



三菱电机 **通用** 可编程控制器

MELSEC iQ-R
series

MELSEC iQ-R编程手册 (运动控制FB篇)

-RD78G4
-RD78G8
-RD78G16
-RD78G32
-RD78G64
-RD78GHV
-RD78GHW

安全注意事项

(使用之前请务必阅读)

在使用MELSEC iQ-R系列可编程控制器之前，请仔细阅读各产品的手册以及各产品的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

在此感谢贵方购买了三菱电机可编程控制器MELSEC iQ-R系列的产品。

本手册是用于让用户了解进行运动模块的编程时所需的运动控制FB、变量及创建方法等有关内容的手册。

在使用之前应熟读本手册及关联手册，在充分了解MELSEC iQ-R系列可编程控制器的功能・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。

对象模块

RD78G4、RD78G8、RD78G16、RD78G32、RD78G64、RD78GHV、RD78GHW

要点

本手册中使用的符号如下所示。

- 带下划线的变量(AxisName等)：由用户定义的变量
-

目录

安全注意事项	1
前言	1
关联手册	5
术语	6
总称/简称	8
第1章 运动控制FB的概要	9
1.1 运动控制FB	9
运动控制FB的使用方法	9
各系统状态的本功能的动作	9
运动控制FB的类型	10
运动控制的类型	14
出错处理	14
控制中使用的单位	15
运动控制FB的输入输出变量	15
注意事项	18
1.2 运动控制FB的配置	19
梯形图	19
FBD/LD(功能块图表/梯形图)	20
ST(结构化文本)	20
第2章 变量及运动控制FB	21
2.1 变量一览	21
轴变量	23
轴组变量	44
系统变量	50
其它结构体的变量	60
2.2 列举符一览	71
ENUM列举符	71
2.3 运动控制FB一览	78
管理系统FB	78
动作系统FB	79
常规FB	79
第3章 运动控制FB	80
3.1 管理系统的FB	80
MC_GroupEnable	80
MC_GroupDisable	83
MC_Power	86
MC_SetPosition	89
MCv_SetTorqueLimit	95
MC_SetOverride	102
MC_ReadParameter	106
MC_WriteParameter	112
MC_Reset	118
MC_GroupReset	121

MC_TouchProbe	125
MC_AbortTrigger	137
MC_CamTableSelect	142
MCv_ChangeCycle	149
MCv_AllPower	159
MC_GroupSetOverride	163
MCv_MotionErrorReset	167
3.2 动作系统的FB	171
MC_Home	171
MC_Stop	177
MC_GroupStop	181
MC_MoveAbsolute	186
MC_MoveRelative	197
MCv_Jog	205
MC_MoveVelocity	213
MC_TorqueControl	220
MCv_SpeedControl	229
MCv_MoveLinearInterpolateAbsolute	237
MCv_MoveLinearInterpolateRelative	251
MCv_MoveCircularInterpolateAbsolute	264
MCv_MoveCircularInterpolateRelative	278
MC_CamIn	294
MC_GearIn	316
MC_CombineAxes	325
MCv_BacklashCompensationFilter	336
MCv_SmoothingFilter	349
MCv_DirectionFilter	358
MCv_SpeedLimitFilter	366
3.3 常规FB	374
MCv_ReadProfileData	374
MCv_WriteProfileData	382
第4章 运动模块的程序	390
4.1 运动模块的程序	390
程序的执行	391
标签的初始化功能	396
标签存储器	396
4.2 运动模块的程序创建	397
使用ST程序的程序的创建步骤	397
使用公开标签功能的程序的创建步骤	421
运动模块FB库的登录	431
使用运动模块的缓冲存储器的数据发送接收	436
附录	437
附1 FBD/LD的样本程序	437
GX Works3中的设置	437
运动控制设置功能中的设置	452
附2 ST的样本程序	455
GX Works3中的设置	455
运动控制设置功能中的设置	459

修订记录	474
质保	475
商标	476

关联手册

关于最新的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	提供形式
MELSEC iQ-R编程手册(运动控制FB篇) [IB-0300534CHN](本手册)	记载了运动控制FB、变量、编程有关内容。	e-Manual PDF
MELSEC iQ-R运动模块用户手册(入门篇) [IB-0300407CHN]	记载了运动模块的规格、投运步骤、系统配置、配线有关内容。	e-Manual PDF
MELSEC iQ-R运动模块用户手册(应用篇) [IB-0300412CHN]	记载了运动模块的功能、输入输出信号、变量、标签、编程、故障排除有关内容。	e-Manual PDF
MELSEC iQ-R运动模块用户手册(网络篇) [IB-0300427CHN]	记载了CC-Link IE TSN的功能、参数设置、故障排除、缓冲存储器有关内容。	e-Manual PDF
MELSEC iQ-R编程手册(运动模块用指令/通用FUN/通用FB篇) [IB-0300432CHN]	记载了运动模块的指令、通用函数/通用FB有关内容。	e-Manual PDF
运动模块快速入门指南 [L03191ENG]	为了初次使用运动模块的人员，记载了系统的启动、参数设置、编程方法有关内容。	e-Manual PDF
运动模块快速入门指南(可编程控制器CPU梯形图程序篇) [L03194ENG]	为了初次使用运动模块的人员，记载了系统的启动、参数设置、编程方法有关内容。	e-Manual PDF
MELSEC iQ-R编程手册(程序设计篇) [SH-081319CHN]	记载了梯形图、ST、FBD/LD、SFC的程序规格有关内容。	e-Manual PDF
GX Works3操作手册 [SH-081271CHN]	说明了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。	e-Manual PDF

关于程序，请参阅下述手册。

 MELSEC iQ-R编程手册(程序设计篇)

要点

e-Manual是可以使用专用工具进行浏览的三菱电机FA电子书籍手册。

e-Manual具有以下特点。

- 可以从多本手册同时搜索需要的信息(跨手册搜索)
- 可以通过手册内的链接浏览其它手册
- 可以通过产品插图的各部分浏览想要了解的硬件规格
- 可以将频繁浏览的信息登录到收藏夹
- 可以将样本程序复制到工程工具中

术语

在本手册中，除非特别标明，将使用以下术语进行说明。

术语	内容
Aborting、Buffered、Blending	缓冲模式的各种模式
GX Works3	MELSEC可编程控制器软件包的产品名
MR Configurator2	伺服设置软件的产品名
MR-J5-G	MR-J5_G_(-RJ)伺服放大器
MR-J5D-G	MR-J5D_ _G_伺服放大器
MR-J5W-G	MR-J5W_ _G_伺服放大器
RD78G	MELSEC iQ-R系列运动模块(CC-Link IE TSN对应)的别称
RD78GH	
RWr	链接软元件的远程寄存器。从从站向主站以16位(1字)单位输入的信息。(在本站中有部分不同。)
RWw	链接软元件的远程寄存器。从主站向从站以16位(1字)单位输出的信息。(在本站中有部分不同。)
RX	链接软元件的远程输入。从从站向主站以位单位输入的信息。(在本站中有部分不同。)
RY	链接软元件的远程输出。从主站向从站以位单位输出的信息。(在本站中有部分不同。)
安全通信	在同一网络的安全站之间，进行安全数据的通信的功能
常规FB	不将轴或轴组作为参数的运动控制FB
常规站	运动管理(控制)站以外的从站
智能功能模块	A/D、D/A转换模块等，具有输入输出以外的功能的模块
进给机械位置	不通过环形计数器化整的指令位置地址。以原点为基准，不受当前位置更改的影响。
对象	CANopen对应的从设备具有的各种各样的数据
外部信号高精度输入	与从站的信号检测时间相关的输入信号
解除连接	数据链接异常时，停止数据链接的处理
虚拟编码器轴	从变量创建指令位置的轴。用于单轴同步控制的主轴。
虚拟轴	未连接到网络上的从设备的轴
虚拟驱动轴	可虚拟创建指令的虚拟轴
虚拟连接轴	单轴同步控制中用于在FB之间连接以传送指令的轴
管理系统FB	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
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
全局标签	在工程内创建了多个程序数据时，对所有的程序数据均有效的标签。可以将运动系统内的全局标签作为管理CPU模块内的模块标签(全局标签)进行公开。
重启	对于具有Execute输入的运动控制FB，在FB执行中重新启动Execute输入
循环传送	在网络的站之间定期进行数据通信的功能
轴	进行运动控制的对象
轴组变量	包含轴组相关的参数及数据，AXES_GROUP型变量的接口
轴变量	包含轴相关的参数及数据，AXIS_*型变量的接口
系统启动	通过系统电源的接通或重启，进行运动模块的初始化处理
系统基本周期	运动运算处理等的恒定周期处理的基本周期
实编码器轴	从从设备上连接的编码器的当前位置创建指令位置的轴。用于单轴同步控制的主轴。
执行	对运动控制FB将Execute输入或Enable输入置为TRUE
实轴	已连接到网络上的从设备的轴
实驱动轴	已连接到与CiA402驱动器配置文件的csp/csv/cst模式(逐次指令)对应的CC-Link IE TSN对应从设备的轴
指定位置	动作系统FB中指定的指令位置地址
指定速度	动作系统FB中指定的指令速度
启动	首次对轴或轴组执行动作系统FB
Jerk	加速度或减速度的时间变化比率
输出变量	FB的输出参数
指令当前位置	通过环形计数器化整的指令位置地址。受当前位置更改的影响。

术语	内容
指令当前速度	通过运动运算创建的当前的控制值
专用指令	用于使用模块的功能的指令
软重启	不执行硬重启，仅复位运动部及网络部
多轴插补控制	直线插补及圆弧插补等多个轴关联的动作控制。指定通过轴组进行协调的轴。
多重启动	在运动控制FB的执行中，对同一轴执行其它运动控制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
之前的FB	缓冲FB的前1个运动控制FB(缓冲的FB为1个的情况下指执行中的FB)
下一个FB	轴及轴组的动作中多重启动的运动控制FB
软元件	模块内部具有的各种存储器。有以位单位及字单位处理的软元件。
瞬时传送	在网络的节点(站)之间非定期进行数据通信的功能。 用于来自于链接专用指令及工程工具的请求时，向对象站发送信息的功能。也可通过中继站或网关与其它网络的站进行通信。
动作系统FB	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
输入变量	FB的输入参数
网络部	运动模块内进行网络控制的软件
硬重启	重新接通系统的电源或复位
缓冲存储器	用于存储设置值、监视值等的数据的智能功能模块的存储器
缓冲模式	多重启动的别称
缓冲	多重启动中，运动控制FB进入待机状态
缓冲FB	进行多重启动，处于执行等待状态(Busy为TRUE)的运动控制FB
反馈位置	对从设备返回的位置地址通过环形计数器进行了化整的位置。受当前位置更改的影响。
反馈速度	实轴中将来自于从设备的反馈值转换为轴单位系统的速度
恢复连接	异常站变为正常时，重启数据链接的处理
运动管理站	通过从标签及运动控制进行循环数据通信的从站
运动控制FB	与运动控制相关的FB。名称以MC_或MCv_开头。
运动控制站	运动管理站的别称
运动部	运动模块内进行运动控制的软件
运动模块	RD78G(H)的别称
目标位置	动作系统FB中指定的指令位置地址
模块标签	将各模块固有定义的存储器(输入输出信号及缓冲存储器)以任意字符串表示的标签。从所使用的模块由GX Works3自动生成，可以作为CPU模块内的全局标签使用。
标签	程序中使用的变量
链接软元件	CC-Link IE的模块内部具有的软元件
链接刷新	在运动模块的链接软元件与CPU模块的软元件之间，自动进行数据传送。
累计当前位置	不通过环形计数器化整的指令位置地址。受当前位置更改的影响。
连续更新	运动控制FB的ContinuousUpdate输入为TRUE期间，将输入值连续反映到控制

总称/简称

在本手册中，除非特别标明，将使用以下总称/简称进行说明。

术语	内容
CC-Link IE	下述总称 <ul style="list-style-type: none"> • CC-Link IE TSN • CC-Link IE控制网络(☐)MELSEC iQ-R CC-Link IE控制网络用户手册(应用篇)) • CC-Link IE现场网络(☐)MELSEC iQ-R CC-Link IE现场网络用户手册(应用篇))
CPU模块	MELSEC iQ-R系列CPU模块的简称
csp	循环位置模式(Cyclic synchronous position mode)的简称(驱动器侧的控制模式的一种)
cst	循环转矩模式(Cyclic synchronous torque mode)的简称(驱动器侧的控制模式的一种)
csv	循环速度模式(Cyclic synchronous velocity mode)的简称(驱动器侧的控制模式的一种)
ct	挡块控制模式(Continuous operation to torque control mode)的简称(驱动器侧的控制模式的一种)
GOT	三菱电机图形操作终端GOT1000、GOT2000系列的总称
hm	原点复位模式(Homing mode)的简称(驱动器侧的控制模式的一种)
MCFB	运动控制FB的简称
MR-J5(W)-G	MR-J5-G(-RJ)/MR-J5W-G/MR-J5D-G_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。
RAS	Reliability(可靠性)Availability(可用性)Serviceability(易维护性)的简称。称为自动化设备的综合易用性。
RD78G(H)	RD78G_、RD78GH_(高性能版)的总称
SDO	Service Data Object的简称。用于对任意的CANopen节点的对象词典内的对象条目进行访问的信息。在站之间非周期性进行发送接收。
SLMP	Seamless Message Protocol的简称。用于从个人计算机及显示器等的外部设备及支持SLMP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。
安全站	进行安全通信及常规通信的站的总称
运算配置文件	各种控制中使用的波形数据的总称
工程工具	GX Works3、MR Configurator2的总称
从站	在CC-Link IE TSN中，本地站、远程站的总称
软件	配置运动系统的固件的总称。由插件、基本系统软件、启动软件构成。
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
运动系统	进行运动控制・网络控制的软件的总称
重启	硬重启及软重启的总称

1 运动控制FB的概要

以下对运动控制FB的使用方法、通用规格有关内容进行说明。

运动系统中可使用的运动控制FB中，包含由PLCopen[®]确定的FB。输入输出信号的基本规格基于PLCopen[®]的运动控制FB。

1.1 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。

各系统状态的本功能的动作

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/动作系统FB/常规FB

运动控制FB根据动作内容有以下类型。

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。
动作系统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FB，有的可以同时执行。 动作系统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FB，有可能引起特定的状态转变。
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，因此动作系统FB及管理系统FB相互无影响。

执行指令(Execute)类型/有效(Enable)类型

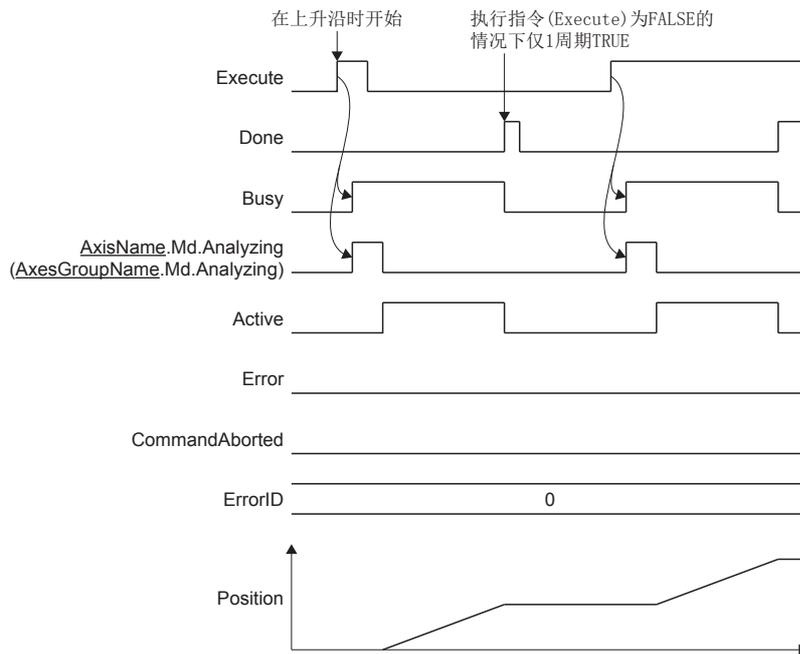
运动控制FB有通过执行指令(Execute)执行及通过有效(Enable)执行的类型。

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(轴出错复位) MC_GroupReset(轴组出错复位) MC_TouchProbe(触摸探头有效) MC_AbortTrigger(触摸探头无效) MC_CamTableSelect(凸轮表选择) MCv_ChangeCycle(1周期当前值更改) MCv_MotionErrorReset(运动出错复位) 	<ul style="list-style-type: none"> MC_Power(允许运行) MC_SetOverride(超驰值设置) MC_ReadParameter(参数读取) MCv_AllPower(所有轴允许运行) MC_GroupSetOverride(轴组超驰值设置) 	—
动作系统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) 	<ul style="list-style-type: none"> MCv_BacklashCompensationFilter(背隙补偿滤波器) 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) 	<ul style="list-style-type: none"> 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) 	—	—

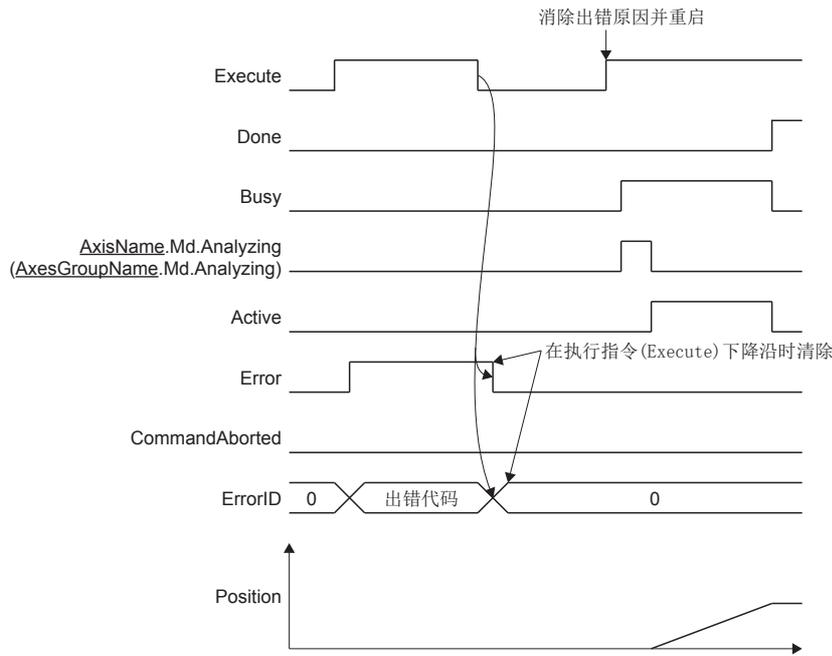
通过执行指令(Execute)及有效(Enable)的运动控制FB的基本动作如下所示。但是，根据运动控制FB，规格有可能不同。

■通过执行指令(Execute)类型的运动控制FB的基本动作

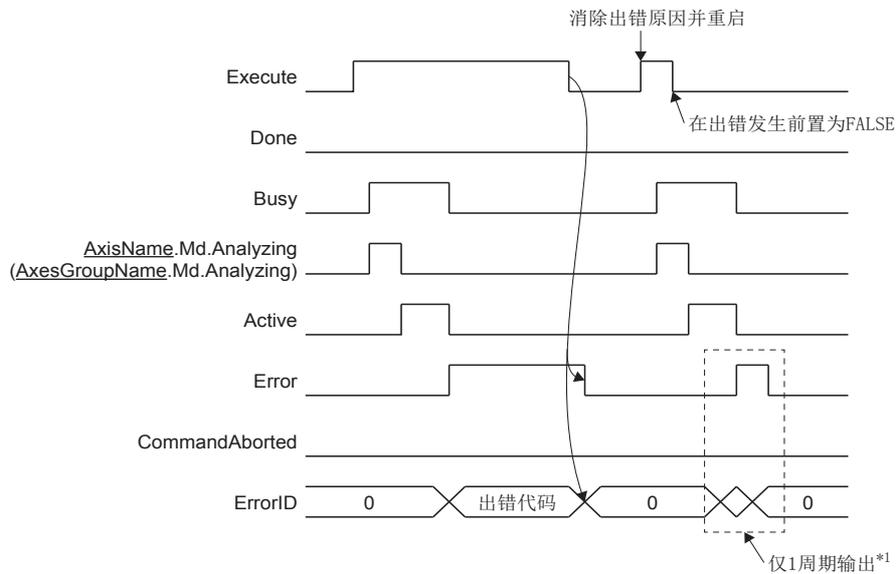
- 执行指令(Execute)类型的运动控制FB在执行指令(Execute)的上升沿读取输入参数，开始动作。一旦开始动作，即使将执行指令(Execute)置为FALSE也将继续动作直至完成为止。
- 开始动作时，执行中(Busy)、执行完成(Done)、出错(Error)、取消受理(CommandAborted)的输出变量中只有1个变为TRUE。
- 执行完成(Done)、出错(Error)、出错代码(ErrorID)、取消受理(CommandAborted)在执行指令(Execute)的下降沿复位。执行中(Busy)、控制中(Active)不受影响。
- 在动作中更改了输入参数的情况下，通过执行指令(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。
- 从执行中(Busy)的上升沿开始分析中(AxisName.Md.Analyzing/AxesGroupName.Md.Analyzing)将变为TRUE，开始动作时分析中(AxisName.Md.Analyzing/AxesGroupName.Md.Analyzing)将变为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。
- 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



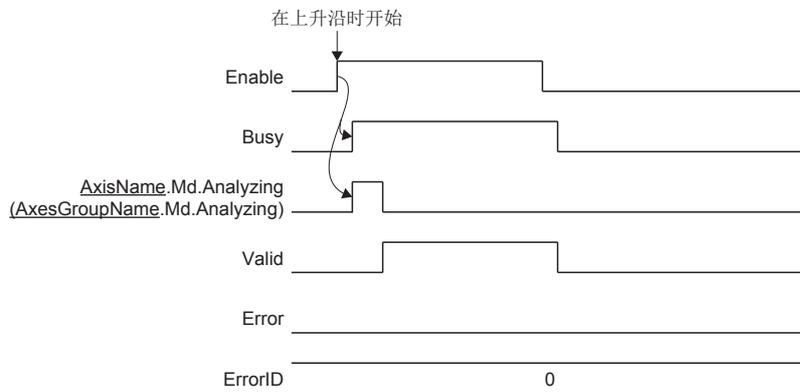
• 输入变量异常时



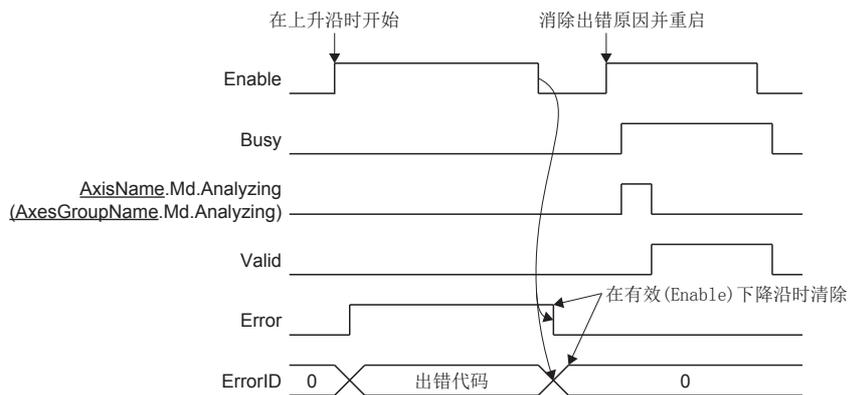
*1 由于FB的结束条件(执行指令(Execute)为FALSE)成立, 因此将变为上述动作。
与轴无关的FB或不减速停止的FB的情况下, 仅在1周期内出错(Error)变为TRUE, 且输出出错代码(ErrorID)。
需要进行减速停止的FB的情况下, 在轴减速停止之前出错(Error)将变为TRUE, 且维持出错代码(ErrorID), 在轴的停止完成时出错(Error)将变为FALSE, 且清除出错代码(ErrorID)。

■通过有效 (Enable) 类型的运动控制FB的基本动作

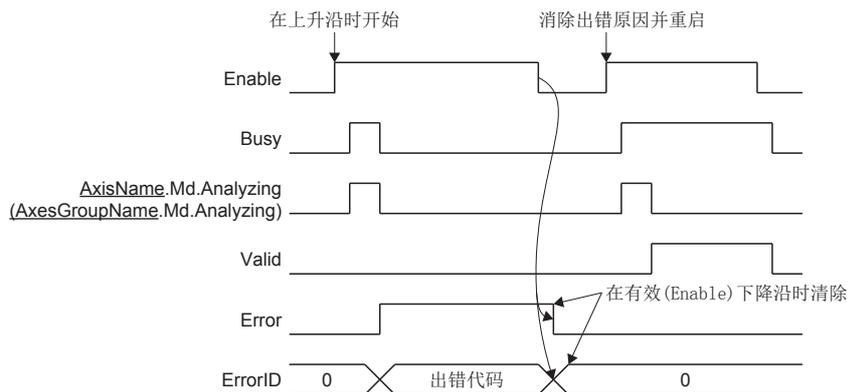
- 有效 (Enable) 类型的运动控制FB在有效 (Enable) 为TRUE期间，将继续重复执行。
- 输出值有效 (Valid) 表示输出为有效值。输出值有效 (Valid) 变为FALSE以后，各输出不变化。
- 输出值有效 (Valid) / 有效中 (Enabled) / 执行中 (Busy)、出错 (Error)、执行中断 (CommandAborted) 的输出变量中只有1个变为TRUE。
- 从执行中 (Busy) 的上升沿开始分析中 (AxisName.Md.Analyzing/AxesGroupName.Md.Analyzing) 将变为TRUE，开始动作时分析中 (AxisName.Md.Analyzing/AxesGroupName.Md.Analyzing) 将变为FALSE。
- 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

根据动作系统运动控制FB，可执行的轴、轴组的控制有以下类型。

大分类	中分类	小分类	内容
轴控制	单轴控制	定位控制	是将轴状态 (<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	是将轴状态 (<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	是将主轴(Master)与从轴(Slave)作为输入输出变量，并将从轴(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。
		原点复位控制	是将轴状态 (<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	是将轴组状态 (<u>AxisGroupName</u> .Md.GroupStatus)置为“5: 动作中(GroupMoving)”，以移动到目标位置的控制。

出错处理

运动控制FB的执行时发生出错时出错(Error)将变为TRUE，并将出错代码输出到出错代码(ErrorID)中。此时，轴的情况下，轴状态 (AxisName.Md.AxisStatus)将转变为“1: 出错停止中(ErrorStop)”，轴组的情况下轴组状态 (AxesGroupName.Md.GroupStatus)将转变为“1: 出错停止中(GroupErrorStop)”。

可动作的轴的状态转变为“1: 出错停止中(ErrorStop)”时，所有的缓冲FB均将中断。中断的FB的出错(Error)将变为TRUE。此后，轴或轴组启动时，需要执行出错复位。

关于轴或轴组的出错确认方法及出错复位方法，请参阅下述手册。

📖 MELSEC iQ-R运动模块用户手册(应用篇)

要点

在出错代码(ErrorID)中，输出与下述某个相同的值(根据控制而不同)。但是，不输出警告代码。

- 轴出错代码 (AxisName.Md.ErrorID)
- 轴组出错代码 (AxesGroupName.Md.ErrorID)
- 运动部最新系统出错代码(System.Md.ErrorID)

对于执行CPU模块侧的运动控制FB时的出错(包含警告)，将作为运动模块的出错进行输出。输出到运动控制FB的出错代码(ErrorID)中的出错代码如下所示。

出错代码	内容
0400H	运动模块在规定时间内没有响应。应再次执行FB。
1C00H	与FB内部使用的专用指令的出错代码(1800H~180FH)相对应。关于专用指令的出错代码，请参阅下述手册的“用户函数执行指令”。 📖 MELSEC iQ-R编程手册(运动模块用指令/通用FUN/通用FB篇)
1C01H	
1C02H	
1C03H	
1C04H	
1C05H	
1C06H	
1C07H	
1C0FH	

注意事项

运动控制FB的参数中指定的运动模块的输入输出编号不正确，且无法特定执行FB的运动模块的情况下，将无处理，或在CPU模块侧输出出错代码。

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。

📖 MELSEC iQ-R运动模块用户手册(应用篇)

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	是实轴中将来自于从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)等要控制的驱动器的轴、轴组。

对于轴、轴组，通过工程工具的轴设置(🔗 407页 轴)、轴组设置(🔗 417页 轴组)对轴进行创建、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。

📖 MELSEC iQ-R运动模块用户手册(应用篇)

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FB的状态、驱动器的状态、出错的有无等。

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。

☞ 22页 数据类型

要点

[关于变量的数组]

变量中有需要设置数组的变量。

通过在数据类型的后面将元素数用 “[]” 括起来表示数组。

设置标签的情况下，应设置 “[]” 内记载的元素数的范围。如果设置的范围与元素数不同，则在程序的转换时将出错。

数组通过工程工具的全局标签或局部标签的“数据类型选择”设置。关于详细内容，请参阅下述章节。

☞ 420页 标签的登录

<例>

- 数据类型“LREAL[0..15]”的情况下 在元素数中设置“16”。

输入参数的省略

省略了FB的输入的情况下，应用各FB中定义的初始值。关于初始值的详细内容，请参阅使用的运动控制FB。

多重启动的FB中省略了速度等的输入的情况下，沿用之前控制的FB的输入值。

输入输出的刷新时机

FB的参数在FB的调用时机进行刷新。与运算周期同步控制FB的输入输出时，应通过与运算周期相同周期的恒定周期程序调用FB。

输入输出编号的指定

在CPU模块侧使用运动控制FB的情况下，运动控制FB中使用的输入输出变量(例：输入输出No. (AxisName. AxisRef. StartIO))中，需要设置对象运动模块的输入输出编号。

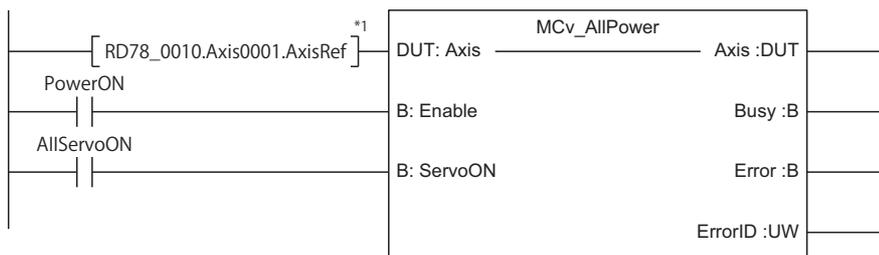
在CPU模块侧设置输入输出编号的情况下，使用作为运动模块的标签公开的轴变量或轴组变量的AXIS_REF(轴变量为 AxisName. AxisRef，轴组为 AxesGroupName. AxesGroupRef)。

由于在轴变量或轴组变量的AXIS_REF的成员(轴No. (AxisNo)、轴组No. (GroupNo)、输入输出No. (StartIO))中输入输出编号已经被设置，因此即使忽略也可通过设置轴的标签来设置输入输出编号。

设置输入输出编号的情况下，建议使用轴信息(AxisName. AxisRef)的标签进行设置。另外，如果作为运动模块的标签公开，则无论哪个轴信息(AxisName. AxisRef)的标签均可设置。

例

使用运动模块(输入输出编号：0010)的轴1(Axis0001)的标签，从CPU模块侧执行MCv_AllPower(所有轴允许运行)的情况下



*1 在CPU模块侧使用公开的标签的情况下，在标签的起始输入“模块名+起始输入输出编号”后设置标签。

RD78_0010.Axis0001.AxisRef



*2 轴信息(AxisName. AxisRef)的标签中包含下述成员的信息。

名称	设置值	标签名
轴No.	1	Axis0001.AxisRef.AxisNo
输入输出No.	H001*3	Axis0001.AxisRef.StartIO

*3 起始输入输出编号为以16进制数4位数表示时的前3位数

在运动模块侧使用运动控制FB的情况下，输入输出变量(例：输入输出No. (AxisName. AxisRef. StartIO))中不需要进行输入输出编号的设置。(设置将被忽略。)

注意事项

- 运动控制FB的输入输出参数的刷新在FB的调用时机进行，但实际控制执行周期根据FB而有所不同。例如，动作系统FB不通过FB调用任务(正常/恒定周期)，而是通过运算周期进行控制。关于详细内容，请参阅使用的FB。
- 在运动控制FB的动作中(执行中(Busy)为TRUE状态)，应调用FB。在IF语句等控制语法内记述FB，且不调用运动控制FB时，即使更改输入变量的值也不被反映到控制中，也无法进行执行完成及出错等的输出变量的更新。
- FB在将执行指令(Execute)或有效(Enable)置为了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FB的重启及连续更新以更改多个输入参数的情况下，为了使各参数的获取时机一致，应在与FB调用任务相同的任务中进行更改。
- 将FB的执行指令(Execute)置为了TRUE之后，将执行指令(Execute)置为FALSE的情况下，应在执行中(Busy)变为TRUE状态之后再置为FALSE。
- 在运动控制功能中，轴的当前位置及FB的目标位置等中，使用实数数据(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中使用的实数数据(浮点数据)的详细内容，请参阅下述手册。

📖 MELSEC iQ-R编程手册(运动模块用指令/通用FUN/通用FB篇)

例

设置(A)的情况下

下述设置(A)中运算误差会带来问题的情况下，更改的候补设置将变为(B)。

设置项目	设置内容	
	设置(A)	设置(B)
位置指令单位	[mm]	[μm]
环形计数器上限值	100.3[mm]	100300[μm]
环形计数器下限值	-100.7[mm]	-100700[μm]
FB输入中设置的目标位置·移动量的最小刻度	0.05[mm]	50[μm]

1.2 运动控制FB的配置

以下对用于使用运动控制FB创建程序的内容进行说明。

在CPU模块侧或运动模块侧使用运动控制FB创建程序的情况下，可创建的程序语言如下所示。

○：可以创建， ×：不能创建

程序语言	CPU模块侧	运动模块侧
梯形图	○	×
功能块图表/梯形图(FBD/LD)	○	×
结构化文本(ST)	○	○

要点

在CPU模块侧创建程序的情况下，为了使用在运动模块中登录的标签需要通过公开标签功能将其公开到CPU模块侧。关于详细内容，请参阅下述章节。

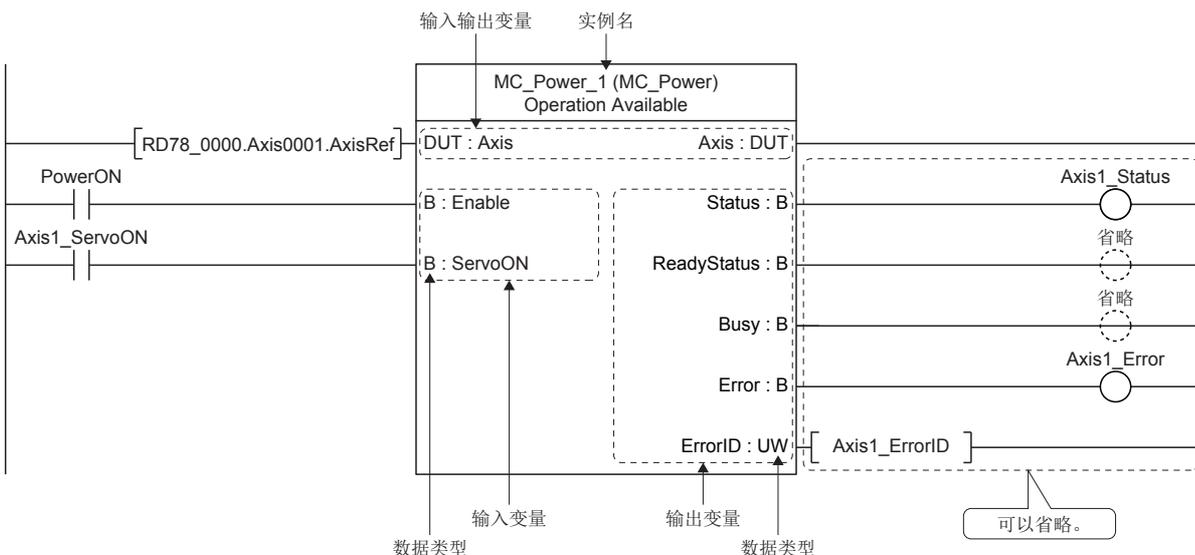
☞ 421页 使用公开标签功能的程序的创建步骤

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



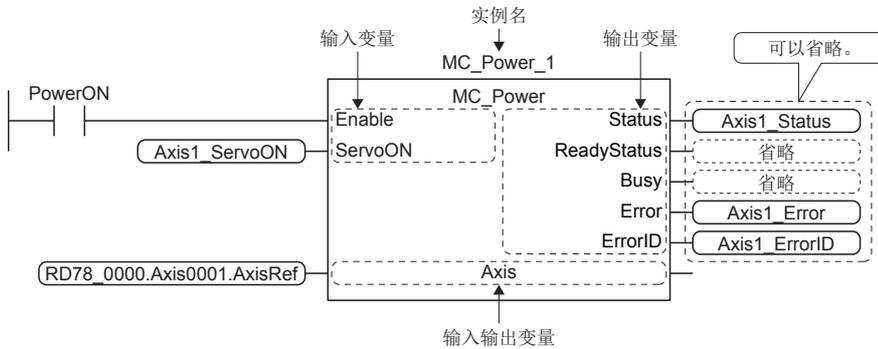
名称	内容
实例名	是对每个FB添加的实例名。 可以更改实例名。
输入输出变量	设置轴信息(Axis)、轴组信息(AxesGroup)等要控制的驱动器等的轴变量名。
输入变量	设置目标位置及指令速度等的动作条件。 可以省略设置。省略的情况下，输入变量变为初始值。
输出变量	输出FB的状态及驱动器的状态。
数据类型	表示输入输出变量、输入变量、输出变量的数据类型。 关于输入输出变量、输入变量、输出变量的数据类型，请参阅下述章节。 ☞ 22页 数据类型

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



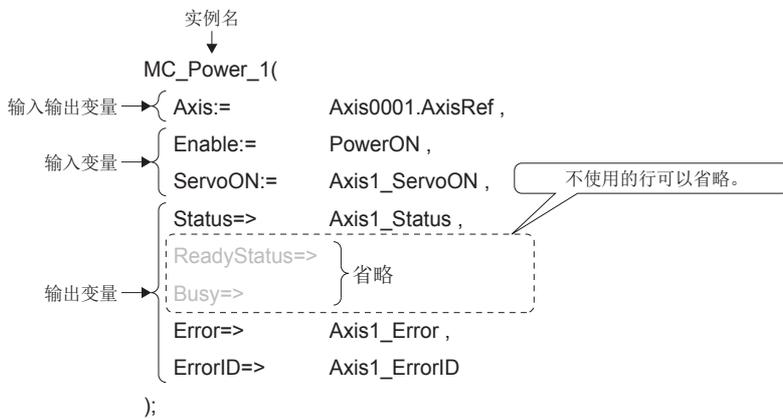
名称	内容
实例名	是对每个FB添加的实例名。 可以更改实例名。
输入输出变量	设置轴信息 (Axis)、轴组信息 (AxesGroup) 等要控制的驱动器等的轴变量名。
输入变量	设置目标位置及指令速度等的动作条件。 可以省略设置。省略的情况下，输入变量变为初始值。
输出变量	输出FB的状态及驱动器的状态。

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的状态及驱动器的状态。

2 变量及运动控制FB

本章对程序中使用的变量及运动控制FB的一览有关内容进行说明。

2.1 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。	☞ 23页 轴变量
轴组变量	是由构成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。	☞ 44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。	☞ 50页 系统变量
其它结构体的变量	是除上述以外分类的结构体的变量。	☞ 60页 其它结构体的变量

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

↓
由用户定义的变量

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
记录设置周期	记录对象数据的采集时机
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	从设备连接时
就绪ON	“可编程控制器就绪(YO)”变为了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

■数据类型

数据类型	内容
BOOL	位
WORD (HEX)	字[无符号]/位串[16位](16进制数)
WORD (UINT)	字[无符号]/位串[16位]
DWORD (HEX)	双字[无符号]/位串[32位](16进制数)
DWORD (UDINT)	双字[无符号]/位串[32位]
INT	字[有符号]
DINT	双字[有符号]
REAL	单精度实数
LREAL	双精度实数
TIME	时间
STRING(□)	字符串 *: □表示除Null以外的可设置的字符数。 *: STRING型的情况下, 将要设置的字符串以“单引号(')'”括起来。
WSTRING(□)	字符串[Unicode] *: □表示除Null以外的可设置的字符数。 *: WSTRING型的情况下, 将要设置的字符串以“双引号(“”)”括起来。

要点

关于数据类型的详细内容, 请参阅下述手册。

 MELSEC iQ-R编程手册(运动模块用指令/通用FUN/通用FB篇)

■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
实驱动轴	AXIS_REAL
实编码器轴	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。（☞ 39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成员。

成员名	数据类型*1*2	内容	参阅
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。	☞ 23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置值。 轴变量初始化后不实施至控制的重新获取。	☞ 24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。	☞ 27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。	☞ 30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。	☞ 38页 AxisName.Cd. (轴控制数据)

*1 □: 各轴类型的数据类型

*2 根据轴类型，数据类型的成员有所不同。

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.。 • 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) *: 在CPU模块侧使用的情况下设置此项。从运动模块侧使用的情况下将忽略。

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	<p>以字符串设置从站的网络地址(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。</p> <p>■单轴设备的情况下</p> <p>192.168.3.1 └── IP地址</p> <p>■多轴设备的情况下</p> <p><例> MR-J5W-G的C轴的情况下</p> <p>192.168.3.1#2 └── IP地址 └── 多点编号</p> <p>*: 以“# + 编号(10进制数)”设置多点编号。</p> <ul style="list-style-type: none"> • #0: A轴 • #1: B轴 • #2: C轴 <p>*: 多点编号可以省略。省略的情况下, 将视为“#0”。</p>
AxisType	轴类型设置	系统启动时	INT (MC_AXIS_TYPE) ☞ 74页 MC_AXIS_TYPE	LIST_WRITE_BACK	<p>设置轴类型。</p> <ul style="list-style-type: none"> • 0: 实驱动轴(DriveAxis) • 2: 实编码器轴(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	实编码器轴类型设置	系统启动时	INT (MC_ENCODER_AXIS_TYPE) ☞ 74页 MC_ENCODER_AXIS_TYPE	LIST_WRITE_BACK	<p>设置实编码器轴的类型。</p> <ul style="list-style-type: none"> • 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 ☞ 63页 SIGNAL_SELECT (信号选择)	LIST_WRITE_BACK	<p>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。</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"> • 0: TRUE时检测(HighLevel) • 1: FALSE时检测(LowLevel) <p>■补偿时间(CompensationTime) 只能设置“0.0[s]”。</p> <p>■滤波器时间(FilterTime) 设置范围为“0.0~5.0[s]”。</p> <p>*: 设置了超出范围的值的的情况下, 将变为“超出参数范围(轴)(出错代码: 1D80H)”。</p>
Encoder_RingCout_LowerValue	编码器环形计数器下限值	系统启动时	DINT	LIST_WRITE_BACK	<p>设置编码器环形计数器的下限值。</p> <p>■PosActualValue为1字</p> <ul style="list-style-type: none"> • -32768~32767 <p>*: 设置了超出1字的范围的值的的情况下, 将变为“超出参数范围(轴)(出错代码: 1D80H)”。</p> <p>■PosActualValue为2字或省略</p> <ul style="list-style-type: none"> • -2147483648~2147483647

变量名	名称	获取	数据类型	属性	内容
Encoder_RingCout_UpperValue	编码器环形计数器上限值	系统启动时	DINT	LIST_WRIT E_BACK	<p>设置编码器环形计数器的上限值。</p> <p>■PosActualValue为1字</p> <ul style="list-style-type: none"> -32768~32767 <p>*: 设置了超出1字的范围的值的的情况下, 将变为“超出参数范围(轴)(出错代码: 1D80H)”。</p> <p>■PosActualValue为2字或省略</p> <ul style="list-style-type: none"> -2147483648~2147483647
HwStrokeLimit_FlsSignal	上限限位信号	系统启动时	SIGNAL_SELECT ☞ 63页 SIGNAL_SELECT(信号选择)	LIST_WRIT E_BACK	<p>设置使用上限限位信号(FLS)的信号。</p> <p>上限限位信号(HwStrokeLimit_FlsSignal)固有的设置及动作如下所示。</p> <p>■IO编号(StartIO)</p> <p>忽略输入值。</p> <p>■对象(Target)</p> <p>无设置时, 将判断为信号无效且置为始终信号未检测状态。</p> <p>*: 设置了不能使用的数据类型的情况下, 将变为“超出参数范围(轴)(出错代码: 1D80H)”。</p> <p>■信号检测方法(Detection)</p> <p>只允许设置下述等级检测。</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的情况下, 将变为“超出参数范围(轴)(出错代码: 1D80H)”。</p> <p>■补偿时间(CompensationTime)</p> <p>忽略输入值。</p> <p>■滤波器时间(FilterTime)</p> <p>设置范围为“0.0~5.0[s]”。</p> <p>*: 设置了超出范围的值的的情况下, 将变为“超出各轴信号的滤波器时间设置范围警告(警告代码: OD24H)”, 且滤波器时间将以“0.0”执行动作。</p>
HwStrokeLimit_RlsSignal	下限限位信号	系统启动时	SIGNAL_SELECT ☞ 63页 SIGNAL_SELECT(信号选择)	LIST_WRIT E_BACK	<p>设置使用下限限位信号(RLS)的信号。</p> <p>下限限位信号(HwStrokeLimit_RlsSignal)固有的设置及动作如下所示。</p> <p>■IO编号(StartIO)</p> <p>忽略输入值。</p> <p>■对象(Target)</p> <p>无设置时, 将判断为信号无效且置为始终信号未检测状态。</p> <p>*: 设置了不能使用的数据类型的情况下, 将变为“超出参数范围(轴)(出错代码: 1D80H)”。</p> <p>■信号检测方法(Detection)</p> <p>只允许设置下述等级检测。</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的情况下, 将变为“超出参数范围(轴)(出错代码: 1D80H)”。</p> <p>■补偿时间(CompensationTime)</p> <p>忽略输入值。</p> <p>■滤波器时间(FilterTime)</p> <p>设置范围为“0.0~5.0[s]”。</p> <p>*: 设置了超出范围的值的的情况下, 将变为“超出各轴信号的滤波器时间设置范围警告(警告代码: OD24H)”, 且滤波器时间将以“0.0”执行动作。</p>
OperationCycle	控制周期设置	系统启动时	INT	LIST_WRIT E_BACK	<p>设置控制运算周期。</p> <ul style="list-style-type: none"> 0: 在第1运算周期动作
PositioningRange	定位范围	系统启动时	LREAL	LIST_WRIT E_BACK	<p>设置定位范围。</p> <ul style="list-style-type: none"> 0.0、200000000.0~100000000000.0

变量名	名称	获取	数据类型	属性	内容
PosRestoration_AbsPosBase	绝对位置基准设置	系统启动时	INT (MC_POS_SOURCE) ☞ 75页 MC_POS_SOURCE	LIST_WRIT E_BACK	使用绝对位置系统的情况下, 设置当前位置恢复时作为基准的当前位置。 • 3: 进给机械位置(FeedMachinePosition)
PosRestoration_AbsPosEnable	绝对位置管理设置	系统启动时	INT (MC_ABS_SYSTEM) ☞ 74页 MC_ABS_SYSTEM	LIST_WRIT E_BACK	设置绝对位置管理。 • 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)
RingCount_Enable	环形计数器有效选择	系统启动时	BOOL	LIST_WRIT E_BACK	设置环形计数器有效/无效。 • FALSE: 无效 • TRUE: 有效
RingCount_LowerValue	环形计数器下限值	系统启动时	LREAL	LIST_WRIT E_BACK	设置环形计数器下限值。 • -10000000000.0~10000000000.0 *: 环形计数器无效时不获取
RingCount_UpperValue	环形计数器上限值	系统启动时	LREAL	LIST_WRIT E_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	从模拟有效	系统启动时	BOOL	LIST_WRIT E_BACK	设置是否作为模拟轴使用。 • FALSE: 无效 • TRUE: 有效
SlaveObject	从对象数据	系统启动时	SLAVE_OBJECT_REAL ☞ 68页 SLAVE_OBJECT_REAL	LIST_WRIT E_BACK	在实驱动轴上设置从对象数据。
SlaveObject	从对象数据	系统启动时	SLAVE_OBJECT_ENCODER ☞ 70页 SLAVE_OBJECT_ENCODER	LIST_WRIT E_BACK	在实编码器轴上设置从对象数据。
SlaveObject	从对象数据	系统启动时	SLAVE_OBJECT_VIRTUAL_ENCODER ☞ 70页 SLAVE_OBJECT_VIRTUAL_ENCODER	LIST_WRIT E_BACK	在虚拟编码器轴上设置从对象数据。
TorqueLimit_Max	转矩限制最大值	系统启动时	LREAL	LIST_WRIT E_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_WRIT E_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_WRIT E_BACK	设置正方向转矩限制值 (AxisName, Cd.TorqueLimit_Positive)的初始值。 • 0.0~1000.0[%]
FastOperationMode	高速模式设置	系统启动时	WORD (HEX)	LIST_WRIT E_BACK	将设置的轴的运算处理切换至高速模式。 • 5FE2H: 高速模式 • 上述以外: 正常模式 *: 仅实驱动轴支持。

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] <p>*: 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。</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) ☞ 75页 MC_ACC_ZERO_MODE	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: 出错(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 <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[U/s²] <p>*: 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。</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>
Drive_UnitConvRatioNum	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的指令单位的分子。</p> <ul style="list-style-type: none"> • 1~2147483647
Drive_UnitConvRatioDen	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 ☞ 63页 SIGNAL_SELECT (信号选择)	LIST_WRITE_BACK	<p>设置使用紧急停止的信号。</p> <p>紧急停止信号 (ForcedStop_Signal) 固有的设置及动作如下所示。</p> <p>■IO编号 (StartIO) 忽略输入值。</p> <p>■对象 (Target) 无设置时, 将判断为信号无效且置为始终信号未检测状态。</p> <p>*: 设置了不能使用的数据类型的情况下, 将变为“超出参数范围(轴) (出错代码: 1D80H)”。</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"> • 0: TRUE时检测 (HighLevel) • 1: FALSE时检测 (LowLevel) <p>*: 设置了边缘检测的情况下, 将变为“超出参数范围(轴) (出错代码: 1D80H)”。</p> <p>■补偿时间 (CompensationTime) 忽略输入值。</p> <p>■滤波器时间 (FilterTime) 设置范围为“0.0~5.0[s]”。</p> <p>*: 设置了超出范围的值的情况下, 将变为“超出各轴信号的滤波器时间设置范围警告 (警告代码: 0D24H)”, 且滤波器时间将以“0.0”执行动作。</p>

变量名	名称	获取	数据类型	属性	内容
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) ☞ 75页 MC_DECEL_STOP_MODE	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。 ■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 • 0.0000、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”的情况下将立即停止。 *: 发生停止原因时超出上限值范围的情况下将以上限值限定，超出下限值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) ☞ 75页 MC_STOP_CURVE_MODE	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) ☞ 75页 MC_DECEL_STOP_MODE	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	发生硬件行程限位出错时停止选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 75页 MC_DECEL_STOP_MODE	LIST_WRITE_BACK	设置发生了硬件行程限位的出错时的动作。 • 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_ServoOfff	运行中伺服OFF指令时处理选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 75页 MC_DECEL_STOP_MODE	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)

变量名	名称	获取	数据类型	属性	内容
StopMode_SwStrokeLimit	发生软件行程限位出错时停止选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 75页 MC_DECEL_STOP_MODE	LIST_WRITE_BACK	设置发生了软件行程限位的出错时的动作。 <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) *: 执行无减速速度指定的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出错而停止。 <ul style="list-style-type: none"> • FALSE: 检测无效 • TRUE: 检测有效
StopSignal	停止信号	就绪ON	SIGNAL_SELECT ☞ 63页 SIGNAL_SELECT(信号选择)	LIST_WRITE_BACK	设置使用外部输入信号的停止信号(STOP)的信号。 停止信号(StopSignal)固有的设置及动作如下所示。 ■IO编号(StartIO) 忽略输入值。 ■对象(Target) 无设置时, 将判断为信号无效且置为始终信号未检测状态。 *: 设置了不能使用的数据类型的情况下, 将变为“超出参数范围(轴)(出错代码: 1D80H)”。 ■信号检测方法(Detection) 只允许设置下述等级检测。 <ul style="list-style-type: none"> • 0: TRUE时检测(HighLevel) • 1: FALSE时检测(LowLevel) *: 设置了边缘检测的情况下, 将输出“超出参数范围(轴)(出错代码: 1D80H)”。 ■补偿时间(CompensationTime) 忽略输入值。 ■滤波器时间(FilterTime) 设置范围为“0.0~5.0[s]”。 *: 设置了超出范围的值的的情况下, 将变为“超出各轴信号的滤波器时间设置范围警告(警告代码: OD24H)”, 且滤波器时间将以“0.0”执行动作。
SwStrokeLimit_Lower	软件行程限位下限值	就绪ON	LREAL	LIST_WRITE_BACK	设置软件行程限位下限值。 <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) ☞ 75页 MC_POS_SOURCE	LIST_WRITE_BACK	设置软件行程限位的对象。 <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	设置软件行程限位上限值。 <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	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 • 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	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。

变量名	名称	获取	数据类型	属性	内容
Unit_Velocity	速度指令单位	就绪ON	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
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。 <ul style="list-style-type: none"> • 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) <small>73页</small> MC_VELOCITY_LIMIT_MODE	LIST_WRITE_BACK	设置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) *: “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 (警告代码: 0D03H)”, 并原样不变地输出超出速度限制值的指令。 *: “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 (出错代码: 1A36H)”, 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。 <ul style="list-style-type: none"> • 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	加速度超驰系数	运算周期	LREAL	LIST_READ_ONLY	存储加速度超驰系数。
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) <small>75页</small> MC_ACC_ZERO_MODE	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。 <ul style="list-style-type: none"> • -1: 出错 (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值, 向负方向 (地址减少方向) 移动的情况下将为负值。
Analyzing	分析中	运算周期	BOOL	LIST_READ_ONLY	显示定位分析中的状态。 <ul style="list-style-type: none"> • 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。 轴组动作中, 构成轴的自动减速中 (AxisName.Md.AutoDeceleration), 将存储FALSE。 <ul style="list-style-type: none"> • 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。

变量名	名称	获取	数据类型	属性	内容
AxisStatus	轴状态	运算周期	INT (MC_AXIS_STATUS) 72页 MC_AXIS_STATUS	LIST_READ _ONLY	显示轴的状态。 <ul style="list-style-type: none"> -1: 轴变量未初始化/轴参数异常(Invalid) 0: 轴无效(Disabled) 1: 出错停止中(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	显示至目标位置为止的剩余距离是否小于或等于指令到位宽度(AxisName.Md.CmdInPos_Width)。 <ul style="list-style-type: none"> 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	显示指令到位宽度。 将为获取了指令到位宽度(AxisName.Pr.CmdInPos_Width)的设置值后的值。 指令到位宽度(AxisName.Md.CmdInPos_Width)为“0.0”的情况下, 相应轴的指令到位功能将变为无效。
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) 轴组动作中, 构成轴的指定加速度(AxisName.Md.CommandedAcceleration)中将存储“0.0”。
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。轴组动作中, 构成轴的指定减速度(AxisName.Md.CommandedDeceleration)中将存储“0.0”。
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)轴组动作中, 构成轴的指定Jerk(AxisName.Md.CommandedJerk)中将存储“0.0”。
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	存储控制中的动作系统的运动控制FB中获取的指定速度。 控制完成时将存储“0.0”。 轴组动作中, 构成轴的指定速度(AxisName.Md.CommandedVelocity)中将存储“0.0”。
Cst_SetTorque	转矩控制时指令当前转矩	运算周期	LREAL	LIST_READ _ONLY	存储循环转矩模式时的指令转矩。
Cst_TargetTorque	转矩控制时目标转矩	运算周期	LREAL	LIST_READ _ONLY	存储循环转矩模式时的目标转矩。
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“-10000000000.0 ≤ 累计当前位置 < 10000000000.0”的环形地址。

变量名	名称	获取	数据类型	属性	内容
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(从设备→运动系统)映射	设备连接时	DWORD(HEX)	LIST_READ_ONLY	显示RPDO映射的信息。
Drive_TPDO[1..64]	TPDO(运动系统→从设备)映射	设备连接时	DWORD(HEX)	LIST_READ_ONLY	显示TPDO映射的信息。
Drive_UnitConvRatioNum	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的指令单位的分子。
Drive_UnitConvRatioDen	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) ☞ 73页 MC_DRIVE_MODE	LIST_READ_ONLY	显示驱动器控制模式的状态。 • 6: 原点复位(hm) • 8: 位置控制(csp) • 9: 速度控制(csv) • 10: 转矩控制(cst) • -104: 挡块控制(ct)
Driver_ReadyOn	驱动器就绪ON状态	运算周期	BOOL	LIST_READ_ONLY	显示连接驱动器的状态。 驱动器状态(AxisName.Md.Driver_State)为下述的情况下, 将变为TRUE。 • 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	显示连接驱动器的状态。 驱动器状态(AxisName.Md.Driver_State)为下述的情况下, 将变为TRUE。 • 6: Operation Enable • 7: Quick Stop Active
Driver_State	驱动器状态	运算周期	INT (MC_DRIVE_STATE) ☞ 75页 MC_DRIVE_STATE	LIST_READ_ONLY	显示驱动器状态。 • 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	驱动器模块出错检测	即时	BOOL	LIST_READ_ONLY	显示驱动器模块出错的状态。 • FALSE: 无驱动器模块出错 • TRUE: 有驱动器模块出错
DriverErrorID	驱动器模块出错代码	即时	WORD(HEX)	LIST_READ_ONLY	发生驱动器模块出错时, 显示从对象的“Current alarm(2A41H)”的高位16位。
DriverErrorDetailID	驱动器模块出错详细代码	即时	WORD(HEX)	LIST_READ_ONLY	发生驱动器模块出错时, 存储从对象的“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	轴出错检测	即时	BOOL	LIST_READ_ONLY	显示轴出错的状态。 • FALSE: 无轴出错 • TRUE: 有轴出错
ErrorID	轴出错代码	即时	WORD(HEX)	LIST_READ_ONLY	显示发生的轴出错的出错代码。

变量名	名称	获取	数据类型	属性	内容
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“-1000000000.0 ≤ 进给机械位置 < 1000000000.0”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 ☞ 63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。 ■IO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”。 ■滤波器时间(FilterTime) 显示参数的获取结果。
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: 发生原点复位出错(速度为0以外) • 0005H: 发生原点复位出错(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、滤波器时间的检测状态。
HwStrokeLimit_Override	硬件行程限位超驰	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

变量名	名称	获取	数据类型	属性	内容
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、滤波器时间的检测状态。
InVelocity	目标速度到达	运算周期	BOOL	LIST_READ _ONLY	显示是否到达目标速度。 轴组动作中, 构成轴的目标速度到达(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PosActualValue的值。(从从设备接收的位置反馈) 实编码器轴、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PosEncoderResolution的值。 实编码器轴、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的值。 实编码器轴的情况下, 将编码器的状态在运动模块侧进行模拟并显示。
Io_TargetPos	对象数据_TargetPos	运算周期	DINT	LIST_READ _ONLY	显示对象数据TargetPos的值。(发送到从设备中的位置指令) 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TargetVelocity的值。(发送到从设备中的速度指令)
Io_TorqueActualValue	对象数据 _TorqueActualValue	运算周期	INT	LIST_READ _ONLY	显示对象数据TorqueActualValue的值。(从从设备接收的转矩反馈)
Io_VelActualValue	对象数据 _VelActualValue	运算周期	DINT	LIST_READ _ONLY	显示对象数据VelActualValue的值。(从从设备接收的速度反馈)
JerkLimit	Jerk限制值	就绪ON	LREAL	LIST_READ _ONLY	存储Jerk限制值。
OperationCycle	控制周期	系统启动时	INT	LIST_READ _ONLY	存储控制运算周期的值。
OverrunOperation	越程时动作设置	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 75页 MC_DECEL_STOP_MODE	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) ☞ 77页 MC_POS_RESTORATION_STATUS	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) *: 从设备断开时将变为“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。

变量名	名称	获取	数据类型	属性	内容
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) 通过符号表示是加速中还是减速中。 到达目标速度时存储“0.0”。 轴组动作中, 存储构成轴的指令当前加速度(SetAcceleration)。 • 0.0: 停止中或定速中 • 正符号: 加速中 • 负符号: 减速中
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的值, 向负方向(地址减少方向)移动的情况下将为负的值。 轴组动作中, 存储构成轴的指令当前速度。
SlaveEmulate_Enable	从模拟中	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) ☞ 75页 MC_STOP_CURVE_MODE	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) ☞ 75页 MC_DECEL_STOP_MODE	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	发生硬件行程限位出错时停止选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 75页 MC_DECEL_STOP_MODE	LIST_READ _ONLY	显示发生硬件行程限位出错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) ☞ 75页 MC_DECEL_STOP_MODE	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)
StopMode_SwStrokeLimit	发生软件行程限位出错时停止选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 75页 MC_DECEL_STOP_MODE	LIST_READ _ONLY	显示发生软件行程限位出错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)
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

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 ☞ 63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。 ■ I0编号(StartI0) 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”。 ■ 滤波器时间(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	软件行程限位超驰	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) ☞ 75页 MC_POS_SOURCE	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) 轴组动作中, 构成轴的目标加速度 (AxisName, Md.TargetAcceleration)中将存储“0.0”。
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) 轴组动作中, 构成轴的目标减速度 (AxisName, Md.TargetDeceleration)中将存储“0.0”。
TargetVelocity	目标速度	运算周期	LREAL	LIST_READ_ONLY	存储考虑了超驰、速度限制值的实际的目标速度。 ■ 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值, 向负方向(地址减少方向)移动的情况下将为负值。 控制完成时将存储“0.0”。 轴组动作中, 构成轴的目标速度 (AxisName, Md.TargetVelocity)中将存储“0.0”。
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。

变量名	名称	获取	数据类型	属性	内容
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。 <ul style="list-style-type: none"> • 0000000H: pulse • 0001000H: m • FD01000H: mm • FA01000H: μm • F701000H: nm • 0041000H: degree • FA41000H: $\times 10^{-6}$degree • 00B4000H: Revolution • 00C0000H: inch • 00FF000H: 任意单位
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	在位置指令单位 (<u>AxisName</u> .Pr.Unit_Position) 中设置了“00FF000H: 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
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_LIM IT_MODE) ☞ 73页 MC_VELOCITY_LIM IT_MODE	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	速度超驰系数	运算周期	LREAL	LIST_READ _ONLY	显示速度超驰系数。
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警告代码。

*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	加速度超驰系数	运算周期	LREAL	LIST_READ _ONLY	设置加速度超驰系数。 • 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	轴出错复位	服务	BOOL	LIST_READ _ONLY	对轴的出错、警告、驱动器模块出错进行复位。出错复位完成时将变为FALSE。 • FALSE: 无执行 • TRUE: 执行出错复位
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	硬件行程限位超驰	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	软件行程限位超驰	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	速度超驰系数	运算周期	LREAL	LIST_READ _ONLY	设置速度超驰系数。 • 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)

○：可以设置，×：不能设置

变量名	名称	轴类型				
		实驱动轴	实编码器轴	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)

○：可以设置，×：不能设置

变量名	名称	轴类型				
		实驱动轴	实编码器轴	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	○	○	○	○	○
PositioningRange	定位范围	○	○	○	○	○
PosRestoration_AbsPosBase	绝对位置基准设置	○	×	×	×	×
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	从模拟有效	○	×	×	×	×
SlaveObject	从对象数据	○	○	×	○	×
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	○	×	×	×	×
FastOperationMode	高速模式设置	○	○	○	○	○

■AxisName.Pr. (轴参数)

○：可以设置， ×：不能设置

变量名	名称	轴类型				
		实驱动轴	实编码器轴	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	发生硬件行程限位出错时停止选择	○	×	○	×	×
StopMode_ServoOff	运行中伺服OFF指令时处理选择	○	×	○	×	×
StopMode_SwStrokeLimit	发生软件行程限位出错时停止选择	○	×	○	×	×
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	○	○	○	○	○
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)

○：可以设置， ×：不能设置

变量名	名称	轴类型				
		实驱动轴	实编码器轴	虚拟驱动轴	虚拟编码器轴	虚拟连接轴
AccelerationLimit	加速度限制值	○	×	○	×	×
AccelerationOverride	加速度超驰系数	○	×	○	×	×
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	转矩控制时指令当前转矩	○	×	×	×	×
Cst_TargetTorque	转矩控制时目标转矩	○	×	×	×	×
CumulativePosition	累计当前位置	○	○	○	○	○
DecelerationLimit	减速度限制值	○	×	○	×	×
Drive_RPDO[1..64]	RPDO(从设备→运动系统)映射	○	×	×	×	×
Drive_TPDO[1..64]	TPDO(运动系统→从设备)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	驱动器模块出错检测	○	×	×	×	×
DriverErrorID	驱动器模块出错代码	○	×	×	×	×
DriverErrorDetailID	驱动器模块出错详细代码	○	×	×	×	×
Encoder_Connected	连接状态	×	×	×	○	×
Encoder_CounterDisable	计数器无效中	×	○	×	○	×
Error	轴出错检测	○	○	○	○	○
ErrorID	轴出错代码	○	○	○	○	○
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	硬件行程限位超驰	○	×	○	×	×

变量名	名称	轴类型				
		实驱动轴	实编码器轴	虚拟驱动轴	虚拟编码器轴	虚拟连接轴
HwStrokeLimit_RlsStatus	下限限位信号状态	○	×	○	×	×
InVelocity	目标速度到达	○	×	○	×	×
Io_PosActualValue	对象数据_PosActualValue	○	○	×	○	×
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	从模拟中	○	×	×	×	×
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	发生硬件行程限位出错时停止选择	○	×	○	×	×
StopMode_ServoOff	运行中伺服OFF指令时处理选择	○	×	○	×	×
StopMode_SwStrokeLimit	发生软件行程限位出错时停止选择	○	×	○	×	×
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	软件行程限位超驰	○	×	○	×	×
SwStrokeLimit_Target	软件行程限位对象	○	×	○	×	×
SwStrokeLimit_Upper	软件行程限位上限值	○	×	○	×	×
Sync_PhaseShift	相位补偿量	○	×	○	×	○
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	○	○	○	○	○
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	速度超驰系数	○	×	○	×	×

变量名	名称	轴类型				
		实驱动轴	实编码器轴	虚拟驱动轴	虚拟编码器轴	虚拟连接轴
Warning	轴警告检测	○	○	○	○	○
WarningID	轴警告代码	○	○	○	○	○

■AxisName.Cd. (轴控制数据)

○：可以设置，×：不能设置

变量名	名称	轴类型				
		实驱动轴	实编码器轴	虚拟驱动轴	虚拟编码器轴	虚拟连接轴
AccelerationOverride	加速度超驰系数	○	×	○	×	×
Encoder_Connect	连接指令	×	×	×	○	×
Encoder_CounterDisable	计数器无效中	×	○	×	○	×
Encoder_InputValue	编码器输入值	×	×	×	○	×
ErrorReset	轴出错复位	○	○	○	○	○
FollowupDisable	跟踪无效中	○	×	×	×	×
Homing_ClearRequest	原点复位请求清除	○	○	○	○	○
HwStrokeLimit_Override	硬件行程限位超驰	○	×	○	×	×
SwStrokeLimit_Override	软件行程限位超驰	○	×	○	×	×
TorqueLimit_Negative	负方向转矩限制值	○	×	×	×	×
TorqueLimit_Positive	正方向转矩限制值	○	×	×	×	×
VelocityOverride	速度超驰系数	○	×	○	×	×

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成员。

成员名	数据类型	内容	参阅
AxesGroupRef	AXES_GROUP_REF	是运动控制用功能块的输入/输出用的数据结构。	☞ 44页 AxesGroupName.AxesGroupRef. (轴组信息)
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。	☞ 44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。	☞ 46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。	☞ 49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.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	设置加速度限制值。 • 0.0000、0.0001~2147483647.0 *: 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。 *: “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) ☞ 75页 MC_ACC_ZERO_MODE	LIST_WRITE_BACK	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。 • -1: 出错(不启动)(ACCError) • 1: 最大加减速(MaximumAcceleration)
Axis[1..16]	构成轴	就绪ON	AXIS_REF ☞ 23页 AxisName.AxisRef. (轴信息)	LIST_WRITE_BACK	设置构成轴组的轴信息(AxisName.AxisRef)的轴No.(AxisNo)。
CmdInPos_Width	指令到位宽度	就绪ON	LREAL	LIST_WRITE_BACK	设置轴组的合成轴上的指令到位宽度。 • 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。

变量名	名称	获取	数据类型	属性	内容
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) ☞ 75页 MC_DECEL_STOP_MODE	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。</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"> • 0.0000、0.000001~8400.0 <p>*: 与加加速度0指定时动作选择 (AccelerationZeroBehavior)无关, “0”的情况下将立即停止。</p> <p>*: 发生停止原因时超出上限值范围的情况下将以上限值限定, 超出下限值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) ☞ 75页 MC_STOP_CURVE_MODE	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	发生轴停止原因时构成轴动作选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 75页 MC_DECEL_STOP_MODE	LIST_WRITE_BACK	<p>设置以轴组运行中, 构成轴中变为驱动器伺服OFF, 且发生了立即停止的轴出错的情况下, 未发生轴出错的轴的动作。</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) ☞ 75页 MC_DECEL_STOP_MODE	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>
Unit_Position	位置指令单位	轴组有效时	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 • FA410000H: $\times 10^{-6}$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_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"> • 0000300H: U/s • 00004700H: U/min • FD000300H: U/ms • FA000300H: U/μs • F7000300H: U/ns
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	加速度超驰系数	运算周期	LREAL	LIST_READ_ONLY	存储加速度超驰系数。
AccelerationZeroBehavior	启动时加减速度0指定时动作选择	就绪ON	INT (MC_ACC_ZERO_MODE) ☞ 75页 MC_ACC_ZERO_MODE	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。 <ul style="list-style-type: none"> • -1: 出错(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。 是构成轴的反馈速度的合成速度。与构成轴的移动方向无关, 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。 <ul style="list-style-type: none"> • 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。 <ul style="list-style-type: none"> • FALSE: 不处于自动减速中 • TRUE: 自动减速中
Axis[1..16]	构成轴	即时	AXIS_REF ☞ 23页 AxisName.AxisRef. (轴信息)	LIST_READ_ONLY	存储构成轴组的轴信息(AxisName.AxisRef)的轴No.(AxisNo)。
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	显示至目标位置为止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)。 <ul style="list-style-type: none"> • FALSE: 大于指令到位宽度 • TRUE: 小于或等于指令到位宽度

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	显示轴组的合成轴上的指令到位宽度。 将为获取了指令到位宽度 (AxesGroupName, Pr.CmdInPos_Width)的设置值后的值。 指令到位宽度 (AxesGroupName, Md.CmdInPos_Width)为“0.0”的情况下，相应轴组的指令到位功能将变为无效。
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)
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。
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)
CommandedVelocity	指定速度	运算周期 (GroupEnable中)	LREAL	LIST_READ_ONLY	存储控制中的动作系统的运动控制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	轴组出错检测	即时	BOOL	LIST_READ_ONLY	显示轴组出错发生的状态。 • FALSE: 无轴组出错 • TRUE: 有轴组出错
ErrorID	轴组出错代码	即时	WORD (HEX)	LIST_READ_ONLY	显示发生的轴组出错的出错代码。
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) ☞ 72页 MC_AXES_GROUP_STATUS	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: 出错停止中(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	DWORD (HEX)	LIST_READ_ONLY	以位显示轴组中插补控制执行中的构成轴。
InVelocity	目标速度到达	运算周期	BOOL	LIST_READ_ONLY	显示是否到达了目标速度。 • 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	构成轴数	即时	INT	LIST_READ_ONLY	显示轴组的构成轴数。
OverrunOperation	越程时动作设置	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 75页 MC_DECEL_STOP_MODE	LIST_READ_ONLY	显示越程动作设置的输入状态。 • 1: 立即停止(ImmediateStop)

变量名	名称	获取	数据类型	属性	内容
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)。(单位: U/s ²) 通过符号表示是加速中还是减速中。 到达目标速度时存储“0.0”。 • 0.0: 停止中或定速中 • 正符号: 加速中 • 负符号: 减速中 *: 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。 与构成轴的移动方向无关, 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) ☞ 75页 MC_STOP_CURVE_MODE	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	发生轴停止原因时构成轴动作选择	就绪ON	INT (MC_DECEL_STOP_MODE) ☞ 75页 MC_DECEL_STOP_MODE	LIST_READ _ONLY	显示发生轴停止原因时构成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) ☞ 75页 MC_DECEL_STOP_MODE	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)
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)
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)
TargetVelocity	目标速度	运算周期 (GroupEnable中)	LREAL	LIST_READ _ONLY	存储考虑了超驰、速度限制值的实际的目标速度。 与构成轴的移动方向无关, 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。
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。 • 0000000H: pulse • 0001000H: m • FD01000H: mm • FA01000H: μm • F701000H: nm • 0041000H: degree • FA41000H: ×10 ⁻⁶ degree • 00B4000H: Revolution • 00C0000H: inch • 00FF000H: 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。

变量名	名称	获取	数据类型	属性	内容
Unit_PositionString	位置指令单位字符串	轴组有效时	WSTRING(31)	LIST_READ _ONLY	在位置指令单位(AxesGroupName, Pr. Unit_Position)中设置了“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(AxesGroupName, Pr. Unit_PositionString)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的速度指令单位。 <ul style="list-style-type: none"> • 00003000H: 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。
VelocityLimit	速度限制值	就绪ON	LREAL	LIST_READ _ONLY	存储轴组的速度限制值。
VelocityOverride	速度超驰系数	运算周期	LREAL	LIST_READ _ONLY	显示速度超驰系数。
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警告代码。

AxesGroupName. Cd. (轴组控制数据)

变量名	名称	获取	数据类型	属性	内容
AccelerationOverride	加速度超驰系数	运算周期	LREAL	LIST_READ _ONLY	设置加速度超驰系数。 <ul style="list-style-type: none"> • 0.01~10.00
ErrorReset	轴组出错复位	服务	BOOL	LIST_READ _ONLY	对轴组的出错、警告进行复位。 <ul style="list-style-type: none"> • FALSE: 无执行 • TRUE: 执行出错复位
VelocityOverride	速度超驰系数	运算周期	LREAL	LIST_READ _ONLY	设置速度超驰系数。 <ul style="list-style-type: none"> • 0.00~10.00

系统变量

系统变量的数据类型以SYSTEM型表示。

系统的数据类型具有下述成员。

成员名	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置值。系统变量初始化后不实施至控制的重新获取。	☞ 50页 System.PrConst. (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。	☞ 53页 System.Pr. (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。	☞ 54页 System.Md. (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。	☞ 58页 System.Cd. (系统控制数据)
LoggingRef	LOGGING_REF[1..10]	是记录控制用功能块的输入/输出用的数据结构。	☞ 59页 System.LoggingRef.

System.PrConst. (系统参数常数)

变量名	名称	获取	数据类型	属性	内容
Addon_AbsSystem	插件AbsSystem参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件AbsSystem参数(Addon_AbsSystem)中使用的存储器的最大使用量。
Addon_Axis	插件Axis参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件Axis参数(Addon_Axis)中使用的存储器的最大使用量。
Addon_ExternalSignal	插件ExternalSignal参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件ExternalSignal参数(Addon_ExternalSignal)中使用的存储器的最大使用量。
Addon_FileTransfer	插件FileTransfer参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件FileTransfer参数(Addon_FileTransfer)中使用的存储器的最大使用量。
Addon_Logging	插件Logging参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件Logging参数(Addon_Logging)中使用的存储器的最大使用量。
Addon_MotionControl_AxisFilter	插件MotionControl_AxisFilter参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件MotionControl_AxisFilter参数(Addon_MotionControl_AxisFilter)中使用的存储器的最大使用量。
Addon_MotionControl_General	插件MotionControl_General参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件MotionControl_General参数(Addon_MotionControl_General)中使用的存储器的最大使用量。
Addon_MotionControl_Sync	插件MotionControl_Sync参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件MotionControl_Sync参数(Addon_MotionControl_Sync)中使用的存储器的最大使用量。
Addon_MotionEngine	插件MotionEngine参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件MotionEngine参数(Addon_MotionEngine)中使用的存储器的最大使用量。
Addon_MotionEventHist	插件MotionEventHist参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件MotionEventHist参数(Addon_MotionEventHist)中使用的存储器的最大使用量。
Addon_NetworkDriver_CCIETSN	插件NetworkDriver_CCIETSN参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件NetworkDriver_CCIETSN参数(Addon_NetworkDriver_CCIETSN)中使用的存储器的最大使用量。
Addon_PackagingApp	插件PackagingApp参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件PackagingApp参数(Addon_PackagingApp)中使用的存储器的最大使用量。
Addon_PlcInstruction	插件PlcInstruction参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件PlcInstruction参数(Addon_PlcInstruction)中使用的存储器的最大使用量。
Addon_ProfileControl	插件ProfileControl参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件ProfileControl参数(Addon_ProfileControl)中使用的存储器的最大使用量。

变量名	名称	获取	数据类型	属性	内容
Addon_Program_ST	插件Program_ST参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件Program_ST参数 (Addon_Program_ST) 中使用的存储器的最大使用量。
Addon_ServoDriver_CANopen	插件ServoDriver_CANopen参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件ServoDriver_CANopen参数 (Addon_ServoDriver_CANopen) 中使用的存储器的最大使用量。
Addon_ServoSystemRecorder	插件ServoSystemRecorder参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件ServoSystemRecorder参数 (Addon_ServoSystemRecorder) 中使用的存储器的最大使用量。
Addon_SignalIO	插件SignalIO参数	系统启动时	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_WRIT E_BACK	设置插件SignalIO参数 (Addon_SignalIO) 中使用的存储器的最大使用量。
BuffermemoryRefreshCycle	缓冲存储器刷新周期设置	系统启动时	CYCLE_PARAM ☞ 64页 CYCLE_PARAM	LIST_WRIT E_BACK	设置缓冲存储器刷新周期的各设置值。 缓冲存储器刷新周期设置 (BuffermemoryRefreshCycle) 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 • -11: 与网络的基本周期同步 • -12: 与网络的中速周期同步 • -13: 与网络的低速周期同步 • -11001~ -118192: 第1运算周期的n倍 (1~8192)
EventHistoryCpu_Exclude	CPU事件履历除外事件	系统启动时	STRING (255)	LIST_WRIT E_BACK	设置不进行至CPU模块的事件登录的事件代码。
EventHistoryMotion_Capacity	运动事件履历容量	系统启动时	INT	LIST_WRIT E_BACK	设置事件履历文件的容量。 <ul style="list-style-type: none"> • 1~2048[k字节]
EventHistoryMotion_Exclude	运动事件履历除外事件	系统启动时	STRING (255)	LIST_WRIT E_BACK	设置不进行至运动系统的事件登录的事件代码。
EventHistoryMotion_Path	运动事件履历路径	系统启动时	STRING (127)	LIST_WRIT E_BACK	设置事件履历文件的保存目标。
ExcludeWarning	除外报警	系统启动时	STRING (255)	LIST_WRIT E_BACK	设置不检测的警告。 <例> <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
FastOperationMode	高速模式设置	系统启动时	WORD (HEX)	LIST_WRIT E_BACK	将整个系统的运算处理切换至高速模式。 <ul style="list-style-type: none"> • 50EFH: 高速模式 • 上述以外: 正常模式
FileTransfer_AcFile_lch	文件传送访问控制(锁存驱动器内文件)	系统启动时	WORD (HEX)	LIST_READ _ONLY	设置对锁存驱动器内文件的访问控制。 <ul style="list-style-type: none"> • 0: 读取禁止/写入禁止 • 1: 读取禁止/写入允许 • 2: 读取允许/写入禁止 • 3: 读取允许/写入允许
FileTransfer_AcFile_ram	文件传送访问控制(RAM驱动器内文件)	系统启动时	WORD (HEX)	LIST_READ _ONLY	设置对RAM驱动器内文件的访问控制。 <ul style="list-style-type: none"> • 0: 读取禁止/写入禁止 • 1: 读取禁止/写入允许 • 2: 读取允许/写入禁止 • 3: 读取允许/写入允许
FileTransfer_AcFile_rom	文件传送访问控制(用户驱动器内文件)	系统启动时	WORD (HEX)	LIST_READ _ONLY	设置对用户驱动器内文件的访问控制。 <ul style="list-style-type: none"> • 0: 读取禁止/写入禁止 • 1: 读取禁止/写入允许 • 2: 读取允许/写入禁止 • 3: 读取允许/写入允许

变量名	名称	获取	数据类型	属性	内容
FileTransfer_AcFile_sdc	文件传送访问控制(SD存储卡内文件)	系统启动时	WORD (HEX)	LIST_READ_ONLY	设置对SD存储卡内文件的访问控制。 <ul style="list-style-type: none"> 0: 读取禁止/写入禁止 1: 读取禁止/写入允许 2: 读取允许/写入禁止 3: 读取允许/写入允许
FileTransfer_AcFile_sys	文件传送访问控制(系统驱动器内文件)	系统启动时	WORD (HEX)	LIST_READ_ONLY	设置对系统驱动器内文件的访问控制。 <ul style="list-style-type: none"> 0: 读取禁止/写入禁止 1: 读取禁止/写入允许 2: 读取允许/写入禁止 3: 读取允许/写入允许
FileTransfer_AcLabel	文件传送访问控制(标签)	系统启动时	WORD (HEX)	LIST_READ_ONLY	设置对标签的访问控制。 <ul style="list-style-type: none"> 0: 读取禁止/写入禁止 1: 读取禁止/写入允许 2: 读取允许/写入禁止 3: 读取允许/写入允许
FileTransfer_LogCapacity	文件传送日志容量	系统启动时	INT	LIST_WRITE_BACK	设置日志文件的容量。 <ul style="list-style-type: none"> -1~2048[k字节]
LabelMemorySize	标签存储器容量	系统启动时	INT	LIST_WRITE_BACK	设置标签存储器容量。 <ul style="list-style-type: none"> 128~8192[k字节]
Link_MotionStationRefreshType	运动控制站发送接收数据更新方法	系统启动时	INT (MC_STATION_REFRESH_MODE) <small>☞ 76页</small> MC_STATION_REFRESH_MODE	LIST_WRITE_BACK	设置运动控制站的发送接收数据的更新方法。 <ul style="list-style-type: none"> 1: 运算周期优先方式 (EmphasisOperationCycle)
Logging_AutoLoggingEnable	自动记录允许	系统启动时	BOOL	LIST_READ_ONLY	设置记录设置文件被存储到SD存储卡中时的自动记录功能的禁止、允许。 <ul style="list-style-type: none"> FALSE: 禁止 TRUE: 允许
MeasuresPrograms	程序信息测量标志	系统启动时	BOOL	LIST_WRITE_BACK	设置是否对程序的执行时间、程序的执行次数、程序的执行状态进行测量。 测量数据将存储到各程序的PROGRAM_INFO结构体中。 <ul style="list-style-type: none"> FALSE: 不测量 TRUE: 测量
OperationCycle[1]	运算周期设置	系统启动时	CYCLE_PARAM <small>☞ 64页</small> CYCLE_PARAM	LIST_WRITE_BACK	设置第1运算周期的各设置值。 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 -11: 与网络的基本周期同步 -12: 与网络的中速周期同步 -13: 与网络的低速周期同步 -100001~-100256: 系统基本处理的n倍(1~256)
SoftReboot_Enable	软重启允许	系统启动时	BOOL	LIST_WRITE_BACK	设置软重启的执行允许、执行禁止。 禁止的情况下, 不执行软重启。 <ul style="list-style-type: none"> FALSE: 禁止 TRUE: 允许
MultiThreadMode	制造商设置用	系统启动时	WORD (HEX)	LIST_WRITE_BACK	—

System.Pr. (系统参数)

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 ☞ 63页 SIGNAL_SELECT (信号选择)	LIST_WRIT E_BACK	<p>设置使用所有轴紧急停止的信号。</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>■ I0编号 (StartI0) 忽略输入值。</p> <p>■ 对象 (Target) 无设置时, 将判断为信号无效且置为始终信号未检测状态。</p> <p>*: 设置了不能使用的数据类型的前提下, 将变为“超出参数范围(系统)(出错代码: 1D82H)”。</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"> • 0: TRUE时检测 (HighLevel) • 1: FALSE时检测 (LowLevel) <p>*: 设置了边缘检测的情况下, 将变为“超出参数范围(系统)(出错代码: 1D82H)”。</p> <p>■ 补偿时间 (CompensationTime) 忽略输入值。</p> <p>■ 滤波器时间 (FilterTime) 设置范围为“0.0~5.0[s]”。</p> <p>*: 设置了超出范围的值的情况下, 将变为“超出系统信号的滤波器时间设置范围警告(警告代码: 0F0FH)”, 且滤波器时间将以“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) ☞ 75页 MC_DECEL_STOP_MODE	LIST_WRIT E_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_WRIT E_BACK	<p>指定发生了所有轴停止原因时的减速停止时的减速度。</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>*: 与加减速速度0指定时动作选择无关, “0”的情况下将立即停止。</p> <p>*: 发生停止原因时超出上限值范围的情况下将以上限值限定, 超出下限值范围的情况下将作为“0(立即停止)”处理。</p>

System.Md. (系统监视数据)

变量名	名称	获取	数据类型	属性	内容
Addon_AbsSystem	插件AbsSystem监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件AbsSystem参数 (Addon_AbsSystem) 中使用的存储器的最大使用量。
Addon_Axis	插件Axis监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件Axis参数 (Addon_Axis) 中使用的存储器的最大使用量。
Addon_ExternalSignal	插件ExternalSignal监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件ExternalSignal参数 (Addon_ExternalSignal) 中使用的存储器的最大使用量。
Addon_FileTransfer	插件FileTransfer监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件FileTransfer参数 (Addon_FileTransfer) 中使用的存储器的最大使用量。
Addon_Logging	插件Logging监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件Logging参数 (Addon_Logging) 中使用的存储器的最大使用量。
Addon_MotionControl_AxisFilter	插件MotionControl_AxisFilter监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件MotionControl_AxisFilter参数 (Addon_MotionControl_AxisFilter) 中使用的存储器的最大使用量。
Addon_MotionControl_General	插件MotionControl_General监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件MotionControl_General参数 (Addon_MotionControl_General) 中使用的存储器的最大使用量。
Addon_MotionControl_Sync	插件MotionControl_Sync监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件MotionControl_Sync参数 (Addon_MotionControl_Sync) 中使用的存储器的最大使用量。
Addon_MotionEngine	插件MotionEngine监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件MotionEngine参数 (Addon_MotionEngine) 中使用的存储器的最大使用量。
Addon_MotionEventHist	插件MotionEventHist监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件MotionEventHist参数 (Addon_MotionEventHist) 中使用的存储器的最大使用量。
Addon_NetworkDriver_CCIETSN	插件NetworkDriver_CCIETSN监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件NetworkDriver_CCIETSN参数 (Addon_NetworkDriver_CCIETSN) 中使用的存储器的最大使用量。
Addon_PackagingApp	插件PackagingApp监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件PackagingApp参数 (Addon_PackagingApp) 中使用的存储器的最大使用量。
Addon_PlcInstruction	插件PlcInstruction监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件PlcInstruction参数 (Addon_PlcInstruction) 中使用的存储器的最大使用量。
Addon_ProfileControl	插件ProfileControl监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件ProfileControl参数 (Addon_ProfileControl) 中使用的存储器的最大使用量。
Addon_Program_ST	插件Program_ST监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件Program_ST参数 (Addon_Program_ST) 中使用的存储器的最大使用量。
Addon_ServoDriver_CANopen	插件ServoDriver_CANopen监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件ServoDriver_CANopen参数 (Addon_ServoDriver_CANopen) 中使用的存储器的最大使用量。
Addon_ServoSystemRecorder	插件ServoSystemRecorder监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件ServoSystemRecorder参数 (Addon_ServoSystemRecorder) 中使用的存储器的最大使用量。
Addon_SignalIO	插件SignalIO监视	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	显示插件SignalIO参数 (Addon_SignalIO) 中使用的存储器的最大使用量。
AddonLimit_Level	插件库配置限制等级	即时	INT	LIST_READ _ONLY	显示设置的限制等级。 • 0: Low • 20: High

变量名	名称	获取	数据类型	属性	内容
AddonLimit_State	插件库配置限制状态	即时	INT (MC_ADDON_LIMIT_STATE) ☞ 77页 MC_ADDON_LIMIT_STATE	LIST_READ _ONLY	显示插件库配置限制状态。 • 0: 无限制 (NoLimitation) • 1: 限制中 (Limitation) • 10: 解除中 (TemporaryNoLimitation) • -1: 锁定中 (Lockout) *: “10: 解除中 (TemporaryNoLimitation)”、“-1: 锁定中 (Lockout)”的情况下, 重新接通电源时将返回到操作前的状态。
BackupRestoreStatus	备份·还原信息	服务	INT	LIST_READ _ONLY	显示备份·还原的状况。 • 0: 备份还原未实施 • 1: 备份停止请求中 • 2: 备份停止中 • 3: 备份停止解除中 • 4: 恢复数据保存中 • 5: 恢复数据保存完成
BuffermemoryFreeMcfbArea	缓冲存储器MCFB通信区域空余容量	即时	DWORD (UDINT)	LIST_READ _ONLY	显示运动控制FB用区域的空余点数。
BuffermemoryRefreshCycle	缓冲存储器刷新周期监视	即时	CYCLE_MONI ☞ 64页 CYCLE_MONI	LIST_READ _ONLY	存储缓冲存储器刷新周期的各监视值。
Error	运动部系统出错检测	即时	BOOL	LIST_READ _ONLY	显示出错的状态。 • FALSE: 无出错 • TRUE: 有出错
ErrorID	运动部最新系统出错代码	即时	WORD (HEX)	LIST_READ _ONLY	显示最新的出错代码。
EventHistoryMotion_ClearStatus	运动事件履历清除执行状态	服务	INT (MC_EXECUTION_STATE) ☞ 76页 MC_EXECUTION_STATE	LIST_READ _ONLY	显示事件履历清除的执行状态。 • 0: 执行请求等待 (Ready) • 1: 执行中 (Executing) • 2: 执行完成 (Done) • -1: 出错发生 (Error_)
EventHistoryMotion_Path	运动事件履历路径	系统启动时	STRING (127)	LIST_READ _ONLY	存储事件履历文件的保存目标 (绝对路径)。
FileTransfer_State	文件传送执行状态	服务	INT (MC_EXECUTION_STATE) ☞ 76页 MC_EXECUTION_STATE	LIST_READ _ONLY	显示文件传送的执行状态。 • 0: 执行请求等待 (Ready) • 1: 执行中 (Executing) • -1: 出错发生 (Error_)
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 ☞ 63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。 ■IO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”。 ■滤波器时间 (FilterTime) 显示参数的获取结果。
HwStatus_OperationTime	通电时间累计	服务	DWORD (UDINT)	LIST_READ _ONLY	存储运动模块的通电时间累计 (分钟)。
HwStatus_RomWriteCycle	闪存写入次数指标值	服务	DWORD (UDINT)	LIST_READ _ONLY	存储闪存的写入次数指标值。
LabelMemoryFreeSize	标签存储器空余容量	即时	INT	LIST_READ _ONLY	存储标签存储器的空余容量。
MemorySize	系统存储器容量	服务	ADDON_PARAM ☞ 64页 ADDON_PARAM	LIST_READ _ONLY	存储系统存储器 (RAM)、系统存储器 (备份RAM) 的最大使用容量。

变量名	名称	获取	数据类型	属性	内容
MemoryUsage	系统存储器空余信息	服务	ADDON_MONI ☞ 64页 ADDON_MONI	LIST_READ _ONLY	存储系统存储器(RAM)、系统存储器(备份RAM)的当前使用量、最大使用量。
NetworkError	网络部出错检测	即时	BOOL	LIST_READ _ONLY	显示网络出错发生的状态。 • FALSE: 无网络出错 • TRUE: 有网络出错
NetworkErrorID	网络部出错代码	即时	WORD(HEX)	LIST_READ _ONLY	存储发生的网络出错的出错代码。
OperationCycle[1]	运算周期监视	即时	CYCLE_MONI ☞ 64页 CYCLE_MONI	LIST_READ _ONLY	存储第1运算周期的监视值。
Program_Ei	EI标志	即时	BOOL	LIST_READ _ONLY	显示恒定周期执行类型程序的指令执行状态。 • FALSE: DI(禁止)中 • TRUE: EI(允许)中
Program_NormalTaskCycle	正常任务处理时间监视	正常	CYCLE_MONI ☞ 64页 CYCLE_MONI	LIST_READ _ONLY	存储正常任务的监视值。 正常任务处理时间监视 (Program_NormalTaskCycle)固有的输出如下所示。 ■设置周期(Cycle) 由于不存在周期设置,因此将始终存储“0”。 ■周期溢出(CycleOver) 由于不进行周期溢出检测,因此将始终变为FALSE。
Ready	准备就绪	服务	BOOL	LIST_READ _ONLY	显示“准备就绪(X0)”的ON/OFF状态。 • FALSE: OFF(准备未就绪) • TRUE: ON(准备就绪)
ServoSystemRecorder_Enable	伺服系统记录器有效	服务	BOOL	LIST_READ _ONLY	显示通过伺服系统记录器的记录采集是有效还是无效。 • FALSE: 无效 • TRUE: 有效
StopMode_All	发生所有轴停止原因时 停止选择	就绪ON	INT (MC_DECEL_STOP_ MODE) ☞ 75页 MC_DECEL_STOP_M ODE	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。
Storage_lch	锁存驱动器信息	服务	STORAGE_INFORMA TION ☞ 64页 STORAGE_INFORMA TION	LIST_READ _ONLY	显示锁存驱动器的信息。
Storage_ram	RAM驱动器信息	服务	STORAGE_INFORMA TION ☞ 64页 STORAGE_INFORMA TION	LIST_READ _ONLY	显示RAM驱动器的信息。
Storage_rom	用户驱动器信息	服务	STORAGE_INFORMA TION ☞ 64页 STORAGE_INFORMA TION	LIST_READ _ONLY	显示用户驱动器的信息。
Storage_sdc	SD存储卡信息	服务	STORAGE_INFORMA TION ☞ 64页 STORAGE_INFORMA TION	LIST_READ _ONLY	显示SD存储卡的信息。
Storage_sdcForcedDisabled	SD存储卡强制使用停止	服务	BOOL	LIST_READ _ONLY	显示SD存储卡的强制使用停止的状态。 • FALSE: 未强制使用停止 • TRUE: 已强制使用停止

变量名	名称	获取	数据类型	属性	内容
Storage_sdcInserted	SD存储卡安装	服务	BOOL	LIST_READ _ONLY	显示SD存储卡的安装状态。 • FALSE: 未安装 • TRUE: 已安装
Storage_sdcProtected	SD存储卡保护	服务	BOOL	LIST_READ _ONLY	显示SD存储卡的写保护开关ON/OFF的状态。 • FALSE: 写保护开关OFF • TRUE: 写保护开关ON
Storage_sys	系统驱动器信息	服务	STORAGE_INFORMATION 64页 STORAGE_INFORMATION	LIST_READ _ONLY	显示系统驱动器的信息。
Sync	同步用标志	服务	BOOL	LIST_READ _ONLY	显示“同步用标志(X1)”的ON/OFF状态。 • FALSE: OFF(模块访问禁止) • TRUE: ON(模块访问允许)
SystemBaseCycle	系统基本周期监视	即时	CYCLE_MONI 64页 CYCLE_MONI	LIST_READ _ONLY	存储系统基本周期的监视值。
SystemBaseCycle_Counter	系统基本周期计数器	系统基本周期	WORD(UINT)	LIST_READ _ONLY	存储系统基本周期计数器。
Version_BaseSystemSw	基本系统软件版本	系统启动时	WORD(UINT)	LIST_READ _ONLY	显示基本系统软件版本。
Version_BootSw	启动软件版本	系统启动时	WORD(UINT)	LIST_READ _ONLY	显示启动软件版本。
Version_NetworkBootSw	网络启动软件版本	系统启动时	WORD(UINT)	LIST_READ _ONLY	显示网络的启动软件版本。
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	存储最新的警告代码。
WDTerror	WDT出错	即时	BOOL	LIST_READ _ONLY	显示WDT出错的状态。 • FALSE: 无WDT出错 • TRUE: 有WDT出错

System.Cd. (系统控制数据)

变量名	名称	获取	数据类型	属性	内容
AddonLimit_Key	插件库配置限制密钥设置	即时	STRING(31)	LIST_READ_ONLY	设置限制插件库配置的密钥。
AddonLimit_Level	插件库配置限制等级	即时	INT	LIST_READ_ONLY	设置限制插件库配置的等级。 • 0: Low • 20: High
AddonLimit_Regist	插件库配置限制请求	服务	INT (MC_ADDON_LIMIT_CMD) 77页 MC_ADDON_LIMIT_CMD	LIST_READ_ONLY	设置插件库配置的限制的状态。 • 0: 无(_None) • 1: 登录(Regist) • 10: 暂时解除(TemporaryNoLimitation) • -1: 初始化(Initialize)
BackupRestore	备份·还原请求	服务	WORD(HEX)	LIST_READ_ONLY	执行备份停止及还原数据的保存。 • 0000H: 无请求 • ABCDH: 备份停止请求 • FFFEH: 恢复数据保存请求
ErrorReset	系统出错复位	服务	BOOL	LIST_READ_ONLY	对所有的出错、警告进行复位。 • FALSE: 无执行 • TRUE: 执行系统出错复位
EventHistoryMotion_Clear	运动事件履历清除执行请求	服务	BOOL	LIST_READ_ONLY	进行事件履历清除的执行请求。 • FALSE: 无执行 • TRUE: 执行清除
FileTransfer_Command	文件传送指令	服务	STRING(255)	LIST_READ_ONLY	设置文件传送的指令。 文件传送完成后, 将自动清除。
FileTransfer_Execute	文件传送执行请求	服务	BOOL	LIST_READ_ONLY	进行文件传送的执行请求。 文件传送完成后, 将自动变为FALSE。 • FALSE: 无执行 • TRUE: 执行文件传送
SequenceReady*1*2	可编程控制器就绪	服务	BOOL	LIST_READ_ONLY	设置“可编程控制器就绪(Y0)”的RUN/STOP。 • FALSE: STOP • TRUE: RUN *: 运动模块的“Y0”的上升沿/下降沿时将自动变为TRUE/FALSE。
SoftRebootRequest	软重启指令	服务	STRING(15)	LIST_READ_ONLY	执行软重启。 • MOTION_RESET: 仅运动部执行复位。 • MOTION_CLEAR: 运动部重启后执行存储器的清除。 • MOTION_QCLEAR: 运动部重启后执行存储器的快速清除。 • 上述以外: 不执行任何操作。 *: 软重启完成后(系统重启后)将删除。
Storage_sdcForcedDisable	SD存储卡强制使用停止指示	服务	BOOL	LIST_READ_ONLY	设置是否停止SD存储卡的强制使用。 • FALSE: 停止解除 • TRUE: 强制停止
Storage_sdcRemovalProhibit	SD存储卡拆装禁止	服务	BOOL	LIST_READ_ONLY	设置是否禁止拆装SD存储卡。 • FALSE: 允许拆装 • TRUE: 禁止拆装

*1 将SequenceReady(可编程控制器就绪)设置为公开标签时, 请勿操作运动模块的“Y0”。否则STOP/RUN可能无法正确切换。

*2 CPU模块的STOP/RUN状态与运动模块的STOP/RUN状态不联动。即使在CPU模块侧的运动控制FB中进行了轴控制的状态下, 将CPU模块置为STOP, 轴控制也不会停止。应根据需要通过运动模块侧的程序监视CPU模块的动作状态, 并进行轴控制的停止。CPU模块的动作状态可以在“本站CPU动作状态(SB004C)”、“本站CPU状态(SW004B)”中进行确认。此外, 在CPU模块中发生了停止型出错的情况下, 运动模块也将自动变为STOP。

System.LoggingRef.

记录控制变量的数据类型以LOGGING_REF型表示。

记录控制的数据类型具有下述成员。

成员名	数据类型	内容	参阅
Cd	LOGGING_CMD	存储记录控制用指令数据。 各记录设置周期中获取最新的值用于控制。	☞ 59页 System.LoggingRef[1..10].Cd. (记录控制数据)
Md	LOGGING_MONI	存储记录控制的监视数据。 以各监视数据中确定的周期实施刷新。	☞ 59页 System.LoggingRef[1..10].Md. (记录监视数据)

■System.LoggingRef[1..10].Cd. (记录控制数据)

变量名	名称	获取	数据类型	属性	内容
SettingPath	记录设置路径	即时	STRING(127)	LIST_READ_ONLY	设置记录设置的存储目标路径(文件夹名、文件名)。
StartLog	记录请求	服务	BOOL	LIST_READ_ONLY	进行记录的执行请求。 • FALSE: STOP • TRUE: RUN
Trigger	触发请求	记录设置周期	BOOL	LIST_READ_ONLY	将触发条件设置为“触发指令”时, 请求触发。 • FALSE: 无请求 • TRUE: 有请求

■System.LoggingRef[1..10].Md. (记录监视数据)

变量名	名称	获取	数据类型	属性	内容
ErrStatus	出错状态	服务	WORD(UINT)	LIST_READ_ONLY	显示与记录相关的出错代码。
LogDataPath	记录数据文件路径	服务	STRING(127)	LIST_READ_ONLY	显示记录数据文件的路径(文件名)。
LogDataSavedCount	保存文件数	服务	WORD(UINT)	LIST_READ_ONLY	显示数据记录文件的保存数。
LoggingStatus	记录状态	服务	BOOL	LIST_READ_ONLY	显示记录的执行状态。 • FALSE: 停止 • TRUE: 执行中
LogType	记录类型	记录设置周期	INT (MC_LOGGING_TYPE) ☞ 76页 MC_LOGGING_TYPE	LIST_READ_ONLY	显示执行中的记录类型。 • 0: 无(_None) • 1: 连续记录(ContinuousLogging) • 2: 触发记录(TriggerLogging)
SamplingTime[0..1]	数据采集间隔	服务	DWORD(UDINT)	LIST_READ_ONLY	以ns单位显示数据采集间隔。 • [0]采集间隔 低位32位 • [1]采集间隔 高位32位
SaveStatus	保存状态	服务	INT (MC_LOGGING_SAVE_STATUS) ☞ 76页 MC_LOGGING_SAVE_STATUS	LIST_READ_ONLY	显示至采集数据的文件的保存状态。 • 0: 未保存(UnSave) • 1: 保存中(Saving) • 2: 保存完成(Saved) • 3: 全部文件保存完成(FullySaved)
SettingPath	记录中设置文件	即时	STRING(127)	LIST_READ_ONLY	显示记录中的记录设置文件的路径(文件名)。
TriggerCount	触发检测次数	记录设置周期	INT	LIST_READ_ONLY	显示触发的检测次数。
TrigerCountIgnored	触发忽略次数	记录设置周期	INT	LIST_READ_ONLY	显示触发后的数据采集中再次发生触发后, 忽略的触发的次数。
Triggered	触发发生	记录设置周期	BOOL	LIST_READ_ONLY	显示触发的发生状态。 • FALSE: 触发未发生 • TRUE: 触发发生
TriggerStatus[1..32]	触发状态	记录设置周期	BOOL	LIST_READ_ONLY	显示各触发条件的状态。 • FALSE: 触发条件未成立 • TRUE: 触发条件成立

变量名	名称	获取	数据类型	属性	内容
TriggerStatusWord	触发状态(字)	记录设置周期	DWORD (HEX)	LIST_READ _ONLY	以位(b0~b31)显示各触发条件的状态。反映与触发状态(TriggerStatus[1..32])相同的状态。

其它结构体的变量

MC_CAM_REF

变量名	名称	数据类型	内容
ProfileData	配置文件	PROFILE_DATA 63页 PROFILE_DATA	设置凸轮数据的配置文件。

MC_CAM_ID

变量名	名称	数据类型	内容
ProfileID	配置文件ID	PROFILE_ID 63页 PROFILE_ID	设置凸轮数据的配置文件ID。

MC_TRIGGER_REF

变量名	名称	数据类型	内容
Signal	触发信号	SIGNAL_SELECT 63页 SIGNAL_SELECT (信号选择)	设置触发信号。

MC_INPUT_REF

变量名	名称	数据类型	内容
Signal	输入信号	SIGNAL_SELECT 63页 SIGNAL_SELECT (信号选择)	设置要输入的信号。

MC_OUTPUT_REF

变量名	名称	数据类型	内容
Signal	输出信号	SIGNAL_SELECT 63页 SIGNAL_SELECT (信号选择)	设置要输出的信号。

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 <ul style="list-style-type: none"> • [类型]^{*1} • (类型)^{*2} • 数据名^{*1} • . 位位置^{*3} • @对象修饰^{*4} ■[类型] (类型) WSTRING型标签 <ul style="list-style-type: none"> • [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]	从设备的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枚举型的列举符。 在数据的(类型)中，只能设置“LREAL”。 • mcSetValue(指令当前值) • mcActualValue(反馈值) • mcLatestSetValue(最新指令当前值) • mcLatestActualValue(最新反馈值) *: 不能对本数据进行写入。	[AXIS]mcSetValue
[DEV]	运动系统内的软元件(包含缓冲存储器、本模块链接软元件)	[DEV] 软元件名	设置本模块内的软元件。可以设置下述软元件。 • 缓冲存储器(G) • 链接特殊继电器(SB) • 链接特殊寄存器(SW)	[DEV]RWr10.5 [DEV]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。可设置的数据的类型如下所示。

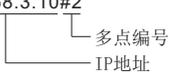
对于类型中设置的数据本身的类型为BOOL型以外的数据设置“(BOOL)”时，将作为省略了“位位置.0”的数据处理。

类型	设置示例
(BOOL)	■[DEV] (BOOL) G11500000的情况下 作为[DEV] (BOOL) G11500000.0处理。
(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(用于识别从设备的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。 @192.168.3.10#2  *: 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。 *: POU(程序部件): <u>Program Organization Unit</u>	
[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 对于参照值，每8周期(2³)进行TRUE与FALSE的切换。(将1字节的值每周期“+1”并获取位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“^”。)

SIGNAL_SELECT (信号选择)

变量名	名称	数据类型	内容
Source	信号	TARGET_REF ☞ 61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) ☞ 75页 MC_SIGNAL_LOGIC	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	滤波器时间	LREAL	以秒单位设置用于去除输入输出信号的振荡的滤波器时间。 *: 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 ☞ 63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 ☞ 63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	STRING (63)	设置文件名称。 (包括扩展名在内可以设置63字符。)*1
Path	文件夹指定	STRING (63)	设置存储文件的文件夹路径。 (最多可以设置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

CYCLE_PARAM

变量名	名称	数据类型	内容
Cycle	周期设置	DINT	设置周期。 <ul style="list-style-type: none"> • 0: 与网络的基本周期同步 • -11: 与网络的基本周期同步 • -12: 与网络的中速周期同步 • -13: 与网络的低速周期同步 • -100001~-100256: 系统基本处理的n倍(1~256) • -118192~-110001: 第1运算周期的n倍(1~8192)
NumOfCycleOverWngDetectTimes	周期溢出警告检测次数	WORD (UINT)	设置检测出各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。 <ul style="list-style-type: none"> • 0~65535
NumOfCycleOverErrDetectTimes	周期溢出出错检测次数	WORD (UINT)	设置检测出各周期的周期溢出出错为止的周期数。 <ul style="list-style-type: none"> • 0~65535
CycleOverErrorType	周期溢出出错选择	INT (MC_ERROR_CLASS) <small>☞ 76页</small> MC_ERROR_CLASS	设置各周期的溢出出错的分配。 <ul style="list-style-type: none"> • 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。

STORAGE_INFORMATION

变量名	名称	数据类型	内容
Capacity	容量	DWORD (UDINT)	存储驱动器的容量[k字节]。
FreeSpace	空余容量	DWORD (UDINT)	存储驱动器的空余容量[k字节]。
Mount	安装状态	BOOL	如果在驱动器的安装状态时可以访问文件, 则将变为TRUE。

INSTANCE_ID

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

变量名	名称 (软件偏置)	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(出错代码: 3410H)”。
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)) • -100.0000000~100.0000000
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。 (0.000000000000001 ≤ 1周期长(CycleLength) ≤ 10000000000.0) • 0.000000000000001~10000000000.0
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(出错代码: 3410H)”。
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。 (0.000000000000001 ≤ 行程量(Stroke) ≤ 10000000000.0) • 0.000000000000001~10000000000.0
NumberOfSections	区间数*1 (+104)	WORD (UINT)	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。

变量名	名称 (软元件偏置)	数据类型	内容
Options	选项 (+106)	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 • 1: 不能设置*3 ■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>

*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。

*3 设置“1”时, 将变为“超出MasterAbsolute范围(出错代码: 341DH)”。

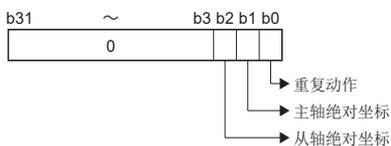
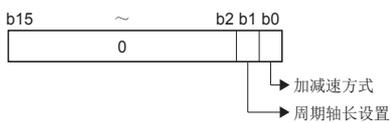
PROFILE_CAM_ELEMENT

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_TYP E) 74页 MC_CAM_CURVE_TYPE	<p>设置凸轮曲线。</p> <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(Trapeclloid) 8: 反梯弦型(ReverseTrapeclloid) 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	<p>设置对1周期长的位置(1周期当前值)。</p> <ul style="list-style-type: none"> 0.0 < 结束点(EndPoint) ≤ 1周期长(CycleLength)
Stroke	行程 (+8)	LREAL	<p>设置行程位置。</p> <ul style="list-style-type: none"> 行程(Stroke)的绝对值 ≤ 行程量(Stroke)
RangeP1	曲线适用范围(P1) (+12)	LREAL	<p>设置凸轮曲线的曲线适用范围(始点: P1, 终点: P2)。</p> <p>设置时应满足“P1 < P2”的条件。但是, “P1 = P2 = 0”时, 将适用“P1 = 0”、“P2 = 1”。</p> <ul style="list-style-type: none"> 0.0~1.0
RangeP2	曲线适用范围(P2) (+16)	LREAL	
RangeL1	加减速范围补偿(范围L1) (+20)	LREAL	<p>设置凸轮曲线中加减速的范围(L1、L2)。</p> <p>根据凸轮曲线可设置的范围有所不同。</p>
RangeL2	加减速范围补偿(范围L2) (+24)	LREAL	<p>“L1 = L2 = 0.0000”时, 适用各凸轮曲线的默认值。</p> <p>在不使用L1、L2的曲线中, 忽略设置值。</p> <ul style="list-style-type: none"> 0.0001 < 范围(L1、L2) < 1.0000
EndVelocity	终点速度*2 (+28)	LREAL	<p>设置凸轮曲线的终点速度。</p> <ul style="list-style-type: none"> -2500000000.0 < 终点速度(EndVelocity) < 2500000000.0
EndAcceleration	终点加速度*2 (+32)	LREAL	<p>设置凸轮曲线的终点加速度。</p> <ul style="list-style-type: none"> -2147483647.0 < 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

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 • 1: 不能设置*1 ■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 = 设置值 × π”进行计算。 ■自动创建选项为“1: 周长”的情况下 作为“周长 = 设置值”进行计算。 (0.0 < 周长 < 10000000000.0) • 0.00000000000001~10000000000.0

变量名	名称 (软元件偏置)	数据类型	内容
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) • -1000000000.0~1000000000.0
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)) • -1000000000.0~1000000000.0
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。

*1 设置“1”时，将变为“超出MasterAbsolute范围(出错代码：341DH)”。

PROGRAM_INFO

变量名	名称	数据类型	内容
ExecutionTime	执行时间	DWORD (UDINT)	以[μs]单位存储程序的执行时间。 在STOP→RUN时将被清零。
Number	执行次数	WORD (UINT)	对程序的执行次数进行计数。 超出65535时将返回到0。 在STOP→RUN时将被清零。 • 0~65535
Status	状态	WORD (HEX)	存储程序的执行状态。 • 0：待机中(MC_PROGRAM_WAITING) • 1：执行中(MC_PROGRAM_RUNNING) • 3：出错停止(MC_PROGRAM_ERROR) • 4：通过DI的待机中(MC_PROGRAM_DI)

SLAVE_OBJECT_REAL

变量名	名称	数据类型	内容
ControlWord	ControlWord	WSTRING (63)	对驱动设备进行状态切换请求。
EncoderIncrements	EncoderIncrements	WSTRING (63)	获取驱动设备的编码器分辨率。
FollowingErrActualValue	FollowingErrActualValue	WSTRING (63)	是驱动设备的滞留脉冲。
GearRatioMotorRevolutions	GearRatioMotorRevolutions	WSTRING (63)	对驱动设备设置伺服电机轴旋转数(分子)。
HomeOffset	HomeOffset	WSTRING (63)	对驱动设备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)	从驱动设备中获取伺服电机的最大速度。
MaxTorque	MaxTorque	WSTRING (63)	从驱动设备中获取伺服电机的最大转矩。
ModesOfOp	ModesOfOp	WSTRING (63)	向驱动设备请求控制模式的切换。
ModesOfOpDisp	ModesOfOpDisp	WSTRING (63)	获取驱动设备的控制模式。
NegativeTorqueLimitValue	NegativeTorqueLimitValue	WSTRING (63)	对驱动设备设置负方向转矩限制值。
Polarity	Polarity	WSTRING (63)	设置驱动设备的旋转方向选择。
PosActualValue	PosActualValue	WSTRING (63)	是驱动设备的当前位置。
PosEncoderResolution	PosEncoderResolution	WSTRING (63)	返回Sub_Index的总数(=2)。
PosEncoderResolutionMotorRevolutions	PosEncoderResolutionMotorRevolutions	WSTRING (63)	是驱动设备的旋转数。
PositiveTorqueLimitValue	PositiveTorqueLimitValue	WSTRING (63)	对驱动设备设置正方向转矩限制值。
ShaftRevolutions	ShaftRevolutions	WSTRING (63)	设置驱动设备的驱动轴旋转数(分母)。
SiUnitPos	SiUnitPos	WSTRING (63)	获取驱动设备的SI单位位置。
SiUnitVel	SiUnitVel	WSTRING (63)	获取驱动设备的SI单位速度。
StatusWord	StatusWord	WSTRING (63)	获取驱动设备的状态。
SupportedDriveModes	SupportedDriveModes	WSTRING (63)	获取驱动设备的对应控制模式。

变量名	名称	数据类型	内容
TargetPos	TargetPos	WSTRING (63)	是输出至驱动设备的指令位置。
TargetTorque	TargetTorque	WSTRING (63)	是输出至驱动设备的指令转矩。
TargetVelocity	TargetVelocity	WSTRING (63)	是输出至驱动设备的指令速度。
TargetActualValue	TorqueActualValue	WSTRING (63)	是驱动设备的当前转矩。
VelActualValue	VelActualValue	WSTRING (63)	是驱动设备的当前速度。
vControlDi1	ControlDi1	WSTRING (63)	对驱动设备进行输入软元件的设置。
vControlDi2	ControlDi2	WSTRING (63)	对驱动设备进行输入软元件的设置。
vControlDi3	ControlDi3	WSTRING (63)	对驱动设备进行输入软元件的设置。
vControlDi4	ControlDi4	WSTRING (63)	对驱动设备进行输入软元件的设置。
vControlDi5	ControlDi5	WSTRING (63)	对驱动设备进行输入软元件的设置。
vControlDi6	ControlDi6	WSTRING (63)	对驱动设备进行输入软元件的设置。
vControlDi7	ControlDi7	WSTRING (63)	对驱动设备进行输入软元件的设置。
vCurrentAlarm	CurrentAlarm	WSTRING (63)	从驱动设备中获取发生中的报警。
vEncoderStatus1	EncoderStatus1	WSTRING (63)	从驱动设备中获取编码器状态。
vHomeAbsCounter	HomeAbsCounter	WSTRING (63)	从驱动设备中获取作为原点保存的编码器多旋转计数器。
vHomeCycleCounter	HomeCycleCounter	WSTRING (63)	从驱动设备中获取电源接通时的编码器1旋转内位置。
vInitialAbsCounter	InitialAbsCounter	WSTRING (63)	从驱动设备中获取电源接通时的编码器多旋转计数器。
vInitialCycleCounter	InitialCycleCounter	WSTRING (63)	从驱动设备中获取电源接通时的编码器1旋转内位置。
vInitialPos	InitialPos	WSTRING (63)	从驱动设备中获取电源接通时的当前位置。
vMaxAbsCounter	MaxAbsCounter	WSTRING (63)	从驱动设备中获取编码器多旋转计数器的最大值。
vStatusDo1	StatusDo1	WSTRING (63)	获取驱动设备的输出软元件状态。
vStatusDo2	StatusDo2	WSTRING (63)	获取驱动设备的输出软元件状态。
vStatusDo3	StatusDo3	WSTRING (63)	获取驱动设备的输出软元件状态。
vStatusDo4	StatusDo4	WSTRING (63)	获取驱动设备的输出软元件状态。
vStatusDo5	StatusDo5	WSTRING (63)	获取驱动设备的输出软元件状态。
vStatusDo6	StatusDo6	WSTRING (63)	获取驱动设备的输出软元件状态。
vStatusDo7	StatusDo7	WSTRING (63)	获取驱动设备的输出软元件状态。
vSupportedControlDi1	SupportedControlDi1	WSTRING (63)	获取驱动设备支持的输入软元件。
vSupportedControlDi2	SupportedControlDi2	WSTRING (63)	获取驱动设备支持的输入软元件。
vSupportedControlDi3	SupportedControlDi3	WSTRING (63)	获取驱动设备支持的输入软元件。
vSupportedControlDi4	SupportedControlDi4	WSTRING (63)	获取驱动设备支持的输入软元件。
vSupportedControlDi5	SupportedControlDi5	WSTRING (63)	获取驱动设备支持的输入软元件。
vSupportedControlDi6	SupportedControlDi6	WSTRING (63)	获取驱动设备支持的输入软元件。
vSupportedControlDi7	SupportedControlDi7	WSTRING (63)	获取驱动设备支持的输入软元件。
vSupportedStatusDo1	SupportedStatusDo1	WSTRING (63)	获取驱动设备支持的输出软元件。
vSupportedStatusDo2	SupportedStatusDo2	WSTRING (63)	获取驱动设备支持的输出软元件。
vSupportedStatusDo3	SupportedStatusDo3	WSTRING (63)	获取驱动设备支持的输出软元件。
vSupportedStatusDo4	SupportedStatusDo4	WSTRING (63)	获取驱动设备支持的输出软元件。
vSupportedStatusDo5	SupportedStatusDo5	WSTRING (63)	获取驱动设备支持的输出软元件。

变量名	名称	数据类型	内容
vSupportedStatusDo6	SupportedStatusDo6	WSTRING (63)	获取驱动设备支持的输出软元件。
vSupportedStatusDo7	SupportedStatusDo7	WSTRING (63)	获取驱动设备支持的输出软元件。
vSyncAbsCounter	SyncAbsCounter	WSTRING (63)	从驱动设备中获取刻度测量编码器的多旋转计数器。
vSyncCycleCounter	SyncCycleCounter	WSTRING (63)	从驱动设备中获取刻度测量编码器的1旋转内位置。
vVelLimitValue	VelLimitValue	WSTRING (63)	对驱动设备设置速度限制值。
vWatchDogCounterDl	WatchDogCounterDl	WSTRING (63)	向驱动设备通知看门狗计数器值。
vWatchDogCounterUl	WatchDogCounterUl	WSTRING (63)	将看门狗计数器值获取到驱动设备中。
vEncoderStatus2	EncoderStatus2	WSTRING (63)	从驱动设备中获取刻度测量编码器状态。
vScaleCycleCounter	ScaleCycleCounter	WSTRING (63)	从驱动设备中获取刻度测量编码器的1旋转内位置。
vScaleAbsCounter	ScaleAbsCounter	WSTRING (63)	从驱动设备中获取刻度测量编码器的多旋转计数器。
vScaleMeasurementEncoderResolution	ScaleMeasurementEncoderResolution	WSTRING (63)	从驱动设备中获取刻度测量编码器的分辨率。
vScaleMeasurementEncoderReceptionStatus	ScaleMeasurementEncoderReceptionStatus	WSTRING (63)	从驱动设备中获取编码器1旋转内位置。

SLAVE_OBJECT_ENCODER

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。</p> <p>在类型中，可以设置[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>*: 省略了设置的情况下，将使用编码器输入值 (<u>AxisName</u>.Cd.Encoder_InputValue)。</p>

2.2 列举符一览

各种参数及监视数据、运动控制FB中使用的列举符的一览如下所示。

ENUM列举符

对于各种参数及监视数据、运动控制FB中使用的列举型的常数，实际上使用INT型的值。

在工程工具中，可以使用“列举型名称_列举符名称”的INT型全局标签。

但是，对于INT型全局标签，仅在运动模块侧创建的程序中可以使用。

对于在CPU模块侧创建的程序，应使用常数。

○：可以设置，×：不能设置

列举符	CPU模块侧			运动模块侧
	梯形图	FBD/LD	ST	ST
INT型全局标签	×	×	×	○*1
常数	○	○	○	○

*1 应使用运动控制设置功能的版本“1.010L”及以后。

例

在工程工具中使用MC_BUFFER_MODE型的列举符“mcBuffered”的情况下

- 以INT型全局标签使用时：设置“MC_BUFFER_MODE_mcBuffered”。
- 以常数使用时：设置“1”。

MC_BUFFER_MODE

列举符	设置值	内容
mcAborting	0	Aborting
mcBuffered	1	Buffered
mcBlendingLow	2	BlendingLow
mcBlendingPrevious	3	BlendingPrevious
mcBlendingNext	4	BlendingNext
mcBlendingHigh	5	BlendingHigh

MC_EXECUTION_MODE

列举符	设置值	内容
mcImmediately	0	立即执行
mcQueued	1	等待完成后执行
mcNextExecute	2	下次启动时执行
mcSpeculatively	3	推测执行

MC_COMBINE_MODE

列举符	设置值	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

列举符	设置值	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

列举符	设置值	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

列举符	设置值	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

列举符	设置值	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

列举符	设置值	内容
mcImmediate	0	即时
mcAbsolute	1	绝对
mcRelative	2	相对

MC_AXIS_STATUS

列举符	设置值	内容
Invalid	-1	轴变量未初始化/轴参数异常
Disabled	0	轴无效
ErrorStop	1	出错停止中
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

列举符	设置值	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	出错停止中
GroupStopping	2	减速停止中
GroupHoming	3	保留
GroupStandby	4	待机中
GroupMoving	5	动作中

MC_DRIVE_MODE

列举符	设置值	内容
NoModeChange	0	不更改
pp	1	保留
v1	2	保留
pv	3	保留
tq	4	保留
Reserved	5	保留
hm	6	hm(原点复位)
ip	7	保留
csp	8	csp(位置控制)
csv	9	csv(速度控制)
cst	10	cst(转矩控制)
cstca	11	保留
ct	-104	ct(挡块控制)

MC_VELOCITY_LIMIT_MODE

列举符	设置值	内容
Ignore	0	忽略
ClampWithRamp	1	限定
Truncate	2	舍去
ImmediateStop	3	立即停止
ClampWithoutRamp	4	限定(无减速时斜率)

MC_INTERPOLATE_SPEED_MODE

列举符	设置值	内容
VectorSpeed	0	合成速度
LongAxisSpeed	1	长轴速度
ReferenceAxisSpeed	2	基准轴速度

MC_SYNTHESIZING_MODE

列举符	设置值	内容
Addition	0	加法
Substitution	1	减法
Combine	2	合并
Interpolation	3	内插

MC_PROFILE_COMBINE_MODE

列举符	设置值	内容
Relative	0	相对
InputRelative	1	输入相对
OutputRelative	2	输出相对
Absolute	3	绝对

MC_RECORD_MODE

列举符	设置值	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_CLUTCH_MODE

列举符	设置值	内容
Exponent	0	指数
Linear	1	直线
LinearInputFollow	2	直线(输入跟踪)

MC_CAM_CURVE_TYPE

列举符	设置值	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

列举符	设置值	内容
DriveAxis	0	实驱动轴
EncoderAxis	2	实编码器轴
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

列举符	设置值	内容
IoModule	0	经由IO设备
Drive	1	经由驱动器模块

MC_ABS_SYSTEM

列举符	设置值	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

列举符	设置值	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

列举符	设置值	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

列举符	设置值	内容
RapidCurve	0	减速度变大的情况下重新创建减速曲线
OverrideCurve	1	重新创建减速曲线
ContinueCurve	2	减速曲线继续

MC_POS_SOURCE

列举符	设置值	内容
Invalid	-1	无效
SetPosition	1	指令当前位置
CumulativePosition	2	累计当前位置
FeedMachinePosition	3	进给机械位置
CommandedPosition	4	指定位置
ActualPosition	5	反馈位置

MC_ACC_ZERO_MODE

列举符	设置值	内容
ACCError	-1	出错(不启动)
MaximumAcceleration	1	最大加减速

MC_SIGNAL_LOGIC

列举符	设置值	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_STATION_REFRESH_MODE

列举符	设置值	内容
EmphasisResponse	0	响应优先方式
EmphasisOperationCycle	1	运算周期优先方式

MC_ERROR_CLASS

列举符	设置值	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_LOGGING_TYPE

列举符	设置值	内容
_None	0	无
ContinuousLogging	1	连续记录
TriggerLogging	2	触发记录

MC_LOGGING_SAVE_STATUS

列举符	设置值	内容
UnSave	0	未保存
Saving	1	保存中
Saved	2	保存完成
FullySaved	3	全部文件保存完成

MC_EXECUTION_STATE

列举符	设置值	内容
Ready	0	执行请求等待
Executing	1	执行中
Done	2	执行完成
Error_	-1	出错发生

MC_ACTIVATION_CMD

列举符	设置值	内容
_None	0	无
Authorize	1	激活密钥认证
SetSuperPassword	10	超级用户口令登录
SetKey	12	激活密钥登录
VerifyKey	13	激活密钥校验
Initialize	-1	初始化

MC_ACTIVATION_STATE

列举符	设置值	内容
NoRegistered	0	无登录
Authorized	1	认证成功
SuperPasswordValid	10	超级用户口令已登录
Unauthorized	12	激活密钥已登录(认证等待)
VerifyDone	13	校验完成
Lockout	-1	锁定中

MC_ADDON_LIMIT_CMD

列举符	设置值	内容
_None	0	无
Regist	1	登录
TemporaryNoLimitation	10	暂时解除
Initialize	-1	初始化

MC_ADDON_LIMIT_STATE

列举符	设置值	内容
NoLimitation	0	无限制
Limitation	1	限制中
TemporaryNoLimitation	10	解除中
Lockout	-1	锁定中

MC_SWITCHING_REQUEST

列举符	设置值	内容
NoRequest	0	无请求
Enable_	1	启用
Disable	2	禁用

MC_POS_RESTORATION_STATUS

列举符	设置值	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_TRANSITION_MODE

列举符	设置值	内容
TMNone	0	标准

2.3 运动控制FB一览

运动控制FB的一览如下所示。

管理系统FB

管理系统的运动控制FB一览如下所示。

运动控制FB	名称	内容	参阅
MC_GroupEnable	轴组有效	将指定的轴组的状态从“0: 轴组无效(GroupDisabled)”转变为“4: 待机中(GroupStandby)”。	☞ 80页 MC_GroupEnable
MC_GroupDisable	轴组无效	将指定的轴组的状态转变为“0: 轴组无效(GroupDisabled)”。	☞ 83页 MC_GroupDisable
MC_Power	允许运行	将指定的轴切换为允许运行状态。	☞ 86页 MC_Power
MC_SetPosition	当前位置更改	更改指定的轴的当前位置(指令位置、反馈位置)。	☞ 89页 MC_SetPosition
MCv_SetTorqueLimit	转矩限制值	执行转矩限制值的更改。	☞ 95页 MCv_SetTorqueLimit
MC_SetOverride	超驰值设置	执行指定的轴的目标速度、目标加速度、目标减速度的更改。	☞ 102页 MC_SetOverride
MC_ReadParameter	参数读取	进行从设备的对象的读取。	☞ 106页 MC_ReadParameter
MC_WriteParameter	参数写入	进行从设备的对象的写入。	☞ 112页 MC_WriteParameter
MC_Reset	轴出错复位	对轴的出错、警告进行复位。	☞ 118页 MC_Reset
MC_GroupReset	轴组出错复位	对轴组及轴组中所属的各轴的出错、警告进行复位。	☞ 121页 MC_GroupReset
MC_TouchProbe	触摸探头有效	通过触发事件发生记录任意数据。	☞ 125页 MC_TouchProbe
MC_AbortTrigger	触摸探头无效	将执行中的锁存置为无效。	☞ 137页 MC_AbortTrigger
MC_CamTableSelect	凸轮表选择	将指定的运算配置文件(凸轮数据)存储到展开区域中。	☞ 142页 MC_CamTableSelect
MCv_ChangeCycle	1周期当前值更改	更改指定的运算配置文件控制FB的1周期当前值。	☞ 149页 MCv_ChangeCycle
MCv_AllPower	所有轴允许运行	将所有轴切换为允许运行状态。	☞ 159页 MCv_AllPower
MC_GroupSetOverride	轴组超驰值设置	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。	☞ 163页 MC_GroupSetOverride
MCv_MotionErrorReset	运动出错复位	对运动系统的所有的出错、警告进行复位。	☞ 167页 MCv_MotionErrorReset

动作系统FB

动作系统的运动控制FB一览如下所示。

运动控制FB	名称	内容	参阅
MC_Home	原点复位	进行指定的轴的原点复位。	☞ 171页 MC_Home
MC_Stop	强制停止	对指定的轴进行减速停止。	☞ 177页 MC_Stop
MC_GroupStop	组强制停止	对指定的轴组进行减速停止。	☞ 181页 MC_GroupStop
MC_MoveAbsolute	绝对值定位	指定绝对位置的目标位置，执行定位。	☞ 186页 MC_MoveAbsolute
MC_MoveRelative	相对值定位	指定相对位置的移动量，执行定位。	☞ 197页 MC_MoveRelative
MCv_Jog	JOG运行	按照指令速度执行JOG运行。	☞ 205页 MCv_Jog
MC_MoveVelocity	速度控制	将驱动器切换为csv，按照指定的速度进行速度控制。	☞ 213页 MC_MoveVelocity
MC_TorqueControl	转矩控制	将驱动器切换为cst，按照指定的目标转矩进行转矩控制。	☞ 220页 MC_TorqueControl
MCv_SpeedControl	速度控制(包含位置循环)	执行包含位置循环的速度控制。	☞ 229页 MCv_SpeedControl
MCv_MoveLinearInterpolateAbsolute	绝对值直线插补控制	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。	☞ 237页 MCv_MoveLinearInterpolateAbsolute
MCv_MoveLinearInterpolateRelative	相对值直线插补控制	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。	☞ 251页 MCv_MoveLinearInterpolateRelative
MCv_MoveCircularInterpolateAbsolute	绝对值圆弧插补控制	使用设置的轴组的构成轴，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。	☞ 264页 MCv_MoveCircularInterpolateAbsolute
MCv_MoveCircularInterpolateRelative	相对值圆弧插补控制	使用设置的轴组的构成轴，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。	☞ 278页 MCv_MoveCircularInterpolateRelative
MC_CamIn	凸轮动作开始	按照指定的凸轮数据开始凸轮动作。	☞ 294页 MC_CamIn
MC_GearIn	齿轮动作开始	按照指定的齿轮比开始齿轮动作。	☞ 316页 MC_GearIn
MC_CombineAxes	加减法定位	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。	☞ 325页 MC_CombineAxes
MCv_BacklashCompensationFilter	背隙补偿滤波器	进行按照移动方向，对机械系统的背隙量进行补偿的滤波器处理。	☞ 336页 MCv_BacklashCompensationFilter
MCv_SmoothingFilter	平滑滤波器	按照指定的频率，进行滤波器处理。	☞ 349页 MCv_SmoothingFilter
MCv_DirectionFilter	移动方向限制滤波器	进行对设置的移动方向进行移动限制的滤波器处理。	☞ 358页 MCv_DirectionFilter
MCv_SpeedLimitFilter	速度限制滤波器	进行将速度限制为设置的限制值的滤波器处理。	☞ 366页 MCv_SpeedLimitFilter

常规FB

用于进行轴控制的运动控制FB一览如下所示。

运动控制FB	名称	内容	参阅
MCv_ReadProfileData	配置文件读取	从展开区域或文件中读取指定的运算配置文件数据。	☞ 374页 MCv_ReadProfileData
MCv_WriteProfileData	配置文件写入	将指定的运算配置文件数据写入到展开区域或文件中。	☞ 382页 MCv_WriteProfileData

3 运动控制FB

3.1 管理系统的FB

MC_GroupEnable

将指定的轴组的状态 (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>

名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
轴组有效	9	5	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 44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

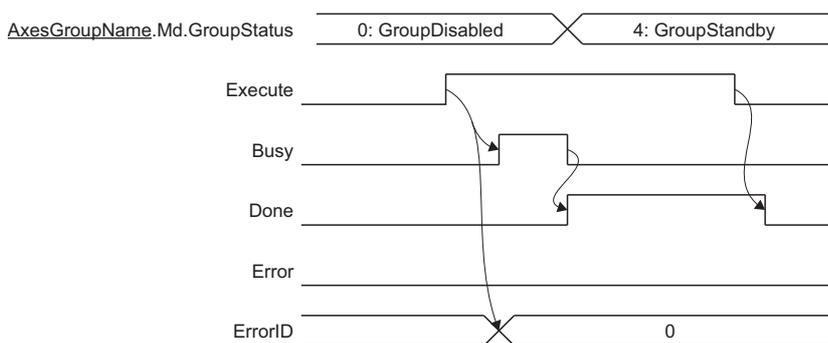
输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	转变为“4: 待机中(GroupStandby)”状态时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_GroupEnable (轴组有效)时，将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册 (应用篇)

功能

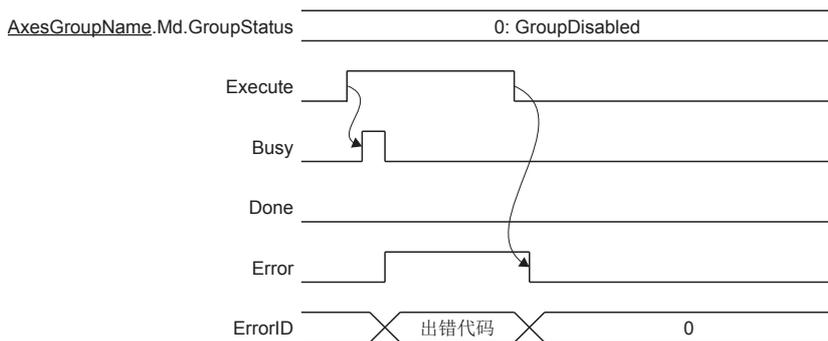
- 将AXES_GROUP_REF结构体的轴组状态 (AxesGroupName. Md. GroupStatus)从“0: 轴组无效(GroupDisabled)”转变为“4: 待机中(GroupStandby)”。
- 执行指令 (Execute)变为TRUE时将执行MC_GroupEnable(轴组有效), 正常开始处理时执行中(Busy)将变为TRUE。
- 处理正常完成时, 执行完成 (Done)将变为TRUE, 执行中 (Busy)将变为FALSE。
- 在MC_GroupEnable(轴组有效)内发生了异常的情况下, 出错 (Error)将变为TRUE, 并将出错代码存储到出错代码 (ErrorID)中。关于出错代码的详细内容, 请参阅下述手册。
 ❷ MELSEC iQ-R运动模块用户手册(应用篇)
- 关于轴组的状态转变, 请参阅下述手册。
 ❷ MELSEC iQ-R运动模块用户手册(应用篇)
- 对于MC_GroupEnable(轴组有效), 只有在所有构成轴的轴状态 (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)变为了“4: 待机中(GroupStandby)”时, 构成轴的轴组使用中 (AxisName. Md. UseInGroup)将变为TRUE。
- 在设置包括轴组使用中 (AxisName. Md. UseInGroup)为TRUE的构成轴的其他轴组后执行了MC_GroupEnable(轴组有效)的情况下, 将变为“轴组构成轴使用中(出错代码: 3496H)”。
- 构成轴之一发生了出错的情况下, 轴组状态 (AxesGroupName. Md. GroupStatus)将变为“1: 出错停止中(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



- 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)。
- 只有在轴组中设置的所有构成轴的轴状态(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)(出错代码: 1D81H)”。此时, 轴组状态(AxesGroupName.Md.GroupStatus)将变为“-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(出错代码: 3402H)”。

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。

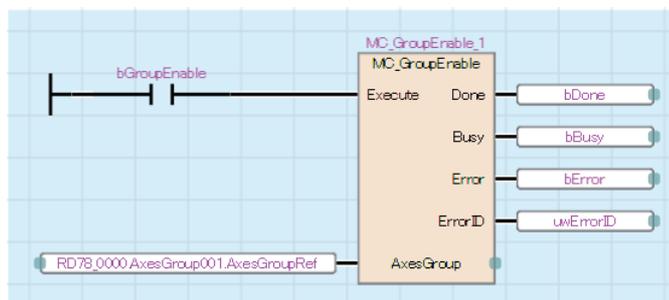
■轴组

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	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)



■ST的程序(运动模块侧)

```
MC_GroupEnable_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bGroupEnable ,
    Done=> bDone ,
    Busy=> bBusy ,
    Error=> bError ,
    ErrorID=> uwErrorID
);
```

MC_GroupDisable

将指定的轴组的状态(AxesGroupName.Md.GroupStatus)转变为“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>

名称	输入区域点数(byte)	输出区域点数(byte)	编译方式	FB动作
轴组无效	9	5	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 44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_GroupDisable(轴组无效)。

■输出变量

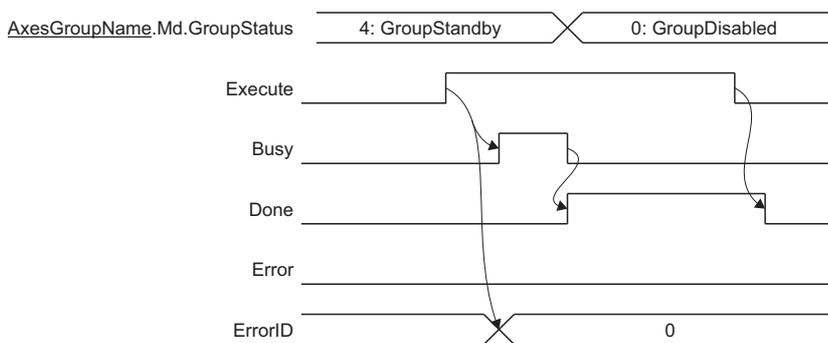
输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	转变为“0: 轴组无效(GroupDisabled)”状态时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_GroupDisable(轴组无效)时，将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD(UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册(应用篇)

功能

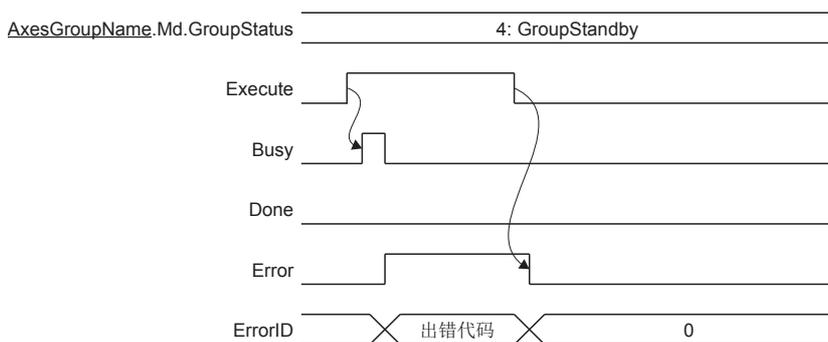
- 将AXES_GROUP_REF结构体的轴组状态 (AxesGroupName. Md. GroupStatus) 转变为“0: 轴组无效(GroupDisabled)”。
- 在执行指令(Execute)为TRUE时执行MC_GroupDisable(轴组无效)，正常开始处理时执行中(Busy)将变为TRUE。
- 处理正常完成时，执行完成(Done)将变为TRUE，执行中(Busy)将变为FALSE。
- 在MC_GroupDisable(轴组无效)内发生了异常的情况下，出错(Error)将变为TRUE，并将出错代码存储到出错代码(ErrorID)中。关于出错代码的详细内容，请参阅下述手册。
 [MELSEC iQ-R运动模块用户手册(应用篇)]
- 关于轴组的状态转变，请参阅下述手册。
 [MELSEC iQ-R运动模块用户手册(应用篇)]
- 对于MC_GroupDisable(轴组无效)，只有在轴组状态 (AxesGroupName. Md. GroupStatus) 为“4: 待机中(GroupStandby)”或“1: 出错停止中(GroupErrorStop)”的情况下才能执行。轴组状态为“1: 出错停止中(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: 出错停止中(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)(出错代码: 1A97H)”，且执行中的FB将停止。
- 轴组状态 (AxesGroupName. Md. GroupStatus) 变为了“0: 轴组无效(GroupDisabled)”时，构成轴的轴组使用中 (AxisName. Md. UseInGroup) 将变为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



- 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)(出错代码: 1D81H)”。此时, 轴组状态(AxesGroupName.Md.GroupStatus)将变为“-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(出错代码: 3402H)”。

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。

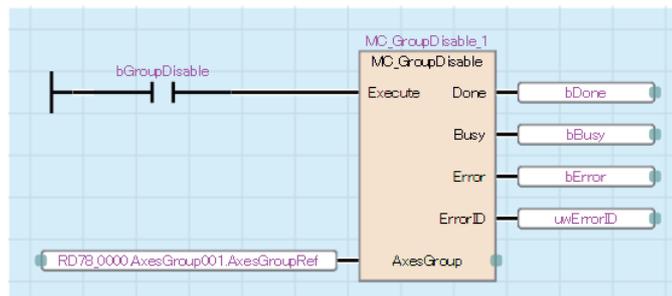
■轴组

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	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

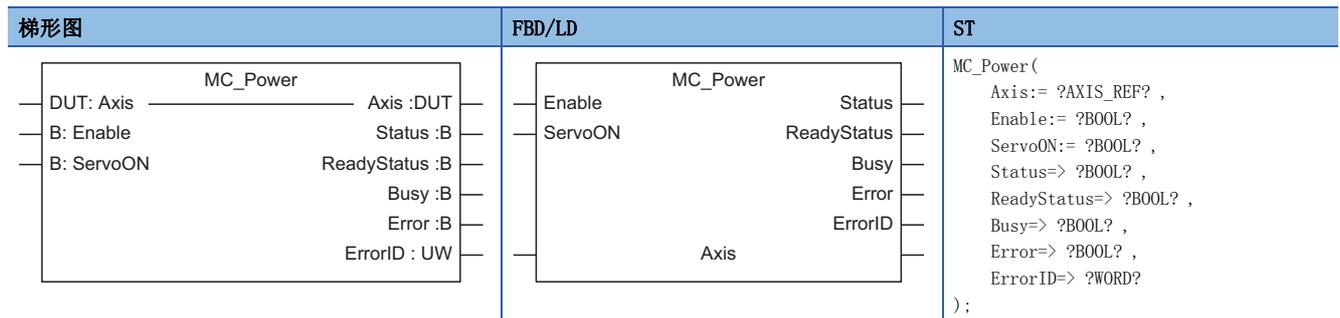


■ST的程序(运动模块侧)

```
MC_GroupDisable_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bGroupDisable ,
    Done=> bDone ,
    Busy=> bBusy ,
    Error=> bError ,
    ErrorID=> uwErrorID
);
```

MC_Power

将指定的轴切换为允许运行状态。



名称	输入区域点数(byte)	输出区域点数(byte)	编译方式	FB动作
允许运行	10	6	子程序型	随时执行型

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.)，请参阅下述章节。 23页 AxisName.AxisRef. (轴信息) *: 在将“可编程控制器就绪(Y0)”切换为ON后，初次调用了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	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD(UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 MELSEC iQ-R运动模块用户手册(应用篇)

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，出错 (Error) 将变为TRUE，并将出错代码存储到出错代码 (ErrorID) 中。关于出错代码的详细内容，请参阅下述手册。

📖 MELSEC iQ-R运动模块用户手册 (应用篇)

- AXIS_MONI结构体的轴状态 (AxisName.Md.AxisStatus) 将从“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。

📖 MELSEC iQ-R运动模块用户手册 (应用篇)

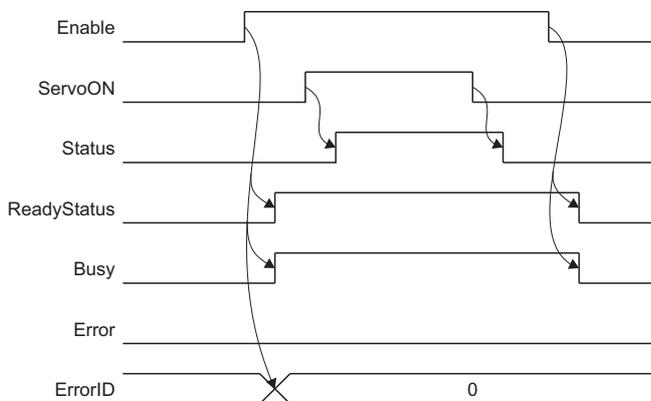
- 根据有效 (Enable) 及伺服ON请求 (ServoON) 的输入，将轴的伺服ON/OFF状态、驱动器状态按以下方式进行切换。

输入变量		输出变量		伺服ON/OFF状态	驱动器状态 (<u>AxisName.Md.Driver_State</u>)
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
	FALSE	TRUE	FALSE	伺服OFF	5: Switched On
FALSE	TRUE	FALSE	FALSE	伺服OFF	3: Switch On Disabled
	FALSE	FALSE	FALSE	伺服OFF	3: Switch On Disabled

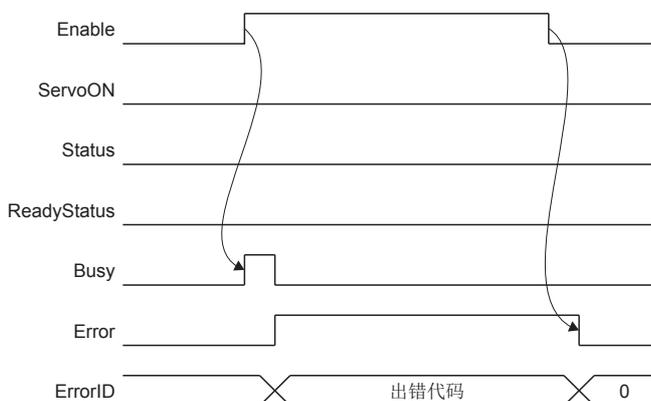
- 伺服OFF中实轴受外力而旋转的情况下，将进行跟踪处理。
- 伺服ON/OFF的控制操作与控制模式无关。伺服OFF时的控制模式取决于驱动器的规格。
- 驱动器模块出错发生中，MC_Power (允许运行) 将发送到驱动器中，因此无需再次将有效 (Enable) 及伺服ON请求 (ServoON) 从FALSE置为TRUE。

■ 时序图

- 正常完成时的情况下



- 异常完成时的情况下



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)。
- 对于MC_Power(允许运行), 在将“可编程控制器就绪(Y0)”切换为ON后, 初次调用了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。
- 对虚拟轴执行了MC_Power(允许运行)的情况下, 与伺服ON请求(ServoON)的输入无关, 允许运行(Status)、就绪ON状态(ReadyStatus)均将变为TRUE。
- “准备就绪(X0)”为OFF中时, 将维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。

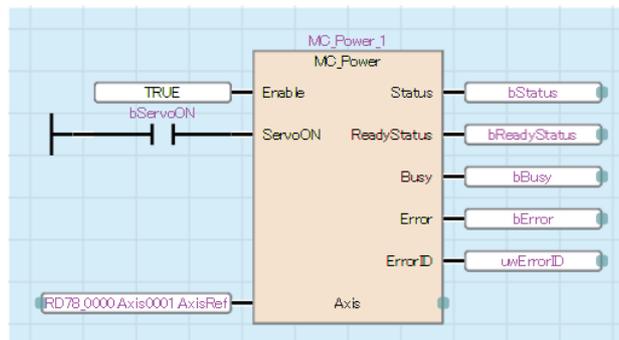
■轴

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	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)



■ST的程序(运动模块侧)

```

MC_Power_1(
    Axis:= Axis0001.AxisRef ,
    Enable:= TRUE ,
    ServoON:= bServoON ,
    Status=> bStatus ,
    ReadyStatus=> bReadyStatus ,
    Busy=> bBusy ,
    Error=> bError ,
    ErrorID=> uwErrorID
);

```

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>

名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
当前位置更改	32	7	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23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，设置值有所不同。 关于详细内容，请参阅下述章节。 ☞ 91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。 ☞ 91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。 ☞ 92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	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 MELSEC iQ-R运动模块用户手册(应用篇)
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。

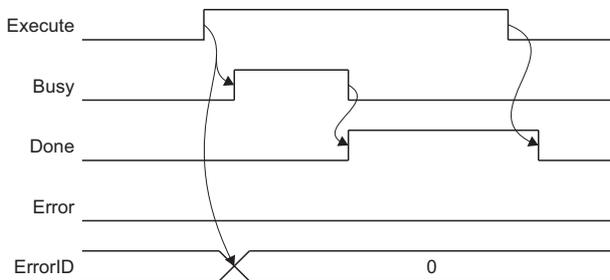
功能

- 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，在处理完成之前轴状态(AxisName.Md.AxisStatus)变为了“0: 轴无效(Disabled)”或“1: 出错停止中(ErrorStop)”的情况下，将变为“MC_SetPosition指令异常(出错代码: 344EH)”。
- 在MC_SetPosition(当前位置更改)内发生了异常的情况下，出错(Error)将变为TRUE，并将出错代码存储到出错代码(ErrorID)中。关于出错代码的详细内容，请参阅下述手册。

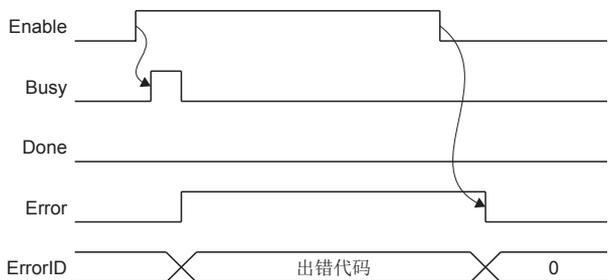
MELSEC iQ-R运动模块用户手册(应用篇)

■时序图

- 正常完成时的情况下



- 异常完成时的情况下



■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。

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)	<p>■环形计数器无效时</p> $-10000000000.0 \leq \text{设置值} < 10000000000.0$ <p>■环形计数器有效时</p> 环形计数器下限值 \leq 设置值 $<$ 环形计数器上限值
TRUE(通过相对距离进行当前位置更改)	<p>■环形计数器无效时</p> $-10000000000.0 \leq \text{设置值} \leq 10000000000.0$ <p>■环形计数器有效时</p> $-(\text{环形计数器上限值} - \text{环形计数器下限值}) / 2 \leq \text{设置值} \leq (\text{环形计数器上限值} - \text{环形计数器下限值}) / 2$

■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。

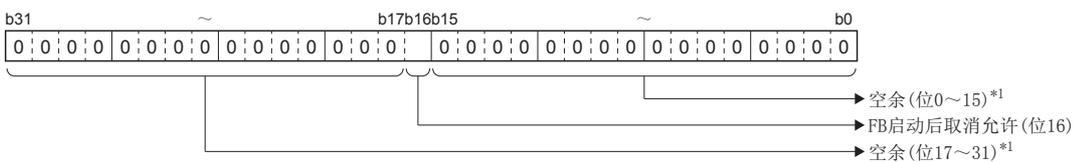
设置值	内容
1: 等待完成后执行 (mcQueued)	<p>执行中的FB结束后执行。 通过执行指令(Execute)的上升沿检测,执行中(Busy)变为TRUE,在轴状态(AxisName.Md.AxisStatus)变为“4: 待机中(Standstill)”之前待机。 轴状态(AxisName.Md.AxisStatus)变为“4: 待机中(Standstill)”时,开始当前位置更改动作。</p> <p>The diagram illustrates the timing of two function blocks: MC_MoveAbsolute and MC_SetPosition. For MC_MoveAbsolute, the Position signal is set to 100.0. The Done signal transitions from low to high, and the Busy signal transitions from high to low. The AxisName.Md.AxisStatus signal transitions from 5: DiscreteMotion to 4: Standstill. For MC_SetPosition, the Position signal is set to 190.0, and the ExecutionMode signal is set to 1: mcQueued. The Relative signal is low. The Execute signal transitions from low to high, then back to low. The Done signal transitions from low to high, and the Busy signal transitions from high to low. The Velocity signal shows a ramp up and then a ramp down. The 指令当前位置 signal shows a ramp from the current position to the FB1 target address, then a step to the current position after the address change.</p>

设置值	内容
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。</p> <p>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(出错代码: 344EH)”, 且不执行当前位置更改。</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(出错代码: 1A4EH)”。

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

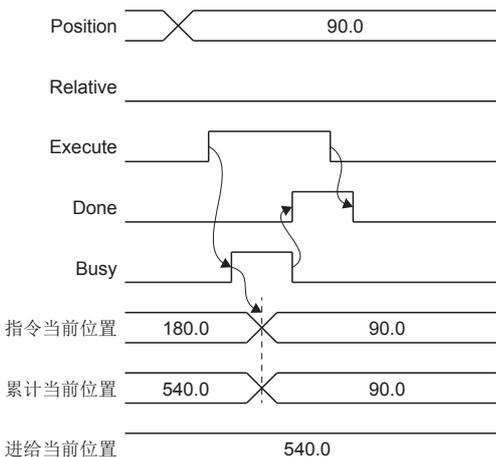
动作概要

- 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”。

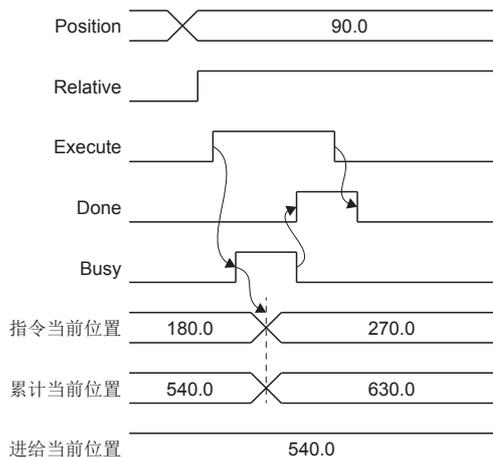
例

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)的情况下>



<相对位置选择(Relative)为TRUE(相对距离)的情况下>



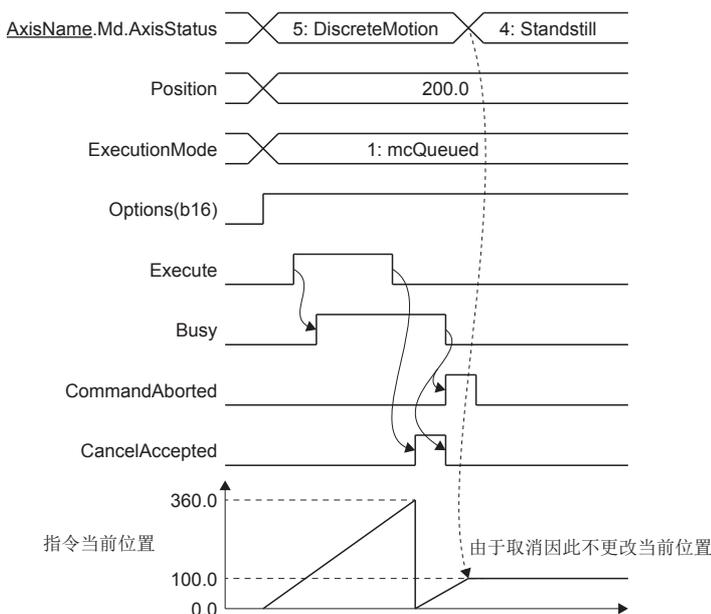
取消

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)切换为“4: 待机中(Standstill)”的切换待机中执行了取消的情况下



注意事项

- 轴动作中也可执行当前位置更改。但是，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

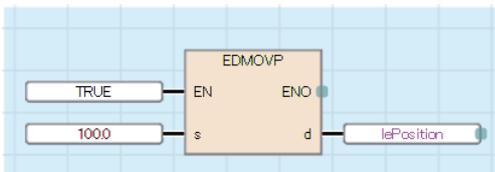
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

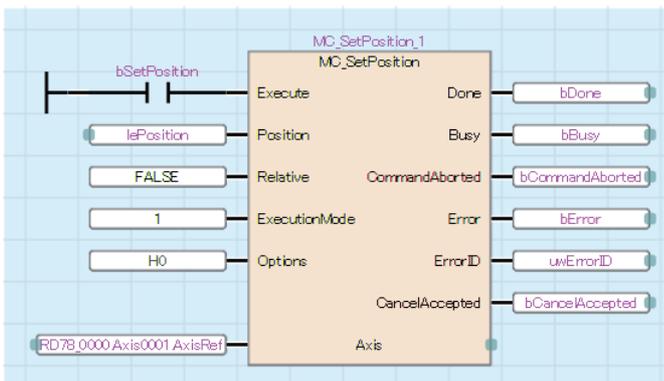
标签名	数据类型	注释
MC_SetPosition_1	MC_SetPosition	当前位置更改FB
bSetPosition	位	当前位置更改指令
lePosition	双精度实数	目标位置
bDone	位	准备就绪
bBusy	位	执行中
bCommandAborted	位	执行中断
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码
bCancelAccepted	位	取消受理

■FBD/LD的程序(CPU模块侧)

- 当前位置更改用数据设置



- 当前位置更改



■ST的程序(运动模块侧)

//-----当前位置更改用数据设置-----

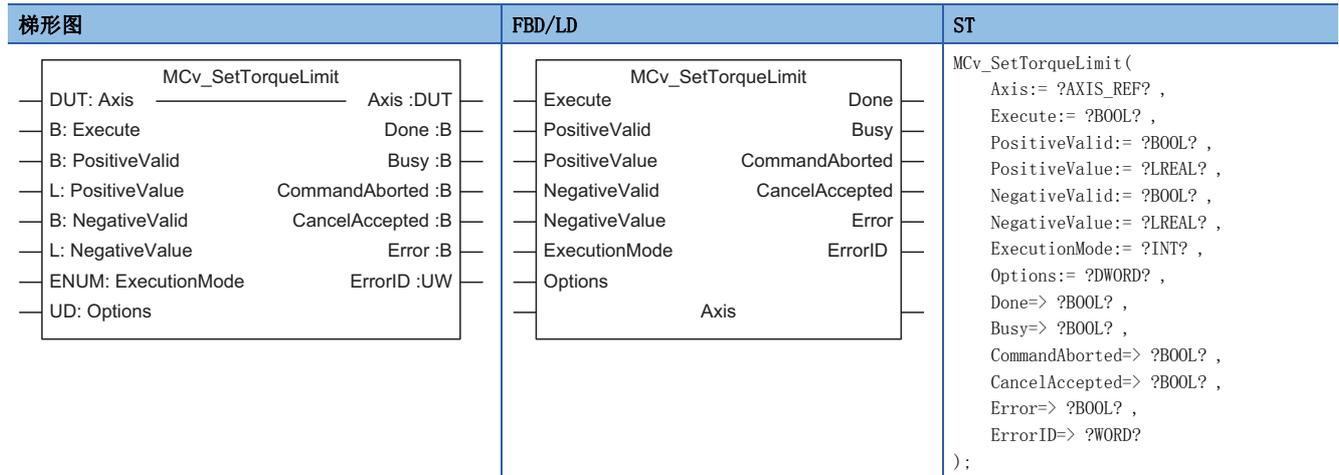
```
lePosition:= 100.0;
```

//-----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
);
```

MCv_SetTorqueLimit

执行转矩限制值的更改。



名称	输入区域点数(byte)	输出区域点数(byte)	编译方式	FB动作
转矩限制值	48	8	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23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置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置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。 ☞ 97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。 ☞ 98页 选项(Options)

■输出变量

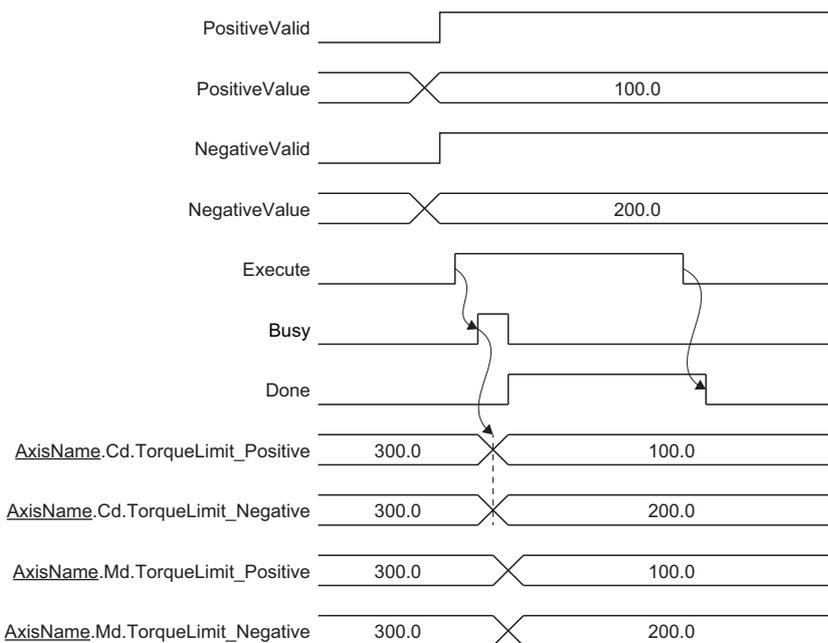
输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	转矩限制值更改正常完成时, 将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_SetTorqueLimit(转矩限制值)时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MCv_SetTorqueLimit(转矩限制值)的执行中断时, 将变为TRUE。
CancelAccepted	取消受理	BOOL	FALSE	MCv_SetTorqueLimit(转矩限制值)受理了取消时, 将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	出错代码	WORD(UINT)	0	发生了异常时, 将返回出错代码。 关于出错代码的详细内容, 请参阅下述手册。 MELSEC iQ-R运动模块用户手册(应用篇)

功能

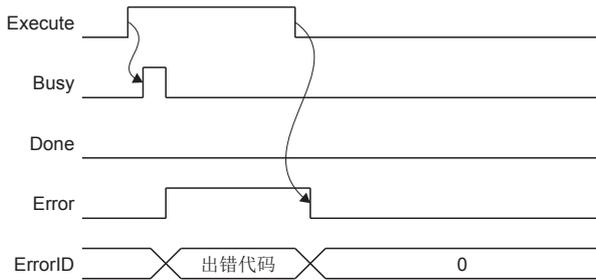
- 更改正方向转矩限制值或负方向转矩限制值。
- 正方向转矩限制有效(PositiveValid)为TRUE的情况下, 将正方向转矩限制值更改为正方向转矩限制值(PositiveValue)的值。
- 负方向转矩限制有效(NegativeValid)为TRUE的情况下, 将负方向转矩限制值更改为负方向转矩限制值(NegativeValue)的值。
- 在执行指令(Execute)为TRUE时执行MCv_SetTorqueLimit(转矩限制值), 正常开始处理时执行中(Busy)将变为TRUE。
- 对于更改转矩限制值的时机, 通过启动模式(ExecutionMode)进行设置。
- 处理完成, 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)”并启动MCv_SetTorqueLimit(转矩限制值)后, 在处理完成之前轴状态(AxisName.Md.AxisStatus)变为了“0: 轴无效(Disabled)”或“1: 出错停止中(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, 出错(Error)将变为TRUE, 并将出错代码存储到出错代码(ErrorID)中。关于出错代码的详细内容, 请参阅下述手册。
 MELSEC iQ-R运动模块用户手册(应用篇)
- 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



■启动模式 (ExecutionMode)

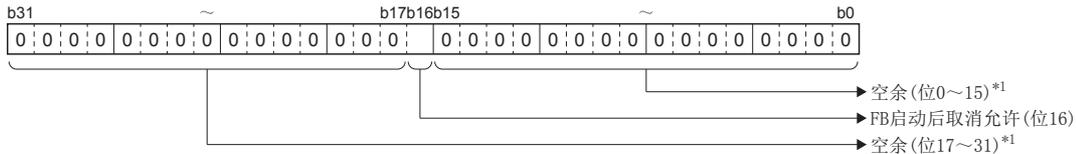
轴动作中执行了MCv_SetTorqueLimit (转矩限制值)的情况下,更改的时机根据设置的启动模式 (ExecutionMode)而有所不同。启动模式 (ExecutionMode)的时机如下所示。

设置值	内容
0: 立即执行 (mcImmediately)	与轴状态 (<u>AxisName</u> .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 (<u>AxisName</u> .Cd.TorqueLimit_Positive)、负方向转矩限制值 (<u>AxisName</u> .Cd.TorqueLimit_Negative)。
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,在轴状态 (<u>AxisName</u>.Md.AxisStatus)变为“4: 待机中 (Standstill)”之前待机。 一旦轴状态 (<u>AxisName</u>.Md.AxisStatus)变为“4: 待机中 (Standstill)”,便更改正方向转矩限制值 (<u>AxisName</u>.Cd.TorqueLimit_Positive)、负方向转矩限制值 (<u>AxisName</u>.Cd.TorqueLimit_Negative)。 执行指令 (Execute)上升沿检测时,如果轴状态 (<u>AxisName</u>.Md.AxisStatus)为“4: 待机中 (Standstill)”,则立即执行。 处理完成前轴状态 (<u>AxisName</u>.Md.AxisStatus)变为了“0: 轴无效 (Disabled)”或“1: 出错停止中 (ErrorStop)”的情况下,将中断处理且执行中断 (CommandAborted)将变为TRUE。</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 (<u>AxisName</u>.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 (<u>AxisName</u>.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 (出错代码: 3450H)”,且不执行转矩限制值更改。</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。

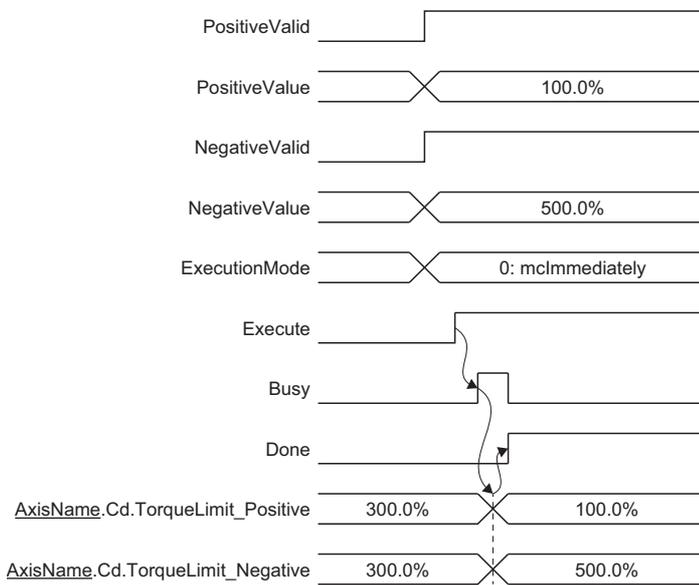


*1 空余中应设置“0”。设置了“0”以外的情况下，将变为“超出Options范围(出错代码: 1A4EH)”。

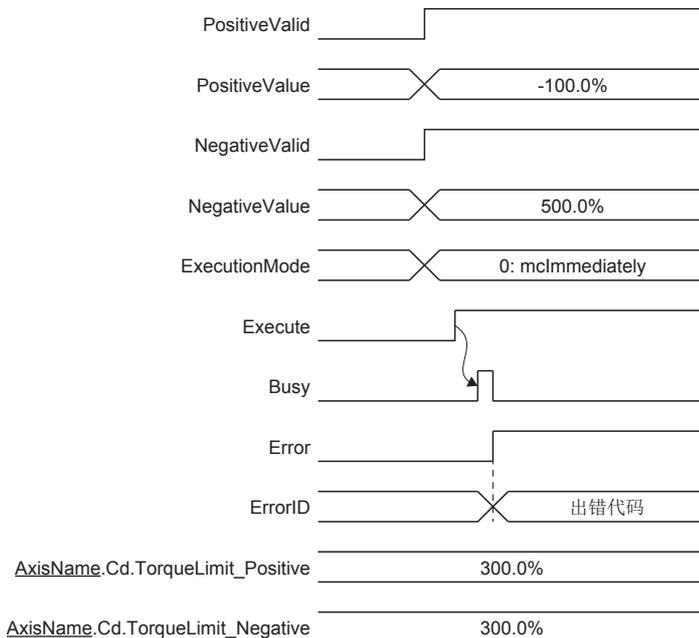
位	名称	内容
16	FB启动后取消允许	设置MCv_SetTorqueLimit(转矩限制值)启动后是否允许取消。 • 0: 不允许 • 1: 允许

■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, 并将执行指令(Execute)置为了TRUE的情况下, MCv_SetTorqueLimit(转矩限制值)将出错, 且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)。



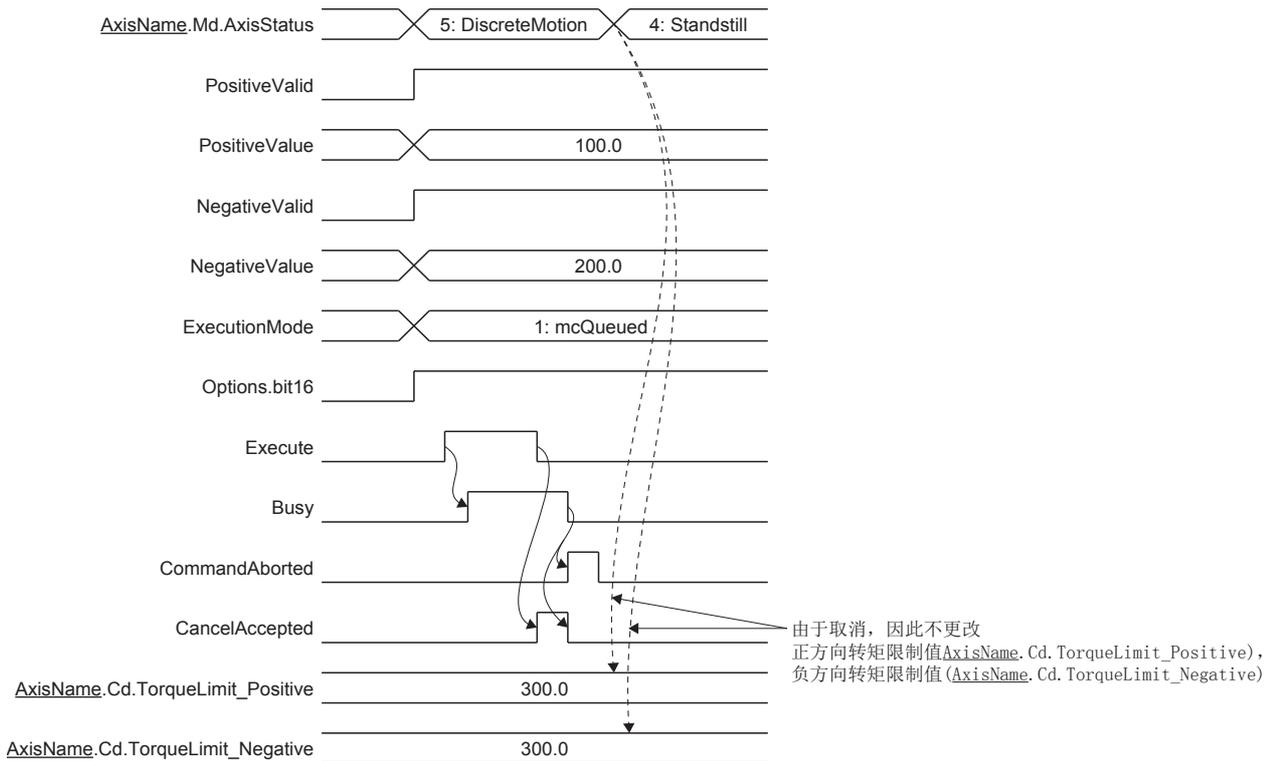
取消

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。
- 执行了取消的情况下，不更正方向转矩限制值(AxisName.Cd.TorqueLimit_Positive)、负方向转矩限制值(AxisName.Cd.TorqueLimit_Negative)。

例

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)切换为“4: 待机中(Standstill)”的切换待机中执行了取消的情况下



注意事项

- 在MCv_SetTorqueLimit (转矩限制值) 中，进行正方向转矩限制值(AxisName.Cd.TorqueLimit_Positive)与负方向转矩限制值(AxisName.Cd.TorqueLimit_Negative)的更改。

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

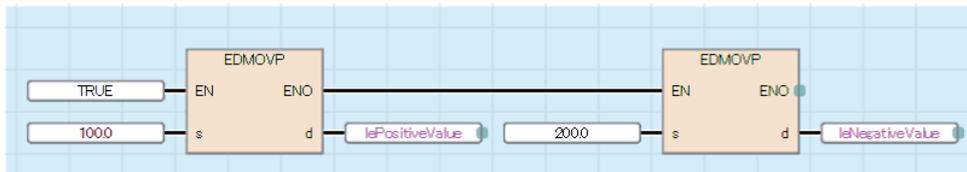
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

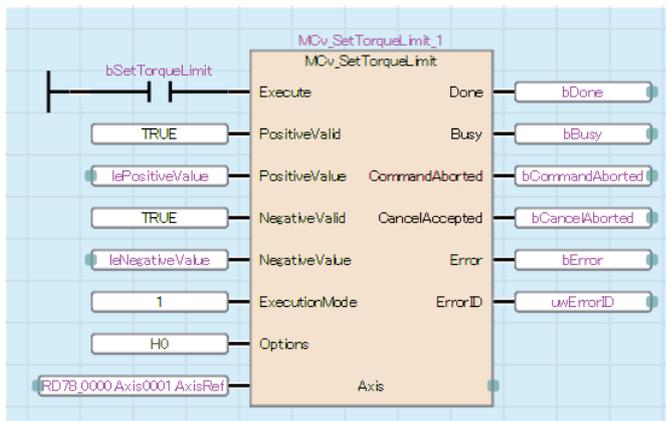
标签名	数据类型	注释
MCv_SetTorqueLimit_1	MCv_SetTorqueLimit	转矩限制值FB
bSetTorqueLimit	位	转矩限制值更改指令
lePositiveValue	双精度实数	正方向转矩限制值
leNegativeValue	双精度实数	负方向转矩限制值
bDone	位	准备就绪
bBusy	位	执行中
bCommandAborted	位	执行中断
bCancelAccepted	位	取消受理
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

- 转矩限制值更改用数据设置



- 转矩限制值更改

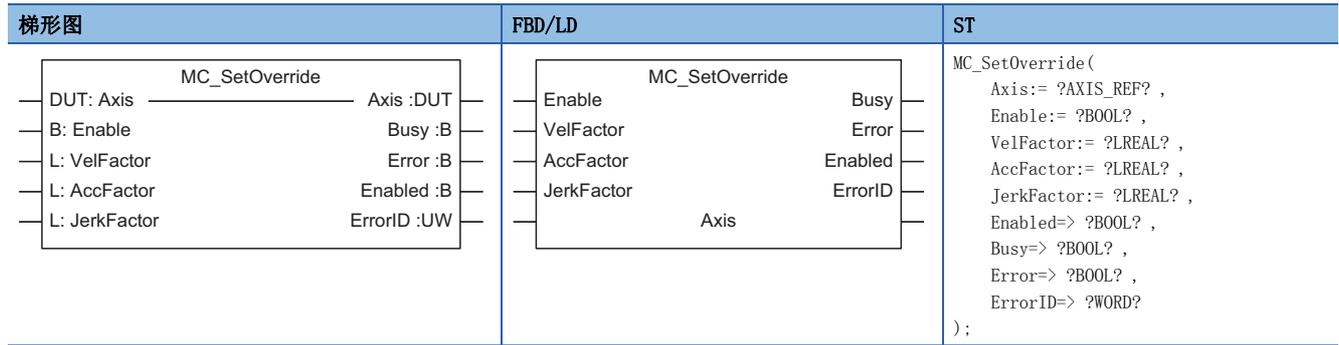


■ST的程序(运动模块侧)

```
//-----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  
);
```

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。



名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
超驰值设置	40	6	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23页 AxisName, AxisRef. (轴信息)

■输入变量

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
Enable	有效	BOOL	始终	TRUE、FALSE	FALSE	TRUE时执行MC_SetOverride(超驰值设置)。
VelFactor	速度超驰系数	LREAL	始终	0.00~10.00	0.00	设置速度的超驰系数。 有效(Enable)TRUE时，将始终获取。
AccFactor	加速度超驰系数	LREAL	始终	0.00、0.01~10.00	0.00	设置加速度的超驰系数。 有效(Enable)为TRUE时，将始终获取。 设置了“0.00”的情况下，不进行加速度超驰系数的更改，维持上次值进行控制。
JerkFactor	Jerk超驰系数	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超驰系数(JerkFactor)范围(出错代码: 349EH)”。

■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	超驰值正常设置的情况下，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_SetOverride(超驰值设置)时，将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD(UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册(应用篇)

功能

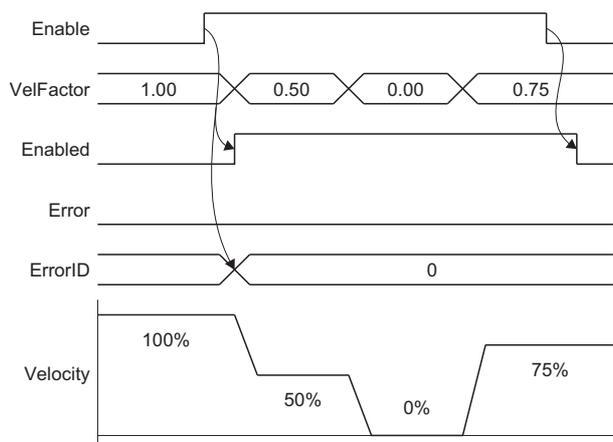
- 更改指定的轴的目标速度、目标加速度、目标减速度。
- 更改为当前动作中的目标速度、目标加速度、目标减速度乘以超驰系数后的值。
- 有效(Enable)变为TRUE时将执行MC_SetOverride(超驰值设置)。超驰系数为有效中时，有效中(Enabled)将变为TRUE。
- 有效(Enable)为TRUE时，如果更改超驰系数的值，则将反映新的超驰系数。
- 在MC_SetOverride(超驰值设置)内发生了异常的情况下，出错(Error)将变为TRUE，并将出错代码存储到出错代码(ErrorID)中。关于出错代码的详细内容，请参阅下述手册。

📖 MELSEC iQ-R运动模块用户手册(应用篇)

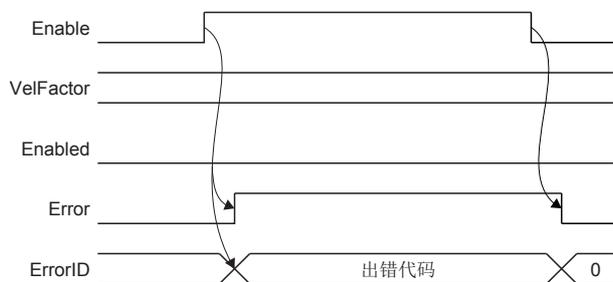
- 在速度超驰系数(VelFactor)中设置“0.00”的值时，在轴状态(AxisName.Md.AxisStatus)不切换为“4: 待机中(Standstill)”的状况下停止轴。
- 在加速度超驰系数(AccFactor)中设置“0.00”的值时，不更改加速度超驰系数，维持上次的加速度超驰系数。

■时序图

- 正常完成的情况下

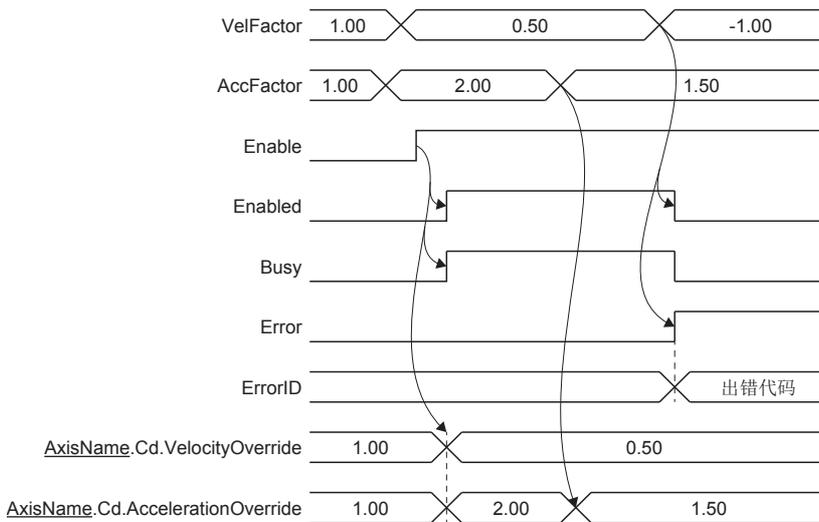


- 异常完成的情况下



■动作概要

- 在MC_SetOverride(超驰值设置)中, 更改速度超驰系数(AxisName.Cd.VelocityOverride)、加速度超驰系数(AxisName.Cd.AccelerationOverride)。



- 在超驰系数中设置了超出范围的值的情况下, MC_SetOverride(超驰值设置)将发生出错并停止以后的获取。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(超驰值设置)。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超驰系数有影响。
- 请勿在MC_SetOverride(超驰值设置)执行中进行速度超驰系数(AxisName.Cd.VelocityOverride)、加速度超驰系数(AxisName.Cd.AccelerationOverride)的直接操作。
- 速度超驰后的速度超出范围的情况下, 请参阅下述手册的“速度范围”。
[MELSEC iQ-R运动模块用户手册\(应用篇\)](#)
- 加速度超驰后的加速度、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”。
[MELSEC iQ-R运动模块用户手册\(应用篇\)](#)

■程序示例

将超驰值更改指令(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超驰系数“1.0”、加减速超驰系数“2.0”后的值的程序示例如下所示。

■轴

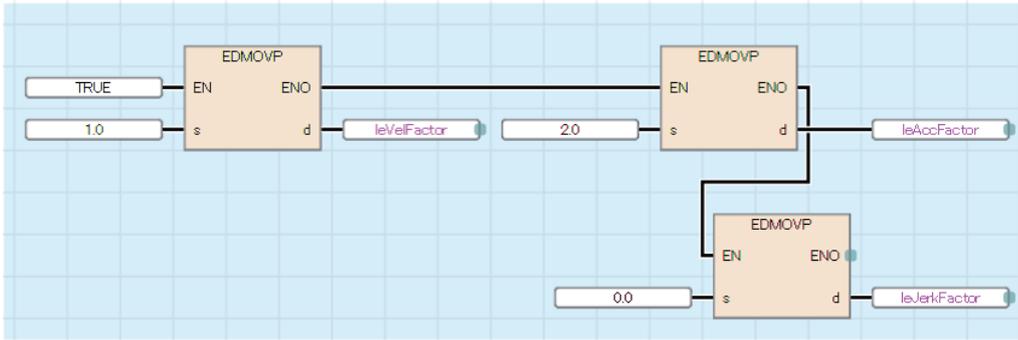
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

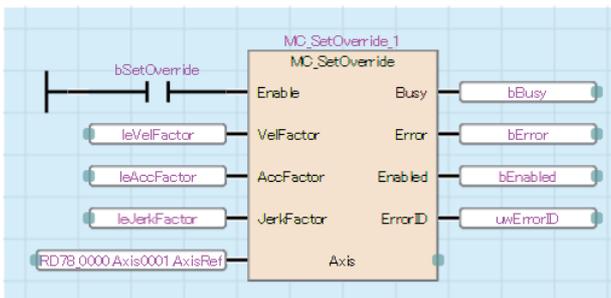
标签名	数据类型	注释
MC_SetOverride_1	MC_SetOverride	超驰值设置FB
bSetOverride	位	超驰值更改指令
leVelFactor	双精度实数	速度超驰系数
leAccFactor	双精度实数	加减速超驰系数
leJerkFactor	双精度实数	Jerk超驰系数
bEnabled	位	有效中
bBusy	位	执行中
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

- 超驰值更改用数据设置



- 超驰值更改



■ST的程序(运动模块侧)

//-----超驰值更改用数据设置-----

leVelFactor:= 1.00;

leAccFactor:= 2.00;

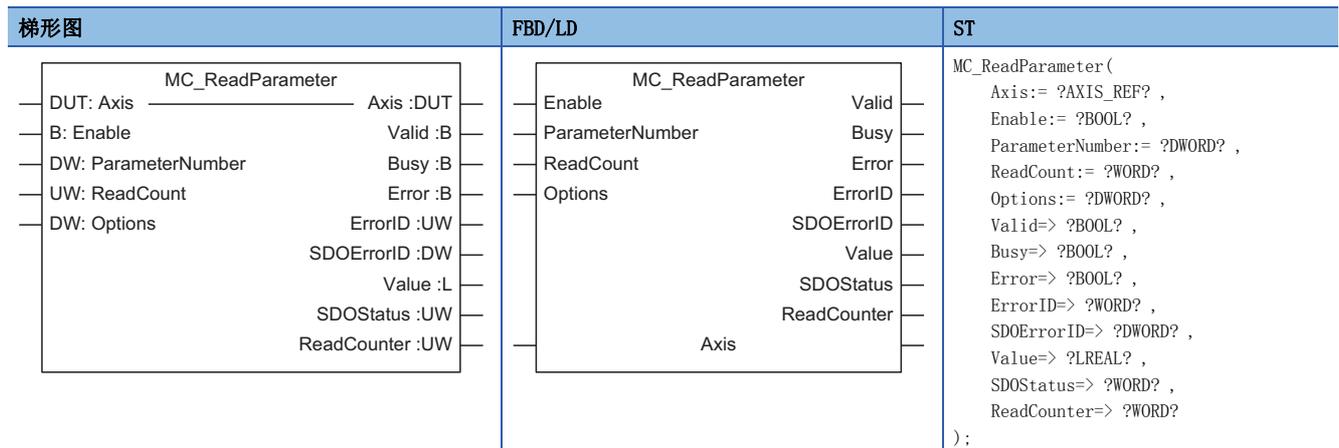
leJerkFactor:= 0.0;

//-----超驰值更改-----

```
MC_SetOverride_1(
    Axis:= Axis0001.AxisRef ,
    Enable:= bSetOverride ,
    VelFactor:= leVelFactor ,
    AccFactor:= leAccFactor ,
    JerkFactor:= leJerkFactor ,
    Enabled=> bEnabled ,
    Busy=> bBusy ,
    Error=> bError ,
    ErrorID=> uwErrorID
);
```

MC_ReadParameter

进行从设备的对象的读取。



名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
参数读取	24	28	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23页 AxisName.AxisRef. (轴信息)

■输入变量

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
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	设置从设备的对象。 关于详细内容，请参阅下述章节。 ☞ 108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。 ☞ 109页 选项(Options)

■输出变量

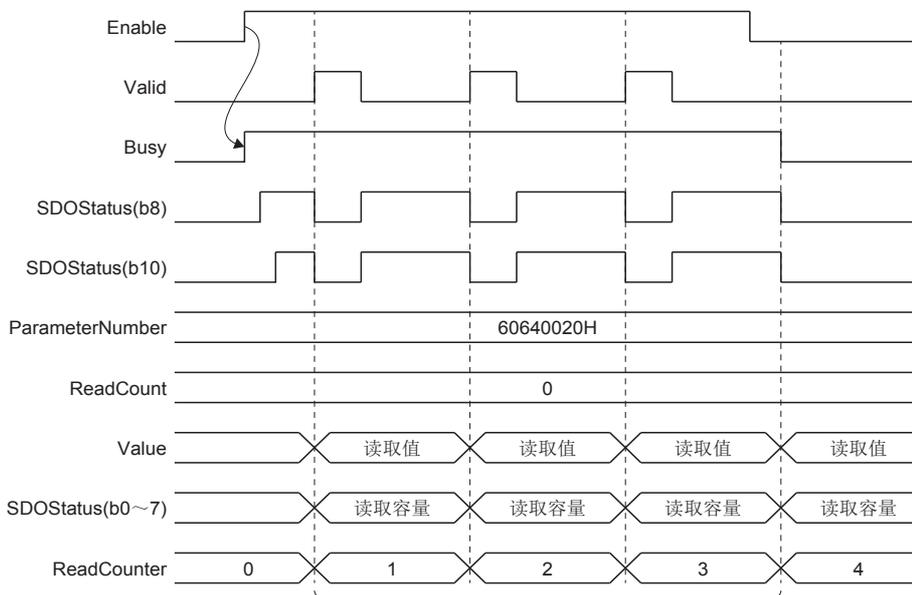
输出变量	名称	数据类型	初始值	内容
Valid	输出值有效	BOOL	FALSE	输出值为有效时, 将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_ReadParameter(参数读取)时, 将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时, 将返回出错代码。 关于出错代码的详细内容, 请参阅下述手册。 【】MELSEC iQ-R运动模块用户手册(应用篇)
SDOErrorID	瞬时出错代码	DWORD (HEX)	0000000H	在SDO通信中发生了异常时, 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。 【】109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及以后来读取从对象。在此情况下, 使用瞬时传送功能发送接收对象数据。
- 从从设备接收了对瞬时请求的出错(响应代码(SDO Abort Code))的情况下, 出错(Error)将变为TRUE, 并在出错代码(ErrorID)中将存储“SDO通信异常(出错代码: 1800H)”, 在瞬时出错代码(SDOErrorID)中将存储从设备的响应代码(SDO Abort Code)。

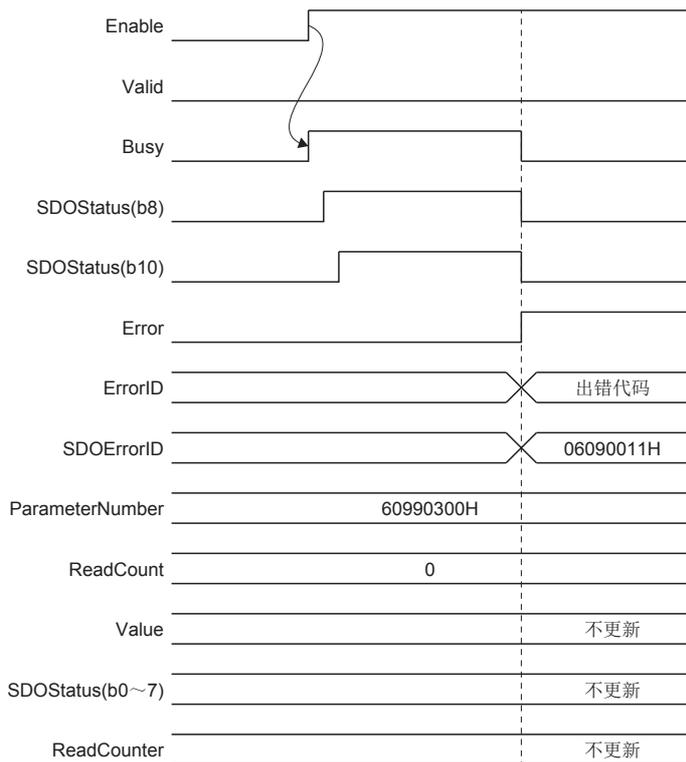
■时序图

- 正常完成的情况下

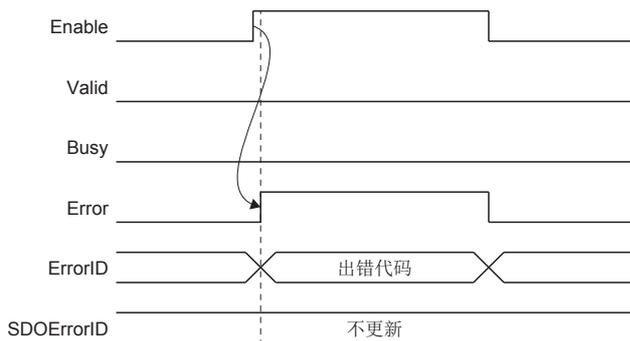


在有效(Enable)变为FALSE之前, 按读取次数(ReadCount)中设置的次数执行读取
(读取次数(ReadCount)为“0”的情况下, 在有效(Enable)变为FALSE之前连续执行)

- 异常完成的情况下
 - 发生SDO通信出错时



- 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	从设备的对象

- 设置从设备的对象的参数编号(ParameterNumber)的内容如下所示。



- 对于从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。设置了设置值以外的值的情况下，将变为“超出参数编号范围(出错代码: 34A2H)”。参数编号(ParameterNumber)中设置的容量与从对象的默认容量不相同时的动作取决于驱动器的规格。

例

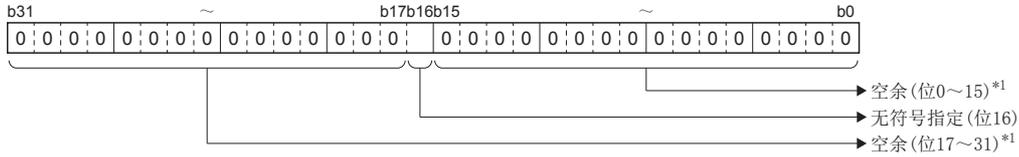
设置对象索引“6099H”、子索引“02H”的UNSIGNED32对象、容量“00H”的情况下设置“60990200H”。

关于各从设备的对象，请参阅从设备的手册。

■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。

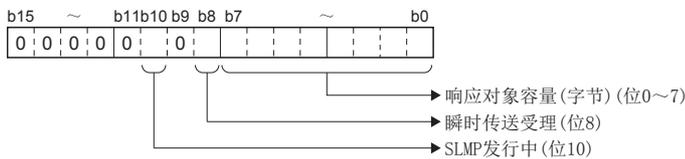


*1 空余中应设置“0”。设置了“0”以外的情况下，将变为“超出Options范围(出错代码: 1A4EH)”。

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，存储从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从对象作为对象的MC_ReadParameter(参数读取)时, 将变为“参数读写FB执行不可出错(出错代码: 34A1H)”, 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的原点数据的获取。因此, 执行MC_Home(原点复位)时正在执行将从对象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基准点读取出错(出错代码: 1AAAH)”。
- 对整数型、BOOL型的参数及从对象进行读取的情况下, 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。从设备对象的情况下, 关于参数的保存方法, 请参阅从设备的手册。
- 瞬时传送中的从设备解除连接的情况下, 将解除连接的从设备作为对象的瞬时请求将全部出错, 并发生“SDO通信异常(出错代码: 1800H)”。

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

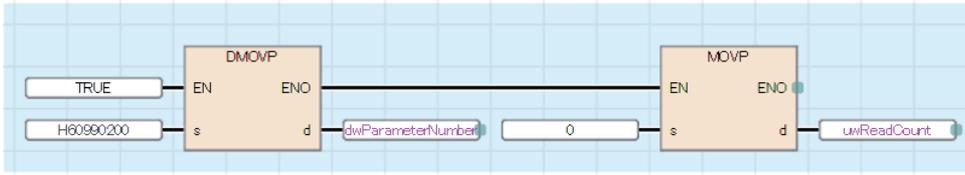
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

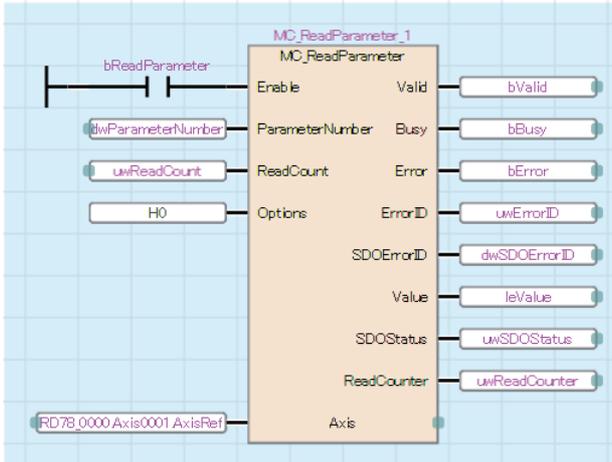
标签名	数据类型	注释
MC_ReadParameter_1	MC_ReadParameter	参数读取FB
bReadParameter	位	参数读取指令
dwParameterNumber	双字[无符号]/位串[32位](16进制数)	参数编号
uwReadCount	字[无符号]/位串[16位]	读取次数
bValid	位	输出值有效
bBusy	位	执行中
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码
dwSDOErrorID	双字[无符号]/位串[32位](16进制数)	瞬时出错代码
leValue	双精度实数	读取值
uwSDOStatus	字[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的程序(CPU模块侧)

- 参数读取用数据设置



- 参数读取



■ST的程序(运动模块侧)

//-----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=> lValue ,
  SDOStatus=> uwSDOStatus ,
  ReadCounter=> uwReadCounter
);
```

MC_WriteParameter

进行从设备的对象的写入。

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>

名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
参数写入	32	14	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23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~ FFFFFFFFH	00000000H	设置从设备的对象。 关于详细内容，请参阅下述章节。 ☞ 114页 参数编号(ParameterNumber)
Value	设置值	LREAL	启动时	—	0.0	设置已设置的参数的设置值。
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。 ☞ 115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。 ☞ 115页 选项(Options)

■输出变量

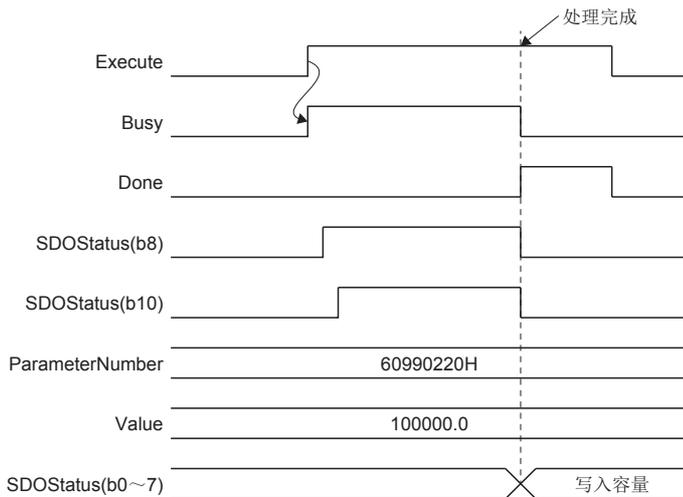
输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	参数的写入完成时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_WriteParameter(参数写入)时，将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 📖 MELSEC iQ-R运动模块用户手册(应用篇)
SDOErrorID	瞬时出错代码	DWORD (HEX)	00000000H	在SDO通信中发生了异常时，将返回响应代码(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。 📖 115页 SDO传送状态(SDOStatus)

功能

- 通过在参数编号(ParameterNumber)中设置00010000H及以后来写入从对象。在此情况下，使用瞬时传送功能发送接收对象数据。
- 从从设备接收了对瞬时请求的出错(响应代码(SDO Abort Code))的情况下，出错(Error)将变为TRUE，并在出错代码(ErrorID)中将存储“SDO通信异常(出错代码: 1800H)”，在瞬时出错代码(SDOErrorID)中将存储从设备的响应代码(SDO Abort Code)。

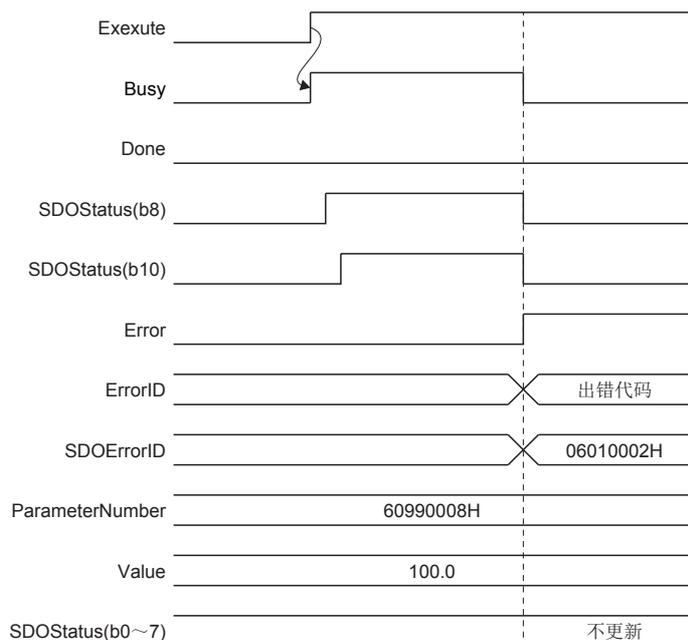
■时序图

- 正常完成的情况下

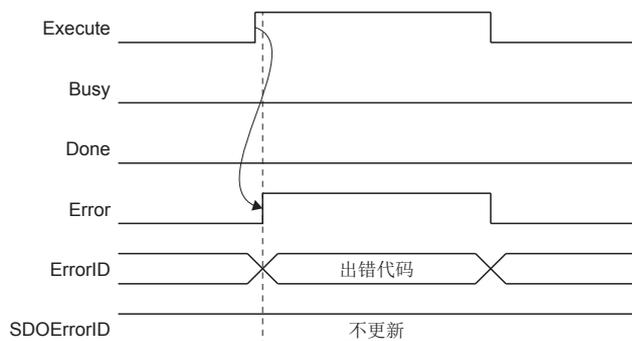


• 异常完成的情况下

- 发生SDO通信出错时



- 输入输出变量、输入变量异常

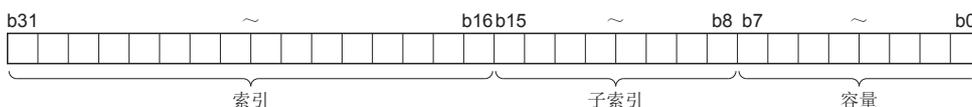


■ 参数编号 (ParameterNumber)

- 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~FFFFFFFFH	从设备的对象

- 设置从设备的对象的参数编号 (ParameterNumber) 的内容如下所示。



- 对于从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。设置了设置值以外的值的情况下，将变为“超出参数编号范围(出错代码: 34A2H)”。参数编号 (ParameterNumber) 中设置的容量与从对象的默认容量不相同时的动作取决于驱动器的规格。

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)

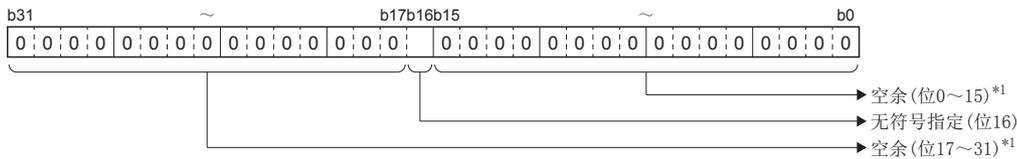
参数写入执行时，将写入的时机通过启动模式 (ExecutionMode) 进行设置。

设置值	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。

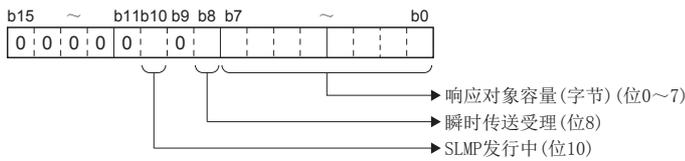


*1 空余中应设置“0”。设置了“0”以外的情况下，将变为“超出Options范围 (出错代码: 1A4EH)”。

位	名称	内容
16	无符号指定	设置将整数数据的设置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，存储从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从对象作为对象的MC_WriteParameter(参数写入)时, 将变为“参数读写FB执行不可出错(出错代码: 34A1H)”, 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的原点数据的获取。因此, 执行MC_Home(原点复位)时正在执行将从对象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基准点读取出错(出错代码: 1AAAH)”。
- 对整数型、BOOL型的参数及从对象进行写入的情况下, MC_WriteParameter(参数写入)的设置值将转换为整数型、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

设置值(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。从设备对象的情况下, 关于参数的保存方法, 请参阅从设备的手册。
- 瞬时传送中的从设备解除连接的情况下, 将解除连接的从设备作为对象的瞬时请求将全部出错, 并发生“SDO通信异常(出错代码: 1800H)”。

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

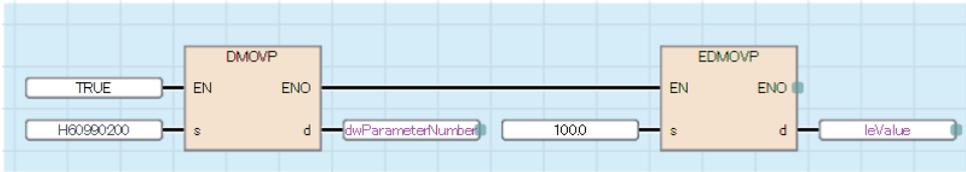
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

使用的标签

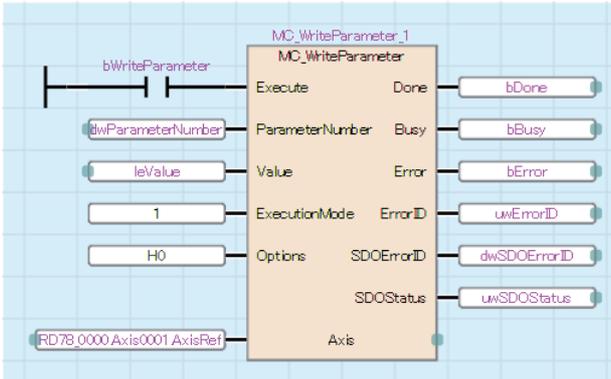
标签名	数据类型	注释
MC_WriteParameter_1	MC_WriteParameter	参数写入FB
bWriteParameter	位	参数写入指令
dwParameterNumber	双字[无符号]/位串[3位](16进制数)	参数编号
leValue	双精度实数	设置值
bDone	位	执行完成
bBusy	位	执行中
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码
dwSDOErrorID	双字[无符号]/位串[3位](16进制数)	瞬时出错代码
uwSDOStatus	字[无符号]/位串[16位]	SDO传送状态

■FBD/LD的程序(CPU模块侧)

- 参数写入用数据设置



- 参数写入



■ST的程序(运动模块侧)

//-----参数写入用数据设置-----

dwParameterNumber:= H60990200;

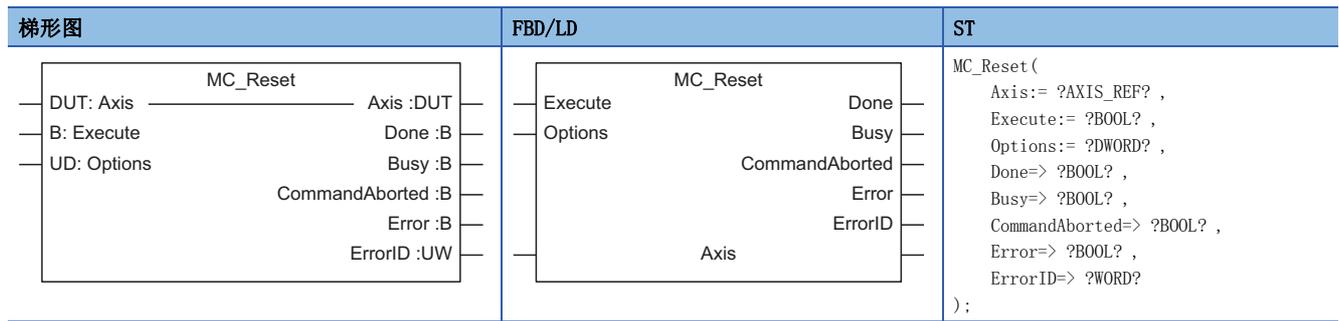
leValid:= 100.0;

//-----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
);
```

MC_Reset

对轴的出错、警告进行复位。



名称	输入区域点数(byte)	输出区域点数(byte)	编译方式	FB动作
轴出错复位	16	6	子程序型	随时执行型

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.)，请参阅下述章节。 23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(轴出错复位)。
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(出错代码: 1A4EH)”。

■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(轴出错复位)时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	由于超时，MC_Reset(轴出错复位)中断时将变为TRUE。 执行指令(Execute)变为FALSE时，将变为FALSE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD(UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 MELSEC iQ-R运动模块用户手册(应用篇)

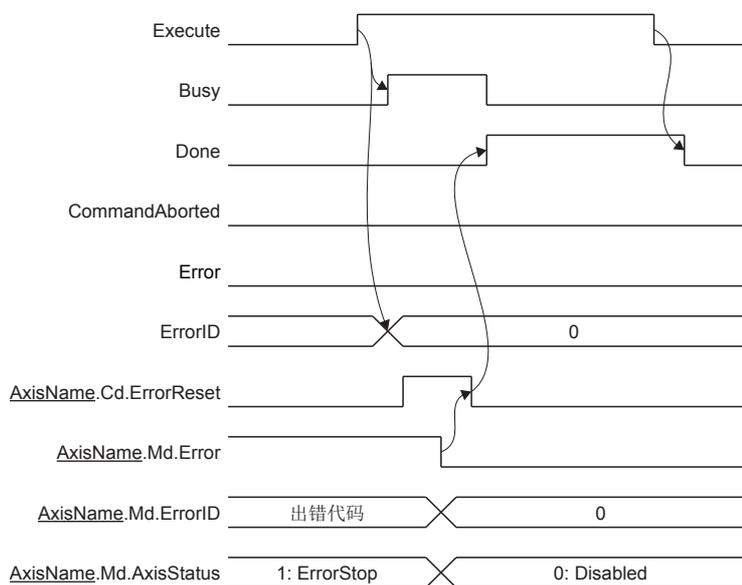
功能

- 在执行指令 (Execute) 为 TRUE 时执行 MC_Reset (轴出错复位)，开始处理时执行中 (Busy) 将变为 TRUE，并将对象轴的轴出错复位 (AxisName. Cd. ErrorReset) 置为 TRUE。
- 轴的出错、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。
- 如果在轴状态 (AxisName. Md. AxisStatus) 为 “1: 出错停止中 (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。
- 即使在残留了轴的出错、警告的原因的状态下将执行指令 (Execute) 置为 TRUE，也不解除出错、警告。在此情况下，执行中断 (CommandAborted) 将变为 TRUE。应将执行指令 (Execute) 置为 FALSE 一次，消除了出错、警告的原因后，再次将执行指令 (Execute) 置为 TRUE。
- 在 MC_Reset (轴出错复位) 内发生了异常的情况下，出错 (Error) 将变为 TRUE，并将出错代码存储到出错代码 (ErrorID) 中。关于出错代码的详细内容，请参阅下述手册。

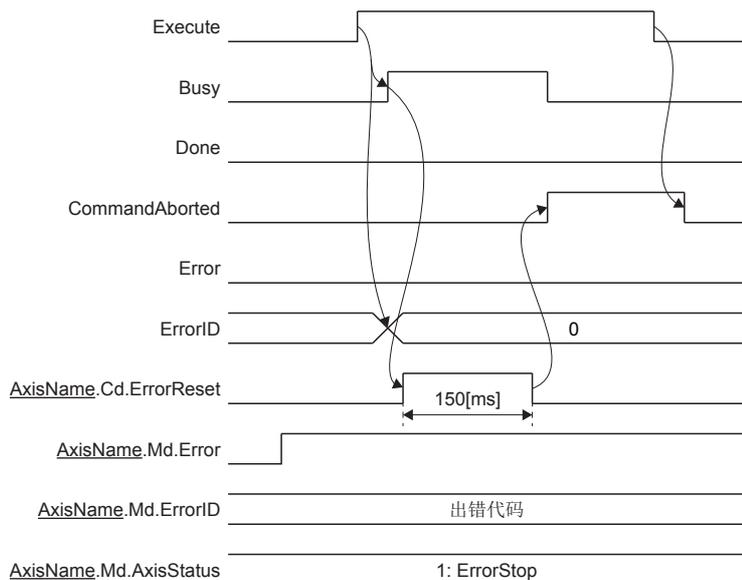
📖 MELSEC iQ-R 运动模块用户手册 (应用篇)

■ 时序图

- 正常完成的情况下



- 超时的情况下



注意事项

- 请勿在MC_Reset (轴出错复位) 的执行中进行轴出错复位 (AxisName.Cd.ErrorReset) 的直接操作。
- 从CPU模块进行轴出错复位的情况下，应不进行轴出错复位 (AxisName.Cd.ErrorReset) 的操作，而使用MC_Reset (轴出错复位)。

程序示例

将出错复位指令(bErrorReset)置为TRUE，并对轴1(Axis0001)的出错、警告进行复位的程序示例如下所示。

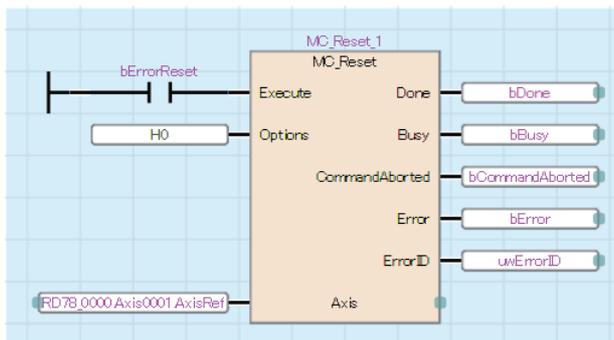
■轴

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	轴出错复位FB
bErrorReset	位	出错复位指令
bDone	位	执行完成
bBusy	位	执行中
bCommandAborted	位	执行中断
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

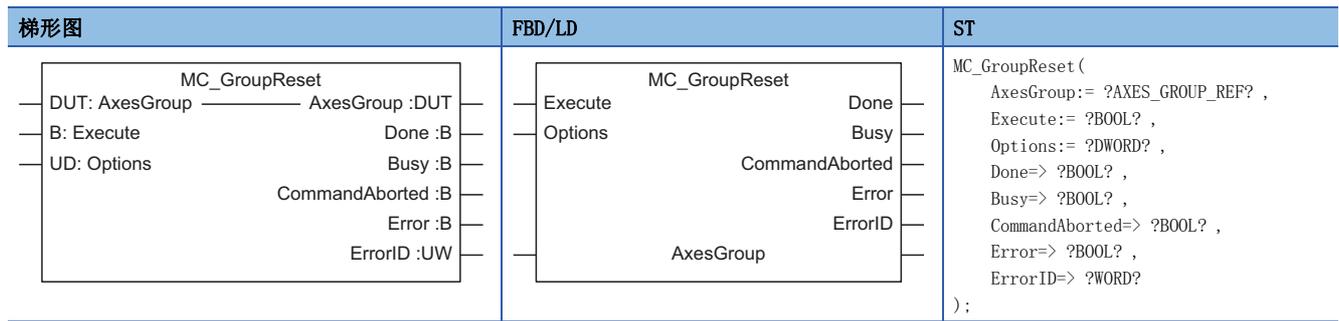


■ST的程序(运动模块侧)

```
MC_Reset_1(
    Axis:= Axis0001.AxisRef ,
    Execute:= bErrorReset ,
    Options:= H00000000 ,
    Done=> bDone ,
    Busy=> bBusy ,
    CommandAborted=> bCommandAborted ,
    Error=> bError ,
    ErrorID=> uwErrorID
);
```

MC_GroupReset

对轴组及轴组中所属的各轴的出错、警告进行复位。



名称	输入区域点数(byte)	输出区域点数(byte)	编译方式	FB动作
轴组出错复位	16	6	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 44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 (轴组出错复位)。
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(出错代码: 1A4EH)”。

■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 (轴组出错复位)时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	由于超时，MC_GroupReset (轴组出错复位) 中断时将变为TRUE。 执行指令(Execute)变为FALSE时，将变为FALSE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册 (应用篇)

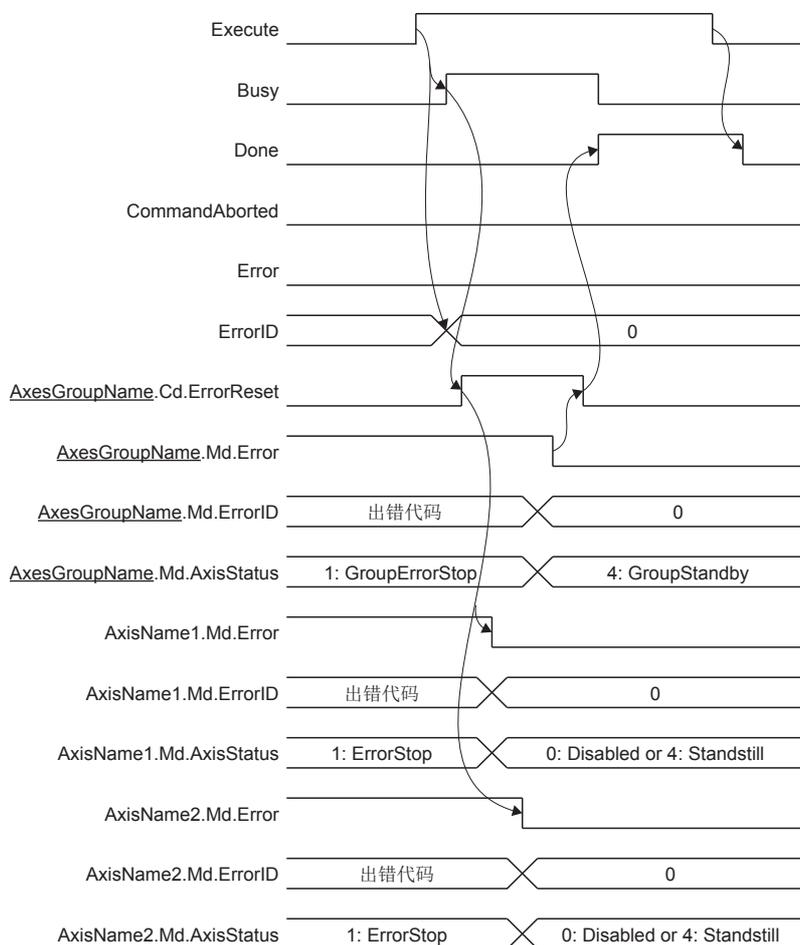
功能

- 在执行指令(Execute)的上升沿检测时执行MC_GroupReset(轴组出错复位), 开始处理时执行中(Busy)将变为TRUE, 并将对象轴组的轴组出错复位(AxesGroupName.Cd.ErrorReset)置为TRUE。
- 轴组的出错或警告及单轴的出错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。
- 如果在轴组状态(AxesGroupName.Md.GroupStatus)为“1: 出错停止中(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。
- 即使在残留了轴的出错或警告的原因的状态下将执行指令(Execute)置为TRUE, 也不解除出错、警告。在此情况下, 从FB执行后1秒以内未解除出错时, 执行中断(CommandAborted)将变为TRUE, 并将对象轴组的轴组出错复位(AxesGroupName.Cd.ErrorReset)置为FALSE。应将执行指令(Execute)置为FALSE一次, 消除了出错或警告的原因后, 再次将执行指令(Execute)置为TRUE。
- 在MC_GroupReset(轴组出错复位)内发生了异常的情况下, 出错(Error)将变为TRUE, 并将出错代码存储到出错代码(ErrorID)中。关于出错代码的详细内容, 请参阅下述手册。

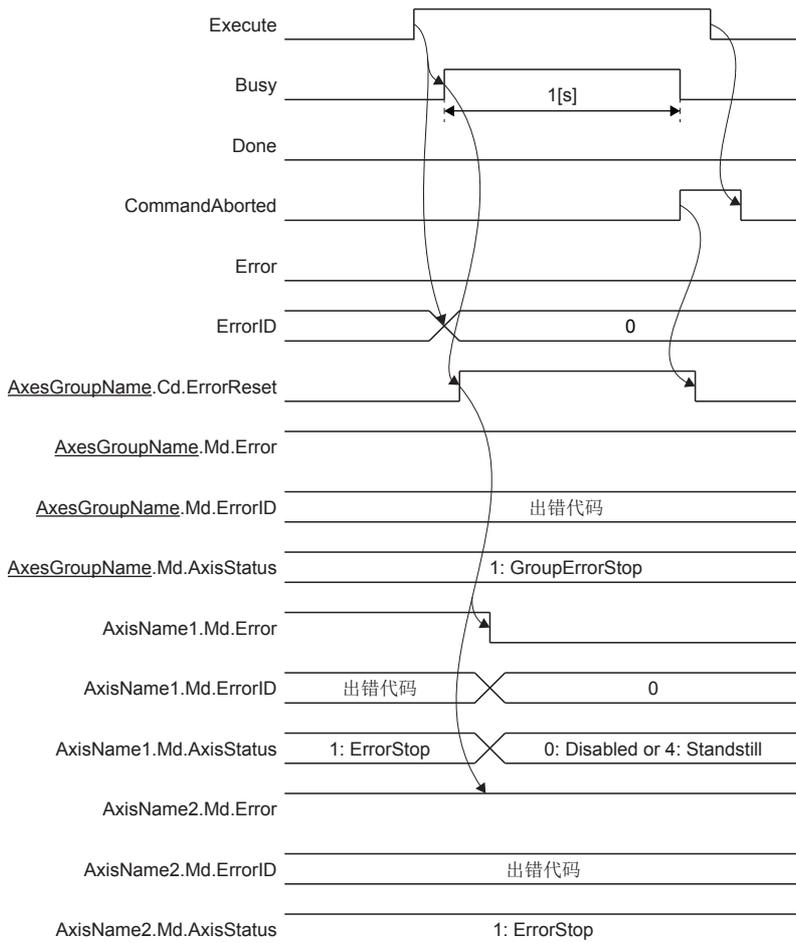
📖 MELSEC iQ-R运动模块用户手册(应用篇)

■时序图

- 正常完成的情况下



• 超时的情况下



注意事项

- 请勿在 `MC_GroupReset` (轴组出错复位) 的执行中进行轴组出错复位 (`AxisGroupName.Cd.ErrorReset`) 的直接操作。
- 从CPU模块进行出错复位的情况下，应不进行轴组出错复位 (`AxisGroupName.Cd.ErrorReset`) 的操作，而使用 `MC_GroupReset` (轴组出错复位)。

程序示例

将轴组出错复位指令(bGroupErrorReset)置为TRUE，并对轴组1(AxesGroup001)的出错、警告进行复位的程序示例如下所示。

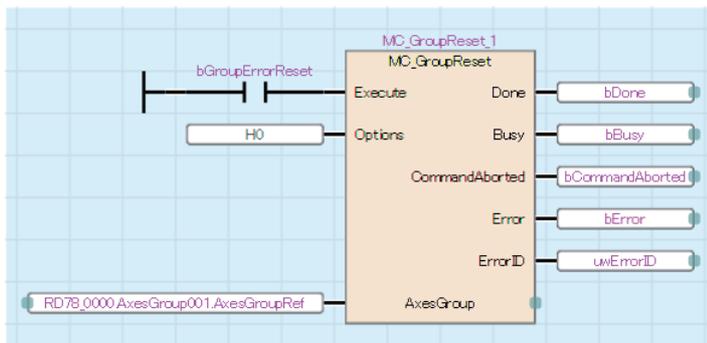
■轴组

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	轴组出错复位FB
bGroupErrorReset	位	轴组出错复位指令
bDone	位	执行完成
bBusy	位	执行中
bCommandAborted	位	执行中断
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)



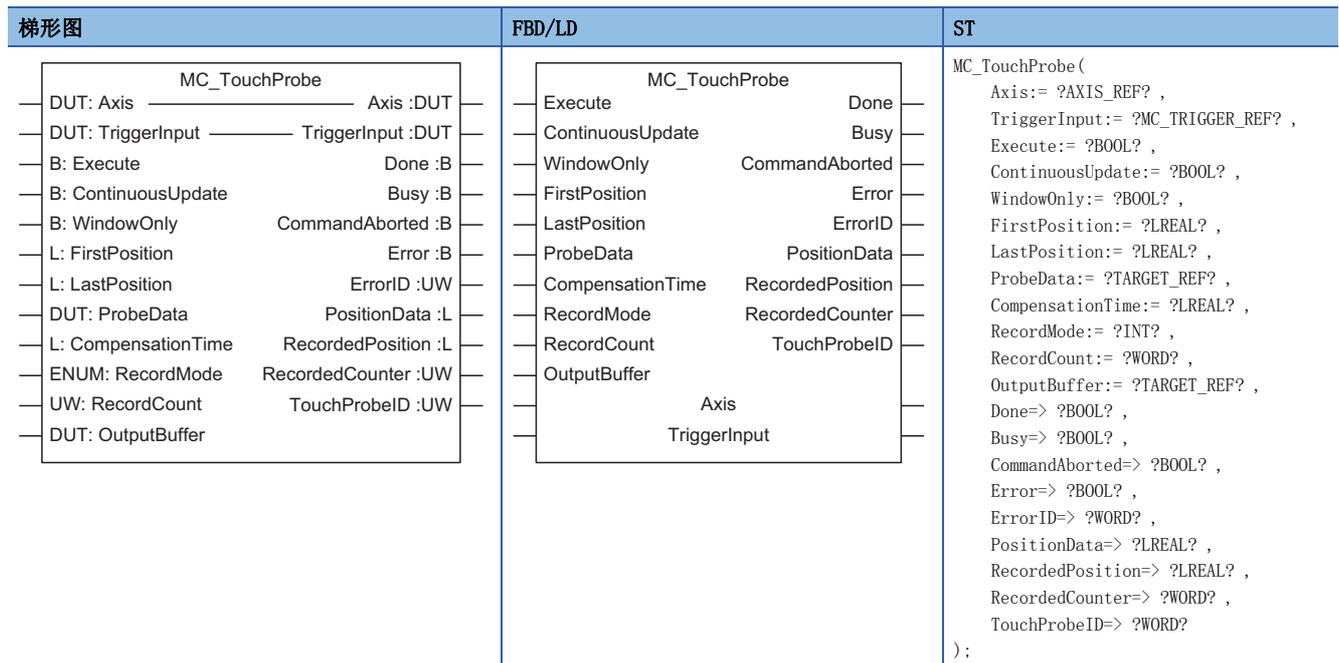
■ST的程序(运动模块侧)

```

MC_GroupReset_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bGroupErrorReset,
    Options:= H00000000 ,
    Done=> bDone ,
    Busy=> bBusy ,
    CommandAborted=> bCommandAborted,
    Error=> bError ,
    ErrorID=> uwErrorID
);
    
```

MC_TouchProbe

通过触发事件发生记录任意数据。



名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
触摸探头有效	480	28	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23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。 ☞ 129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(触摸探头有效)。
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。 ☞ 129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。 ☞ 129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。 ☞ 129页 窗口有效(WindowOnly)/下限位置(FirstPosition)/上限位置>LastPosition)
ProbeData	探头数据	TARGET_REF	启动时	—	—	设置探头数据。关于详细内容，请参阅下述章节。 ☞ 130页 探头数据(ProbeData)
CompensationTime	补偿时间	LREAL	启动时	-5.0~5.0 [s]	0.0	设置对触摸探头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。 ☞ 131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。 ☞ 132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。 ☞ 132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。 ☞ 133页 锁存数据存储目标(OutputBuffer)

■输出变量

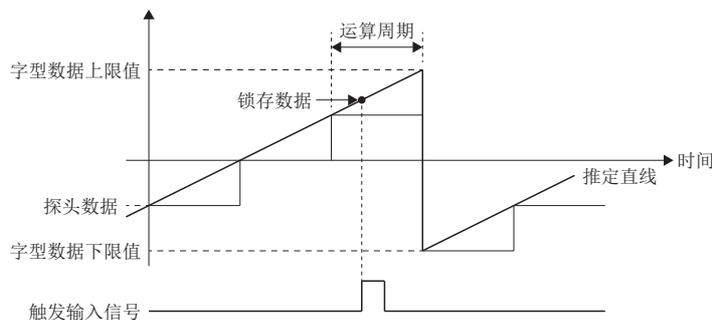
输出变量	名称	数据类型	初始值	内容
Done	完成	BOOL	FALSE	输出值为有效时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_TouchProbe(触摸探头有效)时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	由于出错及多重启动等，MC_TouchProbe(触摸探头有效)的执行中断时，将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时，将返回出错代码。关于出错代码的详细内容，请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册(应用篇)
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	探头ID	WORD (UINT)	0	存储MC_TouchProbe(触摸探头有效)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),将锁存数据存储在锁存位置(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头数据进行推定计算。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。

■探头数据为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从站检测出信号的时间。

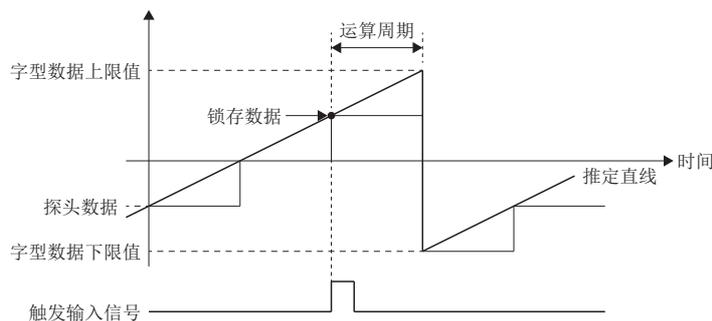


*: 关于外部信号选择的高精度输入的设置方法,请参阅下述手册。

■MELSEC iQ-R运动模块用户手册(应用篇)

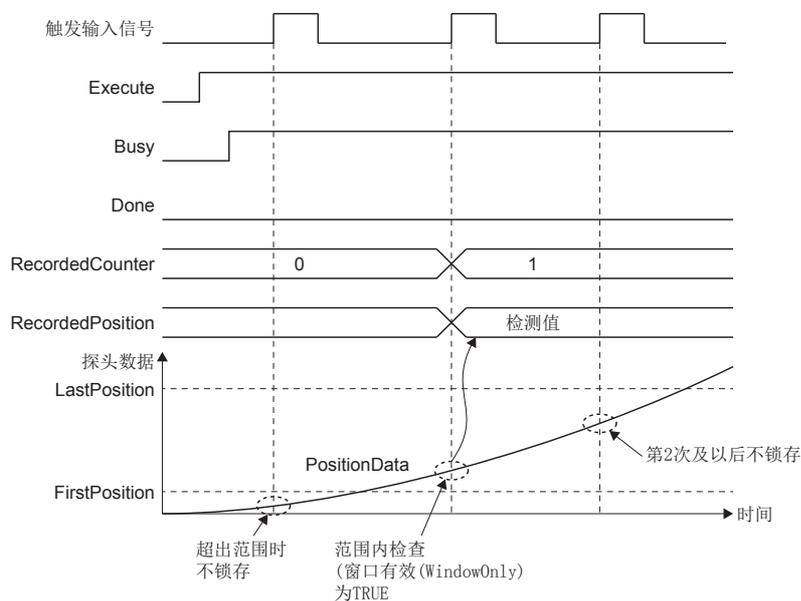
*: 关于信号检测时间的精度,请参阅所使用的从设备的手册。

- 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。



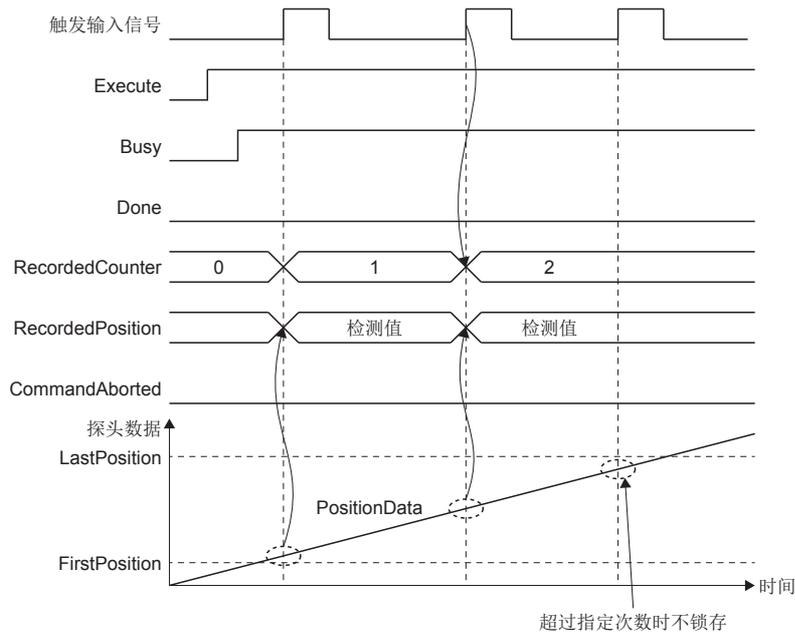
■时序图

- 正常完成的情况下
 - 锁存模式(RecordMode)为“0: 单发模式(OneShot)”的情况下



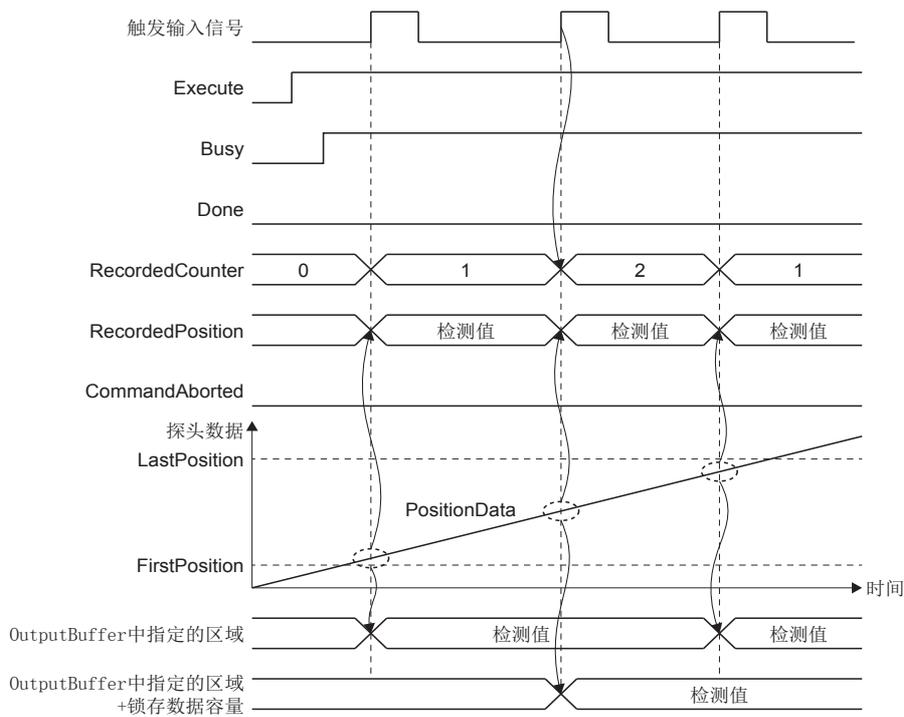
• 锁存模式(RecordMode)为“1: 指定次数模式(RecordCount)”的情况下

*: 锁存次数(RecordCount)为2次的情况下

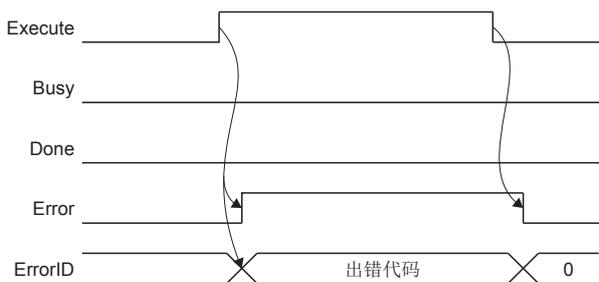


• 锁存模式(RecordMode)为“2: 环形缓冲模式(RingBuffer)”的情况下

*: 锁存次数(RecordCount)为2次的情况下



• 异常完成的情况下



■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。

☞ 60页 MC_TRIGGER_REF

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。

☞ 63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 ☞ 61页 TARGET_REF(输入信号)	■类型 • BOOL ■数据类型 • [OBJ]*1*2 • [VAR] • [DEV] • [CONST]
	Detection(信号检测方法)	INT (MC_SIGNAL_LOGIC) ☞ 75页 MC_SIGNAL_LOGIC	• 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(滤波器时间)	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，将发生出错。

在对象修饰中指定了站地址的情况下，轴信息(Axis)将被忽略。

要参照的站从模拟功能有效且已设置站地址的情况下，将参照模拟的对象。

*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。

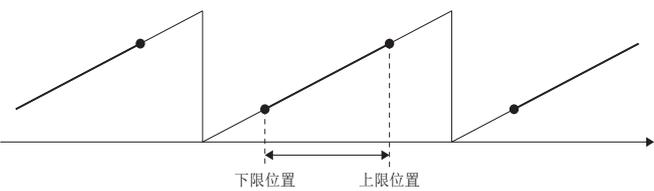
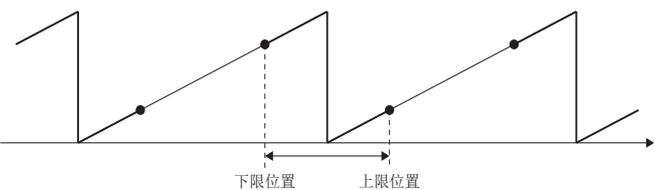
☞ MELSEC iQ-R运动模块用户手册(应用篇)

要点

使用外部信号高精度输入时，应将滤波器时间(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。从设备中存在输入滤波器的情况下，应使用从设备的滤波器。从设备中不存在输入滤波器的情况下，应同时使用触摸探头有效(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，只有探头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头处理。

下限位置(FirstPosition)/上限位置>LastPosition)的关系	触摸探头处理
下限位置(FirstPosition) ≤ 上限位置>LastPosition)	探头数据大于下限位置(FirstPosition)且小于上限位置>LastPosition)时，进行触摸探头处理。 
下限位置(FirstPosition) > 上限位置>LastPosition)	探头数据小于上限位置>LastPosition)或大于下限位置(FirstPosition)时，进行触摸探头处理。 

■探头数据 (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。

☞ 61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，将发生出错。 要参照的站从模拟功能有效且已设置站地址的情况下，将参照模拟的对象。

- addon_base_MotionControlGeneral的版本为“1.15及以前”的情况下，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)”。
- 将探头数据(ProbeData)设置为TARGET_REF结构体的对象(Target)的示例如下所示。

例

对象(Target)的设置示例

- [DEV] (LREAL)G11500000
- [VAR]OutputBufferData1*1
- [VAR] (LREAL)OutputBufferData2*2

*1 OutputBufferData1为LREAL型的实例

*2 OutputBufferData2为WORD型的实例

- 数据将更新为环形计数器。环形计数器的上限值、下限值根据探头数据(ProbeData)的数据类型、数据类型而有所不同。

数据类型	内容
[AXIS]	<p>■在对象修饰中设置了“@Position”的情况下</p> <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。 <p>■在对象修饰中设置了“@CumulativePos”的情况下</p> <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]	<p><例></p> <p>在[VAR]中设置了字型标签的情况下。</p> <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。未满足时探头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)

补偿触摸探头处理的延迟时间。除触发输入信号的补偿时间以外，对触摸探头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。

合计的补偿时间如下所示。

补偿时间(CompensationTime) - 补偿时间(TriggerInput.Signal.CompensationTime)

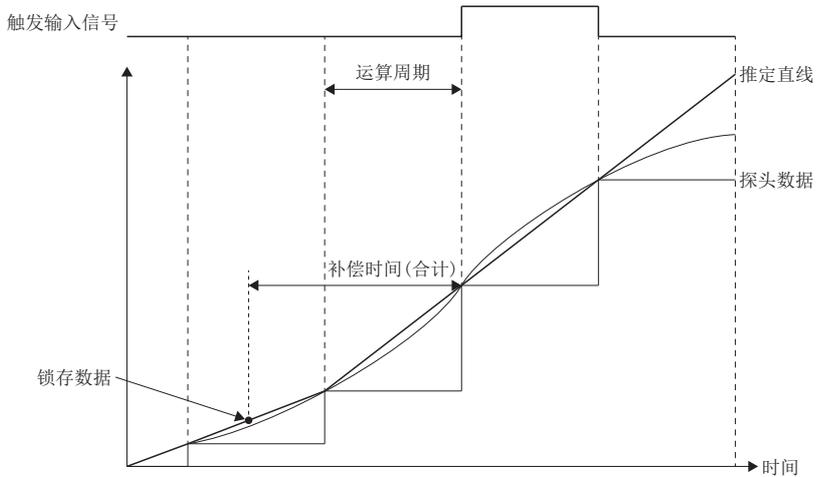
要点

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)

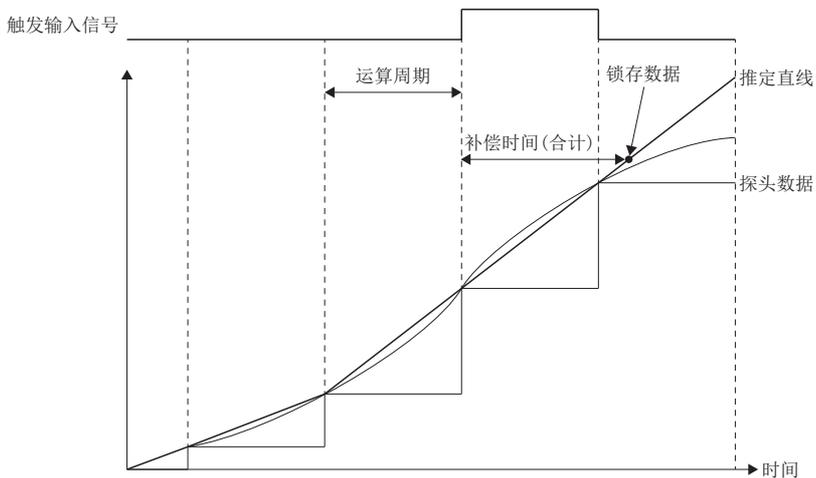
- 补偿延迟的情况下(合计的补偿时间为正的值的情况下)

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。



- 补偿提前的情况下(合计的补偿时间为负的值的情况下)

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。

设置值	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) 的设置值存储锁存数据。 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头功能的情况下, 应执行MC_AbortTrigger (触摸探头无效)。
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) 的设置值的环形缓冲中。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头功能的情况下, 应执行MC_AbortTrigger (触摸探头无效)。

- 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。省略的情况下, 将变为“锁存次数不正确 (出错代码: 3408H)”。
2: 环形缓冲模式 (RingBuffer)	设置环形缓冲的次数。省略的情况下, 将变为“锁存次数不正确 (出错代码: 3408H)”。

■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。

☞ 61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)	每次锁存动作时将“+1”。
2: 环形缓冲模式(RingBuffer)	以下区域 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"> • [DEV] (LREAL)G11500000 • [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"> • [DEV] (LREAL[0..99])G11500000 • [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(出错代码: 349FH)”。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”(出错代码: 349FH)。
- 各项的设置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)	省略/设置	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)	省略	省略/设置	出错	—	
	设置	省略	设置次数	锁存位置(RecordedPosition)	
2: 环形缓冲模式(RingBuffer)	省略	设置	始终	锁存位置(RecordedPosition)/缓冲	
		省略/设置		出错	—
		省略		始终	锁存位置(RecordedPosition)
	设置	设置		锁存位置(RecordedPosition)/缓冲	

注意事项

- 在MC_TouchProbe(触摸探头有效)的执行中(Busy)变为TRUE后发生了运算周期溢出的情况下，推定计算的精度可能会降低。
- 运动控制FB的输入输出参数的刷新在FB的调用时机进行，但是不通过FB调用任务(正常/恒定周期)，而是通过运算周期进行控制。
- 即使对象轴未连接的情况下及出错的情况下，也执行MC_TouchProbe(触摸探头有效)。

程序示例

将触摸探头启动(bTouchProbe)置为TRUE，每当运动系统内的触摸探头信号(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，将触摸探头无效启动(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	公开标签
G_bTouchProbeSignal	位	触摸探头信号	有效

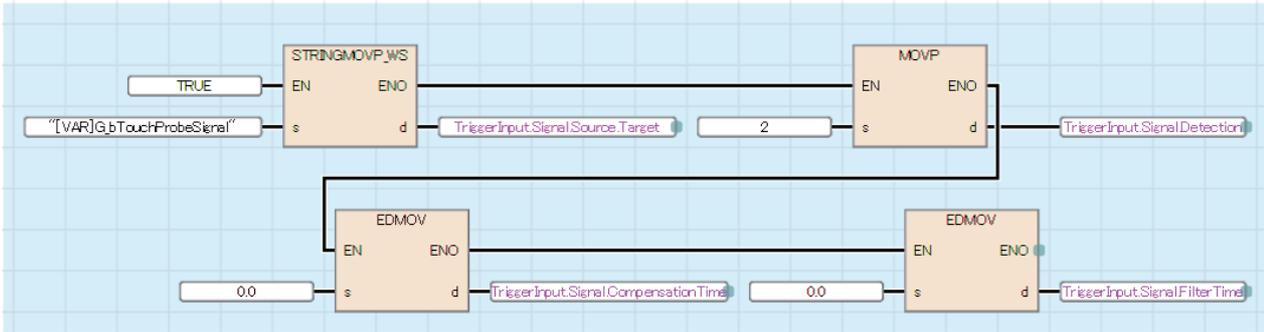
■使用的标签

标签名	数据类型	注释
MC_TouchProbe_1	MC_TouchProbe	触摸探头有效FB
bTouchProbe	位	触摸探头启动
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	探头数据
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	触摸探头有效完成
bTPBusy	位	触摸探头有效执行中
bTPCommandAborted	位	触摸探头有效执行中断
bTPErrors	位	触摸探头有效出错
uwTPErrorsID	字[无符号]/位串[16位]	触摸探头有效出错代码
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]	探头ID
leLatchPos_array	双精度实数(0..1)	锁存位置存储用变量
MC_AbortTrigger_1	MC_AbortTrigger	触摸探头无效FB
bAbortTrigger	位	触摸探头无效启动
bATDone	位	触摸探头无效完成
bATBusy	位	触摸探头无效处理中
bATCommandAborted	位	触摸探头无效执行中断
bATErrors	位	触摸探头无效出错
uwATErrorsID	字[无符号]/位串[16位]	触摸探头无效出错代码
TriggerInput*1	MC_TRIGGER_REF	触发输入信号

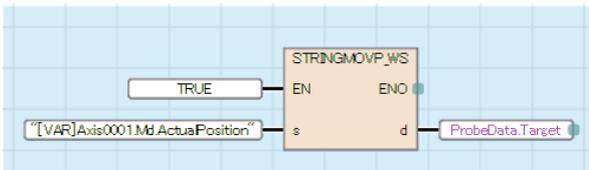
*1 仅使用FBD/LD的程序

■FBD/LD的程序(CPU模块侧)

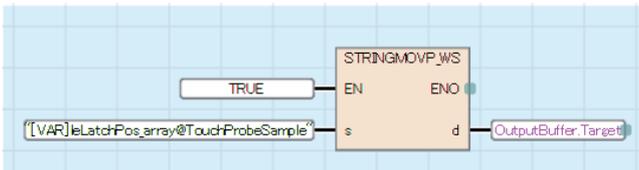
- 触发输入信号(输入输出数据)



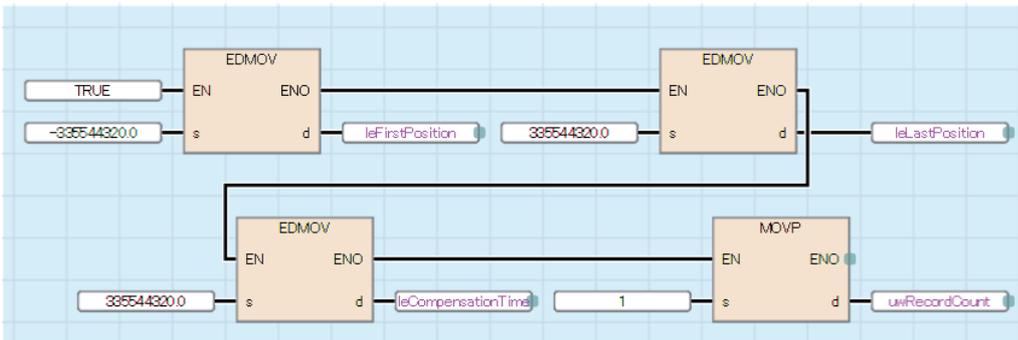
- 锁存对象用数据



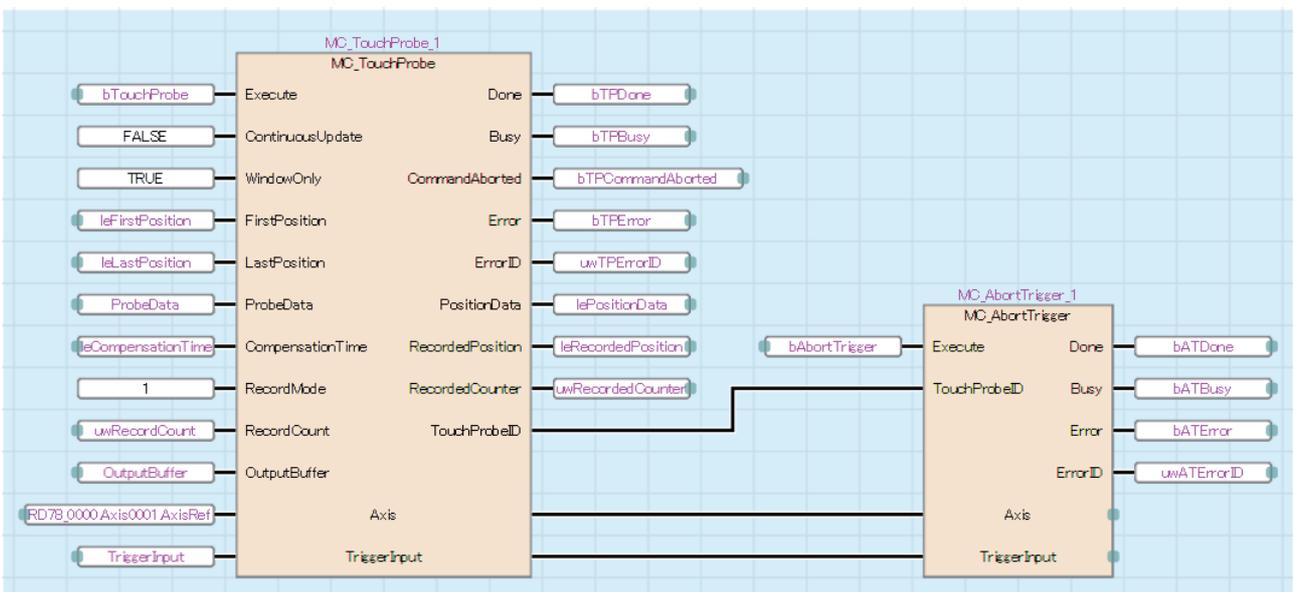
- 触摸探头处理



- 触摸探头用数据设置



- 触摸探头有效/无效



■ST的程序(运动模块侧)

```
//-----锁存对象用数据-----
ProbeData.Target:= "[VAR]Axis0001.Md.ActualPosition";

//-----触摸探头处理-----
OutputBuffer.Target:= "[VAR]leLatchPos_array@TouchProbeSample";

//-----触摸探头用数据设置-----
leFirstPosition:= -335544320.0;
leLastPosition:= 335544320.0;
leCompensationTime:= 0.0;
uwRecordCount:= 1;

//-----触摸探头有效-----
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
);

//-----触摸探头无效-----
MC_AbortTrigger_1(
    Axis:= Axis0001.AxisRef ,
    TriggerInput:= SignalData0001 ,
    Execute:= bAbortTrigger ,
    TouchProbeID:= MC_TouchProbe_1.TouchProbeID ,
    Done=> bATDone ,
    Busy=> bATBusy ,
    Error=> bATError ,
    ErrorID=> uwATErrorID
);
```

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>		
名称	输入区域点数(byte)	输出区域点数(byte)	编译方式	FB动作
触摸探头无效	172	6	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23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。 ☞ 138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(触摸探头无效)。
TouchProbeID	触摸探头ID	WORD(UINT)	启动时	1~65535	0	设置无效的触摸探头的固有ID。 设置无效的MC_TouchProbe(触摸探头有效)的执行指令(Execute)为上升沿后的探头ID(TouchProbeID)中存储的触摸探头ID。

■输出变量

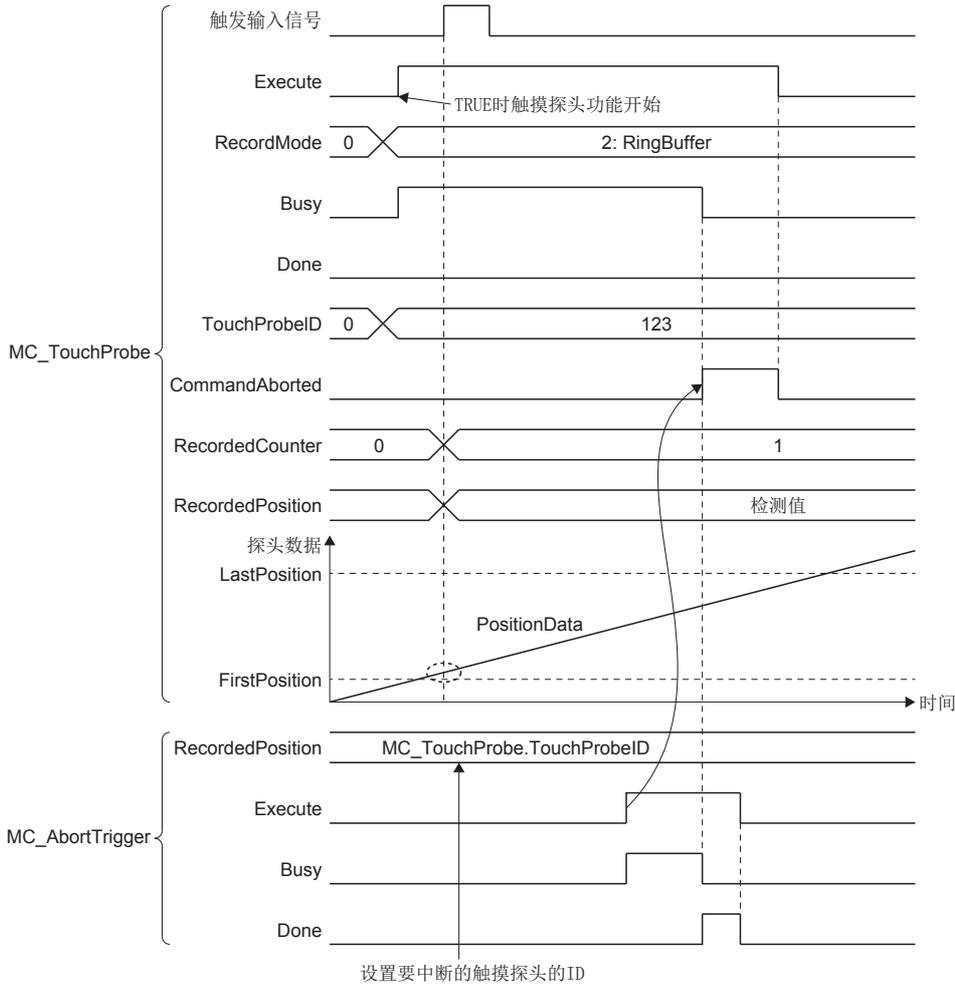
输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	输出值为有效时，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_AbortTrigger(触摸探头无效)时，将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD(UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册(应用篇)

功能

- 将触摸探头ID(TouchProbeID)中设置的MC_TouchProbe(触摸探头有效)置为无效。
- 执行MC_AbortTrigger(触摸探头无效)时，触摸探头ID(TouchProbeID)中设置的MC_TouchProbe(触摸探头有效)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。

☞ 11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。

☞ 60页 MC_TRIGGER_REF

触发输入信号(TriggerInput)中设置的MC_TARGET_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。

☞ 63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 ☞ 61页 TARGET_REF(输入信号)	■类型 <ul style="list-style-type: none"> • BOOL ■数据类型 <ul style="list-style-type: none"> • [VAR] • [DEV] • [CONST]
	Detection(信号检测方法)	INT (MC_SIGNAL_LOGIC) ☞ 75页 MC_SIGNAL_LOGIC	<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(滤波器时间)	LREAL	0.0~5.0[s]

程序示例

将触摸探头启动(bTouchProbe)置为TRUE，每当运动系统内的触摸探头信号(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，将触摸探头无效启动(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	公开标签
G_bTouchProbeSignal	位	触摸探头信号	有效

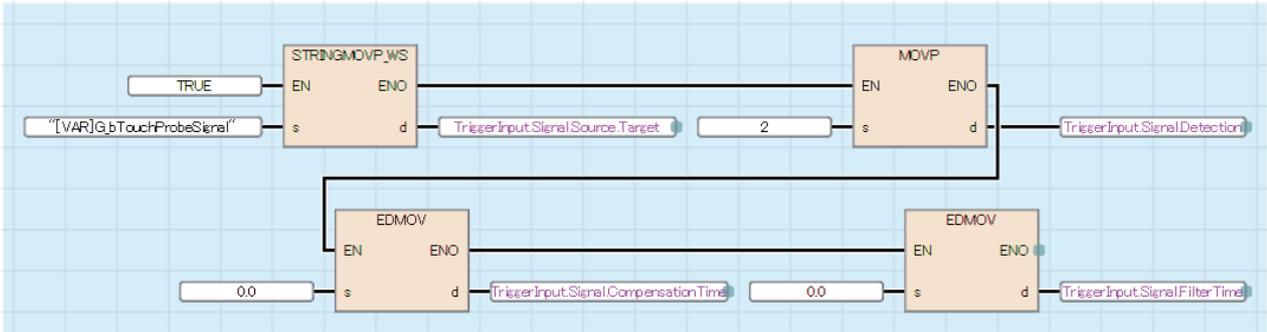
■使用的标签

标签名	数据类型	注释
MC_TouchProbe_1	MC_TouchProbe	触摸探头有效FB
bTouchProbe	位	触摸探头启动
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	探头数据
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	触摸探头有效完成
bTPBusy	位	触摸探头有效执行中
bTPCommandAborted	位	触摸探头有效执行中断
bTPError	位	触摸探头有效出错
uwTPErrorID	字[无符号]/位串[16位]	触摸探头有效出错代码
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]	探头ID
leLatchPos_array	双精度实数(0..1)	锁存位置存储用变量
MC_AbortTrigger_1	MC_AbortTrigger	触摸探头无效FB
bAbortTrigger	位	触摸探头无效启动
bATDone	位	触摸探头无效完成
bATBusy	位	触摸探头无效处理中
bATCommandAborted	位	触摸探头无效执行中断
bATError	位	触摸探头无效出错
uwATErrorID	字[无符号]/位串[16位]	触摸探头无效出错代码
TriggerInput*1	MC_TRIGGER_REF	触发输入信号

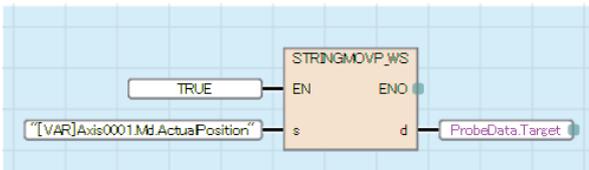
*1 仅使用FBD/LD的程序

■FBD/LD的程序(CPU模块侧)

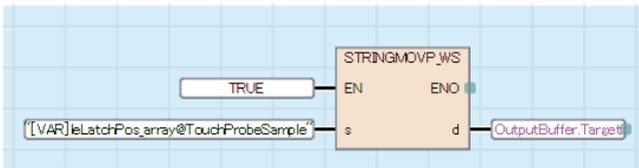
- 触发输入信号(输入输出数据)



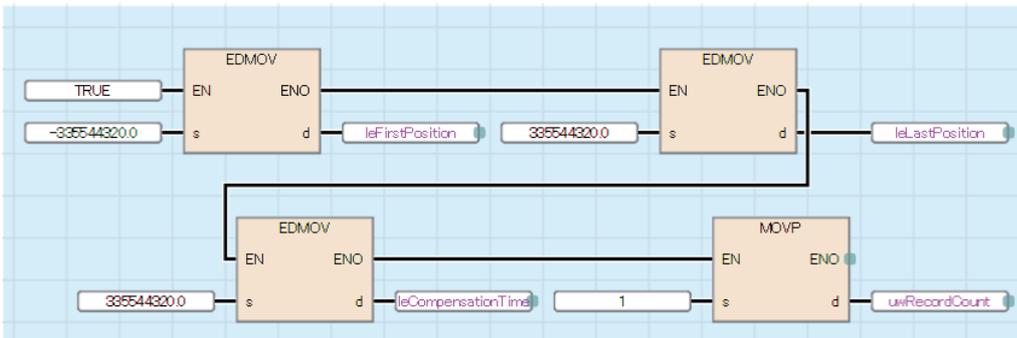
- 锁存对象用数据



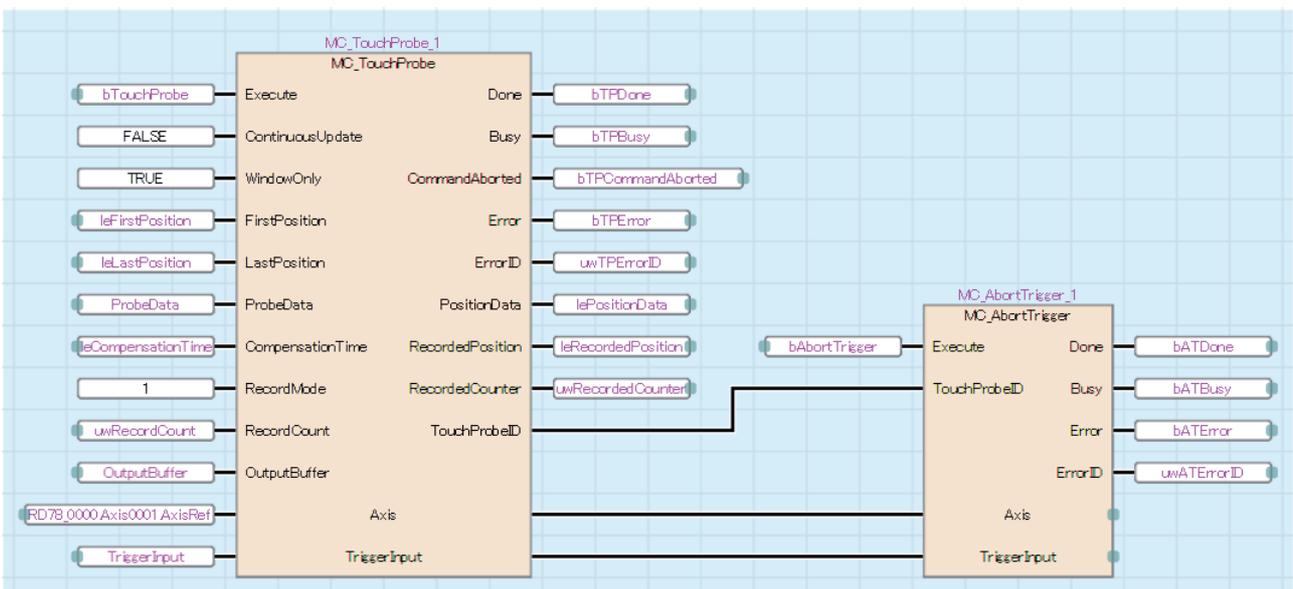
- 触摸探头处理



- 触摸探头用数据设置



- 触摸探头有效/无效



■ST的程序(运动模块侧)

```
//-----锁存对象用数据-----
ProbeData.Target:= "[VAR]Axis0001.Md.ActualPosition";

//-----触摸探头处理-----
OutputBuffer.Target:= "[VAR]leLatchPos_array@TouchProbeSample";

//-----触摸探头用数据设置-----
leFirstPosition:= -335544320.0;
leLastPosition:= 335544320.0;
leCompensationTime:= 0.0;
uwRecordCount:= 1;

//-----触摸探头有效-----
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
);

//-----触摸探头无效-----
MC_AbortTrigger_1(
  Axis:= Axis0001.AxisRef ,
  TriggerInput:= SignalData0001 ,
  Execute:= bAbortTrigger ,
  TouchProbeID:= MC_TouchProbe_1.TouchProbeID ,
  Done=> bATDone ,
  Busy=> bATBusy ,
  Error=> bATError ,
  ErrorID=> uwATErrorID
);
```

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>		
名称	输入区域点数(byte)	输出区域点数(byte)	编译方式	FB动作
凸轮表选择	158	18	子程序型	随时执行型

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	在运动模块侧使用MC_CamTableSelect(凸轮表选择)的情况下,可以省略。设置将被忽略。 在CPU模块侧使用MC_CamTableSelect(凸轮表选择)的情况下,设置输入输出No.(Master.StartIO)。轴No.(Master.AxisNo)将被忽略。 ☞ 23页 AxisName.AxisRef.(轴信息)
Slave	从轴	AXIS_REF	启动时	—	可省略	在运动模块侧使用MC_CamTableSelect(凸轮表选择)的情况下,可以省略。设置将被忽略。 在CPU模块侧使用MC_CamTableSelect(凸轮表选择)的情况下,设置输入输出No.(Slave.StartIO)。轴No.(Slave.AxisNo)将被忽略。 ☞ 23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)。 关于详细内容,请参阅下述章节。 ☞ 144页 凸轮表(CamTable)

要点

在CPU模块侧使用MC_CamTableSelect(凸轮表选择)的情况下,使用主轴(Master)、从轴(Slave)的输入输出No.(StartIO),但是不使用主轴(Master)、从轴(Slave)的轴No.(AxisNo)。
在运动模块侧使用MC_CamTableSelect(凸轮表选择)的情况下,即使设置主轴(Master)、从轴(Slave)也不将轴信息添加到凸轮数据中,也不将其用于控制。

■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。 ☞ 145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。 ☞ 145页 主轴绝对坐标(MasterAbsolute) *: 设置“TRUE”时, 将变为“超出MasterAbsolute范围(出错代码: 341DH)”。
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。 ☞ 146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。 ☞ 146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	出错	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	出错代码	WORD(UINT)	0	发生了异常时, 将返回出错代码。 关于出错代码的详细内容, 请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册(应用篇)
CamTableID	凸轮表ID	MC_CAM_ID	0	输出配置文件ID。

功能

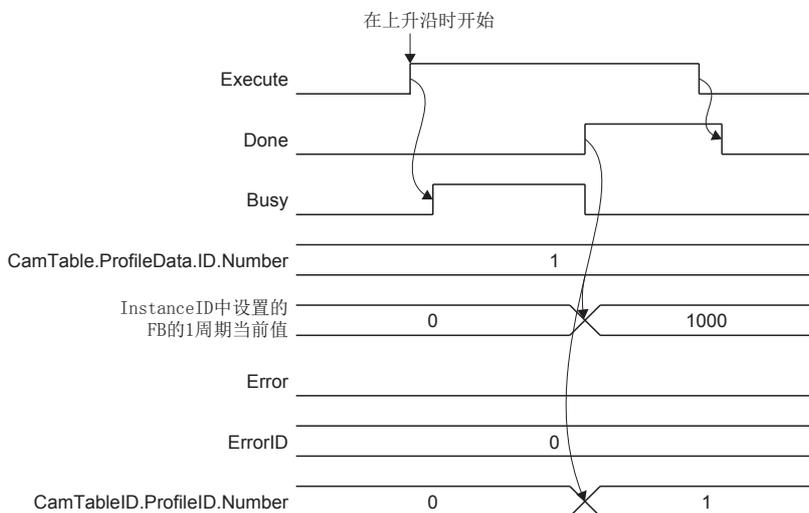
- 使用MC_CamTableSelect (凸轮表选择), 将PROFILE_DATA结构体的运算配置文件存储位置(Location)中设置的运算配置文件(凸轮数据)的文件展开到展开区域中。
- 展开到展开区域中的数据中, 将分配PROFILE_ID结构体的配置文件ID(Number)。

要点

- 关于运算配置文件(凸轮数据)的创建的详细内容, 请参阅下述手册。
 ④ MELSEC iQ-R运动模块用户手册(应用篇)
- MC_CamTableSelect (凸轮表选择)通过服务处理进行。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, 忽略文件的设置值, 并参照输入变量的设置进行展开。(不覆盖文件的设置值。)

■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。

④ 11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PROFILE_DATA结构体中设置运算配置文件(凸轮数据)。关于PROFILE_DATA结构体, 请参阅下述章节。

④ 63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(出错代码: 3452H)”, 且不进行展开处理。
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		已经存在的情况下覆盖, 除此以外新建

- 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(出错代码: 340FH)”, 且不进行展开处理。
- 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(出错代码: 3412H)”, 且不进行展开处理。

■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。

设置值	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。

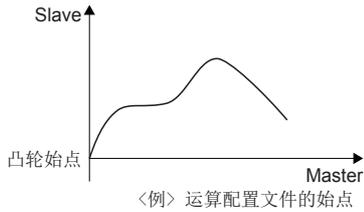
■主轴绝对坐标(MasterAbsolute)

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。)

设置值	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(出错代码: 341DH)”。</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。



设置值	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>时间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>时间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。

设置值	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(出错代码: 3411H)”。</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(出错代码: 3411H)”, 且不更改展开区域。</p>

要点

没有执行中的FB的情况下, 与启动模式(ExecutionMode)的设置无关将立即执行。

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

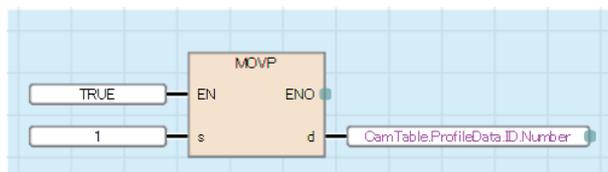
配置文件ID	标签名	数据类型	注释
1	ProfileData0001	MC_CAM_REF	凸轮数据1

■使用的标签

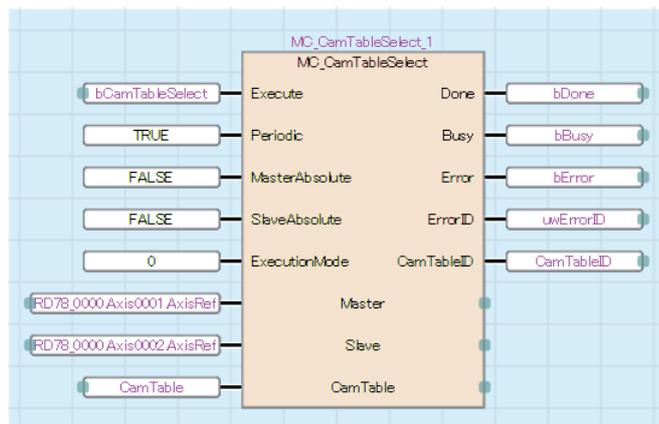
标签名	数据类型	注释
MC_CamTableSelect_1	MC_CamTableSelect	凸轮表选择FB
CamTable	MC_CAM_REF	凸轮表
bCamTableSelect	位	凸轮表选择指令
bDone	位	执行完成
bBusy	位	执行中
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码
CamTableID	MC_CAM_ID	凸轮表ID

■FBD/LD的程序(CPU模块侧)

- 运算配置文件(凸轮数据)设置



- 凸轮表选择



■ST的程序(运动模块侧)

//-----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  
);
```

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>

名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
1周期当前值更改	48	8	子程序型	随时执行型

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(Y0)”ON→OFF之前有效。
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。 ☞ 151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。 ☞ 152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) • 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。 ☞ 153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。 ☞ 156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	由于出错及多重启动等，MCv_ChangeCycle(1周期当前值更改)的执行中断时，将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD(UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 MELSEC iQ-R运动模块用户手册(应用篇)
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	—

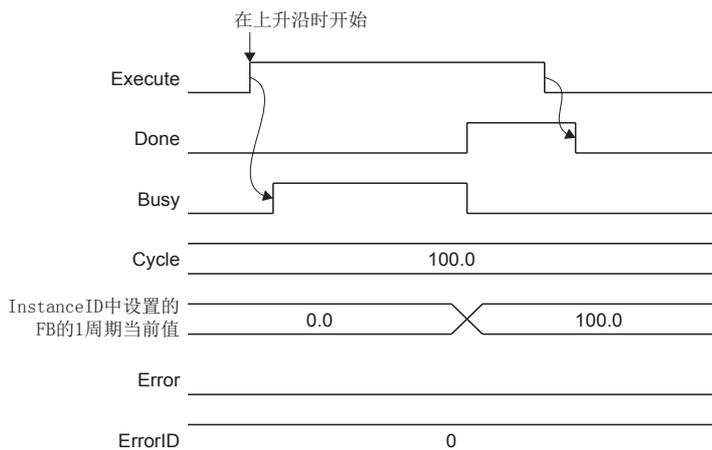
- 对于更改的1周期当前值的设置，根据相对选择(Relative)，设置“绝对位置(FALSE)”或“相对距离(TRUE)”。

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)。

■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。

☞ 11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(出错代码: 3497H(Cycle超出范围(详细代码: 1)))”。

1周期当前值(Cycle)	相对选择(Relative)的设置值	1周期当前值(Cycle)的设置范围
Cycle[0]	FALSE(绝对位置)	$0.0 \leq \text{设置值} < 1\text{周期当前值}$
	TRUE(相对距离)	$-(1\text{周期长}) / 2 \leq \text{设置值} \leq (1\text{周期长}) / 2$
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。

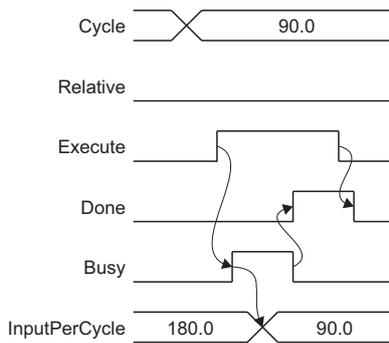
相对选择 (Relative) 的设置值	内容
FALSE (绝对位置)	将1周期当前值 (InputPerCycle)* ¹ 更改为将当前的1周期当前值 (InputPerCycle)* ¹ 与1周期当前值 (Cycle) 相加后的值。
TRUE (相对距离)	将1周期当前值 (InputPerCycle)* ¹ 更改为1周期当前值 (Cycle) 的值。

*1 运算配置文件控制FB (MC_CamIn等) 的公开变量

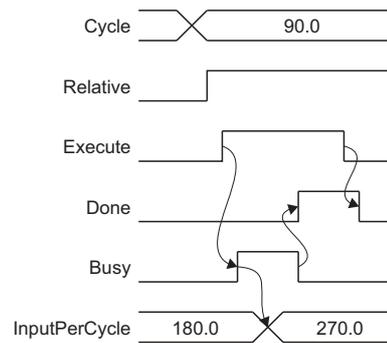
例

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) 的设置值	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

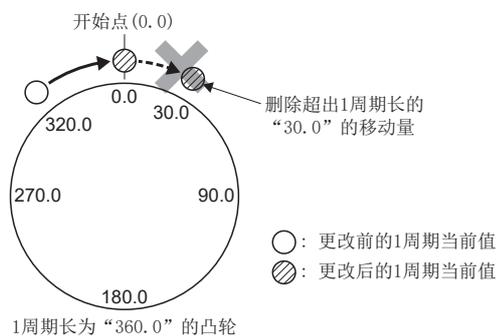
*1 根据MC_CamTableSelect (凸轮表选择) 中设置的重复动作 (Periodic) 的设置值，过峰时的动作有所不同。

重复动作 (Periodic) 的设置值	内容
FALSE (单发动作)	删除超出1周期长的移动量。
TRUE (重复动作)	超出1周期长后更改1周期当前值。

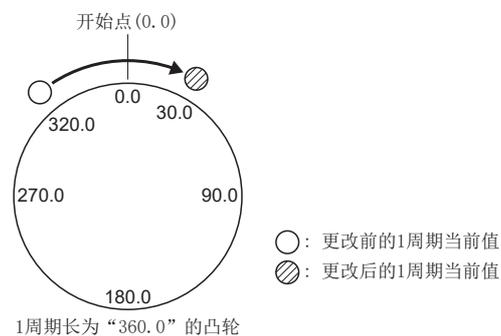
例

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。

设置值	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> 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>*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>

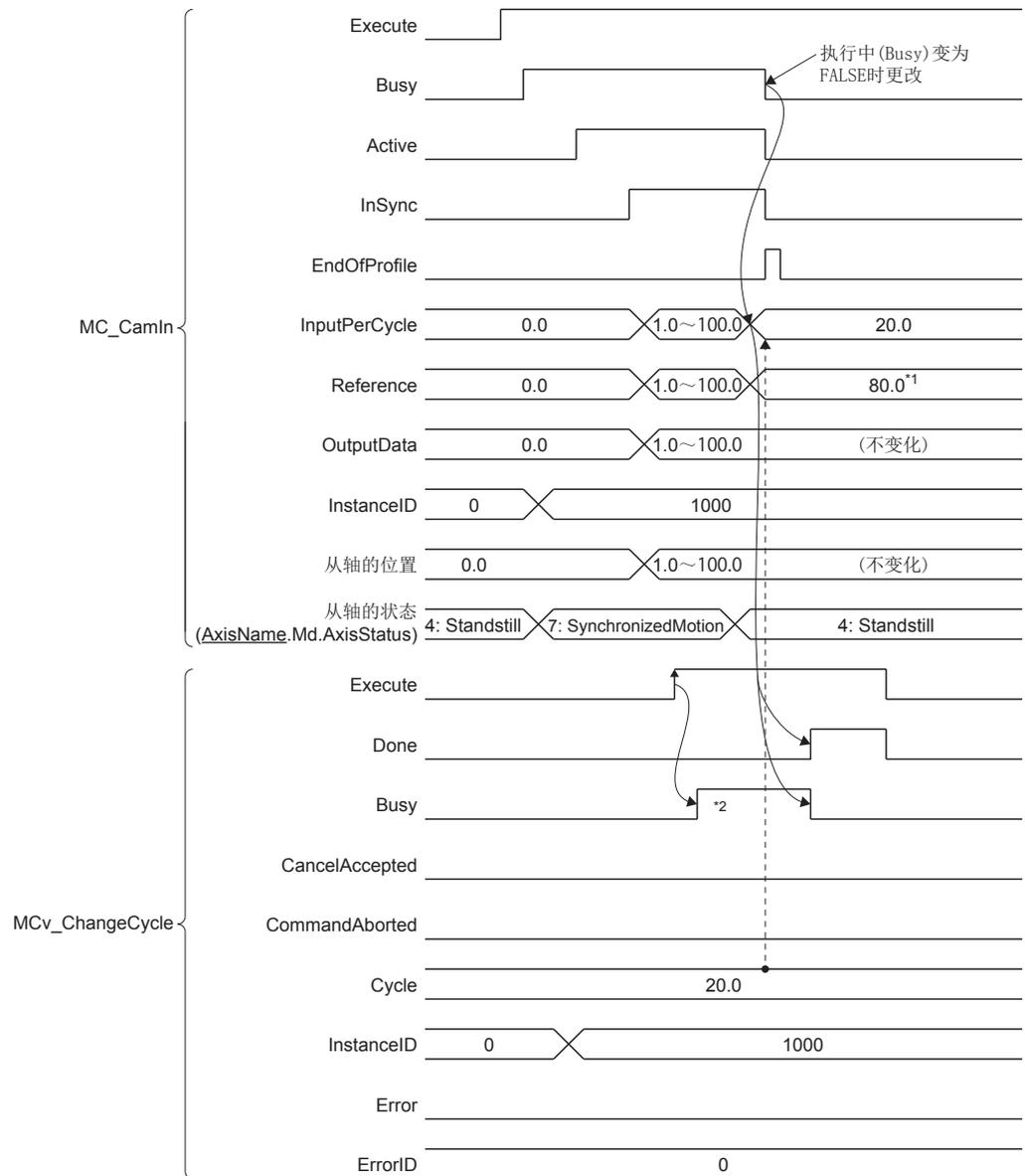
设置值

内容

1: 等待完成后执行 (mcQueued)

通过启动 (Execute) 的上升沿检测执行中 (Busy) 将变为TRUE。
 在运算配置文件控制FB (MC_CamIn等) 的执行中 (Busy) 从TRUE变为FALSE之前将待机并执行。
 <例>

在1周期当前值 (Cycle) 中设置 “20.0”，执行了1周期当前值更改的情况下



*1: 1周期结束时的更新 (100.0) 及通过1周期当前值更改的更新 (-20.0) 将会发生。

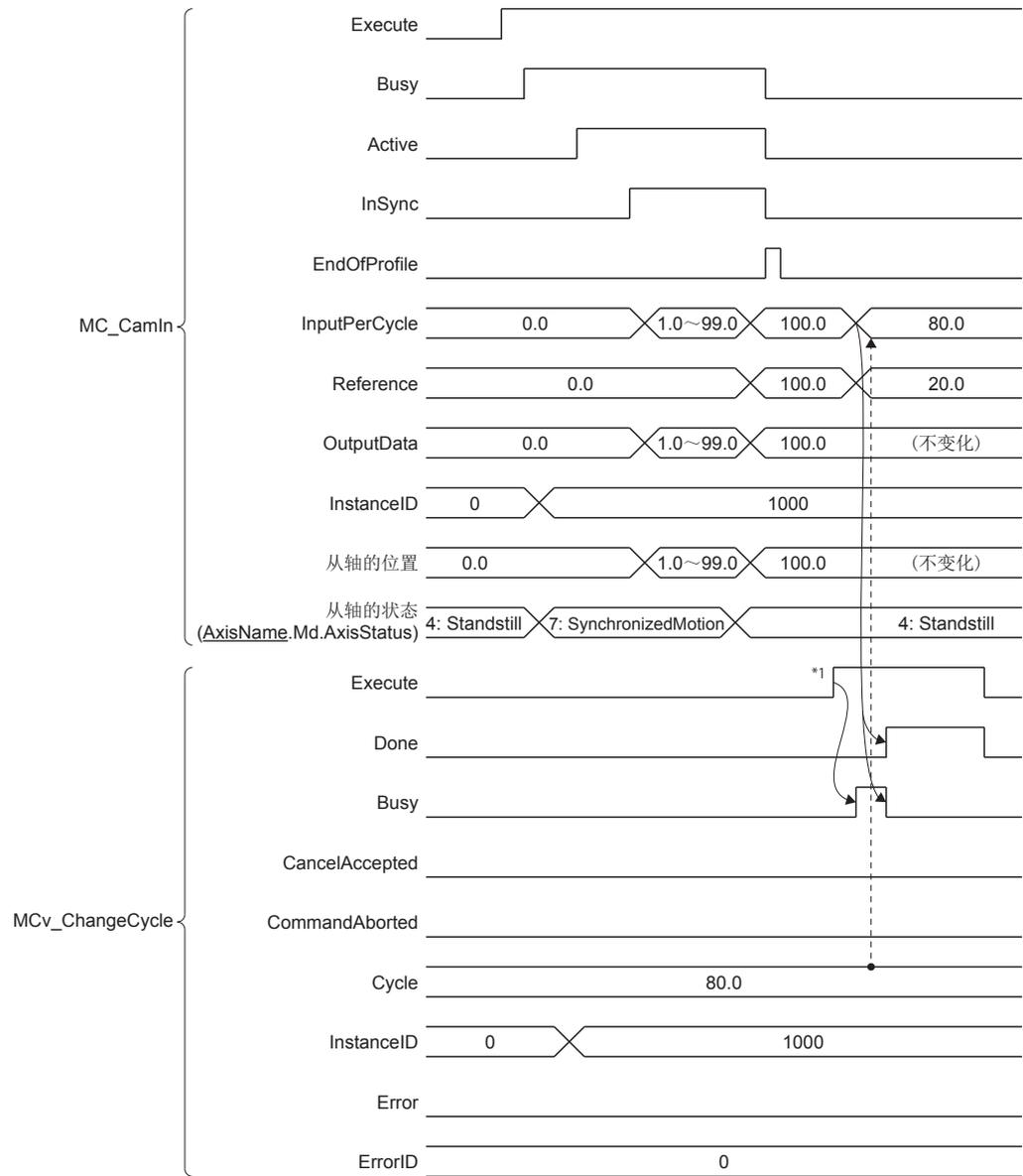
*2: 在MC_CamIn (凸轮动作开始) 的执行中 (Busy) 变为FALSE之前MCv_ChangeCycle (1周期当前值更改) 的执行中 (Busy) 将为TRUE并待机。

设置值

内容

3: 推测执行
(mcSpeculatively)

在运算配置文件控制FB(MC_CamIn等)的执行中(Busy)为FALSE的情况下执行。
 <例>
 在1周期当前值(Cycle)中设置“80.0”，执行了1周期当前值更改的情况下

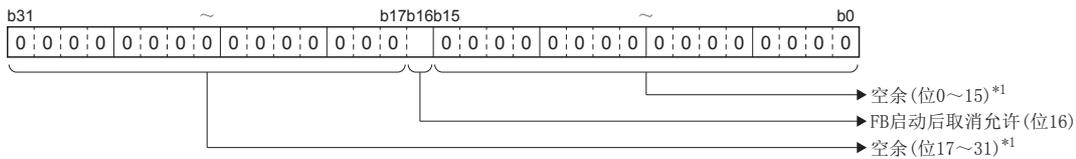


*1: 在MC_CamIn(凸轮动作开始)的执行中(Busy)为FALSE中将MCv_ChangeCycle(1周期当前值更改)的启动(Execute)置为TRUE。

■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。



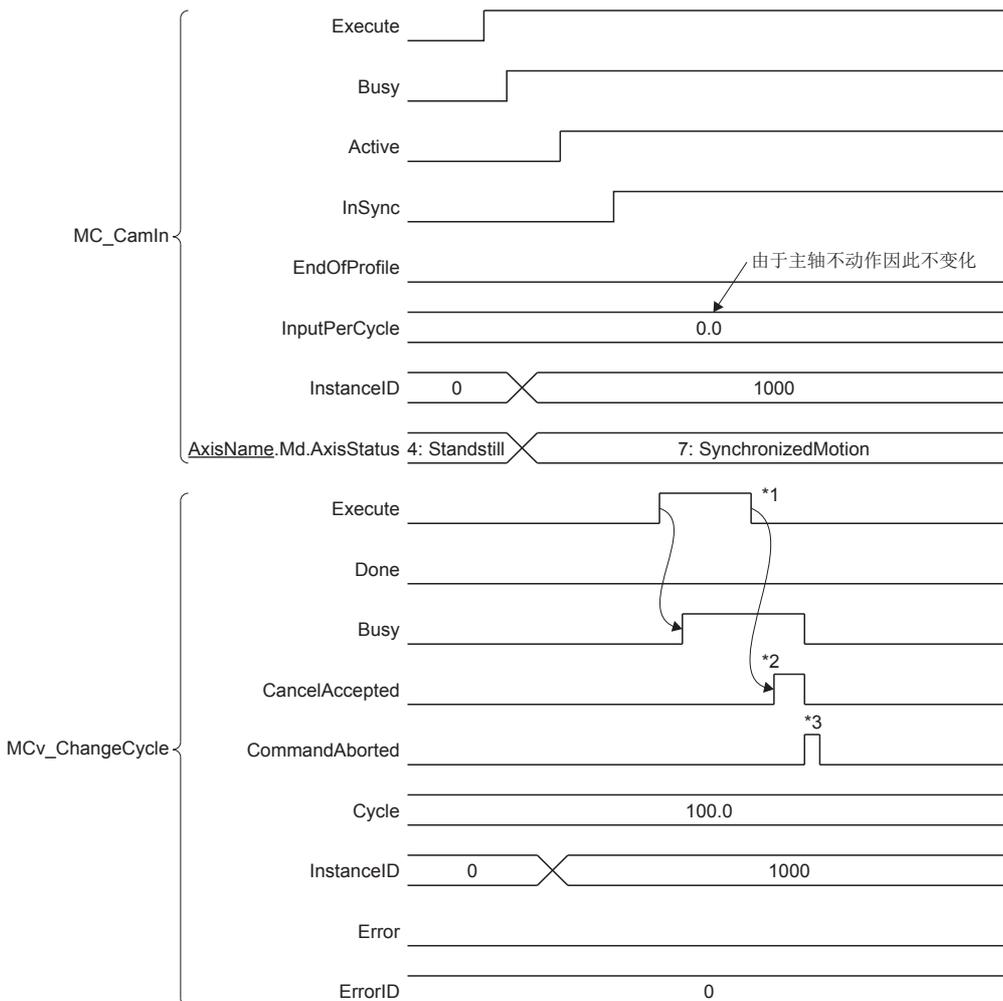
*1 空余中应设置“0”。设置了“0”以外的情况下，将变为“超出Options范围(出错代码: 1A4EH)”。

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)。

例

将启动模式(ExecutionMode)以“1: 等待完成后执行(mcQueued)”启动后，执行中(Busy)从TRUE至FALSE的切换待机中执行了取消的情况下



- *1 FB启动后取消允许(选项(Options): 位16)为“1: 允许”的情况下。“0: 不允许”的情况下，将变为取消无效。
- *2 虽然取消受理时变为TRUE，但是在取消完成或1周期当前值更改完成时将变为FALSE。
- *3 虽然取消完成时变为TRUE，但是由于启动(Execute)为FALSE因此1周期后将变为FALSE。

程序示例

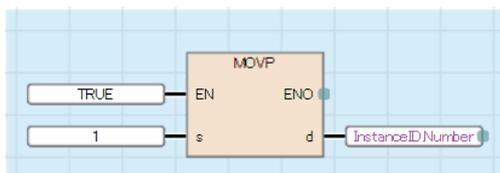
将1周期当前值更改指令 (bChangeCycle) 置为TRUE，并将实例ID (InstanceID) 为“1”的1周期当前值更改为“90.0”的程序示例如下所示。

■使用的标签

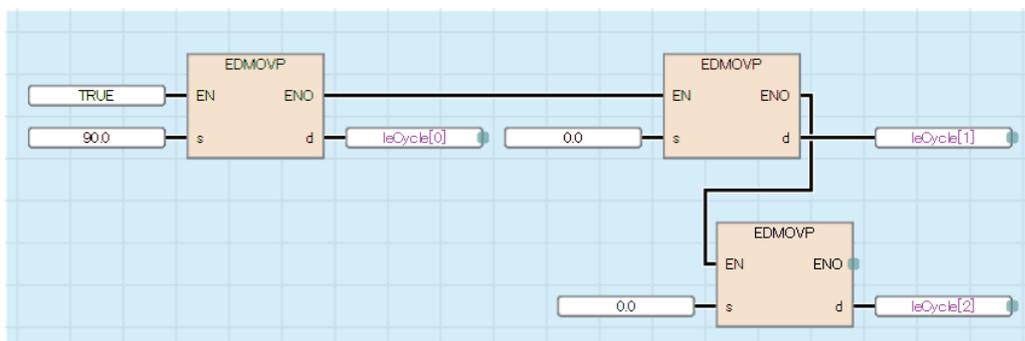
标签名	数据类型	注释
MCv_ChangeCycle_1	MCv_ChangeCycle	1周期当前值更改FB
bChangeCycle	位	1周期当前值更改指令
InstanceID	INSTANCE_ID	实例ID
leCycle	双精度实数(0..2)	1周期当前值
bDone	位	执行完成
bBusy	位	执行中
bCommandAborted	位	执行中断
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码
bCancelAccepted	位	取消受理

■FBD/LD的程序(CPU模块侧)

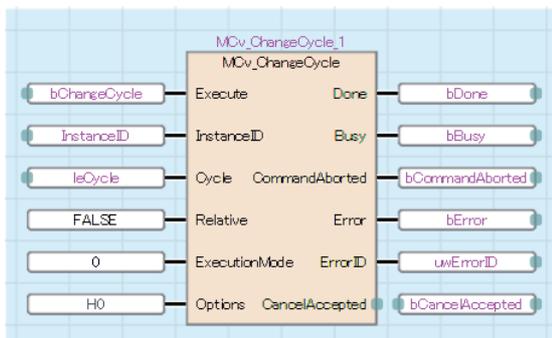
- 运算配置文件设置



- 1周期当前值更改用数据设置



- 1周期当前值更改



■ST的程序(运动模块侧)

//-----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 ,
```

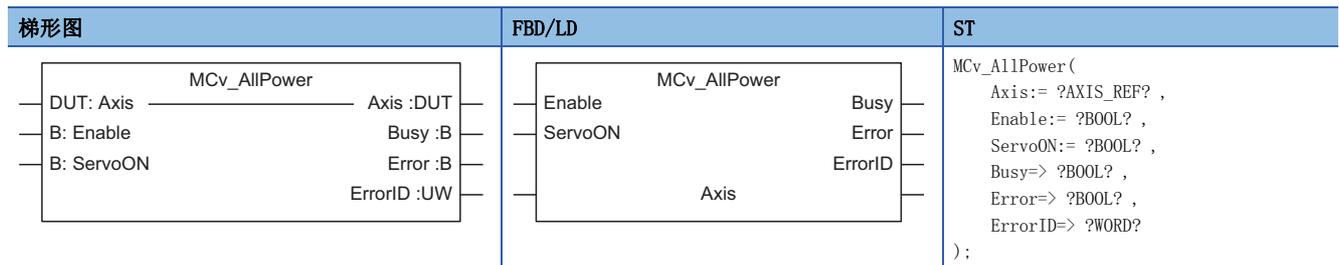
```
    ErrorID=> uwErrorID ,
```

```
    CancelAccepted=> bCancelAccepted
```

```
);
```

MCv_AllPower

将所有轴切换为允许运行状态。



名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
所有轴允许运行	10	4	子程序型	随时执行型

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	在运动模块侧使用MCv_AllPower(所有轴允许运行)的情况下, 可以省略。设置将被忽略。 在CPU模块侧使用MCv_AllPower(所有轴允许运行)的情况下, 设置输入输出No. (StartIO)。轴No. (AxisNo)将被忽略。 ☞ 23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	出错	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时, 将返回出错代码。 关于出错代码的详细内容, 请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册(应用篇)

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。
- 在运动模块侧使用的情况下，轴信息 (Axis) 的设置将被忽略。在CPU模块侧使用的情况下，在轴信息 (Axis) 的输入输出 No. (StartIO) 中设置输入输出编号。关于输入输出编号的指定，请参阅下述章节。

☞ 17页 输入输出编号的指定

- 在MCv_AllPower (所有轴允许运行) 内发生了异常的情况下，出错 (Error) 将变为TRUE，并将出错代码存储到出错代码 (ErrorID) 中。关于出错代码的详细内容，请参阅下述手册。

📖 MELSEC iQ-R运动模块用户手册 (应用篇)

- 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
	FALSE	伺服OFF	5: Switched On
FALSE	TRUE	伺服OFF	3: Switch On Disabled
	FALSE	伺服OFF	3: Switch On Disabled

- 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的规格。
- 驱动器模块出错发生中，MCv_AllPower (所有轴允许运行) 将发送到驱动器中，因此无需再次将有效 (Enable) 及伺服ON请求 (ServoON) 从FALSE置为TRUE。

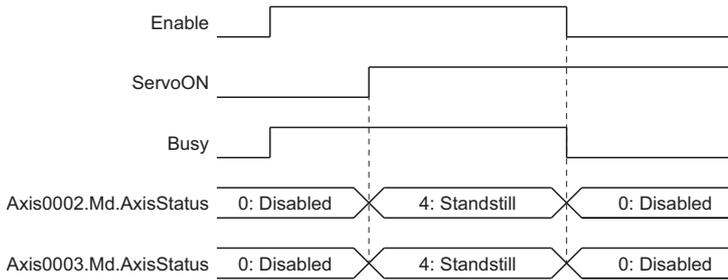
要点

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

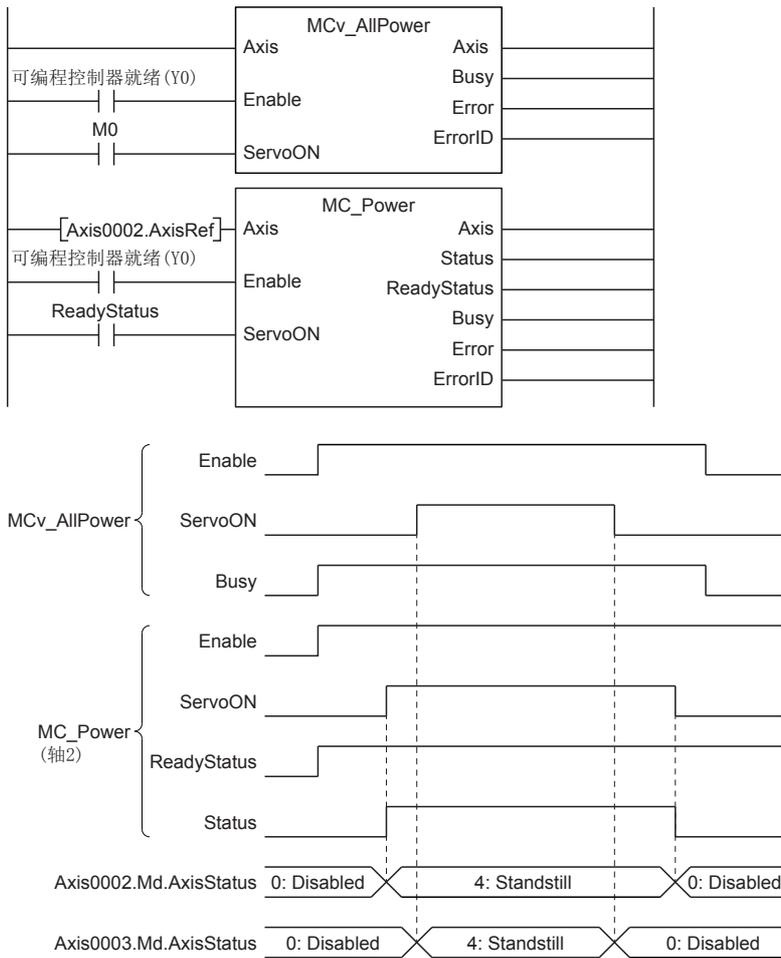
例

将轴2 (Axis0002)、轴3 (Axis0003) 设置为实驱动轴的情况下



例

对轴2 (Axis0002) 同时使用了MC_Power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。

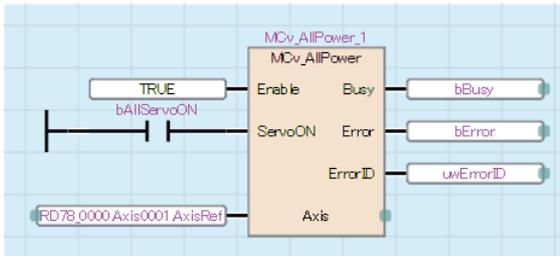
程序示例

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	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序

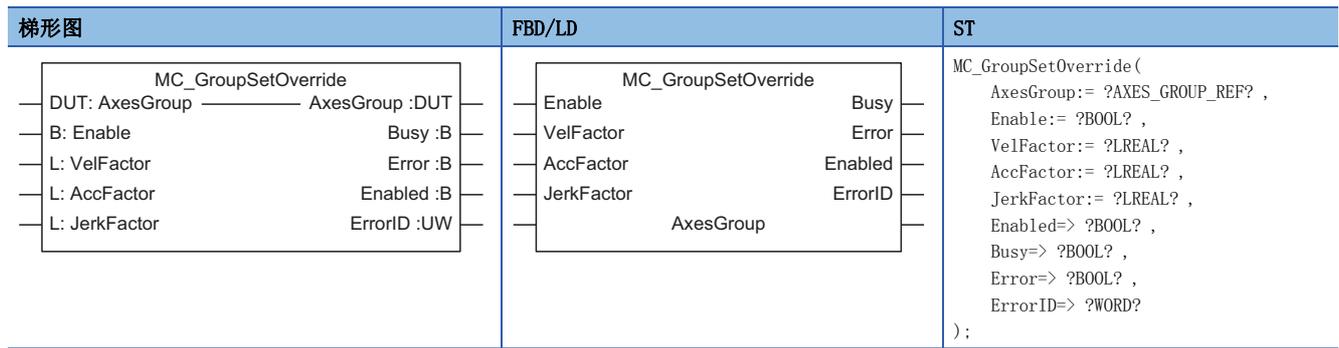


■ST的程序(运动模块侧)

```
MCv_AllPower_1(  
    Enable:= TRUE ,  
    ServoON:= bAllServoON ,  
    Busy=> bBusy ,  
    Error=> bError ,  
    ErrorID=> uwErrorID  
);
```

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。



名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
轴组超驰值设置	40	6	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 44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(轴组超驰值设置)。
VelFactor	速度超驰系数	LREAL	始终	0.00~10.00	0.00	设置速度的超驰系数。 有效(Enable)为TRUE时，将始终获取。
AccFactor	加速度超驰系数	LREAL	始终	0.00、0.01~10.00	0.00	设置加速度的超驰系数。 有效(Enable)为TRUE时，将始终获取。 设置了“0.00”的情况下，不进行加速度超驰系数的更改，维持上次值进行控制。
JerkFactor	Jerk超驰系数	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超驰系数(JerkFactor)范围(出错代码: 349EH)”。

■输出变量

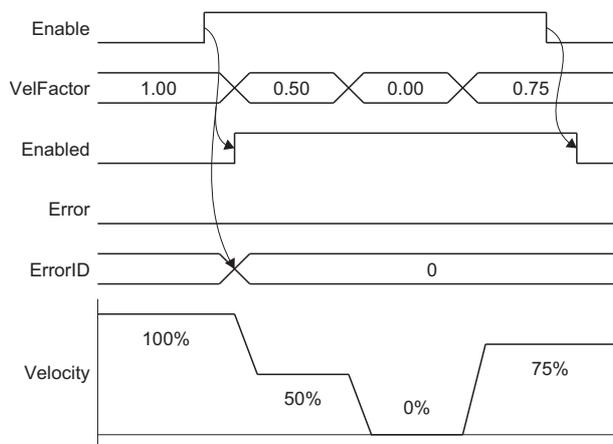
输出变量	名称	数据类型	初始值	内容
Enabled	有效中	BOOL	FALSE	超驰值正常设置的情况下，将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_GroupSetOverride(轴组超驰值设置)时，将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册(应用篇)

功能

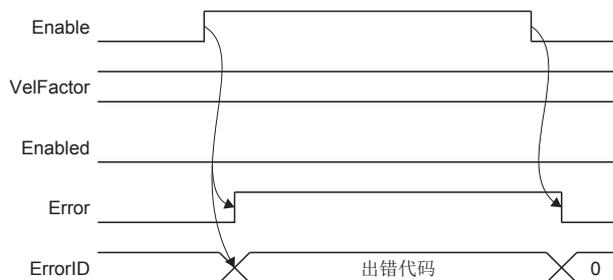
- 更改指定的轴组的目标速度、目标加速度、目标减速度。
- 更改为当前动作中的目标速度、目标加速度、目标减速度乘以超驰系数后的值。
- 有效(Enable)变为TRUE时将执行MC_GroupSetOverride(轴组超驰值设置)。超驰系数为有效中时，有效中(Enabled)将变为TRUE。
- 有效(Enable)为TRUE时，如果更改超驰系数的值，则将反映新的超驰系数。
- 在MC_GroupSetOverride(轴组超驰值设置)内发生了异常的情况下，出错(Error)将变为TRUE，并将出错代码存储到出错代码(ErrorID)中。关于出错代码的详细内容，请参阅下述手册。
📖 MELSEC iQ-R运动模块用户手册(应用篇)
- 在速度超驰系数(VelFactor)中设置“0.00”的值时，在轴状态(AxisName.Md.AxisStatus)不切换为“4: 待机中(Standstill)”的状况下停止轴。
- 在加速度超驰系数(AccFactor)中设置“0.00”的值时，不更改加速度超驰系数，维持上次的加速度超驰系数。

■时序图

- 正常完成的情况下

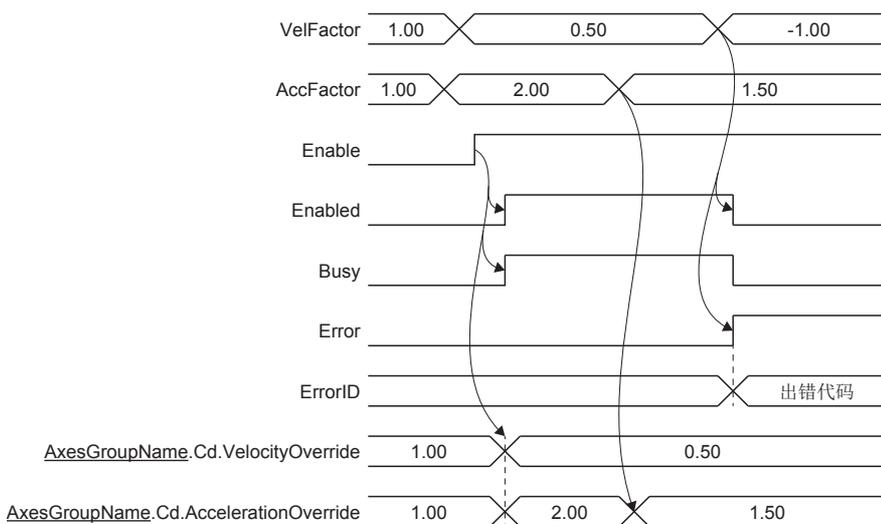


- 异常完成的情况下



■动作概要

- 在MC_GroupSetOverride(轴组超驰值设置)中, 更改速度超驰系数(AxesGroupName.Cd.VelocityOverride)、加速度超驰系数(AxesGroupName.Cd.AccelerationOverride)。



- 在超驰系数中设置了超出范围的值的情况下, MC_GroupSetOverride(轴组超驰值设置)将发生出错并停止以后的获取。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(轴组超驰值设置)。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超驰系数有影响。
- 请勿在MC_GroupSetOverride(轴组超驰值设置)执行中进行速度超驰系数(AxesGroupName.Cd.VelocityOverride)、加速度超驰系数(AxesGroupName.Cd.AccelerationOverride)的直接操作。
- 速度超驰后的速度超出范围的情况下, 请参阅下述手册的“速度范围”。
 📖MELSEC iQ-R运动模块用户手册(应用篇)
- 加速度超驰后的加速度、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”。
 📖MELSEC iQ-R运动模块用户手册(应用篇)

■程序示例

将轴组超驰值更改指令(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超驰系数“1.0”、加减速度超驰系数“2.0”后的值的程序示例如下所示。

■轴组

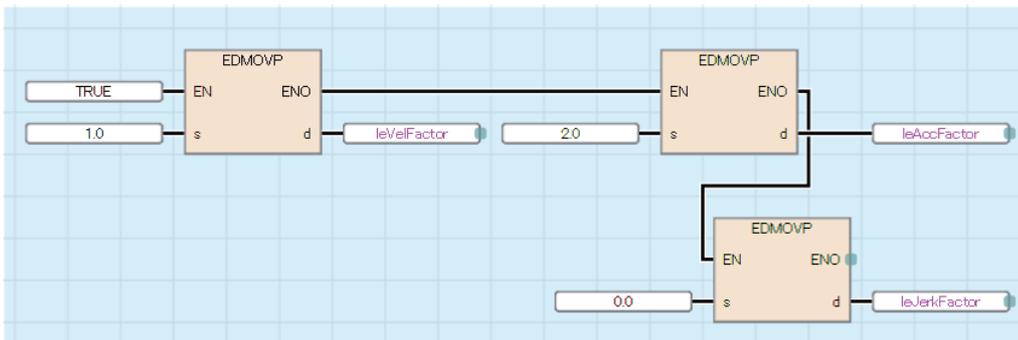
轴组No.	标签名	数据类型	注释
1	AxesGroup001	AXES_GROUP_REF	轴组1

■使用的标签

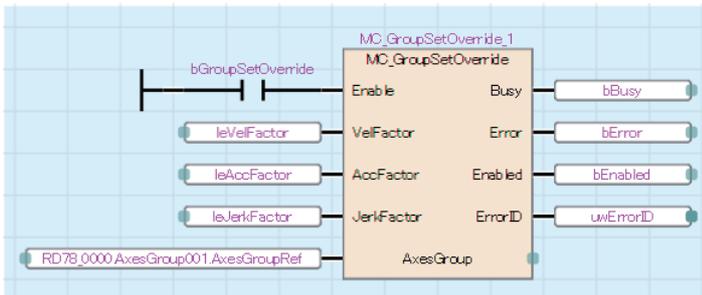
标签名	数据类型	注释
MC_GroupSetOverride_1	MC_GroupSetOverride	轴组超驰值设置FB
bGroupSetOverride	位	轴组超驰值更改指令
leVelFactor	双精度实数	速度超驰系数
leAccFactor	双精度实数	加减速超驰系数
leJerkFactor	双精度实数	Jerk超驰系数
bEnabled	位	有效中
bBusy	位	执行中
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

- 轴组超驰值更改用数据设置



- 轴组超驰值更改



■ST的程序(运动模块侧)

//-----轴组超驰值更改用数据设置-----

leVelFactor:= 1.00;

leAccFactor:= 2.00;

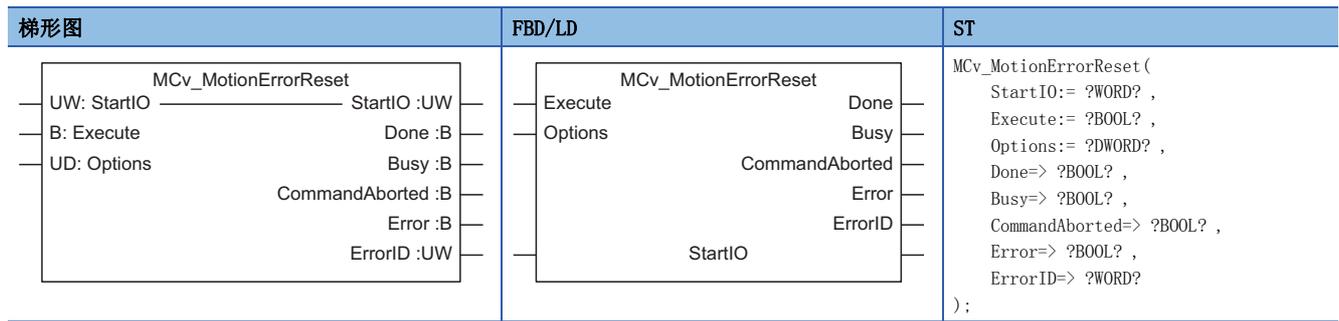
leJerkFactor:= 0.0;

//-----轴组超驰值更改-----

```
MC_GroupSetOverride_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Enable:= bGroupSetOverride ,
    VelFactor:= leVelFactor ,
    AccFactor:= leAccFactor ,
    JerkFactor:= leJerkFactor ,
    Enabled=> bEnabled ,
    Busy=> bBusy ,
    Error=> bError ,
    ErrorID=> uwErrorID
);
```

MCv_MotionErrorReset

对运动系统的所有的出错、警告进行复位。



名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
运动出错复位	8	6	子程序型	随时执行型

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)。在运动模块侧使用MCv_MotionErrorReset (运动出错复位) 的情况下, 可以省略。设置值将被忽略。在CPU模块侧使用MCv_MotionErrorReset (运动出错复位) 的情况下不能省略。

■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MCv_MotionErrorReset (运动出错复位)。
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 (出错代码: 1A4EH)”。

■输出变量

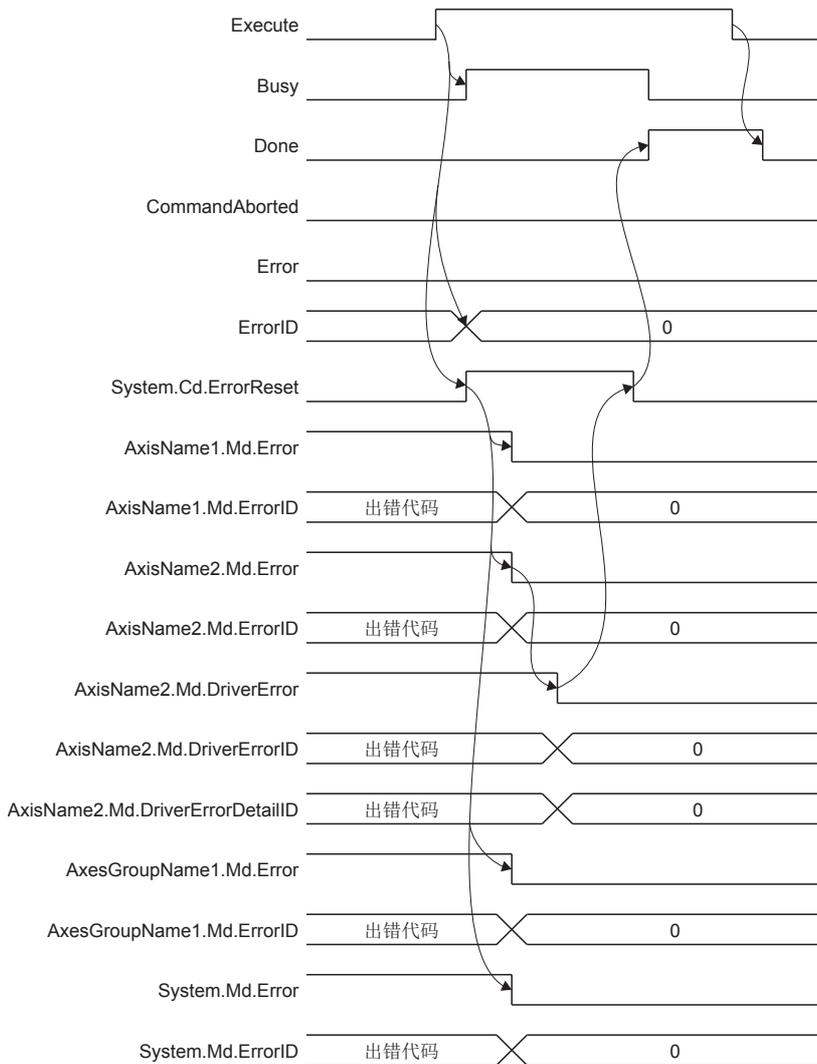
输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	复位完成时, 将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_MotionErrorReset (运动出错复位) 时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	由于超时, MCv_MotionErrorReset (运动出错复位) 中断时将变为TRUE。 执行指令 (Execute) 变为FALSE时, 将变为FALSE。
Error	出错	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时, 将返回出错代码。 关于出错代码的详细内容, 请参阅下述手册。 MELSEC iQ-R运动模块用户手册 (应用篇)

功能

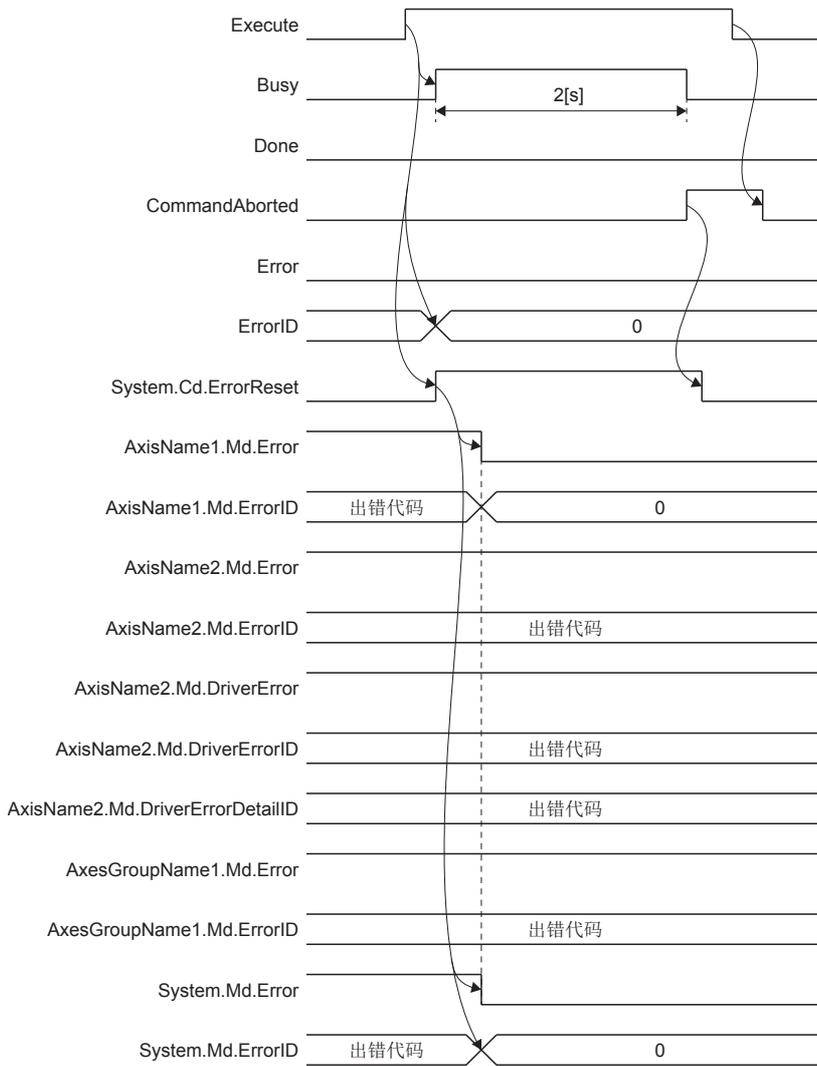
- 在执行指令(Execute)为TRUE时执行MCv_MotionErrorReset (运动出错复位)，开始处理时执行中(Busy)将变为TRUE，并将系统出错复位(System.Cd.ErrorReset)置为TRUE。
- 运动系统的出错、警告的解除完成时执行中(Busy)将变为FALSE，执行完成(Done)将变为TRUE。
- 即使在残留了出错、警告的原因的状态下将执行指令(Execute)置为TRUE，也不解除出错、警告。在此情况下，从FB执行后2秒以内未解除出错时执行中断(CommandAborted)将变为TRUE，并将系统出错复位(System.Cd.ErrorReset)置为FALSE。应将执行指令(Execute)置为FALSE一次，消除了出错、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

- 请勿在MCv_MotionErrorReset (运动出错复位) 的执行中进行系统出错复位 (System. Cd. ErrorReset) 的直接操作。
- 从CPU模块进行出错复位的情况下，应不进行系统出错复位 (System. Cd. ErrorReset) 的操作，而使用MCv_MotionErrorReset (运动出错复位)。

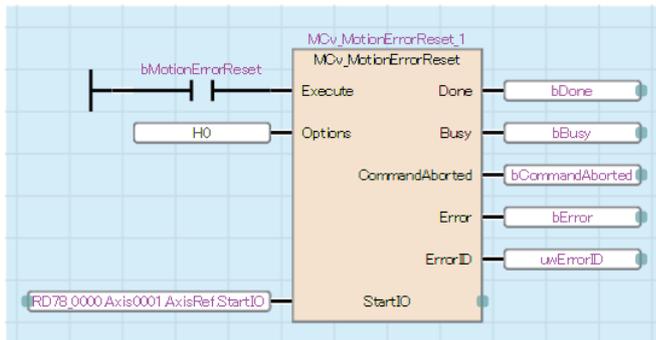
程序示例

将出错复位(bMotionErrorReset)置为TRUE，并对运动系统的所有出错、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	运动出错复位FB
bMotionErrorReset	位	出错复位
bDone	位	执行完成
bBusy	位	执行中
bCommandAborted	位	执行中断
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)



■ST的程序(运动模块侧)

```
MCv_MotionErrorReset_1(  
    Execute:= bMotionErrorReset ,  
    Options:= H00000000 ,  
    Done=> bDone ,  
    Busy=> bBusy ,  
    CommandAborted=> bCommandAborted ,  
    Error=> bError ,  
    ErrorID=> uwErrorID  
);
```

3.2 动作系统的FB

MC_Home

进行指定的轴的原点复位。

梯形图	FBD/LD	ST
		<pre> MC_Home (Axis:= ?AXIS_REF? , Execute:= ?BOOL? , Position:= ?LREAL? , AbsSwitch:= ?MC_INPUT_REF? , Options:= ?DWORD? , Done=> ?BOOL? , Busy=> ?BOOL? , Active=> ?BOOL? , CommandAborted=> ?BOOL? , Error=> ?BOOL? , ErrorID=> ?WORD?); </pre>

名称	输入区域点数(byte)	输出区域点数(byte)	编译方式	FB动作
原点复位	188	8	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23页 AxisName.AxisRef. (轴信息)

■输入变量

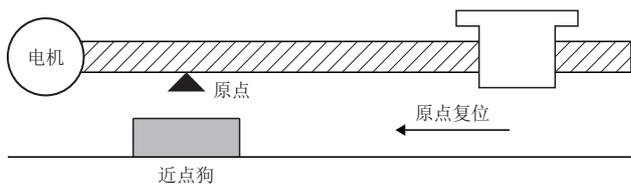
输入变量	名称	数据类型	获取	有效范围	初始值	内容
Execute	执行指令	BOOL	启动时	TRUE、FALSE	FALSE	TRUE时执行MC_Home(原点复位)。
Position	目标位置	LREAL	启动时	-10000000000.0 ~ 10000000000.0	0.0	设置原点地址。 应在下述范围内进行设置。 • -10000000000.0 ≤ 设置值 < 10000000000.0 *: 环形计数器有效的情况下，将变为环形计数器范围。
AbsSwitch	原点开关	MC_INPUT_REF	启动时	—	—	设置驱动器式原点复位中传输至从设备的近点狗信号。 关于详细内容，请参阅下述章节。 ☞ 174页 原点开关(AbsSwitch)
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(出错代码: 1A4EH)”。

■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)的执行中断时，将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册(应用篇)

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”。
根据以下条件，将变为零点复位控制启动时的零点复位方式。

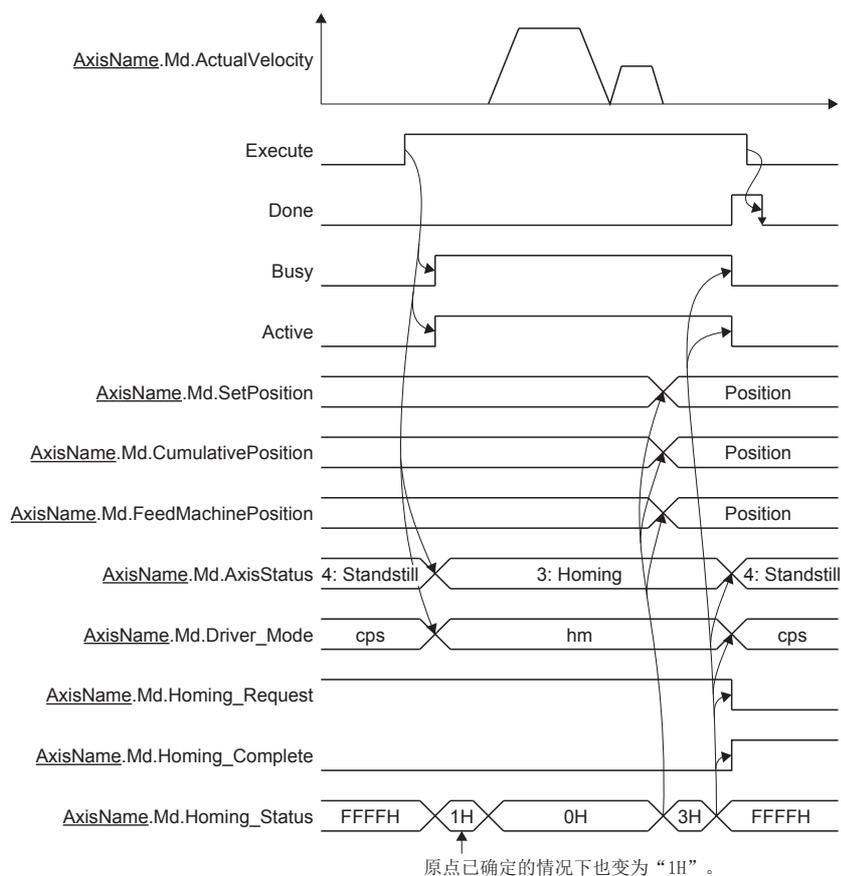
零点复位方式	零点复位方式的条件
驱动器式零点复位	满足下述所有条件的情况下，将变为驱动器式零点复位。 <ul style="list-style-type: none"> 轴类型为实驱动轴时。 驱动器支持Homing模式时。 将“Home offset(607CH)”设置为从对象时。
数据集式零点复位	除上述以外的情况下，将变为数据集式零点复位。

驱动器式零点复位

将驱动器切换为Homing模式，根据驱动器侧中设置的定位模式进行零点复位。

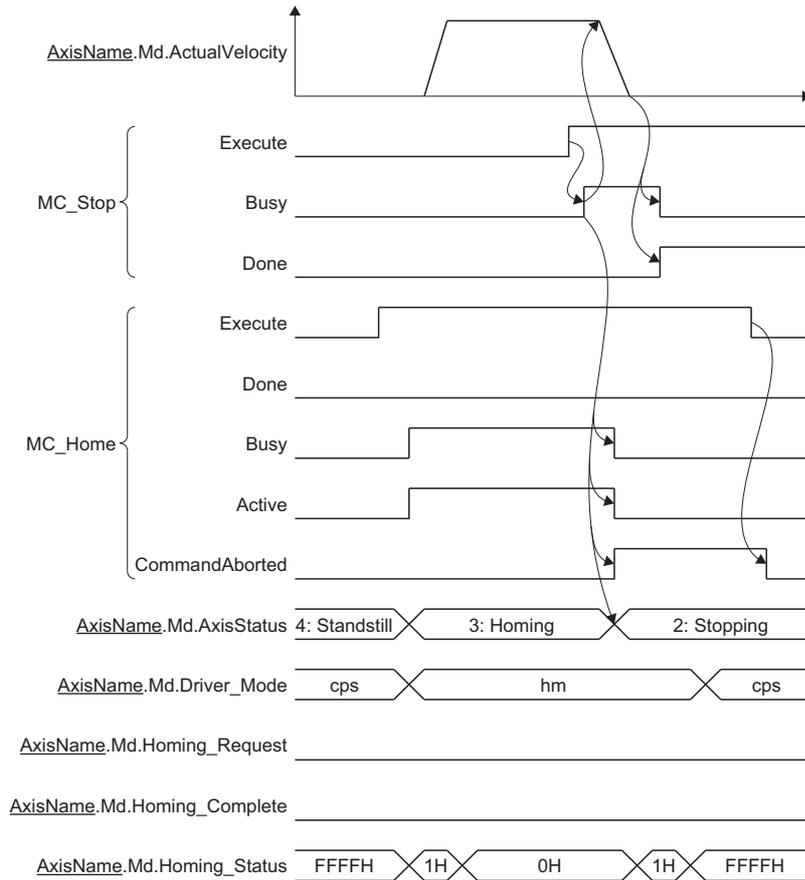
更改零点复位方式及各种参数的情况下，应通过MC_WriteParameter(参数写入)更改驱动器的零点复位数据。零点复位的动作及可设置的参数取决于驱动器的规格。关于驱动器的详细内容，请参阅驱动器的手册。

- 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。
☞ 11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

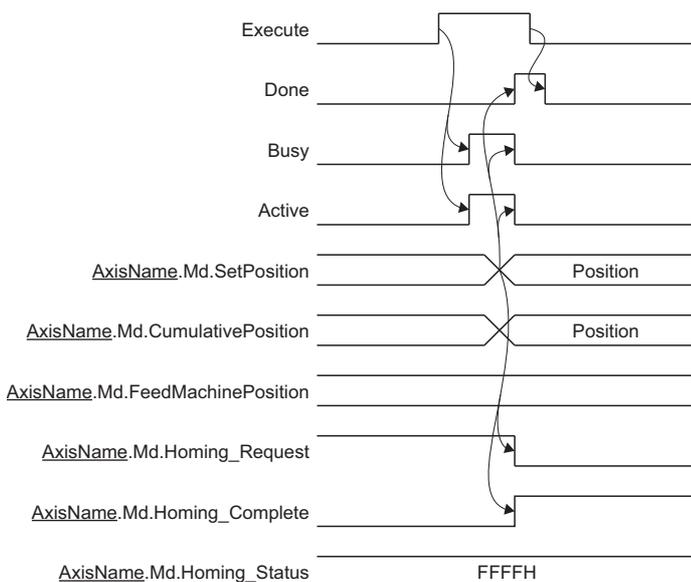


- 原点复位时将MC_Stop(强制停止)置为了TRUE的情况下, 将向驱动器发送“HALT”信号。使用不支持HALT的驱动器的情况下, 由于不通过MC_Stop(强制停止)停止, 因此应使用紧急停止。
- 原点复位时的停止处理按照驱动器的规格。因此, 将忽略MC_Stop(强制停止)的减速度(Deceleration)及Jerk。

■数据集式原点复位

对虚拟轴及从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)作为原点并登录到运动系统中, 并将指令当前位置(AxisName.Md.SetPosition)及累计当前位置(AxisName.Md.CumulativePosition)改写为目标位置(Position)。

• 时序图



■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。对于原点复位请求 (AxisName.Md.Homing_Request)，下述情况下将变为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。
- 接通了驱动器的电源时。^{*1*2}
- 在驱动器侧检测出“绝对位置丢失”时。^{*2}
- 更改了从对象的“Polarity(607EH)(b7: position polarity)”时。^{*2}
- 更改了驱动器侧的电子齿轮时。^{*2}
- 检测出驱动器及电机编码器的更改时。
- 使用模拟功能时变为了上次原点确定时的连接驱动器以外时。
- 在刻度测量编码器中检测出绝对位置丢失时。^{*3}
- 更改了刻度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 仅实驱动轴。

*3 仅实编码器轴(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从设备的近点狗信号。

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。

 60页 MC_INPUT_REF

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  61页 TARGET_REF(输入信号)	■类型 <ul style="list-style-type: none"> • BOOL ■数据类型 <ul style="list-style-type: none"> • [OBJ]*1 • [VAR] • [DEV] • [CONST]
	Detection(信号检测方法)	INT (MC_SIGNAL_LOGIC)  75页 MC_SIGNAL_LOGIC	<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(滤波器时间)	LREAL	0.0~5.0[s]

*1 无法使用外部信号高精度输入。设置了外部信号高精度输入用的信号的情况下，将以与通常的信号相同的精度执行动作。

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。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，建议使用驱动器内置的DI。
- 驱动器式原点复位中，不进行指令当前位置的跟踪。
- 请勿将原点复位中的轴设置为主轴进行同步控制。由于不连续进行指令当前值、反馈值的更新，因此从轴中有可能变为“控制中速度范围溢出(出错代码：1AE8H)”。
- 未对下述对象进行映射的情况下，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。

📖 MELSEC iQ-R运动模块用户手册(应用篇)

- 原点复位中，在运动模块中进行硬件行程限位信号的检测，检测出时则将“HALT”信号发送到驱动器中。使用驱动器侧的限位开关信号进行停止的情况下，应将硬件行程限位超驰(AxisName.Cd.HwStrokeLimit_Override)置为“DISABLE(检查无效)”，并将运动模块侧的硬件行程限位检查暂时置为无效。
- 原点复位启动时，在硬件行程限位超驰(AxisName.Cd.HwStrokeLimit_Override)中设置了“ONLY_INSIDE(仅对范围内方向检查无效)”的情况下，将变为“不可启动(出错代码：1AADH)”，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，并通过原点开关“[DEV](BOOL)G11478000.1”，进行轴1(Axis0001)的原点复位的程序示例如下所示。

轴

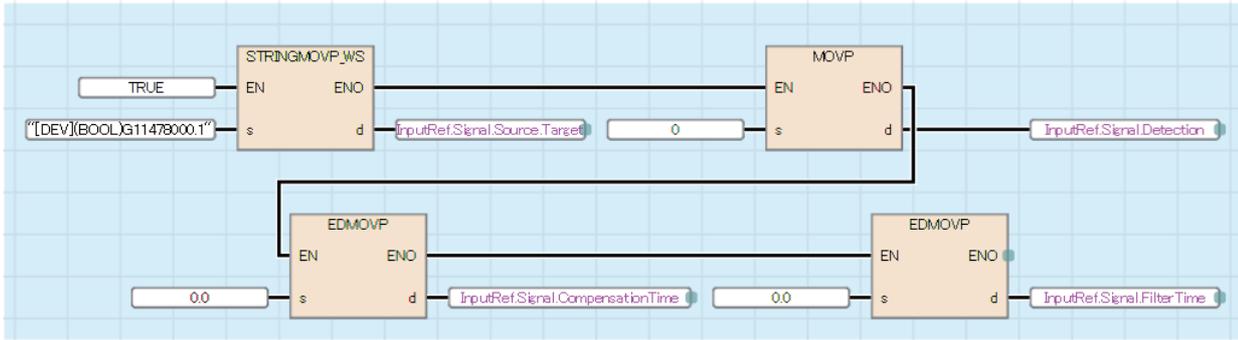
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

使用的标签

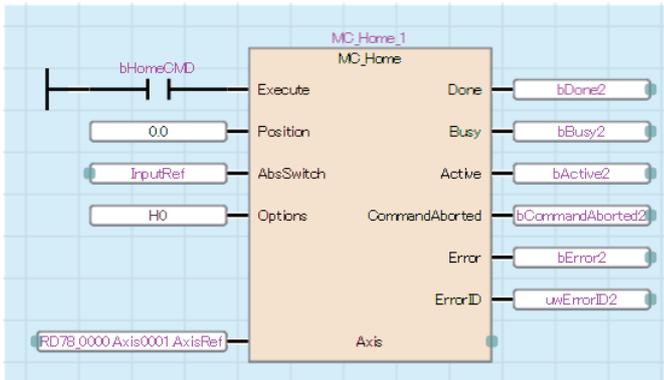
标签名	数据类型	注释
MC_Home_1	MC_Home	原点复位FB
bHomingCMD	位	原点复位指令
InputRef	MC_INPUT_REF	原点开关
bDone	位	执行完成
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

- 原点开关数据设置



- 原点复位



■ST的程序(运动模块侧)

//-----原点开关数据设置-----

```
InputRef.Signal.Source.Target:= "[DEV](BOOL)G11478000.1";
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
);
```

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>

名称	输入区域点数(byte)	输出区域点数(byte)	编译方式	FB动作
强制停止	36	8	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23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。
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(出错代码: 1A13H)”。
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(出错代码: 1A4EH)”。

■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	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD(UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册(应用篇)

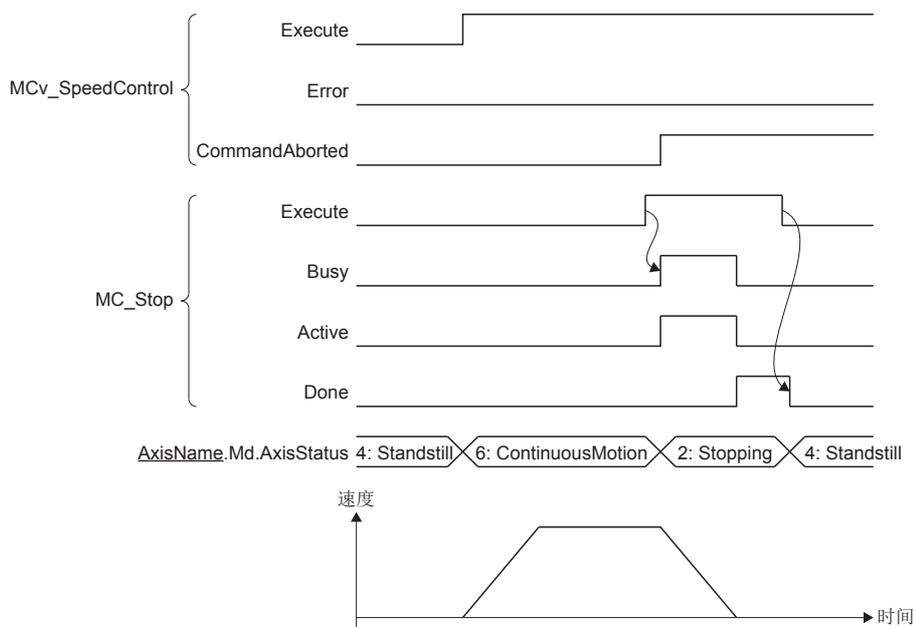
功能

- 对于MC_Stop(强制停止), 设置减速度(Deceleration), 并对当前执行中的FB进行减速停止。
- 执行MC_Stop(强制停止)时, 对于当前执行中的FB, 执行中断(CommandAborted)将变为TRUE, 且轴状态(AxisName.Md.AxisStatus)将转变为“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时将转变为“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

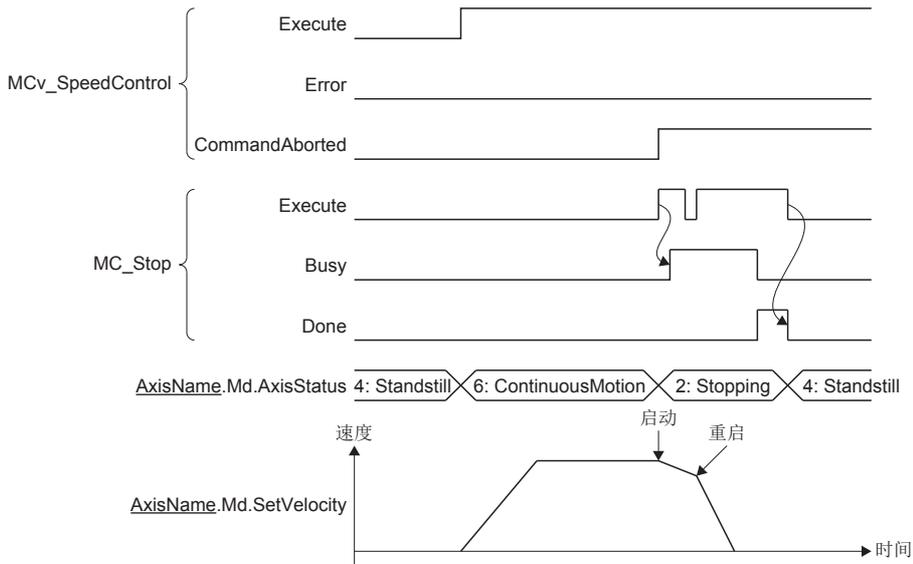
关于异常完成时的时序图, 请参阅下述章节。

☞ 11页 通过执行指令(Execute)类型的运动控制FB的基本动作

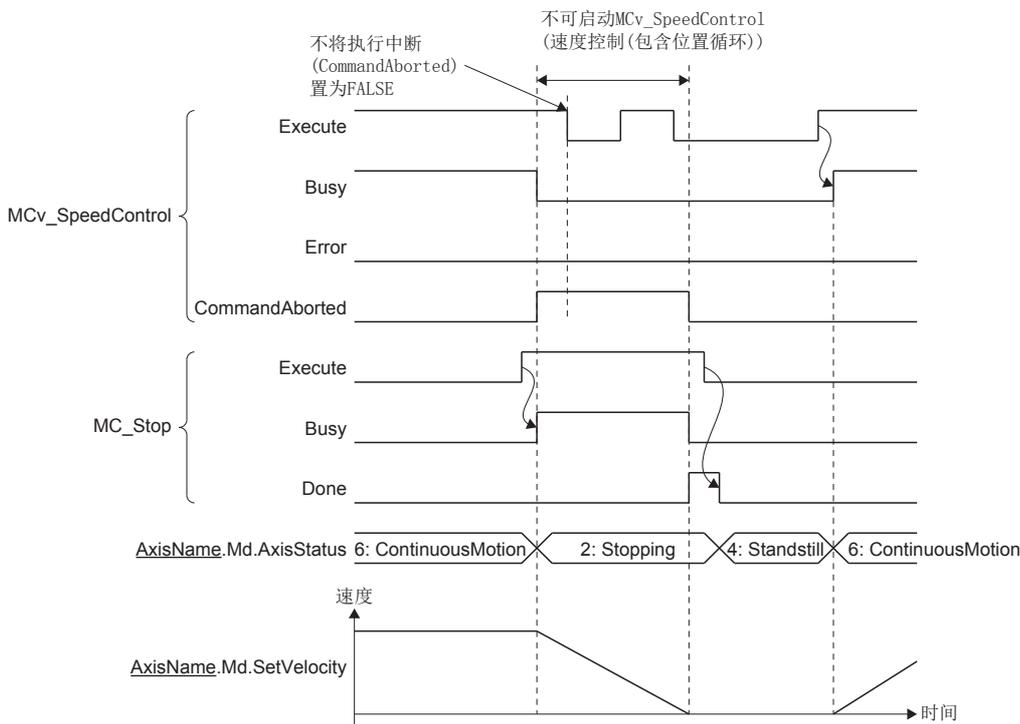
■动作概要

- 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)的设置值重启了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)为“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(出错代码: 1AADH)”,且MC_Stop(强制停止)的出错(Error)将变为TRUE,轴状态(AxisName.Md.AxisStatus)将变为“1: 出错停止中(ErrorStop)”。
- 轴停止后,通过执行轴出错复位(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

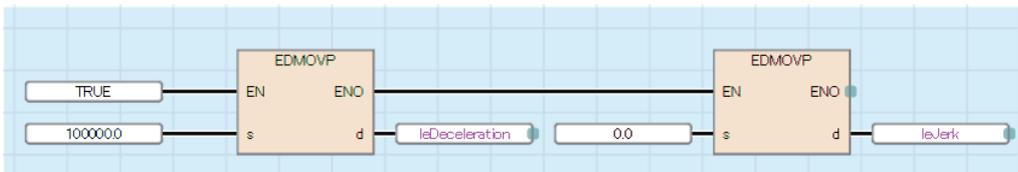
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

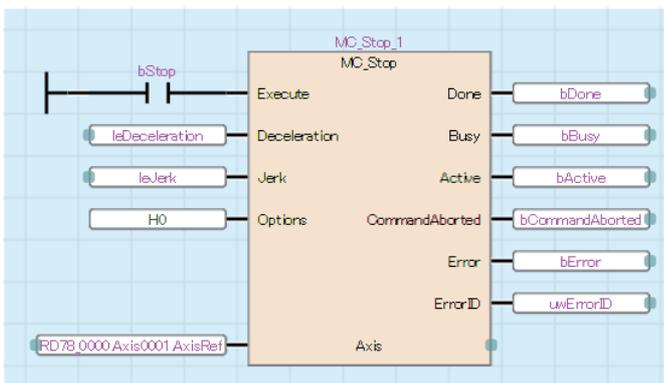
标签名	数据类型	注释
MC_Stop_1	MC_Stop	轴停止FB
bStop	位	轴停止请求
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bDone	位	完成
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

- 轴停止用数据设置



- 轴停止



■ST的程序(运动模块侧)

```
//-----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
);
```

MC_GroupStop

对指定的轴组进行减速停止。

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>

名称	输入区域点数(byte)	输出区域点数(byte)	编译方式	FB动作
组强制停止	36	8	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 44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。
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(出错代码: 1A13H)”。
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(出错代码: 1A4EH)”。

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	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册(应用篇)

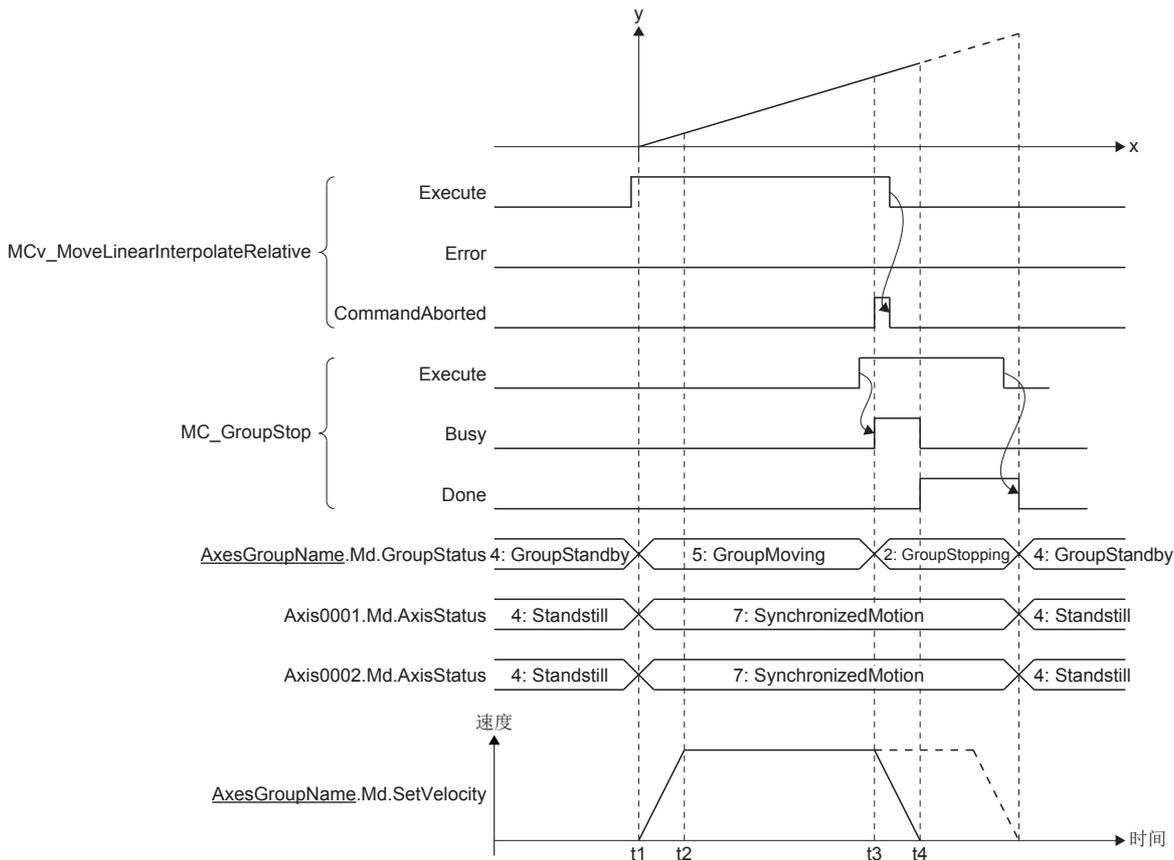
功能

- 对于MC_GroupStop(组强制停止)，设置减速度(Deceleration)，并对当前执行中的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，对于当前执行中的FB，执行中断(CommandAborted)将变为TRUE，且轴组状态(AxesGroupName.Md.GroupStatus)将转变为“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

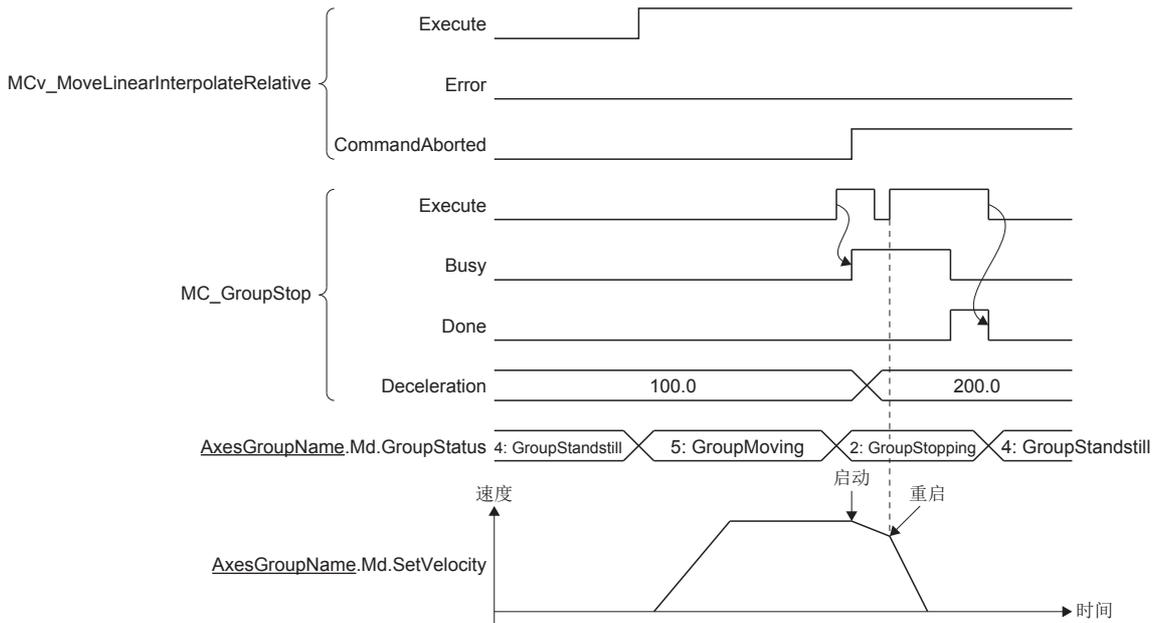
关于异常完成时的时序图，请参阅下述章节。

☞ 11页 通过执行指令(Execute)类型的运动控制FB的基本动作

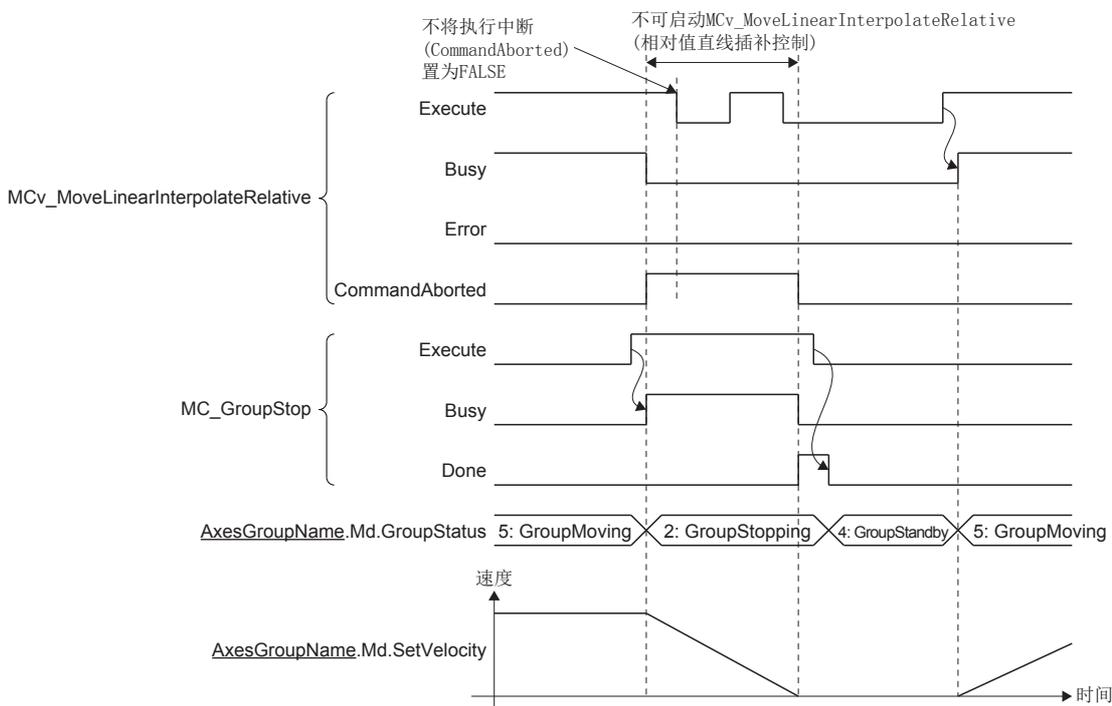
■动作概要

- 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)的设置值重启了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)、执行中的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(出错代码: 1AADH)”, 且 `MC_GroupStop` (组强制停止) 的出错(Error) 将变为 TRUE, 轴组状态 (`AxesGroupName.Md.GroupStatus`) 将变为 “1: 出错停止中(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

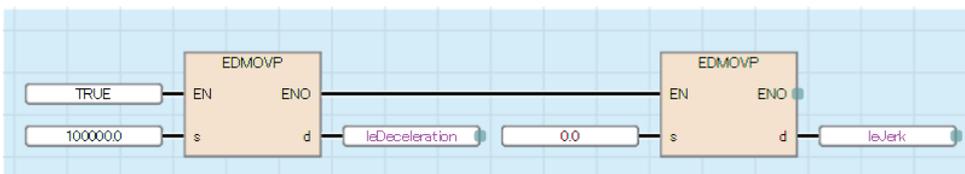
轴组No.	标签名	数据类型	注释
1	AxesGroup001	AXES_GROUP_REF	轴组1

■使用的标签

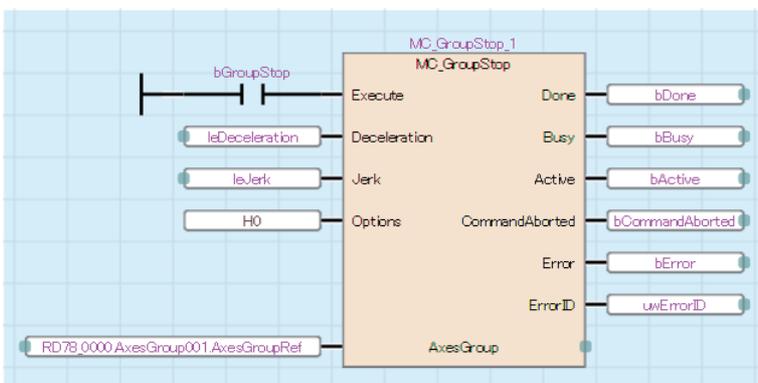
标签名	数据类型	注释
MC_GroupStop_1	MC_GroupStop	轴组停止FB
bGroupStop	位	轴组停止请求
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bDone	位	完成
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

- 轴组停止用数据设置



- 轴组停止

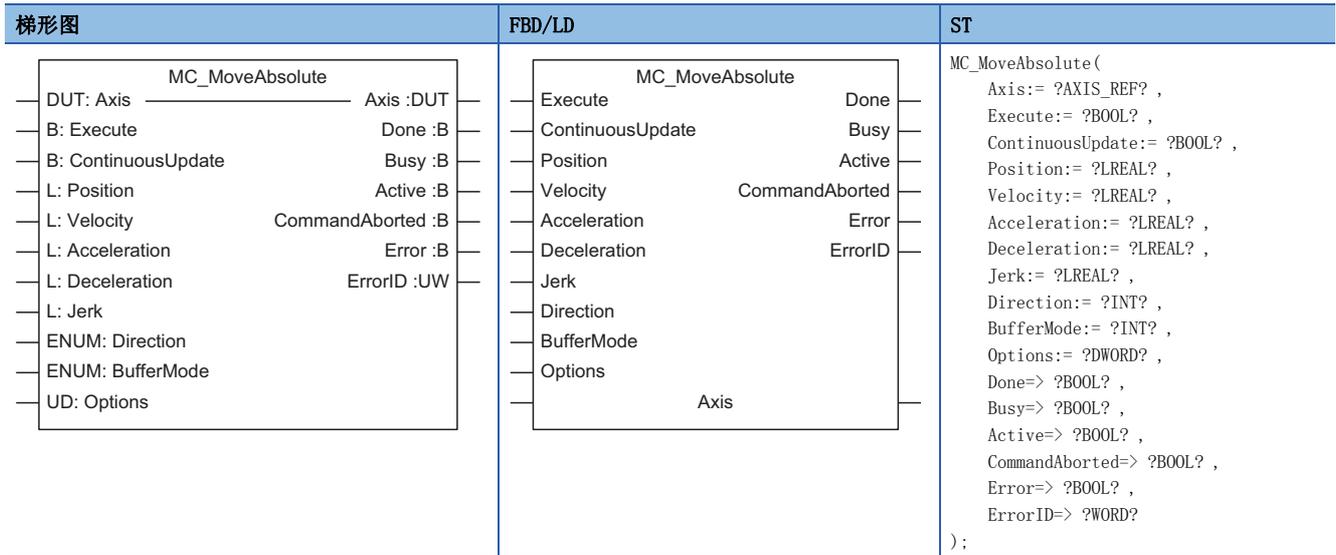


■ST的程序(运动模块侧)

```
//-----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  
);
```

MC_MoveAbsolute

设置绝对位置的目标位置，执行定位。



名称	输入区域点数(byte)	输出区域点数(byte)	编译方式	FB动作
绝对值定位	64	8	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23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。 ☞ 188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。 ☞ 189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。 ☞ 189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。 ☞ 189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。 ☞ 189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(出错代码: 1A37H)”。 关于详细内容，请参阅下述章节。 190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。 193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。 194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	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 MELSEC iQ-R运动模块用户手册(应用篇)

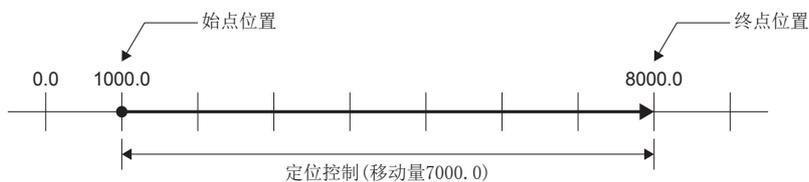
功能

- 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

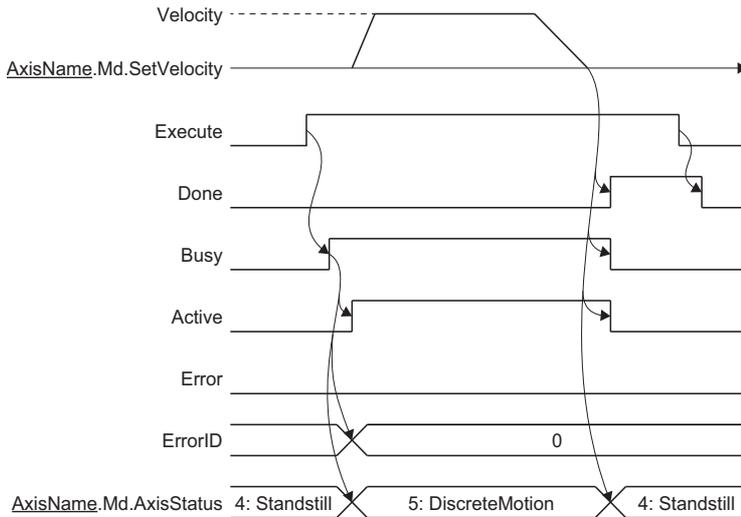
始点位置(当前的停止位置)为“1000.0”时，在目标位置(Position)中设置了“8000.0”的情况下

- 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。

☞ 11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。

设置了超出范围的值的情况下，将变为“超出目标位置范围(出错代码: 1A05H)”且不启动。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)(出错代码: 1A00H)”且不启动。

有效范围

环形计数器下限值 ≤ 目标位置 < 环形计数器上限值

- 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(出错代码: 1A05H)”且不启动。

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。

☞ 190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(出错代码: 1AE4H)”(速度更改时为“超出运算周期换算速度范围警告(警告代码: 0D2FH)”)。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)使运算周期换算后的速度不小于“0.00001”。
- *2 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]为止的正数
1: 加减速时间恒定方式(mcFixedTime)	0.000000 ^{*1} 、0.000001~8400.0[s]为止的正数

- *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]为止的正数
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 ²]为止的正数
1: 加减速时间恒定方式(mcFixedTime)	不使用

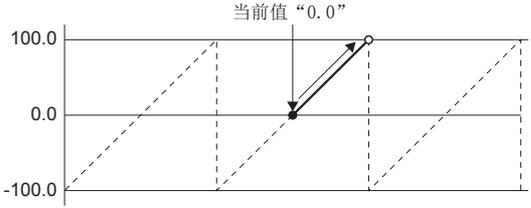
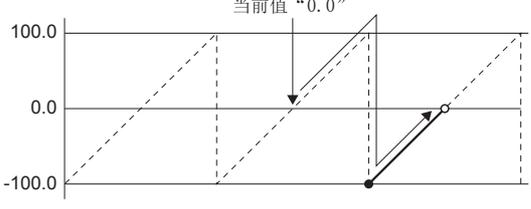
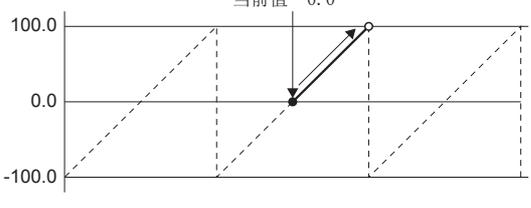
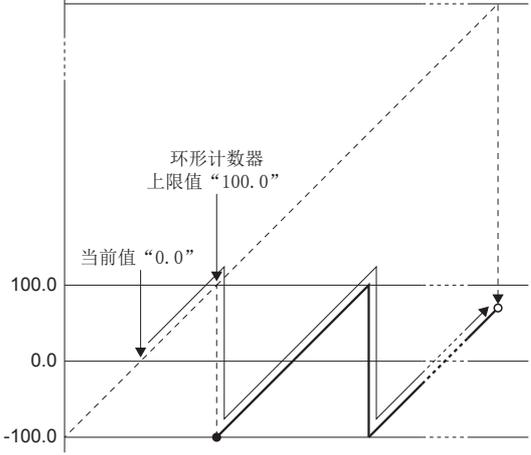
■方向选择 (Direction)

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)

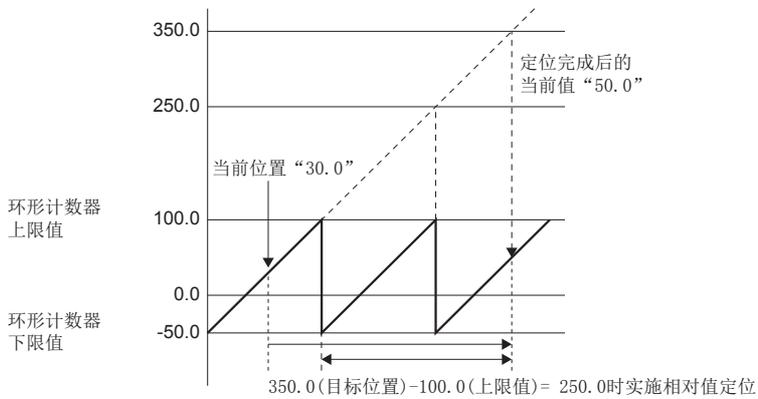
从当前位置向目标位置进行正方向 (地址增加) 的定位。

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
■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”。	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
	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
■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 (出错代码: 1A05H)”且不启动。	• 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	环形计数器上限值  环形计数器下限值
	环形计数器上限值 ≤ 目标位置 < 定位范围上限值	定位范围上限值  环形计数器上限值 环形计数器下限值

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

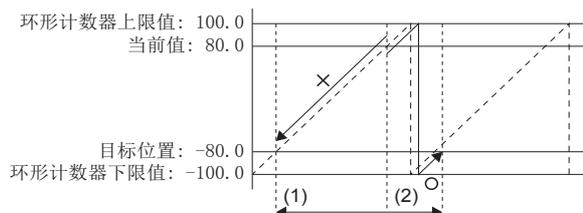
- 3: 最短路径 (mcShortestWay)

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(出错代码: 1A05H)”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 (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。

设置值	内容
0: Aborting(mcAborting)	中断(取消)执行中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在执行中的FB中缓冲下一个FB。 执行中的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) 执行中的FB完成时, 依次执行缓冲FB。
2: BlendingLow(mcBlendingLow)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的FB到达了目标位置时, 将执行中的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)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的FB到达了目标位置时, 将执行中的FB的目标速度作为切换速度。
4: BlendingNext(mcBlendingNext)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的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)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的FB到达了目标位置时, 将执行中的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 在执行中的FB与缓冲FB之间不执行停止。

要点

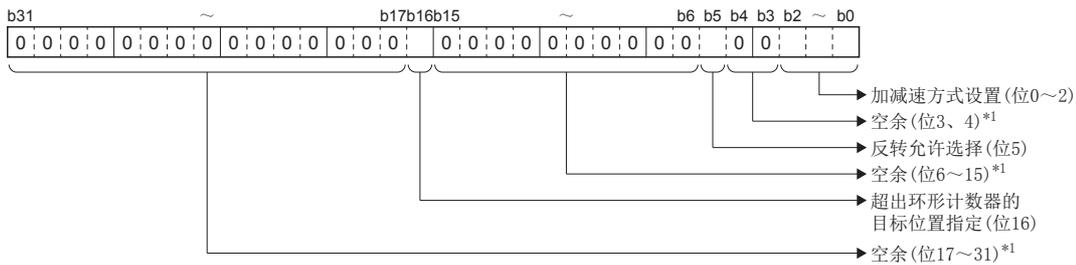
关于多重启动(缓冲模式)的详细内容, 请参阅下述手册。

📖 MELSEC iQ-R运动模块用户手册(应用篇)

■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(出错代码: 1A4EH)”。

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 *: 缓冲模式(BufferMode)的“0: Aborting(mcAborting)”设置时，或目标位置更改时将有效。
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)

设置值	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)

设置值	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“越程出错(出错代码: 1A7EH)”且进行减速停止。

■必要对象

使用MC_MoveAbsolute(绝对值定位)的情况下，应将下述从对象设置为轴。

- Target position(607AH)

未设置从对象的情况下，将变为“必要从对象未设置(出错代码: 1AA8H)”且不启动。

关于从对象设置的详细内容，请参阅下述手册。

📖 MELSEC iQ-R运动模块用户手册(应用篇)

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	设置值
目标位置	80000.0
速度	50000.0
加速度	100000.0
减速度	100000.0
Jerk	0.0

■ 轴

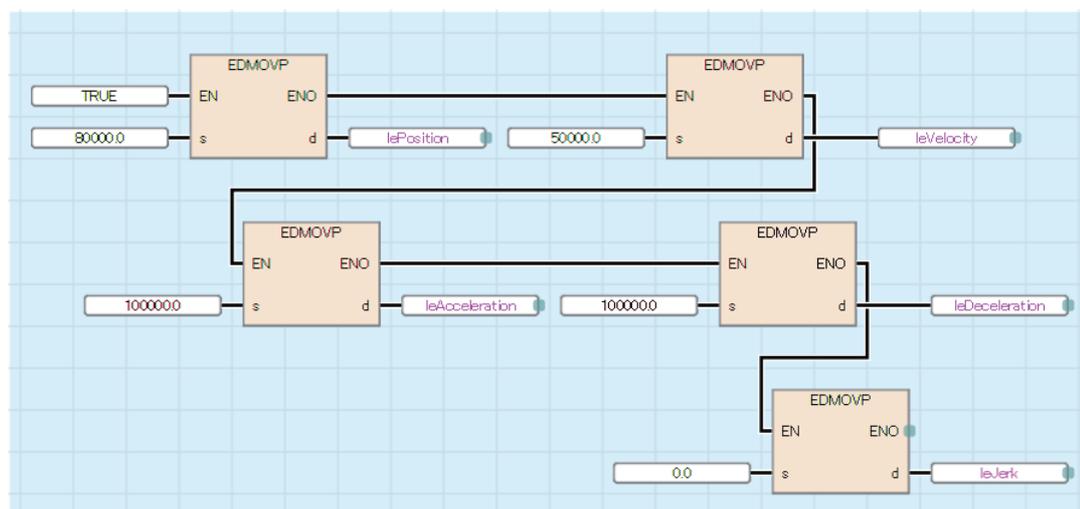
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■ 使用的标签

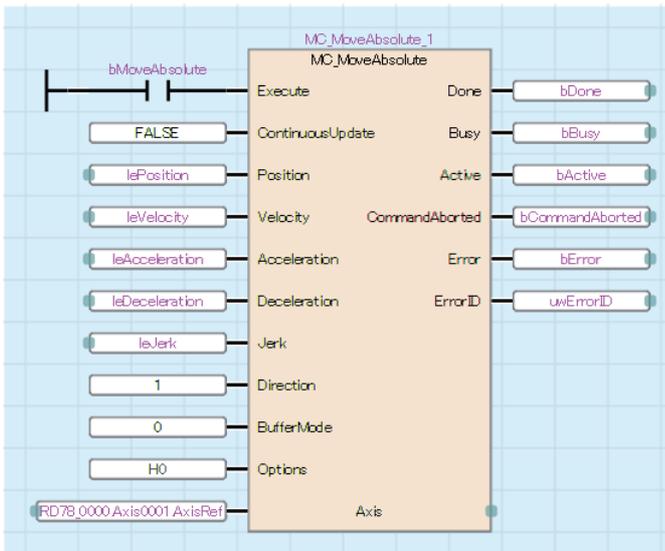
标签名	数据类型	注释
MC_MoveAbsolute_1	MC_MoveAbsolute	绝对值定位FB
bMoveAbsolute	位	单轴绝对值定位启动
lePosition	双精度实数	目标位置
leVelocity	双精度实数	速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bDone	位	执行完成
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■ FBD/LD的程序(CPU模块侧)

• 单轴定位控制用数据设置



• 绝对值定位



■ST的程序(运动模块侧)

//-----单轴定位控制用数据设置-----

```
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
);
```

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>

名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
相对值定位	64	8	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23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。 ☞ 199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。 ☞ 199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。 ☞ 199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。 ☞ 200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。 ☞ 200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。 200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。 201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	出错	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时, 将返回出错代码。 关于出错代码的详细内容, 请参阅下述手册。 MELSEC iQ-R运动模块用户手册 (应用篇)

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)将变为“5: 定位运行中 (DiscreteMotion)”。

例

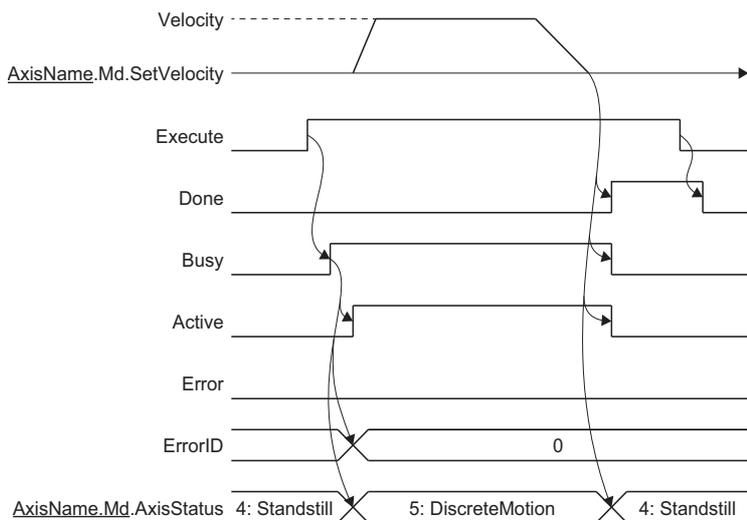
始点位置 (当前的停止位置)为“5000.0”时, 在移动量中设置了“-7000.0”的情况下

- 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。

☞ 11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。

软件行程限位有效时，“当前值+移动量”超出或低于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“超出目标位置范围(出错代码: 1A05H)”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(出错代码: 1AE4H)”(速度更改时为“超出运算周期换算速度范围警告(警告代码: 0D2FH)”)。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)使运算周期换算后的速度不小于“0.00001”。

- *2 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]为止的正数
1: 加减速时间恒定方式(mcFixedTime)	0.000000 ^{*1} 、0.000001~8400.0[s]为止的正数

- *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]为止的正数
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]为止的正数
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。

设置值	内容
0: Aborting(mcAborting)	中断(取消)执行中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在执行中的FB中缓冲下一个FB。 执行中的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) 执行中的FB完成时,依次执行缓冲FB。
2: BlendingLow(mcBlendingLow)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后,依次执行缓冲FB。 执行中的FB到达了目标位置时,将执行中的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)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后,依次执行缓冲FB。 执行中的FB到达了目标位置时,将执行中的FB的目标速度作为切换速度。
4: BlendingNext(mcBlendingNext)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后,依次执行缓冲FB。 执行中的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)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后,依次执行缓冲FB。 执行中的FB到达了目标位置时,将执行中的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 在执行中的FB与缓冲FB之间不执行停止。

要点

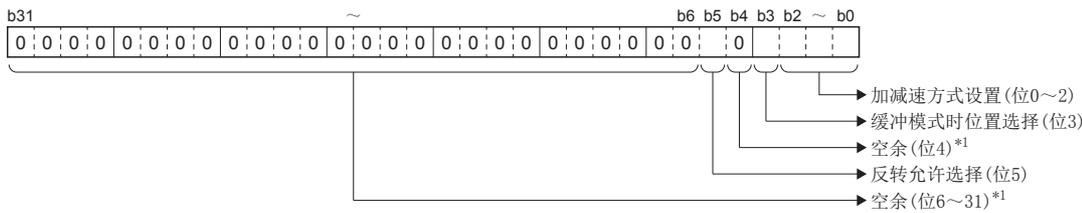
关于多重启动(缓冲模式)的详细内容,请参阅下述手册。

 MELSEC iQ-R运动模块用户手册(应用篇)

■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(出错代码: 1A4EH)”。

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 *: 缓冲模式(BufferMode)的“0: Aborting(mcAborting)”设置时，或目标位置更改时将有效。

• 加减速方式设置(位0~2)

设置值	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)

设置值	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>根据指令当前位置基准 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>根据反馈位置基准 仅移动移动量(Distance)</p>

- 反转允许选择(位5)

设置值	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“越程出错(出错代码: 1A7EH)”且进行减速停止。

■必要对象

使用MC_MoveRelative(相对值定位)的情况下，应将下述从对象设置为轴。

- Target position(607AH)

未设置从对象的情况下，将变为“必要从对象未设置(出错代码: 1AA8H)”且不启动。

关于与从对象设置有关的详细内容，请参阅下述手册。

📖MELSEC iQ-R运动模块用户手册(应用篇)

■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	设置值
目标位置	80000.0
速度	50000.0
加速度	100000.0
减速度	100000.0
Jerk	0.0

■轴

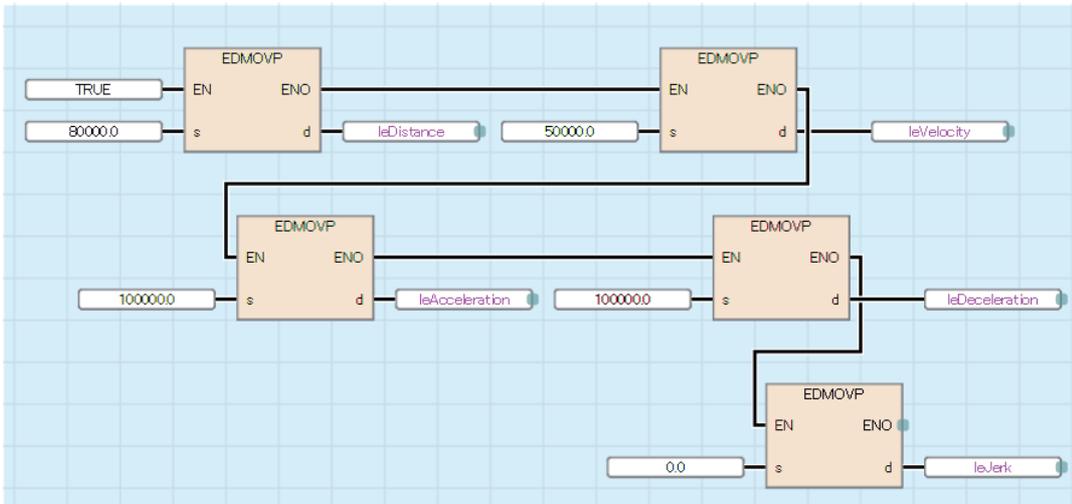
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

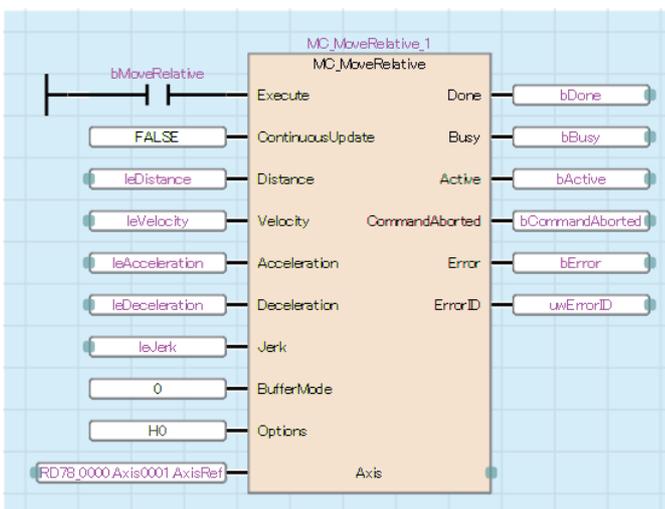
标签名	数据类型	注释
MC_MoveRelative_1	MC_MoveRelative	相对值定位FB
bMoveRelative	位	单轴相对值定位启动
leDistance	双精度实数	移动量
leVelocity	双精度实数	速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bDone	位	执行完成
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

- 单轴定位控制用数据设置



- 相对值定位



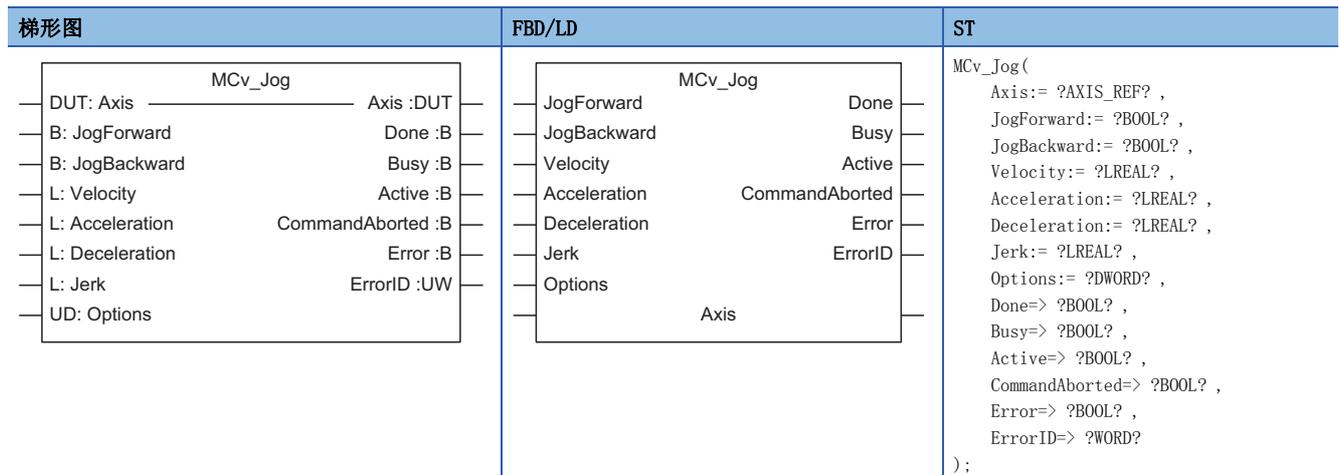
■ST的程序(运动模块侧)

```
//-----单轴定位控制用数据设置-----
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
);
```

MCv_Jog

按照指令速度执行JOG运行。



名称	输入区域点数(byte)	输出区域点数(byte)	编译方式	FB动作
JOG运行	52	8	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23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。 ☞ 207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。 ☞ 207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。 ☞ 208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。 ☞ 208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。 ☞ 209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。 ☞ 209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。 ☞ 209页 选项(Options)

■输出变量

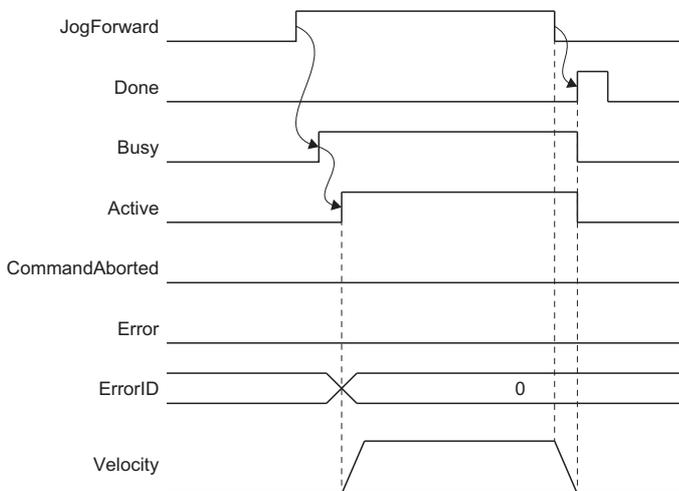
输出变量	名称	数据类型	初始值	内容
Done	执行完成	BOOL	FALSE	通过JOG指令的OFF，减速停止完成时，仅1个扫描变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_Jog(JOG运行)时，将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_Jog(JOG运行)正在控制轴时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MCv_Jog(JOG运行)的执行中断时，将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD(UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。  MELSEC iQ-R运动模块用户手册(应用篇)

功能

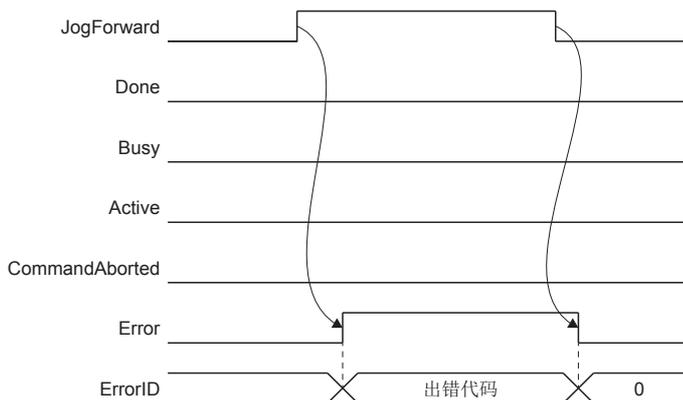
- 将正转JOG指令(JogForward)或反转JOG指令(JogBackward)置为TRUE时，对象轴将向指定的方向移动。
- JOG运行中的轴状态(AxisName.Md.AxisStatus)将变为“6：连续动作运行中(ContinuousMotion)”。
- 将正转JOG指令(JogForward)或反转JOG指令(JogBackward)置为FALSE时，将进行减速停止。
- 减速停止完成时，轴状态(AxisName.Md.AxisStatus)将变为“4：待机中(Standstill)”。
- 通过正转JOG指令(JogForward)或反转JOG指令(JogBackward)的FALSE的减速中，出错(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，出错(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。
- 其它动作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(警告代码：0D01H)”。
- 应在轴状态(AxisName.Md.AxisStatus)为“4：待机中(Standstill)”时进行启动。
- 在JOG运行中更改速度的情况下，应使用通过超驰功能进行的速度更改。关于超驰功能的详细内容，请参阅下述内容。
 MELSEC iQ-R运动模块用户手册(应用篇)

■时序图

- 正常完成的情况下

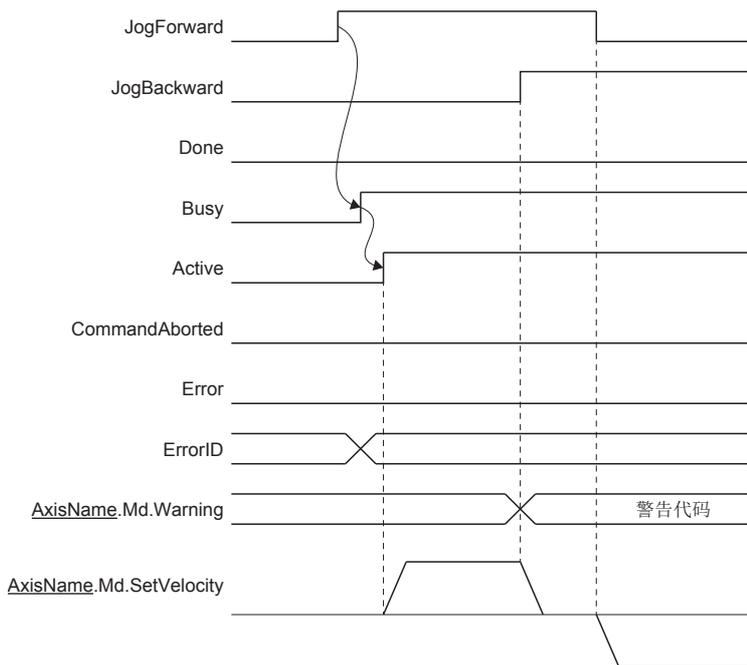


- 异常完成的情况下

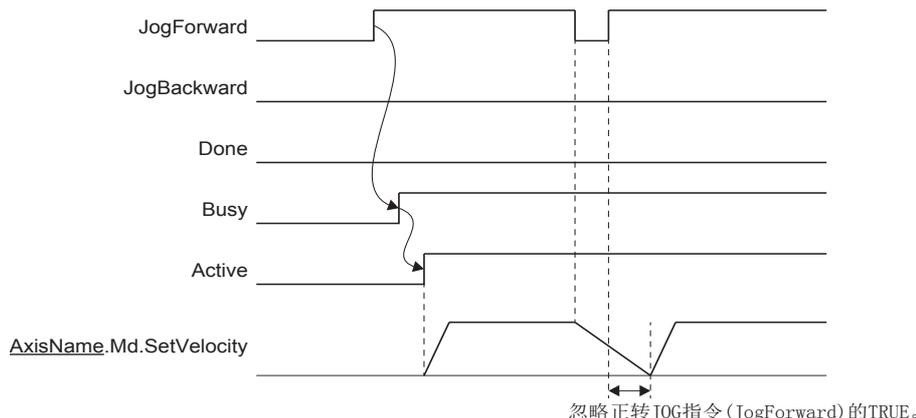


■ 正转JOG指令 (JogForward) / 反转JOG指令 (JogBackward)

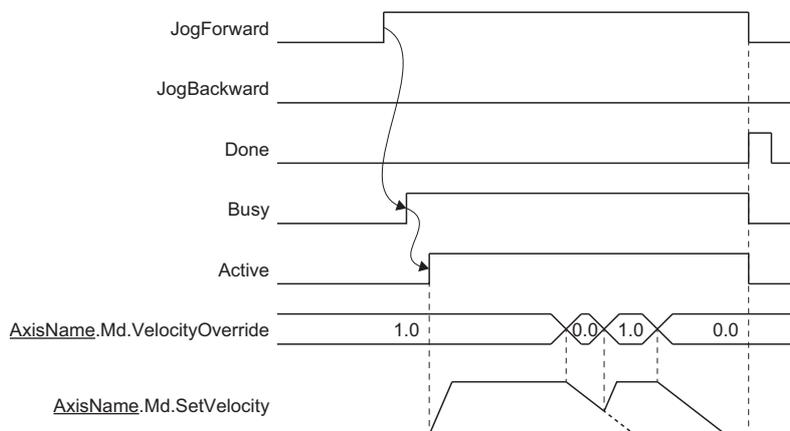
- 将正转JOG指令 (JogForward) 置为TRUE时将启动正转JOG运行，将反转JOG指令 (JogBackward) 置为TRUE时将启动反转JOG运行，且对象轴将向指定的方向移动。
- 同时将正转JOG指令 (JogForward) 与反转JOG指令 (JogBackward) 置为了TRUE的情况下，将变为“两方向JOG指令输入警告 (警告代码: 0D02H)”，执行中断 (CommandAborted) 将变为TRUE，且不进行JOG运行。之后，将两方向的JOG指令置为FALSE，将执行中断 (CommandAborted) 置为了FALSE的状态后，将正转JOG指令 (JogForward) 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 (警告代码: 0D02H)”，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，应使用通过超驰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(出错代码: 1AE4H)”(速度更改时为“超出运算周期换算速度范围警告(警告代码: 0D2FH)”)。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) 使运算周期换算后的速度不小于“0.00001”。

*2 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]为止的正数
1: 加减速时间恒定方式 (mcFixedTime)	0.000000 ^{*1} 、0.000001~8400.0[s]为止的正数

*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]为止的正数
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。

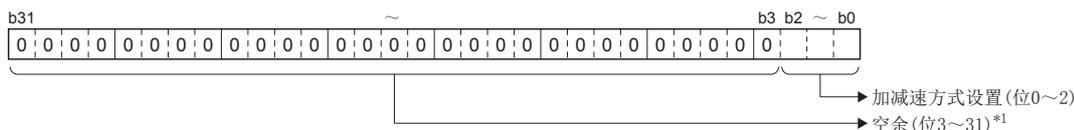
设置了加减速方式设置(选项(Options):位0~2)的“0:加减速速度指定方式(mcAccDec)”的情况下,对设置范围进行设置。设置了“1:加减速时间恒定方式(mcFixedTime)”的情况下,不使用Jerk。

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 ²]为止的正数
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(出错代码:1A4EH)”。

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)

设置值	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

使用MCv_Jog(JOG运行)的情况下,应将下述从对象设置为轴。

- Target position(607AH)

未设置从对象的情况下,将变为“必要从对象未设置(出错代码:1AA8H)”且不启动。

关于与从对象设置有关的详细内容,请参阅下述手册。

📖 MELSEC iQ-R运动模块用户手册(应用篇)

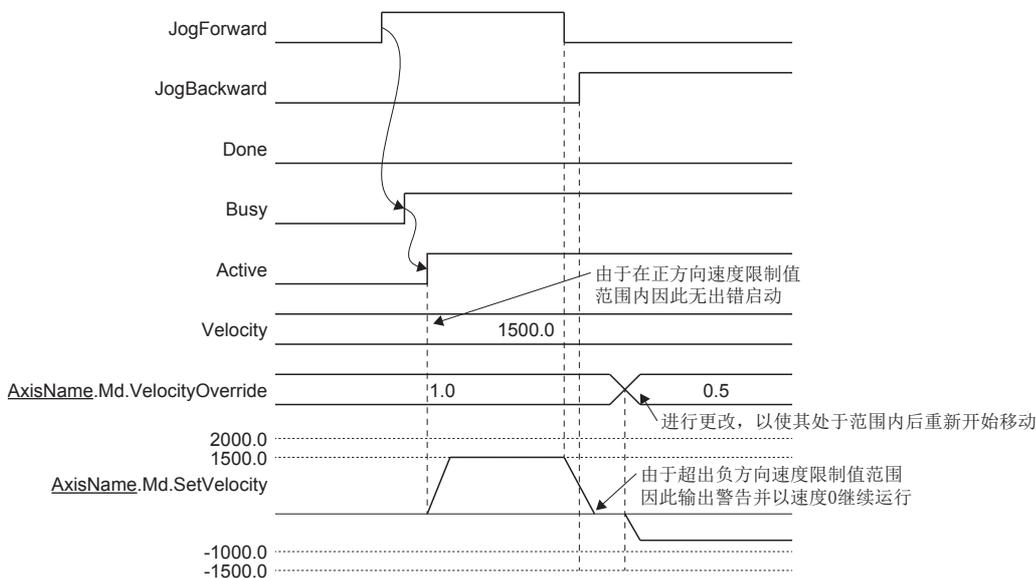
注意事项

- 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。
- 软件行程限位功能为有效，且软件行程限位超驰(AxisName.Cd.SwStrokeLimit_Override)为DISABLE(检查无效)及ONLY_INSIDE(仅至范围内方向检查无效)以外时，在MCv_Jog(JOG运行)执行时将软件行程限位超驰(AxisName.Cd.SwStrokeLimit_Override)改写为ONLY_INSIDE(仅至范围内方向检查无效)，且在JOG运行完成时将其改写为空白。
- 硬件行程限位功能为有效，且硬件行程限位超驰(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，在MCv_Jog(JOG运行)执行时将硬件行程限位超驰(AxisName.Cd.HwStrokeLimit_Override)改写为ONLY_INSIDE(仅至范围内方向检查无效)，且在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超驰(AxisName.Cd.SwStrokeLimit_Override)及硬件行程限位超驰(AxisName.Cd.HwStrokeLimit_Override)。
- JOG运行中进行了多重启动的情况下，运行中将进行下一个FB的分析，因此多重启动时的软件行程限位超驰(AxisName.Cd.SwStrokeLimit_Override)及硬件行程限位超驰(AxisName.Cd.HwStrokeLimit_Override)的值将被沿用到下一个FB，应加以注意。
- 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(警告代码：0D20H)”。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(警告代码：0D32H)”。(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	设置值
速度	1500.0
加速度	2000.0
减速度	2000.0
Jerk	0.0

■轴

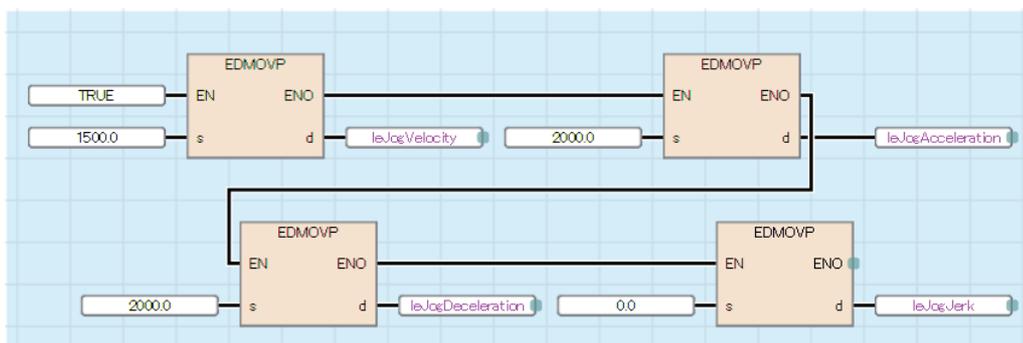
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

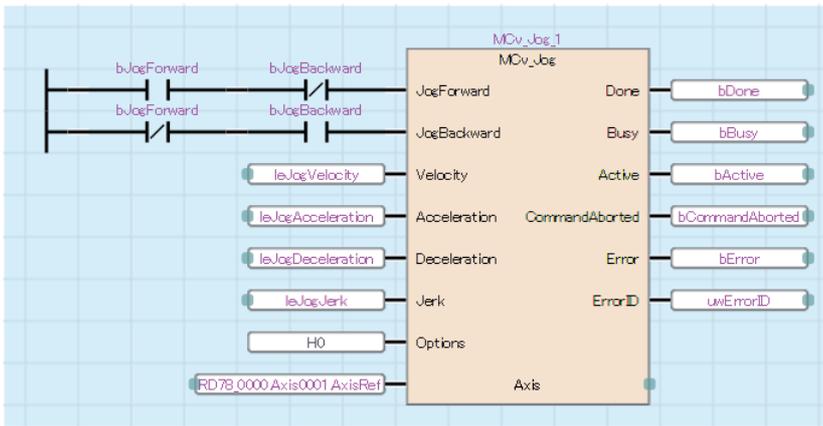
标签名	数据类型	注释
MCv_Jog_1	MCv_Jog	JOG运行FB
bJogForward	位	JOG正转指令
bJogBackward	位	JOG反转指令
leJogVelocity	双精度实数	JOG速度
leJogAcceleration	双精度实数	JOG加速度
leJogDeceleration	双精度实数	JOG减速度
leJogJerk	双精度实数	JOG Jerk
bJogF	位	JOG正转
bJogB	位	JOG反转
bDone	位	执行完成
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

• JOG运行用数据设置



• Axis0001 JOG运行



■ST的程序(运动模块侧)

```
//-----JOG运行用数据设置-----
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
);
```

MC_MoveVelocity

将驱动器切换为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:= ?WORD? , InVelocity=> ?BOOL? , Busy=> ?BOOL? , Active=> ?BOOL? , CommandAborted=> ?BOOL? , Error=> ?BOOL? , ErrorID=> ?WORD?); </pre>

名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
速度控制	56	8	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23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”的情况下，轴不动作但轴状态(AxisName.Md.AxisStatus)将变为“6: 连续动作运行中(ContinuousMotion)”。 关于详细内容，请参阅下述章节。 ☞ 215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。 ☞ 215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。 ☞ 216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。 ☞ 216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“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(出错代码: 1A37H)”。
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。 216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。 217页 选项(Options)

■输出变量

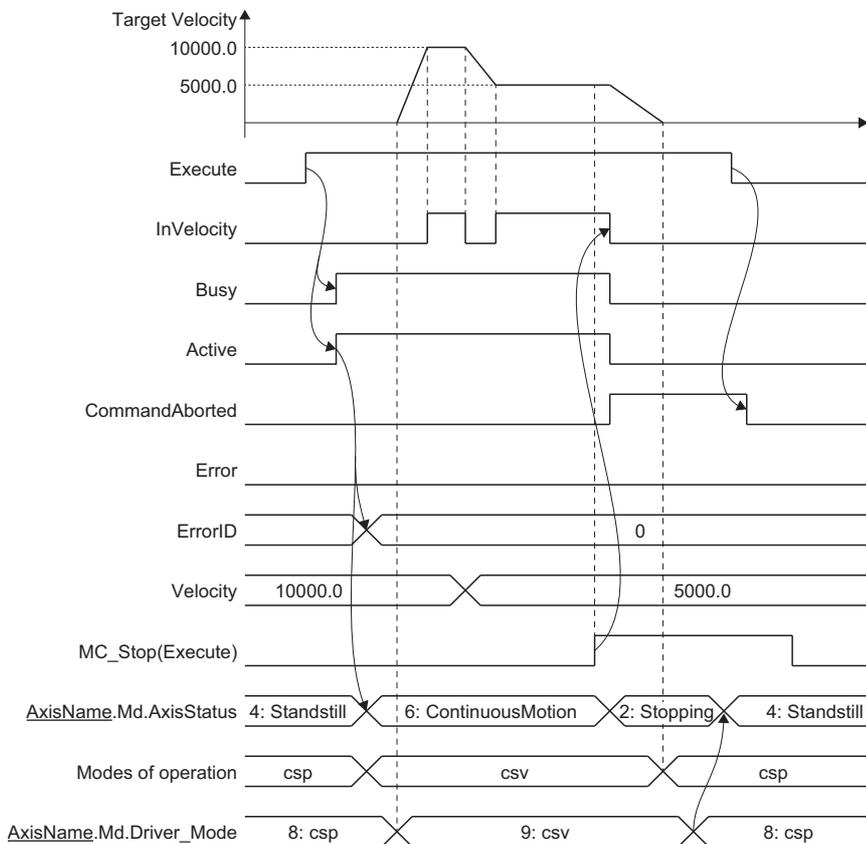
输出变量	名称	数据类型	初始值	内容
InVelocity	目标速度到达	BOOL	FALSE	运动系统中计算的指令速度到达了目标速度时, 将变为TRUE。 通过连接更新(ContinuousUpdate)为有效(TRUE)时的更改更改了目标速度的情况下, 在到达更改后的目标速度之前将变为FALSE。
Busy	执行中	BOOL	FALSE	执行了MC_MoveVelocity(速度控制)时, 将变为TRUE。
Active	控制中	BOOL	FALSE	MC_MoveVelocity(速度控制)正在控制轴时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MC_MoveVelocity(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	出错	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时, 将返回出错代码。 关于出错代码的详细内容, 请参阅下述手册。 MELSEC iQ-R运动模块用户手册(应用篇)

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。

☞ 11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(出错代码: 1AE4H)”(速度更改时为“超出运算周期换算速度范围警告(警告代码: 0D2FH)”)。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)使运算周期换算后的速度不小于“0.00001”。
- *2 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]为止的正数
1: 加减速时间恒定方式(mcFixedTime)	0.00000*1、0.000001~8400.0[s]为止的正数

- *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]为止的正数
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]为止的正数
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。

设置值	内容
0: Aborting(mcAborting)	中断(取消)执行中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在执行中的FB中缓冲下一个FB。 执行中的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) 执行中的FB完成时, 依次执行缓冲FB。

要点

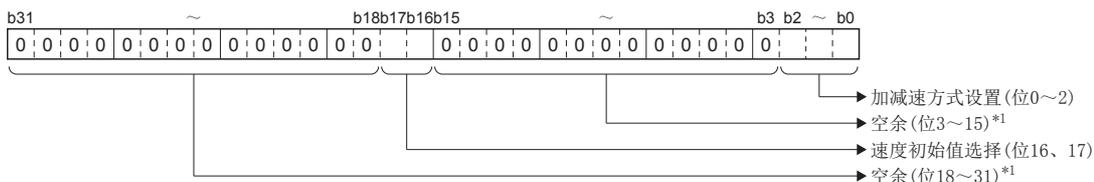
关于多重启动(缓冲模式)的详细内容, 请参阅下述手册。

 MELSEC iQ-R运动模块用户手册(应用篇)

■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(出错代码: 1A4EH)”。

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csp(循环位置模式)至csv(循环速度模式)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)

设置值	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。

• 速度初始值选择(位16、17)

设置值	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 未映射对对象的“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 未映射对对象的“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”。

■注意事项

- 速度超驰系数 (AxisName. Cd. VelocityOverride)、加速度超驰系数 (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。确认了驱动器控制模式 (AxisName. Md. Driver_Mode) 切换为“9: 速度控制(csv)”后, 应启动定位控制用的FB。
- 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, 控制模式切换时有可能产生振动及冲击, 因此应加以注意。
 - 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)”的设置值, 请参阅下述手册。
[MELSEC iQ-R运动模块用户手册\(应用篇\)](#)

程序示例

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	设置值
速度	100000.0
加速度	50000.0
减速度	50000.0
Jerk	0.0

■轴

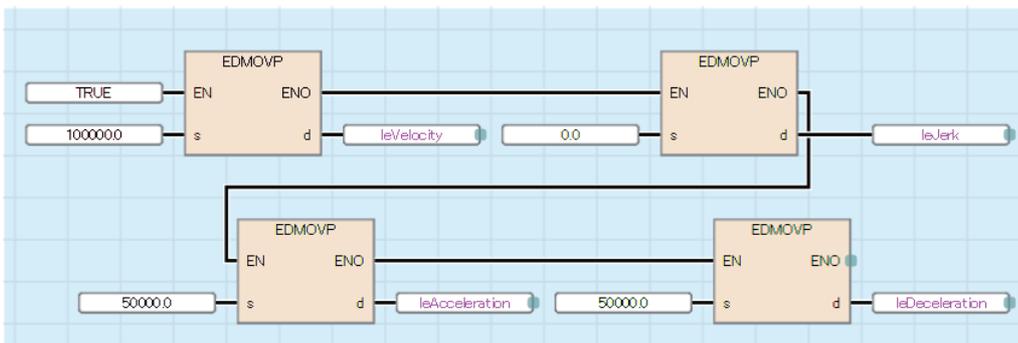
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

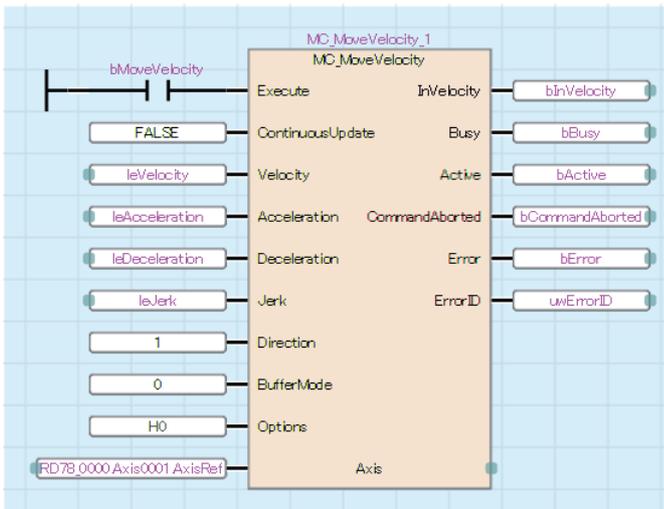
标签名	数据类型	注释
MC_MoveVelocity_1	MC_MoveVelocity	速度控制FB
bMoveVelocity	位	速度控制指令
leVelocity	双精度实数	指令速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bInVelocity	位	目标速度到达
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

- 速度控制用数据设置



• 速度控制



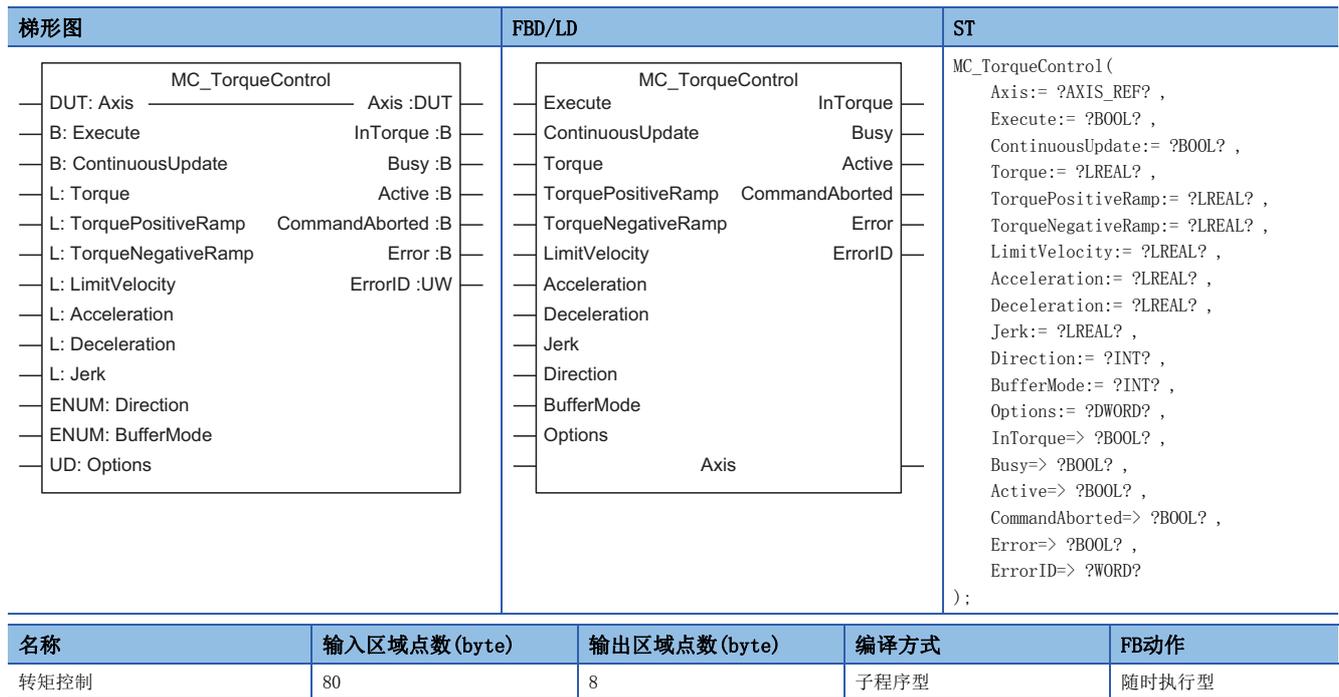
■ST的程序(运动模块侧)

```
//-----速度控制用数据设置-----
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
);
```

MC_TorqueControl

将驱动器切换为cst，按照指定的目标转矩进行转矩控制。



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.)，请参阅下述章节。 ☞ 23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，根据对对象图中分配的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。

输入变量	名称	数据类型	获取	有效范围	初始值	内容
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]	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。 ☞ 223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	■ “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	设置负方向的指令转矩。 ☞ 223页 转矩正方向斜率(TorquePositiveRamp)/ 转矩负方向斜率(TorqueNegativeRamp)
LimitVelocity	限制速度	LREAL	启动时/可重启/可连续更新	0.0001~ 2500000000.0	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, 请参阅驱动器的手册。 ■使用MR-J5(W)-G的情况下 • 速度限制值使用从对象的“Velocity limit value(2D20H)”。 • 未映射从对象的“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)”的设置值将变为有效。) *: 关于从对象的详细内容, 请参阅下述手册。 ☞MELSEC iQ-R运动模块用户手册(应用篇)
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。 ☞ 224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。 ☞ 224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。 ☞ 224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_DIRECTION)	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(出错代码: 1A37H)”。
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。 ☞ 224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。 ☞ 225页 选项(Options)

■输出变量

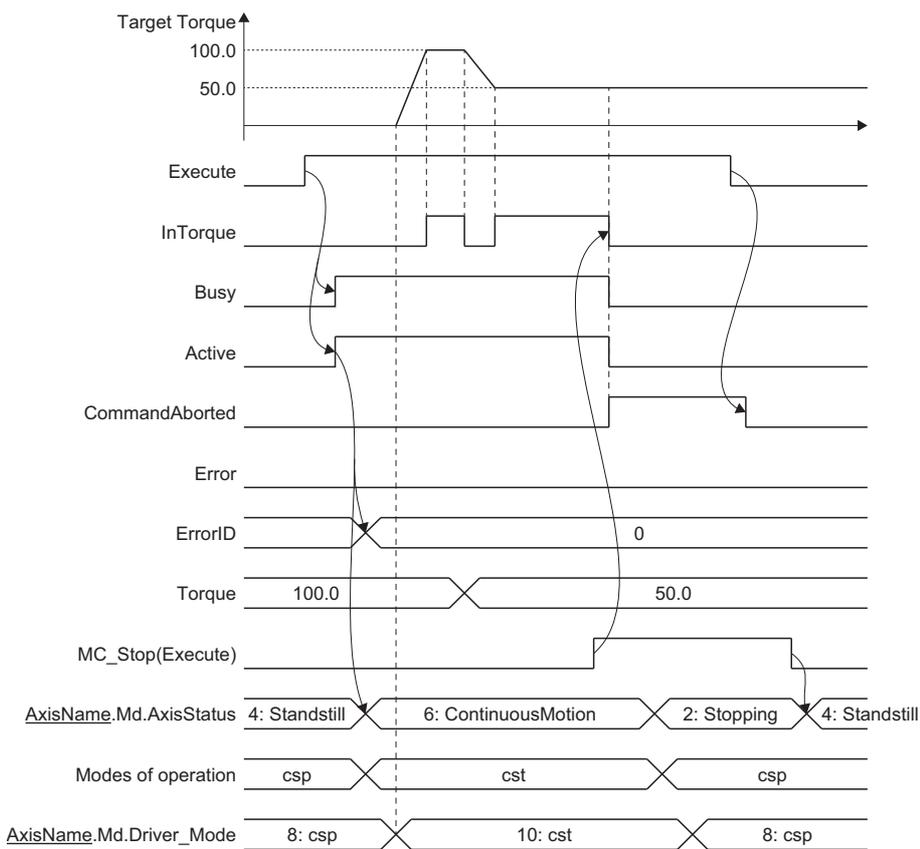
输出变量	名称	数据类型	初始值	内容
InTorque	目标转矩到达	BOOL	FALSE	指令转矩到达了目标转矩时，将变为TRUE。 通过连续更新(ContinuousUpdate)为TRUE时的更改更改了目标转矩的情况下，在到达更改后的目标转矩之前将变为FALSE。
Busy	执行中	BOOL	FALSE	执行了MC_TorqueControl(转矩控制)时，将变为TRUE。
Active	控制中	BOOL	FALSE	MC_TorqueControl(转矩控制)正在控制轴时，将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MC_TorqueControl(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	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD(UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 MELSEC iQ-R运动模块用户手册(应用篇)

功能

- 对于MC_TorqueControl(转矩控制)，将驱动器的控制模式切换为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。

11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置值。

到达目标转矩为止的设置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)的设置值	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>	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>	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>	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)的设置值指令转矩将变为“0.0”，之后，根据转矩正方向斜率(TorquePositiveRamp)的设置值将变为目标转矩。

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]为止的正数
1: 加减速时间恒定方式(mcFixedTime)	0.000000*1、0.000001~8400.0[s]为止的正数

*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]为止的正数
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]为止的正数
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。

设置值	内容
0: Aborting(mcAborting)	中断(取消)执行中的FB并立即执行下一个FB。 立即切换为转矩控制。
1: Buffered(mcBuffered)	在执行中的FB中缓冲下一个FB。 执行中的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) 执行中的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

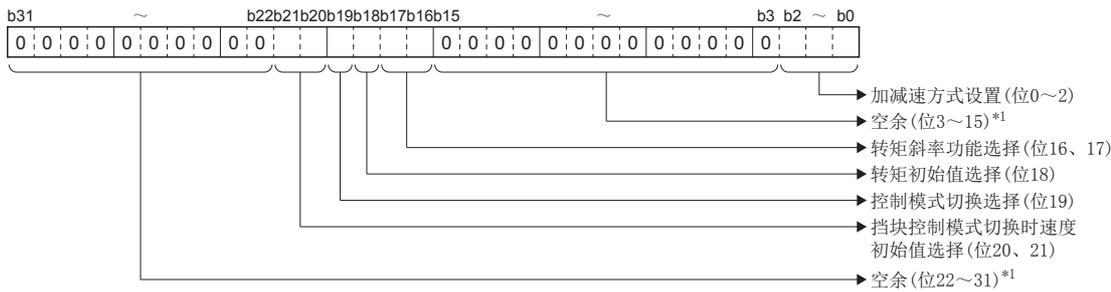
关于多重启动(缓冲模式)的详细内容, 请参阅下述手册。

 MELSEC iQ-R运动模块用户手册(应用篇)

■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(出错代码: 1A4EH)”。

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	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。 <ul style="list-style-type: none"> 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。 223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cst(循环转矩模式)时的转矩初始值。 <ul style="list-style-type: none"> 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。 <ul style="list-style-type: none"> 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csp(循环位置模式)向ct(挡块控制模式)切换控制模式时的速度初始值。 <ul style="list-style-type: none"> 0: 指令速度 1: 反馈速度 2: 自动选择

*2 从对象的“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(出错代码: 1AE9H)”。

关于挡块控制模式的详细内容, 请参阅下述手册。

MESE iQ-R运动模块用户手册(应用篇)

*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)

设置值	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。

• 转矩初始值选择(位18)

设置值	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。

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)

设置值	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 未映射从对象的“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 未映射从对象的“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”。

注意事项

- 速度超驰系数 (AxisName.Cd.VelocityOverride)、加速度超驰系数 (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。确认了驱动器控制模式 (AxisName.Md.Driver_Mode) 切换为“10: 转矩控制(cst)”后，应启动定位控制用的FB。
- 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，控制模式切换时有可能产生振动及冲击，因此应加以注意。
 - 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)”的设置值，请参阅下述手册。
□□MELSEC iQ-R运动模块用户手册(应用篇)

■关于cst(循环转矩模式)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)) 的值的的情况下，将变为“转矩限制值溢出警告(警告代码: 0D12H)”，并以更改前的值执行动作。
- 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) 的情况下，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	设置值
速度	100000.0
加速度	50000.0
减速度	50000.0
Jerk	0.0

■轴

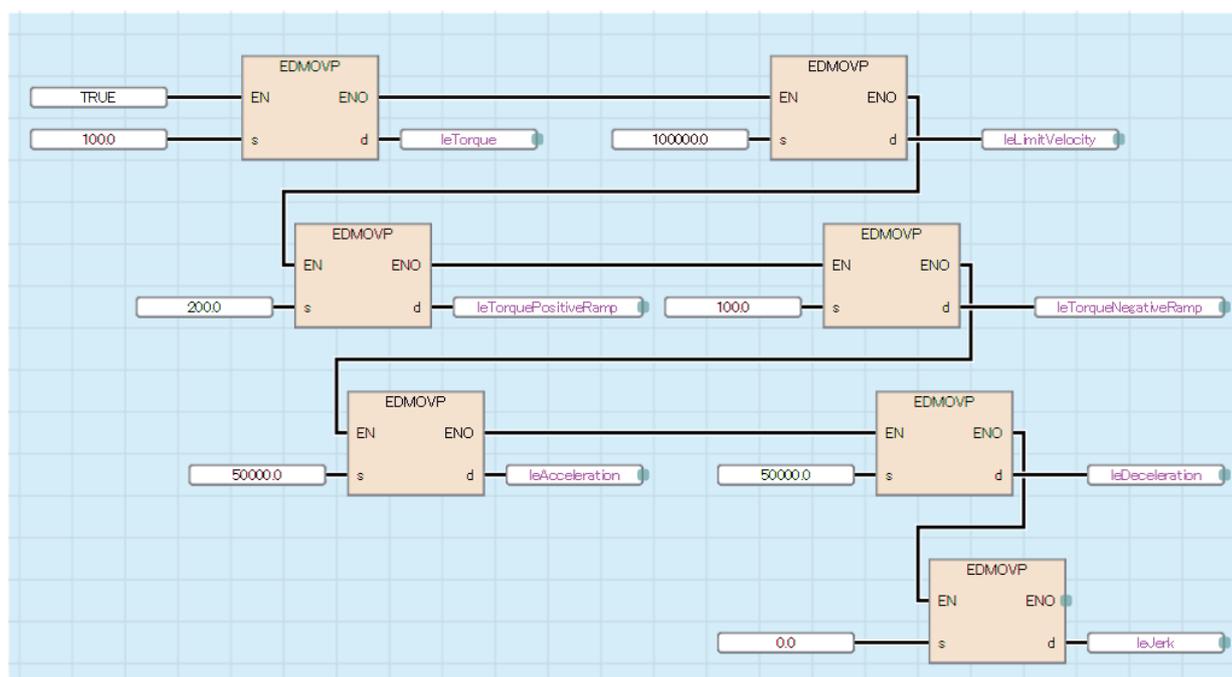
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

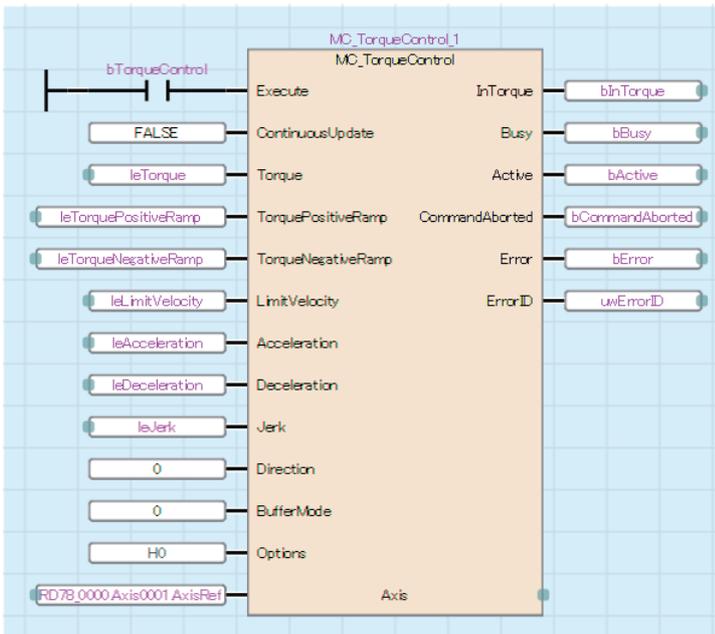
标签名	数据类型	注释
MC_TorqueControl_1	MC_TorqueControl	转矩控制FB
bTorqueControl	位	转矩控制指令
leTorque	双精度实数	目标转矩
leTorquePositiveRamp	双精度实数	转矩正方向斜率
leTorqueNegativeRamp	双精度实数	转矩负方向斜率
leLimitVelocity	双精度实数	限制速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bInTorque	位	目标转矩到达
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

- 转矩控制用数据设置



• 转矩控制



■ST的程序(运动模块侧)

//-----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
);
```

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:= ?WORD? , InVelocity=> ?BOOL? , Busy=> ?BOOL? , Active=> ?BOOL? , CommandAborted=> ?BOOL? , Error=> ?BOOL? , ErrorID=> ?WORD?); </pre>

名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
速度控制 (包含位置循环)	56	8	子程序型	随时执行型

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.AxisRef.</u>)，请参阅下述章节。 ☞ 23页 <u>AxisName.AxisRef.</u> (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。 省略了设置的情况下，轴不动作但轴状态 (<u>AxisName.Md.AxisStatus</u>) 将变为“6: 连续动作运行中 (ContinuousMotion)”。 关于详细内容，请参阅下述章节。 ☞ 231页 <u>速度 (Velocity)</u>
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。 ☞ 231页 <u>加速度 (Acceleration)</u>
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。 ☞ 232页 <u>减速度 (Deceleration)</u>
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。 ☞ 232页 <u>Jerk</u>

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“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(出错代码: 1A37H)”。
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。 232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。 233页 选项(Options)

■输出变量

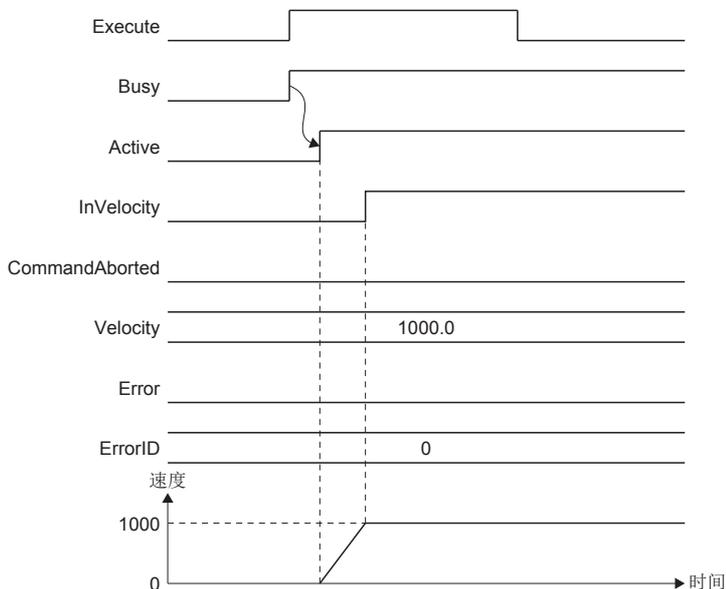
输出变量	名称	数据类型	初始值	内容
InVelocity	目标速度到达	BOOL	FALSE	指令速度到达了目标速度时, 将变为TRUE。 通过连接更新(ContinuousUpdate)为有效(TRUE)时的更改更改了目标速度的情况下, 在到达更改后的目标速度之前将变为FALSE。
Busy	执行中	BOOL	FALSE	执行了MCv_SpeedControl(速度控制(包含位置循环))时, 将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_SpeedControl(速度控制(包含位置循环))正在控制轴时, 将变为TRUE。
CommandAborted	执行中断	BOOL	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	出错	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时, 将返回出错代码。 关于出错代码的详细内容, 请参阅下述手册。 MELSEC iQ-R运动模块用户手册(应用篇)

功能

- 将驱动器侧的控制模式设置为csp，并根据对设置的轴进行了设置的速度执行速度控制。停止轴时，使用MC_Stop(强制停止)，或启动其它的动作系统FB。将驱动器侧的控制模式设置为csv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。

☞ 11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、±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(出错代码: 1AE4H)”(速度更改时为“超出运算周期换算速度范围警告(警告代码: 0D2FH)”)。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)使运算周期换算后的速度不小于“0.00001”。
- *2 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: 加减速度指定方式(mcAccDec)	0.0000*1、0.0001~2147483647.0[U/s ²]为止的正数
1: 加减速时间恒定方式(mcFixedTime)	0.000000*1、0.000001~8400.0[s]为止的正数

- *1 启动时根据启动时加减速度0指定时动作选择(AxisName.Pr.AccelerationZeroBehavior)，动作将变化。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]为止的正数
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]为止的正数
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。

设置值	内容
0: Aborting(mcAborting)	中断(取消)执行中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在执行中的FB中缓冲下一个FB。 执行中的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) 执行中的FB完成时,依次执行缓冲FB。
2: BlendingLow(mcBlendingLow)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后,依次执行缓冲FB。 执行中的FB到达了目标位置时,将执行中的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)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后,依次执行缓冲FB。 执行中的FB到达了目标位置时,将执行中的FB的目标速度作为切换速度。
4: BlendingNext(mcBlendingNext)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后,依次执行缓冲FB。 执行中的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)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后,依次执行缓冲FB。 执行中的FB到达了目标位置时,将执行中的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 在执行中的FB与缓冲FB之间不执行停止。

要点

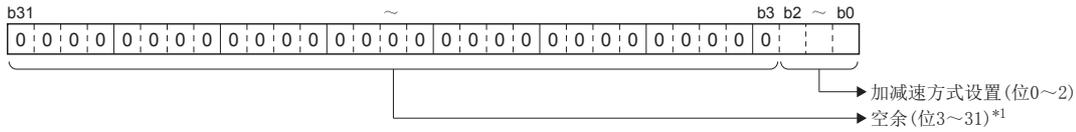
关于多重启动(缓冲模式)的详细内容,请参阅下述手册。

 MELSEC iQ-R运动模块用户手册(应用篇)

■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(出错代码: 1A4EH)”。

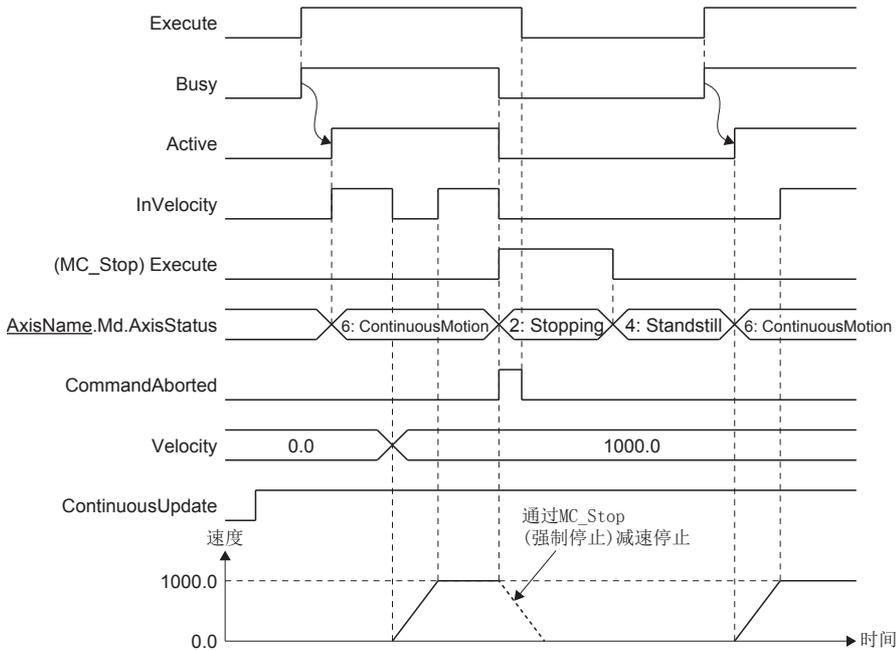
位	名称	内容
0~2	加减速方式设置	设置用于进行控制的加减速方式。 <ul style="list-style-type: none"> • 0: 加减速速度指定方式(mcAccDec) • 1: 加减速时间恒定方式(mcFixedTime)

• 加减速方式设置(位0~2)

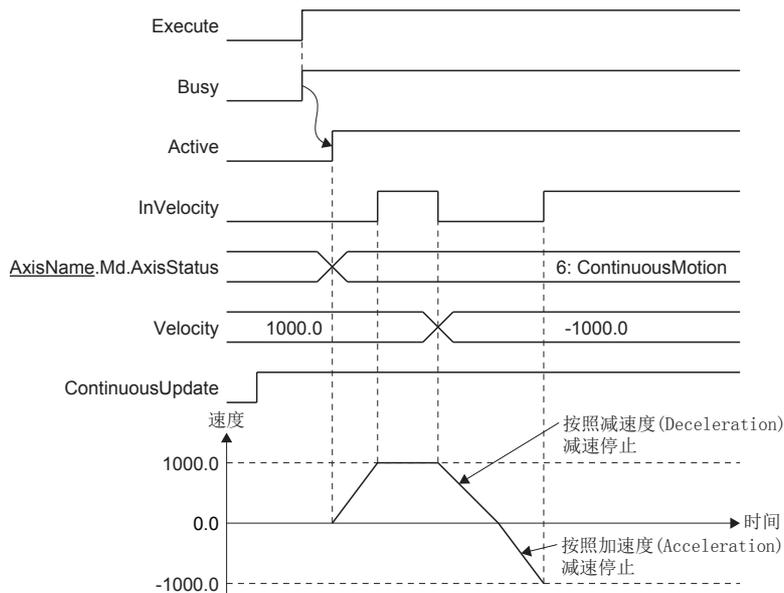
设置值	内容
0: 加减速速度指定方式(mcAccDec)	是使用MCv_SpeedControl(速度控制(包含位置循环))中设置的加速度(Acceleration)、减速度(Deceleration)、Jerk进行加速/减速的方式。
1: 加减速时间恒定方式(mcFixedTime)	是与速度无关，使用MCv_SpeedControl(速度控制(包含位置循环))中设置的加减速时间进行加速/减速的方式。对于加减速时间，设置为加速度(Acceleration)，不使用减速度(Deceleration)、Jerk。

■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

使用单轴的MCv_SpeedControl(速度控制(包含位置循环))的情况下，应将下述从对象设置为轴。

- Target position(607AH)

未设置从对象的情况下，将变为“必要从对象未设置(出错代码: 1AA8H)”且不启动。

关于从对象设置的详细内容，请参阅下述手册。

📖MELSEC iQ-R运动模块用户手册(应用篇)

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	设置值
速度	100000.0
加速度	50000.0
减速度	50000.0
Jerk	0.0

■轴

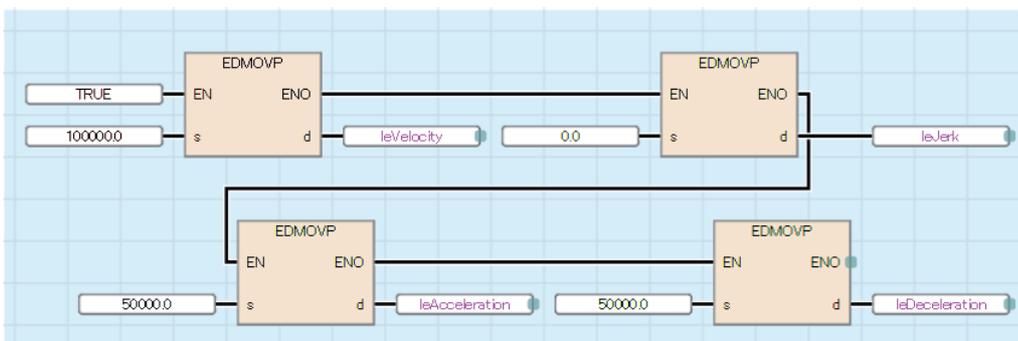
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1

■使用的标签

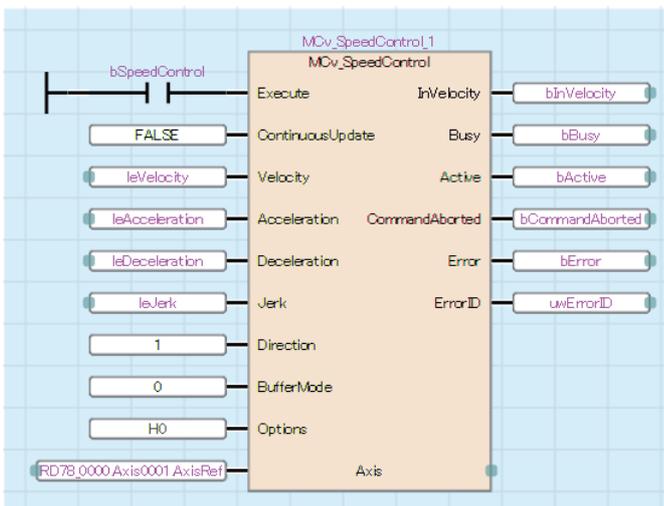
标签名	数据类型	注释
MCv_SpeedControl_1	MCv_SpeedControl	速度控制FB
bSpeedControl	位	速度控制指令
leVelocity	双精度实数	指令速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bInVelocity	位	目标速度到达
bBusy	位	执行中
bActive	位	控制中
bCommandAborted	位	执行中断
bError	位	出错
uwErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

- 速度控制用数据设置



- 速度控制(包含位置循环)



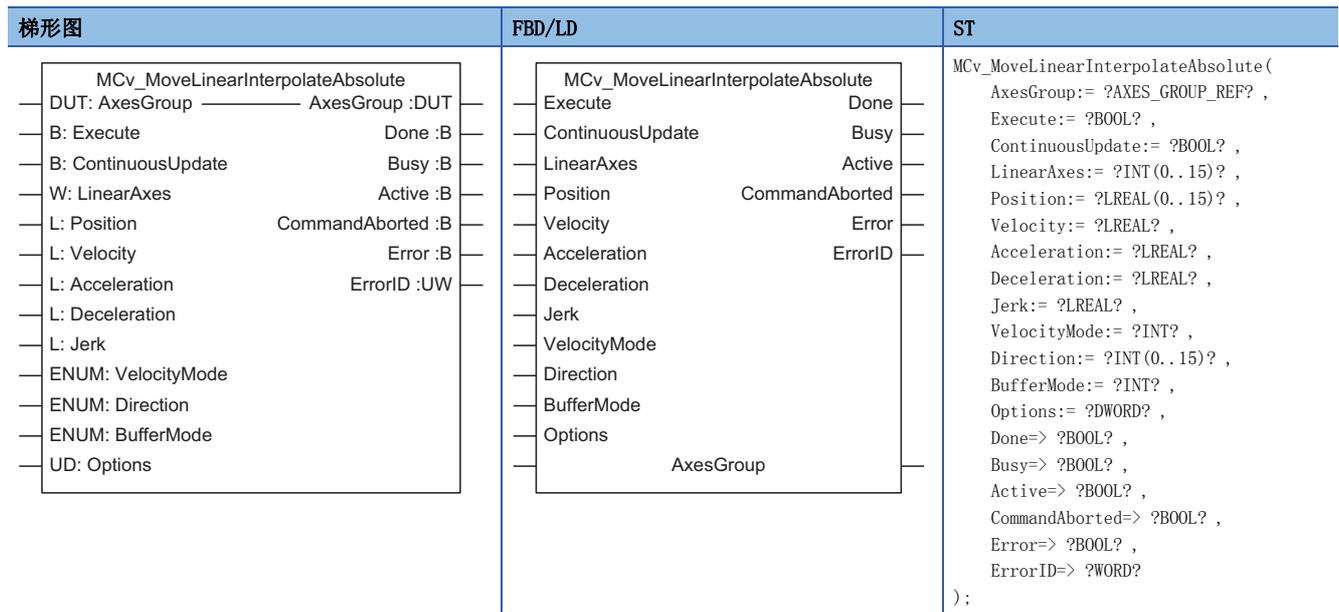
■ST的程序(运动模块侧)

```
//-----速度控制用数据设置-----
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
);
```

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。



名称	输入区域点数(byte)	输出区域点数(byte)	编译方式	FB动作
绝对值直线插补控制	248	8	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 44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_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	从构成轴中设置直线插补控制中使用的轴。 以数组设置构成轴的索引编号(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。 ☞ 242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。作为构成轴1~16的绝对位置处理。 关于详细内容，请参阅下述章节。 ☞ 242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、0.0001~2500000000.0	0.0	设置速度指令值。 关于详细内容，请参阅下述章节。 ☞ 243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。 ☞ 243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。 ☞ 243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。 ☞ 243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。 ☞ 244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。作为构成轴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(出错代码: 1A37H)”。 关于详细内容，请参阅下述章节。 ☞ 245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。 ☞ 245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。 ☞ 246页 选项 (Options)

■输出变量

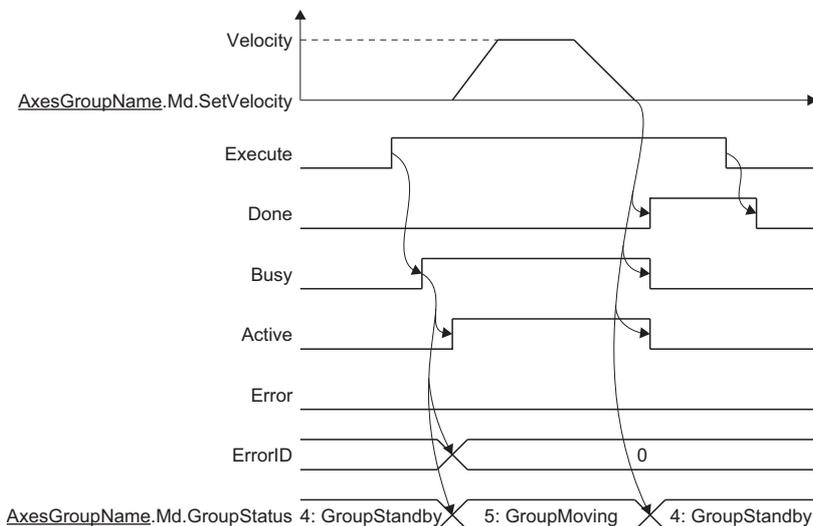
输出变量	名称	数据类型	初始值	内容
Done	完成	BOOL	FALSE	控制完成时，将变为TRUE。 动作完成时根据启动(Execute)的状态将变为如下所示。 ■启动(Execute)为TRUE的情况下 在将启动(Execute)置为FALSE之前将保持为TRUE不变。 ■启动(Execute)为FALSE的情况下 仅1周期为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)时，将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_MoveLinearInterpolateAbsolute(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	由于出错及多重启动等，MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)的执行中断时，将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD(UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 □□MELSEC iQ-R运动模块用户手册(应用篇)

功能

- 直线插补控制时，指定轴组进行插补控制，使从始点(移动开始点)至终点的轨迹成为直线。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。

■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。

☞ 11页 通过执行指令(Execute)类型的运动控制FB的基本动作

■1轴直线插补控制(绝对值)的动作概要

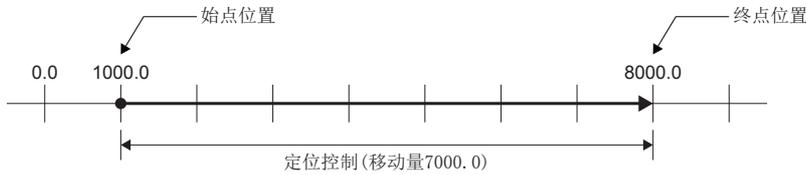
从启动时的当前位置(始点位置)开始向目标位置(Position)中设置的指定位置(终点位置)进行1轴的直线插补。

例

构成轴1的始点位置为“1000.0”时,在目标位置(Position)中设置了“8000.0”的情况下

```
LinearAxes[0]:= 1
```

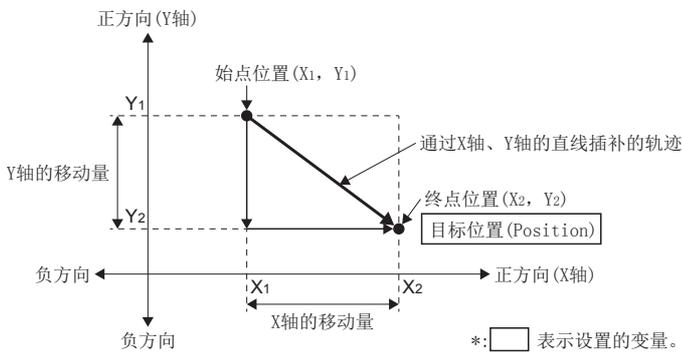
```
Position[0]:= 8000.0
```



■2轴直线插补控制(绝对值)的动作概要

从启动时的当前位置(始点位置)开始向目标位置(Position)中设置的指定位置(终点位置)进行2轴的直线插补。

移动方向取决于各轴的始点位置与终点位置。



例

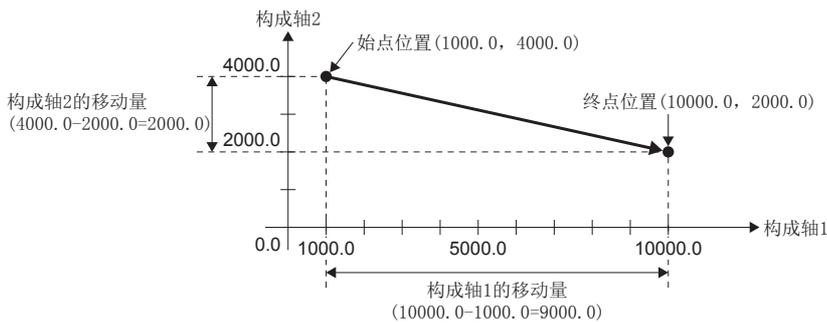
始点位置为“构成轴1: 1000.0, 构成轴2: 4000.0”时,在目标位置(Position)中设置了“构成轴1: 10000.0, 构成轴2: 2000.0”的情况下

```
LinearAxes[0]:= 1
```

```
LinearAxes[1]:= 2
```

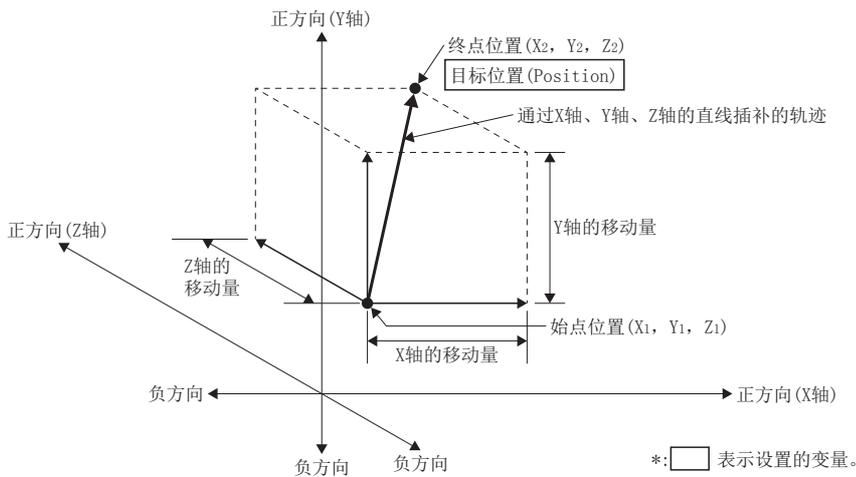
```
Position[0]:= 10000.0
```

```
Position[1]:= 2000.0
```



■3轴直线插补控制(绝对值)的动作概要

从启动时的当前位置(始点位置)开始向目标位置(Position)中设置的指定位置(终点位置)进行3轴的直线插补。
移动方向根据各轴的始点位置与终点位置确定。



例

始点位置为“构成轴1: 1000.0, 构成轴2: 2000.0, 构成轴3: 1000.0”时, 在目标位置(Position)中设置了“构成轴1: 7000.0, 构成轴2: 8000.0, 构成轴3: 4000.0”的情况下

```
LinearAxes[0]:= 1
```

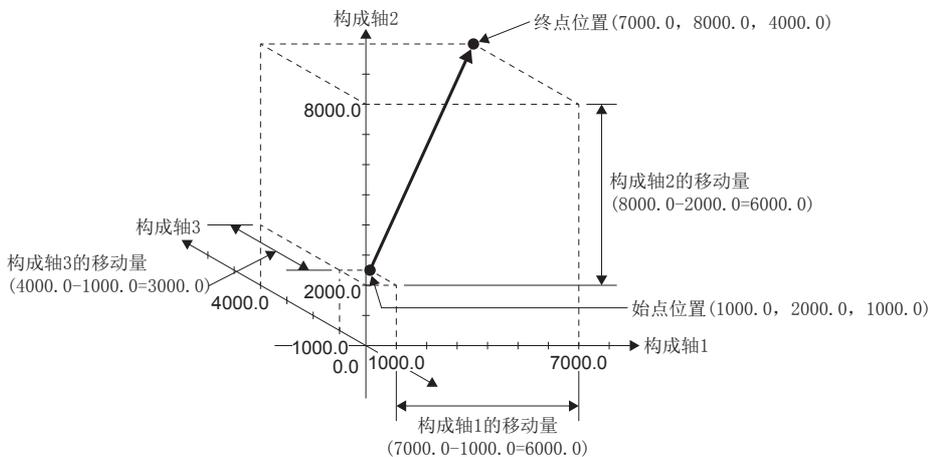
```
LinearAxes[1]:= 2
```

```
LinearAxes[2]:= 3
```

```
Position[0]:= 7000.0
```

```
Position[1]:= 8000.0
```

```
Position[2]:= 4000.0
```



■直线插补轴(LinearAxes)

在直线插补控制中，从轴组中设置的构成轴中，使用任意的轴进行直线插补。

从轴组的构成轴中，通过直线插补轴(LinearAxes)设置进行直线插补的构成轴。直线插补轴(LinearAxes)具有16个数组元素。在数组中，将插补控制中使用的构成轴的索引编号(1~16: 构成轴1~16)仅以插补控制中使用的构成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构成轴数不大于构成轴中登录的轴数。
- 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元素的构成轴将成为基准轴。
- 进行了以下设置的情况下，将变为“直线插补轴设置不正确(出错代码: 1AB9H)”且不启动。
 - 设置了轴未设置的构成轴的情况下
 - 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

将构成轴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 “LinearAxes[3]:= 0;~LinearAxes[15]:= 0;”可以省略。

■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。
 - ☞ 188页 目标位置(Position)
- Position[0..15]表示构成轴1~16的目标位置。对直线插补轴(LinearAxes)中设置的构成轴设置位置。
- 未设置为直线插补轴的构成轴的目标位置(Position)将被忽略。
- 设置为直线插补轴的所有构成轴后面的目标位置(Position)可以省略。

例

将构成轴2、3、4设置为直线插补轴，并在构成轴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]:= 2000.0;  
Position[1]:= 3000.0;  
Position[2]:= 4000.0;  
Position[3]:= 0.0; *1  
:  
Position[15]:= 0.0; *1
```

*1 “LinearAxes[3]:= 0;~LinearAxes[15]:= 0;”与“Position[3]:= 0.0;~Position[15]:= 0.0;”可以省略。

■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(出错代码: 1AE4H)”(速度更改时为“超出运算周期换算速度范围警告(警告代码: 0D2FH)”)。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)使运算周期换算后的速度不小于“0.00001”。
- *2 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]为止的正数
1: 加减速时间恒定方式(mcFixedTime)	0.000000 ^{*1} 、0.000001~8400.0[s]为止的正数

- *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]为止的正数
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

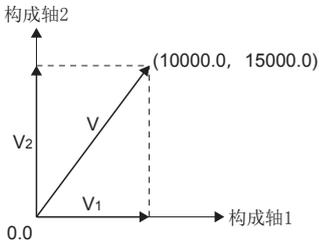
设置MCv_MoveLinearInterpolateAbsolute(绝对值直线插补控制)中的Jerk。

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]为止的正数
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元素中设置的构成轴。

设置值	内容
0: 合成速度 (VectorSpeed)	<p>速度 (Velocity) 设置为合成速度。</p> <p>对于各轴的定位速度 (V_n), 运动系统根据设置的控制对象的定位速度 (V), 通过各轴的移动量 (D_n) 计算。</p> <p><例></p> <p>2轴的直线插补控制的情况下</p>  <ul style="list-style-type: none"> • 构成轴1的移动量 (D_1): 10000.0 [pulse] • 构成轴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"> • 构成轴1的定位速度: $V_1 = V \times D_1 / \sqrt{D_1^2 + D_2^2}$ • 构成轴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。</p> <p>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。</p> <p>对于其它插补轴的定位速度 (V_n), 运动系统通过各插补轴的移动量 (D_n) 计算。</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"> • 构成轴1的移动量 (D_1): 10000.0 [pulse] • 构成轴2的移动量 (D_2): 15000.0 [pulse] • 构成轴3的移动量 (D_3): 5000.0 [pulse] • 构成轴4的移动量 (D_4): 20000.0 [pulse] • 构成轴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构成轴4, 并以长轴速度控制构成轴4。</p> <p>对于其它构成轴的定位速度, 运动系统通过下述计算公式计算。</p> <ul style="list-style-type: none"> • 构成轴1的定位速度: $V_1 = D_1 / D_4 \times V$ • 构成轴2的定位速度: $V_2 = D_2 / D_4 \times V$ • 构成轴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: 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 (D_n) 计算其它插补轴的定位速度 (V_n) 并进行控制。</p> <p>作为插补轴直线插补轴 (LinearAxes) 的第1元素中设置的轴将变为基准轴。</p> <p>设置为基准轴的构成轴的移动量为“0.0”的情况下, 将变为“基准轴移动量0 (出错代码: 1AABH)”。</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"> • 构成轴1的移动量 (D_1): 10000.0 [pulse] • 构成轴2的移动量 (D_2): 15000.0 [pulse] • 构成轴3的移动量 (D_3): 5000.0 [pulse] • 构成轴4的移动量 (D_4): 20000.0 [pulse] • 基准轴速度 (V): 7000.0 [pulse/s] <p>上述的情况下, 基准轴将为构成轴4, 并以构成轴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"> • 构成轴1的定位速度: $V_1 = D_1 / D_4 \times V$ • 构成轴2的定位速度: $V_2 = D_2 / D_4 \times V$ • 构成轴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。

- 在速度模式(VelocityMode)中设置了“0: 合成速度(VectorSpeed)”时, 各轴的移动量超出“4, 294, 967, 296.0(=2³²)”的情况下, 将变为“超出直线移动量范围(出错代码: 1ADEH)”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(出错代码: 1A61H)”。

■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]表示构成轴1~16的方向选择。对直线插补轴(LinearAxes)中设置的构成轴设置值。
- 未设置为直线插补轴的构成轴的方向选择(Direction)将被忽略。
- 设置为直线插补轴的所有构成轴后面的方向选择(Direction)可以省略。
- 设置了设置值以外的值的情况下, 将变为“超出方向选择范围(出错代码: 1A37H)”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。

☞ 190页 方向选择(Direction)

设置值	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。

设置值	内容
0: Aborting(mcAborting)	中断(取消)执行中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在执行中的FB中缓冲下一个FB。 执行中的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) 执行中的FB完成时, 依次执行缓冲FB。
2: BlendingLow(mcBlendingLow)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的FB到达了目标位置时, 将执行中的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)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的FB到达了目标位置时, 将执行中的FB的目标速度作为切换速度。
4: BlendingNext(mcBlendingNext)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的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)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的FB到达了目标位置时, 将执行中的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 在执行中的FB与缓冲FB之间不执行停止。

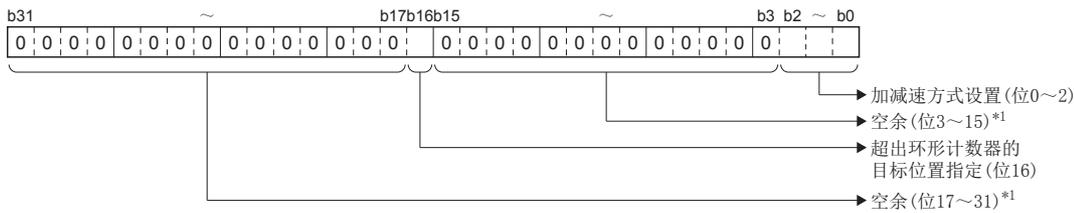
要点

关于多重启动(缓冲模式)的详细内容, 请参阅下述手册。

📖 MELSEC iQ-R运动模块用户手册(应用篇)

■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(出错代码: 1A4EH)”。

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)

设置值	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

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构成轴设置下述从对象。

• Target position(607AH)

存在有未设置从对象的构成轴的情况下，将变为“必要从对象未设置(出错代码: 1AA8H)”且不启动。

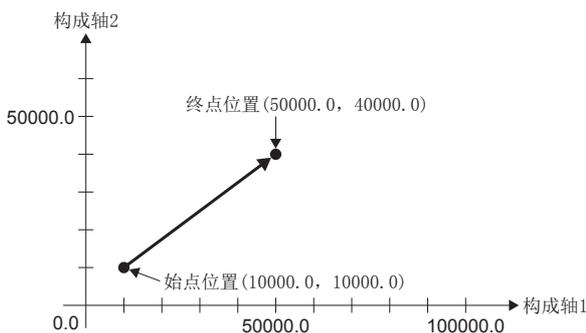
关于与从对象设置有关的详细内容，请参阅下述手册。

📖MELSEC iQ-R运动模块用户手册(应用篇)

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



• 设置

项目	设置值	
	构成轴1	构成轴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

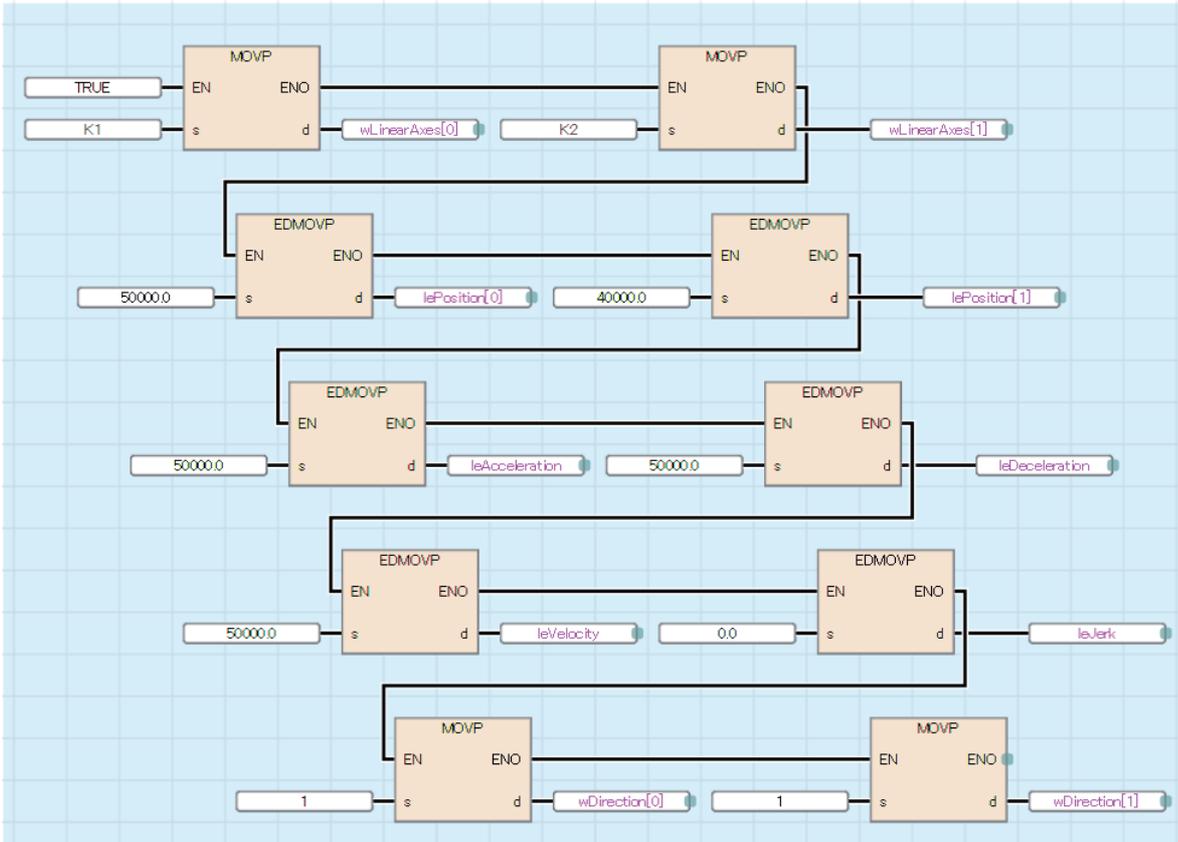
轴组No.	标签名	数据类型	注释
1	AxesGroup001	AXES_GROUP_REF	轴组1

■使用的标签

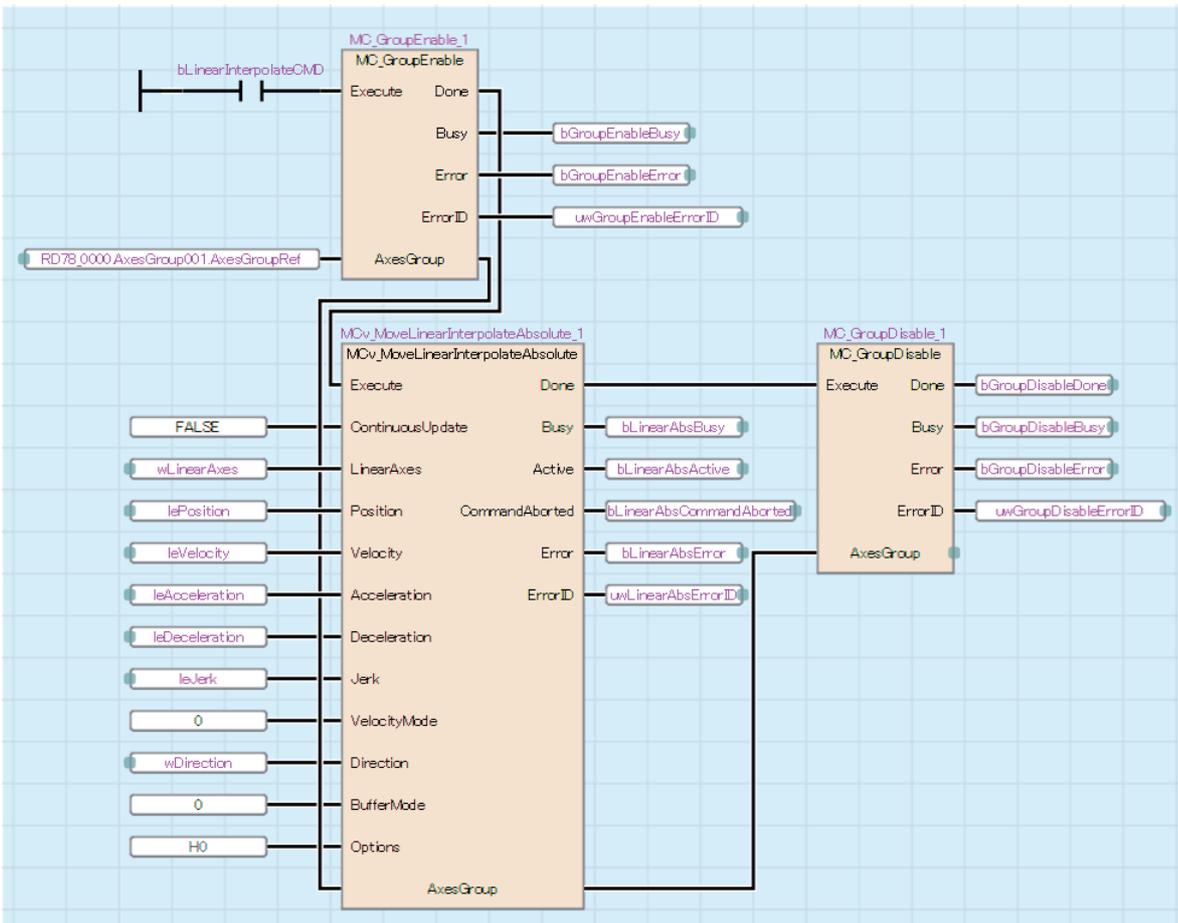
标签名	数据类型	注释
MCv_MoveLinearInterpolateAbsolute_1	MCv_MoveLinearInterpolateAbsolute	绝对值直线插补控制FB
wLinearAxes	字[有符号](0..15)	直线插补轴
lePosition	双精度实数(0..15)	目标位置
leVelocity	双精度实数	速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
wDirection	字[有符号](0..15)	方向选择
bLinearAbsDone	位	执行完成
bLinearAbsBusy	位	执行中
bLinearAbsActive	位	控制中
bLinearAbsCommandAborted	位	执行中断
bLinearAbsError	位	出错
uwLinearAbsErrorID	字[无符号]/位串[16位]	出错代码
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	出错
uwGroupEnableErrorID	字[无符号]/位串[16位]	出错代码
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	出错
uwGroupDisableErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

- 2轴直线插补控制用数据设置



- 轴组有效/2轴直线插补控制/轴组无效



■ST的程序(运动模块侧)

```
//-----2轴直线插补控制用数据设置-----
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
);

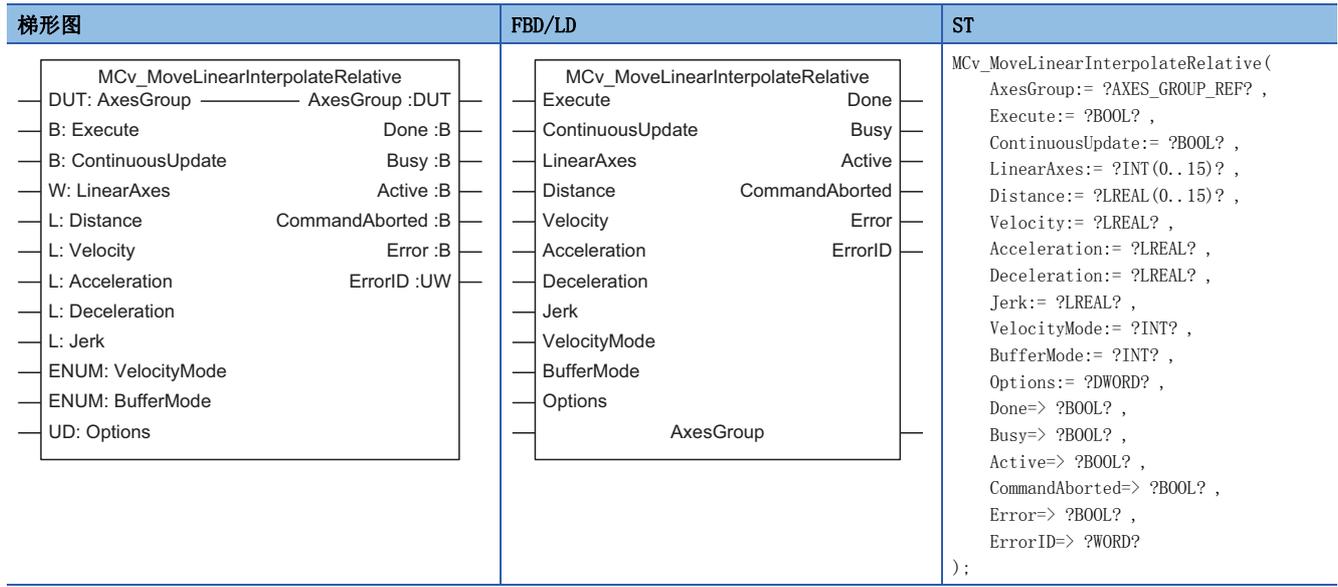
//-----2轴直线插补控制-----
MCv_MoveLinearInterpolateAbsolute_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bGroupEnableDone ,
    ContinuousUpdate:= FALSE ,
    LinearAxes:= wLinearAxes ,
    Position:= lePosition ,
    Velocity:= leVelocity ,
    Acceleration:= leAcceleration ,
    Deceleration:= 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  
);
```

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。



名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
相对值直线插补控制	216	8	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 44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_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	从构成轴中指定直线插补控制中使用的轴。 以数组指定构成轴的索引编号(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。 ☞ 256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。作为构成轴1~16的相对位置进行处理。 关于详细内容，请参阅下述章节。 ☞ 256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、0.0001~2500000000.0	0.0	设置速度指令值。 关于详细内容，请参阅下述章节。 ☞ 257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。 ☞ 257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。 ☞ 257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。 ☞ 257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。 ☞ 258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。 ☞ 259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。 ☞ 259页 选项 (Options)

■输出变量

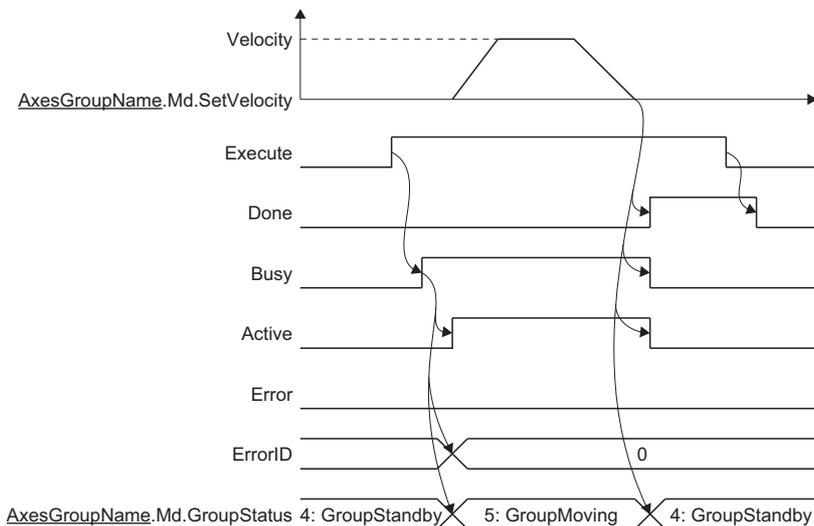
输出变量	名称	数据类型	初始值	内容
Done	完成	BOOL	FALSE	控制完成时，将变为TRUE。 动作完成时根据启动 (Execute) 的状态将变为如下所示。 ■启动 (Execute) 为TRUE的情况下 在将启动 (Execute) 置为FALSE之前将保持为TRUE不变。 ■启动 (Execute) 为FALSE的情况下 仅1周期为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_MoveLinearInterpolateRelative (相对值直线插补控制) 时，将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_MoveLinearInterpolateRelative (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	由于出错及多重启动等， MCv_MoveLinearInterpolateRelative (相对值直线插补控制) 的执行中断时，将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册 (应用篇)

功能

- 直线插补控制时，指定轴组进行插补控制，使从始点(移动开始点)至终点的轨迹成为直线。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。

■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。

☞ 11页 通过执行指令(Execute)类型的运动控制FB的基本动作

■1轴直线插补控制(相对值)的动作概要

从启动时的当前位置(始点位置)开始，以移动量(Distance)中设置的移动量进行1轴的直线插补。

移动方向取决于移动量的符号(+/-)。

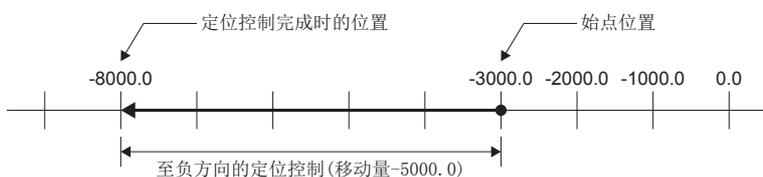
移动量	内容
正(+)	至正方向(地址增加方向)的定位
负(-)	至负方向(地址减少方向)的定位

例

构成轴1的始点位置为“-3000.0”时，在移动量(Distance)中设置了“-5000.0”的情况下

LinearAxes[0]:= 1

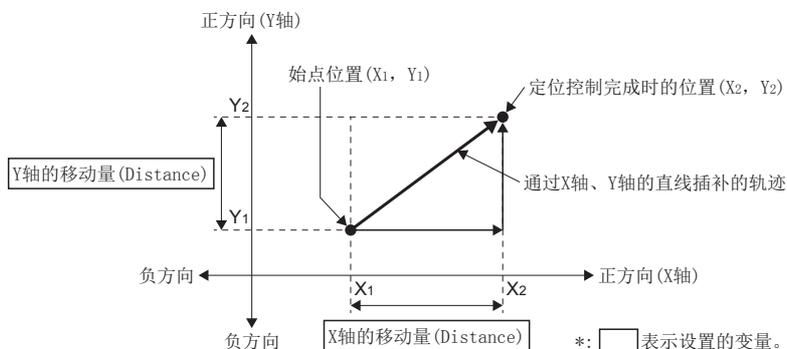
Distance[0]:= -5000.0



■2轴直线插补控制(相对值)的动作概要

从启动时的当前位置(始点位置)开始,以移动量(Distance)中设置的移动量进行2轴的直线插补。
移动方向取决于各轴的移动量的符号(+/-)。

移动量	内容
正(+)	至正方向(地址增加方向)的定位
负(-)	至负方向(地址减少方向)的定位



例

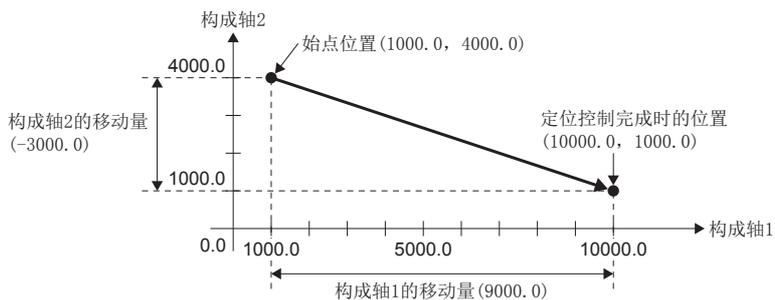
始点位置为“构成轴1: 1000.0, 构成轴2: 4000.0”时,在移动量(Distance)中设置了“构成轴1: 9000.0, 构成轴2: -3000.0”的情况下

LinearAxes[0]:= 1

LinearAxes[1]:= 2

Distance[0]:= 9000.0

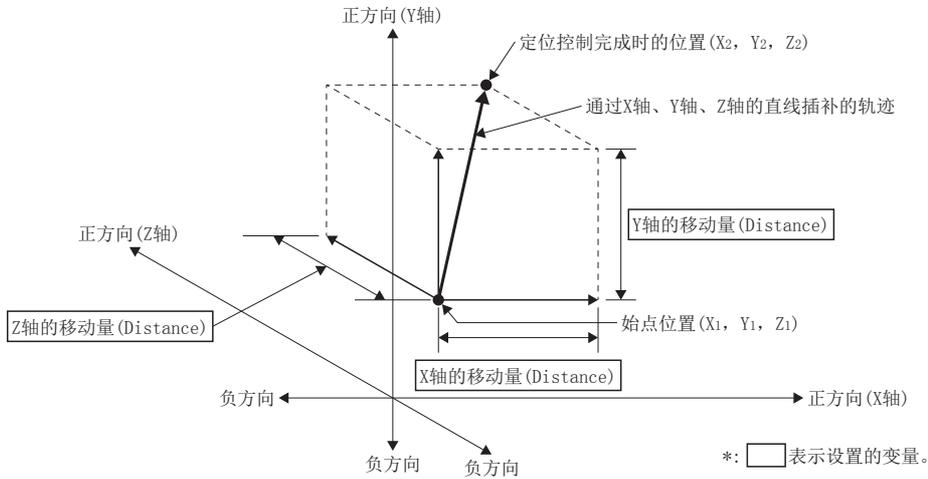
Distance[1]:= -3000.0



■3轴直线插补控制(相对值)的动作概要

从启动时的当前位置(始点位置)开始,以移动量(Distance)中设置的移动量进行3轴的直线插补。移动方向取决于各轴的移动量的符号(+/-)。

移动量	内容
正(+)	至正方向(地址增加方向)的定位
负(-)	至负方向(地址减少方向)的定位



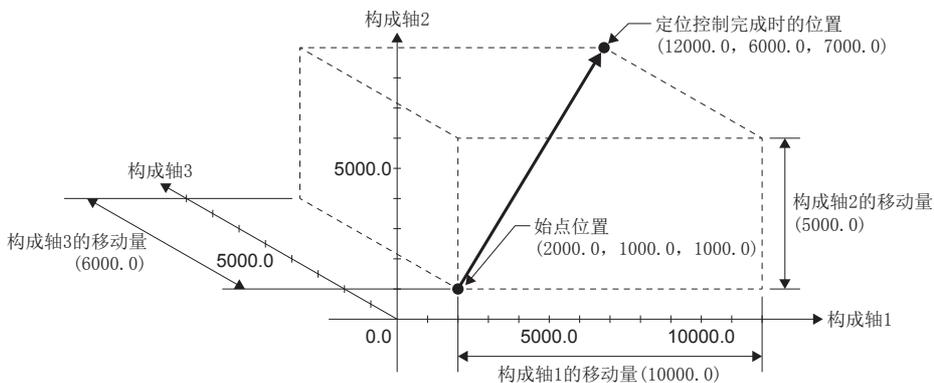
例

始点位置为“构成轴1: 2000.0, 构成轴2: 1000.0, 构成轴3: 1000.0”时,在移动量(Distance)中设置了“构成轴1: 10000.0, 构成轴2: 5000.0, 构成轴3: 6000.0”的情况下

```

LinearAxes[0]:= 1
LinearAxes[1]:= 2
LinearAxes[2]:= 3
Distance[0]:= 10000.0
Distance[1]:= 5000.0
Distance[2]:= 6000.0

```



■直线插补轴(LinearAxes)

在直线插补控制中，从轴组中设置的构成轴中，使用任意的轴进行直线插补。

从轴组的构成轴中，通过直线插补轴(LinearAxes)设置进行直线插补的构成轴。直线插补轴(LinearAxes)具有16个数组元素。在数组中，将插补控制中使用的构成轴的索引编号(1~16: 构成轴1~16)仅以插补控制中使用的构成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构成轴数不大于构成轴中登录的轴数。
- 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元素的构成轴将成为基准轴。
- 进行了以下设置的情况下，将变为“直线插补轴设置不正确(出错代码: 1AB9H)”且不启动。
 - 设置了轴未设置的构成轴的情况下
 - 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

将构成轴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 “LinearAxes[3]:= 0;~LinearAxes[15]:= 0;”可以省略。

■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。

- Distance[0..15]表示构成轴1~16的移动量。对直线插补轴(LinearAxes)中设置的构成轴设置位置。
- 未设置为直线插补轴的构成轴的移动量(Distance)将被忽略。
- 设置为直线插补轴的所有构成轴后面的移动量(Distance)可以省略。

例

将构成轴2、3、4设置为直线插补轴，并在构成轴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]:= 2000.0;  
Position[1]:= 3000.0;  
Position[2]:= 4000.0;  
Position[3]:= 0.0; *1  
:  
Position[15]:= 0.0; *1
```

*1 “LinearAxes[3]:= 0;~LinearAxes[15]:= 0;”与“Position[3]:= 0.0;~Position[15]:= 0.0;”可以省略。

■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(出错代码: 1AE4H)”(速度更改时为“超出运算周期换算速度范围警告(警告代码: 0D2FH)”)。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)使运算周期换算后的速度不小于“0.00001”。
- *2 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]为止的正数
1: 加减速时间恒定方式(mcFixedTime)	0.00000 ^{*1} 、0.000001~8400.0[s]为止的正数

- *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]为止的正数
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

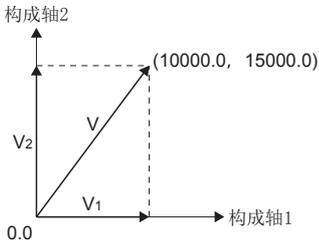
设置MCv_MoveLinearInterpolateRelative(相对值直线插补控制)中的Jerk。

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]为止的正数
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元素中设置的构成轴。

设置值	内容
0: 合成速度 (VectorSpeed)	<p>速度 (Velocity) 设置为合成速度。</p> <p>对于各轴的定位速度 (V_n), 运动系统根据设置的控制对象的定位速度 (V), 通过各轴的移动量 (D_n) 计算。</p> <p><例></p> <p>2轴的直线插补控制的情况下</p>  <p>构成轴2</p> <p>构成轴1</p> <p>0.0</p> <p>(10000.0, 15000.0)</p> <p>V</p> <p>V_2</p> <p>V_1</p> <ul style="list-style-type: none"> • 构成轴1的移动量 (D_1): 10000.0 [pulse] • 构成轴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"> • 构成轴1的定位速度: $V_1 = V \times D_1 / \sqrt{D_1^2 + D_2^2}$ • 构成轴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。</p> <p>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。对于其它插补轴的定位速度 (V_n), 运动系统通过各插补轴的移动量 (D_n) 计算。</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"> • 构成轴1的移动量 (D_1): 10000.0 [pulse] • 构成轴2的移动量 (D_2): 15000.0 [pulse] • 构成轴3的移动量 (D_3): 5000.0 [pulse] • 构成轴4的移动量 (D_4): 20000.0 [pulse] • 构成轴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构成轴4, 并以长轴速度控制构成轴4。对于其它构成轴的定位速度, 运动系统通过下述计算公式计算。</p> <ul style="list-style-type: none"> • 构成轴1的定位速度: $V_1 = D_1 / D_4 \times V$ • 构成轴2的定位速度: $V_2 = D_2 / D_4 \times V$ • 构成轴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: 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 (D_n) 计算其它插补轴的定位速度 (V_n) 并进行控制。作为插补轴直线插补轴 (LinearAxes) 的第1元素中设置的轴将变为基准轴。设置为基准轴的构成轴的移动量为“0.0”的情况下, 将变为“基准轴移动量0 (出错代码: 1AABH)”。</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"> • 构成轴1的移动量 (D_1): 10000.0 [pulse] • 构成轴2的移动量 (D_2): 15000.0 [pulse] • 构成轴3的移动量 (D_3): 5000.0 [pulse] • 构成轴4的移动量 (D_4): 20000.0 [pulse] • 基准轴速度 (V): 7000.0 [pulse/s] <p>上述的情况下, 基准轴将为构成轴4, 并以构成轴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"> • 构成轴1的定位速度: $V_1 = D_1 / D_4 \times V$ • 构成轴2的定位速度: $V_2 = D_2 / D_4 \times V$ • 构成轴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。

- 在速度模式(VelocityMode)中设置了“0: 合成速度(VectorSpeed)”时, 各轴的移动量超出“4, 294, 967, 296.0(=2³²)”的情况下, 将变为“超出直线移动量范围(出错代码: 1ADEH)”且不启动。

■缓冲模式(BufferMode)

设置用于进行多重启动(缓冲模式)的动作。

MCv_MoveLinearInterpolateRelative(相对值直线插补控制)中可设置的缓冲模式如下所示。

设置值	内容
0: Aborting(mcAborting)	中断(取消)执行中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在执行中的FB中缓冲下一个FB。 执行中的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) 执行中的FB完成时, 依次执行缓冲FB。
2: BlendingLow(mcBlendingLow)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的FB到达了目标位置时, 将执行中的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)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的FB到达了目标位置时, 将执行中的FB的目标速度作为切换速度。
4: BlendingNext(mcBlendingNext)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的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)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的FB到达了目标位置时, 将执行中的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 在执行中的FB与缓冲FB之间不执行停止。

要点

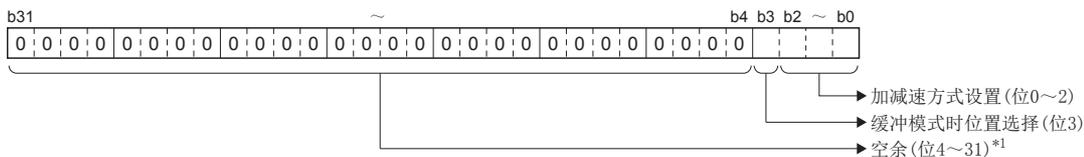
关于多重启动(缓冲模式)的详细内容, 请参阅下述手册。

MESE iQ-R运动模块用户手册(应用篇)

■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(出错代码: 1A4EH)”。

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)
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。 <ul style="list-style-type: none"> 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)

设置值	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)

设置值	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“0000000H(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。</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>

■必要对象

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构成轴设置下述从对象。

- Target position(607AH)

存在有未设置从对象的构成轴的情况下,将变为“必要从对象未设置(出错代码:1AA8H)”且不启动。

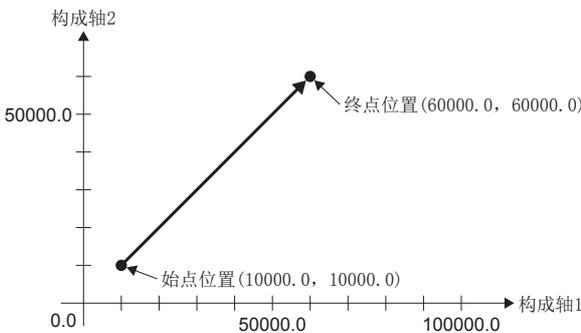
关于与从对象设置有关的详细内容,请参阅下述手册。

📖MELSEC iQ-R运动模块用户手册(应用篇)

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



• 设置

项目	设置值	
	构成轴1	构成轴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

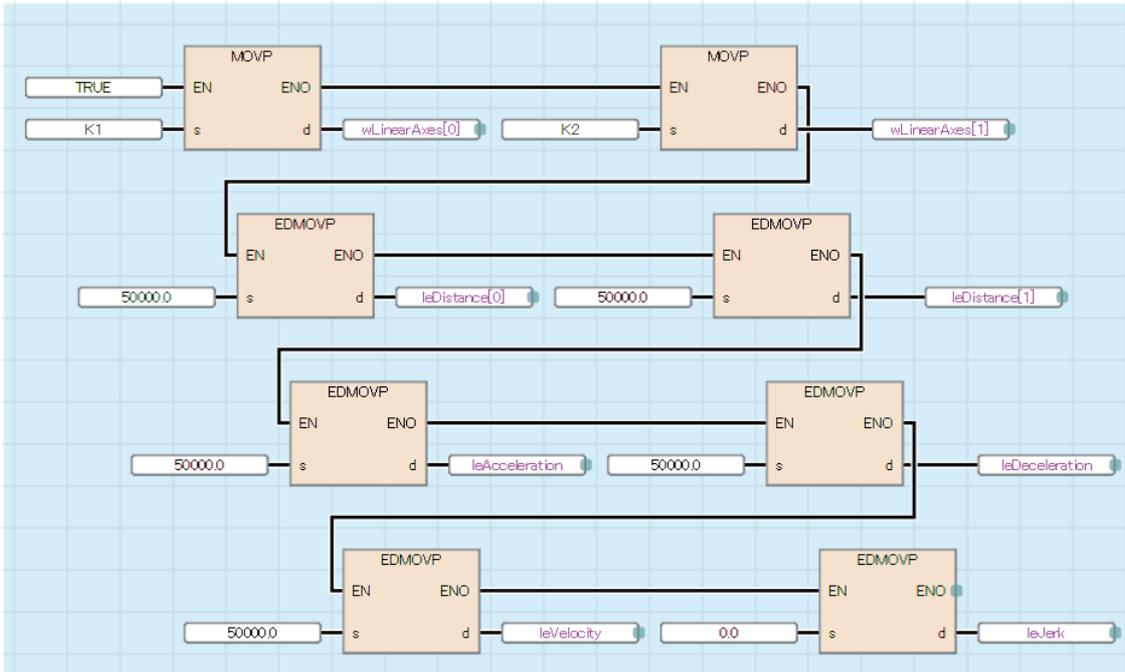
轴组No.	标签名	数据类型	注释
1	AxesGroup001	AXES_GROUP_REF	轴组1

■使用的标签

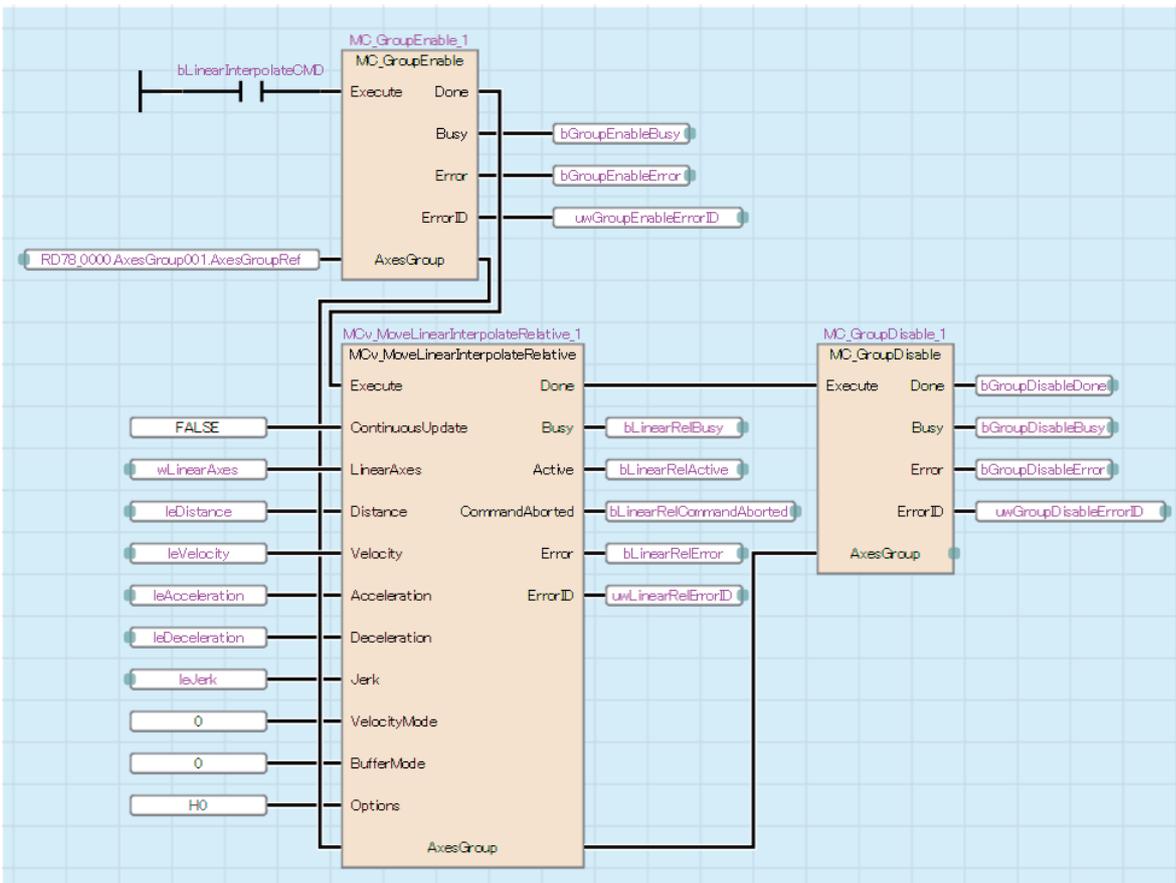
标签名	数据类型	注释
MCv_MoveLinearInterpolateRelative_1	MCv_MoveLinearInterpolateRelative	相对值直线插补控制FB
wLinearAxes	字[有符号](0..15)	直线插补轴
leDistance	双精度实数	移动量
leVelocity	双精度实数	速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bLinearRelDone	位	执行完成
bLinearRelBusy	位	执行中
bLinearRelActive	位	控制中
bLinearCommandAborted	位	执行中断
bLinearRelError	位	出错
uwLinearRelErrorID	字[无符号]/位串[16位]	出错代码
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	出错
uwGroupEnableErrorID	字[无符号]/位串[16位]	出错代码
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	出错
uwGroupDisableErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

- 2轴直线插补控制用数据设置



- 轴组有效/2轴直线插补控制/轴组无效



■ST的程序(运动模块侧)

```
//-----2轴直线插补控制用数据设置-----
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;

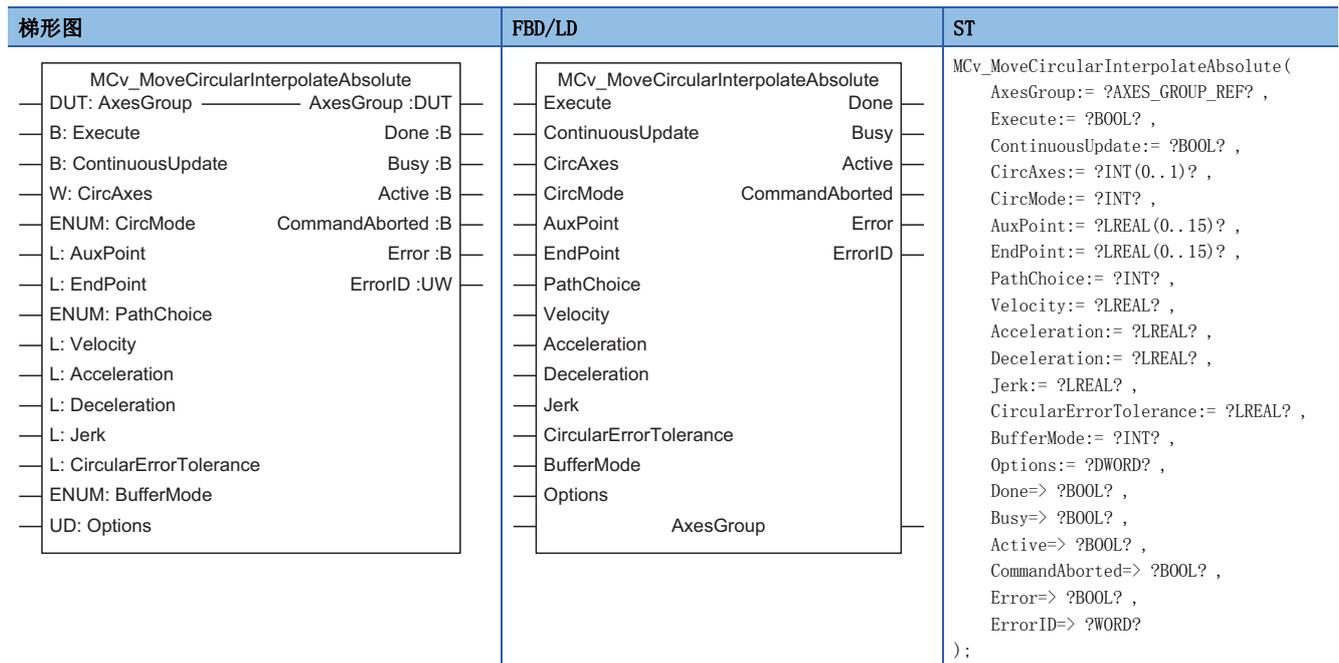
//-----轴组有效-----
MC_GroupEnable_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bLinearInterpolateCMD,
    Done=> bGroupEnableDone,
    Busy=> bGroupEnableBusy ,
    Error=> bGroupEnableError ,
    ErrorID=> uwGroupEnableErrorID
);

//-----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
);
```

MCv_MoveCircularInterpolateAbsolute

使用设置的轴组的构成轴，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。



名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
绝对值圆弧插补控制	328	8	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 44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_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	从构成轴中设置圆弧插补控制中使用的轴。 以数组设置构成轴的索引编号(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。 ☞ 267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。 ☞ 267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	<p>■ “0: 边界点指定 (mcBorder)”、 “1: 中心点指定 (mcCenter)” 的情况下: -10000000000.0 ~10000000000.0</p> <p>■ “2: 半径指定 (mcRadius)” 的情况下: 0.000001~ 2147483647.0</p>	0.0	<p>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。</p> <p>■ “0: 边界点指定 (mcBorder)”、“1: 中心点指定 (mcCenter)” 的情况下 作为构成轴1~16的绝对位置处理。设置边界点、中心点的绝对位置。</p> <p>■ “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。 ☞ 269页 辅助点(AuxPoint)</p>
EndPoint	终点	LREAL[0..15]	启动时	-10000000000.0 ~10000000000.0	0.0	<p>设置终点的位置。 是1维的数组数据。作为构成轴1~16的绝对位置处理。</p> <p>关于详细内容, 请参阅下述章节。 ☞ 270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。 *: 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。 ☞ 270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。 关于详细内容, 请参阅下述章节。 ☞ 271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。 关于详细内容, 请参阅下述章节。 ☞ 271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。 关于详细内容, 请参阅下述章节。 ☞ 271页 减速度(Deceleration)</p>
Jerk	Jerk	LREAL	启动时	0.0000、0.0001 ~2147483647.0	0.0	<p>设置Jerk。 关于详细内容, 请参阅下述章节。 ☞ 272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。 关于详细内容, 请参阅下述章节。 ☞ 272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。 ☞ 272页 缓冲模式(BufferMode)</p>
Options	选项	DWORD (HEX)	启动时	00000000H~ 00000001H	00000000H	<p>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。 ☞ 273页 选项(Options)</p>

■输出变量

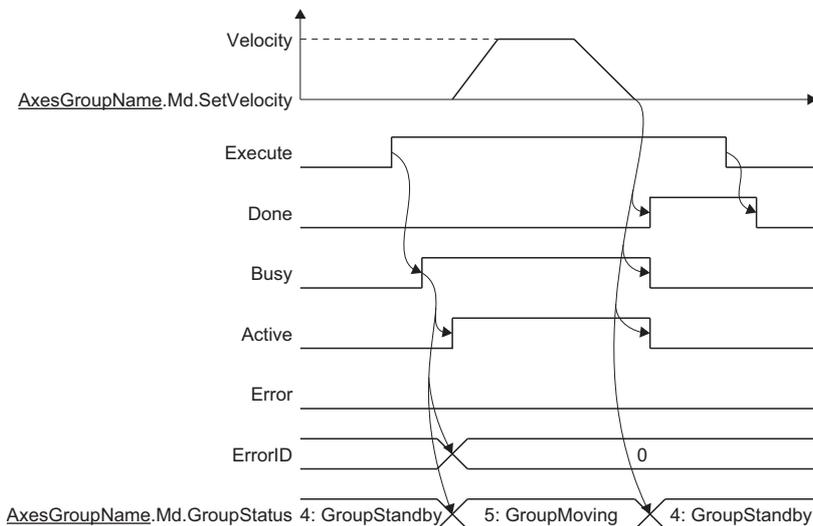
输出变量	名称	数据类型	初始值	内容
Done	完成	BOOL	FALSE	控制完成时，将变为TRUE。 动作完成时根据启动(Execute)的状态将变为如下所示。 ■启动(Execute)为TRUE的情况下 在将启动(Execute)置为FALSE之前将保持为TRUE不变。 ■启动(Execute)为FALSE的情况下 仅1周期为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_MoveCircularInterpolateAbsolute(绝对值圆弧插补控制)时，将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_MoveCircularInterpolateAbsolute(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	由于出错及多重启动等，MCv_MoveCircularInterpolateAbsolute(绝对值圆弧插补控制)的执行中断时，将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD(UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 □□MELSEC iQ-R运动模块用户手册(应用篇)

功能

- 圆弧插补控制时，指定轴组，对机械正交排列的直线轴进行插补控制，使从始点(移动开始点)至终点的轨迹成为圆弧。
- 在圆弧插补控制中，可以从轴组中设置的构成轴中使用任意的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。

☞ 11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，从轴组中设置的构成轴中，使用任意的2轴进行圆弧插补。

从轴组的构成轴中，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构成轴的索引编号(1~16：构成轴1~16)。

数组的第1元素将变为基准轴。

指定了未设置轴的构成轴的情况下将变为“圆弧插补轴未设置(出错代码：1A62H)”。

例

将构成轴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)。

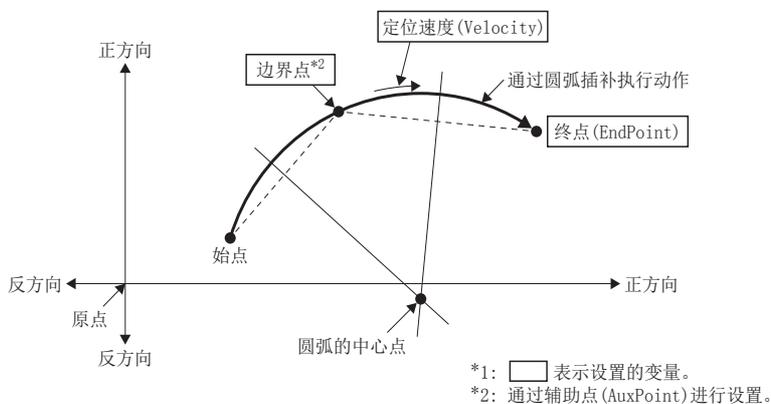
根据圆弧插补模式(CircMode)，辅助点(AuxPoint)的设置内容有所不同。

设置值	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轴的构成轴，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轴的构成轴，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轴的构成轴，以具有设置的半径的圆弧的轨迹进行定位控制。

通过设置的圆弧插补模式(CircMode)的圆弧插补控制的动作如下所示。

- 通过“0: 边界点指定(mcBorder)”的圆弧插补控制的动作

在边界点指定的绝对值圆弧插补控制中，从启动时的当前位置(始点地址)开始，向终点(EndPoint)中设置的地址(终点地址)，以通过辅助点(AuxPoint)中设置的地址(边界点地址)的圆弧的轨迹进行定位。控制的轨迹为，启动时的当前位置与边界点地址，边界点地址与终点地址的垂直二等分线的交点为中心的圆弧。但是，边界点指定的情况下，不绘制正圆。



限制事项

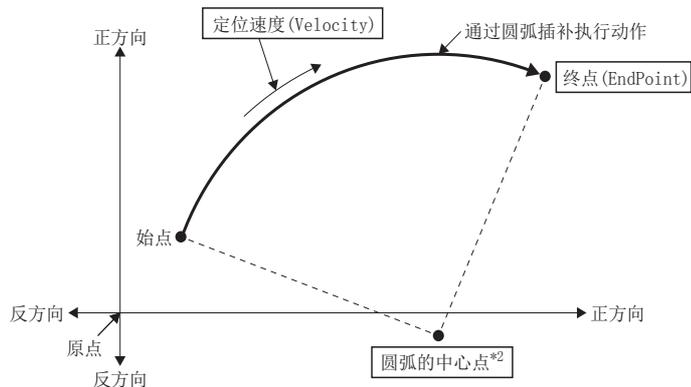
下述的情况下，无法设置通过边界点指定的2轴的圆弧插补控制。

- 半径超出2轴圆弧插补控制中允许的最大半径“2147483647.0”的情况下。(超出半径范围(出错代码：1A6CH))
- 边界点地址超出定位范围的情况下。(超出边界点地址范围(出错代码：1A64H))
- 终点地址超出定位范围的情况下。(超出终点地址范围(出错代码：1A6DH))
- 始点地址、边界点地址、终点地址位于同一直线上的情况下。(始点-边界点-终点地址同一直线(出错代码：1A6AH))
- 始点地址=终点地址的情况下。(始点-终点地址同一值(出错代码：1A66H))
- 始点地址=边界点地址的情况下。(始点-边界点地址同一值(出错代码：1A68H))
- 终点地址=边界点地址的情况下。(终点-边界点地址同一值(出错代码：1A69H))

• 通过“1: 中心点指定(mcCenter)”的圆弧插补控制的动作

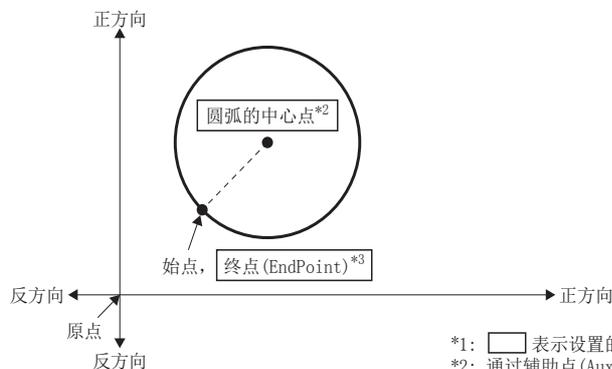
在中心点指定的绝对值圆弧插补控制中，从启动时的当前位置(始点地址)开始，向终点(EndPoint)中设置的终点位置，以将辅助点(AuxPoint)中设置的中心点作为中心的圆弧的轨迹进行定位。对于中心点指定的绝对值圆弧插补控制中的轨迹的路径，通过路径选择(PathChoice)进行设置。关于详细内容，请参阅下述章节。

☞ 270页 路径选择(PathChoice)



*1: 表示设置的变量。
*2: 通过辅助点(AuxPoint)进行设置。

• 将终点(EndPoint)设置为与始点相同时，进行将始点与圆弧的中心点作为半径的正圆的定位。通过将终点(EndPoint)中设置的终点的位置设置为与始点相同的值，始点与终点将变为相同。



*1: 表示设置的变量。
*2: 通过辅助点(AuxPoint)进行设置。
*3: 设置与始点相同的值。

<例>

```
将终点(EndPoint)设置为与始点相同的情况下
AxesGroup001.Pr.Axis[1]:= Axis0001.AxisRef;
AxesGroup001.Pr.Axis[2]:= Axis0002.AxisRef;
EndPoint[0]:= Axis0001.Md.SetPosition;
EndPoint[1]:= Axis0002.Md.SetPosition;
```

如上述所示，将圆弧插补中使用的构成轴的指令当前位置设置为终点(EndPoint)。

限制事项

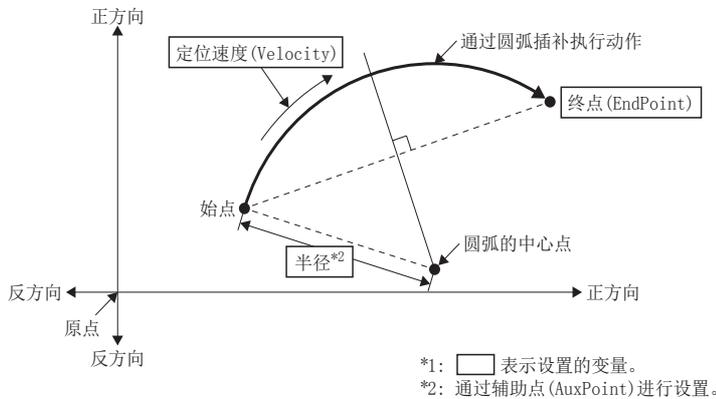
下述的情况下，无法设置通过中心点指定的2轴的圆弧插补控制。

- 半径超出2轴圆弧插补控制中允许的最大半径“2147483647.0”的情况下。(超出半径范围(出错代码: 1A6CH))
- 中心点地址超出定位范围的情况下。(超出中心点地址范围(出错代码: 1A6BH))
- 始点地址=中心点地址的情况下。(始点-中心点地址同一值(出错代码: 1A65H))
- 终点地址=中心点地址的情况下。(终点-中心点地址同一值(出错代码: 1A67H))

• 通过“2: 半径指定(mcRadius)”的圆弧插补控制的动作

在半径指定的绝对值圆弧插补控制中, 从启动时的当前位置(始点地址)开始, 向终点(EndPoint)中设置的地址(终点地址), 以具有辅助点(AuxPoint)中设置的半径的轨迹进行定位。控制的轨迹为, 以启动时的当前位置与终点地址的垂直二等分线与设置的半径的交点为中心点的圆弧。对于半径指定的绝对值圆弧插补控制中的轨迹的路径, 通过路径选择(PathChoice)进行设置。关于详细内容, 请参阅下述章节。

☞ 270页 路径选择(PathChoice)



限制事项

下述的情况下, 无法设置通过半径指定的2轴的圆弧插补控制。

- 半径超出2轴圆弧插补控制中允许的最大半径“2147483647.0”的情况下。(超出半径范围(出错代码: 1A6CH))
- 终点地址超出定位范围的情况下。(超出终点地址范围(出错代码: 1A6DH))
- 始点地址=终点地址的情况下。(始点-终点地址同一值(出错代码: 1A66H))
- 始点地址与终点地址的距离大于半径的情况下。(半径设置出错(出错代码: 1A6EH))

■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)	$-10000000000.0 \leq \text{设置值} < 10000000000.0^{*1}$	辅助点(AuxPoint)设置连接始点与终点的圆弧上的边界点。 AuxPoint[0..15]表示构成轴1~16的辅助点位置。 • 未设置为圆弧插补轴的构成轴的辅助点(AuxPoint)将被忽略。 • 设置为圆弧插补轴的所有构成轴后面的辅助点(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]表示构成轴1~16的辅助点位置。 • 未设置为圆弧插补轴的构成轴的辅助点(AuxPoint)将被忽略。 • 设置为圆弧插补轴的所有构成轴后面的辅助点(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

将构成轴2、3设置为圆弧插补轴, 并在构成轴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]:= 2000.0;
AuxPoint[1]:= 3000.0;
AuxPoint[2]:= 0.0; *2
:
AuxPoint[15]:= 0.0; *2
```

*2 “AuxPoint[2]:= 0.0;~AuxPoint[15]:= 0.0;”可以省略。

■ 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]表示构成轴1~16的终点位置。

- 未设置为圆弧插补轴的构成轴的终点 (EndPoint) 将被忽略。
- 设置为圆弧插补轴的所有构成轴后面的终点 (EndPoint) 可以省略。

终点 (EndPoint) 应在下述范围内进行设置。

设置范围

$-10000000000.0 \leq \text{设置值} < 10000000000.0^{*1}$

*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。

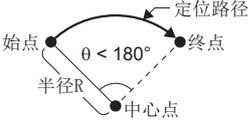
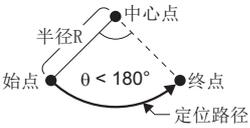
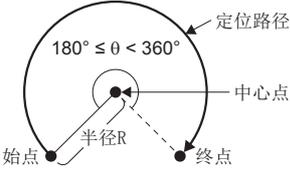
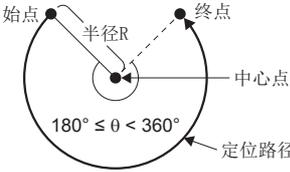
对于路径选择 (PathChoice)，在圆弧插补模式 (CircMode) 中设置了“1: 中心点指定 (mcCenter)”或“2: 半径指定 (mcRadius)”的情况下进行设置。设置了“0: 边界点指定 (mcBorder)”的情况下，将忽略输入。

通过设置的圆弧插补模式 (CircMode) 的路径选择 (PathChoice) 的设置值、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

设置值	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 • 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 (出错代码: 1AB7H)”。	$0^\circ < \theta \leq 180^\circ$	
3: 就远 (mcLongWay)	就远 • 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

设置值	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(出错代码: 1AB7H)”。	$0^\circ < \theta < 180^\circ$	
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(出错代码: 1AB7H)”。	$0^\circ < \theta < 180^\circ$	
4: CW就近 (mcCWLongWay)	CW就近 • 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就近 (mcCCWLongWay)	CCW就近 • 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_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(出错代码: 1AE4H)”(速度更改时为“超出运算周期换算速度范围警告(警告代码: 0D2FH)”)。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)使运算周期换算后的速度不小于“0.00001”。

- *2 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]为止的正数
1: 加减速时间恒定方式(mcFixedTime)	0.000000 ^{*1} 、0.000001~8400.0[s]为止的正数

- *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]为止的正数
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。

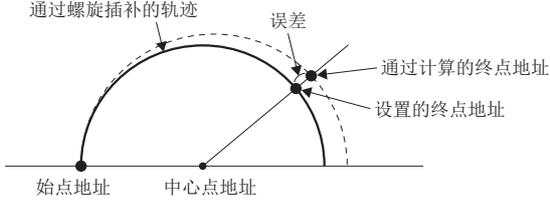
设置了加减速方式设置(选项(Options): 位0~2)的“0: 加减速速度指定方式(mcAccDec)”的情况下, 对设置范围进行设置。设置了“1: 加减速时间恒定方式(mcFixedTime)”的情况下, 不使用Jerk。

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]为止的正数
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)	定位启动时将变为“圆弧插补误差允许值溢出(出错代码: 1A71H)”且不启动。定位控制中的情况下, 检测到出错时将立即停止。

■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。

设置值	内容
0: Aborting(mcAborting)	中断(取消)执行中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在执行中的FB中缓冲下一个FB。 执行中的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) 执行中的FB完成时, 依次执行缓冲FB。
2: BlendingLow(mcBlendingLow)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的FB到达了目标位置时, 将执行中的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)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的FB到达了目标位置时, 将执行中的FB的目标速度作为切换速度。
4: BlendingNext(mcBlendingNext)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的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)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的FB到达了目标位置时, 将执行中的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 在执行中的FB与缓冲FB之间不执行停止。

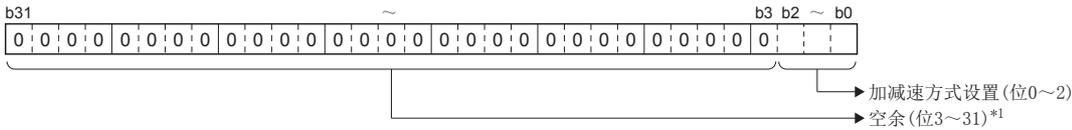
要点

关于多重启动(缓冲模式)的详细内容, 请参阅下述手册。

📖 MELSEC iQ-R运动模块用户手册(应用篇)

■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(出错代码: 1A4EH)”。

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)

设置值	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

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构成轴设置下述从对象。

• Target position(607AH)

存在有未设置从对象的构成轴的情况下, 将变为“必要从对象未设置(出错代码: 1AA8H)”且不启动。

关于与从对象设置有关的详细内容, 请参阅下述手册。

📖 MELSEC iQ-R运动模块用户手册(应用篇)

注意事项

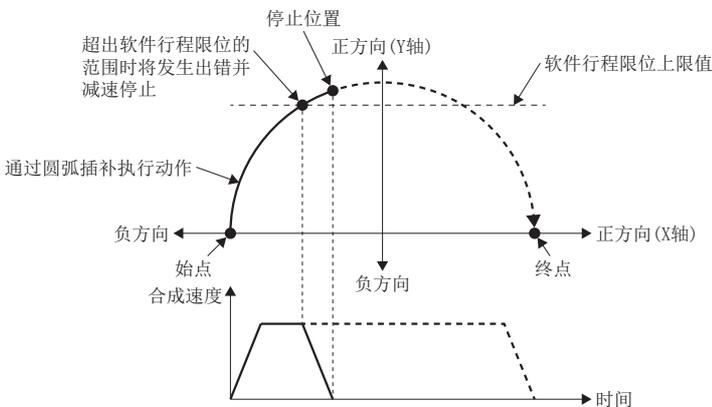
- 不能进行包含将行程限位设置为无效的轴的圆弧插补。否则将变为“圆弧插补时软件行程限位无效(出错代码: 1A72H)”且不启动。
- 插补动作中插补路径超出行程限位范围的情况下, 将变为“软件行程限位溢出(正方向)(出错代码: 1A03H)”或“软件行程限位溢出(负方向)(出错代码: 1A04H)”且停止运行。

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, 在发生出错的同时执行停止。

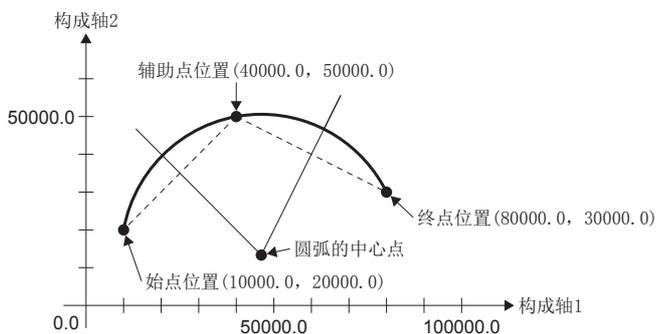
减速停止的情况下, 将如下图所示沿着圆弧的轨迹进行减速, 并停止。



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



• 设置

项目	设置值	
	构成轴1	构成轴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

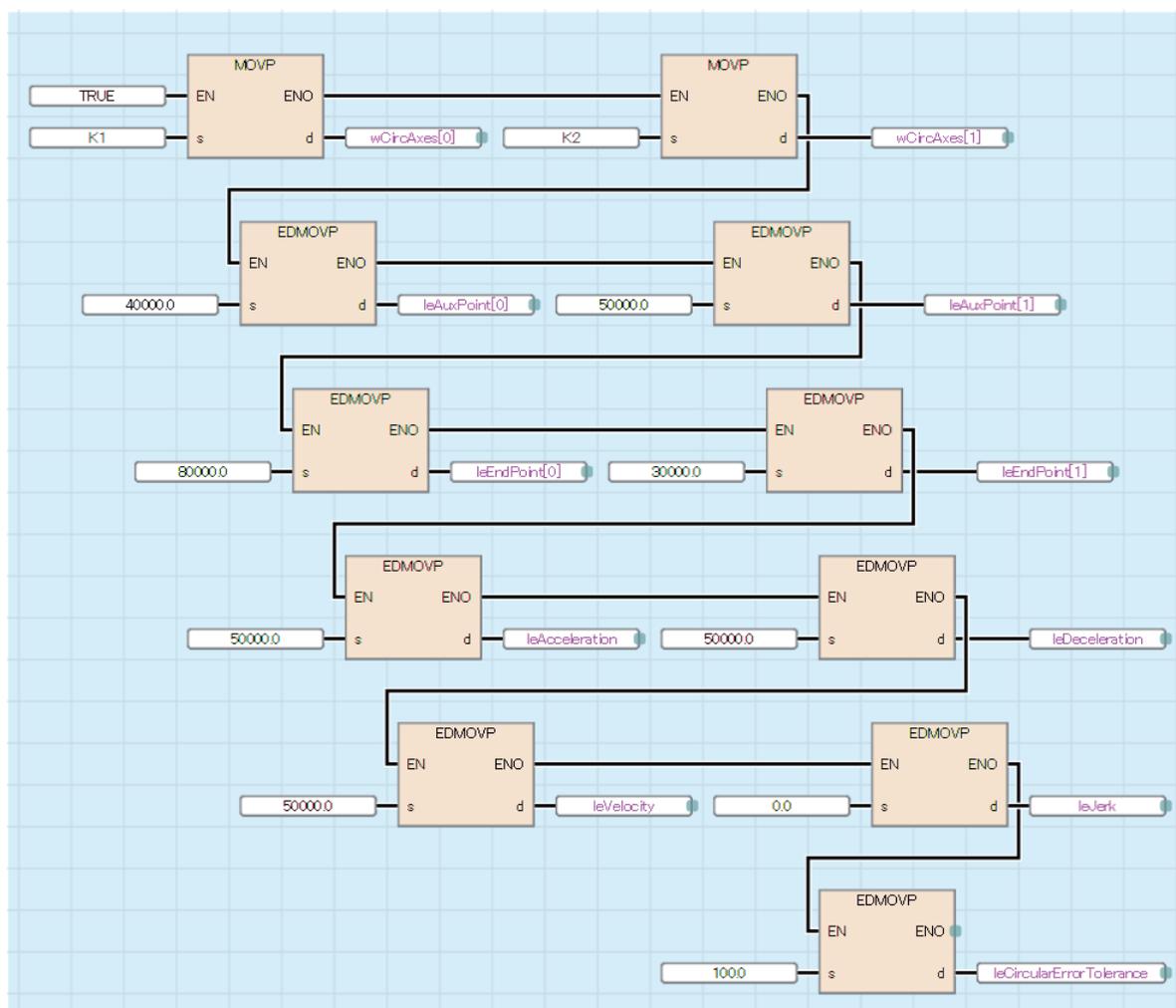
■ 使用的标签

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	出错
uwCircularAbsErrorID	字[无符号]/位串[16位]	出错代码
MC_GroupEnable_1	MC_GroupEnable	轴组有效FB
bCircularInterpolateCMD	位	圆弧插补控制启动

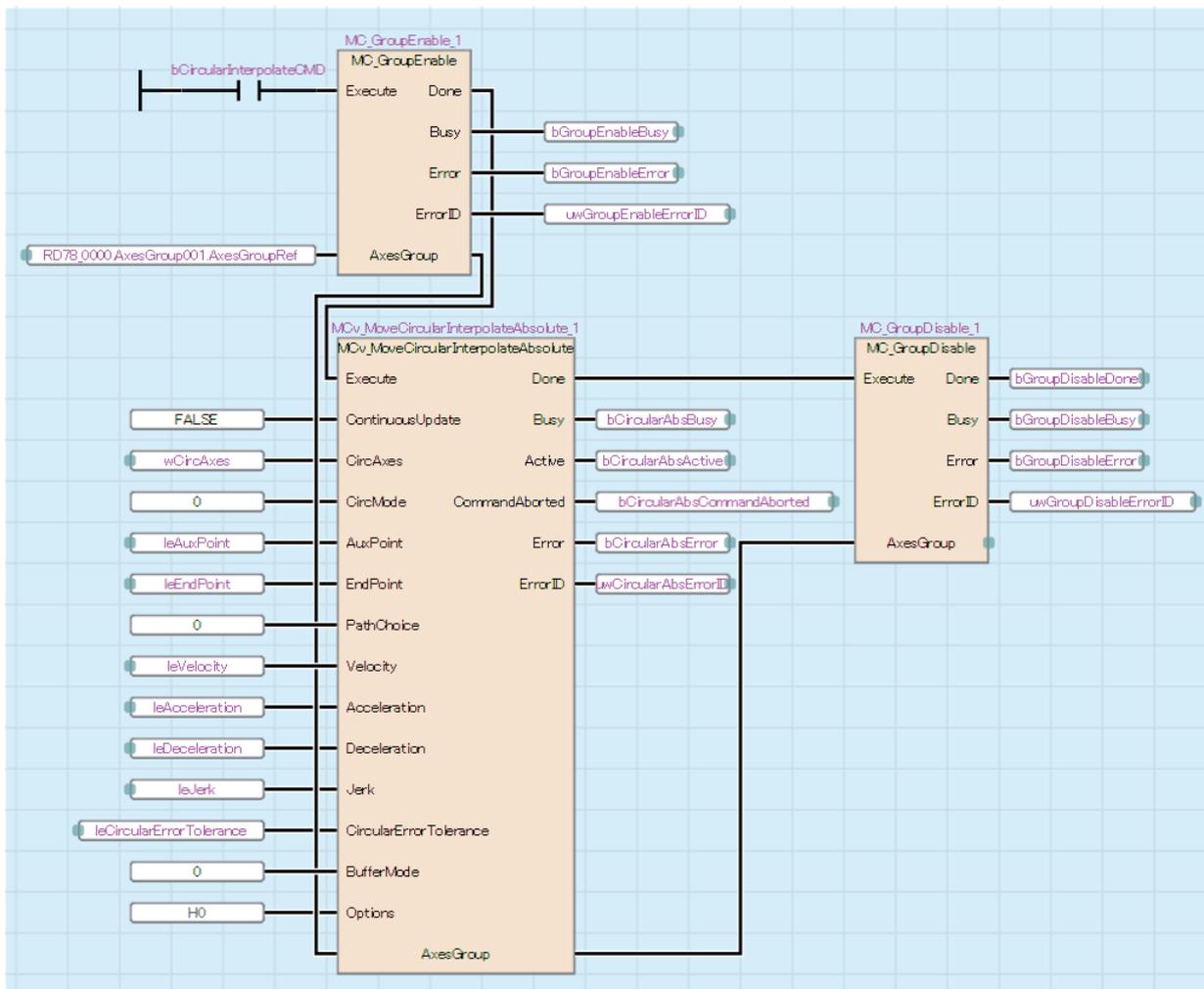
标签名	数据类型	注释
bGroupEnableDone	位	轴组有效完成
bGroupEnableBusy	位	执行中
bGroupEnableError	位	出错
uwGroupEnableErrorID	字[无符号]/位串[16位]	出错代码
MC_GroupDisable_1	MC_GroupDisable	轴组无效FB
bGroupDisableDone	位	轴组无效完成
bGroupDisableBusy	位	执行中
bGroupDisableError	位	出错
uwGroupDisableErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

- 2轴圆弧插补控制用数据设置



- 轴组有效/圆弧插补控制(边界点指定)/轴组无效



■ST的程序(运动模块侧)

//-----2轴圆弧插补控制用数据设置-----

```
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
);
```

```

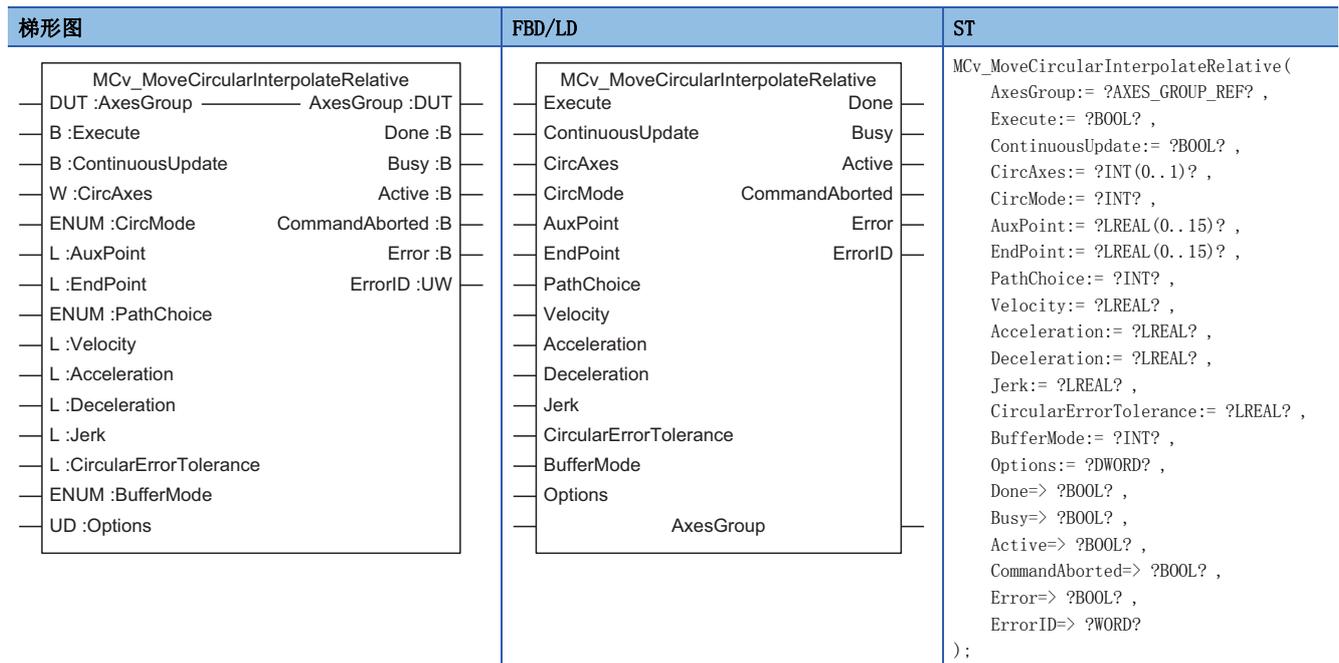
//-----圆弧插补控制(边界点指定)-----
MCv_MoveCircularInterpolateAbsolute_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bGroupEnableDone ,
    ContinuousUpdate:= FALSE ,
    CircAxes:= wCircAxes ,
    CircMode:= MC_CIRC_MODE__mcBorder ,
    AuxPoint:= leAuxPoint ,
    EndPoint:= leEndPoint ,
    PathChoice:= 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
);

```

MCv_MoveCircularInterpolateRelative

使用设置的轴组的构成轴，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。



名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
相对值圆弧插补控制	328	8	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 44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	从构成轴中设置圆弧插补控制中使用的轴。 以数组设置构成轴的索引编号(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。 ☞ 281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。 ☞ 281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	<p>■ “0: 边界点指定 (mcBorder)”、 “1: 中心点指定 (mcCenter)” 的情况下: -10000000000.0 ~10000000000.0</p> <p>■ “2: 半径指定 (mcRadius)” 的情况下: 0.000001~ 2147483647.0</p>	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。 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> <p>■ “0: 边界点指定 (mcBorder)”、“1: 中心点指定 (mcCenter)” 的情况下 作为构成轴1~16的相对位置处理。设置边界点、中心点的绝对位置。</p> <p>■ “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。 ☞ 283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。 是1维的数组数据。作为构成轴1~16的绝对位置处理。</p> <p>关于详细内容, 请参阅下述章节。 ☞ 284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。 *: 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。 ☞ 284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。 关于详细内容, 请参阅下述章节。 ☞ 285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。 关于详细内容, 请参阅下述章节。 ☞ 285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。 关于详细内容, 请参阅下述章节。 ☞ 285页 减速度(Deceleration)</p>
Jerk	Jerk	LREAL	启动时	0.0000、0.0001 ~2147483647.0	0.0	<p>设置Jerk。 关于详细内容, 请参阅下述章节。 ☞ 286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。 关于详细内容, 请参阅下述章节。 ☞ 286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。 ☞ 286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。 关于详细内容, 请参阅下述章节。 ☞ 287页 选项(Options)</p>

■输出变量

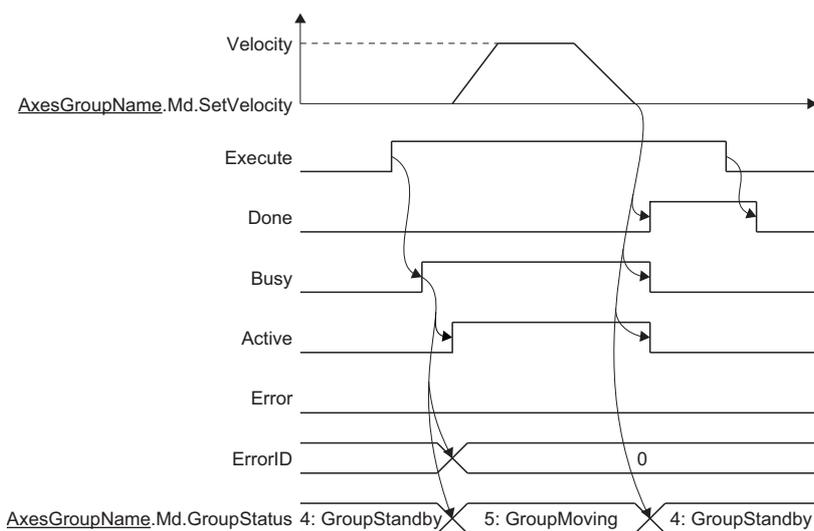
输出变量	名称	数据类型	初始值	内容
Done	完成	BOOL	FALSE	控制完成时，将变为TRUE。 动作完成时根据启动(Execute)的状态将变为如下所示。 ■启动(Execute)为TRUE的情况下 在将启动(Execute)置为FALSE之前将保持为TRUE不变。 ■启动(Execute)为FALSE的情况下 仅1周期为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)时，将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_MoveCircularInterpolateRelative(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	由于出错及多重启动等，MCv_MoveCircularInterpolateRelative(相对值圆弧插补控制)的执行中断时，将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD(UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 □□MELSEC iQ-R运动模块用户手册(应用篇)

功能

- 圆弧插补控制时，指定轴组，对机械正交排列的直线轴进行插补控制，使从始点(移动开始点)至终点的轨迹成为圆弧。
- 在圆弧插补控制中，可以从轴组中设置的构成轴中使用任意的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。

☞ 11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，从轴组中设置的构成轴中，使用任意的2轴进行圆弧插补。

从轴组的构成轴中，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构成轴的索引编号(1~16: 构成轴1~16)。

数组的第1元素将变为基准轴。

设置了未设置轴的构成轴的情况下，将变为“圆弧插补轴未设置(出错代码: 1A62H)”。

例

将构成轴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)。

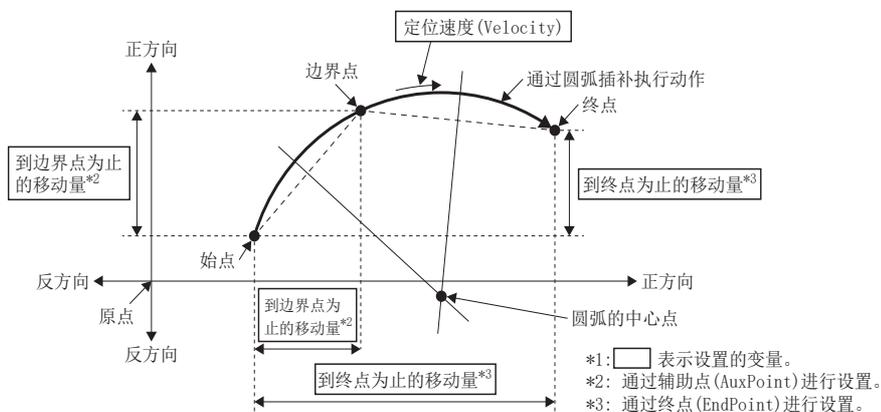
根据圆弧插补模式(CircMode)，辅助点(AuxPoint)的设置内容有所不同。

设置值	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轴的构成轴，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轴的构成轴，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轴的构成轴，以具有设置的半径的圆弧的轨迹进行定位控制。

通过设置的圆弧插补模式的圆弧插补控制的动作如下所示。

- 通过“0: 边界点指定(mcBorder)”的圆弧插补控制

在边界点指定的相对值圆弧插补控制中，从启动时的当前位置(始点地址)开始，向终点(EndPoint)中设置的移动量的地址(终点地址)，以通过辅助点(AuxPoint)中设置的移动量的地址(边界点地址)的圆弧的轨迹进行定位。控制的轨迹为，启动时的当前位置与边界点地址，边界点地址与终点地址的垂直二等分线的交点为中心的圆弧。



限制事项

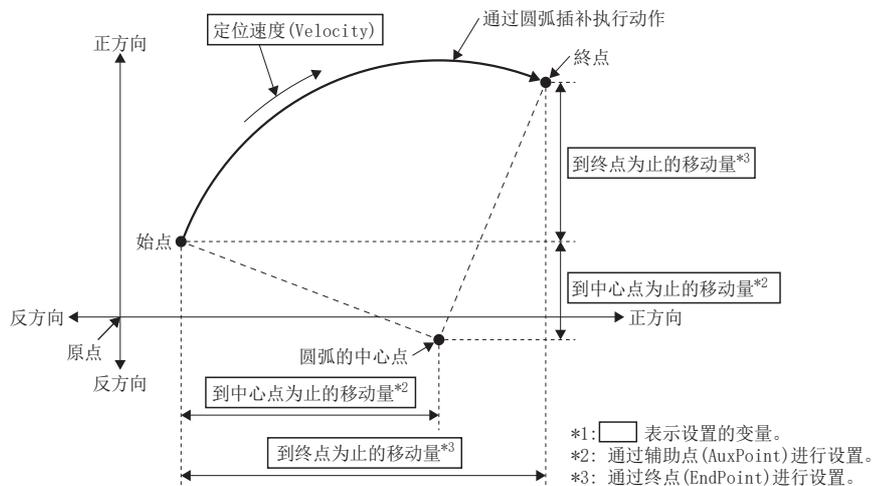
下述的情况下，无法设置通过边界点指定的2轴的圆弧插补控制。

- 半径超出2轴圆弧插补控制中允许的最大半径“2147483647.0”的情况下。(超出半径范围(出错代码: 1A6CH))
- 边界点地址超出定位范围的情况下。(超出边界点地址范围(出错代码: 1A64H))
- 终点地址超出定位范围的情况下。(超出终点地址范围(出错代码: 1A6DH))
- 始点地址、边界点地址、终点地址位于同一直线上的情况下。(始点-边界点-终点地址同一直线(出错代码: 1A6AH))
- 始点地址=终点地址的情况下。(始点-终点地址同一值(出错代码: 1A66H))
- 始点地址=边界点地址的情况下。(始点-边界点地址同一值(出错代码: 1A68H))
- 终点地址=边界点地址的情况下。(终点-边界点地址同一值(出错代码: 1A69H))

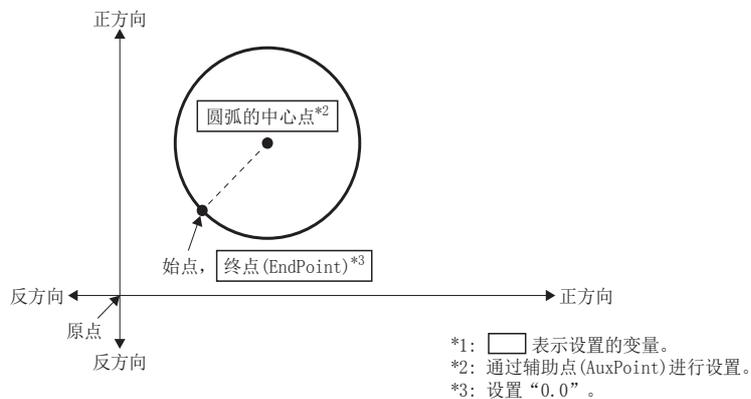
• 通过“1: 中心点指定(mcCenter)”的圆弧插补控制

在中心点指定的相对值圆弧插补控制中，从启动时的当前位置(始点地址)开始，向终点(EndPoint)中设置的移动量的地址(终点地址)，以将辅助点(AuxPoint)中设置的移动量的地址(中心点地址)作为中心的圆弧的轨迹进行定位。对于中心点指定的相对值圆弧插补控制中的轨迹的路径，通过路径选择(PathChoice)进行设置。关于详细内容，请参阅下述章节。

☞ 284页 路径选择(PathChoice)



• 通过将到终点(EndPoint)中设置的终点为止的移动量设置为“0.0”，终点与始点将变为相同，并进行将始点与圆弧的中心点作为半径的正圆的定位。



限制事项

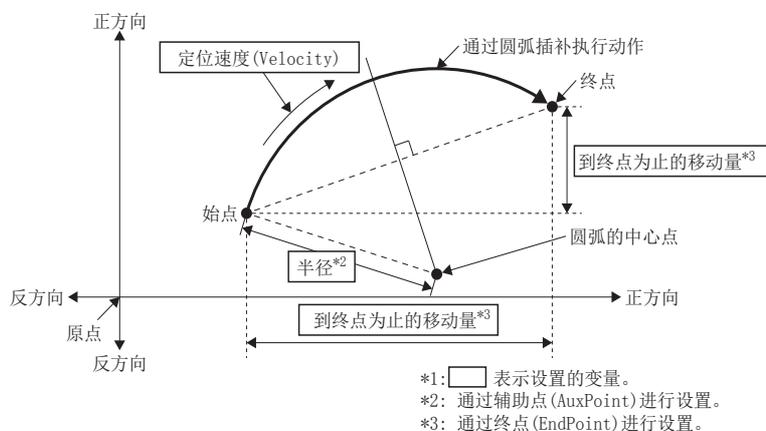
下述的情况下，无法设置通过中心点指定的2轴的圆弧插补控制。

- 半径超出2轴圆弧插补控制中允许的最大半径“2147483647.0”的情况下。(超出半径范围(出错代码: 1A6CH))
- 中心点地址超出定位范围的情况下。(超出中心点地址范围(出错代码: 1A6BH))
- 始点地址=中心点地址的情况下。(始点-中心点地址同一值(出错代码: 1A65H))
- 终点地址=中心点地址的情况下。(终点-中心点地址同一值(出错代码: 1A67H))

- 通过“2: 半径指定(mcRadius)”的圆弧插补控制的动作

在半径指定的相对值圆弧插补控制中, 从启动时的当前位置(始点地址)开始, 向终点(EndPoint)中设置的移动量的地址(终点地址), 以具有辅助点(AuxPoint)中设置的半径的轨迹进行定位。控制的轨迹为, 以启动时的当前位置与终点地址的垂直二等分线与设置的半径的交点为中心点的圆弧。对于半径指定的相对值圆弧插补控制中的轨迹的路径, 通过路径选择(PathChoice)进行设置。关于详细内容, 请参阅下述章节。

☞ 284页 路径选择(PathChoice)



限制事项

下述的情况下, 无法设置通过半径指定的2轴的圆弧插补控制。

- 半径超出2轴圆弧插补控制中允许的最大半径“2147483647.0”的情况下。(超出半径范围(出错代码: 1A6CH))
- 终点地址超出定位范围的情况下。(超出终点地址范围(出错代码: 1A6DH))
- 始点地址=终点地址的情况下。(始点-终点地址同一值(出错代码: 1A66H))
- 始点地址与终点地址的距离大于半径的情况下。(半径设置出错(出错代码: 1A6EH))

■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]表示构成轴1~16的辅助点位置。 • 未设置为圆弧插补轴的构成轴的辅助点(AuxPoint)将被忽略。 • 设置为圆弧插补轴的所有构成轴后面的辅助点(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]表示构成轴1~16的辅助点位置。 • 未设置为圆弧插补轴的构成轴的辅助点(AuxPoint)将被忽略。 • 设置为圆弧插补轴的所有构成轴后面的辅助点(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

将构成轴2、3设置为圆弧插补轴, 并在构成轴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]:= 2000.0;
```

```
AuxPoint[1]:= 3000.0;
```

```
AuxPoint[2]:= 0.0; *1
```

```
⋮
```

```
AuxPoint[15]:= 0.0; *1
```

*1 “AuxPoint[2]:= 0.0;~AuxPoint[15]:= 0.0;”可以省略。

■ 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]表示构成轴1~16的终点位置。

- 未设置为圆弧插补轴的构成轴的终点 (EndPoint) 将被忽略。
- 设置为圆弧插补轴的所有构成轴后面的终点 (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。

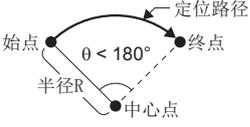
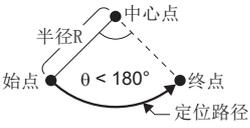
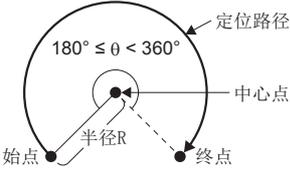
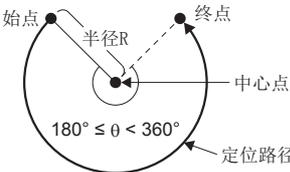
对于路径选择 (PathChoice)，在圆弧插补模式 (CircMode) 中设置了“1: 中心点指定 (mcCenter)”或“2: 半径指定 (mcRadius)”的情况下进行设置。设置了“0: 边界点指定 (mcBorder)”的情况下，将忽略输入。

通过设置的圆弧插补模式 (CircMode) 的路径选择 (PathChoice) 的设置值、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

设置值	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 • 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 (出错代码: 1AB7H)”。	$0^\circ < \theta \leq 180^\circ$	
3: 就远 (mcLongWay)	就远 • 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

设置值	设置内容	可控制的圆弧的中心角	路径
0: CW(mcCW)	CW • 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(出错代码: 1AB7H)”。	$0^\circ < \theta < 180^\circ$	
1: CCW(mcCCW)	CCW • 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(出错代码: 1AB7H)”。	$0^\circ < \theta < 180^\circ$	
4: CW就远 (mcCWLlongWay)	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。	$180^\circ \leq \theta < 360^\circ$	
5: CCW就远 (mcCCWLlongWay)	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。	$180^\circ \leq \theta < 360^\circ$	

■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(出错代码: 1AE4H)”(速度更改时为“超出运算周期换算速度范围警告(警告代码: 0D2FH)”)。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)使运算周期换算后的速度不小于“0.00001”。
- *2 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]为止的正数
1: 加减速时间恒定方式(mcFixedTime)	0.000000 ^{*1} 、0.000001~8400.0[s]为止的正数

- *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]为止的正数
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。

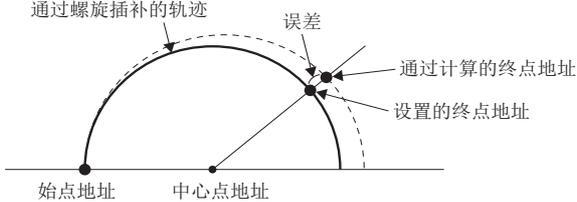
设置了加减速方式设置(选项(Options): 位0~2)的“0: 加减速速度指定方式(mcAccDec)”的情况下, 对设置范围进行设置。设置了“1: 加减速时间恒定方式(mcFixedTime)”的情况下, 不使用Jerk。

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]为止的正数
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)	定位启动时将变为“圆弧插补误差允许值溢出(出错代码: 1A71H)”且不启动。定位控制中的情况下, 检测到出错时将立即停止。

■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。

设置值	内容
0: Aborting(mcAborting)	中断(取消)执行中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在执行中的FB中缓冲下一个FB。 执行中的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) 执行中的FB完成时, 依次执行缓冲FB。
2: BlendingLow(mcBlendingLow)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的FB到达了目标位置时, 将执行中的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)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的FB到达了目标位置时, 将执行中的FB的目标速度作为切换速度。
4: BlendingNext(mcBlendingNext)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的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)	在执行中的FB中缓冲下一个FB。 ^{*1} 执行中的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) 执行中的FB到达目标位置后, 依次执行缓冲FB。 执行中的FB到达了目标位置时, 将执行中的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 在执行中的FB与缓冲FB之间不执行停止。

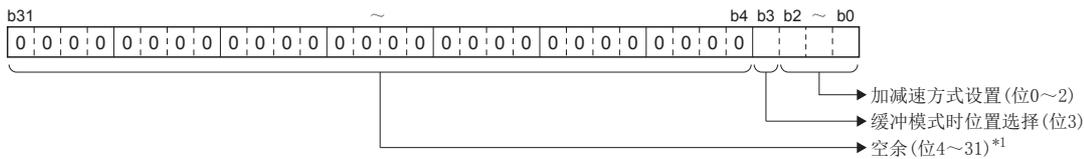
要点

关于多重启动(缓冲模式)的详细内容, 请参阅下述手册。

📖 MELSEC iQ-R运动模块用户手册(应用篇)

■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(出错代码: 1A4EH)”。

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)
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。 <ul style="list-style-type: none"> • 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)

设置值	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)

设置值	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

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构成轴设置下述从对象。

- Target position(607AH)

存在有未设置从对象的构成轴的情况下，将变为“必要从对象未设置(出错代码: 1AA8H)”且不启动。

关于与从对象设置有关的详细内容，请参阅下述手册。

📖MELSEC iQ-R运动模块用户手册(应用篇)

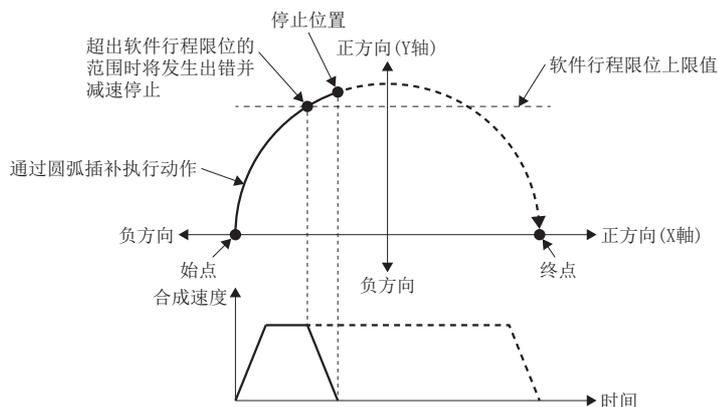
注意事项

- 不能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(出错代码: 1A72H)”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)(出错代码: 1A03H)”或“软件行程限位溢出(负方向)(出错代码: 1A04H)”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, 在发生出错的同时执行停止。

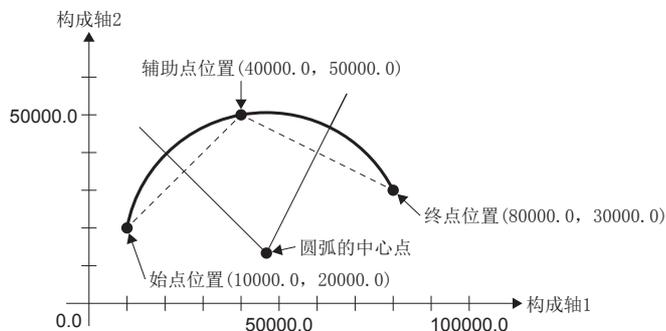
减速停止的情况下, 将如下图所示沿着圆弧的轨迹进行减速, 并停止。



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



• 设置

项目	设置值	
	构成轴1	构成轴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

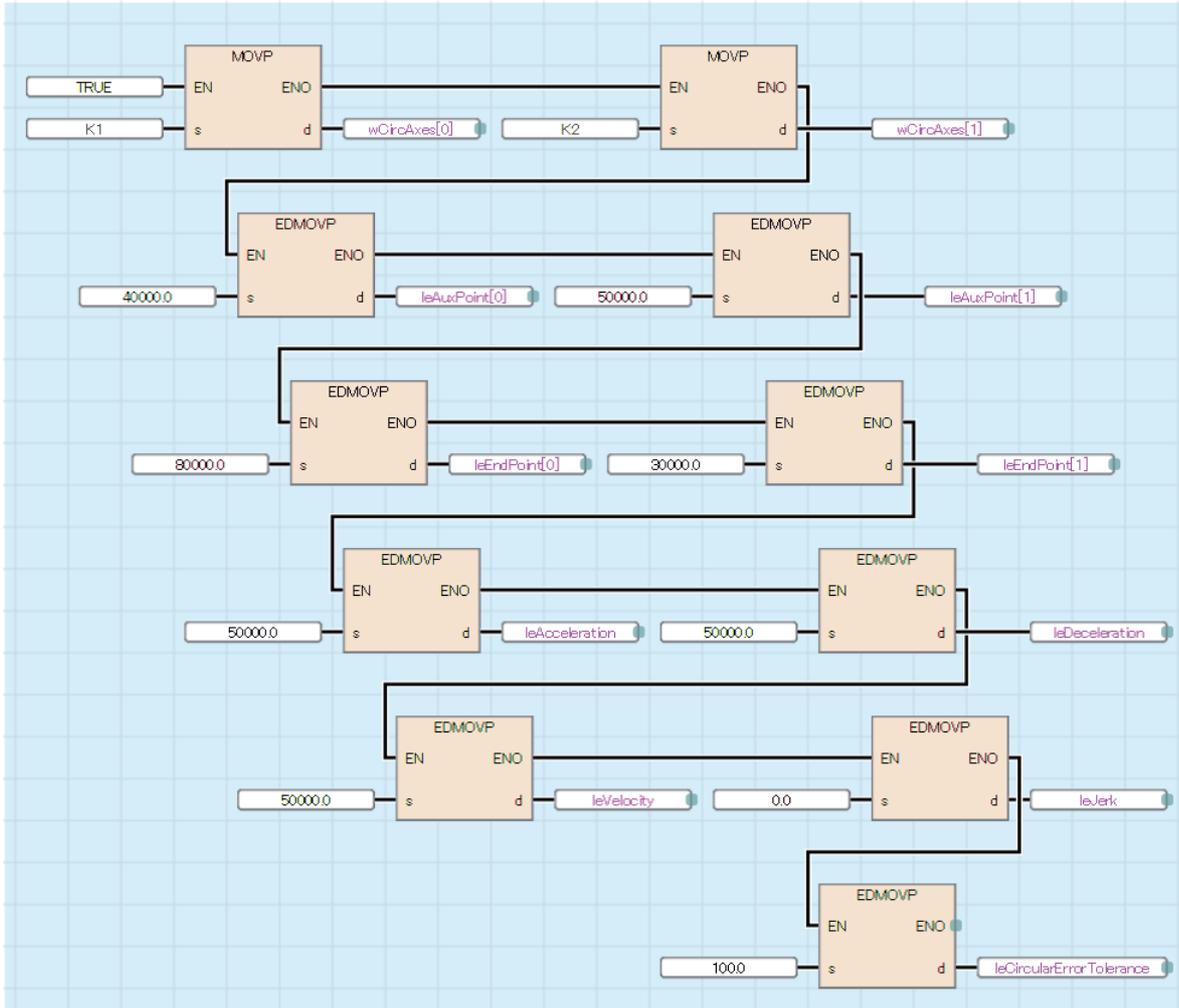
轴组No.	标签名	数据类型	注释
1	AxesGroup001	AXES_GROUP_REF	轴组1

■使用的标签

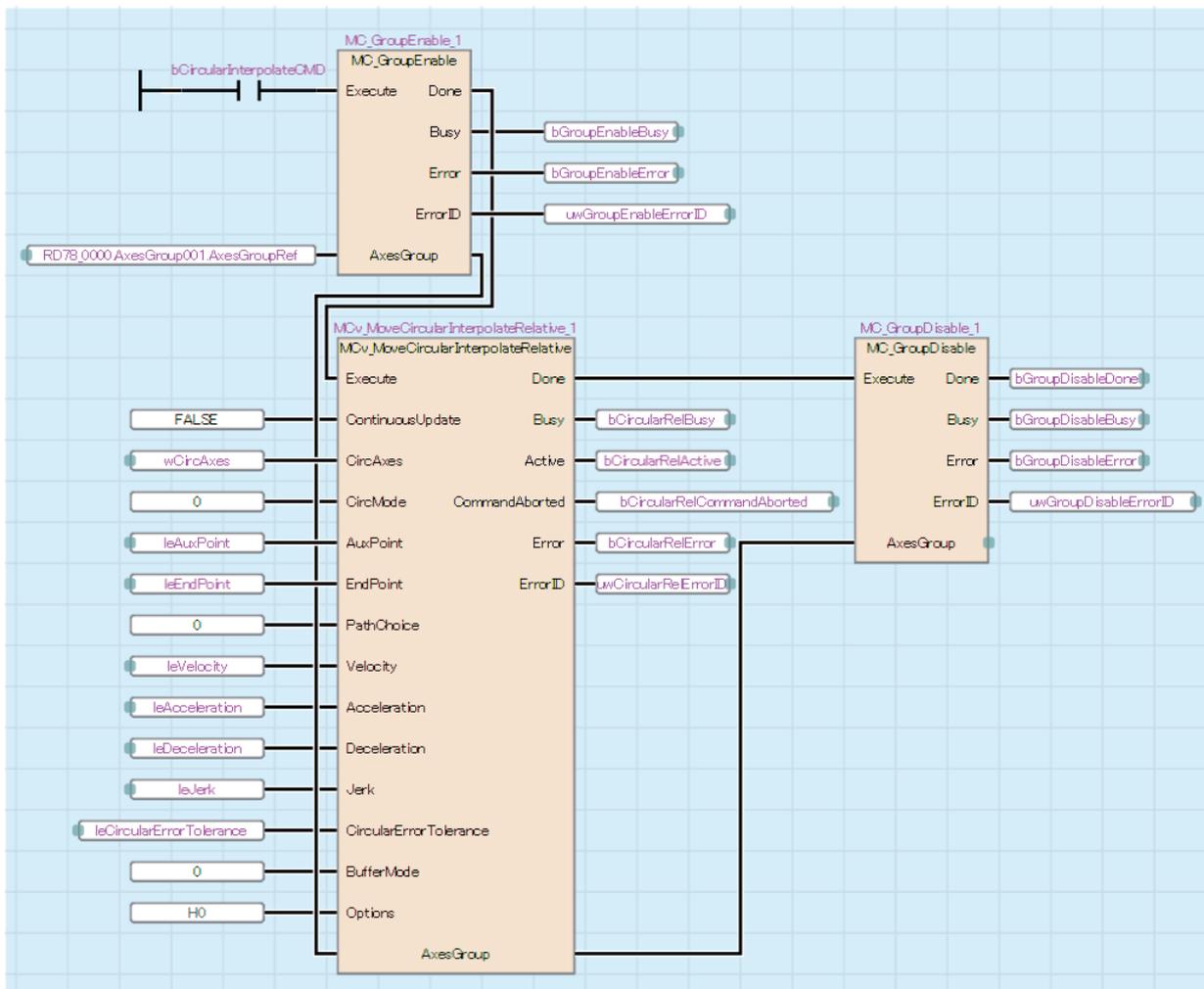
标签名	数据类型	注释
MCv_MoveCircularInterpolateRelative_1	MCv_MoveCircularInterpolateRelative	相对值圆弧插补控制FB
wCircAxes	字[有符号](0..1)	圆弧插补轴
leAuxPoint	双精度实数(0..15)	辅助点
leEndPoint	双精度实数(0..15)	终点
leVelocity	双精度实数	速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
leCircularErrorTolerance	双精度实数	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	出错
uwCircularRelErrorID	字[无符号]/位串[16位]	出错代码
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	出错
uwGroupEnableErrorID	字[无符号]/位串[16位]	出错代码
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	出错
uwGroupDisableErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

- 2轴圆弧插补控制用数据设置



- 轴组有效/圆弧插补控制(边界点指定)/轴组无效



■ST的程序(运动模块侧)

//-----2轴圆弧插补控制用数据设置-----

```
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
);
```

```

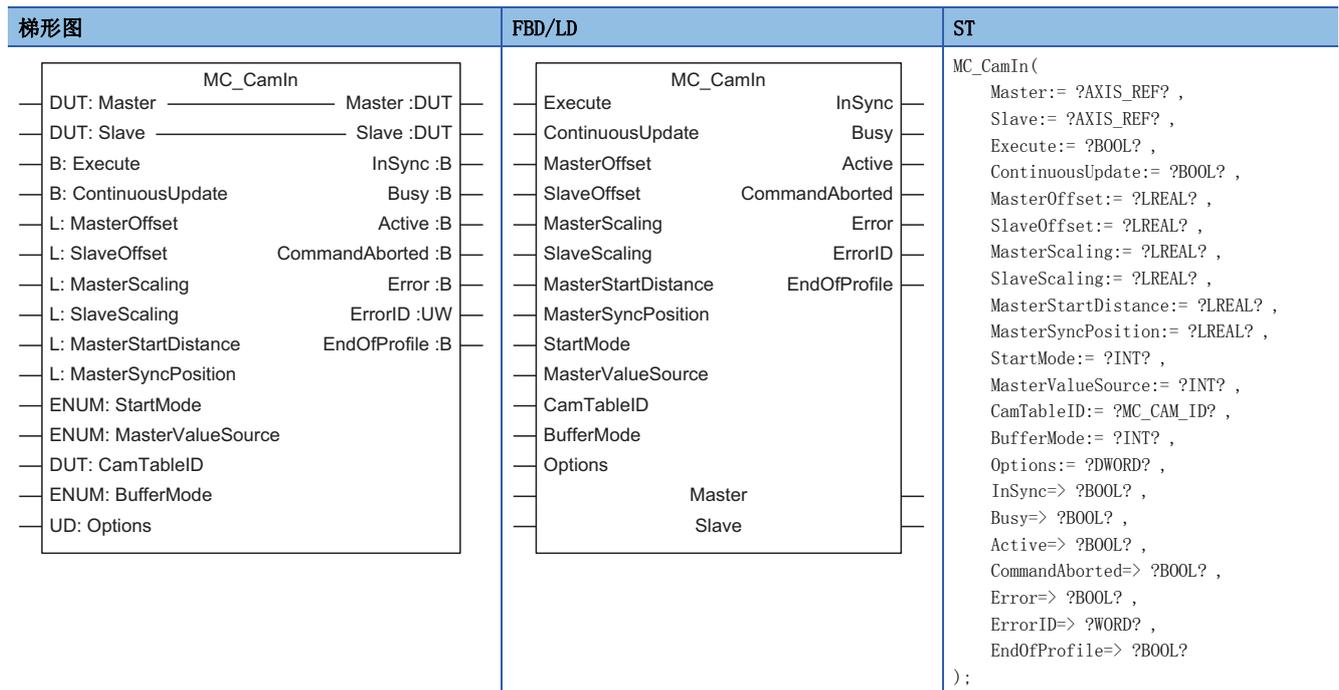
//-----圆弧插补控制(边界点指定)-----
MCv_MoveCircularInterpolateRelative_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bGroupEnableDone ,
    ContinuousUpdate:= FALSE ,
    CircAxes:= wCircAxes ,
    CircMode:= MC_CIRC_MODE__mcBorder ,
    AuxPoint:= leAuxPoint ,
    EndPoint:= leEndPoint ,
    PathChoice:= 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:= bCircularRelDone ,
    Done=> bGroupDisableDone ,
    Busy=> bGroupDisableBusy ,
    Error=> bGroupDisableError ,
    ErrorID=> uwGroupDisableErrorID
);

```

MC_CamIn

按照指定的凸轮数据开始凸轮动作。



名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
凸轮动作开始	96	48	子程序型	随时执行型

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 (<u>AxisName</u> .AxisRef.), 请参阅下述章节。 ☞ 23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 (<u>AxisName</u> .AxisRef.), 请参阅下述章节。 ☞ 23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。 ☞ 304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。 ☞ 305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。 ☞ 306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。 ☞ 306页 从轴系数(SlaveScaling)
MasterStartDistance	主轴跟踪距离	LREAL	启动时	-10000000000.0 ~ 10000000000.0	0.0	设置输出轴(OutputData)开始同步的主轴(Master)的位置(从主轴同步开始位置(MasterSyncPosition)开始的相对位置)。
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。 ☞ 307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。 ☞ 308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	1~6000	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。 ☞ 309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。 ☞ 310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	出错	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时, 将返回出错代码。 关于出错代码的详细内容, 请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册(应用篇)
EndOfProfile	凸轮循环完成	BOOL	FALSE	控制中(Active)变为了TRUE后, 每当移动1周期长时, 仅调用FB的POU(程序部件)的执行周期的1周期变为TRUE。 关于详细内容, 请参阅下述章节。 ☞ 310页 凸轮循环完成(EndOfProfile)

■公开变量

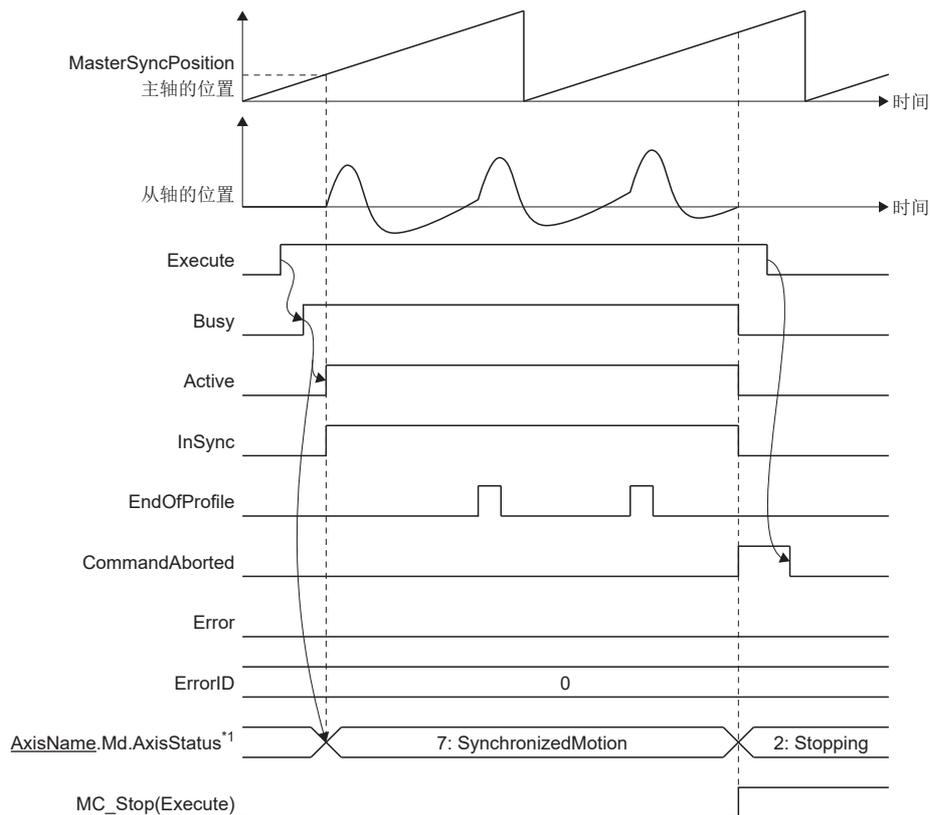
公开变量	名称	数据类型	初始值	内容
InputPerCycle	1周期当前值	LREAL	0.0	存储1周期当前值。 关于详细内容, 请参阅下述章节。 ☞ 310页 1周期当前值 (InputPerCycle)
Reference	基准值	LREAL	0.0	存储基准值。 关于详细内容, 请参阅下述章节。 ☞ 311页 基准值 (Reference)
OutputData	输出值	LREAL	0.0	存储输出值。 关于详细内容, 请参阅下述章节。 ☞ 312页 输出值 (OutputData)
InstanceID	实例ID	WORD (UINT)	0	是实例ID。 实例创建时由系统自动设置。 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。
☞ 11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)如下表所示。

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*3了主轴同步开始位置(MasterSyncPosition)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。

☞ 307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。

☞ 310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/连续更新的控制更改的时机

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,根据开始模式(StartMode)将变为如下表所示。

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。

■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)		
		0: 直线插补	1: 各区间中指定	2: 样条插补
重复动作	Periodic	○	○	○
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)

通过重复动作(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)

FALSE(单发动作)

内容

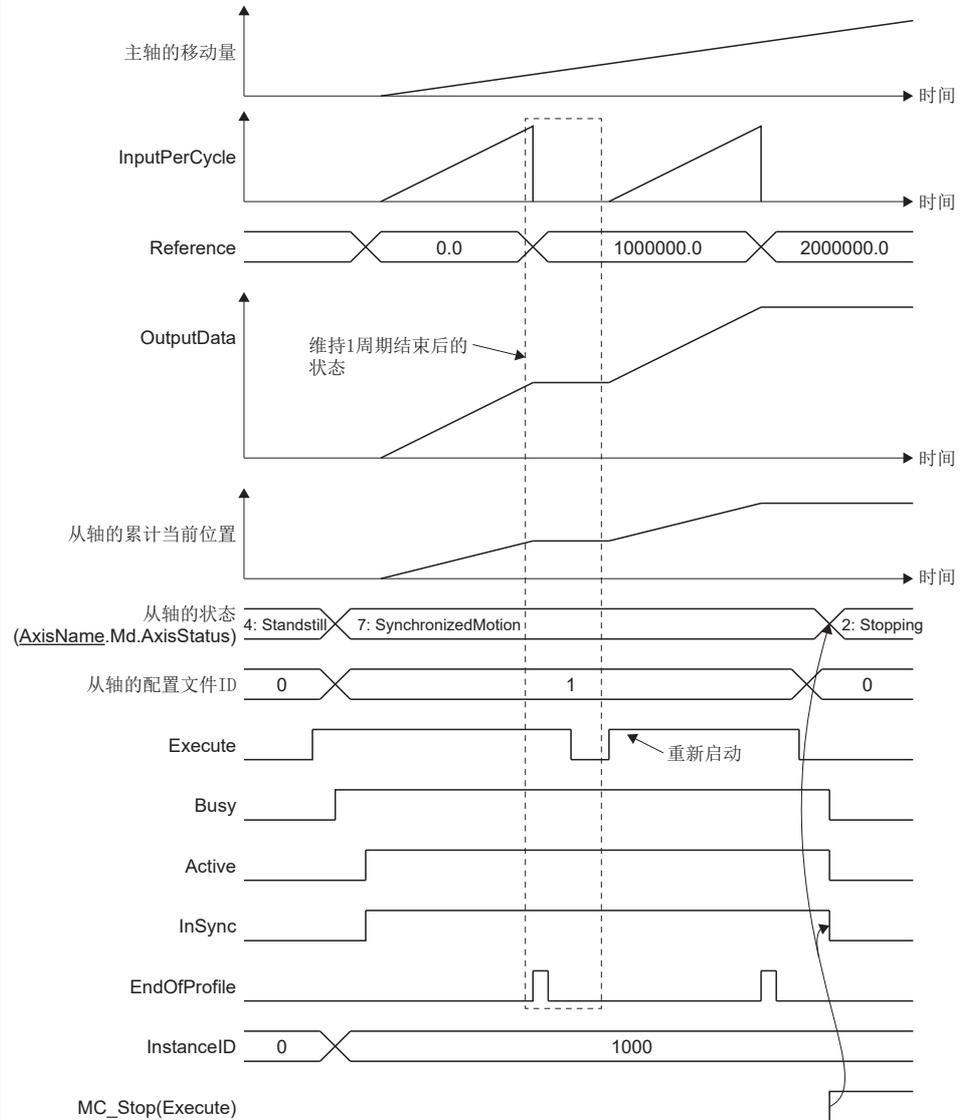
■1周期运行后的动作指定(选项(Options): 位16)为“1: 重启等待”的情况下

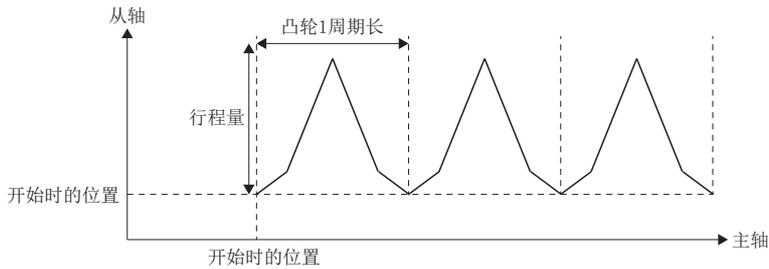
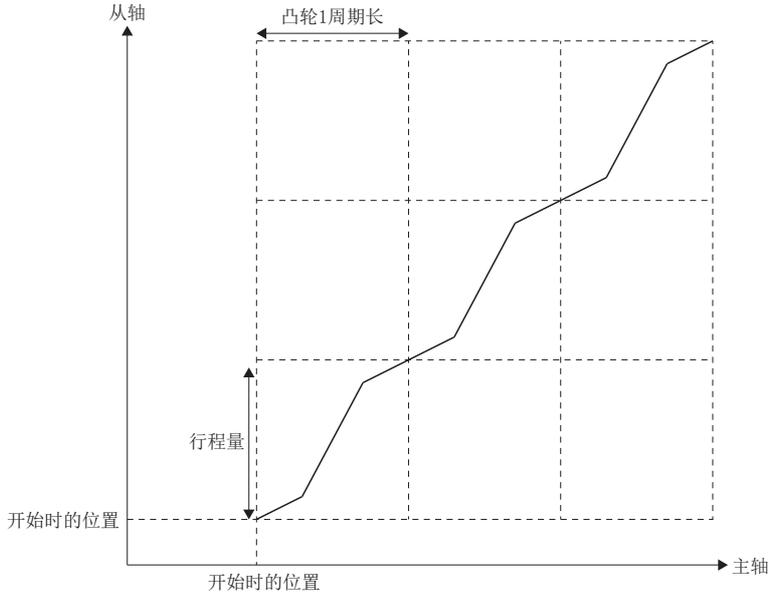
1周期结束后仍维持同步状态, 重新启动执行指令(Execute)时, 也维持基准值(Reference)、输出值(OutputData)的值。同步状态的解除将停止原因输入至从轴。

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)。

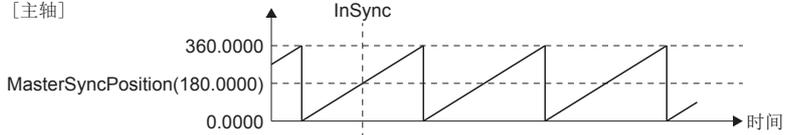
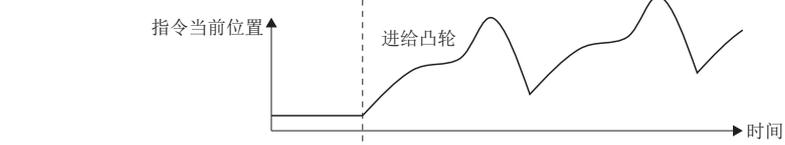
重新启动执行指令(Execute)时, 与起开始模式(StartMode)的设置无关, 从轴均即时开始动作。

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。



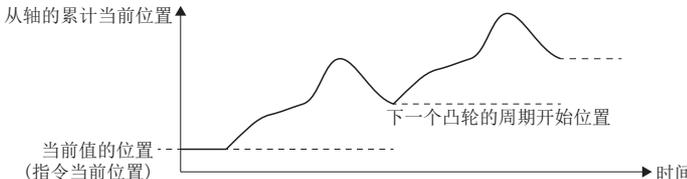
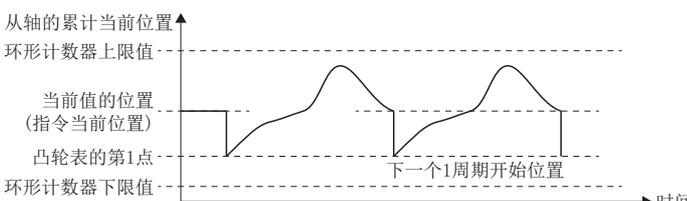
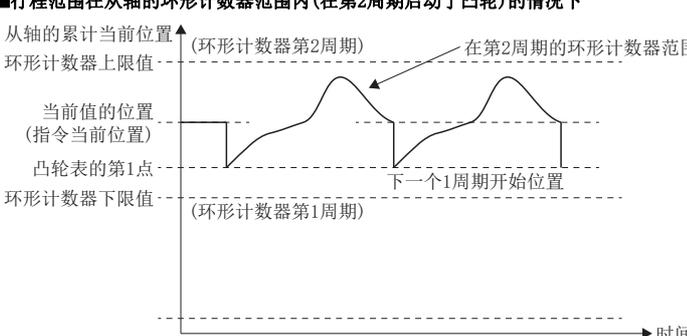
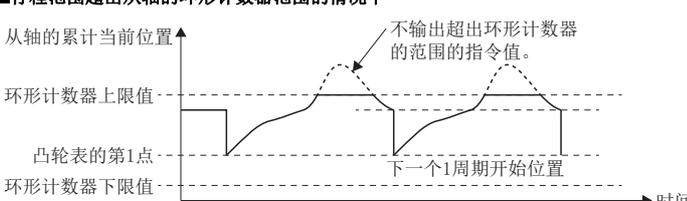
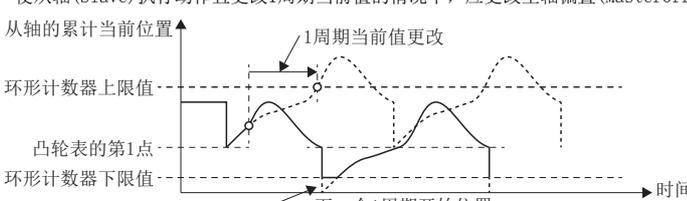
重复动作(Periodic)	内容
TRUE(重复动作)	<p>连续重复凸轮表的执行。根据凸轮动作(往复凸轮或进给凸轮)，其动作如下所示。</p> <p>■往复凸轮 由于重复导致的凸轮表始点的从轴(Slave)的位置每次相同的情况下，将作为往复凸轮执行动作。</p>  <p>■进给凸轮 由于重复导致的凸轮表始点的从轴(Slave)的位置在始点与终点中指令当前位置不相同的情况下，将作为进给凸轮执行动作。</p> 

- 主轴绝对坐标(MasterAbsolute)
 通过主轴绝对坐标(MasterAbsolute)的设置的凸轮动作如下所示。

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> 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>[从轴]</p>  <p>指令当前位置</p>  <p>InSync</p> <p>主轴通过“180.0000”的位置时从轴开始动作。根据主轴的移动量执行动作。</p> <p>进给凸轮</p>

• 从轴绝对坐标 (SlaveAbsolute)

通过从轴绝对坐标 (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>  <p>从轴的累计当前位置</p> <p>下一个凸轮的周期开始位置</p> <p>当前值的位置 (指令当前位置)</p> <p>时间</p>
TRUE (绝对坐标)	<p>在包含同步中 (InSync) 变为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 的伺服出错 ([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>从轴的累计当前位置</p> <p>环形计数器上限值</p> <p>当前值的位置 (指令当前位置)</p> <p>凸轮表的第1点</p> <p>下一个1周期开始位置</p> <p>环形计数器下限值</p> <p>时间</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>从轴的累计当前位置</p> <p>环形计数器上限值</p> <p>当前值的位置 (指令当前位置)</p> <p>凸轮表的第1点</p> <p>下一个1周期开始位置</p> <p>环形计数器下限值</p> <p>时间</p> <p>■行程范围超出从轴的环形计数器范围的情况下</p>  <p>从轴的累计当前位置</p> <p>环形计数器上限值</p> <p>凸轮表的第1点</p> <p>下一个1周期开始位置</p> <p>环形计数器下限值</p> <p>时间</p> <p>不输出超出环形计数器的范围的指令值。</p> <p><注意事项></p> <ul style="list-style-type: none"> • ProfileControl (运算配置文件控制) 的版本为“1.15及以前”的情况下，也输出超出环形计数器范围的指令值。超出环形计数器范围的值将变为在环形计数器范围内化整后的值。 <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)。  <p>从轴的累计当前位置</p> <p>环形计数器上限值</p> <p>凸轮表的第1点</p> <p>下一个1周期开始位置</p> <p>环形计数器下限值</p> <p>时间</p> <p>1周期当前值更改</p> <p>超出环形计数器范围的范围将变化。</p> <p>不输出超出环形计数器范围的指令值。</p>

- 1周期长(CycleLength)

1周期长设置1周期所需的输入量。关于详细内容，请参阅下述章节。

☞ 310页 1周期当前值(InputPerCycle)

- 行程量(Stroke)

行程量设置与行程比100%对应的行程量。关于详细内容，请参阅下述章节。

☞ 312页 输出值(OutputData)

- 开始点(StartPoint)及初始行程量(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。

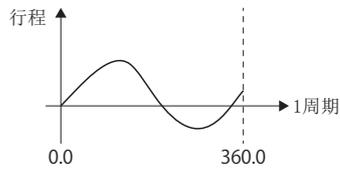
插补方法指定(Interpolate)	凸轮动作的开始点	凸轮动作的初始行程量
0: 直线插补	凸轮表的第1点的输入值	凸轮表的第1点的输出值
1: 各区间中指定	凸轮表的第1点的开始点(StartPoint)	凸轮表的初始行程量(StartStroke)
2: 样条插补		

■ 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”。

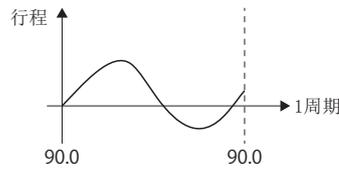
- 凸轮表

<凸轮A(凸轮动作的开始点“0.0”) >



- 插补方法指定: 1: 各区间中指定
- 1周期长: 360.0
- 开始点: 0.0
- 初始行程量: 0.0

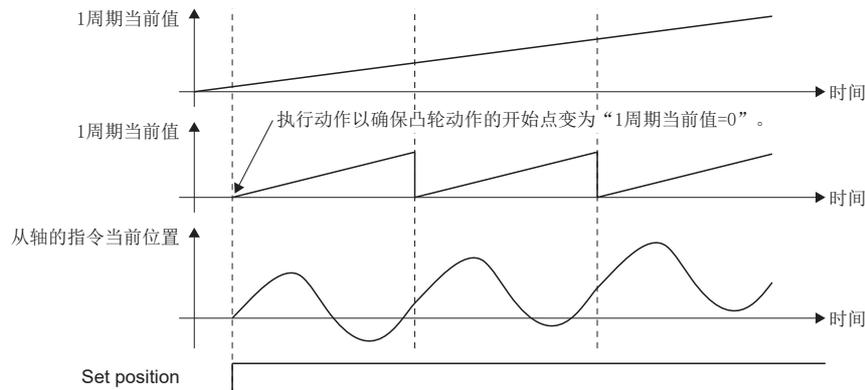
<凸轮B(凸轮动作的开始点“90.0”) >



- 插补方法指定: 1: 各区间中指定
- 1周期长: 360.0
- 开始点: 90.0
- 初始行程量: 0.0

- 凸轮动作

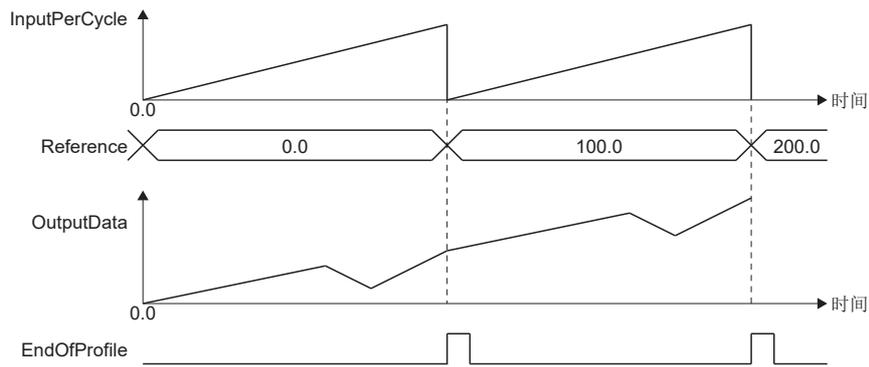
即使使用凸轮A、凸轮B之一也变为下述动作。



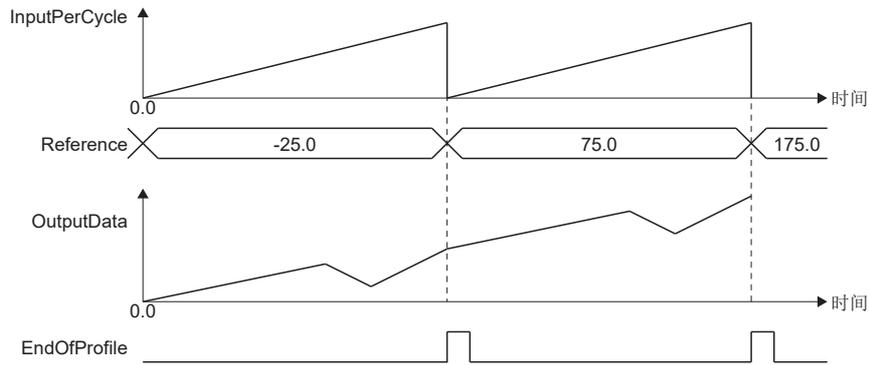
■凸轮动作的初始行程量

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

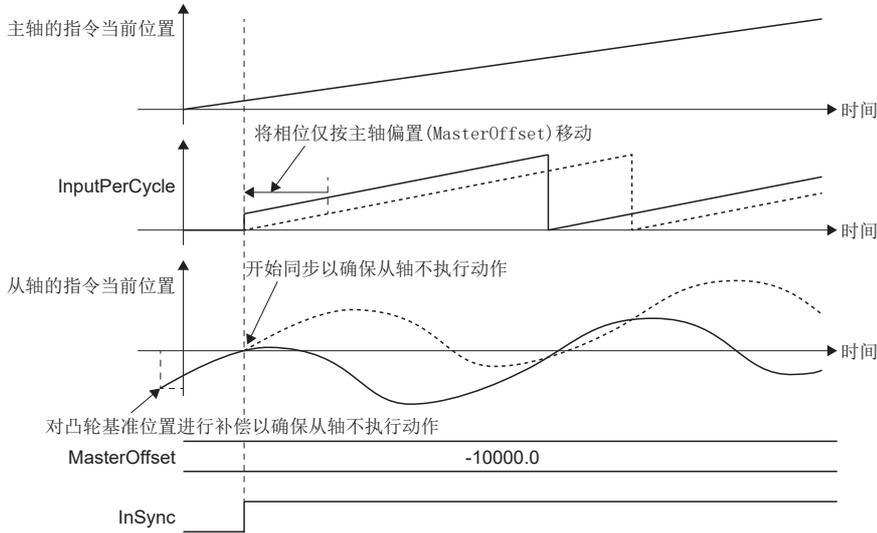


■ 主轴偏置 (MasterOffset)

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)。)

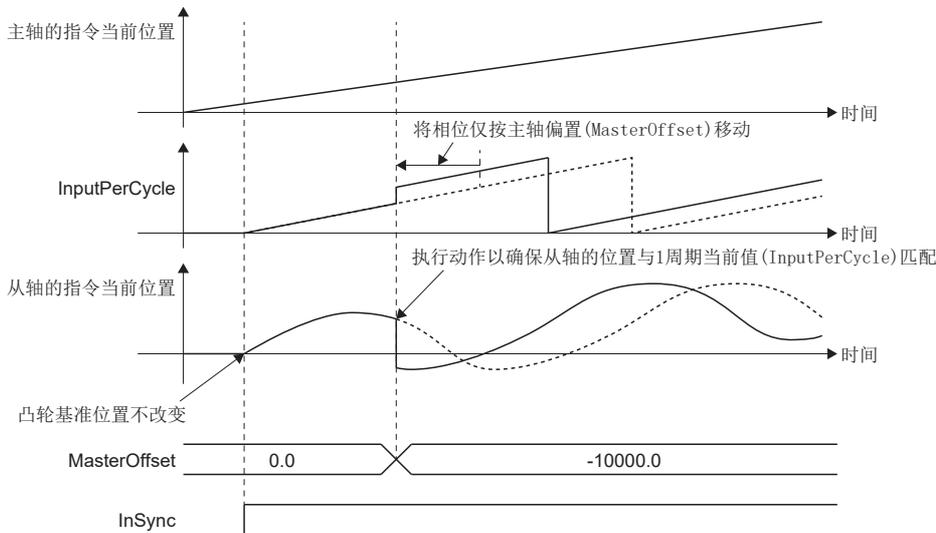
动作开始时设置了“0.0”以外的情况下，同步中 (InSync) 的上升沿时，将变为在主轴 (Master) 的位置加上了偏置量后的凸轮1周期位置。此时，为了防止从轴 (Slave) 执行动作，对基准值 (Reference) 进行补偿。

- 开始模式 (StartMode) 为“0: 即时 (mcImmediate)”的情况下



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。

- 开始模式 (StartMode) 为“0: 即时 (mcImmediate)”的情况下

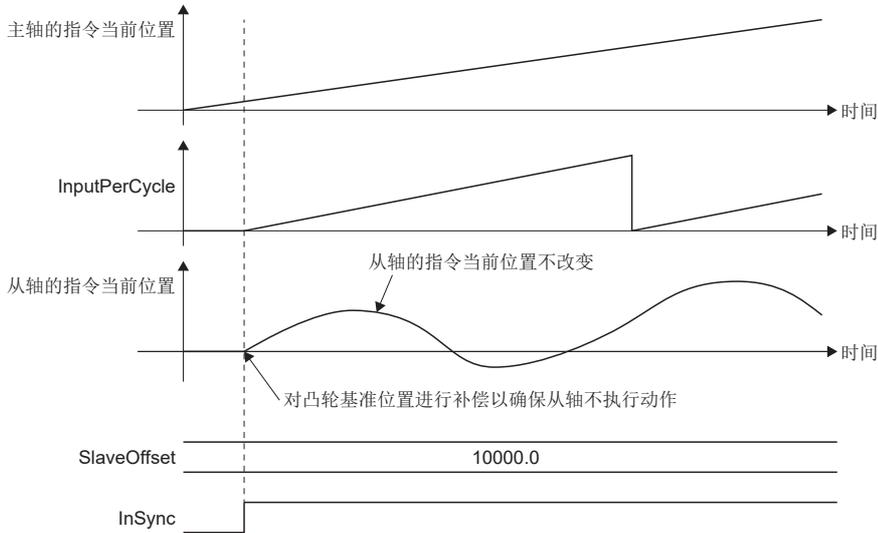


■从轴偏置(SlaveOffset)

将从轴(Slave)的位移按偏置量移动。

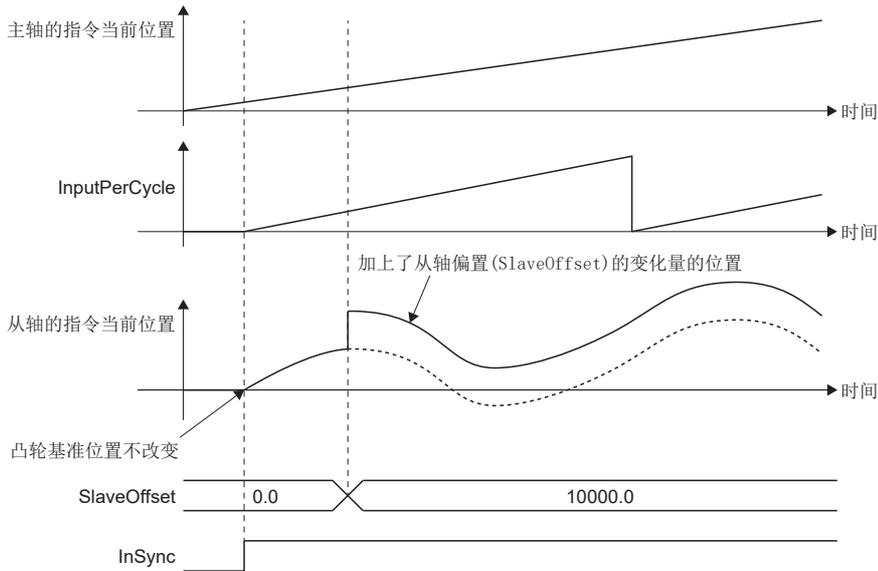
同步中(InSync)的上升沿前设置了“0.0”以外的情况下，为了防止在同步中(InSync)的上升沿时从轴(Slave)执行动作，对基准值(Reference)进行补偿。

- 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。

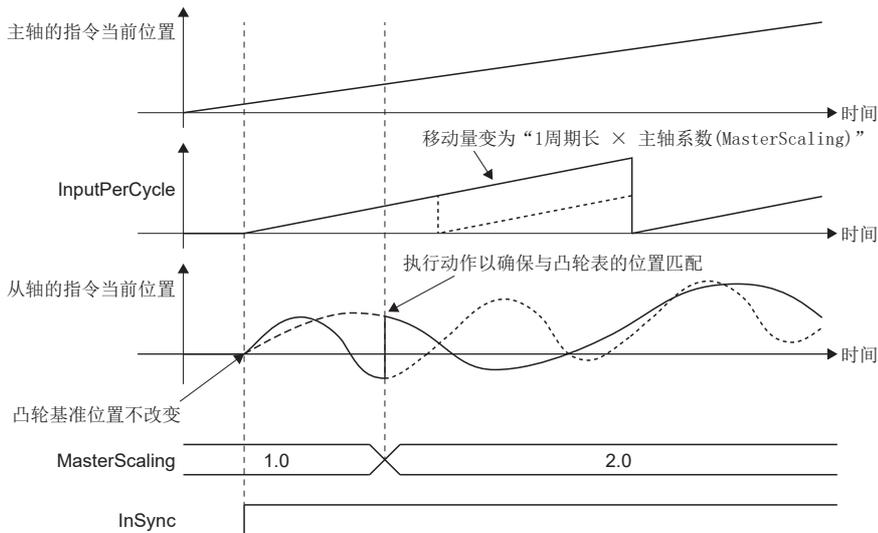
- 开始模式(StartMode)为“0: 即时(mcImmediate)”的情况下



■ 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。

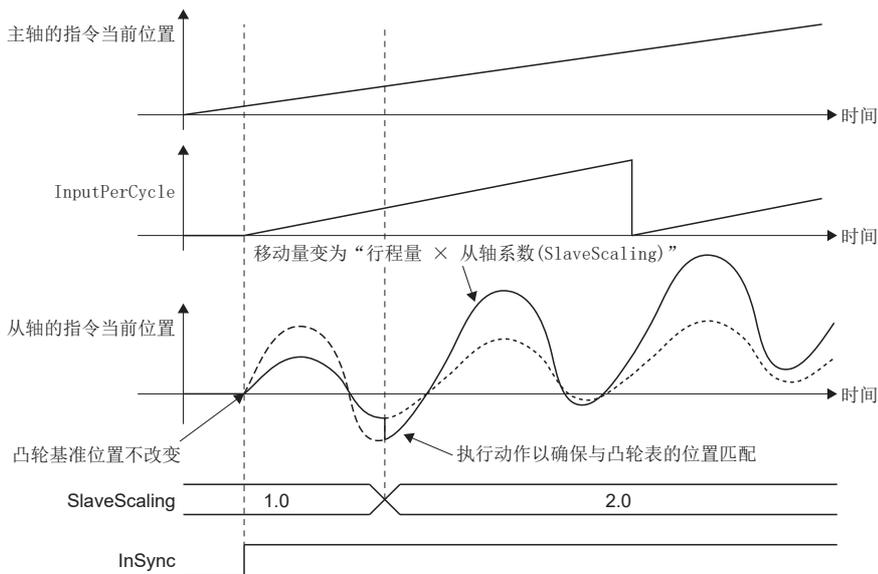
同步中进行了更改的情况下，将输出移动量的指令，以使从轴 (Slave) 的位置与凸轮表的位置匹配。



■ 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。

设置了设置值范围以外的情况下，将变为“超出开始模式范围(出错代码：1A47H)”。

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。

设置值	内容
0: 即时 (mcImmediate)	<p>开始MC_CamIn(凸轮动作开始)时执行。 不使用主轴同步开始位置(MasterSyncPosition)、主轴跟踪距离(MasterStartDistance)。 执行指令(Execute)启动后，从轴开始动作。</p>

设置值	内容
1: 绝对 (mcAbsolute)	<p>主轴的位置通过主轴同步开始位置 (MasterSyncPosition) 时执行。 不使用主轴跟踪距离 (MasterStartDistance)。 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。

设置值	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。

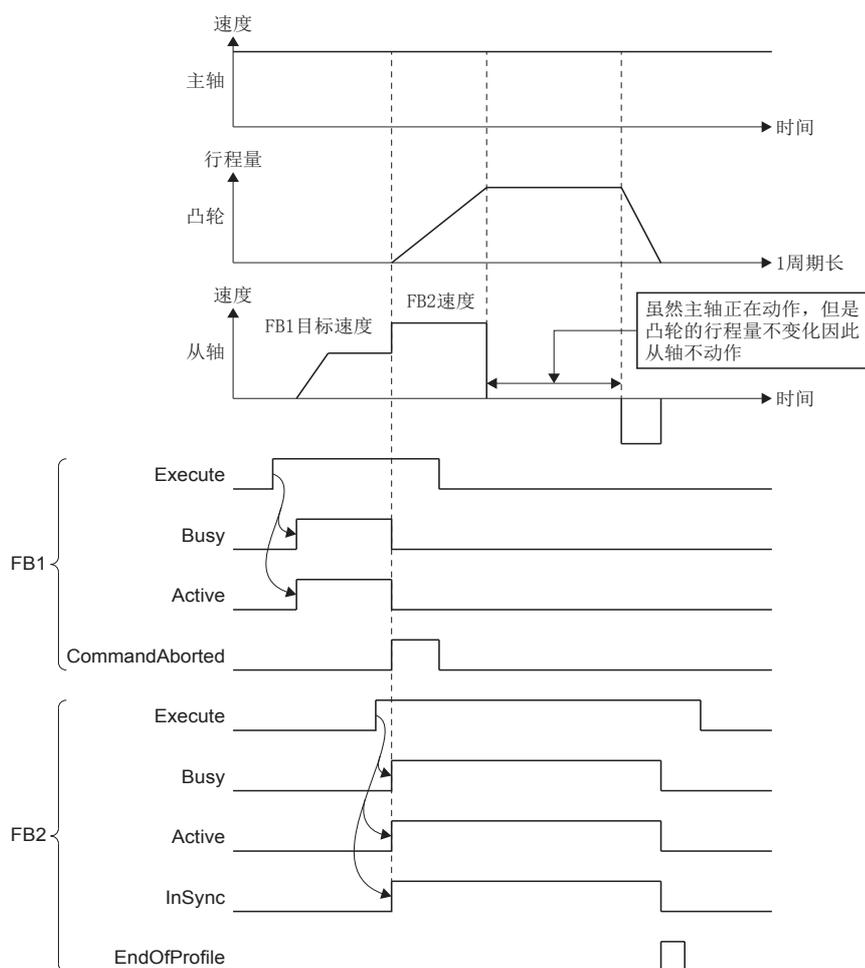
在MC_CamIn(凸轮动作开始)中,可以设置下述缓冲模式。

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。各模式下的切换条件如下所示。

设置值	内容	切换条件
0: Aborting(mcAborting)	中断(取消)执行中的FB并立即执行下一个FB。	始终
1: Buffered(mcBuffered)	在执行中的FB中缓冲下一个FB。 执行中的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) 执行中的FB完成时,依次执行缓冲FB。	凸轮循环完成(EndOfProfile)为TRUE时

例

对执行中的FB1上连接启动FB2(从轴(Slave): MC_CamIn(凸轮动作开始))设置开始模式(StartMode)的“0: 即时(mcImmediate)”,并将缓冲模式(BufferMode)设置为“0: 始终(Aborting)”的情况下



FB2的速度的发生是在同步中(InSync)变为TRUE之后。

开始模式(StartMode)为“0: 即时(mcImmediate)”以外的情况下,FB启动后同步中(InSync)不立即变为TRUE,因此在执行中(Busy)、控制中(Active)及同步中(InSync)变为TRUE之前的期间速度将为0。

要点

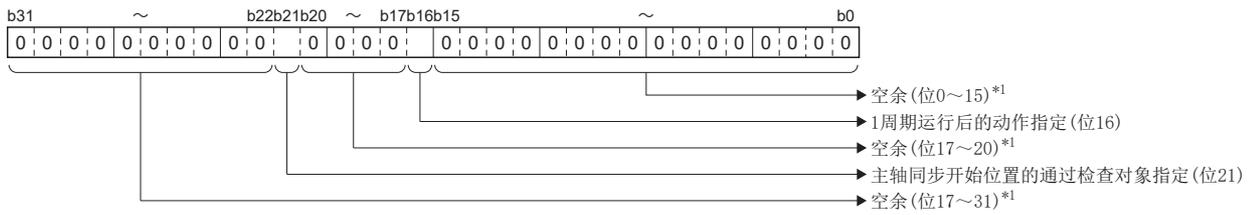
关于多重启动(缓冲模式)的详细内容,请参阅下述手册。

📖 MELSEC iQ-R运动模块用户手册(应用篇)

■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。

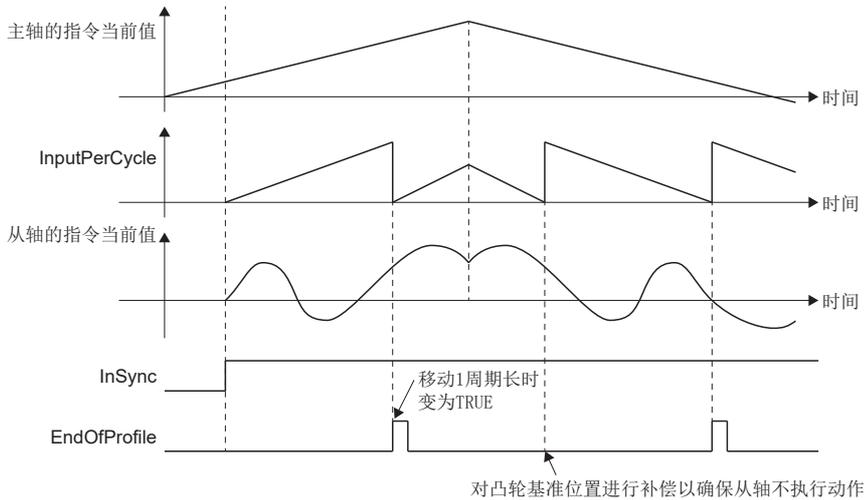


*1 空余中应设置“0”。设置了“0”以外的情况下，将变为“超出Options范围(出错代码: 1A4EH)”。

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)变成了TRUE后, 每当移动1周期长时, 仅调用FB的POU(程序部件)的执行周期的1周期变为TRUE。



■1周期当前值(InputPerCycle)

初始值将变为“0.0”。

控制中(Active)变成了TRUE后, 将按下述所示反映主轴(Master)的移动量。

$$1\text{周期当前值} = (\text{主轴的累计移动量} + \text{主轴偏置} + 1\text{周期当前值更改值}) \text{ MOD } 1\text{周期长}$$

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)

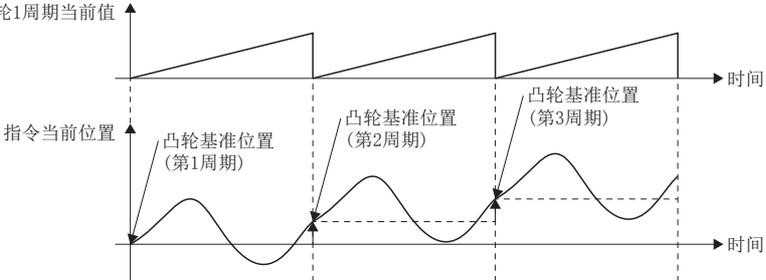
执行MC_CamIn(凸轮动作开始)时,以同步中(InSync)变为TRUE的指令当前位置为基准,从要开始的凸轮位置(凸轮1周期当前值)进行计算。此外,由于减去了凸轮表中的初始行程量,因此初始值的计算公式如下所示。

基准值 = (0 - (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

- 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)	ProfileControl(运算 配置文件控制)	计算公式
FALSE(相对坐标)	版本“1.15”及以后	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
	版本“1.15”以前	基准值 = 0 - (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	按下述公式计算基准值。 • 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) 凸轮1周期当前值 

- 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)

- 监视值的范围根据ProfileControl(运算配置文件控制)的版本将变为下述范围。

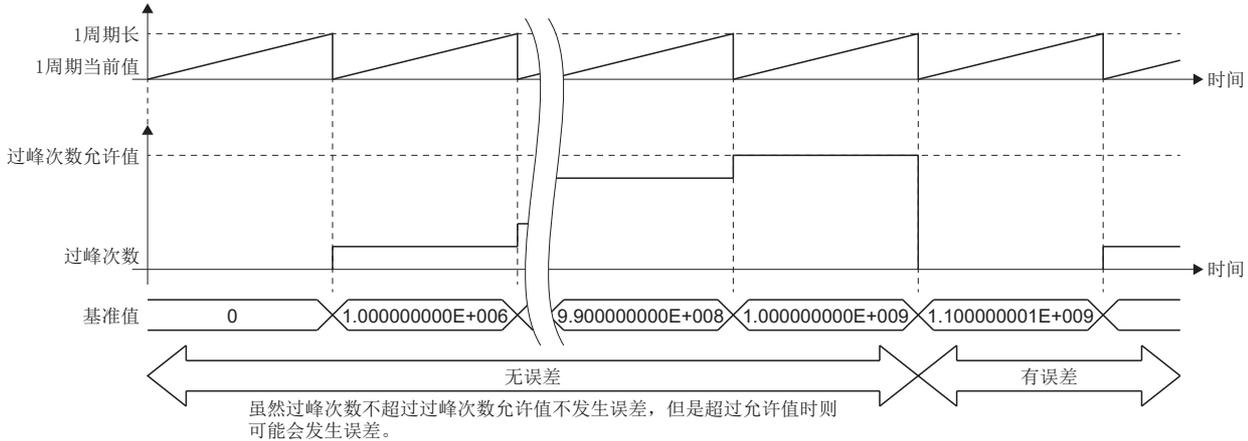
ProfileControl(运算配置文件控制)	监视值的范围
版本“1.15”及以后	从轴的环形计数器下限值 ≤ 基准值 < 从轴的环形计数器上限值
版本“1.15”以前	定位范围的下限值 ≤ 基准值 < 定位范围的上限值

<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”。

■输出值(OutputData)

初始值为凸轮表中的初始行程量。

同步中(InSync)为TRUE时, 将按下述方式进行计算。

- 输出值 = 基准值 + ((行程量 × 与1周期当前值对应的行程比) × 从轴系数) + 从轴偏置
- 或
- 输出值 = 基准值 + ((与1周期当前值对应的输出值) × 从轴系数) + 从轴偏置

监视值的范围根据ProfileControl(运算配置文件控制)的版本将变为下述范围。

ProfileControl(运算配置文件控制)	监视值的范围
版本“1.15”及以后	从轴的环形计数器下限值 ≤ 输出值 < 从轴的环形计数器上限值
版本“1.15”以前	定位范围的下限值 ≤ 输出值 < 定位范围的上限值

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。应充分考虑设置值及更改时机。
- 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		设置值
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

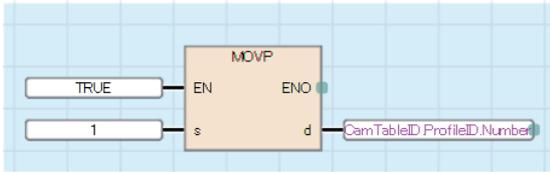
配置文件ID	标签名	数据类型	注释
1	ProfileData0001	MC_CAM_REF	凸轮数据1

使用的标签

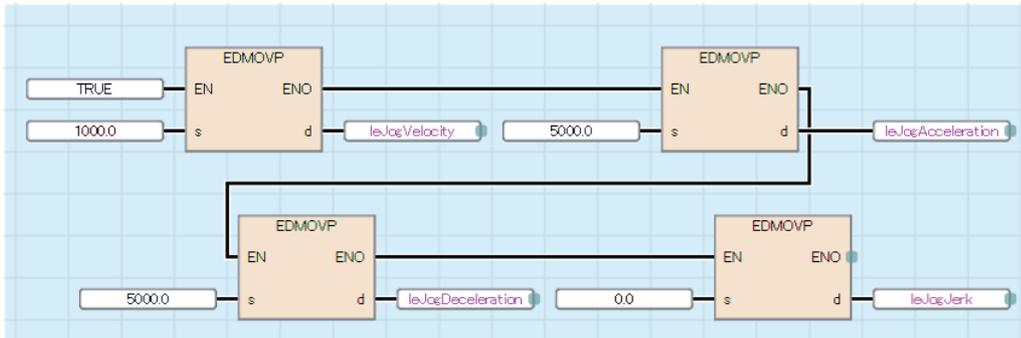
标签名	数据类型	注释
MC_CamIn_1	MC_CamIn	凸轮动作开始FB
bCamInStart	位	凸轮动作开始指令
CamTableID	MC_CAM_ID	凸轮表ID
bInSync	位	同步中
bCamBusy	位	执行中
bCamActive	位	控制中
bCamCommandAborted	位	执行中断
bCamError	位	出错
uwCamErrorID	字[无符号]/位串[16位]	出错代码
bEndOfProfile	位	凸轮循环完成
MCv_Jog_1	MCv_Jog	JOG运行FB
leJogVelocity	双精度实数	JOG速度
leJogAcceleration	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	出错
uwJogErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

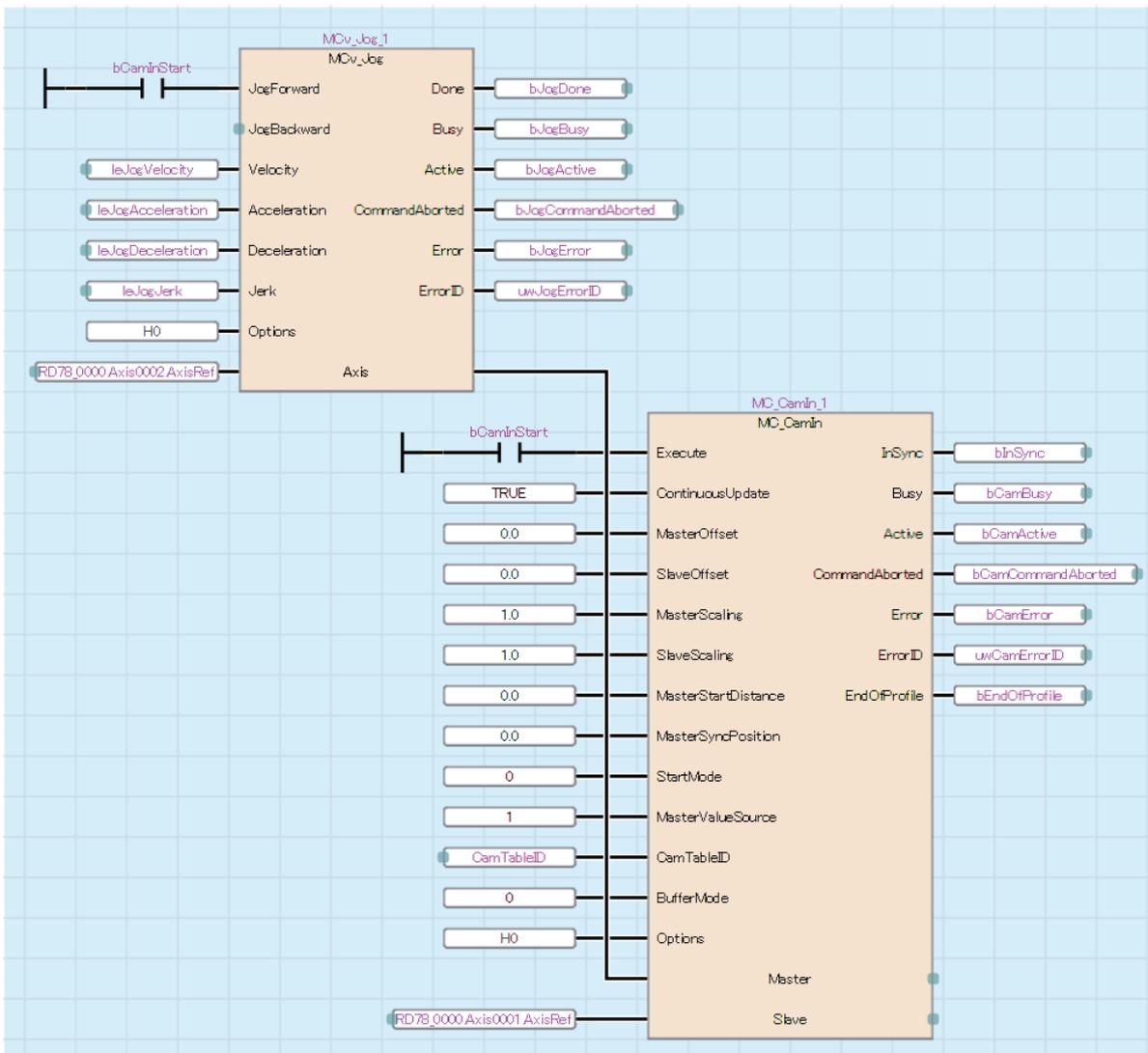
- 运算配置文件(凸轮数据)设置



- JOG运行用数据设置



- JOG运行/凸轮动作开始



■ST的程序(运动模块侧)

```
//-----运算配置文件(凸轮数据)设置-----
CamTableID.ProfileID.Number:= 1;

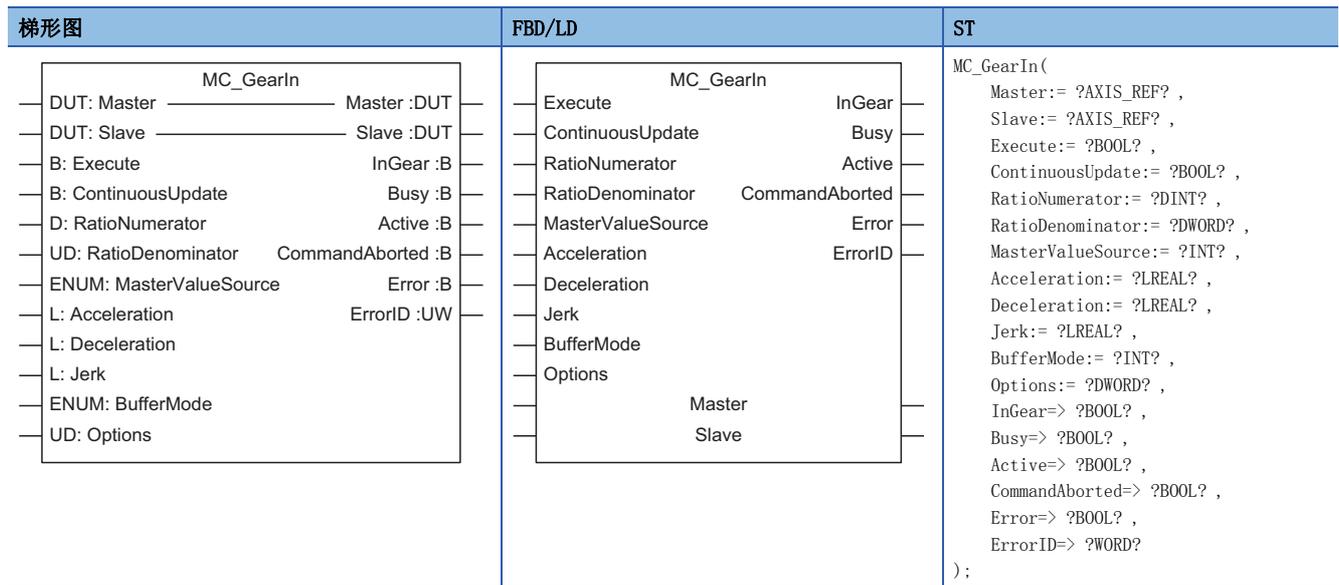
//-----JOG运行用数据设置-----
leJogVelocity:= 1000.0;
leJogAcceleration:= 5000.0;
leJogDeceleration:= 5000.0;
leJogJerk:= 0.0;

//-----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
);
```

MC_GearIn

按照指定的齿轮比开始齿轮动作。



名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
齿轮动作开始	64	8	子程序型	随时执行型

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 (<u>AxisName</u> , <u>AxisRef</u> .), 请参阅下述章节。 ☞ 23页 <u>AxisName</u> . <u>AxisRef</u> . (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 (<u>AxisName</u> , <u>AxisRef</u> .), 请参阅下述章节。 ☞ 23页 <u>AxisName</u> . <u>AxisRef</u> . (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。 ☞ 320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。 ☞ 320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。 321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	设置加减速度动作开始时的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。 321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(出错代码: 1A4EH)”。

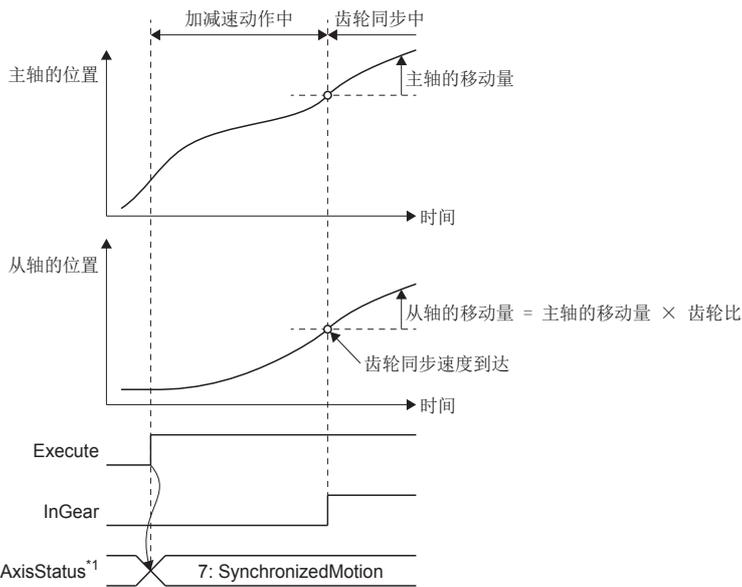
■输出变量

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	出错	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时, 将返回出错代码。 关于出错代码的详细内容, 请参阅下述手册。 MELSEC iQ-R运动模块用户手册(应用篇)

功能

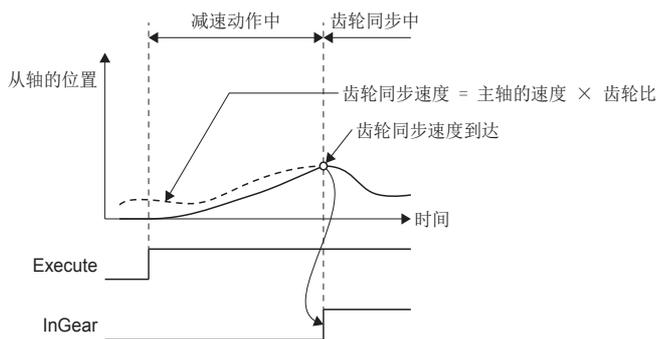
- 对于MC_GearIn(齿轮动作开始), 设置齿轮比分子(RatioNumerator)、齿轮比分母(RatioDenominator)、主轴数据源选择(MasterValueSource)、加速度(Acceleration)、减速度(Deceleration)、Jerk、缓冲模式(BufferMode), 开始齿轮动作。
- 结束动作的情况下, 通过MC_Stop(强制停止)进行。

- 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，并进行加减速度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。



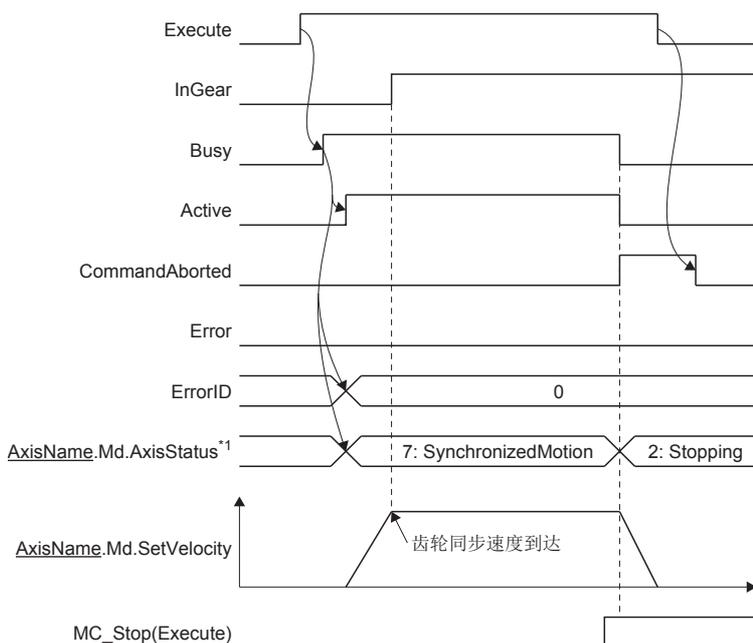
*1 是从轴(Slave)的轴状态。

- 加减速度动作中主轴(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。

☞ 11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) (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) (绝对值) $>$ 齿轮同步速度 (绝对值)	进行减速动作直到齿轮同步速度。
从轴(Slave) = 齿轮同步速度	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)将对主轴(Master)的速度进行了齿轮比换算后的速度进行控制。

■ 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。

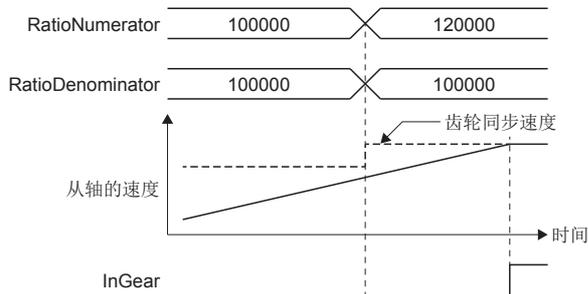
按下述方式转换速度。

$$\text{转换后的主轴 (Master) 速度} = \text{转换前的主轴 (Master) 速度} \times \frac{\text{齿轮比分子 (RatioNumerator)}}{\text{齿轮比分母 (RatioDenominator)}}$$

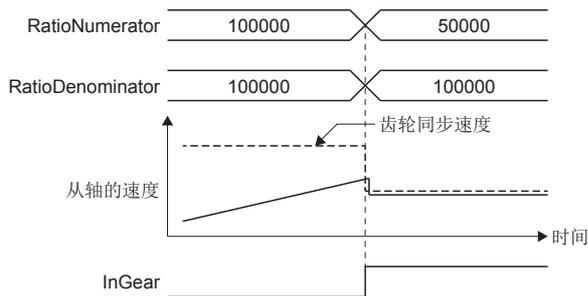
- 齿轮比分母 (RatioDenominator) 中应设置正的值。
- 通过在齿轮比分子 (RatioNumerator) 的设置值中设置负的值，可以反转速度后进行传送。加减速动作中更改了齿轮比的情况下，齿轮同步速度将变化。因此，根据齿轮同步速度的变化，从轴 (Slave) 的速度有可能直接变化并开始齿轮动作。
- 通过在齿轮比分子 (RatioNumerator) 的设置值中设置“0”，可以将转换后的主轴 (Master) 的速度设置为“0”。
- 加减速动作中更改了齿轮比的情况下，目标速度将变化。因此，根据目标速度的变化，从轴 (Slave) 的速度有可能直接变化并开始齿轮动作。

• 加减速动作中的齿轮比更改

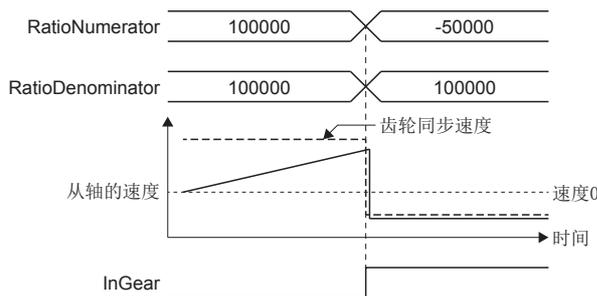
• 齿轮比增加 (将齿轮同步速度设置为高于从轴 (Slave) 的速度的值)



• 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)

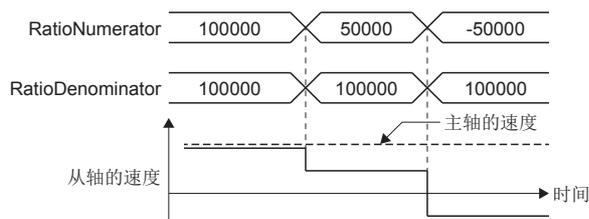


• 齿轮比反转 (将齿轮同步速度设置为低于“0”的值)



• 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。

设置值	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。

设置值	内容
0: Aborting (mcAborting)	中断 (取消) 执行中的FB并立即执行下一个FB。
1: Buffered (mcBuffered)	在执行中的FB中缓冲下一个FB。 执行中的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) 执行中的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。

 MELSEC iQ-R运动模块用户手册 (应用篇)

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 (警告代码: OD04H)” 或 “减速时间限制溢出警告 (警告代码: OD05H)”，停止加减速动作，并以警告检测时的速度继续动作。发生了上述警告的情况下，通过调整转换后的主轴 (Master) 的速度、加速度 (Acceleration) 或减速度 (Deceleration) 以设置为不超出加减速时间上限值，重新开始加减速动作。

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	设置值	
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

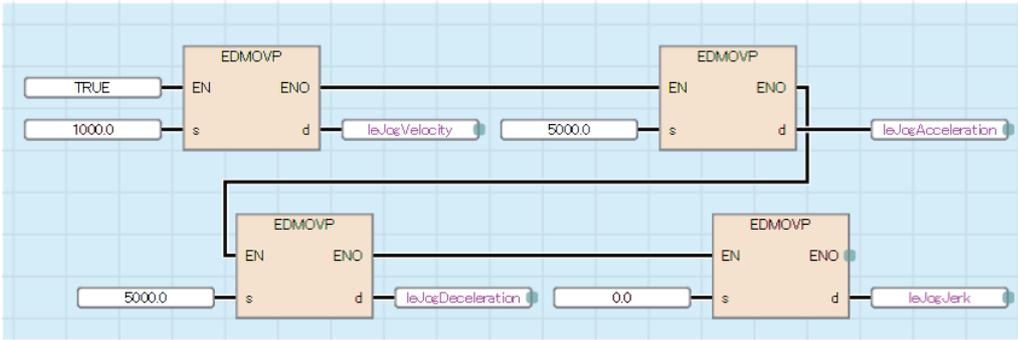
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1
2	Axis0002	AXIS_REF	轴2

■使用的标签

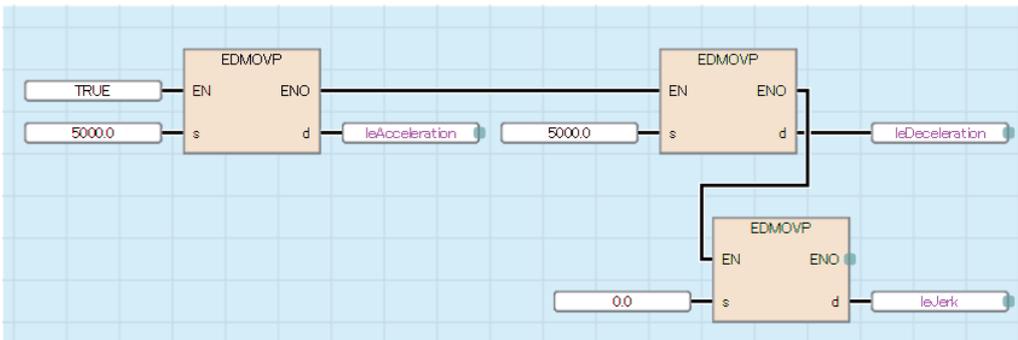
标签名	数据类型	注释
MC_GearIn_1	MC_GearIn	齿轮动作开始FB
bGearInStart	位	齿轮动作开始指令
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bInGear	位	同步中
bGearBusy	位	执行中
bGearActive	位	控制中
bGearCommandAborted	位	执行中断
bGearError	位	出错
uwGearErrorID	字[无符号]/位串[16位]	出错代码
MCv_Jog_1	MCv_Jog	JOG运行FB
leJogVelocity	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	出错
uwJogErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

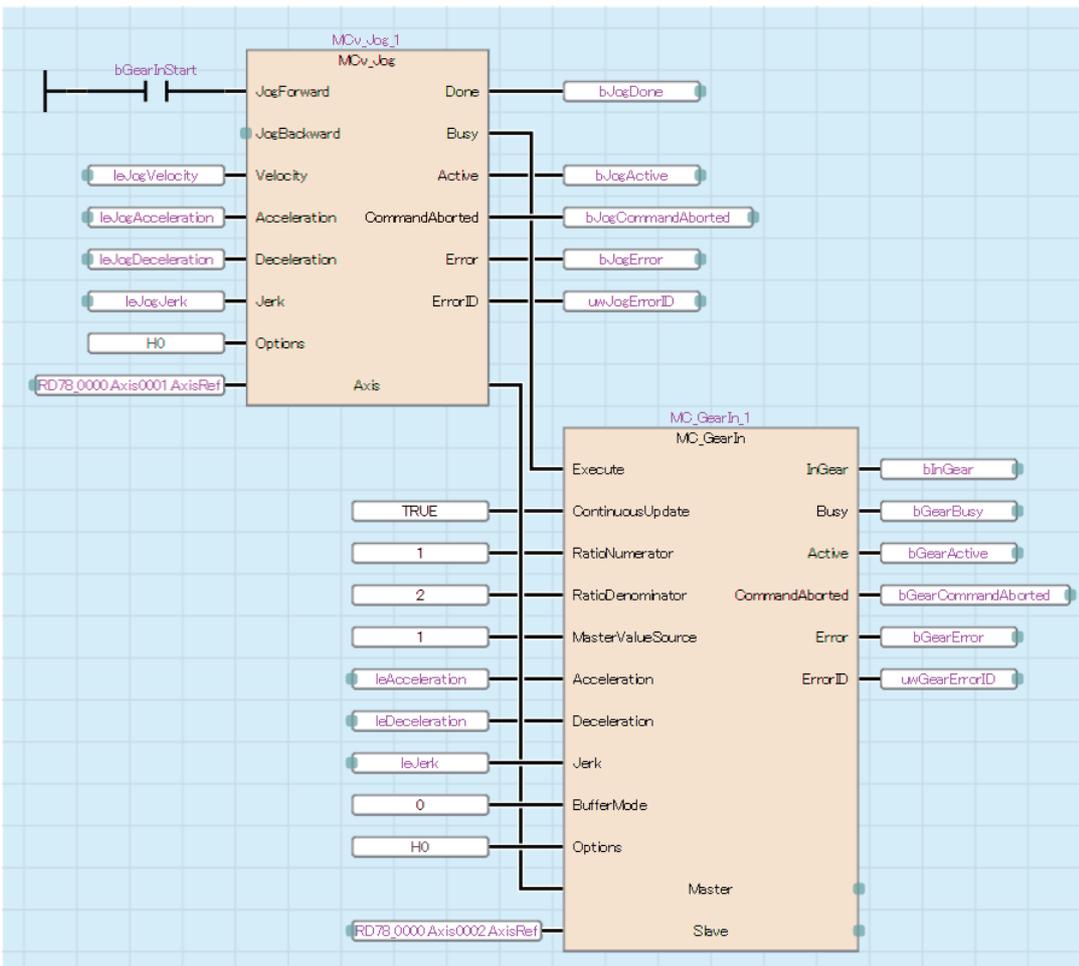
- JOG运行用数据设置



- 齿轮动作数据设置



- JOG运行/齿轮动作开始



■ST的程序(运动模块侧)

```
//-----JOG运行用数据设置-----
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
);
```

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>

名称	输入区域点数(byte)	输出区域点数(byte)	编译方式	FB动作
加减法定位	56	8	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23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.)，请参阅下述章节。 ☞ 23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.)，请参阅下述章节。 ☞ 23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。 • 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。 • 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。 ☞ 328页 加减法方法选择(CombineMode)

输入变量	名称	数据类型	获取	有效范围	初始值	内容
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。 ☞ 328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。 ☞ 328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。 ☞ 329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。 ☞ 329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。 • 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。 ☞ 329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。 • 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。 ☞ 329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。 • 0: Aborting (mcAborting) • 1: Buffered (mcBuffered) 关于详细内容, 请参阅下述章节。 ☞ 330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(出错代码: 1A4EH)”。

■输出变量

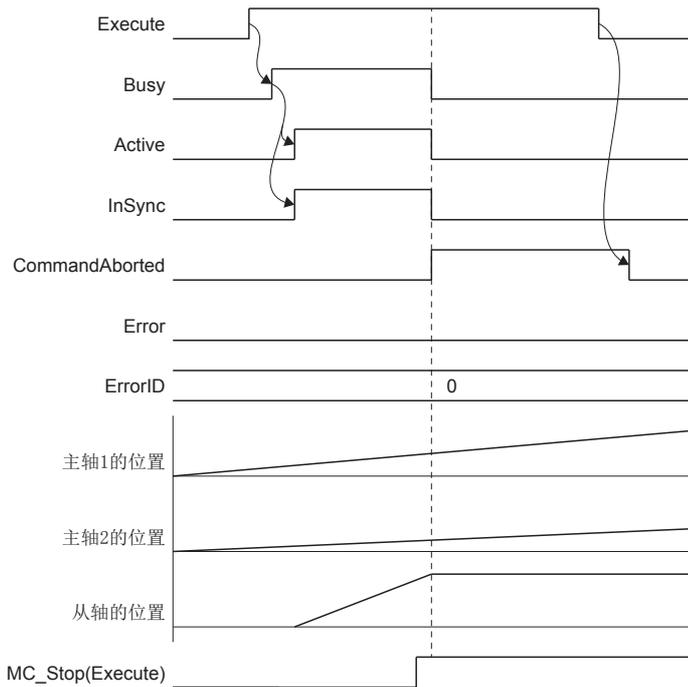
输出变量	名称	数据类型	初始值	内容
InSync	同步中	BOOL	FALSE	从轴 (Slave) 开始了同步时, 将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MC_CombineAxes (加减法定位) 时, 将变为TRUE。
Active	控制中	BOOL	FALSE	正在控制从轴 (Slave) 时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MC_CombineAxes (加减法定位) 的执行中断时, 将变为TRUE。 执行指令 (Execute) 变为FALSE时, 将变为FALSE。
Error	出错	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时, 将返回出错代码。 关于出错代码的详细内容, 请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册 (应用篇)

功能

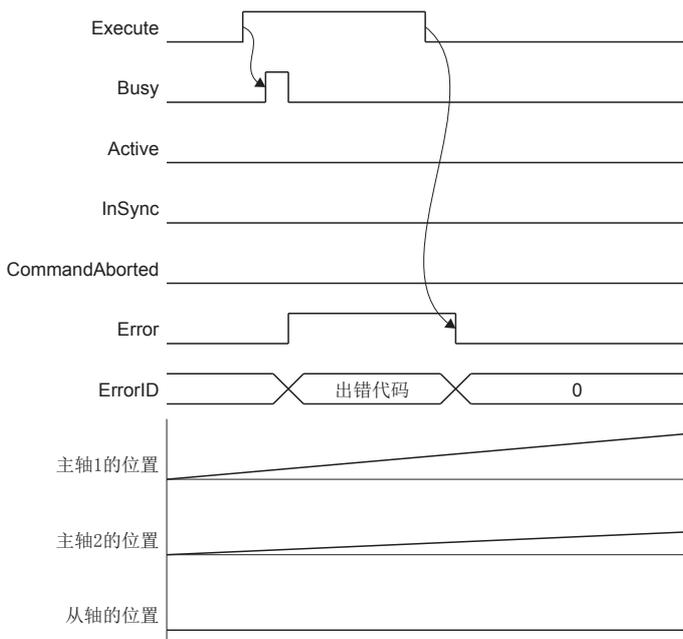
- 在MC_CombineAxes(加减法定位)中, 设置加减法方式选择(CombineMode)、主轴1齿轮比分子(GearRatioNumeratorM1)、主轴1齿轮比分母(GearRatioDenominatorM1)、主轴2齿轮比分子(GearRatioNumeratorM2)、主轴2齿轮比分母(GearRatioDenominatorM2)、主轴1数据源选择(MasterValueSourceM1)、主轴2数据源选择(MasterValueSourceM2)、缓冲模式(BufferMode), 并进行加减法定位。
- 结束动作的情况下, 通过MC_Stop(强制停止)进行。
- 对主轴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



■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。

设置值	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)的设置值中设置负的值,可以反转移动量后进行合成。
- 通过在主轴1齿轮比分子(GearRatioNumeratorM1)的设置值中设置“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)的设置值中设置负的值，可以反转移动量后进行合成。
- 通过在主轴2齿轮比分子(GearRatioNumeratorM2)的设置值中设置“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。

设置值	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。

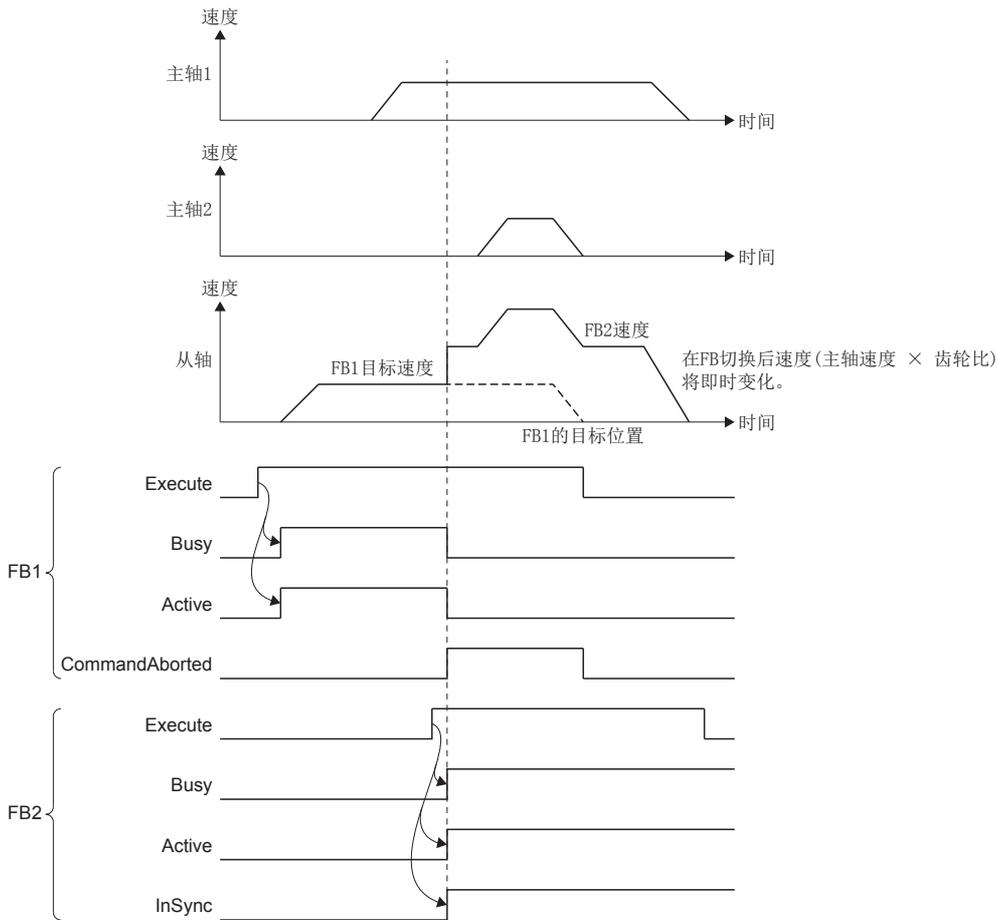
MC_CombineAxes(加减法定位)的速度跟踪同步的FB(主轴),因此FB切换时根据主轴速度与设置的齿轮数据,速度即时变化。

设置值	内容
0: Aborting(mcAborting)	中断(取消)执行中的FB并立即执行下一个FB。
1: Buffered(mcBuffered)	在执行中的FB中缓冲下一个FB。 执行中的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)执行中的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。

📖 MELSEC iQ-R运动模块用户手册(应用篇)

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		设置值	
		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

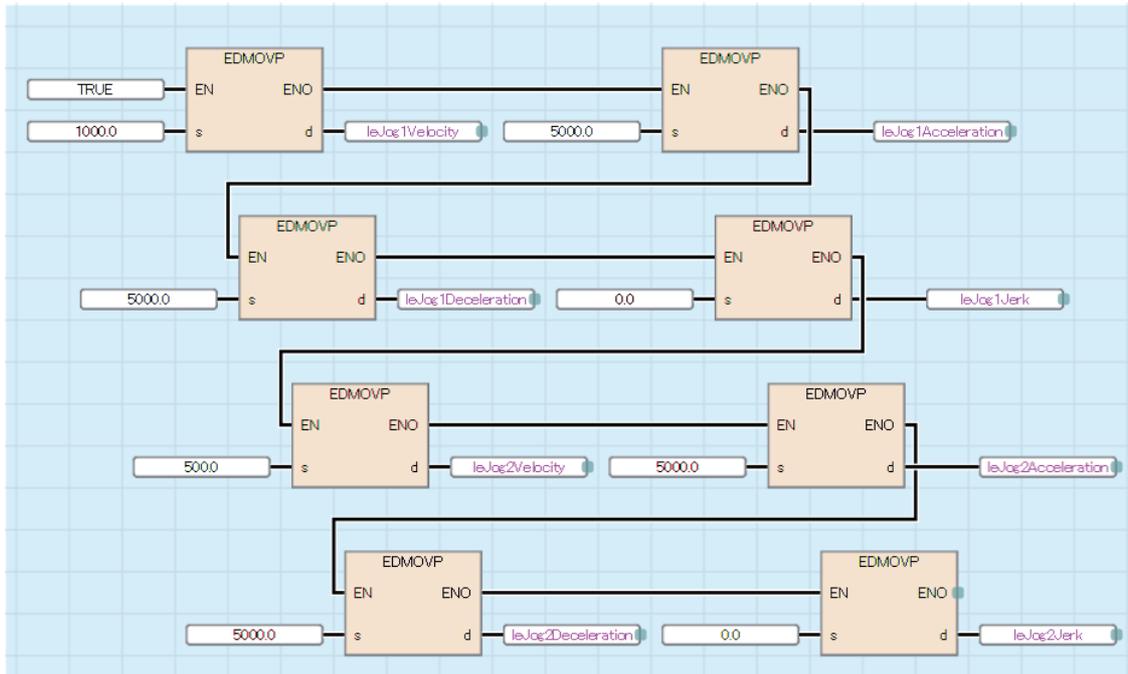
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1
2	Axis0002	AXIS_REF	轴2
3	Axis0003	AXIS_REF	轴3

■使用的标签

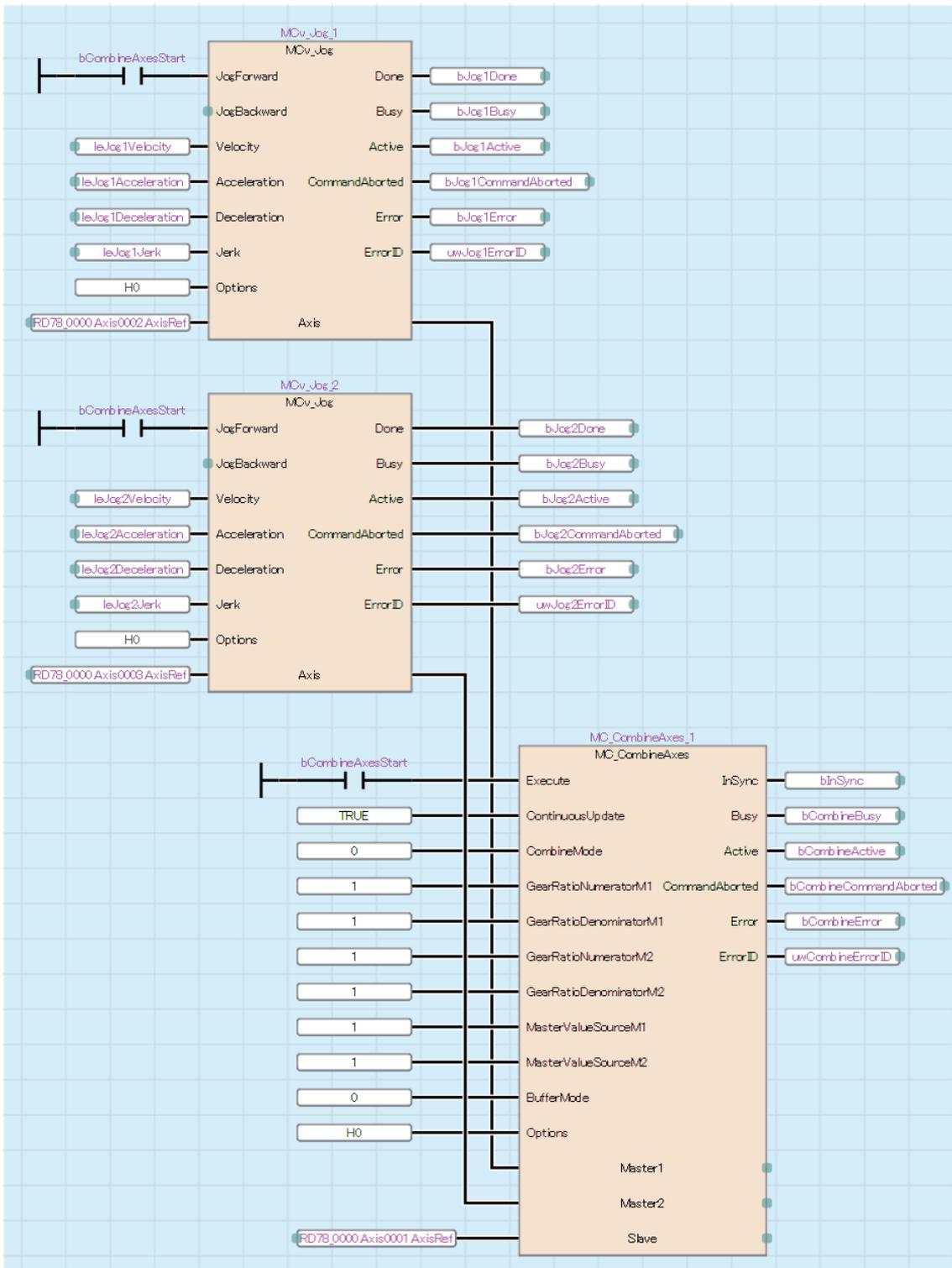
标签名	数据类型	注释
MC_CombineAxes_1	MC_CombineAxes	加减法定位FB
bCombineAxesStart	位	加减法定位开始指令
bInSync	位	同步中
bCombineBusy	位	执行中
bCombineActive	位	控制中
bCombineCommandAborted	位	执行中断
bCombineError	位	出错
uwCombineErrorID	字[无符号]/位串[16位]	出错代码
MCv_Jog_1	MCv_Jog	JOG运行1FB
leJog1Velocity	双精度实数	JOG速度
leJog1Acceleration	双精度实数	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	出错
uwJog1ErrorID	字[无符号]/位串[16位]	出错代码
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	出错
uwJog2ErrorID	字[无符号]/位串[16位]	出错代码

■FBD/LD的程序(CPU模块侧)

- JOG运行用数据设置



• JOG运行1/JOG运行2/加减法定位



■ST的程序(运动模块侧)

```
//-----JOG运行用数据设置-----
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
);

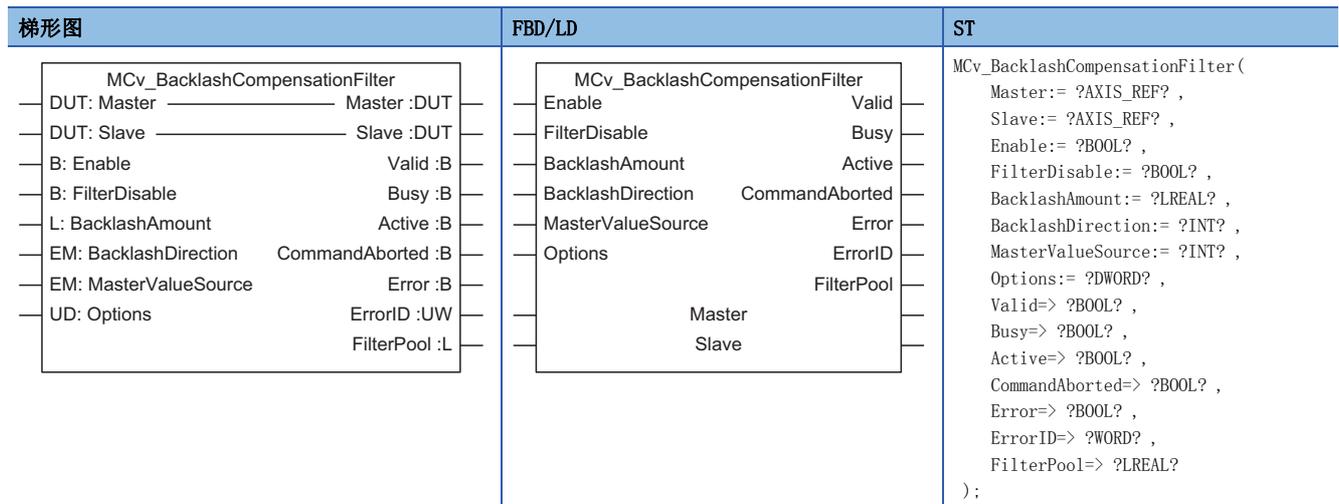
//-----JOG运行2-----
MCv_Jog_2(
    Axis:= Axis0003.AxisRef ,
    JogForward:= bCombineAxesStart ,
    Velocity:= leJog2Velocity ,
    Acceleration:= leJog2Acceleration ,
    Deceleration:= leJog2Deceleration ,
    Jerk:= leJog2Jerk ,
    Options:= H00000000 ,
    Done=> bJog2Done ,
    Busy=> bJog2Busy ,
    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  
);
```

MCv_BacklashCompensationFilter

进行按照移动方向，对机械系统的背隙量进行补偿的滤波器处理。



名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
背隙补偿滤波器	40	24	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23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.)，请参阅下述章节。 ☞ 23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(背隙补偿滤波器)。
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	背隙量	LREAL	启动时	0.0~2500000000.0 [U]	0.0	设置机械系统的背隙(间隙)量。
BacklashDirection	背隙补偿方向	INT (MC_DIRECTION)	启动时	1、2	0	设置进行背隙补偿的方向。 • 1: 正方向 (mcPositiveDirection) • 2: 负方向 (mcNegativeDirection) *: 省略了设置的情况下，将变为“超出背隙补偿方向范围(出错代码: 34A0H)”。 关于详细内容，请参阅下述章节。 ☞ 341页 背隙补偿方向(BacklashDirection)

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、101	1	设置主轴(Master)的数据源。 <ul style="list-style-type: none"> • 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。 341页 主轴数据源选择(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。 342页 选项(Options)

■输出变量

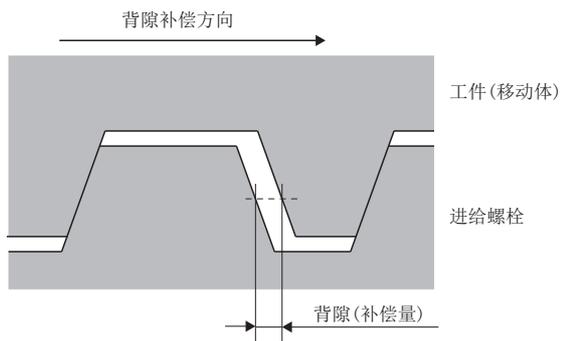
输出变量	名称	数据类型	初始值	内容
Valid	输出值有效	BOOL	FALSE	输出值为有效时, 将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_BacklashCompensationFilter(背隙补偿滤波器)时, 将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_BacklashCompensationFilter(背隙补偿滤波器)正在控制轴时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MCv_BacklashCompensationFilter(背隙补偿滤波器)的执行中断时, 将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时, 将返回出错代码。 关于出错代码的详细内容, 请参阅下述手册。 MELSEC iQ-R运动模块用户手册(应用篇)
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。 背隙滤波器的情况下, 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

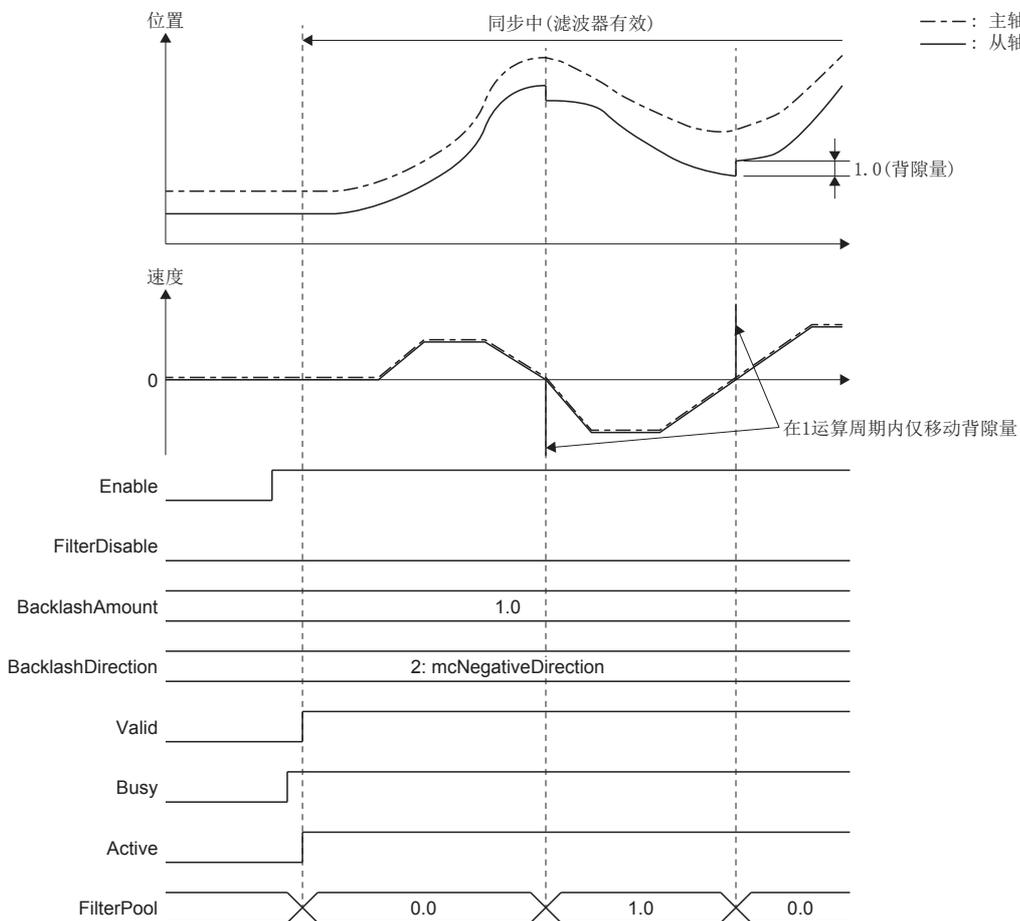
公开变量	名称	数据类型	初始值	内容
PurgedFilterPool	删除的滤波器累计值	LREAL	0.0	通过将滤波器无效(FilterDisable)置为TRUE存储删除的滤波器累计值(FilterPool)。 删除的滤波器累计值(PurgedFilterPool)不累计。 在FB启动时将被清零。

功能

- 背隙补偿滤波器是进行机械系统的背隙(间隙)量的补偿的滤波器。

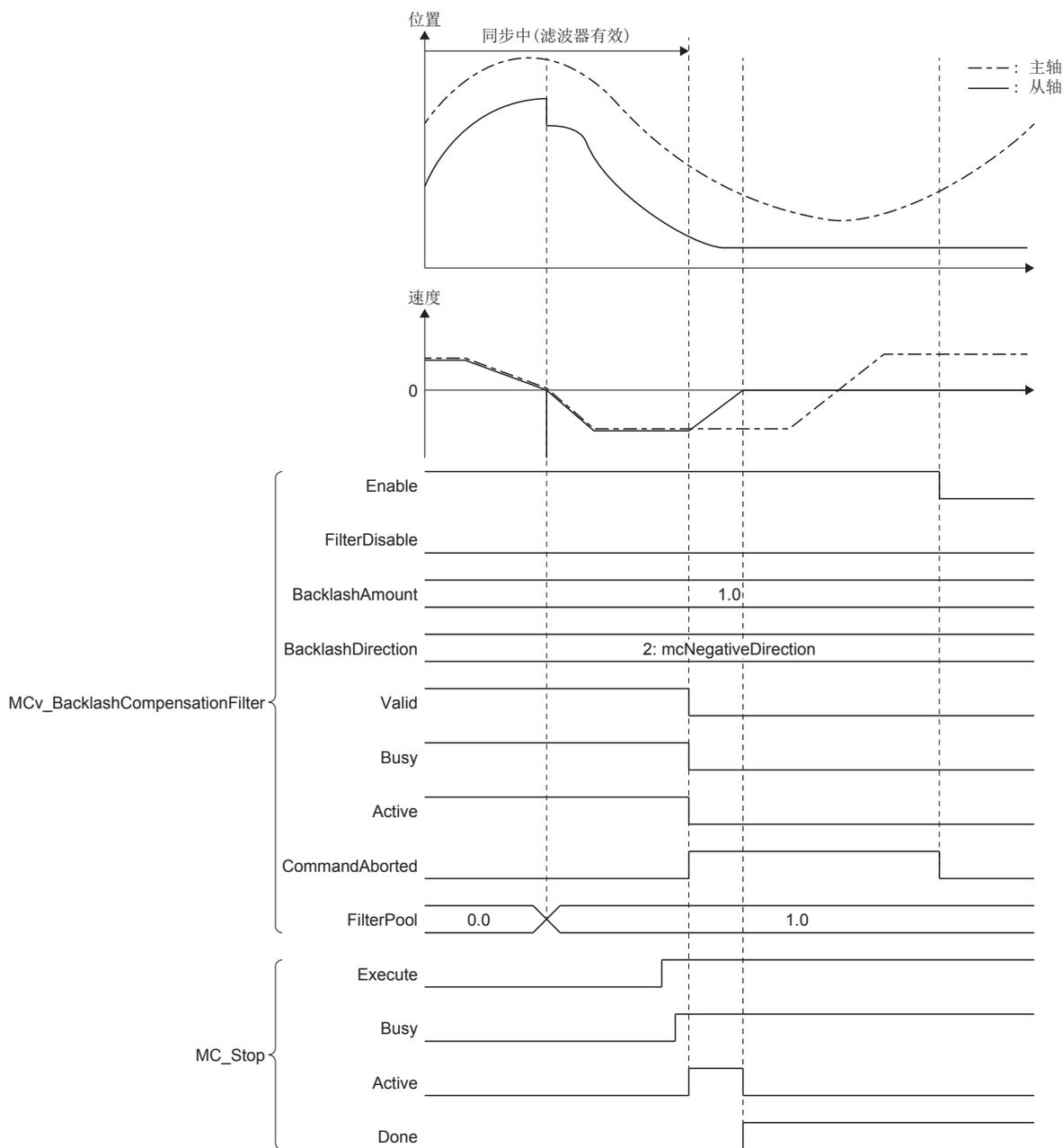


- 主轴(Master)的移动方向与背隙补偿方向(BacklashDirection)相同的情况下，对于将主轴(Master)的当前位置向背隙补偿方向(BacklashDirection)的方向进行了背隙量(BacklashAmount)的移位的位置，从轴(Slave)进行同步。主轴(Master)的移动方向与背隙补偿方向(BacklashDirection)相反的情况下，从轴(Slave)将与主轴(Master)的当前位置进行同步。每当主轴(Master)的移动方向改变时仅按背隙量(BacklashAmount)发生额外的从轴(Slave)的移动量，并进行背隙补偿。



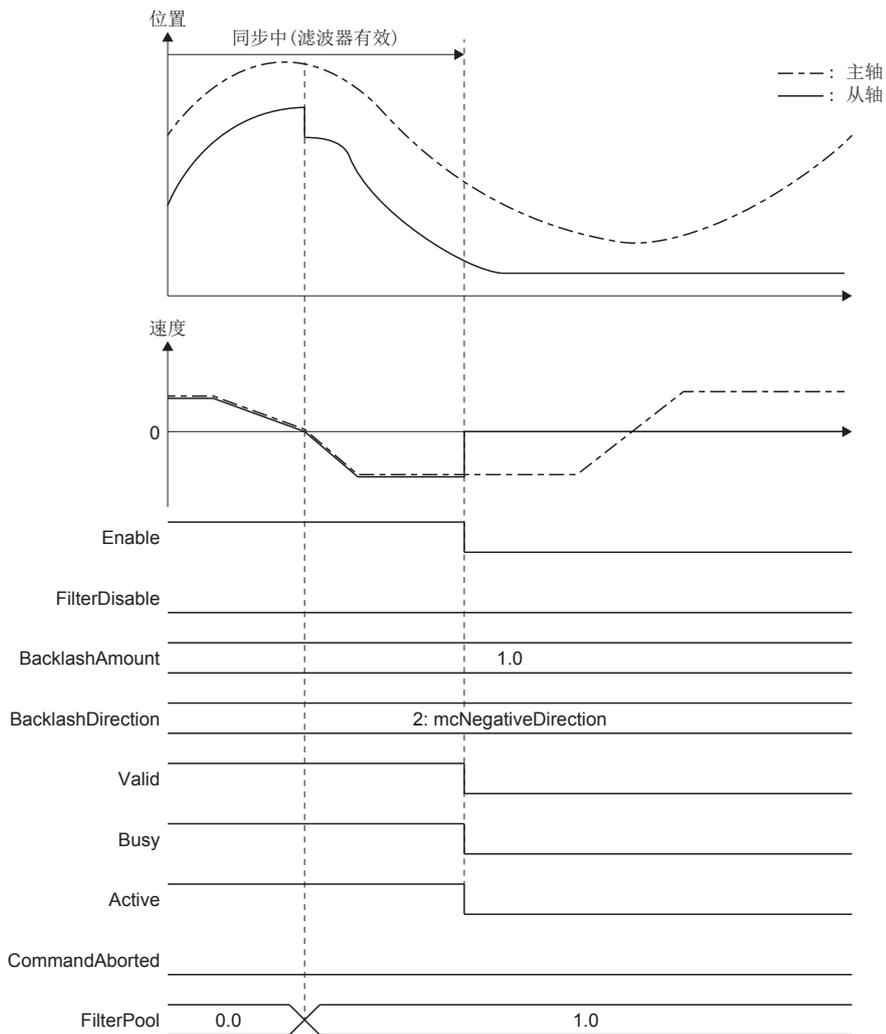
- 背隙补偿滤波器固有的停止动作如下所示。

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。



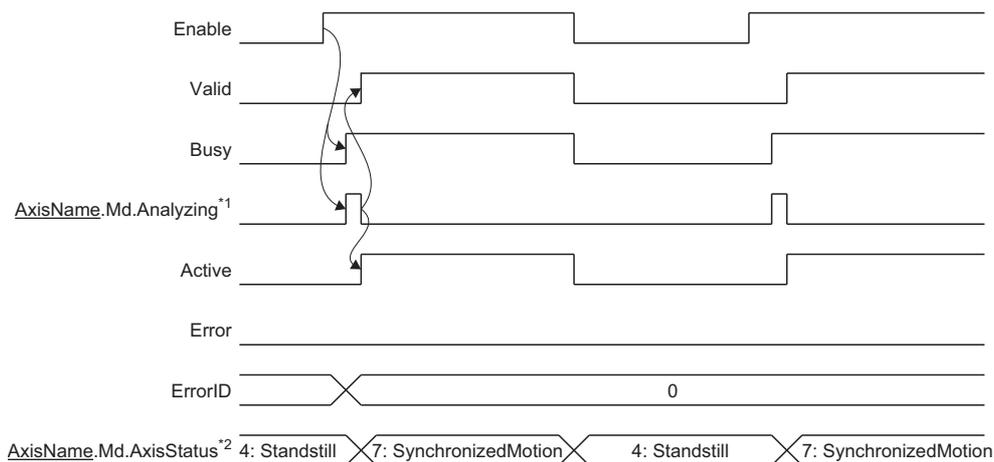
- 滤波器无效时(有效(Enable)下降沿时)的动作

有效(Enable)的下降沿时从轴(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。

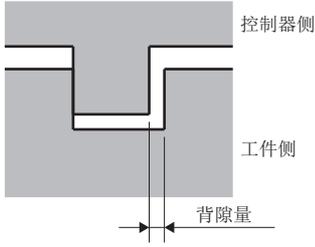
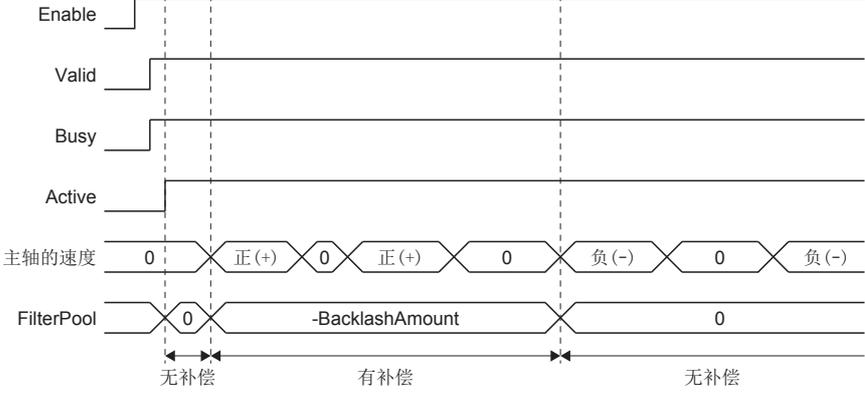
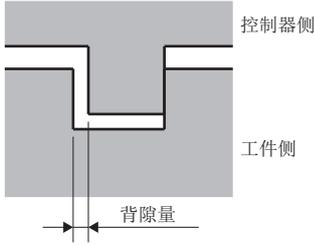
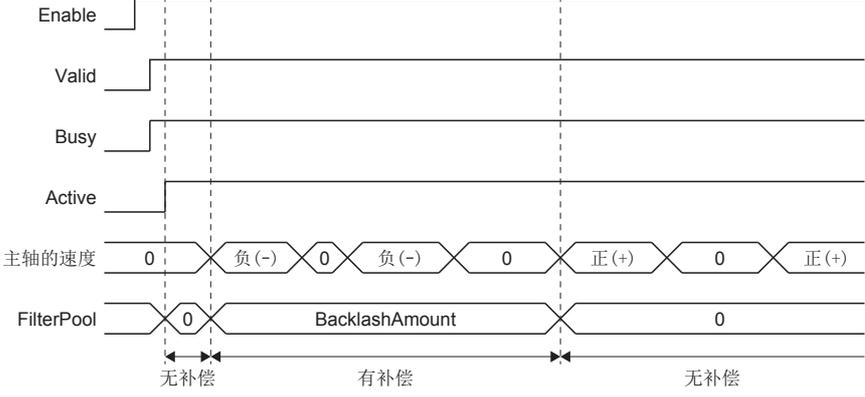
☞ 13页 通过有效(Enable)类型的运动控制FB的基本动作

■背隙补偿方向(BacklashDirection)

设置进行背隙补偿的方向。

主轴(Master)的移动方向为背隙补偿方向(BacklashDirection)的情况下,将补偿量输出到从轴(Slave)中。

背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,机械系统的背隙状态处于下图的关系时则进行补偿。

设置值	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背隙量(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背隙量(BacklashAmount)的移位的位置进行同步。 

■主轴数据源选择(MasterValueSource)

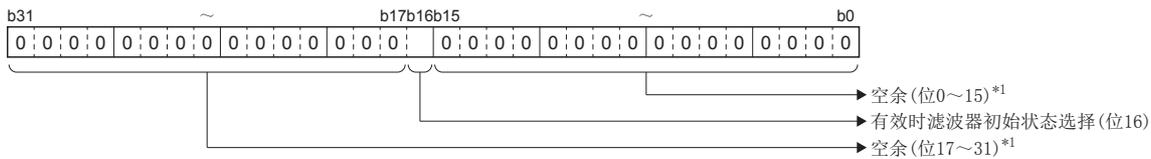
设置从轴(Slave)执行单轴同步控制的主轴(Master)的位置的类型。

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。

设置值	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 (背隙补偿滤波器) 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(出错代码: 1A4EH)”。

位	名称	内容
16	有效时滤波器初始状态选择	设置背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(出错代码: 1A4EH)”且不启动。

• 有效时滤波器初始状态选择(位16)

设置值	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背隙补偿的位置关系时则开始滤波器动作。

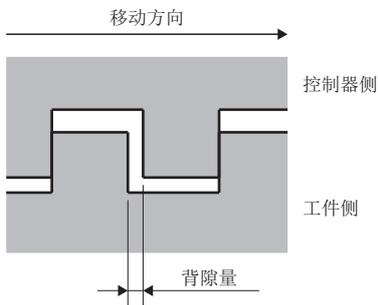
■将背隙补偿滤波器置为有效的步骤

将背隙补偿滤波器置为有效的步骤的使用示例如下所示。

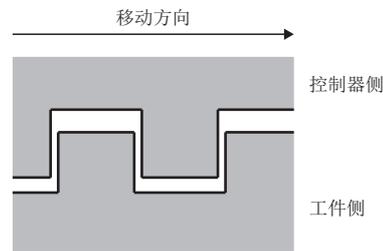
通过原点复位控制确定了机械位置后, 及背隙补偿滤波器途中停止后重新开始的情况下应进行此步骤。

1. 应设置为将实驱动轴通过原点复位控制及JOG运行等进行移动, 按下图的正确示例所示向一个方向产生了机械系统的背隙的状态。

■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驱动轴的指令当前位置一致。

3. 将步骤1. 中实驱动轴的移动方向的反方向设置为背隙补偿方向(BacklashDirection), 启动背隙补偿滤波器。后段中有功能块的情况下也启动这些功能块。

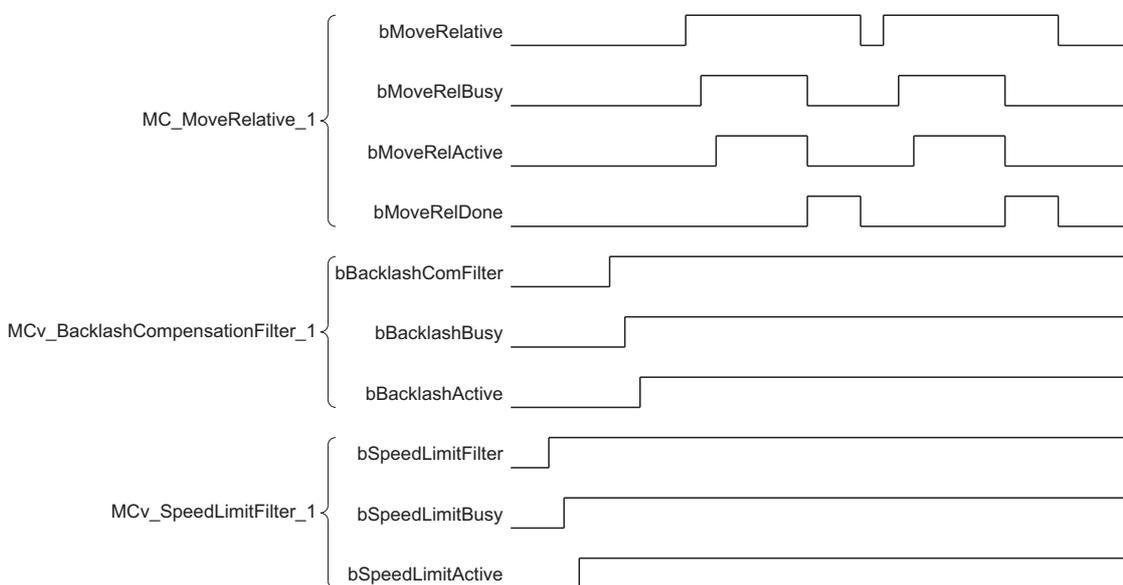
注意事项

- 背隙补偿是以指令方向及背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
 - 背隙量根据机械位置变化的机构(齿条&齿轮机构等)
- 在主轴(Master)的移动方向改变的瞬间从轴(Slave)将在1运算周期内仅移动背隙量(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,因此应在背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)将为加上了背隙量(BacklashAmount)的补偿的位置。主轴(Master)的当前位置不加上背隙量(BacklashAmount)的补偿。
- 指令滤波器的动作中进行滤波器无效(FilterDisable)的切换时,从轴(Slave)的速度有可能急剧变化,因此应加以注意。
- 将指令滤波器置为了有效后,从轴(Slave)中发生停止原因导致未将主轴(Master)的指令传送到从轴(Slave)及后段的同步控制功能块的情况下,同步位置关系将被破坏。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

在背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	设置值	
背隙量	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

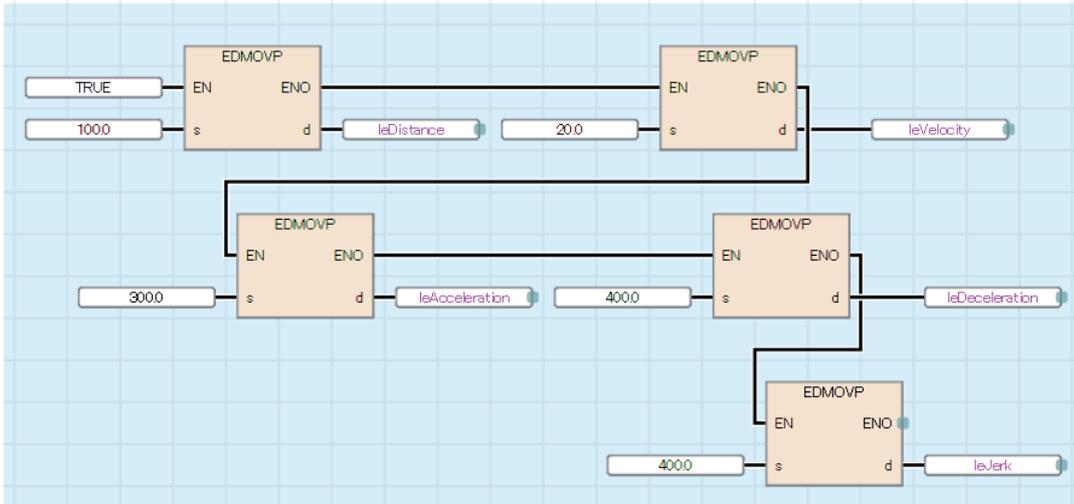
轴No.	标签名	数据类型	注释
1	VirtualAxis0001	AXIS_REF	轴1(虚拟驱动轴)
2	VirtualAxis0002	AXIS_REF	轴2(虚拟驱动轴)
3	Axis0003	AXIS_REF	轴3(实驱动轴)

■使用的标签

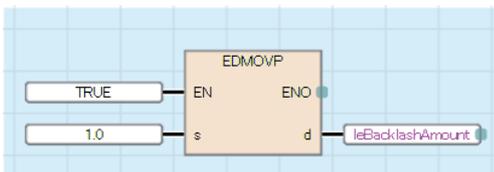
标签名	数据类型	注释
MC_MoveRelative_1	MC_MoveRelative	相对值定位FB
bMoveRelDone	位	单轴相对值定位启动
leDistance	双精度实数	移动量
leVelocity	双精度实数	速度
leAcceleration	双精度实数	加速度
leDeceleration	双精度实数	减速度
leJerk	双精度实数	Jerk
bMoveRelDone	位	执行完成
bMoveRelBusy	位	执行中
bMoveRelActive	位	控制中
bMoveRelCommandAborted	位	执行中断
bMoveRelError	位	出错
uwMoveRelErrorID	字[无符号]/位串[16位]	出错代码
MCv_BacklashCompensationFilter_1	MCv_BacklashCompensationFilter	背隙补偿滤波器FB
bBacklashComFilter	位	背隙补偿滤波器指令
leBacklashAmount	双精度实数	背隙量
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	出错
uwBacklashErrorID	字[无符号]/位串[16位]	出错代码
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	出错
uwSpeedLimitErrorID	字[无符号]/位串[16位]	出错代码
leFilterPool	双精度实数	滤波器累计值

■FBD/LD的程序(CPU模块侧)

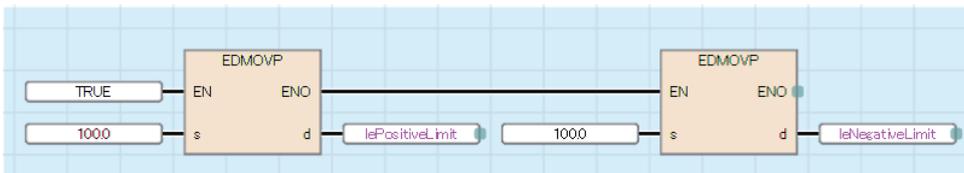
- 单轴定位用数据设置



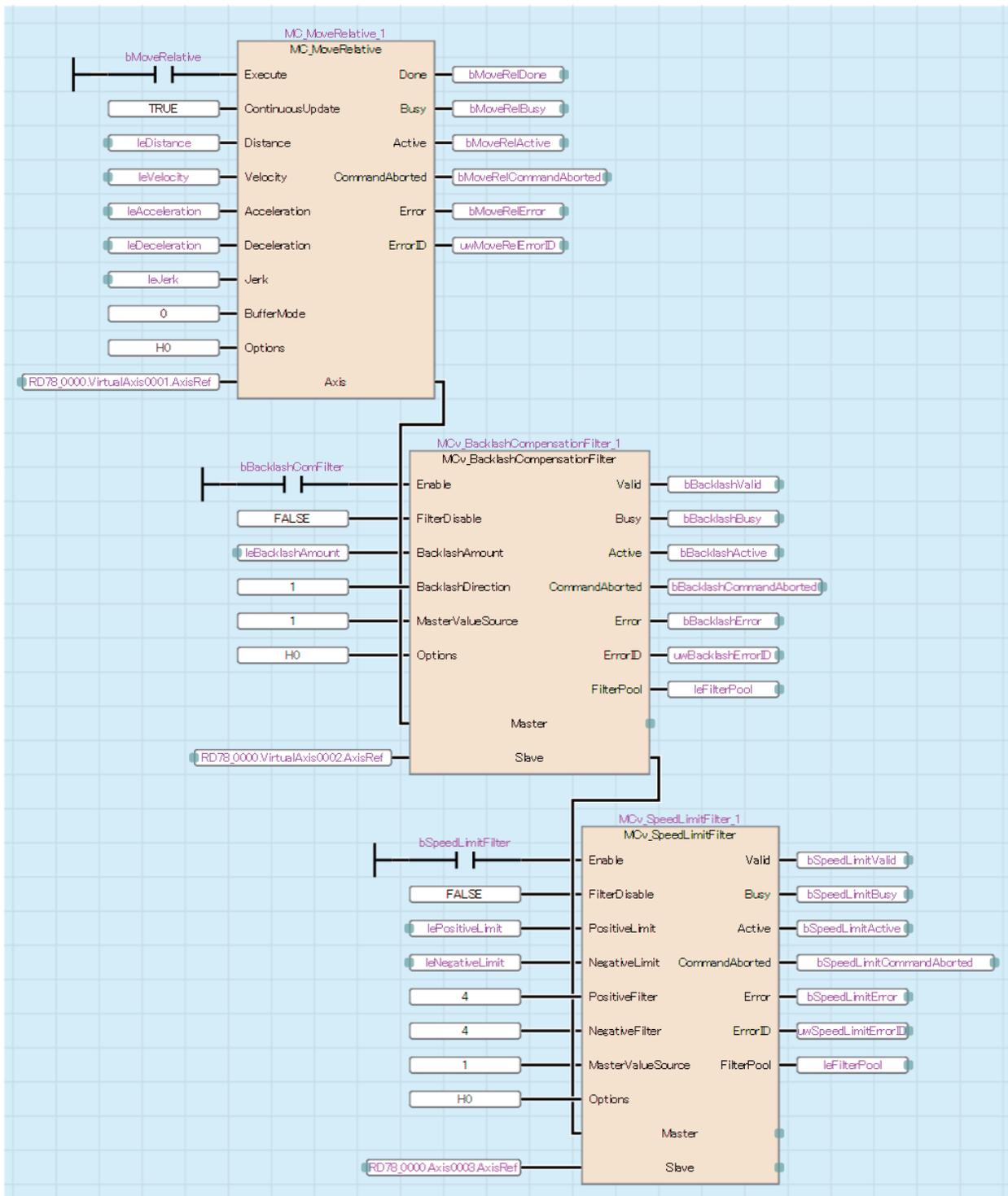
- 背隙量



- 正方向/负方向限制值



• 相对值定位/背隙补偿滤波器/速度限制滤波器



■ST的程序(运动模块侧)

```
//-----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
);

//-----背隙量-----
leBacklashAmount:= 1.0;

//-----背隙补偿滤波器-----
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;
```

```

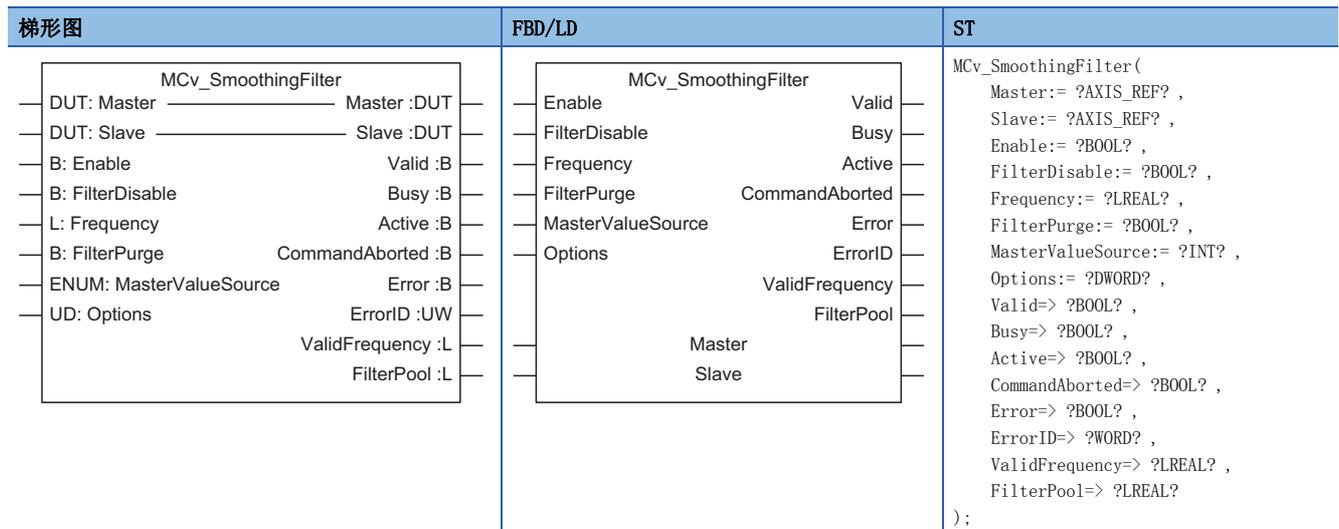
leNegativeLimit:= 100.0;

//-----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
);

```

MCv_SmoothingFilter

按照指定的频率，进行滤波器处理。



名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
平滑滤波器	40	24	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23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.)，请参阅下述章节。 ☞ 23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。 ☞ 352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。 ☞ 352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。 ☞ 353页 滤波器净化(FilterPurge)

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。  354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(出错代码: 1A4EH)”。

■输出变量

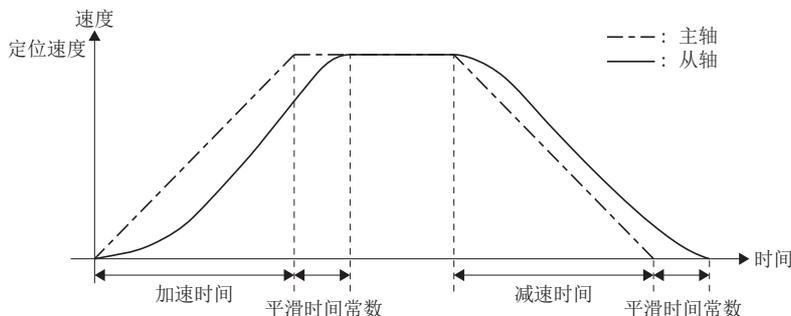
输出变量	名称	数据类型	初始值	内容
Valid	输出值有效	BOOL	FALSE	输出值为有效时, 将变为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_SmoothingFilter(平滑滤波器)时, 将变为TRUE。
Active	控制中	BOOL	FALSE	MCv_SmoothingFilter(平滑滤波器)正在控制轴时, 将变为TRUE。
CommandAborted	执行中断	BOOL	FALSE	MCv_SmoothingFilter(平滑滤波器)的执行中断时, 将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时, 将返回出错代码。 关于出错代码的详细内容, 请参阅下述手册。  MELSEC iQ-R运动模块用户手册(应用篇)
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(警告代码: 0D13H)”。 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

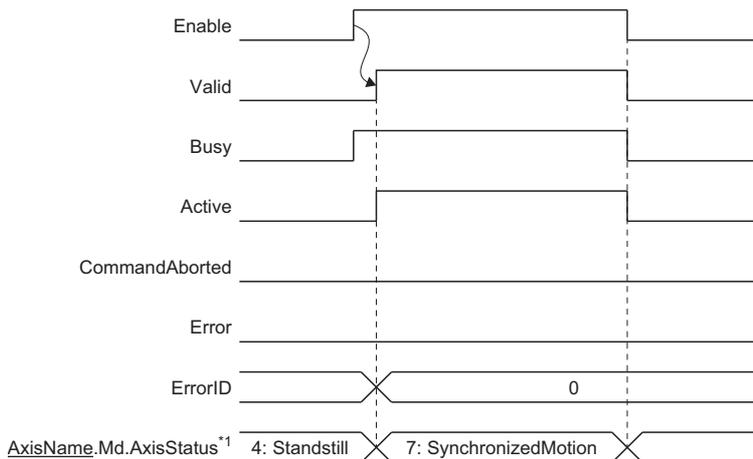
- 平滑滤波器可以去除高于设置的频率(Frequency)的频率，是使高于设置值的整个波形变为平滑的加减速波形的滤波器。
- 平滑滤波器将对主轴数据源选择(MasterValueSource)中设置的主轴(Master)的值执行了滤波器处理的结果输出到从轴(Slave)中。
- 平滑滤波器执行中的从轴(Slave)的轴状态(AxisName.Md.AxisStatus)将变为“7: 同步运行中(SynchronizedMotion)”。
- 平滑的时间常数将变为“ $1/\text{频率(Frequency)}[\text{s}]$ ”，加速时间及减速时间将仅按照平滑时间常数的量变长。频率(Frequency)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(出错代码: 1AADH)”。
- 应在从轴(Slave)的轴状态(AxisName.Md.AxisStatus)为“4: 待机中(Standstill)”时进行启动。
- 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。

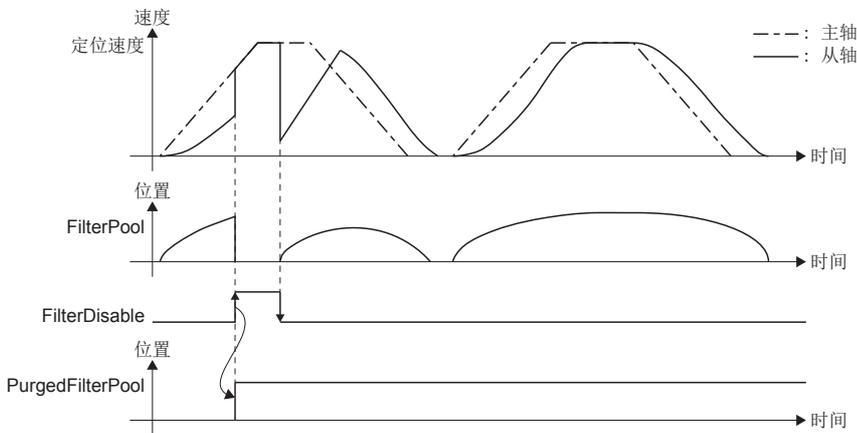
☞ 13页 通过有效(Enable)类型的运动控制FB的基本动作

■滤波器无效(FilterDisable)

设置滤波器处理的方法。

设置值	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。

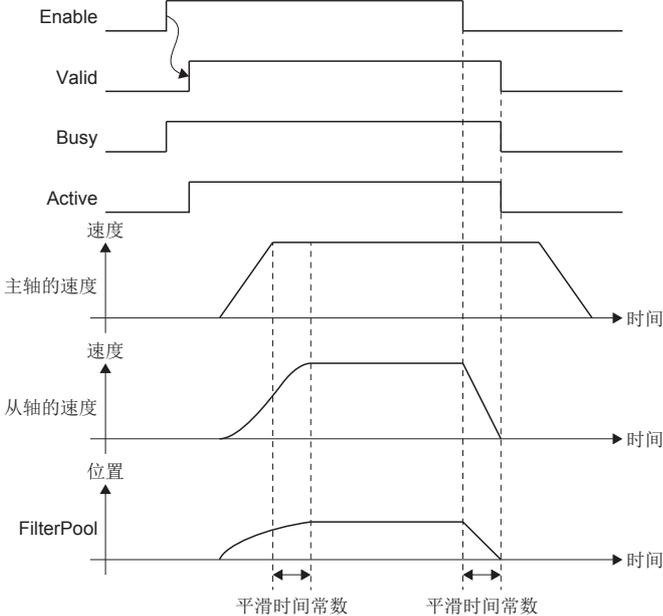
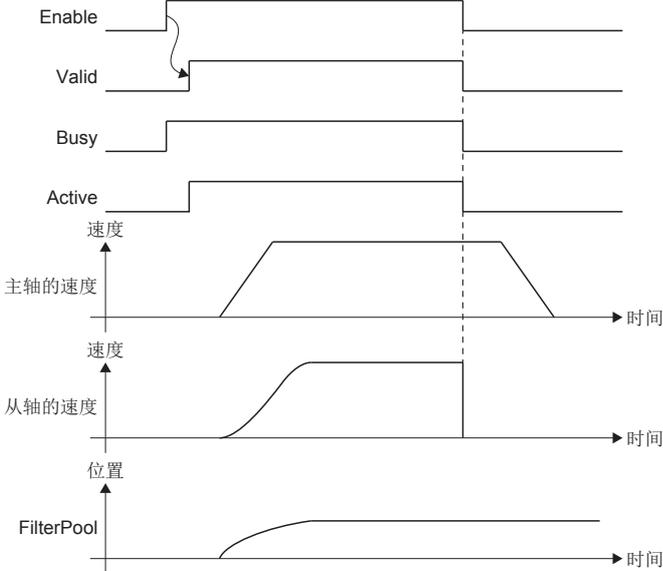
频率(Frequency)的可设置的范围如下所示。

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(出错代码: 1A85H)”。
- 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(出错代码: 1A85H)”。
- 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，按滤波器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。

设置值	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>  <p>速度 ↑ 主轴的速度 → 时间</p> <p>速度 ↑ 从轴的速度 → 时间</p> <p>位置 ↑ FilterPool → 时间</p> <p>平滑时间常数 平滑时间常数</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>  <p>速度 ↑ 主轴的速度 → 时间</p> <p>速度 ↑ 从轴的速度 → 时间</p> <p>位置 ↑ FilterPool → 时间</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。

设置值	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如下所示。

运动模块	运算负载
RD78G	380μs左右
RD78GH	240μs左右

- 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。发生存储器不足的情况下，应更改插件MotionControl_AxisFilter参数 (Addon_MotionControl_AxisFilter) 的RAM最大容量 (System.PrConst.Addon_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。

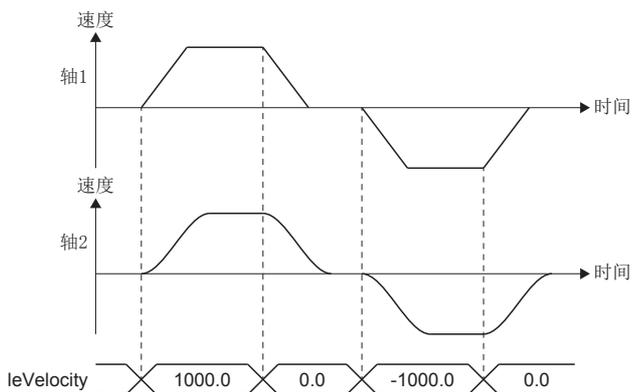
$$\sum_{i=0}^{\text{滤波器级数} - 1} (\text{i 运算周期前的主轴的位置} - \text{控制开始时的主轴的位置})$$

- 进行主轴的移动方向在各运算周期重复交替正方向、负方向的动作时，有可能在内部运算中发生浮点的位遗漏，且主轴与从轴的位置可能会发生误差。
- 指令滤波器的动作中进行滤波器无效 (FilterDisable) 的切换时，从轴 (Slave) 的速度有可能急剧变化，因此应加以注意。
- 将指令滤波器置为了有效后，从轴 (Slave) 中发生停止原因导致未将主轴 (Master) 的指令传送到从轴 (Slave) 及后段的同步控制功能块的情况下，同步位置关系将被破坏。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	设置值	
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

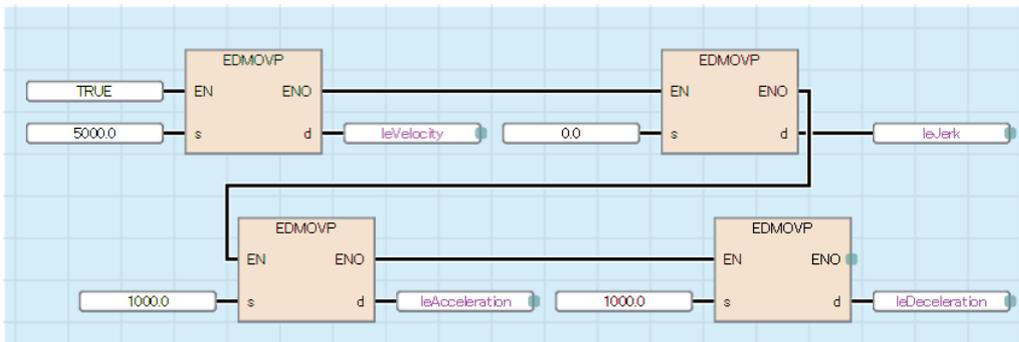
■ 使用的标签

标签名	数据类型	注释
MCv_SpeedControl_1	MCv_SpeedControl	速度控制控制(包含一个循环)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	出错
uwSpeedErrorID	字[无符号]/位串[16位]	出错代码
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	出错
uwSmoothingErrorID	字[无符号]/位串[16位]	出错代码
leSmoothingValidFrequency	双精度实数	有效频率

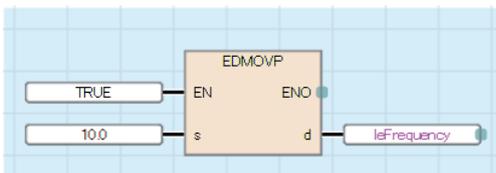
标签名	数据类型	注释
leSmoothingFilterPool	双精度实数	滤波器累计值

■FBD/LD的程序(CPU模块侧)

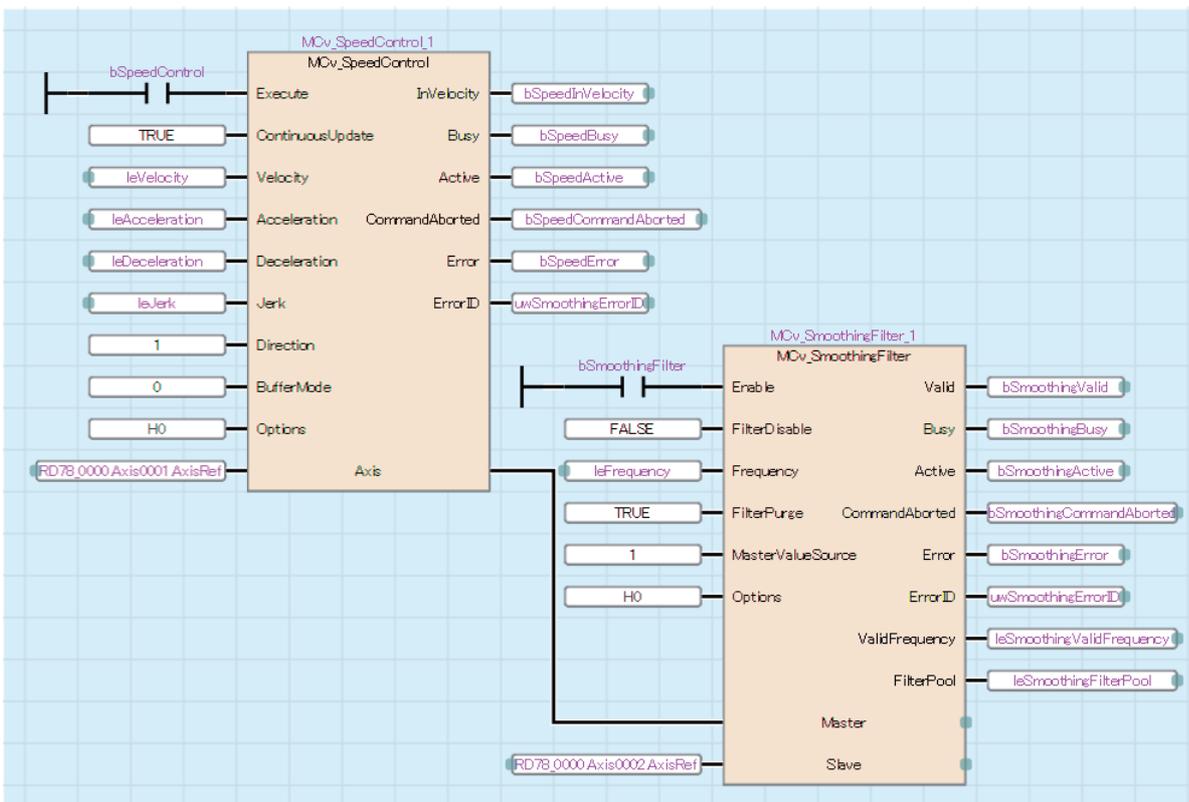
- 速度控制用数据设置



- 频率



- 速度控制/平滑滤波器



■ST的程序(运动模块侧)

```
//-----速度控制用数据设置-----
leVelocity:= 5000.0;
leAcceleration:= 1000.0;
leDeceleration:= 1000.0;
leJerk:= 0.0;

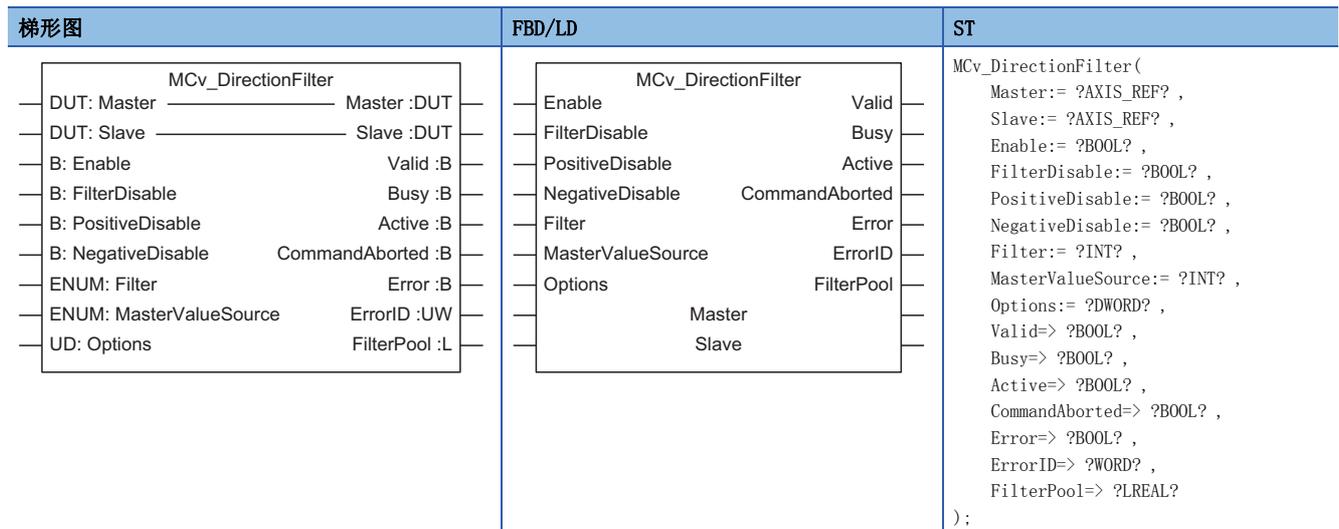
//-----速度控制-----
MCv_SpeedControl_1(
  Axis:= Axis0001.AxisRef ,
  Execute:= bSpeedControl ,
  ContinuousUpdate:= TRUE ,
  Velocity:= leVelocity ,
  Acceleration:= leAcceleration ,
  Deceleration:=leDeceleration ,
  Jerk:= leJerk ,
  Direction:= MC_DIRECTION__mcPositiveDirection ,
  BufferMode:= MC_BUFFER_MODE__mcAborting ,
  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
);
```

MCv_DirectionFilter

进行对设置的移动方向进行移动限制的滤波器处理。



名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
移动方向限制滤波器	28	28	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23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.)，请参阅下述章节。 ☞ 23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。 ☞ 361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，将对正方向的轴动作进行限制。主轴(Master)的正转动动作时从轴(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
NegativeDisable	负方向限制	BOOL	启动时	TRUE、FALSE	FALSE	<p>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。</p> <ul style="list-style-type: none"> • FALSE: 无限制 • TRUE: 有限制
Filter	滤波器动作	INT (MC_VELOCITY_LIMIT_MODE)	启动时	0、2、4	2	<p>设置滤波器动作。</p> <ul style="list-style-type: none"> • 0: 忽略(Ignore) • 2: 舍去(Truncate) • 4: 限定(无减速时斜率)(ClampWithoutRamp) <p>关于详细内容，请参阅下述章节。 ☞ 361页 滤波器动作(Filter)</p>
MasterValueSource	主轴数据源选择	INT (MC_SOURCE)	启动时	1、2、101、102	1	<p>设置主轴(Master)的数据源。</p> <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) <p>关于详细内容，请参阅下述章节。 ☞ 362页 主轴数据源选择(MasterValueSource)</p>
Options	选项	DWORD(HEX)	启动时	0000000H	0000000H	<p>应设置“0000000H”。</p> <p>*: 设置了“0000000H”以外的情况下，将变为“超出Options范围(出错代码: 1A4EH)”。</p>

■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	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD(UINT)	0	<p>发生了异常时，将返回出错代码。</p> <p>关于出错代码的详细内容，请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册(应用篇)</p>
FilterPool	滤波器累计值	LREAL	0.0	<p>输出滤波器处理中累计的从轴的移动量。</p> <p>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。</p> <p>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(警告代码: 0D13H)”。</p> <p>检测出警告时，滤波器累计值(FilterPool)的误差可能会变大。</p>

■公开变量

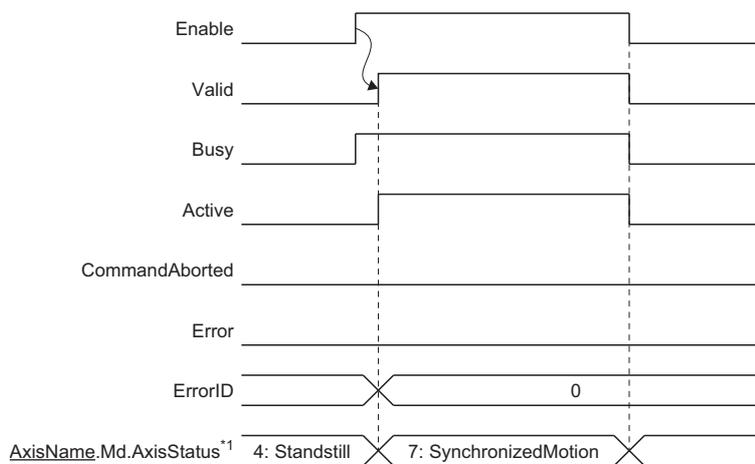
公开变量	名称	数据类型	初始值	内容
PurgedFilterPool	删除的滤波器累计值	LREAL	0.0	<p>通过将滤波器无效(FilterDisable)置为TRUE存储删除的滤波器累计值(FilterPool)。</p> <p>删除的滤波器累计值(PurgedFilterPool)不累计。在FB启动时将被清零。</p>

功能

- 移动方向限制滤波器是相对于主轴的移动，将从轴的移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)将变为“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(出错代码: 1AADH)”。
- 应在从轴(Slave)的轴状态(AxisName. Md. AxisStatus)为“4: 待机中(Standstill)”时进行启动。
- 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。

☞ 13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。

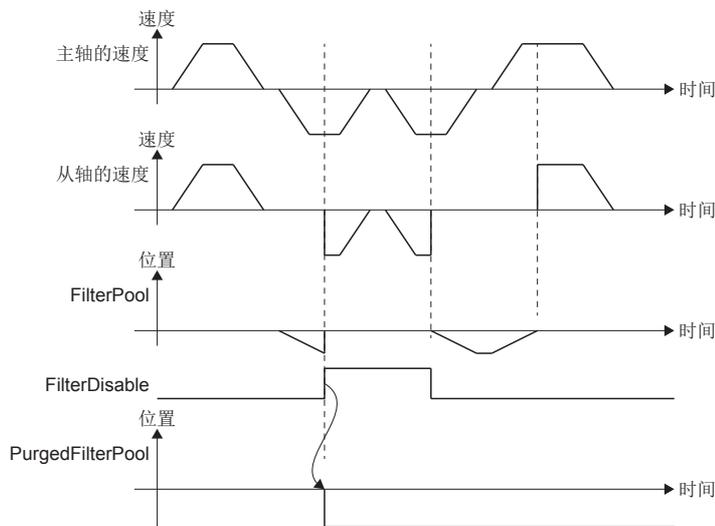
设置值	内容
FALSE	按照滤波器设置执行动作。
TRUE	滤波器将变为无效，将主轴(Master)的输入原样不变地传送到从轴(Slave)。

- 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	设置值
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。

设置值	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> <p>滤波器动作(Filter)为“2: 舍去(Truncate)”的情况下不更新</p>

设置值	内容
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> <p>作为滤波器累计值累计,且变为至允许方向的输入移动量时被反映</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(出错代码:1A86H)”。

■ 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。

设置值	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。

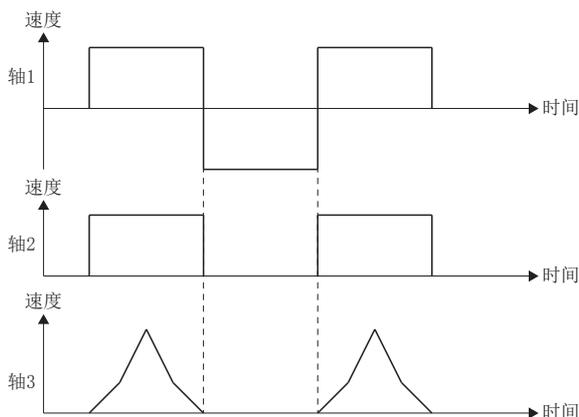
注意事项

- 指令滤波器的动作中进行滤波器无效(FilterDisable)的切换时,从轴(Slave)的速度有可能急剧变化,因此应加以注意。
- 将指令滤波器置为了有效后,从轴(Slave)中发生停止原因导致未将主轴(Master)的指令传送到从轴(Slave)及后段的同步控制功能块的情况下,同步位置关系将被破坏。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	设置值
凸轮ID设置	1

■ 轴

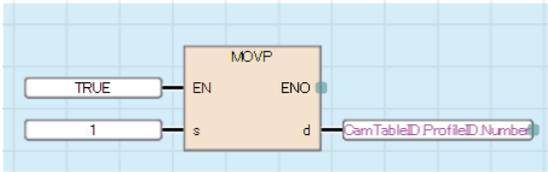
轴No.	标签名	数据类型	注释
1	Axis0001	AXIS_REF	轴1
2	Axis0002	AXIS_REF	轴2
3	Axis0003	AXIS_REF	轴3

■ 使用的标签

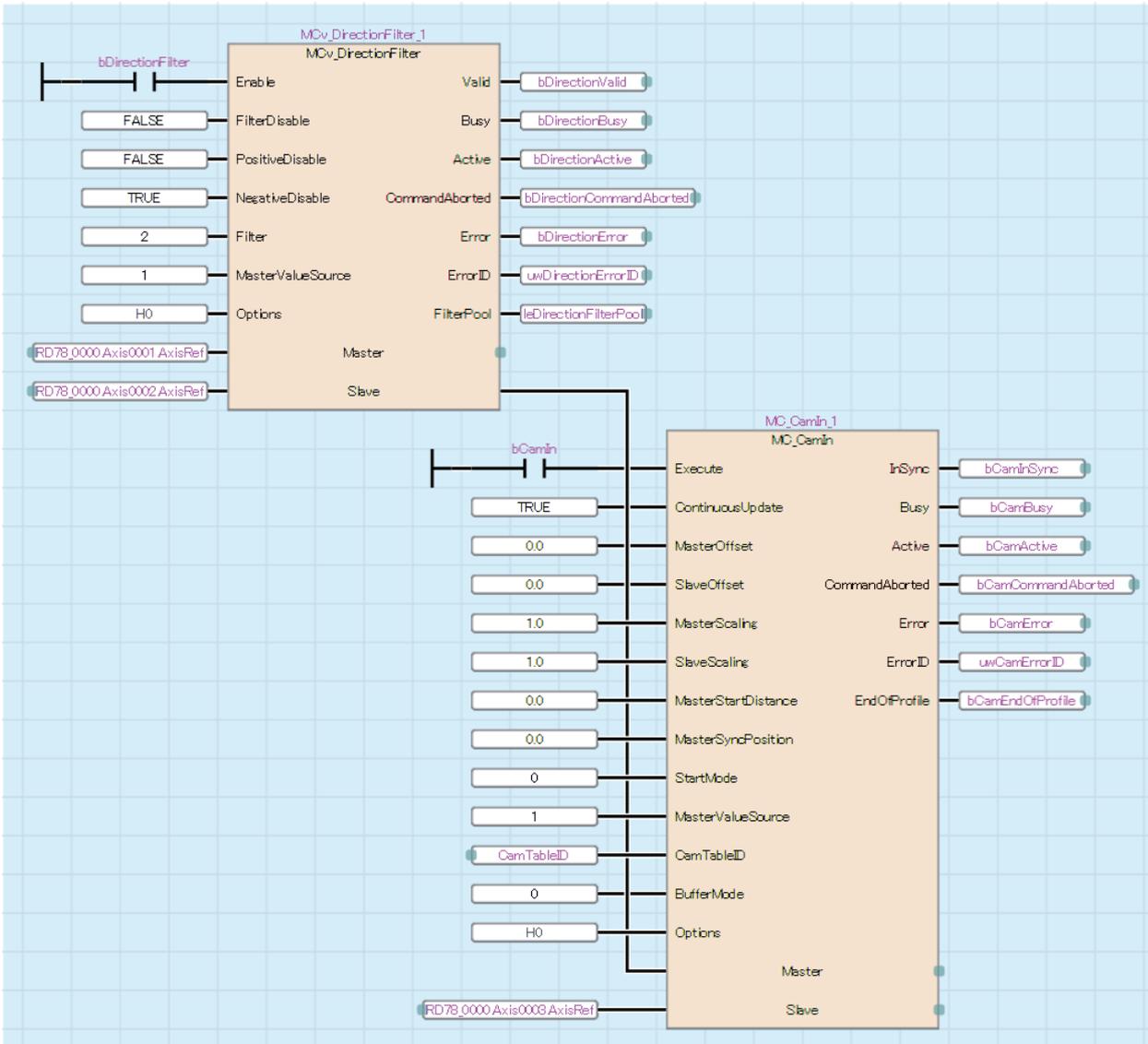
标签名	数据类型	注释
MCv_DirectionFilter_1	MCv_DirectionFilter	移动方向限制滤波器FB
bDirectionFilter	位	移动方向限制滤波器指令
bDirectionValid	位	输出值有效
bDirectionBusy	位	执行中
bDirectionActive	位	控制中
bDirectionCommandAborted	位	执行中断
bDirectionError	位	出错
uwDirectionErrorID	字[无符号]/位串[16位]	出错代码
leDirectionFilterPool	双精度实数	删除的滤波器累计值
MC_CamIn_1	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	出错
uwCamErrorID	字[无符号]/位串[16位]	出错代码
bCamEndOfProfile	位	凸轮循环完成

■FBD/LD的程序(CPU模块侧)

- 凸轮ID设置



- 移动方向限制滤波器/凸轮动作



■ST的程序(运动模块侧)

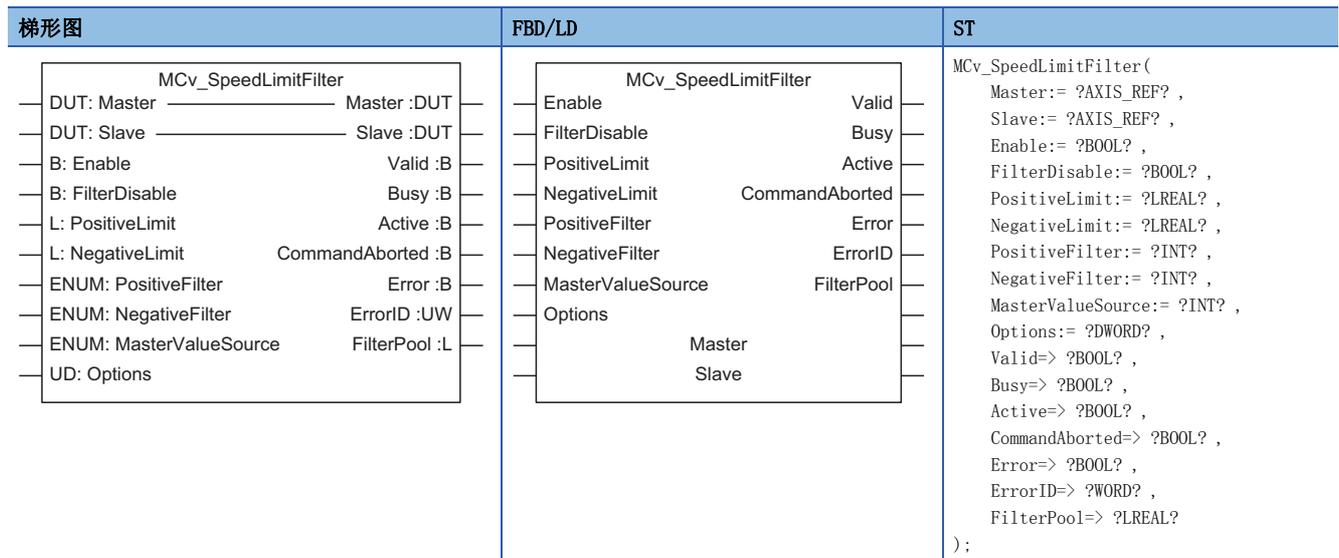
```
//-----移动方向限制滤波器-----
MCv_DirectionFilter_1(
    Master:= Axis0001.AxisRef ,
    Slave:= Axis0002.AxisRef ,
    Enable:= bDirectionFilter ,
    FilterDisable:= FALSE ,
    PositiveDisable:= FALSE ,
    NegativeDisable:= TRUE ,
    Filter:= MC_VELOCITY_LIMIT_MODE__Truncate ,
    MasterValueSource:= 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
);
```

MCv_SpeedLimitFilter

进行将速度限制为设置的限制值的滤波器处理。



名称	输入区域点数(byte)	输出区域点数(byte)	编译方式	FB动作
速度限制滤波器	52	28	子程序型	随时执行型

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.)，请参阅下述章节。 ☞ 23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.)，请参阅下述章节。 ☞ 23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。 ☞ 369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(出错代码: 1A87H)”。
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(出错代码: 1A87H)”。

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。 369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。 369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。 370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 (出错代码: 1A4EH)”。

■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	出错	BOOL	FALSE	发生了异常时, 将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时, 将返回出错代码。 关于出错代码的详细内容, 请参阅下述手册。 MELSEC iQ-R运动模块用户手册 (应用篇)
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 (警告代码: 0D13H)”。 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

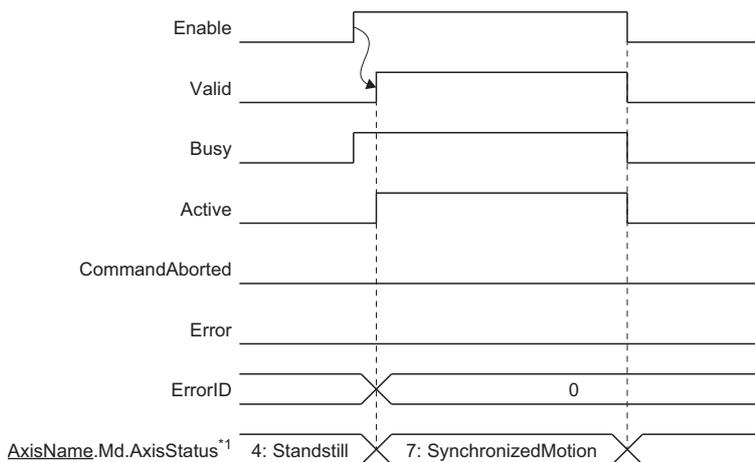
公开变量	名称	数据类型	初始值	内容
PurgedFilterPool	删除的滤波器累计值	LREAL	0.0	通过将滤波器无效 (FilterDisable) 置为TRUE存储删除的滤波器累计值 (FilterPool)。 删除的滤波器累计值 (PurgedFilterPool) 不累计。 在FB启动时将被清零。

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) 将变为 “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 (出错代码: 1AADH)”。
- 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。

☞ 13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。

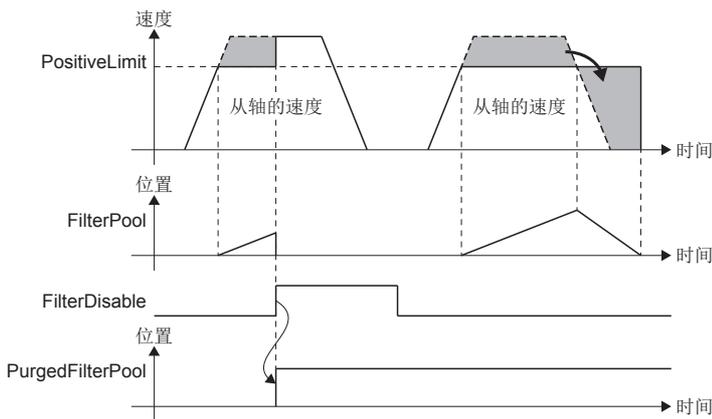
设置值	内容
FALSE	按照滤波器设置执行动作。
TRUE	滤波器将变为无效，将主轴(Master)的输入原样不变地传送到从轴(Slave)。

- 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	设置值
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。

设置值	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。

设置值	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(出错代码:1A86H)”。

■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。

设置值	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。

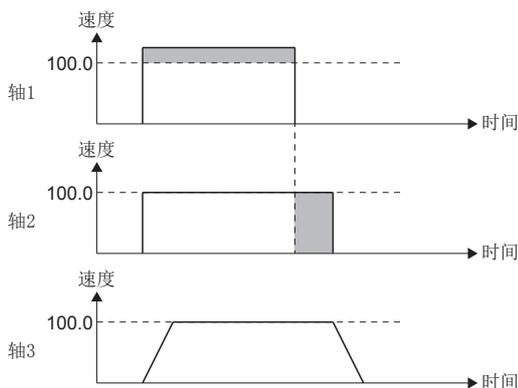
注意事项

- 指令滤波器的动作中进行滤波器无效(FilterDisable)的切换时,从轴(Slave)的速度有可能急剧变化,因此应加以注意。
- 将指令滤波器置为了有效后,从轴(Slave)中发生停止原因导致未将主轴(Master)的指令传送到从轴(Slave)及后段的同步控制功能块的情况下,同步位置关系将被破坏。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		设置值
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
背隙量		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

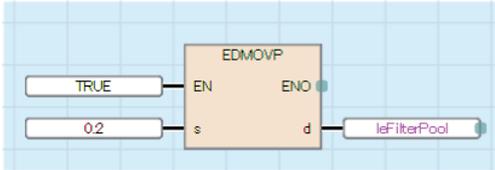
标签名	数据类型	注释
MCv_SpeedLimitFilter_1	MCv_SpeedLimitFilter	速度限制滤波器FB
bSpeedLimitFilter	位	速度限制滤波器指令
lePositiveLimit	双精度实数	正方向限制值
leNegativeLimit	双精度实数	负方向限制值
bSpeedLimitValid	位	输出值有效
bSpeedLimitBusy	位	执行中
bSpeedLimitActive	位	控制中
bSpeedLimitCommandAborted	位	执行中断
bSpeedLimitError	位	出错
uwSpeedLimitErrorID	字[无符号]/位串[16位]	出错代码
leFilterPool	位	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	出错
uwSmoothingErrorID	字[无符号]/位串[16位]	出错代码
leSmoothingValidFrequency	双精度实数	有效频率
leSmoothingFilterPool	双精度实数	滤波器累计值

■FBD/LD的程序(CPU模块侧)

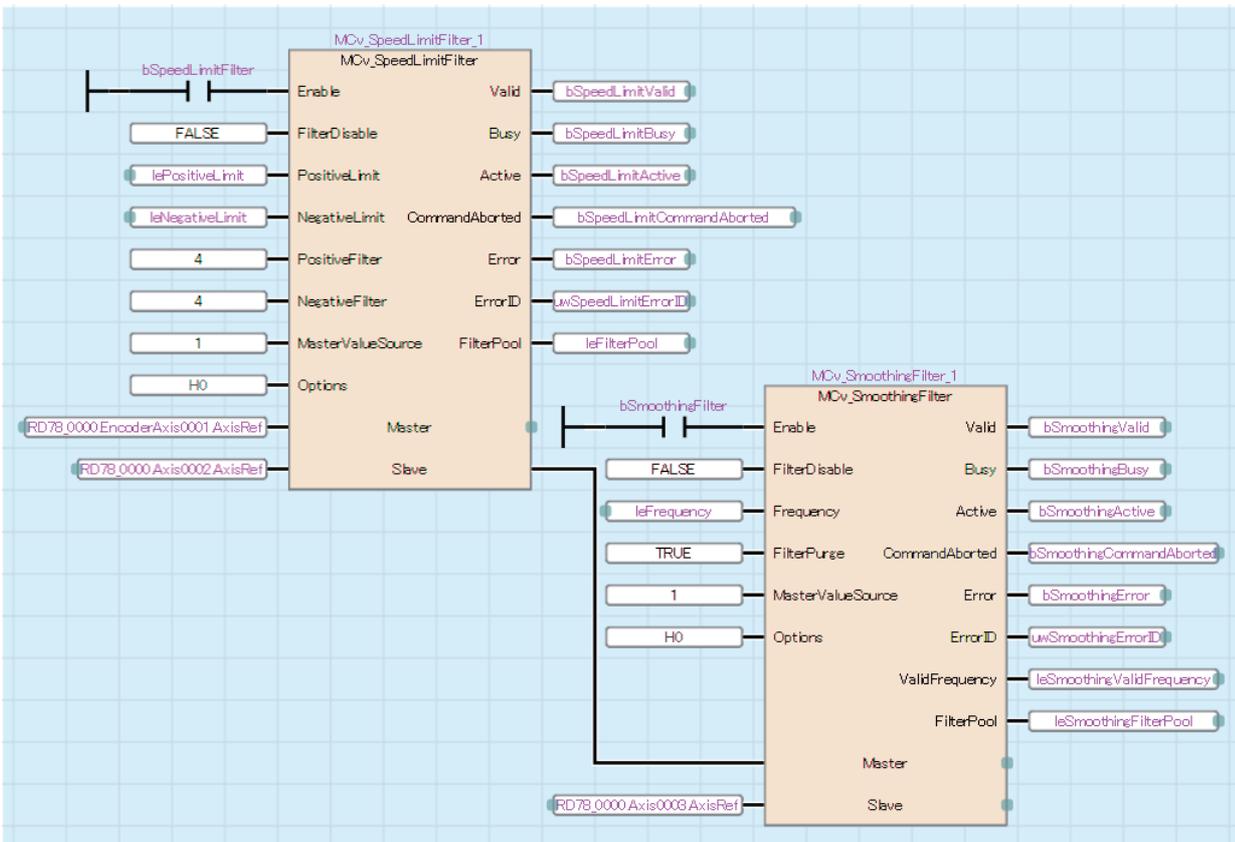
- 正方向/负方向限制值



- 频率



- 速度限制滤波器/平滑滤波器



■ST的程序(运动模块侧)

```
//-----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
);
```

3.3 常规FB

MCv_ReadProfileData

从展开区域或文件中读取指定的运算配置文件数据(凸轮数据)。

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>		
名称	输入区域点数(byte)	输出区域点数(byte)	编译方式	FB动作
配置文件读取	422	10	子程序型	随时执行型

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。 ☞ 63页 PROFILE_DATA 对于定义了读取用数据结构的运算配置文件格式，可以进行读取操作。 设置了不支持读取操作的运算配置文件的情况下，将变为“运算配置文件读取/写入不支持(出错代码: 345EH)”。
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。 ☞ 376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。 ☞ 376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(出错代码: 3413H)”，且不进行读取。 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(出错代码: 3465H)”。

输入变量	名称	数据类型	获取	有效范围	初始值	内容
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(出错代码: 3465H)”.)
Target	读取目标	WORD (UINT)	启动时	0、1	0	设置运算配置文件的读取目标。 • 0: 展开区域 • 1: 文件 关于详细内容，请参阅下述章节。 ☞ 377页 读取目标(Target)

■输出变量

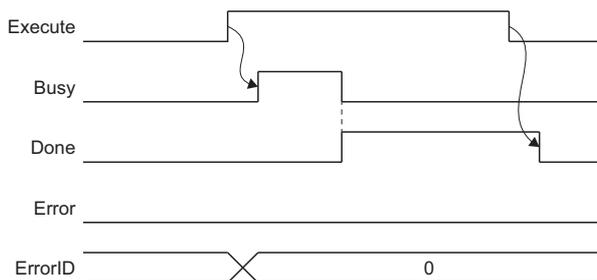
输出变量	名称	数据类型	初始值	内容
Done	完成	BOOL	FALSE	控制完成时，将变为TRUE。 动作完成时根据启动(Execute)的状态将变为如下所示。 ■启动(Execute)为TRUE的情况下 在将启动(Execute)置为FALSE之前将保持为TRUE不变。 ■启动(Execute)为FALSE的情况下 仅1周期为TRUE。
Busy	执行中	BOOL	FALSE	执行了MCv_ReadProfileData(配置文件读取)时，将变为TRUE。
Error	出错	BOOL	FALSE	发生了异常时，将变为TRUE。
ErrorID	出错代码	WORD (UINT)	0	发生了异常时，将返回出错代码。 关于出错代码的详细内容，请参阅下述手册。 ☞ MELSEC iQ-R运动模块用户手册(应用篇)

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

- 正常完成时的情况下



- 异常完成时的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 11页 通过执行指令(Execute)类型的运动控制FB的基本动作

■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。

☞ 61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	☞ 65页 PROFILE_CAM_DATA
旋转刀具用凸轮数据	PROFILE_ROTARY_CUTTER	☞ 67页 PROFILE_ROTARY_CUTTER

例

将凸轮数据格式的运算配置文件以缓冲存储器(G11480000~)10点读取的情况下

Data1.Target := "[DEV](PROFILE_CAM_DATA)G11480000"

• 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)，设置的数据类型的结构体有所不同。数组的元素数与分辨率、坐标数不一致的情况下，将变为“偏置·读取/写入数据数不正确(出错代码：3465H)”。

○：对应，×：不对应

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	0: 展开区域	LREAL[]*1	○	○*3	○*3	—
	1: 文件	LREAL[]*1	○	×	×	—
		PROFILE_CAM_ELEMENT[]*2	×	○	○*4	☞ 66页 PROFILE_CAM_ELEMENT
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]，坐标值[x, y]

元素数(n - m + 1)仅需要读取数据数(Points)的参数的数量。

对整个运算配置文件进行读取的情况下(Points = 0)，应设置为与PROFILE_CAM_DATA结构体的分辨率(Resolution)相同的数。

设置时，应确保维数(p - o + 1)为“2”。

为了将LREAL型2点的数据以元素数进行读取，应准备相同大小的软元件或标签区域。

*2 PROFILE_CAM_ELEMENT的数组将按下述方式设置。

PROFILE_CAM_ELEMENT[m..n]

元素数(n - m + 1)仅需要FB的读取数据数(Points)的参数的数量。

对整个运算配置文件进行读取的情况下(Points = 0)，应设置为与PROFILE_CAM_DATA结构体的区间数(NumberOfSections)相同的数。

*3 通过直线插补对分辨率 + 1点的数据(1周期当前值与行程的成对)进行读取。此时，第1点的数据的X为开始点，Y为初始行程量中设置的值。

*4 仅使用结束点、行程。

例

将凸轮数据格式的运算配置文件以缓冲存储器(G11483000~)10点读取的情况下

Data2.Target := "[DEV](LREAL[1..10, 1..2])G11483000"

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(出错代码：3462H)”。

- 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。

☞ MELSEC iQ-R运动模块用户手册(应用篇)

■ 读取目标 (Target)

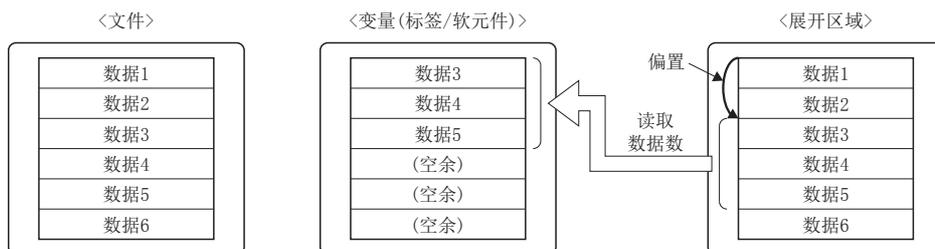
设置运算配置文件的读取目标，对读取数据1(Data1)/读取数据2(Data2)进行读取。

设置值	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。 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(出错代码: 340FH)”。
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。 运算配置文件不存在的情况下，将变为“无运算配置文件(出错代码: 340FH)”。

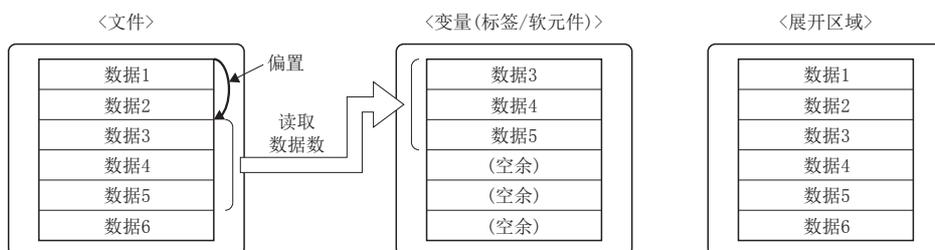
例

设置了下述值的情况下

- 在读取目标(Target)中设置“0: 展开区域”，在偏置(Offset)中设置“2”，在读取数据数(Points)中设置“3”



- 在读取目标(Target)中设置“1: 文件”，在偏置(Offset)中设置“2”，在读取数据数(Points)中设置“3”



要点

- 读取目标(Target)为“1: 文件”的情况下，将以优先度低于运算周期的周期进行处理。因此，根据处理内容及运算配置文件，读取有可能需要耗费一定时间。
- 读取目标(Target)为“0: 展开区域”的情况下，将按照执行的程序的周期进行读取。

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	pulse
	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	出错1
uwErrorID1	字[无符号]/位串[16位]	出错代码1
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	出错2
uwErrorID2	字[无符号]/位串[16位]	出错代码2
ProfileData0002*1	PROFILE_DATA	配置文件数据0002

*1 仅使用FBD/LD的程序

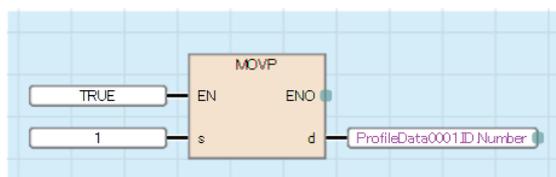
■使用的全局标签

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

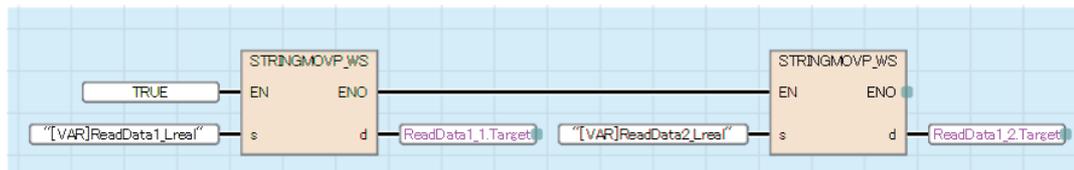
*1 仅使用FBD/LD的程序

■FBD/LD的程序(CPU模块侧)

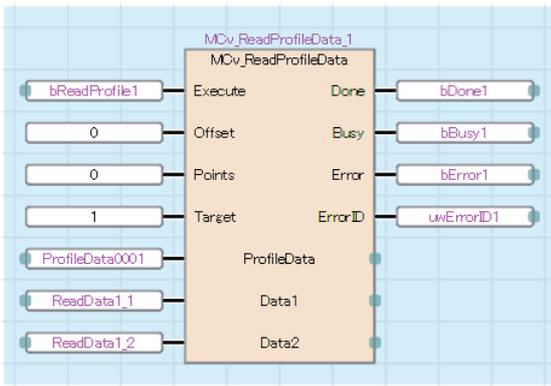
- 配置文件数据0001



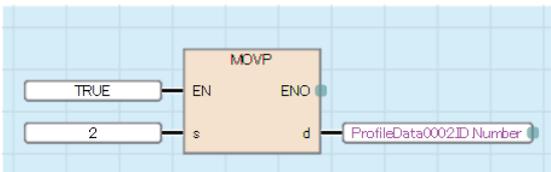
- 读取数据1



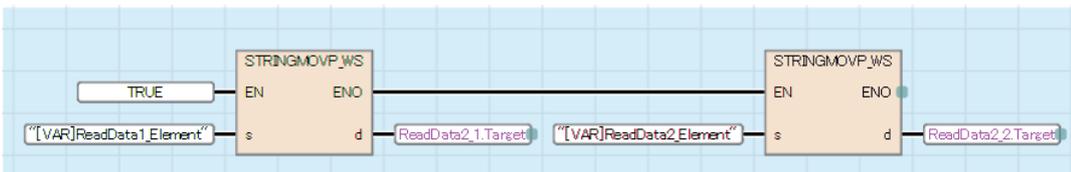
- 配置文件读取1(以LREAL型(2维数组)的格式读取凸轮数据)



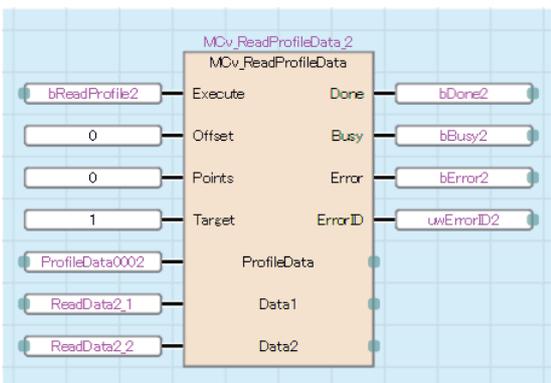
- 配置文件数据0002



- 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(运动模块侧)

```
//-----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:= ProfileData0002.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
);
```

MCv_WriteProfileData

将指定的运算配置文件数据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>

名称	输入区域点数 (byte)	输出区域点数 (byte)	编译方式	FB动作
配置文件写入	424	16	子程序型	随时执行型

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。 63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。 设置了不支持写入操作的运算配置文件的情况下，将变为“运算配置文件读取/写入不支持(出错代码：345EH)”。
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。 384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。 384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(出错代码：3413H)”，且不进行写入。 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(出错代码：3465H)”。)

输入变量	名称	数据类型	获取	有效范围	初始值	内容
Points	写入数据数	DWORD (UDINT)	启动时	0~4294967295	0	<p>设置要写入的数据点数。</p> <p>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。</p> <p>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 (出错代码: 3465H)”.)</p>
Target	写入目标	WORD (UINT)	启动时	0、1	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 <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 (出错代码: 3465H)”.)</p> <p>关于详细内容，请参阅下述章节。  385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。  386页 启动模式 (ExecutionMode)</p>

■输出变量

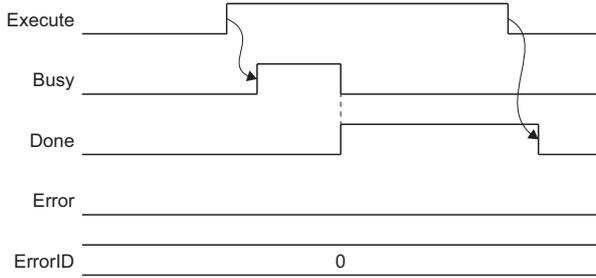
输出变量	名称	数据类型	初始值	内容
Done	完成	BOOL	FALSE	<p>控制完成时，将变为TRUE。</p> <p>动作完成时根据启动 (Execute) 的状态将变为如下所示。</p> <p>■启动 (Execute) 为TRUE的情况下 在将启动 (Execute) 置为FALSE之前将保持为TRUE不变。</p> <p>■启动 (Execute) 为FALSE的情况下 仅1周期为TRUE。</p>
Busy	执行中	BOOL	FALSE	<p>执行了MCv_WriteProfileData (配置文件写入) 时，将变为TRUE。</p>
Error	出错	BOOL	FALSE	<p>发生了异常时，将变为TRUE。</p>
ErrorID	出错代码	WORD (UINT)	0	<p>发生了异常时，将返回出错代码。</p> <p>关于出错代码的详细内容，请参阅下述手册。  MELSEC iQ-R运动模块用户手册 (应用篇)</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) 中设置的数据中写入在写入数据数 (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”。

■时序图

- 正常完成时的情况下



- 异常完成时的情况下

关于异常完成时的时序图，请参阅下述章节。

☞ 11页 通过执行指令(Execute)类型的运动控制FB的基本动作

■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。

☞ 61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	☞ 65页 PROFILE_CAM_DATA
旋转刀具用凸轮数据	PROFILE_ROTARY_CUTTER	☞ 67页 PROFILE_ROTARY_CUTTER

例

将凸轮数据格式的运算配置文件以缓冲存储器(G11480000~)10点写入的情况下

Data1.Target := "[DEV](PROFILE_CAM_DATA)G11480000"

- 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。

○：对应， ×：不对应

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	☞ 66页 PROFILE_CAM_ELEMENT
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], 坐标值[x, y]

元素数(n - m + 1)仅需要写入数据数(Points)的参数的数量。

对整个运算配置文件进行写入的情况下(Points = 0), 应设置为与PROFILE_CAM_DATA结构体的分辨率(Resolution)相同的数。

设置时, 应确保维数(p - o + 1)为“2”。

为了将LREAL型2点的数据以元素数进行写入, 应准备相同大小的软元件或标签区域。

*2 PROFILE_CAM_ELEMENT的数组将按下述方式设置。

PROFILE_CAM_ELEMENT[m..n]

元素数(n - m + 1)仅需要FB的写入数据数(Points)的参数的数量。

对整个运算配置文件进行写入的情况下(Points = 0), 应设置为与PROFILE_CAM_DATA结构体的区间数(NumberOfSections)相同的数。

*3 仅使用结束点、行程。

例

将凸轮数据格式的运算配置文件以缓冲存储器(G11483000~)10点写入的情况下

Data2.Target := "[DEV] (LREAL[1..10, 1..2])G11483000"

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(出错代码：3462H)”。
- 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。
 MELSEC iQ-R运动模块用户手册(应用篇)

■写入目标(Target)

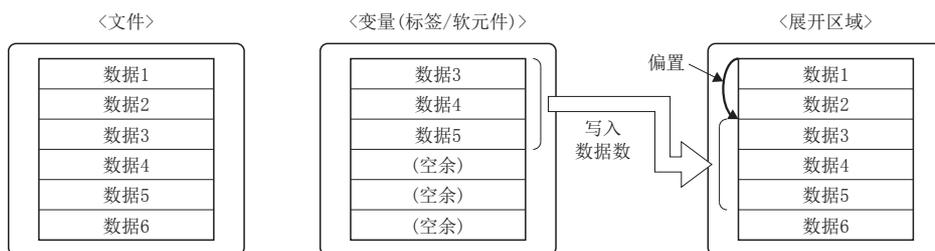
设置运算配置文件的写入目标，对写入数据1(Data1)/写入数据2(Data2)及运算配置文件展开FB的配置文件ID(PROFILE_ID)进行写入。

设置值	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。 ■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(出错代码：3452H)”，且不进行展开处理。 ■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。 • 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(出错代码：3465H)”。(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(出错代码：3465H)”。

例

设置了下述值的情况下

- 在写入目标(Target)中设置“0: 展开区域”，在偏置(Offset)中设置“2”，在写入数据数(Points)中设置“3”



要点

- 写入目标(Target)为“1: 文件”的情况下，将以优先度低于运算周期的周期进行处理。因此，根据处理内容及运算配置文件，写入有可能需要耗费一定时间。
- 写入目标(Target)为“0: 展开区域”的情况下，将按照执行的程序的周期进行写入。

■启动模式(ExecutionMode)

设置执行MCv_WriteProfileData(配置文件写入)的时机。

设置值	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(出错代码:3411H)”。
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(出错代码:3411H)”,且不更改展开区域。

要点

没有执行中的FB的情况下,与启动模式(ExecutionMode)的设置无关将立即执行。

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。
 - 运动系统软件的版本为“11”及以前的情况下,使用MCv_WriteProfileData(配置文件写入)将运算配置文件数据写入到展开区域、文件区域中时,重复动作(Periodic)、主轴绝对坐标(MasterAbsolute)、从轴绝对坐标(SlaveAbsolute)的设置值将变为FALSE。(运动系统软件的版本为“12”及以后的情况下,由于支持设置值的更改,因此无需进行下述设置。)进行设置值的更改时,应按下述步骤实施。
1. 将运算配置文件数据写入到文件区域中。
 2. 将步骤1.中写入的文件通过MC_CamTableSelect(凸轮表选择)进行指定,并设置重复动作(Periodic)、主轴绝对坐标(MasterAbsolute)、从轴绝对坐标(SlaveAbsolute)。

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	pulse
	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	出错1
uwErrorID1	字[无符号]/位串[16位]	出错代码1
ProfileID1	PROFILE_ID	配置文件ID1
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	出错2
uwErrorID2	字[无符号]/位串[16位]	出错代码2
ProfileID2	PROFILE_ID	配置文件ID2
ProfileData0002*1	PROFILE_DATA	配置文件数据0002

*1 仅使用FBD/LD的程序

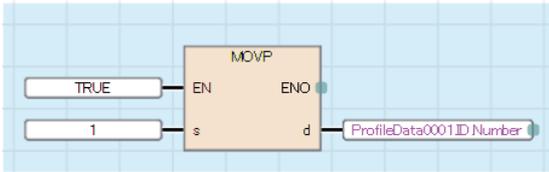
■使用的全局标签

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

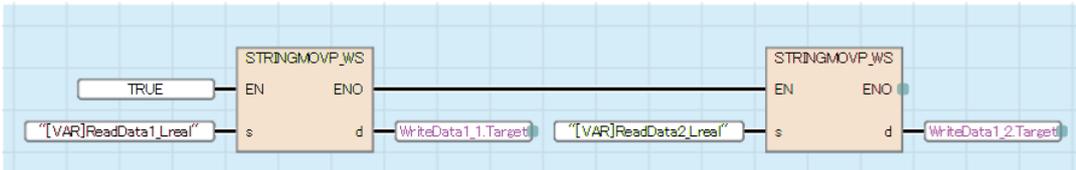
*1 仅使用FBD/LD的程序

■FBD/LD的程序(CPU模块侧)

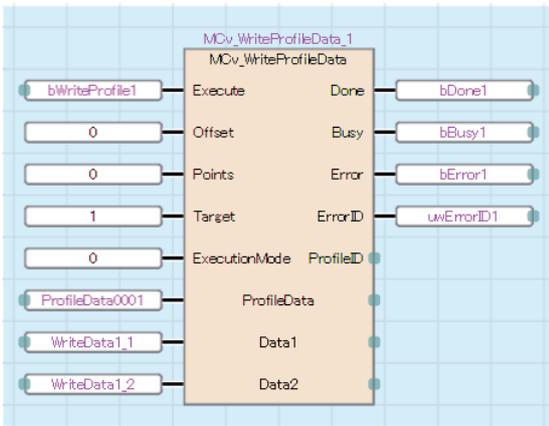
- 配置文件数据0001



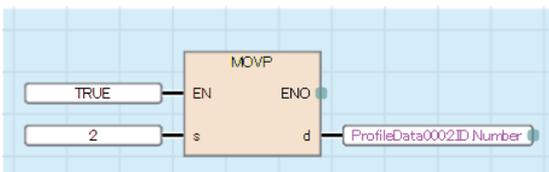
- 写入数据1



- 配置文件写入1(以LREAL型(2维数组)的格式写入凸轮数据)



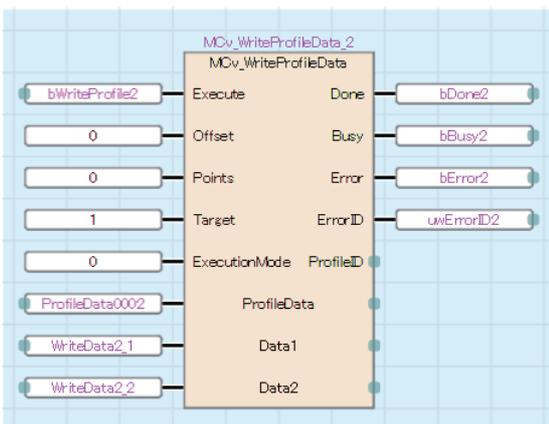
- 配置文件数据0002



- 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(运动模块侧)

```
//-----写入数据1-----
WriteData2_1.Target := "[VAR]WriteData1_Lreal";
WriteData2_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:= WriteData2_1 ,
    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-----
WriteData1_1.Target := "[VAR]WriteData1_Element";
WriteData1_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:= ProfileData0001.ProfileData ,
    Data1:= WriteData1_2 ,
    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
);
```

4 运动模块的程序

以下对运动模块中使用的程序有关内容进行说明。

4.1 运动模块的程序

以下对创建程序时所需的程序的配置及内容进行说明。

支持语言

作为控制运动系统的方法有以下方法。

- 通过CPU模块进行控制的方法
- 通过运动系统内置的程序语言进行控制的方法

作为运动系统内置的程序语言可以使用以下内容。

- 结构化文本(ST语言)

变量

程序中使用的变量有“软元件”与“标签”。

与轴的控制相关的参数、监视数据、控制指令等被分配到由系统提供的标签(结构体)。

类型	内容
软元件	是以名称+编号表示的变量。 名称、类型、使用方法在系统中定义。 ■运动系统中可使用的软元件 <ul style="list-style-type: none">• 缓冲存储器(G)• 链接特殊继电器(SB)• 链接特殊寄存器(SW)
标签	是以任意字符串表示的变量。 有名称、类型、使用方法在系统中定义的标签与由用户定义的标签。

注意事项

■关于工程数据的获取

电源ON中不进行标签定义与初始值、程序等的重新获取。反映已写入的工程的情况下，应进行电源OFF→ON。

■运动控制FB实例的调用

如果从多个程序中调用1个运动控制FB实例，可能会导致意外动作。

■关于字符串型标签的背离

对于字符串型标签进行写入的情况下，如果同时进行执行周期不同的任务及从外部设备进行的读取/写入，则可能会发生背离且可能会导致意外结果。应对其进行修改以确保访问的周期相同，或在访问时设置互锁。

■关于数组的元素

对于定义的数组的元素数，请勿访问超出元素编号的范围的元素。

用常数指定了数组中定义的范围以外的下标的情况下，进行程序的转换时将出错。但是，用常数以外指定了数组的下标的情况下，在程序的转换中不会出错，但在执行时会出错或访问其它软元件进行处理。

■轴类型/轴组类型的局部标签

如果定义轴类型、轴组类型的局部标签，则可能导致意外动作。

■关于程序的容量上限

程序的容量上限因插件Program_ST参数(Addon_Program_ST)中设置的系统存储器(RAM)的容量设置而异。系统存储器不足的情况下，发生“存储器容量不足(出错代码: 350DH)”，且程序不执行RUN。在此情况下，应重新设置系统存储器(RAM)的设置值。

要点

STOP→RUN时将程序文件的内容展开到内部存储器中。此时，作为大致标准需要有程序文件容量的4~7倍(根据程序的内容变动)的系统存储器(RAM)的空余。

程序的执行

程序在“可编程控制器就绪(Y0)”变为ON时变为RUN状态，并根据程序的执行类型及执行步骤的设置执行程序。

程序的执行类型

对各执行类型设置程序的处理在哪个时机执行。

可设置的执行类型如下所示。

执行类型	参阅
初始	☞ 391页 初始执行类型
正常	☞ 391页 正常执行类型
恒定周期	☞ 392页 恒定周期执行类型
待机	☞ 393页 待机执行类型

■初始执行类型

“可编程控制器就绪(Y0)”OFF→ON时仅执行1次程序。

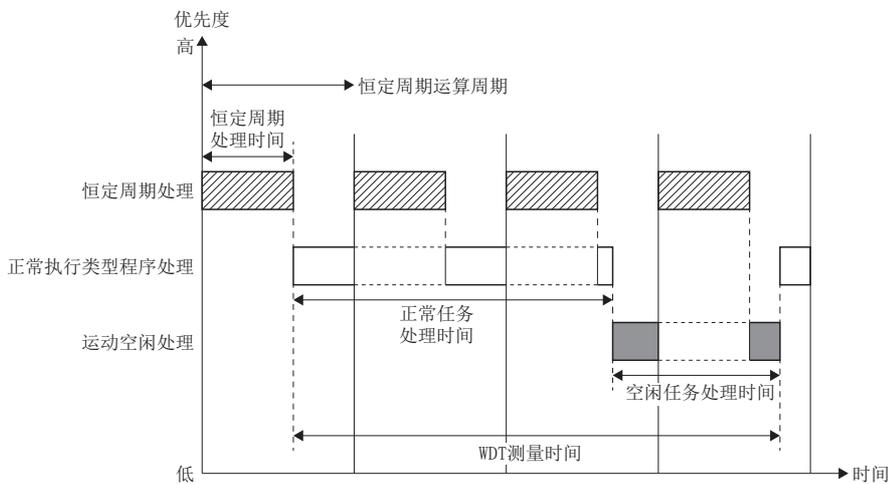
初始执行类型完成后，执行其它执行类型程序。

限制事项

对于初始执行类型程序，请勿在执行完成之前使用需要多个执行周期的指令(完成软元件存在的指令)及运动控制FB。

■正常执行类型

运动系统的正常任务中，执行程序。



限制事项

- 恒定周期执行类型程序的执行中，中断正常执行类型程序的执行。但是，在正常执行类型程序中，可以使用恒定周期执行类型程序的禁止指令(DI)。在从执行恒定周期执行类型程序的禁止指令(DI)开始到执行恒定周期执行类型程序的允许指令(EI)为止的期间，不执行恒定周期执行类型程序。可以通过“EI标志(System. Md. Program_Ei)”确认恒定周期执行类型程序允许/禁止的状态。
- 正常执行类型程序的执行间隔超过1.0[s]的情况下，将发生WDT出错。

■恒定周期执行类型

各指定时间执行程序。对于时间，可以在第1运算周期~60000[ms] (第1运算周期的整数倍) 的范围内进行设置。时间短的程序将优先进行处理。运算周期处理与恒定周期执行类型程序的处理顺序如下所示。

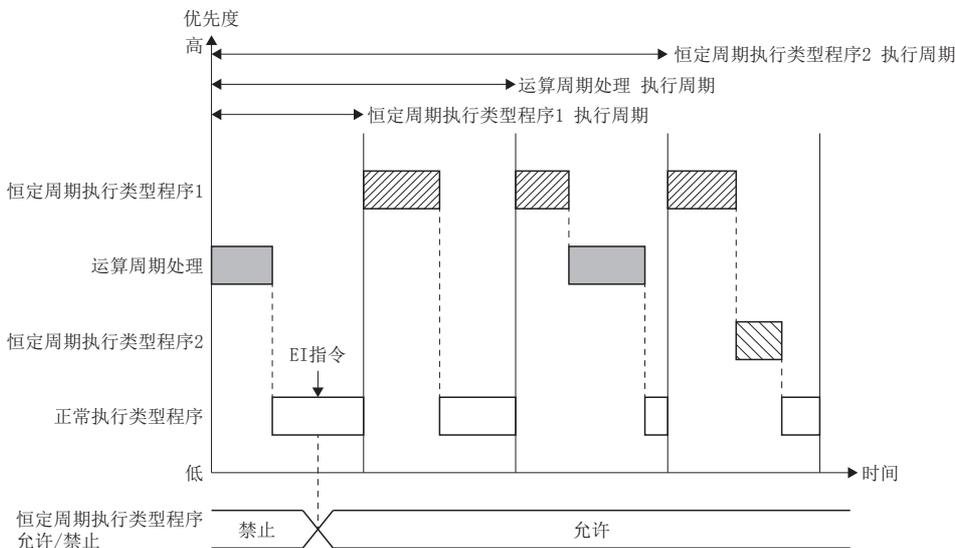
恒定周期间隔	处理顺序
恒定周期间隔 = 运算周期	等待程序执行完成后进行运算周期处理。 在运算周期处理中达到了恒定周期时机的情况下，程序执行与运算周期处理将同时(并行)执行。
恒定周期间隔 > 运算周期	程序执行与运算周期处理同时(并行)执行。

可设置的恒定周期间隔最多为16种类型。对于恒定周期间隔的类型的增加与间隔的长短无关，会对运算周期有影响。因此，运算周期短的情况下(62.5μs及以下)，即使恒定周期间隔小于16种类型也可能发生运算周期溢出。同一间隔的程序设置数没有限制。

要点

执行恒定周期执行类型的程序时，需要通过EI指令将其设置为执行允许状态。

从发出EI指令的下一个恒定周期的起始开始执行恒定周期程序。



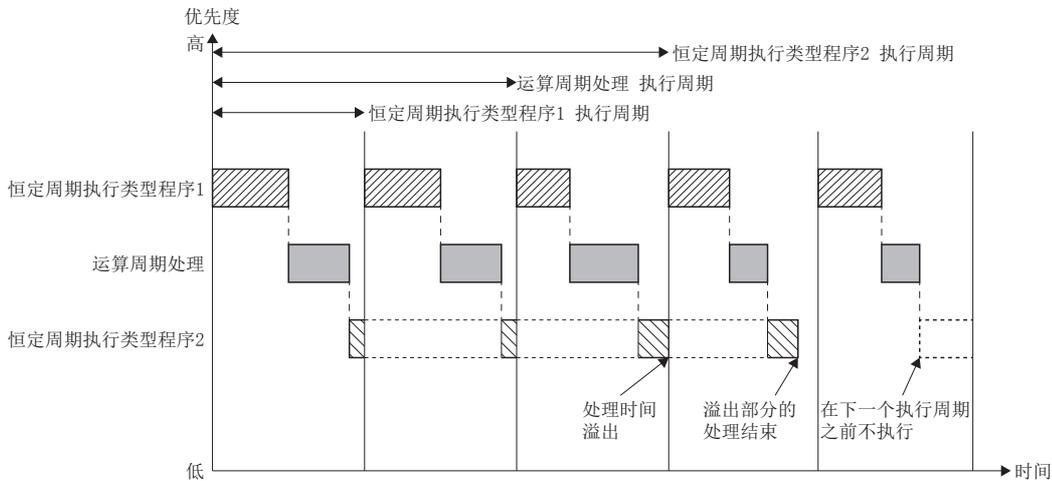
限制事项

在正常执行类型程序中设置为恒定周期执行类型程序的禁止(DI)的情况下，将无法执行恒定周期类型的程序。

• 关于处理的中断

程序执行处理可能会被其它优先级高的处理中断。

恒定周期执行类型程序的执行中，由于中断导致处理被中断的情况下，在中断处理完成后将重新开始处理。另外，处理时间超过了恒定周期间隔的情况下，在处理完成之前即使多次达到执行周期也将被忽略。从处理完成后的下一个执行周期再次开始执行。



限制事项

即使超过恒定周期间隔长于运算周期的程序的处理时间，也不会发生出错或报警。应根据需要使用工程工具监视PROGRAM_INFO结构体的执行时间(ExecutionTime)。

要点

由于恒定周期执行类型程序的运算负载，可能会导致运算周期溢出。

■ 待机执行类型

有执行请求时执行程序。

从其它执行中程序通过PSCAN指令更改为正常执行类型程序后执行。

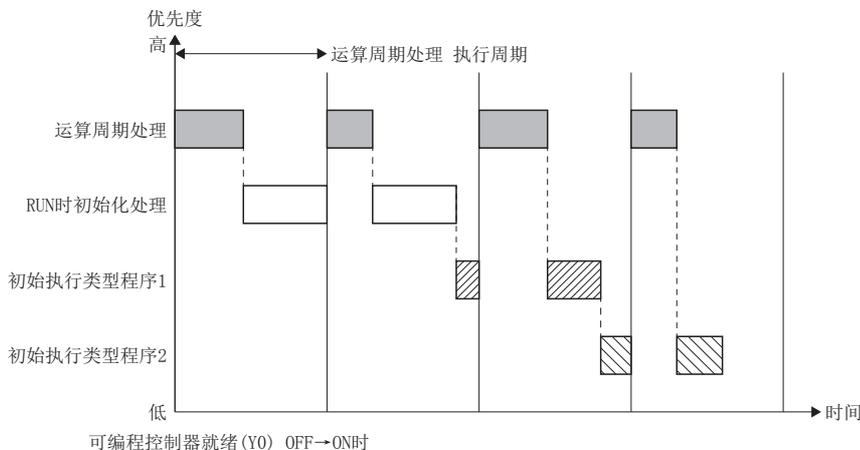
程序执行的流程

各程序执行的流程如下所示。

■ “可编程控制器就绪(Y0)” ON时的初始执行类型的程序处理

“可编程控制器就绪(Y0)” ON时执行的处理的时序图如下所示。

初始执行类型程序的处理被运算周期处理中断。



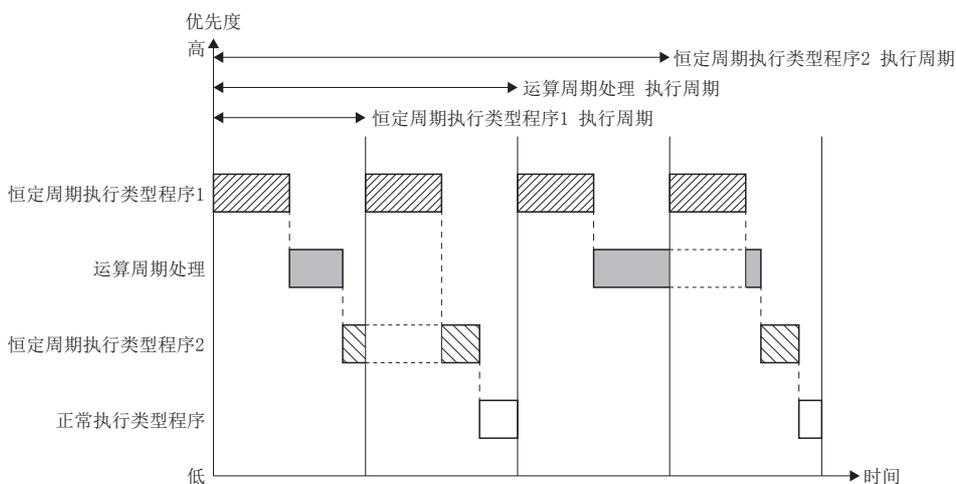
要点

即使在初始执行类型程序中发出EI指令，也不执行在初始执行类型程序处理完成之前的恒定周期类型程序。

■初始执行类型的程序完成时的处理

初始执行类型程序的处理全部完成后的各程序动作的时序图如下所示。

对于3个恒定周期处理(恒定周期执行类型程序1、恒定周期执行类型程序2、运算周期处理)，将按执行周期短的顺序优先进行处理。



要点

同一执行类型・执行周期设置的情况下，从执行顺序的数字小的开始按顺序执行。
可以通过工程工具的“程序执行设置”确认执行顺序。
根据执行类型・执行周期设置的思路优先于执行顺序。

■电源ON时初始化处理

在电源OFF→ON时的标签初始化处理中，进行以下处理。

- 全局标签文件、结构体定义文件、FB定义文件的展开
- 全局标签的构筑、初始值设置
- 程序设置的获取

■首次RUN时初始化处理

电源OFF→ON后，在首次RUN时初始化处理中，进行以下处理。

- 程序文件的展开、程序局部标签的构筑
- 各参数及程序的检查
- 标签/FB的初始值设置

要点

根据参数设置、程序个数、文件数，检查需要花费一定时间，因此对外围设备的响应可能会延迟。

■第2次及以后的RUN时初始化处理

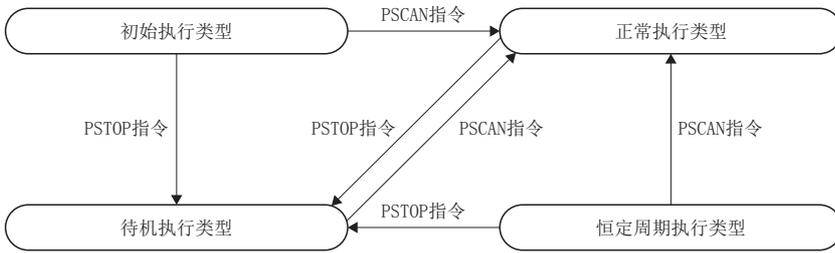
电源OFF→ON后，在第2次及以后的RUN时初始化处理中，不进行各文件的重新获取及标签的重新构筑。只进行初始值的设置处理。

关于要设置的初始值，请参阅标签的初始化功能。(P.396页 标签的初始化功能)

执行类型的更改

“可编程控制器就绪(Y0)” ON中更改程序的执行类型时，将通过以下程序控制用指令进行更改。

- 程序扫描执行登录 (PSCAN)
- 程序待机 (PSTOP)



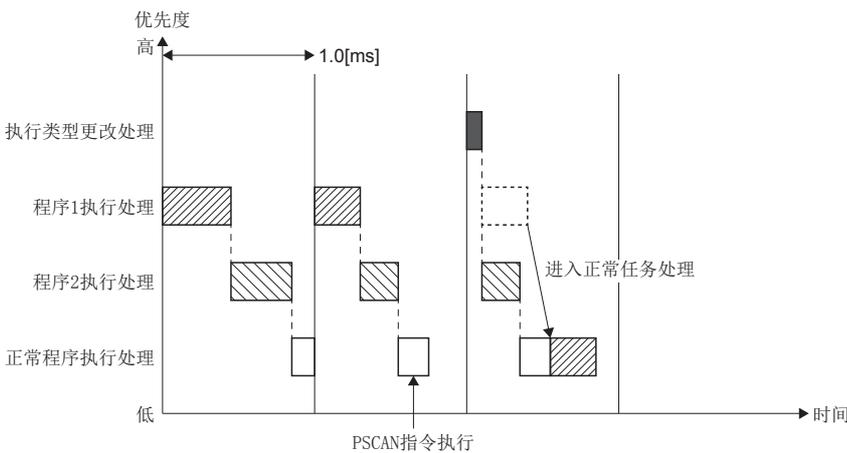
■ 执行类型的更改时机

更改执行类型的时机如下所示。

执行类型	执行指令	
	PSCAN	PSTOP
初始	将变为正常执行类型。	程序的END处理完成后，将变为待机类型。
正常	无处理	
恒定周期	程序的END处理完成后，将变为正常执行类型。	
待机	将变为正常执行类型。	无处理

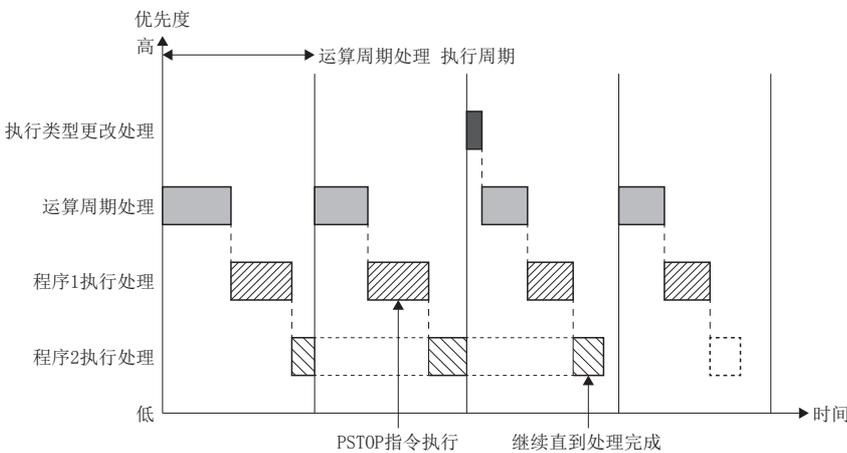
例

恒定周期执行类型1与恒定周期执行类型2中，对程序1执行PSCAN指令的情况下。



例

正常执行类型1与正常执行类型2中，对程序2执行PSTOP指令。



■注意事项

- 如果将初始执行类型程序或恒定周期执行类型程序更改为其它执行类型，则无法返回到原来的执行类型。
- 对同一程序执行了多个指令的情况下，将按照最后执行的指令动作。
- 在更改对象的程序的处理完成之前执行类型的更改不被反映。

发生出错时的动作

执行指令时发生了停止型出错的情况下，“准备就绪(X0)”变为ON→OFF，且程序执行停止。

对于下述出错，可以在工程工具的“异常时动作设置^{*1}”中，设置发生出错时的动作(停止/继续运行)。

- 软元件·标签·缓冲存储器指定不正确
- 文件名指定异常
- 运算异常

*1 “异常时动作设置”是在运动控制设置功能的基本设置中设置。(P 403页 基本设置)

标签的初始化功能

在运动系统中，电源OFF→ON时或STOP→RUN时进行标签的初始化处理。

初始化处理的执行内容如下所示。

标签		初始值设置 有/无	标签初始化时的动作		
			电源OFF→ON	STOP→RUN时	
				首次	第2次及以后
运动标签	参数	—	设置初始值	无处理(维持值) ^{*1}	
	参数以外	—	设置初始值	无处理(维持值)	
用户定义全局标签		有	设置初始值	无处理(维持值) ^{*1}	
		无	设置初始值(=0)	无处理(维持值) ^{*1}	
用户定义局部标签		有	—(标签未构筑)	设置初始值	无处理(维持值) ^{*1}
		无	—(标签未构筑)	设置初始值(=0)	无处理(维持值)

*1 根据设置第2次及以后的STOP→RUN时，可进行初始值的设置。

项目	内容	设置范围	初始值
STOP→RUN时(第2次及以后)的标签初始值反映	设置STOP→RUN时(第2次及以后)是否设置标签初始值。(首次必须反映)	• 有效 • 无效	无效

标签存储器

将分配标签实例的存储器区域称为标签存储器。标签存储器的最大容量通过标签存储器容量

(System.PrConst.LabelMemorySize)进行设置。

当前的标签存储器的空余容量可以通过标签存储器空余容量(System.Md.LabelMemoryFreeSize)进行确认。

4.2 运动模块的程序创建

运动模块通过下述方法创建程序。

- 在运动模块侧使用ST语言创建ST程序。(☞ 397页 使用ST程序的程序的创建步骤)
- 将运动模块中登录的标签在CPU模块侧使用梯形图语言、FBD/LD语言、ST语言创建顺控程序。(☞ 421页 使用公开标签功能的程序的创建步骤)

使用ST程序的程序的创建步骤

以下对使用工程工具进行运动系统中所需的设置及创建ST程序的步骤进行说明。

1. 参数的设置

设置模块配置图、CPU参数。(☞ 398页 参数的设置)

2. 网络的设置

在“模块参数(网络)”中，设置网络相关的参数。

在“网络配置设置”中设置通过CC-LinkIE TSN连接的伺服放大器。

(☞ 399页 网络的设置)

3. 模块参数的设置

在“模块参数(运动)”中，进行与RD78G的动作相关的设置。(☞ 402页 模块参数的设置)

4. 模块扩展参数的设置

在“运动控制设置功能”中进行与RD78G的轴控制相关的设置。

- 基本设置(☞ 403页 基本设置)
- 系统设置(☞ 404页 系统设置)
- 轴(☞ 407页 轴)
- 轴组(☞ 417页 轴组)

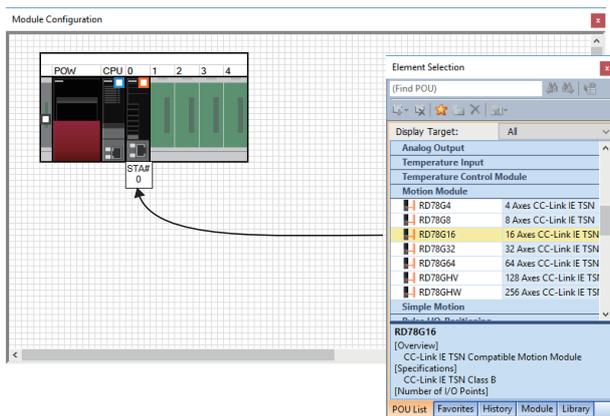
5. ST程序的创建(☞ 419页 ST程序的创建)

在“运动控制设置功能”中使用ST语言，创建使RD78G动作的程序。

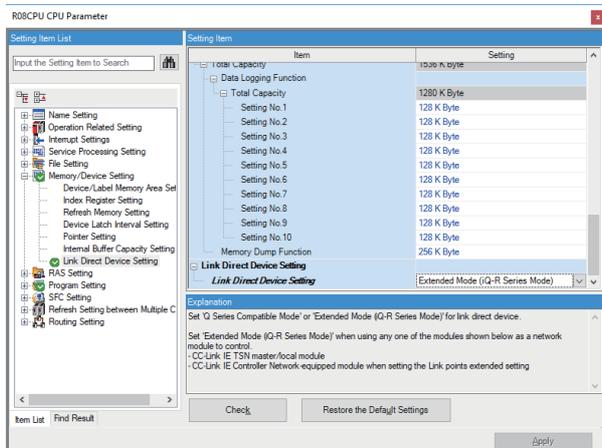
- 运动控制FB的插入(☞ 420页 运动控制FB的插入)
- 标签的登录(☞ 420页 标签的登录)

参数的设置

设置模块配置图、CPU参数。



1. 从[导航窗口]⇒[模块配置图]打开模块配置图，从部件选择窗口添加要使用的RD78G。
选择菜单的[编辑]⇒[参数]⇒[确定]，确定模块配置图。

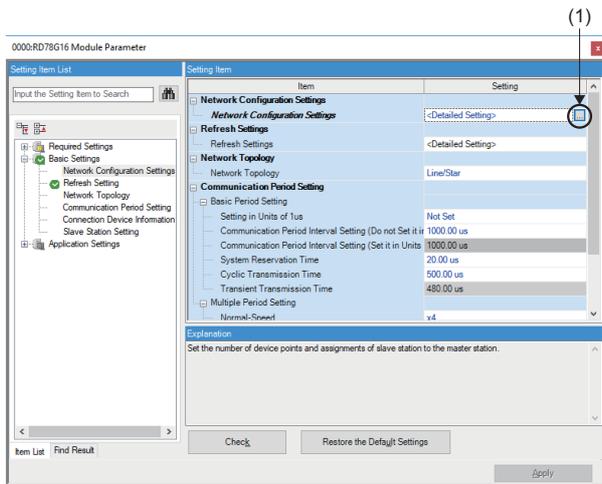


2. 选择[导航窗口]⇒[参数]⇒[模块型号*1]⇒[CPU参数]，显示参数编辑器。
在“存储器/软件设置”⇒“链接直接软件设置”的下拉式列表中选择“扩展模式(iQ-R系列模式)”，单击[应用]按钮。

*1: 显示设置的CPU模块的型号。

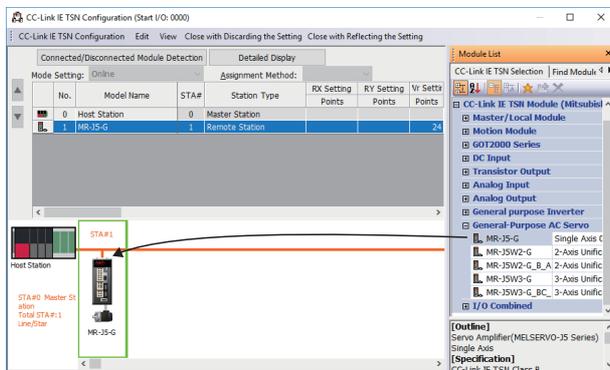
网络的设置

设置模块参数(网络)。

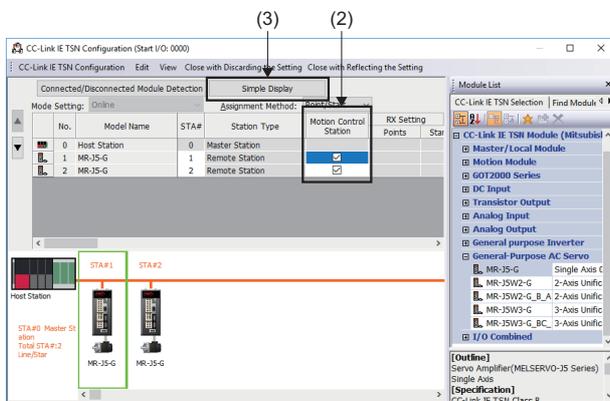


1. 选择[导航窗口]⇒[参数]⇒[模块信息]⇒[模块型号*1]⇒[模块参数(网络)], 显示参数编辑器。
选择“基本设置”⇒“网络配置设置”⇒“详细设置”, 单击[...]按钮(1)。

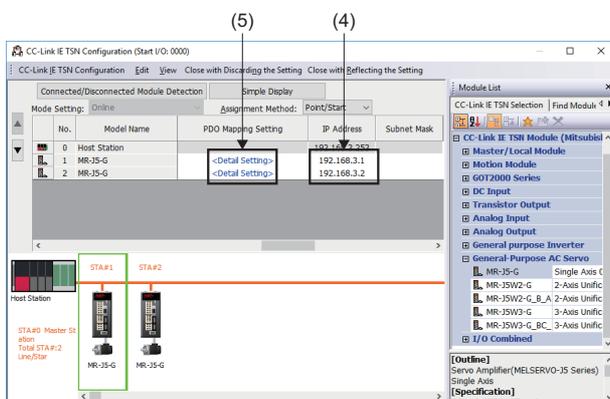
*1: 显示设置的运动模块的型号。



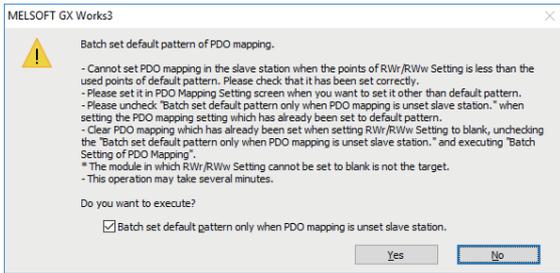
2. 显示“CC-Link IE TSN配置”画面。
从[模块一览]⇒[CC-Link IE TSN设备(三菱电机)]⇒“通用AC伺服”选择伺服放大器, 将其拖放到要放置的位置。



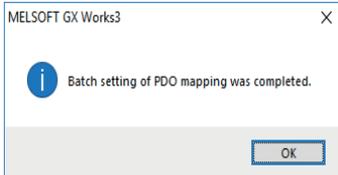
3. 对每个设置的伺服放大器勾选运动管理站(2)的复选框。
*: 未显示运动管理站的情况下, 单击[详细显示]按钮(3), 切换到详细显示。



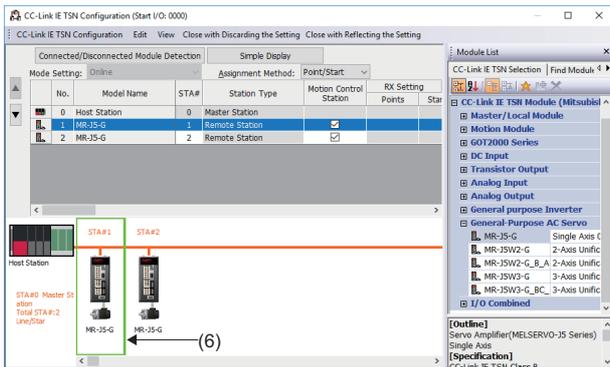
4. 确认被分配到设置的伺服放大器的IP地址(4)。
*: 对于IP地址(4), 在创建要通过运动控制设置功能设置的轴时需要进行设置。
5. 网络配置完成后选择菜单的[CC-Link IE TSN配置]⇒[PDO映射的批量设置]。



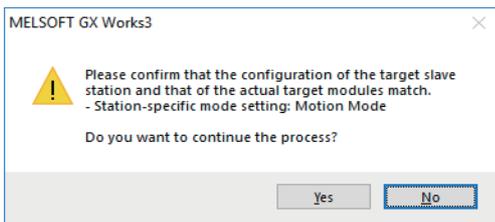
6. 显示“批量设置PDO映射的默认模式。”的信息时，单击[是]按钮。



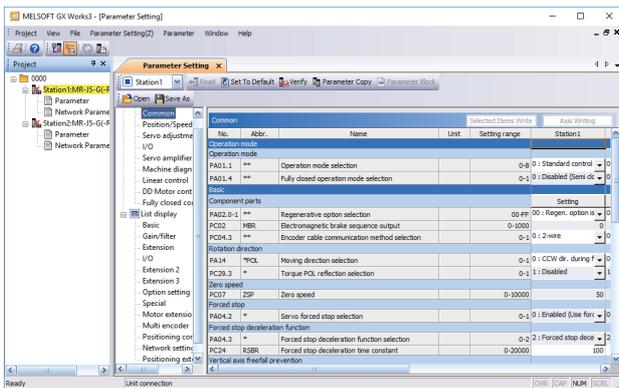
7. 显示“PDO映射的批量设置已完成。”的信息。单击[OK]按钮。



8. 从网络配置图双击要设置参数的伺服放大器(5)。

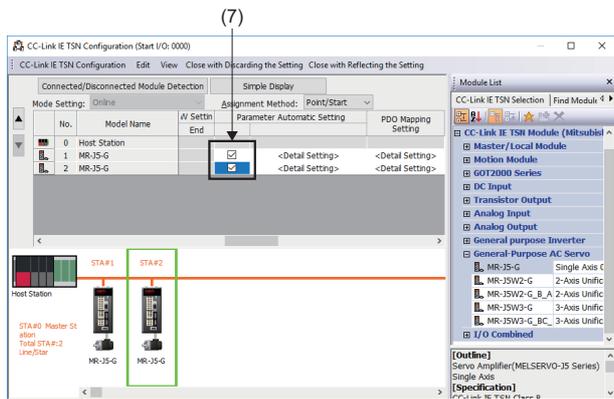


9. 显示“应确认执行对象从站的配置与实际的对象设备一致。”的信息时，单击[是]按钮。



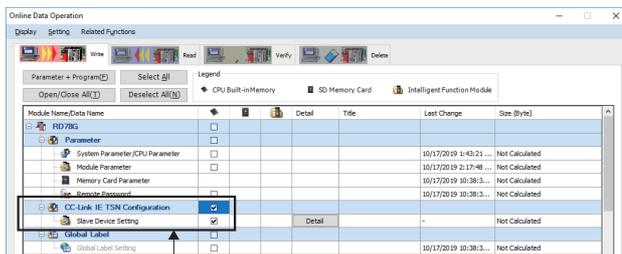
10. 显示参数设置画面。设置各轴的伺服参数。

11. 伺服参数的设置完成后，关闭参数设置画面。



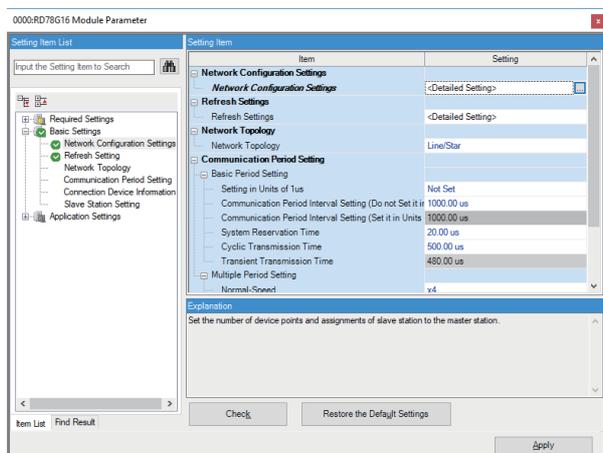
12. 返回到“CC-Link IE TSN配置”画面。勾选参数自动设置(6)的复选框。

*: 进行伺服参数的写入的情况下, 应勾选“在线数据操作”画面的“CC-Link IE TSN配置”(7)。



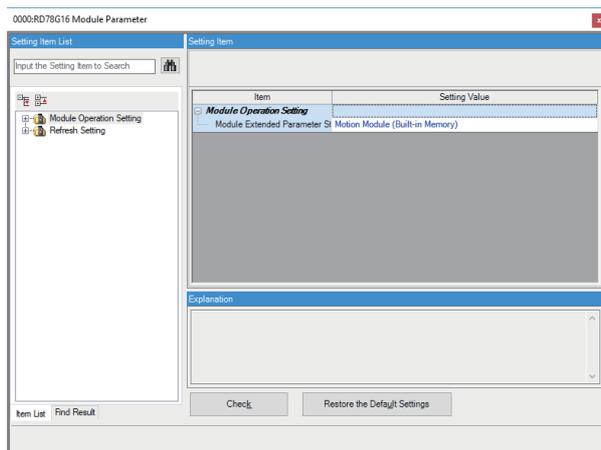
13. 选择[反映设置并关闭], 关闭“CC-Link IE TSN配置”画面。

14. 单击[应用]按钮, 完成模块参数(网络)的设置。



模块参数的设置

设置模块参数(运动)。



1. 选择[导航窗口]⇒[参数]⇒[模块信息]⇒[模块型号*1]⇒[模块参数(运动)], 显示参数编辑器。
设置“模块动作设置”的参数。

*1: 显示设置的运动模块的型号。

显示内容

■ 模块动作设置

项目	设置范围	初始值
模块扩展参数存储目标设置	运动模块(内置存储器)、运动模块(SD卡)	运动模块(内置存储器)

- 模块扩展参数存储目标设置

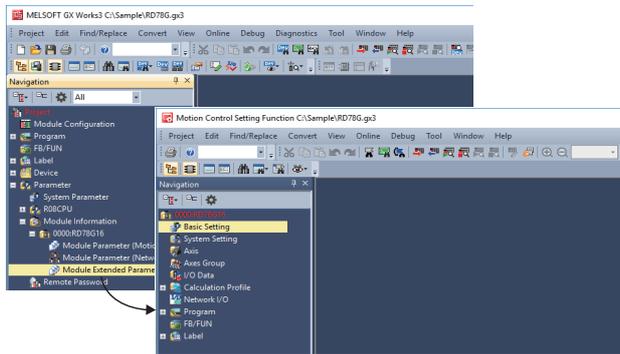
设置运动模块中使用的模块扩展参数的存储目标。从GX Works3写入模块扩展参数时，应在通过本参数指定的存储目标中进行写入。

设置	内容
运动模块(内置存储器)	使用运动模块的闪存中写入的模块扩展参数。
运动模块(SD卡)	使用运动模块的SD存储卡中写入的模块扩展参数。

模块扩展参数的设置

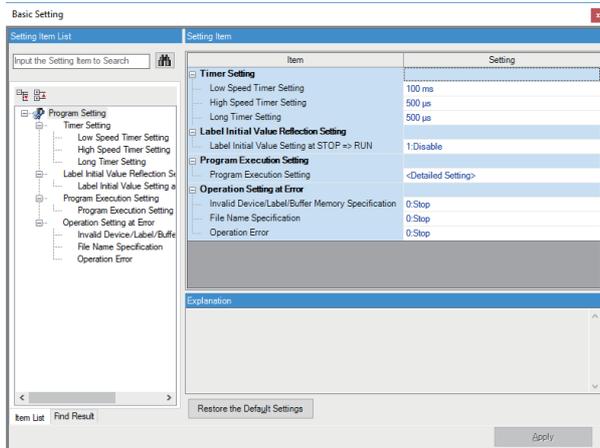
在运动控制设置功能画面上设置模块扩展参数。

■基本设置



1. 选择[导航窗口]⇒[参数]⇒[模块信息]⇒[模块型号*1]⇒[模块扩展参数]，显示“运动控制设置功能”画面。

*1: 显示设置的运动模块的型号。



2. 选择[导航窗口]⇒[基本设置]，显示参数编辑器。设置基本设置的参数。
3. 基本设置的参数设置完成后，单击[应用]按钮。

显示内容

• 程序设置

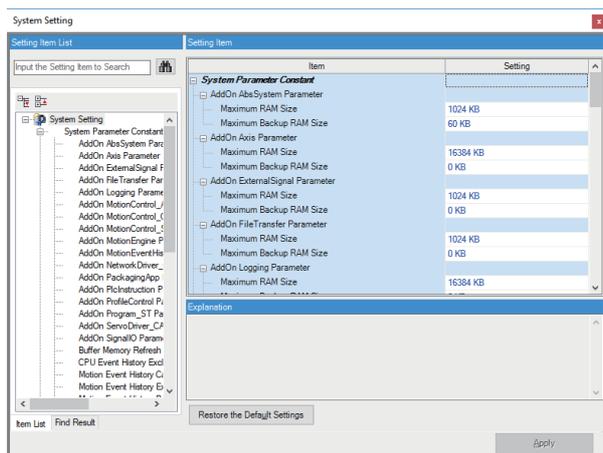
项目		设置范围	初始值
定时器设置	低速定时器设置	1~10000[ms]	100[ms]
	高速定时器设置	1~10000[μs]	500[μs]
	超长定时器设置	1~1000000[μs]	500[μs]
标签初始值反映设置	STOP→RUN时的标签初始值设置	0: 设置为有效 1: 设置为无效	1: 设置为无效
程序执行设置	程序执行设置	<详细设置>*1	—
异常时动作设置	软元件・标签・缓冲存储器指定不正确	0: 停止1:继续运行	0: 停止
	文件名指定		0: 停止
	运算异常		0: 停止

*1 更改要使用的程序的执行顺序、执行类型等的情况下进行此设置。

项目	设置内容	
执行顺序	执行顺序1~2048	
程序名	显示创建的程序名	
执行类型	0: 待机 1: 正常 2: 初始 3: 恒定周期	
程序执行条件*2	恒定周期间隔设置	单位
		恒定周期间隔
■设置1ns时 ・RG78G: 62500~600000[ns] ・RG78GH: 31250~600000[ns] ■设置0.01ms时 ・0.25~6000[ms]		

*2 仅在执行类型中选择“3: 恒定周期”时可以设置。

■ 系统设置



1. 选择[导航窗口]⇒[系统设置]，显示参数编辑器。设置系统设置的参数。
2. 系统设置的参数设置完成后，单击[应用]按钮。

显示内容

• 系统设置

项目	设置范围	初始值
[变量名・结构体名]		
系统参数常数		
插件□□参数 [□□]*1	RAM最大容量 [ADDON_PARAM. RamSizeMax]	—
	备份RAM最大容量 [ADDON_PARAM. BackupRamSizeMax]	—
缓冲存储器刷新周期设置 [System.PrConst.BufferMemoryRefreshCycle]	周期设置 [CYCLE_PARAM. Cycle]	—
	周期溢出警告检测次数 [CYCLE_PARAM. NumOfCycleOverWngDetectTimes]	0~65535
	周期溢出出错检测次数 [CYCLE_PARAM. NumOfCycleOverErrDetectTimes]	0~65535
	周期溢出出错选择 [CYCLE_PARAM. CycleOverErrorType]	0: 无 1: 警告 2: 轻度出错 3: 中度出错 4: 重度出错
CPU事件履历除外事件 [System.PrConst.EventHistoryCpu_Exclude]	—	0x700~0x7ff
运动事件履历容量 [System.PrConst.EventHistoryMotion_Capacity]	1~2048[kB]	128[kB]
运动事件履历除外事件 [System.PrConst.EventHistoryMotion_Exclude]	—	无设置
运动事件履历路径 [System.PrConst.EventHistoryMotion_Path]	—	/1ch/
除外报警 [System.PrConst.ExcludeWarning]	—	无设置
高速模式设置 [System.PrConst.FastOperationMode]	50EF: 高速模式 上述以外: 正常模式	0000
文件传送访问控制(锁存驱动器内文件) [System.PrConst.FileTransfer_AcFile_lch]	读取禁止/写入禁止、 读取禁止/写入允许、 读取允许/写入禁止、 读取允许/写入允许	读取允许/写入允许
文件传送访问控制(RAM驱动器内文件) [System.PrConst.FileTransfer_AcFile_ram]		读取允许/写入允许
文件传送访问控制(用户驱动器内文件) [System.PrConst.FileTransfer_AcFile_rom]		读取禁止/写入允许
文件传送访问控制(SD存储卡内文件) [System.PrConst.FileTransfer_AcFile_sdc]		读取允许/写入允许
文件传送访问控制(系统驱动器内文件) [System.PrConst.FileTransfer_AcFile_sys]		读取禁止/写入允许
文件传送访问控制(标签) [System.PrConst.FileTransfer_AcLabel]		读取允许/写入允许

项目 [变量名·结构体名]	设置范围		初始值		
系统参数常数	文件传送日志容量 [System.PrConst.FileTransfer_LogCapacity]	-1~2048[kB]	128[kB]		
	标签存储器容量 [System.PrConst.LabelMemorySize]	128~8192[kB]	1024[kB]		
	运动控制站发送接收数据更新方法 [System.PrConst.Link_MotionStationRefreshType]	1: 运算周期优先方式	1: 运算周期优先方式		
	自动记录允许 [System.PrConst.Logging_AutoLoggingEnable]	0: 禁止 1: 允许	0: 禁止		
	程序信息测量标志 [System.PrConst.MeasuresPrograms]	0: 不测量 1: 测量	1: 测量		
	制造商设置用 [System.PrConst.MultiThreadMode]	—	0000		
运算周期设置[1] [System.PrConst.OperationCycle[1].Cycle]	周期设置 [CYCLE_PARAM.Cycle]	—	0		
	周期溢出警告检测次数 [CYCLE_PARAM.NumOfCycleOverWngDetectTimes]	0~65535	1		
	周期溢出出错检测次数 [CYCLE_PARAM.NumOfCycleOverErrDetectTimes]	0~65535	5		
	周期溢出出错选择 [CYCLE_PARAM.CycleOverErrorType]	0: 无 1: 警告 2: 轻度出错 3: 中度出错 4: 重度出错	3: 中度出错		
软重启允许 [System.PrConst.SoftReboot_Enable]		0: 禁止 1: 允许	0: 禁止		
系统参数	所有轴紧急停止信号 [System.Pr.ForcedStop_Signal]	信号 [SIGNAL_SELECT.Source]	对象 [TARGET_REF.Target]	*2	无设置
		信号检测方法 [SIGNAL_SELECT.Detection]		0: TRUE时检测 1: FALSE时检测 2: FALSE→TRUE(上升沿)时检测 3: TRUE→FALSE(下降沿)时检测 4: 上升沿/下降沿时检测	0: TRUE时检测
		补偿时间 [SIGNAL_SELECT.CompensationTime]		—	0.0[s]
		滤波器时间 [SIGNAL_SELECT.FilterTime]		—	0.0[s]
	发生所有轴停止原因时停止选择 [System.Pr.StopMode_All]		1: 立即停止 2: 继续执行当前的加减速速度 3: 替代加减速速度		3: 替代加减速速度
所有轴停止时减速度 [System.Pr.StopMode_AllDeceleration]		■加减速方式为“加减速速度指定方式”的情况下 0.0000、0.0001~2147483647.0 ■加减速方式为“加减速时间恒定方式”的情况下 0.000000、0.000001~8400.0 *: “0”的情况下将立即停止。		0.0	

*1 □□为各插件参数的名称及变量名·结构体名。

项目 [变量名·结构体名]	初始值	
	RAM最大容量 [ADDON_PARAM.RamSizeMax]	备份RAM最大容量 [ADDON_PARAM.BackupRamSizeMax]
插件AbsSystem参数 [System.PrConst.Addon_AbsSystem]	1024[kB]	60[kB]
插件Axis参数 [System.PrConst.Addon_Axis]	16384[kB]	0[kB]
插件ExternalSignal参数 [System.PrConst.Addon_ExternalSignal]	1024[kB]	0[kB]
插件FileTransfer参数 [System.PrConst.Addon_FileTransfer]	1024[kB]	0[kB]

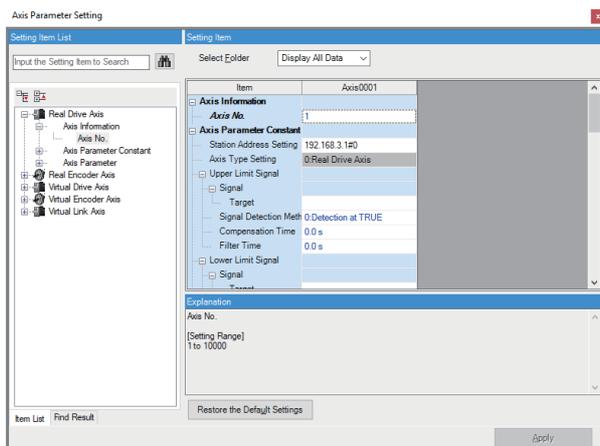
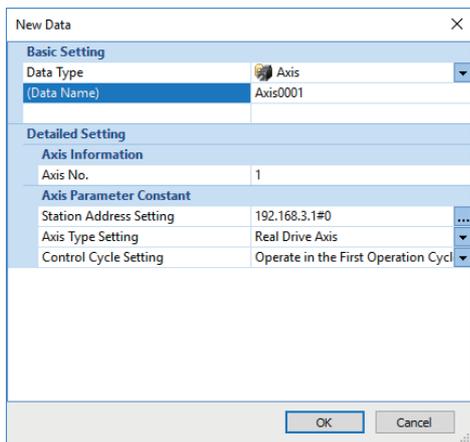
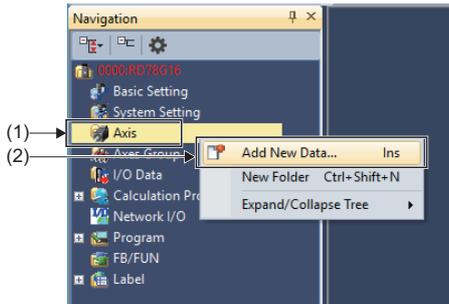
项目 [变量名·结构体名]	初始值	
	RAM最大容量 [ADDON_PARAM. RamSizeMax]	备份RAM最大容量 [ADDON_PARAM. BackupRamSizeMax]
插件Logging参数 [System.PrConst.Addon_Logging]	16384[kB]	0[kB]
插件MotionControl_AxisFilter参数 [System.PrConst.Addon_MotionControl_AxisFilter]	1024[kB]	0[kB]
插件MotionControl_General参数 [System.PrConst.Addon_MotionControl_General]	4096[kB]	0[kB]
插件MotionControl_Sync参数 [System.PrConst.Addon_MotionControl_Sync]	1024[kB]	0[kB]
插件MotionEngine参数 [System.PrConst.Addon_MotionEngine]	4096[kB]	0[kB]
插件MotionEventHist参数 [System.PrConst.Addon_MotionEventHist]	1024[kB]	0[kB]
插件NetworkDriver_CCIETSN参数 [System.PrConst.Addon_NetworkDriver_CCIETSN]	1024[kB]	0[kB]
插件PackagingApp参数 [System.PrConst.Addon_PackagingApp]	1024[kB]	0[kB]
插件PlcInstruction参数 [System.PrConst.Addon_PlcInstruction]	1024[kB]	0[kB]
插件ProfileControl参数 [System.PrConst.Addon_ProfileControl]	16384[kB]	0[kB]
插件Program_ST参数 [System.PrConst.Addon_Program_ST]	16384[kB]	0[kB]
插件ServoDriver_CANopen参数 [System.PrConst.Addon_ServoDriver_CANopen]	2048[kB]	0[kB]
插件ServoSystemRecorder参数 [System.PrConst.Addon_ServoSystemRecorder]	256[kB]	0[kB]
插件SignalIO参数 [System.PrConst.Addon_SignalIO]	1024[kB]	0[kB]

*2 设置对对象使用的信号。

项目	设置内容
源类型	全局标签、CANopen对象、软元件、常数
源型*3	BOOL、INT、DINT、WORD、DWORD、REAL、LREAL
源	—

*3 仅在源类型中选择“软元件”时可以设置。

■轴



*1 站地址设置中设置的值如下所示。

- 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

*2 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

*3 显示设置的数据名。

1. [导航窗口]⇒右键单击[轴](1)并选择[新建数据](2)。

2. 进行创建的轴的设置，并单击[OK]按钮。

- 数据名：设置ST程序中使用的轴的实例名。
- 站地址设置*1：
创建的轴为“实驱动轴”、“实编码器轴”的情况下，设置“CC-Link IE TSN配置”画面中确认的伺服放大器的IP地址、多点编号*2。
- 轴类型设置：
设置创建的轴的类型。轴类型有下述类型。
 - 实驱动轴
 - 实编码器轴
 - 虚拟驱动轴
 - 虚拟编码器轴
 - 虚拟连接轴

3. 在[导航窗口]⇒[轴]中添加[数据名*3]，显示参数编辑器。设置轴的参数。

4. 轴的参数设置完成后，单击[应用]按钮。

显示内容

可设置的轴类型有下述类型。

- 实驱动轴

项目 [变量名・结构体名]	设置范围		初始值	
轴信息 轴No. [AxisName, AxisRef, AxisNo]	1~10000		—	
轴参数常数	站地址设置 [AxisName, PrConst, AddressOfStation]		—	
	轴类型设置 [AxisName, PrConst, AxisType]		—	
	上限限位信号 [AxisName, PrConst, HwStrokeLimit_FlsSignal]	信号 [SIGNAL_SELECT, Source]	对象 [TARGET_REF, Target]	*1
		信号检测方法 [SIGNAL_SELECT, Detection]		0: TRUE时检测 1: FALSE时检测 2: FALSE→TRUE(上升沿)时检测 3: TRUE→FALSE(下降沿)时检测 4: 上升沿/下降沿时检测
		补偿时间 [SIGNAL_SELECT, CompensationTime]		—
		滤波器时间 [SIGNAL_SELECT, FilterTime]		—
	下限限位信号 [AxisName, PrConst, HwStrokeLimit_RlsSignal]	信号 [SIGNAL_SELECT, Source]	对象 [TARGET_REF, Target]	*1
		信号检测方法 [SIGNAL_SELECT, Detection]		0: TRUE时检测 1: FALSE时检测 2: FALSE→TRUE(上升沿)时检测 3: TRUE→FALSE(下降沿)时检测 4: 上升沿/下降沿时检测
		补偿时间 [SIGNAL_SELECT, CompensationTime]		—
		滤波器时间 [SIGNAL_SELECT, FilterTime]		—
	控制周期设置 [AxisName, PrConst, OperationCycle]		0: 在第1运算周期动作	0: 在第1运算周期动作
	绝对位置基准设置 [AxisName, PrConst, PosRestoration_AbsPosBase]		3: 进给机械位置	3: 进给机械位置
绝对位置管理设置 [AxisName, PrConst, PosRestoration_AbsPosEnable]		0: 不使用绝对位置系统 1: 使用绝对位置系统 -1: 自动设置(从连接设备获取)	-1: 自动设置(从连接设备获取)	
环形计数器有效选择 [AxisName, PrConst, RingCount_Enable]		0: 无效 1: 有效	0: 无效	
环形计数器下限值 [AxisName, PrConst, RingCount_LowerValue]		-1000000000000.0~ 1000000000000.0	-10000000000.0	
环形计数器上限值 [AxisName, PrConst, RingCount_UpperValue]		-1000000000000.0~ 1000000000000.0	10000000000.0	
从模拟有效 [AxisName, PrConst, SlaveEmulate_Enable]		0: 无效 1: 有效	0: 无效	
转矩限制最大值 [AxisName, PrConst, TorqueLimit_Max]		0.0~1000.0[%]	1000.0[%]	
负方向转矩限制初始值 [AxisName, PrConst, TorqueLimit_NegativeInitial]		0.0~1000.0[%]	300.0[%]	
正方向转矩限制初始值 [AxisName, PrConst, TorqueLimit_PositiveInitial]		0.0~1000.0[%]	300.0[%]	
高速模式设置 [AxisName, PrConst, FastOperationMode]		5FE2: 高速模式 上述以外: 正常模式	0000	

项目 [变量名·结构体名]			设置范围	初始值	
轴参数	加速度限制值 [AxisName.Pr.AccelerationLimit]		0.0000、0.0001~2147483647.0 *: “0.0000” 时不进行限制。	2147483647.0	
	启动时加减速速度0指定时动作选择 [AxisName.Pr.AccelerationZeroBehavior]		-1: 出错(不启动) 1: 最大加减速	-1: 出错(不启动)	
	指令到位宽度 [AxisName.Pr.CmdInPos_Width]		0.0: 功能无效 0.000000001~10000000000.0: 功能有效	100.0	
	减速度限制值 [AxisName.Pr.DecelerationLimit]		0.0000、0.0001~2147483647.0 *: “0.0000” 时不进行限制。	2147483647.0	
	驱动器单位转换分子 [AxisName.Pr.Drive_UnitConvRatioNum]		1~2147483647	1	
	驱动器单位转换分母 [AxisName.Pr.Drive_UnitConvRatioDen]		1~2147483647	1	
	紧急停止信号 [AxisName.Pr.ForcedStop_Signal]	信号 [SIGNAL_SELECT.Source]	对象 [TARGET_REF.Target]	*1	无设置
		信号检测方法 [SIGNAL_SELECT.Detection]		0: TRUE时检测 1: FALSE时检测 2: FALSE→TRUE(上升沿)时检测 3: TRUE→FALSE(下降沿)时检测 4: 上升沿/下降沿时检测	0: TRUE时检测
		补偿时间 [SIGNAL_SELECT.CompensationTime]		—	0.0[s]
		滤波器时间 [SIGNAL_SELECT.FilterTime]		—	0.0[s]
	原点复位要否设置 [AxisName.Pr.Homing_Required]		0: 不需要原点复位 1: 需要原点复位	1: 需要原点复位	
	Jerk限制值 [AxisName.Pr.JerkLimit]		0.0000、0.0001~2147483647.0 *: “0.0000” 时不进行限制。	2147483647.0	
	越程时动作设置 [AxisName.Pr.OverrunOperation]		1: 立即停止 2: 继续执行当前的加减速速度	1: 立即停止	
	原点复位未完时启动允许 [AxisName.Pr.StartableAtUnhomed]		0: 不允许 1: 允许	0: 不允许	
	停止时减速度 [AxisName.Pr.StopMode_Deceleration]		■加减速方式为“加减速速度指定方式”的情况下 0.0000、0.0001~2147483647.0 ■加减速方式为“加减速时间恒定方式”的情况下 0.0000、0.000001~8400.0	0.0	
	减速停止时停止处理选择 [AxisName.Pr.StopMode_DecelerationCurve]		1: 重新创建减速曲线	1: 重新创建减速曲线	
	发生停止原因时停止选择 [AxisName.Pr.StopMode_General]		1: 立即停止 2: 继续执行当前的加减速速度 3: 替代加减速速度	3: 替代加减速速度	
	发生硬件行程限位出错时停止选择 [AxisName.Pr.StopMode_HwStrokeLimit]		1: 立即停止 2: 继续执行当前的加减速速度 3: 替代加减速速度	1: 立即停止	
	运行中伺服OFF指令时处理选择 [AxisName.Pr.StopMode_ServoOff]		0: 忽略 4: 立即停止后伺服OFF 5: 减速停止后伺服OFF	0: 忽略	
	发生软件行程限位出错时停止选择 [AxisName.Pr.StopMode_SwStrokeLimit]		1: 立即停止 2: 继续执行当前的加减速速度 3: 替代加减速速度	1: 立即停止	
驱动器指令删除检测设置 [AxisName.Pr.StopOption_DriverTargetIgnored]		0: 检测无效 1: 检测有效	1: 检测有效		

项目 [变量名·结构体名]				设置范围	初始值
轴参数	停止信号 [AxisName.Pr.StopSignal]	信号 [SIGNAL_SELECT.Source]	对象 [TARGET_REF.Target]	*1	无设置
		信号检测方法 [SIGNAL_SELECT.Detection]		0: TRUE时检测 1: FALSE时检测 2: FALSE→TRUE(上升沿)时检测 3: TRUE→FALSE(下降沿)时检测 4: 上升沿/下降沿时检测	0: TRUE时检测
		补偿时间 [SIGNAL_SELECT.CompensationTime]		—	0.0[s]
		滤波器时间 [SIGNAL_SELECT.FilterTime]		—	0.0[s]
	软件行程限位下限值 [AxisName.Pr.SwStrokeLimit_Lower]			-1000000000000.0~ 1000000000000.0	-100000000000.0
	软件行程限位对象 [AxisName.Pr.SwStrokeLimit_Target]			-1: 无效 1: 指令当前位置 3: 进给机械位置	-1: 无效
	软件行程限位上限值 [AxisName.Pr.SwStrokeLimit_Upper]			-1000000000000.0~ 1000000000000.0	100000000000.0
	位置指令单位 [AxisName.Pr.Unit_Position]			pulse、m、mm、μm、nm、degree、 Revolution、inch、任意单位	pulse
	位置指令单位字符串 [AxisName.Pr.Unit_PositionString]			—	无设置
	速度指令单位 [AxisName.Pr.Unit_Velocity]			U/s、U/ms、U/μs、U/ns、U/min	U/s
	负方向速度限制值 [AxisName.Pr.VelocityLimit_Negative]			0.0001~2500000000.0	2500000000.0
速度限制值溢出时动作设置 [AxisName.Pr.VelocityLimit_OverOperation]			0: 忽略 3: 立即停止	0: 忽略	
正方向速度限制值 [AxisName.Pr.VelocityLimit_Positive]			0.0001~2500000000.0	2500000000.0	

*1 设置对对象使用的信号。

项目	设置内容
源类型	全局标签、CANopen对象、软元件、常数
源型*2	BOOL、INT、DINT、WORD、DWORD、REAL、LREAL
源	—

*2 仅在源类型中选择“软元件”时可以设置。

• 实编码器轴

项目 [变量名·结构体名]	设置范围		初始值		
轴信息	轴No. [AxisName, AxisRef, AxisNo]	1~10000	—		
轴参数常数	站地址设置 [AxisName, PrConst, AddressOfStation]	—	无设置		
	轴类型设置 [AxisName, PrConst, AxisType]	—	2: 实编码器轴		
	实编码器轴类型设置 [AxisName, PrConst, Encoder_AxisType]	1: 经由驱动器模块	1: 经由驱动器模块		
	计数器禁用信号 [AxisName, PrConst, Encoder_CounterDisableSignal]	信号 [SIGNAL_SELECT, Source]	*1	无设置	
		对象 [TARGET_REF, Target]			
		信号检测方法 [SIGNAL_SELECT, Detection]	0: TRUE时检测 1: FALSE时检测 2: FALSE→TRUE(上升沿)时检测 3: TRUE→FALSE(下降沿)时检测 4: 上升沿/下降沿时检测		0: TRUE时检测
		补偿时间 [SIGNAL_SELECT, CompensationTime]	—		0.0[s]
		滤波器时间 [SIGNAL_SELECT, FilterTime]	—	0.0[s]	
	控制周期设置 [AxisName, PrConst, OperationCycle]	0: 在第1运算周期动作	0: 在第1运算周期动作		
	绝对位置管理设置 [AxisName, PrConst, PosRestoration_AbsPosEnable]	0: 不使用绝对位置系统 1: 使用绝对位置系统	-1: 自动设置(从连接设备获取) -1: 自动设置(从连接设备获取)		
环形计数器有效选择 [AxisName, PrConst, RingCount_Enable]	0: 无效 1: 有效	0: 无效			
环形计数器下限值 [AxisName, PrConst, RingCount_LowerValue]	-1000000000000.0~ 1000000000000.0	-100000000000.0			
环形计数器上限值 [AxisName, PrConst, RingCount_UpperValue]	-1000000000000.0~ 1000000000000.0	100000000000.0			
高速模式设置 [AxisName, PrConst, FastOperationMode]	0000*3	0000			
轴参数	驱动器单位转换分子 [AxisName, Pr, Drive_UnitConvRatioNum]	1~2147483647	1		
	驱动器单位转换分母 [AxisName, Pr, Drive_UnitConvRatioDen]	1~2147483647	1		
	原点复位要否设置 [AxisName, Pr, Homing_Required]	0: 不需要原点复位 1: 需要原点复位	1: 需要原点复位		
	原点复位未完时启动允许 [AxisName, Pr, StartableAtUnhomed]	0: 不允许 1: 允许	0: 不允许		
	位置指令单位 [AxisName, Pr, Unit_Position]	pulse、m、mm、μm、nm、degree、 Revolution、inch、任意单位	pulse		
	位置指令单位字符串 [AxisName, Pr, Unit_PositionString]	—	无设置		
	速度指令单位 [AxisName, Pr, Unit_Velocity]	U/s、U/ms、U/μs、U/ns、U/min	U/s		

*1 设置对对象使用的信号。

项目	设置内容
源类型	全局标签、CANopen对象、软元件、常数
源型*2	BOOL、INT、DINT、WORD、DWORD、REAL、LREAL
源	—

*2 仅在源类型中选择“软元件”时可以设置。

*3 请勿更改本设置。

• 虚拟驱动轴

项目 [变量名·结构体名]		设置范围	初始值
轴信息	轴No. [AxisName, AxisRef, AxisNo]	1~10000	—
轴参数常数	轴类型设置 [AxisName, PrConst, AxisType]	—	3: 虚拟驱动轴
	上限限位信号 [AxisName, PrConst, HwSt rokeLimit_FlsSignal]	信号 [SIGNAL_SELECT, Source] 对象 [TARGET_REF, Target]	*1 无设置
		信号检测方法 [SIGNAL_SELECT, Detection]	0: TRUE时检测 1: FALSE时检测 2: FALSE→TRUE(上升沿)时检测 3: TRUE→FALSE(下降沿)时检测 4: 上升沿/下降沿时检测
		补偿时间 [SIGNAL_SELECT, CompensationTime]	— 0.0[s]
		滤波器时间 [SIGNAL_SELECT, FilterTime]	— 0.0[s]
	下限限位信号 [AxisName, PrConst, HwSt rokeLimit_RlsSignal]	信号 [SIGNAL_SELECT, Source] 对象 [TARGET_REF, Target]	*1 无设置
		信号检测方法 [SIGNAL_SELECT, Detection]	0: TRUE时检测 1: FALSE时检测 2: FALSE→TRUE(上升沿)时检测 3: TRUE→FALSE(下降沿)时检测 4: 上升沿/下降沿时检测
		补偿时间 [SIGNAL_SELECT, CompensationTime]	— 0.0[s]
		滤波器时间 [SIGNAL_SELECT, FilterTime]	— 0.0[s]
	控制周期设置 [AxisName, PrConst, OperationCycle]	0: 在第1运算周期动作	0: 在第1运算周期动作
	绝对位置管理设置 [AxisName, PrConst, PosRestoration_AbsPosEnable]	0: 不使用绝对位置系统 1: 使用绝对位置系统 -1: 自动设置(从连接设备获取)	0: 不使用绝对位置系统
	环形计数器有效选择 [AxisName, PrConst, RingCount_Enable]	0: 无效 1: 有效	0: 无效
	环形计数器下限值 [AxisName, PrConst, RingCount_LowerValue]	-1000000000000.0~ 1000000000000.0	-100000000000.0
	环形计数器上限值 [AxisName, PrConst, RingCount_UpperValue]	-1000000000000.0~ 1000000000000.0	100000000000.0
	高速模式设置 [AxisName, PrConst, FastOperationMode]	0000*3	0000

项目 [变量名·结构体名]			设置范围	初始值	
轴参数	加速度限制值 [AxisName.Pr.AccelerationLimit]		0.0000、0.0001~2147483647.0 *: “0.0000” 时不进行限制。	2147483647.0	
	启动时加减速速度0指定时动作选择 [AxisName.Pr.AccelerationZeroBehavior]		-1: 出错(不启动) 1: 最大加减速	-1: 出错(不启动)	
	指令到位宽度 [AxisName.Pr.CmdInPos_Width]		0.0: 功能无效 0.000000001~1000000000.0: 功能有效	100.0	
	减速度限制值 [AxisName.Pr.DecelerationLimit]		0.0000、0.0001~2147483647.0 *: “0.0000” 时不进行限制。	2147483647.0	
	紧急停止信号 [AxisName.Pr.ForcedStop_Signal]	信号 [SIGNAL_SELECT.Source]	对象 [TARGET_REF.Target]	*1	无设置
		信号检测方法 [SIGNAL_SELECT.Detection]		0: TRUE时检测 1: FALSE时检测 2: FALSE→TRUE(上升沿)时检测 3: TRUE→FALSE(下降沿)时检测 4: 上升沿/下降沿时检测	0: TRUE时检测
		补偿时间 [SIGNAL_SELECT.CompensationTime]		—	0.0[s]
		滤波器时间 [SIGNAL_SELECT.FilterTime]		—	0.0[s]
	原点复位要否设置 [AxisName.Pr.Homing_Required]		0: 不需要原点复位 1: 需要原点复位	1: 需要原点复位	
	Jerk限制值 [AxisName.Pr.JerkLimit]		0.0000、0.0001~2147483647.0 *: “0.0000” 时不进行限制。	2147483647.0	
	越程时动作设置 [AxisName.Pr.OvrerrunOperation]		1: 立即停止 2: 继续执行当前的加减速速度	1: 立即停止	
	原点复位未完时启动允许 [AxisName.Pr.StartableAtUnhomed]		0: 不允许 1: 允许	0: 不允许	
	停止时减速度 [AxisName.Pr.StopMode_Deceleration]		■加减速方式为“加减速速度指定方式”的情况下 0.0000、0.0001~2147483647.0 ■加减速方式为“加减速时间恒定方式”的情况下 0.0000、0.000001~8400.0	0.0	
	减速停止时停止处理选择 [AxisName.Pr.StopMode_DecelerationCurve]		1: 重新创建减速曲线	1: 重新创建减速曲线	
	发生停止原因时停止选择 [AxisName.Pr.StopMode_General]		1: 立即停止 2: 继续执行当前的加减速速度 3: 替代加减速速度	3: 替代加减速速度	
	发生硬件行程限位出错时停止选择 [AxisName.Pr.StopMode_HwStrokeLimit]		1: 立即停止 2: 继续执行当前的加减速速度 3: 替代加减速速度	1: 立即停止	
	运行中伺服OFF指令时处理选择 [AxisName.Pr.StopMode_ServoOff]		0: 忽略 4: 立即停止后伺服OFF 5: 减速停止后伺服OFF	0: 忽略	
	发生软件行程限位出错时停止选择 [AxisName.Pr.StopMode_SwStrokeLimit]		1: 立即停止 2: 继续执行当前的加减速速度 3: 替代加减速速度	1: 立即停止	
	停止信号 [AxisName.Pr.StopSignal1]	信号 [SIGNAL_SELECT.Source]	对象 [TARGET_REF.Target]	*1	无设置
		信号检测方法 [SIGNAL_SELECT.Detection]		0: TRUE时检测 1: FALSE时检测 2: FALSE→TRUE(上升沿)时检测 3: TRUE→FALSE(下降沿)时检测 4: 上升沿/下降沿时检测	0: TRUE时检测
补偿时间 [SIGNAL_SELECT.CompensationTime]		—	0.0[s]		
滤波器时间 [SIGNAL_SELECT.FilterTime]		—	0.0[s]		

项目 [变量名·结构体名]	设置范围	初始值
轴参数		
软件行程限位下限值 [AxisName.Pr.SwStrokeLimit_Lower]	-1000000000000.0~ 1000000000000.0	-100000000000.0
软件行程限位对象 [AxisName.Pr.SwStrokeLimit_Target]	-1: 无效 1: 指令当前位置 3: 进给机械位置	-1: 无效
软件行程限位上限值 [AxisName.Pr.SwStrokeLimit_Upper]	-1000000000000.0~ 1000000000000.0	100000000000.0
位置指令单位 [AxisName.Pr.Unit_Position]	pulse、m、mm、μm、nm、degree、 Revolution、inch、任意单位	pulse
位置指令单位字符串 [AxisName.Pr.Unit_PositionString]	—	无设置
速度指令单位 [AxisName.Pr.Unit_Velocity]	U/s、U/ms、U/μs、U/ns、U/min	U/s
负方向速度限制值 [AxisName.Pr.VelocityLimit_Negative]	0.0001~2500000000.0	2500000000.0
速度限制值溢出时动作设置 [AxisName.Pr.VelocityLimit_OverOperation]	0: 忽略 3: 立即停止	0: 忽略
正方向速度限制值 [AxisName.Pr.VelocityLimit_Positive]	0.0001~2500000000.0	2500000000.0

*1 设置对对象使用的信号。

项目	设置内容
源类型	全局标签、CANopen对象、软元件、常数
源型*2	BOOL、INT、DINT、WORD、DWORD、REAL、LREAL
源	—

*2 仅在源类型中选择“软元件”时可以设置。

*3 请勿更改本设置。

• 虚拟编码器轴

项目 [变量名·结构体名]	设置范围	初始值
轴信息 轴No. [AxisName, AxisRef, AxisNo]	1~10000	—
轴参数常数 轴类型设置 [AxisName, PrConst, AxisType]	—	4: 虚拟编码器轴
计数器禁用信号 [AxisName, PrConst, Encoder_CounterDisableSignal]	信号 [SIGNAL_SELECT, Source] 对象 [TARGET_REF, Target]	*1 无设置
信号检测方法 [SIGNAL_SELECT, Detection]	0: TRUE时检测 1: FALSE时检测 2: FALSE→TRUE(上升沿)时检测 3: TRUE→FALSE(下降沿)时检测 4: 上升沿/下降沿时检测	0: TRUE时检测
补偿时间 [SIGNAL_SELECT, CompensationTime]	—	0.0[s]
滤波器时间 [SIGNAL_SELECT, FilterTime]	—	0.0[s]
编码器环形计数器下限值 [AxisName, PrConst, Encoder_RingCount_LowerValue]	■PosActualValue为1字的情况下 -32768~32767	0
编码器环形计数器上限值 [AxisName, PrConst, Encoder_RingCount_UpperValue]	■PosActualValue为2字或省略的情况下 -2147483648~2147483647	0
控制周期设置 [AxisName, PrConst, OperationCycle]	0: 在第1运算周期动作	0: 在第1运算周期动作
绝对位置管理设置 [AxisName, PrConst, PosRestoration_AbsPosEnable]	0: 不使用绝对位置系统 1: 使用绝对位置系统 -1: 自动设置(从连接设备获取)	-1: 自动设置(从连接设备获取)
环形计数器有效选择 [AxisName, PrConst, RingCount_Enable]	0: 无效 1: 有效	0: 无效
环形计数器下限值 [AxisName, PrConst, RingCount_LowerValue]	-1000000000000.0~ 1000000000000.0	-10000000000.0
环形计数器上限值 [AxisName, PrConst, RingCount_UpperValue]	-1000000000000.0~ 1000000000000.0	10000000000.0
从对象数据 [AxisName, PrConst, SlaveObject]	PosActualValue [SLAVE_OBJECT_VIRTUAL_ENCODER, PosActualValue]	*1 无设置
高速模式设置 [AxisName, PrConst, FastOperationMode]	0000*3	0000
轴参数 驱动器单位转换分子 [AxisName, Pr, Drive_UnitConvRatioNum]	1~2147483647	1
驱动器单位转换分母 [AxisName, Pr, Drive_UnitConvRatioDen]	1~2147483647	1
原点复位要否设置 [AxisName, Pr, Homing_Required]	0: 不需要原点复位 1: 需要原点复位	1: 需要原点复位
原点复位未完时启动允许 [AxisName, Pr, StartableAtUnhomed]	0: 不允许 1: 允许	0: 不允许
位置指令单位 [AxisName, Pr, Unit_Position]	pulse、m、mm、μm、nm、degree、 Revolution、inch、任意单位	pulse
位置指令单位字符串 [AxisName, Pr, Unit_PositionString]	—	无设置
速度指令单位 [AxisName, Pr, Unit_Velocity]	U/s、U/ms、U/μs、U/ns、U/min	U/s

*1 设置对对象使用的信号。

项目	设置内容
源类型	全局标签、CANopen对象、软元件、常数
源型*2	BOOL、INT、DINT、WORD、DWORD、REAL、LREAL
源	—

*2 仅在源类型中选择“软元件”时可以设置。

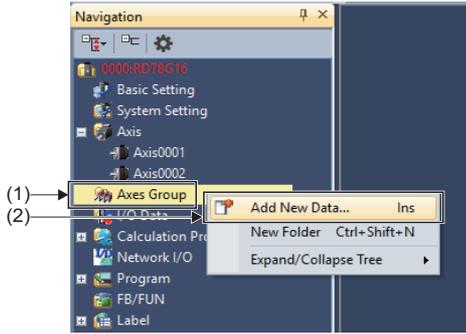
*3 请勿更改本设置。

• 虚拟连接轴

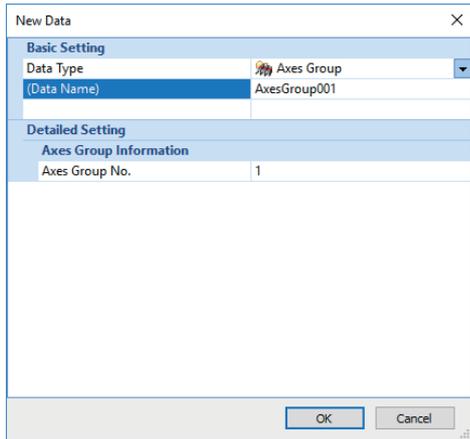
项目 [变量名·结构体名]		设置范围	初始值
轴信息	轴No. [AxisName, AxisRef, AxisNo]	1~10000	—
轴参数常数	轴类型设置 [AxisName, PrConst, AxisType]	—	5: 虚拟连接轴
	控制周期设置 [AxisName, PrConst, OperationCycle]	0: 在第1运算周期动作	0: 在第1运算周期动作
	绝对位置管理设置 [AxisName, PrConst, PosRestoration_AbsPosEnable]	0: 不使用绝对位置系统 1: 使用绝对位置系统 -1: 自动设置(从连接设备获取)	0: 不使用绝对位置系统
	环形计数器有效选择 [AxisName, PrConst, RingCount_Enable]	0: 无效 1: 有效	0: 无效
	环形计数器下限值 [AxisName, PrConst, RingCount_LowerValue]	-1000000000000.0~ 1000000000000.0	-100000000000.0
	环形计数器上限值 [AxisName, PrConst, RingCount_UpperValue]	-1000000000000.0~ 1000000000000.0	10000000000.0
	高速模式设置 [AxisName, PrConst, FastOperationMode]	0000*1	0000
轴参数	原点复位要否设置 [AxisName, Pr, Homing_Required]	0: 不需要原点复位 1: 需要原点复位	0: 不需要原点复位
	原点复位未完时启动允许 [AxisName, Pr, StartableAtUnhomed]	0: 不允许 1: 允许	0: 不允许
	位置指令单位 [AxisName, Pr, Unit_Position]	pulse、m、mm、μm、nm、degree、 Revolution、inch、任意单位	pulse
	位置指令单位字符串 [AxisName, Pr, Unit_PositionString]	—	无设置
	速度指令单位 [AxisName, Pr, Unit_Velocity]	U/s、U/ms、U/μs、U/ns、U/min	U/s

*1 请勿更改本设置。

■轴组

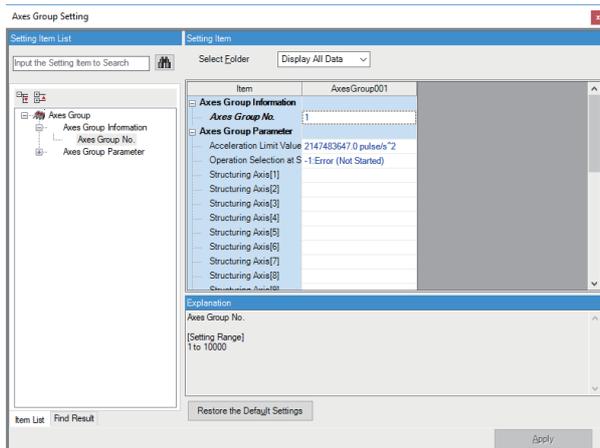


1. [导航窗口]⇒右键单击[轴组](1)并选择[新建数据](2)。



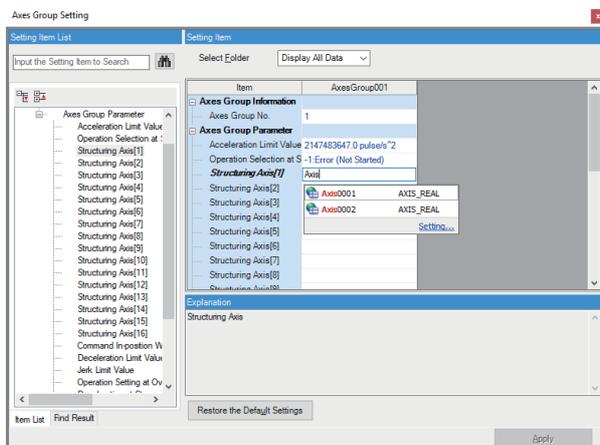
2. 进行创建的轴组的设置，并单击[OK]按钮。

- 数据名：
设置ST程序中使用的轴组的实例名。



3. 在[导航窗口]⇒[轴组]中添加[数据名*1]，显示参数编辑器。

*1: 显示设置的数据名。



4. 设置创建的轴组的构成轴。构成轴设置[导航窗口]⇒[轴]中设置的[数据名](ST程序中使用的轴的实例名)。

5. 轴组的参数设置完成后，单击[应用]按钮。

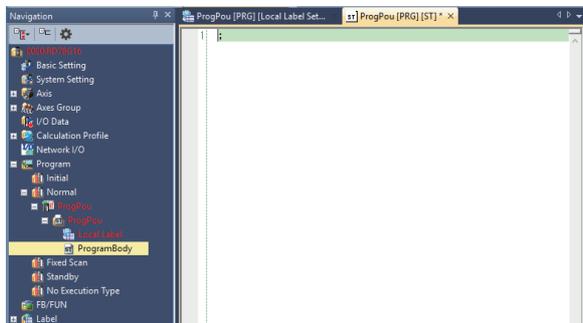
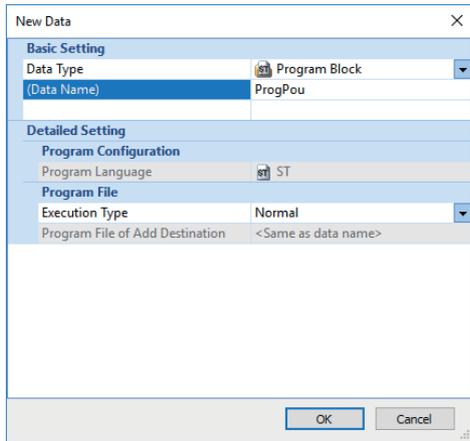
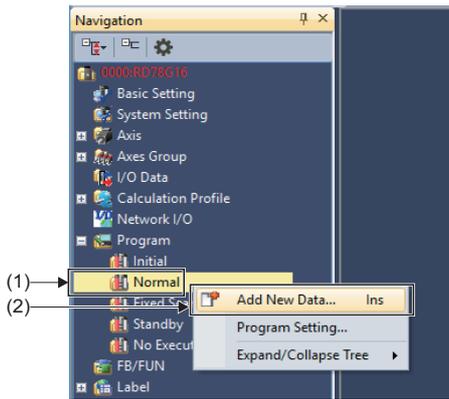
显示内容

• 轴组

项目 [变量名·结构体名]		设置范围	初始值
轴组信息	轴组No. [AxesGroupName, AxesGroupRef, GroupNo]	1~10000	—
轴组参数	加速度限制值 [AxesGroupName, Pr, AccelerationLimit]	0.0000、0.0001~2147483647.0 *: “0.0000” 时不进行限制。	2147483647.0
	启动时加减速速度0指定时动作选择 [AxesGroupName, Pr, AccelerationZeroBehavior]	-1: 出错(不启动) 1: 最大加减速速度	-1: 出错(不启动)
	构成轴[□] [AxesGroupName, Pr, Axis[□]] *: □=1~16	—	无设置
	指令到位宽度 [AxesGroupName, Pr, CmdInPos_Width]	0.0: 功能无效 0.000000001~10000000000.0: 功能有效	100.0
	减速度限制值 [AxesGroupName, Pr, DecelerationLimit]	0.0000、0.0001~2147483647.0 *: “0.0000” 时不进行限制。	2147483647.0
	Jerk限制值 [AxesGroupName, Pr, JerkLimit]	0.0000、0.0001~2147483647.0 *: “0.0000” 时不进行限制。	2147483647.0
	越程时动作设置 [AxesGroupName, Pr, OverrunOperation]	1: 立即停止	1: 立即停止
	停止时减速度 [AxesGroupName, Pr, StopMode_Deceleration]	■加减速方式为“加减速速度指定方式”的情况下 0.0000、0.0001~2147483647.0 ■加减速方式为“加减速时间恒定方式”的情况下 0.0000、0.000001~8400.0	0.0
	减速停止时停止处理选择 [AxesGroupName, Pr, StopMode_DecelerationCurve]	1: 重新创建减速曲线	1: 重新创建减速曲线
	发生轴停止原因时构成轴动作选择 [AxesGroupName, Pr, StopMode_ErrorInGroup]	1: 立即停止	1: 立即停止
	发生停止原因时停止选择 [AxesGroupName, Pr, StopMode_General]	1: 立即停止 2: 继续执行当前的加减速速度 3: 替代加减速速度	3: 替代加减速速度
	位置指令单位 [AxesGroupName, Pr, Unit_Position]	pulse、m、mm、μm、nm、degree、Revolution、inch、任意单位	pulse
	位置指令单位字符串 [AxesGroupName, Pr, Unit_PositionString]	—	无设置
	速度指令单位 [AxesGroupName, Pr, Unit_Velocity]	U/s、U/ms、U/μs、U/ns、U/min	U/s
速度限制值 [AxesGroupName, Pr, VelocityLimit]	0.0001~2500000000.0	2500000000.0	

ST程序的创建

创建使用ST语言的程序。



1. [导航窗口]⇒右键单击[程序](1)并选择[新建数据](2)。

2. 进行创建的程序的设置，并单击[OK]按钮。

- 数据名：
设置程序的实例名。
- 执行类型*1：
设置程序的执行类型。

*1: 关于执行类型的详细内容，请参阅程序的执行。(☞ 391页 程序的执行)

3. 在[导航窗口]⇒[程序]⇒[执行类型*2]中添加 [ProgPou]⇒[ProgPou]⇒[局部标签]/[程序本体]，显示程序的工作窗口。

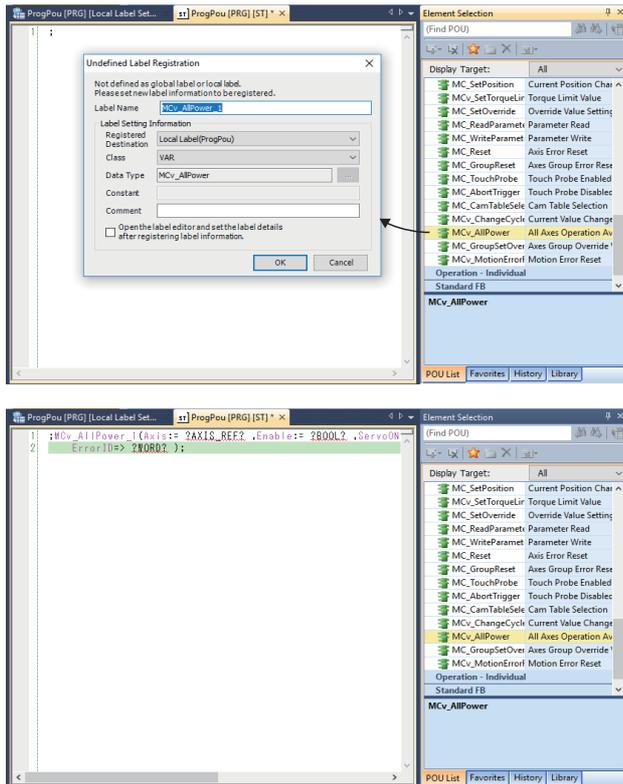
4. 创建ST程序。

*2: 显示执行类型中指定的执行类型名。

*3: 在执行类型中指定的执行类型的树状图中创建程序文件。

运动控制FB的插入

将运动控制FB通过拖放放置到ST程序上并创建程序。



1. 选择[导航窗口]⇒[程序]⇒[执行类型*1]⇒[ProgPou]⇒[ProgPou]⇒[程序本体]，显示程序的工作窗口。
2. 从部件选择窗口选择要插入的FB，将其拖放到要放置的位置时，将显示“未定义标签登录”画面。
3. 设置登录的标签名、标签设置信息，单击[OK]按钮。

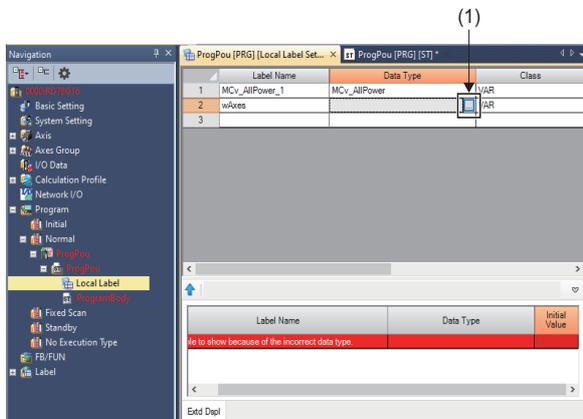
*1: 显示创建的执行类型名。

4. FB被插入到程序的工作窗口中。

*2: 从部件选择窗口插入FB时，标签通过插入的FB的标签名被登录到局部标签设置的一览中。

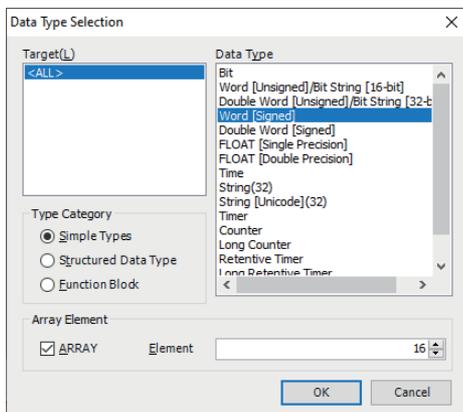
标签的登录

登录ST程序中使用的局部标签。



1. 选择[导航窗口]⇒[程序]⇒[执行类型*1]⇒[ProgPou]⇒[ProgPou]⇒[局部标签]，显示局部标签的工作窗口。
2. 进行标签名、数据类型、分类的设置并登录。
3. 设置数据类型的情况下，单击[...] (1)。

*1: 显示创建的执行类型名。



4. 进行创建的数据类型的设置，并单击[OK]按钮。

- 对象：设置创建的标签的分类。
- 数据类型：设置创建的标签的数据类型。
- 类型分类：设置创建的标签的分类。
- 数组元素：使用数组的情况下，检查数组并设置元素数。

使用公开标签功能的程序的创建步骤

以下对使用工程工具创建使用公开标签功能的顺控程序的步骤进行说明。

1. 进行“使用ST程序的程序的创建步骤”的步骤1.~4.。

- 参数的设置 (☞ 398页 参数的设置)
- 网络的设置 (☞ 399页 网络的设置)
- 模块参数的设置 (☞ 402页 模块参数的设置)
- 模块扩展参数的设置 (☞ 403页 模块扩展参数的设置)

2. 标签的登录

通过“运动控制设置功能”登录运动模块中使用的标签。

在全局标签中登录在CPU模块侧公开的标签。

(☞ 422页 标签的登录)

3. 公开标签的设置

进行用于将在“运动控制设置功能”中登录的全局标签公开到CPU模块侧的设置。

(☞ 423页 公开标签的设置)

4. 标签转换

为了确定在“运动控制设置功能”中设置的内容，进行标签转换。

检查设置的标签并创建要公开的标签信息。

(☞ 424页 标签转换)

5. 公开标签的反映

将通过“运动控制设置功能”创建的公开标签信息反映到CPU模块侧的工程中。

反映完成时，将其作为模块标签自动登录到CPU模块侧。

(☞ 426页 公开标签的反映)

6. 顺控程序的创建

使用公开的运动模块的标签、运动控制FB，创建顺控程序。

- 公开标签的插入 (☞ 429页 公开标签的插入)
- 运动控制FB的插入 (☞ 429页 运动控制FB的插入)

7. 创建的程序的写入

将在CPU模块侧及运动模块侧创建的程序、全局标签及模块参数写入到各模块中。

(☞ 430页 创建的程序的写入)

要点

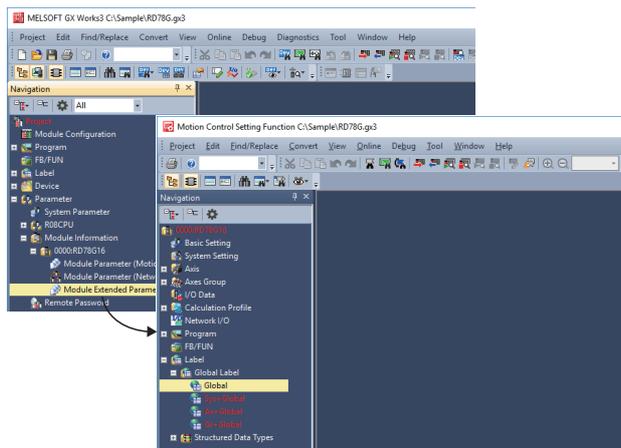
在CPU模块侧使用运动控制FB创建顺控程序的情况下，需要在工程工具的库中登录运动模块FB库。

关于运动模块FB库的登录，请参阅下述章节。

☞ 431页 运动模块FB库的登录

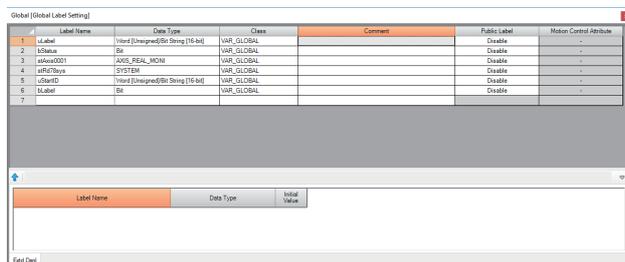
标签的登录

在运动控制设置功能画面上登录运动模块中使用的标签(全局标签)。



1. 选择[导航窗口]⇒[参数]⇒[模块信息]⇒[模块型号*1]⇒[模块扩展参数], 显示“运动控制设置功能”画面。

*1: 显示设置的运动模块的型号。



2. 选择[导航窗口]⇒[标签]⇒[全局标签]⇒[Global*2], 显示全局标签的标签编辑器。登录运动模块中使用的标签。

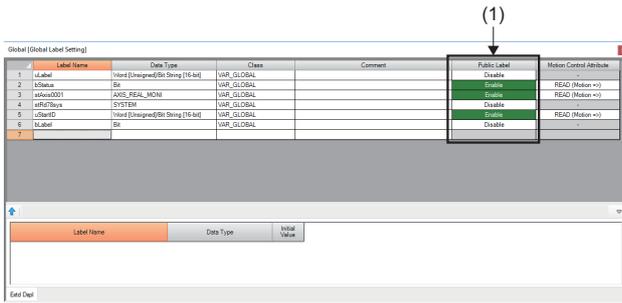
*2: 还可以新建全局标签列表并登录标签。

要点

在CPU模块侧公开全局标签中登录的标签。登录到局部标签的标签不可公开。

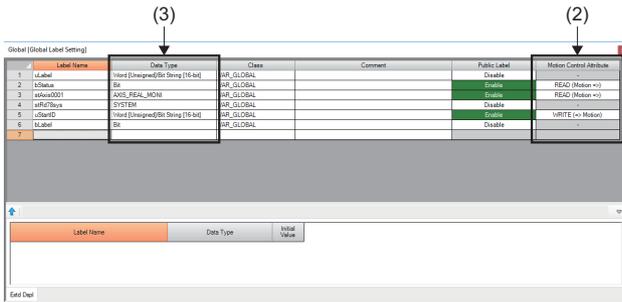
公开标签的设置

为了在CPU模块侧公开登录的全局标签，设置“公开标签”及“运动控制属性”。



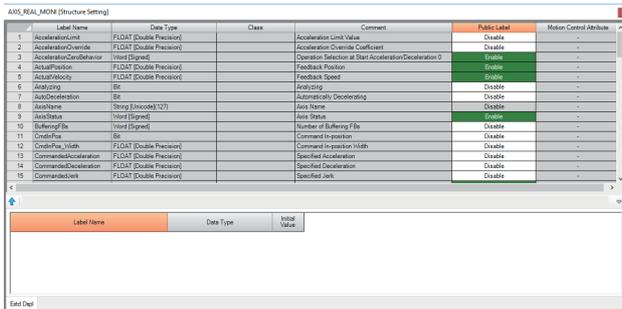
1. 在标签编辑器中，将公开到CPU模块侧的标签名的公开标签(1)设置为“有效”。

*1: 根据标签的数据类型、分类的设置，无法设置公开标签。
(☞ 423页 向CPU模块侧的公开条件)



2. 在将公开标签设置为“有效”的标签名的运动控制属性(2)*2中，设置刷新方向。
设置的标签名的数据类型为结构体的情况下，在数据类型(3)中设置的结构体中进行设置。

*2: 关于设置项目，请参阅运动控制属性的设置项目。
(☞ 424页 运动控制属性的设置项目)



3. 在步骤2. 中将数据类型设置为结构体的情况下，选择[导航窗口]⇒[标签]⇒[结构体]⇒[结构体名*3]，显示结构体的标签编辑器。设置CPU模块侧公开的标签名的公开标签及运动控制属性。

*3: 显示设置的结构体名。

向CPU模块侧的公开条件

可向CPU模块侧公开的条件如下所示。

○：可以设置，×：不能设置，△：根据设置内容不能设置

变量类型	类型	数组选择	公开标签设置可否	备注
全局标签	基本数据类型	无	○	不能设置下述标签、分类。 ■标签 • 字符串型标签 • 定时器型标签 • 计数器型标签 • 超长计数器型标签 • 累计定时器型标签 • 超长累计定时器型标签 • 超长定时器型标签 ■分类 • VAR_GLOBAL_CONSTANT分类
		有	△*1*2	
	结构体	无	△*3	
	FB (包含MCFB)	有	△*1*2*4*5	
		无	×	
程序	—	—	×	
程序块的局部标签	—	—	×	
结构体	—	—	△*3*5	
MCFB结构体	—	—	△*6*7	

- *1 不能设置数组的每个元素的公开标签。
- *2 使用位型的数组的情况下，不能将其设置为“有效”。(结构体的情况下，不能仅将该成员设置为“有效”。)
- *3 对结构体内的成员使用字符串型的情况下，不能将该成员设置为“有效”。
- *4 可公开的结构体的分层最大为4层。
- *5 对结构体内的成员使用结构体的数组的情况下，不能将该成员设置为“有效”。
- *6 有可能在CPU模块侧PLCopen MCFB的程序内使用。
- *7 MCFB结构体中使用字符串型的情况下，不能设置该MCFB结构体本身。

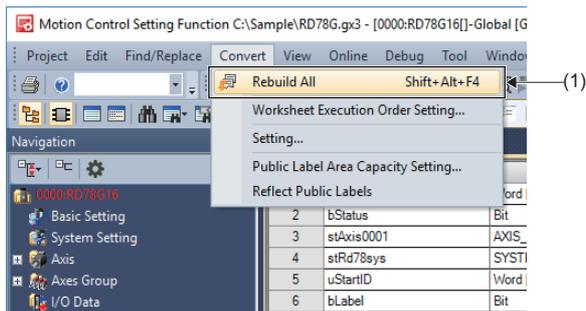
■运动控制属性的设置项目

运动控制属性的设置项目有关内容如下所示。

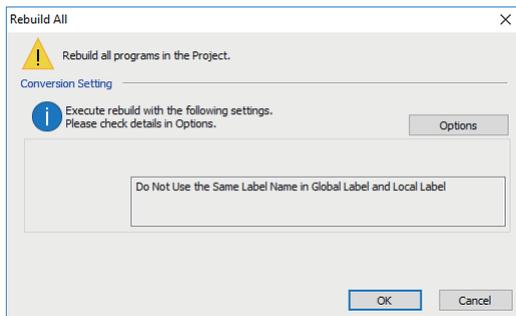
设置项目	内容
READ (运动⇒)	在CPU模块侧只能进行读取。 在CPU模块侧执行了写入的情况下，公开的标签的值不会被反映。(在运动模块侧，将被改写为运动模块中保持的值。)
WRITE (⇒运动)	在CPU模块侧可以进行写入、读取。 在运动模块侧执行了写入的情况下，公开的标签的值不会被反映。(在CPU模块侧，将被改写为CPU模块中保持的值。)
—	不能设置。 <ul style="list-style-type: none"> 有可能由工程工具自动设置。 数据类型为结构体的情况下，将按照结构体的各成员中设置的运动控制属性的设置。

标签转换

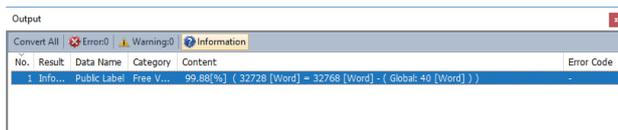
为了确定设置的标签，在运动控制设置功能画面上执行标签转换。



1. 选择菜单的[转换]⇒[全部转换](1)，并转换设置的标签数据。

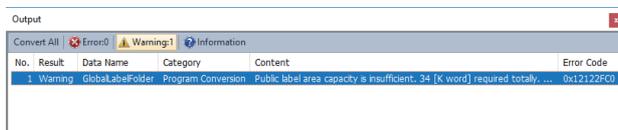


2. 显示“转换工程中的所有程序。”的信息时，单击[OK]按钮。



3. 所有转换正常完成时，在输出窗口的“Information”中显示公开标签区域的空余容量。

*1: 未显示输出窗口的情况下，选择菜单的[显示]⇒[停靠窗口]⇒[输出]以显示。
*2: 发生了转换出错的情况下，应根据出错内容修改设置内容，消除了所有出错后，再次执行转换操作。



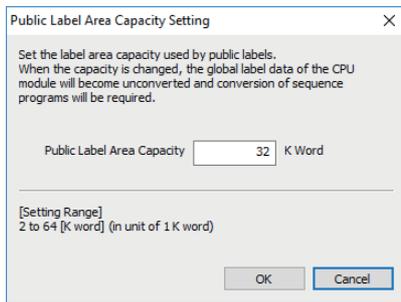
4. 设置的公开标签超出公开标签区域的容量的情况下，在输出窗口的“Warning”中显示报警信息。
应确认公开标签区域容量设置的标签存储器容量，并通过以下方法进行更改以确保在标签区域容量内。

- 通过公开标签区域容量设置增加公开标签区域的容量。
(☞ 425页 公开标签区域容量设置)
- 减少设置为“有效”的公开标签的使用量。

■公开标签区域容量设置

设置公开标签区域容量。

更改公开标签区域的容量的情况下，或者全部转换中设置的公开标签超出公开标签区域的容量的情况下，在公开标签区域容量设置中更改容量。



1. 选择菜单的[转换]⇒[公开标签区域容量设置]，显示“公开标签区域容量设置”画面。设置公开标签区域容量。
2. 设置完成后，单击[OK]按钮。

*: 在输出窗口中显示报警信息的情况下，应确认信息的内容并设置标签容量，使其在范围内。

要点

- 如果在公开标签区域容量设置中更改公开标签区域容量，标签数据将处于未转换状态，并且需要执行“转换”。
- 在转换操作时，显示了公开标签区域容量超出64k字的报警信息的情况下，应减少要公开的标签数。

显示内容

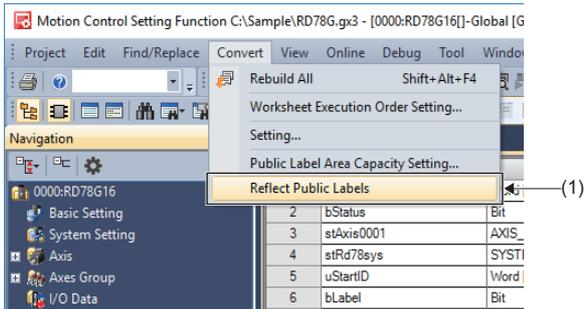
- 公开标签区域容量设置

项目	设置范围	初始值
公开标签区域容量	2~64k字	32k字

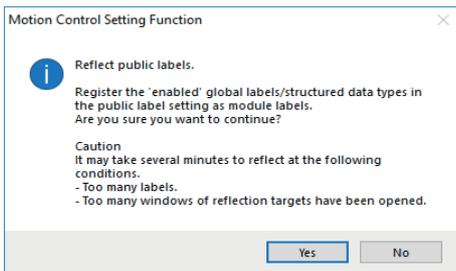
公开标签的反映

将创建的公开标签信息在运动控制设置功能画面上反映到CPU模块侧的工程中。

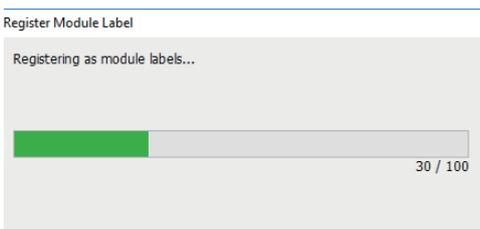
公开标签的反映完成时，将其作为模块标签自动登录到CPU模块侧。



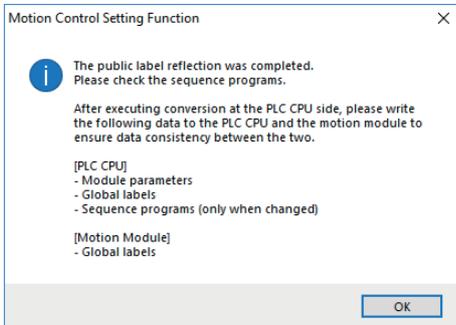
1. 选择菜单的[转换]⇒[公开标签的反映] (1)。



2. 显示“执行公开标签的反映。”的信息时，单击[是]按钮。



3. 执行公开标签的反映。
将公开标签设置为“有效”的标签作为模块标签登录到CPU模块侧。



4. 公开标签的反映完成时，将显示“公开标签的反映完成。”的信息。
单击[OK]按钮。

要点

进行公开标签的反映时，将作为模块标签登录到CPU模块侧的工程中，因此CPU模块侧的工程将处于未转换状态。需要在CPU模块侧的工程中执行转换。

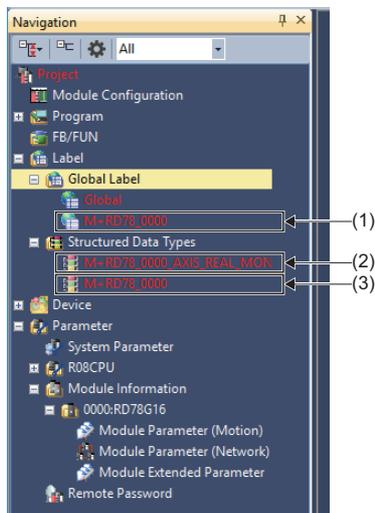
■公开标签反映完成时的CPU模块侧工程

公开标签的反映完成的CPU模块侧的工程中登录的数据如下所示。

• 工程树状图

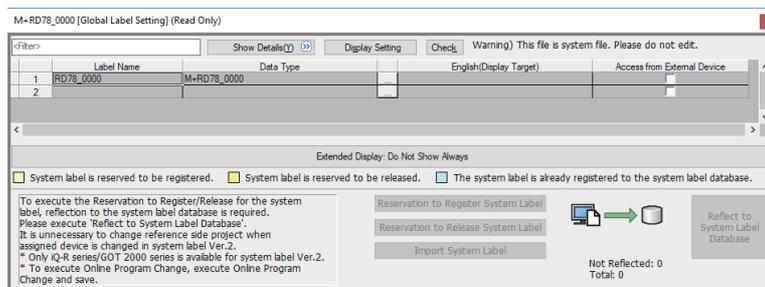
通过运动控制设置功能设置为公开的全局标签及结构体的标签数据被添加到CPU模块侧的全局标签及结构体的工程树状图中。

<CPU模块侧工程树状图>



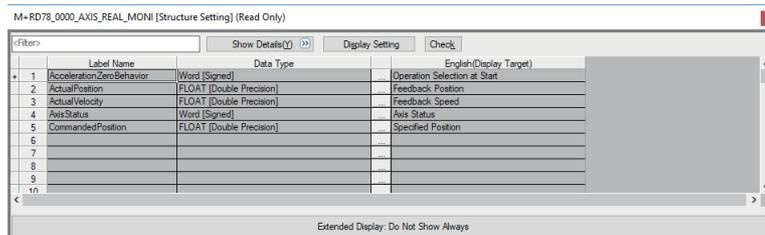
1. 全局标签 (1)

- 1个数据被添加到全局标签的工程树状图中。(对各运动模块创建1个。)
- 全局标签列表的数据名将为“M+RD78_****”^{*1}。
- 添加的全局标签列表中登录1个“RD78_****”^{*1}的标签。
- 登录的标签的数据类型将为“M+RD78_****”^{*1}的结构体型(3)。
- 添加的全局标签列表将为只读。无法进行编辑。



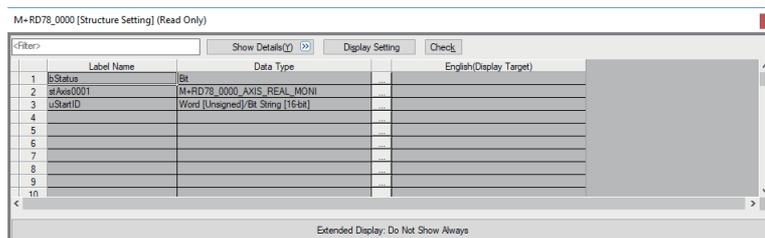
2. 公开的运动模块的结构体 (2)

- 通过运动控制设置功能设置为公开的结构体的成员将作为结构体名“M+RD78_****_####”^{*1*2}的结构体被登录到CPU模块侧。
- 对于结构体的成员，仅在运动控制设置功能中将公开标签设置为“有效”的标签名被登录。
- 添加的结构体为只读。无法进行编辑。



3. 公开的运动模块的全局标签 (3)

- 通过运动控制设置功能设置为公开的标签数据将作为结构体名“M+RD78_****”^{*1}的1个结构体的成员被登录到CPU模块侧。
- 添加的结构体为只读。无法进行编辑。



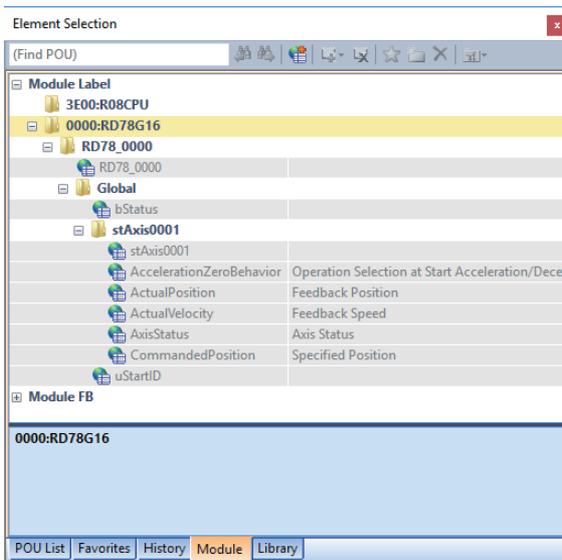
*1 ****=起始输入输出编号

*2 ####=运动控制设置中的结构体名

- 部件选择窗口

通过运动控制设置功能设置为公开的运动模块的标签在部件选择窗口中作为模块标签显示。

<CPU模块侧部件选择窗口>



- 在部件选择窗口的[模块]选项卡上登录为模块标签。
- 模块标签的数据名将为“****:RD78□n” *1*2*3。
- 在“****:RD78□n” *1*2*3的树状图中显示通过运动控制设置功能设置为公开的标签数据的一览。

*1 ****=起始输入输出编号

*2 □=运动模块的型号

- 设置RD78G时: G
- 设置RD78GH时: GH

*3 n=运动模块的控制轴数

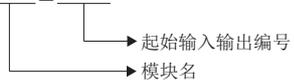
要点

在CPU模块侧使用公开的标签的情况下，在标签的起始输入“模块名+起始输入输出编号”后指定标签。

<例>

- 指定轴1(Axis0001)的标签的情况下

RD78_0000.Axis0001.AxisRef

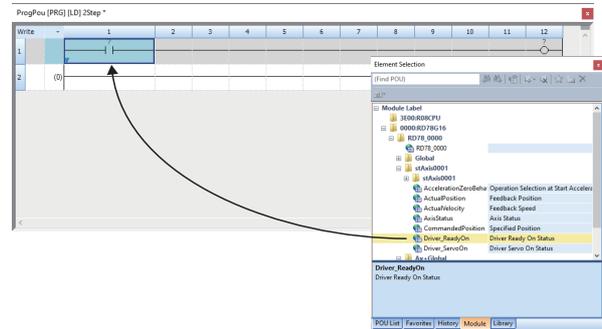


顺控程序的创建

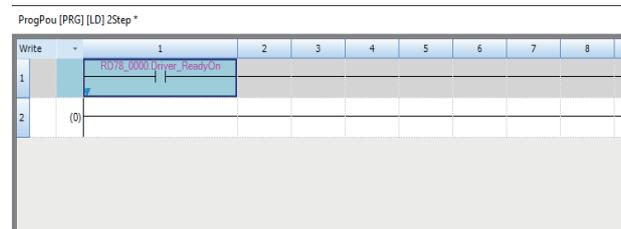
使用公开的运动模块的标签、运动控制FB，创建顺控程序。

■公开标签的插入

将CPU模块侧登录的公开标签通过拖放放置到顺控程序上并创建程序。



1. 从部件选择窗口选择要插入的标签，将其拖放到要放置的位置。

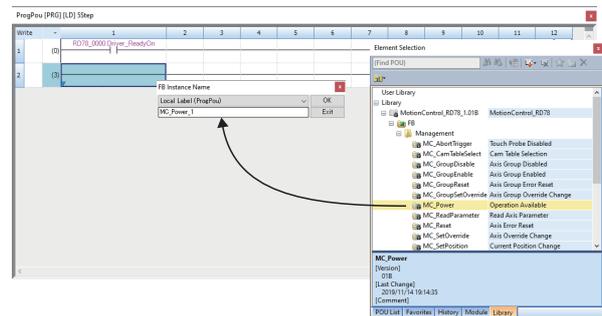


2. 标签将被插入到程序的工作窗口中。

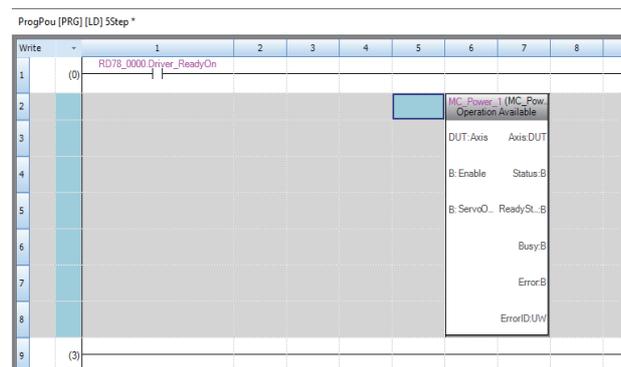
*: 使用的标签已登录为结构体的成员，因此添加结构体的实例名。

■运动控制FB的插入

将CPU模块侧登录的运动控制FB通过拖放放置到顺控程序上并创建程序。



1. 从部件选择窗口选择要插入的FB，将其拖放到要放置的位置时，将显示FB实例名输入画面。
2. 设置登录的实例名，单击[OK]按钮。



3. FB将被插入到程序的工作窗口中。

*: 从部件选择窗口插入FB时，以插入的FB的实例名在局部标签设置的一览中登录为标签名。

要点

运动控制FB的运动模块FB库需要在创建程序之前预先登录到工程工具的库中。

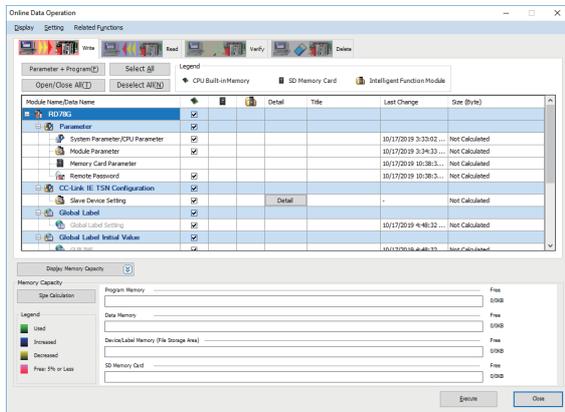
关于运动模块FB库的登录，请参阅下述章节。

☞ 431页 运动模块FB库的登录

创建的程序的写入

将在CPU模块侧及运动模块侧创建的程序、全局标签及模块参数写入到CPU模块及运动模块中。对各模块执行了写入后，应进行CPU模块的复位。

■至CPU模块的写入

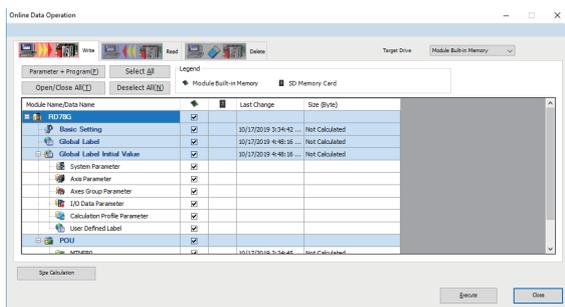


1. 选择菜单的[在线]⇒[至可编程控制器的写入]，显示“在线数据操作”画面。选择进行写入的数据，单击[执行]按钮。执行至可编程控制器CPU的写入。

要点

使用运动控制设置功能执行了“公开标签的反映”操作的情况下，必须选择“模块参数”、“全局标签”的数据。

■至运动模块的写入



1. 使用运动控制设置功能，选择菜单的[在线]⇒[至模块的写入]，显示“在线数据操作”画面。选择写入数据，单击[执行]按钮，并进行至运动模块的写入。

要点

使用运动控制设置功能执行了“公开标签的反映”操作的情况下，必须选择“全局标签”的数据。

⚠注意

- 公开了运动标签的情况下，必须将数据写入到CPU模块侧及运动模块侧。如果CPU模块侧与运动模块侧的标签数据不一致，则可能会发生意外动作。

运动模块FB库的登录

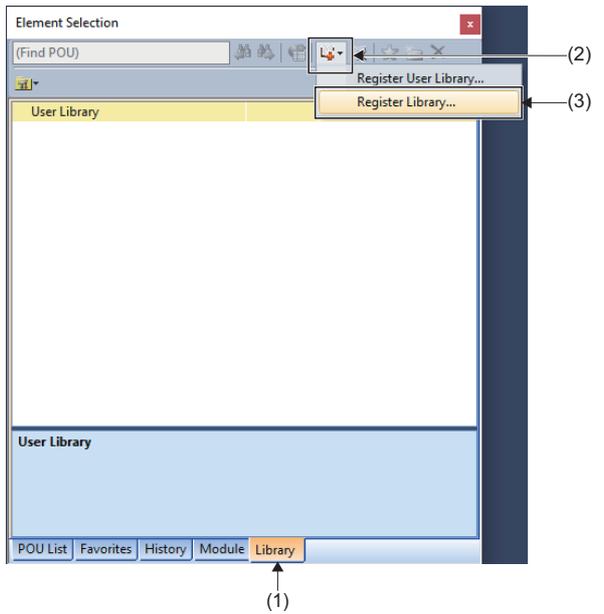
在CPU模块侧使用运动控制FB创建顺控程序的情况下，需要在工程工具的库中登录运动模块FB库。
以下对在工程工具中登录运动模块FB库的步骤进行说明。

要点 

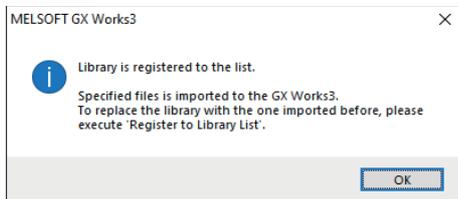
关于运动模块FB库，请向当地三菱电机代理店咨询。

运动模块FB库的登录步骤

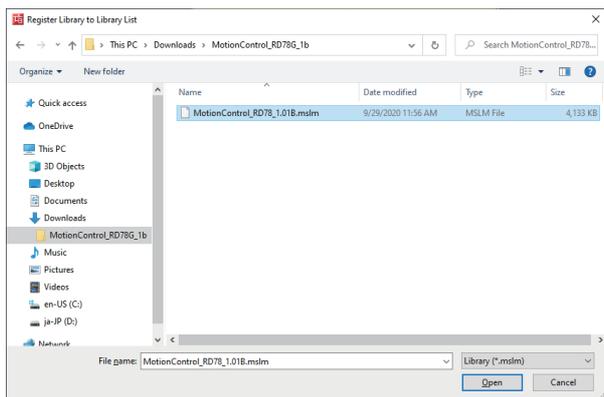
在工程工具的库中登录运动模块FB库。



1. 选择部件选择窗口的[库]选项卡(1)。
2. 单击[登录到库一览中](2)⇒[登录库](3)。

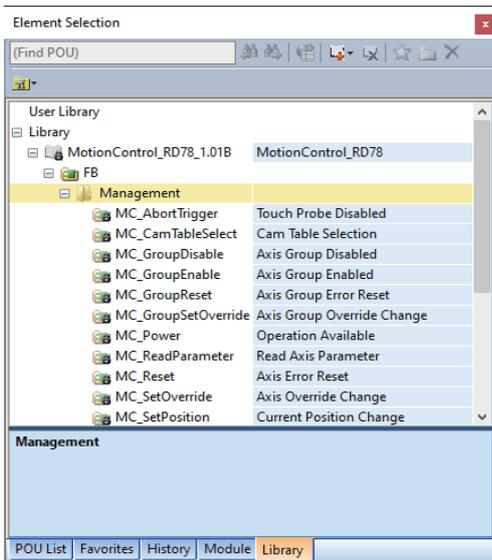


3. 显示“将库登录到一览中。”的信息时，单击[OK]按钮。



4. 显示“将库登录到库一览中”画面。
选择登录的运动模块FB库的文件
(MotionControl_RD78_****.mslm)，单击[打开]。

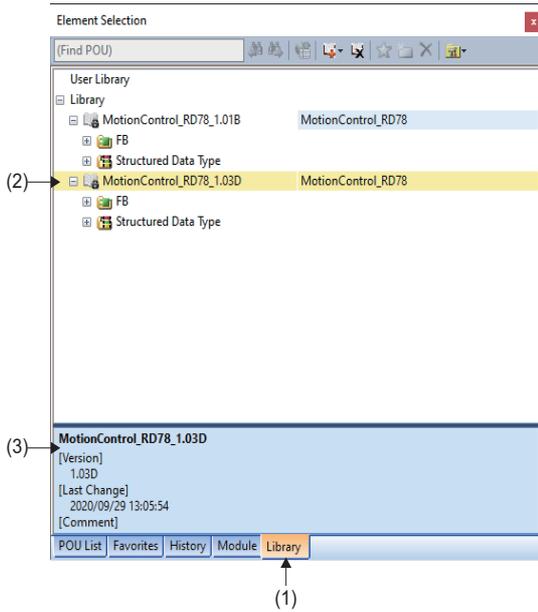
*1: ****=运动模块FB库的版本



5. 显示在部件选择窗口中登录的运动模块FB库的一览。

运动模块FB库的版本升级

登录最新版本的运动模块FB库，进行库部件的更新。



1. 选择部件选择窗口的[库]选项卡(1)。

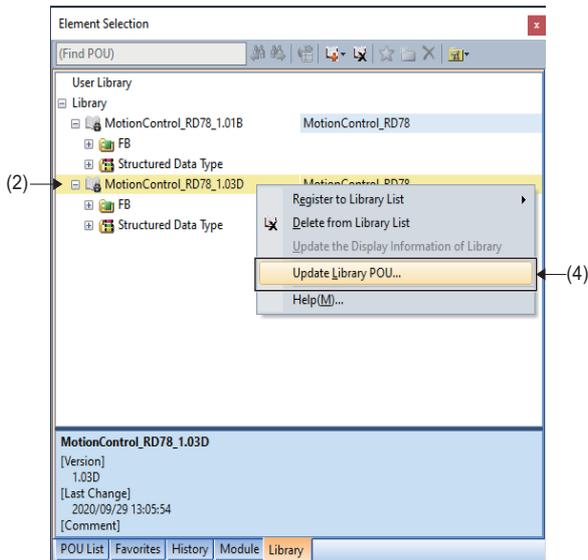
2. 登录最新版本的运动模块FB库。

*: 关于运动模块FB库的登录步骤，请参阅下述章节。

☞ 431页 运动模块FB库的登录步骤

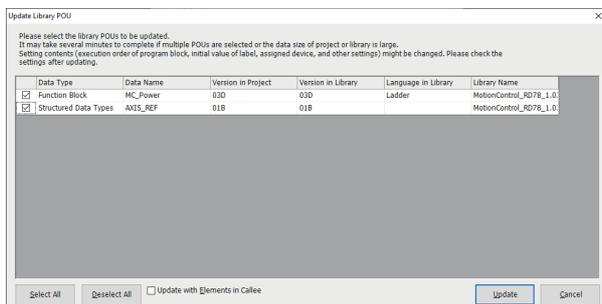
*: 如果选择登录的库(MotionControl_RD78_*****1) (2)，由于在部件选择窗口的下面(3)显示库的版本，因此可以确认当前的版本。

*1: ****=运动模块FB库的版本



3. 右键单击要更新的库(MotionControl_RD78_*****1) (2)，选择[库部件的更新] (4)。

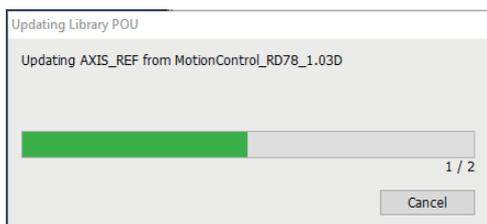
*1: ****=运动模块FB库的版本



4. 显示“库部件的更新”画面。

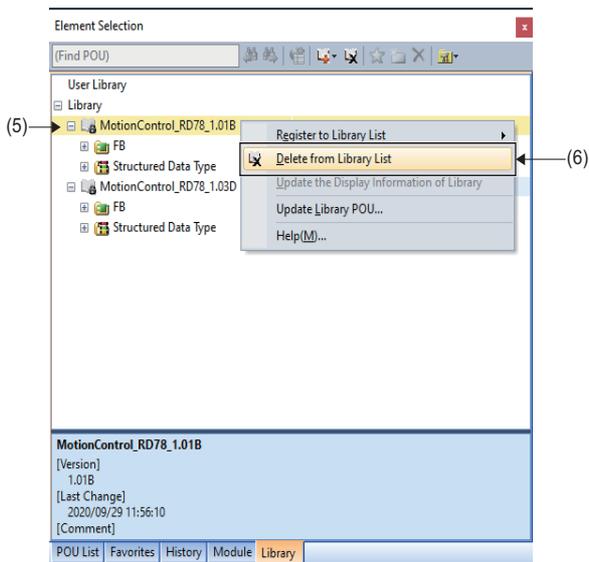
显示工程中使用的FB、结构体的一览。

5. 勾选需要更新的FB、结构体。单击[更新]按钮。



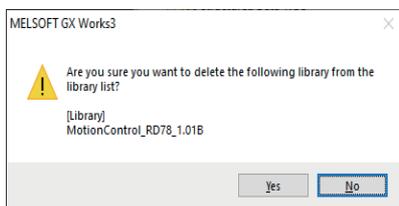
6. 执行库的更新。

更新完成时，关闭“库部件更新中”画面，更新结束。

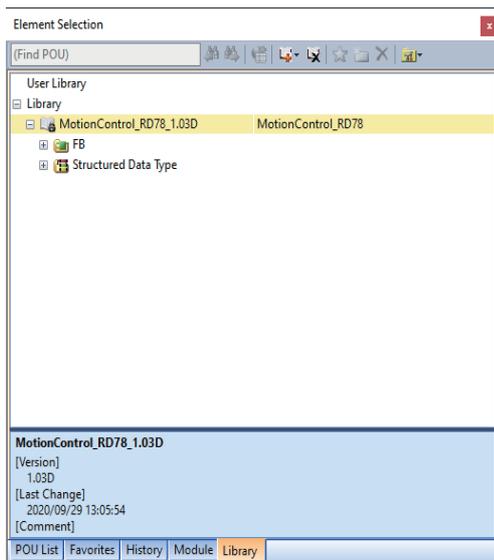


7. 删除旧版本的运动模块FB库的情况下，右键单击要删除的库 (MotionControl_RD78_****^{*1}) (5)，选择[从库一览中删除] (6)。

*1: ****=运动模块FB库的版本



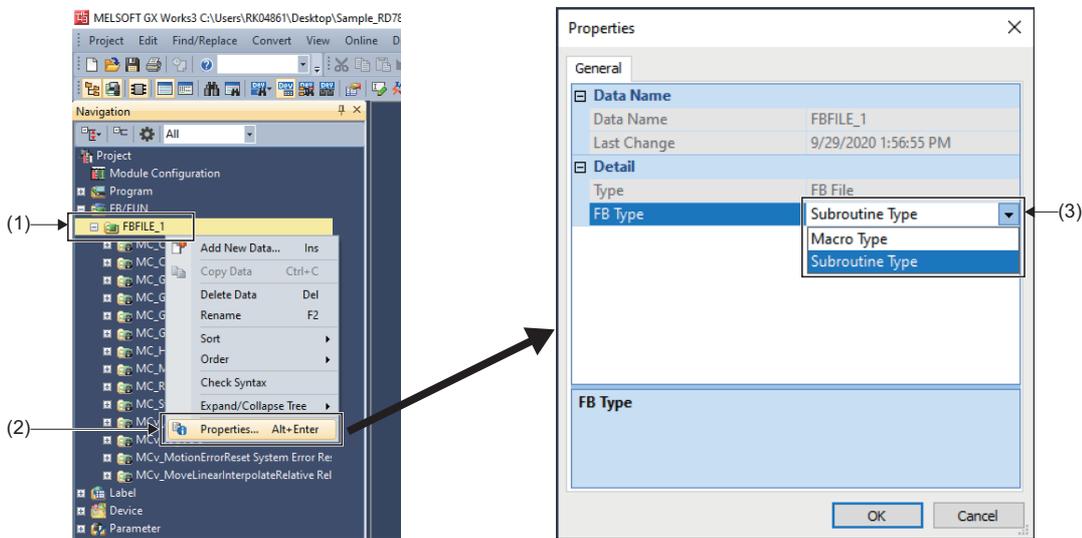
8. 显示“确定要从一览中删除以下库吗？”的信息时，单击[是]按钮。



9. 从库的一览中删除旧版本的运动模块FB库。

■ 注意事项

- 如果进行库的更新，CPU模块侧的工程将处于未转换状态。需要在CPU模块侧的工程中执行转换。
- 进行库的更新的情况下，如果FB的类型为“宏型”，则无法进行库的更新。应更改为“子程序型”后再进行库的更新。
更改FB的类型时，右键单击导航窗口的[FB/FUN]⇒[FBFILE_1] (1)，选择[属性] (2)时，将显示属性画面。从属性画面将FB的类型 (3)更改为“子程序型”。



使用运动模块的缓冲存储器的数据发送接收

CPU模块与运动模块之间的数据发送接收

CPU模块与运动模块之间，可以使用运动模块的缓冲存储器进行数据的发送接收。
使用的软件元件如下所示。

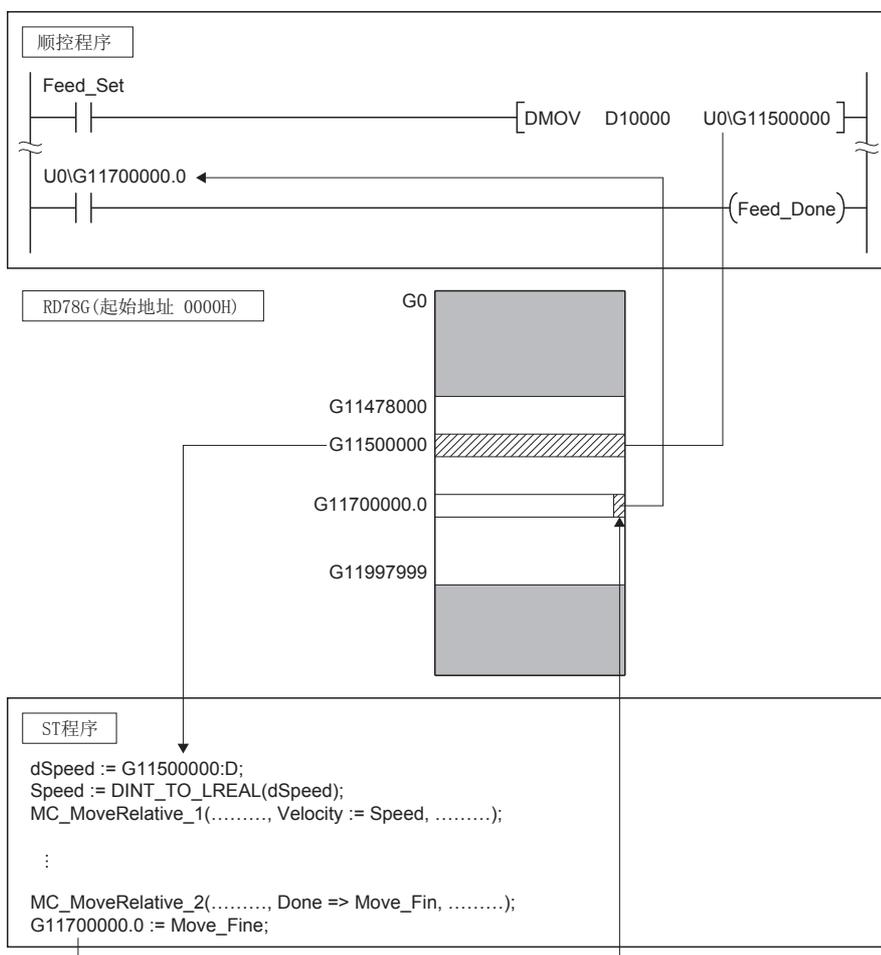
模块	使用的软件元件
CPU模块	Un\Gn (模块访问软件元件)
运动模块	Gn (缓冲存储器)

数据发送接收中可使用的范围如下所示。切勿使用超出设置范围的缓冲存储器。

设置范围
G11478000~G11997999 (520000字)

例

使用运动模块的缓冲存储器的数据发送接收



使用标签的程序

通过向运动模块的缓冲存储器分配全局标签可以进行标签编程，但是不可以在CPU模块与运动模块之间相互参照标签。

要点

将数据从CPU模块传递到运动模块的情况下，不可以通过ST指令将双精度实数(LREAL型)传送到缓冲存储器中。
通过使用专用指令，可以进行双精度实数型标签的读取/写入。

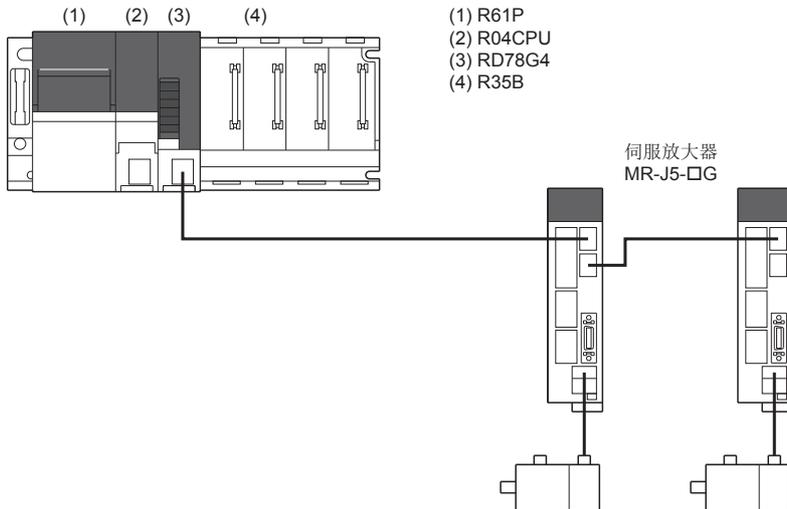
附录

附1 FBD/LD的样本程序

在CPU模块侧通过FBD/LD语言创建的样本程序如下所示。

系统配置

程序示例中使用的系统配置如下所示。



GX Works3中的设置

在GX Works3中进行的以下设置如下所示。

- 系统参数
- CPU参数
- 模块参数(网络)
- 程序

系统参数

系统参数中设置的项目如下所示。

■基板/电源/扩展电缆设置

基板No.	基板	插槽数	电源模块
主基板	R35B	5	R61P

■I/O分配设置

插槽	模块型号	模块状态设置	点数	起始XY	异常检测时的CPU模块动作设置
基本	CPU	R04CPU(本机)	—	3E00	—
	0(0-0)	RD78G4	无设置	32点	0000

CPU参数

CPU参数中设置的项目如下所示。

■存储器/软元件设置

项目	设置
链接直接软元件设置	扩展模式(iQ-R系列模式)

■程序设置

执行顺序	程序名	执行类型		刷新组设置
		类型	详细设置信息	
1	MAIN	扫描	—	(不设置)

模块参数(网络)

模块参数(网络)中设置的项目如下所示。

■基本设置

- 网络配置设置

型号	站号	站类型	运动管理站	RWr设置 点数	RWw设置 点数	参数自动设置	IP地址
本站	0	主站	—	—	—	—	192.168.3.253
MR-J5-G	1	远程站	有勾选	24	20	有勾选	192.168.3.1
MR-J5-G	2	远程站	有勾选	24	20	有勾选	192.168.3.2

软元件

程序中使用的软元件如下所示。

软元件	注释
X0	准备就绪
X1	同步用标志
Y0	可编程控制器就绪
SM400	常开

全局标签

程序中使用的全局标签的设置如下所示。

<Filter>		Easy Display <<	Display Setting	Check	Assign (Device/Label)	Initial Value	Constant	English(Display Target)
1	Label Name	Data Type	Class					
1	G_bSRVOff	Bit	VAR_GLOBAL	▼				Servo OFF
2	G_bJogF1	Bit	VAR_GLOBAL	▼				JOG Positive rotation command Axis0001
3	G_bJogR1	Bit	VAR_GLOBAL	▼				JOG Reverse rotation command Axis0001
4	G_bJogF2	Bit	VAR_GLOBAL	▼				JOG Positive rotation command Axis0002
5	G_bJogR2	Bit	VAR_GLOBAL	▼				JOG Reverse rotation command Axis0002
6	G_bHoming1CMD	Bit	VAR_GLOBAL	▼				Homing command Axis0001
7	G_bHoming2CMD	Bit	VAR_GLOBAL	▼				Homing command Axis0002
8	G_bHoming3CMD	Bit	VAR_GLOBAL	▼				Homing command VirtualAxis0001
9	G_bPosCMD	Bit	VAR_GLOBAL	▼				Single axis positioning start
10	G_bContPosCMD	Bit	VAR_GLOBAL	▼				Single axis continuous positioning start
11	G_bInterpolationCMD	Bit	VAR_GLOBAL	▼				2-axis linear interpolation control start
12	G_bSyncCMD	Bit	VAR_GLOBAL	▼				Synchronous control start
13	G_bErrorReset	Bit	VAR_GLOBAL	▼				Error reset
14	G_bSysErrorReset	Bit	VAR_GLOBAL	▼				System error reset
15	G_leJogVelocity	FLOAT [Double Precision]	VAR_GLOBAL	▼				JOG Velocity
16	G_bHoming1Req	Bit	VAR_GLOBAL	▼				Homing start request Axis0001
17	G_bHoming2Req	Bit	VAR_GLOBAL	▼				Homing start request Axis0002
18	G_bHoming3Req	Bit	VAR_GLOBAL	▼				Homing start request VirtualAxis0001
19	G_bPosReq	Bit	VAR_GLOBAL	▼				Single axis positioning start request
20	G_bContPosReq	Bit	VAR_GLOBAL	▼				Single axis continuous positioning start request
21	G_bInterpolationReq	Bit	VAR_GLOBAL	▼				2-axis linear interpolation control start request
22	G_bSyncReq	Bit	VAR_GLOBAL	▼				Synchronous control start request
23	G_bJog1Busy	Bit	VAR_GLOBAL	▼				JOG operation in progress Axis0001
24	G_bJog2Busy	Bit	VAR_GLOBAL	▼				JOG operation in progress Axis0002

程序

通过FBD/LD语言创建的程序如下所示。

程序文件名	程序名称	处理内容	参阅
MAIN	ServoON_Jog	<ul style="list-style-type: none"> 进行所有轴的伺服ON。 进行轴1 (Axis0001) 与轴2 (Axis0002) 的JOG运行。 	☞ 439页 ServoON_Jog
	Positioning	<ul style="list-style-type: none"> 根据轴1 (Axis0001) 的相对值进行单轴定位控制。 	☞ 441页 Positioning
	ContinuousPositioning	<ul style="list-style-type: none"> 根据轴1 (Axis0001) 与轴2 (Axis0002) 的相对值进行单轴连续定位控制。 	☞ 442页 ContinuousPositioning
	LinearInterpolation	<ul style="list-style-type: none"> 对轴1 (Axis0001) 与轴2 (Axis0002) 的轴组1 (AxesGroup001) 进行2轴直线插补控制。 	☞ 444页 LinearInterpolation
	Homing	<ul style="list-style-type: none"> 进行轴1 (Axis0001)、轴2 (Axis0002)、轴301 (VirtualAxis0001) 的原点复位。 	☞ 447页 Homing
	ErrorReset	<ul style="list-style-type: none"> 进行轴1 (Axis0001)、轴2 (Axis0002)、轴301 (VirtualAxis0001)、轴401 (LinkAxis0001)、轴402 (LinkAxis0002)、轴组1 (AxesGroup001)、运动系统的出错复位。 	☞ 448页 ErrorReset
	Synchronous	<ul style="list-style-type: none"> 进行同步控制。 	☞ 449页 Synchronous

■ ServoON_Jog

- 局部标签

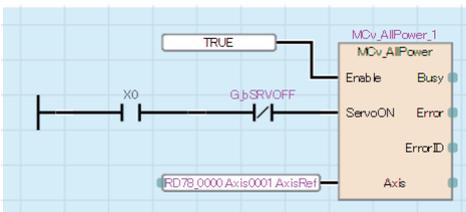
Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)
1 MCv_AllPower_1	MCv_AllPower	VAR			All axes servo ON FB
2 MCv_Jog_1	MCv_Jog	VAR			JOG operation FB Axis0001
3 MCv_Jog_2	MCv_Jog	VAR			JOG operation FB Axis0002
4 leJogAcceleration	FLOAT [Double Precision]	VAR			JOG acceleration
5 leJogDeceleration	FLOAT [Double Precision]	VAR			JOG deceleration
6 leJogJerk	FLOAT [Double Precision]	VAR			JOG jerk

- 程序本体

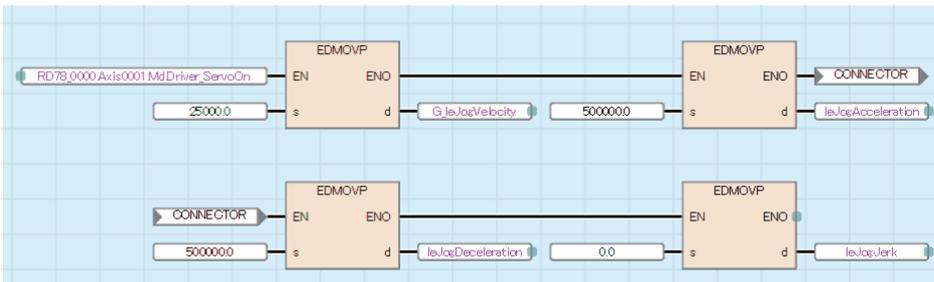
- 可编程控制器就绪ON



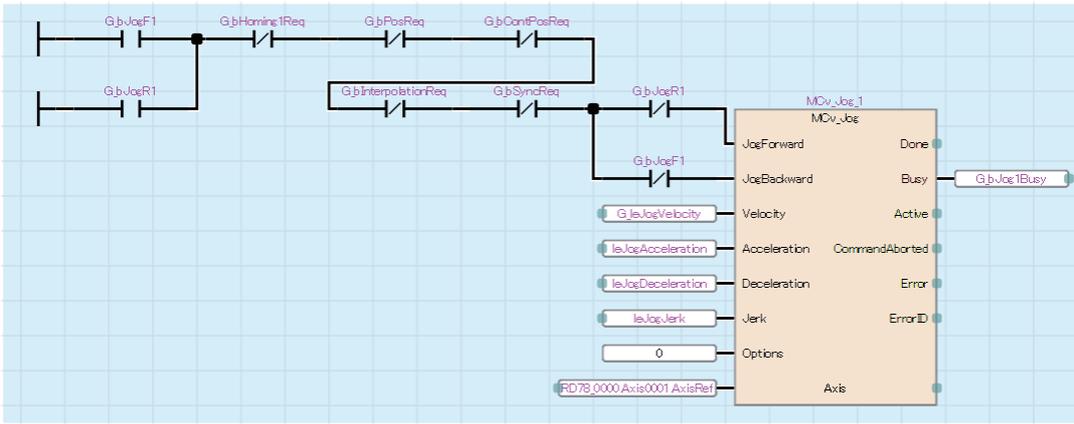
- 所有轴伺服ON



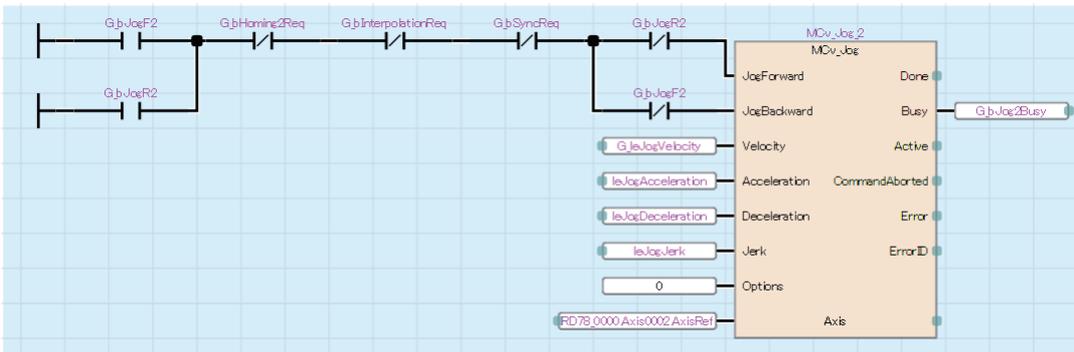
- JOG运行用数据设置



• JOG Axis0001



• JOG Axis0002



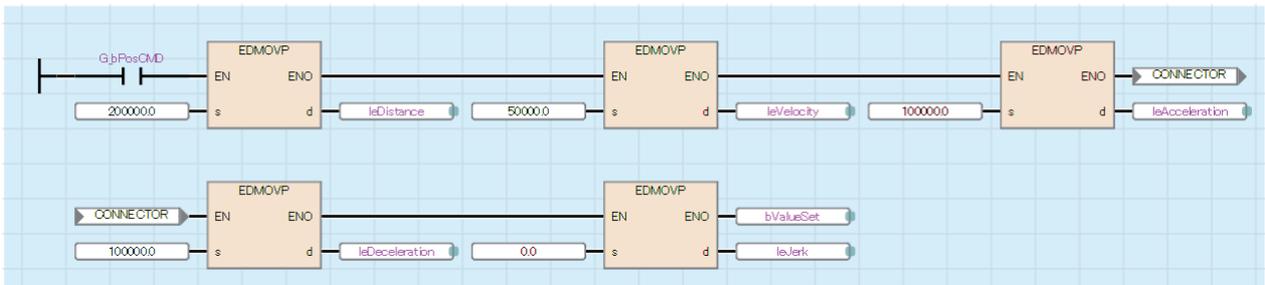
■ Positioning

• 局部标签

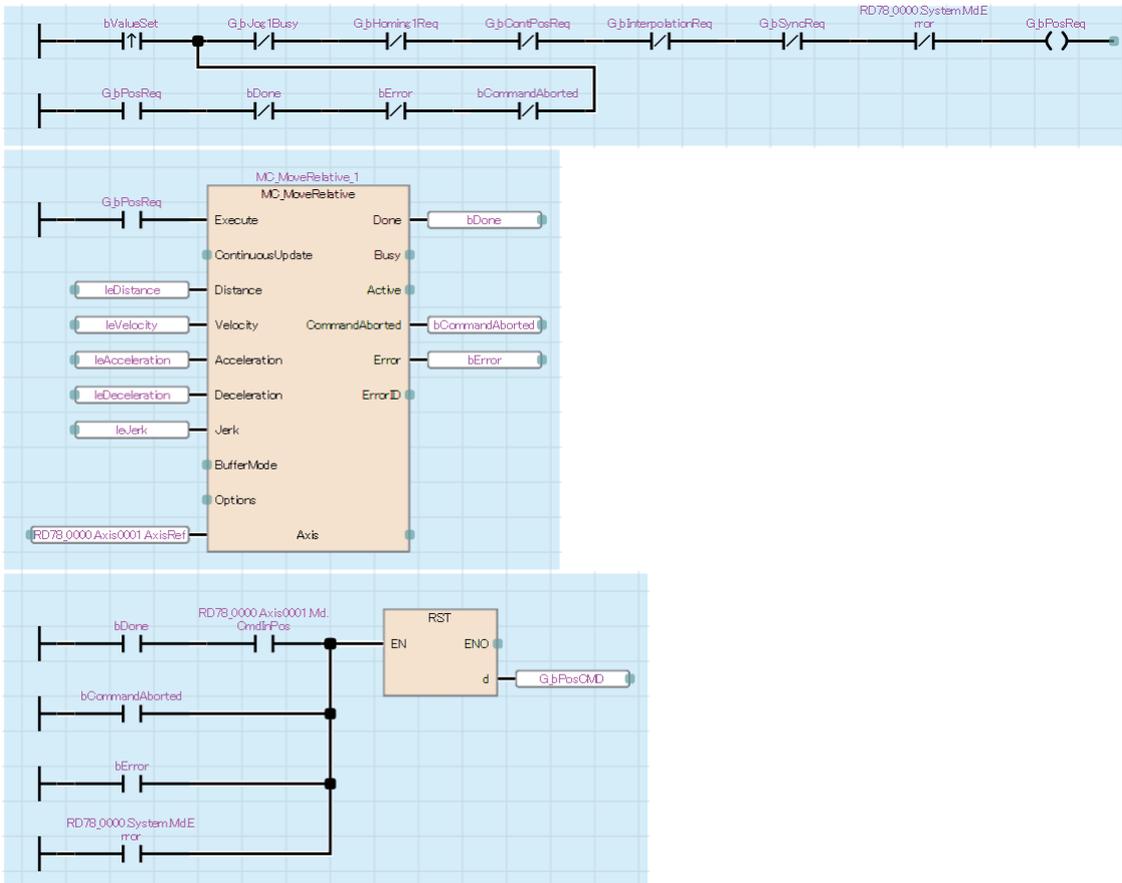
Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)
1 MC_MoveRelative_1	MC_MoveRelative	VAR			Relative value positioning FB
2 leDistance	FLOAT [Double Precision]	VAR			Distance
3 leVelocity	FLOAT [Double Precision]	VAR			Velocity
4 leAcceleration	FLOAT [Double Precision]	VAR			Acceleration
5 leDeceleration	FLOAT [Double Precision]	VAR			Deceleration
6 leJerk	FLOAT [Double Precision]	VAR			Jerk
7 bDone	Bit	VAR			Relative value positioning FB Done output
8 bError	Bit	VAR			Relative value positioning FB Error output
9 bValueSet	Bit	VAR			Variable set complete
10 bCommandAborted	Bit	VAR			Relative value positioning FB CommandAborted output

• 程序本体

- 单轴定位用数据设置



• 相对定位



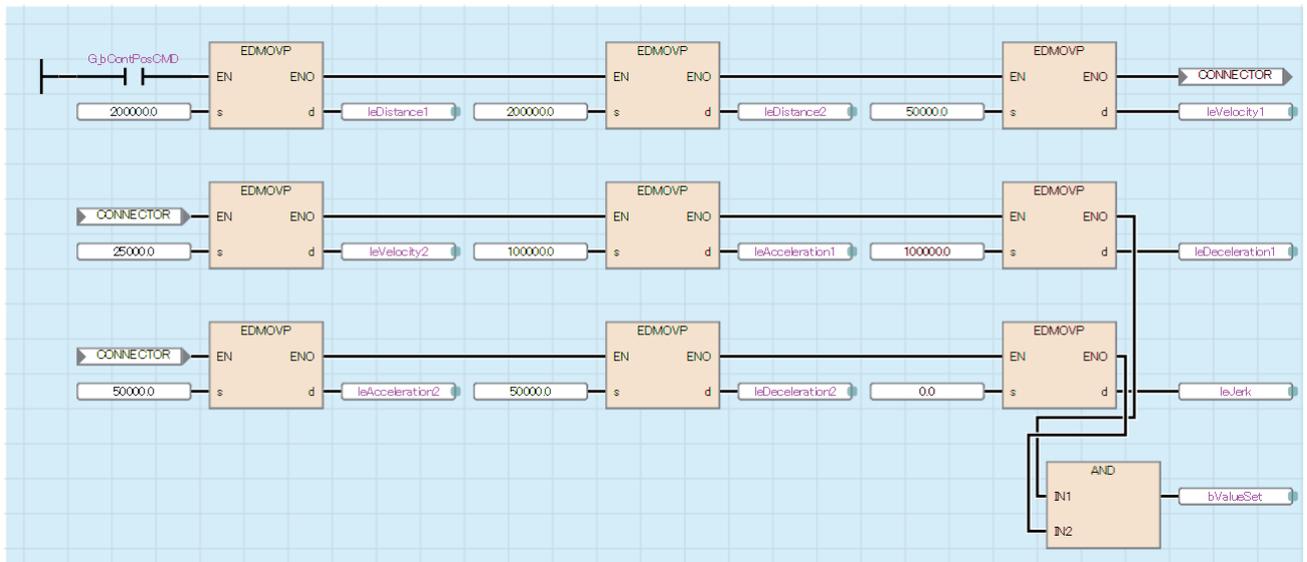
■ Continuous Positioning

- 局部标签

<Filter>	Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)
1	leDistance1	FLOAT [Double Precision]	VAR			Distance1
2	leVelocity1	FLOAT [Double Precision]	VAR			Velocity1
3	leDistance2	FLOAT [Double Precision]	VAR			Distance2
4	leVelocity2	FLOAT [Double Precision]	VAR			Velocity2
5	leAcceleration1	FLOAT [Double Precision]	VAR			Acceleration1
6	leDeceleration1	FLOAT [Double Precision]	VAR			Deceleration1
7	leAcceleration2	FLOAT [Double Precision]	VAR			Acceleration2
8	leDeceleration2	FLOAT [Double Precision]	VAR			Deceleration2
9	leJerk	FLOAT [Double Precision]	VAR			Jerk
10	MC_MoveRelative_1	MC_MoveRelative	VAR			Relative value positioning FB1
11	MC_MoveRelative_2	MC_MoveRelative	VAR			Relative value positioning FB2
12	bError	Bit	VAR			Relative value positioning FB Error output
13	TON_1	TON	VAR			On-delay timer FB
14	bDwell_out	Bit	VAR			Timer output
15	bDwell_in	Bit	VAR			Timer input
16	bValueSet	Bit	VAR			Variable set complete
17	bCommandAborted	Bit	VAR			Relative value positioning FB CommandAborted output

- 程序本体

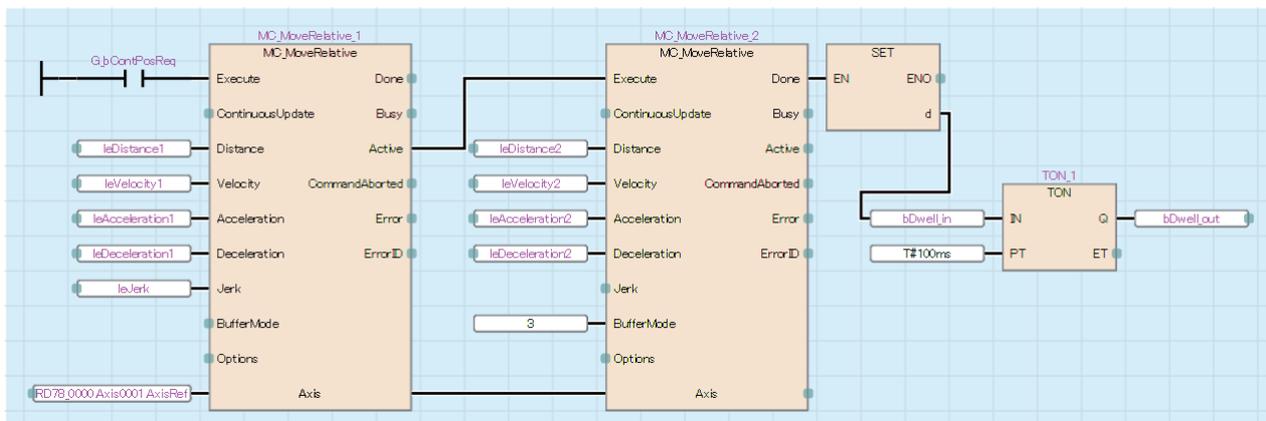
- 单轴连续定位用数据设置



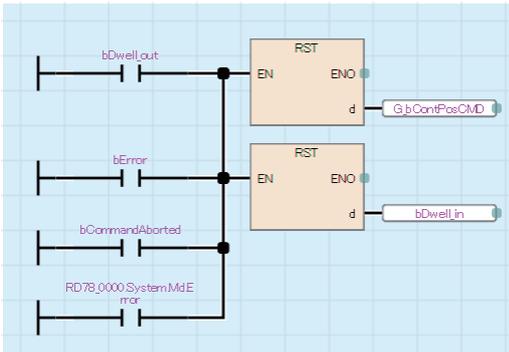
- 单轴连续定位+缓冲模式+停留



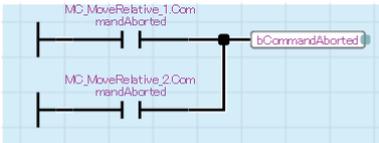
- 相对值定位1/相对值定位2/停留



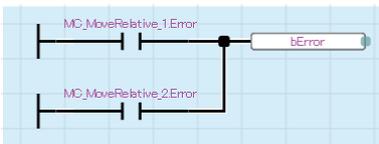
- 定位完成时或出错发生时，复位定位启动请求



- 各FB执行中断输出



- 各FB出错输出



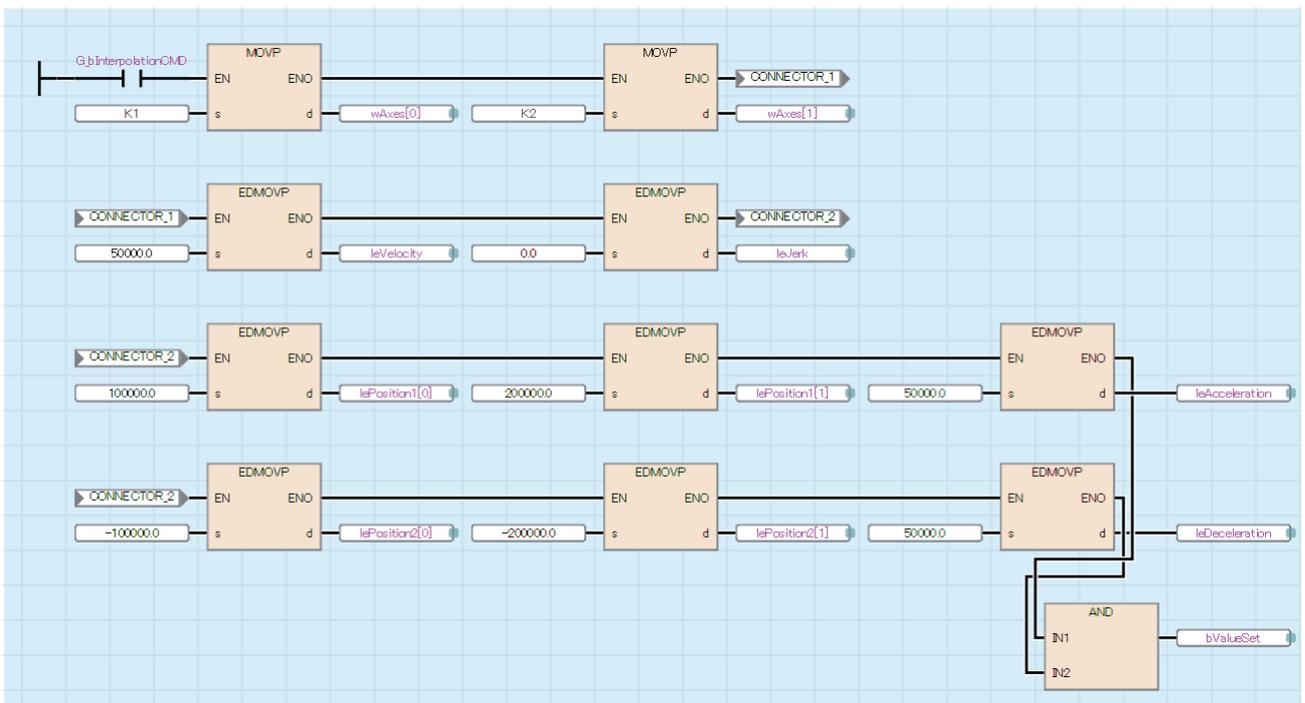
■ Linear Interpolation

• 局部标签

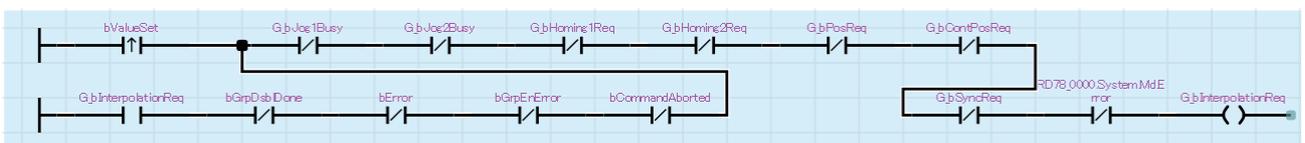
Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)
1 wAxes	Word [Signed](0..15)	VAR			Interpolation axis
2 lePosition1	FLOAT [Double Precision](0..15)	VAR			Position data 1
3 lePosition2	FLOAT [Double Precision](0..15)	VAR			Position data 2
4 leVelocity	FLOAT [Double Precision]	VAR			Velocity
5 leAcceleration	FLOAT [Double Precision]	VAR			Acceleration
6 leDeceleration	FLOAT [Double Precision]	VAR			Deceleration
7 leJerk	FLOAT [Double Precision]	VAR			Jerk
8 MC_GroupEnable_1	MC_GroupEnable	VAR			Axes group enable FB
9 MCV_MoveLinearInterpolateRelative_1	MCV_MoveLinearInterpolateRelative	VAR			Relative value linear interpolation control FB1
10 MCV_MoveLinearInterpolateRelative_2	MCV_MoveLinearInterpolateRelative	VAR			Relative value linear interpolation control FB2
11 MC_GroupDisable_1	MC_GroupDisable	VAR			Axes group disable FB
12 bDone2	Bit	VAR			Relative value linear interpolation control FB2 Done output
13 bGpEnError	Bit	VAR			Axes group enable FB Error output
14 bError	Bit	VAR			Relative value linear interpolation control FB Error output
15 bGpDsblDone	Bit	VAR			Axes group disable FB Done output
16 TON_1	TON	VAR			On-delay timer FB
17 bDwell_in	Bit	VAR			Timer input
18 bDwell_out	Bit	VAR			Timer output
19 bValueSet	Bit	VAR			Variable set complete
20 bCommandAborted	Bit	VAR			Relative value linear interpolation control FB CommandAborted output
21 bDone_Set	Bit	VAR			Done
22 bCommandAborted_Set	Bit	VAR			FB abortion of execution

• 程序本体

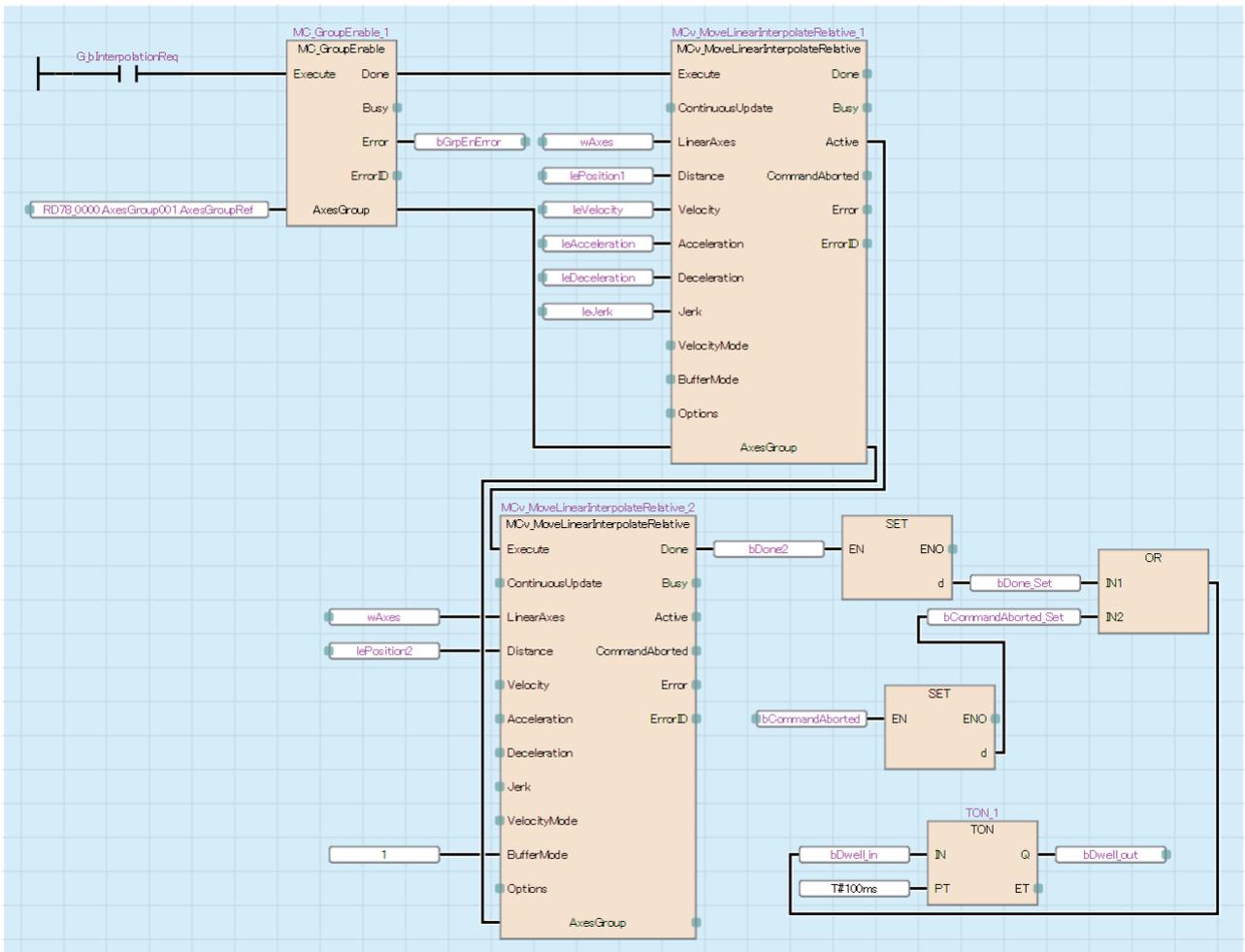
• 2轴直线插补控制用数据设置



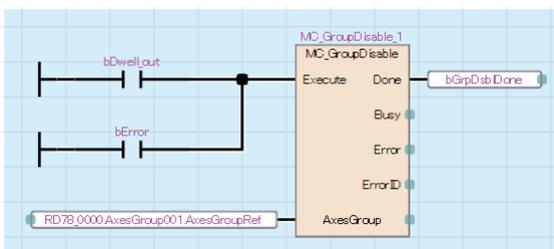
• 直线插补控制+缓冲模式+停留



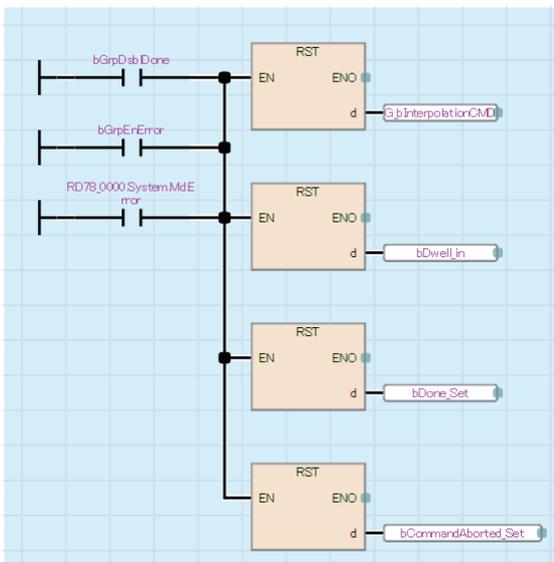
- 轴组启用/直线插补控制1/直线插补控制2/FB执行完成/FB执行中断/停留



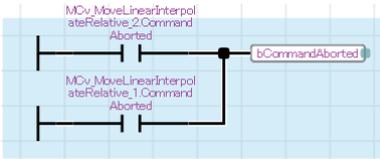
- 轴组禁用



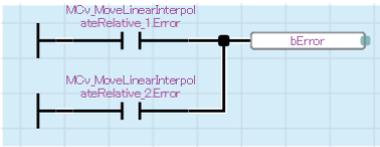
- 复位启动指令、定时器输入



• 各FB执行中断输出



• 各FB出错输出



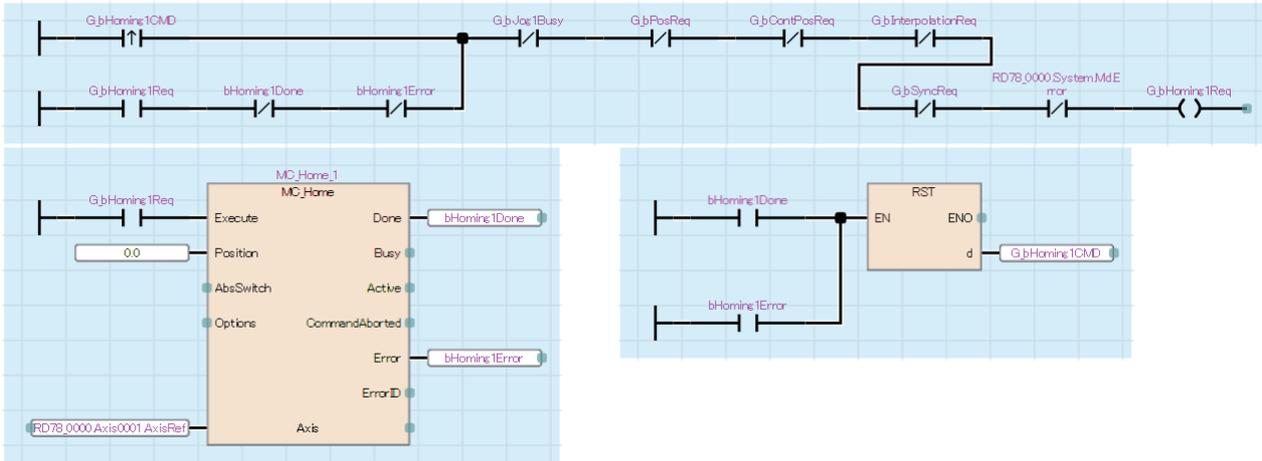
Homing

局部标签

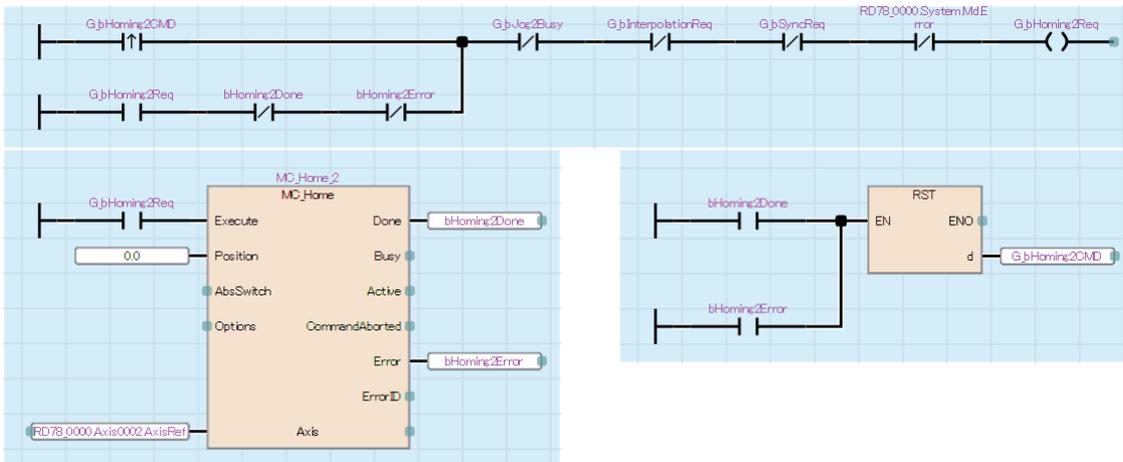
Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)
1 MC_Home_1	MC_Home	VAR			Homing FB1 Axis001
2 MC_Home_2	MC_Home	VAR			Homing FB2 Axis002
3 MC_Home_3	MC_Home	VAR			Homing FB3 VirtualAxis001
4 bHoming1Done	Bit	VAR			Homing FB1 Done output
5 bHoming1Error	Bit	VAR			Homing FB1 Error output
6 bHoming2Done	Bit	VAR			Homing FB2 Done output
7 bHoming2Error	Bit	VAR			Homing FB2 Error output
8 bHoming3Done	Bit	VAR			Homing FB3 Done output
9 bHoming3Error	Bit	VAR			Homing FB3 Error output

程序本体

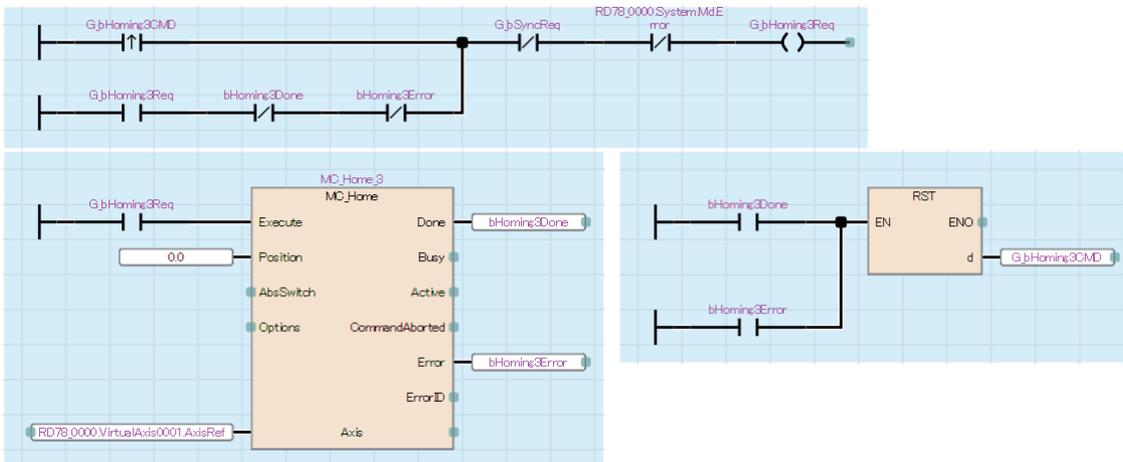
Axis0001 原点复位



Axis0002 原点复位



VirtualAxis0001 原点复位



附

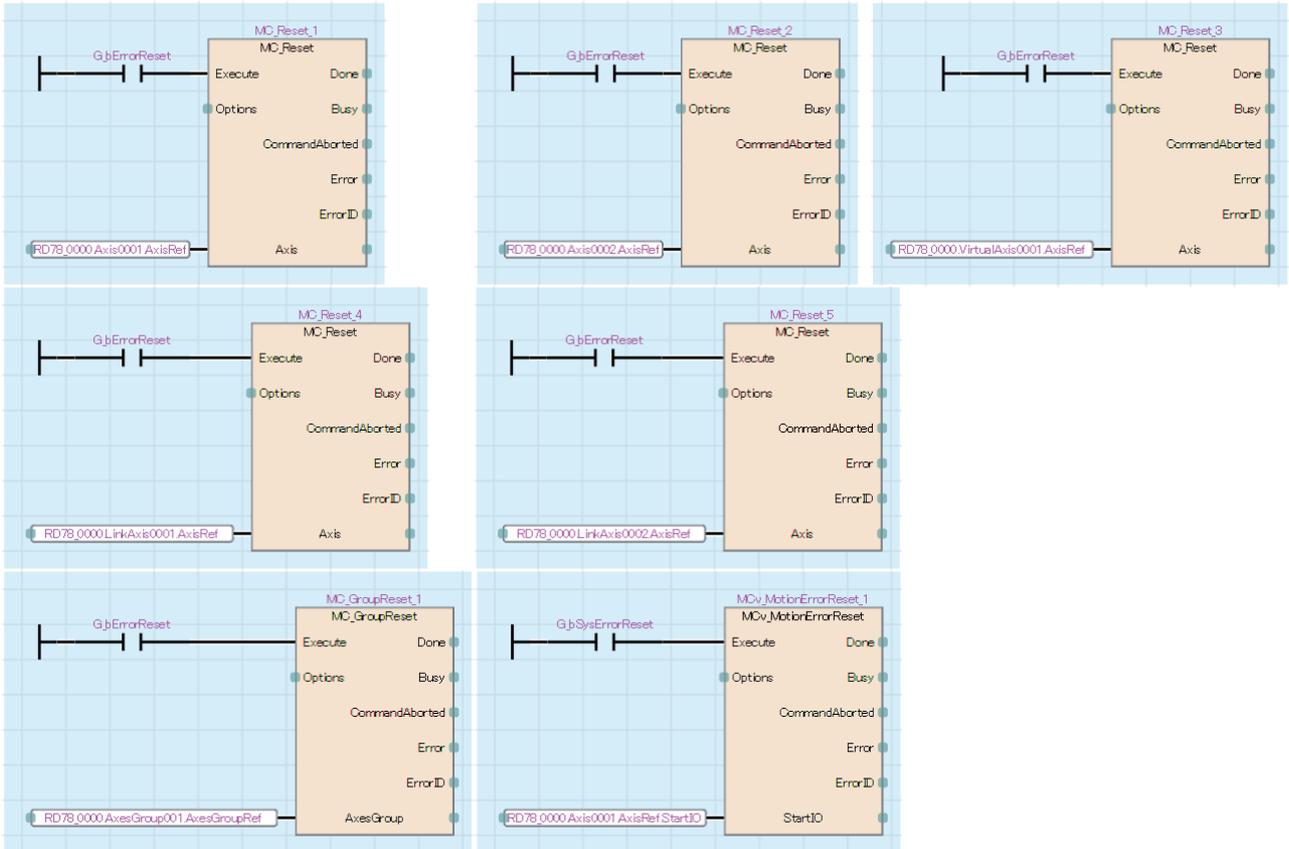
■ ErrorReset

- 局部标签

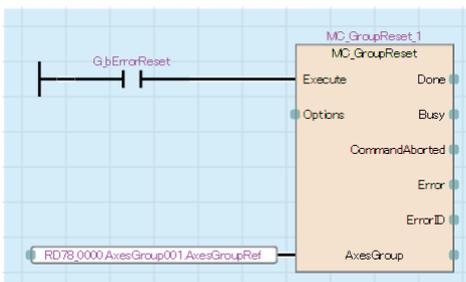
Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)
1 MC_Reset_1	MC_Reset	VAR			Axis error reset FB1 Axis0001
2 MC_Reset_2	MC_Reset	VAR			Axis error reset FB2 Axis0002
3 MC_Reset_3	MC_Reset	VAR			Axis error reset FB3 VirtualAxis0001
4 MC_Reset_4	MC_Reset	VAR			Axis error reset FB4 LinkAxis0001
5 MC_Reset_5	MC_Reset	VAR			Axis error reset FB5 LinkAxis0002
6 MC_GroupReset_1	MC_GroupReset	VAR			Axes group error reset FB
7 MCV_MotionErrorReset_1	MCV_MotionErrorReset	VAR			System error reset FB

- 程序本体

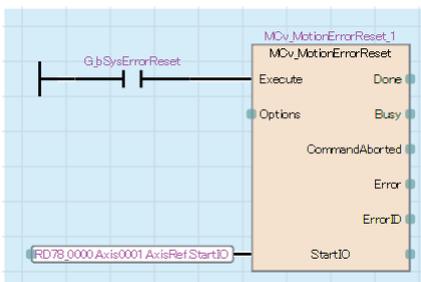
- 轴出错复位



- 轴组出错复位



- 系统出错复位



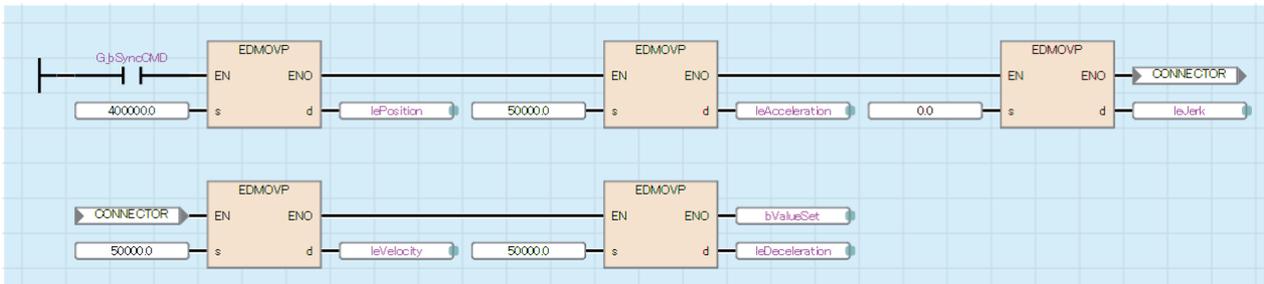
■ Synchronous

- 局部标签

	Label Name	Data Type	Class	Initial Value	Constant	English(Display Target)
1	leVelocity	FLOAT [Double Precision]	VAR			Velocity
2	leAcceleration	FLOAT [Double Precision]	VAR			Acceleration
3	leDeceleration	FLOAT [Double Precision]	VAR			Deceleration
4	leJerk	FLOAT [Double Precision]	VAR			Jerk
5	lePosition	FLOAT [Double Precision]	VAR			Distance
6	MC_GearIn_1	MC_GearIn	VAR			Gear operation FB
7	MC_CombineAxes_1	MC_CombineAxes	VAR			FB combining the motion of two master axes
8	MC_CamIn_1	MC_CamIn	VAR			Cam operation FB
9	CamID	MC_CAM_ID	VAR			Cam ID
10	bInSync	Bit	VAR			Cam operation FB inSync output
11	MC_MoveRelative_1	MC_MoveRelative	VAR			Relative value positioning FB
12	bError	Bit	VAR			Relative value positioning FB Error output
13	bDone1	Bit	VAR			Relative value positioning FB1 Done output
14	MC_MoveRelative_2	MC_MoveRelative	VAR			Relative value positioning FB
15	bDone2	Bit	VAR			Relative value positioning FB2 Done output
16	bStopDone	Bit	VAR			Axis Stop complete
17	bSyncMove	Bit	VAR			Relative value positioning start
18	bValueSet	Bit	VAR			Variable set complete
19	bCommandAborted	Bit	VAR			Relative value positioning FB CommandAborted output
20	MC_Stop_1	MC_Stop	VAR			Axis Stop FB1
21	MC_Stop_2	MC_Stop	VAR			Axis Stop FB2
22	MC_Stop_3	MC_Stop	VAR			Axis Stop FB3

- 程序本体

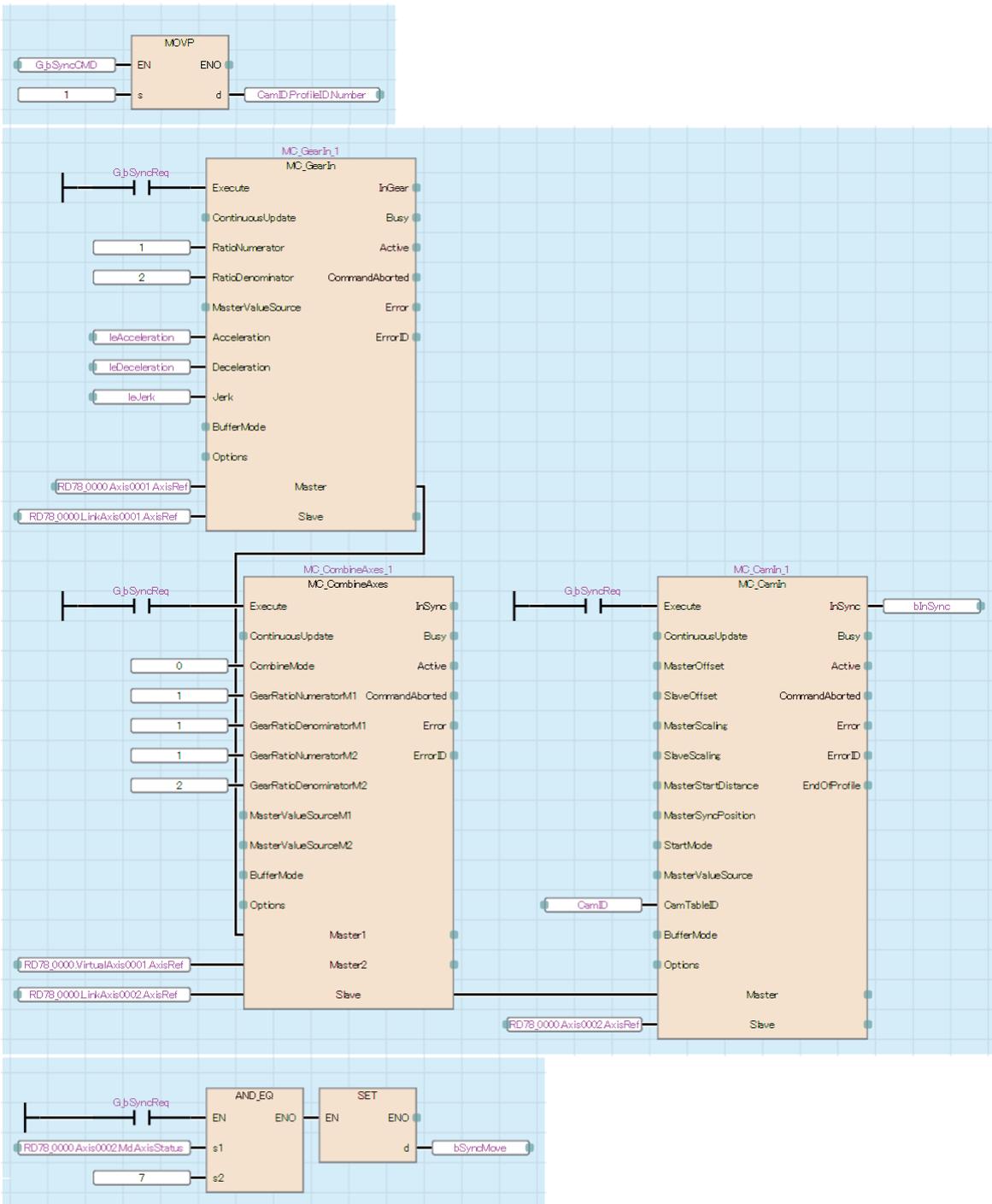
- 同步控制用数据设置



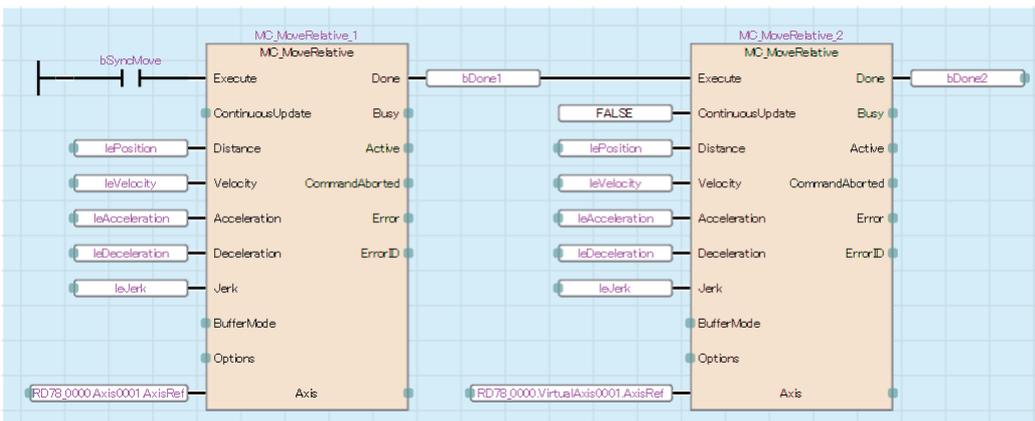
- 同步控制



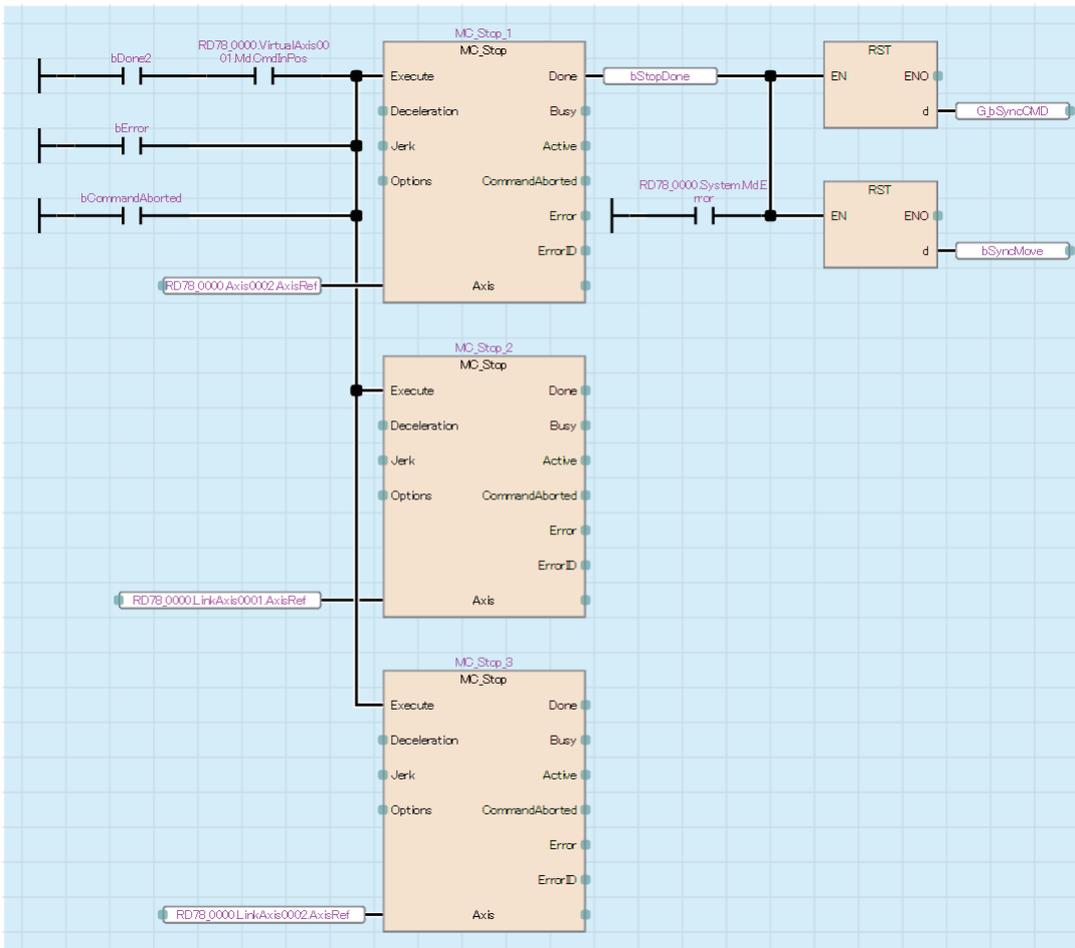
• 齿轮动作/加减法定位/凸轮动作



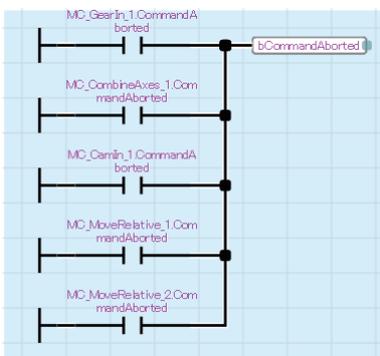
• 相对值定位 实驱动轴 Axis0001/相对值定位 虚拟驱动轴 VirtualAxis0001



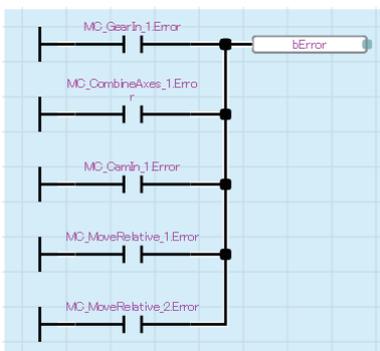
• 同步结束



• 各FB执行中断输出



• 各FB出错输出



运动控制设置功能中的设置

运动控制设置功能中进行的以下设置如下所示。

- 轴
- 轴组
- 全局标签
- 结构体
- 运算配置文件

轴

程序中使用的轴的设置如下所示。

数据名	标题	轴No.	轴类型设置	站地址
Axis0001	[X_Axis]	1	实驱动轴	192.168.3.1
Axis0002	[Y_Axis]	2	实驱动轴	192.168.3.2
VirtualAxis0001	[Vir_Axis01]	301	虚拟驱动轴	—
LinkAxis0001	[Lin_Axis01]	401	虚拟连接轴	—
LinkAxis0002	[Lin_Axis02]	402	虚拟连接轴	—

■实驱动轴

项目	Axis0001			Axis0002
轴信息	轴No.			2
轴参数常数	站地址设置			192.168.3.1
	轴类型设置			0: 实驱动轴
轴参数	驱动器单位转换分子			67108864[pulse]
	驱动器单位转换分母			5000[μm]
	停止信号	信号	对象	源类型
				源
位置指令单位			μm	

■虚拟驱动轴

项目	VirtualAxis0001		
轴信息	轴No.		
轴参数常数	轴类型设置		
轴参数	位置指令单位		

■虚拟连接轴

项目	LinkAxis0001	LinkAxis0002
轴信息	轴No.	402
轴参数常数	轴类型设置	5: 虚拟连接轴
轴参数	位置指令单位	μm

轴组

程序中使用的轴组的设置如下所示。

项目	设置	
数据名	AxesGroup001	
标题	[X-Y Table]	
轴组信息	轴组No.	
轴组参数	构成轴[1]	Axis0001
	构成轴[2]	Axis0002
	位置指令单位	μm

全局标签

程序中使用的全局标签的设置如下所示。

将使用的全局标签的公开标签设置为“有效”。

Global

Label Name	Data Type	Class	Initial Value	Constant	Comment	Remark	Public Label	Motion Control Attribute
1 G_bStopSignalX	Bit	VAR_GLOBAL			Stop command Axis0001		Enable	WRITE (=> Motion)
2 G_bStopSignalY	Bit	VAR_GLOBAL			Stop command Axis0002		Enable	WRITE (=> Motion)

Ax+Global

标签名	数据类型	分类	注释	公开标签
Axis0001	AXIS_REAL	VAR_GLOBAL	[X_Axis]	有效
Axis0002	AXIS_REAL	VAR_GLOBAL	[Y_Axis]	有效
VirtualAxis0001	AXIS_VIRTUAL	VAR_GLOBAL	[Vir_Axis01]	有效
LinkAxis0001	AXIS_VIRTUAL_LINK	VAR_GLOBAL	[Lin_Axis01]	有效
LinkAxis0002	AXIS_VIRTUAL_LINK	VAR_GLOBAL	[Lin_Axis02]	有效

Gr+Global

标签名	数据类型	分类	注释	公开标签
AxesGroup001	AXES_GROUP	VAR_GLOBAL	[X-Y Table]	有效

Sys+Global

标签名	数据类型	分类	注释	公开标签
System	SYSTEM	VAR_GLOBAL		有效

结构体

程序中使用的结构体的设置如下所示。

将使用的结构体的公开标签设置为“有效”。

AXIS_REAL

标签名	数据类型	注释	公开标签
AxisRef	AXIS_REF	轴信息	有效
Md	AXIS_REAL_MONI	轴监视数据	有效

AXIS_REAL_MONI

标签名	数据类型	注释	公开标签
AxisStatus	字[有符号]	轴状态	有效
CmdInPos	位	指令到位	有效
Driver_ServoOn	位	驱动器伺服ON状态	有效
Homing_Request	位	原点复位请求	有效

AXIS_VIRTUAL

标签名	数据类型	注释	公开标签
AxisRef	AXIS_REF	轴信息	有效
Md	AXIS_VIRTUAL_MONI	轴监视数据	有效

AXIS_VIRTUAL_MONI

标签名	数据类型	注释	公开标签
AxisStatus	字[有符号]	轴状态	有效
CmdInPos	位	指令到位	有效
Homing_Request	位	原点复位请求	有效

■AXIS_VIRTUAL_LINK

标签名	数据类型	注释	公开标签
AxisRef	AXIS_REF	轴信息	有效
Md	AXIS_VIRTUAL_LINK_MONI	轴监视数据	有效

■AXES_GROUP

标签名	数据类型	注释	公开标签
AxisGroupRef	AXIS_GROUP_REF	轴组信息	有效
Md	AXIS_GROUP_MONI	轴组监视数据	有效

■AXES_GROUP_MONI

标签名	数据类型	注释	公开标签
CmdInPos	位	指令到位	有效
GroupStatus	位	轴组状态	有效

■SYSTEM

标签名	数据类型	注释	公开标签
Md	SYS_MONI	系统监视数据	有效

■SYS_MONI

标签名	数据类型	注释	公开标签
Error	位	运动部系统出错检测	有效

运算配置文件

程序中使用的运算配置文件数据(凸轮数据)的设置如下所示。

项目	设置值	
数据名	ProfileData0001	
标题	[Cam #1]	
数据格式	凸轮数据	
插补方法指定	各区间中指定	
展开设置	自动展开	是
	配置文件ID	1
	重复动作	有效
	主轴绝对坐标	无效(相对坐标)
	从轴绝对坐标	有效(绝对坐标)
设置方法	分辨率	256
	1周期长	200000
	行程量	200000
	凸轮1周期时间	1.000[s]

■行程设置

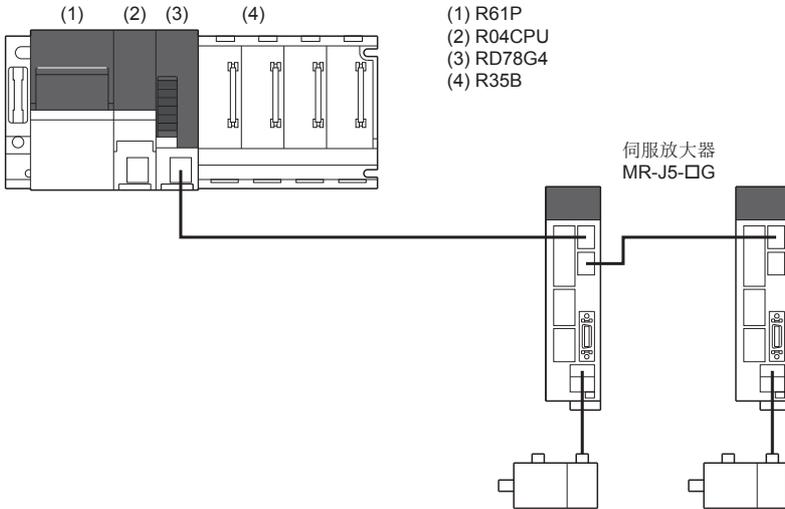
区间No.	开始点	结束点	行程	凸轮曲线类型
1	0	50000	100000	单弦
2	50000	100000	200000	单弦
3	100000	150000	50000	单弦
4	150000	0	0	单弦

附2 ST的样本程序

在运动模块侧通过ST语言创建的样本程序如下所示。

系统配置

程序示例中使用的系统配置如下所示。



GX Works3中的设置

在GX Works3中进行的以下设置如下所示。

- 系统参数
- CPU参数
- 模块参数(网络)
- 程序

系统参数

系统参数中设置的项目如下所示。

■基板/电源/扩展电缆设置

基板No.	基板	插槽数	电源模块
主基板	R35B	5	R61P

■I/O分配设置

插槽	模块型号	模块状态设置	点数	起始XY	异常检测时的CPU模块动作设置
基本	CPU	R04CPU(本机)	—	3E00	—
	0(0-0)	RD78G4	无设置	32点	0000

CPU参数

CPU参数中设置的项目如下所示。

■存储器/软元件设置

项目	设置
链接直接软元件设置	扩展模式(iQ-R系列模式)

■程序设置

执行顺序	程序名	执行类型		刷新组设置
		类型	详细设置信息	
1	MAIN	扫描	—	(不设置)

模块参数(网络)

模块参数(网络)中设置的项目如下所示。

■基本设置

- 网络配置设置

型号	站号	站类型	运动管理站	RWr设置 点数	RWw设置 点数	参数自动设置	IP地址
本站	0	主站	—	—	—	—	192.168.3.253
MR-J5-G	1	远程站	有勾选	24	20	有勾选	192.168.3.1
MR-J5-G	2	远程站	有勾选	24	20	有勾选	192.168.3.2

软元件

程序中使用的软元件如下所示。

软元件	注释
X0	准备就绪
X1	同步用标志
Y0	可编程控制器就绪
SM400	常开
M0	伺服OFF
M1	JOG正转指令 Axis0001
M2	JOG反转指令 Axis0001
M3	JOG正转指令 Axis0002
M4	JOG反转指令 Axis0002
M10	原点复位启动请求 Axis0001
M11	原点复位启动请求 Axis0002
M12	原点复位启动请求 VirtualAxis0001
M20	单轴定位启动
M21	单轴连续定位启动
M22	2轴直线插补控制启动
M23	同步控制启动
M30	出错复位
M31	系统出错复位

程序

创建用于使运动模块的ST程序启动的程序。
通过梯形图语言创建的程序如下所示。

MAIN

- 可编程控制器就绪



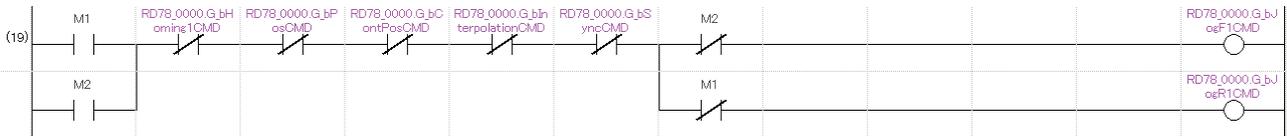
- 伺服ON/OFF



- JOG运行



- Axis0001 JOG指令

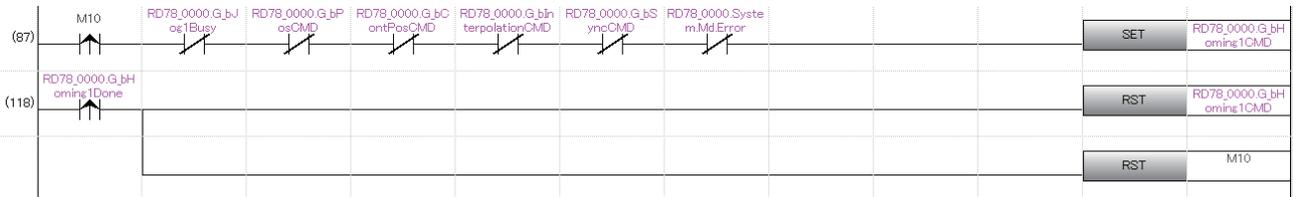


- Axis0002 JOG指令

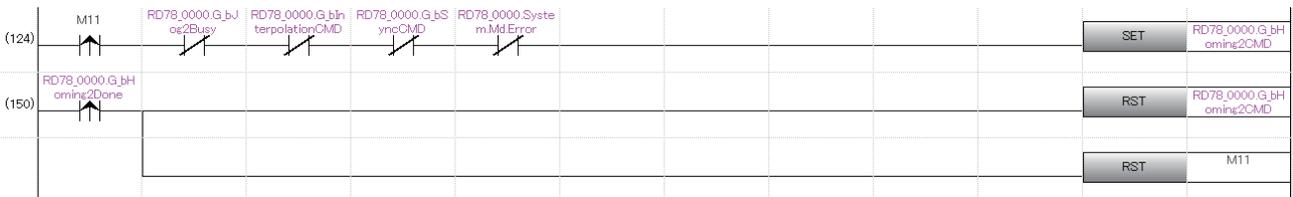


- 原点复位

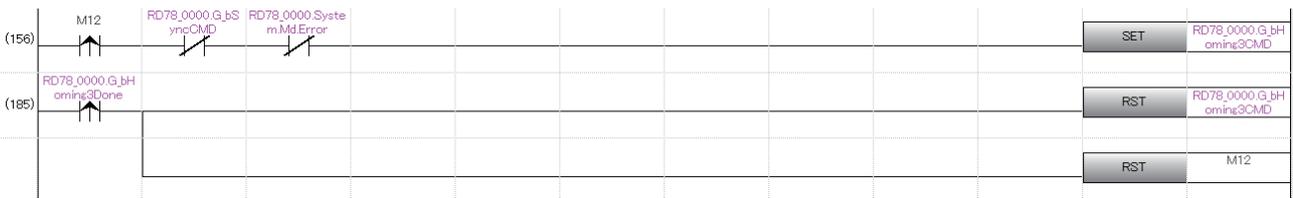
- Axis0001 原点复位



- Axis0002 原点复位



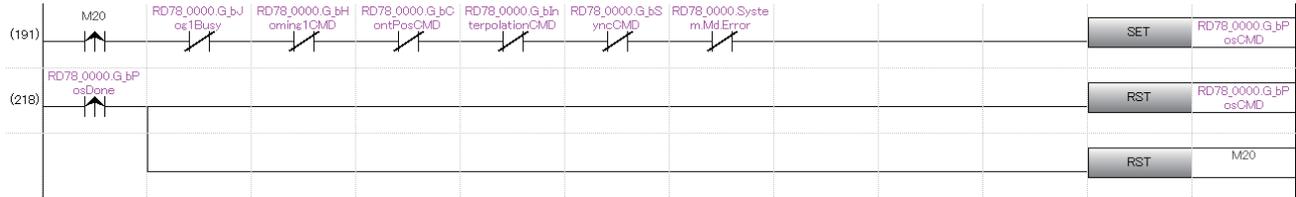
- VirtualAxis0001 原点复位



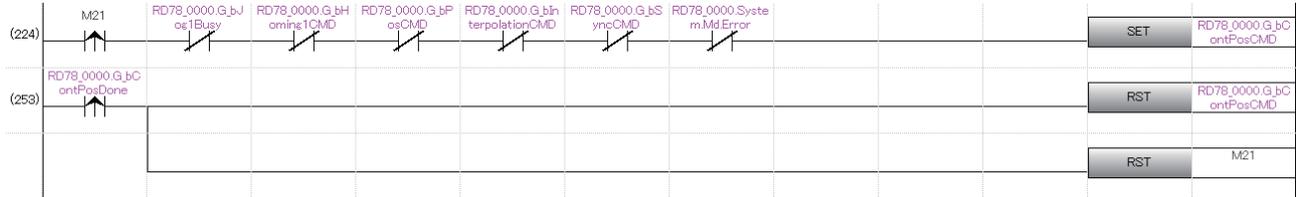
附

• 程序启动

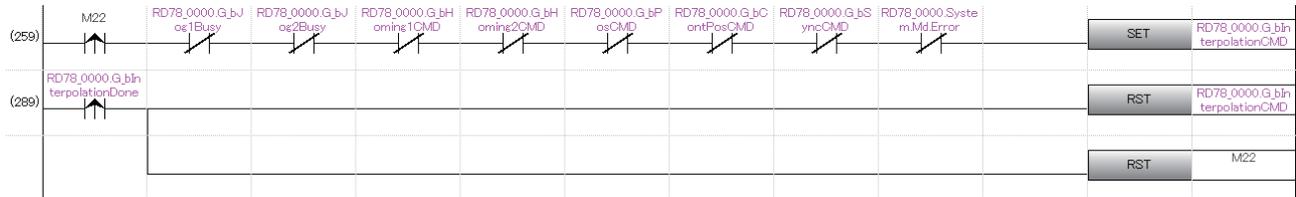
• 单轴定位控制



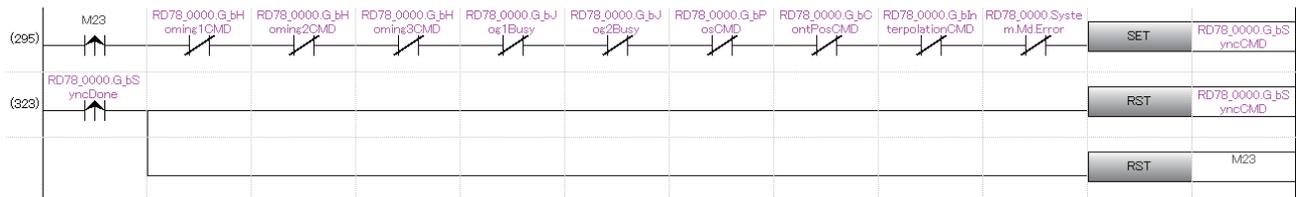
• 单轴连续定位控制



• 2轴直线插补控制



• 同步控制



• 出错复位

• 出错复位



• 系统出错复位



运动控制设置功能中的设置

运动控制设置功能中进行的以下设置如下所示。

- 基本设置
- 轴
- 轴组
- 全局标签
- 运算配置文件
- 程序

基本设置

基本设置中设置的项目如下所示。

■程序执行设置

执行顺序	程序名称	执行类型	程序执行条件
1	ServoON_Jog	1: 正常	—
2	Positioning	1: 正常	—
3	ContinuousPositioning	1: 正常	—
4	LinearInterpolation	1: 正常	—
5	Homing	1: 正常	—
6	ErrorReset	1: 正常	—
7	Synchronous	1: 正常	—

轴

程序中使用的轴的设置如下所示。

数据名	标题	轴No.	轴类型设置	站地址
Axis0001	[X_Axis]	1	实驱动轴	192.168.3.1
Axis0002	[Y_Axis]	2	实驱动轴	192.168.3.2
VirtualAxis0001	[Vir_Axis01]	301	虚拟驱动轴	—
LinkAxis0001	[Lin_Axis01]	401	虚拟连接轴	—
LinkAxis0002	[Lin_Axis02]	402	虚拟连接轴	—

■实驱动轴

项目	Axis0001				Axis0002
轴信息	轴No.				2
轴参数常数	站地址设置				192.168.3.2
	轴类型设置				0: 实驱动轴
轴参数	驱动器单位转换分子				67108864[pulse]
	驱动器单位转换分母				5000[μm]
	停止信号	信号	对象	源类型	全局标签
	源				[VAR]G_bStopSignalY
位置指令单位				μm	

■虚拟驱动轴

项目	VirtualAxis0001			
轴信息	轴No.			
轴参数常数	轴类型设置			
轴参数	位置指令单位			

■虚拟连接轴

项目		LinkAxis0001	LinkAxis0002
轴信息	轴No.	401	402
轴参数常数	轴类型设置	5: 虚拟连接轴	5: 虚拟连接轴
轴参数	位置指令单位	μm	μm

轴组

程序中使用的轴组的设置如下所示。

项目		设置
数据名		AxesGroup001
标题		[X-Y Table]
轴组信息	轴组No.	1
轴组参数	构成轴[1]	Axis0001
	构成轴[2]	Axis0002
	位置指令单位	μm

全局标签

程序中使用的全局标签的设置如下所示。

将使用的全局标签的公开标签设置为“有效”。

■Global

	Label Name	Data Type	Class	Initial Value	Constant	Comment	Remark	Public Label	Motion Control Attribute
1	G_bSRVONCMD	Bit	VAR_GLOBAL			Servo ON/OFF		Enable	WRITE (⇒ Motion)
2	G_bHoming1CMD	Bit	VAR_GLOBAL			Homing command Axis0001		Enable	WRITE (⇒ Motion)
3	G_bHoming2CMD	Bit	VAR_GLOBAL			Homing command Axis0002		Enable	WRITE (⇒ Motion)
4	G_bHoming3CMD	Bit	VAR_GLOBAL			Homing command VirtualAxis001		Enable	WRITE (⇒ Motion)
5	G_bPosCMD	Bit	VAR_GLOBAL			Single axis positioning start		Enable	WRITE (⇒ Motion)
6	G_bContPosCMD	Bit	VAR_GLOBAL			Single axis continuous positioning start		Enable	WRITE (⇒ Motion)
7	G_bInterpolationCMD	Bit	VAR_GLOBAL			2-axis linear interpolation control start		Enable	WRITE (⇒ Motion)
8	G_bSyncCMD	Bit	VAR_GLOBAL			Synchronous control start		Enable	WRITE (⇒ Motion)
9	G_bResetCMD	Bit	VAR_GLOBAL			Error reset		Enable	WRITE (⇒ Motion)
10	G_bMotionResetCMD	Bit	VAR_GLOBAL			System error reset		Enable	WRITE (⇒ Motion)
11	G_bJogF1CMD	Bit	VAR_GLOBAL			JOG Positive rotation command Axis0001		Enable	WRITE (⇒ Motion)
12	G_bJogR1CMD	Bit	VAR_GLOBAL			JOG Reverse rotation command Axis0001		Enable	WRITE (⇒ Motion)
13	G_bJogF2CMD	Bit	VAR_GLOBAL			JOG Positive rotation command Axis0002		Enable	WRITE (⇒ Motion)
14	G_bJogR2CMD	Bit	VAR_GLOBAL			JOG Reverse rotation command Axis0002		Enable	WRITE (⇒ Motion)
15	G_bJogVelocity	FLOAT [Double Precision]	VAR_GLOBAL			JOG Velocity		Enable	WRITE (⇒ Motion)
16	G_bHoming1Done	Bit	VAR_GLOBAL			Homing complete Axis0001		Enable	READ (Motion ⇒)
17	G_bHoming2Done	Bit	VAR_GLOBAL			Homing complete Axis0002		Enable	READ (Motion ⇒)
18	G_bHoming3Done	Bit	VAR_GLOBAL			Homing complete VirtualAxis001		Enable	READ (Motion ⇒)
19	G_bPosDone	Bit	VAR_GLOBAL			Single axis positioning complete		Enable	READ (Motion ⇒)
20	G_bContPosDone	Bit	VAR_GLOBAL			Single axis continuous positioning complete		Enable	READ (Motion ⇒)
21	G_bInterpolationDone	Bit	VAR_GLOBAL			2-axis linear interpolation control complete		Enable	READ (Motion ⇒)
22	G_bSyncDone	Bit	VAR_GLOBAL			Synchronous control complete		Enable	READ (Motion ⇒)
23	G_bJog1Busy	Bit	VAR_GLOBAL			JOG operation in progress Axis0001		Enable	READ (Motion ⇒)
24	G_bJog2Busy	Bit	VAR_GLOBAL			JOG operation in progress Axis0002		Enable	READ (Motion ⇒)
25	G_bStopSignalX	Bit	VAR_GLOBAL			Stop command Axis0001		Enable	READ (Motion ⇒)
26	G_bStopSignalY	Bit	VAR_GLOBAL			Stop command Axis0002		Enable	READ (Motion ⇒)

■Sys+Global

标签名	数据类型	分类	注释	公开标签
System	SYSTEM	VAR_GLOBAL		有效

结构体

程序中使用的结构体的设置如下所示。

将使用的结构体的公开标签设置为“有效”。

■AXIS_REAL

标签名	数据类型	注释	公开标签
AxisRef	AXIS_REF	轴信息	有效

■AXIS_VIRTUAL

标签名	数据类型	注释	公开标签
AxisRef	AXIS_REF	轴信息	有效

■AXIS_VIRTUAL_LINK

标签名	数据类型	注释	公开标签
AxisRef	AXIS_REF	轴信息	有效

■AXES_GROUP

标签名	数据类型	注释	公开标签
AxisGroupRef	AXIS_GROUP_REF	轴组信息	有效

■SYSTEM

标签名	数据类型	注释	公开标签
Md	SYS_MONI	系统监视数据	有效

■SYS_MONI

标签名	数据类型	注释	公开标签
Error	位	运动部系统出错检测	有效

运算配置文件

程序中使用的运算配置文件数据(凸轮数据)的设置如下所示。

项目	设置值	
数据名	ProfileData0001	
标题	[Cam #1]	
数据格式	凸轮数据	
插补方法指定	各区间中指定	
展开设置	自动展开	是
	配置文件ID	1
	重复动作	有效
	主轴绝对坐标	无效(相对坐标)
	从轴绝对坐标	有效(绝对坐标)
设置方法	分辨率	256
	1周期长	200000
	行程量	200000
	凸轮1周期时间	1.000[s]

■行程设置

区间No.	开始点	结束点	行程	凸轮曲线类型
1	0	50000	100000	单弦
2	50000	100000	200000	单弦
3	100000	150000	50000	单弦
4	150000	0	0	单弦

程序

通过ST语言创建的程序如下所示。

程序名称	处理内容	参阅
ServoON_Jog	<ul style="list-style-type: none"> 进行所有轴的伺服ON。 进行轴1(Axis0001)与轴2(Axis0002)的JOG运行。 	☞ 462页 ServoON_Jog
Positioning	<ul style="list-style-type: none"> 根据轴1(Axis0001)的相对值进行单轴定位控制。 	☞ 463页 Positioning
ContinuousPositioning	<ul style="list-style-type: none"> 根据轴1(Axis0001)与轴2(Axis0002)的相对值进行单轴连续定位控制。 	☞ 464页 ContinuousPositioning
LinearInterpolation	<ul style="list-style-type: none"> 对轴1(Axis0001)与轴2(Axis0002)的轴组1(AxesGroup001)进行2轴直线插补控制。 	☞ 466页 LinearInterpolation
Homing	<ul style="list-style-type: none"> 进行轴1(Axis0001)、轴2(Axis0002)、轴301(VirtualAxis0001)的原点复位。 	☞ 468页 Homing
ErrorReset	<ul style="list-style-type: none"> 进行轴1(Axis0001)、轴2(Axis0002)、轴301(VirtualAxis0001)、轴401(LinkAxis0001)、轴402(LinkAxis0002)、轴组1(AxesGroup001)、运动系统的出错复位。 	☞ 469页 ErrorReset
Synchronous	<ul style="list-style-type: none"> 进行同步控制。 	☞ 470页 Synchronous

■ServoON_Jog

- 局部标签

	Label Name	Data Type	Class	Initial Value	Constant	Comment
1	MCv_AllPower_1	MCv_AllPower	VAR			All axes servo ON FB
2	MCv_Jog_1	MCv_Jog	VAR			JOG operation FB1 Axis0001
3	MCv_Jog_2	MCv_Jog	VAR			JOG operation FB2 Axis0002
4	leJogAcceleration	FLOAT [Double Precision]	VAR			JOG acceleration
5	leJogDeceleration	FLOAT [Double Precision]	VAR			JOG deceleration
6	leJogJerk	FLOAT [Double Precision]	VAR			JOG jerk

- 程序本体

```
//-----所有轴伺服ON-----
MCv_AllPower_1(
    Enable:= TRUE ,
    ServoON:= G_bSRVONCMD
);

//-----JOG运行用数据设置-----
IF MCv_AllPower_1.Busy THEN
    leJogAcceleration:= 50000.0;
    leJogDeceleration:= 50000.0;
    leJogJerk:= 0.0;
END_IF;

//-----Axis0001 JOG运行-----
MCv_Jog_1(
    Axis:= Axis0001.AxisRef ,
    JogForward:= G_bJogF1CMD ,
    JogBackward:= G_bJogR1CMD ,
    Velocity:= G_leJogVelocity ,
    Acceleration:= leJogAcceleration ,
    Deceleration:= leJogDeceleration ,
    Jerk:= leJogJerk ,
    BUSY=> G_bJog1Busy
);

//-----Axis0002 JOG运行-----
MCv_Jog_2(
```

```

Axis:= Axis0002.AxisRef ,
JogForward:= G_bJogF2CMD ,
JogBackward:= G_bJogR2CMD ,
Velocity:= G_leJogVelocity ,
Acceleration:= leJogAcceleration ,
Deceleration:= leJogDeceleration ,
Jerk:= leJogJerk ,
BUSY=> G_bJog2Busy
);

```

■Positioning

• 局部标签

	Label Name	Data Type	Class	Initial Value	Constant	Comment
1	MC_MoveRelative_1	MC_MoveRelative	VAR			Relative value positioning FB
2	leDistance	FLOAT [Double Precision]	VAR			Distance
3	leVelocity	FLOAT [Double Precision]	VAR			Velocity
4	leAcceleration	FLOAT [Double Precision]	VAR			Acceleration
5	leDeceleration	FLOAT [Double Precision]	VAR			Deceleration
6	leJerk	FLOAT [Double Precision]	VAR			Jerk
7	bDone	Bit	VAR			Relative value positioning FB Done output
8	bBusy	Bit	VAR			Relative value positioning FB Busy output
9	bError	Bit	VAR			Relative value positioning FB Error output
10	bExecute_P	Bit	VAR			Start
11	bCommandAborted	Bit	VAR			Relative value positioning FB CommandAborted output

• 程序本体

```
//-----单轴定位控制用数据设置・执行指令&复位-----
```

```

IF G_bPosCMD THEN;
    leDistance:= 200000.0;
    leVelocity:= 50000.0;
    leAcceleration:= 100000.0;
    leDeceleration:= 100000.0;
    leJerk:= 0.0;
    bExecute_P:= TRUE;
ELSE
    bExecute_P:= FALSE;
END_IF;

```

```
//-----单轴定位-----
```

```
//相对值定位
```

```

MC_MoveRelative_1(
    Axis:= Axis0001.AxisRef ,
    Execute:= bExecute_P ,
    Distance:= leDistance ,
    Velocity:= leVelocity ,
    Acceleration:= leAcceleration ,
    Deceleration:= leDeceleration ,
    Jerk:= leJerk ,
    Done=> bDone ,
    Busy=> bBusy ,
    CommandAborted=> bCommandAborted ,
    Error=> bError
);

```

```
//发送到可编程控制器 <= 定位完成信号・FB执行中断・FB出错的任意一个输出变为ON *控制结束信号
```

```
G_bPosDone:= ( bDone & Axis0001.Md.CmdInPos ) OR bCommandAborted OR bError;
```

■ ContinuousPositioning

• 局部标签

	Label Name	Data Type	Class	Initial Value	Constant	Comment
1	MC_MoveRelative_1	MC_MoveRelative	VAR			Relative value positioning FB1
2	MC_MoveRelative_2	MC_MoveRelative	VAR			Relative value positioning FB2
3	leDistance1	FLOAT [Double Precision]	VAR			Distance1
4	leDistance2	FLOAT [Double Precision]	VAR			Distance2
5	leVelocity1	FLOAT [Double Precision]	VAR			Velocity1
6	leVelocity2	FLOAT [Double Precision]	VAR			Velocity2
7	leAcceleration1	FLOAT [Double Precision]	VAR			Acceleration1
8	leDeceleration1	FLOAT [Double Precision]	VAR			Deceleration1
9	leAcceleration2	FLOAT [Double Precision]	VAR			Acceleration2
10	leDeceleration2	FLOAT [Double Precision]	VAR			Deceleration2
11	leJerk	FLOAT [Double Precision]	VAR			Jerk
12	bBusy1	Bit	VAR			Relative value positioning FB1 Busy output
13	bActive1	Bit	VAR			Relative value positioning FB1 Active output
14	bDone2	Bit	VAR			Relative value positioning FB2 Done output
15	bBusy2	Bit	VAR			Relative value positioning FB2 Busy output
16	TON_1	TON	VAR			On-delay timer FB
17	bDwell_in	Bit	VAR			Timer input
18	bDwell_out	Bit	VAR			Timer output
19	bExecute_CP	Bit	VAR			Start
20	bCommandAborted	Bit	VAR			Relative value positioning FB CommandAborted output
21	bError	Bit	VAR			Relative value positioning FB Error output

• 程序本体

//-----单轴连续定位用数据设置・执行指令&复位-----

```
IF G_bContPosCMD THEN;
    leDistance1:= 200000.0;
    leDistance2:= 200000.0;
    leVelocity1:= 50000.0;
    leVelocity2:= 25000.0;
    leAcceleration1:= 100000.0;
    leDeceleration1:= 100000.0;
    leAcceleration2:= 50000.0;
    leDeceleration2:= 50000.0;
    leJerk:= 0.0;
    bExecute_CP:= TRUE;
ELSE
    bExecute_CP:= FALSE;
END_IF;
```

//-----单轴连续定位+缓冲模式+停留-----

///相对值定位1

```
MC_MoveRelative_1(
    Axis:= Axis0001.AxisRef ,
    Execute:= bExecute_CP ,
    Distance:= leDistance1 ,
    Velocity:= leVelocity1 ,
    Acceleration:= leAcceleration1 ,
    Deceleration:= leDeceleration1 ,
    Jerk:= leJerk ,
    Busy=> bBusy1 ,
    Active=> bActive1
);
```

//相对值定位2

```
MC_MoveRelative_2(
    Axis:= Axis0001.AxisRef ,
    Execute:= bActive1,
```

```

Distance:= leDistance2 ,
Velocity:= leVelocity2 ,
Acceleration:= leAcceleration2 ,
Deceleration:= leDeceleration2 ,
BufferMode:= MC_BUFFER_MODE__mcBlendingPrevious ,
Done=> bDone2 ,
Busy=> bBusy2
);

//FB执行中断输出
bCommandAborted:= MC_MoveRelative_1.CommandAborted OR MC_MoveRelative_2.CommandAborted;

//FB出错输出
bError:= MC_MoveRelative_1.Error OR MC_MoveRelative_2.Error;

//通过相对值定位2完成信号设置定时器输入
SET( bDone2, bDwell_in );

//停留
TON_1(
    IN:= bDwell_in ,
    PT:= T#100ms ,
    Q=> bDwell_out
);

//发送到可编程控制器 <= 经过停留时间 • FB执行中断 • FB出错的任意一个输出变为ON *控制结束信号
G_bContPosDone:= bDwell_out OR bCommandAborted OR bError ;

//定位完成时，复位相对定位2完成信号保持(定时器输入)
RST( bDwell_out, bDwell_in );

```

■ Linear Interpolation

• 局部标签

	Label Name	Data Type	Class	Initial Value	Constant	Comment
1	MCv_MoveLinearInterpolateRelative_1	MCv_MoveLinearInterpolateRelative	VAR			Relative value linear interpolation control FB1
2	MCv_MoveLinearInterpolateRelative_2	MCv_MoveLinearInterpolateRelative	VAR			Relative value linear interpolation control FB2
3	MC_GroupEnable_1	MC_GroupEnable	VAR			Axes group enable FB
4	MC_GroupDisable_1	MC_GroupDisable	VAR			Axes group disable FB
5	wAxes	Word [Signed](0..15)	VAR			Interpolation axis
6	lePosition1	FLOAT [Double Precision](0..15)	VAR			Position data 1
7	lePosition2	FLOAT [Double Precision](0..15)	VAR			Position data 2
8	leVelocity	FLOAT [Double Precision]	VAR			Velocity
9	leAcceleration	FLOAT [Double Precision]	VAR			Acceleration
10	leDeceleration	FLOAT [Double Precision]	VAR			Deceleration
11	leJerk	FLOAT [Double Precision]	VAR			Jerk
12	bGroupEnableDone	Bit	VAR			Axes group enable done
13	bGroupDisableDone	Bit	VAR			Axes group disable done
14	bBusy1	Bit	VAR			Relative value linear interpolation control FB1 Busy output
15	bActive1	Bit	VAR			Relative value linear interpolation control FB1 Active output
16	bDone2	Bit	VAR			Relative value linear interpolation control FB2 Done output
17	bBusy2	Bit	VAR			Relative value linear interpolation control FB2 Busy output
18	TON_1	TON	VAR			On-delay timer FB
19	bDwell_out	Bit	VAR			Timer output
20	bDwell_in	Bit	VAR			Timer input
21	bExcute_LP	Bit	VAR			Start
22	bCommandAborted	Bit	VAR			Relative value linear interpolation control FB CommandAborted output
23	bError	Bit	VAR			Relative value linear interpolation control FB Error output

• 程序本体

//-----2轴直线插补控制用数据设置・执行指令&复位-----

```

IF G_bInterpolationCMD THEN;
    wAxes[0]:= 1;
    wAxes[1]:= 2;
    lePosition1[0]:= 100000.0;
    lePosition1[1]:= 200000.0;
    lePosition2[0]:= -100000.0;
    lePosition2[1]:= -200000.0;
    leVelocity:= 50000.0;
    leAcceleration:= 50000.0;
    leDeceleration:= 50000.0;
    leJerk:= 0.0;
    bExcute_LP:= TRUE;
ELSE
    bExcute_LP:= FALSE;
END_IF;

//-----直线插补控制+缓冲模式-----
//轴组有效
MC_GroupEnable_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bExcute_LP ,
    Done=> bGroupEnableDone
);

//直线插补控制1
MCv_MoveLinearInterpolateRelative_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bGroupEnableDone ,
    LinearAxes:= wAxes ,
    Distance:= lePosition1 ,
    Velocity:= leVelocity ,

```

```

Acceleration:= leAcceleration ,
Deceleration:= leDeceleration ,
Jerk:= leJerk ,
VelocityMode:= MC_INTERPOLATE_SPEED_MODE__VectorSpeed ,
Busy=> bBusyl ,
Active=> bActive1
);

//直线插补控制2
MCv_MoveLinearInterpolateRelative_2(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= bActive1 ,
    LinearAxes:= wAxes ,
    Distance:= lePosition2 ,
    BufferMode:= MC_BUFFER_MODE__mcBuffered ,
    Done=> bDone2 ,
    Busy=> bBusy2
);

//FB执行中断输出
bCommandAborted:= MCv_MoveLinearInterpolateRelative_1.CommandAborted OR
MCv_MoveLinearInterpolateRelative_2.CommandAborted;

//FB出错输出
bError:= MCv_MoveLinearInterpolateRelative_1.Error OR MCv_MoveLinearInterpolateRelative_2.Error;

//通过直线插补控制2完成信号或FB执行中断设置定时器输入
SET( bDone2 OR bCommandAborted, bDwell_in );

//停留
TON_1(
    IN:= bDwell_in ,
    PT:= T#100ms ,
    Q=> bDwell_out
);

//轴组无效
MC_GroupDisable_1(
    AxesGroup:= AxesGroup001.AxesGroupRef,
    Execute:= bDwell_out OR bError,
    Done=> bGroupDisableDone
);

//发送到可编程控制器 <= 经过停留时间 • FB出错的任意一个输出变为ON *控制结束信号
G_bInterpolationDone:= bDwell_out OR bError;

//定位完成时，复位插补定位2完成信号保持(定时器输入)
RST( bDwell_out, bDwell_in );

```

■Homing

- 局部标签

	Label Name	Data Type	Class	Initial Value	Constant	Comment
1	MC_Home_1	MC_Home	VAR			Homing FB1 Axis0001
2	MC_Home_2	MC_Home	VAR			Homing FB2 Axis0002
3	MC_Home_3	MC_Home	VAR			Homing FB3 VirtualAxis0001
4	bHoming1Done	Bit	VAR			Homing FB1 Done output
5	bHoming1Error	Bit	VAR			Homing FB1 Error output
6	bHoming2Done	Bit	VAR			Homing FB2 Done output
7	bHoming2Error	Bit	VAR			Homing FB2 Error output
8	bHoming3Done	Bit	VAR			Homing FB3 Done output
9	bHoming3Error	Bit	VAR			Homing FB3 Error output

- 程序本体

```
//-----原点复位 Axis0001-----  
//原点复位  
MC_Home_1(  
    Axis:= Axis0001.AxisRef ,  
    Execute:= G_bHoming1CMD ,  
    Position:= 0.0 ,  
    Done=> bHoming1Done,  
    Error=> bHoming1Error  
);  
  
//将发送到可编程控制器的原点复位完成信号置为ON  
G_bHoming1Done:= bHoming1Done OR bHoming1Error;  
  
//-----原点复位 Axis0002-----  
//原点复位  
MC_Home_2(  
    Axis:= Axis0002.AxisRef ,  
    Execute:= G_bHoming2CMD ,  
    Position:= 0.0 ,  
    Done=> bHoming2Done,  
    Error=> bHoming2Error  
);  
  
//将发送到可编程控制器的原点复位完成信号置为ON  
G_bHoming2Done:= bHoming2Done OR bHoming2Error;  
  
//-----原点复位VirtualAxis0001-----  
//原点复位  
MC_Home_3(  
    Axis:= VirtualAxis0001.AxisRef ,  
    Execute:= G_bHoming3CMD ,  
    Position:= 0.0 ,  
    Done=> bHoming3Done,  
    Error=> bHoming3Error  
);  
  
//将发送到可编程控制器的原点复位完成信号置为ON  
G_bHoming3Done:= bHoming3Done OR bHoming3Error;
```

■ ErrorReset

- 局部标签

	Label Name	Data Type	Class	Initial Value	Constant	Comment
1	MC_Reset_1	MC_Reset	VAR			Axis error reset FB1 Axis0001
2	MC_Reset_2	MC_Reset	VAR			Axis error reset FB2 Axis0002
3	MC_Reset_3	MC_Reset	VAR			Axis error reset FB3 VirtualAxis0001
4	MC_Reset_4	MC_Reset	VAR			Axis error reset FB4 LinkAxis0001
5	MC_Reset_5	MC_Reset	VAR			Axis error reset FB5 LinkAxis0002
6	MC_GroupReset_1	MC_GroupReset	VAR			Axes group error reset FB
7	MCv_MotionErrorReset_1	MCv_MotionErrorReset	VAR			System error reset FB

- 程序本体

```
// 出错复位
MC_Reset_1(
    Axis:= Axis0001.AxisRef ,
    Execute:= G_bResetCMD
);

MC_Reset_2(
    Axis:= Axis0002.AxisRef ,
    Execute:= G_bResetCMD
);

MC_Reset_3(
    Axis:= VirtualAxis0001.AxisRef ,
    Execute:= G_bResetCMD
);

MC_Reset_4(
    Axis:= LinkAxis0001.AxisRef ,
    Execute:= G_bResetCMD
);

MC_Reset_5(
    Axis:= LinkAxis0002.AxisRef ,
    Execute:= G_bResetCMD
);

// 轴组出错复位
MC_GroupReset_1(
    AxesGroup:= AxesGroup001.AxesGroupRef ,
    Execute:= G_bResetCMD
);

// 控制器出错复位
MCv_MotionErrorReset_1(
    Execute:= G_bMotionResetCMD
);
```

■ Synchronous

• 局部标签

	Label Name	Data Type	Class	Initial Value	Constant	Comment
1	MC_GearIn_1	MC_GearIn	VAR			Gear operation FB
2	MC_CombineAxes_1	MC_CombineAxes	VAR			FB combining the motion of two master axes
3	MC_CamIn_1	MC_CamIn	VAR			Cam operation FB
4	bGearInBusy	Bit	VAR			Gear operation FB Busy output
5	bCombineAxesBusy	Bit	VAR			FB combining the motion of two master axes Busy output
6	bCamInBusy	Bit	VAR			Cam operation FB Busy output
7	MC_MoveRelative_1	MC_MoveRelative	VAR			Relative value positioning FB
8	MC_MoveRelative_2	MC_MoveRelative	VAR			Relative value positioning FB
9	MC_Stop_1	MC_Stop	VAR			Axis Stop FB1
10	MC_Stop_2	MC_Stop	VAR			Axis Stop FB2
11	MC_Stop_3	MC_Stop	VAR			Axis Stop FB3
12	leAcceleration	FLOAT [Double Precision]	VAR			Acceleration
13	leDeceleration	FLOAT [Double Precision]	VAR			Deceleration
14	leJerk	FLOAT [Double Precision]	VAR			Jerk
15	lePosition1	FLOAT [Double Precision]	VAR			Distance
16	leVelocity	FLOAT [Double Precision]	VAR			Velocity
17	bInSync	Bit	VAR			Cam operation FB inSync output
18	bDone1	Bit	VAR			Relative value positioning FB1 Done output
19	bBusy1	Bit	VAR			Relative value positioning FB1 Busy output
20	bDone2	Bit	VAR			Relative value positioning FB2 Done output
21	bBusy2	Bit	VAR			Relative value positioning FB2 Busy output
22	bStopReq1	Bit	VAR			Axis Stop request 1
23	bStopDone1	Bit	VAR			Axis Stop complete 1
24	bStopDone2	Bit	VAR			Axis Stop complete 2
25	bStopDone3	Bit	VAR			Axis Stop complete 3
26	bSyncMoveCMD	Bit	VAR			Relative value positioning start
27	bExecute_S	Bit	VAR			Start
28	bCommandAborted	Bit	VAR			Relative value positioning FB CommandAborted output
29	bError	Bit	VAR			Relative value positioning FB Error output

• 程序本体

//-----同步控制用数据设置・执行指令&复位-----

IF G_bSyncCMD THEN;

 lePosition1:= 400000.0;

 leVelocity:= 50000.0;

 leAcceleration:= 100000.0;

 leDeceleration:= 100000.0;

 leJerk:= 0.0;

 bExecute_S:= TRUE;

ELSE

 bExecute_S:= FALSE;

END_IF;

//-----同步控制-----

//齿轮动作

MC_GearIn_1(

 Master:= Axis0001.AxisRef ,

 Slave:= LinkAxis0001.AxisRef ,

 Execute:= bExecute_S ,

 RatioNumerator:= 1 ,

 RatioDenominator:= 2 ,

 MasterValueSource:= MC_SOURCE_mcSetValue ,

 Acceleration:= leAcceleration ,

 Deceleration:= leDeceleration ,

 Jerk:= leJerk ,

 Busy=> bGearInBusy

);

//加减法定位

MC_CombineAxes_1(

```

Master1:= LinkAxis0001.AxisRef ,
Master2:= VirtualAxis0001.AxisRef ,
Slave:= LinkAxis0002.AxisRef ,
Execute:= bExecute_S ,
CombineMode:= MC_COMBINE_MODE__mcAddAxes ,
GearRatioNumeratorM1:= 1 ,
GearRatioDenominatorM1:= 1 ,
GearRatioNumeratorM2:= 1 ,
GearRatioDenominatorM2:= 2 ,
MasterValueSourceM1:= MC_SOURCE__mcSetValue ,
MasterValueSourceM2:= MC_SOURCE__mcSetValue ,
Busy=> bCombineAxesBusy
);

//凸轮动作
MC_CamIn_1.CamTableID.ProfileID:= ProfileData0001.ProfileData.ID;

MC_CamIn_1(
    Master:= LinkAxis0002.AxisRef ,
    Slave:= Axis0002.AxisRef ,
    Execute:= bExecute_S ,
    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 ,
    InSync=> bInSync ,
    Busy=> bCamInBusy
);

//输入轴启动
bSyncMoveCMD:= G_bSyncCMD & ( Axis0002.Md.AxisStatus = MC_AXIS_STATUS__SynchronizedMotion );

//相对值定位 实驱动轴 Axis0001
MC_MoveRelative_1(
    Axis:= Axis0001.AxisRef ,
    Execute:= bSyncMoveCMD ,
    Distance:= lePosition1 ,
    Velocity:= leVelocity ,
    Acceleration:= leAcceleration ,
    Deceleration:= leDeceleration ,
    Jerk:= leJerk ,
    Done=> bDone1 ,
    Busy=> bBusy1
);

//相对值定位 虚拟驱动轴 VirtualAxis0001
MC_MoveRelative_2(
    Axis:= VirtualAxis0001.AxisRef ,

```

```

Execute:= bDone1 & Axis0001.Md.CmdInPos ,
Distance:= lePosition1 ,
Velocity:= leVelocity ,
Acceleration:= leAcceleration ,
Deceleration:= leDeceleration ,
Jerk:= leJerk ,
Done=> bDone2 ,
Busy=> bBusy2
);

//FB执行中断输出
bCommandAborted:= MC_GearIn_1.CommandAborted OR MC_CombineAxes_1.CommandAborted OR MC_CamIn_1.CommandAborted
OR MC_MoveRelative_1.CommandAborted OR MC_MoveRelative_2.CommandAborted;

//FB出错输出
bError:= MC_GearIn_1.Error OR MC_CombineAxes_1.Error OR MC_CamIn_1.Error OR MC_MoveRelative_1.Error OR
MC_MoveRelative_2.Error;

//定位完成时或执行中断时、出错发生时将轴停止信号置为ON
bStopReq1:= ( bDone2 & VirtualAxis0001.Md.CmdInPos ) OR bError OR bCommandAborted;

//同步结束
MC_Stop_1(
Axis:= Axis0002.AxisRef ,
Execute:= bStopReq1 ,
Done=> bStopDone1
);

MC_Stop_2(
Axis:= LinkAxis0001.AxisRef ,
Execute:= bStopReq1 ,
Done=> bStopDone2
);

MC_Stop_3(
Axis:= LinkAxis0002.AxisRef ,
Execute:= bStopReq1 ,
Done=> bStopDone3
);

//同步结束后，复位启动请求信号
RST( bStopDone1 OR System.Md.Error, bSyncMoveCMD );
RST( bStopDone1 OR System.Md.Error, bStopReq1 );

//发送到可编程控制器 <= 定位完成信号 • FB执行中断 • FB出错的任意一个输出变为ON *控制结束信号
G_bSyncDone:= ( bStopDone2 & VirtualAxis0001.Md.CmdInPos ) OR bCommandAborted OR bError;

```


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
2020年11月	IB (NA)-0300534CHN-A	第一版
2021年09月	IB (NA)-0300534CHN-B	第二版 部分修改

日文原稿手册：IB-0300532-C

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。

© 2020 MITSUBISHI ELECTRIC CORPORATION

质保

1. 免费质保期限和免费质保范围

如果产品在免费质保期限内发生了因本公司责任而导致的故障或瑕疵(以下统称“故障”)时,本公司将通过销售商或本公司的售后服务公司免费对产品进行修理。但如果需要在国内或海外出差维修时,则要收取派遣技术人员的实际费用。此外,因故障部件的更换而发生的现场再调试、试运行不属于本公司责任范围。

[免费质保期限]

关于产品的免费质保期限,请向您的三菱产品销售商进行咨询。

[免费质保范围]

- (1) 首次故障诊断原则上由贵公司负责实施。但应贵公司要求,本公司或者本公司维修网点可有偿提供该项业务。此时,如果故障是由于本公司原因而导致的,则该项业务免费。
- (2) 仅限于使用状态・使用方法及使用环境等均遵照使用说明书、用户手册、产品本体注意标签等规定的条件・注意事项等,并在正常状态下使用的情况。
- (3) 即使在免费质保期限内,以下情况也要收取维修费用。
 - (i) 因客户保管或使用不当、疏忽、过失等引起的故障,以及因客户的硬件或软件设计内容引起的故障。
 - (ii) 因客户未经本公司允许对产品进行改造等而引起的故障。
 - (iii) 将本公司产品组合安装到用户的机器中时,如果用户的机器上安装了法规规定的安全装置或业界标准要求配备的功能和结构后即可避免的故障。
 - (iv) 如果正常维护、更换使用说明书中指定的消耗品即可避免的故障。
 - (v) 耗材(电池,风扇等)的更换。
 - (vi) 由于火灾、异常电压等不可抗力引起的外部因素以及因地震、雷电、风灾水灾等自然灾害引起的故障。
 - (vii) 根据从本公司出货时的科技标准还无法预知的原因而导致的故障。
 - (viii) 其他任何非本公司责任或客户认为非本公司责任的故障。

2. 产品停产后的有偿维修期限

- (1) 本公司在本产品停产后的7年内受理该产品的有偿维修。关于停产的消息将通过本公司销售和售后服务人员进行通告。
- (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,由本公司在当地的海外FA中心受理维修业务。但是,请注意各个FA中心的维修条件等可能会有所不同。

4. 机会损失和间接损失等不在质保责任范围内

无论是否在免费质保期内,本公司对于以下内容都不承担责任。

- (1) 非本公司责任的原因而导致的损失。
- (2) 因本公司产品故障而引起的用户机会损失、利润损失。
- (3) 无论本公司能否预测的特殊事件引起的损失和间接损失、事故赔偿、对本公司产品以外的损伤。
- (4) 用户更换设备、现场机械设备的再调试、运行测试及其他作业的赔偿。

5. 产品规格的更改

样本、手册或技术资料等所记载的规格如有变更,恕不另行通知。

6. 关于产品的适用范围

- (1) 在使用本公司运动模块时,应该符合以下条件:即使在运动模块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) 本公司运动模块是以一般工业用途等为对象设计和制造的通用产品。

因此,运动模块不适用于面向各电力公司的核电站以及其他发电厂等对公众有较大影响的用途、及面向各铁路公司或行政机关等要求构建特殊质量保证体系的用途。此外,运动模块也不适用于航空航天、医疗、铁路、焚烧・燃料装置、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。

但是,对于上述用途,在用户同意限定用途且无特殊质量要求的条件下,可对其适用性进行研究讨论,请与本公司服务窗口联系。
- (3) 对于由DoS攻击、非法访问、计算机病毒及其它网络攻击而导致发生的可编程控制器及系统故障方面的诸问题,三菱电机将不负责。

商标

Microsoft及Windows是美国Microsoft Corporation在美国及其它国家的注册商标或商标。

Unicode是Unicode, Inc. 在美国及其它国家的注册商标或商标。

PLCopen是PLCopen[®]拥有的注册商标。

本手册中的公司名、系统名和产品名等是相应公司的注册商标或商标。

本手册中，有时未标明商标符号(™、®)。

IB (NA) -0300534CHN-B (2109) MEACH

MODEL: RD78-P-MCFB-C

 **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：021-23223030 传真：021-23223000

网址：<http://cn.MitsubishiElectric.com/fa/zh/>

技术支持热线 **400-821-3030**



扫描二维码,关注官方微博



扫描二维码,关注官方微信

内容如有更改 恕不另行通知