置主轴偏置 (MasterOffset)、从轴偏置 (SlaveOffset)、主轴系数 (MasterScaling)、从轴系数 (SlaveScaling)、主轴跟踪距离 (MasterStartDistance)、主轴同步开始位置 (MasterSyncPosition)、开始模式 (StartMode)、主轴数据源选择 (MasterValueSource)、凸轮表ID (CamTableID)、缓冲模式 (BufferMode), 并执行凸轮动作。
- 结束动作的情况下, 通过MC_Stop(强制停止)进行。

■时序图

- 正常完成的情况下

开始模式 (StartMode) 为“1: 绝对 (mcAbsolute)”, MC_CamTableSelect (凸轮表选择) 中设置的主轴绝对坐标 (MasterAbsolute) 为“FALSE (相对坐标)”的情况下



*1 是从轴 (Slave) 的轴状态。

- 异常完成的情况下

关于异常完成时的时序图, 请参阅下述章节。

☞ 1375页 通过执行指令 (Execute) 类型的运动控制FB的基本动作

■凸轮动作的开始

通过开始模式(StartMode)的设置,可以设置凸轮动作的同步时机及凸轮控制数据的反映时机。

- 凸轮动作的同步时机

根据开始模式(StartMode)与关联参数(主轴同步开始位置(MasterSyncPosition)与主轴跟踪距离(MasterStartDistance),1周期当前值(InputPerCycle)、基准值(Reference)、输出值(OutputData)的同步时机(开始更新的时机)如下所示。

开始模式(StartMode)*1	开始同步的时机		
	1周期当前值(InputPerCycle)	基准值(Reference)	输出值(OutputData)
0: 即时(mcImmediate)	执行了MC_CamIn(凸轮动作开始)时。 主轴同步开始位置(MasterSyncPosition)与主轴跟踪距离(MasterStartDistance)将被忽略。		
1: 绝对(mcAbsolute)	主轴的位置*2通过了主轴同步开始位置(MasterSyncPosition)*3时。 主轴跟踪距离(MasterStartDistance)将被忽略。		

*1 关于开始模式(StartMode),请参阅下述章节。

☞ 1730页 开始模式(StartMode)

*2 通过主轴同步开始位置的通过检查对象指定(选项(Options):位21)设置要使用的位置数据。关于详细内容,请参阅下述章节。

☞ 1736页 选项(Options)

*3 是主轴的位置跨越主轴同步开始位置(MasterSyncPosition)(或主轴同步开始位置(MasterSyncPosition)+主轴跟踪距离(MasterStartDistance))的状态。变为了相同的值的情况下视为未通过,从相同的值进行了移动的情况下视为通过。

- 通过FB的重启/连续更新进行控制更改的时机

对于将通过FB的重启/连续更新进行了更改的参数反映到控制中的时机,开始模式(StartMode)不同的反映时机如下所示。

开始模式(StartMode)	反映时机
0: 即时(mcImmediate)	即时
1: 绝对(mcAbsolute)	1周期当前值(InputPerCycle)通过了凸轮表的第1点时。

更改值超出范围时,输出相应的错误,保持更改前的值并继续控制。

参数	错误
MasterOffset(主轴偏置)	超出主轴偏置范围(错误代码:1AAFH)
SlaveOffset(从轴偏置)	超出从轴偏置范围(错误代码:1AB0H)
MasterScaling(主轴系数)	超出主轴系数范围(错误代码:1AB1H)
SlaveScaling(从轴系数)	超出从轴系数范围(错误代码:1AB2H)
CamTableID(凸轮表ID)	<ul style="list-style-type: none"> • CamTableID(凸轮表ID)的数值超出范围的情况下 超出凸轮表ID范围(错误代码:1AAEH) • 无CamTableID(凸轮表ID)的展开区域的情况下 无凸轮表警告(事件代码:00D44H)

■凸轮表的信息

作为凸轮表的信息，根据MC_CamTableSelect(凸轮表选择)中设置的重复动作、主轴绝对坐标、从轴绝对坐标的MC_CamIn(凸轮动作开始)执行时的凸轮动作如下所示。

○：用于控制， ×：不用于控制

名称	变量名	插补方法指定(Interpolate)			备注
		0: 直线插补	1: 各区间中指定	2: 样条插补	
重复动作	Periodic	○	○	○	是在工程工具中通过运算配置文件的展开设置，或MC_CamTableSelect(凸轮表选择)的输入变量设置的项目。 是与从凸轮数据格式/旋转刀具格式的运算配置文件展开到展开区域中的数据控制有关的项目。
主轴绝对坐标	MasterAbsolute	○	○	○	
从轴绝对坐标	SlaveAbsolute	○	○	○	
插补方法指定	Interpolate	○	○	○	
1周期长	CycleLength	○	○	○	
行程量	Stroke	○	○	○	
开始点	StartPoint	×	×	×	
初始行程量	StartStroke	×	○	○	

• 重复动作(Periodic)

根据重复动作(Periodic)的设置，MC_CamIn(凸轮动作开始)执行时的动作如下所示。

重复动作(Periodic)	内容
FALSE(单发动作)	<p>控制中(Active)变为TRUE之后，仅1周期执行运行。此后，将变为1周期运行后的动作指定(选项(Options): 位16)中设置的动作。</p> <p>■1周期运行后的动作指定(选项(Options): 位16)为“0: 结束”的情况下 1周期结束后同步状态被解除，此后，重新启动执行指令(Execute)时，基准值(Reference)、输出值(OutputData)将变为“0.0(初始值)”。</p>

重复动作 (Periodic)	内容
FALSE (单发动作)	<p>■ 1周期运行后的动作指定 (Options): 位16为“1: 重启等待”的情况下</p> <p>1周期结束后仍维持同步状态, 重新启动执行指令 (Execute) 时, 也维持基准值 (Reference)、输出值 (OutputData) 的值。同步状态的解除将停止原因输入至从轴。</p> <p>1周期结束后的重启中, 更改了可连续更新的输入标签的值的的情况下, 也不进行输入的获取直到重新启动执行指令 (Execute)。</p> <p>重新启动执行指令 (Execute) 时, 与起开始模式 (StartMode) 的设置无关, 从轴均即时开始动作。</p> <p>1周期结束后的重启等待中, 执行1周期当前值更改时, 该值对下一个周期变为有效。</p>

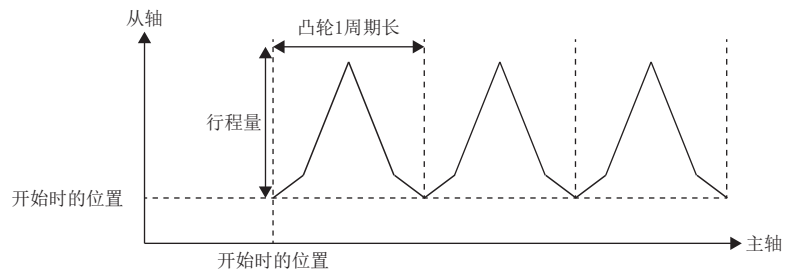
重复动作(Periodic)**内容**

TRUE(重复动作)

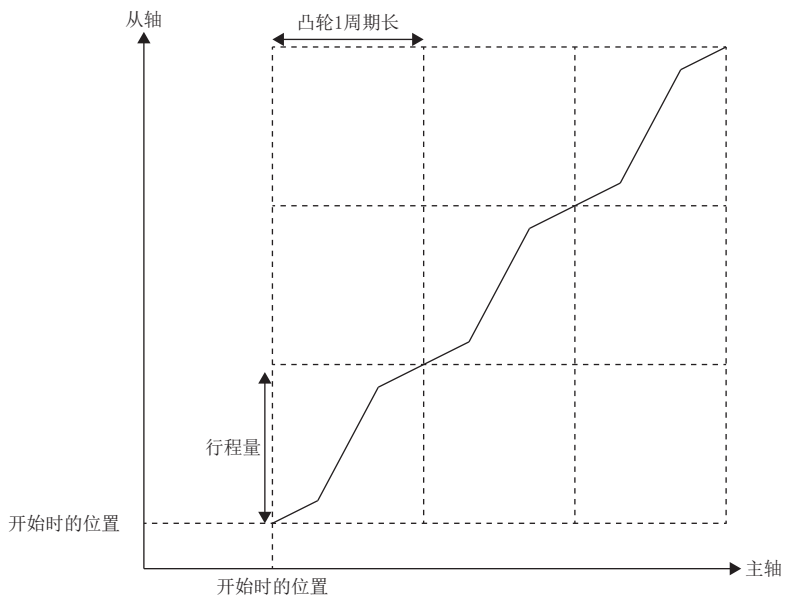
连续重复凸轮表的执行。根据凸轮动作(往复凸轮或进给凸轮)，其动作如下所示。

■往复凸轮

由于重复导致的凸轮表始点的从轴(Slave)的位置每次相同的情况下，将作为往复凸轮执行动作。

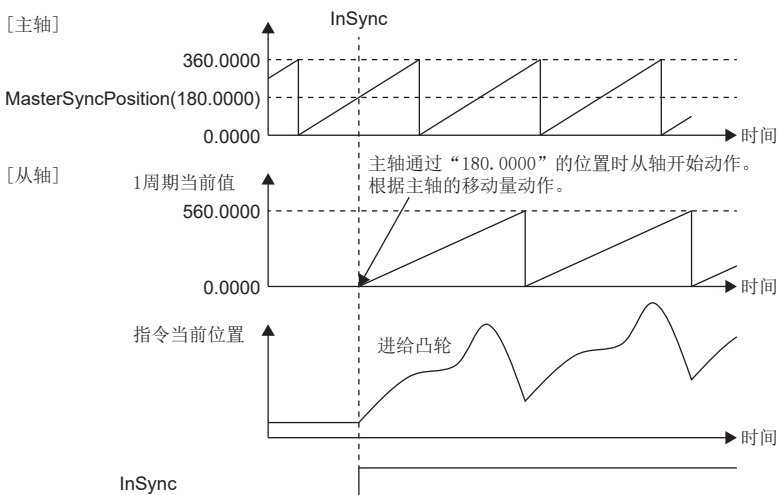
**■进给凸轮**

由于重复导致的凸轮表始点的从轴(Slave)的位置在始点与终点中指令当前位置不相同的情况下，将作为进给凸轮执行动作。



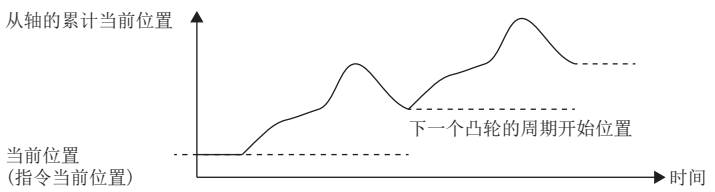
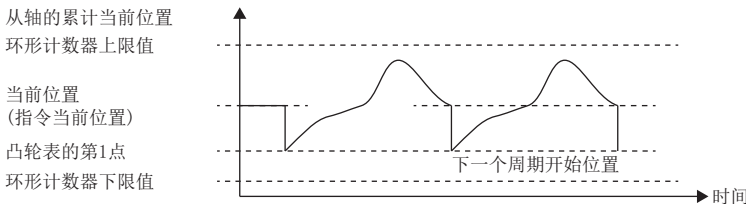
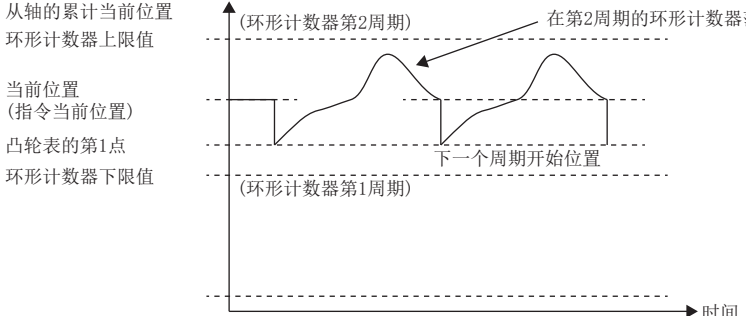
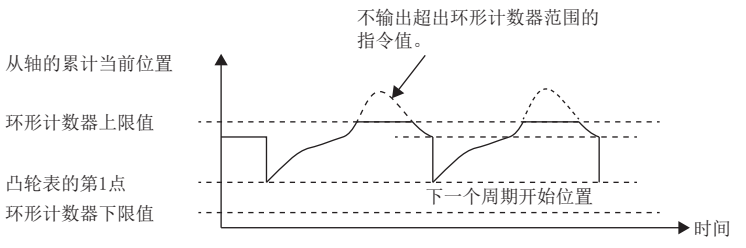
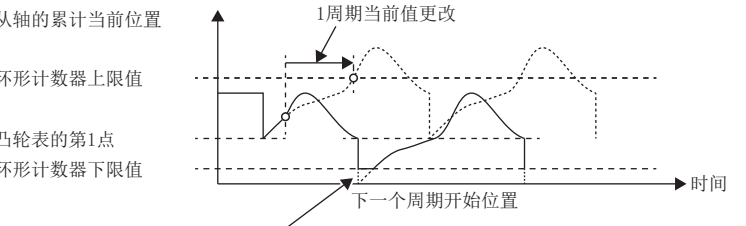
• 主轴绝对坐标 (MasterAbsolute)

根据主轴绝对坐标 (MasterAbsolute) 的设置，凸轮动作如下所示。

主轴绝对坐标 (MasterAbsolute)	内容
FALSE (相对坐标)	<p>主轴同步开始位置 (MasterSyncPosition) 将变为凸轮数据的始点。同步中 (InSync) 变为 TRUE 时，将根据主轴 (Master) 的相对移动量执行凸轮动作。</p> <p>即使凸轮表与主轴 (Master) 的环形计数器不匹配，也将连续执行凸轮动作。</p> <p><例></p> <p>下述设置的情况下</p> <ul style="list-style-type: none"> • 主轴：当前值的环形计数器：0.0000~360.0000 [degree] • 主轴同步开始位置 (MasterSyncPosition)：180.0000 [degree] • 凸轮：凸轮 1 周期长：540.0000 [degree]  <p>[主轴]</p> <p>360.0000 MasterSyncPosition(180.0000) 0.0000</p> <p>[从轴]</p> <p>1周期当前值 560.0000 0.0000</p> <p>指令当前位置</p> <p>进给凸轮</p> <p>InSync</p> <p>时间</p> <p>时间</p> <p>时间</p> <p>时间</p> <p>主轴通过“180.0000”的位置时从轴开始动作。根据主轴的移动量动作。</p>

• 从轴绝对坐标 (SlaveAbsolute)

根据从轴绝对坐标 (SlaveAbsolute) 的设置，凸轮动作如下所示。

从轴绝对坐标 (SlaveAbsolute)	内容
FALSE (相对坐标)	<p>同步中 (InSync) 变为 TRUE 时，从轴将从当前的位置 (指令当前位置) 开始执行凸轮表的动作。</p> <p>凸轮表的重复动作 (Periodic) 为 “TRUE: 重复动作” 的情况下，凸轮表的 1 周期完成时从 1 周期完成时的行程位置 (指令当前位置) 开始下一个周期。</p> 
TRUE (绝对坐标)	<p>在包含同步中 (InSync) 变为了 TRUE 时的从轴 (Slave) 的当前值的环形计数器范围内执行凸轮动作。</p> <p>以 1 运算周期输出指令，以确保在同步中 (InSync) 及凸轮循环完成 (EndOfProfile) 变为了 TRUE 时，从轴 (Slave) 的指令当前位置返回到凸轮表的始点。此时的指令较大的情况下，至驱动器模块的位置指令及速度指令将变大，导致发生 MR-J5 (W)-G 的伺服错误 ([AL. 035 (指令频率异常)])。不输出从轴 (Slave) 的指令当前位置超出环形计数器范围的值。</p> <p>■行程范围在从轴的环形计数器范围内的情况下</p>  <p>■行程范围在从轴的环形计数器范围内 (在第 2 周期启动了凸轮) 的情况下</p>  <p>■行程范围超出从轴的环形计数器范围的情况下</p>  <p>■执行了 1 周期当前值更改的情况下</p> <ul style="list-style-type: none"> • 由于为了避免从轴 (Slave) 执行动作而对基准值 (Reference) 进行补偿，因此超出环形计数器范围的范围发生变化。使从轴 (Slave) 执行动作且更改 1 周期当前值的情况下，应更改主轴偏置 (MasterOffset)。 

- 1周期长(CycleLength)

在1周期长中设置1周期所需的输入量。关于详细内容，请参阅下述章节。

☞ 1736页 1周期当前值(InputPerCycle)

- 行程量(Stroke)

在行程量中设置与行程比100%对应的行程量。关于详细内容，请参阅下述章节。

☞ 1738页 输出值(OutputData)

- 开始点(StartPoint)和初始行程量(StartStroke)

根据插补方法指定(Interpolate)，凸轮动作中使用的开始点(StartPoint)及初始行程量(StartStroke)不相同。

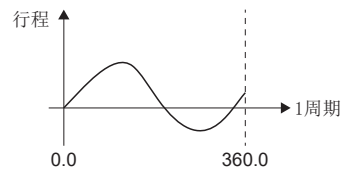
插补方法指定(Interpolate)	凸轮动作的开始点	凸轮动作的初始行程量
0: 直线插补	凸轮表的第1点的输入值	凸轮表的第1点的输出值
1: 各区间中指定	凸轮表的第1点的开始点(StartPoint)	凸轮表的初始行程量(StartStroke)
2: 样条插补		

■ 凸轮动作的开始点

执行动作以确保1周期当前值(InputPerCycle)变为“0.0”。

- 凸轮表

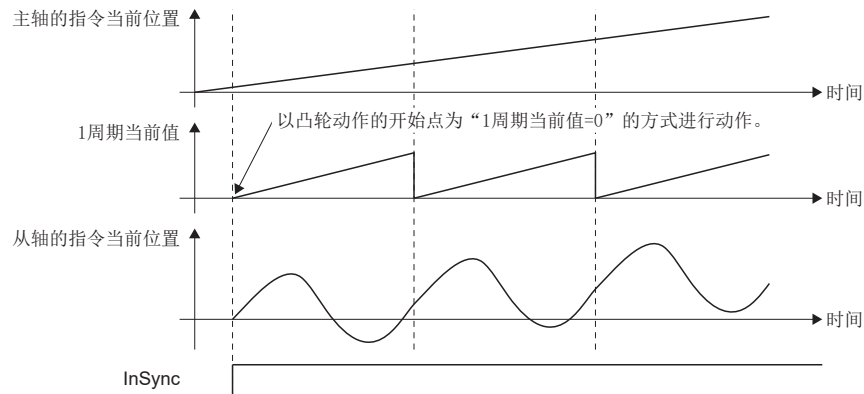
■ 凸轮(1) (凸轮动作的开始点为“0.0”)



- 插补方法：各区间中指定
- 1周期长：360.0
- 开始点：0.0
- 初始行程量：0.0

- 凸轮动作

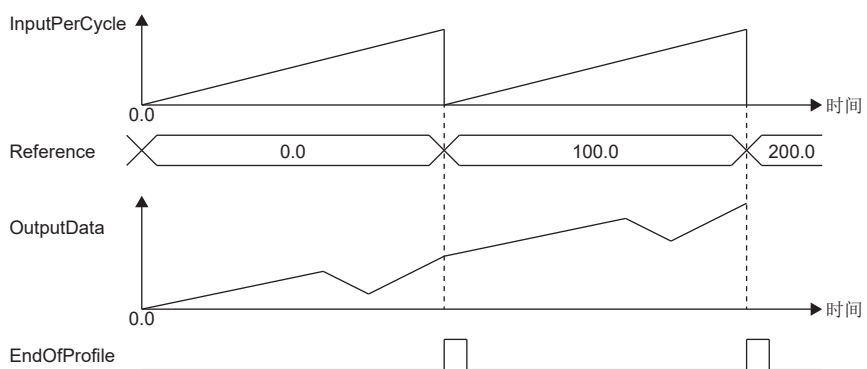
为下述动作。



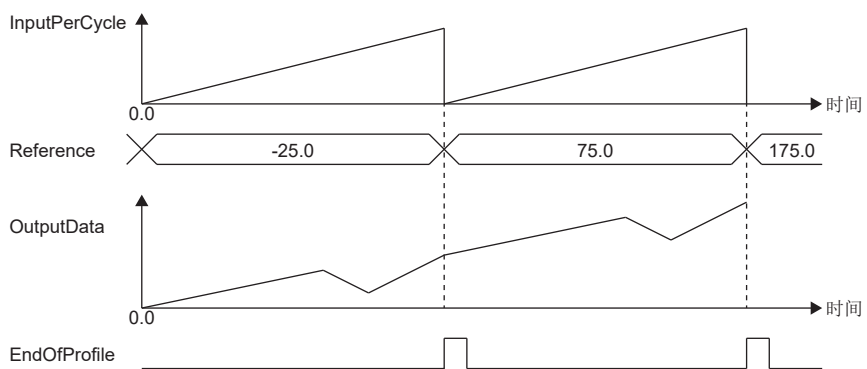
■ 凸轮动作的初始行程量

执行动作以确保其变为同步开始时的输出值(OutputData)。根据凸轮动作的初始行程量，为了防止从轴(Slave)执行动作，将从基准值(Reference)中减去初始行程量。

- 凸轮动作的初始行程量(StartStroke)为“0.0”的情况下



- 凸轮动作的初始行程量(StartStroke)为“25.0”的情况下

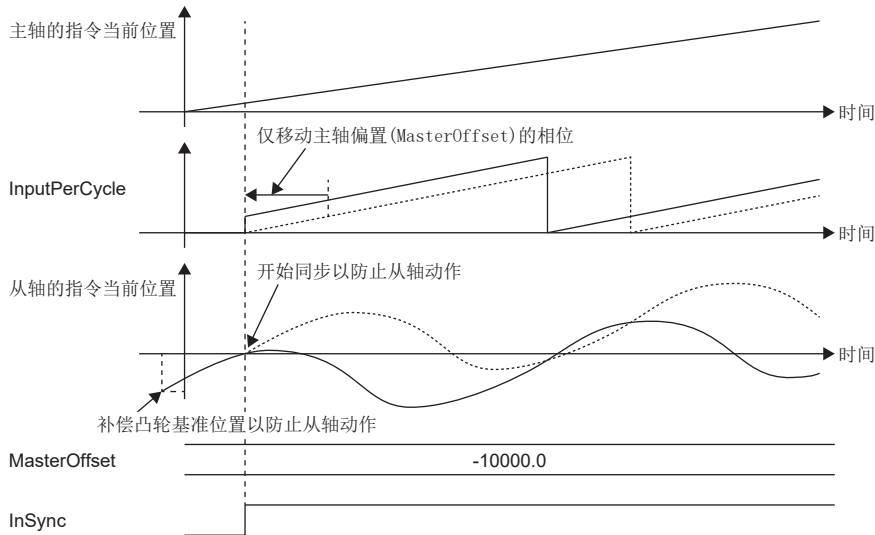


■ 主轴偏置 (MasterOffset)

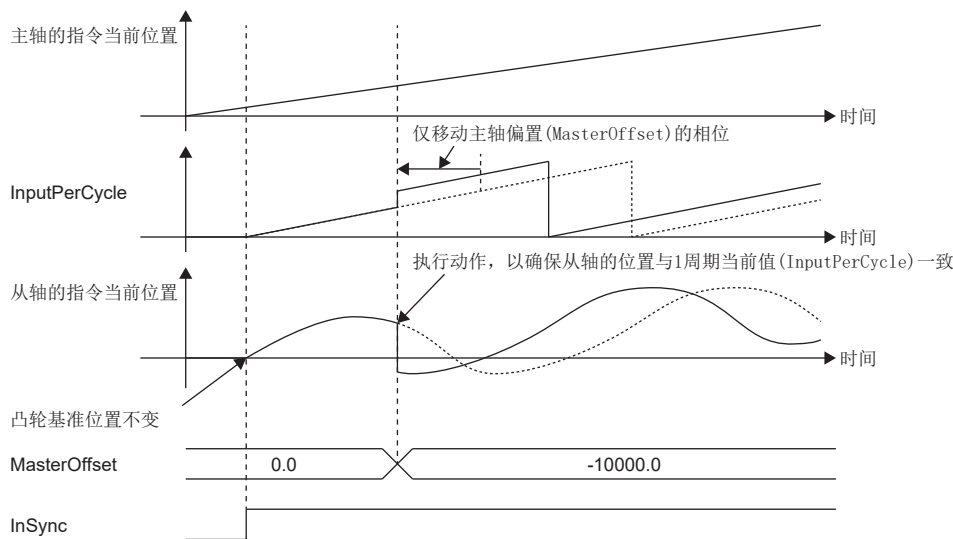
通过设置主轴偏置 (MasterOffset)，将主轴 (Master) 的相位按偏置量移动。(不影响主轴跟踪距离 (MasterStartDistance) 与主轴同步开始位置 (MasterSyncPosition)。)

动作开始时设置了“0.0”以外的情况下，同步中 (InSync) 的上升沿时，将变为在主轴 (Master) 的位置加上了偏置量后的凸轮1周期位置。此时，为了防止从轴 (Slave) 执行动作，对基准值 (Reference) 进行补偿。

- 开始模式 (StartMode) 为“0: 即时 (mcImmediate)”的情况下



- 开始模式 (StartMode) 为“0: 即时 (mcImmediate)”的情况下



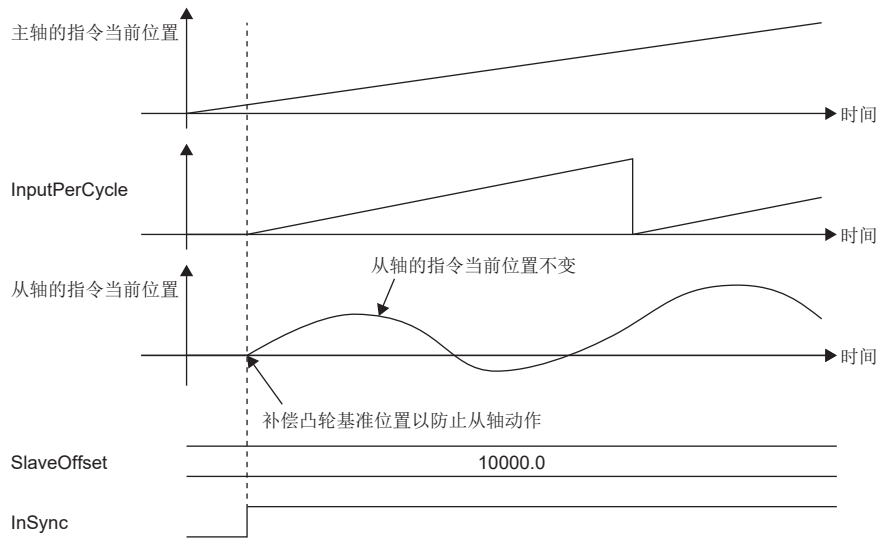
同步中 (InSync) 为 TRUE 时进行了更改的情况下，将执行动作以确保从轴 (Slave) 的位置与加上了主轴偏置 (MasterOffset) 的1周期当前值匹配。

■从轴偏置(SlaveOffset)

将从轴(Slave)的位移按偏置量移动。

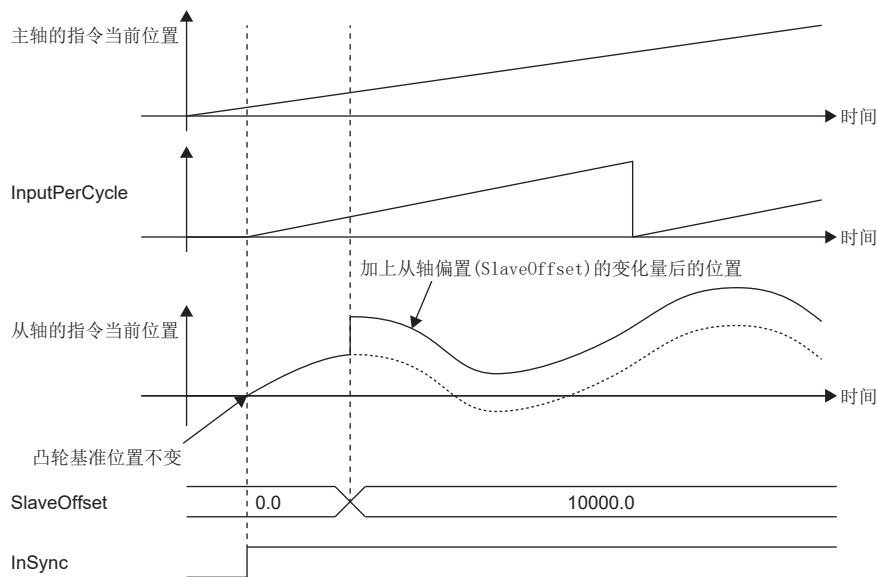
同步中(InSync)的上升沿前设置了“0.0”以外的情况下，为了防止在同步中(InSync)的上升沿时从轴(Slave)执行动作，对基准值(Reference)进行补偿。

- 开始模式(StartMode)为“0: 即时(mcImmediate)”的情况下



同步中(InSync)为TRUE时进行了更改的情况下，将执行动作以确保从轴(Slave)的位置与加上了偏置变化量的位置匹配。

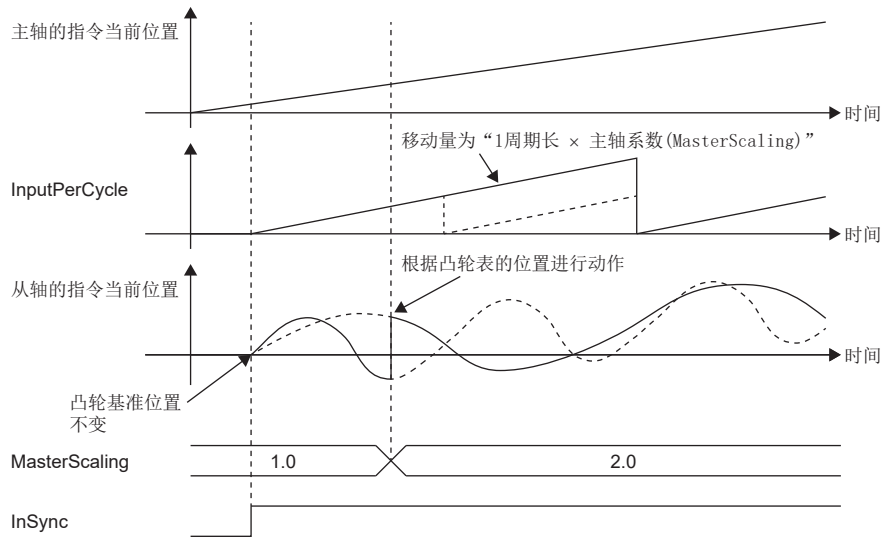
- 开始模式(StartMode)为“0: 即时(mcImmediate)”的情况下



■ 主轴系数 (MasterScaling)

对于主轴系数 (MasterScaling)，对凸轮表的1周期长进行放大、缩小。

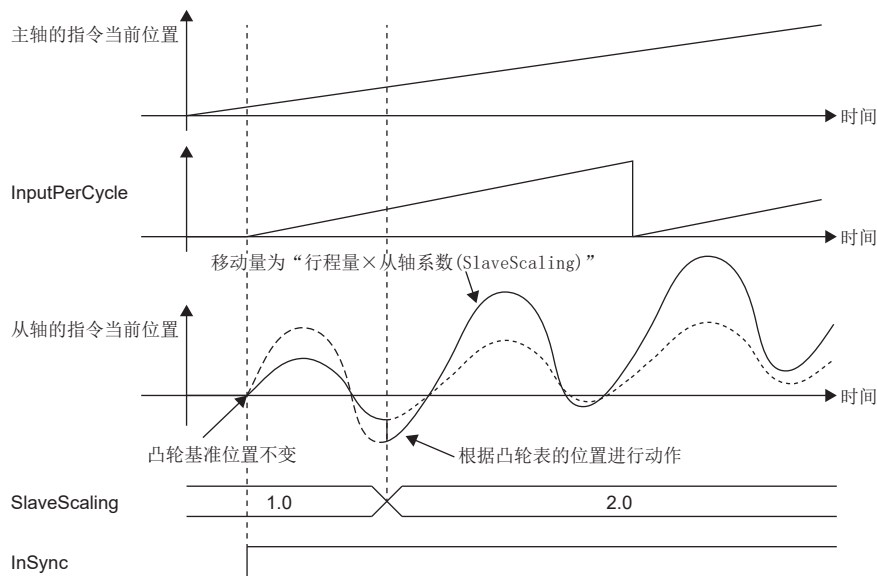
同步中进行了更改的情况下，将输出移动量的指令，以使从轴 (Slave) 的位置与凸轮表的位置匹配。



■ 从轴系数 (SlaveScaling)

对于从轴系数 (SlaveScaling)，对凸轮表的行程量进行放大、缩小。

同步中进行了更改的情况下，将输出移动量的指令，以使从轴 (Slave) 的位置与凸轮表的位置匹配。



■开始模式 (StartMode)

设置凸轮动作的同步时机及凸轮控制数据的反映时机。

关于凸轮动作的同步时机，请参阅下述章节。

📖 1719页 凸轮动作的开始

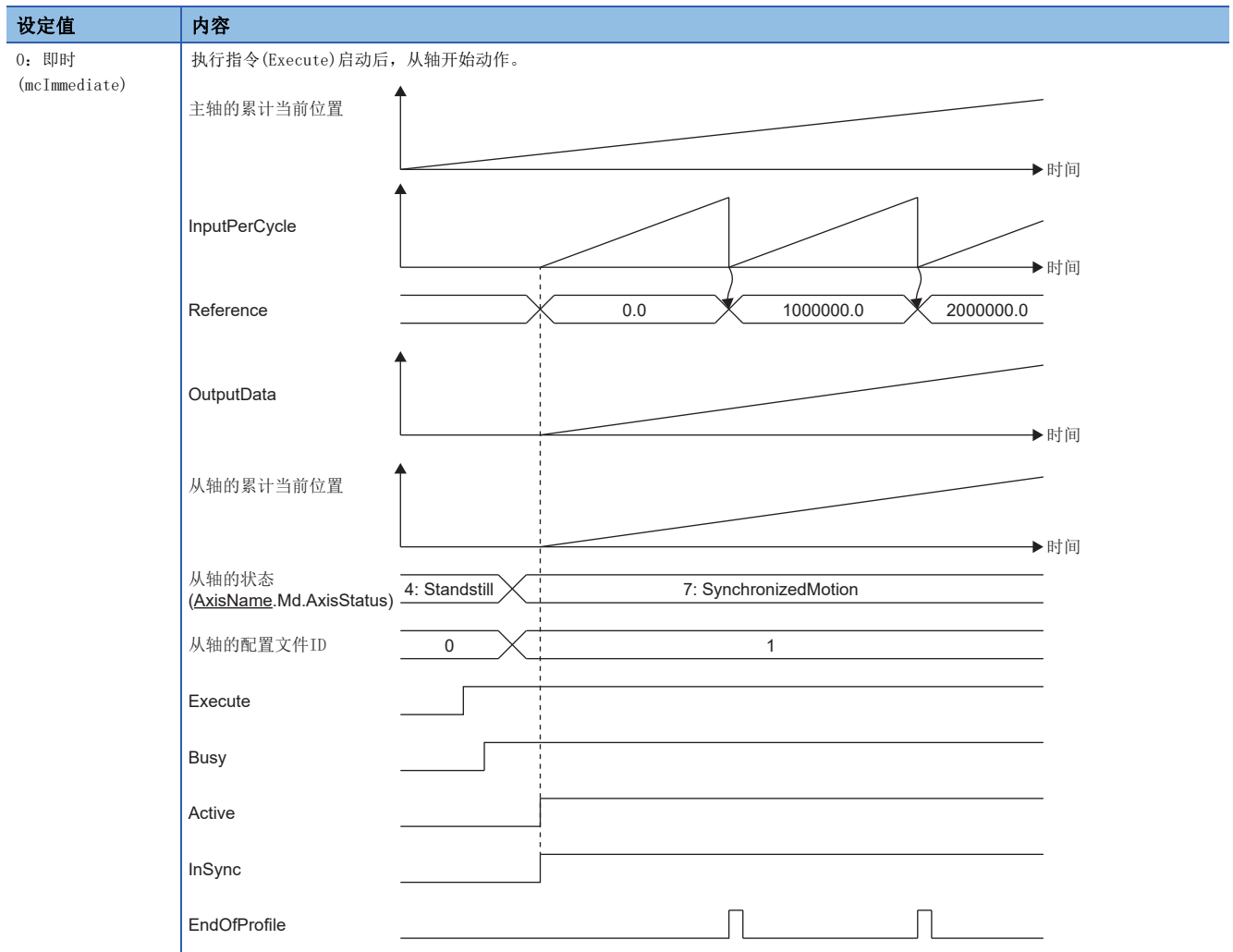
设置了设定值范围以外的情况下，将变为超出开始模式范围(错误代码: 1AB5H)。

执行MC_CamIn(凸轮动作开始)后，1周期当前值(InputPerCycle)同步时，控制中(Active)将变为TRUE，输出值(OutputData)同步时，同步中(InSync)将变为TRUE。

关于各设定值的使用示例，请参阅下述手册的“凸轮动作”。

📖 所使用的控制器的用户手册

根据从轴绝对坐标(SlaveAbsolute)的设置，动作如下所示。



设定值	内容
1: 绝对 (mcAbsolute)	<p>执行指令(Execute)启动后, 主轴的累计当前位置通过了主轴同步开始位置(MasterSyncPosition)后从轴开始动作。 主轴的累计当前位置是否通过了主轴同步开始位置(MasterSyncPosition)的检查从从轴的状态变为了“7: 同步运行中(SynchronizedMotion)”的时机开始。</p>

■ 主轴数据源选择 (MasterValueSource)

设置从轴 (Slave) 执行单轴同步控制的主轴 (Master) 的位置的类型。

根据主轴 (Master) 及从轴 (Slave) 的运算周期不相同的情况下及FB的执行顺序, 动作可能会改变。

将实轴以外的轴类型设置为主轴 (Master) 并在主轴数据源选择 (MasterValueSource) 中设置了反馈值的情况下, 将以与指令当前值相同的值执行动作。

设定值	内容
1: 指令当前值 (mcSetValue)	使用上次的运算周期中的主轴的指令位置。
2: 反馈值 (mcActualValue)	使用上次的运算周期中的主轴的反馈位置。
101: 最新指令当前值 (mcLatestSetValue)	使用本次的运算周期中的主轴的指令位置。
102: 最新反馈值 (mcLatestActualValue)	使用本次的运算周期中的主轴的反馈位置。

要点

在主轴数据源选择 (MasterValueSource) 中设置“1: 指令当前值 (mcSetValue)”、“101: 最新指令当前值 (mcLatestSetValue)”, 主轴由于伺服报警及紧急停止而变为了伺服OFF的情况下, 值的变化量有可能变大。通过在主轴数据源选择 (MasterValueSource) 中设置“2: 反馈值 (mcActualValue)”、“102: 最新反馈值 (mcLatestActualValue)”可以防止。

■缓冲模式(BufferMode)

设置用于进行多重启动(缓冲模式)的动作。

在MC_CamIn(凸轮动作开始)中,可以设置下述缓冲模式。

MC_CamIn(凸轮动作开始)的速度跟踪同步的FB(主轴),因此FB切换时根据主轴速度与设置的凸轮数据,速度即时变化。各模式的切换条件如下所示。

设定值	内容	切换条件
0: Aborting(mcAborting)	中断(取消)控制中的FB并立即执行下一个FB。	始终
1: Buffered(mcBuffered)	在控制中的FB中缓冲下一个FB。 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB完成时,依次执行缓冲FB。	凸轮循环完成(EndOfProfile)为TRUE时

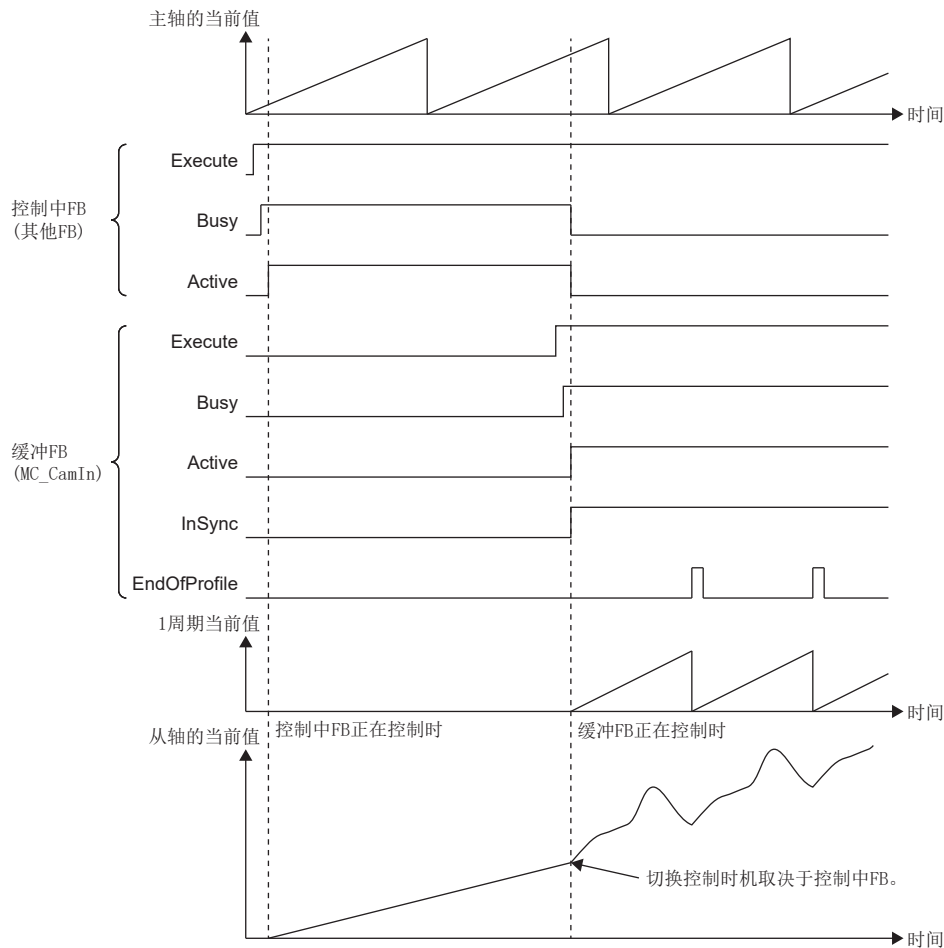
要点

关于多重启动(缓冲模式)的详细内容,请参阅下述手册。

所使用的控制器的用户手册

• 控制中FB与缓冲FB(MC_CamIn(凸轮动作开始))的情况下

如果控制中FB在控制MC_CamIn(凸轮动作开始)中指定的从轴的状态下启动MC_CamIn(凸轮动作开始),从轴的控制将切换为MC_CamIn(凸轮动作开始)。控制切换的时机取决于控制中FB与MC_CamIn(凸轮动作开始)中设置的缓冲模式(BufferMode)。

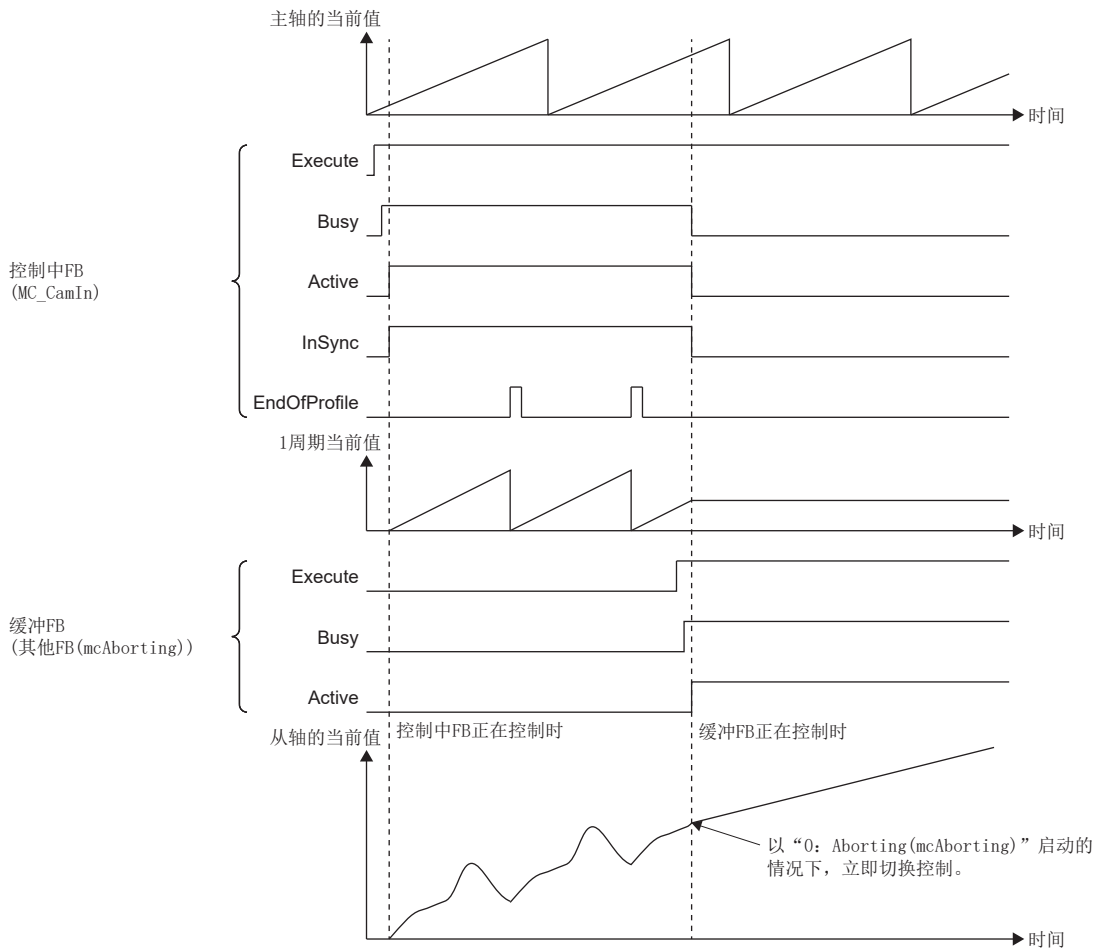


要点

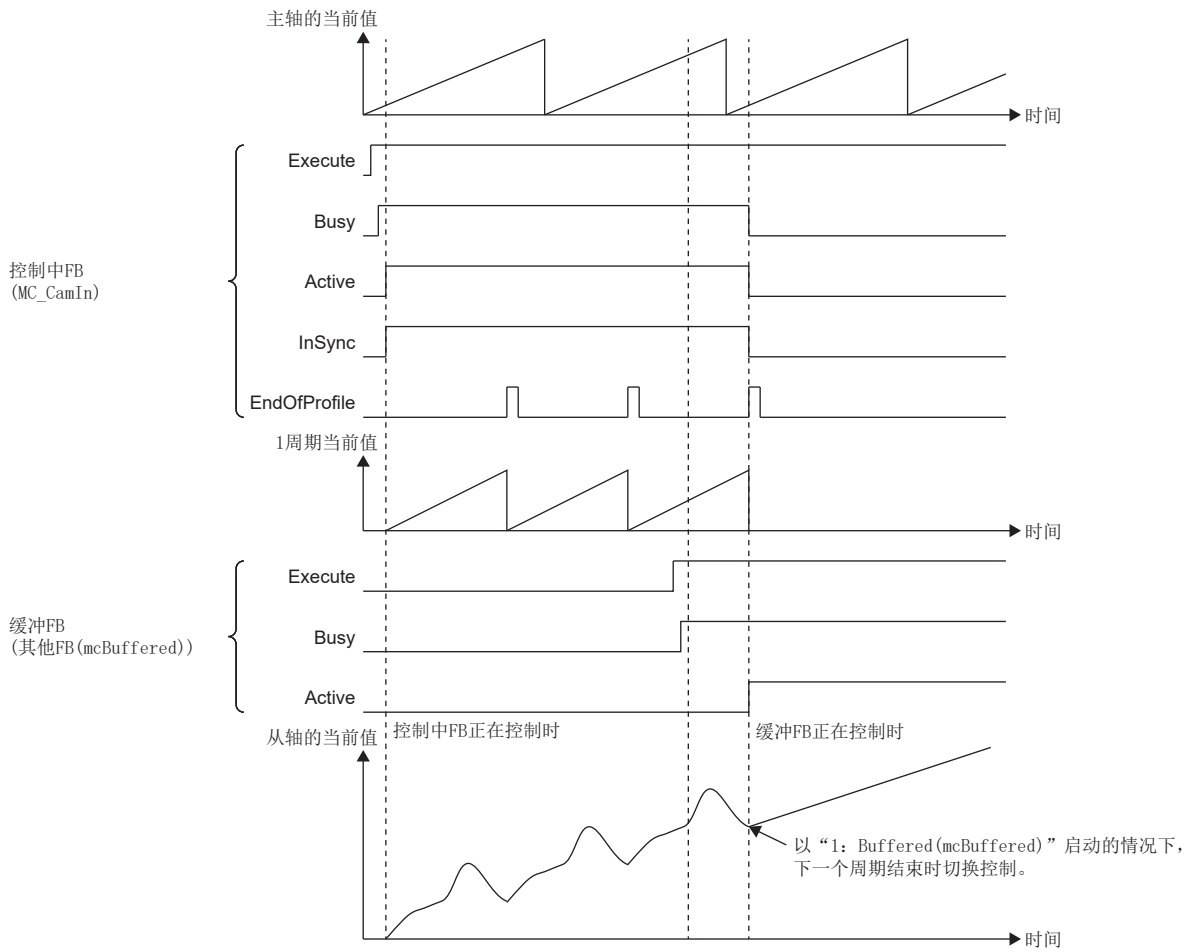
缓冲FB(MC_CamIn(凸轮动作开始))的速度,在同步中(InSync)变为TRUE之后发生。

开始模式(StartMode)为“0:即时(mcImmediate)”以外的情况下,FB启动后同步中(InSync)不会立即变为TRUE,因此在执行中(Busy)、控制中(Active)及同步中(InSync)全部变为TRUE为止的期间速度将为0。

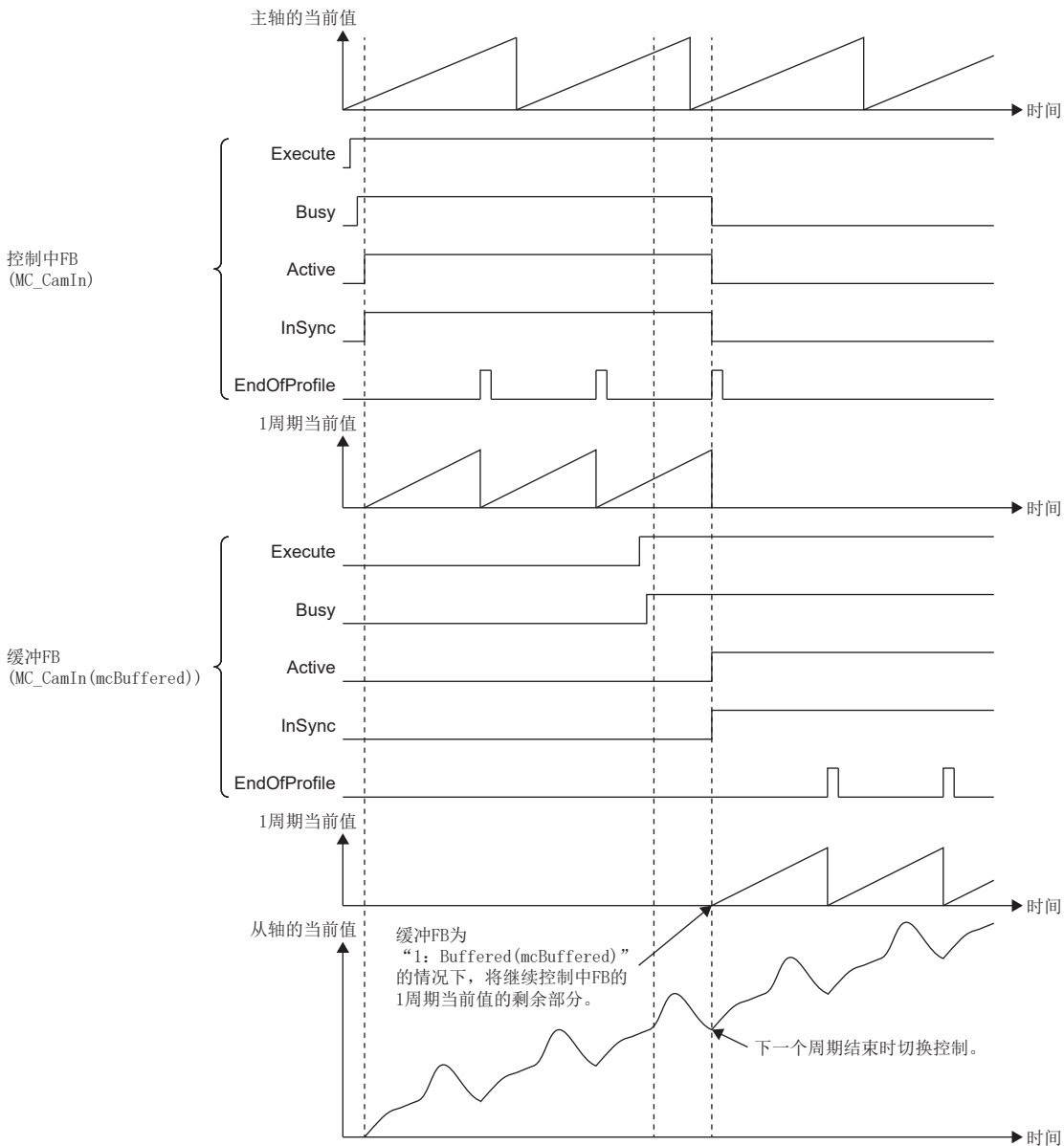
- 控制中FB(MC_CamIn(凸轮动作开始))与缓冲FB(缓冲模式(BufferMode)为“0: Aborting(mcAborting)”)的情况下在MC_CamIn(凸轮动作开始)控制从轴的状态下,对MC_CamIn(凸轮动作开始)中指定的从轴启动其他FB时,从轴的控制将切换为其他FB。控制切换的时机取决于其他FB的缓冲模式(BufferMode),指定为“0: Aborting(mcAborting)”时,立即切换控制。



- 控制中FB(MC_CamIn(凸轮动作开始))与缓冲FB(缓冲模式(BufferMode)为“1: Buffered(mcBuffered)”)的情况下在MC_CamIn(凸轮动作开始)控制从轴的状态下,对MC_CamIn(凸轮动作开始)中指定的从轴启动其他FB时,从轴的控制将切换为其他FB。控制切换的时机取决于其他FB的缓冲模式(BufferMode),指定为“1: Buffered(mcBuffered)”时,下一个周期结束时切换控制。



- 控制中FB(MC_CamIn(凸轮动作开始))与缓冲FB(MC_CamIn(凸轮动作开始))的情况下
在MC_CamIn(凸轮动作开始)控制从轴的状态下,启动其他的MC_CamIn(凸轮动作开始)时,将切换从轴的控制。其他的MC_CamIn(凸轮动作开始)的缓冲模式(BufferMode)为“1: Buffered(mcBuffered)”时,调整开始点的位置,使凸轮数据的结束点与开始点相同。



■选项 (Options)

将MC_CamIn (凸轮动作开始) 中使用的功能选项以位指定进行设置。

以位指定设置的内容如下所示。

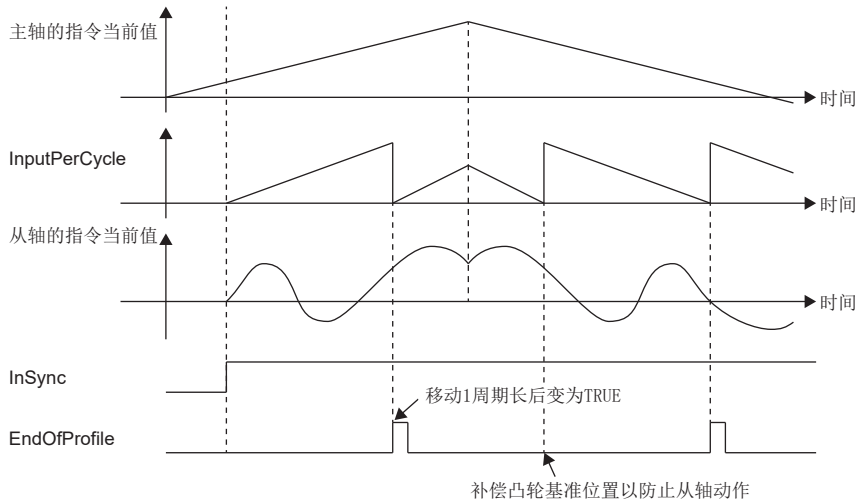


*1 空余中应设置“0”。设置了“0”以外的情况下，将变为超出Options范围 (错误代码: 1ABBH)。

位	名称	内容
16	1周期运行后的动作指定	设置在MC_CamTableSelect (凸轮表选择) 的重复动作 (Periodic) 中设置了“FALSE: 单发动作”时的1周期运行后的动作。 <ul style="list-style-type: none"> • 0: 结束 • 1: 重启等待
21	主轴同步开始位置的通过检查对象指定	设置主轴同步开始位置 (MasterSyncPosition) 的通过检查的对象。 <ul style="list-style-type: none"> • 0: 对主轴的累计当前位置或反馈位置进行了累计位置换算后的位置 • 1: 主轴的指令当前位置或反馈位置

■凸轮循环完成 (EndOfProfile)

控制中 (Active) 变为TRUE后，每次移动1周期长时，仅调用FB的POU (程序部件) 的执行周期的1个周期，将变为TRUE。



■1周期当前值 (InputPerCycle)

初始值将变为“0.0”。

控制中 (Active) 变为TRUE后，主轴 (Master) 的移动量如下所示反映。

$$1\text{周期当前值} = (\text{主轴的累计移动量} - \text{主轴偏置} + 1\text{周期当前值更改值}) \text{ MOD } 1\text{周期长}$$

监视值的范围为凸轮表中设置的1周期长的下述范围。

$$0 \leq \text{MC_CamIn (凸轮动作开始) 的1周期当前值 (InputPerCycle)} < (1\text{周期长} \times \text{主轴系数})$$

■基准值 (Reference)

执行MC_CamIn (凸轮动作开始)时,以同步中(InSync)变为TRUE的指令当前位置为基准,从要开始的凸轮位置(1周期当前值)进行计算。

- 由于减去了凸轮表中的初始行程量,因此初始值的计算公式如下所示。

从轴绝对坐标(SlaveAbsolute)	计算公式
FALSE(相对坐标)	基准值 = 同步开始时的从轴位置 - (同步中(InSync)变为了TRUE时的基于凸轮位置(1周期当前值)的行程量 + 初始行程量) × 从轴系数
TRUE(绝对坐标)	基准值 = 与同步中(InSync)变为了TRUE时的指令当前位置=0相当的累计当前位置

- 往复凸轮动作及进给凸轮动作时的基准值如下所示。

动作	内容
往复凸轮动作或从轴绝对坐标(SlaveAbsolute)为“TRUE(绝对坐标)”时	不更新基准值。
进给凸轮动作 且 从轴绝对坐标(SlaveAbsolute)为“FALSE(相对坐标)”时	<p>按下述公式计算基准值。</p> <ul style="list-style-type: none"> • 基准值 = (原始基准值 + (结束点的行程值 - 第1点的行程值) × 从轴系数)

- 基准值在下述时机更新。

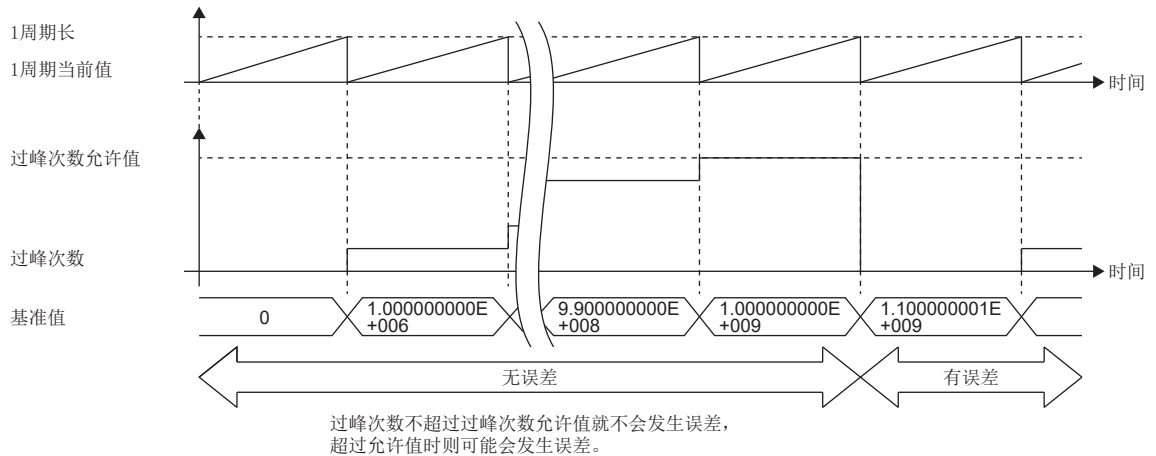
更新时机	基准值的计算公式
1周期当前值(InputPerCycle)向地址增加方向通过了凸轮表的结束点时	原始基准值 + ((结束点的行程值 - 第1点的行程值) × 从轴系数)
1周期当前值(InputPerCycle)向地址减少方向通过了凸轮表的第1点时	原始基准值 - ((结束点的行程值 - 第1点的行程值) × 从轴系数)
更改了1周期当前值时(MCv_ChangeCycle(1周期当前值更改)的完成(Done)为TRUE时)	原始基准值 - (与更改后的1周期当前值相当的行程值 × 从轴系数)

<关于基准值的误差>

执行下述动作之一时可能会导致基准值中发生误差。

- 进给凸轮且同步中(InSync)变为了TRUE后, 凸轮循环完成(EndOfProfile)的TRUE的输出次数超出了过峰次数允许值时。^{*1}
- 进给凸轮且下述控制参数中包含了小数点时。
 - (结束点的行程值 - 第1点的行程值)
 - 从轴系数
- 以相对指定重复了1周期当前值更改时。

*1 虽然过峰次数不超过过峰次数允许值不发生误差, 但是超过过峰次数允许值时则可能会发生误差。(与运算误差的有无无关, 将继续动作。)



基准值的误差变大的情况下, 应解除同步状态或执行绝对位置指定的当前值更改以进行复位。

过峰次数允许值为通过下述计算公式计算的可变的值。

$$\text{过峰次数允许值} = \text{过峰次数极限值} / ((\text{结束点的输出值} - \text{第1点的输出值}) / \text{从轴系数})$$

*: 过峰次数极限值为定位范围上限值“1000000000”。

(例)

将第1点的输出值设置为“0”, 将结束点的输出值设置为“1000000”, 将输出系数设置为“10”的情况下过峰次数允许值将为“100000”。

监视值的范围如下所示。

- 从轴的环形计数器下限值 ≤ 基准值 < 从轴的环形计数器上限值

■输出值(OutputData)

初始值为凸轮表中的初始行程量。

输出值(OutputData)的更新时机及计算方法如下所示。

更新时机	输出值的计算方法
控制中(Active)为TRUE, 同步中(InSync)为FALSE时	基准值 + (凸轮表中的初始行程量 + 从轴偏置) × 从轴系数 开始模式(StartMode)为斜坡逼近(mcRampIn)的情况下, 从上述位置向同步开始时的从轴位置计算输出值。详细内容, 请参阅下述章节。 ☞ 1730页 开始模式(StartMode)
同步中(InSync)为TRUE时	■插补方法为对区间分别指定或样条插补时 基准值 + ((行程量 × 与1周期当前值对应的行程比) + 从轴偏置) × 从轴系数 ■凸轮数据格式为直线插补时 基准值 + ((与1周期当前值对应的输出值) + 从轴偏置) × 从轴系数

注意事项

- 更改主轴偏置(MasterOffset)、从轴偏置(SlaveOffset)、主轴系数(MasterScaling)、从轴系数(SlaveScaling)、凸轮表ID(CamTableID)时, 控制开始时及控制更改时从轴将急剧移动, 可能会给设备带来冲击。应充分考虑设定值及更改时机。
- 通过工程工具监视凸轮控制中的轴时的位置、速度等的单位符合轴的单位设置。不使用运算配置文件中设置的1周期长及行程量的单位。

程序示例

将凸轮动作开始指令(bCamInStart)置为TRUE, 将要通过JOG运行启动的轴2(Axis0002)设置为主轴, 并根据下述设置按照运算配置文件(凸轮数据1)进行从轴(Axis0001)的凸轮动作的程序示例如下所示。

• 设置

项目	设定值	
JOG运行用数据设置	速度	1000.0
	加速度	5000.0
	减速度	5000.0
	Jerk	0.0

■轴

轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1
2	Axis0002	AXIS_REF	轴2

■运算配置文件

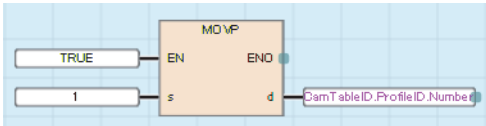
配置文件ID	标签名	数据类型	注释
1	ProfileData0001	MC_CAM_REF	凸轮数据1

■使用的标签

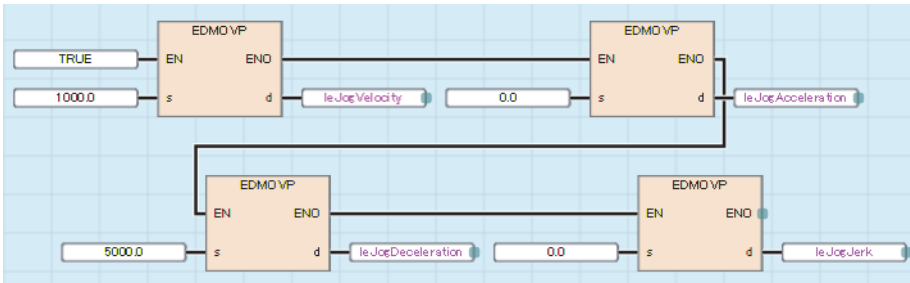
标签名	数据类型	注释
MC_CamIn_1	MC_CamIn	凸轮动作开始FB
bCamInStart	位	凸轮动作开始指令
CamTableID	MC_CAM_ID	凸轮表ID
bInSync	位	同步中
bCamBusy	位	执行中
bCamActive	位	控制中
bCamCommandAborted	位	执行中断
bCamError	位	错误
uwCamErrorID	字[无符号]/位串[16位]	错误代码
bEndOfProfile	位	凸轮循环完成
MCv_Jog_1	MCv_Jog	JOG运行FB
leJogVelocity	双精度实数	JOG速度
leJogAcceleration	双精度实数	JOG加速度
leJogDeceleration	双精度实数	JOG减速度
leJogJerk	双精度实数	JOG Jerk
bJogDone	位	执行完成
bJogBusy	位	执行中
bJogActive	位	控制中
bJogCommandAborted	位	执行中断
bJogError	位	错误
uwJogErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

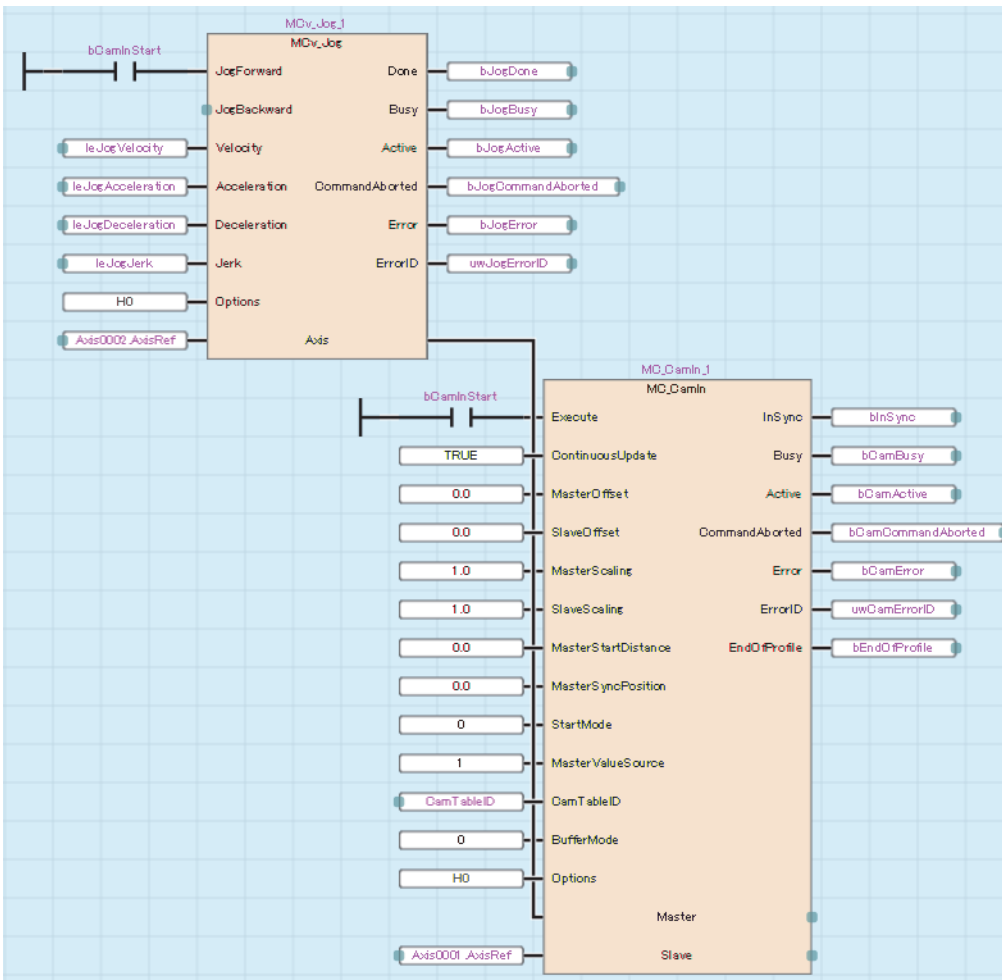
- 运算配置文件(凸轮数据)设置



- JOG运行用数据设置



- JOG运行/凸轮动作开始



■ST程序

```
//-----运算配置文件(凸轮数据)设置-----
CamTableID.ProfileID.Number:= 1;

//-----JOG运行用数据设置-----
leJogVelocity:= 1000.0;
leJogAcceleration:= 5000.0;
leJogDeceleration:= 5000.0;
leJogJerk:= 0.0;

//-----JOG运行-----
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
);

//-----凸轮动作开始-----
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=> bInSync ,
  Busy=> bCamBusy ,
  Active=> bCamActive ,
  CommandAborted=> bCamCommandAborted ,
  Error=> bCamError ,
  ErrorID=> uwCamErrorID ,
  EndOfProfile=> bEndOfProfile
);
```

46.15 齿轮动作开始

MC_GearIn

按照指定的齿轮比开始齿轮动作。

梯形图	FBD/LD	ST
		<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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Master	主轴	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName.AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName.AxisRef.(轴信息)
Slave	从轴	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName.AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName.AxisRef.(轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_GearIn(齿轮动作开始)。
ContinuousUpdate	连续更新	BOOL	启动时	TRUE、FALSE	FALSE	设置将齿轮比分子(RatioNumerator)、齿轮比分母(RatioDenominator)、加速度(Acceleration)、减速度(Deceleration)的连续更改是置为有效，还是置为无效。 • FALSE: 无效 • TRUE: 有效
RatioNumerator	齿轮比分子	DINT	启动时/可重启/可连续更新	-2147483648~2147483647	1	设置主轴(Master)的分子的值。 关于详细内容，请参阅下述章节。 ☞ 1747页 齿轮比分子(RatioNumerator)/齿轮比分母(RatioDenominator)
RatioDenominator	齿轮比分母	DWORD(UDINT)	启动时/可重启/可连续更新	1~2147483647	1	设置主轴(Master)的分母的值。 关于详细内容，请参阅下述章节。 ☞ 1747页 齿轮比分子(RatioNumerator)/齿轮比分母(RatioDenominator)

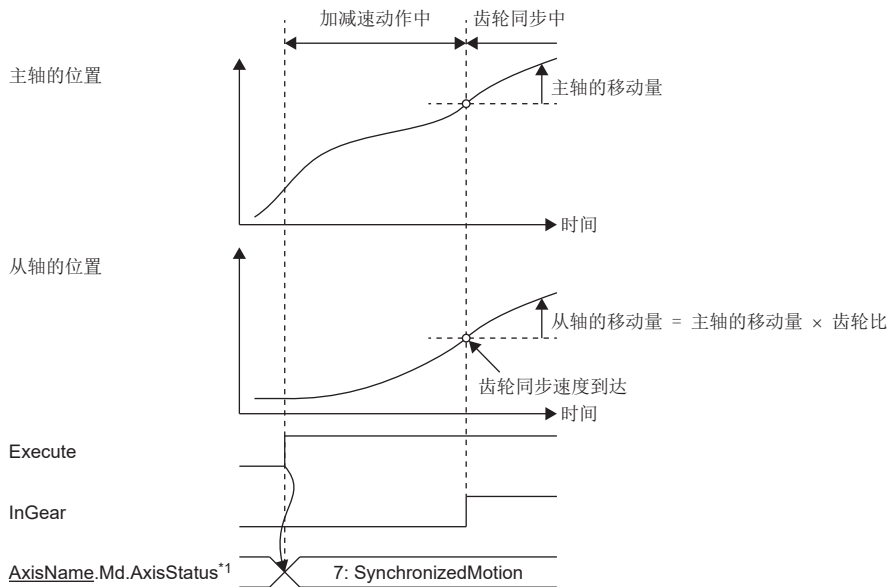
输入变量	名称	数据类型	获取	有效范围	初始值	内容
MasterValueSource	主轴数据源选择	INT (MC_SOURCE)	启动时	1、2、101、102	1	设置主轴(Master)的数据源。 <ul style="list-style-type: none"> • 1: 指令当前值(mcSetValue) • 2: 反馈值(mcActualValue) • 101: 最新指令当前值(mcLatestSetValue) • 102: 最新反馈值(mcLatestActualValue) 关于详细内容, 请参阅下述章节。 1748页 主轴数据源选择 (MasterValueSource)
Acceleration	加速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置加速动作的加速度。 到达齿轮同步速度时齿轮比到达(InGear)将变为TRUE, 且从轴(Slave)将以对主轴(Master)的速度进行了齿轮比换算后的速度进行控制。
Deceleration	减速度	LREAL	启动时/可重启/可连续更新	0.0000、0.0001~2147483647.0	0.0	设置减速动作的减速度。 到达齿轮同步速度时齿轮比到达(InGear)将变为TRUE, 且从轴(Slave)将以对主轴(Master)的速度进行了齿轮比换算后的速度进行控制。
Jerk	Jerk	LREAL	启动时	0.0000、0.0001~2147483647.0	0.0	设置加减速度开始时的Jerk。
BufferMode	缓冲模式	INT (MC_BUFFER_MODE)	启动时	0、1	0	设置缓冲模式。 <ul style="list-style-type: none"> • 0: Aborting(mcAborting) • 1: Buffered(mcBuffered) 关于详细内容, 请参阅下述章节。 1748页 缓冲模式(BufferMode)
Options	选项	DWORD (HEX)	启动时	0000000H	0000000H	应设置“0000000H”。 * 设置了“0000000H”以外的情况下, 将变为超出Options范围(错误代码: 1ABBH)。

■输出变量

输出变量	名称	数据类型	初始值	内容
InGear	齿轮比到达	BOOL	FALSE	到达了齿轮同步速度时, 将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_GearIn(齿轮动作开始)时, 将变为TRUE。
Active	控制中	BOOL	FALSE	正在控制从轴(Slave)时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MC_GearIn(齿轮动作开始)的执行中断时, 将变为TRUE。 执行指令(Execute)变为FALSE时, 将变为FALSE。
Error	错误	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时, 将返回错误代码。 关于错误代码的详细内容, 请参阅下述手册。 所使用的控制器的用户手册

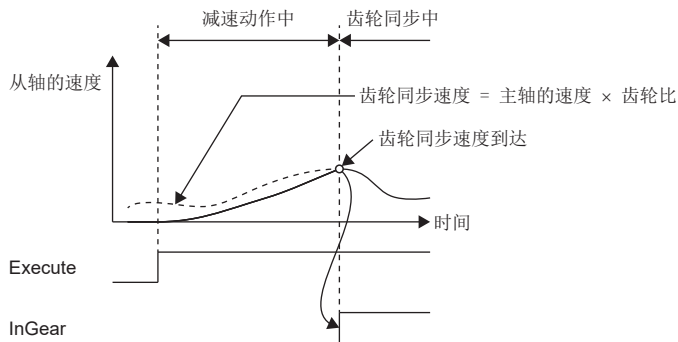
功能

- 对于MC_GearIn(齿轮动作开始)，设置齿轮比分子(RatioNumerator)、齿轮比分母(RatioDenominator)、主轴数据源选择(MasterValueSource)、加速度(Acceleration)、减速度(Deceleration)、Jerk、缓冲模式(BufferMode)，开始齿轮动作。
- 结束动作的情况下，通过MC_Stop(强制停止)进行。
- FB执行后，从轴(Slave)将对主轴(Master)的速度进行了齿轮比换算后的值作为齿轮同步速度，并进行加减速动作直至到达齿轮同步速度为止。齿轮同步速度到达后，齿轮比到达(InGear)将变为TRUE，且从轴(Slave)将以对主轴(Master)的速度进行了齿轮比换算后的速度进行控制。



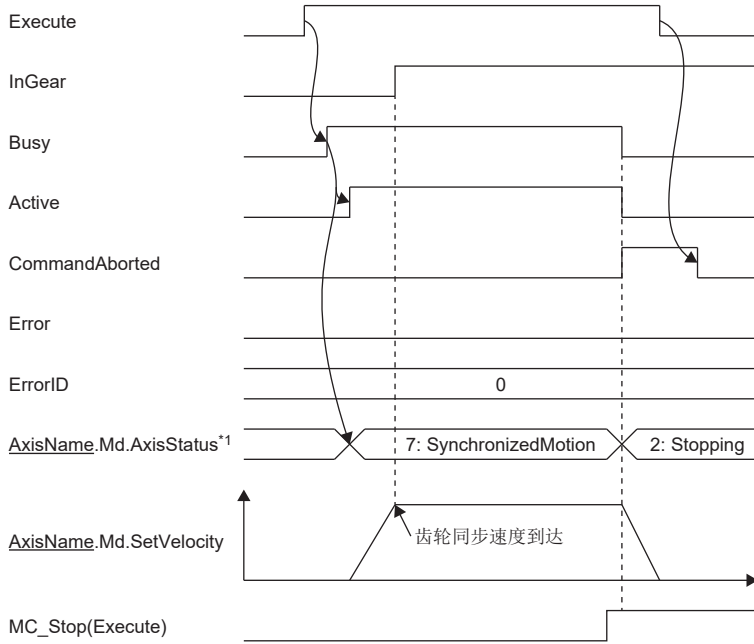
*1 是从轴(Slave)的轴状态。

- 加减速动作中主轴(Master)的速度变动的情况下，齿轮同步速度也将被更新。



■时序图

- 正常完成的情况下



*1 是从轴 (Slave) 的轴状态。

- 异常完成的情况下

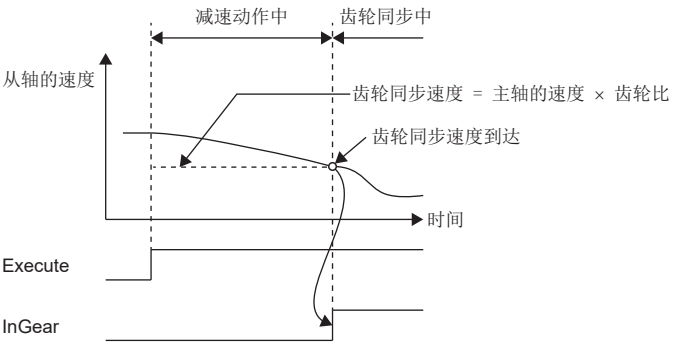
关于异常完成时的时序图，请参阅下述章节。

☞ 1375页 通过执行指令 (Execute) 类型的运动控制FB的基本动作

■直至到达齿轮同步速度为止的加减速动作

- MC_GearIn (齿轮动作开始) 的加减速方式为加减速速度指定方式。
- 根据动作开始时的从轴 (Slave) 的速度与齿轮同步速度，进行下述4个模式的加减速动作。即使在加减速动作中更新齿轮同步速度，也不更改加减速动作的模式。

从轴 (Slave) 的速度及目标速度	加减速动作
从轴 (Slave) 的速度 $< 0 <$ 齿轮同步速度 或 从轴 (Slave) 的速度 $> 0 >$ 齿轮同步速度	进行减速动作直到速度0，并从速度0进行加速动作直到齿轮同步速度。
从轴 (Slave) 的速度 (绝对值) $<$ 齿轮同步速度 (绝对值)	进行加速动作直到齿轮同步速度。

从轴(Slave)的速度及目标速度	加减速动作
<p>从轴(Slave)的速度(绝对值) > 齿轮同步速度(绝对值)</p>	<p>进行减速动作直到齿轮同步速度。</p> 
<p>从轴(Slave)的速度 = 齿轮同步速度</p>	<p>不进行加减速动作。 动作开始后, 齿轮比到达(InGear)将立即变为TRUE, 且从轴(Slave)将以对主轴(Master)的速度进行了齿轮比换算后的速度进行控制。</p>

■ 齿轮比分子 (RatioNumerator) / 齿轮比分母 (RatioDenominator)

齿轮比分子 (RatioNumerator) 与齿轮比分母 (RatioDenominator) 组合进行设置。

齿轮比分子 (RatioNumerator) 设置对主轴 (Master) 的速度进行转换及传送时的分子的值。

齿轮比分母 (RatioDenominator) 设置对主轴 (Master) 的移动量进行转换及合成时的分母的值。

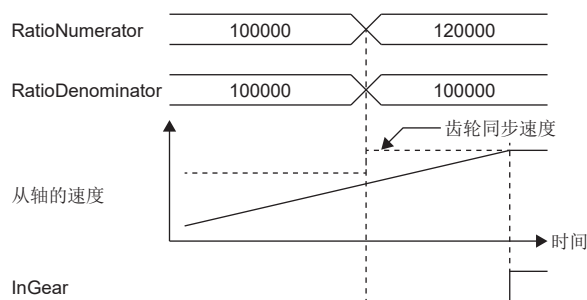
按下述方式转换速度。

$$\text{转换后的主轴 (Master) 速度} = \text{转换前的主轴 (Master) 速度} \times \frac{\text{齿轮比分子 (RatioNumerator)}}{\text{齿轮比分母 (RatioDenominator)}}$$

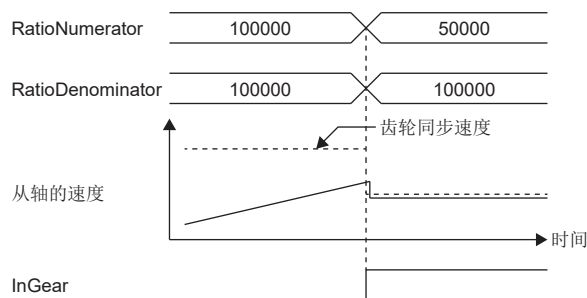
- 齿轮比分母 (RatioDenominator) 中应设置正的值。
- 通过在齿轮比分子 (RatioNumerator) 的设定值中设置负的值，可以反转速度后进行传送。加减速动作中更改了齿轮比的情况下，齿轮同步速度将变化。因此，根据齿轮同步速度的变化，从轴 (Slave) 的速度有可能直接变化并开始齿轮动作。
- 通过在齿轮比分子 (RatioNumerator) 的设定值中设置“0”，可以将转换后的主轴 (Master) 的速度设置为“0”。
- 加减速动作中更改了齿轮比的情况下，目标速度将变化。因此，根据目标速度的变化，从轴 (Slave) 的速度有可能直接变化并开始齿轮动作。

• 加减速动作中的齿轮比更改

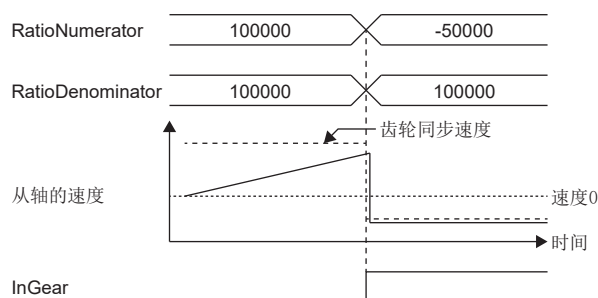
- 齿轮比增加 (将齿轮同步速度设置为高于从轴 (Slave) 的速度的值)



- 齿轮比减少 (将齿轮同步速度设置为低于从轴 (Slave) 的速度的值)

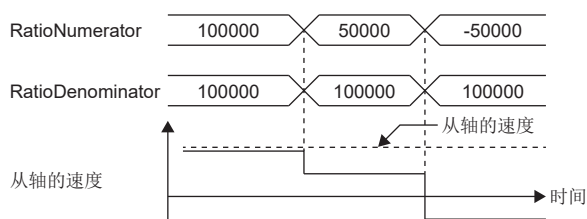


- 齿轮比反转 (将齿轮同步速度设置为低于“0”的值)



- 齿轮同步中更改了齿轮比的情况下，从轴 (Slave) 的速度将直接变化。

• 齿轮同步中的齿轮比更改



■ 主轴数据源选择 (MasterValueSource)

设置从轴 (Slave) 执行单轴同步控制的主轴 (Master) 的位置的类型。

根据主轴 (Master) 及从轴 (Slave) 的运算周期不相同的情况下及FB的执行顺序，动作可能会改变。

将实轴以外的轴类型设置为主轴 (Master) 并在主轴数据源选择 (MasterValueSource) 中设置了反馈值的情况下，将以与指令当前值相同的值执行动作。

设定值	内容
1: 指令当前值 (mcSetValue)	使用上次的运算周期中的主轴的指令位置。
2: 反馈值 (mcActualValue)	使用上次的运算周期中的主轴的反馈位置。
101: 最新指令当前值 (mcLatestSetValue)	使用本次的运算周期中的主轴的指令位置。
102: 最新反馈值 (mcLatestActualValue)	使用本次的运算周期中的主轴的反馈位置。

要点

在主轴数据源选择 (MasterValueSource) 中设置 “1: 指令当前值 (mcSetValue)”、“101: 最新指令当前值 (mcLatestSetValue)”，主轴由于伺服报警及紧急停止而变为了伺服OFF的情况下，值的变化量有可能变大。通过在主轴数据源选择 (MasterValueSource) 中设置 “2: 反馈值 (mcActualValue)”、“102: 最新反馈值 (mcLatestActualValue)” 可以防止。

■ 缓冲模式 (BufferMode)


设置用于进行多重启动 (缓冲模式) 的动作。

在MC_GearIn (齿轮动作开始) 中，可以设置下述缓冲模式。

设定值	内容
0: Aborting (mcAborting)	中断 (取消) 控制中的FB并立即执行下一个FB。
1: Buffered (mcBuffered)	在控制中的FB中缓冲下一个FB。 控制中的FB已经被缓冲的情况下，对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB完成时，依次执行缓冲FB。

要点

关于多重启动 (缓冲模式) 的详细内容，请参阅下述手册。

 所使用的控制器的用户手册

注意事项

- 更改了齿轮比的情况下，从轴的速度将直接变化。希望对速度变化进行平滑处理的情况下，应与MCv_SmoothingFilter (平滑滤波器) 一起使用。
- 加减速动作中转换后的主轴 (Master) 的速度超出加减速时间上限值的情况下，将变为加速时间限制溢出警告 (事件代码: 00D04H) 或减速时间限制溢出警告 (事件代码: 00D05H)，停止加减速动作，并以警告检测时的速度继续动作。发生上述警告的情况下，应调整转换后的主轴 (Master) 的速度、加速度 (Acceleration) 或减速度 (Deceleration)，设置为使其不超过加减速时间上限值。

程序示例

将齿轮动作开始指令(bGearInStart)置为TRUE, 将要通过JOG运行启动的轴1(Axis0001)设置为主轴, 并根据下述设置按照指定的齿轮比进行从轴(Axis0002)的齿轮动作的程序示例如下所示。

• 设置

项目		设定值
齿轮动作数据设置	加速度	5000.0
	减速度	5000.0
	Jerk	0.0
JOG运行数据设置	速度	1000.0
	加速度	5000.0
	减速度	5000.0
	Jerk	0.0

■ 轴

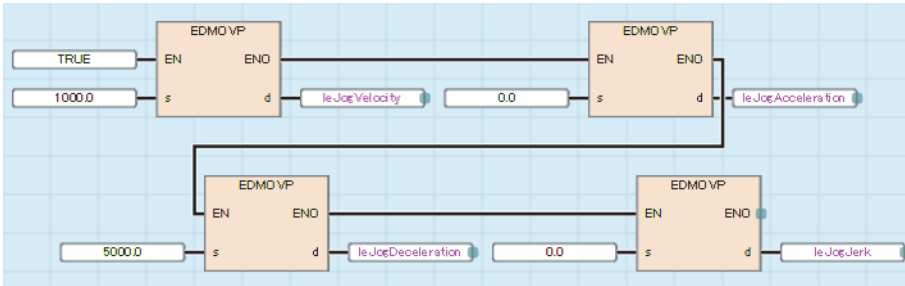
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1
2	Axis0002	AXIS_REF	轴2

■ 使用的标签

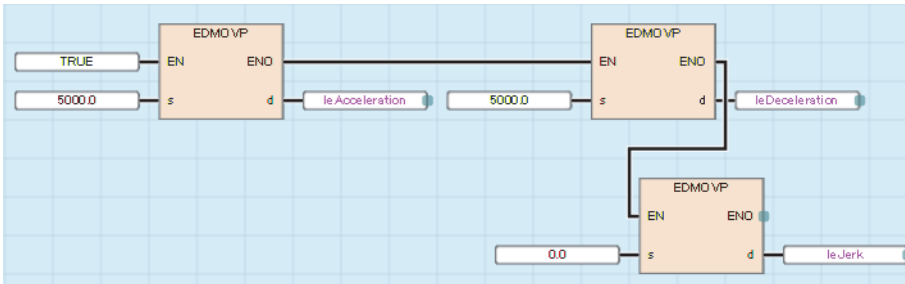
标签名	数据类型	注释
MC_GearIn_1	MC_GearIn	齿轮动作开始FB
bGearInStart	位	齿轮动作开始指令
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bInGear	位	同步中
bGearBusy	位	执行中
bGearActive	位	控制中
bGearCommandAborted	位	执行中断
bGearError	位	错误
uwGearErrorID	字[无符号]/位串[16位]	错误代码
MCv_Jog_1	MCv_Jog	JOG运行FB
leJogVelocity	双精度实数	JOG速度
leJogAcceleration	双精度实数	JOG加速度
leJogDeceleration	双精度实数	JOG减速度
leJogJerk	双精度实数	JOG Jerk
bJogDone	位	执行完成
bJogActive	位	控制中
bJogCommandAborted	位	执行中断
bJogError	位	错误
uwJogErrorID	字[无符号]/位串[16位]	错误代码

■FBD/LD程序

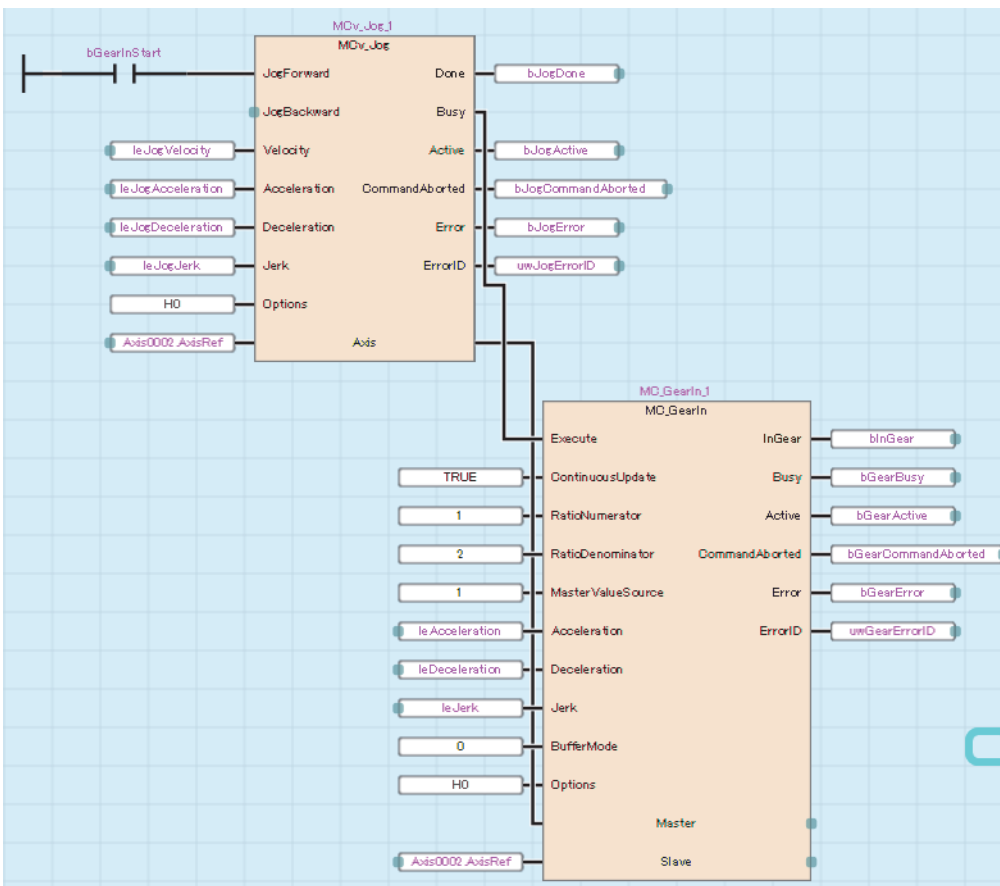
- JOG运行用数据的设置



- 齿轮动作数据设置



- JOG运行/齿轮动作开始



■ST程序

```
//-----JOG运行用数据的设置-----
leJogVelocity:= 1000.0;
leJogAcceleration:= 5000.0;
leJogDeceleration:= 5000.0;
leJogJerk:= 0.0;

//-----齿轮动作数据设置-----
leAcceleration:= 5000.0;
leDeceleration:= 5000.0;
leJerk:= 0.0;

//-----JOG运行-----
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
);

//-----齿轮动作开始-----
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
);
```

46.16 加减法定位

MC_CombineAxes

将对指定的主轴2轴的移动量进行了加减法计算的值作为指令位置进行定位。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Master1	主轴1	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName, AxisRef. (轴信息)
Master2	主轴2	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName, AxisRef. (轴信息)
Slave	从轴	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName, AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_CombineAxes(加减法定位)。
ContinuousUpdate	连续更新	BOOL	启动时	TRUE、FALSE	FALSE	设置将加减法方法选择(CombineMode)、主轴1齿轮比分子(GearRatioNumeratorM1)、主轴1齿轮比分母(GearRatioDenominatorM1)、主轴2齿轮比分子(GearRatioNumeratorM2)、主轴2齿轮比分母(GearRatioDenominatorM2)的连续更改是置为有效,还是置为无效。 <ul style="list-style-type: none"> FALSE: 无效 TRUE: 有效
CombineMode	加减法方法选择	INT (MC_COMBINE_MODE)	启动时/可重启/可连续更新	0、1	1	设置主轴1(Master1)与主轴2(Master2)的移动量的合成方法。 <ul style="list-style-type: none"> 0: 对2个输入轴的位置进行加法运算 (mcAddAxes) 1: 对2个输入轴的位置进行减法运算 (mcSubAxes) 关于详细内容,请参阅下述章节。 ☞ 1755页 加减法方法选择(CombineMode)
GearRatioNumeratorM1	主轴1齿轮比分子	DINT	启动时/可重启/可连续更新	-2147483648~2147483647	1	设置主轴1(Master1)的分子的值。 关于详细内容,请参阅下述章节。 ☞ 1755页 主轴1齿轮比分子 (GearRatioNumeratorM1)/主轴1齿轮比分母 (GearRatioDenominatorM1)
GearRatioDenominatorM1	主轴1齿轮比分母	DWORD (UDINT)	启动时/可重启/可连续更新	1~2147483647	1	设置主轴1(Master1)的分母的值。 关于详细内容,请参阅下述章节。 ☞ 1755页 主轴1齿轮比分子 (GearRatioNumeratorM1)/主轴1齿轮比分母 (GearRatioDenominatorM1)
GearRatioNumeratorM2	主轴2齿轮比分子	DINT	启动时/可重启/可连续更新	-2147483648~2147483647	1	设置主轴2(Master2)的分子的值。 关于详细内容,请参阅下述章节。 ☞ 1756页 主轴2齿轮比分子 (GearRatioNumeratorM2)/主轴2齿轮比分母 (GearRatioDenominatorM2)
GearRatioDenominatorM2	主轴2齿轮比分母	DWORD (UDINT)	启动时/可重启/可连续更新	1~2147483647	1	设置主轴2(Master2)的分母的值。 关于详细内容,请参阅下述章节。 ☞ 1756页 主轴2齿轮比分子 (GearRatioNumeratorM2)/主轴2齿轮比分母 (GearRatioDenominatorM2)
MasterValueSourceM1	主轴1数据源选择	INT (MC_SOURCE)	启动时	1、2、101、102	1	设置主轴1(Master1)的数据源。 <ul style="list-style-type: none"> 1: 指令当前值 (mcSetValue) 2: 反馈值 (mcActualValue) 101: 最新指令当前值 (mcLatestSetValue) 102: 最新反馈值 (mcLatestActualValue) 关于详细内容,请参阅下述章节。 ☞ 1756页 主轴1数据源选择 (MasterValueSourceM1)/主轴2数据源选择 (MasterValueSourceM2)
MasterValueSourceM2	主轴2数据源选择	INT (MC_SOURCE)	启动时	1、2、101、102	1	设置主轴2(Master2)的数据源。 <ul style="list-style-type: none"> 1: 指令当前值 (mcSetValue) 2: 反馈值 (mcActualValue) 101: 最新指令当前值 (mcLatestSetValue) 102: 最新反馈值 (mcLatestActualValue) 关于详细内容,请参阅下述章节。 ☞ 1756页 主轴1数据源选择 (MasterValueSourceM1)/主轴2数据源选择 (MasterValueSourceM2)
BufferMode	缓冲模式	INT (MC_BUFFER_MODE)	启动时	0、1	0	设置缓冲模式。 <ul style="list-style-type: none"> 0: Aborting (mcAborting) 1: Buffered (mcBuffered) 关于详细内容,请参阅下述章节。 ☞ 1757页 缓冲模式(BufferMode)
Options	选项	DWORD (HEX)	启动时	0000000H	0000000H	应设置“0000000H”。 * 设置了“0000000H”以外的情况下,将变为超出Options范围(错误代码: 1ABBH)。

■输出变量

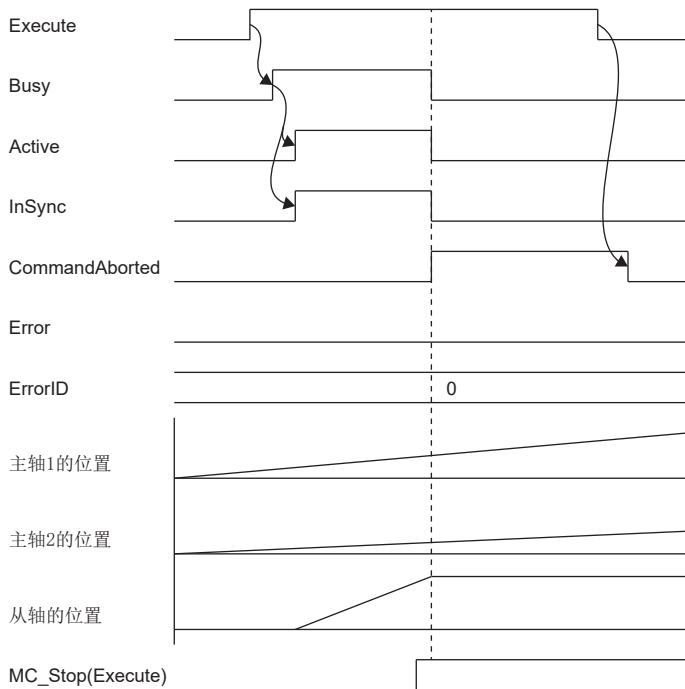
输出变量	名称	数据类型	初始值	内容
InSync	同步中	BOOL	FALSE	从轴(Slave)开始了同步时, 将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_CombineAxes(加减法定位)时, 将变为TRUE。
Active	控制中	BOOL	FALSE	正在控制从轴(Slave)时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MC_CombineAxes(加减法定位)的执行中断时, 将变为TRUE。 执行指令(Execute)变为FALSE时, 将变为FALSE。
Error	错误	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时, 将返回错误代码。 关于错误代码的详细内容, 请参阅下述手册。 ☞所使用的控制器的用户手册

功能

- 在MC_CombineAxes(加减法定位)中, 设置加减法方式选择(CombineMode)、主轴1齿轮比分子(GearRatioNumeratorM1)、主轴1齿轮比分母(GearRatioDenominatorM1)、主轴2齿轮比分子(GearRatioNumeratorM2)、主轴2齿轮比分母(GearRatioDenominatorM2)、主轴1数据源选择(MasterValueSourceM1)、主轴2数据源选择(MasterValueSourceM2)、缓冲模式(BufferMode), 并进行加减法定位。
- 结束动作的情况下, 通过MC_Stop(强制停止)进行。
- 对主轴1(Master1)与主轴2(Master2)的移动量进行合成。此外, 可以对各主轴设置齿轮比。移动量的合成与主轴1(Master1)及主轴2(Master2)的单位无关, 对数值进行加减法运算。

■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图, 请参阅下述章节。

☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■加减法方法选择(CombineMode)

设置主轴1(Master1)与主轴2(Master2)的移动量的合成方法。

移动量的合成与主轴1(Master1)及主轴2(Master2)的单位无关,对数值进行加减法运算。

设定值	内容
0: 对2个输入轴的位置进行加法运算 (mcAddAxes)	<p>将主轴1(Master1)与主轴2(Master2)的移动量相加后输出。</p> <ul style="list-style-type: none"> 从轴(Slave)的移动量 = 主轴1(Master1) + 主轴2(Master2)的移动量
1: 对2个输入轴的位置进行减法运算 (mcSubAxes)	<p>从主轴1(Master1)中减去主轴2(Master2)的移动量后输出。</p> <ul style="list-style-type: none"> 从轴(Slave)的移动量 = 主轴1(Master1) - 主轴2(Master2)的移动量

*1 是从轴(Slave)的轴状态。

■主轴1齿轮比分子(GearRatioNumeratorM1)/主轴1齿轮比分母(GearRatioDenominatorM1)

主轴1齿轮比分子(GearRatioNumeratorM1)与主轴1齿轮比分母(GearRatioDenominatorM1)组合进行设置。

主轴1齿轮比分子(GearRatioNumeratorM1)设置对主轴1(Master1)的移动量进行转换及合成时的分子的值。

主轴1齿轮比分母(GearRatioDenominatorM1)设置对主轴1(Master1)的移动量进行转换及合成时的分母的值。

按下述方式转换移动量。

$$\text{转换后的主轴1(Master1)移动量} = \text{转换前的主轴1(Master1)移动量} \times \frac{\text{主轴1齿轮比分子(GearRatioNumeratorM1)}}{\text{主轴1齿轮比分母(GearRatioDenominatorM1)}}$$

- 主轴1齿轮比分母(GearRatioDenominatorM1)中应设置正的值。
- 通过在主轴1齿轮比分子(GearRatioNumeratorM1)的设定值中设置负的值,可以反转动量后进行合成。
- 通过在主轴1齿轮比分子(GearRatioNumeratorM1)的设定值中设置“0”,可以将转换后的主轴1(Master1)的移动量设置为“0”后进行合成。

■主轴2齿轮比分子(GearRatioNumeratorM2)/主轴2齿轮比分母(GearRatioDenominatorM2)

主轴2齿轮比分子(GearRatioNumeratorM2)与主轴2齿轮比分母(GearRatioDenominatorM2)组合进行设置。

主轴2齿轮比分子(GearRatioNumeratorM2)设置对主轴2(Master2)的移动量进行转换及合成时的分子的值。

主轴2齿轮比分母(GearRatioDenominatorM2)设置对主轴2(Master2)的移动量进行转换及合成时的分母的值。

按下述方式转换移动量。

$$\text{转换后的主轴2(Master2)移动量} = \text{转换前的主轴2(Master2)移动量} \times \frac{\text{主轴2齿轮比分子(GearRatioNumeratorM2)}}{\text{主轴2齿轮比分母(GearRatioDenominatorM2)}}$$

- 主轴2齿轮比分母(GearRatioDenominatorM2)中应设置正的值。
- 通过在主轴2齿轮比分子(GearRatioNumeratorM2)的设定值中设置负的值，可以反转移动量后进行合成。
- 通过在主轴2齿轮比分子(GearRatioNumeratorM2)的设定值中设置“0”，可以将转换后的主轴2(Master2)的移动量设置为“0”后进行合成。

■主轴1数据源选择(MasterValueSourceM1)/主轴2数据源选择(MasterValueSourceM2)

设置从轴(Slave)执行单轴同步控制的主轴1(Master1)或主轴2(Master2)的位置的类型。

根据主轴1(Master1)或主轴2(Master2)及从轴(Slave)的运算周期不相同的情况下及FB的执行顺序，动作可能会改变。

将实轴以外的轴类型设置为主轴1(Master1)或主轴2(Master2)并在主轴1数据源选择(MasterValueSourceM1)或主轴2数据源选择(MasterValueSourceM2)中设置了反馈值的情况下，将以与指令当前值相同的值执行动作。

设定值	内容
1: 指令当前值(mcSetValue)	使用上次的运算周期中的主轴的指令位置。
2: 反馈值(mcActualValue)	使用上次的运算周期中的主轴的反馈位置。
101: 最新指令当前值(mcLatestSetValue)	使用本次的运算周期中的主轴的指令位置。
102: 最新反馈值(mcLatestActualValue)	使用本次的运算周期中的主轴的反馈位置。

要点

在主轴1数据源选择(MasterValueSourceM1)或主轴2数据源选择(MasterValueSourceM2)中设置“1: 指令当前值(mcSetValue)”、“101: 最新指令当前值(mcLatestSetValue)”，主轴由于伺服报警及紧急停止而变为了伺服OFF的情况下，值的变化量有可能变大。通过在主轴1数据源选择(MasterValueSourceM1)或主轴2数据源选择(MasterValueSourceM2)中设置“2: 反馈值(mcActualValue)”、“102: 最新反馈值(mcLatestActualValue)”可以防止。

■缓冲模式(BufferMode)

设置用于进行多重启动(缓冲模式)的动作。

在MC_CombineAxes(加减法定位)中,可以设置下述缓冲模式。

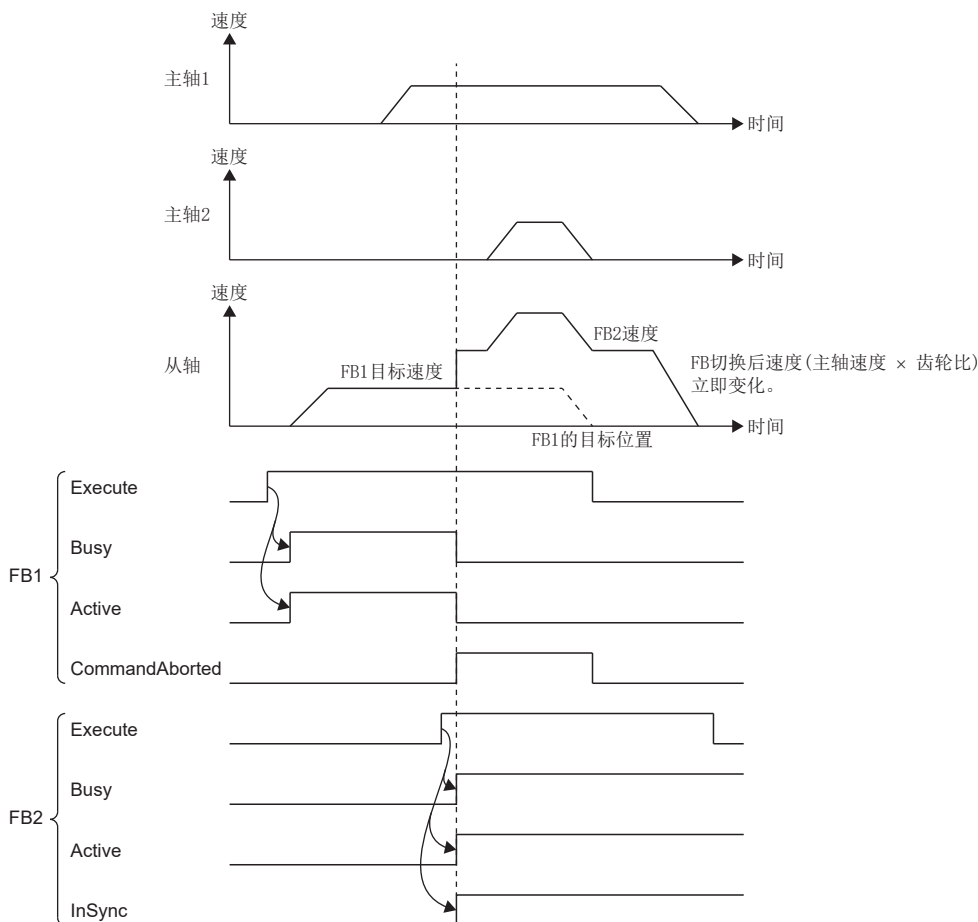
MC_CombineAxes(加减法定位)的速度跟踪同步的FB(主轴),因此FB切换时根据主轴速度与设置的齿轮数据,速度即时变化。

设定值	内容
0: Aborting(mcAborting)	中断(取消)控制中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在控制中的FB中缓冲下一个FB。 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个)控制中的FB完成时,依次执行缓冲FB。

例

对执行中的FB1上连接启动FB2(从轴(Slave): MC_CombineAxes(加减法定位))将缓冲模式(BufferMode)设置为“0: 始终(Aborting)”的情况下

- 加减法方法选择(CombineMode)为“0: 对2个输入轴的位置进行加法运算(mcAddAxes)”时



FB2的速度的发生是在同步中(InSync)变为TRUE之后。

要点

关于多重启动(缓冲模式)的详细内容,请参阅下述手册。

📖 所使用的控制器的用户手册

程序示例

将加减法定位开始指令(bCombineAxesStart)置为TRUE，并根据下述设置将对JOG运行1FB的轴2(Axis0002)与JOG运行1FB的轴3(Axis0003)的主轴2轴的移动量进行了加减法运算的值作为指令值，进行从轴(Axis0001)的运动的程序示例如下所示。

- 设置

项目	设定值		
		JOG运行1用数据	JOG运行2用数据
JOG运行用数据设置	速度	1000.0	500.0
	加速度	5000.0	5000.0
	减速度	5000.0	5000.0
	Jerk	0.0	0.0

■轴

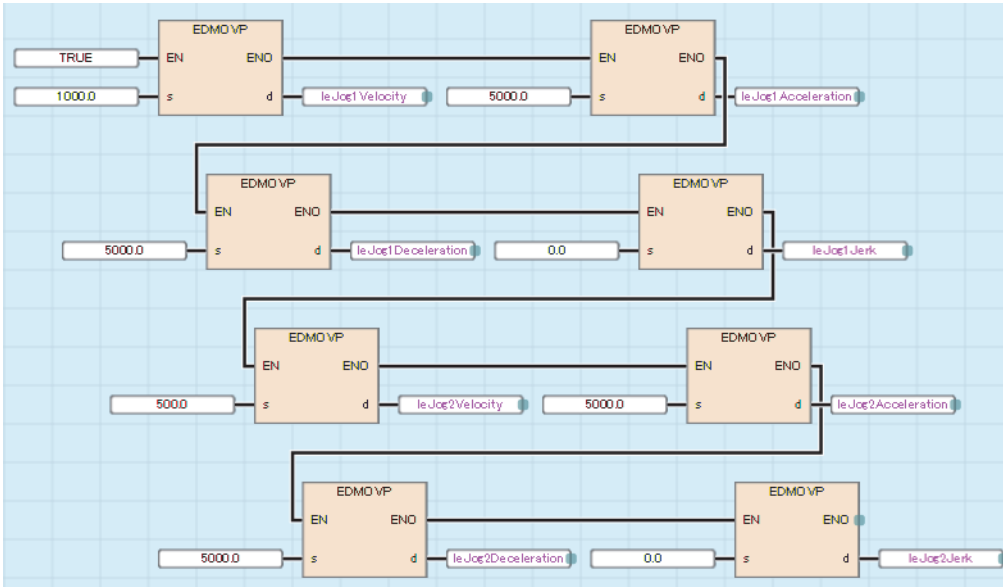
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1
2	Axis0002	AXIS_REF	轴2
3	Axis0003	AXIS_REF	轴3

■使用的标签

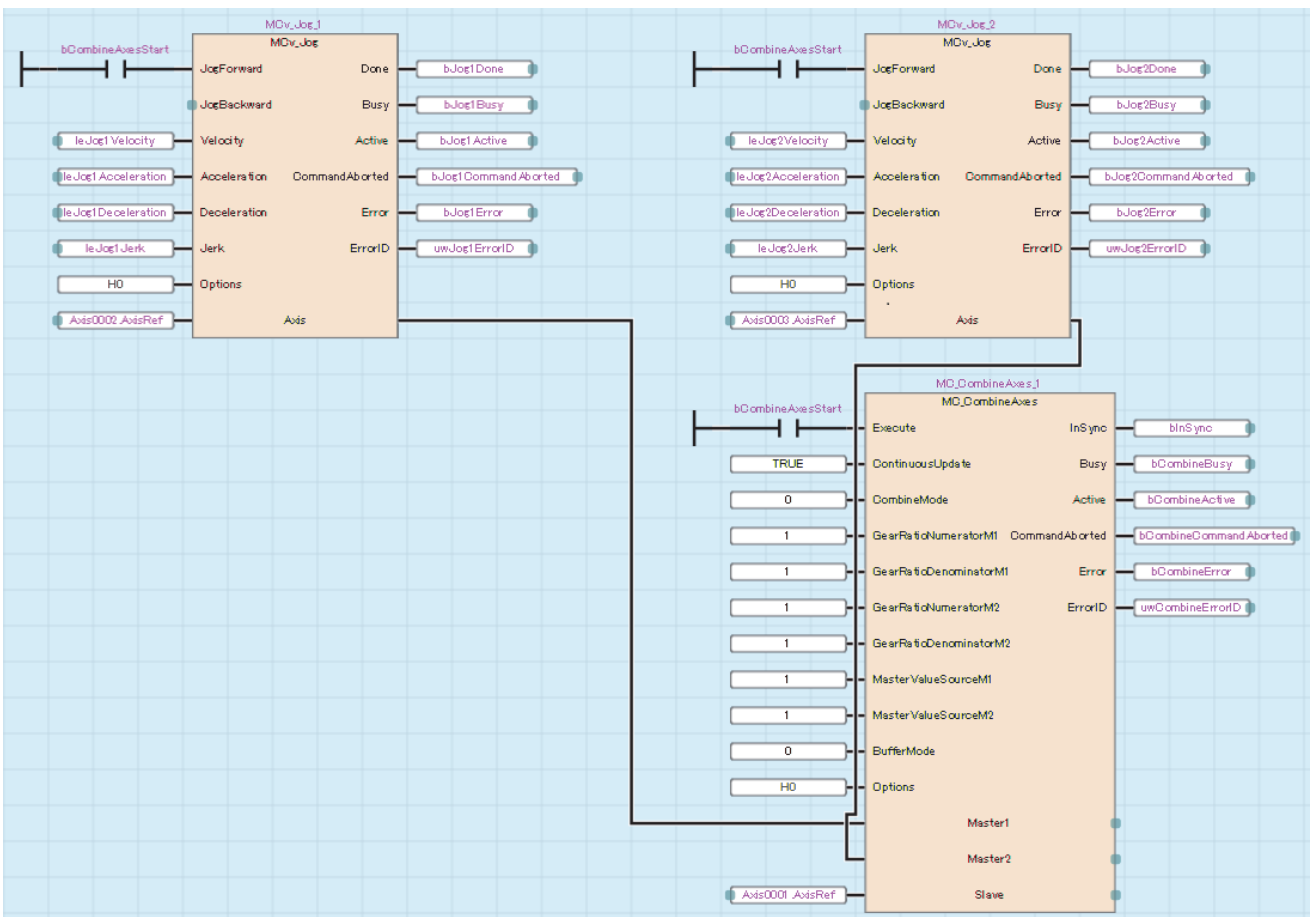
标签名	数据类型	注释
MC_CombineAxes_1	MC_CombineAxes	加减法定位FB
bCombineAxesStart	位	加减法定位开始指令
bInSync	位	同步中
bCombineBusy	位	执行中
bCombineActive	位	控制中
bCombineCommandAborted	位	执行中断
bCombineError	位	错误
uwCombineErrorID	字[无符号]/位串[16位]	错误代码
MCv_Jog_1	MCv_Jog	JOG运行1FB
leJog1Velocity	双精度实数	JOG速度
leJog1Acceleration	双精度实数	JOG加速度
leJog1Deceleration	双精度实数	JOG减速度
leJog1Jerk	双精度实数	JOG Jerk
bJog1Done	位	执行完成
bJog1Busy	位	执行中
bJog1Active	位	控制中
bJog1CommandAborted	位	执行中断
bJog1Error	位	错误
uwJog1ErrorID	字[无符号]/位串[16位]	错误代码
MCv_Jog_2	MCv_Jog	JOG运行2FB
leJog2Velocity	双精度实数	JOG速度
leJog2Acceleration	双精度实数	JOG加速度
leJog2Deceleration	双精度实数	JOG减速度
leJog2Jerk	双精度实数	JOG Jerk
bJog2Done	位	执行完成
bJog2Busy	位	执行中
bJog2Active	位	控制中
bJog2CommandAborted	位	执行中断
bJog2Error	位	错误
uwJog2ErrorID	字[无符号]/位串[16位]	错误代码

■ FBD/LD程序

- JOG运行用数据的设置



- JOG运行1/JOG运行2/加减法定位



■ST程序

```
//-----JOG运行用数据的设置-----
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运行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运行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
);

//-----加减法定位-----
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
);
```

46.17 齿隙补偿滤波器

MCv_BacklashCompensationFilter

进行按照移动方向，对机械系统的齿隙量进行补偿的滤波器处理。

梯形图	FBD/LD	ST
		<pre> 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?); </pre>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Master	主轴	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName.AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName.AxisRef. (轴信息)
Slave	从轴	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName.AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName.AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Enable	有效	BOOL	始终	TRUE、FALSE	FALSE	设置为TRUE时，将执行MCv_BacklashCompensationFilter (齿隙补偿滤波器)。
FilterDisable	滤波器无效	BOOL	始终	TRUE、FALSE	FALSE	设置滤波器处理的方法。 • FALSE: 按照滤波器设置执行动作。 • TRUE: 滤波器将变为无效，将主轴(Master)的输入原样不变地传送到从轴(Slave)。 * 设置为TRUE的情况下，滤波器累计值(FilterPool)将变为“0”，并将删除的滤波器累计值(FilterPool)存储至删除的滤波器累计值(PurgedFilterPool)。
BacklashAmount	齿隙量	LREAL	启动时	0.0~2500000000.0[U]	0.0	设置机械系统的齿隙(间隙)量。
BacklashDirection	齿隙补偿方向	INT (MC_DIRECTION)	启动时	1、2	0	设置进行齿隙补偿的方向。 • 1: 正方向(mcPositiveDirection) • 2: 负方向(mcNegativeDirection) * 省略了设置的情况下，将变为超出齿隙补偿方向范围(错误代码: 1BC0H)。 关于详细内容，请参阅下述章节。 ☞ 1767页 齿隙补偿方向(BacklashDirection)
MasterValueSource	主轴数据来源选择	INT (MC_SOURCE)	启动时	1、101	1	设置主轴(Master)的数据源。 • 1: 指令当前值(mcSetValue) • 101: 最新指令当前值(mcLatestSetValue) 关于详细内容，请参阅下述章节。 ☞ 1767页 主轴数据来源选择(MasterValueSource)
Options	选项	DWORD (HEX)	启动时	0000000H	0000000H	设置功能选项。 关于详细内容，请参阅下述章节。 ☞ 1768页 选项(Options)

■输出变量

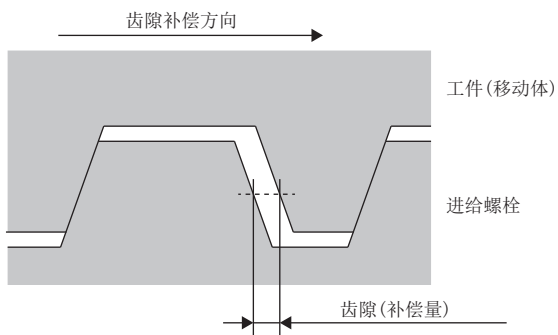
输出变量	名称	数据类型	初始值	内容
Valid	输出值有效	BOOL	FALSE	输出值为有效时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_BacklashCompensationFilter(齿隙补偿滤波器)时，将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_BacklashCompensationFilter(齿隙补偿滤波器)正在控制轴时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MCv_BacklashCompensationFilter(齿隙补偿滤波器)的执行中断时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 ☞所使用的控制器的用户手册
FilterPool	滤波器累计值	LREAL	0.0	输出滤波器处理中累计的从轴的移动量。 齿隙滤波器的情况下，对滤波器累计值(FilterPool)的正负进行了反转后的值将变为传送到从轴中的补偿量。

■公开变量

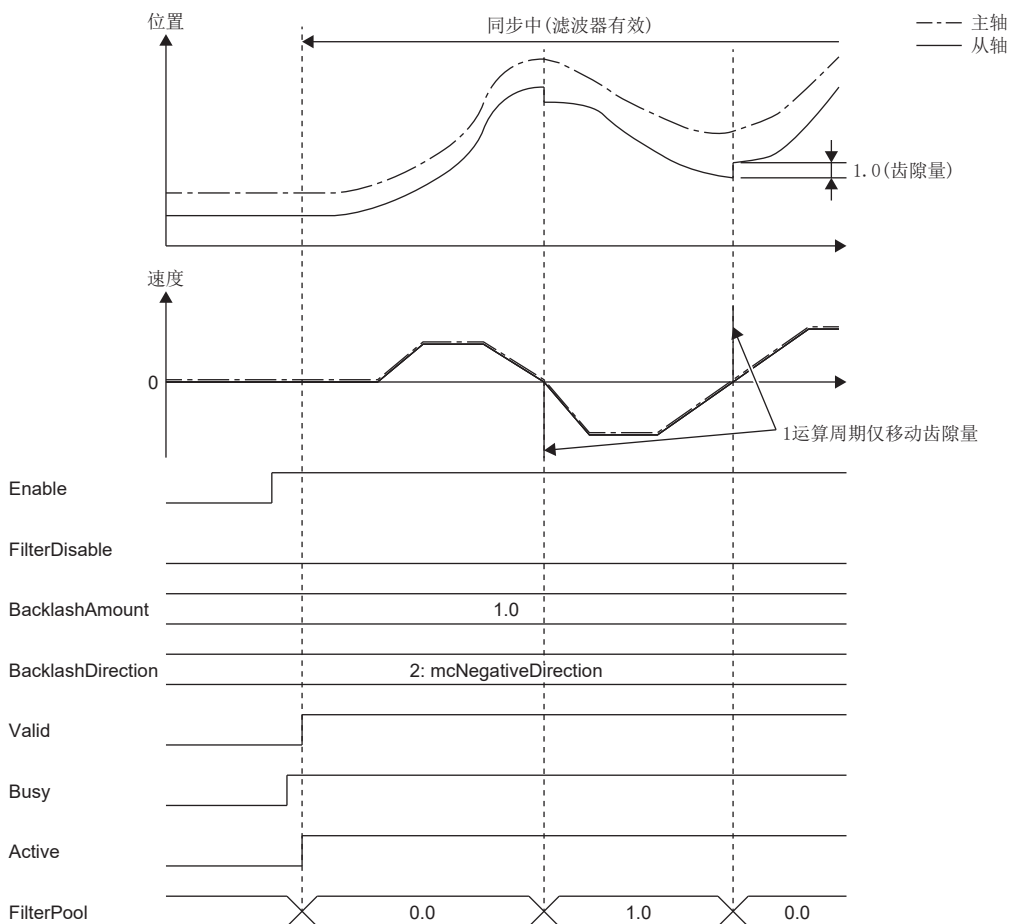
公开变量	名称	数据类型	初始值	内容
PurgedFilterPool	删除的滤波器累计值	LREAL	0.0	通过将滤波器无效(FilterDisable)置为TRUE存储删除的滤波器累计值(FilterPool)。 删除的滤波器累计值(PurgedFilterPool)不累计。 在FB启动时将被清零。

功能

- 齿隙补偿滤波器是进行机械系统的齿隙(间隙)量的补偿的滤波器。

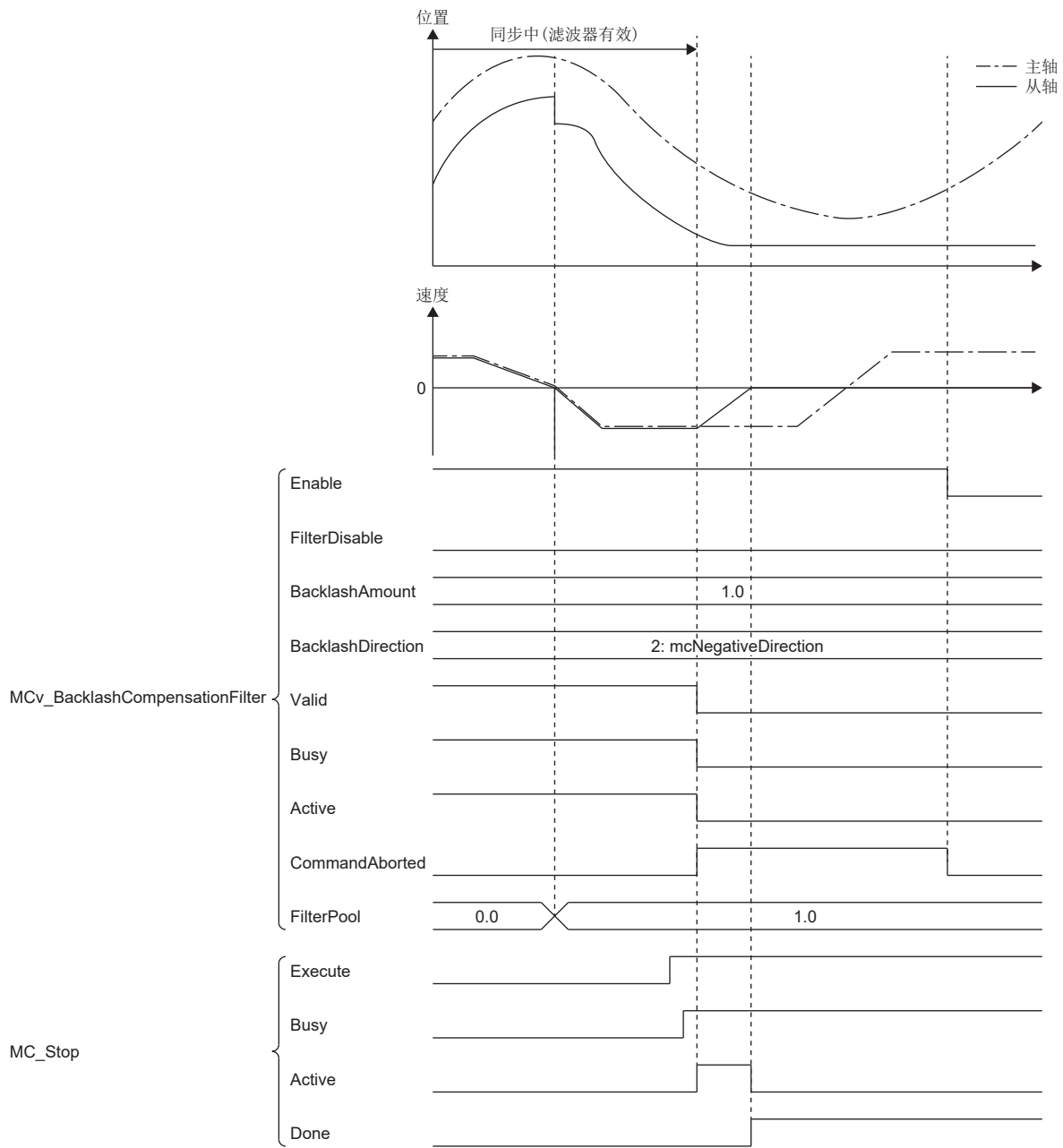


- 主轴(Master)的移动方向与齿隙补偿方向(BacklashDirection)相同的情况下,对于将主轴(Master)的当前位置向齿隙补偿方向(BacklashDirection)的方向进行了齿隙量(BacklashAmount)的移位的位置,从轴(Slave)进行同步。主轴(Master)的移动方向与齿隙补偿方向(BacklashDirection)相反的情况下,从轴(Slave)将与主轴(Master)的当前位置进行同步。每当主轴(Master)的移动方向改变时仅按齿隙量(BacklashAmount)发生额外的从轴(Slave)的移动量,并进行齿隙补偿。



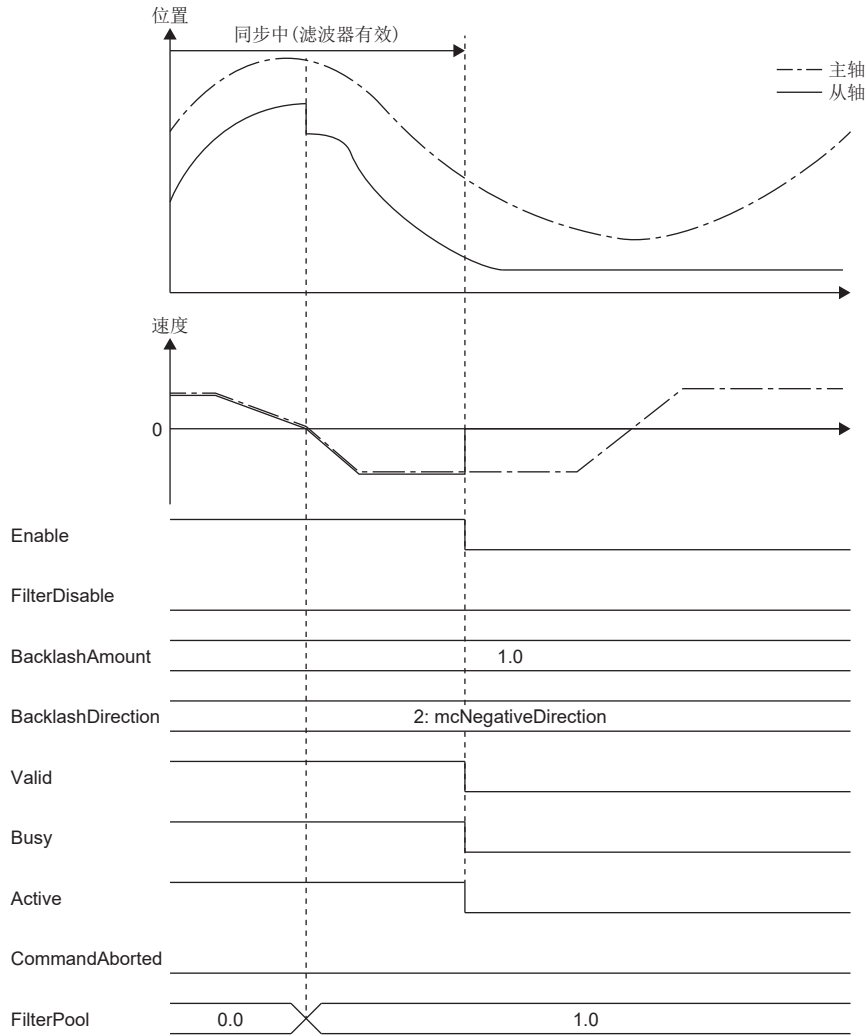
• 齿隙补偿滤波器固有的停止动作如下所示。

从轴(Slave)中发生停止原因时从轴(Slave)将从同步位置开始停止动作。补偿的移动量(滤波器累计值(FilterPool)的值)不撤消。



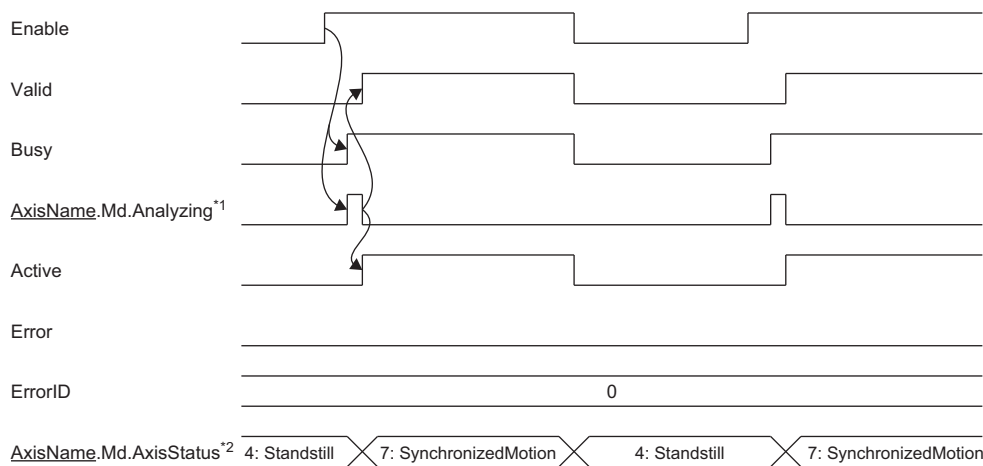
- 滤波器无效时(有效(Enable)下降沿时)的动作

有效(Enable)下降沿时从轴(Slave)将在同步位置立即停止。补偿的移动量(滤波器累计值(FilterPool)的值)不撤消。



■ 时序图

- 正常完成的情况下



*1 是从轴(Slave)的分析中。

*2 是从轴(Slave)的轴状态。

- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

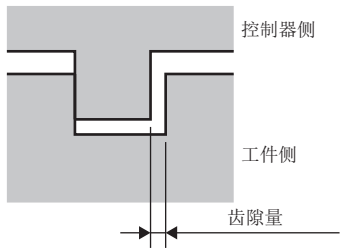
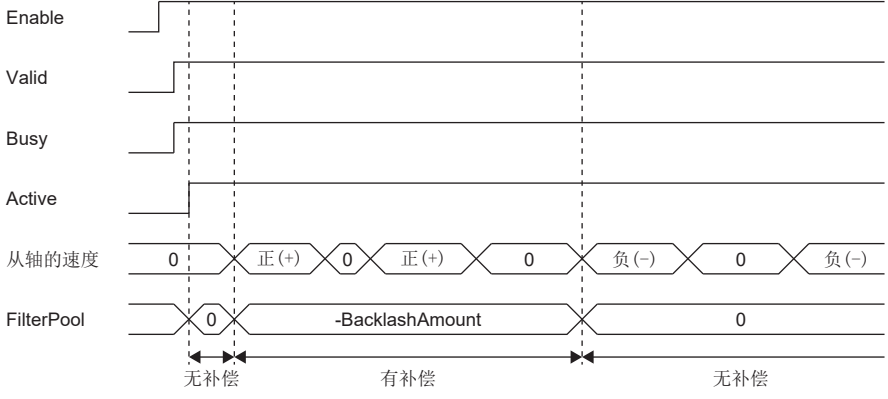
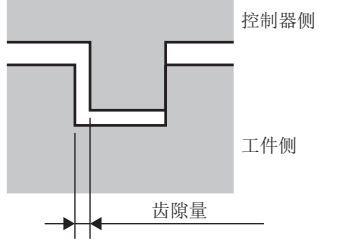
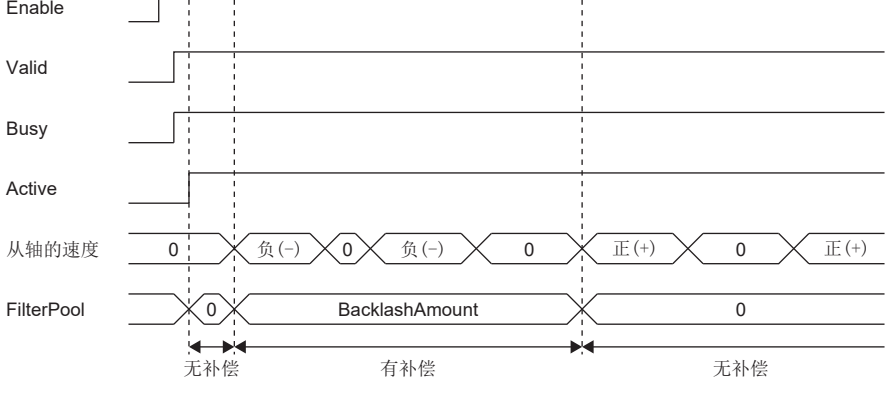
☞ 1378页 通过有效(Enable)类型的运动控制FB的基本动作

■ 齿隙补偿方向 (BacklashDirection)

设置进行齿隙补偿的方向。

主轴 (Master) 的移动方向为齿隙补偿方向 (BacklashDirection) 的情况下, 将补偿量输出到从轴 (Slave) 中。

齿隙补偿滤波器有效时 (在 FALSE 时启动了滤波器无效 (FilterDisable) 时, 或执行中滤波器无效 (FilterDisable) 的下降沿时) 对于主轴 (Master) 的指令当前位置的地址增加方向, 机械系统的齿隙状态处于下述关系时则进行补偿。

设定值	内容
1: 正方向 (mcPositiveDirection) 指令当前位置的地址增加方向 	主轴 (Master) 的移动方向为正方向时, 从轴 (Slave) 将与对主轴 (Master) 的当前位置向正方向进行了齿隙量 (BacklashAmount) 的移位的位置进行同步。 
2: 负方向 (mcNegativeDirection) 指令当前位置的地址增加方向 	主轴 (Master) 的移动方向为负方向时, 从轴 (Slave) 将与对主轴 (Master) 的当前位置向负方向进行了齿隙量 (BacklashAmount) 的移位的位置进行同步。 

■ 主轴数据源选择 (MasterValueSource)

设置从轴 (Slave) 执行单轴同步控制的主轴 (Master) 的位置的类型。

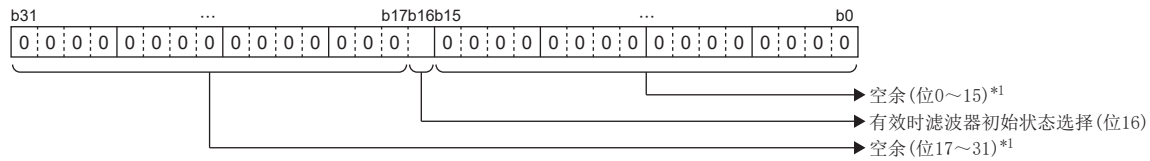
根据主轴 (Master) 及从轴 (Slave) 的运算周期不相同的情况下及 FB 的执行顺序, 动作可能会改变。

设定值	内容
1: 指令当前值 (mcSetValue)	使用上次的运算周期中的主轴的指令位置。
101: 最新指令当前值 (mcLatestSetValue)	使用本次的运算周期中的主轴的指令位置。

■选项 (Options)

将MCv_BacklashCompensationFilter (齿隙补偿滤波器)中使用的功能选项以位指定进行设置。

以位指定设置的内容如下所示。



*1 空余中应设置“0”。设置了“0”以外的情况下，将变为超出Options范围(错误代码：1ABBH)。

位	名称	内容
16	有效时滤波器初始状态选择	设置齿隙补偿滤波器有效时(在FALSE时启动了滤波器无效(FilterDisable)时,或执行中滤波器无效(FilterDisable)的下降沿时)的初始状态。 • 0: 首次补偿 *设置了“0”以外的情况下,将变为超出Options范围(错误代码:1ABBH)且不启动。

• 有效时滤波器初始状态选择(位16)

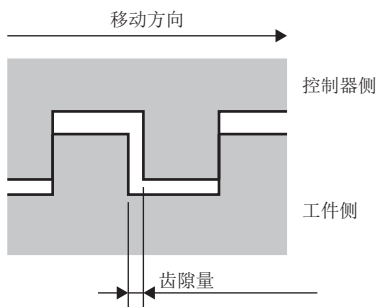
设定值	内容
0: 首次补偿	与滤波器有效时的主轴(Master)的当前方向无关,主轴(Master)与从轴(Slave)的当前位置处于未加上齿隙补偿的位置关系时则开始滤波器动作。

■将齿隙补偿滤波器置为有效的步骤

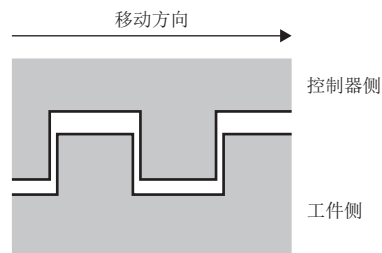
将齿隙补偿滤波器置为有效的步骤的使用示例如下所示。通过原点复位控制确定了机械位置后,及齿隙补偿滤波器途中停止后重新开始的情况下应进行此步骤。

1. 应设置为将实际驱动轴通过原点复位控制及JOG运行等进行移动,按下述的正确示例所示向一个方向产生了机械系统的齿隙的状态。

■正确的示例



■错误的示例



2. 通过当前位置更改等将主轴(Master)的指令当前位置与实际驱动轴的指令当前位置一致。

3. 将步骤1.中实际驱动轴的移动方向的反方向设置为齿隙补偿方向(BacklashDirection),启动齿隙补偿滤波器。后段中有功能块的情况下也启动这些功能块。

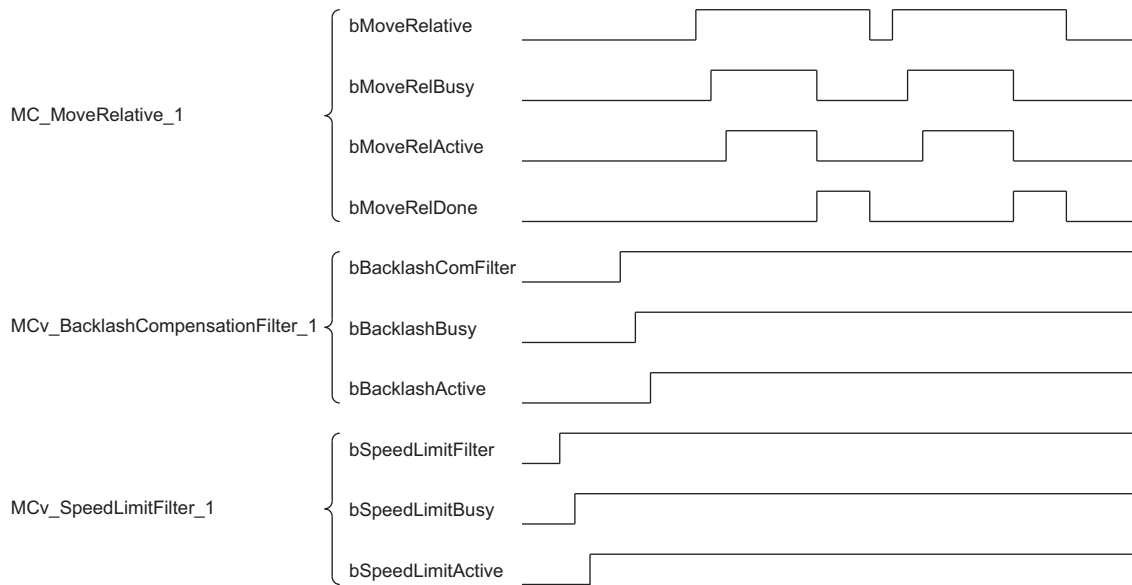
注意事项

- 齿隙补偿是以指令方向及齿隙的状态不通过外力等改变为前提的功能。下述的情况下使用时将不能正确进行补偿。
 - 上下轴等始终向恒定方向施加外力的机械
 - 齿隙量根据机械位置变化的机构(齿条&齿轮机构等)
- 在主轴(Master)的移动方向改变的时刻从轴(Slave)将在1运算周期内仅移动齿隙量(BacklashAmount)。根据值,有可能超出速度限制值及驱动器模块可允许的指令频率,因此应在齿隙补偿滤波器的后段连接速度限制滤波器及平滑滤波器。
- 从轴(Slave)的当前位置(指令当前位置、进给机械位置等)将为加上了齿隙量(BacklashAmount)的补偿的位置。主轴(Master)的当前位置不加上齿隙量(BacklashAmount)的补偿。
- 如果在指令滤波器的动作中进行滤波器无效(FilterDisable)的切换,从轴(Slave)的速度可能会急剧变化。
- 将指令滤波器置为有效后,从轴(Slave)中发生了停止原因,导致从轴(Slave)及后段的同步控制功能块中不再传送主轴(Master)的指令时,同步位置关系将发生破坏。再次将指令滤波器置为有效之前,应根据需要进行同步位置校准。

程序示例

在齿隙补偿滤波器的后段连接速度限制滤波器以进行滤波器处理的程序示例如下所示。

• 时序图



• 设置

项目	设定值	
齿隙量	1.0	
单轴定位用数据设置	移动量	1000.0
	速度	20.0
	加速度	300.0
	减速度	300.0
	Jerk	0.0
正方向/负方向限制值	正方向限制值	100.0
	负方向限制值	100.0

■ 轴

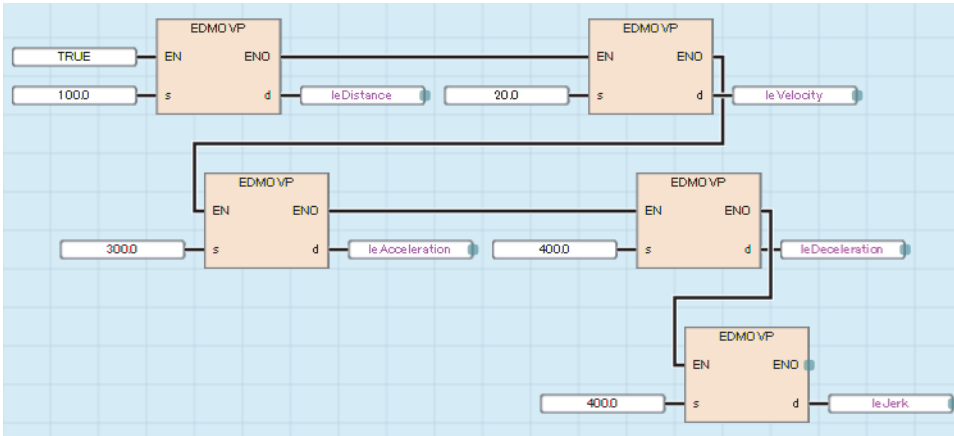
轴No.	标签名	数据类型	注释
1	VirtualAxis0001	AXIS_REF	轴1 (虚拟驱动轴)
2	VirtualAxis0002	AXIS_REF	轴2 (虚拟驱动轴)
3	Axis0003	AXIS_REF	轴3 (实际驱动轴)

■使用的标签

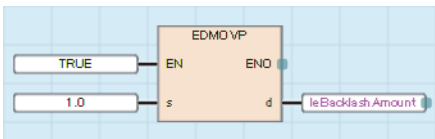
标签名	数据类型	注释
MC_MoveRelative_1	MC_MoveRelative	相对值定位FB
bMoveRelative	位	单轴相对值定位启动
leDistance	双精度实数	移动量
leVelocity	双精度实数	速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bMoveRelDone	位	执行完成
bMoveRelBusy	位	执行中
bMoveRelActive	位	控制中
bMoveRelCommandAborted	位	执行中断
bMoveRelError	位	错误
uwMoveRelErrorID	字[无符号]/位串[16位]	错误代码
MCv_BacklashCompensationFilter_1	MCv_BacklashCompensationFilter	齿隙补偿滤波器FB
bBacklashComFilter	位	齿隙补偿滤波器指令
leBacklashAmount	双精度实数	齿隙量
bBacklashValid	位	输出值有效
bBacklashBusy	位	执行中
bBacklashActive	位	控制中
bBacklashCommandAborted	位	执行中断
bBacklashError	位	错误
uwBacklashErrorID	字[无符号]/位串[16位]	错误代码
leBacklashFilterPool	双精度实数	滤波器累计值
MCv_SpeedLimitFilter_1	MCv_SpeedLimitFilter	速度限制滤波器
bSpeedLimitFilter	位	速度限制滤波器指令
lePositiveLimit	双精度实数	正方向限制值
leNegativeLimit	双精度实数	负方向限制值
bSpeedLimitValid	位	输出值有效
bSpeedLimitBusy	位	执行中
bSpeedLimitActive	位	控制中
bSpeedLimitCommandAborted	位	执行中断
bSpeedLimitError	位	错误
uwSpeedLimitErrorID	字[无符号]/位串[16位]	错误代码
leFilterPool	双精度实数	滤波器累计值

■FBD/LD程序

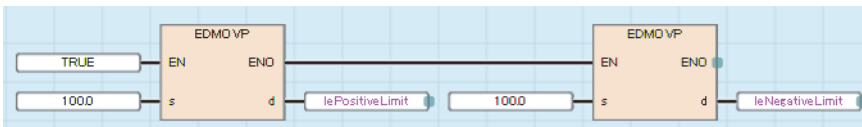
- 单轴定位用数据的设置



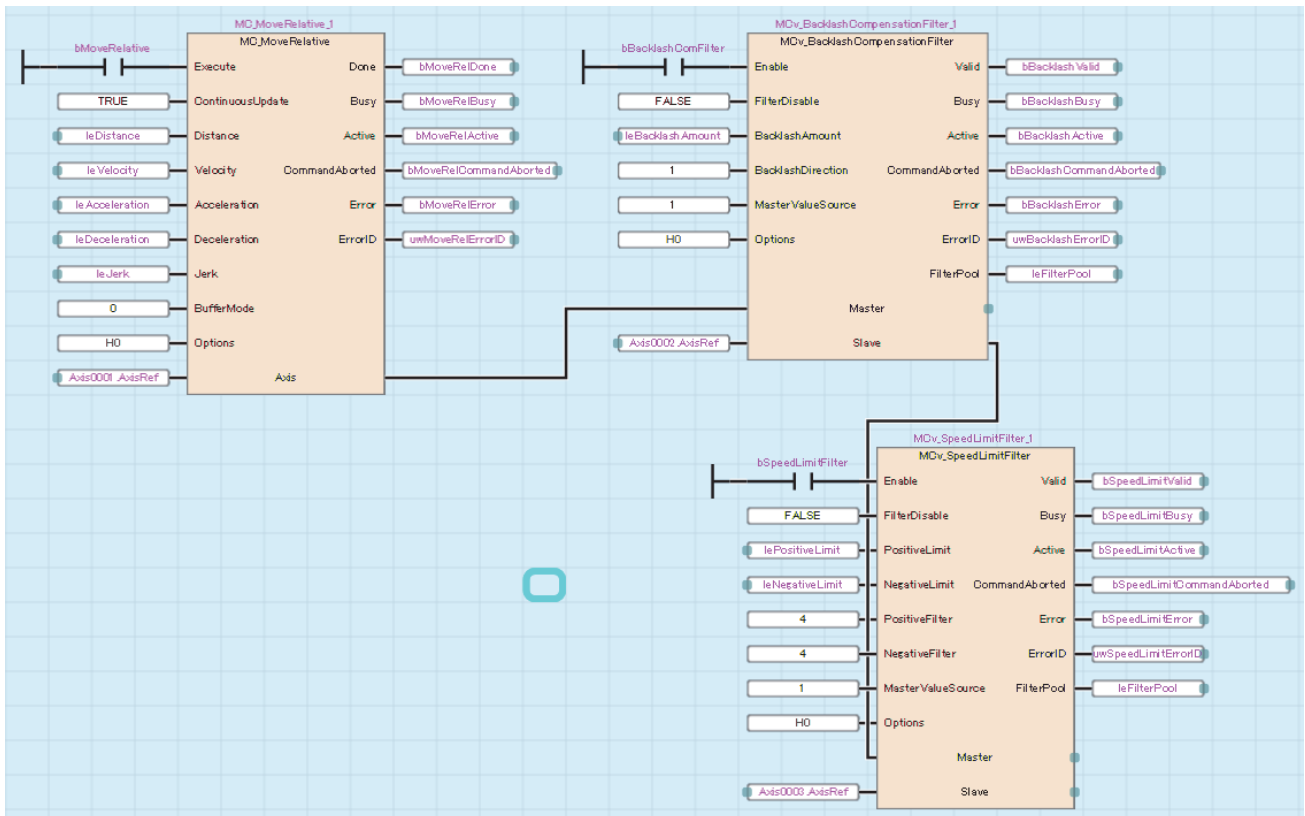
- 齿隙量



- 正方向/负方向限制值



• 相对值定位/齿隙补偿滤波器/速度限制滤波器



■ST程序

```
//-----单轴定位用数据设置-----
leDistance:= 100.0;
leVelocity:= 20.0;
leAcceleration:= 300.0;
leDeceleration:= 400.0;
leJerk:= 0.0;

//-----相对值定位-----
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
);

//-----齿隙量-----
leBacklashAmount:= 1.0;

//-----齿隙补偿滤波器-----
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/负方向限制值-----
lePositiveLimit:= 100.0;
```

```
leNegativeLimit:= 100.0;

//-----速度限制滤波器-----
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
);
```

46.18 平滑滤波器

MCv_SmoothingFilter

按照指定的频率，进行滤波器处理。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Master	主轴	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName, AxisRef. (轴信息)
Slave	从轴	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName, AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Enable	有效	BOOL	始终	TRUE、FALSE	FALSE	设置为TRUE时，将执行MCv_SmoothingFilter(平滑滤波器)。
FilterDisable	滤波器无效	BOOL	始终	TRUE、FALSE	FALSE	设置滤波器处理的方法。 • FALSE：按照滤波器设置执行动作。 • TRUE：滤波器将变为无效，将主轴(Master)的输入原样不变地传送到从轴(Slave)。 关于详细内容，请参阅下述章节。 ☞ 1779页 滤波器无效(FilterDisable)
Frequency	频率	LREAL	始终	0.20~ 250.00[Hz]	10.00	设置抑制振动的频率。 关于详细内容，请参阅下述章节。 ☞ 1779页 频率(Frequency)
FilterPurge	滤波器净化	BOOL	启动时	TRUE、FALSE	FALSE	设置有滤波器累计值的状态下的结束动作。 • FALSE：立即停止从轴。 • TRUE：执行动作直至变为“0”后停止。 关于详细内容，请参阅下述章节。 ☞ 1780页 滤波器净化(FilterPurge)
MasterValueSource	主轴数据源选择	INT (MC_SOURCE)	启动时	1、2、101、102	1	设置主轴(Master)的数据源。 • 1：指令当前值(mcSetValue) • 2：反馈值(mcActualValue) • 101：最新指令当前值(mcLatestSetValue) • 102：最新反馈值(mcLatestActualValue) 关于详细内容，请参阅下述章节。 ☞ 1781页 主轴数据源选择(MasterValueSource)
Options	选项	DWORD (HEX)	启动时	0000000H	0000000H	应设置“0000000H”。 * 设置了“0000000H”以外的情况下，将变为超出Options范围(错误代码：1ABBH)。

■输出变量

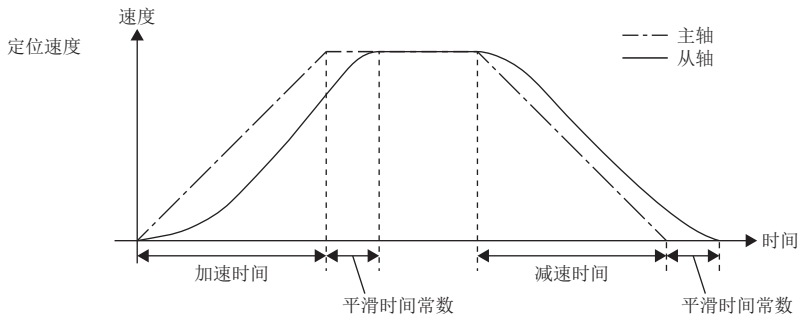
输出变量	名称	数据类型	初始值	内容
Valid	输出值有效	BOOL	FALSE	输出值为有效时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_SmoothingFilter(平滑滤波器)时，将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_SmoothingFilter(平滑滤波器)正在控制轴时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MCv_SmoothingFilter(平滑滤波器)的执行中断时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 □□所使用的控制器的用户手册
ValidFrequency	有效频率	LREAL	0.00	输出有效的频率。
FilterPool	滤波器累计值	LREAL	0.0	输出滤波器处理中累计的从轴的移动量。 滤波器累计值(FilterPool)在将有效(Enable)置为了TRUE时清除。 滤波器累计值(FilterPool)的值超出了定位范围的情况下，将变为滤波器累计值溢出警告(事件代码: 00D13H)。 检测到警告时，滤波器累计值(FilterPool)的误差可能会变大。

■公开变量

公开变量	名称	数据类型	初始值	内容
PurgedFilterPool	删除的滤波器累计值	LREAL	0.0	通过将滤波器无效(FilterDisable)置为TRUE存储删除的滤波器累计值(FilterPool)。 删除的滤波器累计值(PurgedFilterPool)不累计。 在FB启动时将被清零。

功能

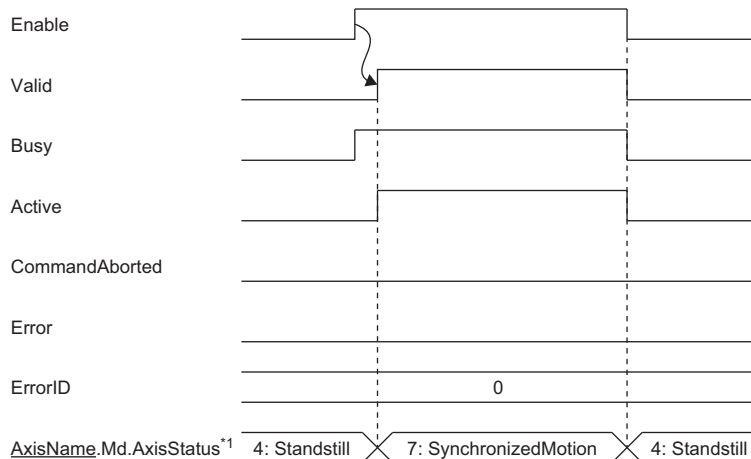
- 平滑滤波器可以去除高于设置的频率(Frequency)的频率，是使高于设定值的整个波形变为平滑的加减速波形的滤波器。
- 平滑滤波器将对主轴数据源选择(MasterValueSource)中设置的主轴(Master)的值执行了滤波器处理的结果输出到从轴(Slave)中。
- 平滑滤波器执行中的从轴(Slave)的轴状态(AxisName.Md.AxisStatus)为“7: 同步运行中(SynchronizedMotion)”。
- 平滑的时间常数将变为“ $1/\text{频率(Frequency)}[\text{s}]$ ”，加速时间及减速时间将仅按照平滑时间常数的量变长。频率(Frequency)为10[Hz]的情况下，平滑时间常数将变为“ $0.1[\text{s}] = 100[\text{ms}]$ ”。



- 从轴(Slave)处于其它动作FB中，启动了平滑滤波器的情况下，将忽略启动请求，且变为启动不可(错误代码: 1AFCH)。应使从轴(Slave)的轴状态(AxisName.Md.AxisStatus)在“4: 待机中(Standstill)”时启动。
- 平滑滤波器执行中，可多重启动的功能块仅为MC_Stop(强制停止)。

■时序图

- 正常完成的情况下



*1 是从轴(Slave)的轴状态。

- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

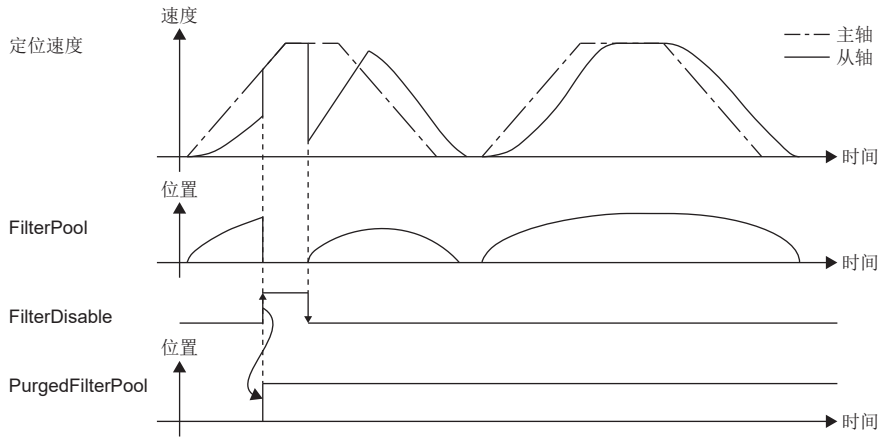
☞ 1378页 通过有效(Enable)类型的运动控制FB的基本动作

■滤波器无效(FilterDisable)

设置滤波器处理的方法。

设定值	内容
FALSE	按照滤波器设置执行动作。
TRUE	滤波器将变为无效，将主轴(Master)的输入原样不变地传送到从轴(Slave)。

- 将滤波器无效(FilterDisable)置为了TRUE的情况下，滤波器累计值(FilterPool)将变为“0.0”，并将删除的滤波器累计值(FilterPool)存储至删除的滤波器累计值(PurgedFilterPool)。



■频率(Frequency)

设置抑制振动的频率。

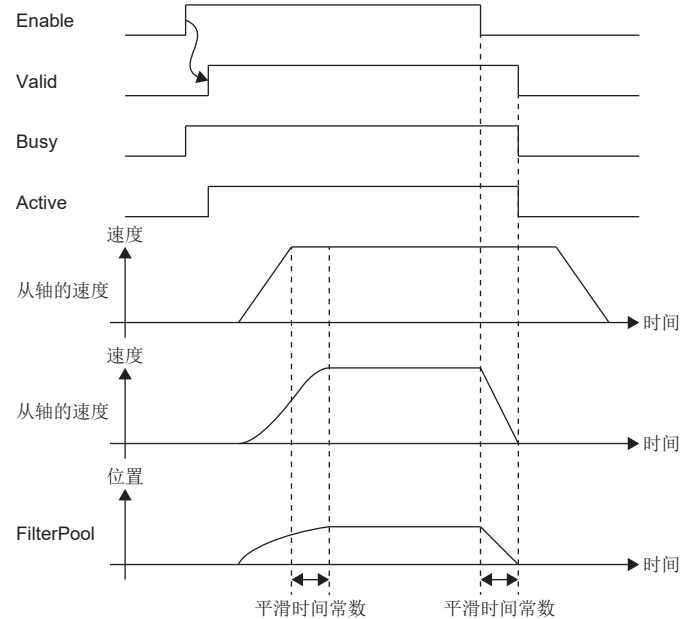
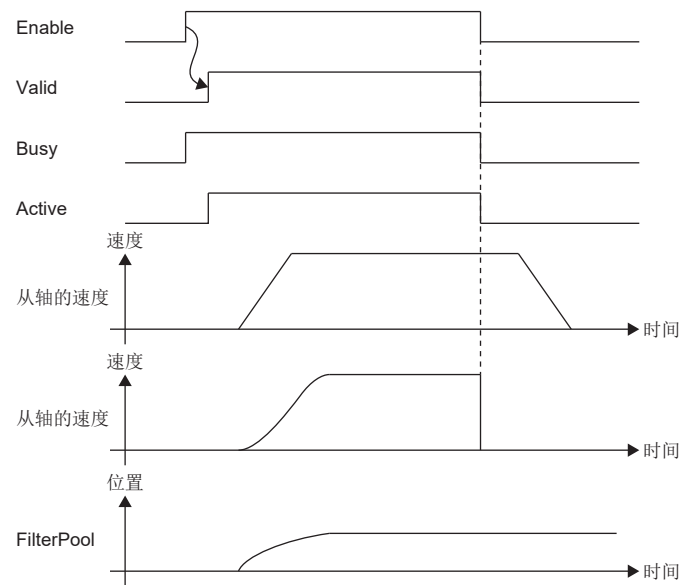
频率(Frequency)的可设置的范围如下所示。

1/从轴运算周期[s]	设置范围
250.00[Hz]及以上	0.20~250.00[Hz]
不足250.00[Hz]	0.20~(1/从轴的运算周期[s])[Hz]

- 在频率(Frequency)中设置了超出范围的值的情况下，将变为超出频率指定范围(错误代码：1AE6H)。
- 对于有效的频率，可以通过有效频率(ValidFrequency)进行确认。
- 更改频率(Frequency)的情况下，应在滤波器累计值(FilterPool)为“0.0”时进行更改。即使在“0.0”以外的状态下进行更改，在变为“0.0”之前也不被反映。
- 对于频率(Frequency)，不能更改为低于有效频率(ValidFrequency)的频率。设置了较低的频率的情况下，将变为超出频率指定范围(错误代码：1AE6H)。
- 在主轴重复正转/反转的运行模式中执行动作的过程中，滤波器累计值(FilterPool)可能会变为“0.0”。在滤波器动作未整定的状态下更改频率(Frequency)的值时，滤波器动作途中被中止，因此导致主轴与从轴的偏差。更改频率(Frequency)的情况下，应确认主轴停止后，按筛选时间常数等待之后再更改。

■滤波器净化(FilterPurge)

在滤波器累计值(FilterPool)为“0.0”以外的状态下,设置将有效(Enable)置为了FALSE时的动作。

设定值	内容
TRUE	<p>有效(Enable)变为了FALSE的情况下,执行动作直至滤波器累计值(FilterPool)变为“0.0”后,停止动作。</p> 
FALSE	<p>有效(Enable)变为了FALSE的情况下,与滤波器累计值(FilterPool)的状态无关,将立即停止从轴。</p> 

■ 主轴数据源选择 (MasterValueSource)

设置从轴 (Slave) 执行单轴同步控制的主轴 (Master) 的位置的类型。

根据主轴 (Master) 及从轴 (Slave) 的运算周期不相同的情况下及FB的执行顺序，动作可能会改变。

将实轴以外的轴类型设置为主轴 (Master) 并在主轴数据源选择 (MasterValueSource) 中设置了反馈值的情况下，将以与指令当前值相同的值执行动作。

设定值	内容
1: 指令当前值 (mcSetValue)	使用上次的运算周期中的主轴的指令位置。
2: 反馈值 (mcActualValue)	使用上次的运算周期中的主轴的反馈位置。
101: 最新指令当前值 (mcLatestSetValue)	使用本次的运算周期中的主轴的指令位置。
102: 最新反馈值 (mcLatestActualValue)	使用本次的运算周期中的主轴的反馈位置。

要点

在主轴数据源选择 (MasterValueSource) 中设置 “1: 指令当前值 (mcSetValue)”、“101: 最新指令当前值 (mcLatestSetValue)”，主轴由于伺服报警及紧急停止而变为了伺服OFF的情况下，值的变化量有可能变大。通过在主轴数据源选择 (MasterValueSource) 中设置 “2: 反馈值 (mcActualValue)”、“102: 最新反馈值 (mcLatestActualValue)” 可以防止。

注意事项

- 对于平滑滤波器，滤波器级数越大运算负载越增加。滤波器级数为5000时的运算负载的大致标准为415[μs]。
- 滤波器级数通过下述公式计算。小数点以下四舍五入。但是，四舍五入后的结果为“0”的情况下，将作为1级执行动作。
滤波器级数 = (1 / 频率 [Hz]) / 运算周期 [s]
- 对于平滑滤波器，将确保启动时滤波器处理中使用的存储器。存储器使用量可以通过下述计算公式求出。发生存储器容量不足的情况下，应更改插件MotionControl_AxisFilter参数 (Addon_MotionControl_AxisFilter) 的RAM最大容量 (AddonSystem.PrConst.MT_MotionControl_AxisFilter.RamSizeMax)。
存储器使用量 = 滤波器级数 × 8 [字节]
- 主轴及从轴的移动量较大且符合下述时，运算误差可能会变大。应重新设置主轴及从轴的单位设置，减小移动范围。误差变大的情况下，应解除同步状态或执行绝对位置指定的当前值更改。
- 控制中的滤波器级数的移动量的总和 (下述计算公式) 超出了可以双精度浮点表示的有效位数 (约15位) 时。

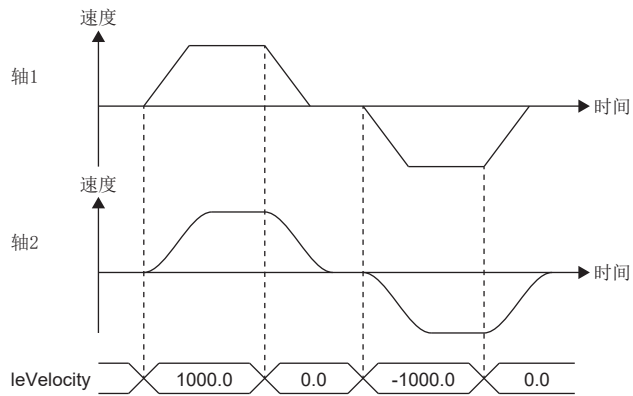
$$\sum_{i=0}^{\text{滤波器级数}-1} (i \text{ 运算周期前的主轴位置} - \text{控制开始时的主轴位置})$$

- 如果在指令滤波器的动作中进行滤波器无效 (FilterDisable) 的切换，从轴 (Slave) 的速度可能会急剧变化。
- 将指令滤波器置为有效后，从轴 (Slave) 中发生了停止原因，导致从轴 (Slave) 及后段的同步控制功能块中不再传送主轴 (Master) 的指令时，同步位置关系将发生破坏。再次将指令滤波器置为有效之前，应根据需要进行同步位置校准。

程序示例

进行将梯形加减速设置为平滑的加减速波形的滤波器处理的程序示例如下所示。

• 动作



• 设置

项目	设定值	
频率	10.00	
速度控制用数据设置	速度	5000.0
	加速度	100.0
	减速度	100.0
	Jerk	0.0

■ 轴

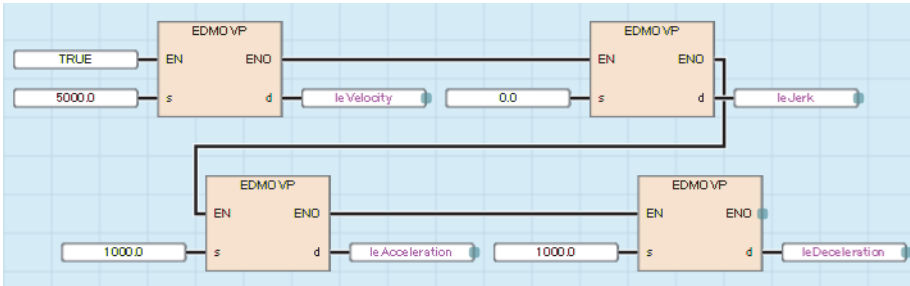
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1
2	Axis0002	AXIS_REF	轴2

■使用的标签

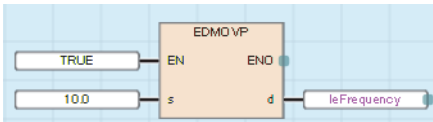
标签名	数据类型	注释
MCv_SpeedControl_1	MCv_SpeedControl	速度控制控制(包含循环)FB
bSpeedControl	位	速度控制指令
leVelocity	双精度实数	指令速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bSpeedInVelocity	位	目标速度到达
bSpeedBusy	位	执行中
bSpeedActive	位	控制中
bSpeedCommandAborted	位	执行中断
bSpeedError	位	错误
uwSpeedErrorID	字[无符号]/位串[16位]	错误代码
MCv_SmoothingFilter_1	MCv_SmoothingFilter	平滑滤波器FB
bSmoothingFilter	位	平滑滤波器指令
leFrequency	双精度实数	频率
bSmoothingValid	位	输出值有效
bSmoothingInVelocity	位	目标速度到达
bSmoothingBusy	位	执行中
bSmoothingActive	位	控制中
bSmoothingCommandAborted	位	执行中断
bSmoothingError	位	错误
uwSmoothingErrorID	字[无符号]/位串[16位]	错误代码
leSmoothingValidFrequency	双精度实数	有效频率
leSmoothingFilterPool	双精度实数	滤波器累计值

■FBD/LD程序

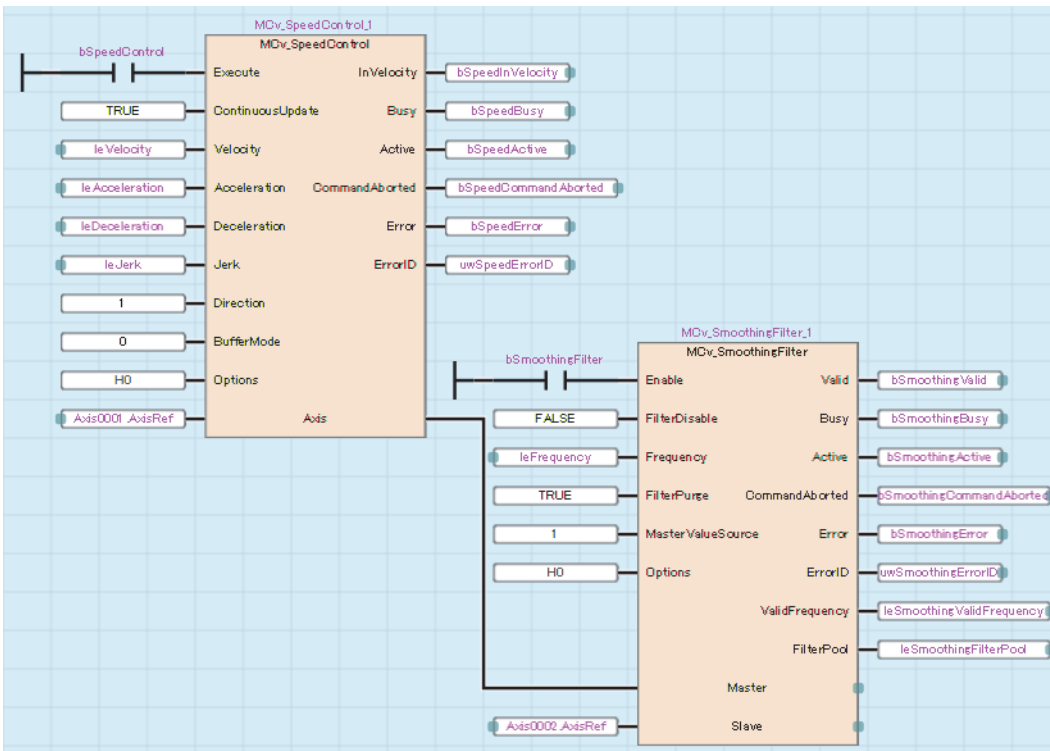
- 速度控制用数据的设置



- 频率



- 速度控制/平滑滤波器



■ST程序

```
//-----速度控制用数据的设置-----
leVelocity:= 5000.0;
leAcceleration:= 1000.0;
leDeceleration:= 1000.0;
leJerk:= 0.0;

//-----速度控制-----
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
);

//-----频率-----
leFrequency:= 10.00;

//-----平滑滤波器-----
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
);
```

46.19 移动方向限制滤波器

MCv_DirectionFilter

进行对设置的移动方向进行移动限制的滤波器处理。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Master	主轴	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName . AxisRef .), 请参阅下述内容。 ☞ 1387页 AxisName . AxisRef . (轴信息)
Slave	从轴	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName . AxisRef .), 请参阅下述内容。 ☞ 1387页 AxisName . AxisRef . (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Enable	有效	BOOL	始终	TRUE、FALSE	FALSE	设置为TRUE时, 将执行MCv_DirectionFilter(移动方向限制滤波器)。
FilterDisable	滤波器无效	BOOL	始终	TRUE、FALSE	FALSE	设置滤波器处理的方法。 • FALSE: 按照滤波器设置执行动作。 • TRUE: 滤波器将变为无效, 将主轴(Master)的输入原样不变地传送到从轴(Slave)。 关于详细内容, 请参阅下述章节。 ☞ 1789页 滤波器无效(FilterDisable)
PositiveDisable	正方向限制	BOOL	启动时	TRUE、FALSE	FALSE	设置为TRUE时, 将对至正方向的轴动作进行限制。主轴(Master)的正转动作时从轴(Slave)不动作。将正方向限制(PositiveDisable)与负方向限制(NegativeDisable)两者置为TRUE时, 即使将主轴(Master)动作从轴(Slave)也不动作。 • FALSE: 无限制 • TRUE: 有限制
NegativeDisable	负方向限制	BOOL	启动时	TRUE、FALSE	FALSE	设置为TRUE时, 将对至负方向的轴动作进行限制。主轴(Master)的反转动作时从轴(Slave)不动作。将正方向限制(PositiveDisable)与负方向限制(NegativeDisable)两者置为TRUE时, 即使将主轴(Master)动作从轴(Slave)也不动作。 • FALSE: 无限制 • TRUE: 有限制

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Filter	滤波器动作	INT (MC_VELOCITY_LIMIT_MODE)	启动时	0、2、4	2	设置滤波器动作。 <ul style="list-style-type: none"> • 0: 忽略(Ignore) • 2: 舍去(Truncate) • 4: 限定(无减速时斜率)(ClampWithoutRamp) 关于详细内容, 请参阅下述章节。 1790页 滤波器动作(Filter)
MasterValueSource	主轴数据源选择	INT (MC_SOURCE)	启动时	1、2、101、102	1	设置主轴(Master)的数据源。 <ul style="list-style-type: none"> • 1: 指令当前值(mcSetValue) • 2: 反馈值(mcActualValue) • 101: 最新指令当前值(mcLatestSetValue) • 102: 最新反馈值(mcLatestActualValue) 关于详细内容, 请参阅下述章节。 1791页 主轴数据源选择(MasterValueSource)
Options	选项	DWORD (HEX)	启动时	0000000H	0000000H	应设置“0000000H”。 * 设置了“0000000H”以外的情况下, 将变为超出Options范围(错误代码: 1ABBH)。

■输出变量

输出变量	名称	数据类型	初始值	内容
Valid	输出值有效	BOOL	FALSE	输出值为有效时, 将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_DirectionFilter(移动方向限制滤波器)时, 将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_DirectionFilter(移动方向限制滤波器)正在控制轴时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MCv_DirectionFilter(移动方向限制滤波器)的执行中断时, 将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时, 将返回错误代码。 关于错误代码的详细内容, 请参阅下述手册。 所使用的控制器的用户手册
FilterPool	滤波器累计值	LREAL	0.0	输出滤波器处理中累计的从轴的移动量。 在滤波器累计值(FilterPool)为“0.0”以外的状态下, 将有效(Enable)置为FALSE的情况下, 滤波器累计值(FilterPool)将保持FALSE设置时的状态。 滤波器累计值(FilterPool)在下次将有效(Enable)置为了TRUE时清除。 滤波器累计值(FilterPool)的值超出了定位范围的情况下, 将变为滤波器累计值溢出警告(事件代码: 00D13H)。 检测出警告时, 滤波器累计值(FilterPool)的误差可能会变大。

■公开变量

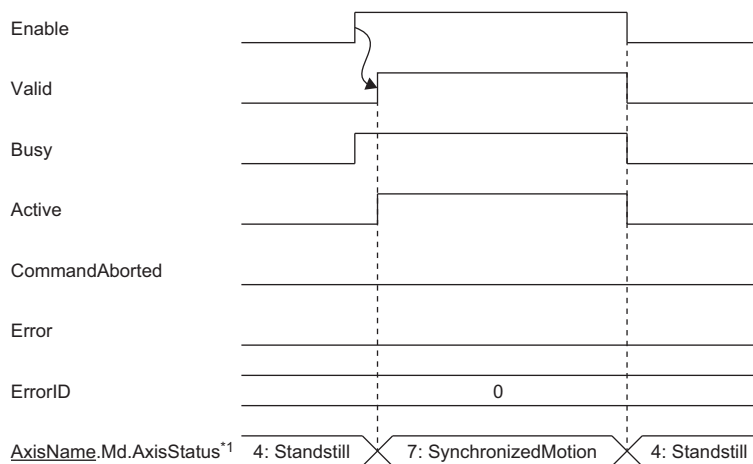
公开变量	名称	数据类型	初始值	内容
PurgedFilterPool	删除的滤波器累计值	LREAL	0.0	通过将滤波器无效(FilterDisable)置为TRUE存储删除的滤波器累计值(FilterPool)。 删除的滤波器累计值(PurgedFilterPool)不累计。 在FB启动时将被清零。

功能

- 移动方向限制滤波器是相对于主轴的移动，将从轴的运动限制为一个方向的滤波器。
- 移动方向限制滤波器将对主轴数据源选择(MasterValueSource)中设置的主轴(Master)的值执行了滤波器处理的结果输出到从轴(Slave)中。
- 移动方向限制滤波器执行中的从轴(Slave)的轴状态(AxisName.Md.AxisStatus)为“7: 同步运行中(SynchronizedMotion)”。
- 从轴处于其它动作FB中，启动了移动方向限制滤波器的情况下，将忽略启动请求，且变为启动不可(错误代码: 1AFCH)。应使从轴(Slave)的轴状态(AxisName.Md.AxisStatus)在“4: 待机中(Standstill)”时启动。
- 移动方向限制滤波器执行中，可多重启动的功能块仅为MC_Stop(强制停止)。
- 将有效(Enable)置为FALSE的情况下，应在从轴处于停止的状态下进行。从轴动作中将有效(Enable)置为FALSE时，从轴将立即停止。

■时序图

- 正常完成的情况下



*1 是从轴(Slave)的轴状态。

- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 1378页 通过有效(Enable)类型的运动控制FB的基本动作

■滤波器无效(FilterDisable)

设置滤波器处理的方法。

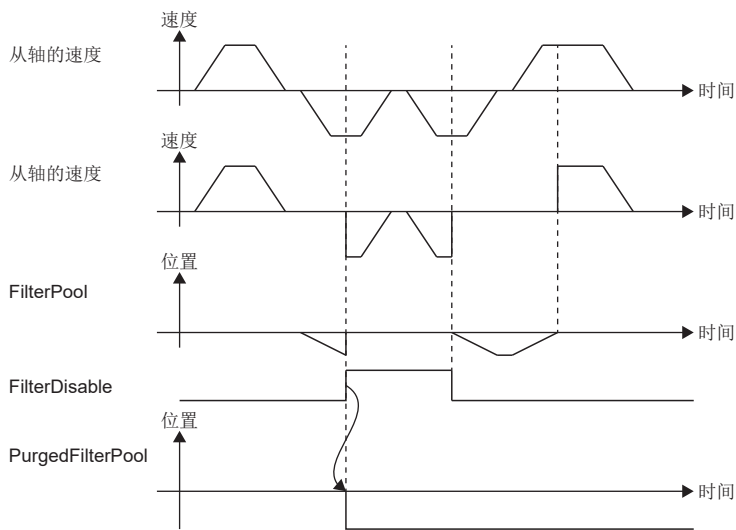
设定值	内容
FALSE	按照滤波器设置执行动作。
TRUE	滤波器将变为无效，将主轴 (Master) 的输入原样不变地传送到从轴 (Slave)。

- 将滤波器无效(FilterDisable)置为了TRUE的情况下，滤波器累计值(FilterPool)将变为“0.0”，并将删除的滤波器累计值(FilterPool)存储至删除的滤波器累计值(PurgedFilterPool)。

例

进行了下述设置的情况下

输入引脚	设定值
滤波器动作(Filter)	4: 限定(无减速时斜率)(ClampWithoutRamp)
正方向限制(PositiveDisable)	FALSE(无限制)
负方向限制(NegativeDisable)	TRUE(有限制)



■滤波器动作(Filter)

- 根据滤波器动作(Filter)的设置, 设置限制时的从轴的动作。

设定值	内容
0: 忽略(Ignore)	<ul style="list-style-type: none"> 不进行移动方向限制。 不更新滤波器累计值(FilterPool)。
2: 舍去(Truncate)	<ul style="list-style-type: none"> 忽略至限制方向的移动量, 且不将其加到滤波器累计值(FilterPool)中。 至限制方向的移动量被舍去, 因此主轴与从轴的位置关系将产生相当于至限制方向的移动量的偏差。 <p><例> 正方向限制(PositiveDisable)为FALSE, 负方向限制(NegativeDisable)为TRUE的情况下</p>
4: 限定(无减速时斜率) (ClampWithoutRamp)	<ul style="list-style-type: none"> 将至限制方向的移动量加到滤波器累计值(FilterPool)中。 滤波器累计值(FilterPool)将反映到至反转方向的移动中。因此, 即使重复至限制方向的动作, 滤波器累计值(FilterPool)为“0.0”的状态的主轴与从轴的位置关系也不会产生偏差。 <p><例> 正方向限制(PositiveDisable)为FALSE(无限制), 负方向限制(NegativeDisable)为TRUE(有限制)的情况下</p>

- 在滤波器动作(Filter)中设置了超出设置范围的值的情况下, 将变为超出滤波器动作指定范围(错误代码: 1AE7H)。

■ 主轴数据源选择 (MasterValueSource)

设置从轴 (Slave) 执行单轴同步控制的主轴 (Master) 的位置的类型。

根据主轴 (Master) 及从轴 (Slave) 的运算周期不相同的情况下及FB的执行顺序，动作可能会改变。

将实轴以外的轴类型设置为主轴 (Master) 并在主轴数据源选择 (MasterValueSource) 中设置了反馈值的情况下，将以与指令当前值相同的值执行动作。

设定值	内容
1: 指令当前值 (mcSetValue)	使用上次的运算周期中的主轴的指令位置。
2: 反馈值 (mcActualValue)	使用上次的运算周期中的主轴的反馈位置。
101: 最新指令当前值 (mcLatestSetValue)	使用本次的运算周期中的主轴的指令位置。
102: 最新反馈值 (mcLatestActualValue)	使用本次的运算周期中的主轴的反馈位置。

要点

在主轴数据源选择 (MasterValueSource) 中设置 “1: 指令当前值 (mcSetValue)”、“101: 最新指令当前值 (mcLatestSetValue)” ，主轴由于伺服报警及紧急停止而变为了伺服OFF的情况下，值的变化量有可能变大。通过在主轴数据源选择 (MasterValueSource) 中设置 “2: 反馈值 (mcActualValue)”、“102: 最新反馈值 (mcLatestActualValue)” 可以防止。

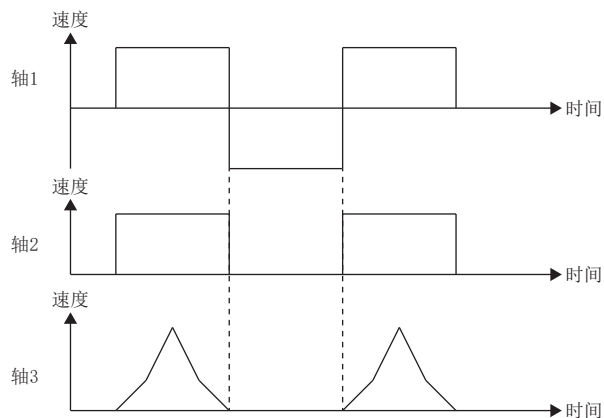
注意事项

- 如果在指令滤波器的动作中进行滤波器无效 (FilterDisable) 的切换，从轴 (Slave) 的速度可能会急剧变化。
- 将指令滤波器置为有效后，从轴 (Slave) 中发生了停止原因，导致从轴 (Slave) 及后段的同步控制功能块中不再传送主轴 (Master) 的指令时，同步位置关系将发生破坏。再次将指令滤波器置为有效之前，应根据需要进行同步位置校准。

程序示例

进行限制凸轮动作的方向的滤波器处理的程序示例如下所示。

• 动作



• 设置

项目	设定值
凸轮ID设置	1

■ 轴

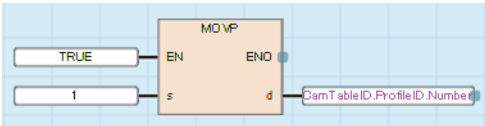
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1
2	Axis0002	AXIS_REF	轴2
3	Axis0003	AXIS_REF	轴3

■ 使用的标签

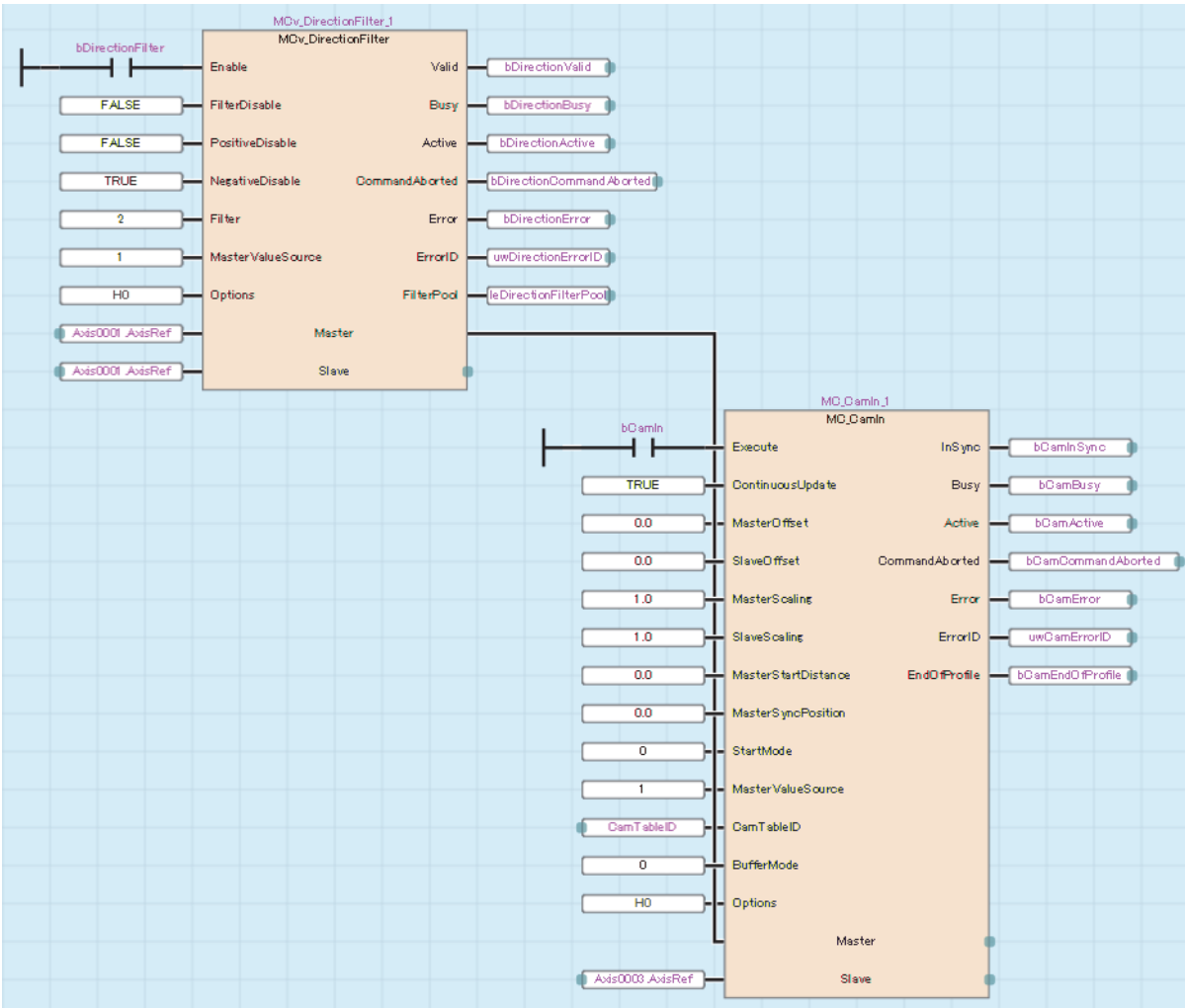
标签名	数据类型	注释
MCv_DirectionFilter_1	MCv_DirectionFilter	移动方向限制滤波器FB
bDirectionFilter	位	移动方向限制滤波器指令
bDirectionValid	位	输出值有效
bDirectionBusy	位	执行中
bDirectionActive	位	控制中
bDirectionCommandAborted	位	执行中断
bDirectionError	位	错误
uwDirectionErrorID	字[无符号]/位串[16位]	错误代码
leDirectionFilterPool	双精度实数	删除的滤波器累计值
MC_CamIn_1	MC_CamIn	凸轮动作开始FB
bCamIn	位	凸轮动作指令
CamTableID	MC_CAM_ID	凸轮ID
bCamInSync	位	同步中
bCamBusy	位	执行中
bCamActive	位	控制中
bCamCommandAborted	位	执行中断
bCamError	位	错误
uwCamErrorID	字[无符号]/位串[16位]	错误代码
bCamEndOfProfile	位	凸轮循环完成

■FBD/LD程序

- 凸轮ID设置



- 移动方向限制滤波器/凸轮动作



■ST程序

```
//-----移动方向限制滤波器-----
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
);

//凸轮ID设置
CamTableID.ProfileID.Number:= 1;

//凸轮动作
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__mcImmediate ,
    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
);
```

46.20 速度限制滤波器

MCv_SpeedLimitFilter

进行将速度限制为设置的限制值的滤波器处理。

梯形图	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:= ?WORD? , Valid=> ?BOOL? , Busy=> ?BOOL? , Active=> ?BOOL? , CommandAborted=> ?BOOL? , Error=> ?BOOL? , ErrorID=> ?WORD? , FilterPool=> ?LREAL?); </pre>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Master	主轴	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName.AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName.AxisRef. (轴信息)
Slave	从轴	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName.AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName.AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Enable	有效	BOOL	始终	TRUE、FALSE	FALSE	设置为TRUE时，将执行MCv_SpeedLimitFilter(速度限制滤波器)。
FilterDisable	滤波器无效	BOOL	始终	TRUE、FALSE	FALSE	设置滤波器处理的方法。 • FALSE：按照滤波器设置执行动作。 • TRUE：滤波器将变为无效，将主轴(Master)的输入原样不变地传送到从轴(Slave)。 关于详细内容，请参阅下述章节。 ☞ 1798页 滤波器无效(FilterDisable)
PositiveLimit	正方向限制值	LREAL	启动时	0.0、0.0001~2500000000.0	0.0	设置正方向的速度限制值。 设置对于主轴(Master)的正方向输入速度，输出到从轴(Slave)中的速度的上限值。 设置了“0.0”的情况下，从轴不向正方向执行动作。 设置了超出范围的值的的情况下，将变为超出方向限制值指定范围(错误代码：1AE8H)。
NegativeLimit	负方向限制值	LREAL	启动时	0.0、0.0001~2500000000.0	0.0	设置负方向的速度限制值。 设置对于主轴(Master)的负方向输入速度，输出到从轴(Slave)中的速度的上限值。 设置了“0.0”的情况下，从轴不向负方向执行动作。 设置了超出范围的值的的情况下，将变为超出方向限制值指定范围(错误代码：1AE8H)。

输入变量	名称	数据类型	获取	有效范围	初始值	内容
PositiveFilter	正方向滤波器动作	INT (MC_VELOCITY_LIMIT_MODE)	启动时	0、2、4	2	设置超出正方向限制值时的滤波器动作。 <ul style="list-style-type: none"> • 0: 忽略 (Ignore) • 2: 舍去 (Truncate) • 4: 限定 (无减速时斜率) (ClampWithoutRamp) 关于详细内容, 请参阅下述章节。 ☞ 1798页 正方向滤波器动作 (PositiveFilter)/ 负方向滤波器动作 (NegativeFilter)
NegativeFilter	负方向滤波器动作	INT (MC_VELOCITY_LIMIT_MODE)	启动时	0、2、4	2	设置超出负方向限制值时的滤波器动作。 <ul style="list-style-type: none"> • 0: 忽略 (Ignore) • 2: 舍去 (Truncate) • 4: 限定 (无减速时斜率) (ClampWithoutRamp) 关于详细内容, 请参阅下述章节。 ☞ 1798页 正方向滤波器动作 (PositiveFilter)/ 负方向滤波器动作 (NegativeFilter)
MasterValueSource	主轴数据源选择	INT (MC_SOURCE)	启动时	1、2、101、102	1	设置主轴 (Master) 的数据源。 <ul style="list-style-type: none"> • 1: 指令当前值 (mcSetValue) • 2: 反馈值 (mcActualValue) • 101: 最新指令当前值 (mcLatestSetValue) • 102: 最新反馈值 (mcLatestActualValue) 关于详细内容, 请参阅下述章节。 ☞ 1799页 主轴数据源选择 (MasterValueSource)
Options	选项	DWORD (HEX)	启动时	0000000H	0000000H	应设置“0000000H”。 * 设置了“0000000H”以外的情况下, 将变为超出Options范围 (错误代码: 1ABBH)。

■输出变量

输出变量	名称	数据类型	初始值	内容
Valid	输出值有效	BOOL	FALSE	输出值为有效时, 将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了Mcv_SpeedLimitFilter (速度限制滤波器) 时, 将变为TRUE。
Active	控制中	BOOL	FALSE	表示FB正在控制轴。 Mcv_SpeedLimitFilter (速度限制滤波器) 正在控制轴时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	Mcv_SpeedLimitFilter (速度限制滤波器) 的执行中断时, 将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时, 将返回错误代码。 关于错误代码的详细内容, 请参阅下述手册。 ☞ 所使用的控制器的用户手册
FilterPool	滤波器累计值	LREAL	0.0	输出滤波器处理中累计的从轴的移动量。 在滤波器累计值 (FilterPool) 为“0.0”以外的状态下, 将有效 (Enable) 置为FALSE的情况下, 滤波器累计值 (FilterPool) 将保持FALSE设置时的状态。 滤波器累计值 (FilterPool) 在下次将有效 (Enable) 置为了TRUE时清除。 滤波器累计值 (FilterPool) 的值超出了定位范围的情况下, 将变为滤波器累计值溢出警告 (事件代码: 00D13H)。 检测出警告时, 滤波器累计值 (FilterPool) 的误差可能会变大。

■公开变量

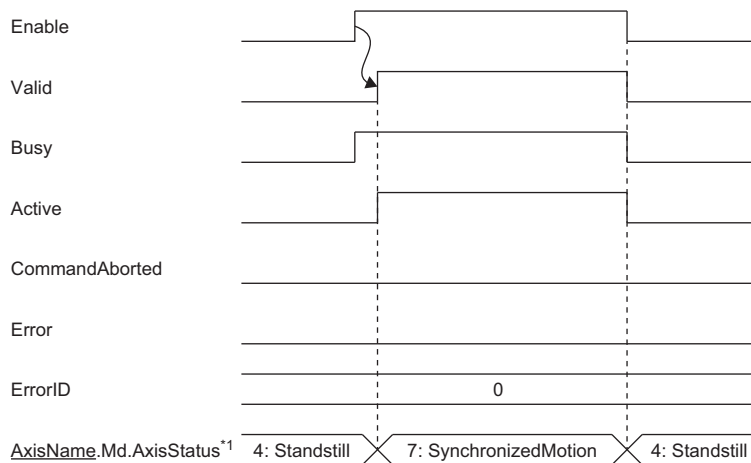
公开变量	名称	数据类型	初始值	内容
PurgedFilterPool	删除的滤波器累计值	LREAL	0.0	通过将滤波器无效 (FilterDisable) 置为TRUE存储删除的滤波器累计值 (FilterPool)。 删除的滤波器累计值 (PurgedFilterPool) 不累计。 在FB启动时将被清零。

功能

- 速度限制滤波器是对主轴的输入速度设置指定的限制值，将设置的限制值的速度输出到从轴中的滤波器。
- 速度限制滤波器将对主轴数据源选择 (MasterValueSource) 中设置的主轴 (Master) 的值执行了滤波器处理的结果输出到从轴 (Slave) 中。
- 速度限制滤波器执行中的从轴 (Slave) 的轴状态 (AxisName.Md.AxisStatus) 为 “7: 同步运行中 (SynchronizedMotion)”。
- 从轴处于其它动作FB中，启动了速度限制滤波器的情况下，将忽略启动请求，且变为启动不可 (错误代码: 1AFCH)。应使从轴 (Slave) 的轴状态 (AxisName.Md.AxisStatus) 在 “4: 待机中 (Standstill)” 时启动。
- 速度限制滤波器执行中，可多重启动的功能块仅为MC_Stop (强制停止)。
- 将有效 (Enable) 置为FALSE的情况下，应在从轴处于停止的状态下进行。从轴动作中将有效 (Enable) 置为FALSE时，从轴将立即停止。

■时序图

- 正常完成的情况下



*1 是从轴 (Slave) 的轴状态。

- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 1378页 通过有效 (Enable) 类型的运动控制FB的基本动作

■滤波器无效(FilterDisable)

设置滤波器处理的方法。

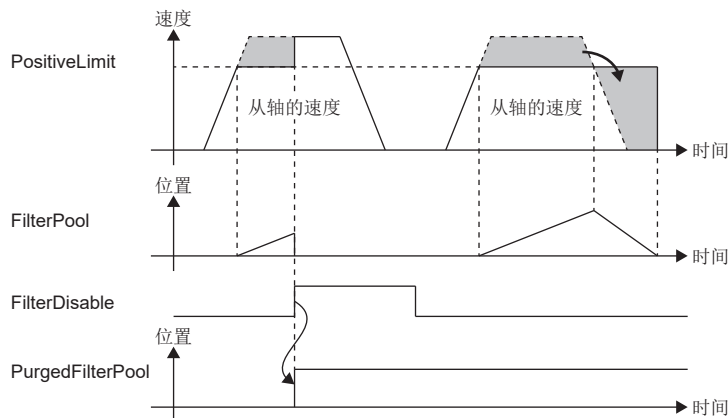
设定值	内容
FALSE	按照滤波器设置执行动作。
TRUE	滤波器将变为无效，将主轴(Master)的输入原样不变地传送到从轴(Slave)。

- 将滤波器无效(FilterDisable)置为了TRUE的情况下，滤波器累计值(FilterPool)将变为“0.0”，并将删除的滤波器累计值(FilterPool)存储至删除的滤波器累计值(PurgedFilterPool)。

例

进行了下述设置的情况下

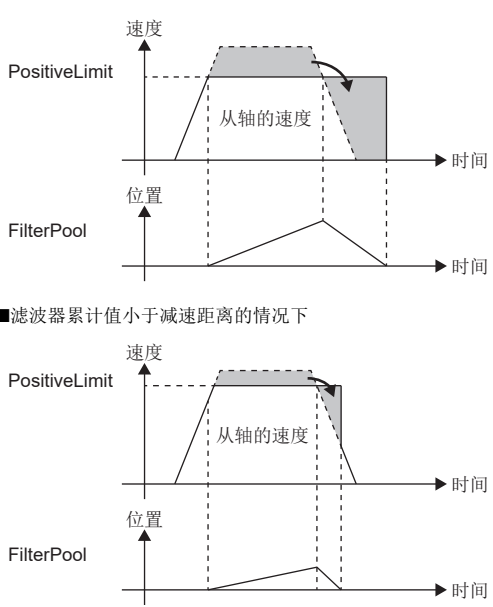
输入引脚	设定值
正方向滤波器动作(PositiveFilter)	4: 限定(无减速时斜率)(ClampWithoutRamp)



■正方向滤波器动作(PositiveFilter)/负方向滤波器动作(NegativeFilter)

- 根据正方向滤波器动作(PositiveFilter)与负方向滤波器动作(NegativeFilter)的设置，设置限制时的从轴的动作。

设定值	内容
0: 忽略(Ignore)	<ul style="list-style-type: none"> • 不进行速度限制。 • 不更新滤波器累计值(FilterPool)。
2: 舍去(Truncate)	<ul style="list-style-type: none"> • 超出了限制值的部分不输出到从轴(Slave)中。 • 限制的移动量被舍去，因此主轴与从轴的位置关系将产生相当于舍去部分的偏差。

设定值	内容
4: 限定(无减速时斜率) (ClampWithoutRamp)	<ul style="list-style-type: none"> 超出了限制值的部分将作为滤波器累计值(FilterPool)累计,且累计部分将被延迟并输出。滤波器累计值(FilterPool)的输出不进行减速动作。 在限制值中设置了“0.0”的情况下,至相应方向的主轴的输入将全部作为滤波器累计值(FilterPool)累计,且不向从轴延迟后输出。在反方向的主轴的输入中减去累计部分,且在滤波器累计值(FilterPool)变为“0.0”之前从轴不执行动作。 <p>■滤波器累计值大于减速距离的情况下</p>  <p>■滤波器累计值小于减速距离的情况下</p>

• 在滤波器动作(Filter)中设置了超出设置范围的值的情况下,将变为超出滤波器动作指定范围(错误代码:1AE7H)。

■主轴数据源选择(MasterValueSource)

设置从轴(Slave)执行单轴同步控制的主轴(Master)的位置的类型。

根据主轴(Master)及从轴(Slave)的运算周期不相同的情况下及FB的执行顺序,动作可能会改变。

将实轴以外的轴类型设置为主轴(Master)并在主轴数据源选择(MasterValueSource)中设置了反馈值的情况下,将以与指令当前值相同的值执行动作。

设定值	内容
1: 指令当前值(mcSetValue)	使用上次的运算周期中的主轴的指令位置。
2: 反馈值(mcActualValue)	使用上次的运算周期中的主轴的反馈位置。
101: 最新指令当前值(mcLatestSetValue)	使用本次的运算周期中的主轴的指令位置。
102: 最新反馈值(mcLatestActualValue)	使用本次的运算周期中的主轴的反馈位置。

要点

在主轴数据源选择(MasterValueSource)中设置“1: 指令当前值(mcSetValue)”、“101: 最新指令当前值(mcLatestSetValue)”,主轴由于伺服报警及紧急停止而变为了伺服OFF的情况下,值的变化量有可能变大。通过在主轴数据源选择(MasterValueSource)中设置“2: 反馈值(mcActualValue)”、“102: 最新反馈值(mcLatestActualValue)”可以防止。

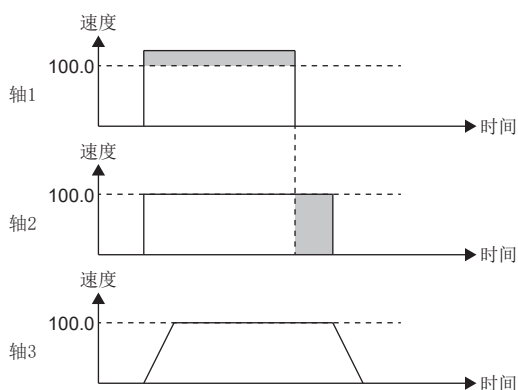
注意事项

- 如果在指令滤波器的动作中进行滤波器无效(FilterDisable)的切换,从轴(Slave)的速度可能会急剧变化。
- 将指令滤波器置为有效后,从轴(Slave)中发生了停止原因,导致从轴(Slave)及后段的同步控制功能块中不再传送主轴(Master)的指令时,同步位置关系将发生破坏。再次将指令滤波器置为有效之前,应根据需要进行同步位置校准。

程序示例

进行限制同步编码器的速度的滤波器处理的程序示例如下所示。

• 动作



• 设置

项目	设定值	
正方向/负方向限制值	正方向限制值	100.0
	负方向限制值	0.0
齿隙量	1.0	

■轴

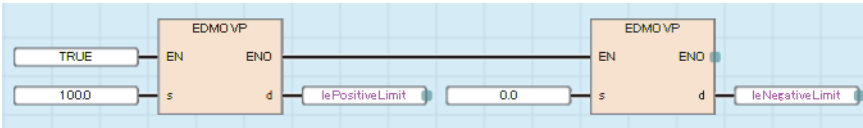
轴No.	标签名	数据类型	注释
1	EncoderAxis0001	AXIS_REF	轴1
2	Axis0002	AXIS_REF	轴2
3	Axis0003	AXIS_REF	轴3

■使用的标签

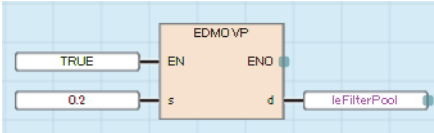
标签名	数据类型	注释
MCv_SpeedLimitFilter_1	MCv_SpeedLimitFilter	速度限制滤波器FB
bSpeedLimitFilter	位	速度限制滤波器指令
lePositiveLimit	双精度实数	正方向限制值
leNegativeLimit	双精度实数	负方向限制值
bSpeedLimitValid	位	输出值有效
bSpeedLimitBusy	位	执行中
bSpeedLimitActive	位	控制中
bSpeedLimitCommandAborted	位	执行中断
bSpeedLimitError	位	错误
uwSpeedLimitErrorID	字[无符号]/位串[16位]	错误代码
leFilterPool	双精度实数	滤波器累计值
MCv_SmoothingFilter_1	MCv_SmoothingFilter	平滑滤波器FB
bSmoothingFilter	位	平滑滤波器指令
leFrequency	双精度实数	频率
bSmoothingValid	位	输出值有效
bSmoothingInVelocity	位	目标速度到达
bSmoothingBusy	位	执行中
bSmoothingActive	位	控制中
bSmoothingCommandAborted	位	执行中断
bSmoothingError	位	错误
uwSmoothingErrorID	字[无符号]/位串[16位]	错误代码
leSmoothingValidFrequency	双精度实数	有效频率
leSmoothingFilterPool	双精度实数	滤波器累计值

■FBD/LD程序

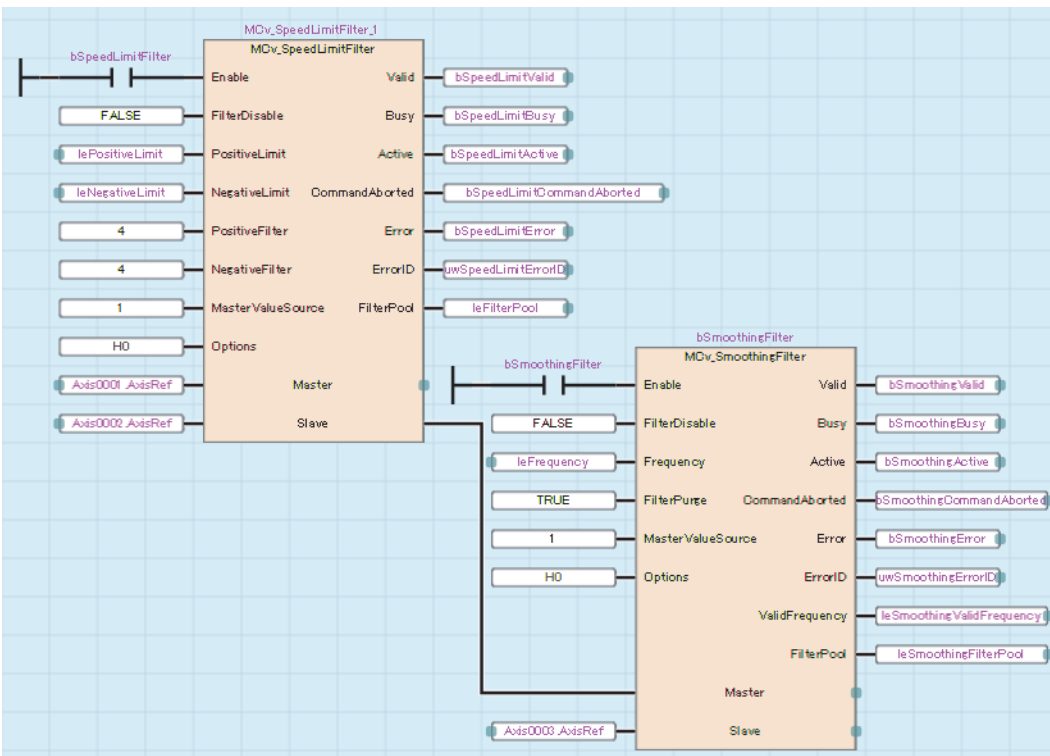
- 正方向/负方向限制值



- 频率



- 速度限制滤波器/平滑滤波器



■ST程序

```
//-----正方向/负方向限制值-----
lePositiveLimit:= 100.0;
leNegativeLimit:= 0.0;

//-----速度限制滤波器-----
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
);

//-----频率-----
leFrequency:= 0.20;

//-----平滑滤波器-----
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
);
```

46.21 轴控制开始等待

MCv_MoveWait

通过允许执行信号的输入即时执行指定轴多重启动了的运行系统FB的运行。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(<u>AxisName</u> .AxisRef.)，请参阅下述内容。 ☞ 1387页 <u>AxisName</u> .AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	通过TRUE执行MCv_MoveWait(轴控制开始等待)。
ExecutionPermission	执行允许信号	MC_INPUT_REF	启动时	—	不能省略	设置用于解除轴控制开始等待的信号。
Options	选项	DWORD(HEX)	启动时	00000000H	00000000H	应设置“00000000H”。 设置“00000000H”以外的情况下，将变为错误。

■输出变量

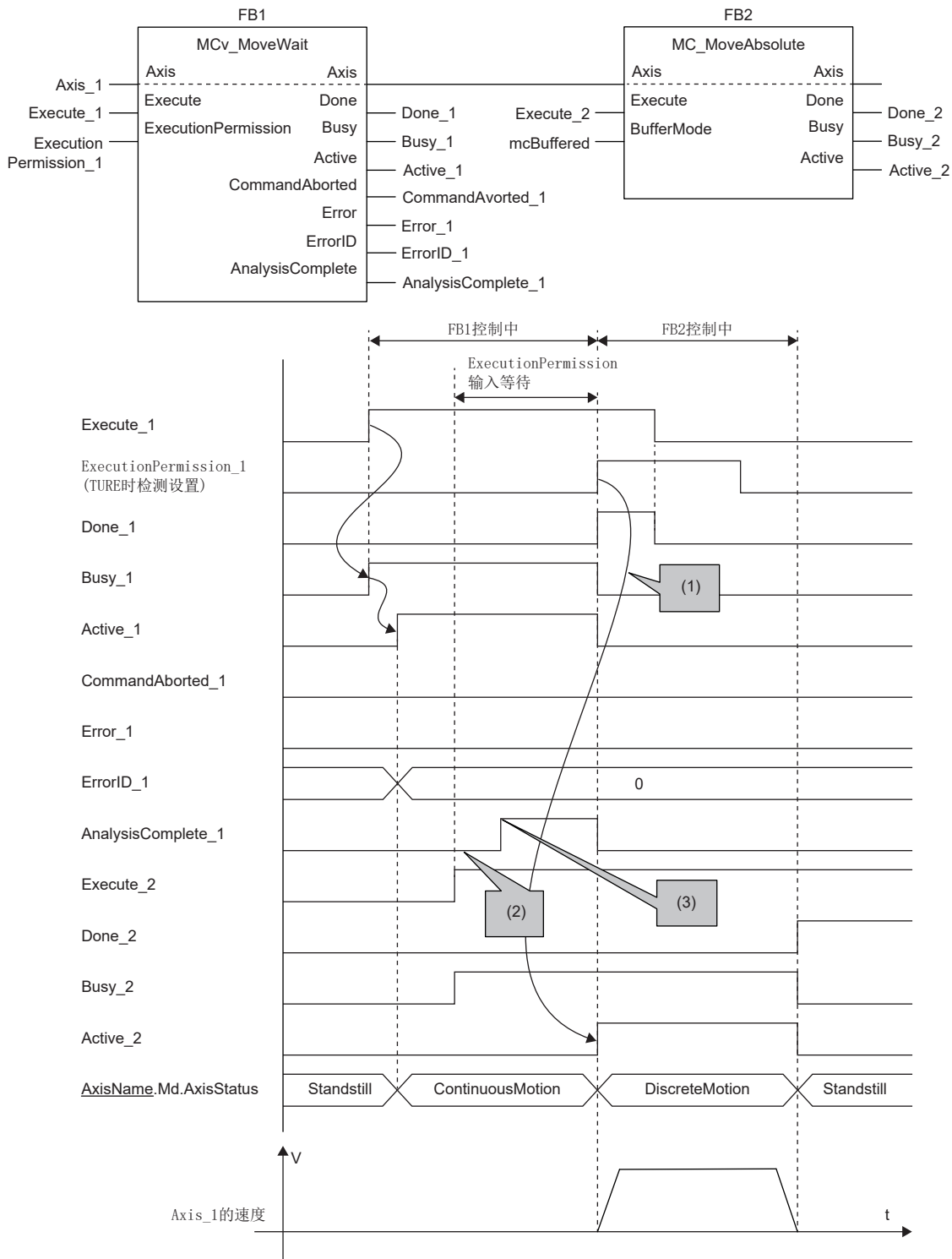
输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	表示在ExecutionPermission(执行允许信号)中设置的信号的输入检测以及AnalysisComplete(分析完成)变为TRUE时，变为TRUE，MCv_MoveWait(轴控制开始等待)的控制已完成。
Busy	执行中	BOOL	FALSE	MCv_MoveWait(轴控制开始等待)执行中时，将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_MoveWait(轴控制开始等待)轴控制中时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	通过其他FB中断执行时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 ☞ 所使用的控制器的用户手册
AnalysisComplete	解析完成	BOOL	FALSE	多重启动的运行系统FB的分析完成时，将变为TRUE。

功能

- 在ExecutionPermission(执行允许信号)的输入时机立即开始多重启动的单轴控制的运行系统FB的轴动作。
- MCv_MoveWait(轴控制开始等待)启动中, 如果对其他运行系统FB进行多重启动, 多重启动的FB将执行至解析完成为止, 待机直至输入ExecutionPermission(执行允许信号)。多重启动的FB的解析完成时机可通过AnalysisComplete(解析完成)进行确认。
- 应在AnalysisComplete(分析完成)变为TRUE后执行ExecutionPermission(执行允许信号)。
- 在输入ExecutionPermission(执行允许信号)之前的期间, 轴状态 (AxisName. Md. AxisStatus)将变为“6: 连续动作运行中(ContinuousMotion)”。
- 输入ExecutionPermission(执行允许信号)时, 在检测出信号输入的1个运算周期后执行多重启动的FB。此时, 轴状态 (AxisName. Md. AxisStatus)将转换为与多重启动的FB对应的状态。
- 应在AnalysisComplete(分析完成)变为TRUE后输入ExecutionPermission(执行允许信号)。

■ 时序图

• 正常完成的情况下



- (1) 检测到ExecutionPermission输入的1运算周期后, FB2将启动。
- (2) FB2的解析中
- (3) FB2的解析完成

• 异常完成的情况下

关于异常完成时的时序图, 请参阅下述章节。

☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■ExecutionPermission(执行允许信号)

指定用于解除MCv_MoveWait(轴控制开始等待)的信号。

ExecutionPermission(执行允许信号)通过MC_INPUT_REF结构体进行设置。关于MC_INPUT_REF结构体,请参阅下述章节。

☞ 1426页 MC_INPUT_REF[M]

ExecutionPermission(执行允许信号)中设置的输入信号(ExecutionPermission.Signal)的设置范围如下所示。

输入信号(ExecutionPermission.Signal)的类型为SIGNAL_SELECT结构体。

结构体	变量名	类型	设置范围
SIGNAL_SELECT(信号选择)	Source.Target(信号)*1	TARGET_REF ☞ 1428页 TARGET_REF(输入信号)	<ul style="list-style-type: none"> ■类型 • BOOL ■数据类型 • [OBJ] • [VAR] • [DEV] • [CONST]
	Detection(信号检测方法)	INT (MC_SIGNAL_LOGIC) ☞ 1459页 MC_SIGNAL_LOGIC[M]	<ul style="list-style-type: none"> • 0: TRUE时检测(HighLevel) • 1: FALSE时检测(LowLevel)
	CompensationTime(补偿时间)*2	LREAL	-5.0~0.0[s]
	FilterTime(筛选时间)	LREAL	0.0~5.0[s]

*1 无法使用外部信号高精度输入。指定了外部信号高精度输入用信号的情况下,将与普通信号相同的精度进行动作。

*2 在补偿时间(ExecutionPermission.Signal.CompensationTime)中指定了小于1运算周期的值的情况下,将作为0.0(无补偿)执行动作。
[OBJ]指定时省略了对象修饰的情况下,参照轴(Axis)的轴的对象。但是,轴(Axis)的轴不具有站地址,或站地址未设置的情况下,输出外部信号选择不正确(错误代码:1B31H、1B32H、1BD6H、1BD7H)。

[OBJ]指定时参照的站在轴模拟功能有效且站地址已设置的情况下,参照正在模拟的对象。

对ExecutionPermission(执行允许信号)的前进进行补偿的情况下,或希望调整多重启动的运行系统FB的执行时机的情况下,应设置补偿时间(ExecutionPermission.Signal.CompensationTime)。

希望防止由于ExecutionPermission(执行允许信号)的抖动而导致的信号误检测的情况下,应设置筛选时间(ExecutionPermission.Signal.FilterTime)。

信号名(ExecutionPermission.Signal.Source.Target)中使用了无法使用的信号的情况下,输出外部信号选择字符串不正确(错误代码:1B05H)、外部信号选择不正确(错误代码:1B31H、1B32H、1BD6H、1BD7H)、插件SignalIO内部异常(错误代码:1B33H、1BD8H)的任意一个。

注意事项

- ExecutionPermission (执行允许信号) 的输入应在AnalysisComplete (分析完成) 变为TRUE后进行。多重启动的运行系统FB不存在的状态或分析未完成的状态 (AnalysisComplete (分析完成) 为FALSE的状态)，即使检测出ExecutionPermission (执行允许信号) 的信号输入，Done (执行完成) 也不变为TRUE。
- 应在MCv_MoveWait (轴控制开始等待) 的Active (执行中) 变为TRUE之后再多重启动运行系统FB。执行MCv_MoveWait (轴控制开始等待) 时发生了错误的情况下，有可能不等待ExecutionPermission (执行允许信号) 的输入而执行多重启动的运行系统FB。
- 使用运算周期混合功能时，应将MCv_MoveWait (轴控制开始等待) 的对象轴的运算周期与ExecutionPermission (执行允许信号) 的刷新周期设置为相同。刷新周期不同的情况下，定位的开始时机可能会发生偏差。
- 同时启动不同运算周期的轴时，应将轴的运算周期设置为相同。轴的启动时机可能会偏离1个运算周期。希望同时启动不同运算周期的轴的情况下，应使用信号名 (ExecutionPermission.Signal.Source.Target) 与信号逻辑 (ExecutionPermission.Signal.Detection) 相同设置的多个ExecutionPermission (执行允许信号)，调整补偿时间 (ExecutionPermission.Signal.CompensationTime)。
- 执行MCv_MoveWait (轴控制开始等待) 时，应确认轴状态 (AxisName.Md.AxisStatus) 为“4: 待机中 (Standstill)”。无法启动的轴状态 (AxisName.Md.AxisStatus) 时，将发生不可启动 (错误代码: 1AFCH)，停止运行中的控制。
- 在其他运行系统FB执行中启动了MCv_MoveWait (轴控制开始等待) 的情况下，启动请求将被忽略，并发生运行中启动警告 (事件代码: 00D01H)。
- 应在多重启动的运行系统FB的缓冲模式 (BufferMode) 中设置Buffered。设置了Aborting并多重启动的情况下，中断MCv_MoveWait (轴控制开始等待)，ExecutionPermission (执行允许信号) 的输入状态将被忽略，并立即执行多重启动的运行系统FB。设置了Blending的多重启动的情况下，将发生BufferMode不可指定 (错误代码: 1AE5H)。

程序示例

通过将控制开始(bExecute)置为TRUE，对MC_MoveRelative和MCv_MoveLinearInterpolateRelative进行解析并待机，通过将执行允许信号(bExecutionPermission)置为TRUE，同时启动待机中的FB的程序示例如下所示。

■轴

轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REAL	轴1
2	Axis0002	AXIS_REAL	轴2
3	Axis0003	AXIS_REAL	轴3

■轴组

轴组No.	标签名	数据类型	注释
1	AxisGroup0001	AXIS_GROUP	轴组1

项目	设定值
配置轴[1]	Axis0001
配置轴[2]	Axis0002

■输入输出数据

标签名	数据类型	数据类型	源类型	源	信号检测方法	补偿时间	筛选时间
SignalData0001	输入信号	MC_INPUT_REF	标签	bExecutionPermission	HighLevel	0.0	0.0

■使用的标签

标签名	数据类型	注释
bExecute	位	控制开始
bExecutionPermission	位	执行允许信号
MCv_MoveWait_1	MCv_MoveWait	轴控制开始等待FB
MC_MoveRelative_1	MC_MoveRelative	相对值定位FB
MCv_GroupMoveWait_1	MCv_GroupMoveWait	轴组控制开始等待FB
MCv_MoveLinearInterpolateRelative_1	MCv_MoveLinearInterpolateRelative	相对值直线插补控制FB
wMoveLinearInterpolateRelativeLinearAxes	字[有符号](0..15)	直线插补轴
lMoveLinearInterpolateRelativePosition	双精度实数(0..15)	目标位置

■ST程序

```

SET( MCv_MoveWait_1.Active , bMoveRelativeExecute );
SET( MCv_GroupMoveWait_1.Active , bMoveLinearInterpolateRelativeExecute );
RST( NOT bExecute , bMoveRelativeExecute );
RST( NOT bExecute , bMoveLinearInterpolateRelativeExecute );
//-----轴控制开始等待-----
MCv_MoveWait_1(
    Axis:= Axis0001.AxisRef ,
    Execute:= bExecute ,
    ExecutionPermission:= SignalData0001 ,
    Options:= H00000000
);
//-----相对值定位-----
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
);
//-----轴组控制开始等待-----
MCv_GroupMoveWait_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bExecute ,
    ExecutionPermission:= SignalData0001 ,
    Options:= H00000000
);
//-----相对值直线插补控制用数据的设置-----
wMoveLinearInterpolateRelativeLinearAxes[0] := 1;
wMoveLinearInterpolateRelativeLinearAxes[1] := 2;
leMoveLinearInterpolateRelativeDistance[0] := 10000.0;
leMoveLinearInterpolateRelativeDistance[1] := 10000.0;
//-----相对值直线插补控制-----
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
);

```

46.22 轴组控制开始等待

MCv_GroupMoveWait

通过允许执行信号的输入即时执行指定轴组多重启动了的运行系统FB的运行。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
AxesGroup	轴组信息	AXES_GROUP_REF	启动时	—	不能省略	设置轴组。 关于使用的变量 (<u>AxesGroupName</u> , <u>AxesGroupRef</u> .), 请参阅下述内容。 ☞ 1408页 <u>AxesGroupName</u> , <u>AxesGroupRef</u> . (轴组信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	在TRUE中执行MCv_GroupMoveWait (轴组控制开始等待)。
ExecutionPermission	执行允许信号	MC_INPUT_REF	启动时	—	不能省略	设置用于解除轴组控制开始等待的信号。
Options	选项	DWORD (HEX)	启动时	00000000H	00000000H	应设置“00000000H”。 设置“00000000H”以外的情况下, 将变为错误。

■输出变量

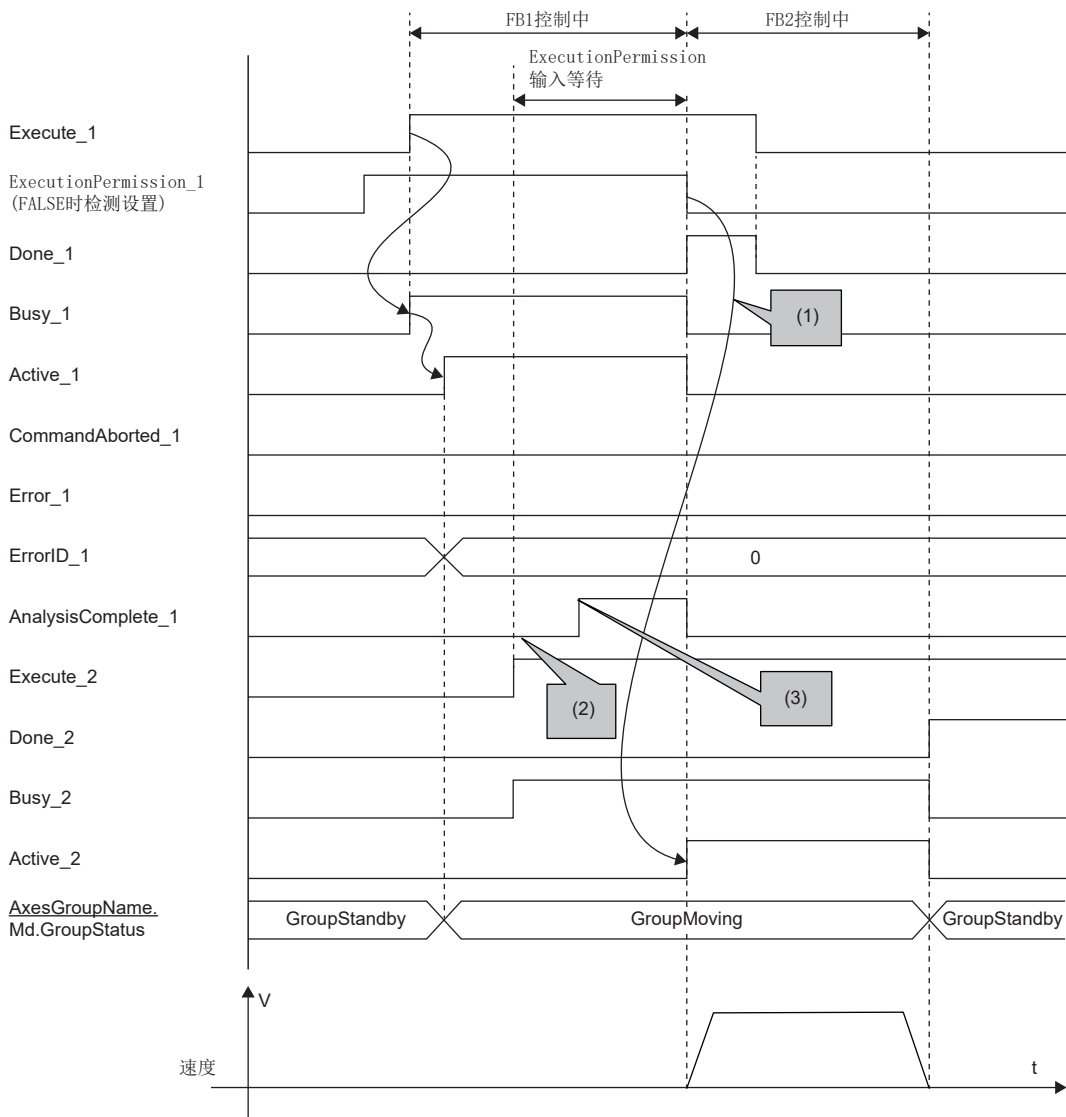
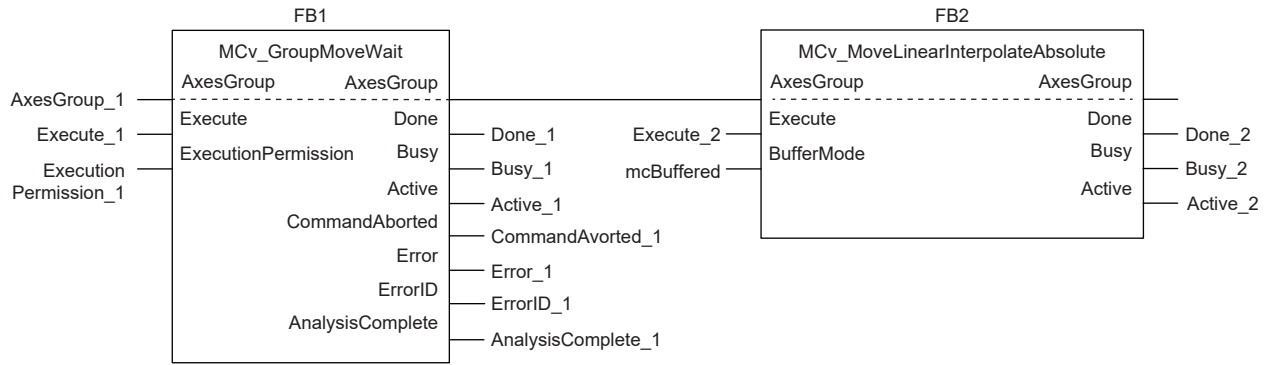
输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	表示在ExecutionPermission(执行允许信号)中设置的信号的输入检测以及AnalysisComplete(分析完成)变为TRUE时, 变为TRUE, MCv_GroupMoveWait (轴组控制开始等待)的控制已完成。
Busy	执行中	BOOL	FALSE	MCv_GroupMoveWait (轴组控制开始等待)执行中时, 将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_GroupMoveWait (轴组控制开始等待)轴控制中时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	通过其他FB中断执行时, 将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时, 将返回错误代码。 关于错误代码的详细内容, 请参阅下述手册。 ☞ 所使用的控制器的用户手册
AnalysisComplete	解析完成	BOOL	FALSE	多重启动的运行系统FB的分析完成时, 将变为TRUE。

功能

- 在ExecutionPermission(执行允许信号)的输入时机立即执行多重启动的多轴控制的运行系统FB的轴组动作。
- MCv_GroupMoveWait(轴组控制开始等待)启动中, 如果对其他运行系统FB进行多重启动, 多重启动的FB将执行至解析完成为止, 待机直至输入ExecutionPermission(执行允许信号)。多重启动的FB的解析完成时机可通过AnalysisComplete(解析完成)进行确认。
- 应在AnalysisComplete(分析完成)变为TRUE后执行ExecutionPermission(执行允许信号)。
- 在输入ExecutionPermission(执行允许信号)之前的期间, 轴组状态(AxisGroupName.Md.GroupStatus)将变为“5: 动作中(GroupMoving)”。
- 输入ExecutionPermission(执行允许信号)时, 在检测出信号输入的1个运算周期后执行多重启动的FB。
- 应在AnalysisComplete(分析完成)变为TRUE后输入ExecutionPermission(执行允许信号)。

■ 时序图

- 正常完成的情况下



- (1) 检测出ExecutionPermission输入的1运算周期后，FB2将启动。
- (2) FB2的解析中
- (3) FB2的解析完成

- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■ExecutionPermission(执行允许信号)

指定用于解除MCv_GroupMoveWait(轴组控制开始等待)的信号。

ExecutionPermission(执行允许信号)通过MC_INPUT_REF结构体进行设置。关于MC_INPUT_REF结构体, 请参阅下述章节。

☞ 1426页 MC_INPUT_REF[M]

ExecutionPermission(执行允许信号)中设置的输入信号(ExecutionPermission.Signal)的设置范围如下所示。

输入信号(ExecutionPermission.Signal)的类型为SIGNAL_SELECT结构体。

结构体	变量名	类型	设置范围
SIGNAL_SELECT(信号选择)	Source.Target(信号)*1	TARGET_REF ☞ 1428页 TARGET_REF(输入信号)	<ul style="list-style-type: none"> ■类型 • BOOL ■数据类型 • [OBJ] • [VAR] • [DEV] • [CONST]
	Detection(信号检测方法)	INT (MC_SIGNAL_LOGIC) ☞ 1459页 MC_SIGNAL_LOGIC[M]	<ul style="list-style-type: none"> • 0: TRUE时检测(HighLevel) • 1: FALSE时检测(LowLevel)
	CompensationTime(补偿时间)*2	LREAL	-5.0~0.0[s]
	FilterTime(筛选时间)	LREAL	0.0~5.0[s]

*1 无法使用外部信号高精度输入。指定了外部信号高精度输入用信号的情况下, 将以与普通信号相同的精度进行动作。

*2 在补偿时间(ExecutionPermission.Signal.CompensationTime)中指定了小于1运算周期的值的情况下, 将作为0.0(无补偿)执行动作。

[OBJ]指定时省略了对象修饰的情况下, 输出外部信号选择不正确(错误代码: 1B31H、1B32H、1BD6H、1BD7H)。

[OBJ]指定时参照的站在轴模拟功能有效且站地址已设置的情况下, 参照正在模拟的对象。

对ExecutionPermission(执行允许信号)的前进进行补偿的情况下, 希望调整多重启动的运行系统FB的执行时机的情况下, 应设置补偿时间(ExecutionPermission.Signal.CompensationTime)。

希望防止由于ExecutionPermission(执行允许信号)的抖动而导致的信号误检测的情况下, 应设置筛选时间(ExecutionPermission.Signal.FilterTime)。

信号名(ExecutionPermission.Signal.Source.Target)中使用了无法使用的信号的情况下, 输出外部信号选择字符串不正确(错误代码: 1B05H)、外部信号选择不正确(错误代码: 1B31H、1B32H、1BD6H、1BD7H)、插件SignalIO内部异常(错误代码: 1B33H、1BD8H)的任意一个。

注意事项

- ExecutionPermission(执行允许信号)的输入应在AnalysisComplete(分析完成)变为TRUE后进行。多重启动的运行系统FB不存在的状态或分析未完成的状态(AnalysisComplete(分析完成)为FALSE的状态)，即使检测出ExecutionPermission(执行允许信号)的信号输入，Done(执行完成)也不变为TRUE。
- 应在MCv_GroupMoveWait(轴组控制开始等待)的Active(执行中)变为TRUE之后再多重启动运行系统FB。执行MCv_GroupMoveWait(轴组控制开始等待)时发生了错误的情况下，有可能不等待ExecutionPermission(执行允许信号)的输入而执行多重启动的运行系统FB。
- 使用运算周期混合功能时，应将MCv_GroupMoveWait(轴组控制开始等待)的对象轴的运算周期与ExecutionPermission(执行允许信号)的刷新周期设置为相同。刷新周期不同的情况下，定位的开始时机可能会发生偏差。
- 同时启动不同运算周期的轴时，应将轴的运算周期设置为相同。轴的启动时机可能会偏离1个运算周期。希望同时启动不同运算周期的轴的情况下，应使用信号名(ExecutionPermission.Signal.Source.Target)与信号逻辑(ExecutionPermission.Signal.Detection)相同设置的多个ExecutionPermission(执行允许信号)，调整补偿时间(ExecutionPermission.Signal.CompensationTime)。
- 执行MCv_GroupMoveWait(轴组控制开始等待)时，应确认轴组状态(AxisGroupName.Md.GroupStatus)为“4: 待机中(GroupStandby)”。为无法启动的轴状态(AxisGroupName.Md.GroupStatus)时，将发生不可启动(错误代码: 1AFCH)，停止运行中的控制。
- 在其他运行系统FB执行中启动了MCv_GroupMoveWait(轴组控制开始等待)的情况下，启动请求将被忽略，将发生运行中启动警告(事件代码: 00D01H)。
- 应在多重启动的运行系统FB的缓冲模式(BufferMode)中设置Buffered。设置了Aborting并多重启动的情况下，中断MCv_GroupMoveWait(轴组控制开始等待)，ExecutionPermission(执行允许信号)的输入状态将被忽略，并立即执行多重启动的运行系统FB。设置了Blending的多重启动的情况下，将发生BufferMode不可指定(错误代码: 1AE5H)。

程序示例

通过将控制开始(bExecute)置为TRUE, 对MC_MoveRelative和MCv_MoveLinearInterpolateRelative进行解析并待机, 通过将执行允许信号(bExecutionPermission)置为TRUE, 同时启动待机中的FB的程序示例如下所示。

■轴

轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REAL	轴1
2	Axis0002	AXIS_REAL	轴2
3	Axis0003	AXIS_REAL	轴3

■轴组

轴组No.	标签名	数据类型	注释
1	AxisGroup0001	AXIS_GROUP	轴组1

项目	设定值
配置轴[1]	Axis0001
配置轴[2]	Axis0002

■输入输出数据

标签名	数据类型	数据类型	源类型	源	信号检测方法	补偿时间	筛选时间
SignalData0001	输入信号	MC_INPUT_REF	标签	bExecutionPermission	HighLevel	0.0	0.0

■使用的标签

标签名	数据类型	注释
bExecute	位	控制开始
bExecutionPermission	位	执行允许信号
MCv_MoveWait_1	MCv_MoveWait	轴控制开始等待FB
MC_MoveRelative_1	MC_MoveRelative	相对值定位FB
MCv_GroupMoveWait_1	MCv_GroupMoveWait	轴组控制开始等待FB
MCv_MoveLinearInterpolateRelative_1	MCv_MoveLinearInterpolateRelative	相对值直线插补控制FB
wMoveLinearInterpolateRelativeLinearAxes	字[有符号](0..15)	直线插补轴
lMoveLinearInterpolateRelativePosition	双精度实数(0..15)	目标位置

■ST程序

```
SET( MCv_MoveWait_1.Active , bMoveRelativeExecute );
SET( MCv_GroupMoveWait_1.Active , bMoveLinearInterpolateRelativeExecute );
RST( NOT bExecute , bMoveRelativeExecute );
RST( NOT bExecute , bMoveLinearInterpolateRelativeExecute );
//-----轴控制开始等待-----
MCv_MoveWait_1(
    Axis:= Axis0001.AxisRef ,
    Execute:= bExecute ,
    ExecutionPermission:= SignalData0001 ,
    Options:= H00000000
);
//-----相对值定位-----
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
);
//-----轴组控制开始等待-----
MCv_GroupMoveWait_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bExecute ,
    ExecutionPermission:= SignalData0001 ,
    Options:= H00000000
);
//-----相对值直线插补控制用数据的设置-----
wMoveLinearInterpolateRelativeLinearAxes[0] := 1;
wMoveLinearInterpolateRelativeLinearAxes[1] := 2;
leMoveLinearInterpolateRelativeDistance[0] := 10000.0;
leMoveLinearInterpolateRelativeDistance[1] := 10000.0;
//-----相对值直线插补控制-----
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
);
```

46.23 高级同步控制

MCv_AdvancedSync

按照指定的高级同步控制设置开始同步控制。



使用时，应确认控制器及工程工具的版本。

所使用的控制器的用户手册

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Master	主轴	AXIS_REF	启动时	—	不能省略	设置输入轴。 省略了的情况下，将变为超出主轴编号范围(错误代码：1B0BH、1B85H)。 没有对应的输入设置的情况下，启动时将变为高级同步控制轴未设置(错误代码：1B22H)。
Slave	从轴	AXIS_REF	启动时	—	不能省略	设置输出轴。 省略了的情况下，将变为超出从轴编号范围(错误代码：1B0CH、1B86H)。 没有对应的输出设置的情况下，启动时将变为高级同步控制轴未设置(错误代码：1B22H)。

■输入变量


输入变量	名称	数据类型	获取	有效范围	初始值	内容
Enable	有效	BOOL	始终	TRUE、FALSE	FALSE	设置为TRUE时，执行本FB。
Acceleration	加速度	LREAL	启动时	0.0	0.0	应设置“0.0”。 设置了“0.0”以外的情况下，将变为超出加速度范围(错误代码：1A8DH)。
Deceleration	减速度	LREAL	启动时	0.0000、0.0001~ 2147483647.0	0.0	设置减速度。 关于详细内容，请参阅下述章节。 1821页 减速度(Deceleration)
Jerk	Jerk	LREAL	启动时	0.0	0.0	应设置“0.0”。 设置了“0.0”以外的情况下，将变为超出Jerk范围(错误代码：1A8FH)。
Options	选项	DWORD(HEX)	启动时	请参阅下述章节。 1821页 选项(Options)	00000000H	以位指定设置本FB的功能选项。 关于详细内容，请参阅下述章节。 1821页 选项(Options)

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	下述情况下，将变为TRUE。 <ul style="list-style-type: none"> 根据本FB下一个执行的运动控制FB中设置的多重启动(缓冲模式)的“1: Buffered”转移到下一个运动控制FB后。 同步控制解析模式下解析完成时。
InSync	同步中	BOOL	FALSE	输出轴开始了同步时，将变为TRUE。
CycleZeroPoint	凸轮循环完成	BOOL	FALSE	控制中(Active)变为TRUE后，使用了直线凸轮或凸轮开始点为“0”的凸轮的情况下，1周期当前位置每次过峰时将变为TRUE。 使用了凸轮开始点为“0”以外的凸轮的情况下，1周期当前位置通过凸轮数据的第0点的位置时将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了本FB时，将变为TRUE。
Active	控制中	BOOL	FALSE	本FB正在控制轴时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	本FB的执行中断时，将变为TRUE。 有效(Enable)变为FALSE时，将变为FALSE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。  所使用的控制器的用户手册

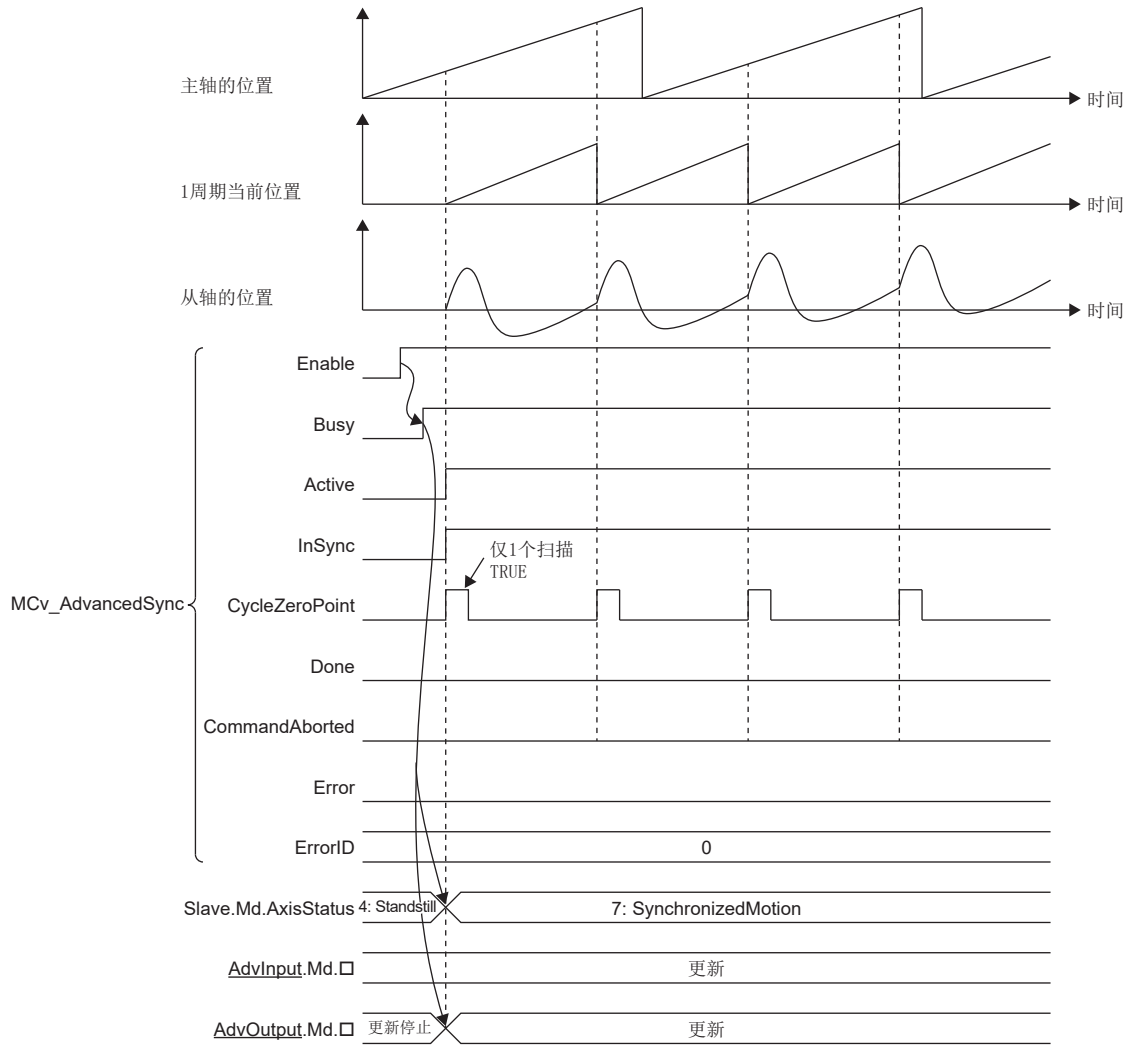
功能

- 执行本FB后，按照指定的高级同步控制设置开始高级同步控制。
- 仅在从轴的轴状态 (AxisName.Md.AxisStatus)为“4: 待机中(Standstill)”的情况下，可以开始同步控制。开始后从轴的轴状态 (AxisName.Md.AxisStatus)变为“7: 同步运行中(SynchronizedMotion)”。同步控制开始时，“4: 待机中(Standstill)”以外的情况下，从轴的轴状态 (AxisName.Md.AxisStatus)将变为“1: 错误停止中(ErrorStop)”、不可启用(错误代码: 1AFCH)。
- 将本FB的有效(Enable)置为了FALSE的情况下或输出轴的轴信息中设置的轴发生了停止原因的情况下，按照停止原因和该停止处理从轴(Slave)变为立即停止/减速停止。
- 关于高级同步控制的详细内容，请参阅下述手册。

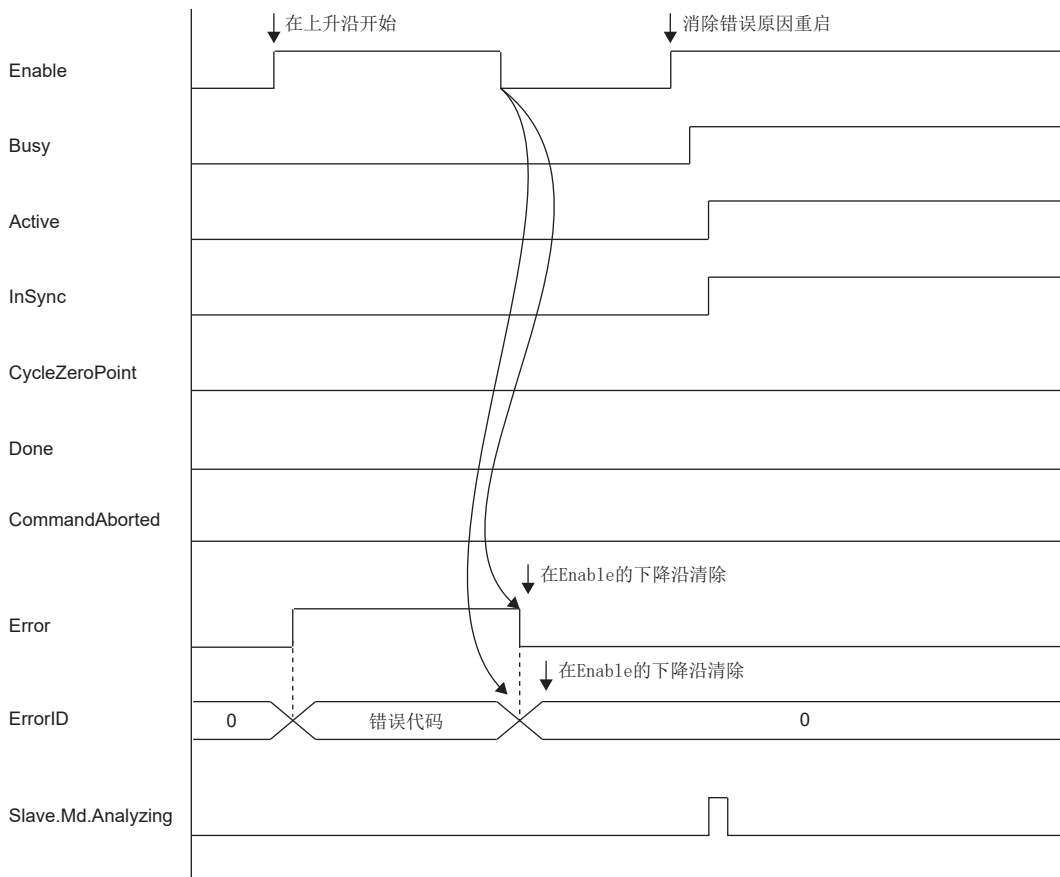
所使用的控制器的用户手册

■时序图

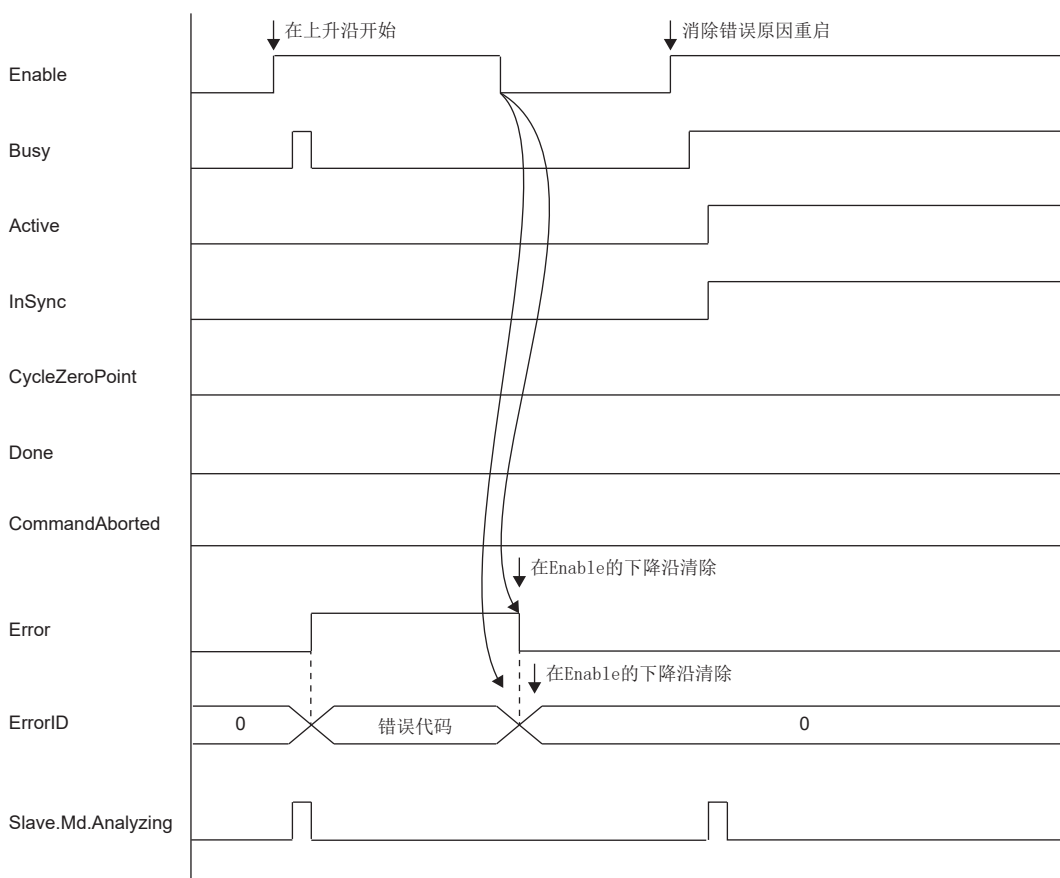
- 正常完成的情况下



• 输入输出变量异常的情况下



• 输入变量异常的情况下



■减速度(Deceleration)

设置减速度。

设置了加减速方式设置(选项(Options):位0~1)的“0:加减速速度指定方式(mcAccDec)”的情况下,对设置范围进行设置。

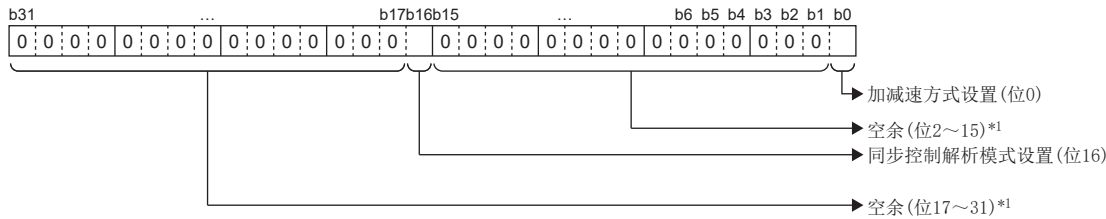
加减速方式设置(选项(Options):位0~1)	设置范围
0:加减速速度指定方式(mcAccDec)	0.0000*1、0.0001~2147483647.0[U/s ²]的正数

*1 启动时,根据启动时加减速速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior),动作将发生变化。加速度更改、减速度更改时,不受理更改。

■选项(Options)

以位指定设置本FB中使用的功能选项。

以位指定设置的内容如下所示。



*1 空余中应设置“0”。

位	名称	内容
0	加减速方式设置	设置用于进行控制的加减速方式。 • 0:加减速速度指定方式(mcAccDec)
16	同步控制解析模式设置	设置同步控制解析模式。 • 0:无效 • 1:有效

• 加减速方式设置(位0~1)

设定值	内容
0:加减速速度指定方式(mcAccDec)	是使用本FB中设置的减速度(Deceleration)进行加速/减速的方式。

注意事项

- 正在执行本FB时，即使主轴 (Master)、辅助轴 (AdvOutputName.Pr.AuxAxis)、子输入轴 (AdvOutputName.Pr.SubAxis) 的输入轴信息中设置的轴发生停止原因，从轴 (Slave) 也通过从输入轴传输的指令继续控制。从轴 (Slave) 的轴状态不变化。此外，即使输出轴的轴信息 (AdvOutputName.Axis) 中设置的轴发生停止原因，也不影响输入轴的轴信息 (AdvInputName.Axis) 中设置的轴。
- 启动本FB时处于下述的情况下，将变为主轴从轴循环参照 (错误代码：1AADH)。

错误的发生条件

连接多个轴时，将主轴 (Master)、辅助轴 (AdvOutputName.Pr.AuxAxis)、子输入轴 (AdvOutputName.Pr.SubAxis) 设置为后段的单轴同步控制FB的从轴 (Slave) 的情况下

连接了257个及以上的单轴同步控制FB的情况下

- 主轴 (Master)、辅助轴 (AdvOutputName.Pr.AuxAxis)、子输入轴 (AdvOutputName.Pr.SubAxis) 由于伺服报警或紧急停止而变为了伺服OFF的情况下，从轴 (Slave) 可能发生预想之外的动作。因此建议使用MC_Stop (强制停止) 停止本FB的动作。
- 本FB启动的解析中 (从轴 (Slave) 的轴状态 (AxisName.Md.AxisStatus) 在变为“7: 同步运行中 (SynchronizedMotion)”之前) 输入轴动作的情况下，同步控制开始之后反映解析中的输入轴的移动量及根据当前值更改等的变化量。根据输入轴的移动量及变化量，输出轴可能会急加速，因此应在确认变为同步控制中后开始输入轴的动作。
- 将同步控制中执行设定值获取的ADV_INPUT及ADV_OUTPUT的构件变量的设定值更改为了本FB执行中超出范围的值的的情况下，将变为超出参数范围 (高级同步控制输入轴) 或超出参数范围 (高级同步控制输出轴)，维持更改前的设定值继续控制。
- 本FB启动时ADV_INPUT及ADV_OUTPUT的构件变量的设定值超出范围的情况下，将变为超出参数范围 (高级同步控制输入轴) 或超出参数范围 (高级同步控制输出轴)，不启动。
- 启动本FB时处于下述的情况下，将变为运算周期不一致 (错误代码：1B29H)。使用了MCv_SyncOperationCycles (控制运算周期同步) 及符合下述条件的情况下，将变为运算周期不一致 (错误代码：1B29H)。


错误的发生条件

主轴 (Master)、辅助轴 (AdvOutputName.Pr.AuxAxis)、子输入轴 (AdvOutputName.Pr.SubAxis) 与从轴 (Slave) 的控制运算周期不同的情况下

输出轴的控制运算周期与离合器信号的刷新周期不同的情况下

程序示例

关于本FB的程序示例，请参阅下述手册。

 所使用的控制器的用户手册

46.24 多轴定位数据运行

MCv_MovePositioningData

按照指定的定位数据，对轴组执行动作。

限制事项

使用时，应确认控制器及工程工具的版本。

所使用的控制器的用户手册

梯形图	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>

设置数据

输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
AxesGroup	轴组信息	AXES_GROUP_REF	启动时	—	不能省略	设置轴组。
PositioningData	定位数据	MC_POSITIONING_DATA_ID	启动时	—	不能省略	设置定位数据的运算配置文件ID。 在MC_POSITIONING_DATA_ID结构体中设置运算配置文件ID。关于MC_POSITIONING_DATA_ID结构体的详细内容，请参阅下述章节。 ☞ 1426页 MC_POSITIONING_DATA_ID


输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行本FB。
StartDataNo	开始定位数据号	WORD (UINT)	启动时	0~5000	0	设置开始的定位数据号。 设置“0”后执行了本FB的情况下，从定位数据No. 1开始。
BufferMode	缓冲模式	INT (MC_BUFFER_MODE)	启动时	0、1	0	设置缓冲模式。 • 0: Aborting (mcAborting) • 1: Buffered (mcBuffered) 关于详细内容，请参阅下述章节。 ☞ 1826页 缓冲模式 (BufferMode)
Options	选项	DWORD (HEX)	启动时	请参阅下述章节。 ☞ 1826页 选项 (Options)	0000000H	以位指定设置本FB的功能选项。 关于详细内容，请参阅下述章节。 ☞ 1826页 选项 (Options)

■输出变量

输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	到达了目标位置时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了本FB时，将变为TRUE。
Active	控制中	BOOL	FALSE	本FB正在控制轴时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	本FB的执行中断时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。  所使用的控制器的用户手册
CurrentDataNo	执行中定位数据号	WORD(UINT)	0	表示执行中的定位数据号。 定位数据号切换时更新。
PrereadingDataNo	预读定位数据号	WORD(UINT)	0	存储预读的定位数据号的终端。 预读从执行中定位数据号(CurrentDataNo)到预读定位数据号(PrereadingDataNo)为止的定位数据号。
RemainingLoopCount	剩余循环次数	WORD(UINT)	0	执行控制方式为LOOP的定位数据时，存储剩余的重复次数。 在重复循环的起始将存储值减少“1”。 存储值为“0”的情况下，控制方式为LEND的定位数据时不循环。

■公开变量

公开变量	名称	数据类型	初始值	内容
InstanceID	实例ID	INSTANCE_ID	0	是实例ID。 实例创建时由系统自动设置。
ResetMcode	M代码复位	BOOL	FALSE	TRUE时执行M代码的复位。 通过复位，M选通(Mstrobe)将变为FALSE。
Mstrobe	M选通	BOOL	FALSE	M代码输出为有效中时，将变为TRUE。 M代码输出时序(选项(Options): 位16)时，将变为TRUE。 根据M代码复位(ResetMcode)的TRUE，将变为FALSE。
ValidMcode	有效M代码	WORD(UINT)	0	存储有效的(最新输出的)M代码。 定位数据的M代码输出有效的情况下，M代码输出时序(选项(Options): 位16)时，更新为定位数据号的M代码。 *: M代码复位(ResetMcode)的TRUE时不复位。
StepMode	步模式	INT (MC_STEP_MODE)	0	设置步模式。 • 0: 忽略(Ignored) • 1: 减速单位步(StepPerDec) • 2: 定位数据单位步(StepPerPositioningData) 关于详细内容，请参阅下述章节。  1826页 步模式(StepMode)
CurrentProfileNo	执行中配置文件No.	WORD(UINT)	0	存储执行中的配置文件ID编号。 关于详细内容，请参阅下述章节。  1827页 执行中配置文件No. (CurrentProfileNo)

- 进行公开变量的获取/更新的周期如下所示。

公开变量	获取/更新周期
InstanceID	调用FB的POU(程序部件)的执行周期
ResetMcode	轴组的配置轴的控制周期*1
Mstrobe	
ValidMcode	
StepMode	调用FB的POU(程序部件)的执行周期
CurrentProfileNo	

- *1 输出变量与更新时机可能不匹配。要与更新时机匹配的情况下，应从相同周期的程序调用FB。

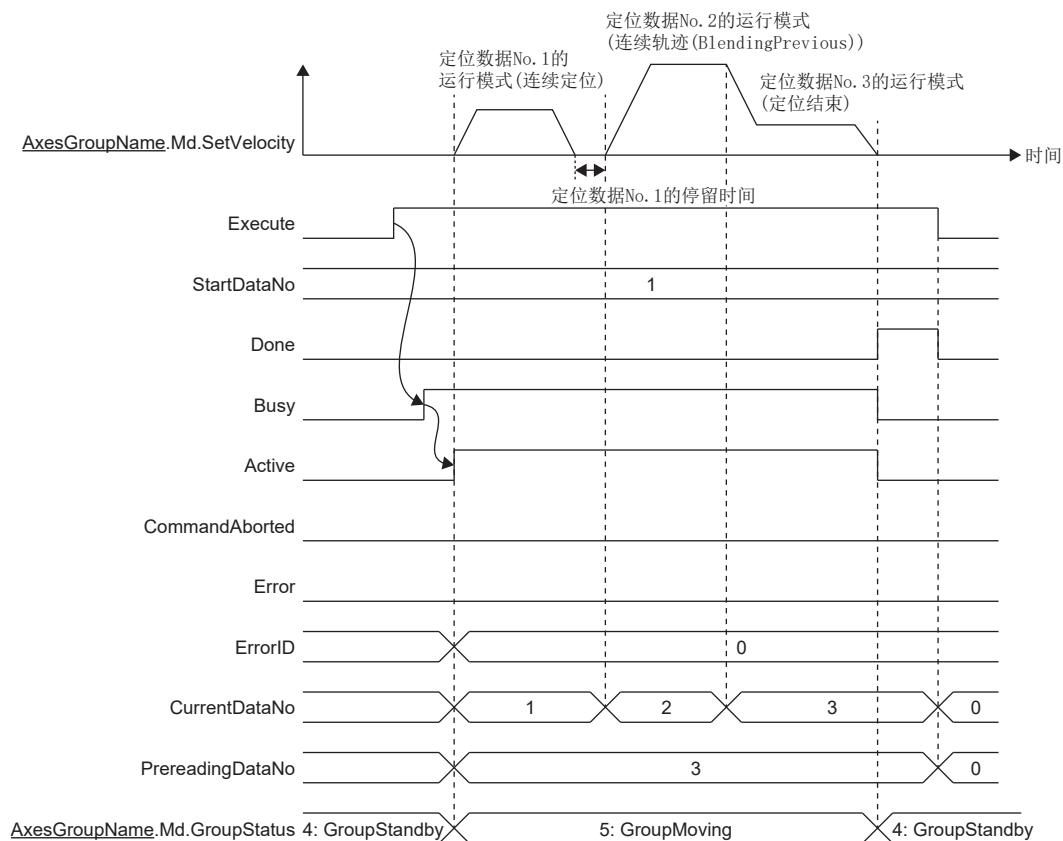
功能

- 对轴组执行运算配置文件(多轴定位数据格式)中指定的连续定位动作。
- 多轴定位数据运行按升序从开始定位数据号(StartDataNo)读入定位数据，并在内部连续执行多轴定位动作。
- 关于多轴定位数据运行的详细内容，请参阅下述手册。

📖 所使用的控制器的用户手册

■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

📖 1378页 通过有效(Enable)类型的运动控制FB的基本动作

■缓冲模式(BufferMode)

设置用于进行多重启动(缓冲模式)的动作。

本FB中可设置的缓冲模式如下所示。

设定值	内容
0: Aborting(mcAborting)	中断(取消)控制中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在控制中的FB中缓冲下一个FB。 控制中的FB已经被缓冲的情况下, 对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB完成时, 依次执行缓冲FB。

要点

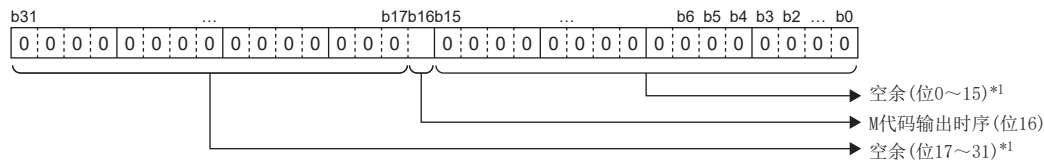
关于多重启动(缓冲模式)的详细内容, 请参阅下述手册。

所使用的控制器的用户手册

■选项(Options)

以位指定设置本FB中使用的功能选项。

以位指定设置的内容如下所示。



*1 空余中应设置0。设置了0以外的情况下, 将变为超出Options范围(错误代码: 1ABBH)。

位	名称	内容
16	M代码输出时序	设置输出M代码的时序。 • 0: WITH模式 • 1: AFTER模式

• M代码输出时序(位16)

设定值	内容
0: WITH模式*2	定位数据的定位开始时将M选通(Mstrobe)置为TRUE, 并将定位数据的M代码(Mcode)存储为有效M代码(ValidMcode)。
1: AFTER模式*2	目标位置到达时(停留时间设置时经过设置时间后)将M选通(Mstrobe)置为TRUE, 并将定位数据的M代码存储为M代码(Mcode)。

*2 关于各模式的详细内容, 请参阅下述手册。

所使用的控制器的用户手册

■步模式(StepMode)

设置步模式。

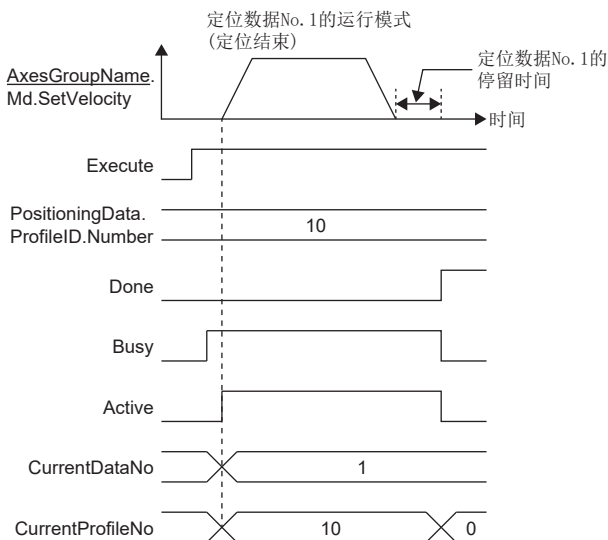
设定值	内容
0: 忽略(Ignored)	将步功能置为无效。 忽略步模式。
1: 减速单位步(StepPerDec)	将步功能置为有效。 以减速单位执行步运行。
2: 定位数据单位步(StepPerPositioningData)	将步功能置为有效。 以定位数据单位执行步运行。

■执行中配置文件No. (CurrentProfileNo)

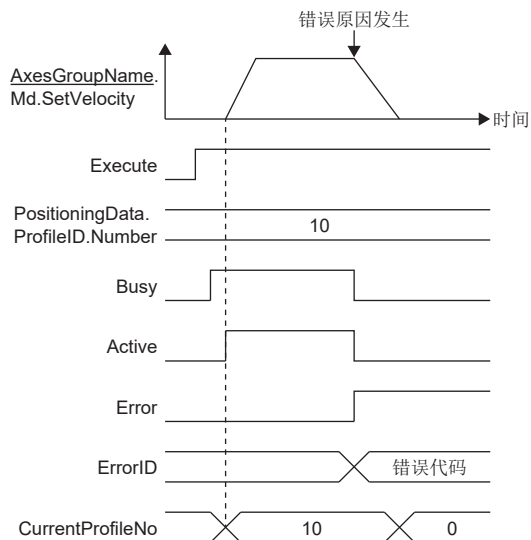
存储执行中的配置文件ID编号。

- 表示正在使用配置文件ID编号相关的展开区域的运算配置文件。
- 动作开始时存储配置文件ID编号，动作完成时清零。

■正常时



■动作开始后发生了错误的情况下



- 执行中配置文件No. (CurrentProfileNo)的存储值不存储在轴组配置轴的执行配置文件ID编号 (AxisName.Md.ProfileID)中。

■必要对象数据

使用FB的情况下，应将指定的轴组的所有配置轴的下述对象数据中分配的对象映射到PDO。

- Target position (607AH)

有未将对象映射到PDO的配置轴的情况下，将变为必要对象数据未设置(错误代码：1AF7H)且不启动。

关于与对象数据设置有关的详细内容，请参阅下述手册。

📖所使用的控制器的用户手册

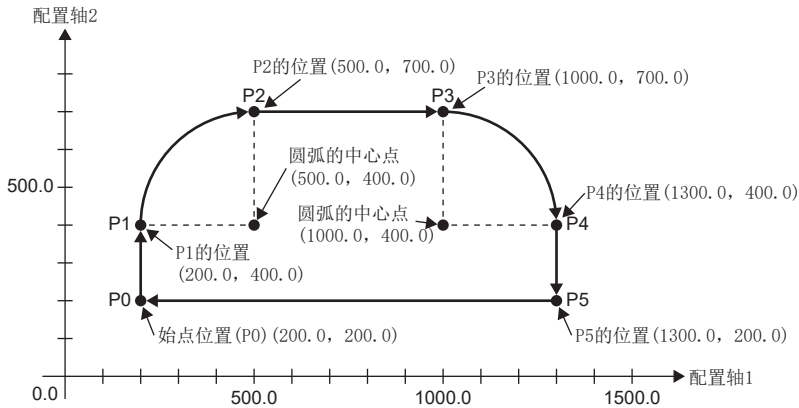
注意事项

- 本FB根据预读或外部信号设置等运算配置文件解析，启动或多重启动(Aborting)的动作可能会延迟。

程序示例

进行轴组1 (AxesGroup001) 的定位控制的示例如下所示。

• 动作



定位数据号	轨迹	定位动作
1	P0→P1	从始点位置(P0)到位置(P1)通过绝对值直线插补(LinearAbsolute)执行定位
2	P1→P2	从位置(P1)到位置(P2)通过绝对值圆弧插补(CircularAbsolute)执行定位
3	P2→P3	从位置(P2)到位置(P3)通过绝对值直线插补(LinearAbsolute)执行定位
4	P3→P4	从位置(P3)到位置(P4)通过相对值圆弧插补(CircularRelative)执行定位
5	P4→P5	从位置(P4)到位置(P5)通过相对值直线插补(LinearRelative)执行定位
6	P5→P0	从位置(P5)到终点位置(P0)通过绝对值直线插补(LinearAbsolute)执行定位

• 轴组

项目	设定值
配置轴[1]	Axis0001
配置轴[2]	Axis0002

■轴

轴No.	标签名	数据类型	注释
1	Axis0001*1	AXIS_REF	轴1
2	Axis0002*1	AXIS_REF	轴2

*1 为了进行圆弧插补控制, 将轴参数的软件行程限位对象(AxisName.Pr.SwStrokeLimit_Target)设置为“有效(-1: 无效(Invalid)”以外”。(程序示例中设置为了“1: 指令当前位置(SetPosition)”。)

■轴组

轴组No.	标签名	数据类型	注释
1	AxesGroup001	AXES_GROUP_REF	轴组1

■运算配置文件

配置文件ID	标签名	数据类型	注释
1	PositioningData0001	MC_POSITIONING_DATA_REF	多轴定位数据1

• 运算配置文件(PositioningData0001(多轴定位数据))的设置

项目	定位数据号					
	1	2	3	4	5	6
运行类型	连续轨迹 (BlendingPrevious)	连续轨迹 (BlendingPrevious)	连续轨迹 (BlendingPrevious)	连续轨迹 (BlendingPrevious)	连续定位 (ContinuousPositioning)	定位结束 (PositioningComplete)
控制方式	绝对值直线插补	绝对值圆弧插补	绝对值直线插补	相对值圆弧插补	相对值直线插补	绝对值直线插补
插补轴	插补轴1	Axis0001	Axis0001	Axis0001	Axis0001	Axis0001
	插补轴2	Axis0002	Axis0002	Axis0002	Axis0002	Axis0002
圆弧插补模式	—	中心点指定	—	中心点指定	—	—
路径选择	—	CW	—	CW	—	—

项目		定位数据号					
		1	2	3	4	5	6
目标位置/移动量/终点	插补轴1	200.0	500.0	1000.0	300.0	0.0	200.0
	插补轴2	400.0	700.0	700.0	-300.0	-200.0	200.0
方向选择	插补轴1	正方向	—	正方向	—	—	正方向
	插补轴2	正方向	—	正方向	—	—	正方向
边界点/中心点/半径	插补轴1	—	500.0	—	0.0	—	—
	插补轴2	—	400.0	—	-300.0	—	—
速度模式		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
加减速方式		加减速速度指定方式	—	—	—	—	—
加速度/减速度时间		400.0	0.0	0.0	0.0	0.0	0.0
减速度		500.0	0.0	0.0	0.0	0.0	0.0
Jerk		500.0	0.0	0.0	0.0	0.0	0.0
圆弧插补误差允许差值		—	—	—	—	—	—
超出环形计数器的目标位置指定		不允许	—	不允许	—	—	不允许
停留时间		500.0	0.0	0.0	0.0	0.3	0.0
M代码		0	0	0	0	0	0
M代码输出时序超驰		—	—	—	—	—	—

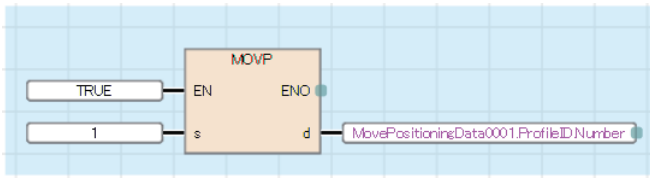
■使用的标签

标签名	数据类型	注释
MCv_MovePositioningData_1	MCv_MovePositioningData	多轴定位数据运行FB
bMovePosDataDone	位	执行完成
bMovePosDataBusy	位	执行中
bMovePosDataActive	位	控制中
bMovePosDataCommandAborted	位	执行中断
bMovePosDataError	位	错误
uwMovePosDataErrorID	字[无符号]/位串[16位]	错误代码
uwMovePosDataCurrentDataNo	字[无符号]/位串[16位]	执行中定位数据号
uwMovePosDataPrereadingDataNo	字[无符号]/位串[16位]	预定定位数据号
uwMovePosDataRemainingLoopCount	字[无符号]/位串[16位]	剩余循环次数
MC_GroupEnable_1	MC_GroupEnable	轴组有效FB
bMovePositioningDataCMD	位	多轴定位数据运行启动
bGroupEnableDone	位	轴组有效完成
bGroupEnableBusy	位	执行中
bGroupEnableError	位	错误
uwGroupEnableErrorID	字[无符号]/位串[16位]	错误代码
MC_GroupDisable_1	MC_GroupDisable	轴组无效FB
bGroupDisableDone	位	轴组无效完成
bGroupDisableBusy	位	执行中
bGroupDisableError	位	错误
uwGroupDisableErrorID	字[无符号]/位串[16位]	错误代码
MovePositioningData0001*1	MC_POSITIONING_DATA_ID	多轴定位数据0001

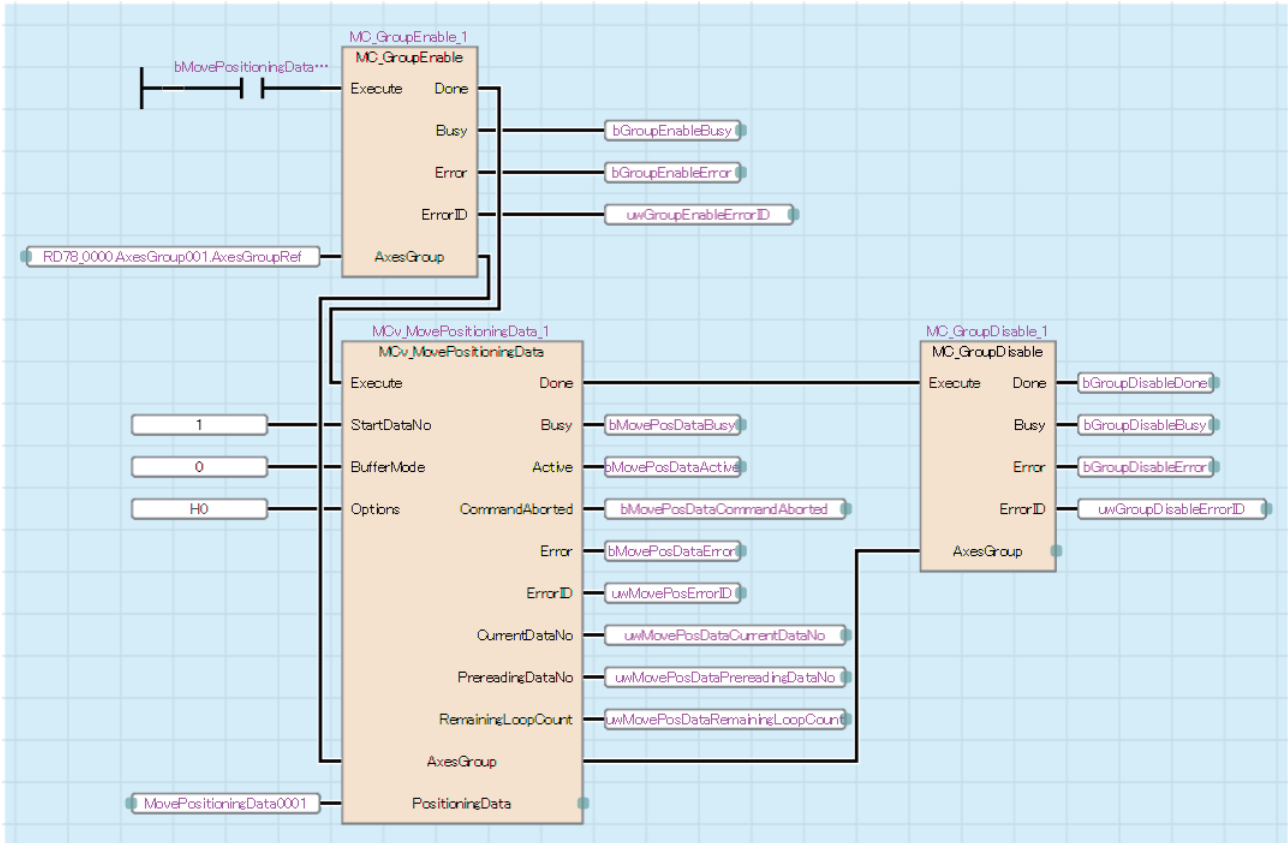
*1 仅FBD/LD的程序使用(为了在FBD/LD的程序中使用运算配置文件需要此项)

■FBD/LD的程序

- 2轴直线插补控制用数据的设置



- 轴组有效/多轴定位数据运行/轴组无效



■ST的程序

```
//-----轴组有效-----
MC_GroupEnable_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bMovePositioningDataCMD ,
    Done=> bGroupEnableDone,
    Busy=> bGroupEnableBusy ,
    Error=> bGroupEnableError ,
    ErrorID=> uwGroupEnableErrorID
);

//-----多轴定位数据运行-----
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
);

//-----轴组无效-----
MC_GroupDisable_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bMovePosDataDone ,
    Done=> bGroupDisableDone ,
    Busy=> bGroupDisableBusy ,
    Error=> bGroupDisableError ,
    ErrorID=> uwGroupDisableErrorID
);
```

46.25 绝对位置跟踪控制

MCv_PositionControl

按照限制速度、加速度、减速度、Jerk将指定的轴移动到目标位置的功能。此外，即使指令到达目标位置，也可通过重启、连续更新来更改目标位置及移动轴。

限制事项

使用时，应确认控制器及工程工具的版本。

📖所使用的控制器的用户手册

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量 (AxisName, AxisRef.)，请参阅下述内容。 📖 1387页 AxisName, AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行本FB。
ContinuousUpdate	连续更新	BOOL	启动时	TRUE、FALSE	FALSE	TRUE期间可以连续更改目标位置、速度、加速度、减速度。
Position	目标位置	LREAL	启动时/可重启/ 可连续更新	<ul style="list-style-type: none"> ■环形计数器无效时 -10000000000.0 ≤ 目标位置 < 10000000000.0 ■环形计数器有效时 环形计数器下限值 ≤ 目标位置 < 环形计数器上限值 	0.0	设置目标位置。 关于详细内容，请参阅下述章节。 📖 1835页 目标位置(Position)
Velocity	速度	LREAL	启动时/可重启/ 可连续更新	0.0、0.0001~ 2500000000.0	0.0	设置速度。 关于详细内容，请参阅下述章节。 📖 1837页 速度(Velocity)
Acceleration	加速度	LREAL	启动时/可重启/ 可连续更新	0.0000、0.0001~ 2147483647.0	0.0	设置加速度。 关于详细内容，请参阅下述章节。 📖 1837页 加速度(Acceleration)
Deceleration	减速度	LREAL	启动时/可重启/ 可连续更新	0.0000、0.0001~ 2147483647.0	0.0	设置减速度。 关于详细内容，请参阅下述章节。 📖 1837页 减速度(Deceleration)

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Jerk	Jerk	LREAL	启动时	0.0000、0.0001~2147483647.0	0.0	设置Jerk。 关于详细内容，请参阅下述章节。 ☞ 1837页 Jerk
Direction	方向选择	INT (MC_DIRECTION [0..15])	启动时/可重启/ 可连续更新	1、2、3	3	设置方向选择。 关于详细内容，请参阅下述章节。 ☞ 1838页 方向选择(Direction)
BufferMode	缓冲模式选择	INT(MC_BUFFER _MODE)	启动时	0、1、2、3、4、5	0	设置缓冲模式。 关于详细内容，请参阅下述章节。 ☞ 1841页 缓冲模式(BufferMode)
Options	选项	DWORD(HEX)	启动时	00000000H~00070021H	00000000H	设置本FB的功能选项。 关于详细内容，请参阅下述章节。 ☞ 1841页 选项(Options)

■输出变量

输出变量	名称	数据类型	初始值	内容
InAddress	目标位置到达	BOOL	FALSE	运动系统中计算的指令位置 (<u>AxisName</u> .Md.SetPosition)到达了目标位置时，将变为TRUE。 连续更新(ContinuousUpdate)为有效(TRUE)时更改了目标位置的情况下，在到达更改后的目标位置之前将变为FALSE。
Busy	执行中	BOOL	FALSE	本FB执行中时，将变为TRUE。
Active	控制中	BOOL	FALSE	本FB正在控制轴时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	通过其他FB中断执行时，将变为TRUE。 发生异常后本FB中止时或异常发生中启动本FB时，将变为TRUE。 执行指令(Execute)变为FALSE时，将变为FALSE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD(UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 ☞ 所使用的控制器的用户手册

功能

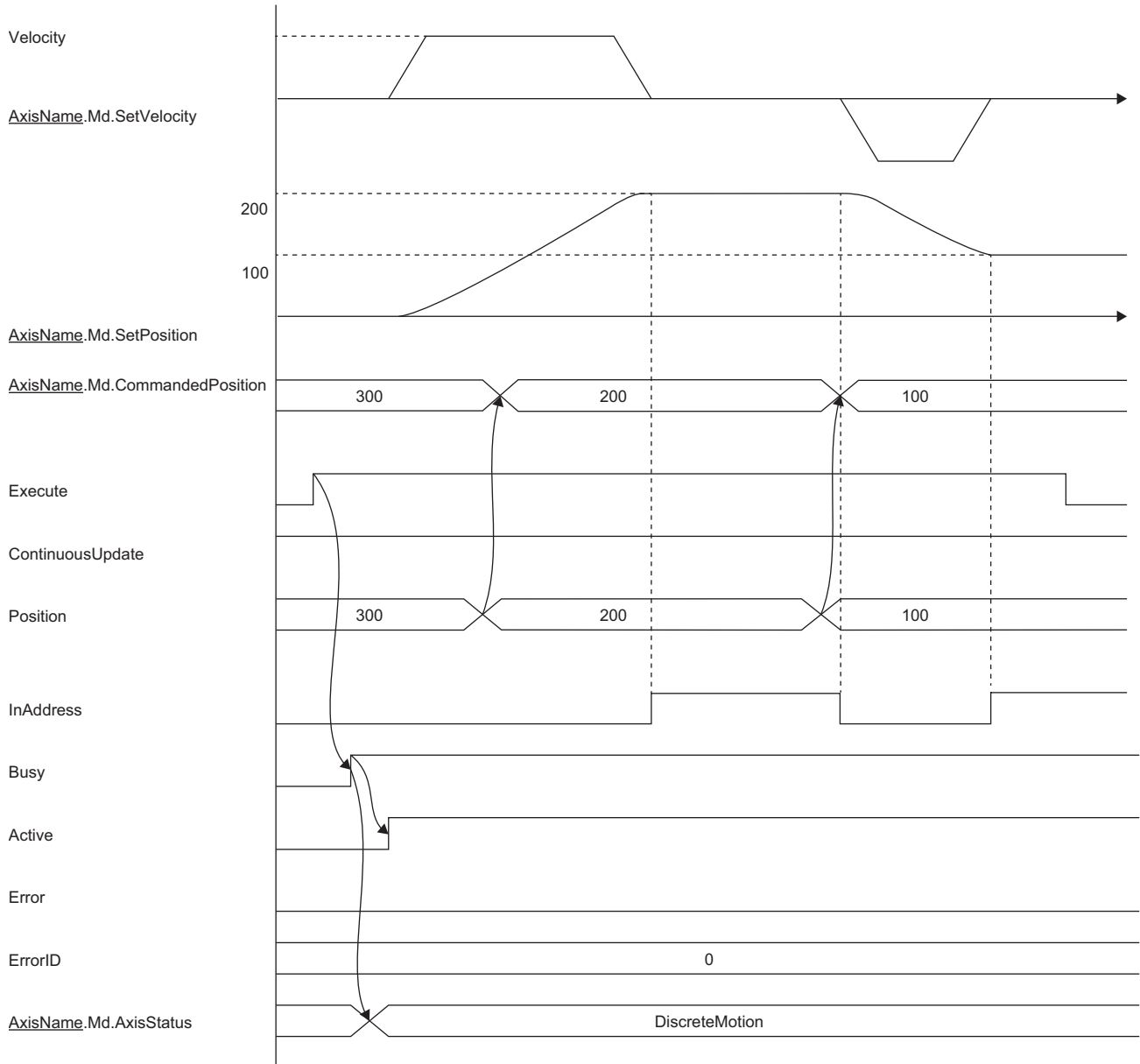
按照限制速度、加速度(Acceleration)、减速度(Deceleration)、Jerk将指定的轴(Axis)移动到目标位置(Position)。此外，即使指令到达目标位置，也可通过重启、连续更新(ContinuousUpdate)更改目标位置(Position)及移动轴。

到达目标位置(Position)前更改了目标位置(Position)的情况下，移动到更改后的目标位置。

本FB通过执行MC_Stop(强制停止)或其他运行系统FB，将变为执行完成。

■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■目标位置(Position)

根据轴的单位设置绝对位置的目标位置。设定值为负值的情况下，向负方向移动。设置了0的情况下，轴不动作但对对象轴的轴状态(AxisName.Md.AxisStatus)将变为“5: 定位运行中(DiscreteMotion)”。

有效范围如下所示。设置了超出范围的值的情况下，将变为超出目标位置范围(错误代码: 1A85H)且不启动。

环形计数器的设置	有效范围
无效	$-10000000000.0 \leq \text{目标位置} < 10000000000.0$
有效	环形计数器下限值 \leq 目标位置 $<$ 环形计数器上限值

软件行程限位有效时(软件行程限位对象(AxisName.Md.SwStrokeLimit_Target)为“-1: 无效(Invalid)”以外时)

- 将软件行程限位溢出的位置作为目标位置设置的情况下，将变为软件行程限位溢出(目标位置)(错误代码: 1A80H)且不启动。
- 忽略方向选择(Direction)的设置，以不超出软件行程限位范围的方向进行定位控制。
- 正方向及负方向不超出软件行程限位范围的情况下，以当前位置为基准，向趋近目标位置(Position)的方向(移动量的绝对值较短的一方)进行定位控制。
- 正方向及负方向距离相同的情况下，沿用前一个执行的运动控制FB的移动方向或上次执行的目标位置更改时的移动方向执行动作。没有上次的移动方向的情况下，向正方向执行动作。

软件行程限位无效时(软件行程限位对象(AxisName.Md.SwStrokeLimit_Target)为“-1: 无效(Invalid)”时)，在有效范围内进行定位控制。

目标位置到达(InAddress)变为TRUE的1运算周期前，忽略目标位置更改。

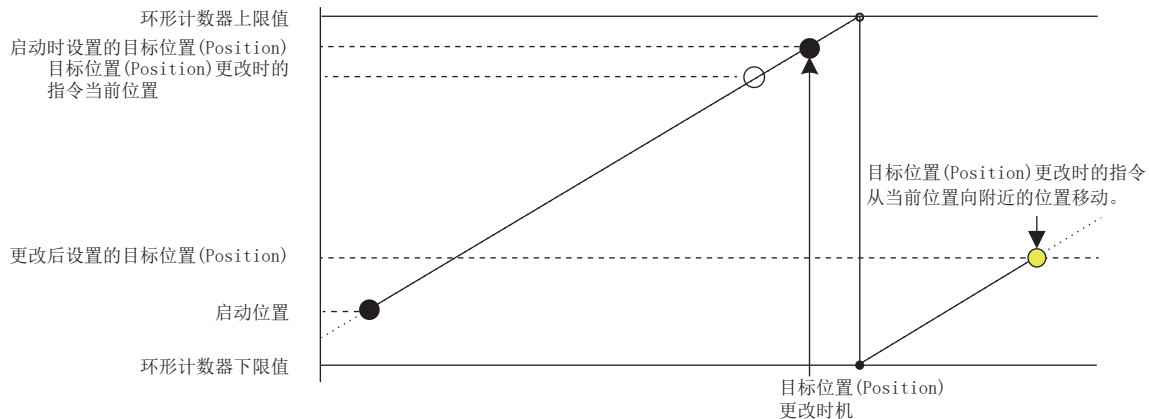
更改了目标位置(Position)的情况下，根据方向选择(Direction)的设置，更改时的动作有所不同。

- 运动控制FB执行时目标位置(Position)中设置了超出范围的值的情况下，将发生超出Position/Distance设置范围警告(事件代码: 00D0DH)，按照更改前的设置继续动作。
- 发生超出Position/Distance设置范围警告(事件代码: 00D0DH)时，更改了方向选择(Direction)的情况下，按照更改前的方向选择(Direction)的设置继续动作。
- 发生超出方向选择范围警告(事件代码: 00D4DH)时，按照更改前的目标位置(Position)的设置继续动作。

例

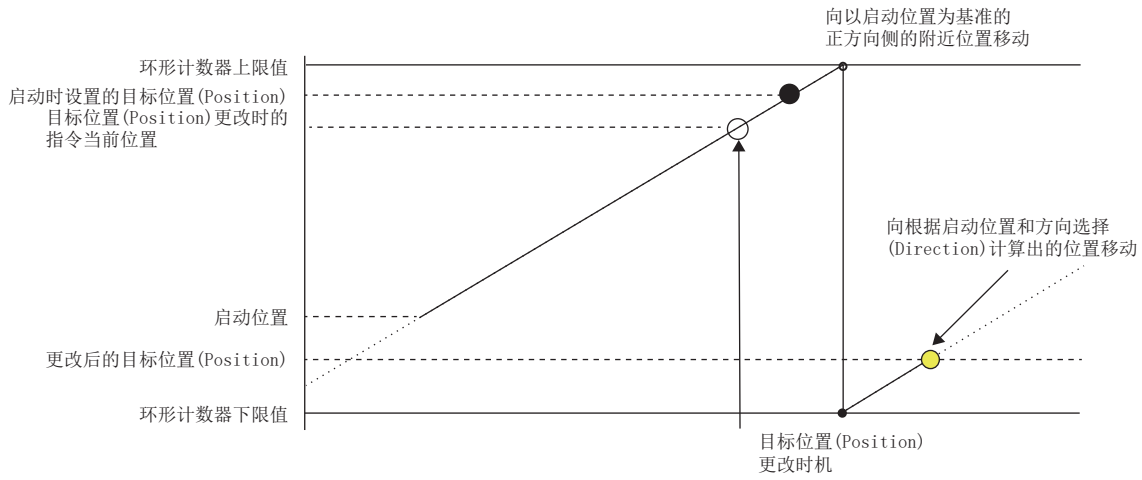
设置最短路径的情况下，从更改时的指令当前位置到更改后的目标位置(Position)向较近的方向执行动作。

下述的“启动”表示首次之后的运动控制FB的启动及上次动作时目标位置到达(InAddress)变为TRUE后的目标位置(Position)的更改。



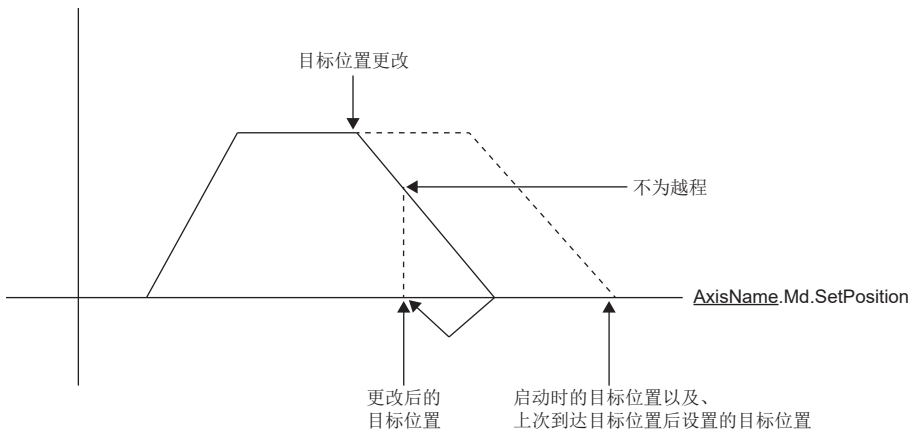
例

设置最短路径以外的情况下，根据更改时的指令当前位置和方向选择(Direction)计算目标位置(Position)。下述的“启动”表示首次之后的运动控制FB的启动及上次动作时目标位置到达(InAddress)变为TRUE后的目标位置(Position)的更改。



根据目标位置(Position)的更改变为反转动作的情况下，或更改时越过更改后的目标位置(Position)的情况下，进行减速停止。减速停止完成后，向更改的目标位置(Position)开始动作。在此情况下，不发生越程错误。

AxisName.Md.SetVelocity



■速度 (Velocity)

设置本FB中的指令速度。

设置范围

0.0、0.0001~2500000000.0

省略了设置或设置了0.0的情况下，轴不动作但对对象轴的轴状态 (AxisName. Md. AxisStatus) 将变为“6: 连续动作运行中 (ContinuousMotion)”。

设置了超出范围的值的情况下，将发生超出速度范围 (错误代码: 1ABAH)，不启动。

关于运动控制FB执行中的动作，请参阅所使用的控制器的用户手册。

■加速度 (Acceleration)

设置本FB中的加速度。关于加减速方式的详细内容，请参阅所使用的控制器的用户手册。

设置了超出范围的值的情况下，将发生超出加速度范围 (错误代码: 1A8DH)，不启动。

关于运动控制FB执行中的动作，请参阅所使用的控制器的用户手册。

■减速度 (Deceleration)

设置本FB中的减速度。关于加减速方式的详细内容，请参阅所使用的控制器的用户手册。

设置了超出范围的值的情况下，将发生超出减速度范围 (错误代码: 1A8EH)，不启动。

关于运动控制FB执行中的动作，请参阅所使用的控制器的用户手册。

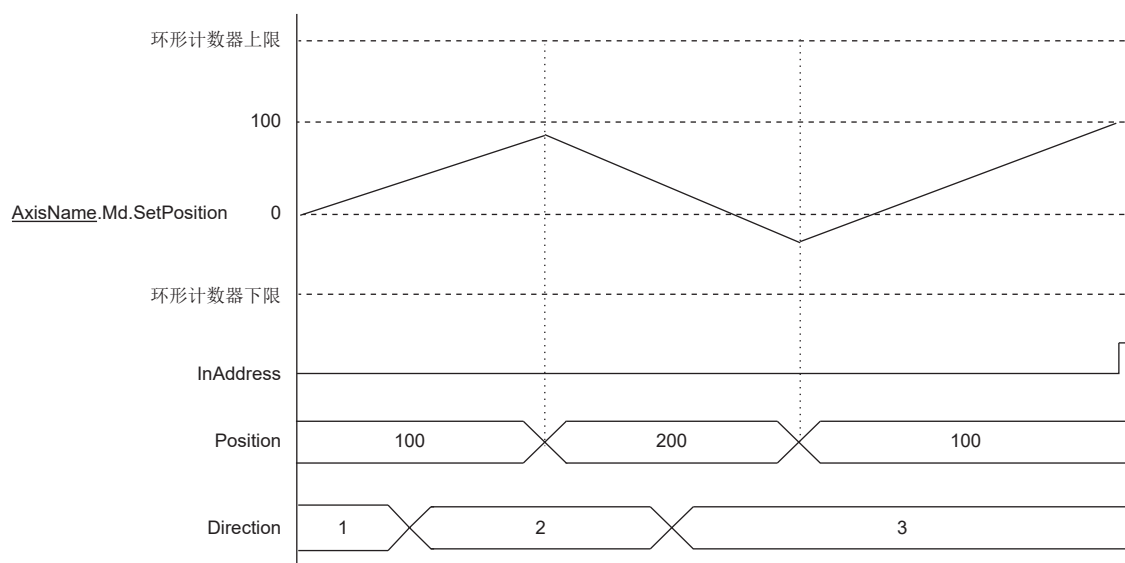
■Jerk

设置本FB中的Jerk。关于加减速方式的详细内容，请参阅所使用的控制器的用户手册。

设置了超出范围的值的情况下，将发生超出Jerk范围 (错误代码: 1A8FH)，不启动。

■方向选择 (Direction)

指定动作的方向。连续更新及重启的情况下，方向选择 (Direction) 不会在更改值的时机反映，而是在更改目标位置 (Position) 时反映。



可以如下进行选择。

- 1: 正方向 (mcPositiveDirection)
- 2: 负方向 (mcNegativeDirection)
- 3: 最短路径 (mcShortestWay)

启动时方向选择 (Direction) 设置了超出范围的值的的情况下，将发生超出方向选择范围 (错误代码: 1AA5H)，不启动。

控制时方向选择 (Direction) 设置了超出范围的值的的情况下，将发生超出方向选择范围警告 (事件代码: 00D4DH)，继续更改前的动作。此时如果同时更改目标位置 (Position) 和方向选择 (Direction)，则按照更改前的目标位置 (Position) 的设置继续动作。

软件行程限位有效时，忽略方向选择 (Direction) 的设置。

软件行程限位无效时，可以从方向选择 (Direction) 的“1: 正方向 (mcPositiveDirection)”、“2: 负方向 (mcNegativeDirection)”、“3: 最短路径 (mcShortestWay)”中选择，并设置从当前位置向目标位置 (Position) 移动的方向。

• 1: 正方向(mcPositiveDirection)

从当前位置向目标位置进行正方向(地址增加)的定位。

设置范围	动作
当前值 ≤ 目标位置 < 环形计数器上限值	<p>The graph shows a sawtooth wave representing a ring counter. The y-axis ranges from -100.0 to 100.0. The current value is at 0.0. The target position is at a positive value less than the upper limit. An arrow points from the current value to the target position in the positive direction.</p>
环形计数器下限值 ≤ 目标位置 < 当前值	<p>The graph shows a sawtooth wave. The current value is at 0.0. The target position is at a positive value greater than the upper limit. An arrow points from the current value to the target position in the positive direction, wrapping around the upper limit.</p>

• 2: 负方向(mcNegativeDirection)

从当前位置向负方向(地址减少)的目标位置进行定位。

设置范围	动作
环形计数器下限值 ≤ 目标位置 ≤ 当前值	<p>The graph shows a sawtooth wave. The current value is at 0.0. The target position is at a negative value greater than the lower limit. An arrow points from the current value to the target position in the negative direction.</p>
当前值 < 目标位置 < 环形计数器上限值	<p>The graph shows a sawtooth wave. The current value is at 0.0. The target position is at a positive value less than the upper limit. An arrow points from the current value to the target position in the negative direction, wrapping around the upper limit.</p>

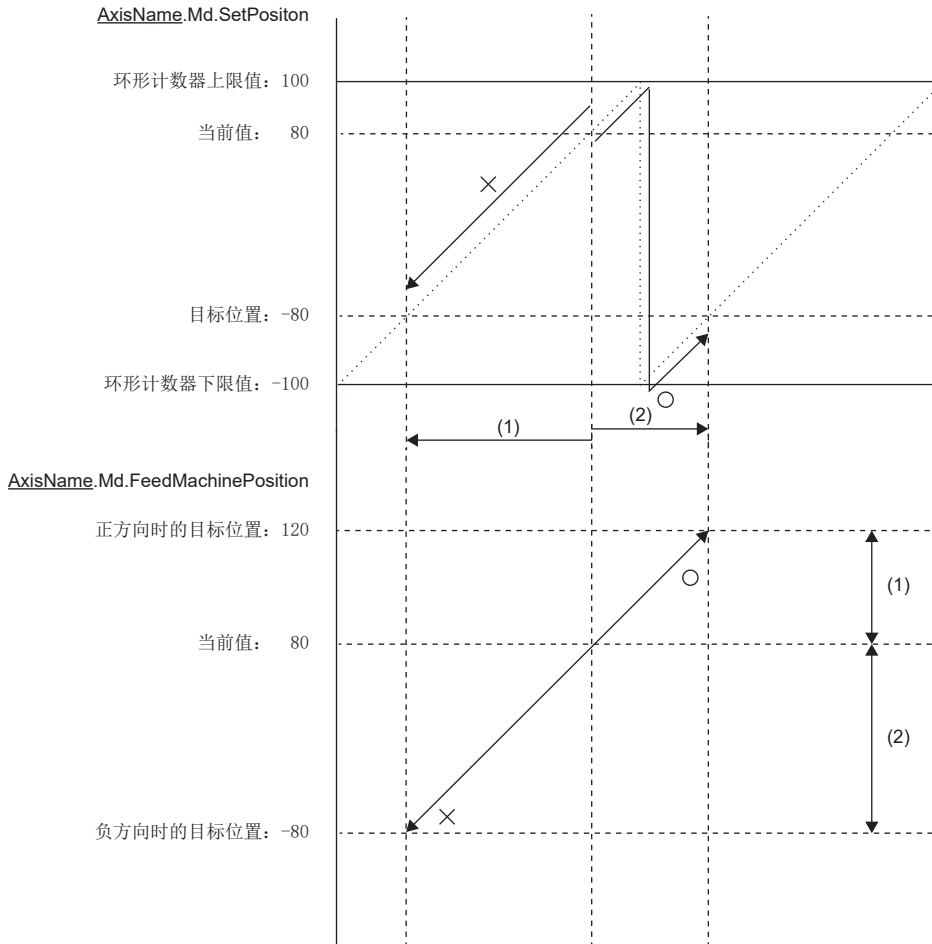
• 3: 最短路径 (mcShortestWay)

以当前位置为基准，以趋近目标位置的方向(移动量的绝对值较短的一方)进行定位控制。正方向、负方向距离相同的情况下以当前方向执行动作。目标位置的设置范围为“环形计数器下限值 ≤ 目标位置 < 环形计数器上限值”。设置了超出范围的值的情况下，将变为超出目标位置范围(错误代码：1A85H)且不启动。

例

设置了下述值时的动作

- 环形计数器上限值：100.0
- 环形计数器下限值：-100.0
- 当前位置：80.0
- 目标位置：-80.0



(1)的情况下移动量变为“160.0”，(2)的情况下移动量变为“40.0”，因此以(2)的正方向移动。

■缓冲模式(BufferMode)

设置用于进行多重启动(缓冲模式)的动作。


本FB中可设置的缓冲模式如下所示。

设定值	内容
0: Aborting(mcAborting)	中断(取消)控制中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在控制中的FB中缓冲下一个FB。 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB完成时,依次执行缓冲FB。
2: BlendingLow(mcBlendingLow)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将控制中的FB与缓冲FB的目标速度中较低一方的速度作为切换速度。
3: BlendingPrevious(mcBlendingPrevious)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将控制中的FB的目标速度作为切换速度。
4: BlendingNext(mcBlendingNext)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将缓冲FB的目标速度作为切换速度。
5: BlendingHigh(mcBlendingHigh)	在控制中的FB中缓冲下一个FB。 ^{*1} 控制中的FB已经被缓冲的情况下,对之前的FB的下一个进行缓冲。(最多2个) 控制中的FB到达目标位置后,依次执行缓冲FB。 控制中的FB到达了目标位置时,将控制中的FB与缓冲FB的目标速度中较高一方的速度作为切换速度。

*1 在控制中的FB与缓冲FB之间不执行停止。

要点

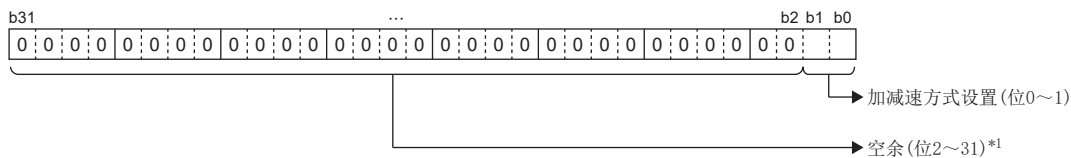
关于多重启动(缓冲模式)的详细内容,请参阅下述手册。

所使用的控制器的用户手册

■选项(Options)

以位指定设置本FB中使用的功能选项。

以位指定设置的内容如下所示。



*1 空余中应设置“0”。设置了“0”以外的情况下,将变为超出Options范围(错误代码:1ABBH)。

位	名称	内容
0~1	加减速方式设置	设置用于进行控制的加减速方式。 • 0: 加减速速度指定方式(mcAccDec) • 1: 加减速时间恒定方式(mcFixedTime)

• 加减速方式设置(位0~1)

设定值	内容
0: 加减速速度指定方式(mcAccDec)	是使用本FB中设置的加速度(Acceleration)、减速度(Deceleration)、Jerk进行加速/减速的方式。
1: 加减速时间恒定方式(mcFixedTime)	是与速度无关,使用本FB中设置的加减速时间进行加速/减速的方式。 对于加减速时间,设置为加速度(Acceleration),不使用减速度(Deceleration)、Jerk。

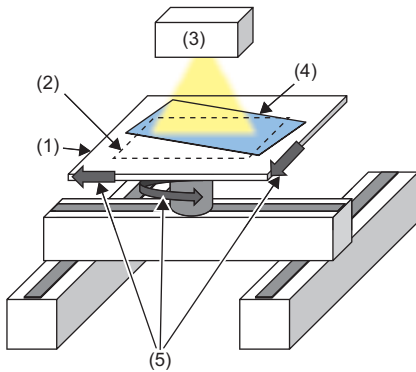
注意事项

同时更改了目标位置(Position)和方向选择(Direction)的情况下,有可能发生背离。通过先更改方向选择(Direction)然后更改目标位置(Position),更改目标位置(Position)时,两个更改都适用。必须同时更改目标位置(Position)和方向选择(Direction)的值的的情况下,应将程序执行周期设置为“恒定周期”后执行本FB,或重启本FB。

程序示例

使用本FB,并基于传感器的补偿信息(θ /X/Y轴补偿值),将XY θ 工作台上的工件移动至基准位置的程序示例如下所示。

- 动作



- (1): XY θ 工作台
- (2): 基准位置
- (3): 传感器
- (4): 工件
- (5): 基于传感器的补偿信息,通过位置跟踪补偿位置。

■轴

设置项目	X轴	Y轴	θ 轴
轴类型	实际驱动轴	实际驱动轴	实际驱动轴
位置指令单位	μm	μm	degree

■使用的标签

标签名	数据类型	注释
bMCv_PositionControl_Execute	位	控制开始
leMCv_PositionControl_Pos_1	双精度实数	至MCv_PositionControl_1.Position的输入
leMCv_PositionControl_Pos_2	双精度实数	至MCv_PositionControl_2.Position的输入
leMCv_PositionControl_Pos_3	双精度实数	至MCv_PositionControl_3.Position的输入
leCorrectionValue_Theta	双精度实数	θ 轴补偿值(输入从传感器获取的补偿值)
leCorrectionValue_XPos	双精度实数	X轴补偿值(输入从传感器获取的补偿值)
leCorrectionValue_YPos	双精度实数	Y轴补偿值(输入从传感器获取的补偿值)

■ST的程序

```
//输入来自传感器的补偿值
leMCv_PositionControl_Pos_1:= leCorrectionValue_Theta;
leMCv_PositionControl_Pos_2:= leCorrectionValue_XPos;
leMCv_PositionControl_Pos_3:= leCorrectionValue_YPos;
```

```
//-----轴1: 绝对位置跟踪控制-----
```

```
MCv_PositionControl_1 (
  Axis:= Axis0001.AxisRef ,
  Execute:= bMCv_PositionControl_Execute,
  ContinuousUpdate:= TRUE ,
  Position:= leMCv_PositionControl_Pos_1
  Velocity:= 100000.0 ,
  Acceleration:= 1000.0 ,
  Deceleration:= 1000.0 ,
  Jerk:= 0.0 ,
  Direction:= MC_DIRECTION_mcPositiveDirection ,
  BufferMode:= MC_BUFFER_MODE_mcAborting ,
  Options:= 0
);
```

```
//-----轴2: 绝对位置跟踪控制-----
```

```
MCv_PositionControl_2 (
  Axis:= Axis0002.AxisRef ,
  Execute:= bMCv_PositionControl_Execute,
  ContinuousUpdate:= TRUE ,
  Position:= leMCv_PositionControl_Pos_2
  Velocity:= 10000000.0 ,
  Acceleration:= 100000.0 ,
  Deceleration:= 100000.0 ,
  Jerk:= 0.0 ,
  Direction:= MC_DIRECTION_mcPositiveDirection ,
  BufferMode:= MC_BUFFER_MODE_mcAborting ,
  Options:= 0
);
```

```
//-----轴3: 绝对位置跟踪控制-----
```

```
MCv_PositionControl_3 (
  Axis:= Axis0003.AxisRef ,
  Execute:= bMCv_PositionControl_Execute,
  ContinuousUpdate:= TRUE ,
  Position:= leMCv_PositionControl_Pos_3
  Velocity:= 10000000.0 ,
  Acceleration:= 100000.0 ,
  Deceleration:= 100000.0 ,
  Jerk:= 0.0 ,
  Direction:= MC_DIRECTION_mcPositiveDirection ,
  BufferMode:= MC_BUFFER_MODE_mcAborting ,
  Options:= 0
);
```

46.26 运动循环位置控制

MCv_CyclicPosition

按照指定的目标位置将目标位置发送到驱动器模块。

限制事项

使用时，应确认控制器及工程工具的版本。

所使用的控制器的用户手册

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量 (AxisName, AxisRef.)，请参阅下述内容。 1387页 AxisName, AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	—	FALSE	TRUE时执行本FB。
ContinuousUpdate	连续更新	BOOL	启动时	—	FALSE	TRUE期间可以连续更改目标位置。
Position	目标位置	LREAL	启动时/可重启/可连续更新	■环形计数器无效的情况下 $-10000000000.0 \leq \text{设定值} < 10000000000.0$ ■环形计数器有效的情况下 $\text{环形计数器下限值} \leq \text{设定值} < \text{环形计数器上限值}$	0	在定位范围内设置绝对位置的目标位置。 关于详细内容，请参阅下述章节。 1847页 目标位置(Position)
Direction	方向选择	MC_DIRECTION	启动时/可重启/可连续更新	1、2、3	0	软件行程限位无效时，指定从当前位置向目标位置移动的方向。 关于详细内容，请参阅下述章节。 1848页 方向选择(Direction)
BufferMode	缓冲模式	MC_BUFFER_MODE	启动时	mcAborting	mcAborting	设置缓冲模式。 关于详细内容，请参阅下述章节。 1852页 缓冲模式(BufferMode)
Options	选项	DWORD (HEX)	启动时	固定为0	0	在本FB中不能设置。应设置为0。设置了0以外的情况下，将变为超出Options范围(错误代码: 1ABBH)。

■输出变量

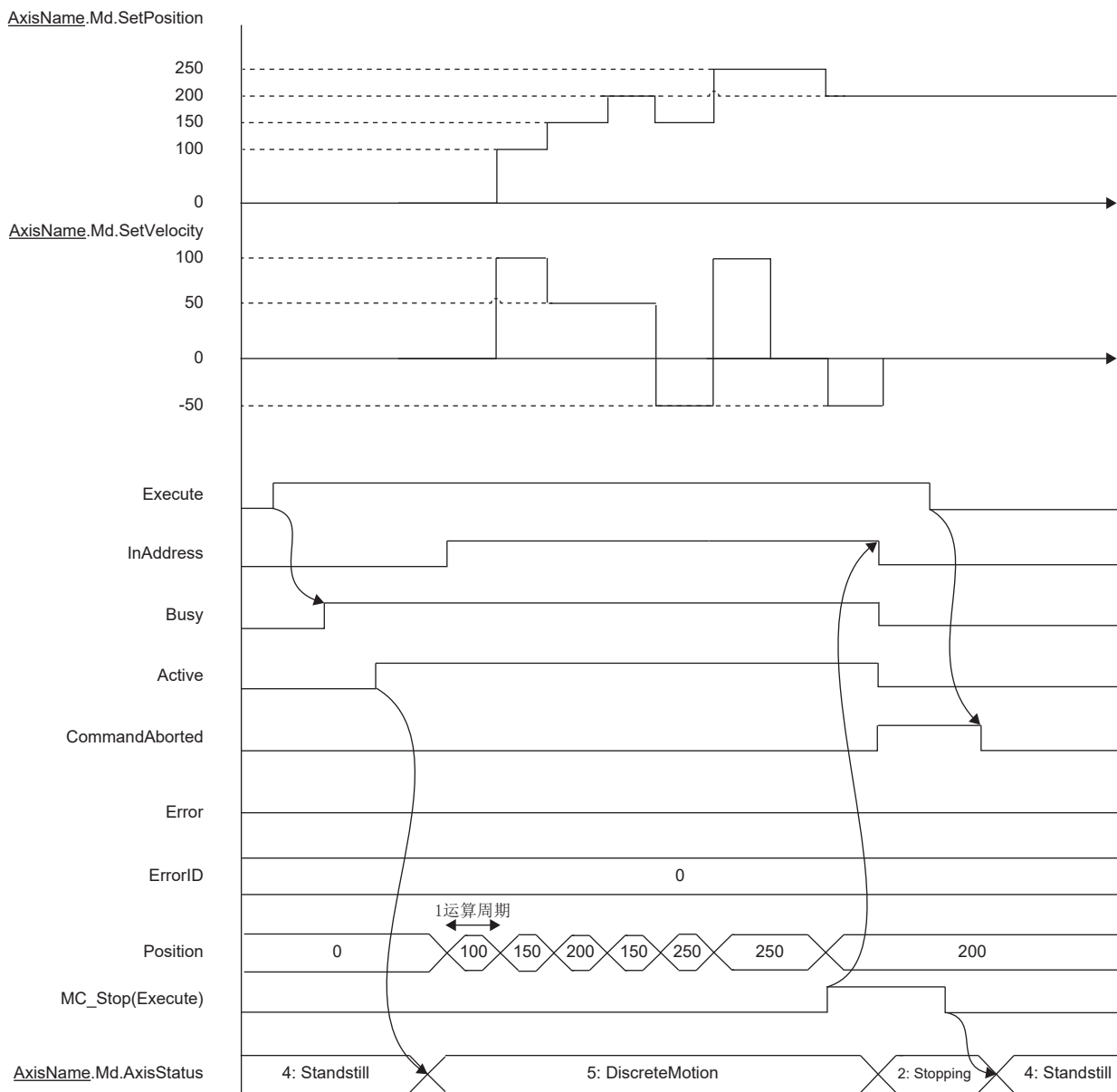
输出变量	名称	数据类型	初始值	内容
InAddress	目标位置到达	BOOL	FALSE	AxisName.Md.SetPosition到达了目标位置时，将变为TRUE。 异常时或发生停止原因时，将变为FALSE。
Busy	执行中	BOOL	FALSE	本FB动作中时，将变为TRUE。
Active	控制中	BOOL	FALSE	本FB正在控制轴时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	通过其他FB中断执行时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 □所使用的控制器的用户手册

功能

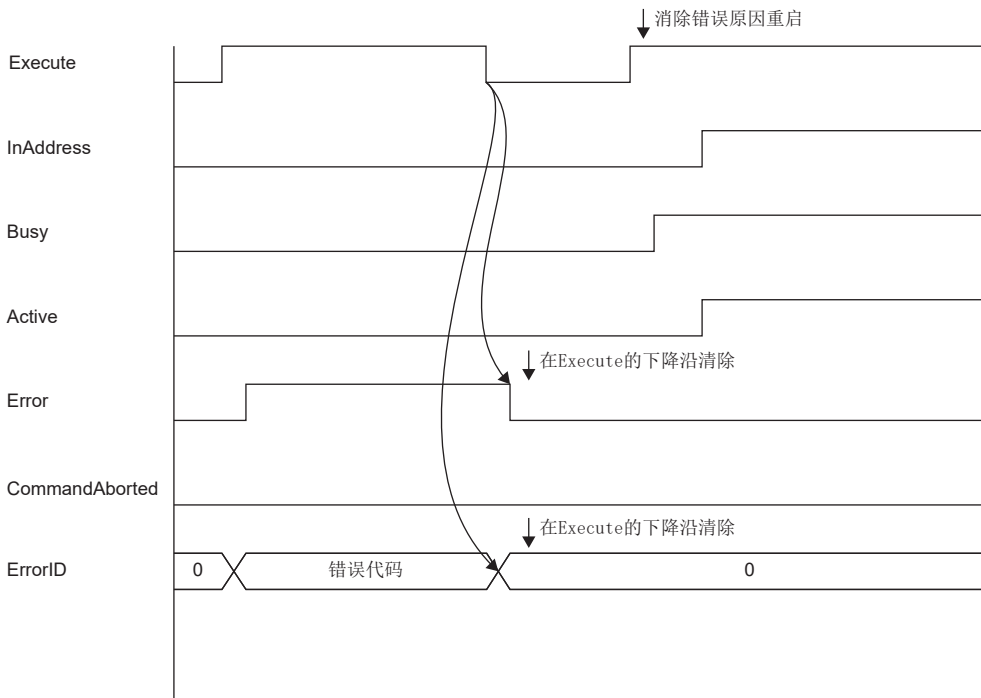
将驱动器模块侧的控制模式切换为csp(循环位置模式)，并发送指令位置。停止本FB时，应使用MC_Stop(强制停止)。

■时序图

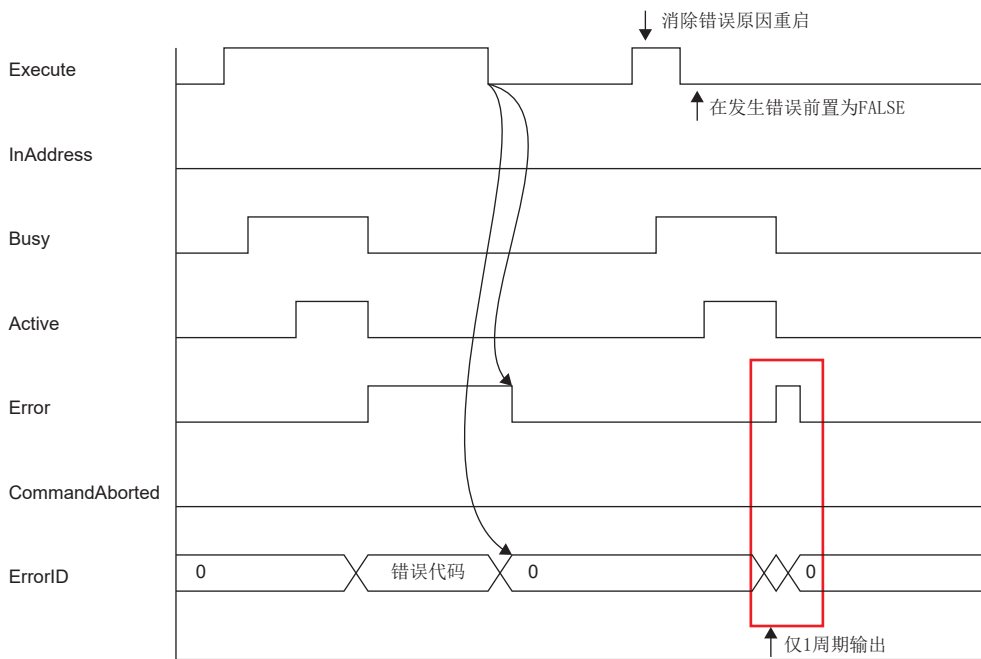
- 正常完成的情况下



• 输入输出变量异常的情况下



• 输入变量异常的情况下



■目标位置(Position)

在定位范围内设置绝对位置的目标位置。有效范围如下所示。设置了超出范围的值的情况下，将变为超出目标位置范围(错误代码：1A85H)。

环形计数器的设置	有效范围
无效	$-10000000000.0 \leq \text{目标位置} < 10000000000.0$
有效	$\text{环形计数器下限值} \leq \text{目标位置} < \text{环形计数器上限值}$

软件行程限位有效时(软件行程限位对象(AxisName.Md.SwStrokeLimit_Target)为“-1: 无效(Invalid)”以外时)，以不超出软件行程限位范围的方向进行定位控制。正方向、负方向均不超出软件行程限位范围的情况下，以当前位置为基准，以趋近目标位置的方向(移动量的绝对值较短的一方)进行定位控制。正方向、负方向距离相同的情况下，沿用上次执行的运动控制FB的移动方向或上次的目标位置更改时的动作方向执行动作。此外，即使在有效范围内将软件行程限位溢出的位置作为目标位置指定的情况下，也将发生软件行程限位溢出(目标位置)(错误代码：1A80H)。

软件行程限位无效时(软件行程限位对象(AxisName.Md.SwStrokeLimit_Target)为“-1: 无效(Invalid)”时)的动作，请参阅下述章节。

☞ 1848页 方向选择(Direction)

更改目标位置(Position)的情况下，应在下述范围内进行更改。不满足下述范围且电子齿轮转换后的值设置了超出31位的值的情况下，驱动器位置转换后将发生超出移动量范围(错误代码：1B3CH)。

目标位置的变化量 \times 驱动器单位转换分子/驱动器单位转换分母 ≤ 2000000000

例

按照下述条件计算目标位置的示例如下所示。

项目	值
电机编码器分辨率	67108864[pulse/rev]
滚珠丝杠螺距	1000[μm]
外置齿轮比	1/10

电子齿轮设置在伺服电机每转的脉冲数为67108864[pulse]，伺服电机每转的移动量为100.0[μm]时，目标位置的变化量在下述范围以下。

$$\text{目标位置的变化量} = 2000000000 \div \frac{\text{驱动器单位转换分子}}{\text{驱动器单位转换分母}} = 2000000000 \div \frac{\text{每转的脉冲数}}{\text{每转的移动量}} = 2000000000 \div \frac{67108864}{100.0} = 2980.23\dots$$

■方向选择 (Direction)

软件行程限位无效时，指定从当前位置向目标位置移动的方向。

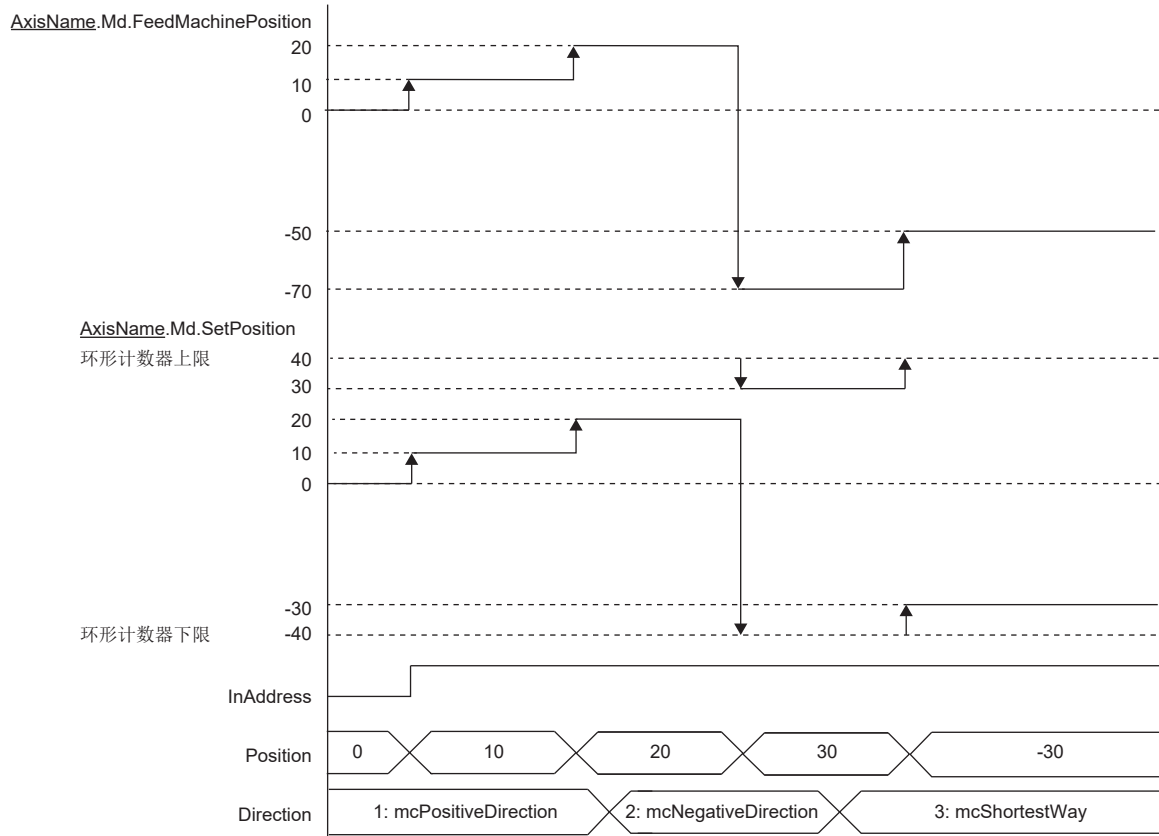
设置范围可以从“1: 正方向(mcPositiveDirection)”、“2: 负方向(mcNegativeDirection)”、“3: 最短路径(mcShortestWay)”中选择。

软件行程限位有效时，忽略方向选择(Direction)的设置。

启动时方向选择(Direction)设置了超出范围的值的的情况下，将发生超出方向选择范围(错误代码: 1AA5H)，不启动。

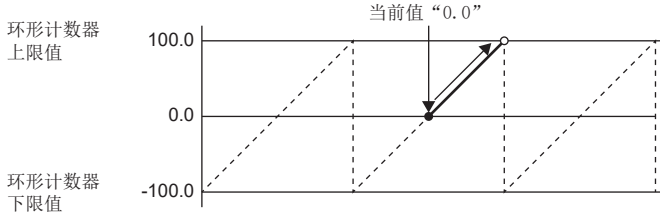
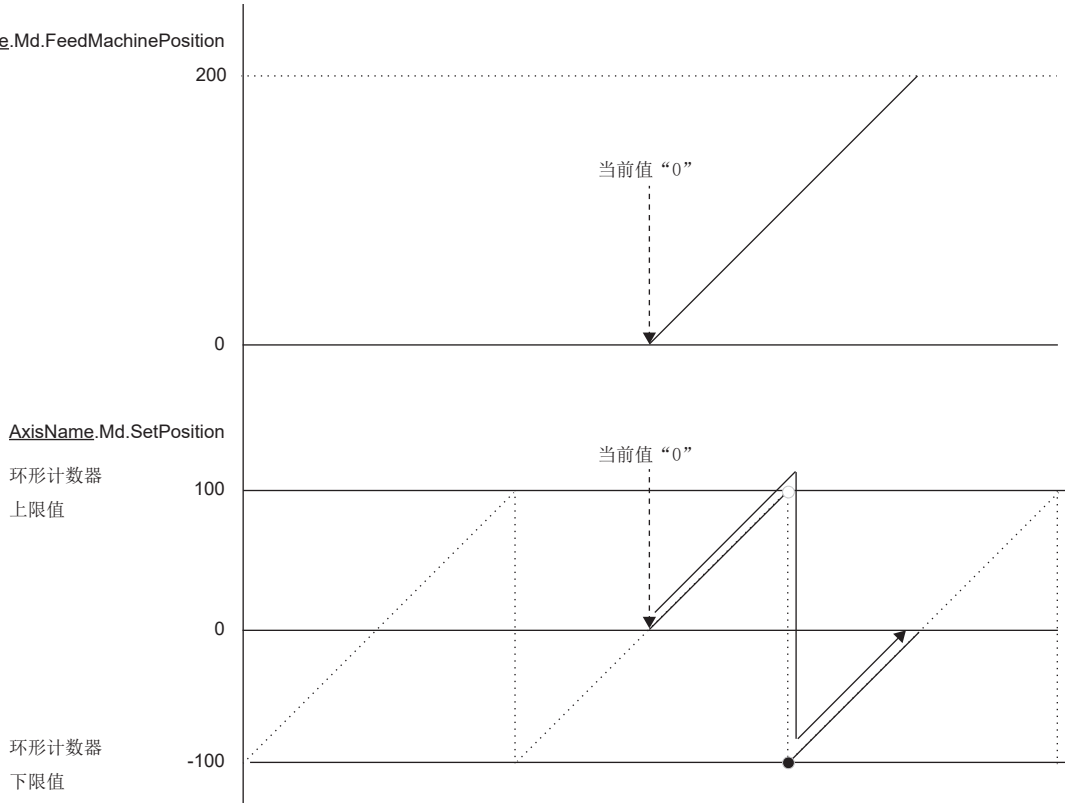
控制中时方向选择(Direction)设置了超出范围的值的的情况下，将发生超出方向选择范围警告(事件代码: 00D4DH)，继续更改前的动作。另外，此时如果同时更改目标位置(Position)和方向选择(Direction)，则按照更改前的目标位置(Position)的设置继续动作。

根据方向选择(Direction)的移动方向，运动控制FB启动时以启动时的指令当前位置为基准执行动作，运动控制FB启动后以目标位置更改时的指令当前位置为基准执行动作。



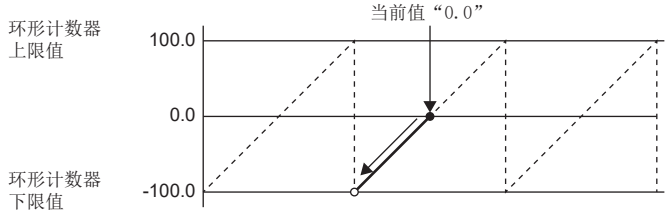
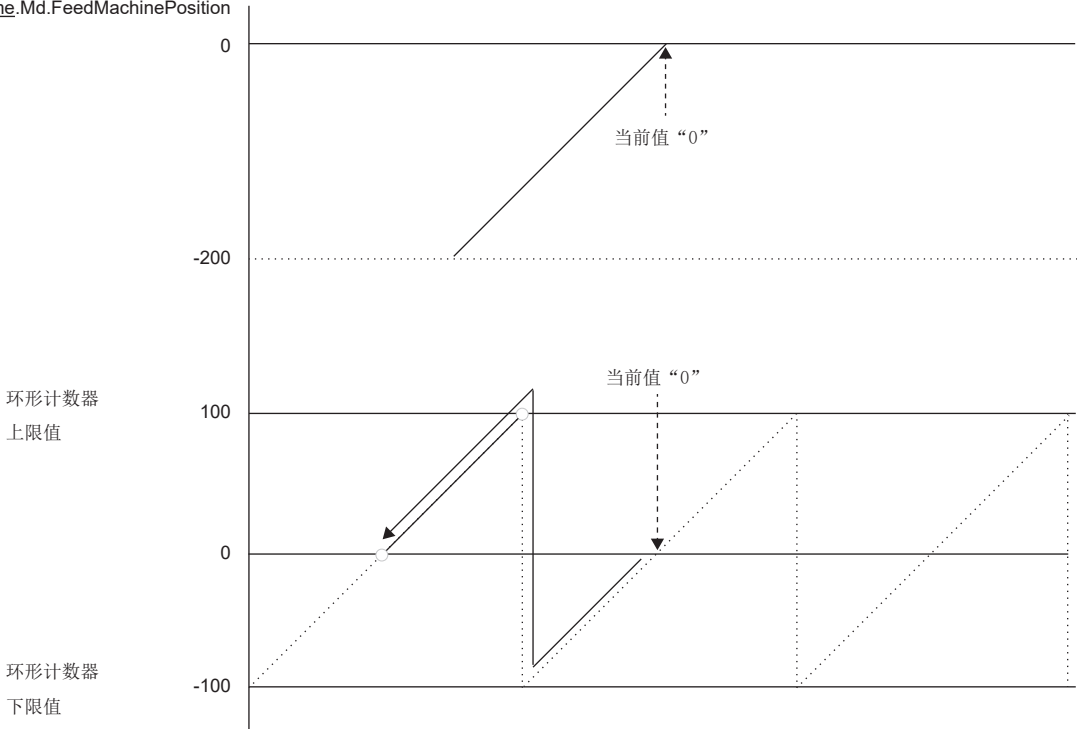
- 1: 正方向(mcPositiveDirection)

从当前位置向目标位置进行正方向(地址增加)的定位。

设置范围	动作
当前值 ≤ 目标位置 < 环形计数器上限值	
环形计数器下限值 ≤ 目标位置 < 当前值	

• 2: 负方向(mcNegativeDirection)

从当前位置向负方向(地址减少)的目标位置进行定位。

设置范围	动作
环形计数器下 限值 ≤ 目标位 置 ≤ 当前值	
当前值 < 目标 位置 < 环形计 数器上限值	<p data-bbox="295 517 598 539"><u>AxisName.Md.FeedMachinePosition</u></p> 

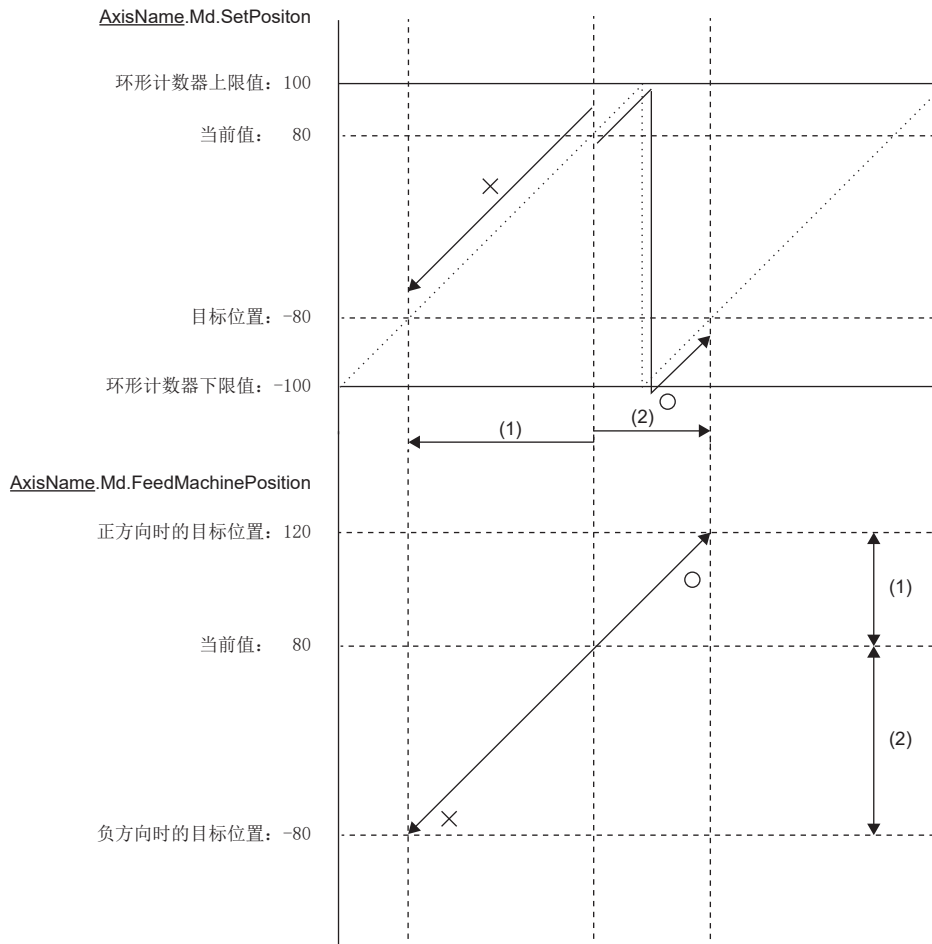
• 3: 最短路径 (mcShortestWay)

以当前位置为基准，以趋近目标位置的方向 (移动量的绝对值较短的一方) 进行定位控制。正方向、负方向距离相同的情况下以当前方向执行动作。目标位置的设置范围为“环形计数器下限值 \leq 目标位置 $<$ 环形计数器上限值”。设置了超出范围的值的情况下，将变为超出目标位置范围 (错误代码: 1A85H) 且不启动。

例

设置了下述值时的动作

- 环形计数器上限值: 100.0
- 环形计数器下限值: -100.0
- 当前位置: 80.0
- 目标位置: -80.0



(1)的情况下移动量变为“160.0”，(2)的情况下移动量变为“40.0”，因此以(2)的正方向移动。

■缓冲模式(BufferMode)


设置用于进行多重启动(缓冲模式)的动作。

本FB中可设置的缓冲模式如下所示。

设定值	内容
0: Aborting(mcAborting)	中断(取消)控制中的FB并立即执行下一个FB。

要点

关于多重启动(缓冲模式)的详细内容, 请参阅下述手册。


所使用的控制器的用户手册

注意事项

- 运动循环位置控制不进行加速度、减速度、Jerk的设置, 而是发送指令位置, 因此有可能进行急剧的加减速。因此, 应在速度限制值(正方向时正方向速度限制值(AxisName.Pr.VelocityLimit_Positive)、负方向时负方向速度限制值(AxisName.Pr.VelocityLimit_Negative))中设置考虑了安全的值。
- 指令当前速度超出各轴的速度限制值时, 按照速度限制值溢出时动作设置(AxisName.Pr.VelocityLimit_OverOperation)。此外, 指令当前速度的绝对值超出最大速度的情况下, 将变为控制中速度范围溢出(错误代码: 1B1AH), 停止运行。
- 在多重启动MCv_CyclicPosition(运动循环位置控制)后, 从csp以外的控制模式切换为csp的情况下, 应确认电机已停止后再进行切换。在未停止电机的状态下进行了切换时, 由于当前位置与MCv_CyclicPosition(运动循环位置控制)的差异将变大, 有可能引起急剧动作。

程序示例

关于本FB的程序示例, 请参阅下述手册。

所使用的控制器的用户手册

46.27 运动循环速度控制

MCv_CyclicVelocity

按照指定的目标速度将目标速度发送到驱动器模块。

限制事项

使用时，应确认控制器及工程工具的版本。

📖所使用的控制器的用户手册

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName, AxisRef.)，请参阅下述内容。 📖 1387页 AxisName, AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	—	FALSE	TRUE时执行本FB。
ContinuousUpdate	连续更新	BOOL	启动时	—	FALSE	TRUE期间可以连续更改目标位置。
Velocity	目标速度	LREAL	启动时/可重启/ 可连续更新	-2500000000.0~2500000000.0	0	设置目标速度。 关于详细内容，请参阅下述章节。 📖 1856页 目标速度(Velocity)
Direction	方向选择	MC_DIRECTION	启动时	mcPositiveDirection正方向 (1)、mcNegativeDirection负 方向(2)	0	指定方向。 关于详细内容，请参阅下述章节。 📖 1856页 方向选择(Direction)
BufferMode	缓冲模式	MC_BUFFER_MODE	启动时	mcAborting	mcAborting	设置缓冲模式。 关于详细内容，请参阅下述章节。 📖 1856页 缓冲模式(BufferMode)
Options	选项	DWORD (HEX)	启动时	固定为0	0	在本FB中不能设置。应设置为0。设置了0以外的情况下，将变为超出Options范围(错误代码: 1ABBH)。

■输出变量

输出变量	名称	数据类型	初始值	内容
InVelocity	目标速度到达	BOOL	FALSE	AxisName.Md.IO_TargetVelocity到达了目标速度时, 将变为TRUE。 异常时或发生停止原因时, 将变为FALSE。
Busy	执行中	BOOL	FALSE	本FB执行中时, 将变为TRUE。
Active	控制中	BOOL	FALSE	本FB正在控制轴时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	通过其他FB中断执行时, 将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时, 将返回错误代码。 关于错误代码的详细内容, 请参阅下述手册。 □□所使用的控制器的用户手册

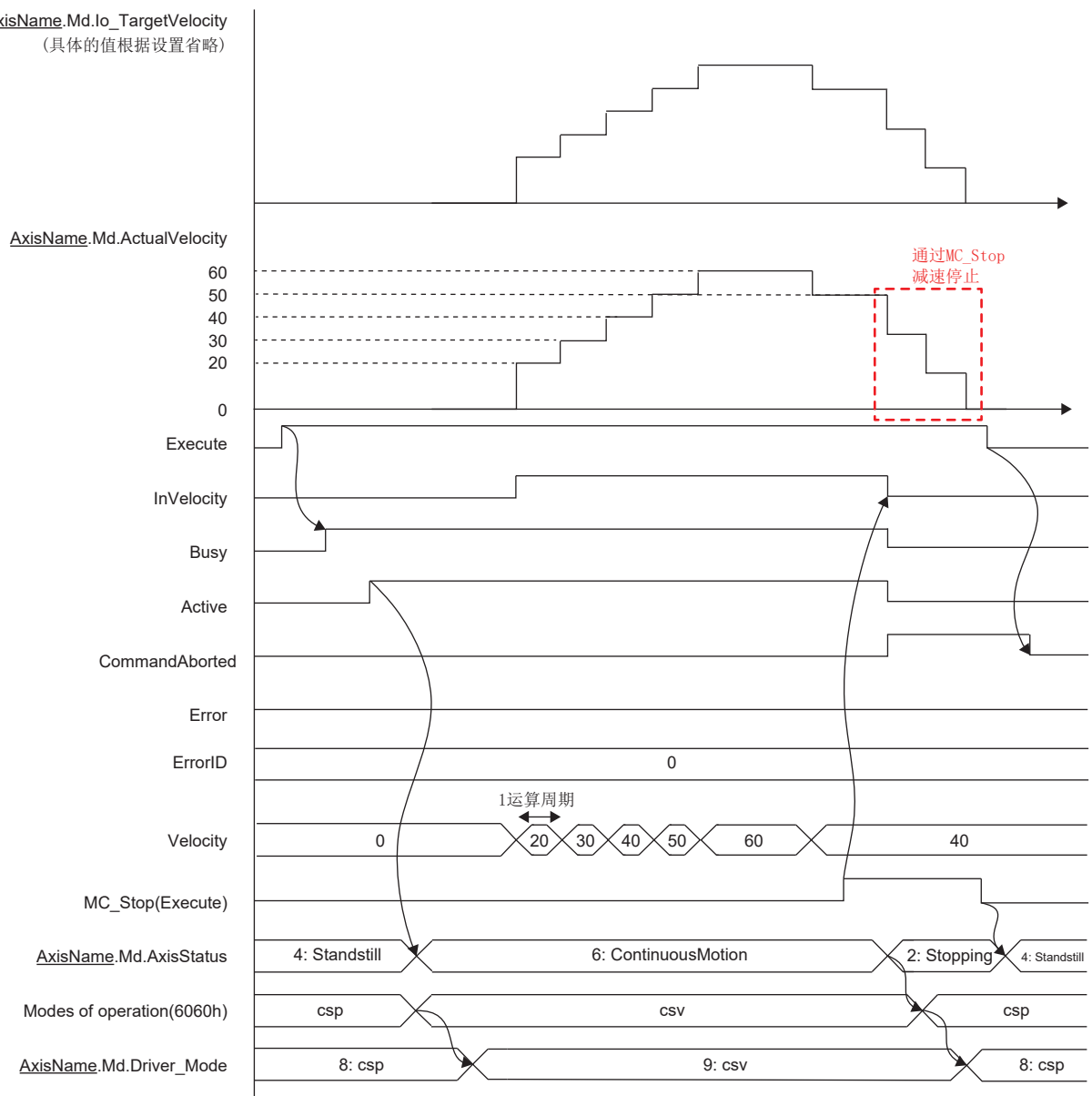
功能

将驱动器模块侧的控制模式切换为csv(循环速度模式), 并发送指令速度。停止本FB时, 应使用MC_Stop(强制停止)。

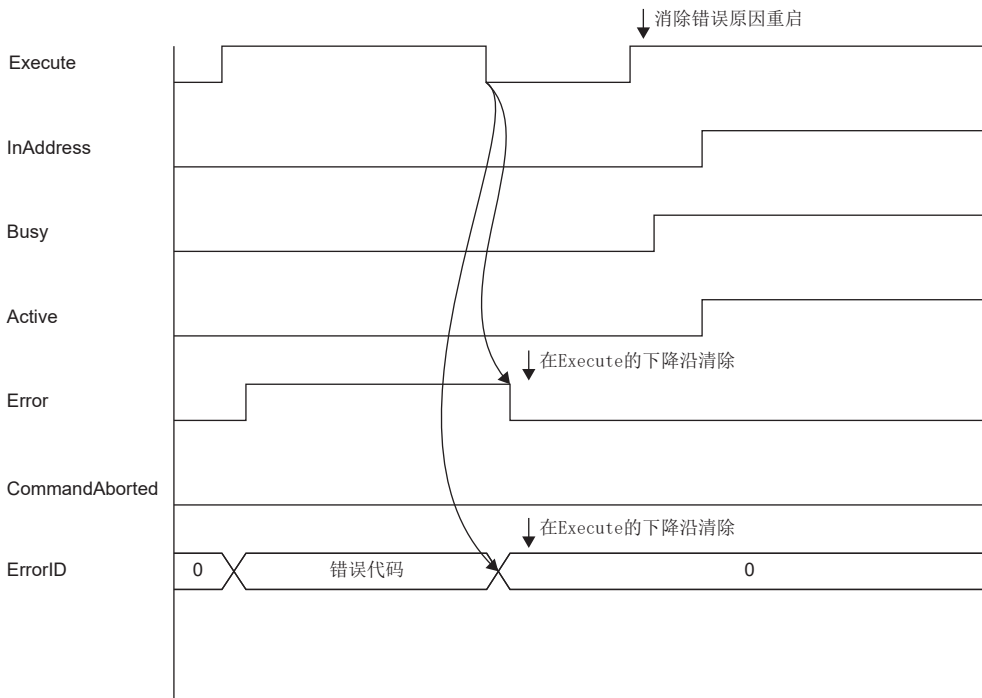
■时序图

- 正常完成的情况下

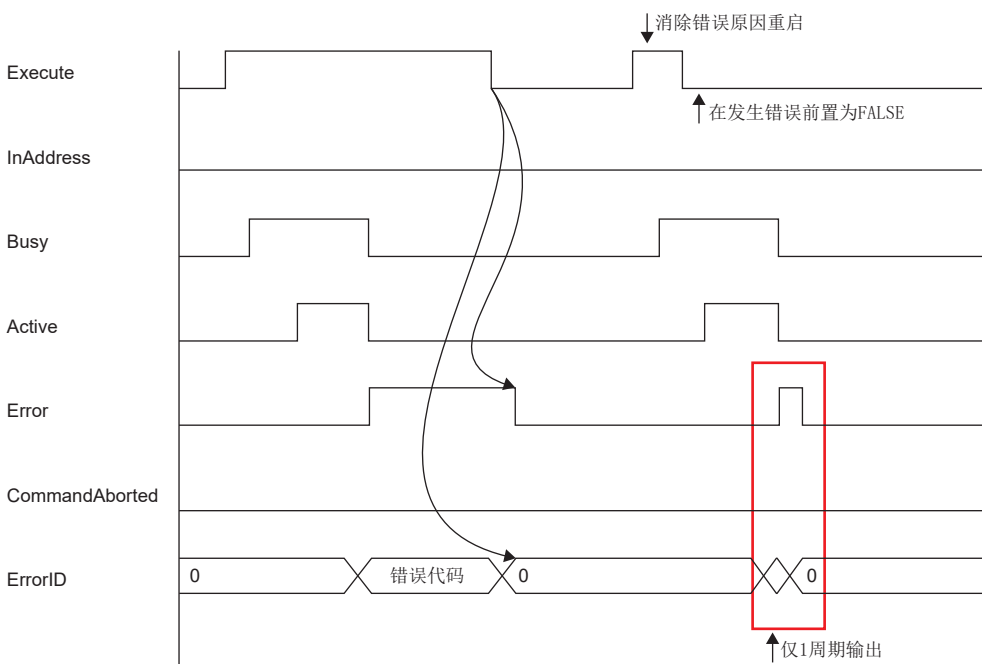
AxisName.Md.io_TargetVelocity
(具体的值根据设置省略)



• 输入输出变量异常的情况下



• 输入变量异常的情况下



■目标速度(Velocity)

设置目标速度。启动时设置了超出范围的值的的情况下，将变为超出速度范围(错误代码：1ABAH)。控制中设置了超出范围的值的的情况下，将发生超出速度范围警告(事件代码：00D0FH)。

设置时应使驱动器速度单位转换后的指令速度在-2147483648~2147483647的范围内。转换后的指令速度超出范围的情况下，与动作状态无关，驱动器速度单位转换后将发生超出指令速度范围(错误代码：1A90H)。

驱动器速度单位转换后的指令速度可通过下述公式求出。

驱动器速度单位转换后的指令速度=速度指令单位×(驱动器单位转换分子÷驱动器单位转换分母)

速度设置为负的情况下，向负方向移动。

启动时设置了0的情况下，轴不动作但对对象轴的轴状态(AxisName.Md.AxisStatus)将变为“6：连续动作运行中(ContinuousMotion)”。

多重启动本FB且目标速度(Velocity)中设置了0的情况下，以之前运动控制FB的指令速度执行动作。

■方向选择(Direction)

指定从当前位置向目标位置移动的方向。

设置范围可以从“1：正方向(mcPositiveDirection)”、“2：负方向(mcNegativeDirection)”、“3：最短路径(mcShortestWay)”中选择。设置“2：负方向(mcNegativeDirection)”，且将目标速度(Velocity)设置为了负的情况下，电机的移动方向将变为正方向。

启动时方向选择(Direction)设置了超出范围的值的的情况下，将发生超出方向选择范围(错误代码：1AA5H)，不启动。

■缓冲模式(BufferMode)


设置用于进行多重启动(缓冲模式)的动作。

本FB中可设置的缓冲模式如下所示。

设定值	内容
0: Aborting(mcAborting)	中断(取消)控制中的FB并立即执行下一个FB。

要点

关于多重启动(缓冲模式)的详细内容，请参阅下述手册。

所使用的控制器的用户手册


注意事项

设置目标速度时，应设置值以避免当前速度与目标速度的差过大。差较大的情况下，轴可能会急加速。

启动本FB中发生了停止原因的情况下，将立即停止。

程序示例

关于本FB的程序示例，请参阅下述手册。

所使用的控制器的用户手册

46.28 运动循环转矩控制

MCv_CyclicTorque

按照指定的目标转矩将目标转矩发送到驱动器模块。

限制事项

使用时，应确认控制器及工程工具的版本。

📖 所使用的控制器的用户手册

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
Axis	轴信息	AXIS_REF	启动时	—	不能省略	设置轴。 关于使用的变量(AxisName.AxisRef.)，请参阅下述内容。 ☞ 1387页 AxisName.AxisRef. (轴信息)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	—	FALSE	TRUE时执行本FB。
ContinuousUpdate	连续更新	BOOL	启动时	—	FALSE	TRUE期间可以连续更改目标位置。
Torque	目标转矩	LREAL	启动时/可重启/ 可连续更新	-1000.0~1000.0[%]	0	以使用的伺服电机的额定转矩的比率进行设置。 关于详细内容，请参阅下述章节。 ☞ 1861页 目标转矩(Torque)
LimitVelocity	限制速度	LREAL	启动时/可重启/ 可连续更新	-2500000000.0~ 2500000000.0	0	指定方向。 关于详细内容，请参阅下述章节。 ☞ 1861页 限制速度(LimitVelocity)
BufferMode	缓冲模式	MC_BUFFER_ MODE	启动时	mcAborting	mcAborting	设置缓冲模式。 关于详细内容，请参阅下述章节。 ☞ 1861页 缓冲模式(BufferMode)
Options	选项	DWORD (HEX)	启动时	固定为0	0	在本FB中不能设置。应设置为0。设置了0以外的 情况下，将变为超出Options范围(错误代码： 1ABBH)。

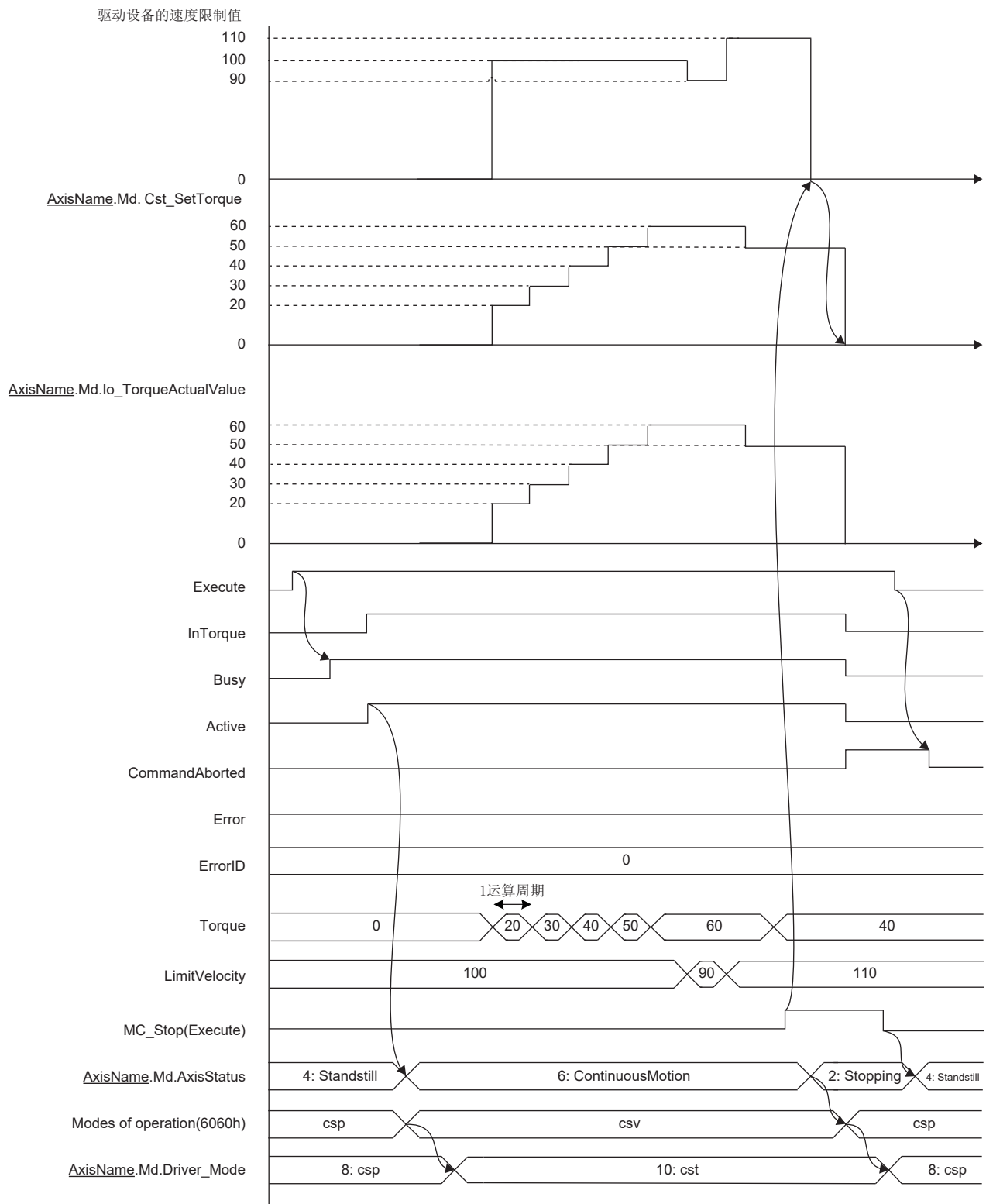
■输出变量

输出变量	名称	数据类型	初始值	内容
InTorque	目标转矩到达	BOOL	FALSE	驱动器模块发送的Cst_SetTorque到达了目标转矩时，将变为TRUE。 异常时或由于发生停止原因导致轴停止后，将变为FALSE。
Busy	执行中	BOOL	FALSE	本FB动作中时，将变为TRUE。
Active	控制中	BOOL	FALSE	本FB正在控制轴时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	通过其他FB中断执行时，将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时，将返回错误代码。 关于错误代码的详细内容，请参阅下述手册。 □□所使用的控制器的用户手册

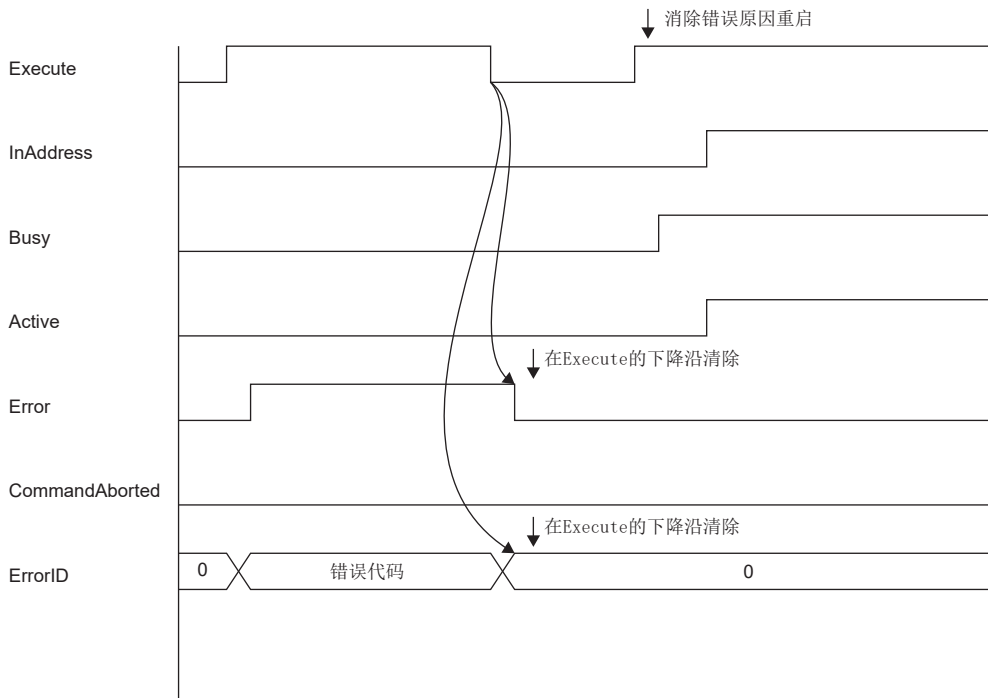
将驱动器模块侧的控制模式切换为cst(循环转矩模式)，并发送指令速度。停止本FB时，应使用MC_Stop(强制停止)。

■时序图

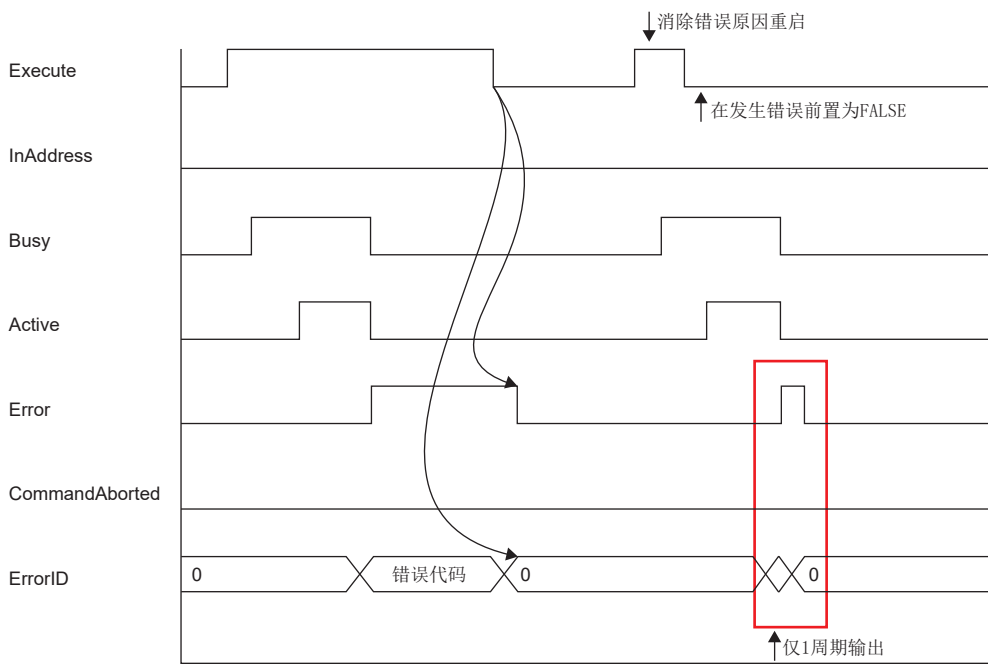
- 正常完成的情况下



• 输入输出变量异常的情况下



• 输入变量异常的情况下



■目标转矩 (Torque)

对使用的伺服电机的额定转矩的比率以%单位进行设置。

启动时指定了超出范围的值的情况下，将发生超出目标转矩范围(错误代码：1A96H)。

控制中设置了超出范围的值的情况下，将发生超出转矩设置范围警告(事件代码：00D16H)。

目标转矩的有效位数根据连接的驱动器模块的Target Torque的规格而有所不同。目标转矩中设置的值中包含Target Torque的有效位数以下的值的情况下，有效位数以下的值将被舍去。

将指令转矩向正方向(地址增加方向)输出的情况下应设置正的值，向负方向(地址减少方向)输出的情况下应设置负的值。

为了安全起见，对目标转矩的方向进行反转的情况下，应先将目标转矩设置为0。

例

将目标转矩从-10.0设置为10.0的情况下，按照下述顺序进行更改。

-10.0→0.0→10.0

■限制速度 (LimitVelocity)

将循环转矩模式时的速度限制值设置在-2500000000.0~2500000000.0的范围内。设置了超出范围的值的情况下，启动时将发生超出限制速度范围(错误代码：1A86H)，控制中将发生超出限制速度范围警告(事件代码：00D1BH)。

设置时应使驱动器速度单位转换后的指令速度不超出-2147483648~2147483647[0.01rpm]的范围。设置了超出范围的值的情况下，驱动器速度单位转换后将发生超出指令速度范围(错误代码：1A90H)。

驱动器速度单位转换后的指令速度可通过下述公式求出。

驱动器速度单位转换后的指令速度=速度指令单位×(驱动器单位转换分子÷驱动器单位转换分母)

启动时设置了0的情况下，轴不动作但对象轴的轴状态(AxisName. Md. AxisStatus)将变为“6: 连续动作运行中(ContinuousMotion)”。

多重启动MC_CyclicTorque(转矩控制)且LimitVelocity(限制速度)中存储了0的情况下，将之前的运动控制FB的电机旋转数作为限制速度(LimitVelocity)进行控制。

限制速度(LimitVelocity)满足下述条件的情况下将有效。

- 对支持连接的驱动器模块的循环转矩模式(cst)中的限制速度(LimitVelocity)的对象进行PDO映射分配。
- 设置映射到vVelLimitValue(AxisName.PrConst.SlaveObject.vVelLimitValue)的对象数据。

将限制速度(LimitVelocity)中的速度限制置为有效的方法，根据驱动器模块而有所不同。

- 使用MR-J5(W)-G的情况下，应对Velocity limit value进行PDO映射分配。
- 使用MR-J5(W)-G以外的驱动器模块的情况下，对支持连接的驱动器模块的循环转矩模式(cst)中的限制速度的对象进行PDO映射分配。此外还应设置映射到vVelLimitValue(AxisName.PrConst.SlaveObject.vVelLimitValue)的对象数据。

限制速度(LimitVelocity)中的速度限制无效的情况下，控制器的速度指令不变为有效。

- 使用MR-J5(W)-G的情况下，应通过工程工具设置伺服参数[Pr.PT67 速度限制]的设定值。
- 使用MR-J5(W)-G以外的驱动器模块的情况下，应使用连接的驱动器模块侧的速度限制功能。关于详细内容，请参阅所连接的驱动器模块的用户手册。

■缓冲模式 (BufferMode)

设置用于进行多重启动(缓冲模式)的动作。

本FB中可设置的缓冲模式如下所示。

设定值	内容
0: Aborting(mcAborting)	中断(取消)控制中的FB并立即执行下一个FB。

要点

关于多重启动(缓冲模式)的详细内容，请参阅下述手册。

📖所使用的控制器的用户手册

注意事项


- 通过跟踪更新指令当前位置、进给机械位置。
- 通过本FB切换控制模式所需的时间，根据所使用的驱动器模块的规格而有所不同。
- 启动本FB中发生了停止原因的情况下，将立即停止。
- 请勿在本FB动作中启动定位控制FB。驱动器控制模式(AxisName.Md.Driver_Mode)切换为“10: 转矩控制(cst)”后，应启动定位控制FB。
- 目标转矩中设置的转矩不足的情况下，根据机构及负载转矩可能会急剧消失。减小目标转矩的情况下，应在充分注意安全的前提下进行设置。
- 轴已停止的情况下，应在切换控制模式后将值存储到限制速度(LimitVelocity)。切换前限制速度(LimitVelocity)中存储了值的情况下，速度可能会急剧增加，发生危险。
- cst中建议使用限制速度(LimitVelocity)。不使用限制速度(LimitVelocity)时，当前转矩及目标转矩的差异较大的情况下，根据机构及施加负载方式，速度可能会急剧增加，发生危险。
- 关于cst的转矩限制更改，应注意在下述更改时的动作。

更改内容	动作
通过重启或连续更新，将目标转矩更改为大于转矩限制值(正方向转矩限制值(AxisName.Md.TorqueLimit_Positive)、负方向转矩限制值(AxisName.Md.TorqueLimit_Negative))的值	发生转矩限制值溢出警告(事件代码: 00D12H)，按照更改前的设置继续动作。
转矩控制中，更改为小于目标转矩的正方向转矩限制值(AxisName.Cd.TorqueLimit_Positive)、负方向转矩限制值(AxisName.Cd.TorqueLimit_Negative)	将目标转矩更改为转矩限制值。此时，指令转矩以1运算周期进行更改。

- 本FB动作中多重启动MC_TorqueControl(转矩控制)后，不能进行挡块控制模式的切换。执行了控制模式切换的情况下，将发生控制模式不可切换警告(事件代码: 00D31H)，以当前控制模式执行动作。
- 挡块控制中不能执行本FB。执行了切换的情况下，将发生控制模式切换异常(错误代码: 1A95H)。
- 在多重启动MCv_CyclicTorque(运动循环转矩控制)后，从cst切换控制模式时，将立即发送MCv_CyclicTorque(运动循环转矩控制)中设置的指令转矩。设置时应避免多重启动前后的指令转矩及限制速度的差变大。若差异较大，则可能会引起急剧动作。

程序示例

关于本FB的程序示例，请参阅下述手册。

所使用的控制器的用户手册

47.1 配置文件读取

MCv_ReadProfileData

从展开区域或文件读取指定的运算配置文件。

梯形图	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>

设置数据

■输入输出变量

输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
ProfileData	配置文件数据	PROFILE_DATA	启动时	—	不能省略	设置要读取的运算配置文件。 在PROFILE_DATA结构体中设置运算配置文件。关于PROFILE_DATA结构体，请参阅下述章节。 ☞ 1431页 PROFILE_DATA 对于定义了读取用数据结构体的运算配置文件格式，可以进行读取操作。
Data1	读取数据1	TARGET_REF	启动时	—	不能省略	根据要读取的运算配置文件格式设置读取用数据结构体。关于详细内容，请参阅下述章节。 ☞ 1870页 读取数据1(Data1)/读取数据2(Data2)
Data2	读取数据2	TARGET_REF	启动时	—	不能省略	根据要读取的运算配置文件格式设置读取用数据结构体。关于详细内容，请参阅下述章节。 ☞ 1870页 读取数据1(Data1)/读取数据2(Data2)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	启动	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MCv_ReadProfileData(配置文件读取)。
Offset	偏置	DWORD(UDINT)	启动时	0~分辨率(2~65535)	0	设置从运算配置文件起始开始的偏置。 偏置(Offset)超出运算配置文件的范围时，将变为超出偏置范围(错误代码：1B92H)，不进行读取。 在偏置(Offset)中设置“0”以外的值的情况下，读取数据数(Points)中应设置“0”以外的值。(在读取数据数(Points)中设置了“0”的情况下，将变为偏置·读取/写入数据数不正确(错误代码：1BADH)。) * 根据偏置(Offset)与读取数据数(Points)的设定值，读取的动作有所不同。关于详细内容，请参阅下述章节。 ☞ 1865页 偏置(Offset)与读取数据数(Points)的设置
Points	读取数据数	DWORD(UDINT)	启动时	0~4294967295	0	设置要读取的数据点数。 读取数据数(Points)超出运算配置文件的元素数的情况下，将在运算配置文件的元素数的范围内进行读取。 运算配置文件的插补方法指定(Interpolate)为“1:各区间中指定”或“2:样条插补”的情况下，应设置“0”。(设置了“0”以外的情况下，将变为偏置·读取/写入数据数不正确(错误代码：1BADH)。)

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Target	读取目标	WORD (UINT)	启动时	0、1、2	0	设置运算配置文件的读取目标。 <ul style="list-style-type: none"> • 0: 展开区域 • 1: 文件 • 2: 展开区域(运动服务处理) 关于详细内容, 请参阅下述章节。 ☞ 1872页 读取目标(Target)

■输出变量

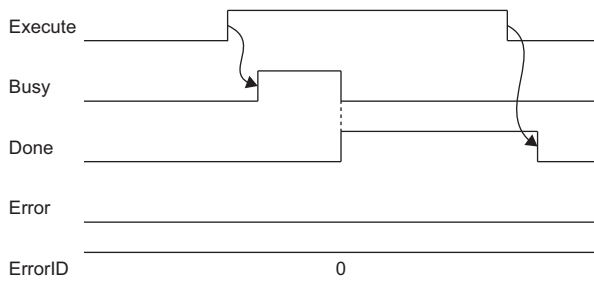
输出变量	名称	数据类型	初始值	内容
Done	完成	BOOL	FALSE	控制完成时, 将变为TRUE。 动作完成时根据启动(Execute)的状态将变为如下所示。 ■启动(Execute)为TRUE的情况下 在将启动(Execute)置为FALSE之前将保持为TRUE不变。 ■启动(Execute)为FALSE的情况下 仅1周期为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_ReadProfileData(配置文件读取)时, 将变为TRUE。
Error	错误	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	错误代码	WORD (UINT)	0	发生了异常时, 将返回错误代码。 关于错误代码的详细内容, 请参阅下述手册。 ☞ 所使用的控制器的用户手册

功能

- 使用MCv_ReadProfileData(配置文件读取)进行运算配置文件的读取操作。
- 对于配置文件数据(ProfileData)及读取目标(Target)中设置的文件或展开区域的运算配置文件, 从偏置(Offset)中设置的数据中读取在读取数据数(Points)中设置的点数。
- 读取整个运算配置文件时, 在偏置(Offset)与读取数据数(Points)的两者中设置“0”。
- 读取的数据存储到读取数据1(Data1)、读取数据2(Data2)中设置的变量中。

■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图, 请参阅下述章节。

[☞ 1375页 通过执行指令\(Execute\)类型的运动控制FB的基本动作](#)

■偏置(Offset)与读取数据数(Points)的设置

根据偏置(Offset)与读取数据数(Points)中设置的值,将变为下述动作。

偏置(Offset)	读取数据数(Points)	与运算配置文件的元素数的大小关系	动作
0	0	—	读取运算配置文件的所有元素。
	0以外	偏置(Offset) + 读取数据数(Points) ≤ 运算配置文件的元素数	从起始开始按照读取数据数(Points)中设置的点数读取运算配置文件的元素。
	0以外	偏置(Offset) + 读取数据数(Points) > 运算配置文件的元素数	读取运算配置文件的所有元素。
0以外	0	偏置(Offset) ≤ 运算配置文件的元素数	发生偏置・读取/写入数据数不正确(错误代码: 1BADH), 不进行读取。
	0以外	偏置(Offset) + 读取数据数(Points) ≤ 运算配置文件的元素数	从偏置(Offset)开始按照读取数据数(Points)中指定的点数读取运算配置文件的元素。
	0以外	偏置(Offset) + 读取数据数(Points) > 运算配置文件的元素数	从偏置(Offset)开始读取直到运算配置文件的最终元素。
	—	偏置(Offset) > 运算配置文件的元素数	发生超出偏置范围(错误代码: 1B92H), 不进行读取。

对于偏置(Offset)、读取数据数(Points)中设置的值的内容,根据运算配置文件格式而有所不同。

在运算配置文件格式中设置“凸轮数据”、“旋转刀具”的内容如下所示。

- 运算配置文件格式为“凸轮数据”的情况下

插补方法指定(Interpolate)	内容																																											
0: 直线插补	<p>通过偏置(Offset)设置要读取的坐标的起始,通过读取数据数(Points)设置坐标数。 读取数据1(Data1)中存储的数据将变为插补方法指定(Interpolate)为“0: 直线插补”的PROFILE_CAM_DATA型的数据。 读取数据2(Data2)中存储的数据将变为将已设置的坐标数据(输入值(1周期当前值)与输出值(行程)的成对)通过LREAL型的二维数组进行了定义的数据。</p> <p><例> 在偏置(Offset)中设置了“2”,在读取数据数(Points)中设置了“3”时的读取数据2(Data2)的读取结果 (将读取数据2(Data2)中设置的变量设置为数据类型“LREAL[1..8,1..2]”的标签“CamData2”的情况下)</p> <p style="text-align: center;"><变量(标签/软元件)> LREAL型的二维数组的标签 (CamData2[1..8,1..2])</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th></th> <th>CamData[n][1]</th> <th>CamData[n][2]</th> </tr> </thead> <tbody> <tr><td>n=1</td><td>第2点的输入值</td><td>第2点的输出值</td></tr> <tr><td>n=2</td><td>第3点的输入值</td><td>第3点的输出值</td></tr> <tr><td>n=3</td><td>第4点的输入值</td><td>第4点的输出值</td></tr> <tr><td>n=4</td><td>(不更新)</td><td>(不更新)</td></tr> <tr><td>n=5</td><td>(不更新)</td><td>(不更新)</td></tr> <tr><td>n=6</td><td>(不更新)</td><td>(不更新)</td></tr> <tr><td>n=7</td><td>(不更新)</td><td>(不更新)</td></tr> <tr><td>n=8</td><td>(不更新)</td><td>(不更新)</td></tr> </tbody> </table> </div> <div style="text-align: center;"> <p>偏置(Offset)</p> <p>读取数据数(Points)</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: right;"><展开区域/文件>*1</p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tbody> <tr><td>第0点的输入值</td><td>第0点的输出值</td></tr> <tr><td>第1点的输入值</td><td>第1点的输出值</td></tr> <tr><td>第2点的输入值</td><td>第2点的输出值</td></tr> <tr><td>第3点的输入值</td><td>第3点的输出值</td></tr> <tr><td>第4点的输入值</td><td>第4点的输出值</td></tr> <tr><td>第5点的输入值</td><td>第5点的输出值</td></tr> <tr><td>第6点的输入值</td><td>第6点的输出值</td></tr> <tr><td>第7点的输出值</td><td>第7点的输出值</td></tr> </tbody> </table> </div> </div> <p>*1: 取决于读取目标(Target)。</p>		CamData[n][1]	CamData[n][2]	n=1	第2点的输入值	第2点的输出值	n=2	第3点的输入值	第3点的输出值	n=3	第4点的输入值	第4点的输出值	n=4	(不更新)	(不更新)	n=5	(不更新)	(不更新)	n=6	(不更新)	(不更新)	n=7	(不更新)	(不更新)	n=8	(不更新)	(不更新)	第0点的输入值	第0点的输出值	第1点的输入值	第1点的输出值	第2点的输入值	第2点的输出值	第3点的输入值	第3点的输出值	第4点的输入值	第4点的输出值	第5点的输入值	第5点的输出值	第6点的输入值	第6点的输出值	第7点的输出值	第7点的输出值
	CamData[n][1]	CamData[n][2]																																										
n=1	第2点的输入值	第2点的输出值																																										
n=2	第3点的输入值	第3点的输出值																																										
n=3	第4点的输入值	第4点的输出值																																										
n=4	(不更新)	(不更新)																																										
n=5	(不更新)	(不更新)																																										
n=6	(不更新)	(不更新)																																										
n=7	(不更新)	(不更新)																																										
n=8	(不更新)	(不更新)																																										
第0点的输入值	第0点的输出值																																											
第1点的输入值	第1点的输出值																																											
第2点的输入值	第2点的输出值																																											
第3点的输入值	第3点的输出值																																											
第4点的输入值	第4点的输出值																																											
第5点的输入值	第5点的输出值																																											
第6点的输入值	第6点的输出值																																											
第7点的输出值	第7点的输出值																																											

插补方法指定 (Interpolate)	内容																																													
1: 各区间中指定	根据读取目标(Target), 可读取的数据有所不同。																																													
2: 样条插补	<p>■读取目标(Target)为“0: 展开区域”、“2: 展开区域(运动服务处理)”的情况下 通过偏置(Offset)设置要读取的区间的起始, 通过读取数据数(Points)设置区间数(NumberOfSections)。 读取数据1(Data1)中存储的数据将变为插补方法指定(Interpolate)为“0: 直线插补”的PROFILE_CAM_DATA型的数据。 读取数据2(Data2)中存储的数据将变为将对象的凸轮数据以凸轮分辨率的点数等分后定义的数据(1周期当前值与行程的成对)通过LREAL型的二维数组进行了定义的数据。</p> <p><例> 凸轮分辨率为“256”的情况下, 在偏置(Offset)中设置了“2”, 在读取数据数(Points)中设置了“3”时的读取数据2(Data2)的读取结果 (将读取数据2(Data2)中设置的变量设置为数据类型“LREAL[1..8, 1..2]”的标签“CamData2”的情况下)</p> <div style="text-align: center;"> <p><变量(标签/软元件)> LREAL型的二维数组的标签 (CamData2[1..8, 1..2])</p> </div> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td></td> <td style="text-align: center;">CamData[n][1]</td> <td style="text-align: center;">CamData[n][2]</td> <td></td> </tr> <tr> <td>n=1</td> <td>第2点的1周期当前值</td> <td>第2点的行程</td> <td rowspan="8"> <div style="display: flex; align-items: center; justify-content: center;"> <div style="margin-right: 10px;">←</div> <div style="text-align: center;"> <p>偏置 (Offset)</p> <p>读取数据数 (Points)</p> </div> <div style="margin-left: 10px;">→</div> </div> </td> </tr> <tr> <td>n=2</td> <td>第3点的1周期当前值</td> <td>第3点的行程</td> </tr> <tr> <td>n=3</td> <td>第4点的1周期当前值</td> <td>第4点的行程</td> </tr> <tr> <td>n=4</td> <td>(不更新)</td> <td>(不更新)</td> </tr> <tr> <td>n=5</td> <td>(不更新)</td> <td>(不更新)</td> </tr> <tr> <td>n=6</td> <td>(不更新)</td> <td>(不更新)</td> </tr> <tr> <td>n=7</td> <td>(不更新)</td> <td>(不更新)</td> </tr> <tr> <td>n=8</td> <td>(不更新)</td> <td>(不更新)</td> </tr> </table> <div style="margin-left: auto; margin-right: auto; margin-top: 10px;"> <p><展开区域></p> <table border="1" style="border-collapse: collapse;"> <tr> <td style="width: 50px;">开始点</td> <td>初始行程</td> </tr> <tr> <td>第1点的1周期当前值</td> <td>第1点的行程</td> </tr> <tr> <td>第2点的1周期当前值</td> <td>第2点的行程</td> </tr> <tr> <td>第3点的1周期当前值</td> <td>第3点的行程</td> </tr> <tr> <td>第4点的1周期当前值</td> <td>第4点的行程</td> </tr> <tr> <td>第5点的1周期当前值</td> <td>第5点的行程</td> </tr> <tr> <td style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</td> </tr> <tr> <td>第256点的1周期当前值</td> <td>第256点的行程</td> </tr> </table> <p style="text-align: center;">↑ 以凸轮分辨率点数对凸轮数据进行等分的数据</p> </div> <p>■读取目标(Target)为“1: 文件”的情况下 在偏置(Offset)及读取数据数(Points)中设置“0”, 读取所有元素。(设置了“0”以外的情况下, 将变为偏置·读取/写入数据数不正确(错误代码: 1BADH)。) 读取数据1(Data1)中存储的数据将变为插补方法指定(Interpolate)为“1: 各区间中指定”或“2: 样条插补”的PROFILE_CAM_DATA型的数据。 读取数据2(Data2)中存储的数据将变为通过PROFILE_CAM_ELEMENT型的数组进行了定义的区间数据。</p>		CamData[n][1]	CamData[n][2]		n=1	第2点的1周期当前值	第2点的行程	<div style="display: flex; align-items: center; justify-content: center;"> <div style="margin-right: 10px;">←</div> <div style="text-align: center;"> <p>偏置 (Offset)</p> <p>读取数据数 (Points)</p> </div> <div style="margin-left: 10px;">→</div> </div>	n=2	第3点的1周期当前值	第3点的行程	n=3	第4点的1周期当前值	第4点的行程	n=4	(不更新)	(不更新)	n=5	(不更新)	(不更新)	n=6	(不更新)	(不更新)	n=7	(不更新)	(不更新)	n=8	(不更新)	(不更新)	开始点	初始行程	第1点的1周期当前值	第1点的行程	第2点的1周期当前值	第2点的行程	第3点的1周期当前值	第3点的行程	第4点的1周期当前值	第4点的行程	第5点的1周期当前值	第5点的行程	⋮	⋮	第256点的1周期当前值	第256点的行程
	CamData[n][1]	CamData[n][2]																																												
n=1	第2点的1周期当前值	第2点的行程	<div style="display: flex; align-items: center; justify-content: center;"> <div style="margin-right: 10px;">←</div> <div style="text-align: center;"> <p>偏置 (Offset)</p> <p>读取数据数 (Points)</p> </div> <div style="margin-left: 10px;">→</div> </div>																																											
n=2	第3点的1周期当前值	第3点的行程																																												
n=3	第4点的1周期当前值	第4点的行程																																												
n=4	(不更新)	(不更新)																																												
n=5	(不更新)	(不更新)																																												
n=6	(不更新)	(不更新)																																												
n=7	(不更新)	(不更新)																																												
n=8	(不更新)	(不更新)																																												
开始点	初始行程																																													
第1点的1周期当前值	第1点的行程																																													
第2点的1周期当前值	第2点的行程																																													
第3点的1周期当前值	第3点的行程																																													
第4点的1周期当前值	第4点的行程																																													
第5点的1周期当前值	第5点的行程																																													
⋮	⋮																																													
第256点的1周期当前值	第256点的行程																																													

• 运算配置文件格式为“旋转刀具”的情况下

读取目标 (Target)	内容																																													
0: 展开区域 2: 展开区域(运动服务处理)	<p>通过偏置(Offset)设置要读取的区间的起始, 通过读取数据数(Points)设置区间数(NumberOfSections)。读取数据1(Data1)中存储的数据将变为运算配置文件格式为“凸轮数据”, 插补方法指定(Interpolate)为“0: 直线插补”的PROFILE_CAM_DATA型的数据。读取数据2(Data2)中存储的数据将变为将对象的凸轮数据以凸轮分辨率的点数等分后定义的数据(1周期当前值与行程的成对)通过LREAL型的二维数组进行了定义的数据。</p> <p><例> 在偏置(Offset)中设置了“2”, 在读取数据数(Points)中设置了“3”时的读取数据2(Data2)的读取结果(将读取数据2(Data2)中设置的变量设置为数据类型“LREAL[1..8, 1..2]”的标签“CamData2”的情况下)</p> <p style="text-align: center;"><变量(标签/软元件)> LREAL型的二维数组的标签 (CamData2[1..8, 1..2])</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th></th> <th>CamData[n][1]</th> <th>CamData[n][2]</th> </tr> </thead> <tbody> <tr><td>n=1</td><td>第2点的1周期当前值</td><td>第2点的行程</td></tr> <tr><td>n=2</td><td>第3点的1周期当前值</td><td>第3点的行程</td></tr> <tr><td>n=3</td><td>第4点的1周期当前值</td><td>第4点的行程</td></tr> <tr><td>n=4</td><td>(不更新)</td><td>(不更新)</td></tr> <tr><td>n=5</td><td>(不更新)</td><td>(不更新)</td></tr> <tr><td>n=6</td><td>(不更新)</td><td>(不更新)</td></tr> <tr><td>n=7</td><td>(不更新)</td><td>(不更新)</td></tr> <tr><td>n=8</td><td>(不更新)</td><td>(不更新)</td></tr> </tbody> </table> <div style="text-align: center;"> <p>偏置 (Offset)</p> <p>读取数据数 (Points)</p> </div> <table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th colspan="2"><展开区域></th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td></tr> <tr><td>第1点的1周期当前值</td><td>第1点的行程</td></tr> <tr><td>第2点的1周期当前值</td><td>第2点的行程</td></tr> <tr><td>第3点的1周期当前值</td><td>第3点的行程</td></tr> <tr><td>第4点的1周期当前值</td><td>第4点的行程</td></tr> <tr><td>第5点的1周期当前值</td><td>第5点的行程</td></tr> <tr><td>⋮</td><td>⋮</td></tr> <tr><td>第255点的1周期当前值</td><td>第255点的行程</td></tr> </tbody> </table> <p style="text-align: center;">↑ 以凸轮分辨率点数对凸轮数据进行等分的数据</p> </div>		CamData[n][1]	CamData[n][2]	n=1	第2点的1周期当前值	第2点的行程	n=2	第3点的1周期当前值	第3点的行程	n=3	第4点的1周期当前值	第4点的行程	n=4	(不更新)	(不更新)	n=5	(不更新)	(不更新)	n=6	(不更新)	(不更新)	n=7	(不更新)	(不更新)	n=8	(不更新)	(不更新)	<展开区域>		0	0	第1点的1周期当前值	第1点的行程	第2点的1周期当前值	第2点的行程	第3点的1周期当前值	第3点的行程	第4点的1周期当前值	第4点的行程	第5点的1周期当前值	第5点的行程	⋮	⋮	第255点的1周期当前值	第255点的行程
	CamData[n][1]	CamData[n][2]																																												
n=1	第2点的1周期当前值	第2点的行程																																												
n=2	第3点的1周期当前值	第3点的行程																																												
n=3	第4点的1周期当前值	第4点的行程																																												
n=4	(不更新)	(不更新)																																												
n=5	(不更新)	(不更新)																																												
n=6	(不更新)	(不更新)																																												
n=7	(不更新)	(不更新)																																												
n=8	(不更新)	(不更新)																																												
<展开区域>																																														
0	0																																													
第1点的1周期当前值	第1点的行程																																													
第2点的1周期当前值	第2点的行程																																													
第3点的1周期当前值	第3点的行程																																													
第4点的1周期当前值	第4点的行程																																													
第5点的1周期当前值	第5点的行程																																													
⋮	⋮																																													
第255点的1周期当前值	第255点的行程																																													
1: 文件	<p>偏置(Offset)及读取数据数(Points)的设置将被忽略。 读取数据1(Data1)中存储的数据将变为运算配置文件格式为“旋转刀具”的PROFILE_ROTARY_CUTTER型的数据。无读取数据2(Data2)中存储的数据。</p>																																													

• 运算配置文件格式为“多轴定位数据”的情况下

读取目标(Target)	内容																																																																																																										
0: 展开区域	<p>应通过偏置(Offset)指定读取定位数据的区间的起始, 通过读取数据数(Points)指定区间数(NumberOfSections)。</p>																																																																																																										
2: 展开区域(运动服务处理)	<p>在Data1 (TARGET_REF型)中指定的PROFILE_POSITIONING_DATA型的变量中读取数据。 在Data2 (TARGET_REF型)中指定PROFILE_POSITIONING_DATA_ELEMENT型数组的变量的情况下, 在指定的变量中读取定位数据。 <例> 在偏置(Offset)中设置了“2”, 在读取数据数(Points)中设置了“3”时的读取数据2(Data2)的读取结果 (将读取数据2(Data2)中设置的变量设置为数据类型“PROFILE_POSITIONING_DATA_ELEMENT[1..8]”的标签“PositioningData2”的情况下)</p> <div style="text-align: center;"> <p><变量(标签/软元件)> PROFILE_POSITIONING_DATA_ELEMENT型的标签 (PositioningData2[1..8])</p> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>n=1</td><td>第3点的DataNo (3)</td><td>第3点的 OperationPattern</td><td>...</td></tr> <tr><td>n=2</td><td>第4点的DataNo (4)</td><td>第4点的 OperationPattern</td><td>...</td></tr> <tr><td>n=3</td><td>第5点的DataNo (5)</td><td>第5点的 OperationPattern</td><td>...</td></tr> <tr><td>n=4</td><td>(不更新)</td><td>(不更新)</td><td>...</td></tr> <tr><td>n=5</td><td>(不更新)</td><td>(不更新)</td><td>...</td></tr> <tr><td>n=6</td><td>(不更新)</td><td>(不更新)</td><td>...</td></tr> <tr><td>n=7</td><td>(不更新)</td><td>(不更新)</td><td>...</td></tr> <tr><td>n=8</td><td>(不更新)</td><td>(不更新)</td><td>...</td></tr> </table> <p style="margin-left: 100px;">偏置(Offset) →</p> <p style="margin-left: 100px;">← 读取数据数(Points)</p> <table border="1" style="display: inline-table;"> <caption><展开区域></caption> <tr><td>第1点的DataNo (1)</td><td>第1点的 OperationPattern</td><td>...</td></tr> <tr><td>第2点的DataNo (2)</td><td>第2点的 OperationPattern</td><td>...</td></tr> <tr><td>第3点的DataNo (3)</td><td>第3点的 OperationPattern</td><td>...</td></tr> <tr><td>第4点的DataNo (4)</td><td>第4点的 OperationPattern</td><td>...</td></tr> <tr><td>第5点的DataNo (5)</td><td>第5点的 OperationPattern</td><td>...</td></tr> <tr><td>第6点的DataNo (6)</td><td>第6点的 OperationPattern</td><td>...</td></tr> <tr><td>...</td><td>...</td><td>...</td></tr> </table> </div> <p>读取区间中, 展开或写入时存在未设置的定位数据的情况下, 在相应DataNo的区间数据中存储初始值。(DataNo变为0。) <例> 上述示例中, 在展开或写入中未设置展开区域的第4点(定位数据No. 4)的情况下</p> <div style="text-align: center;"> <p><变量(标签/软元件)> PROFILE_POSITIONING_DATA_ELEMENT型的标签 (PositioningData2[1..8])</p> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>n=1</td><td>第3点的DataNo (3)</td><td>第3点的 OperationPattern</td><td>...</td></tr> <tr style="background-color: #cccccc;"><td>n=2</td><td>0</td><td>0</td><td>...</td></tr> <tr><td>n=3</td><td>第5点的DataNo (5)</td><td>第5点的 OperationPattern</td><td>...</td></tr> <tr><td>n=4</td><td>(不更新)</td><td>(不更新)</td><td>...</td></tr> <tr><td>n=5</td><td>(不更新)</td><td>(不更新)</td><td>...</td></tr> <tr><td>n=6</td><td>(不更新)</td><td>(不更新)</td><td>...</td></tr> <tr><td>n=7</td><td>(不更新)</td><td>(不更新)</td><td>...</td></tr> <tr><td>n=8</td><td>(不更新)</td><td>(不更新)</td><td>...</td></tr> </table> <p style="margin-left: 100px;">偏置(Offset) →</p> <p style="margin-left: 100px;">← 读取数据数(Points)</p> <table border="1" style="display: inline-table;"> <caption><展开区域></caption> <tr><td>第1点的DataNo (1)</td><td>第1点的 OperationPattern</td><td>...</td></tr> <tr><td>第2点的DataNo (2)</td><td>第2点的 OperationPattern</td><td>...</td></tr> <tr><td>第3点的DataNo (3)</td><td>第3点的 OperationPattern</td><td>...</td></tr> <tr style="background-color: #cccccc;"><td>第4点的DataNo</td><td>第4点的 OperationPattern</td><td>...</td></tr> <tr><td>第5点的DataNo (5)</td><td>第5点的 OperationPattern</td><td>...</td></tr> <tr><td>第6点的DataNo (6)</td><td>第6点的 OperationPattern</td><td>...</td></tr> <tr><td>...</td><td>...</td><td>...</td></tr> </table> </div> <p>在Data2 (TARGET_REF型)中指定PROFILE_POSITIONING_DATA_EXTENDED型的变量的情况下 变量的构件ConditionSignal、SkipSignal及PositioningData中指定了WSTRING型的变量中, 将分别存储条件信号、跳过信号及定位数据。 条件信号、跳过信号读取PROFILE_SIGNAL_SELECT_ELEMENT型数组的数据。定位数据读取PROFILE_POSITIONING_DATA_ELEMENT型数组的数据。(与在Data2 (TARGET_REF型)中指定了PROFILE_POSITIONING_DATA_ELEMENT型的数组的情况相同。)</p>	n=1	第3点的DataNo (3)	第3点的 OperationPattern	...	n=2	第4点的DataNo (4)	第4点的 OperationPattern	...	n=3	第5点的DataNo (5)	第5点的 OperationPattern	...	n=4	(不更新)	(不更新)	...	n=5	(不更新)	(不更新)	...	n=6	(不更新)	(不更新)	...	n=7	(不更新)	(不更新)	...	n=8	(不更新)	(不更新)	...	第1点的DataNo (1)	第1点的 OperationPattern	...	第2点的DataNo (2)	第2点的 OperationPattern	...	第3点的DataNo (3)	第3点的 OperationPattern	...	第4点的DataNo (4)	第4点的 OperationPattern	...	第5点的DataNo (5)	第5点的 OperationPattern	...	第6点的DataNo (6)	第6点的 OperationPattern	n=1	第3点的DataNo (3)	第3点的 OperationPattern	...	n=2	0	0	...	n=3	第5点的DataNo (5)	第5点的 OperationPattern	...	n=4	(不更新)	(不更新)	...	n=5	(不更新)	(不更新)	...	n=6	(不更新)	(不更新)	...	n=7	(不更新)	(不更新)	...	n=8	(不更新)	(不更新)	...	第1点的DataNo (1)	第1点的 OperationPattern	...	第2点的DataNo (2)	第2点的 OperationPattern	...	第3点的DataNo (3)	第3点的 OperationPattern	...	第4点的DataNo	第4点的 OperationPattern	...	第5点的DataNo (5)	第5点的 OperationPattern	...	第6点的DataNo (6)	第6点的 OperationPattern
n=1	第3点的DataNo (3)	第3点的 OperationPattern	...																																																																																																								
n=2	第4点的DataNo (4)	第4点的 OperationPattern	...																																																																																																								
n=3	第5点的DataNo (5)	第5点的 OperationPattern	...																																																																																																								
n=4	(不更新)	(不更新)	...																																																																																																								
n=5	(不更新)	(不更新)	...																																																																																																								
n=6	(不更新)	(不更新)	...																																																																																																								
n=7	(不更新)	(不更新)	...																																																																																																								
n=8	(不更新)	(不更新)	...																																																																																																								
第1点的DataNo (1)	第1点的 OperationPattern	...																																																																																																									
第2点的DataNo (2)	第2点的 OperationPattern	...																																																																																																									
第3点的DataNo (3)	第3点的 OperationPattern	...																																																																																																									
第4点的DataNo (4)	第4点的 OperationPattern	...																																																																																																									
第5点的DataNo (5)	第5点的 OperationPattern	...																																																																																																									
第6点的DataNo (6)	第6点的 OperationPattern	...																																																																																																									
...																																																																																																									
n=1	第3点的DataNo (3)	第3点的 OperationPattern	...																																																																																																								
n=2	0	0	...																																																																																																								
n=3	第5点的DataNo (5)	第5点的 OperationPattern	...																																																																																																								
n=4	(不更新)	(不更新)	...																																																																																																								
n=5	(不更新)	(不更新)	...																																																																																																								
n=6	(不更新)	(不更新)	...																																																																																																								
n=7	(不更新)	(不更新)	...																																																																																																								
n=8	(不更新)	(不更新)	...																																																																																																								
第1点的DataNo (1)	第1点的 OperationPattern	...																																																																																																									
第2点的DataNo (2)	第2点的 OperationPattern	...																																																																																																									
第3点的DataNo (3)	第3点的 OperationPattern	...																																																																																																									
第4点的DataNo	第4点的 OperationPattern	...																																																																																																									
第5点的DataNo (5)	第5点的 OperationPattern	...																																																																																																									
第6点的DataNo (6)	第6点的 OperationPattern	...																																																																																																									
...																																																																																																									

读取目标(Target)	内容																																								
1: 文件	<p>应在偏置(Offset)及读取数据数(Points)中设置0, 并读取所有元素。 (设置了0以外的情况下, 将发生偏置・读取/写入数据数不正确(错误代码: 1BADH)。) 在Data1(TARGET_REF型)中指定的PROFILE_POSITIONING_DATA型的变量中读取数据。 在Data2(TARGET_REF型)中指定PROFILE_POSITIONING_DATA_ELEMENT型数组的变量的情况下, 在指定的变量中读取定位数据。 在文件中空出DataNo设置了定位数据的情况下, 在空出的DataNo中的区间数据中存储初始值。(DataNo变为0。) <例> 在偏置(Offset)中设置了“0”, 在读取数据数(Points)中设置了“0”时的读取数据2(Data2)的读取结果 (将读取数据2(Data2)中设置的变量设置为数据类型“PROFILE_POSITIONING_DATA_ELEMENT[1..100]”的标签“PositioningData2”的情况下)</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p><变量(标签/软元件)> PROFILE_POSITIONING_DATA_ELEMENT型的标签 (PositioningData2[1..100])</p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>n=1</td><td>0</td><td>0</td><td>...</td></tr> <tr><td>n=2</td><td>第1点的DataNo(2)</td><td>第1点的OperationPattern</td><td>...</td></tr> <tr><td>n=3</td><td>第2点的DataNo(3)</td><td>第2点的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>第3点的DataNo(5)</td><td>第3点的OperationPattern</td><td>...</td></tr> <tr><td>...</td><td>...</td><td>...</td><td>...</td></tr> <tr><td>n=100</td><td>第4点的DataNo(100)</td><td>第4点的OperationPattern</td><td>...</td></tr> </table> </div> <div style="text-align: center;"> <p><文件></p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>第1点的DataNo(2)</td><td>第1点的OperationPattern</td><td>...</td></tr> <tr><td>第2点的DataNo(3)</td><td>第2点的OperationPattern</td><td>...</td></tr> <tr><td>第3点的DataNo(5)</td><td>第3点的OperationPattern</td><td>...</td></tr> <tr><td>第4点的DataNo(100)</td><td>第4点的OperationPattern</td><td>...</td></tr> </table> </div> </div> <p>在Data2(TARGET_REF型)中指定PROFILE_POSITIONING_DATA_EXTENDED型的变量的情况下 变量的构件ConditionSignal、SkipSignal及PositioningData中指定了WSTRING型的变量中, 将分别存储条件信号、跳过信号及定位数据。 条件信号、跳过信号读取PROFILE_SIGNAL_SELECT_ELEMENT型数组的数据。 定位数据读取PROFILE_POSITIONING_DATA_ELEMENT型数组的数据。(与在Data2(TARGET_REF型)中指定了PROFILE_POSITIONING_DATA_ELEMENT型数组的变量的情况相同。)</p>	n=1	0	0	...	n=2	第1点的DataNo(2)	第1点的OperationPattern	...	n=3	第2点的DataNo(3)	第2点的OperationPattern	...	n=4	0	0	...	n=5	第3点的DataNo(5)	第3点的OperationPattern	n=100	第4点的DataNo(100)	第4点的OperationPattern	...	第1点的DataNo(2)	第1点的OperationPattern	...	第2点的DataNo(3)	第2点的OperationPattern	...	第3点的DataNo(5)	第3点的OperationPattern	...	第4点的DataNo(100)	第4点的OperationPattern	...
n=1	0	0	...																																						
n=2	第1点的DataNo(2)	第1点的OperationPattern	...																																						
n=3	第2点的DataNo(3)	第2点的OperationPattern	...																																						
n=4	0	0	...																																						
n=5	第3点的DataNo(5)	第3点的OperationPattern	...																																						
...																																						
n=100	第4点的DataNo(100)	第4点的OperationPattern	...																																						
第1点的DataNo(2)	第1点的OperationPattern	...																																							
第2点的DataNo(3)	第2点的OperationPattern	...																																							
第3点的DataNo(5)	第3点的OperationPattern	...																																							
第4点的DataNo(100)	第4点的OperationPattern	...																																							

• 运算配置文件格式为“数字凸轮开关数据”的情况下

读取目标(Target)	内容																																																																								
0: 展开区域	<p>通过偏置(Offset)设置要读取的区间的起始, 通过读取数据数(Points)设置区间数(NumberOfSections)。 读取数据1(Data1)中存储的数据将变为PROFILE_CAMSWITCH_DATA型的数据。</p>																																																																								
2: 展开区域(运动服务处理)	<p>对于读取数据2(Data2)中存储的数据, 将变为将指定的开关设置的数据(FirstOnPosition、LastOnPosition、Direction、CamSwitchMode、Duration)通过PROFILE_CAMSWITCH_ELEMENT型的数组进行了定义的数据。 <例> 在偏置(Offset)中设置了“2”, 在读取数据数(Points)中设置了“3”时的读取数据2(Data2)的读取结果 (将读取数据2(Data2)中设置的变量设置为数据类型“PROFILE_CAMSWITCH_ELEMENT[1..8]”的标签“DigitalCamSwitch Data2”的情况下)</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p><变量(标签/软元件)> PROFILE_CAMSWITCH_ELEMENT型的标签 (PositioningData2[1..8])</p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>n=1</td><td>第3点的FirstOnPosition</td><td>第3点的LastOnPosition</td><td>...</td><td>第3点的Duration</td></tr> <tr><td>n=2</td><td>第4点的FirstOnPosition</td><td>第4点的LastOnPosition</td><td>...</td><td>第4点的Duration</td></tr> <tr><td>n=3</td><td>第5点的FirstOnPosition</td><td>第5点的LastOnPosition</td><td>...</td><td>第5点的Duration</td></tr> <tr><td>n=4</td><td>(不更新)</td><td>(不更新)</td><td>...</td><td>(不更新)</td></tr> <tr><td>n=5</td><td>(不更新)</td><td>(不更新)</td><td>...</td><td>(不更新)</td></tr> <tr><td>n=6</td><td>(不更新)</td><td>(不更新)</td><td>...</td><td>(不更新)</td></tr> <tr><td>n=7</td><td>(不更新)</td><td>(不更新)</td><td>...</td><td>(不更新)</td></tr> <tr><td>n=8</td><td>(不更新)</td><td>(不更新)</td><td>...</td><td>(不更新)</td></tr> </table> </div> <div style="text-align: center;"> <p><展开区域></p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>第1点的FirstOnPosition</td><td>第1点的LastOnPosition</td><td>...</td><td>第1点的Duration</td></tr> <tr><td>第2点的FirstOnPosition</td><td>第2点的LastOnPosition</td><td>...</td><td>第2点的Duration</td></tr> <tr><td>第3点的FirstOnPosition</td><td>第3点的LastOnPosition</td><td>...</td><td>第3点的Duration</td></tr> <tr><td>第4点的FirstOnPosition</td><td>第4点的LastOnPosition</td><td>...</td><td>第4点的Duration</td></tr> <tr><td>第5点的FirstOnPosition</td><td>第5点的LastOnPosition</td><td>...</td><td>第5点的Duration</td></tr> <tr><td>第6点的FirstOnPosition</td><td>第6点的LastOnPosition</td><td>...</td><td>第6点的Duration</td></tr> <tr><td>...</td><td>...</td><td>...</td><td>...</td></tr> <tr><td>第10000点的FirstOnPosition</td><td>第10000点的LastOnPosition</td><td>...</td><td>第10000点的Duration</td></tr> </table> </div> </div> <p>偏置(Offset) → (指向展开区域表格的起始行) 读取数据数(Points) → (指向展开区域表格的列数)</p>	n=1	第3点的FirstOnPosition	第3点的LastOnPosition	...	第3点的Duration	n=2	第4点的FirstOnPosition	第4点的LastOnPosition	...	第4点的Duration	n=3	第5点的FirstOnPosition	第5点的LastOnPosition	...	第5点的Duration	n=4	(不更新)	(不更新)	...	(不更新)	n=5	(不更新)	(不更新)	...	(不更新)	n=6	(不更新)	(不更新)	...	(不更新)	n=7	(不更新)	(不更新)	...	(不更新)	n=8	(不更新)	(不更新)	...	(不更新)	第1点的FirstOnPosition	第1点的LastOnPosition	...	第1点的Duration	第2点的FirstOnPosition	第2点的LastOnPosition	...	第2点的Duration	第3点的FirstOnPosition	第3点的LastOnPosition	...	第3点的Duration	第4点的FirstOnPosition	第4点的LastOnPosition	...	第4点的Duration	第5点的FirstOnPosition	第5点的LastOnPosition	...	第5点的Duration	第6点的FirstOnPosition	第6点的LastOnPosition	...	第6点的Duration	第10000点的FirstOnPosition	第10000点的LastOnPosition	...	第10000点的Duration
n=1	第3点的FirstOnPosition	第3点的LastOnPosition	...	第3点的Duration																																																																					
n=2	第4点的FirstOnPosition	第4点的LastOnPosition	...	第4点的Duration																																																																					
n=3	第5点的FirstOnPosition	第5点的LastOnPosition	...	第5点的Duration																																																																					
n=4	(不更新)	(不更新)	...	(不更新)																																																																					
n=5	(不更新)	(不更新)	...	(不更新)																																																																					
n=6	(不更新)	(不更新)	...	(不更新)																																																																					
n=7	(不更新)	(不更新)	...	(不更新)																																																																					
n=8	(不更新)	(不更新)	...	(不更新)																																																																					
第1点的FirstOnPosition	第1点的LastOnPosition	...	第1点的Duration																																																																						
第2点的FirstOnPosition	第2点的LastOnPosition	...	第2点的Duration																																																																						
第3点的FirstOnPosition	第3点的LastOnPosition	...	第3点的Duration																																																																						
第4点的FirstOnPosition	第4点的LastOnPosition	...	第4点的Duration																																																																						
第5点的FirstOnPosition	第5点的LastOnPosition	...	第5点的Duration																																																																						
第6点的FirstOnPosition	第6点的LastOnPosition	...	第6点的Duration																																																																						
...																																																																						
第10000点的FirstOnPosition	第10000点的LastOnPosition	...	第10000点的Duration																																																																						
1: 文件	<p>应指定偏置(Offset)及读取数据数(Points), 并读取所有元素。 (设置了0以外的情况下, 将发生偏置・读取/写入数据数不正确(错误代码: 1BADH)。) 读取数据1(Data1)中存储的数据将变为PROFILE_CAMSWITCH_DATA型的数据。 对于读取数据2(Data2)中存储的数据, 将变为将指定的开关设置的数据(FirstOnPosition、LastOnPosition、Direction、CamSwitchMode、Duration)通过PROFILE_CAMSWITCH_ELEMENT型的数组进行了定义的数据。</p>																																																																								

■读取数据1(Data1)/读取数据2(Data2)

在TARGET_REF结构体中设置运算配置文件的读取数据。可使用的数据类型为[VAR]、[DEV]。

关于TARGET_REF结构体的详细内容，请参阅下述章节。

☞ 1428页 TARGET_REF(输入信号)

可使用的数据类型为各运算配置文件格式的读取用数据结构体的类型。

- 读取数据1(Data1)

读取数据1(Data1)中设置的运算配置文件格式及数据类型如下所示。

运算配置文件格式	数据类型	参阅
凸轮数据	PROFILE_CAM_DATA	1433页 PROFILE_CAM_DATA[P]
旋转刀具用凸轮数据	PROFILE_ROTARY_CUTTER	1435页 PROFILE_ROTARY_CUTTER[P]

例

使用软元件、标签读取的情况下

- 读取PROFILE_CAM_DATA型的标签(CamData1)的情况下

```
Data1.Target := "[VAR]CamData1"
```

• 读取数据2(Data2)

读取数据2(Data2)中设置的运算配置文件格式及数据类型如下所示。根据读取目标(Target)及成为对象的凸轮数据的插补方法指定(Interpolate), 设置的数据类型的结构体有所不同。

此外, 读取整个运算配置文件的情况下, 指定的数组的元素数小于(分辨率 +1/坐标数)的情况下, 将发生读取/写入数据数不一致(错误代码: 1BACH)。

○: 支持, ×: 不支持

运算配置文件格式	读取目标(Target)	数据类型	插补方法指定(Interpolate)			参阅
			0: 直线插补	1: 各区间中指定	2: 样条插补	
凸轮数据	2: 展开区域(运动服务处理)	LREAL[]*1	○	○*3	○*3	—
	1: 文件	LREAL[]*1	○	×	×	—
		PROFILE_CAM_ELEMENT[]*2	×	○	○*4	1434页 PROFILE_CAM_ELEMENT[P]
旋转刀具用凸轮数据	无需设置。 即使设置也将被忽略。					—

*1 LREAL型的2维数组将按下述方式设置。

LREAL[m..n, o..p]

项目	内容
元素数(n-m+1)	设置为数组元素为要读取的点数及以上。 对整个运算配置文件进行读取的情况下(将偏置(Offset)、读取数据数(Points)设置为“0”的情况下), 应进行设置以确保其为对象的凸轮数据的分辨率+1/坐标数及以上。 但是, 大于读取数据1(Data1)中读取的分辨率+1/坐标数的区域不被更新。
次元数(p-o+1)	读取各点的1周期当前位置与行程。 设置时, 应确保数组元素为“2”。 将数组元素设置为“3”及以上的情况下, 可以读取, 但是由于数据已被删除可能无法获得预期的结果, 因此设置时必须确保其为“2”。

*2 PROFILE_CAM_ELEMENT的数组将按下述方式设置。

PROFILE_CAM_ELEMENT[m..n]

项目	内容
元素数(n-m+1)	设置为数组元素为要读取的点数及以上。 对整个运算配置文件进行读取的情况下(将偏置(Offset)、读取数据数(Points)设置为“0”的情况下), 应进行设置以确保其为对象的凸轮数据的区间数(NumberOfSections)及以上。 但是, 大于读取数据1(Data1)中读取的区间数(NumberOfSections)的区域不被更新。

*3 作为展开区域中展开的行程数据(1周期当前值与行程的成对)进行读取。第1点的数据的X为开始点, Y为设置为初始行程量的值。

第2点及以后的数据将变为以分辨率分割的行程数据。因此, 展开区域中的读取点数将变为“分辨率+1”。

*4 仅使用结束点、行程。

例

通过插补方法指定(Interpolate)的LREAL型的情况下

插补方法指定(Interpolate)	设置示例
0: 直线插补	在TARGET_REF结构体的对象(Target)中设置坐标数“100”的二维标签的示例如下所示。 Data2.Target := "[VAR]CamData2" * 对于CamData2, 在全局标签的标签名中声明“CamData2”, 在数据类型中声明“LREAL[1..100, 1..2]”的标签, 在对象(Target)中指定字符串。要读取的数据如下所示。 · CamData2[n, 1]: 第n点的1周期当前位置 · CamData2[n, 2]: 第n点的行程
1: 各区间中指定	在TARGET_REF结构体的对象(Target)中设置分辨率“100”的二维标签的示例如下所示。 TARGET_REF.Target := "[VAR]CamData2" * 对于CamData2, 在全局标签的标签名中声明“CamData2”, 在数据类型中声明“LREAL[1..101, 1..2]”的标签, 在对象(Target)中指定字符串。要读取的数据如下所示。 · CamData2[1, 1]: 设置为开始点的值 · CamData2[1, 2]: 初始行程量的设定值 · CamData2[n+1, 1]: 第n点的1周期当前位置 · CamData2[n+1, 2]: 第n点的行程
2: 样条插补	

例

PROFILE_CAM_ELEMENT型的情况下

设置示例


在TARGET_REF结构体的对象(Target)中设置区间数“10”的二维标签的示例如下所示。

```
Data2.Target := "[VAR]CamData2"
```

* 对于CamData2, 在全局标签的标签名中声明“CamData2”, 在数据类型中声明“PROFILE_CAM_ELEMENT[1..10]”的标签, 在对象(Target)中指定字符串。要读取的数据如下所示。

· CamData2[n]: 第n点的区间数据 (PROFILE_CAM_ELEMENT型)

要点

- 设置的读取用数据结构体的类型与各运算配置文件格式中请求的类型不一致的情况下, 将变为读取/写入数据类型不正确(错误代码: 1BAAH)。
- 关于读取数据1(Data1)/读取数据2(Data2)中设置的运算配置文件的凸轮数据的详细内容, 请参阅下述手册
所使用的控制器的用户手册

■读取目标(Target)

设置运算配置文件的读取目标。

设定值	内容
0: 展开区域	<ul style="list-style-type: none">• 参照PROFILE_DATA结构体的配置文件ID(ID)的值, 对PROFILE_ID结构体的配置文件ID编号(Number)表示的展开区域的运算配置文件进行读取。• 将按照执行的程序的周期进行读取处理。• 运算配置文件不存在的情况下(未展开的情况下), 将变为无运算配置文件(错误代码: 1B8EH)。
1: 文件	<ul style="list-style-type: none">• 参照PROFILE_DATA结构体的运算配置文件存储位置(Location)的值, 对文件名称及路径表示的文件的运算配置文件进行读取。• 将按照运动服务处理的周期进行读取处理。• 运算配置文件不存在的情况下, 将变为无运算配置文件(错误代码: 1B8EH)。
2: 展开区域(运动服务处理)	<ul style="list-style-type: none">• 参照PROFILE_DATA结构体的配置文件ID(ID)的值, 对PROFILE_ID结构体的配置文件ID编号(Number)表示的展开区域的运算配置文件进行读取。• 将按照运动服务处理的周期进行读取处理。• 运算配置文件不存在的情况下(未展开的情况下), 将变为无运算配置文件(错误代码: 1B8EH)。

要点

- 读取目标(Target)为“1: 文件”、“2: 展开区域(运动服务处理)”的情况下, 由于在运动服务处理中执行, 因此到处理完成为止可能需要花费较长时间。应根据需要将Done(执行完成)用于互锁。
- Target = “0: 展开区域”的情况下, 由于是在调用了FB的任务内处理, 因此立即完成。写入的数据点数较多时, 程序的执行时间将会延长, 因此应根据需要分多次执行写入, 或者使用Target = “2: 展开区域(运动服务处理)”。

注意事项

一次可读取的数据点数有限制。1次的FB执行中读取未完成时, 应分为多次执行读取。

程序示例

从文件中读取“LREAL型(2维数组)的格式”的凸轮数据1(ProfileData0001)与“PROFILE_CAM_ELEMENT型的格式”的凸轮数据2(ProfileData0002)的程序示例如下所示。

■运算配置文件

配置文件ID	标签名	数据类型	注释
1	ProfileData0001	MC_CAM_REF	凸轮数据1
2	ProfileData0002	MC_CAM_REF	凸轮数据2

• 运算配置文件(凸轮数据)的设置

项目	ProfileData0001	ProfileData0002
插补方法指定	直线插补	各区间中指定
分辨率	—	256
1周期长设置	单位	degree
	1周期长	360.00000
行程量	单位	%
	行程量	100.0
凸轮1周期时间	—	1.000[s]
初始行程值	—	0.0000000
凸轮1周期最小值	0.00000	—
凸轮1周期最大值	360.00000	—

• ProfileData0001

区间No.	输入值[degree]	输出值[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

区间No.	开始点[degree]	结束点[degree]	行程[%]	凸轮曲线类型
1	0.00000	90.00000	60.0000000	单弦
2	90.00000	180.00000	100.0000000	单弦
3	180.00000	270.00000	30.0000000	单弦
4	270.00000	0.00000	0.0000000	单弦

■使用的标签

标签名	数据类型	注释
MCv_ReadProfileData_1	MCv_ReadProfileData	配置文件读取FB 1
ReadData1_1	TARGET_REF	读取数据1 配置文件1
ReadData2_1	TARGET_REF	读取数据2 配置文件1
ReadData1_Lreal	PROFILE_CAM_DATA	配置文件数据 凸轮数据1
ReadData2_Lreal	双精度实数(0..4,0..1)	输入值/输出值数据 LREAL型
bReadProfile1	位	配置文件读取指令1
bDone1	位	完成1
bBusy1	位	执行中1
bError1	位	错误1
uwErrorID1	字[无符号]/位串[16位]	错误代码1
Cam_ProfileData0001*1	PROFILE_DATA	配置文件数据0001
MCv_ReadProfileData_2	MCv_ReadProfileData	配置文件读取FB 2
ReadData1_2	TARGET_REF	读取数据1 配置文件2
ReadData2_2	TARGET_REF	读取数据2 配置文件2
ReadData1_Element	PROFILE_CAM_DATA	配置文件数据 凸轮数据2
ReadData2_Element	PROFILE_CAM_ELEMENT(0..3)	输入值/输出值数据 PROFILE_CAM_ELEMENT型
bReadProfile2	位	配置文件读取指令2
bDone2	位	完成2
bBusy2	位	执行中2
bError2	位	错误2
uwErrorID2	字[无符号]/位串[16位]	错误代码2
Cam_ProfileData0002*1	PROFILE_DATA	配置文件数据0002

*1 仅FBD/LD的程序使用(为了在FBD/LD的程序中使用运算配置文件需要此项)

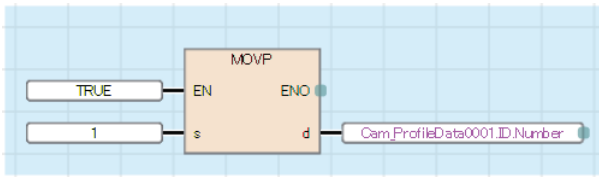
■使用的全局标签

标签名	数据类型	注释
G_ProfileCamData*1	PROFILE_CAM_DATA	配置文件凸轮数据
G_ProfileCamELEMENT*1	PROFILE_CAM_ELEMENT	配置文件凸轮元件

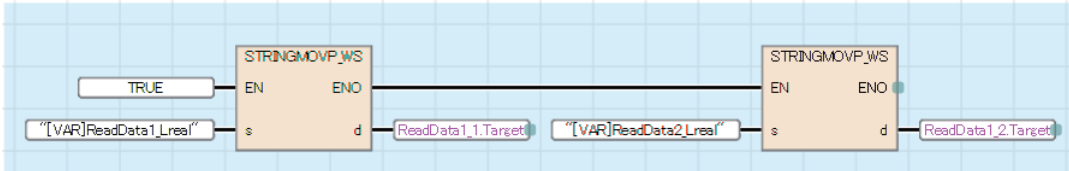
*1 仅FBD/LD的程序使用(为了在FBD/LD的程序中使用运算配置文件需要此项)

■FBD/LD的程序

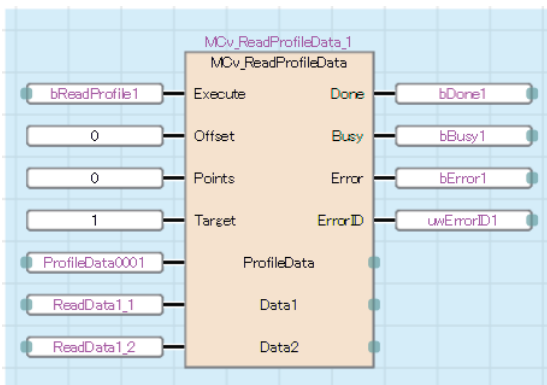
- 配置文件数据0001



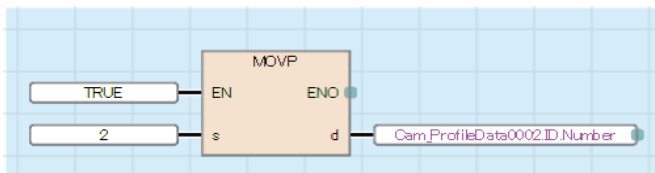
- 读取数据1



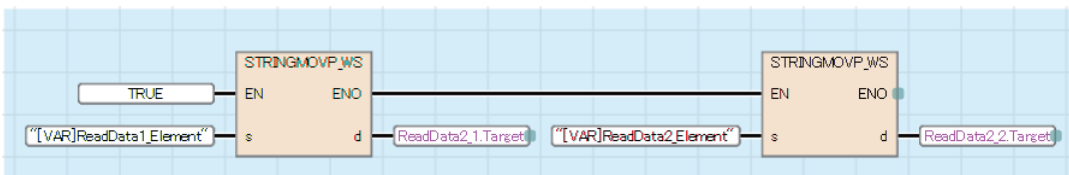
- 配置文件读取1(以LREAL型(2维数组)的格式读取凸轮数据)



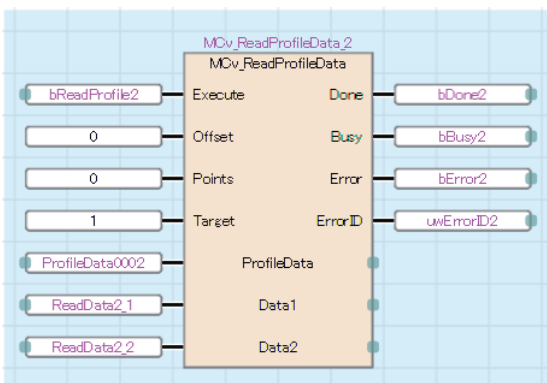
- 配置文件数据0002



- 读取数据2



- 配置文件读取2(以PROFILE_CAM_ELEMENT型的格式读取凸轮数据)



■ST的程序

```
//-----读取数据1-----
ReadData1_1.Target := "[VAR]ReadData1_Lreal";
ReadData2_1.Target := "[VAR]ReadData2_Lreal";

//-----配置文件读取1-----
//-----以LREAL型(2维数组)的格式读取凸轮数据-----
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
);

//-----读取数据2-----
ReadData1_2.Target := "[VAR]ReadData1_Element";
ReadData2_2.Target := "[VAR]ReadData2_Element";

//-----配置文件读取2-----
//-----以PROFILE_CAM_ELEMENT型的格式读取凸轮数据-----
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
);
```

47.2 配置文件写入

MCv_WriteProfileData

将指定的运算配置文件写入到展开区域或文件中。

梯形图	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>

设置数据

■输入输出变量


输入输出变量	名称	数据类型	输入获取	有效范围	初始值	内容
ProfileData	配置文件数据	PROFILE_DATA	启动时	—	不能省略	设置要写入的运算配置文件。 在PROFILE_DATA结构体中设置运算配置文件。关于PROFILE_DATA结构体，请参阅下述章节。 ☞ 1431页 PROFILE_DATA 对于定义了写入用数据结构体的运算配置文件格式，可以进行写入操作。
Data1	写入数据1	TARGET_REF	启动时	—	不能省略	根据要写入的运算配置文件格式设置写入用数据结构体。关于详细内容，请参阅下述章节。 ☞ 1883页 写入数据1(Data1)/写入数据2(Data2)
Data2	写入数据2	TARGET_REF	启动时	—	不能省略	根据要写入的运算配置文件格式设置写入用数据结构体。关于详细内容，请参阅下述章节。 ☞ 1883页 写入数据1(Data1)/写入数据2(Data2)

■输入变量

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	启动	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MCv_WriteProfileData(配置文件写入)。
Offset	偏置	DWORD(UDINT)	启动时	0~分辨率(2~65535)	0	设置从运算配置文件起始开始的偏置。 偏置(Offset)超出运算配置文件的范围时，将变为超出偏置范围(错误代码: 1B92H)，不进行写入。 在偏置(Offset)中设置“0”以外的值的情况下，写入数据数(Points)中应设置“0”以外的值。(在写入数据数(Points)中设置了“0”的情况下，将变为偏置·读取/写入数据数不正确(错误代码: 1BADH)。) * 根据偏置(Offset)与写入数据数(Points)的设定值，写入的动作有所不同。关于详细内容，请参阅下述章节。 ☞ 1880页 偏置(Offset)与写入数据数(Points)的设置
Points	写入数据数	DWORD(UDINT)	启动时	0~4294967295	0	设置要写入的数据点数。 写入数据数(Points)超出运算配置文件的元素数的情况下，将在运算配置文件的元素数的范围内进行写入。 运算配置文件写入中，写入目标(Target)为“1: 文件”的情况下，应设置“0”。(设置了“0”以外的情况下，将变为偏置·读取/写入数据数不正确(错误代码: 1BADH)。)

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Target	写入目标	WORD(UINT)	启动时	0、1、2	0	<p>设置运算配置文件的写入目标。</p> <ul style="list-style-type: none"> • 0: 展开区域 • 1: 文件 • 2: 展开区域(运动服务处理) <p>* 在写入目标(Target)中设置了“1: 文件”的情况下, 偏置(Offset)及写入数据数(Points)中, 应设置“0”。(设置了“0”以外的情况下, 将变为偏置·读取/写入数据数不正确(错误代码: 1BADH)。)</p> <p>关于详细内容, 请参阅下述章节。  1885页 写入目标(Target)</p>
ExecutionMode	启动模式	INT (MC_EXECUTION_MODE)	启动时	0、1、3	0	<p>设置MCv_WriteProfileData(配置文件写入)的执行时机。</p> <ul style="list-style-type: none"> • 0: 立即执行(mcImmediately) • 1: 等待完成后执行(mcQueued) • 3: 推测执行(mcSpeculatively) <p>关于详细内容, 请参阅下述章节。  1886页 启动模式(ExecutionMode)</p>

■输出变量

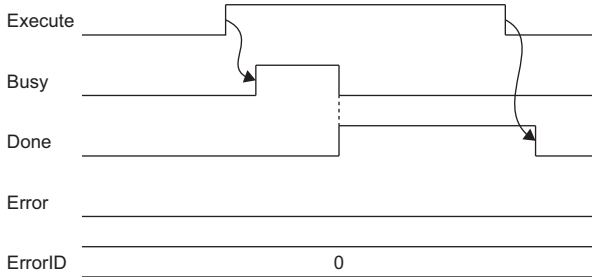
输出变量	名称	数据类型	初始值	内容
Done	完成	BOOL	FALSE	<p>控制完成时, 将变为TRUE。 动作完成时根据启动(Execute)的状态将变为如下所示。</p> <ul style="list-style-type: none"> ■启动(Execute)为TRUE的情况下 在将启动(Execute)置为FALSE之前将保持为TRUE不变。 ■启动(Execute)为FALSE的情况下 仅1周期为TRUE。
Busy	执行中	BOOL	FALSE	<p>执行了MCv_WriteProfileData(配置文件写入)时, 将变为TRUE。</p>
Error	错误	BOOL	FALSE	<p>发生了异常时, 将变为TRUE。</p>
ErrorID	错误代码	WORD(UINT)	0	<p>发生了异常时, 将返回错误代码。 关于错误代码的详细内容, 请参阅下述手册。  所使用的控制器的用户手册</p>
ProfileID	配置文件ID	PROFILE_ID	0	<p>输出配置文件ID。</p>

功能

- 使用MCv_WriteProfileData(配置文件写入)进行运算配置文件的写入操作。
- 对于配置文件数据(ProfileData)及写入目标(Target)中设置的文件或展开区域的运算配置文件，从偏置(Offset)中设置的数据进行写入，写入的点数数据数(Points)中设置的点数。
- 写入整个运算配置文件时，在偏置(Offset)与写入数据数(Points)的两者中设置“0”。
- 写入的数据通过写入数据1(Data1)、写入数据2(Data2)中设置的变量进行设置。

■时序图

- 正常完成的情况下



- 异常完成的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 1375页 通过执行指令(Execute)类型的运动控制FB的基本动作

■偏置(Offset)与写入数据数(Points)的设置

根据偏置(Offset)与写入数据数(Points)中设置的值,将变为下述动作。

对于偏置(Offset)、写入数据数(Points)中设置的值的内容,根据运算配置文件格式而有所不同。

在运算配置文件格式中设置“凸轮数据”、“旋转刀具”的内容如下所示。

- 运算配置文件格式为“凸轮数据”的情况下

插补方法指定 (Interpolate)	内容																																													
0: 直线插补	<p>应通过偏置(Offset)指定写入对象的坐标的起始,通过写入数据数(Points)指定坐标数。</p> <p>对于写入数据1(Data1),设置插补方法指定(Interpolate)为“0: 直线插补”的PROFILE_CAM_DATA型的数据。对于写入数据2(Data2),设置将已设置的坐标数据(1周期当前值与行程的成对)通过LREAL型的二维数组进行了定义的数据。</p> <p><例></p> <p>在偏置(Offset)中设置了“2”,在写入数据数(Points)中设置了“3”时的写入数据2(Data2)的写入结果(将写入数据2(Data2)中设置的变量设置为数据类型“LREAL[1..8,1..2]”的标签“CamData2”的情况下)</p> <p style="text-align: center;"><变量(标签/软元件)> LREAL型的二维数组的标签 (CamData2[1..8,1..2])</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th></th> <th>CamData[n][1]</th> <th>CamData[n][2]</th> </tr> </thead> <tbody> <tr><td>n=1</td><td>第2点的输入值</td><td>第2点的输出值</td></tr> <tr><td>n=2</td><td>第3点的输入值</td><td>第3点的输出值</td></tr> <tr><td>n=3</td><td>第4点的输入值</td><td>第4点的输出值</td></tr> <tr><td>n=4</td><td>(不更新)</td><td>(不更新)</td></tr> <tr><td>n=5</td><td>(不更新)</td><td>(不更新)</td></tr> <tr><td>n=6</td><td>(不更新)</td><td>(不更新)</td></tr> <tr><td>n=7</td><td>(不更新)</td><td>(不更新)</td></tr> <tr><td>n=8</td><td>(不更新)</td><td>(不更新)</td></tr> </tbody> </table> <div style="text-align: center;"> <p>偏置 (Offset)</p> <p>写入数据数 (Points)</p> </div> <table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th colspan="2" style="text-align: right;"><展开区域/文件>*1</th> </tr> </thead> <tbody> <tr><td>第0点的输入值</td><td>第0点的输出值</td></tr> <tr><td>第1点的输入值</td><td>第1点的输出值</td></tr> <tr><td>第2点的输入值</td><td>第2点的输出值</td></tr> <tr><td>第3点的输入值</td><td>第3点的输出值</td></tr> <tr><td>第4点的输入值</td><td>第4点的输出值</td></tr> <tr><td>第5点的输入值</td><td>第5点的输出值</td></tr> <tr><td>第6点的输入值</td><td>第6点的输出值</td></tr> <tr><td>第7点的输出值</td><td>第7点的输出值</td></tr> </tbody> </table> </div> <p style="text-align: center;">*1: 取决于读取目标(Target)。</p>		CamData[n][1]	CamData[n][2]	n=1	第2点的输入值	第2点的输出值	n=2	第3点的输入值	第3点的输出值	n=3	第4点的输入值	第4点的输出值	n=4	(不更新)	(不更新)	n=5	(不更新)	(不更新)	n=6	(不更新)	(不更新)	n=7	(不更新)	(不更新)	n=8	(不更新)	(不更新)	<展开区域/文件>*1		第0点的输入值	第0点的输出值	第1点的输入值	第1点的输出值	第2点的输入值	第2点的输出值	第3点的输入值	第3点的输出值	第4点的输入值	第4点的输出值	第5点的输入值	第5点的输出值	第6点的输入值	第6点的输出值	第7点的输出值	第7点的输出值
	CamData[n][1]	CamData[n][2]																																												
n=1	第2点的输入值	第2点的输出值																																												
n=2	第3点的输入值	第3点的输出值																																												
n=3	第4点的输入值	第4点的输出值																																												
n=4	(不更新)	(不更新)																																												
n=5	(不更新)	(不更新)																																												
n=6	(不更新)	(不更新)																																												
n=7	(不更新)	(不更新)																																												
n=8	(不更新)	(不更新)																																												
<展开区域/文件>*1																																														
第0点的输入值	第0点的输出值																																													
第1点的输入值	第1点的输出值																																													
第2点的输入值	第2点的输出值																																													
第3点的输入值	第3点的输出值																																													
第4点的输入值	第4点的输出值																																													
第5点的输入值	第5点的输出值																																													
第6点的输入值	第6点的输出值																																													
第7点的输出值	第7点的输出值																																													
1: 各区间中指定	不能部分进行写入。																																													
2: 样条插补	在偏置(Offset)及写入数据数(Points)中设置“0”,对整个运算配置文件进行写入。设置了“0”以外的情况下,将变为偏置·读取/写入数据数不正确(错误代码: 1BADH)。对于写入数据1(Data1),设置插补方法指定(Interpolate)为“1: 各区间中指定”或“2: 样条插补”的PROFILE_CAM_DATA型的数据。对于写入数据2(Data2),设置将区间数据通过PROFILE_CAM_ELEMENT型的数组进行了定义的数据。																																													

- 运算配置文件格式为“旋转刀具”的情况下

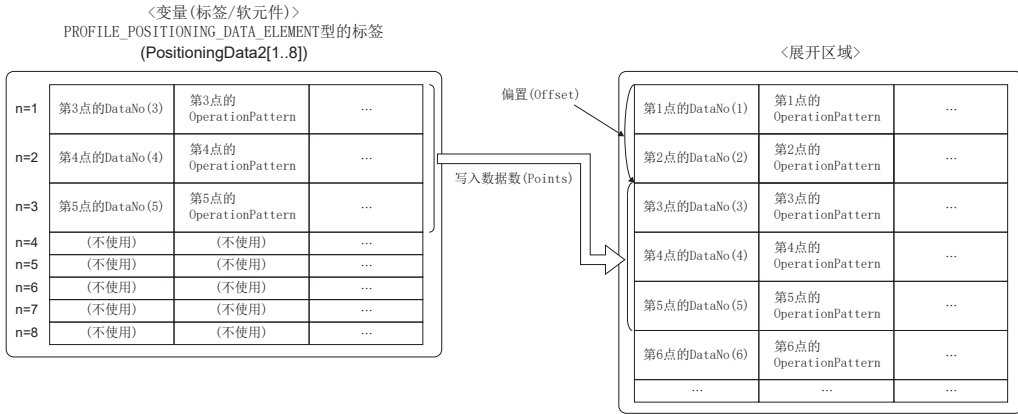
不能部分进行写入。在偏置(Offset)及写入数据数(Points)中设置“0”,对整个运算配置文件进行写入。偏置(Offset)及写入数据数(Points)的设置将被忽略。对于写入数据1(Data1),设置运算配置文件格式为“旋转刀具”的PROFILE_ROTARY_CUTTER型的数据。不使用写入数据2(Data2)。

• 运算配置文件格式为“多轴定位数据”的情况下

写入目标 (Target)	内容
---------------	----

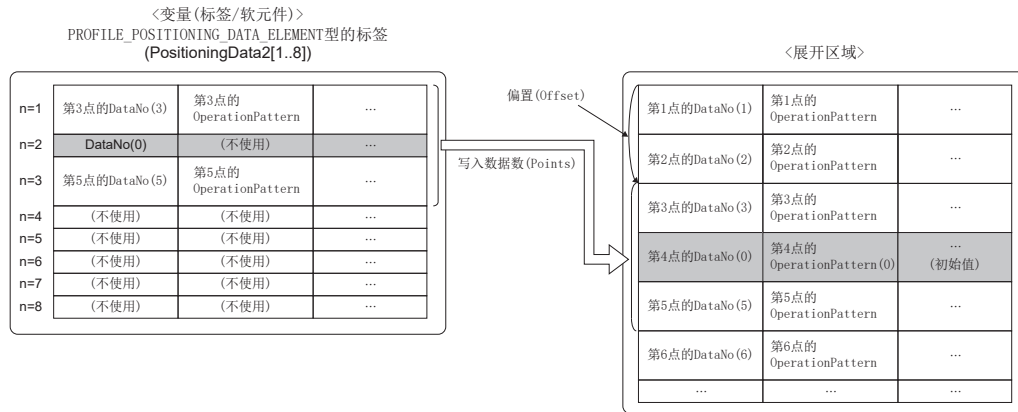
0: 展开区域
 2: 展开区域(运动服务处理)

应通过偏置(Offset)指定写入定位数据的区间的起始, 通过写入数据数(Points)指定区间数(NumberOfSections)。
 在Data1 (TARGET_REF型)中指定的PROFILE_POSITIONING_DATA_ELEMENT型的变量中写入数据。
 在Data2 (TARGET_REF型)中指定PROFILE_POSITIONING_DATA_ELEMENT型数组的变量的情况下, 从指定的变量写入定位数据。
 <例>
 在偏置(Offset)中设置了“2”, 在写入数据数(Points)中设置了“3”时的写入数据2(Data2)的写入结果
 (将写入数据2(Data2)中设置的变量设置为数据类型“PROFILE_POSITIONING_DATA_ELEMENT[1..8]”的标签“PositioningData2”的情况下)



应将指定变量的各定位数据的DataNo指定为与写入目标相同的DataNo。
 • 存在DataNo为“0”的定位数据的情况下, 将视为未设置, 并在相应DataNo的区间数据中存储初始值。(DataNo变为0。)

<例>
 上述示例中将变量n=2的DataNo设置为0的情况下



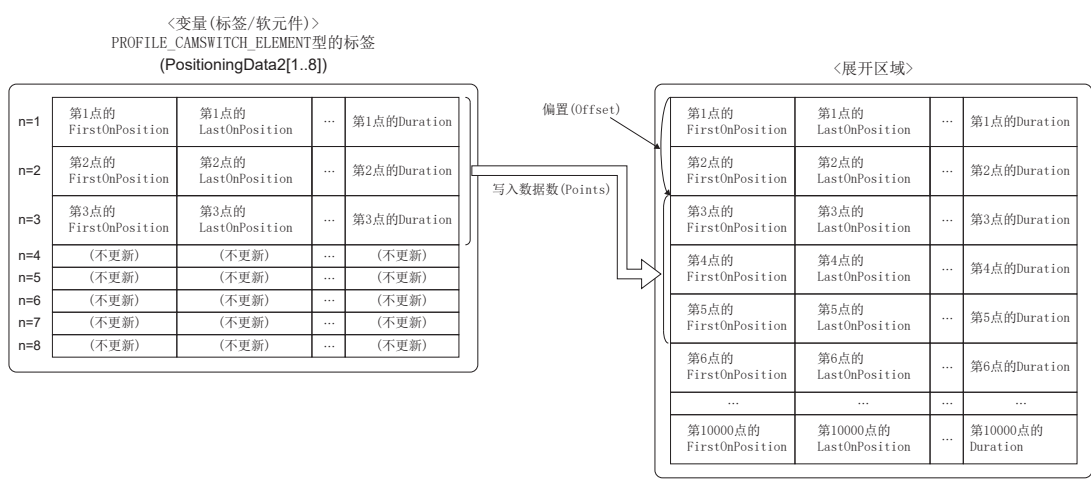
存在定位数据的DataNo为“0”或定位数据不是写入目标DataNo中的任意一个的情况下, 将发生偏置・读取/写入数据数不正确(错误代码: 1BADH)。

在Data2 (TARGET_REF型)中指定PROFILE_POSITIONING_DATA_EXTENDED型的变量的情况下
 从变量的构件ConditionSignal、SkipSignal及PositioningData中指定了WSTRING型的变量中写入条件信号、跳过信号及定位数据。
 应对条件信号、跳过信号指定PROFILE_SIGNAL_SELECT_ELEMENT型数组的数据。
 应对定位数据指定PROFILE_POSITIONING_DATA_ELEMENT型数组的数据。(与在Data2 (TARGET_REF型)中指定了PROFILE_POSITIONING_DATA_ELEMENT型数组的变量的情况相同。)

写入目标 (Target)	内容																																																
1: 文件	<p>应在偏置 (Offset) 及写入数据数 (Points) 中指定 0，并对整个运算配置文件进行写入。 (设置了 0 以外的情况下，将发生偏置・读取/写入数据数不正确 (错误代码: 1BADH)。)</p> <p>从 Data1 (TARGET_REF 型) 中指定的变量写入 PROFILE_POSITIONING_DATA 型的数据。 在 Data2 (TARGET_REF 型) 中指定 PROFILE_POSITIONING_DATA_ELEMENT 型数组的变量的情况下，从指定的变量写入定位数据。 写入区间中，存在定位数据号为“0”的定位数据的情况下，不写入相应 DataNo 的区间数据。</p> <p><例> 在偏置 (Offset) 中设置了“0”，在写入数据数 (Points) 中设置了“0”时的写入数据 2 (Data2) 的写入结果 (将写入数据 2 (Data2) 中设置的变量设置为数据类型“PROFILE_POSITIONING_DATA_ELEMENT [1..100]”的标签“PositioningData2”的情况下)</p> <div style="text-align: center;"> <p><变量 (标签/软元件)> PROFILE_POSITIONING_DATA_ELEMENT 型的标签 (PositioningData2[1..100])</p> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>n=1</td><td>DataNo(0) (不使用)</td><td>(不使用)</td><td>...</td></tr> <tr><td>n=2</td><td>第1点的DataNo(2)</td><td>第1点的OperationPattern</td><td>...</td></tr> <tr><td>n=3</td><td>第2点的DataNo(3)</td><td>第2点的OperationPattern</td><td>...</td></tr> <tr><td>n=4</td><td>DataNo(0) (不使用)</td><td>(不使用)</td><td>...</td></tr> <tr><td>n=5</td><td>第3点的DataNo(5)</td><td>第3点的OperationPattern</td><td>...</td></tr> <tr><td>n=6</td><td>DataNo(0) (不使用)</td><td>(不使用)</td><td>...</td></tr> <tr><td>...</td><td>...</td><td>...</td><td>...</td></tr> <tr><td>n=99</td><td>DataNo(0) (不使用)</td><td>(不使用)</td><td>...</td></tr> <tr><td>n=100</td><td>第4点的DataNo(100)</td><td>第4点的OperationPattern</td><td>...</td></tr> </table> <p><文件></p> <table border="1" style="display: inline-table;"> <tr><td>第1点的DataNo(2)</td><td>第1点的OperationPattern</td><td>...</td></tr> <tr><td>第2点的DataNo(3)</td><td>第2点的OperationPattern</td><td>...</td></tr> <tr><td>第3点的DataNo(5)</td><td>第3点的OperationPattern</td><td>...</td></tr> <tr><td>第4点的DataNo(100)</td><td>第4点的OperationPattern</td><td>...</td></tr> </table> </div> <p>在 Data2 (TARGET_REF 型) 中指定 PROFILE_POSITIONING_DATA_EXTENDED 型的变量的情况下 从变量的构件 ConditionSignal、SkipSignal 及 PositioningData 中指定了 WSTRING 型的变量中写入条件信号、跳过信号及定位数据。 应对条件信号、跳过信号指定 PROFILE_SIGNAL_SELECT_ELEMENT 型数组的数据。 应对定位数据指定 PROFILE_POSITIONING_DATA_ELEMENT 型数组的数据。(与在 Data2 (TARGET_REF 型) 中指定了 PROFILE_POSITIONING_DATA_ELEMENT 型数组的变量的情况相同。)</p>	n=1	DataNo(0) (不使用)	(不使用)	...	n=2	第1点的DataNo(2)	第1点的OperationPattern	...	n=3	第2点的DataNo(3)	第2点的OperationPattern	...	n=4	DataNo(0) (不使用)	(不使用)	...	n=5	第3点的DataNo(5)	第3点的OperationPattern	...	n=6	DataNo(0) (不使用)	(不使用)	n=99	DataNo(0) (不使用)	(不使用)	...	n=100	第4点的DataNo(100)	第4点的OperationPattern	...	第1点的DataNo(2)	第1点的OperationPattern	...	第2点的DataNo(3)	第2点的OperationPattern	...	第3点的DataNo(5)	第3点的OperationPattern	...	第4点的DataNo(100)	第4点的OperationPattern	...
n=1	DataNo(0) (不使用)	(不使用)	...																																														
n=2	第1点的DataNo(2)	第1点的OperationPattern	...																																														
n=3	第2点的DataNo(3)	第2点的OperationPattern	...																																														
n=4	DataNo(0) (不使用)	(不使用)	...																																														
n=5	第3点的DataNo(5)	第3点的OperationPattern	...																																														
n=6	DataNo(0) (不使用)	(不使用)	...																																														
...																																														
n=99	DataNo(0) (不使用)	(不使用)	...																																														
n=100	第4点的DataNo(100)	第4点的OperationPattern	...																																														
第1点的DataNo(2)	第1点的OperationPattern	...																																															
第2点的DataNo(3)	第2点的OperationPattern	...																																															
第3点的DataNo(5)	第3点的OperationPattern	...																																															
第4点的DataNo(100)	第4点的OperationPattern	...																																															

• 运算配置文件格式为“数字凸轮开关数据”的情况下
通过偏置 (Offset) 设置要写入的区间的起始，通过写入数据数 (Points) 设置区间数 (NumberOfSections)。
应对写入数据 1 (Data1) 指定 PROFILE_CAMSWITCH_DATA 型的数据。
对于写入数据 2 (Data2)，应将开关设置的数据 (FirstOnPosition、LastOnPosition、Direction、CamSwitchMode、Duration) 指定为通过 PROFILE_CAMSWITCH_ELEMENT 型的数组定义的数据。

<例>
在偏置 (Offset) 中设置了“0”，在写入数据数 (Points) 中设置了“3”时的写入数据 2 (Data2) 的写入结果
(将写入数据 2 (Data2) 中设置的变量设置为数据类型“PROFILE_CAMSWITCH_ELEMENT [1..8]”的标签“DigitalCamSwitchData2”的情况下)



■写入数据1(Data1)/写入数据2(Data2)

在TARGET_REF结构体中设置运算配置文件的写入数据。可使用的数据类型为[VAR]、[DEV]。

关于TARGET_REF结构体的详细内容，请参阅下述章节。

☞ 1428页 TARGET_REF(输入信号)

可使用的数据类型为各运算配置文件格式的写入用数据结构体的类型。

- 写入数据1(Data1)

写入数据1(Data1)中设置的运算配置文件格式及数据类型如下所示。

运算配置文件格式	数据类型	参阅
凸轮数据	PROFILE_CAM_DATA	1433页 PROFILE_CAM_DATA[P]
旋转刀具用凸轮数据	PROFILE_ROTARY_CUTTER	1435页 PROFILE_ROTARY_CUTTER[P]

例

使用软元件、标签写入的情况下

- 写入PROFILE_CAM_DATA型的标签(CamData1)的情况下

Data1.Target := "[VAR]CamData1"

- 写入数据2(Data2)

写入数据2(Data2)中设置的运算配置文件格式及数据类型如下所示。根据写入目标(Target)及成为对象的凸轮数据的插补方法指定(Interpolate)，设置的数据类型的结构体有所不同。

写入整个运算配置文件的情况下，指定的数组的元素数小于(分辨率+1/坐标数)的情况下，将发生读取/写入数据数不一致(错误代码：1BACH)。

○：支持，×：不支持

运算配置文件格式	数据类型	插补方法指定(Interpolate)			参阅
		0: 直线插补	1: 各区间中指定	2: 样条插补	
凸轮数据	LREAL[]*1	○	×	×	—
	PROFILE_CAM_ELEMENT[]*2	×	○	○*3	1434页 PROFILE_CAM_ELEMENT[P]
旋转刀具用凸轮数据	无需设置。 即使设置也将被忽略。				—

*1 LREAL型的2维数组将按下述方式设置。

LREAL[m..n, o..p]

项目	内容
元素数(n-m+1)	设置为数组元素为要读取的点数及以上。 对整个运算配置文件进行写入的情况下(将偏置(Offset)、写入数据数(Points)设置为“0”的情况下)，应进行设置以确保其为对象的凸轮数据的分辨率+1/坐标数及以上。 但是，大于写入数据1(Data1)中指定的分辨率+1/坐标数的区域的数据不使用。
次元数(p-o+1)	写入各点的1周期当前位置与行程。 设置时，应确保数组元素为“2”。 将数组元素设置为“3”及以上的情况下，可以写入，但是由于数据已被删除可能无法获得预期的结果，因此设置时必须确保其为“2”。

*2 PROFILE_CAM_ELEMENT的数组将按下述方式设置。

PROFILE_CAM_ELEMENT[m..n]

项目	内容
元素数(n-m+1)	设置为数组元素为要读取的点数及以上。 对整个运算配置文件进行读取的情况下(将偏置(Offset)、读取数据数(Points)设置为“0”的情况下)，应进行设置以确保其为对象的凸轮数据的区间数(NumberOfSections)及以上。 但是，大于写入数据1(Data1)中指定的区间数(NumberOfSections)的区域的数据不使用。

*3 仅使用结束点、行程。

例

LREAL型的情况下


插补方法指定 (Interpolate)	设置示例
0: 直线插补	<p>在TARGET_REF结构体的对象(Target)中设置坐标数“100”的二维标签的示例如下所示。</p> <p>Data2.Target := "[VAR]CamData2"</p> <p>* 对于CamData2, 在全局标签的标签名中声明“CamData2”, 在数据类型中声明“LREAL[1..100,1..2]”的标签, 在对象(Target)中指定字符串。要写入的数据如下所示。</p> <ul style="list-style-type: none"> · CamData2[n,1]: 第n点的1周期当前位置 · CamData2[n,2]: 第n点的行程
1: 各区间中指定	<p>在TARGET_REF结构体的对象(Target)中设置分辨率“100”的二维标签的示例如下所示。</p> <p>TARGET_REF.Target := "[VAR]CamData2"</p> <p>* 对于CamData2, 在全局标签的标签名中声明“CamData2”, 在数据类型中声明“LREAL[1..101,1..2]”的标签, 在对象(Target)中指定字符串。要写入的数据如下所示。</p> <ul style="list-style-type: none"> · CamData2[1,1]: 设置为开始点的值 · CamData2[1,2]: 初始行程量的设定值 · CamData2[n+1,1]: 第n点的1周期当前位置 · CamData2[n+1,2]: 第n点的行程
2: 样条插补	<p>在TARGET_REF结构体的对象(Target)中设置分辨率“100”的二维标签的示例如下所示。</p> <p>TARGET_REF.Target := "[VAR]CamData2"</p> <p>* 对于CamData2, 在全局标签的标签名中声明“CamData2”, 在数据类型中声明“LREAL[1..101,1..2]”的标签, 在对象(Target)中指定字符串。要写入的数据如下所示。</p> <ul style="list-style-type: none"> · CamData2[1,1]: 设置为开始点的值 · CamData2[1,2]: 初始行程量的设定值 · CamData2[n+1,1]: 第n点的1周期当前位置 · CamData2[n+1,2]: 第n点的行程

例

PROFILE_CAM_ELEMENT型的情况下

设置示例
<p>在TARGET_REF结构体的对象(Target)中设置区间数“10”的二维标签的示例如下所示。</p> <p>Data2.Target := "[VAR]CamData2"</p> <p>* 对于CamData2, 在全局标签的标签名中声明“CamData2”, 在数据类型中声明“PROFILE_CAM_ELEMENT[1..10]”的标签, 在对象(Target)中指定字符串。要写入的数据如下所示。</p> <ul style="list-style-type: none"> · CamData2[n]: 第n点的区间数据(PROFILE_CAM_ELEMENT型)

要点

- 设置的写入用数据结构体的类型与各运算配置文件格式中请求的类型不一致的情况下, 将变为读取/写入数据类型不正确(错误代码: 1BAAH)。
- 关于写入数据1(Data1)/写入数据2(Data2)中设置的运算配置文件的凸轮数据的详细内容, 请参阅下述手册。
 所使用的控制器的用户手册
- LREAL型的数组中在MCv_WriteProfileData(配置文件写入)的写入目标(Target)中设置“1: 文件”进行了写入的情况下, 将凸轮数据格式的插补方法指定(Interpolate)设置为“0: 执行插补”并输出至文件。

■写入目标(Target)

设置运算配置文件的写入目标，对写入数据1(Data1)/写入数据2(Data2)及运算配置文件展开FB的配置文件ID(PROFILE_ID)进行写入。

设定值	内容
0: 展开区域	<ul style="list-style-type: none"> 参照PROFILE_DATA结构体的配置文件ID(ID)的值，对PROFILE_ID结构体的配置文件ID编号(Number)表示的展开区域的运算配置文件进行写入。 将按照执行的程序的周期进行写入处理。 ■在PROFILE_ID结构体的配置文件ID编号(Number)中设置了“0”的情况下 自动分配空余ID，并将ID存储到配置文件ID编号(Number)中。 对于同一运算配置文件，多次执行配置文件ID编号(Number)为“0”的展开FB时，将被展开到多个不同的展开区域中。 没有空余ID的情况下，将变为运算配置文件ID不足(错误代码：1BA0H)，不进行展开处理。 ■在PROFILE_ID结构体的配置文件ID编号(Number)中设置了“1~60000”的情况下 在不改变设置的ID的状况下进行展开。 设置了已展开的ID的情况下，将覆盖展开数据。
1: 文件	<ul style="list-style-type: none"> 参照PROFILE_DATA结构体的配置文件ID(ID)的值，对PROFILE_ID结构体的配置文件ID编号(Number)表示的展开区域的运算配置文件进行写入。此后，参照PROFILE_DATA结构体的运算配置文件存储位置(Location)的值，对FILE_LOCATION结构体的文件名称(Filename)及文件夹指定(Path)表示的文件的运算配置文件进行写入。对于运算配置文件的配置文件ID，将输出变量的配置文件ID(ProfileID)的值写入到文件中。 将按照运动服务处理的周期进行写入处理。 未设置PROFILE_DATA结构体的运算配置文件存储位置(Location)的情况下，将变为无运算配置文件(错误代码：1B8EH)，不进行写入处理。 设置的文件已存在的情况下，进行覆盖。 设置的文件不存在的情况下，如果偏置(Offset)及写入数据数(Points)的两者的值为“0”则进行新建，且字符代码以“Unicode(有UTF-16LEBOM)”进行写入。 新建文件的情况下，如果偏置(Offset)及写入数据数(Points)的两者的值不为“0”则变为偏置·读取/写入数据数不正确(错误代码：1BADH)。(执行至展开区域的写入。) 设置了“1: 文件”的情况下，偏置(Offset)及写入数据数(Points)中，应设置“0”。(设置了“0”以外的情况下，将变为偏置·读取/写入数据数不正确(错误代码：1BADH)。)
2: 展开区域(运动服务处理)	<ul style="list-style-type: none"> 参照PROFILE_DATA结构体的配置文件ID(ID)的值，对PROFILE_ID结构体的配置文件ID编号(Number)表示的展开区域的运算配置文件进行写入。 将按照运动服务处理的周期进行写入处理。 ■在PROFILE_ID结构体的配置文件ID编号(Number)中设置了“0”的情况下 自动分配空余ID，并将ID存储到配置文件ID编号(Number)中。 对于同一运算配置文件，多次执行配置文件ID编号(Number)为“0”的展开FB时，将被展开到多个不同的展开区域中。 没有空余ID的情况下，将变为运算配置文件ID不足(错误代码：1BA0H)，不进行展开处理。 ■在PROFILE_ID结构体的配置文件ID编号(Number)中设置了“1~60000”的情况下 在不改变设置的ID的状况下进行展开。 设置了已展开的ID的情况下，将覆盖展开数据。

要点

- 读取目标(Target)为“1: 文件”、“2: 展开区域(运动服务处理)”的情况下，由于在运动服务处理中执行，因此到处理完成为止可能需要花费较长时间。应根据需要将Done(执行完成)用于互锁。
- Target = “0: 展开区域”的情况下，由于是在调用了FB的任务内处理，因此立即完成。写入的数据点数较多时，程序的执行时间将会延长，因此应根据需要分多次执行写入，或者使用Target = “2: 展开区域(运动服务处理)”。

■启动模式 (ExecutionMode)

设置执行MCv_WriteProfileData(配置文件写入)的时机。

设定值	内容
0: 立即执行 (mcImmediately)	立即写入展开区域的内容。 有可能对执行中的控制带来影响。 但是, FB的执行中进行写入的情况下, 运算配置文件的格式、分辨率不一致时, 将变为运算配置文件操作中(错误代码: 1B90H)。
1: 等待完成后执行 (mcQueued)	等待执行中的FB的执行完成后写入。 多个FB处于等待状态的情况下, 下一个要执行的FB将成为在优先级较高的任务中执行的FB。 相同的优先度的情况下, 将按启动顺序执行。
3: 推测执行 (mcSpeculatively)	将变为运算配置文件操作中(错误代码: 1B90H), 不更改展开区域。

要点

如果没有正在执行的FB, 则无论启动模式 (ExecutionMode) 的设置如何, 都将立即执行。
访问展开区域的各FB执行中, 各FB的执行中 (Busy) 将变为TRUE。应根据需要将启动模式 (ExecutionMode) 用于互锁。

注意事项

一次可写入的数据点数有限制。1次的FB执行中写入未完成时, 应分为多次执行写入。

程序示例

将“LREAL型(2维数组)的格式”的凸轮数据1(ProfileData0001)与“PROFILE_CAM_ELEMENT型的格式”的凸轮数据2(ProfileData0002)写入到文件中的程序示例如下所示。

■运算配置文件

配置文件ID	标签名	数据类型	注释
1	ProfileData0001	MC_CAM_REF	凸轮数据1
2	ProfileData0002	MC_CAM_REF	凸轮数据2

• 运算配置文件(凸轮数据)的设置

项目	ProfileData0001	ProfileData0002
插补方法指定	直线插补	各区间中指定
分辨率	—	256
1周期长设置	单位	degree
	1周期长	360.00000
行程量	单位	%
	行程量	100.0
凸轮1周期时间	—	1.000[s]
初始行程值	—	0.0000000
凸轮1周期最小值	0.00000	—
凸轮1周期最大值	360.00000	—

• ProfileData0001

区间No.	输入值[degree]	输出值[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

区间No.	开始点[degree]	结束点[degree]	行程[%]	凸轮曲线类型
1	0.00000	90.00000	60.0000000	单弦
2	90.00000	180.00000	100.0000000	单弦
3	180.00000	270.00000	30.0000000	单弦
4	270.00000	0.00000	0.0000000	单弦

■使用的标签

标签名	数据类型	注释
MCv_WriteProfileData_1	MCv_WriteProfileData	配置文件写入FB 1
WriteData1_1	TARGET_REF	写入数据1 配置文件1
WriteData2_1	TARGET_REF	写入数据2 配置文件1
WriteData1_Lreal	PROFILE_CAM_DATA	配置文件数据1 凸轮数据1
WriteData2_Lreal	双精度实数(0..4, 0..1)	配置文件数据1 LREAL型
bWriteProfile1	位	配置文件写入指令1
bDone1	位	完成1
bBusy1	位	执行中1
bError1	位	错误1
uwErrorID1	字[无符号]/位串[16位]	错误代码1
ProfileID1	PROFILE_ID	配置文件ID1
Cam_ProfileData0001* ¹	PROFILE_DATA	配置文件数据0001
MCv_WriteProfileData_2	MCv_WriteProfileData	配置文件写入FB 2
WriteData1_2	TARGET_REF	写入数据1 配置文件2
WriteData2_2	TARGET_REF	写入数据2 配置文件2
WriteData1_Element	PROFILE_CAM_DATA	配置文件数据2 凸轮数据2
WriteData2_Element	PROFILE_CAM_ELEMENT(0..3)	配置文件数据2 PROFILE_CAM_ELEMENT型
bWriteProfile2	位	配置文件写入指令2
bDone2	位	完成2
bBusy2	位	执行中2
bError2	位	错误2
uwErrorID2	字[无符号]/位串[16位]	错误代码2
ProfileID2	PROFILE_ID	配置文件ID2
Cam_ProfileData0002* ¹	PROFILE_DATA	配置文件数据0002

*1 仅FBD/LD的程序使用(为了在FBD/LD的程序中使用运算配置文件需要此项)

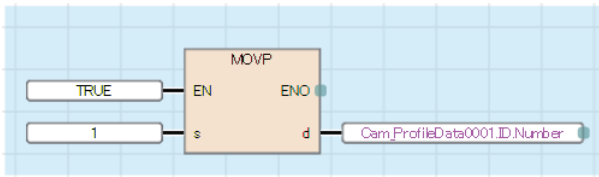
■使用的全局标签

标签名	数据类型	注释	公开标签
G_ProfileCamData* ¹	PROFILE_CAM_DATA	配置文件凸轮数据	有效
G_ProfileCamELEMENT* ¹	PROFILE_CAM_ELEMENT	配置文件凸轮元件	有效

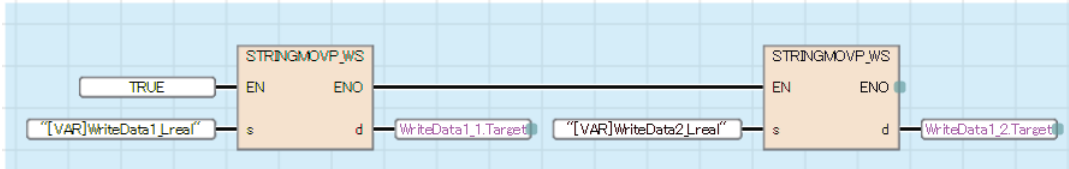
*1 仅FBD/LD的程序使用(为了在FBD/LD的程序中使用运算配置文件需要此项)

■FBD/LD的程序

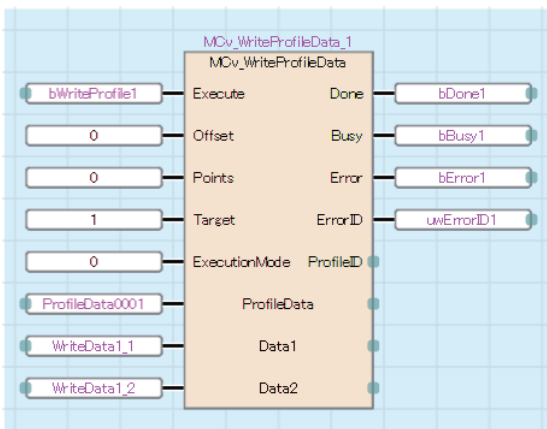
- 配置文件数据0001



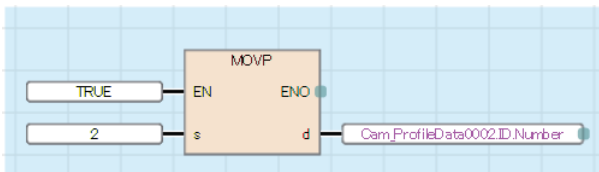
- 写入数据1



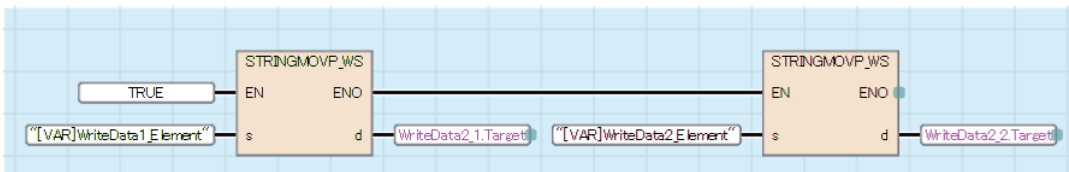
- 配置文件写入1(以LREAL型(2维数组)的格式写入凸轮数据)



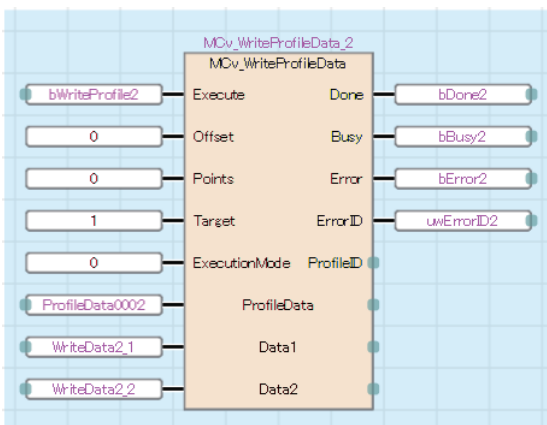
- 配置文件数据0002



- 写入数据2



- 配置文件写入2(以PROFILE_CAM_ELEMENT型的格式写入凸轮数据)



■ST的程序

```
//-----写入数据1-----
WriteData1_1.Target := "[VAR]WriteData1_Lreal";
WriteData1_2.Target := "[VAR]WriteData2_Lreal";

//-----配置文件写入1-----
//-----以LREAL型(2维数组)的格式写入凸轮数据-----
MCv_WriteProfileData_1(
    ProfileData:= ProfileData0001.ProfileData ,
    Data1:= WriteData1_1 ,
    Data2:= WriteData1_2 ,
    Execute:= bWriteProfile1 ,
    Offset:= 0 ,
    Points:= 0 ,
    Target:= 1 ,
    ExecutionMode:= MC_EXECUTION_MODE__mcImmediately ,
    Done=> bDone1 ,
    Busy=> bBusy1 ,
    Error=> bError1 ,
    ErrorID=> uwErrorID1 ,
    ProfileID=> ProfileID1
);

//-----写入数据2-----
WriteData2_1.Target := "[VAR]WriteData1_Element";
WriteData2_2.Target := "[VAR]WriteData2_Element";

//-----配置文件写入2-----
//-----以PROFILE_CAM_ELEMENT型的格式写入凸轮数据-----
MCv_WriteProfileData_2(
    ProfileData:= ProfileData0002.ProfileData ,
    Data1:= WriteData2_1 ,
    Data2:= WriteData2_2 ,
    Execute:= bWriteProfile2 ,
    Offset:= 0 ,
    Points:= 0 ,
    Target:= 1 ,
    ExecutionMode:= MC_EXECUTION_MODE__mcImmediately ,
    Done=> bDone2 ,
    Busy=> bBusy2 ,
    Error=> bError2 ,
    ErrorID=> uwErrorID2 ,
    ProfileID=> ProfileID2
);
```

第9部分 网络FB

本部分由下述章构成。

48 以太网用FB

49 CC-Link IE TSN用FB

48 以太网用FB

要点

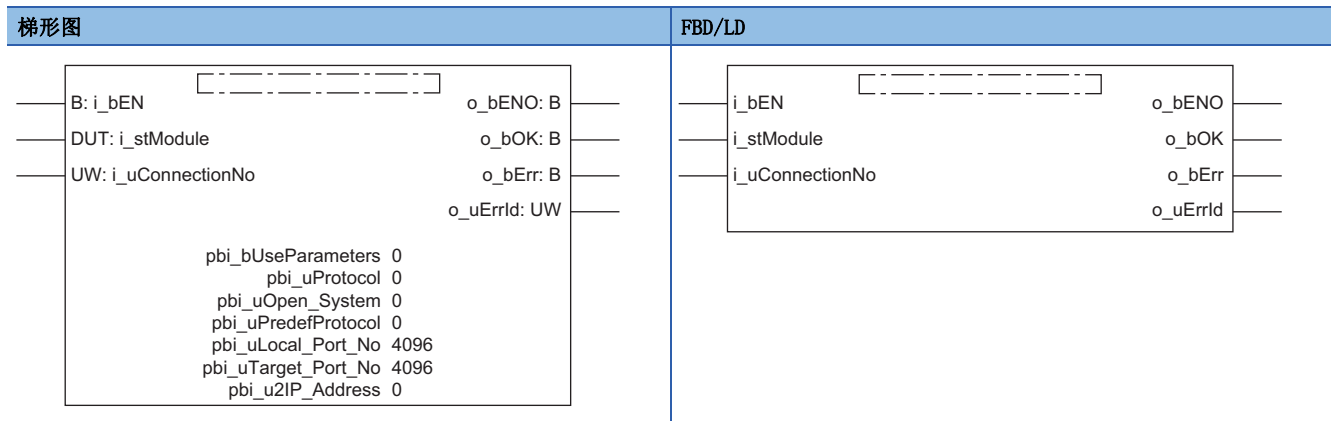
本章的FB为控制器的模块FB。应使用工程工具的部件选择窗口的[Module (模块)]⇒[Module FB (模块FB)]。

48.1 连接的建立/断开

M+MXF-SQ_ConnectionOpen

打开(建立)连接。

指定加密通信协议(TLS和DTLS)时,也需要对象设备支持。



ST

```
M_MXF_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?
);
```

设置数据

■输入参数

变量名	名称	数据类型	有效范围	内容
i_bEN	执行指令	位	—	ON: 启动FB。 OFF: 不启动FB。
i_stModule	模块标签	结构体	—	指定执行FB的控制器的模块标签。 (例: MX_SQ)
i_uConnectionNo	连接No.	字[无符号]/位串[16位]	1~64	指定要打开的连接编号。

■输出参数

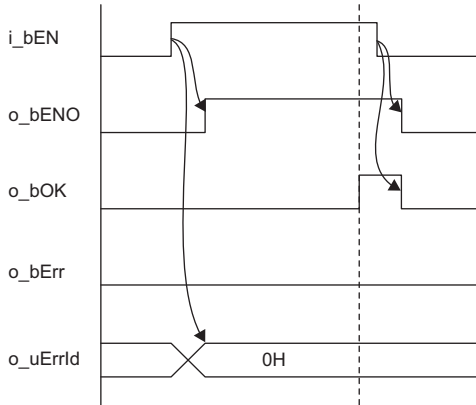
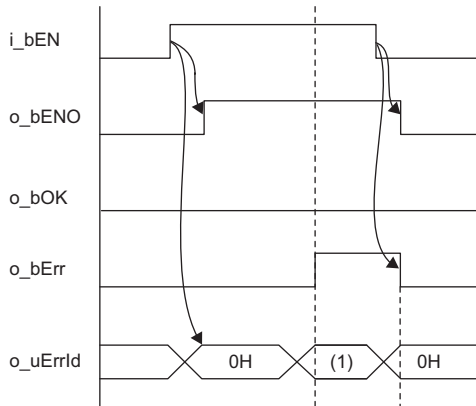
变量名	名称	数据类型	内容	默认值
o_bENO	执行状态	位	ON: 执行指令ON OFF: 执行指令OFF	OFF
o_bOK	正常完成	位	ON时, 表示FB的处理已正常完成。	OFF
o_bErr	异常完成	位	ON时, 表示FB的处理已异常完成。	OFF
o_uErrId	错误代码	字[无符号]/位串[16位]	异常完成时将存储错误代码。	0

■动作参数

变量名	名称	数据类型	有效范围	内容	默认值															
pbi_bUseParameters	使用参数	位	ON、OFF	指定连接的打开处理时，选择是使用通过工程工具的参数设置值，还是使用下列动作参数的设置值。 <ul style="list-style-type: none"> • OFF：以工程工具的对象设备连接配置设置中设置的内容进行打开处理。（不需要设置下列动作参数的值。即使设置值也会被无视。） • ON：以下列动作参数中设置的内容进行打开处理。 	OFF															
pbi_uProtocol	协议	字[无符号]/位串[16位]	0~3	选择要打开的连接协议。 <ul style="list-style-type: none"> • pbi_uPredefProtocol (通信协议设置)为0的情况下 0: TCP 1: UDP 2: TLS 3: DTLS • pbi_uPredefProtocol (通信协议设置)为1的情况下 0: TCP 1: UDP 	0															
pbi_uOpen_System	打开方式	字[无符号]/位串[16位]	0~2	选择连接的打开方式。 <ul style="list-style-type: none"> • 0: Active打开或UDP或DTLS • 1: Unpassive打开 • 2: Fullpassive打开 	0															
pbi_uPredefProtocol	通信协议设置	字[无符号]/位串[16位]	0~1	选择有无使用通信协议功能。 <ul style="list-style-type: none"> • 0: 不使用通信协议支持功能。（使用套接字通信功能。） • 1: 使用通信协议支持功能。 	0															
pbi_uLocal_Port_No	自节点端口编号	字[无符号]/位串[16位]	1~4999、5010~65534	指定自节点的端口编号。 端口编号1~1023一般分配至保留的端口编号(WELL KNOWN PORT NUMBERS)，因此建议使用端口编号1024~4999、5010~65534。	4096															
pbi_uTarget_Port_No	对象设备端口编号	字[无符号]/位串[16位]	1~65534、65535	指定对象设备的端口编号。 指定了65535的连接(仅限协议选择了UDP或DTLS时)中，将全部端口编号作为对象进行接收。由于指定了65535的连接无法发送数据，发送数据时应指定1~65534。	4096															
pbi_u2IP_Address	对象设备IP地址	字[无符号]/位串[16位](0..1)	0.0.0.1~255.255.255.255(00000001H~FFFFFFFH)	指定对象设备的IP地址。 第1字指定第3~4八位字节，第2字指定第1~2八位字节。 进行广播轮询通信时，指定255.255.255.255(FFFFFFFH)。 <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <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 style="text-align: right;">+0</td> <td style="border: 1px solid black; width: 40px; text-align: center;">(3)</td> <td style="border: 1px solid black; width: 40px; text-align: center;">(4)</td> <td colspan="2"></td> </tr> <tr> <td style="text-align: right;">+1</td> <td style="border: 1px solid black; width: 40px; text-align: center;">(1)</td> <td style="border: 1px solid black; width: 40px; text-align: center;">(2)</td> <td colspan="2"></td> </tr> </table>		b15	b8	b7	b0	+0	(3)	(4)			+1	(1)	(2)			192.168.1.1 (COA80101H)
	b15	b8	b7	b0																
+0	(3)	(4)																		
+1	(1)	(2)																		

(1)~(4)：表示IP地址的八位字节。

功能

项目	内容	
对象设备	对象模块	MELSEC MX控制器(以太网用端口)
	对象CPU	MELSEC MX控制器
	对象工程工具	GX Works3
使用语言	梯形图	
标签使用量	标签: 36点 锁存标签: 0点 程序中内置的标签使用量根据使用的控制器、参数中指定的软元件及GX Works3的选项设置而不同。有关GX Works3的选项设置, 请参阅下述内容。 □□GX Works3 操作手册	
功能说明	通过i_bEN(执行指令)的ON, 打开与对象设备数据通信的连接。	
FB编译方式	宏型	
FB动作	脉冲型(多个扫描中执行型)	
输入输出信号的动作	<ul style="list-style-type: none"> 正常完成的情况下  <ul style="list-style-type: none"> 异常完成的情况下(与模块错误时相同)  <p>(1) 错误代码</p>	
注意事项	<ul style="list-style-type: none"> ■关于FB的规格 <ul style="list-style-type: none"> 对于其他模块FB或专用指令正在使用的连接, 无法执行本模块FB。对于使用中的连接执行本模块FB时, 将变为错误。 将pbi_bUseParameters(使用参数)置为ON以动作参数的内容进行打开处理时, 可以使用的通信手段仅限套接字通信及通信协议。 对于通过对对象设备连接配置设置了参数的连接执行本FB时, 将依照本FB指定的动作参数进行打开处理。 本FB使用各程序的标签初始值。控制器的引导运行中, 通过引导文件设置指定使用了本FB的程序文件时, 也应同样通过引导文件设置指定各程序的标签初始值文件。此外, 发生了□□1894页 错误代码中未记载的错误代码时, 可能是引导文件设置中未指定各程序的标签初始值文件。应在引导文件设置中指定各程序的标签初始值文件。 执行了本FB时, 有可能发生控制器的运算异常。此时, 应确认事件履历运算异常的详细信息, 重新修改该模块FB的输入参数。 ■关于FB的动作 <ul style="list-style-type: none"> 应在o_bOK(正常完成)或者o_bErr(异常完成)变为ON后, 将i_bEN(执行指令)置为OFF。当i_bEN(执行指令)变为OFF, o_bOK(正常完成)和o_bErr(异常完成)也变为OFF, o_uErrld(错误代码)被清除为0。 	

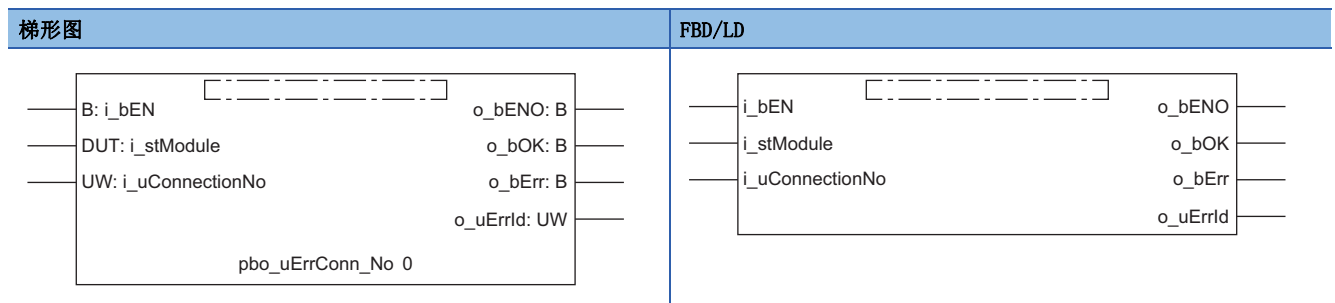
■错误代码

本FB发生的错误代码与SOCOPEN指令相同。

□□1110页 GP. SOCOPE、SP. SOCOPE

M+MXF-SQ_ConnectionClose

关闭(断开)连接。



ST

```
M_MXF_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?
);
```

设置数据

■输入参数

变量名	名称	数据类型	有效范围	内容
i_bEN	执行指令	位	—	ON: 启动FB。 OFF: 不启动FB。
i_stModule	模块标签	结构体	—	指定执行FB的控制器的模块标签。 (例: MX_SQ)
i_uConnectionNo	连接No.	字[无符号]/位串[16位]	1~64、65535	指定要关闭的连接编号。 指定了65535 (FFFFH)时, 将关闭全部连接。

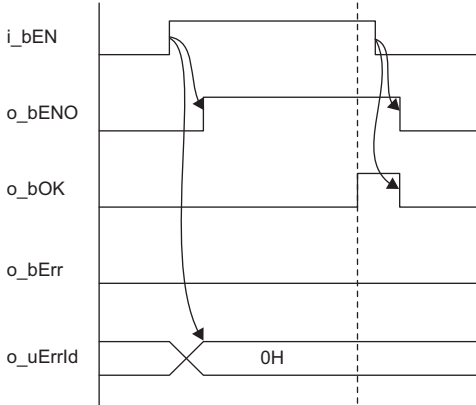
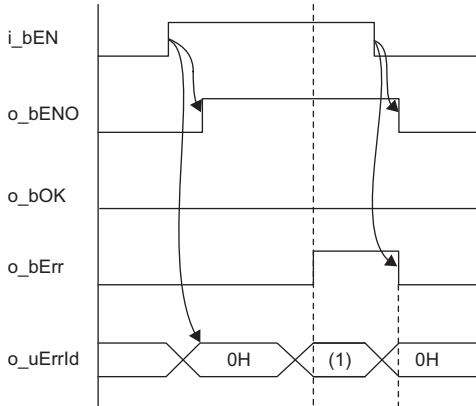
■输出参数

变量名	名称	数据类型	内容	默认值
o_bENO	执行状态	位	ON: 执行指令ON OFF: 执行指令OFF	OFF
o_bOK	正常完成	位	ON时, 表示FB的处理已正常完成。	OFF
o_bErr	异常完成	位	ON时, 表示FB的处理已异常完成。	OFF
o_uErrId	错误代码	字[无符号]/位串[16位]	异常完成时将存储错误代码。	0

■公开变量

变量名	名称	数据类型	内容	默认值
pbo_uErrConn_No	异常发生连接No.	字[无符号]/位串[16位]	存储关闭处理异常完成的连接编号。 连接No. (i_uConnectionNo)中指定了65535 (FFFFH)时, 存储最先关闭处理异常完成的连接编号。	0

功能

项目	内容	
对象设备	对象模块	MELSEC MX控制器(以太网用端口)
	对象CPU	MELSEC MX控制器
	对象工程工具	GX Works3
使用语言	梯形图	
标签使用量	标签: 28点 锁存标签: 0点 程序中内置的标签使用量根据使用的控制器, 参数中指定的软元件及GX Works3的选项设置而不同。有关GX Works3的选项设置, 请参阅下述内容。 GX Works3 操作手册	
功能说明	<ul style="list-style-type: none"> 通过i_bEN(执行指令)的ON, 关闭与对象设备数据通信的连接。 输入参数的连接No. (i_uConnectionNo)中指定了65535(FFFFH)时, 将关闭全部连接。 关闭对象中如果有1个指定的连接无法关闭时, 将异常完成。 	
FB编译方式	宏型	
FB动作	脉冲型(多个扫描中执行型)	
输入输出信号的动作	<ul style="list-style-type: none"> 正常完成的情况下  <ul style="list-style-type: none"> 异常完成的情况下(与模块错误时相同)  <p>(1) 错误代码</p>	
注意事项	<ul style="list-style-type: none"> 对于其他模块FB或专用指令正在使用的连接, 无法执行本模块FB。对于使用中的连接执行本模块FB时, 将变为错误。 应在o_bOK(正常完成)或者o_bErr(异常完成)变为ON后, 将i_bEN(执行指令)置为OFF。当i_bEN(执行指令)变为OFF, o_bOK(正常完成)和o_bErr(异常完成)也变为OFF, o_uErrld(错误代码)被清除为0。 执行了本FB时, 有可能发生控制器的运算异常。此时, 应确认事件履历运算异常的详细信息, 重新修改该模块FB的输入参数。 	

出错

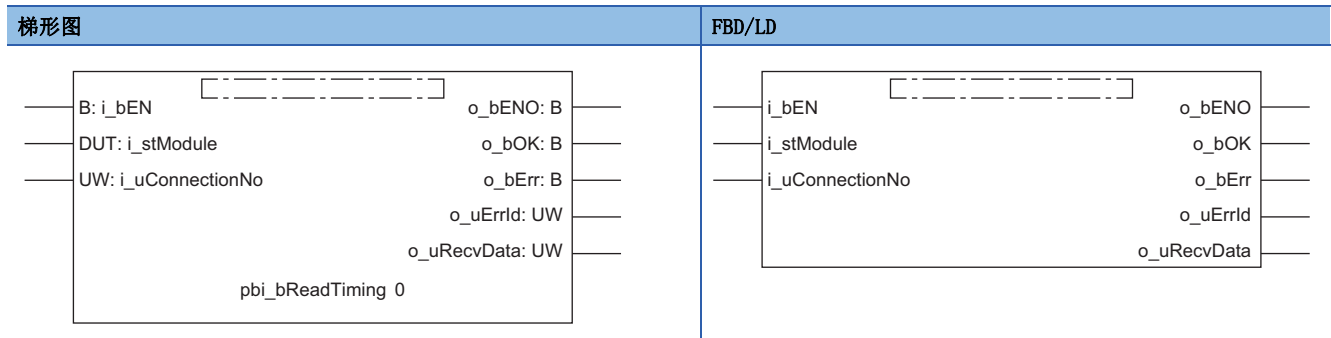
本FB发生的错误代码与SOCCLD指令相同。

☞ 1114页 GP. SOCCLD、SP. SOCCLD

48.2 套接字通信

M+MXF-SQ_Recv_Socket

读取指定的连接中通过套接字通信接收的数据。



ST

```

M_MXF_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?
);
    
```

设置数据

■输入参数

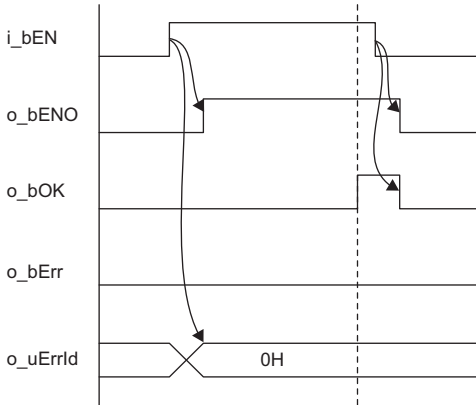
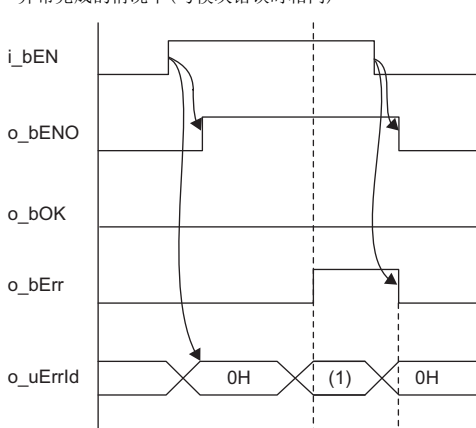
变量名	名称	数据类型	有效范围	内容
i_bEN	执行指令	位	—	ON: 启动FB。 OFF: 不启动FB。
i_stModule	模块标签	结构体	—	指定执行FB的控制器的模块标签。 (例: MX_SQ)
i_uConnectionNo	连接No.	字[无符号]/位串[16位]	1~64	指定要接收的连接编号。

■输出参数

变量名	名称	数据类型	内容	默认值								
o_bENO	执行状态	位	ON: 执行指令ON OFF: 执行指令OFF	OFF								
o_bOK	正常完成	位	ON时, 表示FB的处理已正常完成。	OFF								
o_bErr	异常完成	位	ON时, 表示FB的处理已异常完成。	OFF								
o_uErrId	错误代码	字[无符号]/位串[16位]	异常完成时将存储错误代码。	0								
o_uRecvData	接收数据存储目标	字[无符号]/位串[16位]	<p>指定接收数据长和存储接收数据的元件的起始编号。如下读取的数据将从小编号开始依次被存储。</p> <ul style="list-style-type: none"> 数据单位为字节的情况下 <p>第1字: 接收数据长(单位: 字节) 第2~n字:</p> <table border="1" style="margin-left: 40px;"> <tr> <td style="text-align: center;">b15···b8</td> <td style="text-align: center;">b7···b0</td> </tr> <tr> <td style="text-align: center;">(2)</td> <td style="text-align: center;">(1)</td> </tr> <tr> <td colspan="2" style="text-align: center;">⋮</td> </tr> <tr> <td style="text-align: center;">(4)</td> <td style="text-align: center;">(3)</td> </tr> </table> <p>(1) 接收数据1 (2) 接收数据2 (3) 接收数据m-1 (4) 接收数据m</p> <p>接收数据的数据形式及单位、数据长的范围根据模块的类型及连接编号的不同而不同。</p> <p>接收数据按照字区域的前半部(b0~7)→后半部(b8~15)的顺序存储。</p> <p>作为参数无法进行下列指定。如果指定, 有可能发生错误(2820H: 元件)</p> <ul style="list-style-type: none"> 标签·缓冲存储器指定不正确。 动态指定的数组要素(例: wLabel[D0]) 位指定的标签(例: K4bLabel) 间接指定(例: @W0) 局部元件指定(例: #D0) 	b15···b8	b7···b0	(2)	(1)	⋮		(4)	(3)	0
b15···b8	b7···b0											
(2)	(1)											
⋮												
(4)	(3)											

■动作参数

变量名	名称	数据类型	有效范围	内容	默认值
pbi_bReadTiming	读取时机	位	ON、OFF	<p>指定数据读取处理的执行时机。</p> <ul style="list-style-type: none"> OFF: 模块FB启动后, 立即开始读取。 ON: 模块FB启动后, 最初的END处理时读取。 	OFF

项目	内容	
对象设备	对象模块	MELSEC MX控制器(以太网用端口)
	对象CPU	MELSEC MX控制器
	对象工程工具	GX Works3
使用语言	梯形图	
标签使用量	标签: 24点 锁存标签: 0点 程序中内置的标签使用量根据使用的控制器, 参数中指定的软元件及GX Works3的选项设置而不同。有关GX Works3的选项设置, 请参阅下述内容。 □□GX Works3 操作手册	
功能说明	通过i_bEN(执行指令)的ON, 读取输入参数中指定的连接接收的数据。	
FB编译方式	宏型	
FB动作	脉冲型(多个扫描中执行型)	
输入输出信号的动作	<ul style="list-style-type: none"> 正常完成的情况下  <ul style="list-style-type: none"> 异常完成的情况下(与模块错误时相同)  <p>(1) 错误代码</p>	
注意事项	<ul style="list-style-type: none"> 对于其他模块FB或专用指令正在使用的连接, 无法执行本模块FB。对于使用中的连接执行本模块FB时, 将变为错误。 应在o_bOK(正常完成)或者o_bErr(异常完成)变为ON后, 将i_bEN(执行指令)置为OFF。当i_bEN(执行指令)变为OFF, o_bOK(正常完成)和o_bErr(异常完成)也变为OFF, o_uErrId(错误代码)被清除为0。 本FB的执行指令可以在任意的时机执行, 但在数据接收后执行时, 需要在执行指令的条件中添加G7889~G7892(套接字通信接收状态信号)或者该模块标签。 如果在动作参数的读取时机指定ON(模块FB启动后, 最初的END处理时读取)并执行, 则会在本模块FB的1次END处理内完成数据的读取处理, 因此扫描时间将延长。 本FB使用各程序的标签初始值。控制器的引导运行中, 通过引导文件设置指定使用了本FB的程序文件时, 也应同样通过引导文件设置指定各程序的标签初始值文件。此外, 发生了□□ 1899页 错误代码中未记载的错误代码时, 可能是引导文件设置中未指定各程序的标签初始值文件。应在引导文件设置中指定各程序的标签初始值文件。 执行了本FB时, 有可能发生控制器的运算异常。此时, 应确认事件履历运算异常的详细信息, 重新修改该模块FB的输入参数。 	

■错误代码

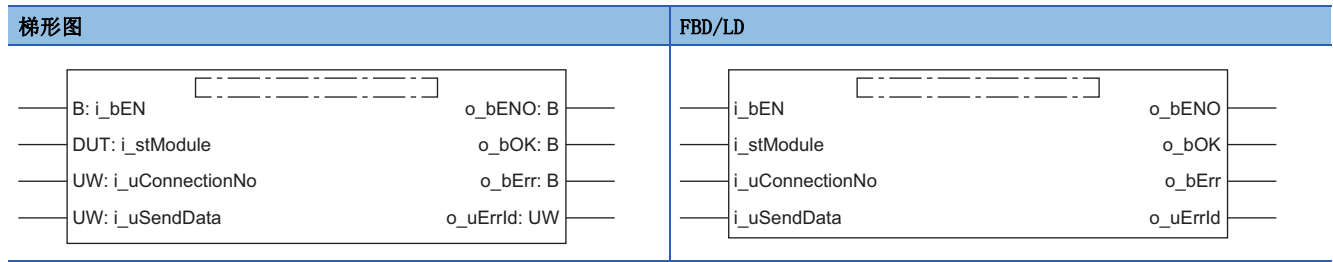
本FB发生的错误代码与SOCRCV指令、SOCRCVS指令相同。

□□ 1116页 GP. SOCRCV、SP. SOCRCV

□□ 1119页 G. SOCRCVS、S. SOCRCVS

M+MXF-SQ_Send_Socket

将套接字通信的数据发送到指定连接的对象设备。



ST

```

M_MXF_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?
);

```

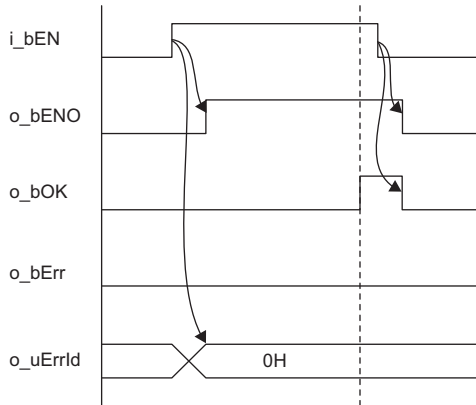
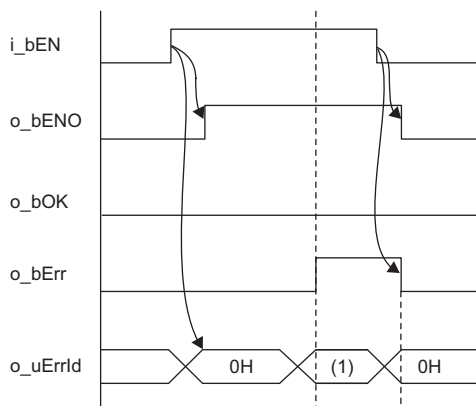
■输入参数

变量名	名称	数据类型	有效范围	内容								
i_bEN	执行指令	位	—	ON: 启动FB。 OFF: 不启动FB。								
i_stModule	模块标签	结构体	—	指定执行FB的控制器的模块标签。 (例: MX_SQ)								
i_uConnectionNo	连接No.	字[无符号]/位串[16位]	1~64	指定要发送的连接编号。								
i_uSendData	发送数据存储目标	字[无符号]/位串[16位]	—	<p>指定发送数据长和存储发送数据的软元件的起始编号。</p> <ul style="list-style-type: none"> • 数据单位为字节的情况下 <p>第1字: 发送数据长(单位: 字节) 第2~n字:</p> <table border="1" style="margin-left: 40px;"> <tr> <td style="text-align: center;">b15···b8</td> <td style="text-align: center;">b7···b0</td> </tr> <tr> <td style="text-align: center;">(2)</td> <td style="text-align: center;">(1)</td> </tr> <tr> <td style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</td> </tr> <tr> <td style="text-align: center;">(4)</td> <td style="text-align: center;">(3)</td> </tr> </table> <p>(1) 发送数据1 (2) 发送数据2 (3) 发送数据m-1 (4) 发送数据m</p> <p>发送数据的数据形式及数据长的范围根据模块的类型及使用的连接设置的不同而不同。 按照字区域的前半部(b0~7)→后半部(b8~15)的顺序发送数据。 作为参数无法进行下列指定。如果指定, 有可能发生错误(2820H: 软元件·标签·缓冲存储器指定不正确)。</p> <ul style="list-style-type: none"> • 动态指定的数组要素(例: wLabel[D0]) • 位指定的标签(例: K4bLabel) • 间接指定(例: @W0) • 局部软元件指定(例: #D0) 	b15···b8	b7···b0	(2)	(1)	⋮	⋮	(4)	(3)
b15···b8	b7···b0											
(2)	(1)											
⋮	⋮											
(4)	(3)											

■输出参数

变量名	名称	数据类型	内容	默认值
o_bENO	执行状态	位	ON: 执行指令ON OFF: 执行指令OFF	OFF
o_bOK	正常完成	位	ON时, 表示FB的处理已正常完成。	OFF
o_bErr	异常完成	位	ON时, 表示FB的处理已异常完成。	OFF
o_uErrId	错误代码	字[无符号]/位串[16位]	异常完成时将存储错误代码。	0

功能

项目	内容	
对象设备	对象模块	MELSEC MX控制器(以太网用端口)
	对象CPU	MELSEC MX控制器
	对象工程工具	GX Works3
使用语言	梯形图	
标签使用量	标签: 24点 锁存标签: 0点 程序中内置的标签使用量根据使用的控制器, 参数中指定的软元件及GX Works3的选项设置而不同。有关GX Works3的选项设置, 请参阅下述内容。 GX Works3 操作手册	
功能说明	通过i_bEN(执行指令)的ON, 发送数据至输入参数中指定连接的对象设备。	
FB编译方式	宏型	
FB动作	脉冲型(多个扫描中执行型)	
输入输出信号的动作	<ul style="list-style-type: none"> 正常完成的情况下  <ul style="list-style-type: none"> 异常完成的情况下(与模块错误时相同)  <p>(1) 错误代码</p>	
注意事项	<ul style="list-style-type: none"> 对于其他模块FB或专用指令正在使用的连接, 无法执行本模块FB。对于使用中的连接执行本模块FB时, 将变为错误。 应在o_bOK(正常完成)或者o_bErr(异常完成)变为ON后, 将i_bEN(执行指令)置为OFF。当i_bEN(执行指令)变为OFF, o_bOK(正常完成)和o_bErr(异常完成)也变为OFF, o_uErrId(错误代码)被清除为0。 执行了本FB时, 有可能发生控制器的运算异常。此时, 应确认事件履历运算异常的详细信息, 重新修改该模块FB的输入参数。 	

出错

本FB发生的错误代码与SOCSND指令相同。

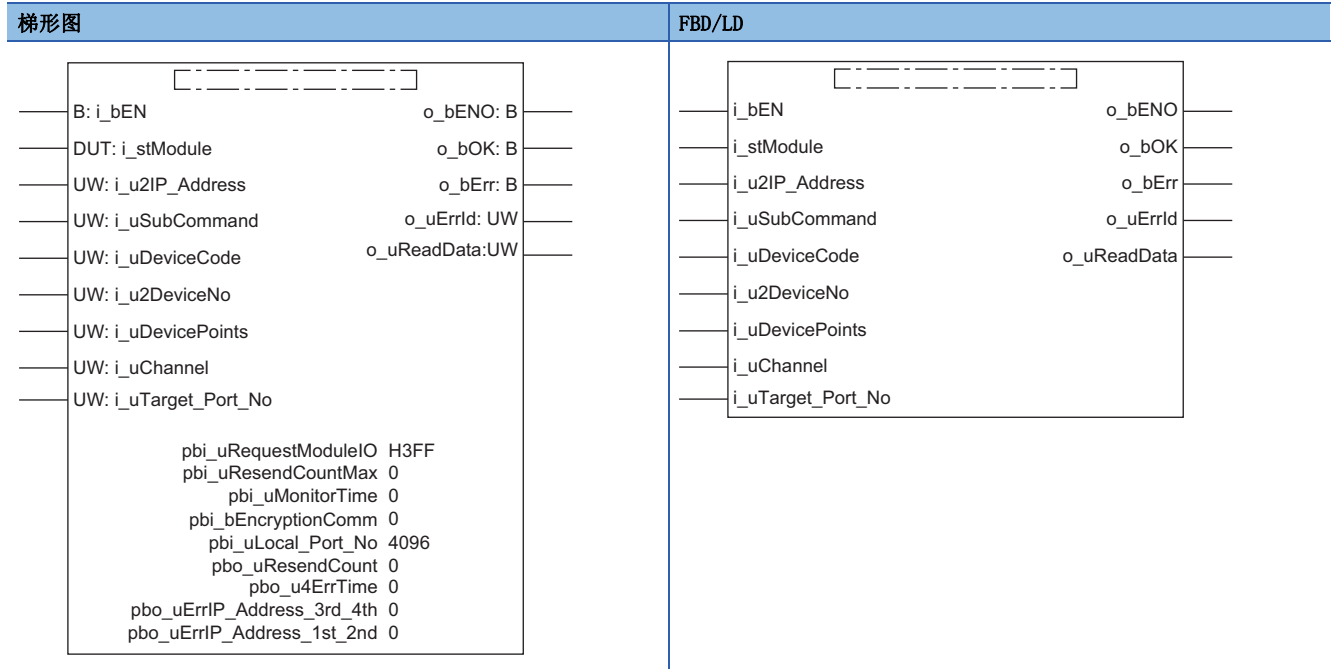
☞ 1122页 GP. SOCSND、SP. SOCSND

M+MXF-SQ_SLMP_DeviceRead_IP

通过IP地址指定读取SLMP对应设备的软元件数据。

对象设备需要支持SLMP指令(Device Read)。

进行加密通信时，对象设备除上述之外，还需要支持加密通信协议(DTLS)。



ST

```

M_MXF_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?
);
    
```

设置数据

■输入参数

变量名	名称	数据类型	有效范围	内容															
i_bEN	执行指令	位	—	ON: 启动FB。 OFF: 不启动FB。															
i_stModule	模块标签	结构体	—	指定执行FB的控制器的模块标签。 (例: MX_SQ)															
i_u2IP_Address	对象设备IP地址	字[无符号]/位串 [16位](0..1)	0.0.0.1~ 253.255.255.254(00000001H~ DFFFFFFFEH)	指定对象设备的IP地址。 第1字指定第3~4八位字节, 第2字指定第1~2八位字节。 但是, 不能设置第4八位字节为0或者为255(FFH)。 <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 style="border: 1px solid black; text-align: center;">(4)</td> <td colspan="2"></td> </tr> <tr> <td>+1</td> <td style="border: 1px solid black; text-align: center;">(1)</td> <td style="border: 1px solid black; text-align: center;">(2)</td> <td colspan="2"></td> </tr> </table> (1)~(4): 表示IP地址的八位字节。		b15	b8	b7	b0	+0	(3)	(4)			+1	(1)	(2)		
	b15	b8	b7	b0															
+0	(3)	(4)																	
+1	(1)	(2)																	
i_uSubCommand	子指令	字[无符号]/位串 [16位]	—	指定读取单位及软元件的指定方法。 ■第0位: 读取单位 0: 字单位 1: 位单位 ■第1位: 读取软元件的指定方法 0: 以2位指定软元件代码, 以6位指定起始软元件编号 (MELSEC-Q/L系列用) 1: 以4位指定软元件代码, 以8位指定起始软元件编号 (MELSEC iQ-R系列用)															
i_uDeviceCode	软元件代码*1	字[无符号]/位串 [16位]	—	以二进制代码指定读取软元件的软元件代码。 • 子指令的第1位为0时: 2位 • 子指令的第1位为1时: 4位															
i_u2DeviceNo	起始软元件编号	字[无符号]/位串 [16位](0..1)	—	以二进制代码指定读取软元件的起始软元件编号。 • 子指令的第1位为0时: 6位 • 子指令的第1位为1时: 8位															
i_uDevicePoints	软元件点数	字[无符号]/位串 [16位]	—	以二进制代码指定读取软元件的软元件点数。 • 子指令的第0位为0时: 1~960 • 子指令的第0位为1时: 1~3972															
i_uChannel	本站使用通道	字[无符号]/位串 [16位]	1~9	指定本站使用的通道。 根据通道决定是否在请求报文中附加序列号*2, 因此要根据用途指定通道。 • 1: 不附加序列号的通道(使用3E帧) • 2~9: 附加序列号的通道(使用4E帧)															
i_uTarget_Port_No	对象设备端口编号	字[无符号]/位串 [16位]	1~65534	指定对象设备的UDP端口编号或DTLS端口编号。															

*1 关于软元件代码的详细内容, 请参阅下述手册。

 SLMP参考手册

*2 向同一SLMP对应设备发送多个请求报文的情况下附加。附加的序列号由系统自动确定。关于序列号, 请参阅下述手册。

 SLMP参考手册

■动作参数

变量名	名称	数据类型	有效范围	内容	默认值
pbi_uRequestModuleIO	请求目标模块I/O编号	字[无符号]/位串 [16位]	03E0H~03E3H、03FFH	指定访问目标的模块。 • 03FFH: 本站、管理CPU • 03E0H: 多CPU1号机 • 03E1H: 多CPU2号机 • 03E2H: 多CPU3号机 • 03E3H: 多CPU4号机	03FFH
pbi_uResendCountMax	最多重新发送次数	字[无符号]/位串 [16位]	0~15	设置在pbi_uMonitorTime(到达监视时间)中指定的监视时间内未完成的情况下重新发送的次数。 • 0次~15次	5
pbi_uMonitorTime	到达监视时间	字[无符号]/位串 [16位]	0、1~32767	指定处理完成为止的监视时间。在监视时间内未完成的情况下将重新发送直至达到pbi_uResendCountMax(最多重新发送次数)中指定的次数为止。 • 0: 10秒 • 1~32767: 1~32767秒	0
pbi_bEncryptionComm	有无加密通信	位串	ON、OFF	指定SLMP帧的有无加密。 • ON: 有效(加密) • OFF: 无效(未加密)	OFF
pbi_uLocal_Port_No	自节点端口编号	字[无符号]/位串 [16位]	1~4999、5010~65534	指定自节点的端口编号。端口编号1~1023一般分配至保留的端口编号(WELL KNOWN PORT NUMBERS), 因此建议使用端口编号1024~4999、5010~65534。 指定的值仅在pbi_bEncryptionComm(有无加密通信)为ON时有效。	4096

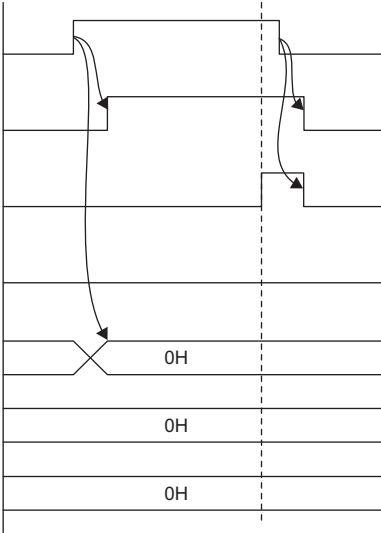
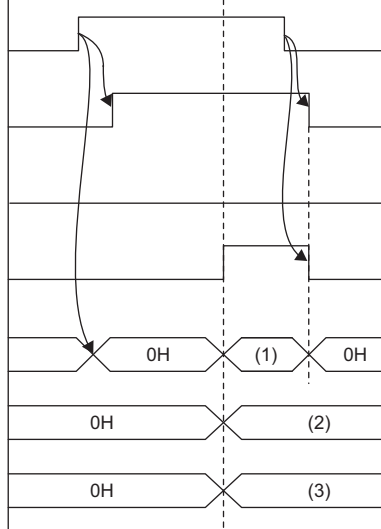
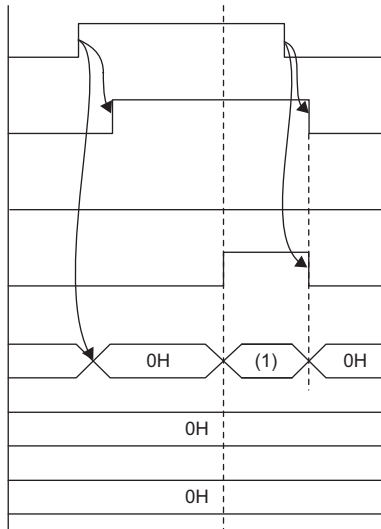
■公开变量

变量名	名称	数据类型	内容	默认值
pbo_uResendCount	重新发送次数	字[无符号]/位串 [16位]	存储重新发送的进行次数(结果)。 存储至检测出异常而中断重新发送为止的重新发送的进行次数(结果)。	0
pbo_u4ErrTime	异常发生时间	字[无符号]/位串 [16位](0..3)	存储异常发生时的时钟数据(仅限异常时)。 第1字 • 高位8位: 月(01H~12H) • 低位8位: 年(00H~99H) 公历低2位 第2字 • 高位8位: 时(00H~23H) • 低位8位: 日(01H~31H) 第3字 • 高位8位: 秒(00H~59H) • 低位8位: 分(00H~59H) 第4字 • 高位8位: 年(00H~99H) 公历高2位 • 低位8位: 星期(00H(日)~06H(六))	0
pbo_uErrIP_Address_3rd_4th	异常检测站IP地址(第3、4八位字节)	字[无符号]/位串 [16位](0..1)	存储检测出异常的站的IP地址(第3八位字节、4八位字节)。*1 例: 192.168.1.2的情况下 0102h	0
pbo_uErrIP_Address_1st_2nd	异常检测站IP地址(第1、2八位字节)	字[无符号]/位串 [16位](0..1)	存储检测出异常的站的IP地址(第1八位字节、2八位字节)。*1 例: 192.168.1.2的情况下 C0A8h	0

*1 仅在专用指令异常完成时存储值。

在异常检测设备IP地址(第3、4八位字节)和异常检测设备IP地址(第1、2八位字节)存储输入参数的对象站地址中设置的值。

项目	内容	
对象设备	对象模块	MELSEC MX控制器(以太网用端口)
	对象CPU	MELSEC MX控制器
	对象工程工具	GX Works3
使用语言	梯形图	
标签使用量	标签：1096点 锁存标签：0点 程序中内置的标签使用量根据使用的控制器，参数中指定的软件元件及GX Works3的选项设置而不同。有关GX Works3的选项设置，请参阅下述内容。 GX Works3 操作手册	
功能说明	<ul style="list-style-type: none"> 通过i_bEN(执行指令)的ON，读取SLMP对应设备的软元件数据。 本FB指定对象设备的IP地址后执行。 本FB使用SLMP的Read指令(指令：0401)。SLMP的指令报文为二进制代码。(见SLMP参考手册) 将pbi_bEncryptionComm(有无加密通信)置为ON执行本FB时，使用pbi_uLocal_Port_No(自节点端口编号)指定的端口编号通过DTLS以加密的SLMP帧进行通信。 	
FB编译方式	宏型	
FB动作	脉冲型(多个扫描中执行型)	

项目	内容
输入输出信号的动作	<p>• 正常完成的情况下</p>  <p>• 异常完成的情况下(专用指令异常完成时)</p>  <p>(1) 错误代码 (2) 异常发生时间 (3) 异常检测设备IP地址</p> <p>• 异常完成的情况下(专用指令正常完成/结束代码异常时)</p>  <p>(1) 错误代码</p>

项目	内容
注意事项	<ul style="list-style-type: none"> • 本FB不包含从错误恢复的处理。关于错误处理，应根据系统或请求动作，另行创建。 • 本FB使用SLMPSND指令或SLMPSNDC指令。SLMPSND指令和SLMPSNDC指令，即使对象设备返回了异常响应的情况下也将正常完成。因此，本FB除了专用指令的完成状态外，还通过响应帧的结束代码，判定正常完成/异常完成。通过结束代码判定为异常完成时，应确认o_uErrId(错误代码)中是否存储了结束代码，以及pbo_u4ErrTime(异常发生时间)与pbo_uErrIP_Address_3rd_4th(异常检测设备IP地址(第3、4八位字节))、pbo_uErrIP_Address_1st_2nd(异常检测设备IP地址(第1、2八位字节))中是否存储了0(初始值)。关于异常原因及处置方法，请参阅所使用的SLMP对应设备的手册。(见SLMP参考手册) • 应在o_bOk(正常完成)或者o_bErr(异常完成)变为ON后，将i_bEN(执行指令)置为OFF。当i_bEN(执行指令)变为OFF，o_bOk(正常完成)和o_bErr(异常完成)也变为OFF，o_uErrId(错误代码)被清除为0。 • 本FB无法通过SLMP的扩展指定读取访问的软元件(链接直接软元件等)。 • 本FB中不能将其他网络的站作为对象站。 • 对于设置了远程口令的对象设备的端口执行本FB时，应先进行远程口令的解锁处理后再执行。对于设置了远程口令的对象设备的端口执行了本FB时，将变为错误。 • 对象站需支持SLMP指令的“Read(指令: 0401)”。 • 本FB仅以二进制代码的通信为对象(不能进行ASCII代码的通信)。 • 本FB通过UDP或DTLS进行通信。对象设备也应置为相同设置。 • 本FB使用各程序的标签初始值。控制器的引导运行中，通过引导文件设置指定使用了本FB的程序文件时，也应同样通过引导文件设置指定各程序的标签初始值文件。此外，发生了图1909页 错误代码中未记载的错误代码时，可能是引导文件设置中未指定各程序的标签初始值文件。应在引导文件设置中指定各程序的标签初始值文件。 • pbi_bEncryptionComm(有无加密通信)为ON(有效)时，应避免使pbi_uLocal_Port_No(自节点端口编号)设置的值与模块参数(以太网)的“External Device Connection Configuration(对象设备连接配置设置)”设置的“PLC Port No.(可编程控制器 端口编号)”中设置的本站的端口编号相同。设置的端口编号重复时，本FB有可能异常完成。 • 执行了本FB时，有可能发生控制器的运算异常。此时，应确认事件履历运算异常的详细信息，重新修改该模块FB的输入参数。

■ 错误代码

错误代码	内容	处理方法
100H	参数的软元件点数(i_uDevicePoints)设置了超出范围的值。	应将软元件点数(i_uDevicePoints)的范围修改为范围内。
100H以外	<p>应确认SLMPSND指令的错误代码并进行处置。</p> <p>图1146页 G(P). SLMPSND、J(P). SLMPSND、SP. SLMPSND</p> <p>进行加密通信时，应确认SLMPSNDC指令的错误代码并进行处置。</p> <p>图1155页 G(P). SLMPSNDC</p>	

M+MXF-SQ_SLMP_DeviceWrite_IP

通过IP地址指定将软元件数据写入至SLMP对应设备。

对象设备需要支持SLMP指令(Device Write)。

进行加密通信时，对象设备除上述之外，还需要支持加密通信协议(DTLS)。

梯形图	FBD/LD

ST

```

M_MXF_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?
);

```

■输入参数

变量名	名称	数据类型	有效范围	内容															
i_bEN	执行指令	位	—	ON: 启动FB。 OFF: 不启动FB。															
i_stModule	模块标签	结构体	—	指定执行FB的控制器的模块标签。 (例: MX_SQ)															
i_u2IP_Address	对象设备IP地址	字[无符号]/位串[16位](0..1)	0.0.0.1~ 253.255.255.254(00000001H~ DFFFFFFEH)	指定对象站的IP地址。 第1字指定第3~4八位字节, 第2字指定第1~2八位字节。 但是, 不能设置第4八位字节为0或者为255(FFH)。 <table border="1" 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="text-align: center;">(3)</td> <td colspan="2"></td> <td style="text-align: center;">(4)</td> </tr> <tr> <td>+1</td> <td colspan="2" style="text-align: center;">(1)</td> <td colspan="2" style="text-align: center;">(2)</td> </tr> </table> (1)~(4): 表示IP地址的八位字节。		b15	b8	b7	b0	+0	(3)			(4)	+1	(1)		(2)	
	b15	b8	b7	b0															
+0	(3)			(4)															
+1	(1)		(2)																
i_uSubCommand	子指令	字[无符号]/位串[16位]	—	指定写入单位及软元件的指定方法。 ■第0位: 写入单位 0: 字单位 1: 位单位 ■写入软元件的指定方法 0: 以2位指定软元件代码, 以6位指定起始软元件编号(MELSEC-Q/L系列用) 1: 以4位指定软元件代码, 以8位指定起始软元件编号(MELSEC iQ-R系列用)															
i_uDeviceCode	软元件代码*1	字[无符号]/位串[16位]	—	以二进制代码指定写入软元件的软元件代码。 • 子指令的第1位为0时: 2位 • 子指令的第1位为1时: 4位															
i_u2DeviceNo	起始软元件编号	字[无符号]/位串[16位](0..1)	—	以二进制代码指定写入软元件的起始软元件编号。 • 子指令的第1位为0时: 6位 • 子指令的第1位为1时: 8位															
i_uDevicePoints	软元件点数	字[无符号]/位串[16位]	—	以二进制代码指定写入软元件的软元件点数。 • 子指令的第0位为0时: 1~960 • 子指令的第0位为1时: 1~3972															

变量名	名称	数据类型	有效范围	内容																																																																																																
i_uWriteData	写入数据存储目标	字[无符号]/位串[16位]	—	<p>指定存储写入数据的软件的起始软元件编号。</p> <p>■子指令的第0位为0时 以字单位写入软元件数据。 例：以字单位写入位软元件M100~M115(1字)时 第1字：</p> <table border="1" style="margin-left: 20px;"> <tr> <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 style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> <td style="text-align: center;">4</td> </tr> <tr> <td style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">M115</td> <td style="text-align: center;">⋯</td> <td style="text-align: center;">M100</td> <td></td> </tr> </table> <p>例：以字单位写入字软元件D0~D2时 第1字：</p> <table border="1" style="margin-left: 20px;"> <tr> <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 style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> <td style="text-align: center;">4</td> </tr> <tr> <td colspan="4" style="text-align: center;">D0</td> </tr> </table> <p>第2字：</p> <table border="1" style="margin-left: 20px;"> <tr> <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 style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">2</td> </tr> <tr> <td colspan="4" style="text-align: center;">D1</td> </tr> </table> <p>第3字：</p> <table border="1" style="margin-left: 20px;"> <tr> <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 style="text-align: center;">1</td> <td style="text-align: center;">D</td> <td style="text-align: center;">E</td> <td style="text-align: center;">F</td> </tr> <tr> <td colspan="4" style="text-align: center;">D2</td> </tr> </table> <p>■子指令的第0位为1时 以位单位写入软元件数据。 例：以位单位写入位软元件M100~M107时 第1字：</p> <table border="1" style="margin-left: 20px;"> <tr> <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 style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">M102</td> <td style="text-align: center;">M103</td> <td style="text-align: center;">M100</td> <td style="text-align: center;">M101</td> </tr> </table> <p>第2字：</p> <table border="1" style="margin-left: 20px;"> <tr> <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 style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">M106</td> <td style="text-align: center;">M107</td> <td style="text-align: center;">M104</td> <td style="text-align: center;">M105</td> </tr> </table>	b15	b8	b7	b0	1	2	3	4	⋮	⋮	⋮	⋮	0	0	1	0	0	1	0	0	0	0	1	1	0	1	0	1	0	0	0	0	M115	⋯	M100		b15	b8	b7	b0	1	2	3	4	D0				b15	b8	b7	b0	0	0	0	2	D1				b15	b8	b7	b0	1	D	E	F	D2				b15	b8	b7	b0	0	1	0	0	M102	M103	M100	M101	b15	b8	b7	b0	1	1	0	0	M106	M107	M104	M105
b15	b8	b7	b0																																																																																																	
1	2	3	4																																																																																																	
⋮	⋮	⋮	⋮																																																																																																	
0	0	1	0																																																																																																	
0	1	0	0																																																																																																	
0	0	1	1																																																																																																	
0	1	0	1																																																																																																	
0	0	0	0																																																																																																	
M115	⋯	M100																																																																																																		
b15	b8	b7	b0																																																																																																	
1	2	3	4																																																																																																	
D0																																																																																																				
b15	b8	b7	b0																																																																																																	
0	0	0	2																																																																																																	
D1																																																																																																				
b15	b8	b7	b0																																																																																																	
1	D	E	F																																																																																																	
D2																																																																																																				
b15	b8	b7	b0																																																																																																	
0	1	0	0																																																																																																	
M102	M103	M100	M101																																																																																																	
b15	b8	b7	b0																																																																																																	
1	1	0	0																																																																																																	
M106	M107	M104	M105																																																																																																	
i_uChannel	本站使用通道	字[无符号]/位串[16位]	1~9	<p>指定本站使用的通道。</p> <p>根据通道决定是否在请求报文中附加序列号*2，因此要根据用途指定通道。</p> <ul style="list-style-type: none"> • 1：不附加序列号的通道(使用3E帧) • 2~9：附加序列号的通道(使用4E帧) 																																																																																																
i_uTarget_Port_No	对象设备端口编号	字[无符号]/位串[16位]	1~65534	指定对象设备的UDP端口编号或DTLS端口编号。																																																																																																

*1 关于软元件代码的详细内容，请参阅下述手册。

📖 SLMP参考手册

*2 向同一SLMP对应设备发送多个请求报文的情况下附加。附加的序列号由系统自动确定。关于序列号，请参阅下述手册。

📖 SLMP参考手册

■输出参数

变量名	名称	数据类型	内容	默认值
o_bENO	执行状态	位	ON: 执行指令ON OFF: 执行指令OFF	OFF
o_bOK	正常完成	位	ON时, 表示FB的处理已正常完成。	OFF
o_bErr	异常完成	位	ON时, 表示FB的处理已异常完成。	OFF
o_uErrId	错误代码	字[无符号]/位串[16位]	异常完成时将存储错误代码。	0

■动作参数

变量名	名称	数据类型	有效范围	内容	默认值
pbi_uRequestModuleIO	请求目标模块I/O编号	字[无符号]/位串[16位]	03E0H~03E3H、03FFH	指定访问目标的模块。 • 03FFH: 本站、管理CPU • 03E0H: 多CPU1号机 • 03E1H: 多CPU2号机 • 03E2H: 多CPU3号机 • 03E3H: 多CPU4号机	03FFH
pbi_uResendCountMax	最多重新发送次数	字[无符号]/位串[16位]	0~15	设置在pbi_uMonitorTime(到达监视时间)中指定的监视时间内未完成的情况下重新发送的次数。 • 0次~15次	5
pbi_uMonitorTime	到达监视时间	字[无符号]/位串[16位]	0、1~32767	指定处理完成为止的监视时间。在监视时间内未完成的情况下将重新发送直至达到pbi_uResendCountMax(最多重新发送次数)中指定的次数为止。 • 0: 10秒 • 1~32767: 1~32767秒	0
pbi_bEncryptionComm	有无加密通信	位串	ON、OFF	指定SLMP帧的有无加密。 • ON: 有效(加密) • OFF: 无效(未加密)	OFF
pbi_uLocal_Port_No	自节点端口编号	字[无符号]/位串[16位]	1~4999、5010~65534	指定自节点的端口编号。端口编号1~1023一般分配至保留的端口编号(WELL KNOWN PORT NUMBERS), 因此建议使用端口编号1024~4999、5010~65534。 指定的值仅在pbi_bEncryptionComm(有无加密通信)为ON时有效。	4096

■公开变量

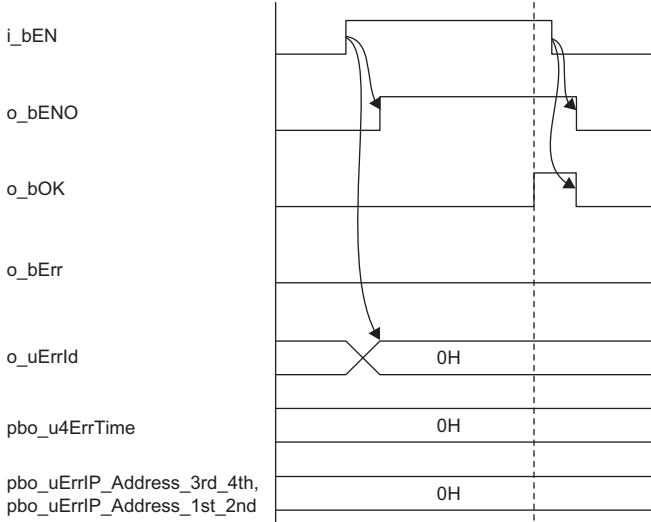
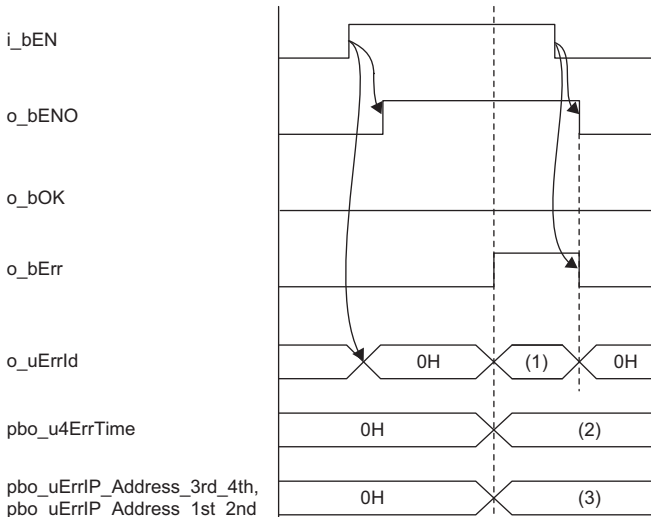
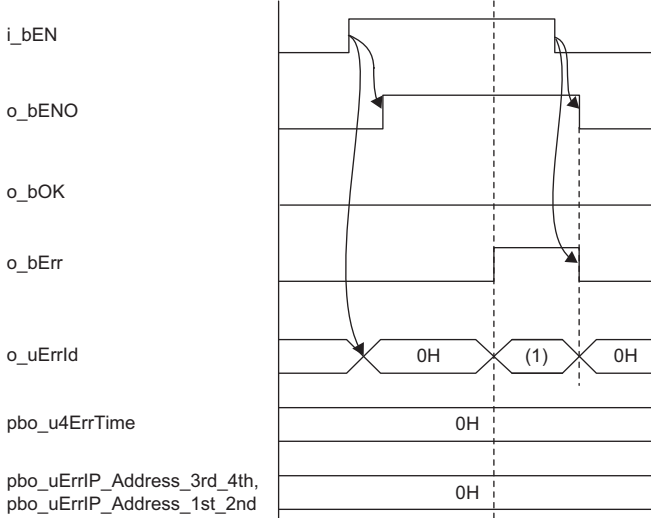
变量名	名称	数据类型	内容	默认值
pbo_uResendCount	重新发送次数	字[无符号]/位串[16位]	存储重新发送的进行次数(结果)。	0
pbo_u4ErrTime	异常发生时间	字[无符号]/位串[16位](0..3)	存储异常发生时的时钟数据(仅限异常时)。 第1字 • 高位8位: 月(01H~12H) • 低位8位: 年(00H~99H) 农历低2位 第2字 • 高位8位: 时(00H~23H) • 低位8位: 日(01H~31H) 第3字 • 高位8位: 秒(00H~59H) • 低位8位: 分(00H~59H) 第4字 • 高位8位: 年(00H~99H) 农历高2位 • 低位8位: 星期(00H(日)~06H(六))	0
pbo_uErrIP_Address_3rd_4th	异常检测站IP地址(第3、4八位字节)	字[无符号]/位串[16位](0..1)	存储检测出异常的站的IP地址(第3八位字节、4八位字节)。*1 例: 192.168.1.2的情况下 0102h	0
pbo_uErrIP_Address_1st_2nd	异常检测站IP地址(第1、2八位字节)	字[无符号]/位串[16位](0..1)	存储检测出异常的站的IP地址(第1八位字节、2八位字节)。*1 例: 192.168.1.2的情况下 COA8h	0

*1 仅在专用指令异常完成时存储值。

在异常检测设备IP地址(第3、4八位字节)和异常检测设备IP地址(第1、2八位字节)存储输入参数的对象站地址中设置的值。

功能

项目	内容	
对象设备	对象模块	MELSEC MX控制器(以太网用端口)
	对象CPU	MELSEC MX控制器
	对象工程工具	GX Works3
使用语言	梯形图	
标签使用量	标签: 1096点 锁存标签: 0点 程序中内置的标签使用量根据使用的控制器, 参数中指定的软元件及GX Works3的选项设置而不同。有关GX Works3的选项设置, 请参阅下述内容。 GX Works3 操作手册	
功能说明	<ul style="list-style-type: none"> 通过i_bEN(执行指令)的ON, 写入SLMP对应设备的软元件数据。 本FB指定对象设备的IP地址后执行。 本FB使用SLMP的Write指令(指令: 1401)。SLMP的指令报文为二进制代码。(见SLMP参考手册) 将pbi_bEncryptionComm(有无加密通信)置为ON执行本FB时, 使用pbi_uLocal_Port_No(自节点端口编号)指定的端口编号通过DTLS以加密的SLMP帧进行通信。 	
FB编译方式	宏型	
FB动作	脉冲型(多个扫描中执行型)	

项目	内容
输入输出信号的动作	<p>• 正常完成的情况下</p>  <p>• 异常完成的情况下(专用指令异常完成时)</p>  <p>(1) 错误代码 (2) 异常发生时间 (3) 异常检测设备IP地址</p> <p>• 异常完成的情况下(专用指令正常完成/结束代码异常时)</p>  <p>(1) 错误代码</p>

项目	内容
注意事项	<ul style="list-style-type: none"> • 本FB不包含从错误恢复的处理。关于错误处理，应根据系统或请求动作，另行创建。 • 本FB使用SLMPSND指令或SLMPSNDC指令。SLMPSND指令和SLMPSNDC指令，即使对象设备返回了异常响应的情况下也将正常完成。因此，本FB除了专用指令的完成状态外，还通过响应帧的结束代码，判定正常完成/异常完成。通过结束代码判定为异常完成时，应确认o_uErrId(错误代码)中是否存储了结束代码，以及pbo_u4ErrTime(异常发生时间)与pbo_uErrIP_Address_3rd_4th(异常检测设备IP地址(第3、4八位字节))、pbo_uErrIP_Address_1st_2nd(异常检测设备IP地址(第1、2八位字节))中是否存储了0(初始值)。关于异常原因及处置方法，请参阅所使用的SLMP对应设备的手册。(见SLMP参考手册) • 应在o_bOk(正常完成)或者o_bErr(异常完成)变为ON后，将i_bEN(执行指令)置为OFF。当i_bEN(执行指令)变为OFF，o_bOk(正常完成)和o_bErr(异常完成)也变为OFF，o_uErrId(错误代码)被清除为0。 • 本FB无法通过SLMP的扩展指定写入访问的软元件(链接直接软元件等)。 • 本FB中不能将其他网络的站作为对象站。 • 对于设置了远程口令的对象设备的端口执行本FB时，应先进行远程口令的解锁处理后再执行。对于设置了远程口令的对象设备的端口执行了本FB时，将变为错误。 • 对象站需支持SLMP指令的“Write(指令: 1401)”。 • 本FB仅以二进制代码的通信为对象(不能进行ASCII代码的通信)。 • 本FB通过UDP或DTLS进行通信。对象设备也应置为相同设置。 • 本FB使用各程序的标签初始值。控制器的引导运行中，通过引导文件设置指定使用了本FB的程序文件时，也应同样通过引导文件设置指定各程序的标签初始值文件。此外，发生了 1916页 错误代码中未记载的错误代码时，可能是引导文件设置中未指定各程序的标签初始值文件。应在引导文件设置中指定各程序的标签初始值文件。 • pbi_bEncryptionComm(有无加密通信)为ON(有效)时，应避免使pbi_uLocal_Port_No(自节点端口编号)设置的值与模块参数(以太网)的“External Device Connection Configuration(对象设备连接配置设置)”设置的“PLC Port No.(可编程控制器 端口编号)”中设置的本站的端口编号相同。设置的端口编号重复时，本FB有可能异常完成。 • 执行了本FB时，有可能发生控制器的运算异常。此时，应确认事件履历运算异常的详细信息，重新修改该模块FB的输入参数。

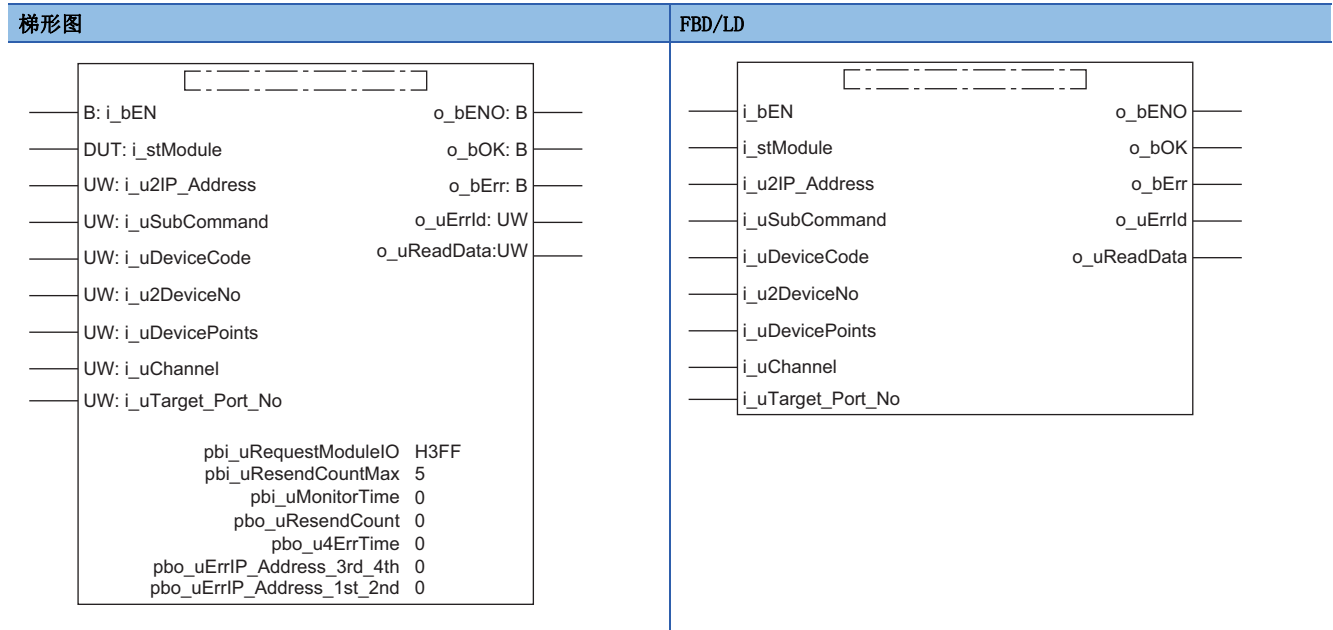
■ 错误代码

错误代码	内容	处理方法
100H	参数的软元件点数(i_uDevicePoints)设置了超出范围的值。	应将软元件点数(i_uDevicePoints)的范围修改为范围内。
100H以外	<p>应确认SLMPSND指令的错误代码并进行处置。</p> <p>1146页 G(P). SLMPSND、J(P). SLMPSND、SP. SLMPSND 进行加密通信时，应确认SLMPSNDC指令的错误代码并进行处置。</p> <p>1155页 G(P). SLMPSNDC</p>	

M+MXF-TSN_SLMP_DeviceRead_IP

通过IP地址指定读取SLMP对应设备的软元件数据。

对象设备需要支持SLMP指令(Device Read)。



ST

```

M_MXF_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?
);

```

设置数据

■输入参数

变量名	名称	数据类型	有效范围	内容															
i_bEN	执行指令	位	—	ON: 启动FB。 OFF: 不启动FB。															
i_stModule	模块标签	结构体	—	指定执行FB的控制器的模块标签。 (例: MX_TSN)															
i_u2IP_Address	对象设备IP地址	字[无符号]/位串[16位](0..1)	0.0.0.1~ 253.255.255.254(00000001H~ DFFFFFFFEH)	指定对象设备的IP地址。 第1字指定第3~4位字节, 第2字指定第1~2位字节。 但是, 不能设置第4位字节为0或者为255(FFH)。 <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 style="border: 1px solid black; text-align: center;">(4)</td> <td></td> <td></td> </tr> <tr> <td>+1</td> <td style="border: 1px solid black; text-align: center;">(1)</td> <td style="border: 1px solid black; text-align: center;">(2)</td> <td></td> <td></td> </tr> </table> (1)~(4): 表示IP地址的八位字节。		b15	b8	b7	b0	+0	(3)	(4)			+1	(1)	(2)		
	b15	b8	b7	b0															
+0	(3)	(4)																	
+1	(1)	(2)																	
i_uSubCommand	子指令	字[无符号]/位串[16位]	—	指定读取单位及软元件的指定方法。 ■第0位: 读取单位 0: 字单位 1: 位单位 ■第1位: 读取软元件的指定方法 0: 以2位指定软元件代码, 以6位指定起始软元件编号(MELSEC-Q/L系列用) 1: 以4位指定软元件代码, 以8位指定起始软元件编号(MELSEC-iQ-R系列用)															
i_uDeviceCode	软元件代码*1	字[无符号]/位串[16位]	—	以二进制代码指定读取软元件的软元件代码。 • 子指令的第1位为0时: 2位 • 子指令的第1位为1时: 4位															
i_u2DeviceNo	起始软元件编号	字[无符号]/位串[16位](0..1)	—	以二进制代码指定读取软元件的起始软元件编号。 • 子指令的第1位为0时: 6位 • 子指令的第1位为1时: 8位															
i_uDevicePoints	软元件点数	字[无符号]/位串[16位]	—	以二进制代码指定读取软元件的软元件点数。 • 子指令的第0位为0时: 1~960 • 子指令的第0位为1时: 1~3972															
i_uChannel	本站使用通道	字[无符号]/位串[16位]	1~9	指定本站使用的通道。 • 1: 不附加序列号的通道(使用3E帧) • 2~9: 附加序列号的通道(使用4E帧) • 10~17: 站号扩展帧															
i_uTarget_Port_No	对象设备端口编号	字[无符号]/位串[16位]	1~65534	指定对象设备的UDP端口编号。															

*1 关于软元件代码的详细内容, 请参阅下述手册。

📖 SLMP参考手册

■输出参数

变量名	名称	数据类型	内容	默认值																																																																																												
o_bENO	执行状态	位	ON: 执行指令ON OFF: 执行指令OFF	OFF																																																																																												
o_bOK	正常完成	位	ON时, 表示FB的处理已正常完成。	OFF																																																																																												
o_bErr	异常完成	位	ON时, 表示FB的处理已异常完成。	OFF																																																																																												
o_uErrId	错误代码	字[无符号]/位串 [16位]	异常完成时将存储错误代码。	0																																																																																												
o_uReadData	读取数据存储目标	字[无符号]/位串 [16位]	<p>指定存储读取数据的软元件的起始软元件编号。以二进制代码存储读取的数据。</p> <p>■子指令的第0位为0时 以字单位读取软元件数据。 例: 以字单位读取位软元件M100~M115(1字)时 第1字:</p> <table border="1" style="margin-left: 20px;"> <tr> <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 style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> <td style="text-align: center;">4</td> </tr> <tr> <td style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">M115</td> <td style="text-align: center;">⋯</td> <td style="text-align: center;">M100</td> <td></td> </tr> </table> <p>例: 以字单位读取字软元件D0~D2时 第1字:</p> <table border="1" style="margin-left: 20px;"> <tr> <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 style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> <td style="text-align: center;">4</td> </tr> <tr> <td colspan="4" style="text-align: center;">D0</td> </tr> </table> <p>第2字:</p> <table border="1" style="margin-left: 20px;"> <tr> <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 style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">2</td> </tr> <tr> <td colspan="4" style="text-align: center;">D1</td> </tr> </table> <p>第3字:</p> <table border="1" style="margin-left: 20px;"> <tr> <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 style="text-align: center;">1</td> <td style="text-align: center;">D</td> <td style="text-align: center;">E</td> <td style="text-align: center;">F</td> </tr> <tr> <td colspan="4" style="text-align: center;">D2</td> </tr> </table> <p>■子指令的第0位为1时 以位单位读取软元件数据。 例: 以位单位读取位软元件M100~M107时 第1字:</p> <table border="1" style="margin-left: 20px;"> <tr> <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 style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">M102</td> <td style="text-align: center;">M103</td> <td style="text-align: center;">M100</td> <td style="text-align: center;">M101</td> </tr> </table> <p>第2字:</p> <table border="1" style="margin-left: 20px;"> <tr> <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 style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">M106</td> <td style="text-align: center;">M107</td> <td style="text-align: center;">M104</td> <td style="text-align: center;">M105</td> </tr> </table>	b15	b8	b7	b0	1	2	3	4	⋮	⋮	⋮	⋮	0	0	1	0	0	1	0	0	0	0	1	1	0	1	0	0	M115	⋯	M100		b15	b8	b7	b0	1	2	3	4	D0				b15	b8	b7	b0	0	0	0	2	D1				b15	b8	b7	b0	1	D	E	F	D2				b15	b8	b7	b0	0	1	0	0	M102	M103	M100	M101	b15	b8	b7	b0	1	1	0	0	M106	M107	M104	M105	0
b15	b8	b7	b0																																																																																													
1	2	3	4																																																																																													
⋮	⋮	⋮	⋮																																																																																													
0	0	1	0																																																																																													
0	1	0	0																																																																																													
0	0	1	1																																																																																													
0	1	0	0																																																																																													
M115	⋯	M100																																																																																														
b15	b8	b7	b0																																																																																													
1	2	3	4																																																																																													
D0																																																																																																
b15	b8	b7	b0																																																																																													
0	0	0	2																																																																																													
D1																																																																																																
b15	b8	b7	b0																																																																																													
1	D	E	F																																																																																													
D2																																																																																																
b15	b8	b7	b0																																																																																													
0	1	0	0																																																																																													
M102	M103	M100	M101																																																																																													
b15	b8	b7	b0																																																																																													
1	1	0	0																																																																																													
M106	M107	M104	M105																																																																																													

■动作参数

变量名	名称	数据类型	有效范围	内容	默认值
pbi_uRequestModuleIO	请求目标模块I/O编号	字[无符号]/位串[16位]	03E0H~03E3H、03FFH	指定访问目标的模块。 <ul style="list-style-type: none"> • 03FFH: 本站、管理CPU • 03E0H: 多CPU1号机 • 03E1H: 多CPU2号机 • 03E2H: 多CPU3号机 • 03E3H: 多CPU4号机 	03FFH
pbi_uResendCountMax	最多重新发送次数	字[无符号]/位串[16位]	0~15	设置在pbi_uMonitorTime(到达监视时间)中指定的监视时间内未完成的情况下重新发送的次数。 <ul style="list-style-type: none"> • 0次~15次 	5
pbi_uMonitorTime	到达监视时间	字[无符号]/位串[16位]	0、1~32767	指定处理完成为止的监视时间。在监视时间内未完成的情况下将重新发送直至达到pbi_uResendCountMax(最多重新发送次数)中指定的次数为止。 <ul style="list-style-type: none"> • 0: 10秒 • 1~32767: 1~32767秒 	0

■公开变量

变量名	名称	数据类型	内容	默认值
pbo_uResendCount	重新发送次数	字[无符号]/位串[16位]	存储重新发送的进行次数(结果)。 存储至检测出异常而中断重新发送为止的重新发送的进行次数(结果)。	0
pbo_u4ErrTime	异常发生时间	字[无符号]/位串[16位](0..3)	存储异常发生时的时钟数据(仅限异常时)。 第1字 <ul style="list-style-type: none"> • 高位8位: 月(01H~12H) • 低位8位: 年(00H~99H) 公历低2位 第2字 <ul style="list-style-type: none"> • 高位8位: 时(00H~23H) • 低位8位: 日(01H~31H) 第3字 <ul style="list-style-type: none"> • 高位8位: 秒(00H~59H) • 低位8位: 分(00H~59H) 第4字 <ul style="list-style-type: none"> • 高位8位: 年(00H~99H) 公历高2位 • 低位8位: 星期(00H(日)~06H(六)) 	0
pbo_uErrIP_Address_3rd_4th	异常检测站IP地址(第3、4八位字节)	字[无符号]/位串[16位](0..1)	存储检测出异常的站的IP地址(第3八位字节、4八位字节)。*1 例: 192.168.1.2的情况下 0102h	0
pbo_uErrIP_Address_1st_2nd	异常检测站IP地址(第1、2八位字节)	字[无符号]/位串[16位](0..1)	存储检测出异常的站的IP地址(第1八位字节、2八位字节)。*1 例: 192.168.1.2的情况下 COA8h	0

*1 仅在专用指令异常完成时存储值。

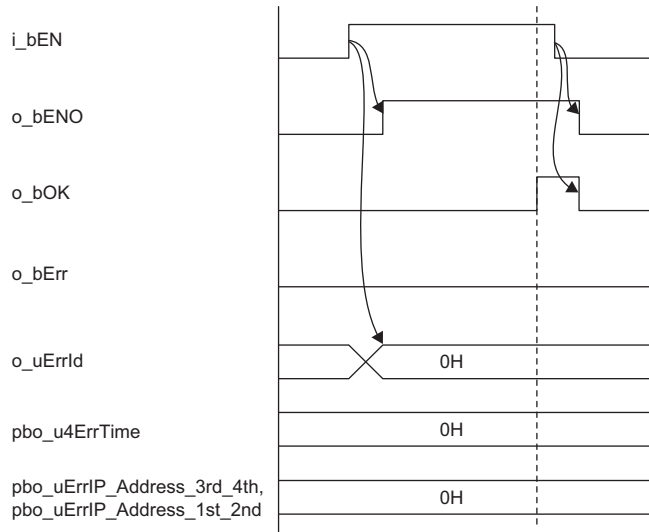
在异常检测设备IP地址(第3、4八位字节)和异常检测设备IP地址(第1、2八位字节)存储输入参数的对象站地址中设置的值。

项目	内容	
对象设备	对象模块	MELSEC MX控制器 (CC-Link IE TSN用端口)
	对象CPU	MELSEC MX控制器
	对象工程工具	GX Works3
使用语言	梯形图	
标签使用量	标签：1076点 锁存标签：0点 程序中内置的标签使用量根据使用的控制器，参数中指定的软件及GX Works3的选项设置而不同。有关GX Works3的选项设置，请参阅下述内容。 GX Works3 操作手册	
功能说明	<ul style="list-style-type: none"> 通过i_bEN(执行指令)的ON，读取SLMP对应设备的软元件数据。 本FB指定对象设备的IP地址后执行。 本FB使用SLMP的Read指令(指令：0401)。SLMP的指令报文为二进制代码。(SLMP参考手册) 	
FB编译方式	宏型	
FB动作	脉冲型(多个扫描中执行型)	

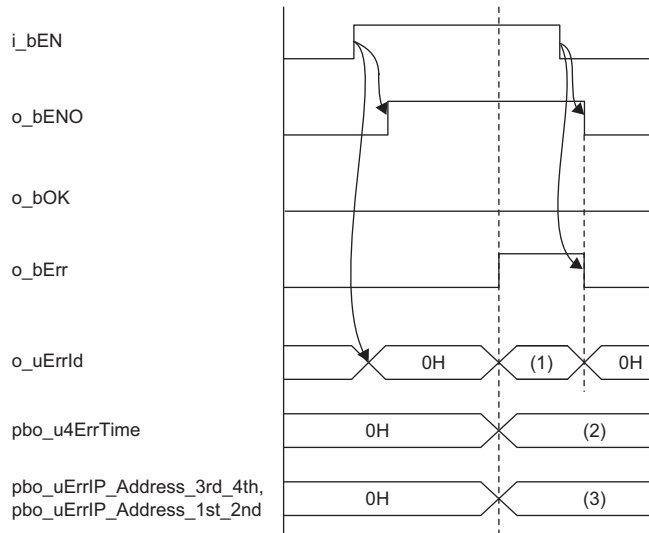
项目	内容
----	----

输入输出信号的动作

• 正常完成的情况下

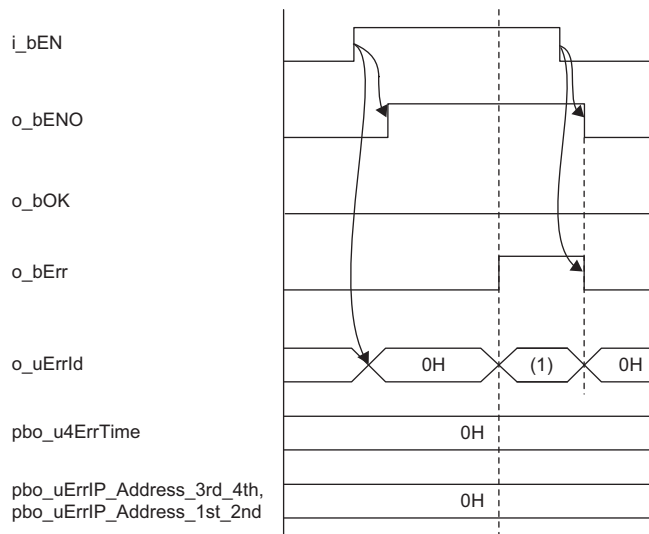


• 异常完成的情况下(专用指令异常完成时)



- (1) 错误代码
- (2) 异常发生时间
- (3) 异常检测设备IP地址

• 异常完成的情况下(专用指令正常完成/结束代码异常时)



- (1) 错误代码

项目	内容
注意事项	<ul style="list-style-type: none"> • 本FB不包含从错误恢复的处理。关于错误处理，应根据系统或请求动作，另行创建。 • 本FB使用SLMPSND指令。SLMPSND指令，即使对象设备返回了异常响应的情况下也将变为正常完成。因此，本FB除了专用指令的完成状态外，还通过响应帧的结束代码，判定正常完成/异常完成。通过结束代码判定为异常完成时，应确认o_uErrId(错误代码)中是否存储了结束代码，以及pbo_u4ErrTime(异常发生时间)与pbo_uErrIP_Address_3rd_4th(异常检测设备IP地址(第3、4八位字节))、pbo_uErrIP_Address_1st_2nd(异常检测设备IP地址(第1、2八位字节))中是否存储了0(初始值)。关于异常原因及处置方法，请参阅所使用的SLMP对应设备的手册。(见SLMP参考手册) • 应在o_bOk(正常完成)或者o_bErr(异常完成)变为ON后，将i_bEN(执行指令)置为OFF。当i_bEN(执行指令)变为OFF，o_bOk(正常完成)和o_bErr(异常完成)也变为OFF，o_uErrId(错误代码)被清除为0。 • 本FB无法通过SLMP的扩展指定读取访问的软元件(链接直接软元件等)。 • 本FB中不能将其他网络的站作为对象站。 • 对于设置了远程口令的对象设备的端口执行本FB时，应先进行远程口令的解锁处理后再执行。对于设置了远程口令的对象设备的端口执行了本FB时，将变为错误。 • 对象站需支持SLMP指令的“Read(指令: 0401)”。 • 本FB仅以二进制代码的通信为对象(不能进行ASCII代码的通信)。 • 本FB通过UDP进行通信。对象设备也应置为相同设置。 • 本FB使用各程序的标签初始值。控制器的引导运行中，通过引导文件设置指定使用了本FB的程序文件时，也应同样通过引导文件设置指定各程序的标签初始值文件。此外，发生了图1923页 错误代码中未记载的错误代码时，可能是引导文件设置中未指定各程序的标签初始值文件。应在引导文件设置中指定各程序的标签初始值文件。 • 执行了本FB时，有可能发生控制器的运算异常。此时，应确认事件履历运算异常的详细信息，重新修改该模块FB的输入参数。

■ 错误代码

错误代码	内容	处理方法
100H	参数的软元件点数(i_uDevicePoints)设置了超出范围的值。	应将软元件点数(i_uDevicePoints)的范围修改为范围内。
100H以外	应确认SLMPSND指令的错误代码并进行处置。 图1146页 G(P). SLMPSND、J(P). SLMPSND、SP. SLMPSND	

M+MXF-TSN_SLMP_DeviceWrite_IP

通过IP地址指定将软元件数据写入至SLMP对应设备。

对象设备需要支持SLMP指令(Device Write)。

梯形图	FBD/LD

ST

```

M_MXF_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?
);

```

■输入参数

变量名	名称	数据类型	有效范围	内容															
i_bEN	执行指令	位	—	ON: 启动FB。 OFF: 不启动FB。															
i_stModule	模块标签	结构体	—	指定执行FB的控制器的模块标签。 (例: MX_TSN)															
i_u2IP_Address	对象设备IP地址	字[无符号]/位串 [16位] (0..1)	0.0.0.1~ 253.255.255.254(0000001H~ DFFFFFFEH)	指定对象站的IP地址。 第1字指定第3~4八位字节, 第2字指定第1~2八位字节。 但是, 不能设置第4八位字节为0或者为255(FH)。 <table border="1" 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="text-align: center;">(3)</td> <td style="text-align: center;">(4)</td> <td colspan="2"></td> </tr> <tr> <td>+1</td> <td style="text-align: center;">(1)</td> <td style="text-align: center;">(2)</td> <td colspan="2"></td> </tr> </table> (1)~(4): 表示IP地址的八位字节。		b15	b8	b7	b0	+0	(3)	(4)			+1	(1)	(2)		
	b15	b8	b7	b0															
+0	(3)	(4)																	
+1	(1)	(2)																	
i_uSubCommand	子指令	字[无符号]/位串 [16位]	—	指定写入单位及软元件的指定方法。 ■第0位: 写入单位 0: 字单位 1: 位单位 ■写入软元件的指定方法 0: 以2位指定软元件代码, 以6位指定起始软元件编号 (MELSEC-Q/L系列用) 1: 以4位指定软元件代码, 以8位指定起始软元件编号 (MELSEC iQ-R系列用)															
i_uDeviceCode	软元件代码*1	字[无符号]/位串 [16位]	—	以二进制代码指定写入软元件的软元件代码。 • 子指令的第1位为0时: 2位 • 子指令的第1位为1时: 4位															
i_u2DeviceNo	起始软元件编号	字[无符号]/位串 [16位] (0..1)	—	以二进制代码指定写入软元件的起始软元件编号。 • 子指令的第1位为0时: 6位 • 子指令的第1位为1时: 8位															
i_uDevicePoints	软元件点数	字[无符号]/位串 [16位]	—	以二进制代码指定写入软元件的软元件点数。 ■子指令的第0位为0时 • 1~960 ■子指令的第0位为1时: 1~3972 • 本站使用通道1~9: 1~960 • 本站使用通道10~17: 1~3960															

■输出参数

变量名	名称	数据类型	内容	默认值
o_bENO	执行状态	位	ON: 执行指令ON OFF: 执行指令OFF	OFF
o_bOK	正常完成	位	ON时, 表示FB的处理已正常完成。	OFF
o_bErr	异常完成	位	ON时, 表示FB的处理已异常完成。	OFF
o_uErrId	错误代码	字[无符号]/位串[16位]	异常完成时将存储错误代码。	0

■动作参数

变量名	名称	数据类型	有效范围	内容	默认值
pbi_uRequestModuleIO	请求目标模块I/O编号	字[无符号]/位串[16位]	03E0H~03E3H、 03FFH	指定访问目标的模块。 • 03FFH: 本站、管理CPU • 03E0H: 多CPU1号机 • 03E1H: 多CPU2号机 • 03E2H: 多CPU3号机 • 03E3H: 多CPU4号机	03FFH
pbi_uResendCountMax	最多重新发送次数	字[无符号]/位串[16位]	0~15	设置在pbi_uMonitorTime(到达监视时间)中指定的监视时间内未完成的情况下重新发送的次数。 • 0次~15次	5
pbi_uMonitorTime	到达监视时间	字[无符号]/位串[16位]	0、1~32767	指定处理完成为止的监视时间。在监视时间内未完成的情况下将重新发送直至达到pbi_uResendCountMax(最多重新发送次数)中指定的次数为止。 • 0: 10秒 • 1~32767: 1~32767秒	0

■公开变量

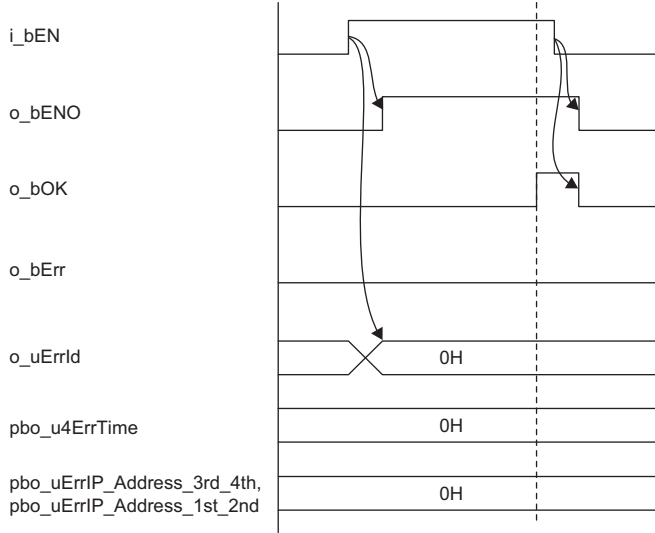
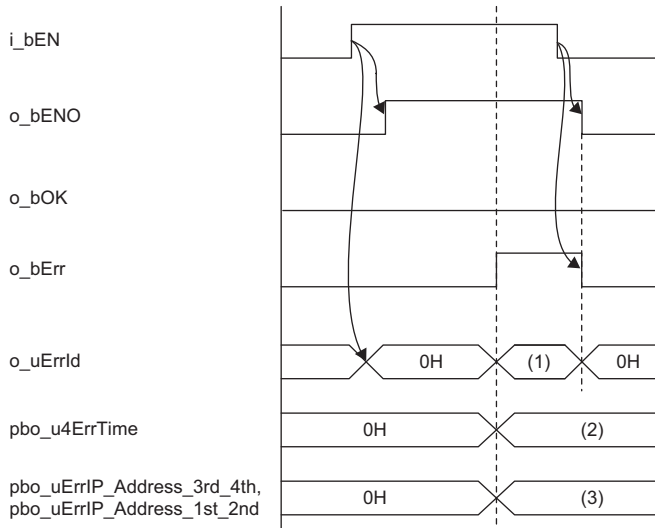
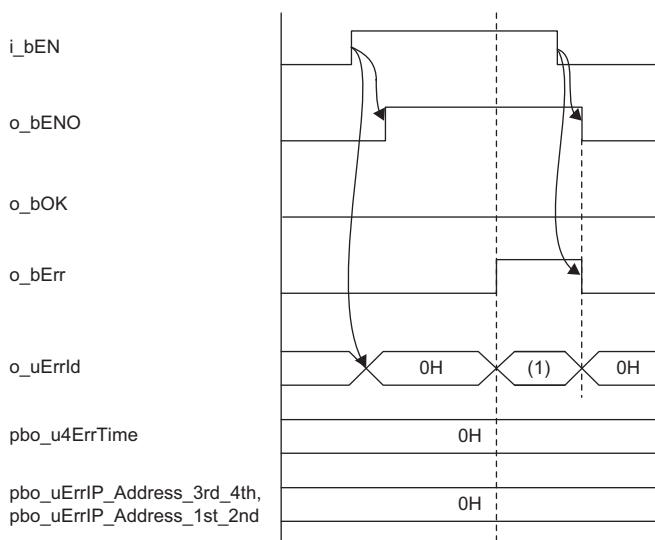
变量名	名称	数据类型	内容	默认值
pbo_uResendCount	重新发送次数	字[无符号]/位串[16位]	存储重新发送的进行次数(结果)。	0
pbo_u4ErrTime	异常发生时间	字[无符号]/位串[16位](0..3)	存储异常发生时的时钟数据(仅限异常时)。 第1字 • 高位8位: 月(01H~12H) • 低位8位: 年(00H~99H) 公历低2位 第2字 • 高位8位: 时(00H~23H) • 低位8位: 日(01H~31H) 第3字 • 高位8位: 秒(00H~59H) • 低位8位: 分(00H~59H) 第4字 • 高位8位: 年(00H~99H) 公历高2位 • 低位8位: 星期(00H(日)~06H(六))	0
pbo_uErrIP_Address_3rd_4th	异常检测站IP地址(第3、4八位字节)	字[无符号]/位串[16位](0..1)	存储检测出异常的站的IP地址(第3八位字节、4八位字节)。*1 例: 192.168.1.2的情况下 0102h	0
pbo_uErrIP_Address_1st_2nd	异常检测站IP地址(第1、2八位字节)	字[无符号]/位串[16位](0..1)	存储检测出异常的站的IP地址(第1八位字节、2八位字节)。*1 例: 192.168.1.2的情况下 C0A8h	0

*1 仅在专用指令异常完成时存储值。

在异常检测设备IP地址(第3、4八位字节)和异常检测设备IP地址(第1、2八位字节)存储输入参数的对象站地址中设置的值。

功能

项目	内容	
对象设备	对象模块	MELSEC MX控制器 (CC-Link IE TSN用端口)
	对象CPU	MELSEC MX控制器
	对象工程工具	GX Works3
使用语言	梯形图	
标签使用量	标签：1076点 锁存标签：0点 程序中内置的标签使用量根据使用的控制器，参数中指定的软元件及GX Works3的选项设置而不同。有关GX Works3的选项设置，请参阅下述内容。 GX Works3 操作手册	
功能说明	<ul style="list-style-type: none"> 通过i_bEN(执行指令)的ON，写入SLMP对应设备的软元件数据。 本FB指定对象设备的IP地址后执行。 本FB使用SLMP的Write指令(指令：1401)。SLMP的指令报文为二进制代码。(SLMP参考手册) 	
FB编译方式	宏型	
FB动作	脉冲型(多个扫描中执行型)	

项目	内容
输入输出信号的动作	<p>• 正常完成的情况下</p>  <p>• 异常完成的情况下(专用指令异常完成时)</p>  <p>(1) 错误代码 (2) 异常发生时间 (3) 异常检测设备IP地址</p> <p>• 异常完成的情况下(专用指令正常完成/结束代码异常时)</p>  <p>(1) 错误代码</p>

项目	内容
注意事项	<ul style="list-style-type: none"> • 本FB不包含从错误恢复的处理。关于错误处理，应根据系统或请求动作，另行创建。 • 本FB使用SLMPSND指令。SLMPSND指令，即使对象设备返回了异常响应的情况下也将变为正常完成。因此，本FB除了专用指令的完成状态外，还通过响应帧的结束代码，判定正常完成/异常完成。通过结束代码判定为异常完成时，应确认o_uErrId(错误代码)中是否存储了结束代码，以及pbo_u4ErrTime(异常发生时间)与pbo_uErrIP_Address_3rd_4th(异常检测设备IP地址(第3、4八位字节))、pbo_uErrIP_Address_1st_2nd(异常检测设备IP地址(第1、2八位字节))中是否存储了0(初始值)。关于异常原因及处置方法，请参阅所使用的SLMP对应设备的手册。(见SLMP参考手册) • 应在o_bOk(正常完成)或者o_bErr(异常完成)变为ON后，将i_bEN(执行指令)置为OFF。当i_bEN(执行指令)变为OFF，o_bOk(正常完成)和o_bErr(异常完成)也变为OFF，o_uErrId(错误代码)被清除为0。 • 本FB无法通过SLMP的扩展指定写入访问的软元件(链接直接软元件等)。 • 本FB中不能将其他网络的站作为对象站。 • 对于设置了远程口令的对象设备的端口执行本FB时，应先进行远程口令的解锁处理后再执行。对于设置了远程口令的对象设备的端口执行了本FB时，将变为错误。 • 对象站需支持SLMP指令的“Write(指令: 1401)”。 • 本FB仅以二进制代码的通信为对象(不能进行ASCII代码的通信)。 • 本FB通过UDP进行通信。对象设备也应置为相同设置。 • 本FB使用各程序的标签初始值。控制器的引导运行中，通过引导文件设置指定使用了本FB的程序文件时，也应同样通过引导文件设置指定各程序的标签初始值文件。此外，发生了图1930页 错误代码中未记载的错误代码时，可能是引导文件设置中未指定各程序的标签初始值文件。应在引导文件设置中指定各程序的标签初始值文件。 • 执行了本FB时，有可能发生控制器的运算异常。此时，应确认事件履历运算异常的详细信息，重新修改该模块FB的输入参数。

■ 错误代码

错误代码	内容	处理方法
100H	参数的软元件点数(i_uDevicePoints)设置了超出范围的值。	应将软元件点数(i_uDevicePoints)的范围修改为范围内。
100H以外	应确认SLMPSND指令的错误代码并进行处置。 图1146页 G(P). SLMPSND、J(P). SLMPSND、SP. SLMPSND	

要点 🔍

本章的FB为控制器的模块FB。请使用工程工具的部件选择窗口的[Module (模块)]⇒[Module FB (模块FB)]

49.1 远程复位

M+MXF-TSN_RemoteReset2_IP

通过IP地址指定，向对象站发送SLMP请求的远程复位。

梯形图	FBD/LD

ST

```

M_MXF_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?
);
    
```

设置数据

■输入参数

变量名	名称	数据类型	有效范围	内容															
i_bEN	执行指令	位	—	ON: 启动FB。 OFF: 不启动FB。															
i_stModule	模块标签	结构体	—	指定执行FB的控制器的模块标签。 (例: MX_TSN)															
i_u2TargetAddress	对象设备IP地址	字[无符号]/位串[16位](0..1)	0. 0. 0. 1~ 253. 255. 255. 254 (00000001H~ DFFFFFFFEH)	指定对象设备的IP地址。通过标签指定的情况下, 数据类型应使用数组。 • 00000001H~DFFFFFFFEH 第4八位字节应指定1~254 (FEH)。 <table style="margin-left: 40px;"> <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 style="border: 1px solid black; text-align: center;">4</td> <td colspan="2"></td> </tr> <tr> <td>+1</td> <td style="border: 1px solid black; text-align: center;">1</td> <td style="border: 1px solid black; text-align: center;">2</td> <td colspan="2"></td> </tr> </table> 1~4: 表示IP地址的八位字节。		b15	b8	b7	b0	+0	3	4			+1	1	2		
	b15	b8	b7	b0															
+0	3	4																	
+1	1	2																	
i_uTarget_Port_No	对象设备端口编号	字[无符号]/位串[16位]	1~65534	指定对象设备的UDP端口编号。 关于指定的端口编号, 应确认对象设备的手册。															
i_uChannel	本站使用通道	字[无符号]/位串[16位]	1~17	指定本站使用的通道。*1															

*1 通过本FB使用无序列号帧进行通信时, 将本站使用的通道指定为1。指定为2~9时, 将使用有序列号帧进行通信。指定为10~17时, 将使用站号扩展帧进行通信。

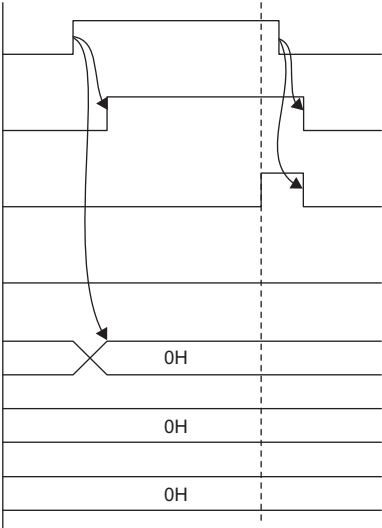
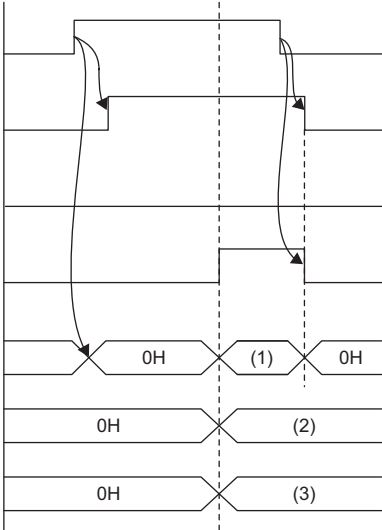
■输出参数

变量名	名称	数据类型	内容	默认值
o_bENO	执行状态	位	ON: 执行指令ON OFF: 执行指令OFF	OFF
o_bOK	正常完成	位	ON时, 表示FB的处理已正常完成。	OFF
o_bErr	异常完成	位	ON时, 表示FB的处理已异常完成。	OFF
o_uErrId	错误代码	字[无符号]/位串[16位]	异常完成时将存储错误代码。	0

■公开变量

变量名	名称	数据类型	内容	默认值
pbo_u4ErrTime	异常发生时间	字[无符号]/位串[16位](0..3)	存储异常发生时的时钟数据(仅限异常时)。*1 第1字 • 高位8位: 月(01H~12H) • 低位8位: 年(00H~99H) 公历低2位 第2字 • 高位8位: 时(00H~23H) • 低位8位: 日(01H~31H) 第3字 • 高位8位: 秒(00H~59H) • 低位8位: 分(00H~59H) 第4字 • 高位8位: 年(00H~99H) 公历高2位 • 低位8位: 星期(00H(日)~06H(六))	0
pbo_uErrIP_Address_3rd_4th	异常检测设备IP地址(第3、4八位字节)	字[无符号]/位串[16位]	存储检测出异常的站的IP地址(第3八位字节、4八位字节)。*1 例: 192.168.1.2的情况下 0102h	0
pbo_uErrIP_Address_1st_2nd	异常检测设备IP地址(第1、2八位字节)	字[无符号]/位串[16位]	存储检测出异常的站的IP地址(第1八位字节、2八位字节)。*1 例: 192.168.1.2的情况下 C0A8h	0

*1 仅在专用指令异常完成时存储值。
在异常检测设备IP地址(第3、4八位字节)和异常检测设备IP地址(第1、2八位字节)存储输入参数的对象站地址中设置的值。

项目	内容	
对象设备	对象模块	MELSEC MX控制器 (CC-Link IE TSN用端口)
	对象CPU	MELSEC MX控制器
	对象工程工具	GX Works3
使用语言	梯形图	
标签使用量	标签：72点 锁存标签：0点 程序中内置的标签使用量根据使用的控制器，参数中指定的软件件及GX Works3的选择设置而不同。有关GX Works3的选择设置，请参阅下述内容。 GX Works3 操作手册	
功能说明	通过i_bEN(执行指令)的ON, 向对象站发送SLMP请求的远程复位。	
FB编译方式	宏型	
FB动作	脉冲型(多个扫描执行型)	
FB_EN的输入条件	无	
输入输出信号的动作	<ul style="list-style-type: none"> 正常完成的情况下  <ul style="list-style-type: none"> 异常完成的情况下(专用指令异常完成时)  <p>(1) 错误代码 (2) 异常发生时间 (3) 异常检测站地址</p>	

项目	内容
注意事项	<ul style="list-style-type: none"> • 本FB不包含从错误恢复的处理。关于错误处理，应根据系统或请求动作，另行创建。 • 本FB使用SLMPSND指令。SLMPSND指令，即使对象设备返回了异常响应的情况下也将正常完成。 • SLMPSND指令正常完成时，不在公开变量的异常发生时间、pbo_uErrIP_Address_3rd_4th(异常检测设备IP地址(第、3、4八位字节))和pbo_uErrIP_Address_1st_2nd(异常检测设备IP地址(第1、2八位字节))中存储值。^{*1} • 应在o_bOK(正常完成)或者o_bErr(异常完成)变为ON后，将i_bEN(执行指令)置为OFF。当i_bEN(执行指令)变为OFF，o_bOK(正常完成)和o_bErr(异常完成)也变为OFF，o_uErrId(错误代码)被清除为0。 • 当远程复位的请求发送正常完成时，o_bOK(正常完成)变为ON。实际上对象站是否远程复位要取决于对象站的状态。 • 本FB中不能将其他网络的站作为对象站。 • 对象设备需要对应SLMP指令的“Remote Reset(1006H)”。 • 本FB通过UDP进行通信。对象设备也应置为相同设置。

*1 在异常发生时间、异常检测设备IP地址(第3、4八位字节)和异常检测设备IP地址(第1、2八位字节)中存储0(初始值)时，应确认所使用的SLMP对应设备的手册并进行处置。

出错

本FB发生的错误代码与SLMPSND指令相同。

☞ 1146页 G(P).SLMPSND、J(P).SLMPSND、SP.SLMPSND

附1 指令处理时间

各指令的处理时间如下所示。

处理时间根据源和目标的内容以及程序相应的处理优化而变动，因此参考时应以下表中的值为大致基准。

指令的分类			指令名	条件	处理时间[ns]	
大分类	中分类	小分类			最小	
顺控指令	触点指令	—	LD	位软元件	1.25	
				位型标签	1.25	
	输出指令	—	OUT	位软元件	34.40	
				位型标签	1.04	
				OUT T	—	68.42
				OUT LT	—	102.46
基本指令	数据传送指令	—	MOV	—	2.71	
			BMOV	n=1	533.02	
				n=1024	1227.24	
			FMOV	n=1	324.21	
				n=1024	1035.68	
	算术运算指令	递增	INC	—	5.27	
		四则运算指令(整数)	+	—	1.36	
			-	—	1.36	
			*	—	1.36	
			/	—	1.36	

指令的分类			指令名	条件	处理时间[ns]
大分类	中分类	小分类			最小
应用指令	浮点指令	四则运算指令(单精度实数)	E+	$s1=2^{126}, s2=2^{127}$	118.2
			E-	$s1=2^{126}, s2=2^{127}$	118.43
			E*	$s1=2^{63}, s2=2^{64}$	116.89
			E/	$s1=2^{126}, s2=2^{127}$	155.7
		四则运算指令(双精度实数)	ED+	$s1=2^{1022}, s2=2^{1023}$	110.53
			ED-	$s1=2^{1022}, s2=2^{1023}$	110.71
			ED*	$s1=2^{511}, s2=2^{512}$	109.45
			ED/	$s1=2^{1022}, s2=2^{1023}$	124.91
		三角函数(单精度实数)	SIN	$s=E0.7853982(45度)$	308.08
		三角函数(双精度实数)	SIND	$s=E0.7853982(45度)$	234.33
		实数数据传送(单精度实数)	EMOV	$s=$ 内部软元件、 $d=$ 内部软元件	2.73
				$s=E1.23,$ $d=$ 内部软元件	1.36
		实数数据传送(双精度实数)	EDMOV	$s=$ 内部软元件、 $d=$ 内部软元件	2.73
				$s=E1.23,$ $d=$ 内部软元件	2.5
	PID控制指令	不完全微分	S.PIDINIT	1环路	3845.1
				32环路	7456.2
			S.PIDCONT	1环路	677.4
				32环路	11362.6
	结构化指令	重复指令	FOR	—	2.72
			NEXT	—	1.06
		子程序调用指令	CALL	局部指针	1514.5
				全局指针	2175.8
			RET	返回至本程序	736.7
				返回至其他程序	1058.3
模块访问指令	—	TO	$n=1$	10866	
			$n=1024$	715892	
		FROM	$n=1$	11697	
			$n=1024$	694319	
专用指令	内置以太网用指令	SLMP报文发送	GP.SLMPSND	Read(指令: 0401H)(字单位读取): 读取点数=1点	2687
	智能功能模块用指令	其他站可编程控制器的数据读取	GP.READ	—	121173
FB/FUN调用	FB调用	FB调用	—	局部FB	2793
			—	全局FB	3042
	FUN调用	FUN调用	—	—	2073
	内嵌ST调用	内嵌ST调用	—	—	856.77
ST控制语法	条件分支	—	IF	—	2.61
	重复	—	CASE	—	3.07
			ST_FOR	—	11.26

索引

符号

-	.92
*	.92
**	.92
/	.92
&	.92
+	.92
<	.92
<=	.92
<>	.92
=	.92
>	.92
>=	.92

数字

16位数据(字数据)	.32
32位数据(双字数据)	.34

A

AND	.92
ASCII	103, 115

B

保留字	.91
-----	-----

C

CASE	.97
程序	62, 74
程序块	.59

D

代入语句	.94
单精度实数数据	.37

E

EN	61, 73
ENO	61, 73
EXIT	.97

F

FB调用语句	.96
FB文件	65, 74
FOR...DO	.97, 101
FUN文件	60, 62

G

功能代码和功能参数	.1174
功能块(FB)	58, 64

H

函数(FUN)	58, 60
函数调用语句	.96

缓冲存储器	.22
-------	-----

I

IF THEN	.97
IF...ELSE	.97
IF...ELSIF	.97
iQSS对应设备	.24

J

基本指令	.251
------	------

L

类型指定	104, 115
类型转换	.95, 108
链接软元件	.22

M

MOD	.92
目标(d)	.26

N

NOT	.92
内部变量	.66
内置以太网功能用指令	.1110

O

OR	.92
----	-----

P

PID控制指令	.1016
PID指令的种类	.1000

R

REPEAT...UNTIL	.97
RETURN	.96
软元件数/传送数/数据数/字符串数等的数值(n)	.27

S

STRING	103, 115
设备站	.22
声明	.89
实例	.67
实数数据(浮点数据)	.37
输出变量	61, 65
数据链接	.24
输入变量	61, 65
输入输出变量	.65
双精度实数数据	.38
顺控程序指令	.200

T

梯形图语言	84
通信执行状态输出软元件	1175
通用FB	1343
通用函数	1179

U

Unicode	103
-------------------	-----

W

WHILE...DO	97
WSTRING	103, 115
外部变量	66
网络FB	1891
位数据	31

X

XOR	92
---------------	----

Y

移位JIS	103, 115
应用指令	515
源(s)	26
运动控制FB	1371

Z

站子ID编号	24
指令的构成	26
执行条件	42
中断程序	59
主程序	59
注解	89
主站	22
子程序	59
子程序型FB	65
字符串	103, 115
字符串[Unicode]	103, 115
字符串数据	40

指令索引

符号

-(P) (_U)	265, 267
*(P) (_U)	277
/(P) (_U)	279
+(P) (_U)	261, 263
\$+(P)	702, 704
\$MOV	706
\$MOV(P)_WS	708

A

ABS(_E)	1264
ACOS(_E)	1275
ACOS(P)	848
ACOSD(P)	860
ADD_TIME(_E)	1335
ADD(_E)	1277
ADPRW	1173
ADRSET(P)	934
ANB	208
AND	200
AND_EQ(_U)	251
AND_GE(_U)	251
AND_GT(_U)	251
AND_LE(_U)	251
AND_LT(_U)	251
AND_NE(_U)	251
AND(_E)	1300
AND<(_U)	251
AND<=(_U)	251
AND<>(_U)	251
AND=(_U)	251
AND>(_U)	251
AND>=(_U)	251
AND\$<	699
AND\$<=	699
AND\$<>	699
AND\$=	699
AND\$>	699
AND\$>=	699
ANDD_EQ(_U)	253
ANDD_GE(_U)	253
ANDD_GT(_U)	253
ANDD_LE(_U)	253
ANDD_LT(_U)	253
ANDD_NE(_U)	253
ANDD<(_U)	253
ANDD<=(_U)	253
ANDD<>(_U)	253
ANDD=(_U)	253
ANDD>(_U)	253
ANDD>=(_U)	253
ANDDT_EQ	960
ANDDT_GE	960
ANDDT_GT	960
ANDDT_LE	960
ANDDT_LT	960
ANDDT_NE	960
ANDDT<	960
ANDDT<=	960
ANDDT<>	960

ANDDT=	960
ANDDT>	960
ANDDT>=	960
ANDE_EQ	788
ANDE_GE	788
ANDE_GT	788
ANDE_LE	788
ANDE_LT	788
ANDE_NE	788
ANDE<	788
ANDE<=	788
ANDE<>	788
ANDE=	788
ANDE>	788
ANDE>=	788
ANDED_EQ	790
ANDED_GE	790
ANDED_GT	790
ANDED_LE	790
ANDED_LT	790
ANDED_NE	790
ANDED<	790
ANDED<=	790
ANDED<>	790
ANDED=	790
ANDED>	790
ANDED>=	790
ANDF	202
ANDFI	205
ANDP	202
ANDPI	205
ANDSTRING_EQ	699
ANDSTRING_GE	699
ANDSTRING_GT	699
ANDSTRING_LE	699
ANDSTRING_LT	699
ANDSTRING_NE	699
ANDTM_EQ	964
ANDTM_GE	964
ANDTM_GT	964
ANDTM_LE	964
ANDTM_LT	964
ANDTM_NE	964
ANDTM<	964
ANDTM<=	964
ANDTM<>	964
ANDTM=	964
ANDTM>	964
ANDTM>=	964
ANI	200
ASC2INT(P)	462
ASIN(_E)	1274
ASIN(P)	846
ASIND(P)	858
ATAN(_E)	1276
ATAN(P)	850
ATAND(P)	862

B

B-(P)	288, 289
B*(P)	297

DBMINUS (P)	295
DBMULTI (P)	301
DBPLUS (P)	292
DCML (P)	491
DDABCD (P)	453
DDABIN (P) (_U)	441
DDEC (P) (_U)	318
DDIVISION (P) (_U)	283
DDRVA	1077
DDRVI	1072
DDSZR	1054
DDVIT	1061
DEC (P) (_U)	316
DECO (P)	468
DEG (P)	878
DEGD (P)	882
DELETE (_E)	1328
DELTA (P)	238
DFMOV (P)	502
DFMOVL (P)	504
DFROM (P)	980
DFROMD (P)	988
DGBIN (P) (_U)	432
DGRY (P) (_U)	428
DHABIN (P)	447
DHCMOV (P)	1093
DHIOEN (P)	1051
DI	520, 523
DI_1	523
DINC (P) (_U)	317
DINT_TO_BCD (_E)	1212
DINT_TO_BITARR (_E)	1255
DINT_TO_BOOL (_E)	1208
DINT_TO_DWORD (_E)	1210
DINT_TO_INT (_E)	1211
DINT_TO_LREAL (_E)	1215
DINT_TO_REAL (_E)	1214
DINT_TO_STRING (_E)	1217
DINT_TO_TIME (_E)	1216
DINT_TO_WORD (_E)	1209
DINT2DBL (P)	827
DINT2FLT (P)	819
DINT2INT (P)	414
DINT2UDINT (P)	418
DINT2UINT (P)	416
DIS (P)	474
DITRG	1084
DIV_TIME (_E)	1341
DIV (_E)	1285
DIVISION (P) (_U)	279
DLIMIT (P) (_U)	643
DMAX (P) (_U)	675
DMEAN (P) (_U)	691
DMIN (P) (_U)	679
DMINUS (P) (_U)	275
DMOV (P)	488
DMULTI (P) (_U)	281
DNEG (P)	467
DOR (P)	330, 331
DPLSV	1067
DPLUS (P) (_U)	271
DPWMH	1100
DRCL (P)	577
DRCR (P)	575
DROL (P)	577
DROR (P)	575

DRVA	1077
DRVI	1072
DRVTBL	1082
DSCL (P) (_U)	656
DSCL2 (P) (_U)	661
DSERDATA (P)	665
DSERMM (P)	669
DSFL (P)	368
DSFR (P)	366
DSORTD (_U)	683
DSPDH	1107
DSQRT (P)	694
DSTR (P) (_U)	730
DSUM (P)	672
DSWAP (P)	511
DSZR	1054
DTEST (P)	355
DTO (P)	983
DTOD (P)	992
DUTY	975
DVAL (P) (_U)	459
DVIT	1061
DWORD_TO_BOOL (_E)	1192
DWORD_TO_DINT (_E)	1195
DWORD_TO_INT (_E)	1194
DWORD_TO_STRING (_E)	1197
DWORD_TO_TIME (_E)	1196
DWORD_TO_WORD (_E)	1193
DWSUM (P) (_U)	687
DXCH (P)	507
DXNR (P)	345, 346
DXOR (P)	338, 339
DZONE (P) (_U)	651

E

E- (P)	797, 799
E* (P)	809
E/ (P)	811
E+ (P)	793, 795
ECALL (P)	552
ED- (P)	805, 807
ED* (P)	813
ED/ (P)	815
ED+ (P)	801, 803
EDDIVISION (P)	815
EDIVISION (P)	811
EDMAX (P)	910
EDMIN (P)	914
EDMINUS (P)	807
EDMOV (P)	839
EDMULTI (P)	813
EDNEG (P)	837
EDPLUS (P)	803
EDSQRT (P)	886
EFCALL (P)	559
EGF	211
EGP	211
EI	520
EMAX (P)	908
EMIN (P)	912
EMINUS (P)	799
EMOD (P)	464
EMOV (P)	838
EMULTI (P)	809
ENCO (P)	470

END	247
ENEG(P)	836
EPLUS(P)	795
EQ(_E)	1313
EREXP(P)	834
ESQRT(P)	884
ESTR(P)	742
EVAL(P)	830
EXP(_E)	1270
EXP(P)	888
EXPD(P)	890
EXPT(_E)	1289

F

F_TRIG(_E)	1350
FCALL(P)	546
FDEL(P)	587
FEND	246
FF	237
FIFR(P)	579
FIFW(P)	583
FIND(_E)	1333
FINS(P)	585
FLT2DBL(P)	829
FLT2DINT(P)	390
FLT2INT(P)	386
FLT2UDINT(P)	392
FLT2UINT(P)	388
FMOV(P)	498
FMOVL(P)	500
FOR	534
FPOP(P)	581
FROM(P)	980
FROMD(P)	988

G

G_SOCRVS	1119
G.SOCRVS	1119
G(P).SLMPSND	1146
G(P).SLMPSNDC	1155
G(P).SOCRDATA	1136
GBIN(P)(_U)	430
GE(_E)	1313
GET_BIT_OF_INT(_E)	1257
GET_BOOL_ADDR	1263
GET_INT_ADDR	1263
GET_WORD_ADDR	1263
GOEND	519
GP_ECPRTCL	1138
GP_FTPCGET	1168
GP_FTPCPUT	1163
GP_SOCCINF	1126
GP_SOCCLOSE	1114
GP_SOCCSET	1129
GP_SOCOPEN	1110
GP_SOCRVC	1116
GP_SOCRMODE	1131
GP_SOCSND	1122
GP.ECPRTCL	1138
GP.FTPCGET	1168
GP.FTPCPUT	1163
GP.SOCCINF	1126
GP.SOCCLOSE	1114
GP.SOCCSET	1129

GP.SOCOPEN	1110
GP.SOCRVC	1116
GP.SOCRMODE	1131
GP.SOCSND	1122
GRY(P)(_U)	426
GT(_E)	1313
SP.SOCCINF	1126

H

HABIN(P)	445
HCMOV(P)	1090
HIOEN(P)	1048

I

IMASK	528
INC(P)(_U)	315
INSERT(_E)	1326
INSTR(P)	782
INT_TO_BCD(_E)	1202
INT_TO_BITARR(_E)	1254
INT_TO_BOOL(_E)	1198
INT_TO_DINT(_E)	1201
INT_TO_DWORD(_E)	1200
INT_TO_LREAL(_E)	1204
INT_TO_REAL(_E)	1203
INT_TO_STRING(_E)	1206
INT_TO_TIME(_E)	1205
INT_TO_WORD(_E)	1199
INT2ASC(P)	747
INT2DBL(P)	825
INT2DINT(P)	404
INT2FLT(P)	817
INT2UDINT(P)	406
INT2UINT(P)	402
INV	209
IRET	532

J

J(P).SLMPSND	1146
JMP	516

L

LD	200
LD_EQ(_U)	251
LD_GE(_U)	251
LD_GT(_U)	251
LD_LE(_U)	251
LD_LT(_U)	251
LD_NE(_U)	251
LD<(_U)	251
LD<=(_U)	251
LD<>(_U)	251
LD=(_U)	251
LD>(_U)	251
LD>=(_U)	251
LD\$<	699
LD\$<=	699
LD\$<>	699
LD\$=	699
LD\$>	699
LD\$>=	699
LDD_EQ(_U)	253

LDD_GE (_U)	253
LDD_GT (_U)	253
LDD_LE (_U)	253
LDD_LT (_U)	253
LDD_NE (_U)	253
LDD< (_U)	253
LDD<= (_U)	253
LDD<> (_U)	253
LDD= (_U)	253
LDD> (_U)	253
LDD>= (_U)	253
LDDT_EQ	960
LDDT_GE	960
LDDT_GT	960
LDDT_LE	960
LDDT_LT	960
LDDT_NE	960
LDDT<	960
LDDT<=	960
LDDT<>	960
LDDT=	960
LDDT>	960
LDDT>=	960
LDE_EQ	788
LDE_GE	788
LDE_GT	788
LDE_LE	788
LDE_LT	788
LDE_NE	788
LDE<	788
LDE<=	788
LDE<>	788
LDE=	788
LDE>	788
LDE>=	788
LDED_EQ	790
LDED_GE	790
LDED_GT	790
LDED_LE	790
LDED_LT	790
LDED_NE	790
LDED<	790
LDED<=	790
LDED<>	790
LDED=	790
LDED>	790
LDED>=	790
LDF	202
LDFI	205
LDI	200
LDP	202
LDPI	205
LDSTRING_EQ	699
LDSTRING_GE	699
LDSTRING_GT	699
LDSTRING_LE	699
LDSTRING_LT	699
LDSTRING_NE	699
LDTM_EQ	964
LDTM_GE	964
LDTM_GT	964
LDTM_LE	964
LDTM_LT	964
LDTM_NE	964
LDTM<	964
LDTM<=	964

LDTM<>	964
LDTM=	964
LDTM>	964
LDTM>=	964
LE (_E)	1313
LEDR.	697
LEFT (_E)	1319
LEFT (P)	775
LEN (_E)	1317
LEN (P)	771
LIMIT (_E)	1308
LIMIT (P) (_U)	641
LN (_E)	1267
LOG (_E)	1268
LOG (P)	892
LOG10 (P)	904
LOG10D (P)	906
LOGD (P)	894
LREAL_TO_DINT (_E)	1232
LREAL_TO_INT (_E)	1231
LREAL_TO_REAL (_E)	1233
LT (_E)	1313

M

M+MXF-SQ_ConnectionClose.	1895
M+MXF-SQ_ConnectionOpen	1892
M+MXF-SQ_Recv_Socket	1897
M+MXF-SQ_Send_Socket	1900
M+MXF-SQ_SLMP_DeviceRead_IP	1903
M+MXF-SQ_SLMP_DeviceWrite_IP	1910
M+MXF-TSN_RemoteReset2_IP	1931
M+MXF-TSN_SLMP_DeviceRead_IP	1917
M+MXF-TSN_SLMP_DeviceWrite_IP	1924
MAX (_E)	1306
MAX (P) (_U)	673
MC	242
MC_AbortTrigger.	1529
MC_CamIn.	1716
MC_CamTableSelect.	1534
MC_CombineAxes	1752
MC_DigitalCamSwitch.	1571
MC_GearIn	1742
MC_GroupDisable.	1467
MC_GroupEnable	1463
MC_GroupReset	1511
MC_GroupSetOverride.	1553
MC_GroupStop	1591
MC_Home	1577
MC_MoveAbsolute.	1597
MC_MoveRelative.	1610
MC_MoveVelocity.	1629
MC_Power.	1471
MC_ReadParameter	1495
MC_Reset.	1508
MC_SetOverride	1490
MC_SetPosition	1475
MC_Stop	1586
MC_TorqueControl	1637
MC_TouchProbe	1515
MC_WriteParameter.	1502
MCR	242
MCv_AdvancedSync	1817
MCv_AdvCamSetPositionCalc	1568
MCv_AdvPositionPerCycleCalc	1565
MCv_AllPower	1550

MCv_BacklashCompensationFilter 1762
 MCv_ChangeCycle 1542
 MCv_CyclicPosition 1844
 MCv_CyclicTorque 1857
 MCv_CyclicVelocity 1853
 MCv_DirectionFilter 1786
 MCv_GroupMoveWait 1810
 MCv_Jog, MCv. 1619
 MCv_MotionErrorReset. 1558
 MCv_MoveCircularInterpolateAbsolute 1683
 MCv_MoveCircularInterpolateRelative 1699
 MCv_MoveLinearInterpolateAbsolute 1656
 MCv_MoveLinearInterpolateRelative 1670
 MCv_MovePositioningData 1823
 MCv_MoveWait 1803
 MCv_PositionControl 1832
 MCv_ReadProfileData 1863
 MCv_SetTorqueLimit 1482
 MCv_SmoothingFilter 1776
 MCv_SpeedControl 1647
 MCv_SpeedLimitFilter. 1795
 MCv_SyncOperationCycles 1562
 MCv_WriteProfileData. 1877
 MEAN (P) (_U) 689
 MEF 210
 MEP 210
 MID (_E) 1322
 MIDR (P) 777
 MIDW (P) 779
 MIN (_E) 1306
 MIN (P) (_U) 677
 MINUS (P) (_U) 267
 MOD (_E) 1287
 MOV (P) 486
 MOV (P) 512
 MOVE (_E) 1290
 MUL_TIME (_E) 1339
 MUL (_E) 1280
 MULTI (P) (_U) 277
 MUX (_E) 1311

N

NDIS (P) 478
 NE (_E) 1315
 NEG (P) 466
 NEXT 534
 NOPLF 250
 NOT (_E) 1303
 NUNI (P) 480

O

OR 200
 OR_EQ (_U) 251
 OR_GE (_U) 251
 OR_GT (_U) 251
 OR_LE (_U) 251
 OR_LT (_U) 251
 OR_NE (_U) 251
 OR (_E) 1300
 OR< (_U) 251
 OR<= (_U) 251
 OR<> (_U) 251
 OR= (_U) 251
 OR> (_U) 251

OR>= (_U) 251
 OR\$<. 699
 OR\$<= 699
 OR\$<> 699
 OR\$=. 699
 OR\$>. 699
 OR\$>= 699
 ORB 208
 ORD_EQ (_U) 253
 ORD_GE (_U) 253
 ORD_GT (_U) 253
 ORD_LE (_U) 253
 ORD_LT (_U) 253
 ORD_NE (_U) 253
 ORD< (_U) 253
 ORD<= (_U) 253
 ORD<> (_U) 253
 ORD= (_U) 253
 ORD> (_U) 253
 ORD>= (_U) 253
 ORDT_EQ 960
 ORDT_GE 960
 ORDT_GT 960
 ORDT_LE 960
 ORDT_LT 960
 ORDT_NE 960
 ORDT< 960
 ORDT<= 960
 ORDT<> 960
 ORDT= 960
 ORDT> 960
 ORDT>= 960
 ORE_EQ 788
 ORE_GE 788
 ORE_GT 788
 ORE_LE 788
 ORE_LT 788
 ORE_NE 788
 ORE<. 788
 ORE<= 788
 ORE<> 788
 ORE=. 788
 ORE>. 788
 ORE>= 788
 ORED_EQ 790
 ORED_GE 790
 ORED_GT 790
 ORED_LE 790
 ORED_LT 790
 ORED_NE 790
 ORED< 790
 ORED<= 790
 ORED<> 790
 ORED= 790
 ORED> 790
 ORED>= 790
 ORF 202
 ORFI 205
 ORI 200
 ORP 202
 ORPI 205
 ORSTRING_EQ 699
 ORSTRING_GE 699
 ORSTRING_GT 699
 ORSTRING_LE 699
 ORSTRING_LT 699

ORSTRING_NE.	699
ORTM_EQ	964
ORTM_GE	964
ORTM_GT	964
ORTM_LE	964
ORTM_LT	964
ORTM_NE	964
ORTM<	964
ORTM<=	964
ORTM<>	964
ORTM=	964
ORTM>	964
ORTM>=	964
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
OUT_T	215
OUTH.	215
OUTH ST	215
OUTH T	215

P

PID	1012
PIDCONT(P)	1041
PIDINIT(P)	1039
PIDPRMW(P)	1046
PIDRUN(P)	1045
PIDSTOP(P)	1044
PLF	235
PLS	233
PLSV	1067
PLUS(P) (_U)	263
POW(P)	900
POWD(P)	902
PWMH	1095

R

R_TRIG(_E)	1348
RAD(P)	876
RADD(P)	880
RCL(P)	572
RCR(P)	569
REAL_TO_DINT(_E)	1226
REAL_TO_INT(_E)	1225
REAL_TO_LREAL(_E)	1227
REAL_TO_STRING(_E)	1228
REPLACE(_E)	1330
RET	545
RFS(P)	978
RIGHT(_E)	1319
RIGHT(P)	773
RND(P)	916
ROL(_E)	1296
ROL(P)	572
ROR(_E)	1298
ROR(P)	569
RS(_E)	1346
RS2	1176

RSET(P)	928
RST	228
RST F	231

S

S_SOCR CVS	1119
S_SOCR CVS	1119
S(P)_DATEMINUS	973
S(P)_DATEPLUS	971
S(P)_DATERD	969
S(P)_DEVLD	590
S(P)_PIDCONT	1030
S(P)_PIDINIT	1027
S(P)_PIDPRMW	1035
S(P)_PIDRUN	1034
S(P)_PIDSTOP	1033
S(P)_SOCRDATA	1136
S(P).DATE-	973
S(P).DATE+	971
S(P).DATERD	969
S(P).DEVLD	590
S(P).PIDCONT	1030
S(P).PIDINIT	1027
S(P).PIDPRMW	1035
S(P).PIDRUN	1034
S(P).PIDSTOP	1033
S(P).SOCRDATA	1136
SCJ	516
SCL(P) (_U)	653
SCL2(P) (_U)	659
SEC2DATE(P) (_U)	958
SEC2TIME(P)	954
SEG(P)	472
SEL(_E)	1304
SERDATA(P)	663
SERMM(P)	667
SET	226
SET F	229
SET_BIT_OF_INT(_E)	1259
SFL(P)	360
SFR(P)	358
SFT(P)	240
SFTBL(P)	372
SFTBR(P)	370
SFTWL(P)	376
SFTWR(P)	374
SHL(_E)	1292
SHR(_E)	1294
SIMASK	530
SIN(_E)	1271
SIN(P)	840
SIND(P)	852
SJIS2WS(P)	767
SJIS2WSB(P)	769
SORTD(_U)	681
SP_DEVST	592
SP_ECPRTCL	1138
SP_FCOPY	624
SP_FDELETE	621
SP_FMOVE	629
SP_FREAD	594
SP_FRENAME	634
SP_FSTATUS	637
SP_FTPGET	1168
SP_FTPPUT	1163

SP_FWRITE	608
SP_SLMP SND	1146
SP_SOCKINF	1126
SP_SOCKCLOSE	1114
SP_SOCKSET	1129
SP_SOCOPEN	1110
SP_SOCRVCV	1116
SP_SOCRMODE	1131
SP_SOCSND	1122
SP.DEVST	592
SP.ECPRTCL	1138
SP.FCOPY	624
SP.FDELETE	621
SP.FMOVE	629
SP.FREAD	594
SP.FRENAME	634
SP.FSTATUS	637
SP.FTPGET	1168
SP.FTPPUT	1163
SP.FWRITE	608
SP_SLMP SND	1146
SP_SOCKCLOSE	1114
SP_SOCKSET	1129
SP_SOCOPEN	1110
SP_SOCRVCV	1116
SP_SOCRMODE	1131
SP_SOCSND	1122
SPDH	1105
SPF(P)	749
SQRT(_E)	1266
SQRT(P)	693
SR(_E)	1344
SRND(P)	917
STMR	942
STOP	249
STR(P)(_U)	727
STRDEL(P)	786
STRING_TO_BCD(_E)	1246
STRING_TO_BOOL(_E)	1240
STRING_TO_DINT(_E)	1244
STRING_TO_DWORD(_E)	1242
STRING_TO_INT(_E)	1243
STRING_TO_REAL(_E)	1248
STRING_TO_TIME(_E)	1251
STRING_TO_WORD(_E)	1241
STRINGMOV(P)	706
STRINGMOV(P)_WS	708
STRINGPLUS(P)	704
STRINS(P)	784
SUB_TIME(_E)	1337
SUB(_E)	1282
SUM(P)	671
SWAP(P)	510

T

TAN(_E)	1273
TAN(P)	844
TAND(P)	856
TCMP(P)	967
TEST(P)	353
TIMCHK	977
TIME_TO_BOOL(_E)	1234
TIME_TO_DINT(_E)	1238
TIME_TO_DWORD(_E)	1236
TIME_TO_INT(_E)	1237

TIME_TO_STRING(_E)	1239
TIME_TO_WORD(_E)	1235
TIME2SEC(P)	952
TIMER_10_FB_M	1367
TIMER_100_FB_M	1367
TIMER_CONT_FB_M	1367
TIMER_CONTHFB_M	1367
TIMER_HIGH_FB_M	1367
TIMER_LOW_FB_M	1367
TO(P)	983
TOD(P)	992
TOF(_E)	1365
TON(_E)	1363
TP(_E)	1361
TTMR	940

U

UDCNT1	936
UDCNT2	938
UDINT2DBL(P)	828
UDINT2DINT(P)	424
UDINT2FLT(P)	821
UDINT2INT(P)	420
UDINT2UINT(P)	422
UINT2DBL(P)	826
UINT2DINT(P)	410
UINT2FLT(P)	818
UINT2INT(P)	408
UINT2UDINT(P)	412
UNI(P)	476
UNIINFRD(P)	996

V

VAL(P)(_U)	456
------------	-----

W

WAND(P)	319, 320
WDT(P)	533
WOR(P)	327, 328
WORD_TO_BOOL(_E)	1186
WORD_TO_DINT(_E)	1189
WORD_TO_DWORD(_E)	1187
WORD_TO_INT(_E)	1188
WORD_TO_STRING(_E)	1191
WORD_TO_TIME(_E)	1190
WS2SJIS(P)	765
WSUM(P)(_U)	685
WTOB(P)	482
WXNR(P)	343, 344
WXOR(P)	335, 336

X

XCALL	564
XCH(P)	506
XOR(_E)	1300

Z

ZONE(P)(_U)	649
ZPOP(P)	921, 926
ZPOP(P)_2	926
ZPUSH(P)	918, 922

ZPUSH(P)_2	922
ZRRDB(P)	930
ZRWRB(P)	932

修订记录

*本手册号在封底的左下角。

修订日期	*手册编号	修改内容
2024年10月	SH(NA)-082692CHN-A	第一版
2025年5月	SH(NA)-082692CHN-B	■添加/修改位置 9章、13.2节、13.3节、22.1节、27.1节、27.2节、28.1节、28.2节、28.3节、29.1节、43.1节、44章、45章、46章、附录1、关于产品的应用
2025年10月	SH(NA)-082692CHN-C	■添加/修改位置 总称/简称、6章、9章、13章、23章、25章、26章、27章、28章、44章、45章、46章
2026年4月	SH(NA)-082692CHN-D	■添加/修改位置 17.1节、32.42节、32.44节、32.45节、39.1节、39.2节、39.8节、46.9节、46.10节、46.17节、47.7节、47.8节、47.14节

日语版手册编号：SH-082690-D

本手册不授予工业产权或任何其它类型的权利，也不授予任何专利许可。三菱电机对由于使用了本手册中的内容而引起的涉及工业产权的任何问题不承担责任。

© 2024 MITSUBISHI ELECTRIC CORPORATION

关于产品的应用

(1) 在使用三菱电机可编程控制器时，应该符合以下条件：即使在可编程控制器设备出现问题或故障时也不会导致重大事故，并且应在设备外部系统地配备能应付任何问题或故障的备用设备及失效安全功能。

(2) 三菱电机可编程控制器是以一般工业用途等为对象设计和制造的通用产品。

因此，三菱电机可编程控制器不应用于以下设备·系统等特殊用途。如果用于以下特殊用途，对于三菱电机可编程控制器的质量、性能、安全等所有相关责任（包括但不限于债务未履行责任、瑕疵担保责任、质量保证责任、违法行为责任、制造物责任），三菱电机将不负责。

- 面向各电力公司的核电站以及其它发电厂等对公众有较大影响的用途。
- 用于各铁路公司或公用设施目的等有特殊质量保证体系要求的用途。
- 航空航天、医疗、铁路、焚烧·燃料装置、载人移动设备、载人运输装置、娱乐设备、安全设备等预计对人身财产有较大影响的用途。

然而，对于上述应用，如果在限于具体用途，无需特殊质量（超出一般规格的质量等）要求的条件下，经过三菱电机的判断也可以使用三菱电机可编程控制器，详细情况请与当地三菱电机代表机构协商。

(3) 因拒绝服务攻击（DoS攻击）、非法访问、计算机病毒以及其他网络攻击引发的可编程控制器与系统方面的各种问题，三菱电机不承担责任。

质保

使用之前请确认以下产品质保的详细说明。

1. 免费质保期限和免费质保范围

在免费质保期内使用本产品时如果出现任何属于三菱电机责任的故障或缺陷（以下称“故障”），则经销商或三菱电机服务公司负责免费维修。

但是如果需要在国内现场或海外维修时，则要收取派遣工程师的费用。对于涉及到更换故障模块后的任何再试运转、维护或现场测试，三菱电机将不负任何责任。

[免费质保期限]

免费质保期限为自购买日或交货的一年内。

注意产品从三菱电机生产并出货之后，最长分销时间为 6 个月，生产后最长的免费质保期为 18 个月。维修零部件的免费质保期不得超过修理前的免费质保期。

[免费质保范围]

- (1) 范围局限于按照使用手册、用户手册及产品上的警示标签规定的使用状态、使用方法和使用环境正常使用的情况下。
- (2) 以下情况下，即使在免费质保期内，也要收取维修费用。
 - ① 因不当存储或搬运、用户过失或疏忽而引起的故障。因用户的硬件或软件设计而导致的故障。
 - ② 因用户未经批准对产品进行改造而导致的故障等。
 - ③ 对于装有三菱电机产品的用户设备，如果根据现有的法定安全措施或工业标准要求配备必需的功能或结构后本可以避免的故障。
 - ④ 如果正确维护或更换了使用手册中指定的耗材（电池、背光灯、保险丝等）后本可以避免的故障。
 - ⑤ 因火灾或异常电压等外部因素以及因地震、雷电、大风或水灾等不可抗力而导致的故障。
 - ⑥ 根据从三菱电机出货时的科技标准还无法预知的原因而导致的故障。
 - ⑦ 任何非三菱电机或用户责任而导致的故障。

2. 产品停产后的有偿维修期限

- (1) 三菱电机在本产品停产后的 7 年内受理该产品的有偿维修。
停产的消息将以三菱电机技术公告等方式予以通告。
- (2) 产品停产，将不再提供产品（包括维修零件）。

3. 海外服务

在海外，维修由三菱电机在当地的海外 FA 中心受理。注意各个 FA 中心的维修条件可能会不同。

4. 机会损失和间接损失不在质保责任范围内

无论是否在免费质保期内，凡以下事由三菱电机将不承担责任。

- (1) 任何非三菱电机责任原因而导致的损失。
- (2) 因三菱电机产品故障而引起的用户机会损失、利润损失。
- (3) 无论三菱电机能否预测，由特殊原因而导致的损失和间接损失、事故赔偿、以及三菱电机产品以外的损伤。
- (4) 对于用户更换设备、现场机械设备的再调试、运行测试及其它作业等的补偿。

5. 产品规格的改变

目录、手册或技术文档中的规格如有改变，恕不另行通知。

资讯与服务

关于资讯与服务，请向当地三菱电机代理店咨询。
请访问三菱电机官网查找当地三菱电机代理店。

MITSUBISHI ELECTRIC Factory Automation Global Website
Locations Worldwide
www.MitsubishiElectric.com/fa/about-us/overseas/

商标

The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.

In some cases, trademark symbols such as ‘™’ or ‘®’ are not specified in this manual.

SH(NA)-082692CHN-D

三菱电机自动化(中国)有限公司

地址：上海市虹桥路1386号三菱电机自动化中心

邮编：200336

电话：86-21-2322-3030 传真：86-21-2322-3000

官网：<https://www.MitsubishiElectric-FA.cn>

技术支持热线 **400-821-3030**



内容如有更改 恕不另行通知