

机器自动化控制器

NJ/NX系列

用户手册

软件篇

NX701-17□□

NX701-16□□

NJ501-□5□□

NJ501-□4□□

NJ501-□3□□

NJ301-12□□

NJ301-11□□

NJ101-10□□

声明

- 禁止誊写、复制或转载本手册的部分或全部内容。
- 本手册的内容可能因改良需要而有所变更，恕不另行通知，敬请谅解。
- 本手册的内容经严格审核，若发现不明之处或错误，请与本公司联系。联系时，请告知卷末记载的手册编号（Man.No.）。

商标

- Sysmac是欧姆龙株式会社FA机器产品在日本及其他国家的商标或注册商标。
- Microsoft及Windows、Windows Vista、Excel、Visual Basic是美国Microsoft Corporation在美国、日本及其他国家的注册商标或商标。
- EtherCAT®是德国倍福自动化有限公司提供的注册商标，是已获得专利的技术。
- ODVA、CIP、CompoNet、DeviceNet、EtherNet/IP是ODVA的商标。
- SD、SDHC标识是SD-3C, LLC的商标。



此外，记载的公司名称和产品名称为各公司的注册商标或商标。

著作权

Screen shot已取得Microsoft Corporation的使用许可。

前言

感谢您购买 NJ/NX 系列 CPU 单元。

本手册是运动控制指令的详细说明书。请对相关功能、操作方法等进行充分理解，正确使用运动控制功能模块（以下称作 MC 功能模块）。

使用时，请同时参考 NJ/NX 系列 CPU 单元的用户手册。

此外，请妥善保管本手册，阅读之后放于随时可取阅之处。

阅读对象

本手册提供给下列阅读对象。

具有电工专业知识的人员（合格的电气工程师或具有同等知识的人员）：

- 引进 FA 设备的人员；
- 设计 FA 系统的人员；
- 安装和连接 FA 设备的人员；
- FA 现场管理人员。

此外，编程语言的对象为掌握国际标准规格 IEC 61131-3 或日本国内标准规格 JIS B 3503 之规定内容的人员。

对象产品

本手册的对象为以下产品：

NX 系列 CPU 单元

- NX701-17□□
- NX701-16□□

NJ 系列 CPU 单元

- NJ501-□5□□
- NJ501-□4□□
- NJ501-□3□□
- NJ301-12□□
- NJ301-11□□
- NJ101-10□□

各产品的部分规格或限制事项可能在其他手册中有记载。□□请确认「分册构成 (P.2)」及 □□「相关手册 (P.24)」。

分册构成

本产品的手册如下表所示分册。请根据不同目的选读，灵活运用本产品。

本产品的操作主要使用自动化软件 Sysmac Studio。关于 Sysmac Studio，请参考 □ 《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

使用目的	手册											
	基本信息											
	NX 系列 CPU 单元 用户手册 硬件篇	NJ 系列 CPU 单元 用户手册 硬件篇	NJ/NX 系列 CPU 单元 用户手册 软件篇	NJ/NX 系列 指令基准手册 基本篇	NJ/NX 系列 CPU 单元 用户手册 运动控制篇	NJ/NX 系列 指令基准手册 运动篇	NJ/NX 系列 CPU 单元 内置 EtherCAT 端口 用户手册	NJ/NX 系列 CPU 单元 内置 EtherNet/IP 端口 用户手册	NJ 系列 数据库连接 CPU 单元 用户手册	NJ 系列 配备 SECS/GEM 的 CPU 单元 用户手册	NJ 系列 NJ Robotics CPU 单元 用户手册	NJ/NX 系列 故障排除手册
了解 NX 系列的概要	●											
了解 NJ 系列的概要		●										
进行安装、设置和硬件设定												
进行运动控制时					●							
使用 EtherCAT 时						●						
使用 EtherNet/IP 时	●	●					●					
使用数据库连接服务时								●				
使用 GEM 服务时									●			
进行机器人控制时										●		
进行软件设定												
进行运动控制时					●							
使用 EtherCAT 时						●						
使用 EtherNet/IP 时			●				●					
使用数据库连接服务时								●				
使用 GEM 服务时									●			
进行机器人控制时										●		
创建用户程序												
进行运动控制时					●	●						
使用 EtherCAT 时							●					
使用 EtherNet/IP 时								●				
使用数据库连接服务时			●	●					●			
使用 GEM 服务时										●		
进行机器人控制时											●	
组合异常处理时												●
进行动作确认、调试时												
进行运动控制时					●							
使用 EtherCAT 时							●					
使用 EtherNet/IP 时			●					●				
使用数据库连接服务时									●			
使用 GEM 服务时										●		
进行机器人控制时											●	

使用目的	手册											
	基本信息											
	NX 系列 CPU 单元 用户手册 硬件篇	NJ 系列 CPU 单元 用户手册 硬件篇	NJ/NX 系列 CPU 单元 用户手册 软件篇	NJ/NX 系列 指令基准手册 基本篇	NJ/NX 系列 CPU 单元 用户手册 运动控制篇	NJ/NX 系列 指令基准手册 运动篇	NJ/NX 系列 CPU 单元 内置 EtherCAT 端口 用户手册	NJ/NX 系列 CPU 单元 内置 EtherNet/IP 端口 用户手册	NJ 系列 数据库连接 CPU 单元 用户手册	NJ 系列 配备 SECS/GEM 的 CPU 单元 用户手册	NJ 系列 NJ Robotics CPU 单元 用户手册	NJ/NX 系列 故障排除手册
了解异常管理功能或故障发生时的处理方法 ^{*1}	△	△	△		△		△	△	△	△		●
了解维护作业												
进行运动控制时					●							
使用 EtherCAT 时	●	●					●					
使用 EtherNet/IP 时								●				

*1. 关于异常管理的说明和异常项目的概要，请参考 □ 《NJ/NX 系列 故障排除手册（SBCA-CN5-361）》。关于异常的详情，请根据异常的内容参考带 △ 标志的手册。

手册说明

页面构成

本手册各页面的构成及符号如下所示。

节标题 → **MC_Power**

项标题 → **变量**

手册名称 → NJ/NX 系列 命令参考手册 (运动控制) (SBCE-CN5-364)


章标题 → 3 轴指令

节标题 → 表示当前页的节标题。

章编号 → 表示当前页的章编号。

项标题 → 表示当前页的项标题。

注记、补充、参照处
以图形表示注记、补充、参照处等项目。



3 轴指令

MC_Power

将伺服驱动器切换为可运行状态。

指令	名称	FB/FUN	反馈取得	ST 表现
MC_Power	可运行	FB	MC_Power_instance Axis Enable	MC_Power_instance { Axis := 《参数》; Enable := 《参数》; Status => 《参数》; Busy => 《参数》; Error => 《参数》; ErrorID := 《参数》; };

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、FALSE	FALSE	若设为 TRUE，则变为可运行状态。若设为 FALSE，则解除可运行状态。

输出变量

输出变量	名称	数据类型	有效范围	内容
Status	可运行	BOOL	TRUE、FALSE	变为可运行状态时变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	"1"	发生异常时，输出错误代码。16#0000 表示正常。

*1: □ [A-1 错误代码一览 (P.A-2)]。

输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Status	指定轴变为可运行状态时	<ul style="list-style-type: none"> 指定的可运行状态解除时 Error 变为 TRUE 时
Busy	Enable 的上升沿	<ul style="list-style-type: none"> Enable 变为 FALSE 时 Error 变为 TRUE 时
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

3 - 3

3 轴指令

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	-	指定轴。*1

*1: 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称 (默认 "MC_Axis****") 或系统定义变量的轴变量名称 (_MC_AX[*]、_MC1_AX[*]、_MC2_AX[*])。

功能说明

- 将 Enable (有效) 设为 TRUE 后，Axis (轴) 中指定的轴将变为可运行状态。将轴设为可运行状态后，可进行轴控制。
- 将 Enable (有效) 设为 FALSE 后，将解除 Axis (轴) 中指定轴的可运行状态。解除了可运行状态时，轴将无法接收动作指令，不可进行轴控制。此外，对解除了可运行状态的轴发出的动作指令将发生异常。但是，在解除状态下，MC_Power (可运行) 指令、MC_Reset (错误重置) 指令仍可执行。
- 还可将动作中的轴切换为解除状态。此时，动作中的轴将变为 CommandAborted (执行中断)，停止输出动作指令，切换为解除状态。
- 如果使用绝对编码器的伺服电机，轴切换为可运行状态时，若处于原点未确定状态，将进行 [绝对编码器原点位置偏置] 补偿，变为原点确定状态。

在单元版本 1.10 以上的 CPU 单元中，EtherCAT 的过程数据通信从未建立状态变为建立状态时，也可变为原点确定状态。

[绝对编码器原点位置偏置] 请参考 □ [NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)]。

使用注意事项

- 本指令可用于轴种类为何服轴或虚拟何服轴的轴。用于编码器轴或虚拟编码器轴时，将发生异常。
- 对 NX 系列脉冲输出单元执行本指令时，本指令的功能不同。详情请参考 □ [NX 系列位置接口单元 用户手册 (SBCE-CN5-374)]。
- 对同步控制的主轴执行本指令时**

在垂直轴中，主轴从可运行状态解除时，主轴的位置可能发生剧烈变化。因此，从轴的动作也可能剧烈变化。

为避免从轴发生剧烈动作，请为主轴设置制动器或在结束同步控制后再解除主轴的运行状态。

3 - 4

NJ/NX 系列 命令参考手册 (运动控制) (SBCE-CN5-364)

(注) 本页为说明用样本。与实际内容不同。

图标

本手册中使用的图标含义如下。



安全要点

表示为了安全使用，应该实施或避免的行为。



使用注意事项

表示为了防止产品出现动作不良、误动作或严重影响其性能、功能，应该实施或避免的行为。



参考

需要时阅读的项目。
了解后有助于使用的信息以及使用时可参考的内容。



版本相关信息

介绍不同版本的 CPU 单元或 Sysmac Studio 的性能和功能区别。



表示记载有详细信息、相关信息的位置。

目录结构

		1
1	运动控制指令的概要	2
2	变量和指令	3
3	轴指令	4
4	轴组指令	5
5	共通指令	A
A	附录	I
I	索引	

目录

前言	1
分册构成	2
手册说明	4
目录结构	7
承诺事项	14
安全注意事项	16
安全要点	17
使用注意事项	18
法规和标准	19
版本	21
相关手册	24
手册修订记录	27

第 1 章 运动控制指令的概要

1-1 运动控制指令	1-2
PLCopen [®] 运动控制用功能块	1-2
运动控制指令的概要	1-3
同步控制的主轴及辅助轴的注意事项	1-6
1-2 运动控制指令的基础知识	1-8
运动控制指令的名称	1-8
运动控制指令的语言	1-8
运动控制指令的配置	1-9
运动控制指令的同时启动	1-18
运动控制指令的在线编辑	1-19
CPU 单元的动作模式切换	1-19

第 2 章 变量和指令

2-1 变量一览	2-2
MC 共通变量	2-3
轴变量	2-4
轴组变量	2-8
运动控制指令的输入变量	2-11
运动控制指令的输出变量	2-22
运动控制指令的输入输出变量	2-24
2-2 指令一览	2-26
共通指令	2-26
轴指令	2-27
轴组指令	2-29
2-3 PDO 映射	2-30
必须对象	2-30

不同指令的设定对象	2-32
-----------------	------

第 3 章 轴指令

MC_Power	3-3
变量	3-3
功能说明	3-4
MC_MoveJog	3-8
变量	3-8
功能说明	3-9
MC_Home	3-15
变量	3-15
功能说明	3-17
MC_HomeWithParameter	3-36
变量	3-36
功能说明	3-39
MC_Move	3-41
变量	3-41
功能说明	3-43
MC_MoveAbsolute	3-47
变量	3-47
功能说明	3-49
示例程序 1	3-57
示例程序 2	3-64
MC_MoveRelative	3-73
变量	3-73
功能说明	3-75
MC_MoveVelocity	3-80
变量	3-80
功能说明	3-82
示例程序	3-87
MC_MoveZeroPosition	3-95
变量	3-95
功能说明	3-97
MC_MoveFeed	3-102
变量	3-103
功能说明	3-106
示例程序	3-118
MC_Stop	3-129
变量	3-129
功能说明	3-131
MC_ImmediateStop	3-138
变量	3-138
功能说明	3-139
MC_SetPosition	3-143
变量	3-143
功能说明	3-145
MC_SetOverride	3-149
变量	3-149
功能说明	3-150
MC_ResetFollowingError	3-154
变量	3-154
功能说明	3-156
MC_CamIn	3-161
变量	3-161

功能说明	3-164
示例程序 1	3-187
示例程序 2	3-198
MC_CamOut	3-215
变量	3-215
功能说明	3-216
MC_GearIn	3-220
变量	3-220
功能说明	3-222
示例程序	3-229
MC_GearInPos	3-240
变量	3-240
功能说明	3-243
示例程序	3-250
MC_GearOut	3-261
变量	3-261
功能说明	3-263
MC_MoveLink	3-266
变量	3-266
功能说明	3-269
示例程序	3-278
MC_CombineAxes	3-288
变量	3-288
功能说明	3-290
MC_Phasing	3-299
变量	3-299
功能说明	3-301
MC_TorqueControl	3-305
变量	3-305
功能说明	3-307
MC_SetTorqueLimit	3-316
变量	3-316
功能说明	3-318
MC_ZoneSwitch	3-323
变量	3-323
功能说明	3-324
MC_TouchProbe	3-329
变量	3-329
功能说明	3-331
示例程序	3-343
MC_AbortTrigger	3-350
变量	3-350
功能说明	3-352
MC_AxesObserve	3-354
变量	3-354
功能说明	3-356
MC_SyncMoveVelocity	3-360
变量	3-360
功能说明	3-362
MC_SyncMoveAbsolute	3-369
变量	3-369
功能说明	3-371
MC_Reset	3-375
变量	3-375
功能说明	3-376

MC_ChangeAxisUse	3-379
变量	3-379
功能说明	3-381
MC_DigitalCamSwitch	3-383
变量	3-384
功能说明	3-385
示例程序	3-395
MC_TimeStampToPos	3-402
变量	3-402
功能说明	3-403
示例程序	3-407
MC_PeriodicSyncVariables	3-415
变量	3-415
功能说明	3-416
示例程序	3-420
MC_SyncOffsetPosition	3-423
变量	3-423
功能说明	3-425

第 4 章 轴组指令

MC_GroupEnable	4-2
变量	4-2
功能说明	4-3
MC_GroupDisable	4-6
变量	4-6
功能说明	4-7
MC_MoveLinear	4-11
变量	4-11
功能说明	4-14
示例程序	4-24
MC_MoveLinearAbsolute	4-37
变量	4-37
功能说明	4-39
MC_MoveLinearRelative	4-40
变量	4-40
功能说明	4-42
MC_MoveCircular2D	4-43
变量	4-43
功能说明	4-46
示例程序	4-55
MC_GroupStop	4-67
变量	4-67
功能说明	4-69
MC_GroupImmediateStop	4-75
变量	4-75
功能说明	4-76
MC_GroupSetOverride	4-79
变量	4-79
功能说明	4-80
MC_GroupReadPosition	4-83
变量	4-83
功能说明	4-85
MC_ChangeAxesInGroup	4-87
变量	4-87

功能说明	4-89
MC_GroupSyncMoveAbsolute	4-91
变量	4-91
功能说明	4-93
MC_GroupReset	4-97
变量	4-97
功能说明	4-98

第 5 章 共通指令

MC_SetCamTableProperty	5-2
变量	5-2
功能说明	5-3
MC_SaveCamTable	5-8
变量	5-8
功能说明	5-10
MC_Write	5-13
变量	5-13
功能说明	5-16
MC_GenerateCamTable	5-18
变量	5-18
功能说明	5-20
示例程序	5-33
MC_WriteAxisParameter	5-45
变量	5-45
功能说明	5-47
MC_ReadAxisParameter	5-58
变量	5-58
功能说明	5-60

附录

A-1 错误代码一览	A-2
A-2 错误代码详情	A-22
A-2-1 控制器异常说明的解释	A-22
A-2-2 异常说明	A-23
A-3 多重启动可否	A-88
A-3-1 轴的状态、轴组的状态	A-89
A-3-2 状态切换和多重启动的可否	A-91
A-4 版本相关信息	A-96

索引

承诺事项

关于“本产品”，若无特殊协议，无论客户从何处购买，均适用本承诺事项中的条件。

● 定义

本承诺事项中术语的定义如下所示。

- “本公司产品”：“本公司”的 FA 系统设备、通用控制设备、传感设备、电子和机械零件
- “产品样本等”：与“本公司产品”相关的欧姆龙工控设备、电子和机械零件综合样本、其他产品样本、规格书、使用说明书、手册等，还包括通过电磁介质提供的资料。
- “使用条件等”：“产品样本等”中的“本公司产品”的使用条件、额定值、性能、运行环境、使用方法、使用注意事项、禁止事项等
- “用户用途”：用户使用“本公司产品”的方法，包括直接使用或将“本公司产品”装入用户制造的零件、印刷电路板、机械、设备或系统等。
- “适用性等”：“用户用途”中“本公司产品”的（a）适用性、（b）动作、（c）不侵犯第三方知识产权、（d）遵守法律以及（e）遵守各种标准

● 记载内容的注意事项

关于“产品样本等”中的内容，请注意以下几点。

- 额定值和性能值是在各条件下进行单独试验后获取的值，并不保证在复合条件下可获取各额定值和性能值。
- 参考数据仅供参考，并不保证在该范围内始终正常运行。
- 使用实例仅供参考，“本公司”不保证“适用性等”。
- “本公司”可能会因产品改良、本公司的原因而中止“本产品”的生产或变更“本产品”的规格。

● 使用注意事项

使用时，请注意以下几点。

- 使用时请符合额定值、性能以及“使用条件等”。
- 请用户自行确认“适用性等”，判断是否可使用“本公司产品”。“本公司”对“适用性等”不作任何保证。
- 用户将“本公司产品”用于整个系统时，请务必事先自行确认配电、设置是否恰当。
- 使用“本公司产品”时，请注意以下各事项。（i）使用“本公司产品”时，应在额定值和性能方面留有余量，采用冗余设计等安全设计，（ii）采用安全设计，即使“本公司产品”发生故障，也可将“用户用途”造成的危险降至最低程度，（iii）对整个系统采取安全措施，以便向使用者告知危险，（iv）定期维护“本公司产品”及“用户用途”。
- 本公司设计并制造面向一般工业产品的通用产品。但是，不可用于以下用途。如果用户将“本公司产品”用于以下用途，则“本公司”不对“本公司产品”作任何保证。但如果属于本公司许可的特别产品用途或与本公司签订特殊协议的场合除外。
 - （a）需高安全性的用途（例：原子能控制设备、燃烧设备、航空航天设备、铁路设备、起重设备、娱乐设备、医疗设备、安全装置以及其他危及生命、健康的用途）

- (b) 需高可靠性的用途（例：煤气、自来水、电力等供应系统、24小时连续运行的系统、支付系统等涉及权利、财产的用途等）
- (c) 用于严格条件或环境下（例：需设置在室外的设备、会受化学污染的设备、会受电磁波干扰的设备、会受振动、冲击影响的设备等）
- (d) 在“产品样本等”中未记载的条件或环境下使用
- 上述（a）～（d）以及“本产品样本等中记载的产品”不可用于汽车（含两轮车。下同）。请勿装入汽车进行使用。关于可装入汽车的产品，请咨询本公司销售负责人。

● 保修条件

“本产品”的保修条件如下所述。

- 保修期为购买本产品后的1年内。
（“产品样本等”中另有记载的情况除外。）
- 保修内容 对发生故障的“本公司产品”，经“本公司”判断后提供以下任一服务。
 - (a) 发生故障的“本公司产品”可在本公司维修服务网点免费维修
（不提供电子和机械零件的维修服务。）
 - (b) 免费提供与发生故障的“本公司产品”数量相同的替代品
- 非保修范围 如果因以下任一原因造成故障，则不在保修范围内。
 - (a) 用于非“本公司产品”原本用途的用途时
 - (b) 未按“使用条件等”进行使用
 - (c) 违反本承诺事项中的“使用注意事项”进行使用
 - (d) 改造或维修未经“本公司”
 - (e) 使用的软件程序非由“本公司”人员编制
 - (f) 因以出厂时的科学技术水平无法预见的原因
 - (g) 除上述以外，因“本公司”或“本公司产品”以外的原因（包括自然灾害等不可抗力）

● 责任免除

本承诺事项中的保修即与“本公司产品”相关的保修的所有内容。




对因“本公司产品”造成的损害，“本公司”及“本公司产品”的销售店概不负责。

● 出口管理

出口“本公司产品”或技术资料或向非居民的人员提供时，应遵守日本及各国安全保障贸易管理相关的法律法规。如果用户违反上述法律法规，则可能无法向其提供“本公司产品”或技术资料。




安全注意事项

关于安全注意事项，请参考以下手册。

-  《NX 系列 CPU 单元用户手册 硬件篇（SBCA-CN5-418）》
-  《NJ 系列 CPU 单元 用户手册 硬件篇（SBCA-CN5-358）》
-  《NJ/NX 系列 CPU 单元 用户手册 软件篇（SBCA-CN5-359）》




安全要点

关于安全要点，请参考以下手册。

-  《NX 系列 CPU 单元用户手册 硬件篇（SBCA-CN5-418）》
-  《NJ 系列 CPU 单元 用户手册 硬件篇（SBCA-CN5-358）》
-  《NJ/NX 系列 CPU 单元 用户手册 软件篇（SBCA-CN5-359）》

使用注意事项

关于使用注意事项，请参考以下手册。

-  《NX 系列 CPU 单元用户手册 硬件篇（SBCA-CN5-418）》
-  《NJ 系列 CPU 单元 用户手册 硬件篇（SBCA-CN5-358）》
-  《NJ/NX 系列 CPU 单元 用户手册 软件篇（SBCA-CN5-359）》

法规和标准

日本国外的使用

出口（或向非居住者提供）本产品中符合外汇及外国贸易法规定的出口许可、批准对象货物（或技术）要求的产品时，须依照该法获得出口许可、批准（或劳务交易许可）。

符合 EU 指令

适用指令

- EMC 指令
- 低电压指令

概念

● EMC 指令

由于 OMRON 公司的产品为安装在各种机械、制造装置中使用的电气设备，为了使安装 PLC 的机械、装置更容易符合 EMC 标准，必须先使产品本身达到相关 EMC 标准*。

但是，由于客户使用的机械、装置各不相同，EMC 性能因符合 EU 指令的产品安装的设备和控制柜的构成、配线及配置等条件而异，无法在客户使用状态下确认其适合性。因此，机械、装置整体最终的 EMC 适用性确认请客户自行实施。

* EMC（Electro-Magnetic Compatibility：电磁兼容性）相关标准中，
EMS（Electro-Magnetic Susceptibility：电磁敏感度）应符合 EN61131-2、
EMI（Electro-Magnetic Interference：电磁干扰）应符合 EN61131-2
此外，Radiated emission 基于 10m 法。

● 低电压指令

对于以电源电压 50V AC ~ 1000V AC 以及 75V DC ~ 1500V DC 工作的设备，要求必须确保必要的安全性。适用标准为 EN61010-2-201。

● 符合 EU 指令

NJ/NX 系列符合 EU 指令。为确保客户使用的机械、装置符合 EU 指令，必须注意下列几点。

- NJ/NX 系列 PLC 必须安装在控制柜内。
- 连接到 DC 电源单元和 I/O 单元的 DC 电源，请使用 SELV 规格的电源。
- NJ/NX 系列同时符合 EC 指令和 EMI 相关的通用发射标准。特别是 Radiated emission(10m 法)，因控制柜的配置、连至控制柜的其它设备、接线和其它条件而异。
因此，即使使用符合 EU 指令的 NJ/NX 系列时，客户也必须确认机械、装置整体是否符合 EU 指令。

KC 标准的适用

在韩国使用本产品时，请遵守以下注意事项。

A 급 기기 (업무용방송통신기자재)

이 기기는 업무용(A 급) 전저파작합기기로서 판매자 또는 사용자는 이 점을 주의하시기 바라며, 가정외의 지역에서 사용하는 것을 목적으로 합니다.

ClassA 设备 (商用广播通信设备)

本设备为商用电磁波发生设备 (ClassA)，设计用于家庭以外的场所。经销商和用户应注意以下内容。

船舶标准的适用

本产品符合各类船舶标准。适用各类船舶标准时，设定了使用条件，根据安装场所不同，可能无法使用。使用时，请务必与本公司联系。

各船舶标准的使用条件 (NK、LR)

- 本产品必须安装在控制柜内。
- 控制柜开关口的缝隙应用垫片等完全密封。
- 电源线应与以下噪声滤波器连接。

噪声滤波器

厂家	型号
COSEL	TAH-06-683

软件许可和著作权

本产品中组装有第三方开发的软件。关于软件的许可和著作权，请参考 http://www.fa.omron.co.jp/nj_info_j/。

版本

NJ/NX 系列的各单元及各 EtherCAT 从站的硬件或软件按照硬件修订版本或单元版本等编号管理。每次硬件或软件发生规格变更后，硬件修订版本或单元版本都会更新。因此，即使是同一型号的单元或 EtherCAT 从站，若硬件修订版本或单元版本不同，配备的功能或性能可能不同。

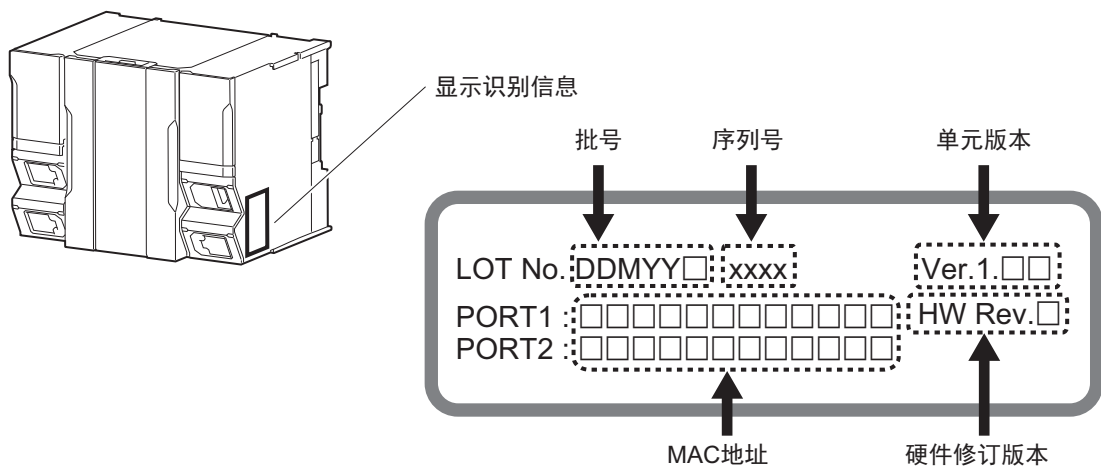
版本的确认方法

版本可通过标示的识别信息或 Sysmac Studio 确认。

通过标示的识别信息确认

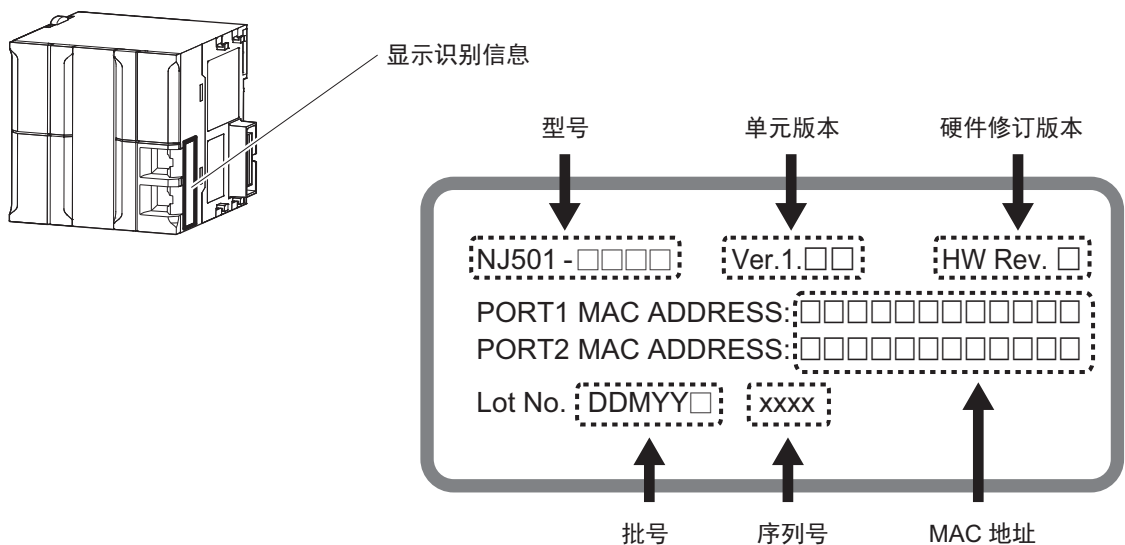
可通过产品侧面标示的识别信息确认版本。

NX 系列 CPU 单元 NX701-□□□□ 的识别信息标示如下图所示。



(注) 硬件修订版本为“无”的单元中，不标示硬件修订版本。

NJ 系列 CPU 单元 NJ501-□□□□ 的识别信息标示如下图所示。



(注) 硬件修订版本为“无”的单元中，不标示硬件修订版本。

通过 Sysmac Studio 确认的方法

可通过 Sysmac Studio 确认版本。单元和 EtherCAT 从站的确认方法不同。

● NX 系列 CPU 单元的版本确认方法

单元版本可通过在线状态的 [生产信息] 确认。但是，只能确认 CPU 单元的版本。

- 1 在多视图浏览器中右击 [配置和设定] - [CPU·扩展装置] - [CPU 装置]，选择 [显示生产信息]。
显示 [生产信息] 对话框。

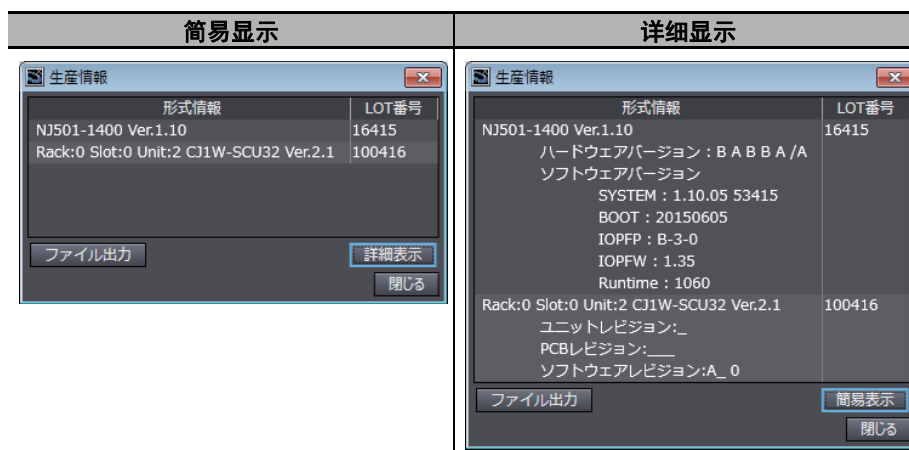
● NJ 系列 CPU 单元的版本确认方法

单元版本可通过在线状态的 [生产信息] 确认。但是，只能确认 CPU 单元、CJ 系列高性能 I/O 单元及 CJ 系列 CPU 总线单元的版本。CJ 单元的基本 I/O 单元的版本无法通过 Sysmac Studio 确认。

- 1 在多视图浏览器中双击 [配置和设定] - [CPU·扩展装置]。或者右击 [配置和设定] - [CPU·扩展装置]，然后选择 [编辑]。
显示单元编辑器。
- 2 右击单元编辑器的空白栏，选择 [显示生产信息]。
显示 [生产信息] 对话框。

● 切换生产信息的显示内容

- 1 选择 [生产信息] 对话框右下方的 [简易显示] 或 [详细显示]。
切换 [生产信息] 的简易显示和详细显示。



显示内容因简易显示还是详细显示而异。详细显示中，显示单元版本、硬件版本及软件版本。简易显示中，仅显示单元版本。

(注) 硬件修订版本在硬件版本的右端以“/”分隔显示。硬件修订版本为“无”的单元中，不标示硬件修订版本。

● EtherCAT 从站的版本确认方法

EtherCAT 从站的版本可通过在线状态的 [生产信息] 确认。确认方法如下所示。

- 1 在多视图浏览器中双击 [配置和设定] - [EtherCAT]。或者右击 [配置和设定] - [EtherCAT]，然后选择 [编辑]。

显示 EtherCAT 配置编辑画面。

- 2 右击 EtherCAT 配置编辑画面中的主机，选择 [显示生产信息]。

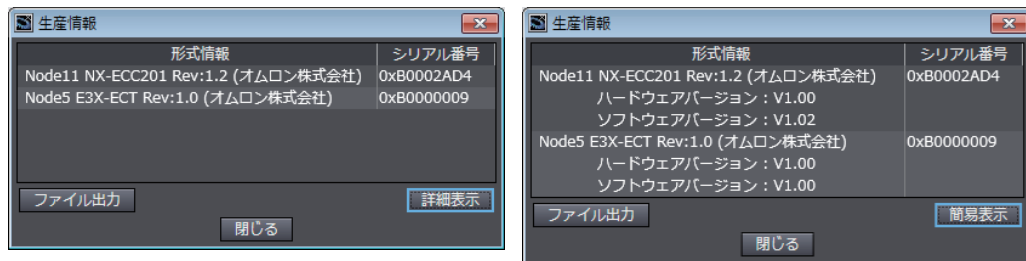
显示生产信息对话框。

单元版本以带“Rev”的形式显示。

● 切换生产信息的显示内容

选择 [生产信息] 对话框右下方的 [简易显示] 或 [详细显示]。

切换 [生产信息] 的简易显示和详细显示。



简易显示

详细显示

CPU 单元的单元版本和 Sysmac Studio 的版本

NJ/NX 系列 CPU 单元的单元版本不同，所配备的功能也不同。使用升级后添加的功能时，需要使用对应版本的 Sysmac Studio。

CPU 单元的单元版本的种类和 Sysmac Studio 的版本的的关系、以及

单元版本的支持功能一览表，请参考 □□ 《NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCE-CN5-359)》。

相关手册

本手册相关的手册如下表所示。请同时参考。

手册名称	Man.No.	型号	用途	内容
NX 系列 CPU 单元 用户手册 硬件篇	SBCA-CN5-418	NX701-□□□□	希望了解 NX 系列 CPU 单元的概要 / 设计 / 安装 / 保养等基本规格时。 主要是硬件相关的信息。	对 NX 系列的系统整体概要及 CPU 单元相关的如下内容进行说明。 • 特长及系统构成 • 概要 • 各部分的名称和功能 • 一般规格 • 安装和配线 • 维护检查 请同时使用用户手册 软件篇 (SBCA-CN5-359)。
NJ 系列 CPU 单元 用户手册 硬件篇	SBCA-CN5-358	NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	希望了解 NJ 系列 CPU 单元的概要 / 设计 / 安装 / 保养等基本规格时。 主要是硬件相关的信息。	对 NJ 系列的系统整体概要及 CPU 单元相关的如下内容进行说明。 • 特长及系统构成 • 概要 • 各部分的名称和功能 • 一般规格 • 安装和配线 • 维护检查 请同时使用用户手册 软件篇 (SBCA-CN5-359)。
NJ/NX 系列 CPU 单元 用户手册 软件篇	SBCA-CN5-359	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	希望了解 NJ/NX 系列 CPU 单元的编程 / 系统启动时。 与软件相关的信息为主。	对 NJ/NX 系列 CPU 单元相关的如下内容进行说明。 • CPU 单元的动作 • CPU 单元的功能 • 初始设定 • 基于 IEC 61131-3 的语言规格和编程 请同时使用用户手册 硬件篇 (SBCA-CN5-418 或 SBCA-CN5-358)。
NJ/NX 系列指令基准手册 基本篇	SBCA-CN5-360	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	希望了解 NJ/NX 系列的基本指令规格详情时。	对各指令 (IEC 61131-3 规格) 的详情进行说明。 编程时请同时使用用户手册 硬件篇 (SBCA-CN5-418 或 SBCA-CN5-358) / 软件篇 (SBCA-CN5-359)。
NJ/NX 系列 CPU 单元 用户手册 运动控制篇	SBCE-CN5-363	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	希望了解运动控制的设定及编程思路时。	对用于运动控制的 CPU 单元的设定、动作及编程思路进行说明。 编程时请同时使用用户手册 硬件篇 (SBCA-CN5-418 或 SBCA-CN5-358) / 软件篇 (SBCA-CN5-359)。
NJ/NX 系列指令基准手册 运动篇	SBCE-CN5-364	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	希望了解运动指令规格的详情时。	对各运动指令的详情进行说明。 编程时请同时使用用户手册 硬件篇 (SBCA-CN5-418 或 SBCA-CN5-358) / 软件篇 (SBCA-CN5-359)、用户手册 运动控制篇 (SBCE-CN5-363)。
NJ/NX 系列 CPU 单元 内置 EtherCAT® 端口 用户手册	SBCD-CN5-358	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	使用 NJ/NX 系列 CPU 单元的内置 EtherCAT 端口时。	对内置 EtherCAT 端口进行说明。 对概要、构成、功能、安装进行说明。 请同时使用用户手册 硬件篇 (SBCA-CN5-418 或 SBCA-CN5-358) / 软件篇 (SBCA-CN5-359)。

手册名称	Man.No.	型号	用途	内容
NJ/NX 系列 CPU 单元 内置 EtherNet/IP™ 端口 用户手册	SBCD-CN5 -359	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	使用 NJ/NX 系列 CPU 单元的内置 EtherNet/IP 端口。	对内置 EtherNet/IP 端口进行说明。 对基本设定、标签数据链接、其它功能进行说明。 请同时使用用户手册 硬件篇 (SBCA-CN5-418 或 SBCA-CN5-358) / 软件篇 (SBCA-CN5-359)。
NJ 系列 数据库连接 CPU 单元 用户手册	SBCA-CN5 -411	NJ501-1 □ 20	在 NJ 系列中使用数 据库连接服务功能 时。	对数据库连接服务功能进行说明。
NJ 系列 配备 SECS/GEM 的 CPU 单元 用户手册	SBCA-CN5 -412	NJ501-1340	在 NJ 系列中使用 GEM 服务功能时。	对 GEM 服务的功能进行说明。
NJ 系列 NJ Robotics CPU 单元 用户手册	SBCA-CN5 -421	NJ501-4□□□	用 NJ 系列进行机器 人控制时。	对机器人控制功能进行说明。
NJ/NX 系列 故障排除手册	SBCA-CN5 -361	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	希望了解 NJ/NX 系列 检出的异常详情时。	对 NJ/NX 系列系统检出的异常的管理思路和各异常项目进行说明。 请同时使用用户手册 硬件篇 (SBCA-CN5-418 或 SBCA-CN5-358) / 软件篇 (SBCA-CN5-359)。
Sysmac Studio Version 1 操作手册	SBCA-CN5 -362	SYSMAC -SE2 □□□	希望了解 Sysmac Studio 的操作方法、 功能时。	对 Sysmac Studio 的操作方法进行说明。
NX 系列 EtherCAT® 耦合器单元 用户手册	SBCD-CN5 -361	NX-ECC □□□	希望了解 NX 系列 EtherCAT 耦合器单元 及 EtherCAT 从站终 端的使用方法时。	对 NX 系列 EtherCAT 耦合器单元和 NX 单元构成的 EtherCAT 从站终端的系统概要、配置方法, 以及通过 EtherCAT 对 NX 单元进行设定、控制、监视所需的 EtherCAT 耦合器单元的硬件、设定方法和功能进行说明。
NX 系列 NX 单元 用户手册	SBCA-CN5 -407	NX-ID □□□□ NX-IA □□□□ NX-OC □□□□ NX-OD □□□□	希望了解 NX 单元的 使用方法时。	对 NX 单元的硬件、设定方法和功能进行说明。 包括以下单元的手册。 数字 I/O 单元、模拟量 I/O 单元、系统单元、 位置接口单元
	SBCA-CN5 -408	NX-AD □□□□ NX-DA □□□□ NX-TS □□□□		
	SBCA-CN5 -409	NX-PD1 □□□ NX-PF0 □□□ NX-PC0 □□□ NX-TBX □□		
	SBCE-CN5 -374	NX-EC0 □□□□ NX-ECS □□□□ NX-PG0 □□□□		
NX 系列 数据基准手册	SBCA-CN5 -410	NX-□□□□□□	希望列表查看 NX 系 列各单元的系统构建 所需的数据时。	汇总记载 NX 系列各单元的消费电力、质量等 系统构建所需的数据。
EtherCAT 远程 I/O 终端 GX 系列 EtherCAT 从站 用户手册	SBCD-CN5 -350	GX-ID □□□□ GX-OD □□□□ GX-OC □□□□ GX-MD □□□□ GX-AD □□□□ GX-DA □□□□ GX-EC □□□□ XWT-ID □□ XWT-OD □□	希望了解 EtherCAT 远程 I/O 终端的使用 方法时。	对 EtherCAT 远程 I/O 终端的硬件、设定方法 和功能进行说明。

手册名称	Man.No.	型号	用途	内容
AC 伺服电机 / 驱动器 G5 系列 EtherCAT® 通信内置型 用户手册	SBCE-CN5 -360	R88M-K □ R88D-KN □ -ECT-R	希望了解 EtherCAT 通信内置型伺服电机 / 驱动器的使用方法 时。	对 EtherCAT 通信内置型伺服电机 / 驱动器的 硬件、设定方法和功能进行说明。 G5 系列包括直线电机型和位置控制限定机 型。
	SBCE-CN5 -365	R88M-K □ R88D-KN □ -ECT		
	SBCE-CN5 -366	R88L-EC- □ R88D-KN □ -ECT-L		

手册修订记录

手册修订符号在封面和封底中追加记载的 Man.No. 末尾标注。

例

Man.No.	SBCE-CN5-364M
----------------	----------------------

↑
修订符号

修订符号	修订日期	修订内容和修订页码
A	2011 年 7 月	第一版
B	2012 年 1 月	以下的轴组指令的追加 <ul style="list-style-type: none"> • MC_GroupReadPosition (轴组位置获取) • MC_ChangeAxesInGroup (轴组构成轴写入) • MC_GroupSyncMoveAbsolute (轴组周期性同步绝对位置控制) 误记修正
C	2012 年 5 月	因 CPU 单元版本升级到 1.02 发生的修订 误记修正
D	2012 年 8 月	因 CPU 单元版本升级到 1.03 发生的修订
E	2013 年 2 月	因 CPU 单元版本升级到 1.04 发生的修订
F	2013 年 4 月	因 CPU 单元版本升级到 1.05 发生的修订 误记修正
G	2013 年 6 月	因 CPU 单元版本升级到 1.06 发生的修订 误记修正
H	2013 年 12 月	因 CPU 单元版本升级到 1.08 发生的修订 误记修正
J	2014 年 7 月	因 CPU 单元版本升级到 1.09 发生的修订 误记修正
K	2015 年 1 月	因 CPU 单元版本升级到 1.10 发生的修订 误记修正
L	2015 年 4 月	因追加 NX 系列 CPU 单元 NX701-□□□□、NJ 系列 CPU 单元 NJ101-10□□ 发生的修订 误记修正
M	2016 年 4 月	因 CPU 单元版本升级到 1.11 发生的修订 误记修正

运动控制指令的概要

下面介绍 NJ/NX 系列 CPU 单元中内置的运动控制指令的概要。

1-1 运动控制指令	1-2
PLCopen® 运动控制用功能块	1-2
运动控制指令的概要	1-3
同步控制的主轴及辅助轴的注意事项	1-6
1-2 运动控制指令的基础知识	1-8
运动控制指令的名称	1-8
运动控制指令的语言	1-8
运动控制指令的配置	1-9
运动控制指令的同时启动	1-18
运动控制指令的在线编辑	1-19
CPU 单元的动作模式切换	1-19

1-1 运动控制指令

若要通过 NJ/NX 系列的用户程序执行运动控制功能，需要使用定义为功能块的运动控制指令。

MC 功能模块的运动控制指令以 PLCopen[®] 的运动控制功能块技术规格为基础。

运动控制指令分为 PLCopen[®] 中定义的指令和 MC 功能模块独立的指令两种。

下面介绍 PLCopen[®] 运动控制用功能块的概要和运动控制指令的概要。

运动控制指令的详情请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

在 CPU 单元版本 1.05 以上和 Sysmac Studio Ver.1.06 以上的组合中，可使用 NX 系列位置接口单元。

使用 NX 系列位置接口单元时请参考 □ 《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

PLCopen[®] 运动控制用功能块

PLCopen[®] 通过对运动控制用功能块进行标准化，定义了 IEC 61131-3 (JIS B 3503) 标准语言的程序界面。

除了单轴定位、电子凸轮功能以及多轴插补控制等功能，还对启动指令所需的基本步骤做出定义。

利用 PLCopen[®] 运动控制功能块，可减少硬件的依赖，提高用户程序的重复利用性。

此外，还可减少培训和支持的费用。



参考

什么是 PLCopen[®]

PLCopen[®] 为总部位于欧洲的 IEC 61131-3 普及团体，是一个全球性会员组织。

IEC 61131-3 为 PLC 编程的国际标准规格。

PLCopen[®] Japan 为该组织在日本市场的推广委员会，由关注日本市场的人员组成。

- PLCopen[®] Japan 网站 <http://www.plcopen-japan.jp/>
- PLCopen[®] 欧洲总部网站 <http://www.plcopen.org/>

运动控制指令的概要

下面介绍 PLCopen[®] 运动控制用功能块技术规格中定义的项目和 MC 功能模块的概要。

运动控制指令的种类

下表为运动控制指令的种类和概要。

种类	种类	功能项目	概要
共通指令	共通的管理类指令	凸轮表	该指令可控制 MC 功能模块的共通状态、操作或监视各种数据。
		参数	
轴指令	单轴的动作类指令	单轴位置控制	使轴进行单轴动作的指令。
		单轴速度控制	
		单轴扭矩控制	
		单轴同步控制	
		单轴手动操作	
	单轴的管理类指令	单轴控制辅助	控制和监视轴状态的指令。
轴组指令	多轴的动作类指令	多轴协调控制	使轴组进行协调动作的指令。
	多轴的管理类指令	多轴协调控制辅助	控制和监视轴组状态的指令。

状态变化

对轴的状态、轴组的状态以及启动指令后的状态变化进行定义。

MC 功能模块的状态和状态变化请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

运动控制指令的启动和状态

对启动指令的变量和表示指令执行状态的变量作出定义，作为指令的共通规则。

启动指令所需的输入变量有 Execute 和 Enable 两种。

表示指令执行状态的输出变量有 Busy、Done、CommandAborted、Error 等。

MC 功能模块的详细规格请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。



使用注意事项

手册中记载的时序图可能与 Sysmac Studio 的数据追踪表中显示的时序不同。

数据追踪的详情请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCE-CN5-359)》。

异常处理

MC 功能模块中的运动控制通过运动控制指令来执行。执行运动控制指令时，对输入参数的错误或指令处理引起的异常进行检查。

指令中有异常时，指令的输出变量 Error（错误）将变为 TRUE，并在 ErrorID（错误代码）中输出异常代码。

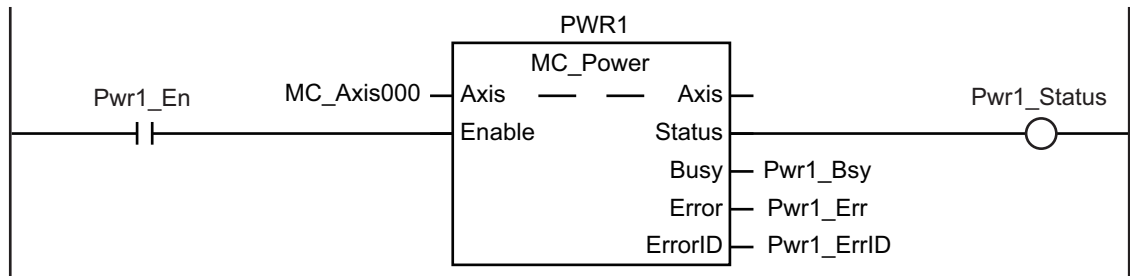
运动控制指令中，可用两种方法创建异常处理程序。

● 各指令的异常处理

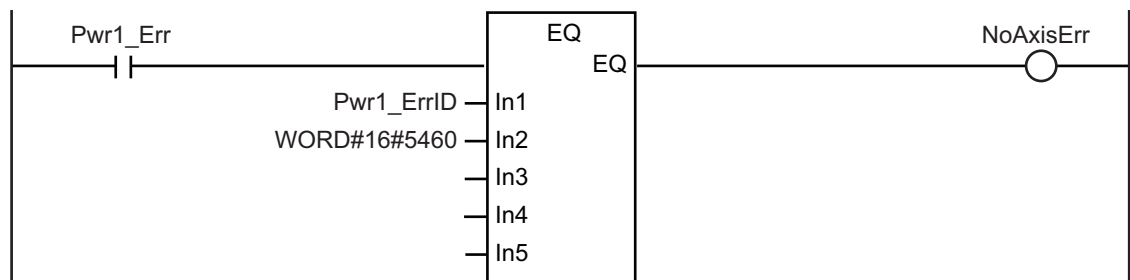
使用指令的输出变量 Error（错误）或 ErrorID（错误代码），可对发生异常的各个指令进行异常处理。

下面表示实例名称为“PWR1”的指令发生“轴指定不正确”时的示例。程序设计为在“NoAxisErr”TRUE时执行异常处理。

设为伺服 ON



确认指定的轴是否存在

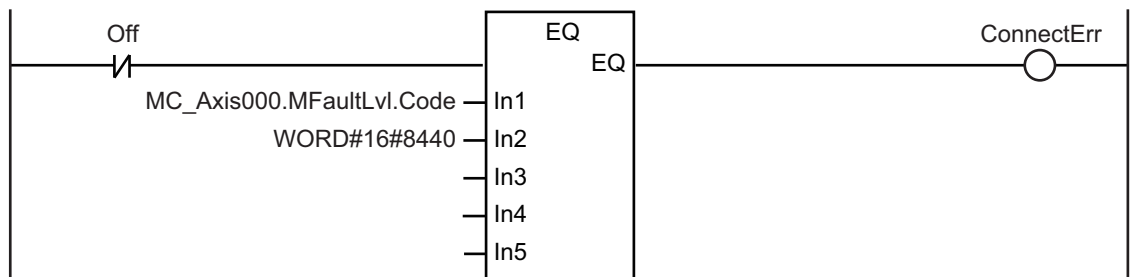


● 各类型的异常处理

使用运动控制系统变量中的异常状态，可按异常的种类进行异常处理。

下面表示“MC_Axis000”的轴发生“从站通信异常”时的示例。程序设计为在“ConnectErr”TRUE时执行异常处理。

确认 CPU 单元和伺服驱动器间的通信异常



正在执行运动控制指令时输入变量的变更（指令重新启动）

在执行运动控制指令的过程中，若变更同一个指令实例的输入变量值，并再次将 Execute 设为 TRUE，将按变更后的值动作。

MC 功能模块的指令重新启动请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

通过选择缓冲模式进行指令多重启动

动作过程中可执行其他的指令实例。

通过设定缓冲模式选择（BufferMode）这一输入，可指定开始动作的时间。

缓冲模式选择（BufferMode）中包含以下模式。

- 中断 : 中断 (Aborting)
- 缓冲 : 待机 (Buffered)
- 低共混 : 以低速度同时存在 (BlendingLow)
- 前共混 : 以前速度同时存在 (BlendingPrevious)
- 后共混 : 以下一速度同时存在 (BlendingNext)
- 高共混 : 以高速度同时存在 (BlendingHigh)

在中断模式下，将中断其他动作后立即动作。

在其他缓冲模式下，将保持待机，直至执行中指令的输出变量 Done 或 InVelocity 变为 TRUE。

在缓冲（待机）模式下，执行中指令的动作正常结束（Done = TRUE）后，执行下一指令。

在共混中，无需停止 2 个指令的动作，可连续执行。连续时的速度可从 4 种缓冲模式中选择。

MC 功能模块中，通过缓冲模式选择（BufferMode）进行的指令多重启动单独表述为指令多重启动并说明。

MC 功能模块的指令多重启动请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

MC 功能模块上是否可进行指令多重启动，取决于当前轴的状态或当前轴组的状态，以及所执行指令的种类。它们的关系详情请参考 □□「A-3 多重启动可否 (P.A-88)」。

运动控制中使用的结构体

PLCopen[®] 技术资料中，运动控制所需的信息定义为结构体。对数据类型的名称和概要进行定义，但未对结构体的内容作出定义。

PLCopen[®] 中的主要数据类型和 MC 功能模块中使用的数据类型如下。

数据类型		定义
PLCopen [®]	MC 功能模块	
AXIS_REF	_sAXIS_REF	包含对应轴信息的结构体。
AXES_GROUP_REF	_sGROUP_REF	包含对应轴组信息的结构体。
TRIGGER_REF	_sTRIGGER_REF	包含触发输入相关信息的结构体。 • 触发的指定 • 检测模式信息（正、负、两者、边缘、等级、模式识别等）
INPUT_REF	—	包含输入规格相关信息的结构体。也可能是虚拟数据。 MC 功能模块中不使用。
OUTPUT_REF	_sOUTPUT_REF	物理输出相关的结构体。 MC 功能模块可在 CPU 单元版本 1.06 以上和 Sysmac Studio Ver.1.07 以上的组合中使用。

MC 功能模块中，如上表所述，使用 PLCopen[®] 中定义的部分数据类型和 MC 功能模块独立定义的数据类型。

MC 功能模块中处理的数据类型和结构体的定义请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

同步控制的主轴及辅助轴的注意事项

下面记载同步控制的主轴及辅助轴在速度剧烈变化时的注意事项和发生异常的条件。

速度剧烈变化时

开始同步动作时或同步动作过程中，主轴及辅助轴的速度发生剧烈变化时，可能从轴的动作也会剧烈变化，对装置造成过大的力。

以下条件时，主轴及辅助轴的速度可能发生剧烈变化，敬请注意。

- 对主轴或辅助轴执行了以下 4 个指令时

MC_ImmediateStop（立即停止）指令

MC_SetPosition（当前位置变更）指令

MC_ResetFollowingError（偏差计数器复位）指令

MC_SyncMoveVelocity（周期性同步速度控制）指令

为避免从轴发生剧烈动作，请将上述指令的输入参数或启动时间设定为合适的值，或在解除同步控制后在启动上述指令。

- 主轴或辅助轴的立即停止输入信号、极限停止输入信号变为 TRUE 时

- 主轴或辅助轴从伺服 ON 状态变为伺服 OFF 状态时
如果主轴或辅助轴为垂直轴，变为伺服 OFF 状态时，可能发生剧烈变化。
为避免从轴发生剧烈动作，请对主轴或辅助轴加装制动器，或在结束同步控制后，再设为伺服 OFF 状态。
- 切换伺服驱动器的控制模式时
请注意启动指令时的速度变化。
请将指令的输入参数设定为合适的值。



版本相关信息

在单元版本 1.10 以上的 CPU 单元中，即使使用 MC_SetPosition（当前位置变更）指令变更同步控制指令的主轴或辅助轴的指令当前位置，从轴的动作也不会发生剧烈变化。

发生异常的条件

分为 NJ/NX 系列 CPU 单元共通的条件和只适用 NX 系列 CPU 单元的条件。

● NJ/NX 系列 CPU 单元共通的条件

开始同步动作时或同步动作中，主轴及辅助轴发生以下 4 个条件时，从轴将发生“主轴位置读取异常”或“辅助轴位置读取异常”。

同时，同步控制指令的输出变量“CommandAborted（执行中断）”变为 TRUE。

- 未建立 EtherCAT 的过程数据通信时
- 在“EtherCAT 通信未建立”状态下，发生“从站通信异常”时
- 发生“无法计算绝对编码器当前位置”的异常时
- 从站发生脱离时

此外，对从轴多重启动了同步控制指令时，情况如下。

- 多重启动时，即使主轴或辅助轴为上述 4 个条件中的任意一个，仍将正常受理多重启动，并变为缓冲状态。
- 多重启动且缓冲状态的指令开始动作时，若不是上述 4 个条件之一，则开始动作。



参考

对主轴或辅助轴启动 MC_Home 或 MC_HomeWithParameter，或对使用绝对编码器的轴执行了 MC_Power 时，从轴将忽略主轴或辅助轴的位置变化。
因此，在原点确定时的处理中，从轴不会发生剧烈动作。

● NX 系列 CPU 单元的条件

主轴及辅助轴的任务和从轴的任务分配到不同的任务中时，从轴将发生“主轴指定不正确（错误代码：5462 Hex）”，同步控制指令的输出变量“Error（错误）”变为 TRUE。

1-2 运动控制指令的基础知识

下面介绍使用CPU单元中的内置MC功能模块的运动控制指令创建程序时，需要的基本规格和限制事项。

关于运动控制指令的详情，请参考 □「第3章 轴指令」、□「第4章 轴组指令」、□「第5章 共通指令」。

运动控制指令的名称

MC 功能模块的所有运动控制指令前面均带“MC_”。

如何分辨是PLCopen®定义的指令，还是MC功能模块独立的指令，请参考 □「2-2 指令一览(P.2-26)」。

运动控制指令的语言

MC 功能模块的运动控制指令可使用以下编程语言。

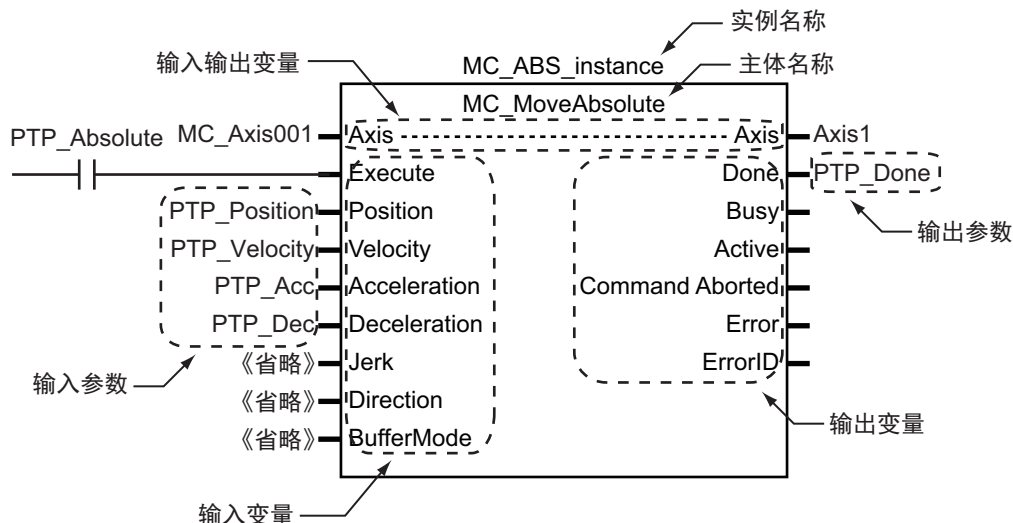
- 梯形图 (LD)
- 结构文本 (ST)

梯形图 (LD)

将运动控制指令的运动实例配置到梯形图中。

可命名每个指令实例的实例名称。

下面为 MC_MoveAbsolute (绝对值定位) 指令的记述示例。



- 在输入输出变量 Axis 中，指定要控制的伺服驱动器等轴变量名称。
- 在各输入变量中，指定目标位置或目标速度等动作条件。
- 各输出变量中，将输出指令的状态或伺服驱动器的状态。
- 省略了各输入参数时，为各输入变量的初始值。

结构文本 (ST)

指定指令实例名称。

指令的变量按从左上到左下、右上到右下的顺序记述。

下面为 MC_MoveAbsolute (绝对值定位) 指令的记述示例。

```
MC_ABS_instance (
  Axis := MC_Axis001 ,
  Execute := PTP_Absolute ,
  Position := PTP_Position ,
  Velocity := PTP_Velocity ,
  Acceleration := PTP_Acc ,
  Deceleration := PTP_Dec ,
  Jerk := PTP_Jerk ,
  Direction := _mcNoDirection ,
  BufferMode := _mcAborting ,
  Axis => MC_Axis001 ,
  Done => PTP_Done
);
```

运动控制指令的配置

下面介绍运动控制指令可分配到哪个任务中，或配置到程序的哪个位置后将发生怎样的动作。

任务类型

运动控制指令可在主固定周期任务、固定周期任务（执行优先度 5）和固定周期任务（执行优先度 16）中使用。若在上述以外的任务中使用运动控制指令，执行链接时将发生异常。

任务类型	可否使用	备注
主固定周期任务	可以	<ul style="list-style-type: none"> 不指定轴 / 轴组的共通指令 对主固定周期任务中分配的轴 / 轴组执行的指令
固定周期任务（执行优先度 5）	可以 ^{*1}	<ul style="list-style-type: none"> 不指定轴 / 轴组的共通指令 对固定周期任务（执行优先度 5）中分配的轴 / 轴组执行的指令
固定周期任务（执行优先度 16）	可以	<ul style="list-style-type: none"> 不指定轴 / 轴组的共通指令 对主固定周期任务中分配的轴 / 轴组执行的指令
固定周期任务（执行优先度 17）	不可	
固定周期任务（执行优先度 18）	不可	
事件任务（执行优先度 8）	不可	
事件任务（执行优先度 48）	不可	

*1. 仅在 NX 系列 CPU 单元时可使用。不可在 NJ 系列 CPU 单元中使用。

功能块定义内

运动控制指令还可在用户创建的功能块定义中使用。

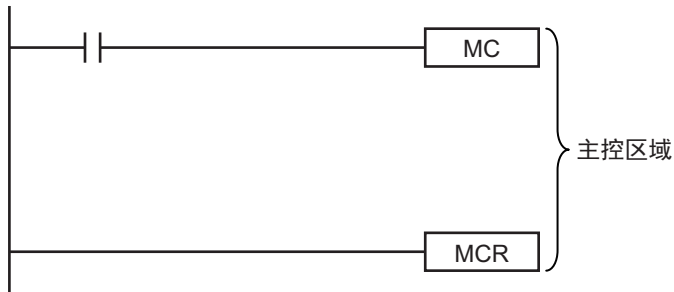


参考

将多个动作组合而成的工序汇总为一个功能块，利用程序的结构化，提高设计效率和程序的识别性。

主控区域

在梯形图程序中，从 MC 指令（主控开始指令）到 MCR 指令（主控结束指令）之间的区域成为主控区域。

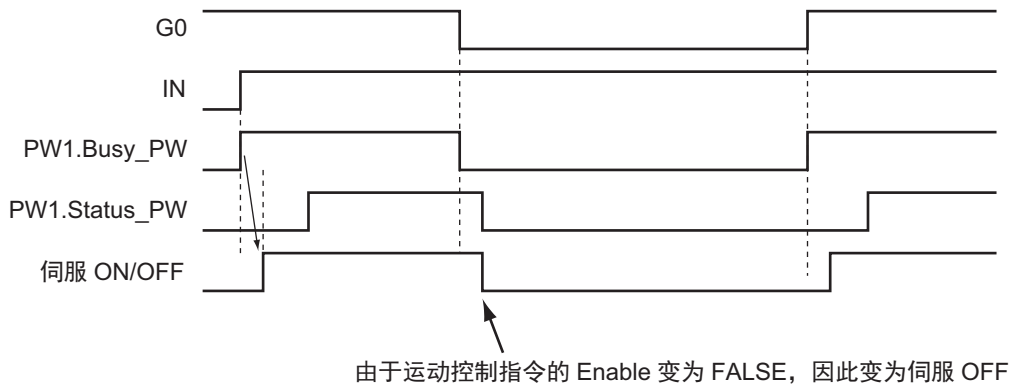
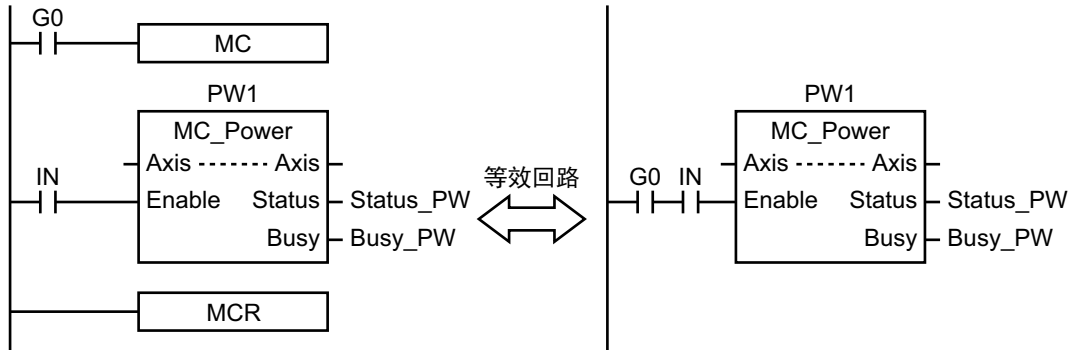


将运动控制指令配置到主控区域时，MC 的输入条件为 FALSE 时，动作如下。

- 从左母线直接输入到运动控制指令的输入变量 Enable 或 Execute 的指令，将作为 FALSE 动作。
- 实例 ST 部分正常动作。
- 即使运动控制指令的输入变量 Enable 或 Execute 作为 FALSE 动作，输出参数的值仍将正常更新。

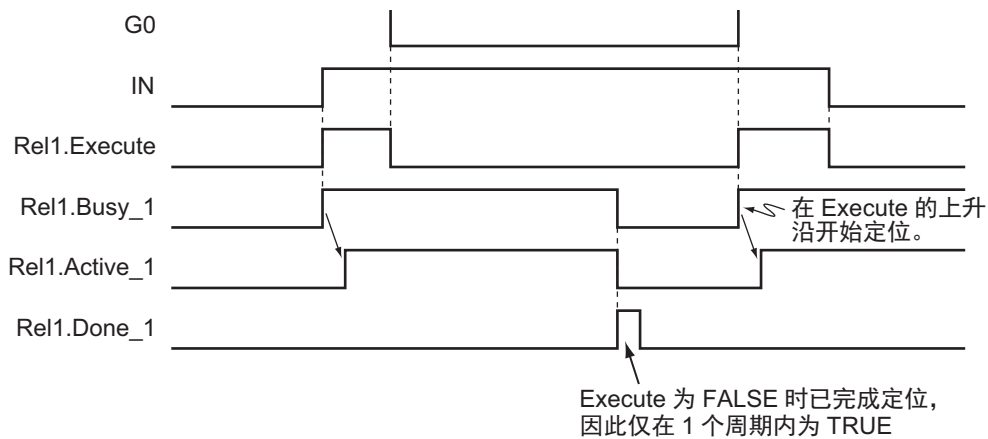
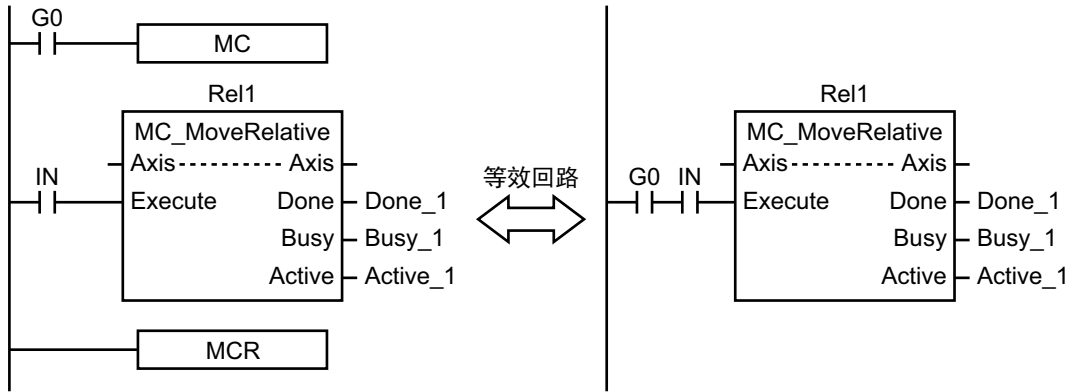
● Enable 型运动控制指令时

- 主控区域中配置的指令与下图右侧的回路等效。
- G0 为 TRUE 时，正常动作。
- G0 为 FALSE 时，Enable 输入与 FALSE 相同。



● Execute 型运动控制指令时

- 主控区域中配置的指令与下图右侧的回路等效。
- G0 为 TRUE 时，正常动作。
- G0 为 FALSE 时，Execute 输入与 FALSE 相同。
- G0 为 TRUE 时启动的指令即使在动作过程中 G0 变为 FALSE，也会继续动作，直至完成。此时，输出参数的值也会正常更新。



使用注意事项

Execute 型运动控制指令会在上升沿开始动作，因此不建议在主控区域内使用。如果使用，请时刻注意其动作。



参考

- ST 程序中，与 MC 指令（主控开始指令）的功能无效。所有指令均正常动作。MC 指令 / MCR 指令的详情请参考 □ 《NJ/NX 系列 指令基准手册 基本篇（SBCE-CN5-360）》。

ST 语法指令中

下面介绍在 IF 指令、CASE 指令、WHILE 指令、REPEAT 指令等 ST 语法指令中配置了运动控制指令的情况。

与各ST语法指令的条件式评估结果不一致时，不执行运动控制指令。此外，指令的输出变量值也不更新。开始执行 Execute 型指令后，即使评估结果变为不一致，仍会继续执行，直至处理完成。此时，指令的输出变量值不更新。



使用注意事项

条件式的评估结果不一致时，执行中的 Execute 型指令执行状态会变为不详，因此不建议在 ST 语法指令中使用。
如果使用，请时刻注意其动作。



参考

- 通过条件式切换执行的 Execute 型指令时，只将 Execute 的输入参数配置到 ST 语法指令中。Execute 型指令本身配置到 ST 语法指令之外。
ST 语法指令的详情请参考 □ 《NJ/NX 系列 指令基准手册 基本篇（SBCE-CN5-364）》。

实数（REAL 型、LREAL 型）的处理

实数（REAL 型、LREAL 型）是指拥有浮动小数点型数据格式的数据类型。
下面介绍其表述方法和运算处理。

● REAL 型、LREAL 型的数据大小

REAL 型和 LREAL 型的数据大小不同。

REAL 型的数据大小为 32 位，LREAL 型为 64 位。

● 浮动小数点格式的数据格式

浮动小数点格式是指用符号、指数、尾数表示实数的格式。

用以下公式表示某实数时，s 的值为符号，e 的值为指数，f 的值为尾数。

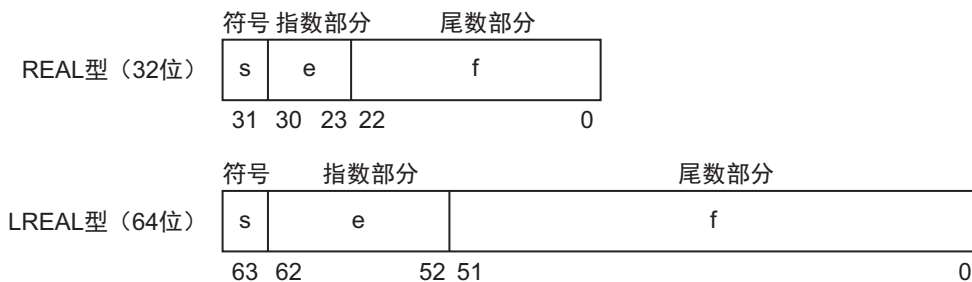
• REAL 型

$$\text{数值} = (-1)^s 2^{e-127} (1+f \times 2^{-23})$$

• LREAL 型

$$\text{数值} = (-1)^s 2^{e-1023} (1+f \times 2^{-52})$$

浮动小数点格式的数据格式符合 IEEE754 标准。分别如下所示。



(例) REAL 型，表示“-86.625”时

1 符号的设定

由于为负数，所以 s=1。

2 进制数的表示

若用 2 进制数表示“86.625”，则为“1010110.101”。

3 2 进制数的标准化

若将上述值进行标准化，则变为“1.010110101×2⁶”。

4 指数部的表示

利用上述公式，e-127=6，因此 e=133。

“133”在 2 进制数中为“10000101”，这就是指数部的表示。

5 尾数部的表示

“1.010110101”的小数点以下为“010110101”。

为使小数点以下的数值表示为 23 位，不足的位数需要以 0 补充。达到 23 位的值变为 f。

因此，f = 0101101010000000000000。

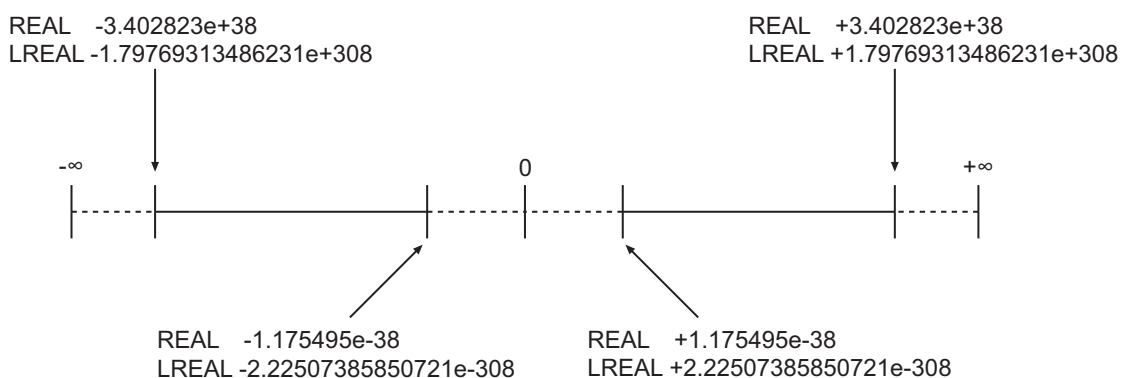
所以，“-86.625”可以如下图所示表示。



● 有效范围

REAL 型、LREAL 型的有效范围如下表所示。

数据类型	$-\infty$	负数	0	正数	$+\infty$
REAL	$-\infty$	$-3.402823e+38 \sim -1.175495e-38$	0	$+1.175495e-38 \sim +3.402823e+38$	$+\infty$
LREAL	$-\infty$	$-1.79769313486231e+308 \sim -2.22507385850721e-308$	0	$+2.22507385850721e-308 \sim +1.79769313486231e+308$	$+\infty$



● 特殊值

“ $+\infty$ ”、“ $-\infty$ ”、“+0”、“-0”、非数值称为特殊值。

非数值是指无法视为浮动小数点数的数据，不作为数值处理。

“+0”和“-0”在数学上都是“0”的意思，但运算处理内容不同。

详细信息后面再介绍。

特殊值的符号 s、指数 e、尾数 f 的值如下表所示。

数据类型	特殊值	符号 s	指数 e	尾数 f
REAL	$+\infty$	0	255	0
	$-\infty$	1	255	0
	+0	0	0	0
	-0	1	0	0
	非数值	-	255	0 以外
LREAL	$+\infty$	0	2047	0
	$-\infty$	1	2047	0
	+0	0	0	0
	-0	1	0	0
	非数值	-	2047	0 以外

● 非标准数

浮动小数点格式无法表示“0”附近（绝对值非常小）的值。

为了扩大“0”附近的有效范围而设计了非标准数。

与正常数据格式表示的值相比，非标准数可以表示绝对值较小的值。



参考

以正常数据格式表示的值称为标准数。

指数 $e = 0$ 且尾数 $f \neq 0$ 的数值将视为非标准数，值的表示如下。

- REAL 型

$$\text{数值} = (-1)^s 2^{-126} (f \times 2^{-23})$$

- LREAL 型

$$\text{数值} = (-1)^s 2^{-1022} (f \times 2^{-52})$$

(例) REAL 型，表示“ 0.75×2^{-127} ”时

1 符号的设定

由于为正数，所以 $s=0$ 。

2 2 进制数的表示

若用 2 进制数表示“0.75”，则为“0.11”。

3 尾数部的计算

因为 $(0.11)_2 \times 2^{-127} = 2^{-126} (f \times 2^{-23})$ ，所以 $f = (0.11)_2 \times 2^{22}$ 。

4 尾数部的表示

根据以上公式， $f = 0110000000000000000000$ 。

所以，“ 0.75×2^{-127} ”可以如下图表示。

	符号	指数部分	尾数部分
REAL型 (32位)	0	00000000	011000000000000000000000
	31	30 23 22	0

非标准数比标准数的有效位数少。因此，标准数的运算结果为非标准数时，或在运算过程中结果变为非标准数时，运算结果的有效位数可能小于标准数的有效位数。

● 运算处理

浮动小数点格式为近似值表现，因此与真值存在误差。此外，值的有效范围也有限制。因此，运算时，需要进行以下处理。

取整

真值超出尾数的有效位数时，按照以下规格取整数值。

- 在浮动小数点格式可表现的数值中，将最接近真值的值作为运算结果。
 - 在浮动小数点格式可表现的数值中，最接近真值的值有 2 个时，将尾数最低位的位值有“0”的值作为运算结果。
- 最接近真值的值有 2 个的情况是指真值为 2 个近似值的中间值时。

溢出、下溢

真值的绝对值大于浮动小数点格式可表现的数值时，称为溢出。反之，小于浮动小数点格式可表现的数值时，称为下溢。

- 溢出时，如果真值的符号为正，则运算结果为“+∞”。
- 如果真值的符号为负，则运算结果为“-∞”。
- 下溢时，如果真值的符号为正，则运算结果为“+0”。
- 如果真值的符号为负，则运算结果为“-0”。

特殊值的运算

特殊值的运算按以下规则进行。

关于特殊值，请参考「特殊值 (P.1-15)」。

- “+∞”和“-∞”的相加结果为非数值。
- 符号相同的无限大值的相减结果为非数值。
- “+0”和无限大、“-0”和无限大的相乘结果为非数值。
- “+0”的相除结果、“-0”的相除结果、无限大的相除结果为非数值。
- “+0”和“-0”的相加结果为“+0”。
- “+0”的相减结果、“-0”的相减结果为“+0”。
- 包含非数值的四则运算结果为非数值。
- Cmp 指令等比较指令中，“+0”和“-0”判断为相等。
- 比较指令中，如果比较对象中包含非数值，将始终判断为不相等。



使用注意事项

MC 功能模块中，电子齿轮的设定值或运动控制指令的目标位置使用浮动小数点 (LREAL) 的变量。因此，运算结果中含有取整误差。

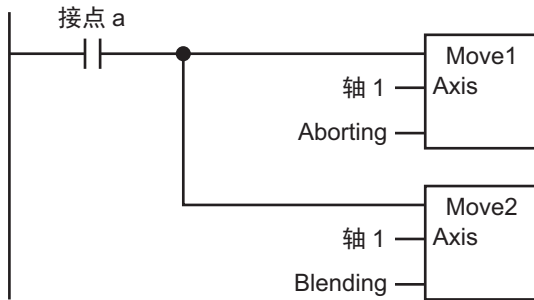
例如，反复执行 MC_MoveRelative（相对值定位）指令时，误差会累计。

误差累计导致出现问题时，请将指令单位设定为 [pulse]，或指定类似 MC_MoveAbsolute（绝对值定位）指令的绝对位置。

运动控制指令的同时启动

下面介绍对同一轴、在同一任务周期内启动多个运动控制指令的情况。

- 下图程序时，若将接点 a 设为“ON”，指令实例 [Move 1] 和 [Move 2] 将在同一任务周期内启动。
- 程序中记述的指令实际会从上往下依次启动，因此先启动 [Move1]，在 [Move1] 完成动作前，启动 [Move2]。
- 这相当于运动控制指令的多重启动。在该程序中，相对于 [Move1]，[Move2] 通过 Blending 进行多重启动。

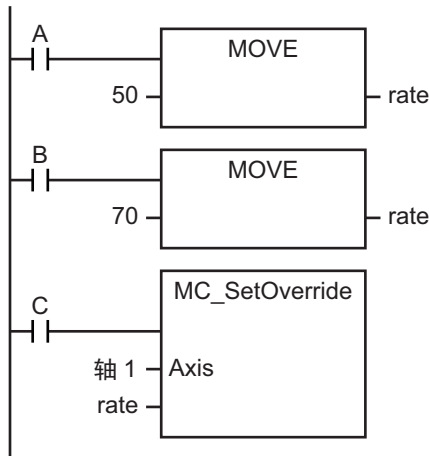


各运动控制指令的多重启动详情请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。



参考

如上同时启动 MC_SetOverride（超驰值设定）指令时，下侧的超驰值将变为有效。在 MC_SetOverride（超驰值设定）指令中，分别使用多个超驰值时，建议按以下方式编程。



运动控制指令的在线编辑

Sysmac Studio 的在线编辑功能可对运动控制指令执行以下操作。

在线编辑操作
运动控制指令的删除
运动控制指令的追加
运动控制指令的输入、输出、输入输出变量的追加
运动控制指令的输入、输出、输入输出变量的变更
运动控制指令的输入、输出、输入输出变量的删除



使用注意事项

轴的动作过程中，若删除 MC_Stop（强制停止）指令或 MC_GroupStop（轴组强制停止）指令等停止轴动作的指令，在有些用户程序的内容下，轴可能会无法停止。
对运动控制指令进行在线编辑时，请在确认安全后再使用。

CPU 单元的动作模式切换

NJ/NX 系列的 CPU 单元动作模式中，包括程序模式和运行模式两种。

下面介绍切换动作模式 MC 功能模块的动作。

从运行模式切换为程序模式

- 中断执行中的运动控制指令。输出变量 CommandAborted（执行中断）保持 FALSE，轴动作与 CommandAborted（执行中断）为 TRUE 时相同。
- 轴正在动作时，以最大减速度减速停止。伺服 ON/OFF 状态继续。
- 正在利用凸轮表保存指令保存凸轮表时，继续执行保存处理。
- 正在利用凸轮表生成指令执行凸轮表生成时，继续执行生成处理。
- 执行优先级为 16 的固定周期任务中配置的运动控制指令将在执行该任务的结束指令后，执行上述处理。

从程序模式切换为运行模式

- 运动控制指令的输出变量将清除。
- 从运行模式变更为程序模式后，轴将减速停止。若在减速停止过程中恢复为运行模式，运动控制指令的输出变量将清除，因此所执行的运动控制指令的 CommandAborted（执行中断）保持 FALSE。



参考

- 切换动作模式后，若要参照运动控制指令的输出变量，请将有保持属性的变量分配到输出参数中。
通过参照分配的输出参数，可以参照切换动作模式前的输出变量。
 - 即使切换动作模式，伺服 ON/OFF 的状态仍将保持。
-

变量和指令

表示 MC 功能模块中处理的变量和指令的一览表。

2-1 变量一览	2-2
MC 共通变量	2-3
轴变量	2-4
轴组变量	2-8
运动控制指令的输入变量	2-11
运动控制指令的输出变量	2-22
运动控制指令的输入输出变量	2-24
2-2 指令一览	2-26
共通指令	2-26
轴指令	2-27
轴组指令	2-29
2-3 PDO 映射	2-30
必须对象	2-30
不同指令的设定对象	2-32

2-1 变量一览

MC 功能模块中处理的变量分为两种。

一种是用于监视轴等的状态或部分参数设定内容的系统定义变量。MC功能模块中处理的系统定义变量称为运动控制系统变量。

另一种是运动控制指令将指令参数作为输入接收或将指令的执行状态作为输出所需的变量。运动控制指令的输入变量中，包括从选项中选择列举型变量。

本节介绍各种变量的一览、运动控制指令的输入变量有效范围及列举型的一览。

● 运动控制系统变量

1 级菜单	2 级菜单	3 级菜单	内容
系统定义变量	运动控制系统变量	MC 共通变量	可监视 MC 功能模块共通的状态
		轴变量	可监视各轴的状态和部分轴参数的设定内容
		轴组变量	可监视各轴组的状态和部分轴组参数的设定内容

运动控制系统变量的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。



参考

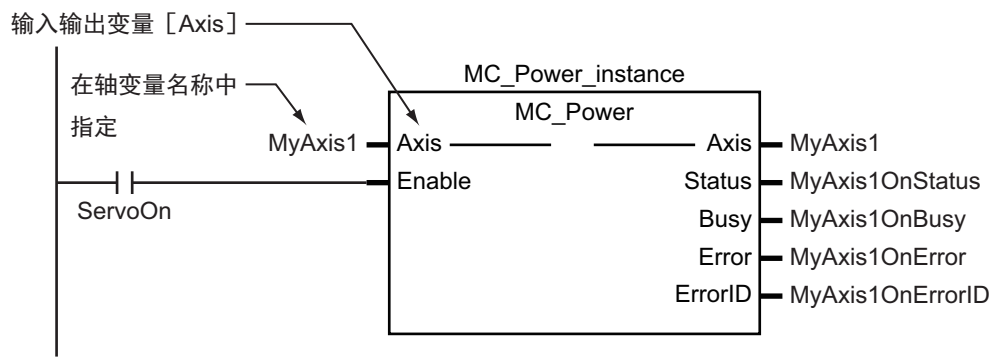
轴变量和轴组变量为系统定义变量。

在用户程序中使用这些变量时，使用系统定义变量名称（_MC_AX[*]、_MC1_AX[*]、_MC2_AX[*]、_MC_GRP[*]、_MC1_GRP[*]、_MC2_GRP[*]）。

在用户程序中使用时，除了上述变量名称之外，还可使用通过 Sysmac Studio 创建的变量名称。

通过 Sysmac Studio 创建的轴或轴组的变量名称可变更为任意名称。

下图示例表示针对系统定义变量的轴变量名称为“_MC_AX[0]”的轴，将通过 Sysmac Studio 追加时的轴变量名称变更为“MyAxis1”。



● 运动控制指令的变量

种类	概要
输入变量	变量的参数
输出变量	监视指令的执行状态
输入输出变量	指定指令的执行对象



参考

- 开头为“_e”的数据类型表示枚举型。
- 开头为“_s”的数据类型表示结构体型。

MC 功能模块中处理的数据类型请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

MC 共通变量

MC 共通变量是指变量名称为 `_MC_COM`、数据类型为 `_sCOMMON_REF` 型的结构体型变量。

下面介绍 MC 共通变量的构成及其成员的详情。

变量名称	数据类型	名称	功能
<code>_MC_COM</code>	<code>_sCOMMON_REF</code>	MC 共通变量	
Status	<code>_sCOMMON_REF_STA</code>	MC 共通状态	
RunMode	BOOL	MC 运行中	MC 功能模块正在运行时为 TRUE
TestMode	BOOL	MC 调试中	正在通过 Sysmac Studio 调试时为 TRUE
CamTableBusy	BOOL	凸轮表文件保存执行中 / 待机中	正在保存凸轮表文件或待机时变为 TRUE
GenerateCam Busy ^{*1}	BOOL	凸轮表生成执行中	正在生成凸轮表时变为 TRUE
PFaultLvl	<code>_sMC_REF_EVENT</code>	MC 共通 部分停止故障	
Active	BOOL	MC 共通 发生部分停止故障	发生 MC 共通的停止故障时变为 TRUE
Code	WORD	MC 共通 部分停止故障代码	输出 MC 共通的停止故障代码。与事件代码的前面 4 位相同的值
MFaultLvl	<code>_sMC_REF_EVENT</code>	MC 共通 轻度故障	
Active	BOOL	MC 共通 发生轻度故障	发生 MC 共通的轻度故障时变为 TRUE
Code	WORD	MC 共通 轻度故障代码	输出 MC 共通的轻度故障代码。与事件代码的前面 4 位相同的值
Obsr	<code>_sMC_REF_EVENT</code>	MC 共通 监视信息	
Active	BOOL	MC 共通 监视信息发生中	发生 MC 共通的监视信息时变为 TRUE
Code	WORD	MC 共通 监视信息代码	输出 MC 共通的监视信息代码。与事件代码的前面 4 位相同的值

*1. 可在 CPU 单元版本 1.08 以上和 Sysmac Studio Ver.1.09 以上的组合中使用。

轴变量

轴变量是指系统定义变量的变量名称为 `_MC_AX[0-255]`、`_MC1_AX[0-255]`、`_MC2_AX[0-255]`、数据类型为 `_sAXIS_REF` 型的结构体型变量。

下面以 `_MC_AX[0-255]` 为例，介绍轴变量的构成及其成员的详情。`_MC1_AX[0-255]`、`_MC2_AX[0-255]` 亦同。

变量名称	数据类型	名称	功能
<code>_MC_AX[0-255]</code>	<code>_sAXIS_REF</code>	轴变量	
Status	<code>_sAXIS_REF_STA</code>	轴状态	
Ready	BOOL	轴启动准备完成	轴已经完成启动准备、处于停止状态时变为 TRUE。 <code>_MC_AX[*].Status.Standstill</code> 表示与“TRUE: 停止中”时相同的状态
Disabled	BOOL	轴无效	轴为伺服 OFF、处于停止状态时变为 TRUE。 与以下轴状态互斥，只有一个会变为 TRUE Disabled/ Standstill/ Discrete/ Continuous/ Synchronized/ Homing/ Stopping/ ErrorStop/ Coordinated
Standstill	BOOL	停止中	轴为伺服 ON、处于停止状态时变为 TRUE
Discrete	BOOL	定位动作中	正在向目标位置执行定位控制时变为 TRUE。 包括在定位动作中将超驰值设定为“0”后速度变为“0”的状态
Continuous	BOOL	连续动作中	没有目标位置，连续动作时变为 TRUE。 速度控制和扭矩控制时变为该状态。 包括将目标速度设为“0”后速度为“0”的状态或在连续动作中将超驰值设定为“0”后速度变为“0”的状态
Synchronized	BOOL	同步动作中	正在执行同步控制时变为 TRUE。 包括切换同步控制指令后的同步等待状态
Homing	BOOL	原点复位中	正在通过 <code>MC_Home</code> 指令或 <code>MC_HomeWithParameter</code> 指令执行原点复位时变为 TRUE
Stopping	BOOL	减速停止中	利用 <code>MC_Stop</code> 指令或 <code>MC_TouchProbe</code> 指令停止轴时，轴完全停止前为 TRUE。 包括利用 <code>MC_Stop</code> 指令停止轴后，Execute 为 TRUE 时。 减速停止中时，无法启动轴动作指令 (<code>CommandAborted = TRUE</code>)
ErrorStop	BOOL	错误减速停止中	<code>MC_ImmediateStop</code> 指令或轴发生轻度故障 (<code>_MC_AX[*].MFaultLvl.Active</code> 为“TRUE: 轴发生轻度故障”) 时，轴停止前和停止的状态。 在此状态下，无法启动轴动作指令 (<code>CommandAborted = TRUE</code>)

变量名称	数据类型	名称	功能
Coordinated	BOOL	多轴协调动作中	因多轴协调指令，轴组处于有效状态时变为 TRUE
Details	_sAXIS_REF_DET	轴控制状态 *1	
Idle	BOOL	停止中	除位置等待外，未计算指令值时变为 TRUE。*2 与 Idle 和 InPosWaiting 互斥，因此不会同时变为 TRUE
InPosWaiting	BOOL	位置等待	位置等待时变为 TRUE。 在带位置检查的定位中，正在进行位置检查的状态
Homed	BOOL	原点确定	原点确定状态时变为 TRUE *3 TRUE：原点确定 FALSE：原点未确定
InHome	BOOL	原点停止	在原点位置的范围内时为 TRUE。 以下的 AND 条件 • 原点确定状态 • 反馈当前位置进入以原点为中心的“原点位置范围”内时 在指令状态下轴正在动作时，若通过原点位置，也会变为 TRUE
VelLimit *4	BOOL	指令速度饱和	同步控制中指令速度受最高速度限制时变为 TRUE
Dir	_sAXIS_REF_DIR	指令方向状态 *5	
Posi	BOOL	正方向指令中	正在向正方向指令时变为 TRUE
Nega	BOOL	负方向指令中	正在向负方向指令时变为 TRUE
DrvStatus	_sAXIS_REF_STA_DRV	伺服驱动器状态 *6	
ServoOn	BOOL	伺服 ON	伺服电机通电时为 TRUE
Ready	BOOL	伺服就绪	伺服就绪状态时变为 TRUE
MainPower	BOOL	主回路电源	伺服驱动器的主回路电源通电时为 TRUE
P_OT	BOOL	正方向极限输入	正方向极限输入为有效时变为 TRUE
N_OT	BOOL	负方向极限输入	负方向极限输入为有效时变为 TRUE
HomeSw	BOOL	近原点输入	近原点输入为有效时变为 TRUE
Home	BOOL	原点输入	原点输入为有效时变为 TRUE *7
ImdStop	BOOL	立即停止输入	立即停止输入为有效时变为 TRUE
Latch1	BOOL	外部锁定输入 1	锁定输入 1 为有效时变为 TRUE
Latch2	BOOL	外部锁定输入 2	锁定输入 2 为有效时变为 TRUE
DrvAlarm	BOOL	驱动器错误输入	发生驱动器错误时变为 TRUE
DrvWarning	BOOL	驱动器警告输入	发生驱动器警告时变为 TRUE
ILA	BOOL	驱动器内部功能限制中	正在利用伺服驱动器的限制功能实际进行限制时变为 TRUE。*8
CSP	BOOL	周期性同步位置 (CSP) 模式中	伺服驱动器为伺服 ON、处于 CSP 模式时变为 TRUE *9
CSV	BOOL	周期性同步速度 (CSV) 模式中	伺服驱动器为伺服 ON、处于 CSV 模式时变为 TRUE *9

变量名称	数据类型	名称	功能
CST	BOOL	周期性同步扭矩 (CST) 模式中	伺服驱动器为伺服 ON、处于 CST 模式时变为 TRUE ^{*9}
Cmd	_sAXIS_REF_CMD_DATA	轴指令值	
Pos	LREAL	指令当前位置	输出指令位置的当前值。(单位: 指令单位) 伺服 OFF 时及位置控制模式以外时, 输出反馈当前位置 ^{*10}
Vel	LREAL	指令当前速度	输出指令速度的当前值。(单位: 指令单位/s) 正方向移动时加 + 号, 负方向移动时加 - 号。 根据指令当前位置的差计算得到的速度。伺服 OFF 时及位置控制模式以外时, 根据反馈当前位置计算得到的速度
AccDec	LREAL	指令当前加减速度	输出指令加减速度的当前值。(单位: 指令单位/s ²) 根据指令当前速度的差计算得到的加减速度。加速时加 + 号, 减速时加 - 号。 执行中的指令加减速度为“0”时, 变为“0”
Jerk	LREAL	指令当前跃度	输出指令跃度的当前值。(指令单位/s ³) 加减速度的绝对值增加时加 + 号, 减少时加 - 号。 执行中的指令加减速度、指令跃度为“0”时, 变为“0”
Trq	LREAL	指令当前扭矩	输出指令扭矩的当前值。(单位:%) 正方向移动时加 + 号, 负方向移动时加 - 号。 扭矩控制模式以外时, 输出与反馈当前扭矩相同的值
Act	_sAXIS_REF_ACT_DATA	轴当前值	
Pos	LREAL	反馈当前位置	输出反馈位置的当前值 (单位: 指令单位) ^{*10}
Vel	LREAL	反馈当前速度	输出反馈速度的当前值。(单位: 指令单位/s) 正方向移动时加 + 号, 负方向移动时加 - 号
Trq	LREAL	反馈当前扭矩	输出反馈扭矩的当前值。(单位:%) 正方向移动时加 + 号, 负方向移动时加 - 号
TimeStamp ^{*11}	ULINT	时间戳	输出轴当前位置的更新时刻。对变化时刻获取功能正在运行的轴有效。 (单位:ns)
MFaultLvl	_sMC_REF_EVENT	轴 轻度故障	
Active	BOOL	轴 发生轻度故障	发生轴的轻度故障时变为 TRUE
Code	WORD	轴 轻度故障代码	输出轴的轻度故障代码。 与事件代码的前面 4 位相同的值

变量名称	数据类型	名称	功能
Obsr	_sMC_REF_EVENT	轴 监视信息	
Active	BOOL	轴 监视信息发生中	发生轴的监视信息时变为 TRUE
Code	WORD	轴 监视信息代码	输出轴的监视信息代码。 与事件代码的前面 4 位相同的值
Cfg	_sAXIS_REF_CFG	轴基本设定 *12	
AxNo	UINT	轴编号	表示轴的逻辑编号。 参照 _sAXIS_REF 时，需要识别轴编号时参照
AxEnable	_eMC_AXIS_USE	轴使用	表示是否使用轴 0: _mcNoneAxis (未创建轴) 1: _mcUnusedAxis (未使用轴) 2: _mcUsedAxis (使用轴)
AxType	_eMC_AXIS_TYPE	轴种类	表示轴种类。 虚拟轴不需要输入输出配线 0: _mcServo (伺服轴) 1: _mcEncdr (编码器轴) 2: _mcVirServo (虚拟伺服轴) 3: _mcVirEncdr (虚拟伺服轴)
NodeAddress	UINT	节点地址	表示 EtherCAT 的从站地址。*13 16#FFFF 表示无。
ExecID *14	UINT	执行 ID	表示各任务的执行 ID。 0: 未分配任务 (未创建轴) 1: 分配到主固定周期任务 2: 分配到固定周期任务 (执行优先度 5)
Scale	_sAXIS_REF_SCALE	单位转换 *15	
Num	UDINT	电机每转 1 圈的脉冲数	表示指令位置的电机每转 1 圈的脉冲数。 根据电子齿轮比，从指令值转换为脉冲量
Den	LREAL	电机每转 1 圈的移动量	表示指令位置的电机每转 1 圈的工件移动量 *16
Units	_eMC_UNITS	显示单位	表示指令位置的显示单位 0: _mcPls (pulse) 1: _mcMm (mm) 2: _mcUm (μm) 3: _mcNm (nm) 4: _mcDeg (degree) 5: _mcInch (inch)
CountMode *14	_eMC_COUNT_MODE	计数模式	表示计数模式。 0: _mcCountModeLinear (直线模式) 1: _mcCountModeRotary (旋转模式)
MaxPos *14	LREAL	当前位置上限值	表示当前位置显示的上限值。*17
MinPos *14	LREAL	当前位置下限值	表示当前位置显示的下限值。*18

*1. 表示指令的控制状态。

*2. 以速度“0”动作、偏差计数器复位中、同步控制中及多轴协调动作中也视为正在运算的状态。

*3. 即使已变为 TRUE，在以下情况下，仍需重新确定原点。

变更了位置计数设定或单位转换设定时。

伺服驱动器上发生绝对位置信息丢失的异常或执行了此类操作时。例如，编码器电缆断线或清除了绝对编码器数据等。

*4. VelLimit 只能用于同步控制中的从轴。

*5. 表示指令的移动方向。

*6. 表示驱动器等的状态。

*7. 在 Sysmac Studio 的 [轴基本设定] 画面的 [详细设定] 中，[数字输入] 的 [编码器 Z 相检测] 中设定的信号的状态在其他公司生产的驱动器中，此信号可能无法映射到 PDO 中。请确认所连接驱动器的手册

*8. 显示 PDO 中映射的“状态字 (6041Hex)”第 11 位 (内部限制功能有效) 的状态。变为 TRUE 的条件取决于伺服驱动器的规格。请参考所连接伺服驱动器的手册。在欧姆龙生产的伺服驱动器 G5 系列中，为扭矩限制 / 速度限制 / 驱动禁止输入 / 软件限位之一。

- *9. 根据 PDO 中映射的“操作模式显示 (6061Hex)”的值显示。CSP、CSV、CST 变为 TRUE 的条件取决于伺服驱动器的规格。请参考所连接伺服驱动器的手册。
PDO 中未映射“操作模式显示 (6061Hex)”时，始终为 FALSE。
- *10. 轴中分配的 EtherCAT 从站或 NX 单元与 CPU 单元尚未建立过程数据通信时，不同 CPU 单元版本中的情况如下。
CPU 单元版本 1.09 以下时，轴变量的反馈当前位置及指令当前位置为“0”或下限值。变为下限值的条件：计数模式为旋转模式，位置范围中不包含“0”时。
在单元版本 1.10 以上的 CPU 单元中，轴变量的反馈当前位置及指令当前位置为过程数据通信未建立前的反馈当前位置。
- *11. 可在 CPU 单元版本 1.06 以上和 Sysmac Studio Ver.1.07 以上的组合中使用。
- *12. 表示轴基本设定的设定内容。
- *13. NX 系列位置接口单元时，表示安装了位置接口单元的 EtherCAT 耦合器单元的节点地址。
- *14. 可在 CPU 单元版本 1.10 以上和 Sysmac Studio Ver.1.12 以上的组合中使用。
- *15. 表示电子齿轮比的设定值。
- *16. 单元版本 1.11 以上的 CPU 单元中，将追加的单位转换设定的减速机使用设定为“使用”时，该参数无效。确认有效参数“工件侧每转 1 圈的移动量”、“工件侧齿轮比”、“电机侧齿轮比”的值时，请使用 MC_ReadAxisParameter（轴参数读取）指令。
- *17. 计数模式为直线模式时，显示即将溢出时的位置。为旋转模式时，显示循环计数器上限值。
- *18. 计数模式为直线模式时，显示即将下溢时的位置。为旋转模式时，显示循环计数器下限值。

轴组变量

轴组变量是指系统定义变量的变量名称为 `_MC_GRP[0-63]`、`_MC1_GRP[0-63]`、`_MC2_GRP[0-63]`、数据类型为 `_sGROUP_REF` 型的结构体变量。

下面以 `_MC_GRP[0-63]` 为例，介绍轴组变量的构成及其成员的详情。`_MC1_GRP[0-63]` 或 `_MC2_GRP[0-63]` 亦同。

此外，功能说明中以 `_MC_AX[*]` 为例对轴变量进行说明，`_MC1_AX[*]`、`_MC2_AX[*]` 亦同。

变量名称	数据类型	名称	功能
<code>_MC_GRP[0-63]</code>	<code>_sGROUP_REF</code>	轴组变量	
Status	<code>_sGROUP_REF_STA</code>	轴组状态	
Ready	BOOL	启动准备完成	轴组处于停止状态、已经完成启动准备时变为 TRUE。 以下的 AND 条件为启动准备完成的条件 <ul style="list-style-type: none"> 构成轴未执行 MC_Stop 指令。 <code>_MC_GRP[*].Status.Standby</code>（停止中）为 TRUE 构成轴为伺服 ON 状态 构成轴的 <code>_MC_AX[*].Details.Homed</code>（原点确定）为 TRUE
Disabled	BOOL	轴组无效	轴组无效、处于停止状态时变为 TRUE。 与以下轴组状态互斥，只有一个会变为 TRUE。 Disable/ Standby/ Moving/ Stopping/ ErrorStop

变量名称	数据类型	名称	功能
Standby	BOOL	停止中	轴组动作指令处于停止状态时变为 TRUE。 与轴组构成轴的伺服 ON/OFF 状态无关。
Moving	BOOL	动作中	正在向目标位置执行轴组动作指令时变为 TRUE。 包括位置等待的状态或因超驰而使速度变为“0”的状态
Stopping	BOOL	减速停止中	利用 MC_GroupStop 指令停止轴组时，轴组完全停止前为 TRUE 包括利用 MC_GroupStop 指令停止轴后，Execute 为 TRUE 时。 在本状态下，无法启动轴组动作指令 (CommandAborted=TRUE)
ErrorStop	BOOL	错误减速停止中	因 MC_GroupImmediateStop 指令或轴组发生轻度故障 (<code>_MC_GRP[*].MFaultLvl.Active=TRUE</code>)，轴组停止前及停止时变为 TRUE。在本状态下，无法启动轴组动作指令 (CommandAborted=TRUE)
Details	<code>_sGROUP_REF_DET</code>	轴组控制状态 *1	
Idle	BOOL	停止中	除位置等待外，未计算指令值时变为 TRUE。*2 与 Idle 和 InPosWaiting 互斥，因此不会同时变为 TRUE
InPosWaiting	BOOL	位置等待	任一构成轴处于位置等待时变为 TRUE。 在带位置检查的定位中，正在进行位置检查的状态
Cmd	<code>_sGROUP_REF_CMD_DATA</code>	轴组指令值	
Vel	LREAL	指令插补速度	输出指令插补速度的当前值。 根据插补指令当前位置的差计算得到的插补速度。正方向移动时加 + 号，负方向移动时加 - 号。 轴组为无效时，变为“0”
AccDec	LREAL	指令插补加减速度	输出指令插补加减速度的当前值。 根据指令插补速度的差计算得到的插补加减速度。加速时加 + 号，减速时加 - 号。 轴组为无效时以及执行中的轴组动作指令加减速度为“0”时，变为“0”
MFaultLvl	<code>_sMC_REF_EVENT</code>	轴组 轻度故障	
Active	BOOL	轴组 发生轻度故障	发生轴组轻度故障时变为 TRUE
Code	UINT	轴组 轻度故障代码	输出轴组轻度故障代码。 与事件代码的前面 4 位相同的值

变量名称	数据类型	名称	功能
Obsr	_sMC_REF_EVENT	轴组 监视信息	
Active	BOOL	轴组 监视信息发生中	发生轴组的监视信息时变为 TRUE
Code	WORD	轴组 监视信息 代码	输出轴组的监视信息代码。 与事件代码的前面 4 位相同的值
Cfg	_sGROUP_REF_CFG	轴组基本设定	
GrpNo	UINT	轴组编号	表示轴组的逻辑编号。 参照 _sGROUP_REF 时，需要识别轴组编号 时参照
GrpEnable	_eMC_GROUP_USE	轴组使用	表示是否使用轴组。 0: _mcNoneGroup (未创建轴组) 1: _mcUnusedGroup (未使用轴组) 2: _mcUsedGroup (使用轴组)
ExecID *3	UINT	执行 ID	表示已分配的各任务的执行 ID。 0: 未分配任务 (未创建组) 1: 分配到主固定周期任务 2: 分配到固定周期任务 (执行优先度 5)
Kinematics	_sGROUP_REF_KIM	运动学转换设定 *4	
GrpType	_eMC_GROUP_TYPE	轴构成	表示多轴协调控制中的轴构成 0: _mcXY (2 轴) 1: _mcXYZ (3 轴) 2: _mcXYZU (4 轴)
Axis[0]	UINT	构成轴选择 (轴 A0)	表示轴 A0 中分配的轴编号
Axis[1]	UINT	构成轴选择 (轴 A1)	表示轴 A1 中分配的轴编号
Axis[2]	UINT	构成轴选择 (轴 A2)	表示轴 A2 中分配的轴编号
Axis[3]	UINT	构成轴选择 (轴 A3)	表示轴 A3 中分配的轴编号

*1. 表示指令的控制状态。

*2. 以速度“0”动作时也视为正在运算的状态。

*3. 可在 CPU 单元版本 1.10 以上和 Sysmac Studio Ver.1.12 以上的组合中使用。

*4. 表示轴组的运动学转换所需的定义。

运动控制指令的输入变量

运动控制指令的输入变量一览和有效范围一览，以及列举型的有效范围一览。

输入变量一览

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。 Execute 上升时，导入其他输入变量。 要更新输入值时，将变更输入值，变为再次启动 Execute 的状态。 执行执行完成后，Execute 为 TRUE 期间，输出变量仍为有效。 然后，在 Execute 的下降沿，Error 和 ErrorID 以外的输出变量变为无效。 如果在指令执行完成前，将 Execute 设为 FALSE，输出变量将至少在 1 个周期内保持有效
Enable	有效	BOOL	TRUE、FALSE	FALSE	设为 TRUE 后，指令的功能将变为有效，设为 FALSE，则变为无效。 Enable 为 TRUE 期间，其他输入变量也会每个周期都导入。 将 Enable 设为 FALSE 后，Error 和 ErrorID 以外的输出变量变为无效
Positive Enable	正方向有效	BOOL	TRUE、FALSE	FALSE	<ul style="list-style-type: none"> MC_MoveJog 指令 若设为 TRUE，则开始向正方向移动。 若设为 FALSE，则结束移动。 MC_MoveJog 指令的 Velocity、Acceleration、Deceleration 将导入 PositiveEnable 上升时的值。 MC_SetTorqueLimit 指令 若设为 TRUE，则正方向的扭矩限制变为有效。若设为 FALSE，则正方向的扭矩限制变为无效。
Negative Enable	负方向有效	BOOL	TRUE、FALSE	FALSE	<ul style="list-style-type: none"> MC_MoveJog 指令 若设为 TRUE，则开始向负方向移动。 若设为 FALSE，则结束移动。 MC_MoveJog 指令的 Velocity、Acceleration、Deceleration 将导入 NegativeEnable 上升时的值。 MC_SetTorqueLimit 指令 若设为 TRUE，则负方向的扭矩限制变为有效。若设为 FALSE，则负方向的扭矩限制变为无效
BufferMode	缓冲模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 ^{*1}	指定运动控制指令多重启动时的动作。 0: 中断 1: 缓冲 2: 低共混 3: 前共混 4: 后共混 5: 高共混
Velocity	目标速度	LREAL	正数 ^{*2}	0	指定目标速度 ^{*3}
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度 ^{*4}
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度 ^{*4}
Jerk	跃度	LREAL	正数或“0”	0	指定跃度 ^{*5}

输入变量	名称	数据类型	有效范围	初始值	内容
Distance	移动距离	LREAL	负数、正数、“0”	0	指定指令当前位置起的移动距离
		ARRAY [0..3] OF LREAL	负数、正数、“0”	0	指定直线插补的目标位置 *6
Position	目标位置	LREAL	负数、正数、“0”	0	指定绝对坐标的目标位置 *6
		ARRAY [0..3] OF LREAL	负数、正数、“0”	0	指定直线插补的目标位置 *6
VelFactor	速度 超驰值	LREAL	0 ~ 500	100	指定速度的超驰值。 超驰值的有效范围为“0.01 ~ 500.00”。 “500.00 以上”视为“500”，“0.01 以下 (包括负数)”视为“0.01”。 仅在指定为“0”时作为“0”动作。 单位为 [%]
AccFactor (Reserved)	加减速度 超驰值	LREAL	0 ~ 500	100	(Reserved)
JerkFactor (Reserved)	跃度 超驰值	LREAL	0 ~ 500	100	(Reserved)
Reference Type *7	位置种类选择	_eMC_ REFERENCE_ TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 *1	指定主轴的输入信息。 0: 指令位置 (前一任务周期 *8 中的计算 值) 1: 反馈位置 (同一任务周期 *8 中的获取 值) 2: 指令位置 (同一任务周期 *8 中的计算 值)
Feed Distance	固定尺寸距离	LREAL	负数、正数、“0”	0	指定中断输入后的移动距离
FeedVelocity	固定尺寸速度	LREAL	正数	0	指定中断输入后的移动目标速度
ErrorDetect	错误检测选择	BOOL	TRUE、FALSE	FALSE	无中断输入时, 指定是否进行异常检测。 TRUE: 有异常检测 FALSE: 无异常检测
Periodic	重复模式	BOOL	TRUE、FALSE	FALSE	指定是否重复执行指定的凸轮表, 或只执 行 1 次。 TRUE: 重复 FALSE: 不重复
StartMode	开始位置方式 指定	_eMC_ START_MODE	0: _mcAbsolutePosition 1: _mcRelativePosition	0 *1	指定使用 MasterStartDistance(主轴追踪 距离) 的坐标系 0: 绝对位置 1: 相对位置
StartPosition	凸轮表 开始位置	LREAL	负数、正数、“0”	0	以主轴的绝对位置指定凸轮表的起点 (相 位 =0)
MasterStart Distance	主轴追踪距离	LREAL	负数、正数、“0”	0	指定从轴开始凸轮动作时主轴的位置。 StartMode (开始位置方式指定) 指定为 [绝对定位] 时, 指定主轴的绝对位置。指 定为相对定位时, 指定从 StartPosition (凸轮表开始位置) 起的相对量。
Master Scaling	主轴系数	LREAL	正值 (>0.0)	1.0	按指定的比例, 放大 / 缩小主轴的相位
SlaveScaling	从轴系数	LREAL	正值 (>0.0)	1.0	按指定的比例, 放大 / 缩小从轴的相位
MasterOffset	主轴偏置	LREAL	负数、正数、“0”	0	按指定的偏置值, 错开主轴的相位
SlaveOffset	从轴偏置	LREAL	负数、正数、“0”	0	按指定的偏置值, 错开从轴的相位

输入变量	名称	数据类型	有效范围	初始值	内容
Cam Transition (Reserved)	凸轮转换指定	_eMC_CAM_TRANSITION	0: _mcCTNone	0 ^{*1}	(Reserved)
OutMode (Reserved)	同步解除模式选择	_eMC_OUT_MODE	0: _mcStop	0 ^{*1}	(Reserved)
Direction	方向选择	_eMC_DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	4/0 ^{*1}	指定动作的方向 0: 正方向指定 1: 最短距离指定 2: 负方向指定 3: 当前方向指定 4: 无方向指定
Continuous (Reserved)	持续方向选择	BOOL	TRUE、FALSE	FALSE	(Reserved)
Ratio Numerator	齿轮比分子	DINT ^{*9}	正数或负数 ^{*9}	10000	指定主轴和从轴间电子齿轮的分子
Ratio Denominator	齿轮比分母	UDINT ^{*10}	正数	10000	指定主轴和从轴间电子齿轮的分母
MasterSync Position	主轴同步位置	LREAL	负数、正数、“0”	0	指定绝对坐标的主轴同步位置
SlaveSync Position	从轴同步位置	LREAL	负数、正数、“0”	0	指定绝对坐标的从轴同步位置
Slave Distance	从轴移动距离	LREAL	负数、正数、“0”	0	指定从轴的移动距离
Master Distance	主轴移动距离	LREAL	正数或“0”	0	指定主轴的移动距离
Master Distanceln ACC	主轴加速移动距离	LREAL	正数或“0”	0	指定从轴加速时主轴的移动距离
Master Distanceln DEC	主轴减速移动距离	LREAL	正数或“0”	0	指定从轴减速时主轴的移动距离
LinkOption	同步开始条件选择	_eMC_LINKOPTION	0: _mcCommandExecution 1: _mcTriggerDetection 2: _mcMasterReach	0 ^{*1}	指定从轴与主轴同步的条件 0: 启动时 1: 检测到触发时 2: 主轴到达主轴追踪距离时
Combine Mode	加减法选择	_eMC_COMBINE_MODE	0: _mcAddAxes 1: _mcSubAxes	0 ^{*1}	指定合成方法 0: 加法 1: 减法
Ratio Numerator Master (Reserved)	主轴齿轮比分子	DINT ^{*9}	正数或负数 ^{*9}	10000	指定主轴和从轴间电子齿轮的分子
Ratio DenominatorMaster (Reserved)	主轴齿轮比分母	UDINT ^{*10}	正数	10000	指定主轴和从轴间电子齿轮的分母
Ratio Numerator Auxiliary (Reserved)	辅助轴齿轮比分子	DINT ^{*9}	正数或负数 ^{*9}	10000	指定辅助轴和从轴间电子齿轮的分子
Ratio DenominatorAuxiliary (Reserved)	辅助轴齿轮比分母	UDINT ^{*10}	正数	10000	指定辅助轴和从轴间电子齿轮的分母
Reference TypeMaster	主轴位置类别选择	_eMC_REFERENCE_TYPE	1: _mcFeedback 2: _mcLatestCommand	2 ^{*1}	指定主轴的位置种类 1: 反馈位置 (同一任务周期 ^{*8} 中的获取值) 2: 指令位置 (同一任务周期 ^{*8} 中的计算值)
Reference Type Auxiliary	辅助轴位置类别选择	_eMC_REFERENCE_TYPE	1: _mcFeedback 2: _mcLatestCommand	2 ^{*1}	指定辅助轴的位置种类 1: 反馈位置 (同一任务周期 ^{*8} 中的获取值) 2: 指令位置 (同一任务周期 ^{*8} 中的计算值)

输入变量	名称	数据类型	有效范围	初始值	内容
PhaseShift	相位补偿量	LREAL	负数、正数、“0”	0	指定主轴的相位补偿量 ⁶
Torque	目标扭矩	LREAL	0 ~ 1000.0	300.0	以 0.1% 为单位指定输出到伺服驱动器的目标扭矩。 将额定扭矩视为 100%，按比例指定。单位为“%”
TorqueRamp	扭矩斜率	LREAL	正数或“0”	0	指定从当前扭矩到输出目标扭矩为止的扭矩变化率。 单位为“%/s”
Positive Value	正方向扭矩限制值	LREAL	0.1 ~ 1000.0、 或“0.0”	300.0	以 0.1% 为单位指定正方向的扭矩限制值。 如果输入的值超出轴参数中的 [正方向扭矩限制上限值]，将变为 [正方向扭矩限制上限值]。 指定为“0”及负数时，作为“0”动作
Negative Value	负方向扭矩限制值	LREAL	0.1 ~ 1000.0、 或“0.0”	300.0	以 0.1% 为单位指定负方向的扭矩限制值。 如果输入的值超出轴参数中的 [负方向扭矩限制上限值]，将变为 [负方向扭矩限制上限值]。 指定为“0”及负数时，作为“0”动作
WindowOnly	窗口有效	BOOL	TRUE、FALSE	FALSE	指定窗口主机的有效 / 无效
FirstPosition	开始位置	LREAL	负数、正数、“0”	0	指定开始位置。
LastPosition	结束位置	LREAL	负数、正数、“0”	0	指定结束位置。
StopMode	停止方法选择	_eMC_STOP_MODE	1: _mcImmediateStop 2: _mcImmediateStop FEReset 3: _mcFreeRunStop 4: _mcNonStop	4 ^{*1}	指定停止方法 1: 立即停止 2: 立即停止，进行偏差计数器复位 3: 立即停止，变为伺服 OFF 4: 不停止
Relative (Reserved)	相对位置选择	BOOL	TRUE、FALSE	FALSE	(Reserved)
Execution Mode (Reserved)	执行模式选择	_eMC_EXECUTION_MODE	0: _mcImmediately	0 ^{*1}	(Reserved)
Permitted Deviation	轴间偏差容许值	LREAL	正数或“0”	0	指定容许的主轴和从轴的位置偏差最大值
CmdPos Mode	指令当前位置计数选择	_eMC_CMDPOS_MODE	0: _mcCount	0 ^{*1}	0: 使用反馈当前位置，更新指令当前位置。 保持原点确定状态
CoordSystem	坐标系	_eMC_COORD_SYSTEM	0: _mcACS	0 ^{*1}	指定坐标系 0: 轴坐标系 (ACS)
Transition Mode	转换模式 (变化模式)	_eMC_TRANSITION_MODE	0: _mcTMNone 10: _mcTMCorner Superimposed	0 ^{*1}	指定动作的路径 0: 转换无效 10: 重叠
MoveMode	移动方法选择	_eMC_MOVE_MODE	0: _mcAbsolute 1: _mcRelative	0 ^{*1}	选择移动方法 0: 绝对值定位 1: 相对值定位
CircAxes	圆弧轴指定	ARRAY [0,1] OF UINT	0 ~ 3	0	指定进行圆弧插补的轴 0: 轴 A0 1: 轴 A1 2: 轴 A2 3: 轴 A3
CircMode	圆弧插补模式	_eMC_CIRC_MODE	0: _mcBorder 1: _mcCenter 2: _mcRadius	0 ^{*1}	指定圆弧插补的方法 0: 通过点指定 1: 中心点指定 2: 半径指定
AuxPoint	辅助点	ARRAY [0,1] OF LREAL	负数、正数、“0”	0	指定通过点位置 / 中心位置 / 半径
EndPoint	终点	ARRAY [0,1] OF LREAL	负数、正数、“0”	0	指定目标位置

输入变量	名称	数据类型	有效范围	初始值	内容
PathChoice	路径选择	_eMC_CIRC_PATHCHOICE	0: _mcCW 1: _mcCCW	0 ^{*1}	指定路径方向 0: CW 1: CCW
Parameter Number	参数编号	_eMC_PARAMETER_NUMBER	0: _mcChkVel 1: _mcChkAcc 2: _mcChkDec 3: _mcPosiChkTrq 4: _mcNegaChkTrq 5: _mcFELmt 6: _mcChkFELmt 7: _mcSwLmtMode 8: _mcPosiSwLmt 9: _mcNegaSwLmt 10: _mcInPosTime 11: _mcInPosRange ^{*11} 12: _mcStartVel ^{*12}	0 ^{*1}	指定写入目标的参数 0: 速度警告值 / 插补速度警告值 1: 加速度警告值 / 插补加速度警告值 2: 减速度警告值 / 插补减速度警告值 3: 正方向扭矩警告值 4: 负方向扭矩警告值 5: 位置偏差超限值 6: 位置偏差警告值 7: 软件限位功能 8: 正方向软件限位 9: 负方向软件限位 10: 位置检查时间 11: 位置宽度 12: 启动速度
HomingMode ^{*13}	原点复位动作	_eMC_HOMING_MODE	0: _mcHomeSwTurnHomeSwOff 1: _mcHomeSwTurnHomeSwOn 4: _mcHomeSwOff 5: _mcHomeSwOn 8: _mcLimitInputOff 9: _mcHomeSwTurnHomeMask 11: _mcLimitInputOnly 12: _mcHomeSwTurnHoldingTime 13: _mcNoHomeSwHoldingHomeInput 14: _mcHomePreset	0 ^{*1}	指定要重写的 [原点复位动作]。 0: 附近退避、近原点输入 OFF 指定 1: 附近退避、近原点输入 ON 指定 4: 近原点输入 OFF 指定 5: 近原点输入 ON 指定 8: 极限输入 OFF 指定 9: 附近退避、原点输入掩码距离指定 11: 仅极限输入 12: 附近退避、推压时间指定 13: 无近原点输入、推压原点输入指定 14: 原点预设
AxisUse ^{*14}	轴使用	_eMC_AXIS_USE	1: _mcUnusedAxis 2: _mcUsedAxis	1 ^{*1}	指定使用轴或未使用轴。 1: 未使用轴 2: 使用轴
EnableMask ^{*15}	轨道有效	WORD	16#0000 ~ FFFF	0	指定各轨道的有效或无效。最多 16 个轨道，第 0 位的值为轨道编号 0 的有效无效，第 15 位为轨道编号 15 的有效无效。 0: 无效、1: 有效
ValueSource (Reserved)	输入信息	_sMC_SOURCE	—	—	(Reserved)
TimeStamp ^{*15}	时间戳	ULINT	正数、“0”	0	指定用于计算位置的时间戳。 指定以支持时间戳方式的数字输入单元或编码器输入单元中的时刻为基准的时间戳。单位为 [ns]。
ExecID ^{*16}	执行 ID	UINT	2	2	指定变量同步指定目标的 ID。 2: 固定周期任务（执行优先级 5）
OffsetPosition	位置偏置	LREAL	负数、正数、“0”	0	指定要加到指令当前位置的位置偏置 ^{*6}

*1. 如果是有效范围为枚举型的变量，实际初始值不是数值，而是枚举值。

*2. 包括 MC_MoveJog 指令及 MC_MoveVelocity 指令等可以设定为“0”的指令。

*3. 变为指令单位 /s。指令单位以 [mm]、[μm]、[nm]、[degree]、[inch] 或 [pulse] 表示。

*4. 变为指令单位 /s²。

*5. 变为指令单位 /s³。

*6. 变为指令单位。

*7. 使用 _mcLatestCommand 时，使用的主轴和从轴必须满足以下关系。

Master（主轴）中设定的运动控制系统变量的轴编号 < Slave（从轴）中设定的运动控制系统变量的轴编号

*8. 任务为主固定周期任务时，任务周期为主周期，为固定周期任务（执行优先级 5）时，任务周期为固定周期任务（执行优先级 5）的任务周期。

- *9. 可在 CPU 单元版本 1.02 以上和 Sysmac Studio Ver.1.03 以上的组合中使用。在更低版本的组合中，数据类型为“UINT”，有效范围为“正数”。
- *10. 可在 CPU 单元版本 1.02 以上和 Sysmac Studio Ver.1.03 以上的组合中使用。在更低版本的组合中，数据类型为“UINT”。
- *11. 可在 CPU 单元版本 1.01 以上和 Sysmac Studio Ver.1.02 以上的组合中使用。
- *12. 可在 CPU 单元版本 1.10 以上和 Sysmac Studio Ver.1.12 以上的组合中使用。
- *13. 可在 CPU 单元版本 1.03 以上和 Sysmac Studio Ver.1.04 以上的组合中使用。
- *14. 可在 CPU 单元版本 1.04 以上和 Sysmac Studio Ver.1.05 以上的组合中使用。
- *15. 可在 CPU 单元版本 1.06 以上和 Sysmac Studio Ver.1.07 以上的组合中使用。
- *16. NJ 系列 CPU 单元中无。

输入变量的有效范围

下面介绍运动控制指令的输入变量有效范围。各指令的有效范围请参考各指令的说明。

● BOOL 型输入变量

FALSE 以外视为 TRUE。因此，不会发生超出范围的异常。

● 列举（ENUM）型输入变量

超出有效范围时视为异常。

● 有效范围记载为“全范围”、“正数”、“负数”的输入变量

设定了各输入变量的有效范围和超出有效范围的数值时，动作如下表所示。



输入变量	名称	有效范围	超出最大值范围	超出最小值范围 (0 除外)
Velocity	速度	0 或 ($-1 \leq \text{且} \leq \text{“最高速度”}$)、 ($1 \leq \text{且} \leq \text{“最高速度”}$) ^{*1}	正数时：“最高速度” 负数时：“-“最高速度”” 分别固定 ^{*2}	正数时：1pulse/s 负数时：-1pulse/s 分别固定 ^{*2}
Acceleration	加速度	0 或 ($0.004 \leq \text{且} \leq \text{“最大加速度”}$) ^{*3}	固定为最大加速度。 但是，加速时间小于 ^{*4} 125 μ s 时，固定为 125 μ s。	正数时，固定为 0.004pulse/s ² 。 但是，加速时间 ^{*4} 超过 250s 时，固定为 250s。 负数时：发生异常
Deceleration	减速度	0 或 ($0.004 \leq \text{且} \leq \text{“最大减速度”}$) ^{*5}	固定为最大减速度。 但是，减速时间 ^{*4} 小于 125 μ s 时，固定为 125 μ s。	正数时，固定为 0.004pulse/s ² 。 但是，减速时间 ^{*4} 超过 250s 时，固定为 250s。 负数时：发生异常
Jerk	跃度	0 或 ($0.000016 \leq \text{且}$ $25,600,000,000,000,000 \text{ pulse/s}^3$)	固定为 25,600,000,000,000,000 pulse/s ³ 。 但是，加速跃度适用时间 ^{*6} 或减速跃度适用时间 ^{*6} 小于 125 μ s 时，分别固定为 125 μ s	固定为 0.000016 pulse/s ³ 。 但是，加速跃度适用时间 ^{*6} 或减速跃度适用时间 ^{*6} 超过 250s 时，分别固定为 250s。 负数时，发生异常。

输入变量	名称	有效范围	超出最大值范围	超出最小值范围 (0 除外)
Distance	移动距离	$(0x\text{FFFFFF}0000000001) \leq \text{且} \leq (0x000000\text{FFFFFFFF})$	发生异常	不会超出最小值范围
Position	指令位置	$(0x\text{FFFFFF}8000000000) \leq \text{且} < (0x0000007\text{FFFFFFFF}+1)^{*7}$	发生异常	不会超出最小值范围
VelFactor	超驰值	0 或 $0.01 \leq \text{且} \leq 500.00^{*8}$	超过 500.00% 时, 固定为 500.00	小于 0.01% 时, 固定为 0.01%
Velocity	插补速度	$0.000\ 000\ 000\ 1 \leq \text{且} \leq \text{“最高插补速度”}^{*9}$	固定为“最高插补速度”。	固定为 0.000 000 000 1 pulse/s
Acceleration	插补加速度	0 或 $(0.000\ 000\ 000\ 000\ 4 \leq \text{且} \leq \text{“最大插补加速度”})^{*10}$	固定为“最大插补加速度”。但是, 插补加速时间小于 $^{*11} 125\mu\text{s}$ 时, 固定为 $125\mu\text{s}$	正数时: 固定为 0.000 000 000 000 4 pulse/s ² 。但是, 插补加速时间 *11 超过 250s 时, 固定为 250s。 负数时: 发生异常
Deceleration	插补减速度	0 或 $(0.000\ 000\ 000\ 000\ 000\ 04 \leq \text{且} \leq \text{“最大插补减速度”})^{*12}$	固定为“最大插补减速度”。但是, 插补减速时间 *11 小于 $125\mu\text{s}$ 时, 固定为 $125\mu\text{s}$	正数时: 固定为 0.000 000 000 000 4 pulse/s ² 。但是, 插补减速时间 *11 超过 250s 时, 固定为 250s。 负数时: 发生异常
Jerk	插补跃度	0 或 $(0.000\ 000\ 000\ 000\ 001\ 6 \leq \text{且} \leq 51,200,000,000,000,000\ \text{pulse/s}^3)$	固定为 51,200,000,000,000,000 pulse/s ³ 。但是, 插补加速跃度适用时间 *13 或插补减速跃度适用时间 *13 小于 $125\mu\text{s}$ 时, 分别固定为 $125\mu\text{s}$	固定为 0.000 000 000 000 001 6 pulse/s ³ 。但是, 插补加速跃度适用时间 *13 或插补减速跃度适用时间 *13 超过 250s 时, 分别固定为 250s。 负数时, 发生异常

*1. 可设定的最大值为转换为脉冲单位的值, 如下所示。

单元版本 1.02 以下的 CPU 单元中, 400,000,000[pulse/s];

单元版本 1.03 以上的 CPU 单元中, 500,000,000[pulse/s];

单元版本 1.11 以上的 CPU 单元中, 2,147,483,647[pulse/s]。

*2. 有效范围中不含负数或“0”时, 若指定为负数或“0”, 将发生异常。

*3. 轴参数“最大加速度”的上限为 3,200,000,000,000 pulse/s²。

*4. 按“加速时间=速度/加速度”、“减速时间=速度/减速度”、“加减速时间=加速时间+减速时间”计算。

*5. 轴参数“最大减速度”的上限为 3,200,000,000,000 pulse/s²。

*6. “加速跃度适用时间”、“减速跃度适用时间”是指适用跃度的时间。

按“加速跃度适用时间=加速度/跃度”、“减速跃度适用时间=减速度/跃度”计算。

*7. 脉冲单位的绝对位置, 带符号需要在 40bit 的范围内。

*8. 单位为“%”。

*9. 轴参数“最高插补速度”的上限为轴参数“最高速度”上限的 2 倍。

*10. 轴参数“最大插补加速度”的上限为 6,400,000,000,000 pulse/s²。

*11. 按“插补加速时间=插补速度/插补加速度”、“插补减速时间=插补速度/插补减速度”、“加减速时间=加速时间+减速时间”计算

*12. 轴参数“最大插补减速度”的上限为 6,400,000,000,000 pulse/s²。

*13. “插补加速跃度适用时间”、“插补减速跃度适用时间”是指适用插补跃度的时间。

按“插补加速跃度适用时间=加速度/跃度”、“插补减速跃度适用时间=减速度/跃度”计算。

列举型一览

运动控制指令的输入变量中处理的 ENUM 型一览。

列举型输入变量利用列举值进行设定，而不是数值。

数据类型	有效范围	内容	适用指令的变量 (变量的名称)
_eMC_BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	指定运动控制指令多重启动时的动作 0: 中断 1: 缓冲 2: 低共混 3: 前共混 4: 后共混 5: 高共混	BufferMode (缓冲模式选择)
_eMC_CIRC_ MODE	0: _mcBorder 1: _mcCenter 2: _mcRadius	指定圆弧插补的方法 0: 通过点指定 1: 中心点指定 2: 半径指定	CircMode (圆弧插补模式)
_eMC_CAM_ TRANSITION	0: _mcCTNone	指定凸轮重启时从轴指令值的输出方法 0: 无限制和修改	CamTransition (凸轮转换指定)
_eMC_CIRC_ PATHCHOICE	0: _mcCW 1: _mcCCW	指定路径方向 0: CW 1: CCW	PathChoice (路径选择)
_eMC_COMBINE_ MODE	0: _mcAddAxes 1: _mcSubAxes	指定合成方法 0: 加法 1: 减法	CombineMode (加减法选择)
_eMC_COORD_ SYSTEM	0: _mcACS	指定坐标系 0: 轴坐标系 (ACS)	CoordSystem (坐标系)
eMC DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	指定各种动作的方向 0: 正方向指定 1: 最短距离指定 2: 负方向指定 3: 当前方向指定 4: 无方向指定	Direction (方向选择)
eMC EXECUTION_ MODE	0: _mcImmediately	(Reserved)	ExecutionMode (执行模式选择)
eMC LINKOPTION	0: _mcCommandExecution 1: _mcTriggerDetection 2: _mcMasterReach	指定从轴与主轴同步的条件 0: 指令开始时 1: 检测到触发时 2: 主轴到达主轴追踪距离时	LinkOption (同步开始条件选择)
_eMC_MOVE_ MODE	0: _mcAbsolute 1: _mcRelative 2: _mcVelocity	选择移动方法 0: 绝对值定位 1: 相对值定位 2: 速度控制	MoveMode (移动方法选择)
eMC _OUT_MODE	0: _mcStop	指定解除同步控制指令的模式 0: 减速停止	OutMode (同步解除模式选择) (Reserved)

数据类型	有效范围	内容	适用指令的变量 (变量的名称)
eMC PARAMETER_ NUMBER	0: _mcChkVel 1: _mcChkAcc 2: _mcChkDec 3: _mcPosiChkTrq 4: _mcNegaChkTrq 5: _mcFELmt 6: _mcChkFELmt 7: _mcSwLmtMode 8: _mcPosiSwLmt 9: _mcNegaSwLmt 10: _mcInPosTime 11: _mcInPosRange* ¹ 12: _mcStartVel* ²	指定写入目标的参数 0: 速度警告值 / 插补速度警告值 1: 加速度警告值 / 插补加速度警告值 2: 减速度警告值 / 插补减速度警告值 3: 正方向扭矩警告值 4: 负方向扭矩警告值 5: 位置偏差超限值 6: 位置偏差警告值 7: 软件限位功能 8: 正方向软件限位 9: 负方向软件限位 10: 位置检查时间 11: 位置宽度 12: 启动速度	ParameterNumber (参数编号)
eMC SWLMT_MODE	0: _mcNonSwLmt 1: _mcCmdDecelerationStop 2: _mcCmdImmediateStop 3: _mcActDecelerationStop 4: _mcActImmediateStop	指定软件限位功能的有效 / 无效及停止方法 0: 软件限位功能无效 1: 有效, 对指令位置减速停止 2: 有效, 对指令位置立即停止 3: 有效, 对反馈位置减速停止 4: 有效, 对反馈位置立即停止	SettingValue (设定值)
eMC REFERENCE_ TYPE * ³	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	指定位置的类别 0: 指令位置 (前一任务周期* ⁴ 中的计算值) 1: 反馈位置 (同一任务周期* ⁴ 中的获取值) 2: 指令位置 (同一任务周期* ⁴ 中的计算值)	ReferenceType (位置种类选择) ReferenceTypeMaster (主轴位置种类选择) ReferenceTypeAuxiliary (辅助轴位置种类选择)
eMC START_MODE	0: _mcAbsolutePosition 1: _mcRelativePosition	指定使用 MasterStartDistance(主轴追踪距离) 的坐标系 0: 绝对位置 1: 相对位置	StartMode (开始位置方式指定)
_eMC_STOP_ MODE	0: _mcDecelerationStop 1: _mcImmediateStop 2: _mcImmediateStopFEReset 3: _mcFreeRunStop 4: _mcNonStop	指定停止方法 0: 减速停止指令值 1: 立即停止 2: 立即停止, 进行偏差计数器复位 3: 伺服 OFF 4: 不停止	StopMode (停止方法选择)
_eMC_TRIGGER_ LATCH_ID	0: _mcLatch1 1: _mcLatch2	指定使用两个锁定功能中的哪个。 0: 锁定功能 1 1: 锁定功能 2	LatchID (锁定 ID)
eMC CMDPOS_MODE	0: _mcCount	0: 使用反馈当前位置, 更新指令当前位置保持原点确定状态	CmdPosMode (指令当前位置计数选择)
eMC TRANSITION_ MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	指定动作的路径 0: 转换无效 10: 重叠	TransitionMode (转换模式 (变化模式))
_eMC_TRIGGER_ MODE	0: _mcDrive 1: _mcController	指定触发的模式 0: 驱动器模式 1: 控制器模式	Mode (模式)

数据类型	有效范围	内容	适用指令的变量 (变量的名称)
_eMC_TRIGGER_ INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	指定驱动器模式时的触发信号 0: Z相 1: 外部输入	InputDrive (驱动器触发 输入信号)
_eMC_HOMING_ MODE *5	0: _mcHomeSwTurnHomeSw Off 1: _mcHomeSwTurnHomeSwOn 4: _mcHomeSwOff 5: _mcHomeSwOn 8: _mcLimitInputOff 9: _mcHomeSwTurnHomeMask 11: _mcLimitInputOnly 12: _mcHomeSwTurnHolding Time 13: _mcNoHomeSwHolding HomeInput 14: _mcHomePreset	指定要重写的 [原点复位动作] 0: 附近退避、近原点输入 OFF 指定 1: 附近退避、近原点输入 ON 指定 4: 近原点输入 OFF 指定 5: 近原点输入 ON 指定 8: 极限输入 OFF 指定 9: 附近退避、原点输入掩码距离指定 11: 仅极限输入 12: 附近退避、推压时间指定 13: 无近原点输入、推压原点输入指 定 14: 原点预设	—
_eMC_HOME_ INPUT *5	0: _mcZPhase 1: _mcExternalSignal	选择原点输入信号 0: 使用 Z 相输入 1: 使用外部原点输入	—
_eMC_LIMIT_ REVERSE_MODE *5	0: _mcErrorStop 1: _mcRevImmediateStop 2: _mcRevDecelerationStop	设定在原点复位过程中将极限输入设 为“ON”时的停止方法 0: 不反转、轴异常停止 (停止方法遵 照极限输入停止方法) 1: 反转、立即停止 2: 反转、减速停止	—
_eMC_CAM_CUR VE *6	0: _mcConstantLine 1: _mcStraightLine 2: _mcParabolic 3: _mcModifiedConstant Vel 4: _mcModifiedTrapezoid 5: _mcModifiedSine 6: _mcCycloidal 7: _mcTrapeclloid 8: _mcReverseTrapeclloid 9: _mcSimpleHarmonic 10: _mcDoubleHarmonic 11: _mcReverseDouble Harmonic 12: _mcNC2Curve 13: _mcPolynomic3 14: _mcPolynomic5	指定到节点为止的凸轮曲线形状。 0: 直线 (保持位移) 1: 直线 2: 等加速度 3: 变形等速度 4: 变形梯形 5: 变形正弦 6: 摆线 7: Trapeclloid 8: 反 Trapeclloid 9: 单弦 10: 复弦 11: 反复弦 12: NC2 曲线 13: 3 次曲线 14: 5 次曲线	Curve (曲线形状)
eMC ACCDECOVER *6	0: _mcAccDecOverBuffer 1: _mcAccDecOverRapid 2: _mcAccDecOverError Stop	在轴的加减速控制中, 为了优先停止 在目标位置, 会发生减速超限, 导致 超出最大加减速速度, 指定这种情况 下的动作。 0: 剧烈加减速 (将共混切换为缓冲) ^{*7} 1: 剧烈加减速 2: 异常停止 ^{*8}	—
eMC REVERSE_MODE *6	0: _mcReverseMode DecelerationStop 1: _mcReverseMode ImmediateStop	指定指令多重启动时、指令重新启动 时、中断固定尺寸定位时的反转时动 作。 0: 减速停止 1: 立即停止	—

数据类型	有效范围	内容	适用指令的变量 (变量的名称)
eMC COUNT_MODE *6	0: _mcCountModeLinear 1: _mcCountModeRotary	设定位置的计数模式。 0: 直线模式 (长度有限) 1: 旋转模式 (长度无限)	—
_eMC_UNITS *6	0: _mcPls 1: _mcMm 2: _mcUm 3: _mcNm 4: _mcDeg 5: _mcInch	设定指令位置的单位。 0: pulse 1: mm 2: μm 3: nm 4: degree 5: inch	—

- *1. 可在 CPU 单元版本 1.01 以上和 Sysmac Studio Ver.1.02 以上的组合中指定。
- *2. 可在 CPU 单元版本 1.10 以上和 Sysmac Studio Ver.1.12 以上的组合中指定。
- *3. 使用 _mcLatestCommand 时, 使用的主轴和从轴必须满足以下关系。
Master (主轴) 中设定的运动控制系统变量的轴编号 < Slave (从轴) 中设定的运动控制系统变量的轴编号
- *4. 任务为主固定周期任务时, 任务周期为主周期, 为固定周期任务 (执行优先级 5) 时, 任务周期为固定周期任务 (执行优先级 5) 的任务周期。
- *5. 可在 CPU 单元版本 1.03 以上和 Sysmac Studio Ver.1.04 以上的组合中指定。
- *6. 可在 CPU 单元版本 1.08 以上和 Sysmac Studio Ver.1.09 以上的组合中指定。
- *7. 单元版本 1.10 以上的 CPU 单元中, 无法将共混切换为缓冲。详情请参考 [] 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。
- *8. 单元版本 1.10 以上的 CPU 单元中, 发生共混动作时, 将视为发生异常而停止。详情请参考 [] 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

运动控制指令的输出变量

运动控制指令的输出变量一览如下。

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。 此时，输出变量 Active、Error、CommandAborted 变为 FALSE。 指令完成时，若输入变量 Execute 为 FALSE，则 Done 至少在 1 个周期内保持 TRUE。 Execute 为 TRUE 时，Done 在 Execute 变为 FALSE 前保持 TRUE
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。 控制中是指该指令正在实际控制轴或轴组。此时，输出变量 Done、Error、CommandAborted 变为 FALSE。
Enabled	有效	BOOL	TRUE、FALSE	控制过程中变为 TRUE
CommandAborted	执行中断	BOOL	TRUE、FALSE	无法启动指令或在执行过程中发生中断时，变为 TRUE。 对象轴或轴组发生异常时，无法启动指令。同样，正在减速停止时，也无法启动指令。 启动了其他指令或发生本指令以外的异常时，指令将中断。 此时，输出变量 Done、Active、Error 变为 FALSE。 若在输入变量 Execute 为 FALSE 时发生中断，CommandAborted 至少会在 1 个周期内保持 TRUE。 Execute 或 Enable 为 TRUE 时，Execute 或 Enable 变为 FALSE 前，CommandAborted 保持 TRUE
Error	错误	BOOL	TRUE、FALSE	输入变量错误或指令处理发生异常时，变为 TRUE *1
ErrorID	错误代码	WORD	*2	发生异常时，输出错误代码。 16#0000 表示正常
Failure	错误结束	BOOL	TRUE、FALSE	未正常执行指令时变为 TRUE
Status	可运行	BOOL	TRUE、FALSE	变为可运行状态时变为 TRUE
EndOfProfile	凸轮周期完成	BOOL	TRUE、FALSE	执行了凸轮表的终点时变为 TRUE
Index	索引	UINT	正数或“0”	输出凸轮数据的索引
StartSync	追踪中	BOOL	TRUE、FALSE	为了同步而开始加减速动作时变为 TRUE
RecordedPosition	锁定位置	LREAL	负数、正数、“0”	输出锁定的位置 *3
Invalid	轴间偏差超限	BOOL	TRUE、FALSE	超出轴间偏差容许值时变为 TRUE
DeviatedValue	轴间偏差值	LREAL	负数、正数、“0”	输出指定主轴和从轴的差 *3
EndPointIndex	终点索引	UINT	正数或“0”	输出凸轮表的终点索引

输出变量	名称	数据类型	有效范围	内容
MaxDataNumber	最大凸轮数据数	UINT	正数	输出最大凸轮数据数
InVelocity	达到目标速度	BOOL	TRUE、FALSE	达到目标速度时，变为 TRUE
InSync	同步中	BOOL	TRUE、FALSE	已与主轴同步或从轴到达从轴同步位置时，变为 TRUE
InGear	达到齿轮比	BOOL	TRUE、FALSE	从轴达到目标速度时，变为 TRUE
InCombination	加减运算中	BOOL	TRUE、FALSE	加减运算过程中变为 TRUE
InCam	凸轮动作中	BOOL	TRUE、FALSE	执行了凸轮表的起点时变为 TRUE
InTorque	达到目标扭矩	BOOL	TRUE、FALSE	达到目标扭矩时，变为 TRUE
InFeed	固定尺寸输送中	BOOL	TRUE、FALSE	接收锁定输入后，进行固定尺寸输送时变为 TRUE
InZone	区域内	BOOL	TRUE、FALSE	轴位置在区域范围内时变为 TRUE
Valid ^{*4}	有效	BOOL	TRUE、FALSE	控制过程中变为 TRUE
Command Position ^{*4}	指令当前位置	ARRAY [0..3] OF LREAL	负数、正数、“0”	输出指令位置的当前值 ^{*3}
ActualPosition ^{*4}	反馈当前位置	ARRAY [0..3] OF LREAL	负数、正数、“0”	输出反馈位置的当前值 ^{*3}
InPosition ^{*4}	位置	BOOL	TRUE、FALSE	所有构成轴的反馈当前位置进入目标位置的位置宽度内时变为 TRUE
InOperation ^{*5}	动作中	BOOL	TRUE、FALSE	指令的动作过程中变为 TRUE。
CalcPosition ^{*5}	计算位置	LREAL	负数、正数、“0”	输出指定时间戳的位置。 ^{*3}
ErrorParameterCode ^{*6}	参数详细代码	WORD	*7	有些错误代码存在附属信息。存在时，输出发生异常的参数之详细代码。
ErrorNodePointIndex ^{*6}	节点要素编号	UINT	*7	有些错误代码存在附属信息。存在时，输出发生异常的节点要素编号。
OutputtedOffsetPosition ^{*8}	位置偏置输出值	LREAL	负数、正数、“0”	指定已加到指令当前位置的位置偏置。 Active 为 TRUE 时更新值。 CommandAborted 或 Error 为 TRUE 时，停止更新，保持值。

*1. Error (错误) 不会清零，直至执行 MC_Reset (轴错误重置) 指令、MC_GroupReset (轴组错误重置) 指令或 Reset-MCError (所有错误重置) 指令。

这个规格与 PLCopen[®] 不同。PLCopen[®] 中，在 Execute (启动) 的下降沿，变为 FALSE。

此外，Error 为 TRUE 期间，不会启动运动控制指令。解除异常后，即使 Execute 为 TRUE，由于不是上升沿，指令也不会启动。Enable 型运动控制指令在 Enable (有效) 变为 TRUE 时启动。

*2. 与事件代码的前面 4 位相同的数字。详情请参考 □「A-1 错误代码一览 (P.A-2)」。

*3. 变为指令单位。指令单位以 [mm]、[μm]、[nm]、[degree]、[inch] 或 [pulse] 表示。

*4. 可在 CPU 单元版本 1.01 以上和 Sysmac Studio Ver.1.02 以上的组合中使用。

*5. 可在 CPU 单元版本 1.06 以上和 Sysmac Studio Ver.1.07 以上的组合中使用。

*6. 可在 CPU 单元版本 1.08 以上和 Sysmac Studio Ver.1.09 以上的组合中指定。

*7. 详情请参考 □「A-1 错误代码一览 (P.A-2)」。

*8. 可在 CPU 单元版本 1.10 以上和 Sysmac Studio Ver.1.12 以上的组合中使用。



参考

变更运行模式后，若要参照运动控制指令的输出变量，请将有保持属性的变量分配到输出参数中。

通过参照分配的输出参数，可以参照变更运行模式前的输出变量。

运动控制指令的输入输出变量

运动控制指令的输入输出变量一览如下。

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴
AxesGroup	轴组	_sGROUP_REF	—	指定轴组
Auxiliary	辅助轴	_sAXIS_REF	—	指定辅助轴
Master	主轴	_sAXIS_REF	—	指定主轴
Slave	从轴	_sAXIS_REF	—	指定从轴
CamTable	凸轮表	ARRAY[0..N] OF _sMC_CAM_REF	—	将凸轮数据结构体 _sMC_CAM_REF 型的排列变量指定为凸轮表 ^{*1}
TriggerInput	触发输入条件	_sTRIGGER_REF	—	设定触发条件
TriggerVariable	触发变量	BOOL	TRUE、 FALSE	在触发条件中指定了控制器模式时，指定作为触发的输入变量
Target	写入对象	_sAXIS_REF 或 _sGROUP_REF	—	指定作为写入对象的轴或轴组
SettingValue	设定值	取决于指定变量的 类型	—	指定要写入的值。 有效范围遵照 ParameterNumber（参数编号） 中指定的运动控制参数。 ^{*2} 初始值为“0”
Axes ^{*3}	轴组构成轴	ARRAY [0..3] OF UINT	—	指定重写后构成轴的轴编号
Homing Parameter ^{*4}	原点复位参数	_sHOMING_REF	—	设定原点复位参数。
Switches ^{*5}	开关	ARRAY[0..255] OF _sCAMSWITCH_ REF	—	将开关结构体 _sCAMSWITCH_REF 型排列变量作为开关的 ON/OFF 模式数据指定。 排列要素编号表示开关编号。
Outputs ^{*5}	输出信号	ARRAY[0..15] OF _sOUTPUT_REF	—	指定输出信号 _sOUTPUT_REF 型的排列变量， 作为根据开关的 ON/OFF 模式数据计算得到的 数字输出 ON/OFF 时刻的输出目标。排列要素 编号表示轨道编号。 将该排列变量指定为 NX_AryDOutTimeStamp 指令的输入输出变量，可对实际的数字输出进行 ON/OFF 控制。
TrackOptions ^{*5}	轨道选项	ARRAY[0..15] OF _sTRACK_REF	—	将轨道选项结构体 _sTRACK_REF 型排列变量 作为开关的动作条件指定。排列要素编号表示轨 道编号。

输入输出变量	名称	数据类型	有效范围	内容
CamProperty ^{*6}	凸轮属性	_sMC_CAM_PROPERTY	—	指定凸轮属性结构体 _sMC_CAM_PROPERTY 型的变量。 请指定 _sMC_CAM_PROPERTY 型用户定义变量或 Sysmac studio 的凸轮编辑器中创建的凸轮属性变量。
CamNodes ^{*6}	凸轮节点	ARRAY[0..N] OF _sMC_CAM_NODE	—	指定凸轮节点结构体 _sMC_CAM_NODE 型的排列变量。 请指定 _sMC_CAM_NODE 型用户定义变量 ^{*7} 或 Sysmac studio 的凸轮编辑器中创建的凸轮节点变量。
AxisParameter ^{*6}	轴参数	_sAXIS_PARAM	—	重写时，指定轴参数。 读取时，指定 _sAXIS_PARAM 型变量，作为所读取轴参数的保存位置。

*1. 排列要素 [N] 通过 Sysmac Studio 自动设定。指定在 Sysmac Studio 的凸轮编辑器中创建的凸轮数据变量。

*2. 关于变量的数据类型，请参考 □□「参数的数据类型和有效范围 (P.5-14)」。

*3. 可在 CPU 单元版本 1.01 以上和 Sysmac Studio Ver.1.02 以上的组合中指定。

*4. 可在 CPU 单元版本 1.03 以上和 Sysmac Studio Ver.1.04 以上的组合中指定。

*5. 可在 CPU 单元版本 1.06 以上和 Sysmac Studio Ver.1.07 以上的组合中指定。

*6. 可在 CPU 单元版本 1.08 以上和 Sysmac Studio Ver.1.09 以上的组合中指定。

*7. 用户定义变量时，应创建为排列变量的要素开始编号为 0，排列要素 [N] 小于 358 以下。

2-2 指令一览

运动控制指令有以下三种。

种类	概要
共通指令	MC 功能模块共通的指令
轴指令	MC 功能模块执行单轴控制所需的指令
轴组指令	MC 功能模块执行多轴协调控制所需的指令

共通指令的详情请参考「第 5 章 共通指令」，轴指令的详情请参考「第 3 章 轴指令」，轴组的详情请参考「第 4 章 轴组指令」。

对 NX 系列位置接口单元使用运动控制指令时，可能包含功能有限制的指令和无法使用的指令。详情请参考 □□ 《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

共通指令

下面介绍 MC 功能模块共通的指令。

指令类别栏的“管理”表示非动作型指令，“动作”表示动作型指令。

P : PLCopen® 技术规格中定义的指令

O : MC 功能模块独有的指令。

指令	指令名称	概要	指令类别	
MC_SetCamTableProperty	凸轮表属性更新	更新输入参数中指定的凸轮表的终点索引	管理	O
MC_SaveCamTable	凸轮表保存	将输入参数中指定的凸轮表保存至非易失性存储器中	管理	O
MC_Write	MC 设定写入	重写运动控制参数设定的一部分	管理	O
MC_GenerateCamTable *1	凸轮表生成	根据输入输出参数中指定的凸轮属性和凸轮节点生成凸轮表	管理	O
MC_WriteAxisParameter *1	轴参数写入	重写运动控制参数设定中的轴参数设定	管理	O
MC_ReadAxisParameter *1	轴参数读取	读取运动控制参数设定中的轴参数设定	管理	O

*1. 可在 CPU 单元版本 1.08 以上和 Sysmac Studio Ver.1.09 以上的组合中使用。

关于启动指令后轴状态的变化，请参考 □□ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

轴指令

下面介绍 MC 功能模块执行单轴控制所需的指令。

指令类别栏的“管理”表示非动作型指令，“动作”表示动作型指令。

P：PLCopen® 技术规格中定义的指令

O：MC 功能模块独有的指令。

指令	指令名称	概要	指令类别	
MC_Power	可运行	将驱动器切换为可运行状态	管理	P
MC_MoveJog	点动进给	依据指定的目标速度执行点动进给	动作	O
MC_Home	原点复位	驱动电机，使用极限信号、近原点信号、原点信号确定机械原点	动作	P
MC_HomeWithParameter ^{*1}	参数指定原点复位	指定原点复位参数，驱动电机，使用极限信号、近原点信号、原点信号确定机械原点。	动作	O
MC_Move	定位	进行绝对值定位、相对值定位	动作	O
MC_MoveAbsolute	绝对值定位	指定绝对坐标的目标位置，进行定位	动作	P
MC_MoveRelative	相对值定位	指定指令当前位置起的移动距离，进行定位	动作	P
MC_MoveVelocity	速度控制	使用伺服驱动器的位置控制模式，进行模拟速度控制	动作	P
MC_MoveZeroPosition	高速原点复位	将绝对坐标“0”作为目标位置，进行定位并返回原点	动作	O
MC_MoveFeed	中断固定尺寸定位	指定基于外部输入的中断输入发生位置起的移动距离，进行定位	动作	O
MC_Stop	强制停止	将轴减速停止	动作	P
MC_ImmediateStop	立即停止	与轴状态无关，按照输入变量“StopMode（停止方法选择）”的停止方法停止	动作	O
MC_SetPosition	当前位置变更	将轴的指令当前位置和反馈当前位置变更为任意值	管理	P
MC_SetOverride	超驰值设定	变更轴的目标速度	管理	P
MC_ResetFollowingError	偏差计数器复位	重置指令位置和反馈位置的偏差	动作	O
MC_CamIn	凸轮动作开始	使用指定的凸轮表开始凸轮动作	动作	P
MC_CamOut	凸轮动作解除	完成输入参数中指定的轴的凸轮动作	动作	P
MC_GearIn	齿轮动作开始	设定主轴与从轴间的齿轮比，进行齿轮动作	动作	P
MC_GearInPos	位置指定齿轮动作	设定主轴与从轴间的齿轮比，进行电子齿轮动作。指定开始同步时主轴及从轴的位置	动作	P
MC_GearOut	齿轮动作解除	停止正在执行的 MC_GearIn、MC_GearInPos	动作	P
MC_MoveLink	梯形模式凸轮	与指定的主轴同步进行定位	动作	O
MC_CombineAxes	加减法定位	将 2 个轴的指令位置相加或相减得到的值，作为指令位置输出	动作	O
MC_Phasing	主轴相对值相位补偿	执行同步控制中的主轴相位补偿	动作	P
MC_TorqueControl	扭矩控制	使用伺服驱动器的扭矩控制模式进行扭矩控制	动作	P
MC_SetTorqueLimit	扭矩限制	使用伺服驱动器的扭矩限制功能，限制伺服驱动器的输出扭矩	管理	O
MC_ZoneSwitch	区域监视	判断轴的指令位置或反馈当前位置是否存在于指定范围（区域）内	管理	O
MC_TouchProbe	外部锁定有效	发生触发信号，记录轴位置	管理	P
MC_AbortTrigger	外部锁定无效	将执行中的锁定变为无效	管理	P
MC_AxesObserve	轴间偏差监视	监视指定轴的指令位置或反馈位置的差异量是否超出容许值	管理	O
MC_SyncMoveVelocity	周期性同步速度控制	将目标速度中设定的值，按任务周期 ^{*2} ，以周期性同步速度模式（CSV）输出到伺服驱动器	动作	O

指令	指令名称	概要	指令类别	
MC_SyncMoveAbsolute ^{*1}	周期性同步绝对定位	周期性输出轴中指定的目标位置	动作	O
MC_Reset	轴错误复位	解除轴异常	管理	P
MC_ChangeAxisUse ^{*3}	轴使用变更	临时切换轴参数中的 [轴使用]	管理	P
MC_DigitalCamSwitch ^{*4}	数字凸轮开关有效	根据轴的位置将数字输出设为 ON 或 OFF。	管理	P
MC_TimeStampToPos ^{*4}	时间戳→轴位置计算	计算指定时间戳的轴位置。	管理	O
MC_PeriodicSync Variables ^{*5}	轴变量任务间固定周期同步	按固定周期执行轴变量的任务间同步。	管理	O
MC_SyncOffsetPosition ^{*6}	周期性同步位置偏置补偿	执行同步控制中的从轴位置补偿。	动作	O

*1. 可在 CPU 单元版本 1.03 以上和 Sysmac Studio Ver.1.04 以上的组合中使用。

*2. 任务为主固定周期任务时，任务周期为主周期，为固定周期任务（执行优先度 5）时，任务周期为固定周期任务（执行优先度 5）的任务周期。

*3. 可在 CPU 单元版本 1.04 以上和 Sysmac Studio Ver.1.05 以上的组合中使用。

*4. 可在 CPU 单元版本 1.06 以上和 Sysmac Studio Ver.1.07 以上的组合中使用。

*5. 不可在 NJ 系列 CPU 单元中使用。

*6. 可在 CPU 单元版本 1.10 以上和 Sysmac Studio Ver.1.12 以上的组合中使用。

关于启动指令后轴状态的变化，请参考 □《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

关于针对 PLCopen[®] 技术规格的依据内容，请参考合规清单。

合规清单可从 PLCopen[®] 的网站获取。

轴组指令

下面介绍 MC 功能模块执行多轴协调控制所需的指令。

指令类别栏的“组管理”表示非动作型指令，“组动作”表示动作型指令。

P : PLCopen[®] 技术规格中定义的指令

O : MC 功能模块独有的指令。

指令	指令名称	概要	指令类别	
MC_GroupEnable	轴组有效	将轴组设为有效	组管理	P
MC_GroupDisable	轴组无效	将轴组设为无效	组管理	P
MC_MoveLinear	直线插补	进行直线插补	组动作	O
MC_MoveLinearAbsolute	绝对值直线插补	指定绝对位置进行直线插补	组动作	P
MC_MoveLinearRelative	相对值直线插补	指定相对位置进行直线插补	组动作	P
MC_MoveCircular2D	2 轴圆弧插补	进行 2 轴圆弧插补	组动作	O
MC_GroupStop	轴组强制停止	使插补动作中的所有轴减速停止	组动作	P
MC_GroupImmediateStop	轴组立即停止	按轴参数中选择的方法，使插补动作中的所有轴立即停止	组动作	O
MC_GroupSetOverride	轴组超驰值设定	变更插补动作中的合成目标速度	组管理	P
MC_GroupReadPosition ^{*1}	轴组位置获取	获取轴组指令的当前位置和反馈当前位置	组管理	O
MC_ChangeAxesInGroup	轴组构成轴写入	暂时改写轴组参数中的 [构成轴]	组管理	O
MC_GroupSyncMove Absolute ^{*1}	轴组周期性同步绝对位置控制	周期性输出各轴中指定的目标位置	组动作	O
MC_GroupReset	轴组错误复位	解除轴组及轴异常	组管理	P

*1. 可在 CPU 单元版本 1.01 以上和 Sysmac Studio Ver.1.02 以上的组合中使用。

关于启动指令后轴组状态的变化，请参考 □《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

关于针对 PLCopen[®] 技术规格的依据内容，请参考合规清单。

合规清单可从 PLCopen[®] 的网站获取。

2-3 PDO 映射

要使用运动控制的各功能，需要将使用的功能所需的对象映射到过程数据通信中。
事先将需要的对象汇总得到的数据即为 PDO 映射。

使用欧姆龙产伺服驱动器 G5 系列 R88D-KN□□□-ECT (Ver.2.1 以上)、R88D-KN□□□-ECT-L (Ver.1.1 以上) 时，PDO 映射可使用 Sysmac Studio 的默认设定。

RxPDO (1704h)	控制字 (6040Hex)、目标位置 (607AHex)、目标速度 (60FFHex)、目标扭矩 (6071Hex)、操作模式 (6060Hex)、锁定功能 (60B8Hex)、最大曲线速度 (607FHex)、正转扭矩限制值 (60E0Hex)、反转扭矩限制值 (60E1Hex)
TxPDO (1B02h)	错误代码 (603FHex)、状态字 (6041Hex)、反馈位置 (6064Hex)、反馈扭矩 (6077Hex)、操作模式显示 (6061Hex)、锁定状态 (60B9Hex)、锁定位置 1(60BAHex)、锁定位置 2(60BCHex)、数字输入 (60FDHex)



参考

用欧姆龙生产的伺服驱动器 G5 系列 R88D-KN □□□ -ECT 进行全闭环控制时，RxPDO 请选择 1701Hex 或对象总大小为 12 字节以下的 1600Hex (Ver.2.1 以上时)。

PDO 映射的设定方法请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

使用 NX 系列位置接口单元时请参考 □□《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》的“I/O 入口映射”。

必须对象

必须对象包括轴种类为伺服轴时的对象和编码器轴时的对象。
未设定任何必须对象时，将发生必须过程数据对象未设定 (错误代码：3460Hex)。

伺服轴

将 MC 功能模块的指令用于伺服轴时，需要设定以下对象。

输入输出	功能名称	过程数据
输出	控制字	6040Hex
	目标位置	607AHex
输入	状态字	6041Hex
	反馈位置	6064Hex

版本相关信息

- 在单元版本 1.09 以下的 CPU 单元中，使用欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型以外的产品时，需要操作模式 (6060Hex) 和操作模式显示 (6061Hex)。
- 使用单元版本 1.10 以上的 CPU 单元时，因操作模式 (6060Hex) 及操作模式显示 (6061Hex) 的映射有无，动作如下。

	有操作模式显示 (6061Hex)	无操作模式显示 (6061Hex)
有操作模式 (6060Hex)	<ul style="list-style-type: none"> 可执行使用 CSP^{*1}、CSV^{*2}、CST^{*3} 的指令。 CSP、CSV、CST 以外时，为伺服 OFF。 	<ul style="list-style-type: none"> 可执行使用 CSP 的指令。使用其他控制模式的指令将在执行时发生“过程数据对象设定不足（错误代码：3461Hex）”。 MC 功能模块将伺服驱动器的控制模式视为 CSP。向伺服驱动器发出 CSP 指令。
无操作模式 (6060Hex)	<ul style="list-style-type: none"> 可执行使用 CSP 的指令。使用其他控制模式的指令将在执行时发生“过程数据对象设定不足（错误代码：3461Hex）”。 CSP 以外时，为伺服 OFF。 	<ul style="list-style-type: none"> 可执行使用 CSP 的指令。使用其他控制模式的指令将在执行时发生“过程数据对象设定不足（错误代码：3461Hex）”。

*1. 伺服驱动器控制模式中的周期同步位置模式 (CSP)。

*2. 伺服驱动器控制模式中的周期性同步速度模式 (CSV)。

*3. 伺服驱动器控制模式中的周期同步扭矩模式 (CST)。

编码器轴

将 MC 功能模块的指令用于编码器轴时，需要设定以下对象。

输入输出	功能名称	过程数据
输入	反馈位置	4010Hex

不同指令的设定对象

有些指令有需要设定的对象。

伺服轴、编码器轴需要分别设定。

此外，未设定各指令的必要对象时，将发生过程数据对象设定不足（错误代码：3461Hex）。

伺服轴

要作为伺服轴使用时，需要根据指令设定必要对象。

请在参考以下一览的基础上，设定必要对象。

此外，以下一览表中未记载的指令，只需设定必须对象即可进行伺服轴的动作。

● 输出设定

指令名称	功能名称						
	目标速度 60FFHex	目标扭矩 6071Hex	操作模式 6060Hex ^{*1}	锁定功能 60B8Hex	最大曲线速度 607FHex	正转侧扭矩 限制值 60E0Hex	反转侧扭矩 限制值 60E1Hex
MC_Home, MC_HomeWith Parameter				△ ^{*2}			
MC_MoveFeed				△ ^{*3}			
MC_MoveLink				△ ^{*4}			
MC_TorqueControl		○	○		△ ^{*5}		
MC_SetTorqueLimit						○	○
MC_TouchProbe				△ ^{*2}			
MC_SyncMoveVelocity	○		○				

*1. 设定操作模式 (6060Hex) 时，请同时设定操作模式显示 (6061Hex)。若只设定其中的一个，将无法正常工作。

*2. 除原点复位动作模式“11”、“12”、“14”外，需要设定。

*3. Mode 选择为 [驱动器模式] 时。

*4. LinkOption（同步开始条件选择）选择为 [_mcTriggerDetection]，Mode 选择为 [驱动器模式] 时。

*5. 仅在使用欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型时检查。

● 输入设定

指令名称	功能名称				
	反馈扭矩 6077Hex	操作模式显示 6061Hex ^{*1}	锁定状态 60B9Hex	锁定位置 1 60BAHex	锁定位置 2 60BCHex
MC_Home, MC_HomeWith Parameter			△ ^{*2}	△ ^{*2}	
MC_MoveFeed			△ ^{*3}	△ ^{*4}	△ ^{*5}
MC_MoveLink			△ ^{*6}	△ ^{*7}	△ ^{*8}
MC_TorqueControl	○	○			
MC_TouchProbe			△ ^{*3}	△ ^{*4}	△ ^{*5}
MC_SyncMoveVelocity		○			

*1. 设定操作模式显示 (6061Hex) 时, 请同时设定操作模式 (6060Hex)。若只设定其中的一个, 将无法正常工作。

*2. 除原点复位动作模式“11”、“12”、“14”外, 需要设定。

*3. Mode 选择为 [驱动器模式] 时。

*4. Mode 选择为 [驱动器模式], LatchID (锁定 ID 选择) 选择为 [_mcLatch1 (锁定功能 1)] 时

*5. Mode 选择为 [驱动器模式], LatchID (锁定 ID 选择) 选择为 [_mcLatch2 (锁定功能 2)] 时

*6. LinkOption (同步开始条件选择) 选择为 [_mcTriggerDetection], Mode 选择为 [驱动器模式] 时。

*7. LinkOption (同步开始条件选择) 选择为 [_mcTriggerDetection], Mode 选择为 [驱动器模式], LatchID (锁定 ID 选择) 选择为 [_mcLatch1 (锁定功能 1)] 时。

*8. LinkOption (同步开始条件选择) 选择为 [_mcTriggerDetection], Mode 选择为 [驱动器模式], LatchID (锁定 ID 选择) 选择为 [_mcLatch2 (锁定功能 2)] 时。

编码器轴

要作为编码器轴使用时, 需要根据指令设定必要对象。

请在参考以下一览表的基础上, 设定必要对象。

此外, 以下一览表中未记载的指令, 只需设定必须对象即可进行编码器轴的动作。

● 输出设定

指令名称	功能名称	
	锁定功能 4020Hex	编码器输入从站软交换 4020Hex
MC_TouchProbe	△ ^{*1}	△ ^{*2}

*1. Mode 选择为 [驱动器模式] 时。

*2. 使用欧姆龙生产的编码器从站, Mode 选择为 [驱动器模式] 时。

● 输入设定

指令名称	功能名称			
	锁定状态 4030Hex	锁定位置 1 4012Hex	锁定位置 2 4013Hex	编码器输入从站状态 4030Hex
MC_TouchProbe	△ *1	△ *2	△ *3	△ *4

*1. Mode 选择为 [驱动器模式] 时。

*2. Mode 选择为 [驱动器模式]，LatchID（锁定 ID 选择）选择为 [_mcLatch1（锁定功能 1）] 时。

*3. Mode 选择为 [驱动器模式]，LatchID（锁定 ID 选择）选择为 [_mcLatch2（锁定功能 2）] 时。

*4. 使用欧姆龙生产的编码器从站，Mode 选择为 [驱动器模式] 时。

3

轴指令

下面介绍 MC 功能模块执行单轴控制所需的指令。

MC_Power	3-3
MC_MoveJog	3-8
MC_Home	3-15
MC_HomeWithParameter	3-36
MC_Move	3-41
MC_MoveAbsolute	3-47
MC_MoveRelative	3-73
MC_MoveVelocity	3-80
MC_MoveZeroPosition	3-95
MC_MoveFeed	3-102
MC_Stop	3-129
MC_ImmediateStop	3-138
MC_SetPosition	3-143
MC_SetOverride	3-149
MC_ResetFollowingError	3-154
MC_CamIn	3-161
MC_CamOut	3-215
MC_GearIn	3-220
MC_GearInPos	3-240
MC_GearOut	3-261
MC_MoveLink	3-266
MC_CombineAxes	3-288
MC_Phasing	3-299
MC_TorqueControl	3-305
MC_SetTorqueLimit	3-316
MC_ZoneSwitch	3-323
MC_TouchProbe	3-329

MC_AbortTrigger	3-350
MC_AxesObserve	3-354
MC_SyncMoveVelocity	3-360
MC_SyncMoveAbsolute	3-369
MC_Reset	3-375
MC_ChangeAxisUse	3-379
MC_DigitalCamSwitch	3-383
MC_TimeStampToPos	3-402
MC_PeriodicSyncVariables	3-415
MC_SyncOffsetPosition	3-423

MC_Power

将伺服驱动器切换为可运行状态。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_Power	可运行	FB		<pre>MC_Power_instance (Axis := 《参数》, Enable := 《参数》, Status => 《参数》, Busy => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、 FALSE	FALSE	若设为 TRUE，则变为可运行状态，若设为 FALSE，则解除可运行状态。

输出变量

输出变量	名称	数据类型	有效范围	内容
Status	可运行	BOOL	TRUE、 FALSE	变为可运行状态时变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*1	发生异常时，输出错误代码。16#0000 表示正常。

*1. □「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Status	指定轴变为可运行状态时	<ul style="list-style-type: none"> 指定的可运行状态解除时 Error 变为 TRUE 时
Busy	Enable 的上升沿	<ul style="list-style-type: none"> Enable 变为 FALSE 时 Error 变为 TRUE 时
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

功能说明

- 将 Enable（有效）设为 TRUE 后，Axis（轴）中指定的轴将变为可运行状态。
将轴设为可运行状态后，可进行轴控制。
- 将 Enable（有效）设为 FALSE 后，将解除 Axis（轴）中指定轴的可运行状态。
解除了可运行状态时，轴将无法接收动作指令，不可进行轴控制。此外，对解除了可运行状态的轴发出的动作指令将发生异常。但是，在解除状态下，MC_Power（可运行）指令、MC_Reset（轴错误重置）指令仍可执行。
- 还可将动作中的轴切换为解除状态。此时，动作中的轴将变为 CommandAborted（执行中断），停止输出动作指令，切换为解除状态。
- 如果使用带绝对编码器的伺服电机，轴切换为可运行状态时，若处于原点未确定状态，将进行 [绝对编码器原点位置偏置] 补偿，变为原点确定状态。
在单元版本 1.10 以上的 CPU 单元中，EtherCAT 的过程数据通信从未建立状态变为建立状态时，也可变为原点确定状态。
[绝对编码器原点位置偏置] 请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。



使用注意事项

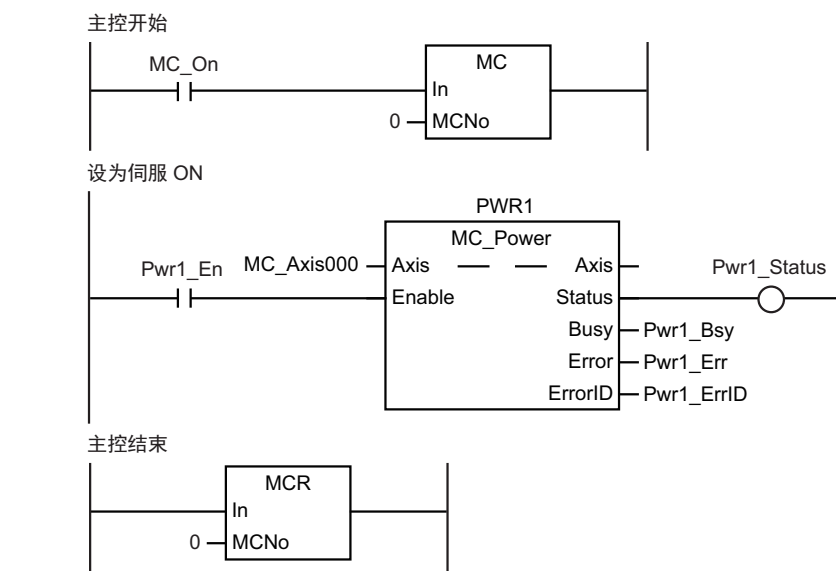
- 本指令可用于轴种类为伺服轴或虚拟伺服轴的轴。用于编码器轴或虚拟编码器轴时，将发生异常。
- 对 NX 系列脉冲输出单元执行本指令时，本指令的功能不同。详情请参考 □ □ 《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。
- **对同步控制的主轴执行本指令时**
在垂直轴中，主轴从可运行状态解除时，主轴的位置可能发生剧烈变化。因此，从轴的动作也可能剧烈变化。
为避免从轴发生剧烈动作，请为主轴设置制动器或在结束同步控制后再解除主轴的运行状态。



参考

Execute 型运动控制指令在 Execute 输入所连接的接点上升时开始动作，并持续动作至指令的控制完成。即使正在通过主控执行复位，若在执行复位前指令已经在执行，将继续动作，直至指令的控制完成。

若要将 Execute 型运动控制指令联锁，请如下图所示，将 MC_Power（可运行）指令配置到主控区域。将 MC_On 设为 FALSE，可确保变为伺服 OFF。



● 与 CPU 单元动作模式的关系

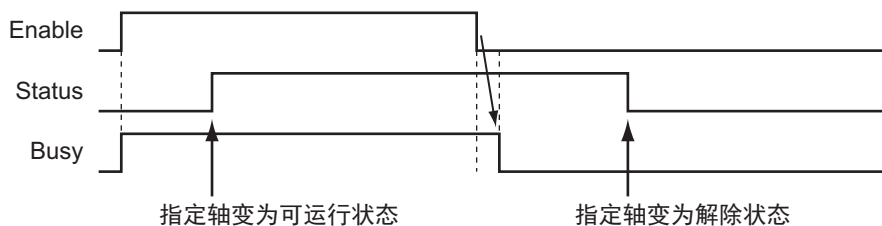
在运行模式下通过本指令变为可运行状态后，再切换为程序模式时，将保持运行状态。

● 通过在线编辑删除指令时

通过本指令变为可运行状态后，即使在在线编辑中删除了本指令，仍将保持可运行状态。

时序图

- 若将 Enable（有效）设为 TRUE，表示指令接收中的 Busy（执行中）将变为 TRUE。
- 之后，若将轴变为可运行状态，Status（可运行）变为 TRUE。
- 若将 Enable（有效）设为 FALSE，Busy（执行中）将变为 FALSE。Status（可运行）在解除可运行状态后变为 FALSE。Status（可运行）与 Enable（有效）的 TRUE/ FALSE 无关，始终输出轴的可运行状态。





使用注意事项

- 将 Enable（有效）设为 TRUE 后，如果轴的处理尚未结束，则 Status（可运行）不会变为 TRUE。运行轴时，请务必确认 Status（可运行）已变为 TRUE 后，再开始动作。
- 创建用户程序时，请确保在接通控制器的电源后开始运行等时，先确认已建立 EtherCAT 通信，再启动运动控制指令。此外，请在用户程序中嵌入联锁功能，以在运行过程中监视 EtherCAT 通信是否发生异常。

运动指令重新执行

输入为“Enable 型”的指令没有运动指令重新执行。

运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

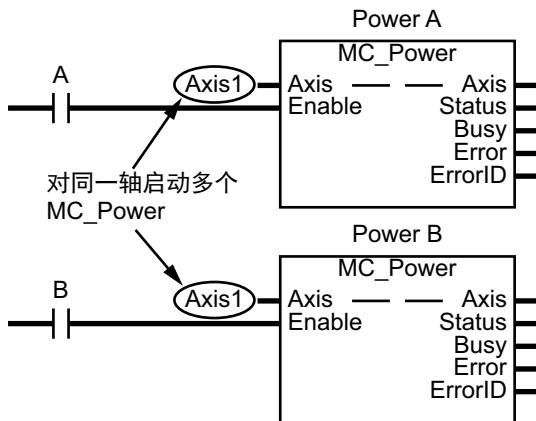
● 本指令的多重启动

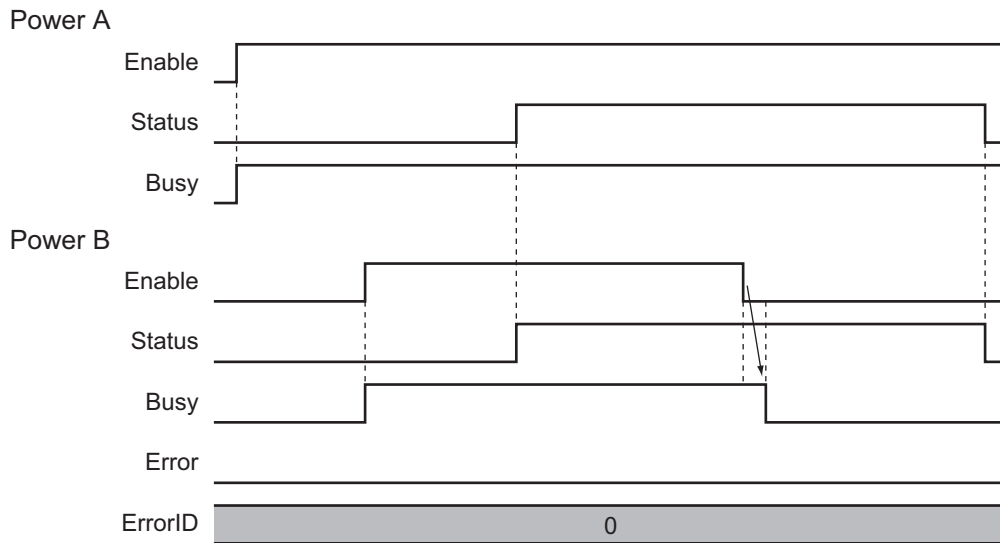


使用注意事项

创建程序时，不可对正在执行 MC_Power（可运行）指令的轴，启动其他实例的 MC_Power（可运行）指令。原则上，1 个轴只能对应 1 个 MC_Power（可运行）指令。

如果对正在执行 MC_Power（可运行）指令的轴，启动了其他实例的 MC_Power（可运行）指令，将优先执行后面的 MC_Power（可运行）指令。





错误代码

指令中发生的异常请参考 □ 「A-1 错误代码一览 (P.A-2)」。

MC_MoveJog

依据指定的目标速度执行点动进给。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_MoveJog	点动进给	FB		<pre>MC_MoveJog_instance (Axis := 《参数》, PositiveEnable := 《参数》, NegativeEnable := 《参数》, Velocity := 《参数》, Acceleration := 《参数》, Deceleration := 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
PositiveEnable	正方向有效	BOOL	TRUE、FALSE	FALSE	若设为 TRUE，则开始向正方向移动。若设为 FALSE，则结束移动。
NegativeEnable	负方向有效	BOOL	TRUE、FALSE	FALSE	若设为 TRUE，则开始向负方向移动。若设为 FALSE，则结束移动。
Velocity	目标速度	LREAL	正数或“0”	0	指定目标速度。 单位为 [指令单位 /s]。*1
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度。 单位为 [指令单位 /s ²]。*1
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。 单位为 [指令单位 /s ²]。*1

*1. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》的“单位转换设定”。

输出变量

输出变量	名称	数据类型	有效范围	内容
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*1	发生异常时，输出错误代码。16#0000 表示正常。

*1. □ □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Busy	PositiveEnable 或 NegativeEnable 的上升沿	<ul style="list-style-type: none"> 停止了轴时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动（中断），本指令中止时 因发生异常，本指令中止时 正在发生异常时，启动了本指令 正在执行 MC_Stop 指令时，启动了本指令 	<ul style="list-style-type: none"> PositiveEnable 为 TRUE 时，与 PositiveEnable 的 FALSE 同时 NegativeEnable 为 TRUE 时，与 NegativeEnable 的 FALSE 同时 PositiveEnable 及 NegativeEnable 两者为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

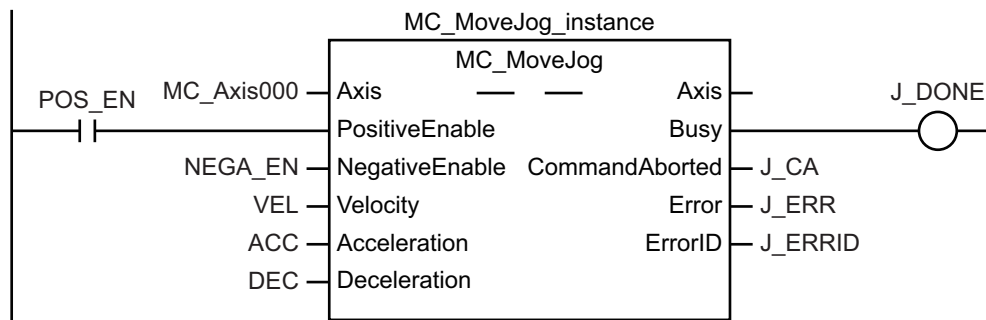
功能说明

- 依据指定的 Velocity（目标速度）执行点动进给。
- 要向正方向进行点动进给时，将 PositiveEnable（正方向有效）设为 TRUE，要向负方向进行点动进给时，将 NegativeEnable（负方向有效）设为 TRUE。
- 将 PositiveEnable（正方向有效）和 NegativeEnable（负方向有效）同时设为 TRUE 时，将优先 PositiveEnable（正方向有效），向正方向进行点动进给。
- 指定的 MC_MoveJog（点动进给）指令的指令速度超出轴参数 [点动最高速度] 时，为 [点动最高速度]。

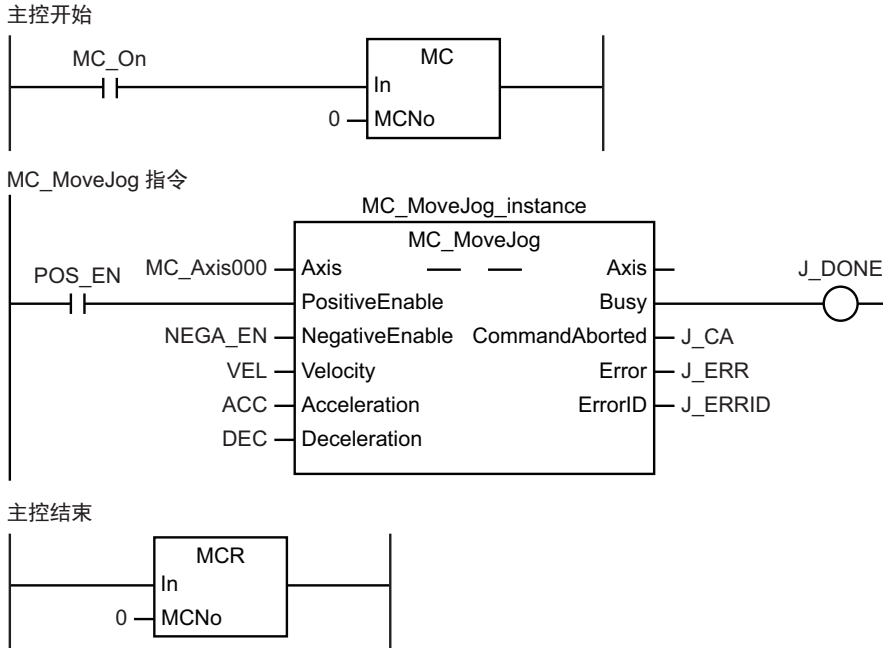


使用注意事项

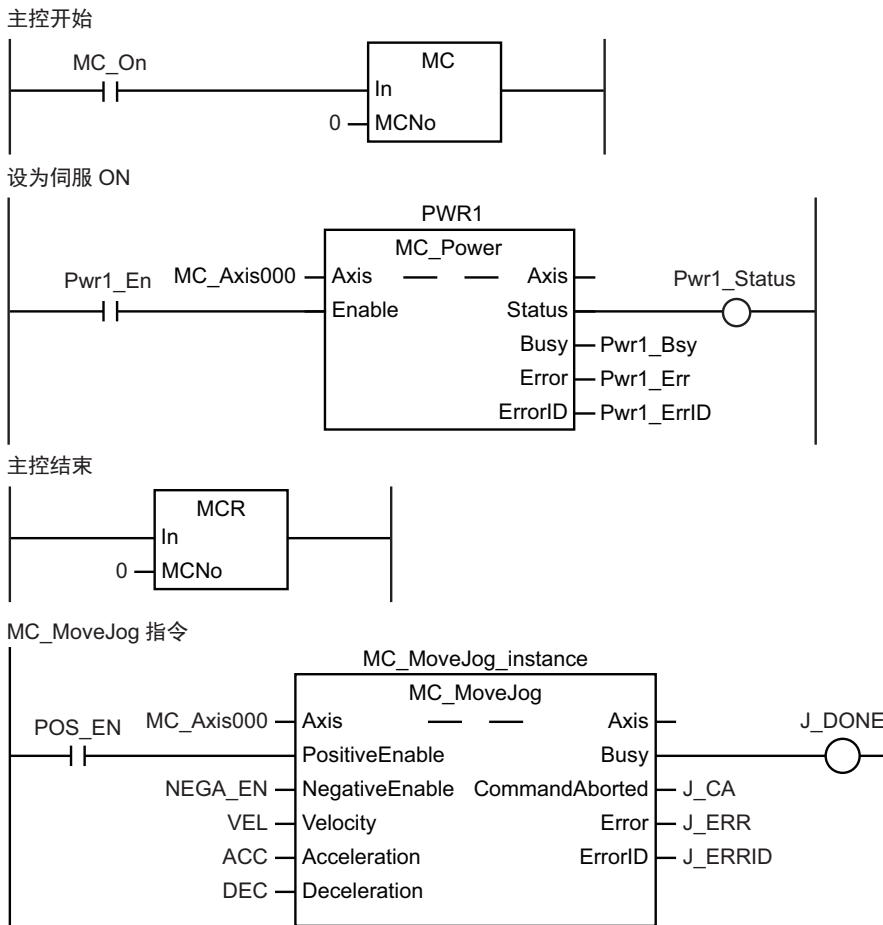
- 创建梯形图程序时，必须如下所示，将输入变量“PositiveEnable（正方向有效）”与左母线连接，将“NegativeEnable（负方向有效）”指定为变量。



如果对 MC_MoveJog（点动进给）使用主控指令（MC 指令），请勿进行如下记述。若如下记述，主控将只对 PositiveEnable（正方向有效）有用，对 NegativeEnable（负方向有效）无用。

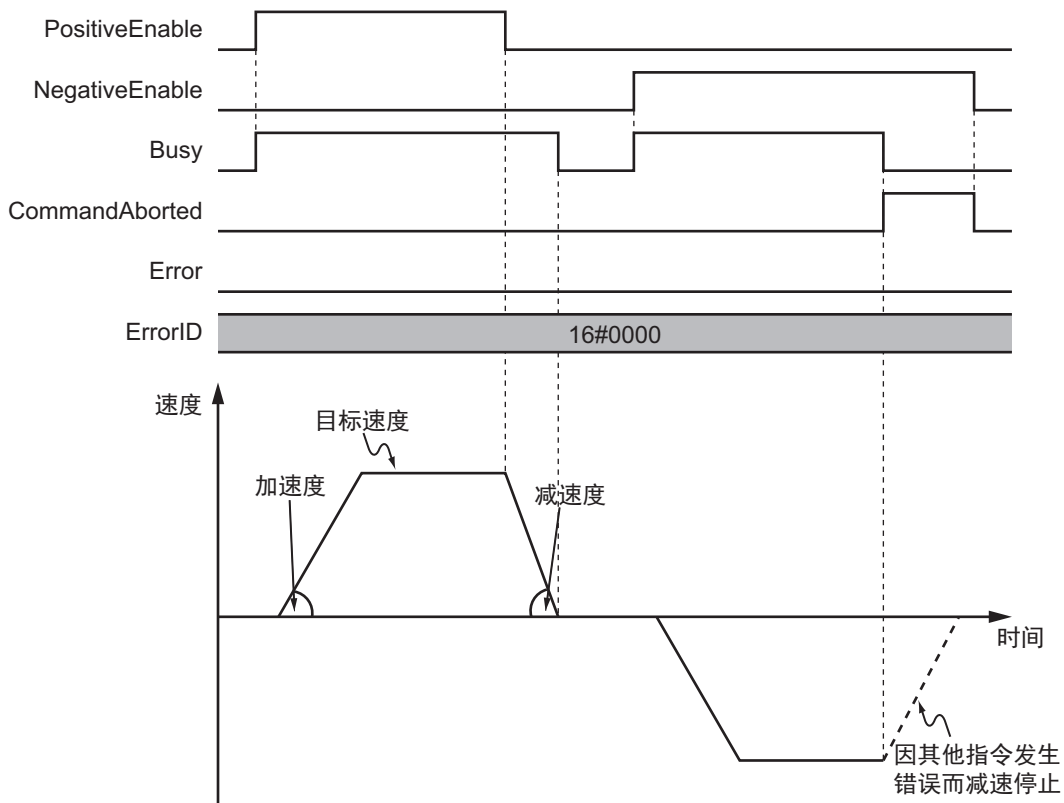


请务必对 MC_Power（可运行）使用主控指令。



时序图

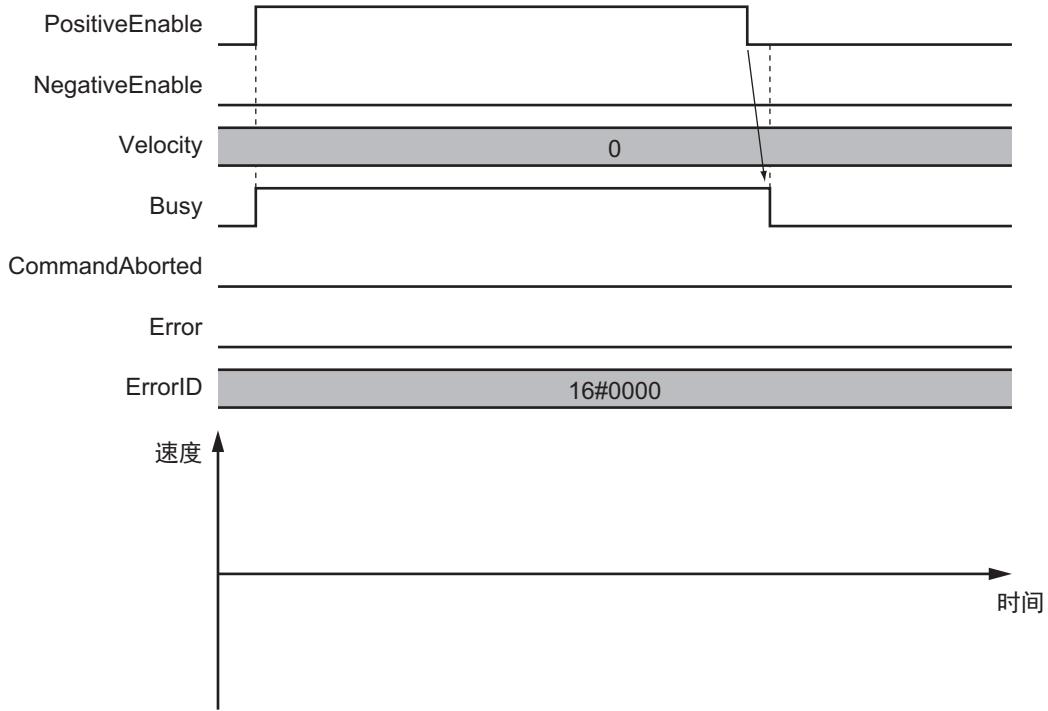
- Busy（执行中）与 PositiveEnable（正方向有效）或 NegativeEnable（负方向有效）的上升沿同时变为 TRUE。
- 在 PositiveEnable（正方向有效）或 NegativeEnable（负方向有效）的下降沿开始减速，在轴停止的同时，Busy（执行中）变为 FALSE。
- 通过其他指令中止了本指令时，CommandAborted（执行中断）变为 TRUE，Busy（执行中）变为 FALSE。



输入变量中可指定 Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）。输入变量 Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）仅在 PositiveEnable（正方向有效）或 NegativeEnable（负方向有效）上升时反映到动作中。因此，PositiveEnable（正方向有效）或 NegativeEnable（负方向有效）为 TRUE 期间，即使变更 Velocity（目标速度），速度也不会变更。

● 目标速度为“0”时的时序图

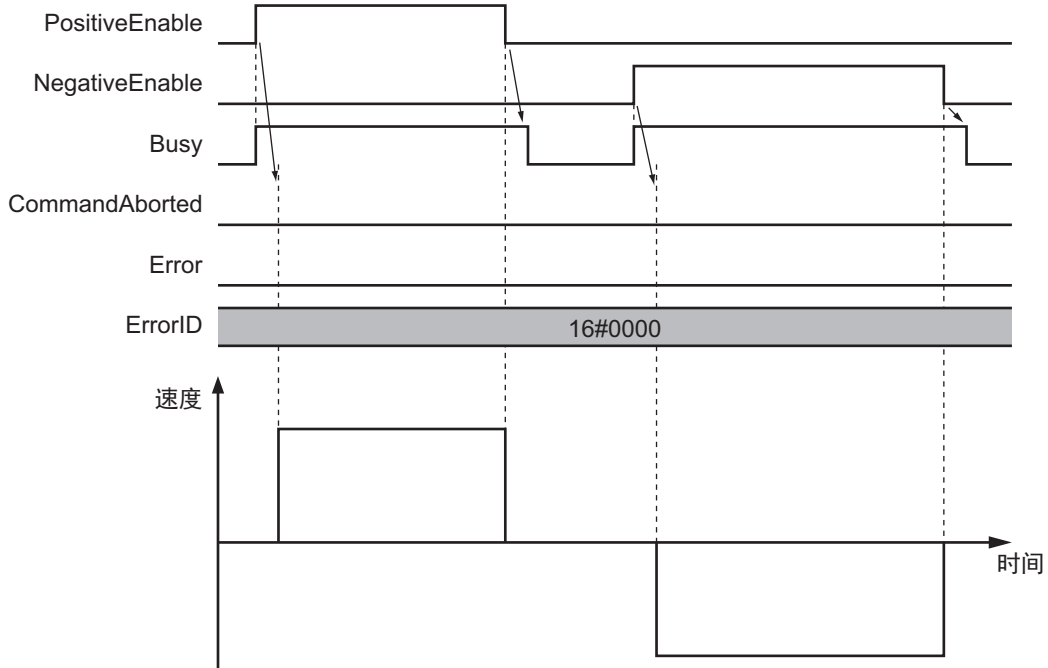
将 Velocity（目标速度）指定为“0”并启动时，轴不会移动，变为连续动作中。将 Velocity（目标速度）设为“0”并启动时的动作示例如下。



● 加减速为“0”时的时序图

将 Acceleration（加速度）或 Deceleration（减速度）指定为“0”并启动时，将不进行加减速而达到目标速度。

Acceleration（加速度）及 Deceleration（减速度）两者为“0”时的动作示例如下。



运动指令重新执行

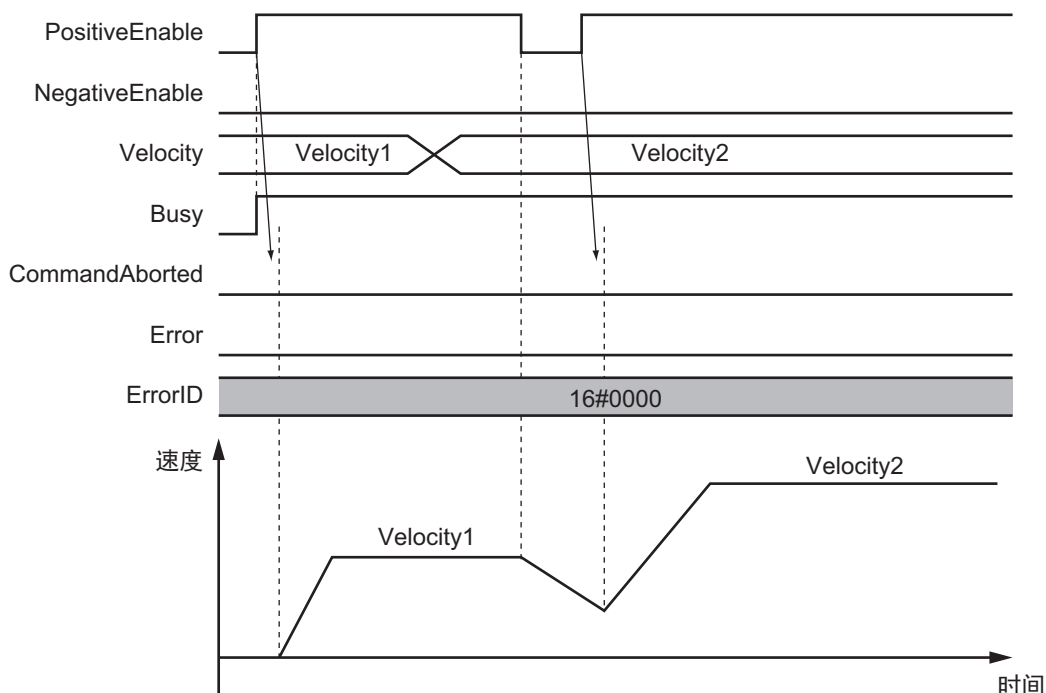
● 重新启动同一方向的 Enable 时

将 PositiveEnable（正方向有效）或 NegativeEnable（负方向有效）设为 FALSE 并正在减速时，再将 PositiveEnable（正方向有效）或 NegativeEnable（负方向有效）设为 TRUE 时，将在减速途中进行加减速，向目标速度移动。

此时，若 Velocity（目标速度）、Acceleration（加速度）或 Deceleration（减速度）有变更，将按新的输入变量动作。

此外，由于轴尚未停止，Busy（执行中）不会变为 FALSE。

在减速过程中将 PositiveEnable（正方向有效）设为 TRUE 时的动作示例如下。



● 重新启动不同方向的 Enable 时

将 PositiveEnable（正方向有效）设为 TRUE，并正在向正方向进行点动进给时，若将 NegativeEnable（负方向有效）设为 TRUE，将反转方向后向负方向进行点动进给。

此时，根据 NegativeEnable（负方向有效）为 TRUE 时的输入变量进行点动进给。输入变量中包括 Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）。

反转时的减速度及反转后的加速度与轴参数的 [反转时动作] 无关，始终以将 NegativeEnable（负方向有效）设为 TRUE 时的输入变量为准。

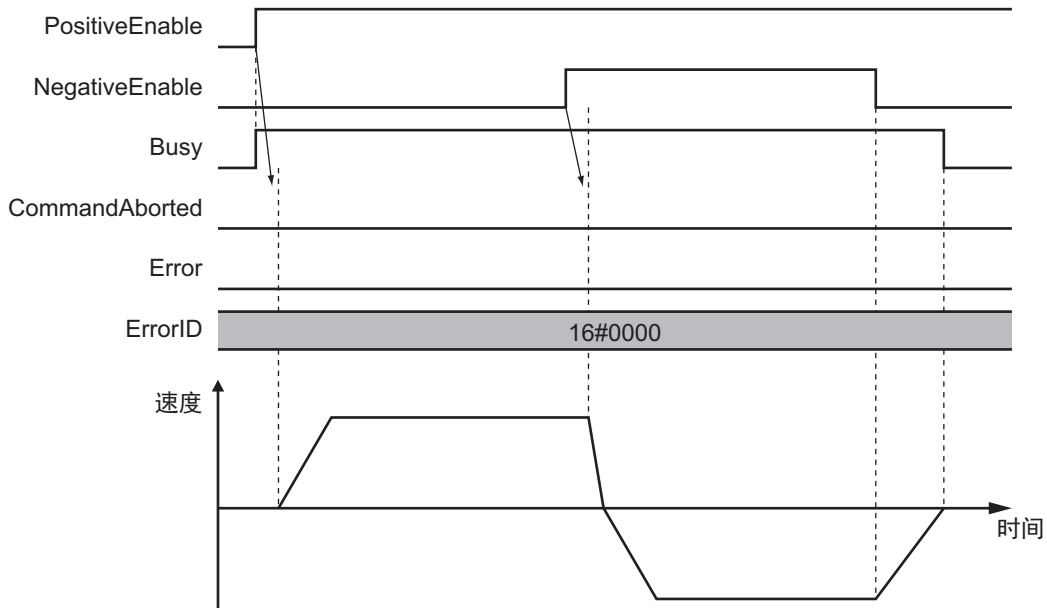
将 NegativeEnable（负方向有效）设为 TRUE，且正在向负方向进行点动进给时，若将 PositiveEnable（正方向有效）设为 TRUE，动作相同。

将 PositiveEnable（正方向有效）设为 TRUE，然后将 NegativeEnable（负方向有效）设为 TRUE 并正在向负方向进行点动进给时，即使在 PositiveEnable（正方向有效）保持 TRUE 的状态下，将 NegativeEnable（负方向有效）设为 FALSE，也不会向正方向进行点动进给。

若要向正方向进行点动进给，请先将 PositiveEnable（正方向有效）设为 FALSE，再设为 TRUE。

相反的情况下动作亦同。

将 PositiveEnable（正方向有效）设为 TRUE 后再将 NegativeEnable（负方向有效）设为 TRUE 时的动作示例如下。



运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。
参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 □ 「A-1 错误代码一览 (P.A-2)」。

MC_Home

驱动电机，使用极限信号、近原点信号、原点信号确定机械原点。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_Home	原点复位	FB		<pre>MC_Home_instance (Axis := 《参数》, Execute := 《参数》, Done => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、 FALSE	FALSE	上升时开始指令。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、 FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、 FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*1	发生异常时，输出错误代码。16#0000 表示正常。

*1. □「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	指令完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动 (中断), 本指令中止时 因发生异常, 本指令中止时 正在发生异常时, 启动了本指令 正在执行 MC_Stop 指令时, 启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称 (默认 "MC_Axis****") 或系统定义变量的轴变量名称 (_MC_AX[*], _MC1_AX[*], _MC2_AX[*])。

功能说明

- 对 Axis（轴）中指定的轴，在 Execute（启动）的上升沿开始原点复位动作。
- 原点复位指令中使用的各种参数在轴参数中设定。
- 原点复位指令中包括 10 种原点复位动作模式。
请通过 Sysmac Studio 的轴参数 [原点复位动作] 设定。



使用注意事项

关于主轴的注意事项，请参考 □□「同步控制的主轴及辅助轴的注意事项 (P.1-6)」。

● 数据对象的映射

使用 MC_Home（原点复位）指令时，请在 Sysmac Studio 的轴基本设定的 [详细设定] 中，映射以下对象数据。

但是，使用原点复位动作模式“11”、“12”、“14”时，无需设定。

- 锁定功能（60B8Hex）
- 锁定状态（60B9Hex）
- 锁定位置 1（60BAHex）

未设定以上内容时，将发生过程数据对象设定不足（错误代码：3461Hex）。

数据对象的映射请参考 □□「2-3 PDO 映射 (P.2-30)」，□□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

使用 NX 系列位置接口单元时请参考 □□《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》的“I/O 入口映射”。

欧姆龙生产的伺服驱动器 G5 系列的设定

原点复位指令中使用的近原点信号等输入信号在欧姆龙生产的伺服驱动器 G5 系列中设定。

输入型号的设定方法请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“伺服驱动器连接”及 □□《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型 用户手册 (SBCE-CN5-365)》、□□《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型 用户手册 (SBCE-CN5-366)》的“时序输入输出信号”。

变量名称	名称	数据类型	有效范围	单位	初始值	内容
ProximitySignalReference	接近信号选择	Input_REF	—	—	—	选择近原点输入信号的输入目标。
HomeSignalReference	原点信号选择	Input_REF	—	—	—	选择原点输入信号的输入目标。

NX 系列位置接口单元的设定

使用 NX 系列位置接口单元时的设定请参考 □□《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

原点复位动作模式

作为决定机械原点的动作，可从 10 种动作中选择。

- 附近退避、近原点输入 OFF 指定
- 附近退避、近原点输入 ON 指定
- 近原点输入 OFF 指定
- 近原点输入 ON 指定
- 极限输入 OFF 指定
- 附近退避、原点输入掩码距离指定
- 仅极限输入
- 附近退避、推压时间指定
- 无近原点输入、推压原点输入指定
- 原点预设

各原点复位动作中使用的原点复位参数如下表所示。

(○：使用参数、x：不使用参数)

原点复位动作	原点复位参数														
	原点输入信号	原点复位开始方向	原点检测方向	正方向极限输入时动作	负方向极限输入时动作	原点复位速度	原点复位接近速度	原点复位加速度	原点复位减速度	原点复位跃度	原点输入掩码距离	原点位置偏置	原点复位推压时间	原点复位修正值	原点复位修正速度
附近退避、近原点输入 OFF 指定	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
附近退避、近原点输入 ON 指定	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
近原点输入 OFF 指定	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
近原点输入 ON 指定	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
极限输入 OFF 指定	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
附近退避、原点输入掩码距离指定	○	○	○	○	○	○	○	○	○	○	○	○	×	○	○
仅极限输入	×	○	○	○	○	○	○	○	○	○	×	○	×	○	○
附近退避、推压时间指定	×	○	○	○	○	○	○	○	○	○	×	○	○	○	○
无近原点输入、推压原点输入指定	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
原点预设	×	×	×	×	×	×	×	×	×	×	×	○	×	×	×

各原点复位模式的动作详情请参考 □ 「原点确定的动作 (P.3-22)」。

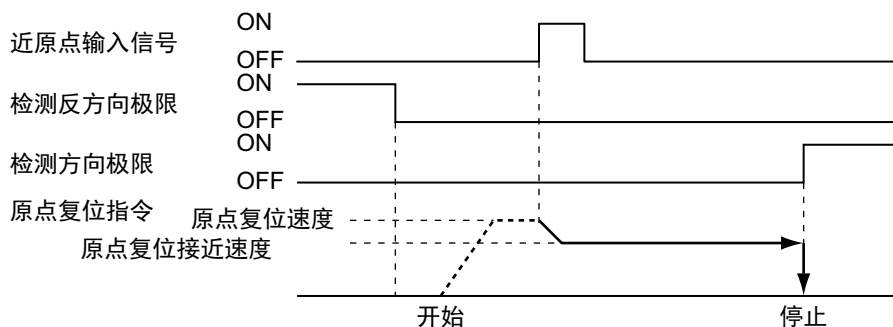


参考

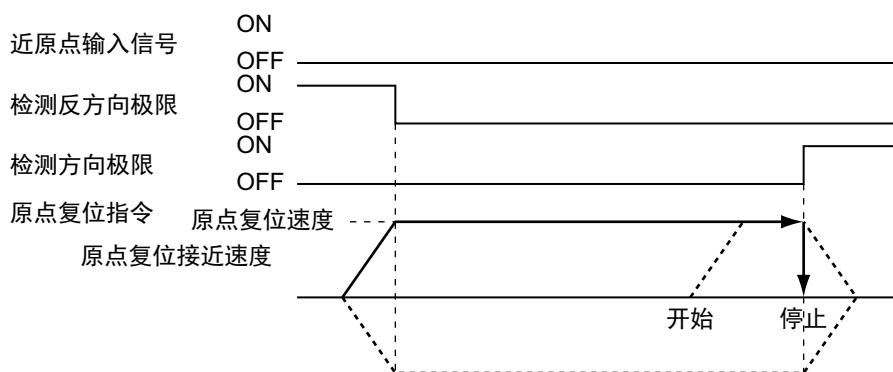
使用 NX 系列位置接口单元时，无法选择使用推压的原点复位动作模式。详情请参考 □ 《NX 系列位置接口单元用户手册 (SBCE-CN5-374)》。

正方向极限输入时动作和负方向极限输入时动作

- 原点复位过程中达到动作方向的极限输入时，选择是反转后继续执行原点复位还是不反转并异常停止。反转时，应同时选择停止方法。
- 若设定为 [反转]，在原点复位接近速度下原点检测方向的极限信号变为“ON”时，则异常停止。但是，在不使用接近信号的原点复位动作模式 [13: 无近原点输入、推压原点输入指定] 时，不会变为异常停止。



- 此外，两个方向的极限输入时动作都设定为 [反转] 时，如果在相反侧的极限输入到其他极限输入之间向原点检测方向动作，但未检测到原点时，将异常停止。



原点复位开始方向

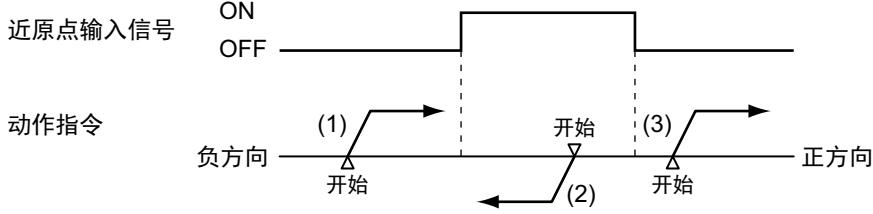
选择开始原点复位时的方向。

但是，在包含附近退避引起的反转动作的原点复位动作模式中，若在近原点信号为“ON”的状态下启动原点复位，无论原点复位开始方向的设定如何，始终向原点检测方向的反方向运动。

包含附近退避引起的反转动作的原点复位动作模式有以下 4 种。

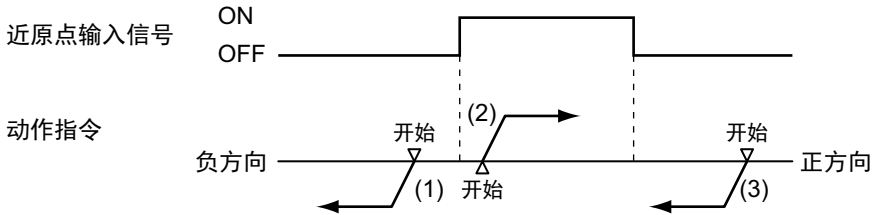
- 0: 附近退避、近原点输入 OFF 指定
- 1: 附近退避、近原点输入 ON 指定
- 9: 附近退避、原点输入掩码距离指定
- 12: 附近退避、推压时间指定

原点复位开始方向：正
原点检测方向：正



- (1)、(3) : 在近原点信号为“OFF”的状态下开始，按原点复位开始方向开始动作
- (2) : 在近原点信号为“ON”的状态下开始，按原点检测方向的反方向开始动作

原点复位开始方向：负
原点检测方向：负



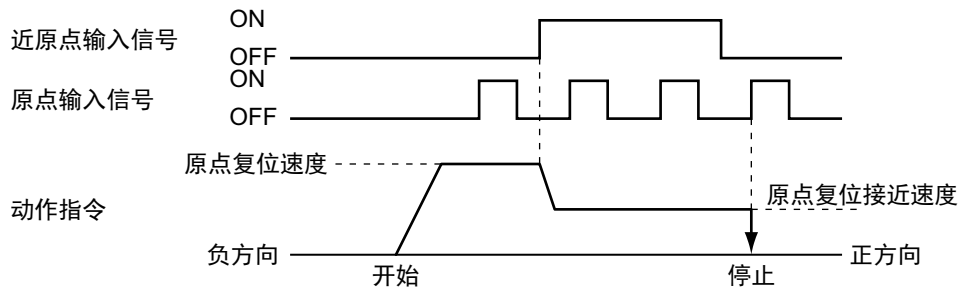
- (1)、(3) : 在近原点信号为“OFF”的状态下开始，按原点复位开始方向开始动作
- (2) : 在近原点信号为“ON”的状态下开始，按原点检测方向的反方向开始动作

原点检测方向

选择原点输入检测方向。

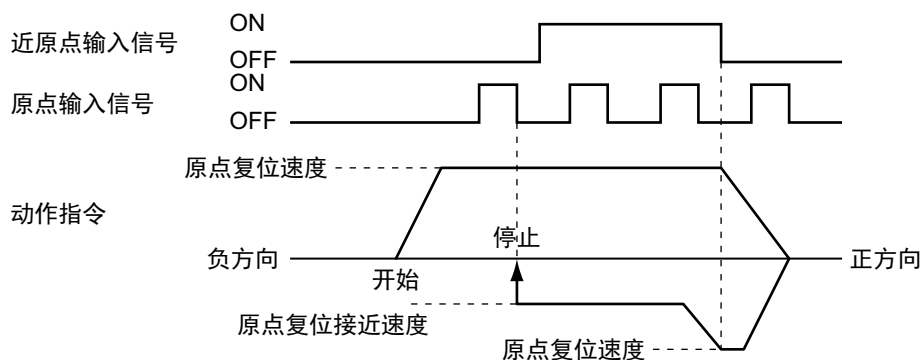
原点检测方向为 FALSE（正方向）时，动作如下所示。

动作示例：附近退避、近原点输入OFF指定



此外，原点检测方向为 TRUE（负方向）时，动作如下所示。

动作示例：附近退避、近原点输入OFF指定



原点输入掩码距离

设定原点复位动作模式为“9：附近退避、原点输入掩码距离指定”时的固定尺寸距离。
动作详情请参考 □ 「9：附近退避、原点输入掩码距离指定的动作 (P.3-25)」。

原点复位修正值

设定原点检测后的原点复位修正量。
此时的移动速度在 [原点复位修正速度] 中设定。
动作详情请参考 □ 「原点修正 (P.3-31)」。

原点位置偏置

原点复位动作完成时的当前位置偏置。
[原点位置偏置] 中设定的值为当前位置。

原点复位速度

原点复位时的高速速度。

原点复位接近速度

原点复位时的接近速度。

原点复位修正速度

设定了 [原点复位修正值] 时的移动速度。
动作详情请参考 □ 「原点修正 (P.3-31)」。

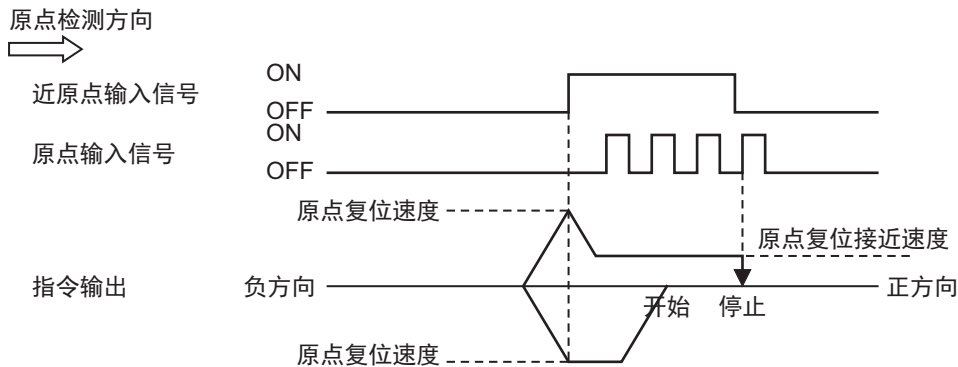
指令的详情

● 原点确定的动作

下面介绍 10 种原点复位动作模式。

0: 附近退避、近原点输入 OFF 指定的动作

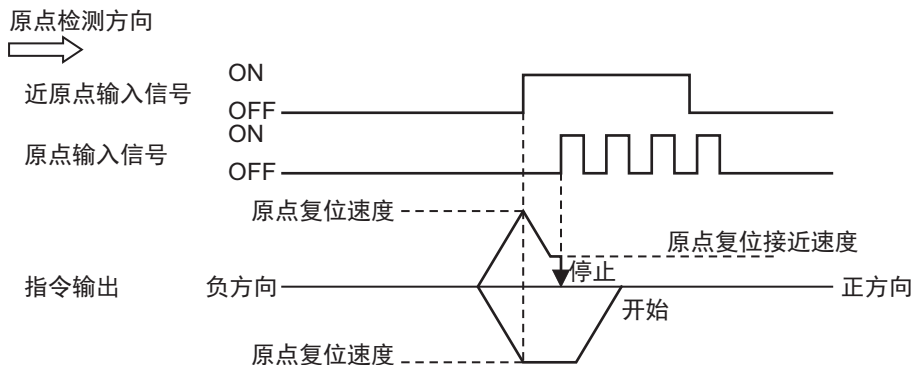
- 1** 以原点复位速度启动，利用近原点输入信号的“OFF” → “ON”，开始减速至原点复位接近速度。
- 2** 达到原点复位接近速度，根据近原点输入信号从“ON”变为“OFF”后的最初原点输入停止，确定机械原点。



- 若在近原点输入信号为“ON”的状态下进行原点复位，将向原点检测方向的反方向以原点复位速度开始动作。在近原点输入信号为“ON”的状态下，先进行退避后，再向原点检测方向以原点复位速度开始原点复位动作。
- 原点复位开始后，在低于原点复位接近速度的状态下，近原点输入信号从“OFF”变为“ON”再变为“OFF”后，将原点输入信号变为“ON”，也可确定机械原点。

1: 附近退避、近原点输入 ON 指定的动作

- 1** 以原点复位速度启动，利用近原点输入信号的“OFF” → “ON”，开始减速至原点复位接近速度。
- 2** 根据达到原点复位接近速度后的最初原点输入停止，确定机械原点。
设为“ON”后的近原点输入信号状态不会影响动作。

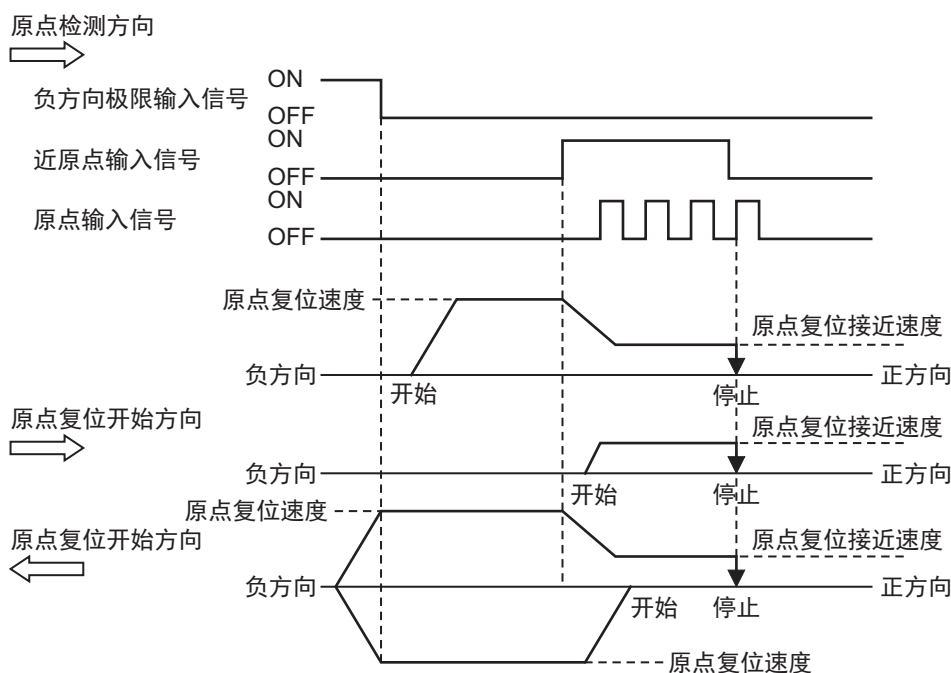


- 若在近原点输入信号为“ON”的状态下进行原点复位，将向原点检测方向的反方向以原点复位速度开始动作。在近原点输入信号为“ON”的状态下，先进行退避后，再向原点检测方向以原点复位速度开始原点复位动作。

- 原点复位开始后，在低于原点复位接近速度的状态下，近原点输入信号变为“ON”后，将原点输入信号变为“ON”，也可确定机械原点。

4: 近原点输入 OFF 指定的动作

- 1 以原点复位速度启动，利用近原点输入信号的“OFF”→“ON”，开始减速至原点复位接近速度。
- 2 达到原点复位接近速度，根据近原点输入信号从“ON”变为“OFF”后的最初原点输入停止，确定机械原点。



- 若在近原点输入信号为“ON”的状态下进行原点复位，将根据原点复位开始方向的设定，进行以下动作。

原点复位开始方向和原点检测方向相同时

不执行退避动作，向原点检测方向以原点复位接近速度开始原点复位动作。

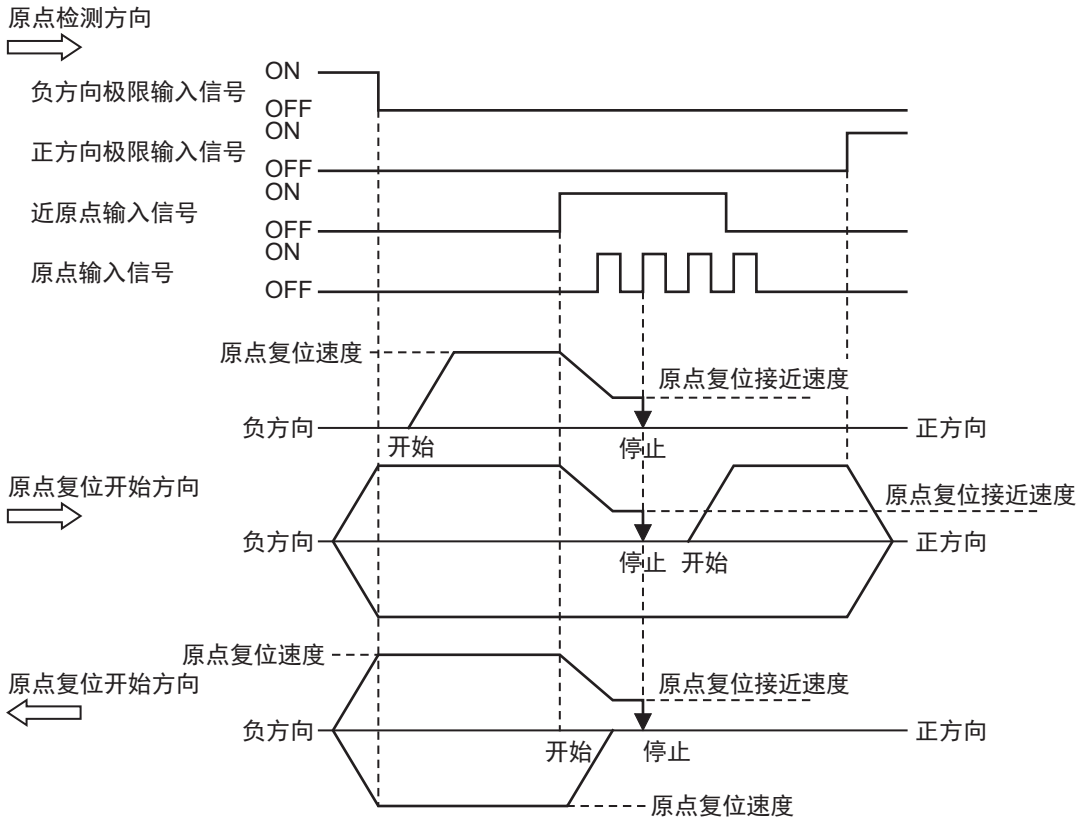
原点复位开始方向和原点检测方向不同时

不变为近原点输入信号的状态，以原点复位速度向原点复位开始方向开始动作。原点复位开始时间的极限输入时动作设定为“反转”时，检测到极限输入时反转，向原点检测方向执行原点复位动作。

- 原点复位开始后，在低于原点复位接近速度的状态下，近原点输入信号从“OFF”变为“ON”再变为“OFF”后，将原点输入信号变为“ON”，也可确定机械原点。

5: 近原点输入 ON 指定的动作

- 1 以原点复位速度启动，利用近原点输入信号的“OFF” → “ON”，开始减速至原点复位接近速度。
- 2 根据达到原点复位接近速度后的最初原点输入停止，确定机械原点。
设为“ON”后的近原点输入信号状态不会影响动作。



- 若在近原点输入信号为“ON”的状态下进行原点复位，将根据原点复位开始方向的设定，进行以下动作。

原点复位开始方向和原点检测方向相同时

不执行退避动作，向原点检测方向以原点复位速度开始原点复位动作。

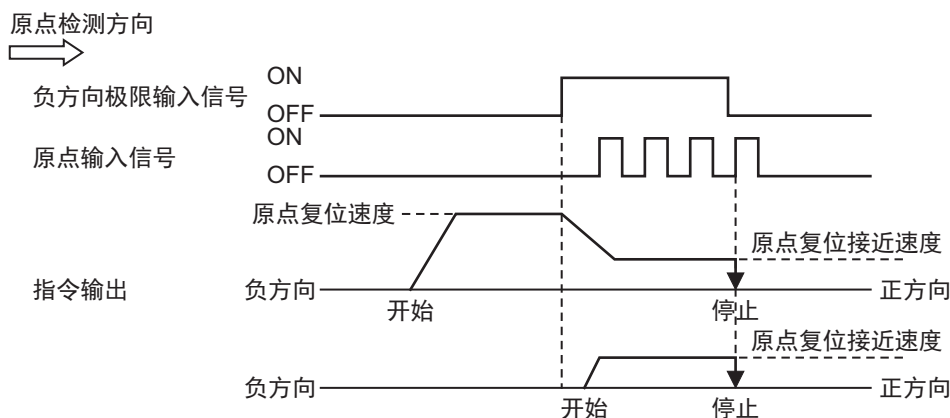
原点复位开始方向和原点检测方向不同时

不变为近原点输入信号的状态，以原点复位速度向原点复位开始方向开始动作。原点复位开始时间的极限输入时动作设定为“反转”时，检测到极限输入时反转，向原点检测方向执行原点复位动作。

- 原点复位开始后，在低于原点复位接近速度的状态下，近原点输入信号从“OFF”变为“ON”后，将原点输入信号变为“ON”，也可确定机械原点。

8: 极限输入 OFF 指定的动作

- 1 以原点复位速度启动，利用与原点检测方向反方向的极限信号的“OFF”→“ON”，开始减速至原点复位接近速度。
- 2 达到原点复位接近速度，根据极限信号从“ON”变为“OFF”后的最初原点输入停止，确定机械原点。



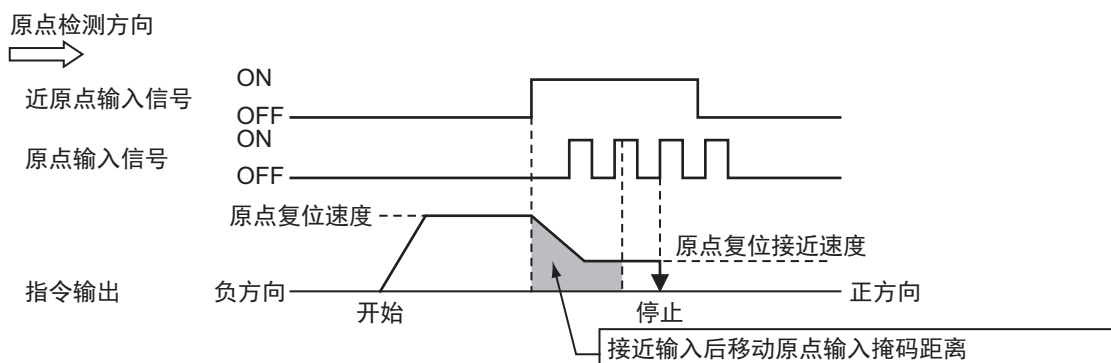
- 若在与原点检测方向相反的极限输入为“ON”的状态下进行原点复位，将向原点检测方向以原点复位接近速度执行原点复位动作。
- 原点复位开始后，在低于原点复位接近速度的状态下，与检测方向相反的极限信号从“OFF”变为“ON”再变为“OFF”后，将原点输入信号变为“ON”，也可确定机械原点。

9: 附近退避、原点输入掩码距离指定的动作

- 1 以原点复位速度启动，利用近原点输入信号的“OFF”→“ON”，开始减速至原点复位接近速度。
- 2 近原点输入信号从“OFF”变为“ON”后，按原点输入掩码距离移动结束后，以最初原点输入停止，确定机械原点。

设为“ON”后的近原点输入信号状态不会影响动作。

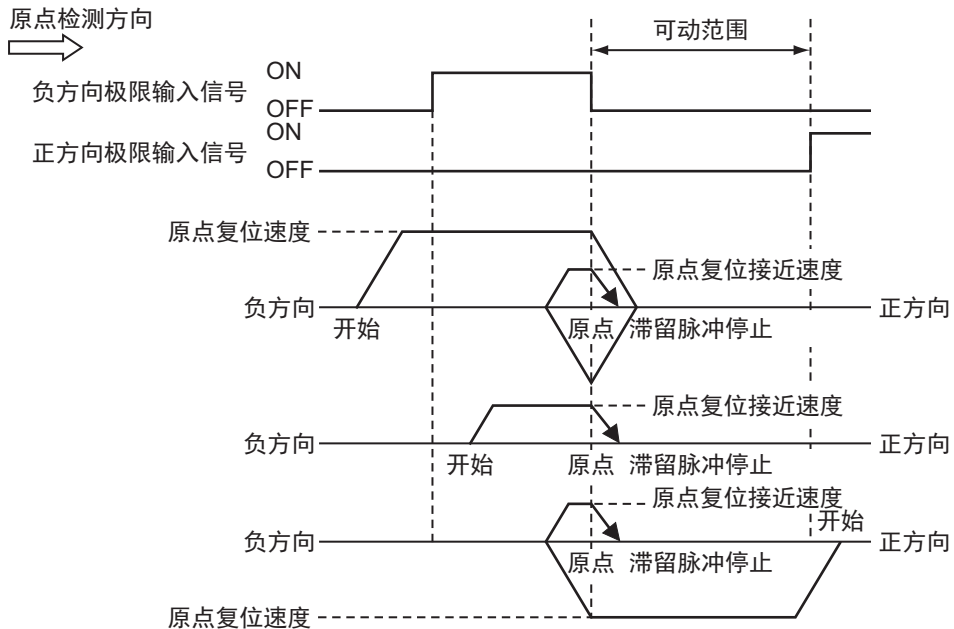
此时，如果设定的移动量较短，在减速至原点复位接近速度前达到移动量时，如果进行原点复位，将发生“原点输入掩码距离不正确（错误代码：742B Hex）”。



- 若在近原点输入信号为“ON”的状态下进行原点复位，将向原点检测方向的反方向以原点复位速度开始动作。在近原点输入信号为“ON”的状态下，先进行退避后，再向原点检测方向以原点复位接近速度开始原点复位动作。
- 原点复位开始后，在低于原点复位接近速度的状态下，近原点输入信号变为“ON”后，按原点输入掩码距离移动，将原点输入信号变为“ON”，也可确定机械原点。

11: 仅极限输入时的动作

- 1 以原点复位速度启动，利用与原点检测方向反方向的极限信号的“ON”→“OFF”，减速停止。
- 2 停止后，根据原点复位速度反转，利用极限信号的“OFF”→“ON”减速停止。
- 3 停止后，根据原点复位接近速度反转，利用极限信号的“ON”→“OFF”，将该位置作为原点，立即停止（滞留脉冲停止）。但是，不会恢复到机械原点位置。



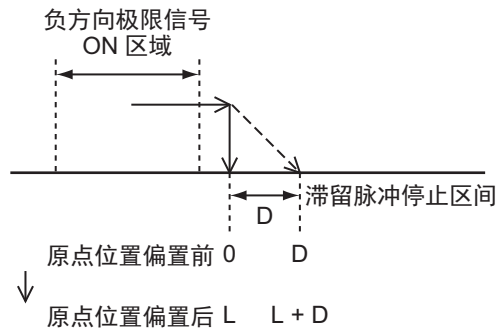
- 只使用极限信号进行原点复位时，将朝原点检测方向动作时的极限信号的“ON”→“OFF”变化点作为原点检测位置。
- 若在原点检测方向的极限信号为“ON”的状态下进行原点复位，将向原点检测方向的反方向以原点复位速度开始动作。利用与原点检测方向反方向的极限信号的“OFF”→“ON”，减速停止。
- 检测到极限信号后，如果在减速停止前将极限信号设为“OFF”，不会确定机械原点，将继续减速。此时，不会发生异常。
- 在反转后或极限信号输入状态下启动后，如果在达到原点复位接近速度前将极限信号从“ON”变为“OFF”，将确定机械原点。

原点复位动作模式 11 和其他原点复位动作模式的不同

模式 11 是指利用 MC 功能模块的处理来检测“极限信号的 ON → OFF”，以确定原点的原点复位动作。因此，与将原点输入作为外部锁定信号使用的原点复位动作模式相比，有以下不同。请了解模式 11 和其他原点复位动作模式的不同，然后再使用。

- 在极限输入的检测中，不使用伺服驱动器等的外部锁定功能，通过 MC 功能模块的处理进行。MC 功能模块的处理是指：任务为主固定周期任务时，以主周期间隔执行的处理；为固定周期任务（执行优先度 5）时，以固定周期任务（执行优先度 5）的任务周期执行的处理。因此，原点确定的精度取决于原点复位接近速度、主固定周期任务、固定周期任务（执行优先度 5）的控制周期。
- 不会恢复到机械原点。原点修正量为“0”时，将停止在与机械原点不同的位置，并结束原点复位处理。
- 原点修正量设定为“0”时，不执行原点修正动作。原点修正量为“0”时，如上所述，将在与机械原点不同的位置结束原点复位处理。原点修正量为“0”以外时，与其他原点复位动作模式相同，将原点补偿量作为与机械原点的相对量进行动作。

- 原点位置偏置将变更机械原点位置。如下图所示，停止位置从原点错开“D”时，如果原点位置偏置为“L”、原点修正量为“0”，则原点复位处理结束时的位置为“L + D”。

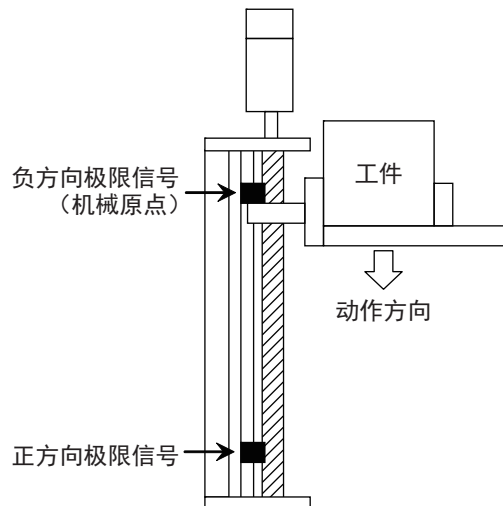


使用注意事项

- 检测到“极限信号 ON → OFF”后，在进行滞留脉冲停止并开始原点修正动作前，将忽略与原点检测方向反方向的极限信号。
- 如果原点修正量为“0”，且在原点确定后与原点检测方向反方向的极限输入信号立即变为“ON”，请将原点修正量设定为远离极限信号输入的方向。

原点复位动作模式 11 的使用示例

如下图所示，没有空间同时安装“负方向极限信号”和“原点信号”时，可令“负方向极限信号”同时具备极限信号和机械原点的作用。



12: 附近退避、推压时间指定的动作

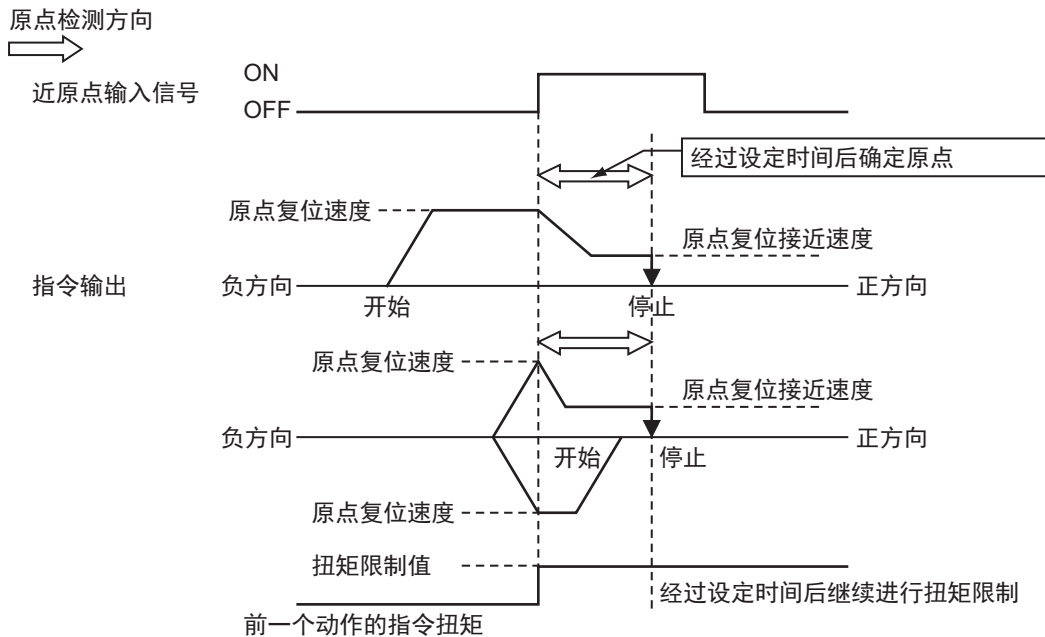
- 1** 以原点复位速度启动，利用近原点输入信号的“OFF” → “ON”，开始减速至原点复位接近速度。在开始减速的同时，开始时间监视。开始推压时的扭矩限制在欧姆龙生产的伺服驱动器 G5 系列和其他伺服上有以下不同。

G5 系列 : 自动执行扭矩限制。
 上述以外的伺服 : 按设定执行扭矩限制。

推压动作请参考 □□「推压动作（使用欧姆龙生产的伺服驱动器 G5 系列时）(P.3-31)」。

2 经过设定时间后，确定机械原点。

设为“ON”后的近原点输入信号状态不会影响动作。



- 若在近原点输入信号为“ON”的状态下进行原点复位，将向原点检测方向的反方向以原点复位速度开始动作。在近原点输入信号为“ON”的状态下，先进行退避后，再向原点检测方向以原点复位接近速度开始原点复位动作。
- 扭矩限制的解除在欧姆龙生产的伺服驱动器 G5 系列和其他伺服上也不同。

G5 系列 : 原点复位后，首次向与原点复位相反的方向运动时，自动变为“OFF”。

上述以外的伺服 : 使用扭矩限制时，在原点复位后，首次向与原点复位相反的方向移动时，请解除扭矩限制。重写扭矩限制时，请使用 EC_CoESDOWrite (CoE SDO 写入) 指令。

- 检测到近原点输入信号后，达到原点复位接近速度前，即使经过推压时间，也不会发生异常，仍会确定机械原点。
- 原点复位开始后，在低于原点复位接近速度的状态下，近原点输入信号从“OFF”变为“ON”后，即使经过推压时间，也可确定机械原点。

13: 无近原点输入、推压原点输入指定的动作

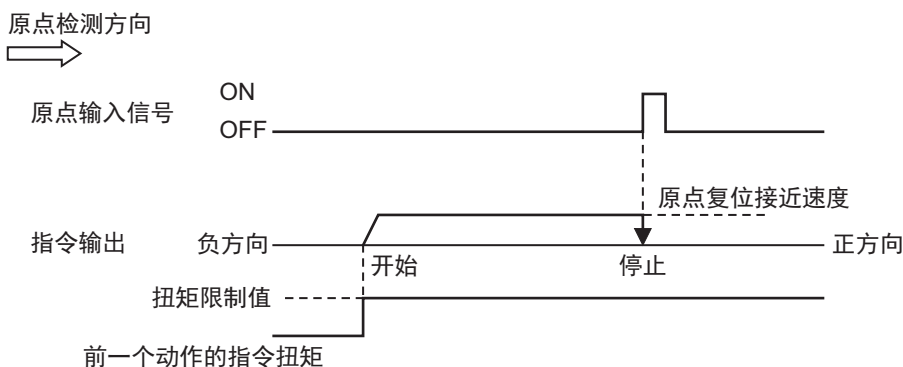
1 以原点复位接近速度启动。

开始推压时的扭矩限制在欧姆龙生产的伺服驱动器 G5 系列和其他伺服上有以下不同。

G5 系列	: 自动执行扭矩限制。
上述以外的伺服	: 根据需要执行扭矩限制。

推压动作请参考 □ 「推压动作（使用欧姆龙生产的伺服驱动器 G5 系列时）(P.3-31)」。

2 利用原点输入的“OFF”→“ON”，确定机械原点。



- 扭矩限制的解除在欧姆龙生产的伺服驱动器 G5 系列和其他伺服上也不同。

G5 系列	: 原点复位后，首次向与原点复位相反的方向运动时，自动变为“OFF”。
上述以外的伺服	: 使用扭矩限制时，在原点复位后，首次向与原点复位相反的方向移动时，请解除扭矩限制。重写扭矩限制时，请使用 EC_CoESDOWrite（CoE SDO 写入）指令。

- 原点复位开始后，在达到原点复位接近速度前，即使将原点输入信号变为“ON”，也可确定机械原点。

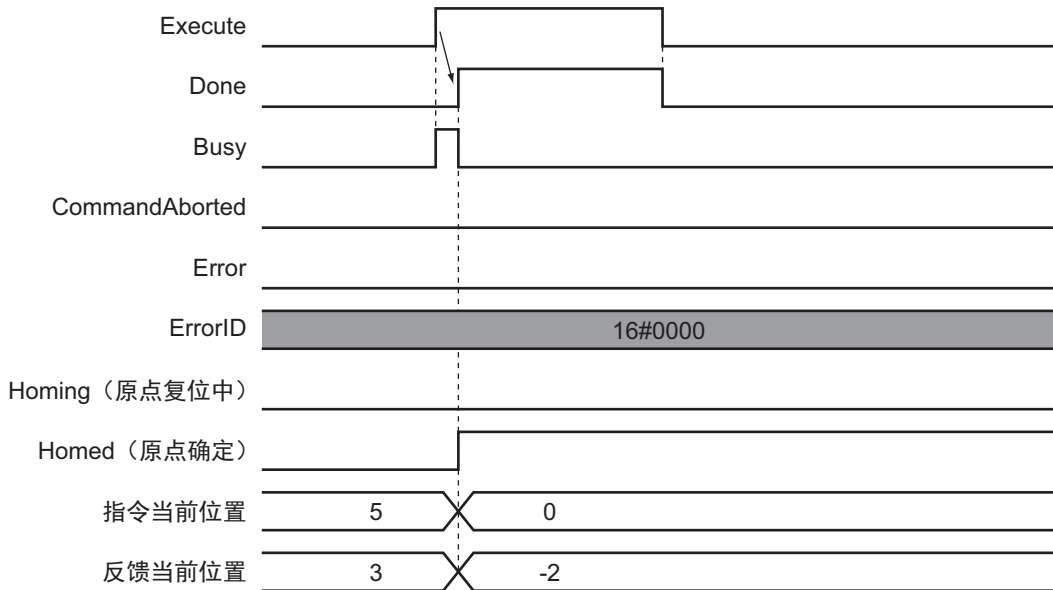


使用注意事项

在欧姆龙产伺服驱动器 G5 系列 EtherCAT 通信内置直线电机型中，无法将 Z 相输入映射到 PDO 中。因此，若使用原点复位动作中的“13: 无近原点输入、推压原点输入指定”，需要用到 PDO 中映射的 Z 相输入，因此请勿在原点输入信号中选择 Z 相输入。

14: 原点预设的动作

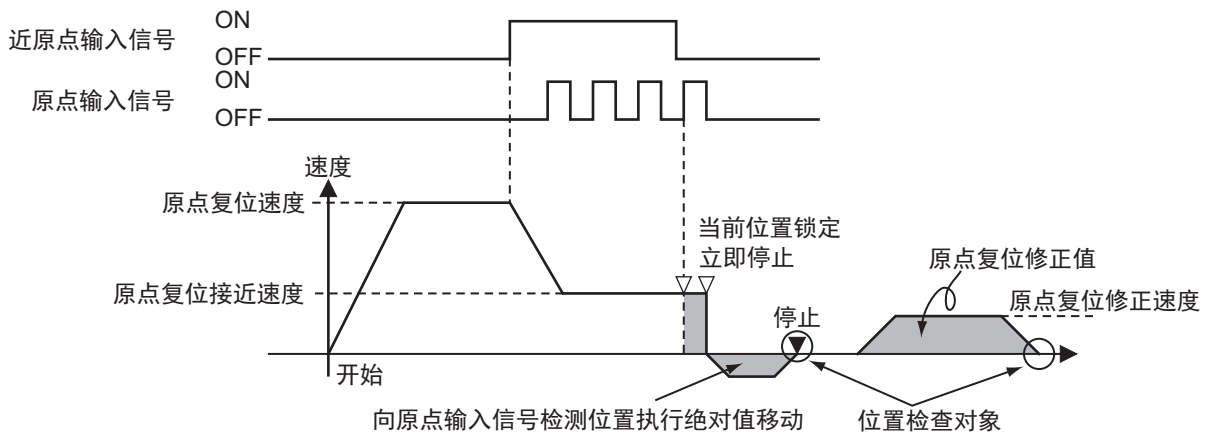
将指令当前位置变更为原点位置偏置，确定机械原点。
此外，将保持此时的指令当前位置和反馈位置的偏差。



● 原点确定时的停止方法和位置检查

原点确定时，按以下步骤停止。

- 1** 锁定检测到原点输入信号时的当前位置，立即停止。
- 2** 立即停止后，以绝对移动的方式反转移动到 1 中锁定的位置。



- 无论轴参数“位置检查时间”的设定如何，本指令将始终等待定位完成。
特别是向原点输入信号检测位置进行绝对移动时，以及位置检查时间设定为“0”时，将一直等待定位完成，请设定为确定进入位置宽度内。
在原点复位修正值的动作中，即使位置检查时间设定为“0”，也不会进行位置等待。
此外，是否处于位置等待状态可通过运动控制系统变量“lnPosWaiting（位置等待）”确认。

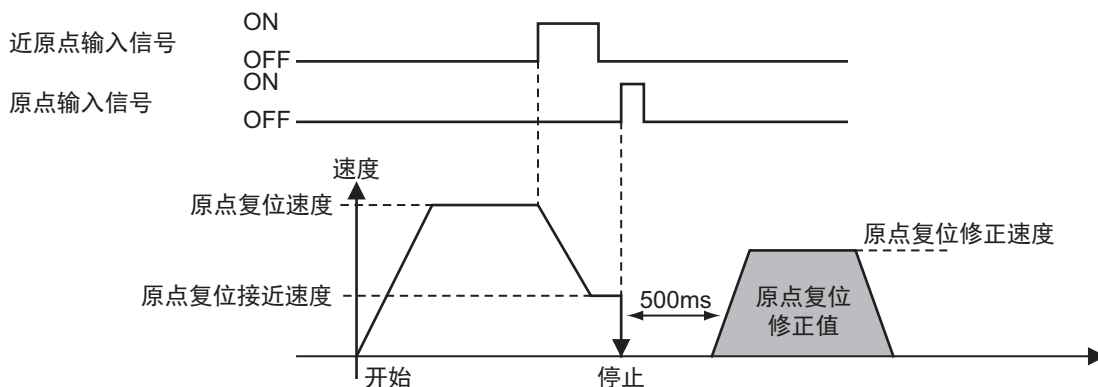
● 原点修正

设定原点复位修正值，在检测到原点输入后，将根据原点复位修正值移动，并确定原点。

通过机械类确定机械原点后，将很难调整工件，需要微调时，请将从最初确定的机械原点开始的偏差量设定为修正数据。

此时的移动速度为原点复位修正速度，检测到原点输入并停止后，经过 500ms 再开始移动。

原点复位修正值的符号表示相对于坐标轴的方向，原点复位修正值为负数时，向负方向移动。



● 超驰

超驰对本指令无效。

● 扭矩限制的自动控制

使用欧姆龙生产的伺服驱动器 G5 系列时，原点复位动作选择为“12: 附近退避、推压时间指定”、“13: 无近原点输入、推压原点输入指定”之一时，将对推压方向自动开始扭矩限制。

原点复位时的扭矩限制自动控制在原点复位后仍将保持扭矩限制状态，向与原点检测方向相反的方向动作时，扭矩限制变为无效。

此外，以下情况下也会自动解除扭矩限制。

- 伺服 OFF（伺服解锁）
- 从周期性同步位置控制（CSP）切换为其他控制模式时

原点复位动作过程中，向与原点检测方向相反的方向动作时，将解除扭矩限制。

例如，极限输入时的反转动作方向与原点检测方向相反时，将在反转完成时解除扭矩限制。

此外，反转后动作方向为原点检测方向时，再次将扭矩限制设为有效。

● 推压动作（使用欧姆龙生产的伺服驱动器 G5 系列时）

原点复位的 [12: 附近退避、推压时间指定]、[13: 无近原点输入、推压原点输入指定] 将使用伺服驱动器中事先设定的扭矩限制值，向原点检测方向自动开始扭矩限制。



使用注意事项

使用其他公司生产的伺服驱动器时，MC_Home（原点复位）指令的自动扭矩限制功能不会启用。

请使用 MC_SetTorqueLimit（扭矩限制）指令、SDO 通信、支持工具等，设定为合适的值。



参考

- 原点复位正常完成后，继续进行扭矩限制。
- 启动了向反方向移动的指令时，将自动解除扭矩限制。

欧姆龙生产的伺服驱动器 G5 系列的设定

要使用推压动作，需要利用伺服驱动器的支持软件，设定伺服驱动器 G5 系列的扭矩限制选择（3521 Hex）。

- 原点复位推压动作时，对原点检测方向进行扭矩控制，且在其他动作中使用 MC_SetTorqueLimit（扭矩限制）指令的扭矩限制方向或扭矩限制值时，请使用“6”。
此时，在原点复位的推压动作过程中，MC_SetTorqueLimit（扭矩限制）指令的输入变量的内容将被忽略。
- 设定为“4”时，始终适用 MC_SetTorqueLimit（扭矩限制）指令的输入变量内容。原点复位的推压动作时以及其他动作时，需要设定为合适的扭矩限制值。

		扭矩限制选择（3521 Hex）	
		6（推荐）	4
正方向扭矩限制值	原点复位中	适用第 3 扭矩限制（3525 Hex）。	适用 MC_SetTorqueLimit（扭矩限制）指令的 PositiveValue（正方向扭矩限制值）和第 1 扭矩限制（3013 Hex）中的较小者。
	原点复位以外	适用 MC_SetTorqueLimit（扭矩限制）指令的 PositiveValue（正方向扭矩限制值）和第 1 扭矩限制（3013 Hex）中的较小者。	
负方向扭矩限制值	原点复位中	适用第 4 扭矩限制（3526 Hex）。	适用 MC_SetTorqueLimit（扭矩限制）指令的 NegativeValue（负方向扭矩限制值）和第 2 扭矩限制（3522 Hex）中的较小者。
	原点复位以外	适用 MC_SetTorqueLimit（扭矩限制）指令的 NegativeValue（负方向扭矩限制值）和第 2 扭矩限制（3522 Hex）中的较小者。	

扭矩限制请参考 □ 「MC_SetTorqueLimit(P.3-316)」。

此外，伺服驱动器 G5 系列的设定请参考 □ 《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型 用户手册 (SBCE-CN5-365)》、□ 《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型 用户手册 (SBCE-CN5-366)》。

● 推压动作中的位置偏差监视

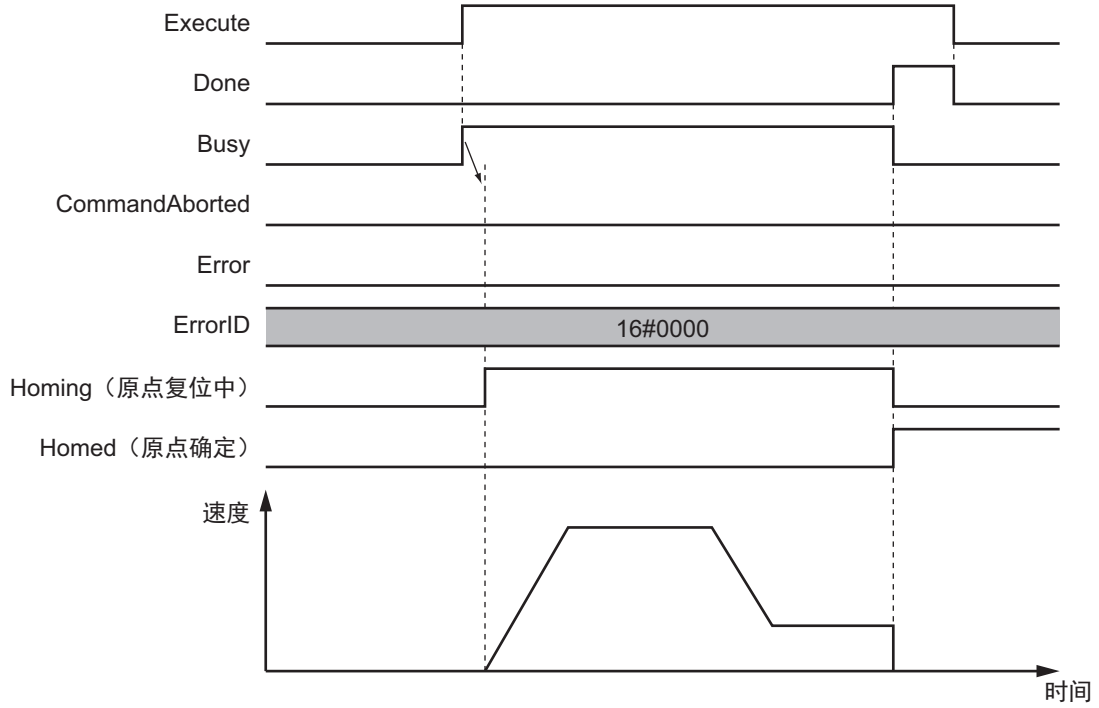
原点复位的推压动作中，位置偏差监视功能不会启用。

位置偏差监视功能的详情请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

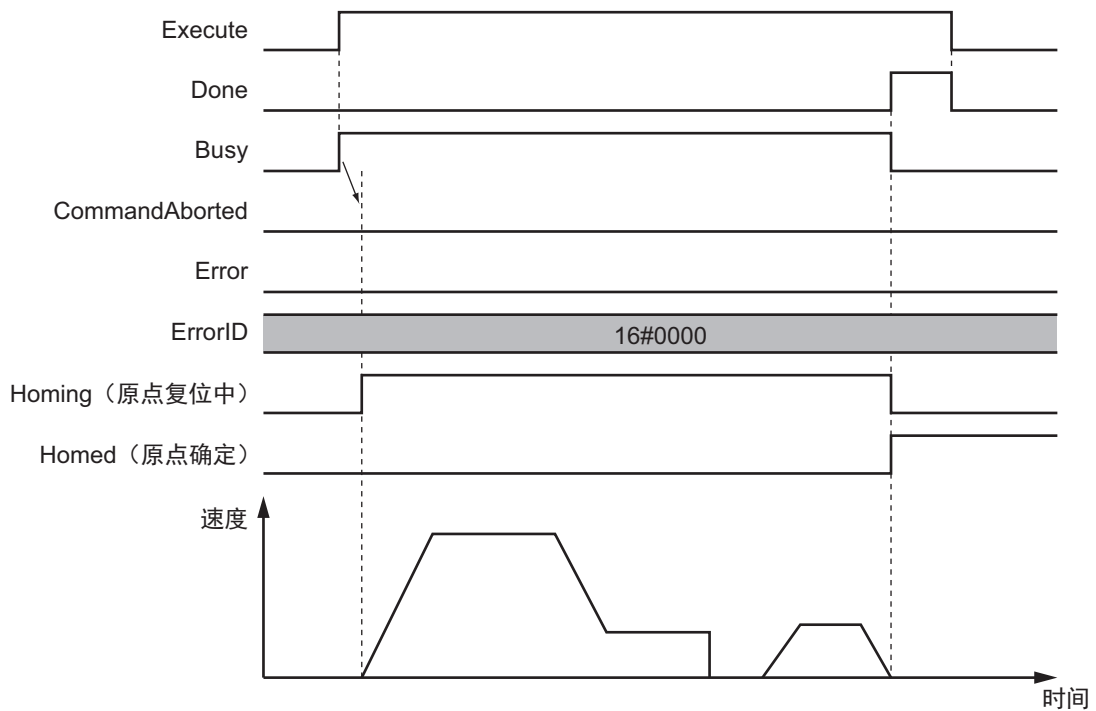
时序图

执行了原点复位时的时序图如下。

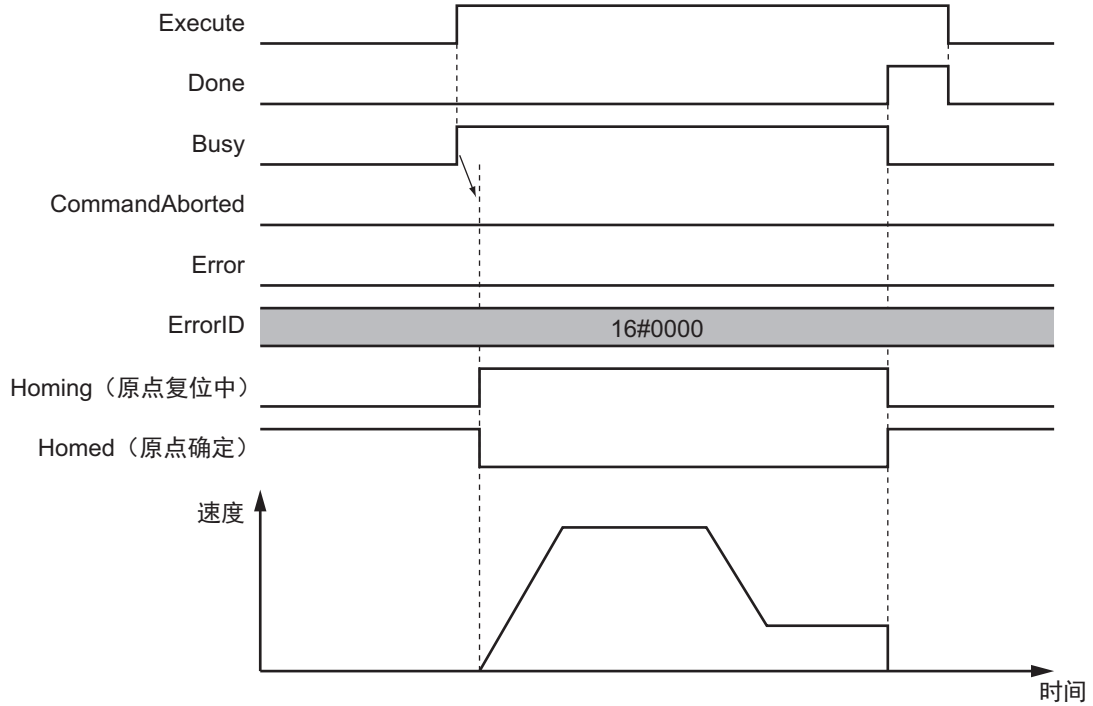
● 无原点修正时



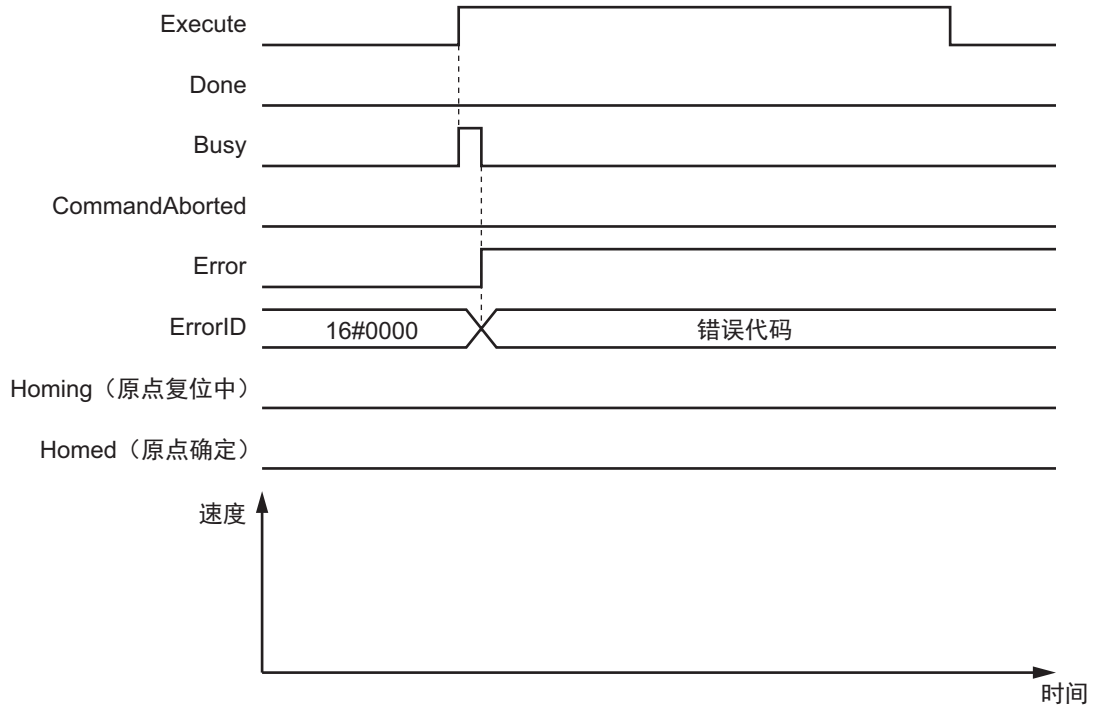
● 有原点修正时



● 在**原点确定**状态下执行了本指令时



● 在**参数不正确**的状态下执行了本指令，以及在**运动指令不可执行**的状态下执行了本指令时



运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

错误代码

指令中发生的异常请参考 □ 「A-1 错误代码一览 (P.A-2)」。

MC_HomeWithParameter

指定原点复位参数，驱动电机，使用极限信号、近原点信号、原点信号确定机械原点。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_HomeWithParameter	参数指定原点复位	FB		<pre>MC_HomeWithParameter_instance (Axis := 《参数》, HomingParameter := 《参数》, Execute := 《参数》, Done => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>



版本相关信息

本指令可在 CPU 单元版本 1.03 以上和 Sysmac Studio Ver.1.04 以上的组合中使用。

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、 FALSE	FALSE	上升时开始指令。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、 FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、 FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*1	发生异常时，输出错误代码。16#0000 表示正常。

*1. □□「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	指令完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动 (中断), 本指令中止时 因发生异常, 本指令中止时 正在发生异常时, 启动了本指令 正在执行 MC_Stop 指令时, 启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。*1
HomingParameter	原点复位参数	_sHOMING_REF	—	设定原点复位参数。*2

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称 (默认 “MC_Axis****”) 或系统定义变量的轴变量名称 (_MC_AX[*], _MC1_AX[*], _MC2_AX[*])。

*2. 请定义 _sHOMING_REF 型用户定义变量。

● _sHOMING_REF

成员变量	名称	数据类型	有效范围	功能
Homing Mode	原点复位动作	_eMC_HOMING_MODE	0: _mcHomeSwTurn HomeSwOff 1: _mcHomeSwTurn HomeSwOn 4: _mcHomeSwOff 5: _mcHomeSwOn 8: _mcLimitInputOff 9: _mcHomeSwTurn HomeMask 11: _mcLimitInputOnly 12: _mcHomeSwTurn HoldingTime 13: _mcNoHomeSw HoldingHomeInput 14: _mcHomePreset	指定要重写的原点复位动作。 0: 附近退避、近原点输入 OFF 指定 1: 附近退避、近原点输入 ON 指定 4: 近原点输入 OFF 指定 5: 近原点输入 ON 指定 8: 极限输入 OFF 指定 9: 附近退避、原点输入掩码距离指定 11: 仅极限输入 12: 附近退避、推压时间指定 13: 无近原点输入、推压原点输入指定 14: 原点预设
HomeInput	原点输入信号	_eMC_HOME_INPUT	0: _mcZPhase 1: _mcExternalSignal	选择原点输入信号。 0: 使用 Z 相输入 1: 使用外部原点输入*1
StartDir	原点复位开始方向	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection	设定原点复位启动时的开始方向。 0: 正方向 2: 负方向
HomeDir	原点检测方向	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection	设定原点复位的原点输入检测方向。 0: 正方向 2: 负方向

成员变量	名称	数据类型	有效范围	功能
PosiLmt Mode	正方向极限输入时动作	_eMC_ LIMIT_ REVERSE_ MODE	0:_mcErrorStop 1:_mcRevImmediateStop 2:_mcRevDeceleration Stop	设定在原点复位过程中将正方向极限输入设为“ON”时的停止方法。 0: 不反转、轴异常停止（停止方法遵照极限输入停止方法） 1: 反转、立即停止 2: 反转、减速停止
NegaLmt Mode	负方向极限输入时动作	_eMC_ LIMIT_ REVERSE_ MODE	0:_mcErrorStop 1:_mcRevImmediateStop 2:_mcRevDeceleration Stop	设定在原点复位过程中将负方向极限输入设为“ON”时的停止方法。 0: 不反转、轴异常停止（停止方法遵照极限输入停止方法） 1: 反转、立即停止 2: 反转、减速停止
Vel	原点复位速度	LREAL	正数	设定原点复位时的速度。 请设定为最高速度以下且原点复位接近速度 \leq 原点复位速度。 单位为 [指令单位 /s]。 ^{*2}
ApproachVel	原点复位接近速度	LREAL	正数	设定近原点输入变为“ON”后的速度。 请设定为最高速度以下且原点复位接近速度 \leq 原点复位速度。 单位为 [指令单位 /s]。 ^{*2}
Acc	原点复位加速度	LREAL	正数或“0”	指定原点复位时的加速度。 若设定为“0”，将不加速。 单位为 [指令单位 /s ²]。 ^{*2}
Dec	原点复位减速度	LREAL	正数或“0”	指定原点复位时的减速度。 若设定为“0”，将不减速。 单位为 [指令单位 /s ²]。 ^{*2}
Jerk	原点复位跃度	LREAL	正数或“0”	指定原点复位时的跃度。 若设定为“0”，将没有跃度。 脉冲单位请设定为 40 位以内。 单位为 [指令单位 /s ³]。 ^{*2}
Mask	原点输入掩码距离	LREAL	正数或“0”	设定原点复位动作模式为“附近退避、原点输入掩码距离指定”时的原点输入掩码距离。 单位为 [指令单位]。 ^{*2}
Offset	原点位置偏置	LREAL	负数、正数、“0”	原点复位完成时按设定的值预设当前位置。 旋转模式时，请设定为循环计数器下限值 \leq 原点位置偏置 $<$ 循环计数器上限值。此外，脉冲单位请设定为 40 位以内。 单位为 [指令单位]。 ^{*2}
PushTime	原点复位推压时间	UINT	0 ~ 10,000	以毫秒为单位设定原点复位动作模式为“附近退避、推压时间指定”时的推压时间。 单位为 [ms]。

成员变量	名称	数据类型	有效范围	功能
Compensation	原点复位修正值	LREAL	负数、正数、“0”	设定原点复位的原点确定后的原点复位修正量。 旋转模式时，请设定为 原点复位修正量 < 循环计数器上限值 - 循环计数器下限值 。 此外，脉冲单位请设定为 40 位以内。 单位为 [指令单位]。*2
Compensation Vel	原点复位修正速度	LREAL	正数	设定原点复位修正时的速度。 请设定为小于最高速度。 单位为 [指令单位 /s]。*2

*1. 使用欧姆龙生产的伺服驱动器 G5 系列时可以指定。将伺服驱动器的锁定功能 1 中分配的输入作为外部原点输入使用。在伺服驱动器 G5 系列的初始状态下，锁定功能 1 中分配了外部锁定输入 1。

详情请参考 □ □ 《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型 用户手册 (SBCE-CN5-365)》及 □ □ 《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型 用户手册 (SBCE-CN5-366)》。

*2. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

功能说明

- 对 Axis（轴）中指定的轴，在 Execute（启动）的上升沿开始原点复位动作。
- 参数指定原点复位指令中使用的各种参数在输入输出变量 HomingParameter 中设定。
- 执行本指令，不会变更轴参数。
- 本指令与 MC_Home（原点复位）指令的不同仅在于各种参数的设定方法。本指令的各种参数在输入输出变量 HomingParameter 中设定。MC_Home（原点复位）指令的各种参数在 Sysmac Studio 的轴参数 [原点复位动作] 中设定。

各种参数的设定方法以外的指令规格请参考 □ □ 「MC_Home(P.3-15)」。



使用注意事项

本指令中指定的原点复位参数仅在执行本指令时有效。执行本指令后执行了 MC_Home（原点复位）指令时，将根据轴参数中设定的原点复位参数进行动作。

原点复位动作和原点复位参数的关系

原点复位参数包含不按原点复位动作设定使用的参数。不使用的参数不在范围检查和整合检查的对象范围内。范围检查和整合检查在执行本指令时执行。

与原点复位动作的设定对应的各原点复位参数的使用和不使用如下表所示。

(○：使用参数、×：不使用参数)

原点复位动作	原点复位参数														
	原点输入信号	原点复位开始方向	原点检测方向	正方向极限输入时动作	负方向极限输入时动作	原点复位速度	原点复位接近速度	原点复位加速度	原点复位减速度	原点复位跃度	原点输入掩码距离	原点位置偏置	原点复位推压时间	原点复位修正值	原点复位修正速度
附近退避、近原点输入 OFF 指定	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
附近退避、近原点输入 ON 指定	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
近原点输入 OFF 指定	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
近原点输入 ON 指定	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
极限输入 OFF 指定	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
附近退避、原点输入掩码距离指定	○	○	○	○	○	○	○	○	○	○	○	○	×	○	○
仅极限输入	×	○	○	○	○	○	○	○	○	○	×	○	×	○	○
附近退避、推压时间指定	×	○	○	○	○	○	○	○	○	○	×	○	○	○	○
无近原点输入、推压原点输入指定	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
原点预设	×	×	×	×	×	×	×	×	×	×	×	○	×	×	×



参考

使用 NX 系列位置接口单元时，无法选择使用推压的原点复位动作模式。详情请参考 □□《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

运动指令多重启动的详情请参考 □□《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

错误代码

指令中发生的异常请参考 □□「A-1 错误代码一览 (P.A-2)」。

MC_Move

进行绝对值定位或相对值定位。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_Move	定位	FB		<pre> MC_Move_instance (Axis := 《参数》, Execute := 《参数》, Position := 《参数》, Velocity := 《参数》, Acceleration := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, Direction := 《参数》, BufferMode := 《参数》, MoveMode := 《参数》, Done => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》); </pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令
Position	目标位置	LREAL	负数、正数、“0”	0	在移动方法选择中指定绝对定位时，指定绝对坐标的目标位置。在移动方法选择中指定相对定位时，指定相对位置。单位为 [指令单位]。 ^{*1}
Velocity	目标速度	LREAL	正数	0	指定目标速度。 ^{*2} 单位为 [指令单位 /s]。 ^{*1}
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度。单位为 [指令单位 /s ²]。 ^{*1}
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。单位为 [指令单位 /s ²]。 ^{*1}
Jerk	跃度	LREAL	正数或“0”	0	指定跃度。单位为 [指令单位 /s ³]。 ^{*1}

输入变量	名称	数据类型	有效范围	初始值	内容
Direction	方向选择	_eMC_DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	0 *3	MoveMode 为 [0: 绝对值定位] *4, 计数模式为 [旋转模式] 时, 指定旋转方向。 0: 正方向指定 1: 最短距离指定 2: 负方向指定 3: 当前方向指定 4: 无方向指定
BufferMode	缓冲模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	指定运动指令多重启动时的动作。 0: 中断 1: 缓冲 2: 低共混 3: 前共混 4: 后共混 5: 高共混
MoveMode	移动方法选择	_eMC_MOVE_MODE	0: _mcAbsolute 1: _mcRelative	0 *3	选择移动方法。 0: 绝对值定位 1: 相对值定位

*1. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. 请务必设定目标速度。若不设定而直接动作, 将发生超出目标速度范围 (错误代码: 5422Hex)。

*3. 如果是有效范围为列举型的变量初始值, 实际不是数值, 而是列举值。

*4. MoveMode 为 [1: 相对值定位] 时, 根据 Position 的符号决定移动方向。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时, 变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*1	发生异常时, 输出错误代码。16#0000 表示正常。

*1. □ □ 「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	定位完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	开始了指令时	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动 (中断), 本指令中止时 因发生异常, 本指令中止时 正在发生异常时, 启动了本指令 正在执行 MC_Stop 指令时, 启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称 (默认 "MC_Axis****") 或系统定义变量的轴变量名称 (_MC_AX[*], _MC1_AX[*], _MC2_AX[*])。

功能说明

- 可进行绝对值定位或相对值定位。
- MoveMode (移动方法选择) 指定为 "绝对值定位" 时, 与 MC_MoveAbsolute (绝对值定位) 指令的动作相同。同样, 指定为 "相对值定位" 时, 与 MC_MoveRelative (相对值定位) 指令的动作相同。
- 作为相对值定位指令动作时, 不使用 Direction (方向选择)。

详情请参考 □ 「MC_MoveAbsolute(P.3-47)」或 □ 「MC_MoveRelative(P.3-73)」。

指令的详情

下面介绍本指令的详情。



使用注意事项

进行绝对值定位时，请在轴参数 [循环计数器上限设定值] 及 [循环计数器下限设定值] 的范围内指定目标位置。

如果指定的目标位置超出 [循环计数器上限设定值] 和 [循环计数器下限设定值] 的范围，将发生“超出目标位置设定范围（错误代码：5478Hex）”。

● 位置检查

本指令的位置检查根据轴参数 [位置宽度]、[位置检查时间] 的设定执行。

运动指令重新执行

在定位动作中变更输入变量，并再次将 Execute（启动）设为 TRUE，可变更本指令的动作。通过运动指令重新执行可变更的输入变量包括 Position（目标位置）、Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）。

运动指令重新执行的详情请参考  《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 其他指令执行过程中本指令的启动

对当前正在执行的指令启动本指令后，可切换为本指令或缓冲本指令。每个轴最多可缓冲 1 个。在 BufferMode（缓冲模式选择）中指定多重启动时本指令的动作。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 切换指令后如果轴的动作方向发生反转，将按照轴参数 [反转时动作] 执行反转动作。
缓冲	当前正在执行的指令正常完成后，自动启动已缓冲的本指令。
共混	将当前正在执行的指令到达目标位置时的速度（中继速度）作为启动速度，连续执行已缓冲的本指令。变更当前正在执行的指令动作，使其以中继速度到达目标位置。中继速度的指定方法有以下 4 种。
低共混	将当前正在执行的目标速度和缓冲中的目标速度中较低的速度作为中继速度。
前共混	将当前正在执行的目标速度作为中继速度。
后共混	将已缓冲的本指令的目标速度作为中继速度。
高共混	将当前正在执行的目标速度和缓冲中的目标速度中较高的速度作为中继速度。

BufferMode（缓冲模式选择）请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 本指令执行过程中其他指令的启动

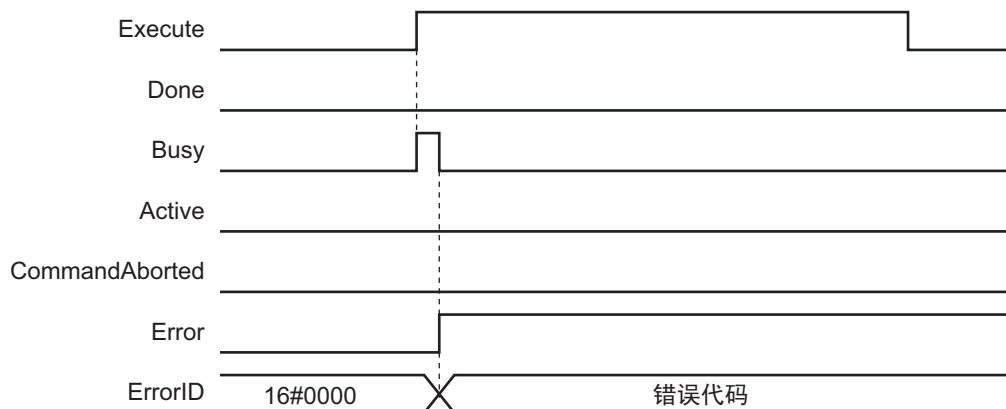
对本指令执行运动指令多重启动时，可进行中断、缓冲、共混。

异常

● 发生异常时的时序图

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。



● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

MC_MoveAbsolute

指定绝对坐标的目标位置，进行定位。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_MoveAbsolute	绝对值 定位	FB		<pre> MC_MoveAbsolute_instance (Axis := 《参数》, Execute := 《参数》, Position := 《参数》, Velocity := 《参数》, Acceleration := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, Direction := 《参数》, BufferMode := 《参数》, Done => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》); </pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Position	目标位置	LREAL	负数、正数、“0”	0	指定绝对坐标的目标位置。 单位为 [指令单位]。 *1
Velocity	目标速度	LREAL	正数	0	指定目标速度。 ^{*2} 单位为 [指令单位 /s]。*1
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度。 单位为 [指令单位 /s ²]。*1
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。 单位为 [指令单位 /s ²]。*1
Jerk	跃度	LREAL	正数或“0”	0	指定跃度。 单位为 [指令单位 /s ³]。*1

输入变量	名称	数据类型	有效范围	初始值	内容
Direction	方向选择	_eMC_ DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	0 *3	计数模式为 [旋转模式] 时, 指定旋转方向。 0: 正方向指定 1: 最短距离指定 2: 负方向指定 3: 当前方向指定 4: 无方向指定
BufferMode	缓冲 模式选择	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	指定运动指令多重启动时的动作。 0: 中断 1: 缓冲 2: 低共混 3: 前共混 4: 后共混 5: 高共混

*1. 指令单位请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. 请务必设定目标速度。若不设定而直接动作, 将发生异常。

*3. 如果是有效范围为列举型的变量初始值, 实际不是数值, 而是列举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、 FALSE	指令执行完成时, 变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、 FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、 FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*1	发生异常时, 输出错误代码。16#0000 表示正常。

*1. □ 「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	定位完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	开始了指令时	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动 (中断), 本指令中止时 因发生异常, 本指令中止时 正在发生异常时, 启动了本指令 正在执行 MC_Stop 指令时, 启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

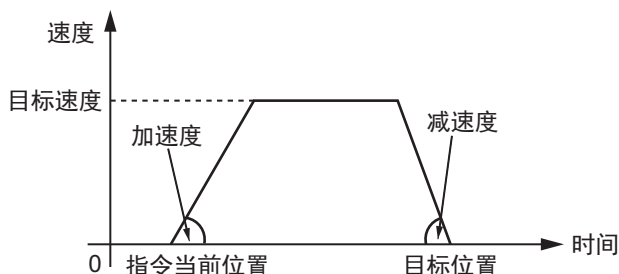
输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称 (默认“MC_Axis****”) 或系统定义变量的轴变量名称 (_MC_AX[*], _MC1_AX[*], _MC2_AX[*])。

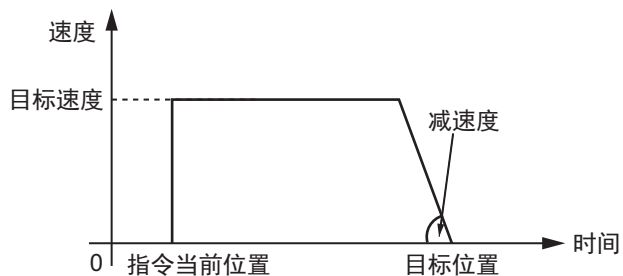
功能说明

- 指定绝对坐标的目标位置, 进行定位。
- 在 Execute (启动) 的上升沿开始绝对定位动作。
- 原点未确定状态下, 也可执行本指令。
- 输入变量中可指定 Velocity (目标速度)、Acceleration (加速度)、Deceleration (减速度)、Jerk (跃度)。

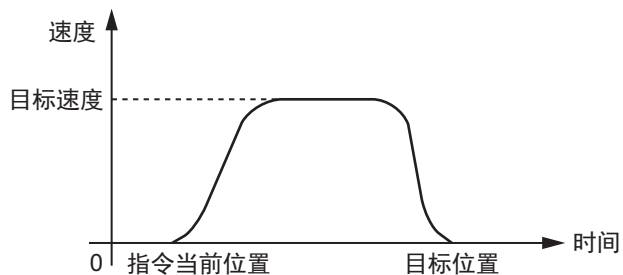
绝对定位的动作示例如下所示。



将加速度或减速度指定为“0”并启动时，将不进行加减速而达到目标速度。
加速度为“0”时的动作示例如下。



希望均匀加减速时，需指定 Jerk（跃度）。
指定了 Jerk（跃度）时的动作示例如下。



Jerk（跃度）的详情请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

将 Position（目标位置）指定为与指令当前位置相同并启动时，轴不会移动，但 Done（完成）变为 TRUE。

指令的详情

下面介绍指令的详情。

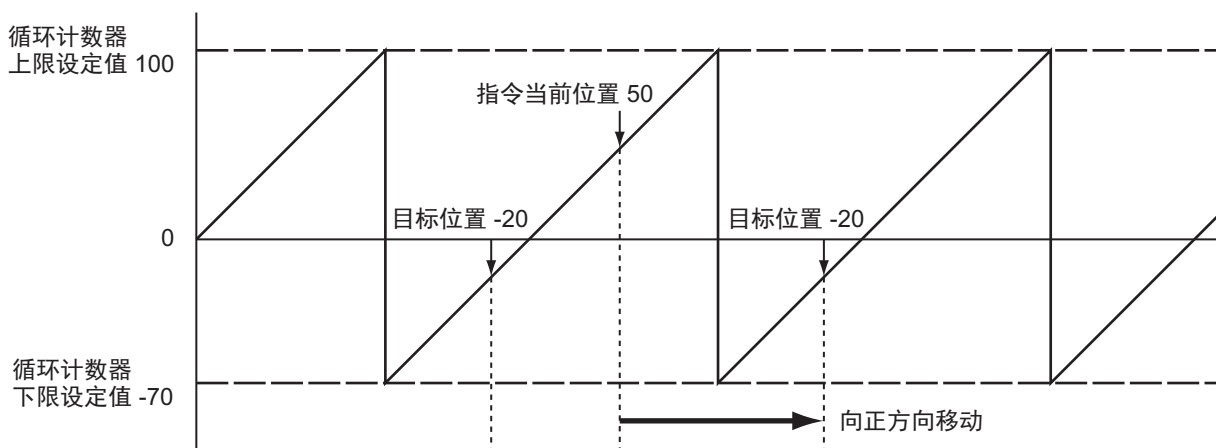
● Direction（方向选择）

计数模式为 [旋转模式] 时，在 Direction（方向选择）中指定开始定位的方向。

计数模式为 [直线模式] 时，将忽略 Direction（方向选择）。

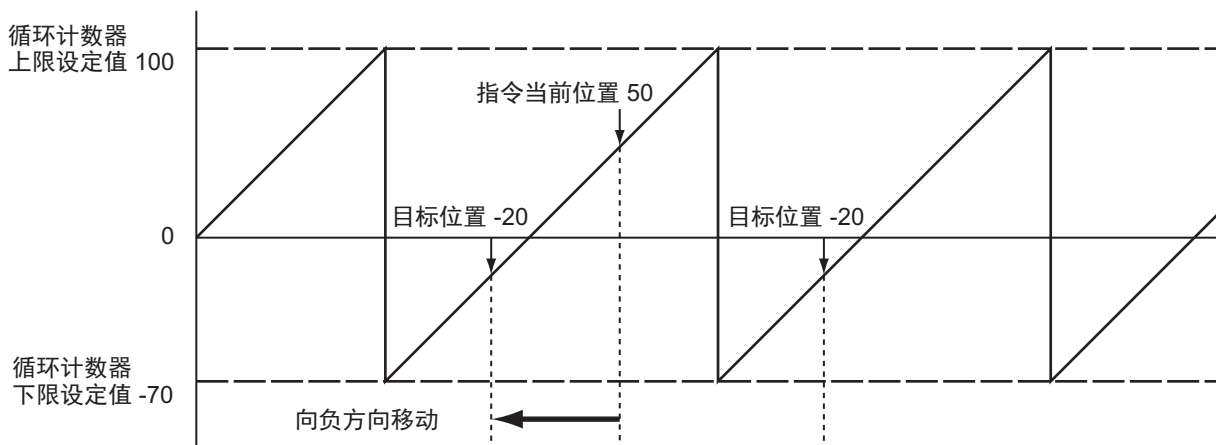
Direction（方向选择）为“正方向指定”时，向正方向的目标位置进行定位。

指令位置从“50”向“-20”进行定位时的动作示例如下所示。

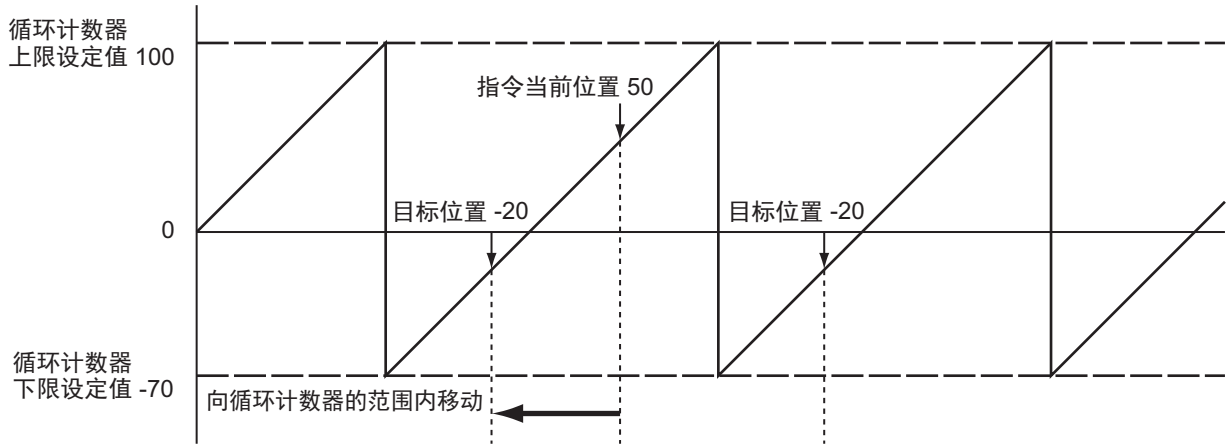


Direction（方向选择）为“负方向指定”时，向负方向的目标位置进行定位。

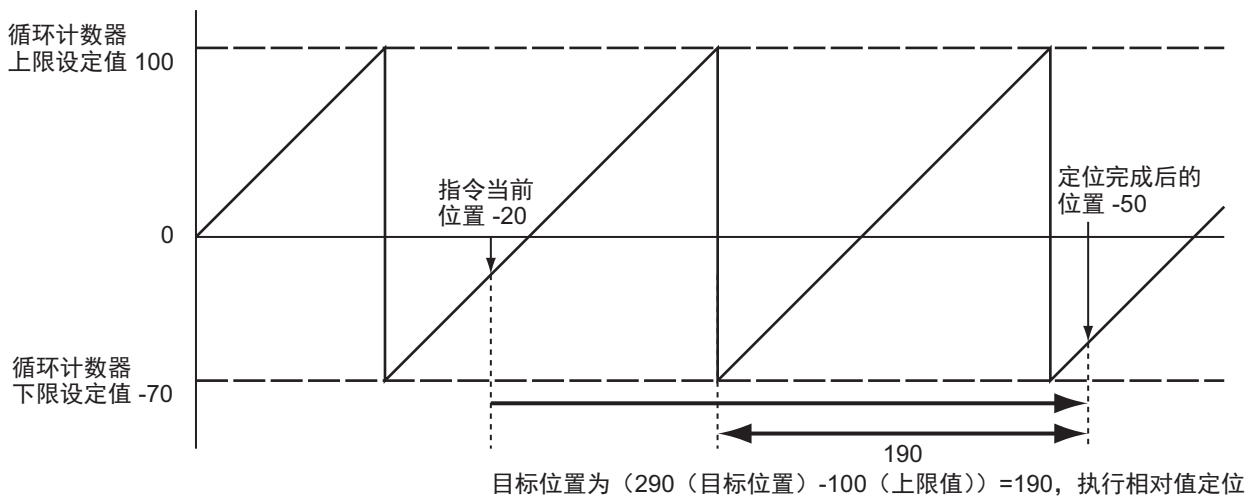
指令位置从“50”向“-20”进行定位时的动作示例如下所示。



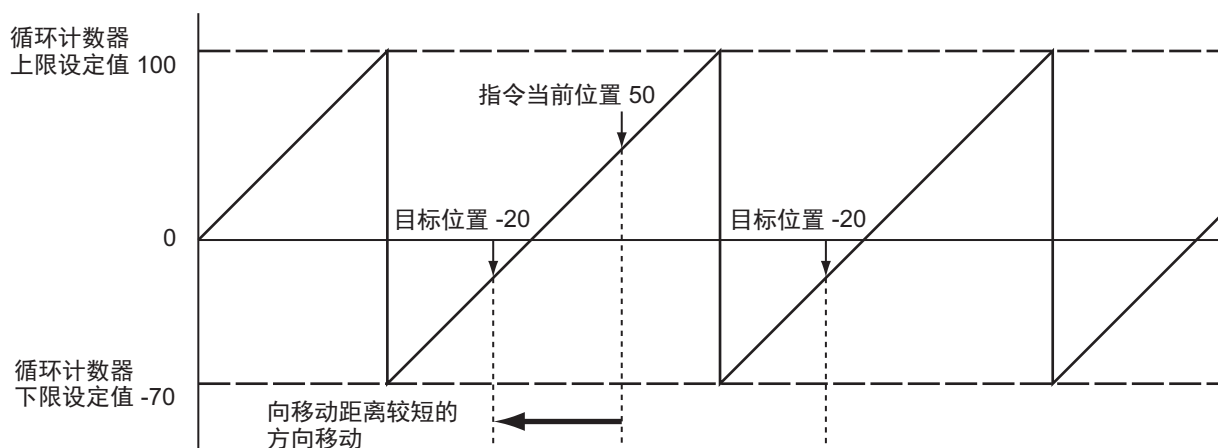
Direction（方向选择）为“无方向指定”时，向循环计数器范围内的目标位置进行定位。因此，将根据指令当前位置和目标位置的大小关系确定移动方向。
指令位置从“50”向“-20”进行定位时的动作示例如下所示。



Direction（方向选择）为“无方向指定”时，可指定为超出循环计数器上下限设定值范围的 Position（目标位置）。
指定为超出循环计数器上限设定值的 Position（目标位置）时，将超出循环计数器上限设定值的移动量作为超出后的相对值进行定位。因此，可进行循环多圈定位。
指定为超出循环计数器下限设定值的 Position（目标位置）时，以相同的方式进行定位。
指令当前位置为“-20”，Position（目标位置）指定为“290”时的动作示例如下。



Direction（方向选择）为“最短距离指定”时，向指令当前位置和目标位置距离较短的方向进行定位。指令位置从“50”向“-20”进行定位时的动作示例如下所示。



正方向的移动距离和负方向的移动距离都相同时，与“当前方向指定”时的动作相同。



使用注意事项

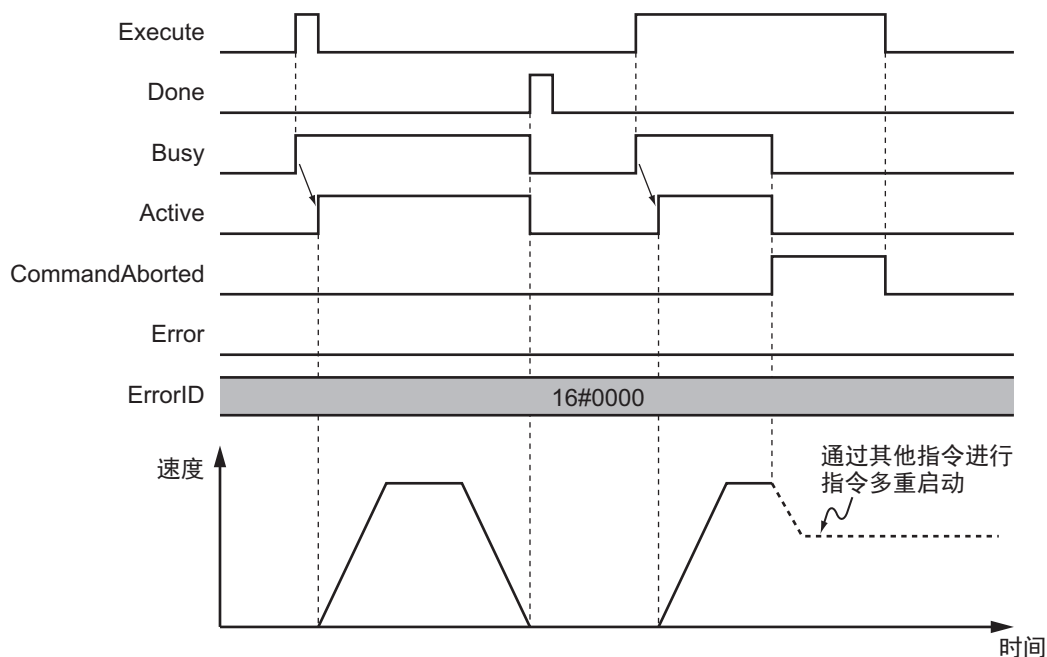
- 进行绝对值定位时，请在小于轴参数 [循环计数器上限设定值] 及大于 [循环计数器下限设定值] 的范围内指定目标位置。
如果指定的目标位置大于 [循环计数器上限设定值] 或小于 [循环计数器下限设定值]，将发生“超出目标位置设定范围（错误代码：5478Hex）”。
但是，Direction（方向选择）为“无方向指定”时，可指定为超出循环计数器上下限设定值范围的 Position（目标位置）。
- Direction（方向选择）选择为“3: _mcCurrentDirection（当前方向指定）”时，将沿用前一动作的指令方向进行动作。因此，因指令组合的不同，可能与前一动作的运动指令输入所指示的方向不一致。
使用“3: _mcCurrentDirection（当前方向指定）”时，请通过轴参数 Dir.Posi（正方向指令中）及 Dir.Nega（负方向指令中）确认当前方向。

● 位置检查

本指令的位置检查根据轴参数 [位置宽度]、[位置检查时间] 的设定执行。

时序图

- 在 Execute (启动) 上升的同时 Busy (执行中) 变为 TRUE, 在下一周期, Active (控制中) 变为 TRUE。
- 到达 Position (目标位置), 完成定位后, Done (完成) 变为 TRUE。
- 通过其他指令中止了本指令时, CommandAborted (执行中断) 变为 TRUE, Busy (执行中)、Active (控制中) 变为 FALSE。



运动指令重新执行

在定位动作中变更输入变量, 并再次将 Execute (启动) 设为 TRUE, 可变更本指令的动作。通过运动指令重新执行可变更的输入变量包括 Position (目标位置)、Velocity (目标速度)、Acceleration (加速度)、Deceleration (减速度)。

运动指令重新执行的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 其他指令执行过程中本指令的启动

对当前正在执行的指令启动本指令后，可切换为本指令或缓冲本指令。

每个轴最多可缓冲 1 个。

在 BufferMode（缓冲模式选择）中指定多重启动时本指令的动作。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 切换指令后如果轴的动作方向发生反转，将按照轴参数 [反转时动作] 执行反转动作。
缓冲	当前正在执行的指令正常完成后，自动启动已缓冲的本指令。
共混	将当前正在执行的指令到达目标位置时的速度（中继速度）作为启动速度，连续执行已缓冲的本指令。变更当前正在执行的指令动作，使其以中继速度到达目标位置。中继速度的指定方法有以下 4 种。
低共混	将当前正在执行的目标速度和缓冲中的目标速度中较低的速度作为中继速度。
前共混	将当前正在执行的目标速度作为中继速度。
后共混	将已缓冲的本指令的目标速度作为中继速度。
高共混	将当前正在执行的目标速度和缓冲中的目标速度中较高的速度作为中继速度。

BufferMode（缓冲模式选择）请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

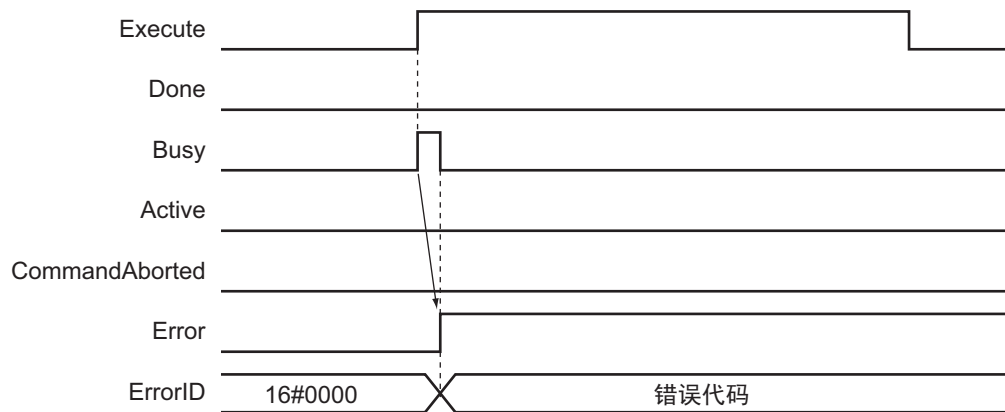
● 本指令执行过程中其他指令的启动

对本指令执行运动指令多重启动时，可进行中断、缓冲、共混。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。
参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

示例程序 1

下面介绍通过反复进行多重启动来定位时的示例程序。

参数设定

下面表示本示例程序中使用的最低限度的设定。

● 设定轴参数

轴种类

轴	轴种类
轴 1	伺服轴

计数器模式

轴	计数器模式
轴 1	直线模式

坐标单位选择

轴	坐标单位
轴 1	mm

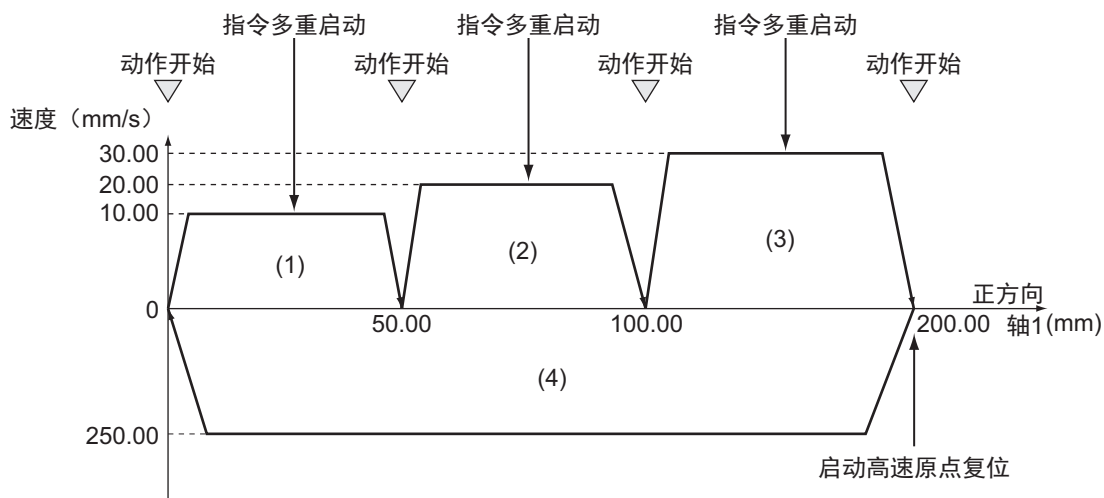
动作示例

将 MC_MoveAbsolute（绝对值定位）指令的 BufferMode（缓冲模式选择）设定为 [缓冲]，在运动指令多重启动的同时，移动到最终目标位置。

到达最终目标位置后，通过 MC_ZeroPosition（高速原点复位）指令返回原点。

进行多重启动时，在前一指令的输出变量“Active（控制中）”为 TRUE 的状态下执行。单轴动作的指令多重启动最多为 1 个指令。

● 动作模式



1 动作开始开关的“ON”

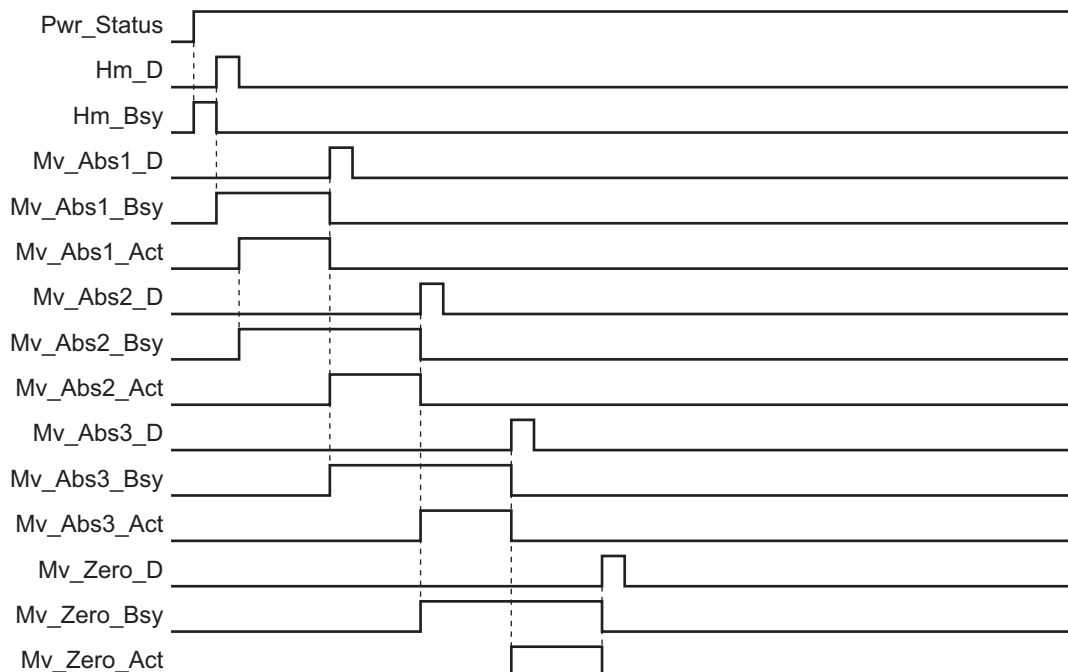
在原点处将动作开始开关设为“ON”后，轴 1 将向正方向进行 50.00mm 定位。

2 接着将动作开始开关设为“ON”

接着每次将动作开始开关设为“ON”，轴 1 将向正方向进行 100.00mm、200.00mm 定位，回到原点后停止。

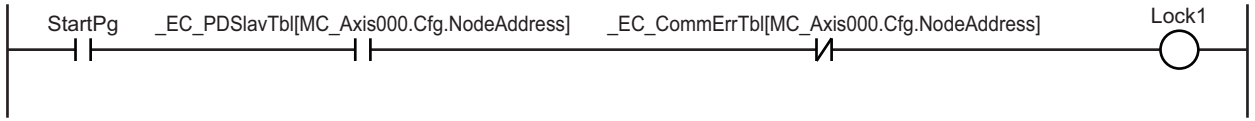
梯形图**● 主要变量**

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
Pwr_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR 的输出变量 Status 中的变量。若变为伺服 ON 状态，该变量变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态，则变为伺服 ON 状态。

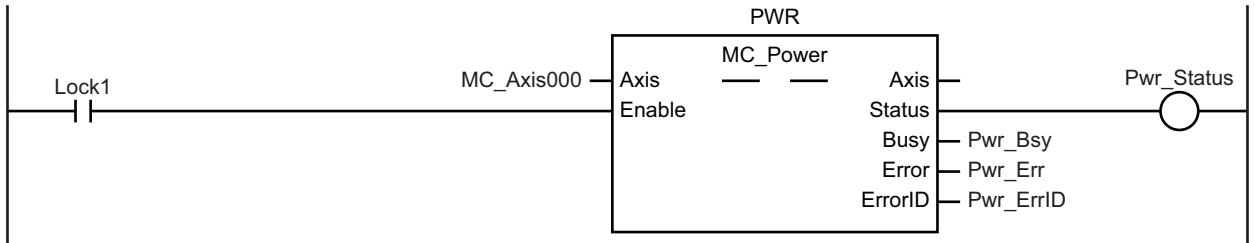
● 时序图

● 示例程序

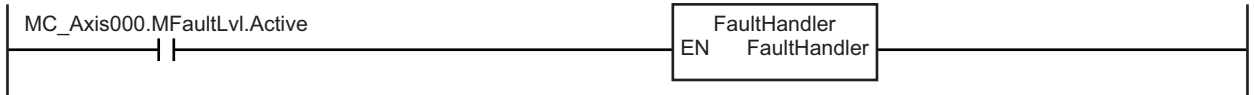
接点 StartPg 变为 TRUE 后，确认 EtherCAT 通信是否正常进行过程数据通信



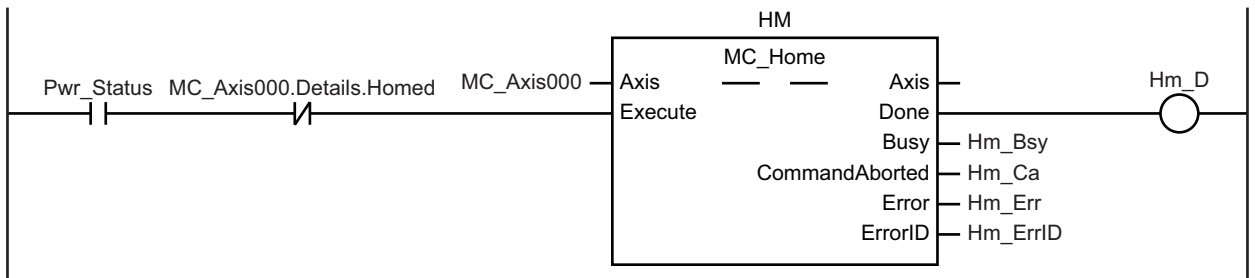
若正在进行过程数据通信，则变为伺服 ON 状态



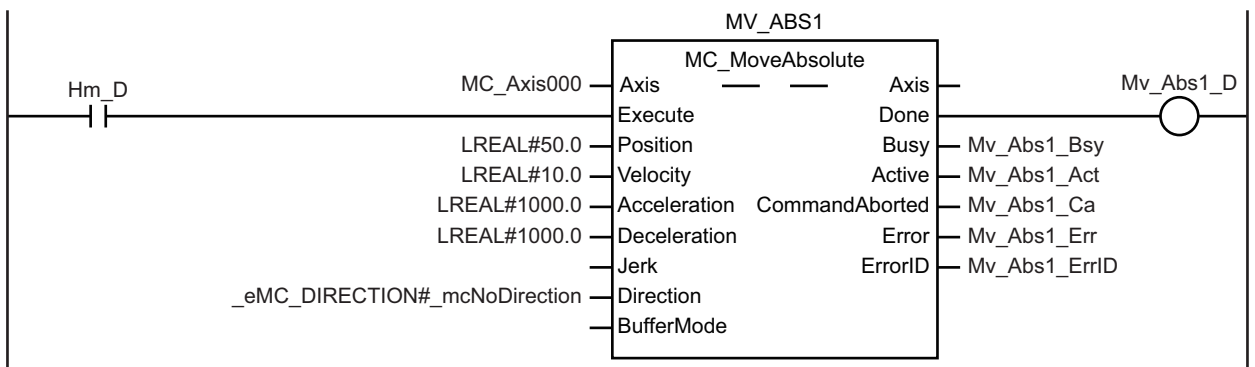
轴 1 发生轻度故障等级的异常时，执行装置发生异常时的处理 (FaultHandler)。发生异常时的处理 (FaultHandler) 请根据装置由客户进行编程



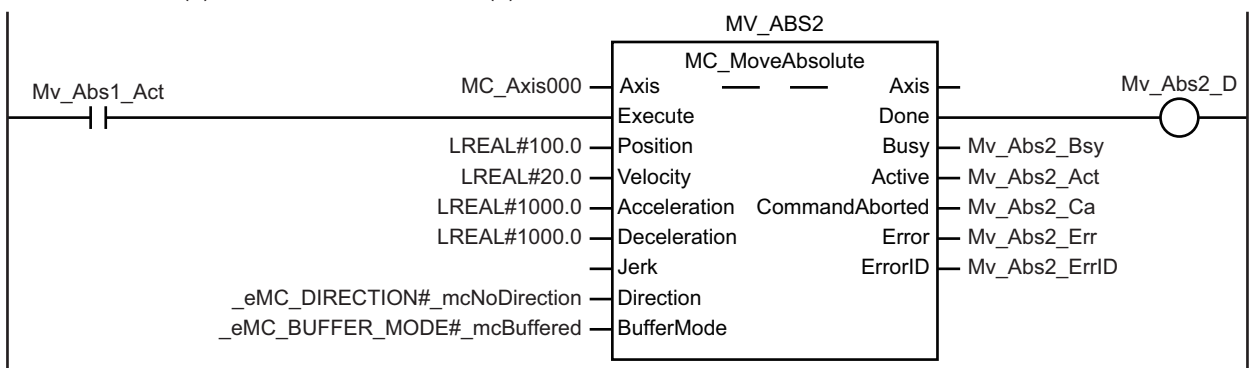
变为伺服 ON 状态且处于原点未确定状态时，执行原点复位



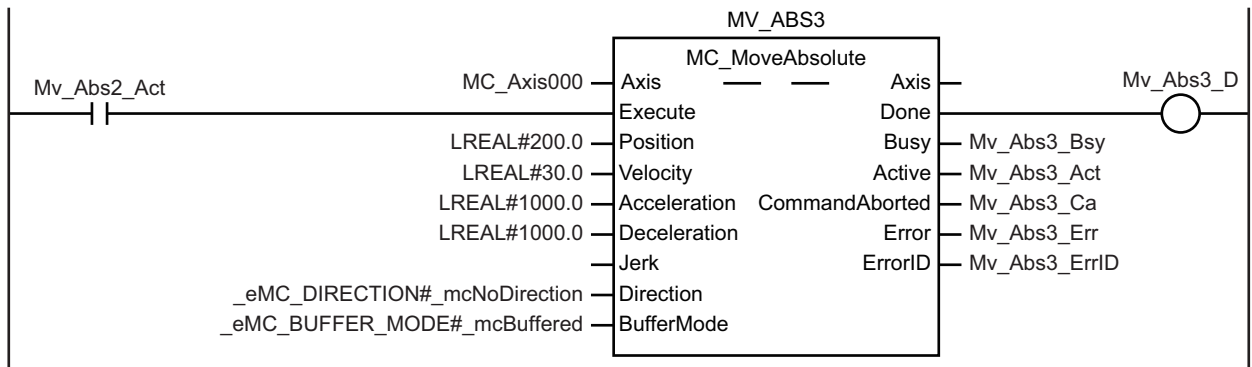
原点确定后，启动绝对值定位 (1)



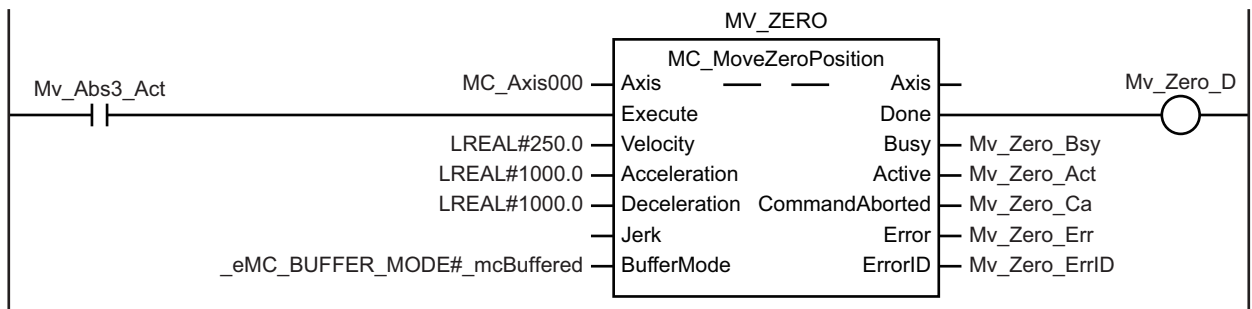
开始绝对值定位 (1) 后，多重启动绝对值定位 (2)



开始绝对值定位 (2) 后, 多重启动绝对值定位 (3)



开始绝对值定位 (3) 后, 执行高速原点复位

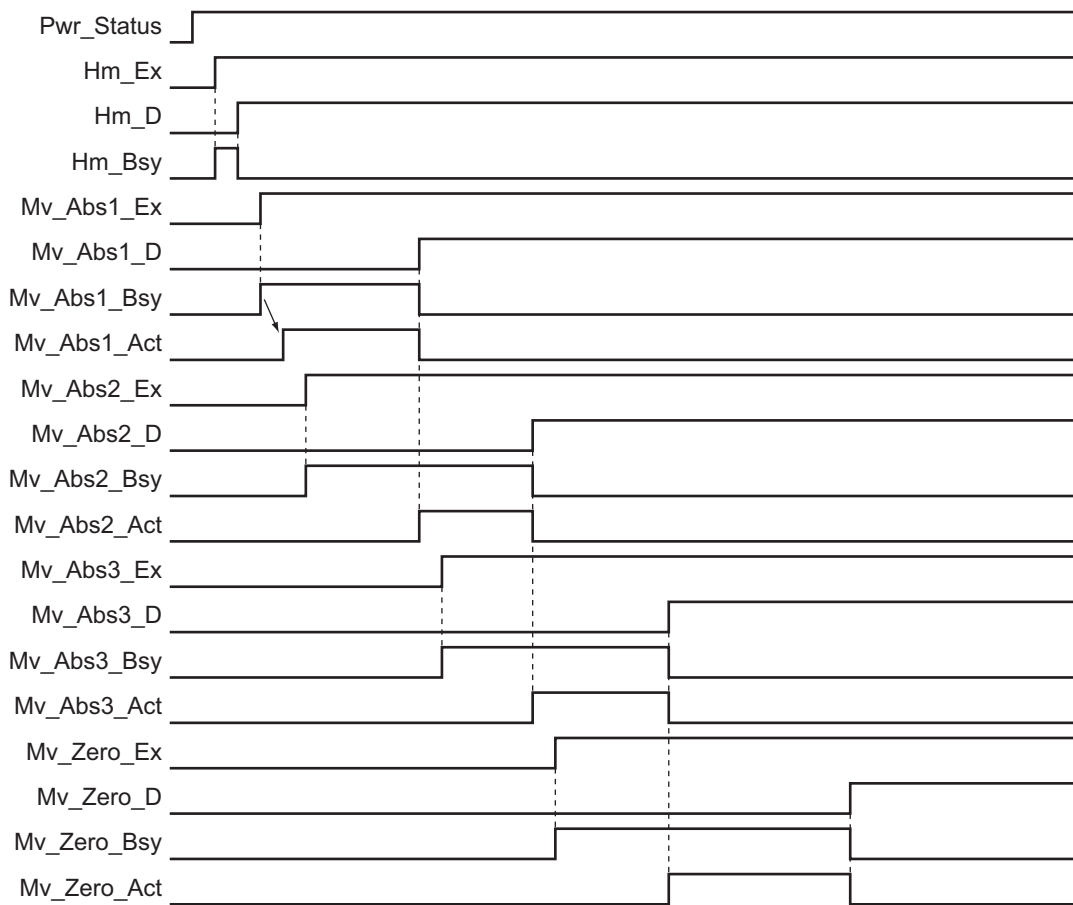


结构文本 (ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时, 变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
Pwr_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR 的输出变量 Status 中的变量。若变为伺服 ON 状态, 该变量变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态, 则变为伺服 ON 状态。
Hm_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_Home 的实例 HM。
Mv_Abs1_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_MoveAbsolute 的实例 MV_ABS1。
Mv_Abs2_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_MoveAbsolute 的实例 MV_ABS2。
Mv_Abs3_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_MoveAbsolute 的实例 MV_ABS3。
Mv_Zero_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_MoveZeroPosition 的实例 MV_ZERO。
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE 时, 设定输入参数。输入参数设定完成后, 变为 TRUE。

● 时序图



● 示例程序

```
// 未设定输入参数时的处理
IF InitFlag = FALSE THEN
```

```
  //MV_ABS1 参数
```

```
  Mv_Abs1_Pos := LREAL#50.0;
  Mv_Abs1_Vel := LREAL#10.0;
  Mv_Abs1_Acc := LREAL#1000.0;
  Mv_Abs1_Dec := LREAL#1000.0;
  Mv_Abs1_Dir := _eMC_DIRECTION#_mcNoDirection;
```

```
  //MV_ABS2 参数
```

```
  Mv_Abs2_Pos := LREAL#100.0;
  Mv_Abs2_Vel := LREAL#20.0;
  Mv_Abs2_Acc := LREAL#1000.0;
  Mv_Abs2_Dec := LREAL#1000.0;
  Mv_Abs2_Dir := _eMC_DIRECTION#_mcNoDirection;
  Mv_Abs2_Bm := _eMC_BUFFER_MODE#_mcBuffered;
```

```
  //MV_ABS3 参数
```

```
  Mv_Abs3_Pos := LREAL#200.0;
  Mv_Abs3_Vel := LREAL#30.0;
```

```

Mv_Abs3_Acc := LREAL#1000.0;
Mv_Abs3_Dec := LREAL#1000.0;
Mv_Abs3_Dir := _eMC_DIRECTION#_mcNoDirection;
Mv_Abs3_Bm := _eMC_BUFFER_MODE#_mcBuffered;

//MV_ZERO 参数
Mv_Zero_Vel := LREAL#250;
Mv_Zero_Acc := LREAL#1000.0;
Mv_Zero_Dec := LREAL#1000.0;
Mv_Zero_Bm := _eMC_BUFFER_MODE#_mcBuffered;

// 设定输入参数后, 将 InitFlag 设为 TRUE
InitFlag:=TRUE;

END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 设为伺服 ON 状态。
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
  Pwr_En:=TRUE;
ELSE
  Pwr_En:=FALSE;
END_IF;

// 发生轻度故障等级异常时的处理
// 发生异常时的处理 (FaultHandler) 请根据装置由客户进行编程。
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
  FaultHandler();
END_IF;

// 在伺服 ON 状态下, 且原点未确定时, 进行原点复位
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
  Hm_Ex:=TRUE;
END_IF;

// 确定原点后, 执行 MV_ABS1
IF Hm_D=TRUE THEN
  Mv_Abs1_Ex:=TRUE;
END_IF;

// 开始 MV_ABS1 后, 多重启动 MV_ABS2
IF Mv_Abs1_Act=TRUE THEN
  Mv_Abs2_Ex:=TRUE;
END_IF;

// 开始 MV_ABS2 后, 多重启动 MV_ABS3
IF Mv_Abs2_Act=TRUE THEN
  Mv_Abs3_Ex:=TRUE;
END_IF;

// 开始 MV_ABS3 后, 多重启动 MV_ZERO
IF Mv_Abs3_Act=TRUE THEN
  Mv_Zero_Ex:=TRUE;
END_IF;

```

```

//MC_Power
PWR(
  Axis      := MC_Axis000,
  Enable    := Pwr_En,
  Status    => Pwr_Status,
  Busy      => Pwr_Bsy,
  Error     => Pwr_Err,
  ErrorID   => Pwr_ErrID
);

//MC_Home
HM(
  Axis          := MC_Axis000,
  Execute       := Hm_Ex,
  Done          => Hm_D,
  Busy          => Hm_Bsy,
  CommandAborted => Hm_Ca,
  Error         => Hm_Err,
  ErrorID       => Hm_ErrID
);

//MC_MoveAbsolute
MV_ABS1(
  Axis          := MC_Axis000,
  Execute       := Mv_Abs1_Ex,
  Position      := Mv_Abs1_Pos,
  Velocity      := Mv_Abs1_Vel,
  Acceleration  := Mv_Abs1_Acc,
  Deceleration  := Mv_Abs1_Dec,
  Direction     := Mv_Abs1_Dir,
  Done          => Mv_Abs1_D,
  Busy          => Mv_Abs1_Bsy,
  Active        => Mv_Abs1_Act,
  CommandAborted => Mv_Abs1_Ca,
  Error         => Mv_Abs1_Err,
  ErrorID       => Mv_Abs1_ErrID
);

MV_ABS2(
  Axis          := MC_Axis000,
  Execute       := Mv_Abs2_Ex,
  Position      := Mv_Abs2_Pos,
  Velocity      := Mv_Abs2_Vel,
  Acceleration  := Mv_Abs2_Acc,
  Deceleration  := Mv_Abs2_Dec,
  Direction     := Mv_Abs2_Dir,
  BufferMode     := Mv_Abs2_Bm,
  Done          => Mv_Abs2_D,
  Busy          => Mv_Abs2_Bsy,
  Active        => Mv_Abs2_Act,
  CommandAborted => Mv_Abs2_Ca,
  Error         => Mv_Abs2_Err,
  ErrorID       => Mv_Abs2_ErrID
);

```

```

MV_ABS3(
  Axis           := MC_Axis000,
  Execute        := Mv_Abs3_Ex,
  Position       := Mv_Abs3_Pos,
  Velocity       := Mv_Abs3_Vel,
  Acceleration   := Mv_Abs3_Acc,
  Deceleration   := Mv_Abs3_Dec,
  Direction      := Mv_Abs3_Dir,
  BufferMode     := Mv_Abs3_Bm,
  Done           => Mv_Abs3_D,
  Busy           => Mv_Abs3_Bsy,
  Active         => Mv_Abs3_Act,
  CommandAborted => Mv_Abs3_Ca,
  Error          => Mv_Abs3_Err,
  ErrorID        => Mv_Abs3_ErrID
);

//MC_MoveZeroPosition
MV_ZERO(
  Axis           := MC_Axis000,
  Execute        := Mv_Zero_Ex,
  Velocity       := Mv_Zero_Vel,
  Acceleration   := Mv_Zero_Acc,
  Deceleration   := Mv_Zero_Dec,
  BufferMode     := Mv_Zero_Bm,
  Done           => Mv_Zero_D,
  Busy           => Mv_Zero_Bsy,
  Active         => Mv_Zero_Act,
  CommandAborted => Mv_Zero_Ca,
  Error          => Mv_Zero_Err,
  ErrorID        => Mv_Zero_ErrID
);

```

示例程序 2

计数模式为旋转模式时，为了定位到目标位置，将自动判断左右方向哪个较近，然后进行定位。下面介绍工具更换的最短距离控制时的示例程序。

参数设定

下面表示本示例程序中使用的最低限度的设定。

● 设定轴参数

轴种类

轴	轴种类
轴 1	伺服轴

计数器模式

轴	计数器模式
轴 1	旋转模式

循环计数器

轴	上限值	下限值
轴 1	360	0

坐标单位选择

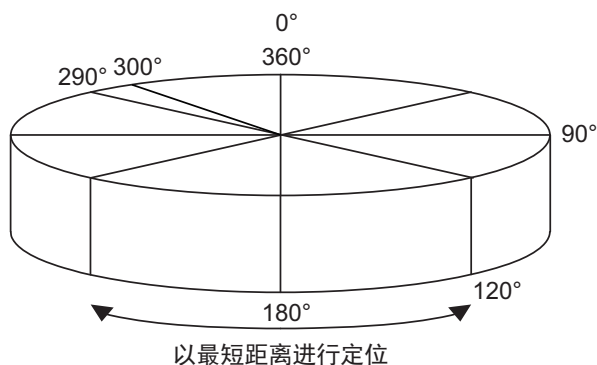
轴	坐标单位
轴 1	degree

动作示例

利用绝对值定位指令的多重启动，在 $0 \sim 360^\circ$ 的范围内的动作示例。此外，如果当前位置超出 $0 \sim 360^\circ$ 的范围，将恢复为 0° 。

进行多重启动时，在前一指令的输出变量“Active（控制中）”为 TRUE 的状态下执行。单轴动作的指令多重启动最多为 1 个指令。

本示例中，将 BufferMode（缓冲模式）设为缓冲后进行多重启动。



指定为原点 (0°)、 90° 、 120° 、 290° 中的任意位置后，移动到该位置。

此时的旋转方向为向较近的方向旋转。此外，移动速度为 $250^\circ/\text{s}$ 。

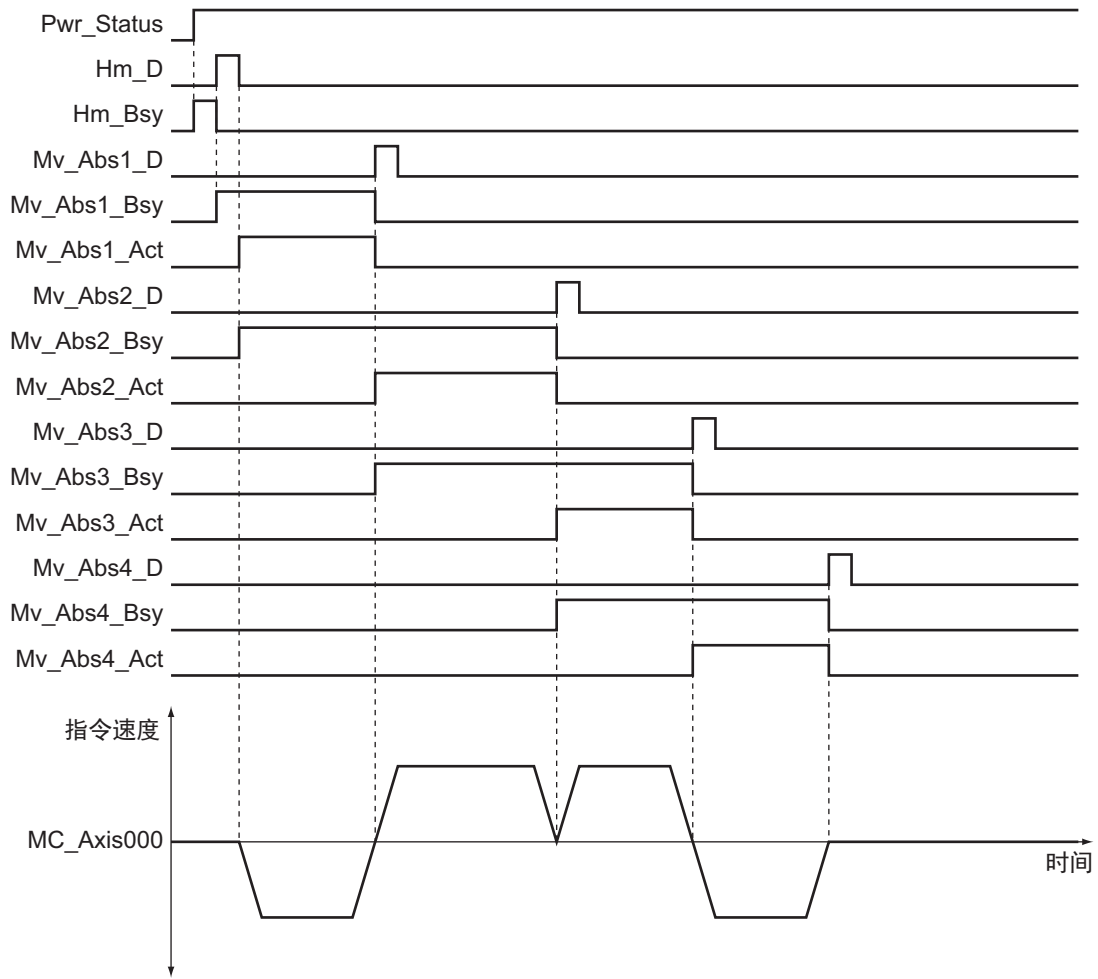
在示例程序中， $290^\circ \rightarrow 90^\circ \rightarrow 120^\circ \rightarrow$ 原点 (0°) 按最短距离指定进行定位。

梯形图

● 主要变量

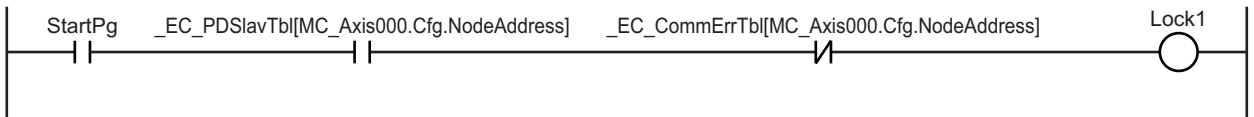
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
Pwr_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR 的输出变量 Status 中的变量。若变为伺服 ON 状态，该变量变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态，则变为伺服 ON 状态。

● 时序图

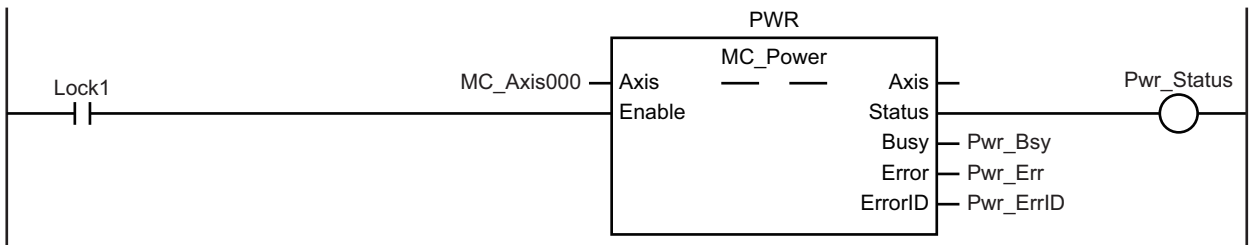


● 示例程序

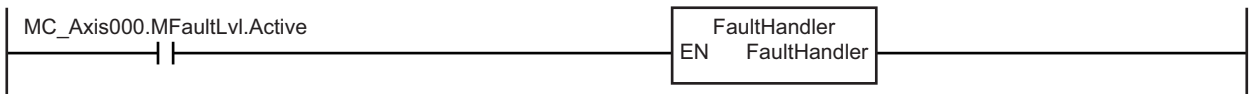
接点 StartPg 变为 TRUE 后，确认 EtherCAT 通信是否正常进行过程数据通信



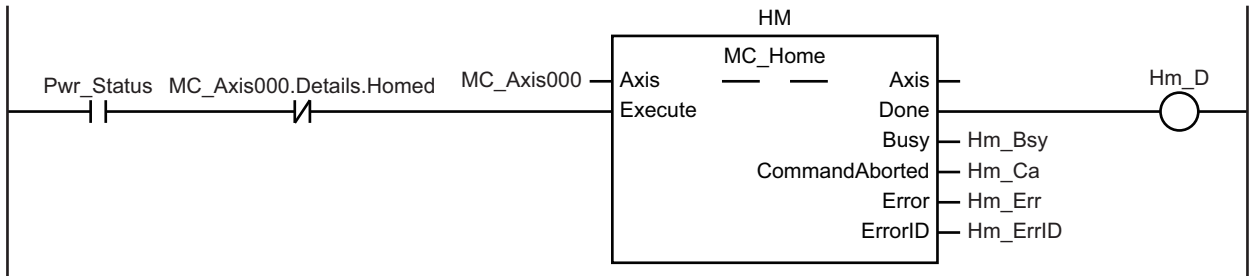
若正在进行过程数据通信，则变为伺服 ON 状态



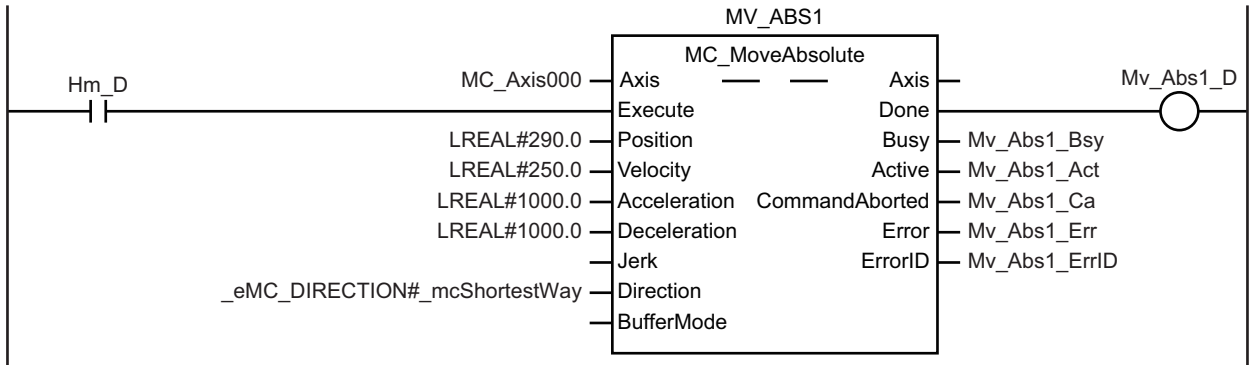
轴 1 发生轻度故障等级的异常时，执行装置发生异常时的处理（FaultHandler）。发生异常时的处理（FaultHandler）请根据装置由客户进行编程



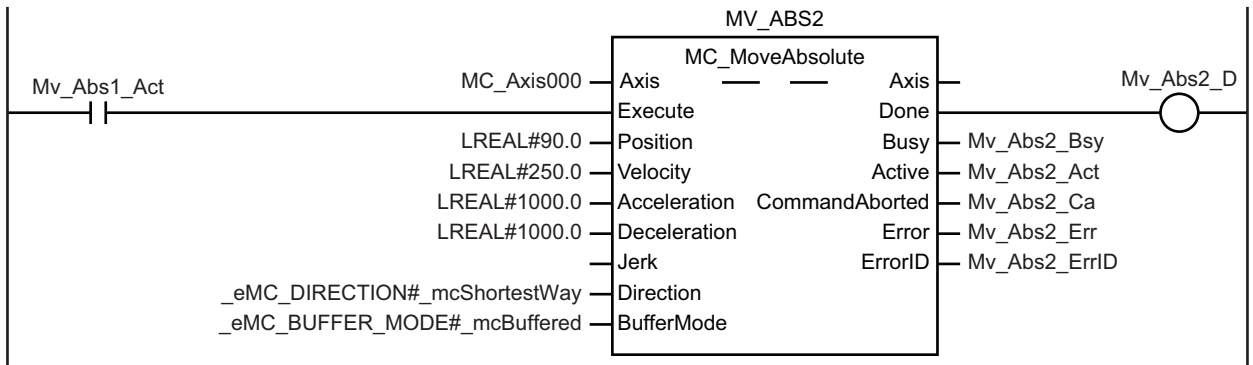
变为伺服 ON 状态且处于原点未确定状态时，执行原点复位



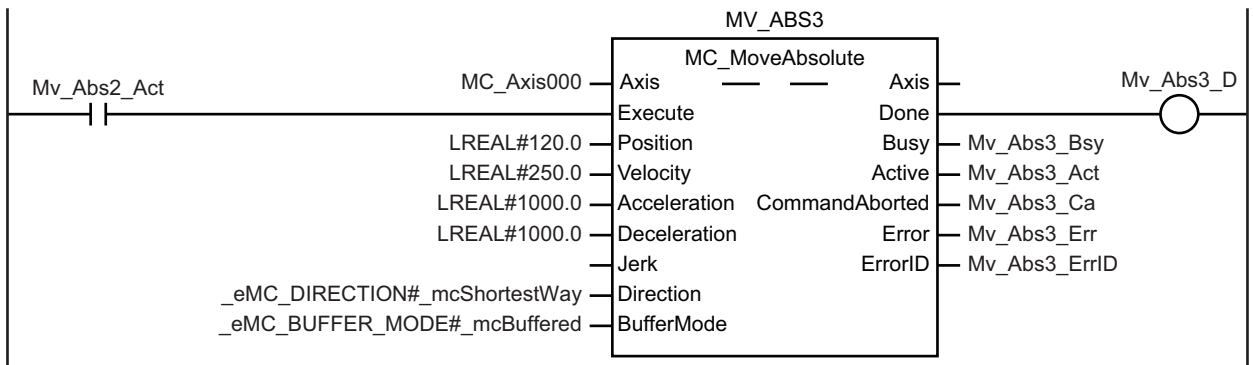
原点确定后，执行绝对值定位 (1)，向 290.0° 移动。动作方向为最短距离指定



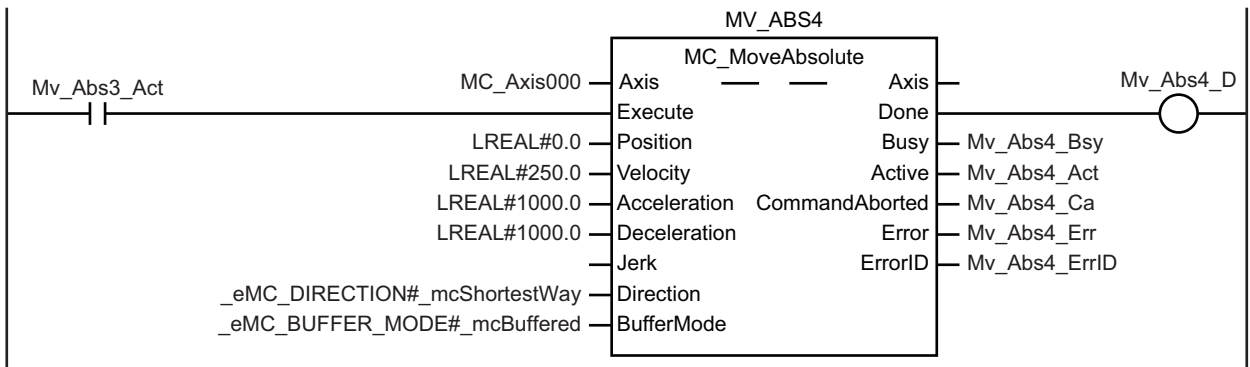
多重启动绝对值定位 (2)，从 290.0°向 90.0°移动。动作方向为最短距离



多重启动绝对值定位 (3)，从 90.0° 向 120.0° 移动。动作方向为最短距离



多重启动绝对值定位 (4)，从 120.0°向 0.0°移动。动作方向为最短距离

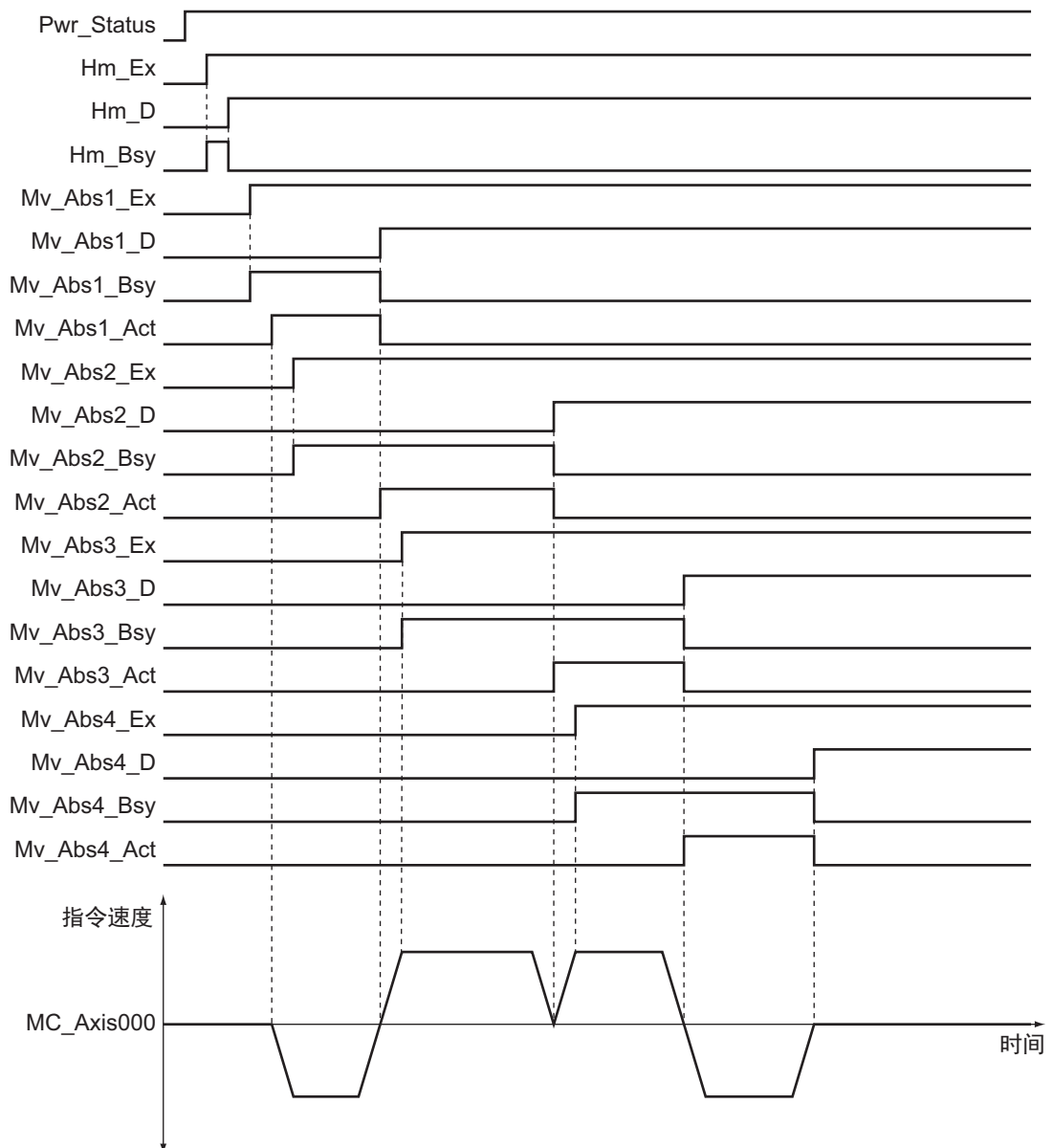


结构文本 (ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
Pwr_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR 的输出变量 Status 中的变量。若变为伺服 ON 状态，该变量变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态，则变为伺服 ON 状态。
Hm_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时，启动 MC_Home 的实例 HM。
Mv_Abs1_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时，启动 MC_MoveAbsolute 的实例 MV_ABS1。
Mv_Abs2_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时，启动 MC_MoveAbsolute 的实例 MV_ABS2。
Mv_Abs3_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时，启动 MC_MoveAbsolute 的实例 MV_ABS3。
Mv_Abs4_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时，启动 MC_MoveAbsolute 的实例 MV_ABS4。
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE 时，设定输入参数。输入参数设定完成后，变为 TRUE。

● 时序图



● 示例程序

```
// 未设定输入参数时的处理
IF InitFlag = FALSE THEN

  //MV_ABS1 参数
  Mv_Abs1_Pos := LREAL#290.0;
  Mv_Abs1_Vel := LREAL#250.0;
  Mv_Abs1_Acc := LREAL#1000.0;
  Mv_Abs1_Dec := LREAL#1000.0;
  Mv_Abs1_Dir := _eMC_DIRECTION#_mcShortestWay;

  //MV_ABS2 参数
  Mv_Abs2_Pos := LREAL#90.0;
  Mv_Abs2_Vel := LREAL#250.0;
  Mv_Abs2_Acc := LREAL#1000.0;
  Mv_Abs2_Dec := LREAL#1000.0;
  Mv_Abs2_Dir := _eMC_DIRECTION#_mcShortestWay;
  Mv_Abs2_Bm := _eMC_BUFFER_MODE#_mcBuffered;
```

```

//MV_ABS3 参数
Mv_Abs3_Pos := LREAL#120.0;
Mv_Abs3_Vel := LREAL#250.0;
Mv_Abs3_Acc := LREAL#1000.0;
Mv_Abs3_Dec := LREAL#1000.0;
Mv_Abs3_Dir := _eMC_DIRECTION#_mcShortestWay;
Mv_Abs3_Bm := _eMC_BUFFER_MODE#_mcBuffered;

//MV_ABS4 参数
Mv_Abs4_Pos := LREAL#0.0;
Mv_Abs4_Vel := LREAL#250.0;
Mv_Abs4_Acc := LREAL#1000.0;
Mv_Abs4_Dec := LREAL#1000.0;
Mv_Abs4_Dir := _eMC_DIRECTION#_mcShortestWay;
Mv_Abs4_Bm := _eMC_BUFFER_MODE#_mcBuffered;

// 设定输入参数后，将 InitFlag 设为 TRUE
InitFlag := TRUE;

END_IF;

//StartPg 为 TRUE 时，确认 EtherCAT 通信正常通信，
// 设为伺服 ON 状态。
//EtherCAT 通信未正常通信时，设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbI[MC_Axis000.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbI[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
  Pwr_En:=TRUE;
ELSE
  Pwr_En:=FALSE;
END_IF;

// 发生轻度故障等级异常时的处理
// 发生异常时的处理 (FaultHandler) 请根据装置由客户进行编程。
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
  FaultHandler();
END_IF;

// 在伺服 ON 状态下，且原点未确定时，进行原点复位
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
  Hm_Ex:=TRUE;
END_IF;

// 确定原点后，执行 MV_ABS1
IF Hm_D=TRUE THEN
  Mv_Abs1_Ex:=TRUE;
END_IF;

// 开始 MV_ABS1 后，多重启动 MV_ABS2
IF Mv_Abs1_Act=TRUE THEN
  Mv_Abs2_Ex:=TRUE;
END_IF;

// 开始 MV_ABS2 后，多重启动 MV_ABS3
IF Mv_Abs2_Act=TRUE THEN
  Mv_Abs3_Ex:=TRUE;
END_IF;

```

```

// 开始 MV_ABS3 后, 多重启动 MV_ABS4
IF Mv_Abs3_Act=TRUE THEN
  Mv_Abs4_Ex:=TRUE;
END_IF;

// MC_Power
PWR(
  Axis      := MC_Axis000,
  Enable    := Pwr_En,
  Status    => Pwr_Status,
  Busy      => Pwr_Bsy,
  Error     => Pwr_Err,
  ErrorID   => Pwr_ErrID
);

// MC_Home
HM(
  Axis          := MC_Axis000,
  Execute       := Hm_Ex,
  Done          => Hm_D,
  Busy          => Hm_Bsy,
  CommandAborted => Hm_Ca,
  Error         => Hm_Err,
  ErrorID       => Hm_ErrID
);

// 绝对值定位 (1)
MV_ABS1(
  Axis          := MC_Axis000,
  Execute       := Mv_Abs1_Ex,
  Position      := Mv_Abs1_Pos,
  Velocity      := Mv_Abs1_Vel,
  Acceleration  := Mv_Abs1_Acc,
  Deceleration  := Mv_Abs1_Dec,
  Direction     := Mv_Abs1_Dir,
  Done          => Mv_Abs1_D,
  Busy          => Mv_Abs1_Bsy,
  Active        => Mv_Abs1_Act,
  CommandAborted => Mv_Abs1_Ca,
  Error         => Mv_Abs1_Err,
  ErrorID       => Mv_Abs1_ErrID
);

// 绝对值定位 (2)
MV_ABS2(
  Axis          := MC_Axis000,
  Execute       := Mv_Abs2_Ex,
  Position      := Mv_Abs2_Pos,
  Velocity      := Mv_Abs2_Vel,
  Acceleration  := Mv_Abs2_Acc,
  Deceleration  := Mv_Abs2_Dec,
  Direction     := Mv_Abs2_Dir,
  BufferMode     := Mv_Abs2_Bm,
  Done          => Mv_Abs2_D,
  Busy          => Mv_Abs2_Bsy,
  Active        => Mv_Abs2_Act,
  CommandAborted => Mv_Abs2_Ca,

```

```

    Error          => Mv_Abs2_Err,
    ErrorID        => Mv_Abs2_ErrID
);

```

```
// 绝对值定位 (3)
```

```

MV_ABS3(
    Axis           := MC_Axis000,
    Execute        := Mv_Abs3_Ex,
    Position       := Mv_Abs3_Pos,
    Velocity       := Mv_Abs3_Vel,
    Acceleration   := Mv_Abs3_Acc,
    Deceleration   := Mv_Abs3_Dec,
    Direction      := Mv_Abs3_Dir,
    BufferMode      := Mv_Abs3_Bm,
    Done           => Mv_Abs3_D,
    Busy           => Mv_Abs3_Bsy,
    Active         => Mv_Abs3_Act,
    CommandAborted => Mv_Abs3_Ca,
    Error          => Mv_Abs3_Err,
    ErrorID        => Mv_Abs3_ErrID
);

```

```
// 绝对值定位 (4)
```

```

MV_ABS4(
    Axis           := MC_Axis000,
    Execute        := Mv_Abs4_Ex,
    Position       := Mv_Abs4_Pos,
    Velocity       := Mv_Abs4_Vel,
    Acceleration   := Mv_Abs4_Acc,
    Deceleration   := Mv_Abs4_Dec,
    Direction      := Mv_Abs4_Dir,
    BufferMode      := Mv_Abs4_Bm,
    Done           => Mv_Abs4_D,
    Busy           => Mv_Abs4_Bsy,
    Active         => Mv_Abs4_Act,
    CommandAborted => Mv_Abs4_Ca,
    Error          => Mv_Abs4_Err,
    ErrorID        => Mv_Abs4_ErrID
);

```

MC_MoveRelative

指定指令当前位置起的移动距离，进行定位。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_MoveRelative	相对值 定位	FB		<pre>MC_MoveRelative_instance (Axis := 《参数》, Execute := 《参数》, Distance := 《参数》, Velocity := 《参数》, Acceleration := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, BufferMode := 《参数》, Done => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Distance	移动距离	LREAL	负数、正数、“0”	0	指定指令当前位置起的移动距离。 单位为 [指令单位]。*1
Velocity	目标速度	LREAL	正数	0	指定目标速度。*2 单位为 [指令单位 /s]。*1
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度。 单位为 [指令单位 /s ²]。*1
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。 单位为 [指令单位 /s ²]。*1
Jerk	跃度	LREAL	正数或“0”	0	指定跃度。 单位为 [指令单位 /s ³]。*1
BufferMode	缓冲 模式选择	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0*3	指定运动指令多重启动时的动作。 0: 中断 1: 缓冲 2: 低共混 3: 前共混 4: 后共混 5: 高共混

*1. 指令单位请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. 请务必设定目标速度。若不设定而直接动作，将发生异常。

*3. 如果是有效范围为枚举型的变量初始值，实际不是数值，而是枚举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*1	发生异常时，输出错误代码。16#0000 表示正常。

*1. □□「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	定位完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	开始了指令时	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动（中断），本指令中止时 因发生异常，本指令中止时 正在发生异常时，启动了本指令 正在执行 MC_Stop 指令时，启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

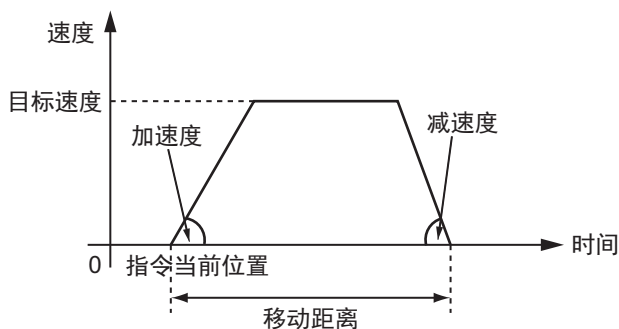
输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。*1

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis***”）或系统定义变量的轴变量名称（_MC_AX[*]、_MC1_AX[*]、_MC2_AX[*]）。

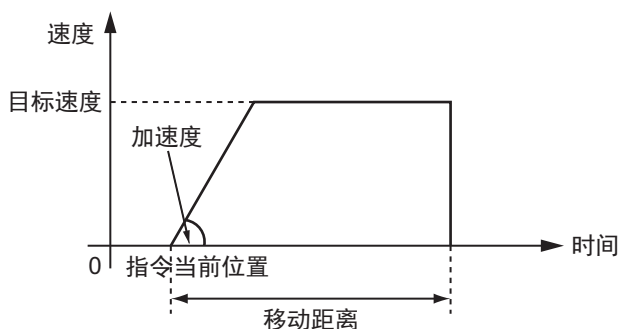
功能说明

- 指定指令当前位置起的移动距离，进行定位。
- 在 Execute（启动）的上升沿开始相对定位动作。
- 输入变量中可指定 Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）、Jerk（跃度）。

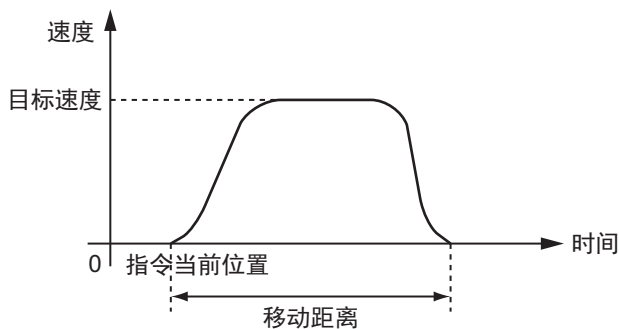
相对定位的动作示例如下所示。



将加速度或减速度指定为“0”并启动时，将不进行加减速而达到目标速度。减速度为“0”时的动作示例如下。



希望均匀加减速时，需指定 Jerk（跃度）。指定了 Jerk（跃度）时的动作示例如下。

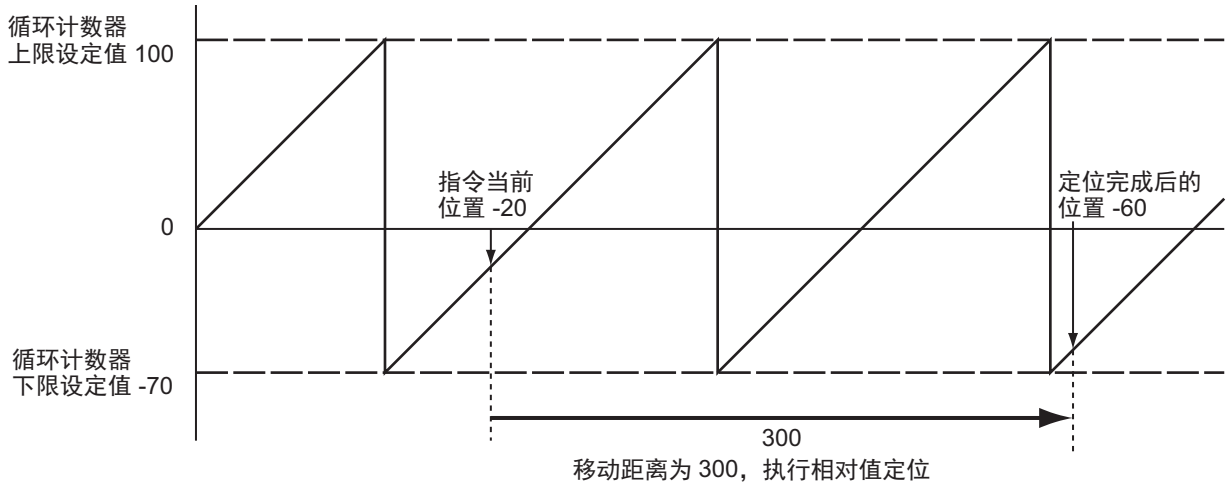


Jerk（跃度）的详情请参考 [□](#) 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

指令的详情

计数模式为 [旋转模式] 时，可指定为超出轴参数 [循环计数器下限设定值] 到 [循环计数器上限设定值] 相对距离范围的 Distance（移动距离），进行循环多圈定位。

指令当前位置为“-20”，Distance（移动距离）指定为“300”时的动作示例如下。



将 Distance（移动距离）指定为“0”并启动时，轴不会移动，但 Done（完成）变为 TRUE。



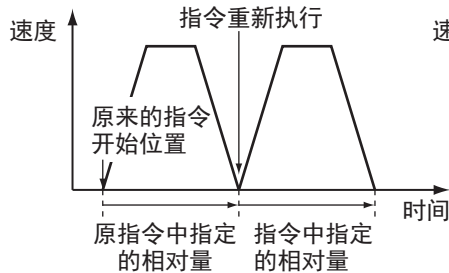
使用注意事项

完成定位前重新启动相对值定位时，请注意以下几点。

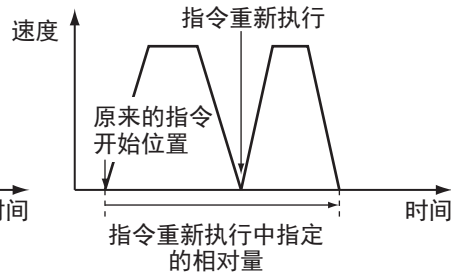
如果在重新启动 MC 功能模块前完成定位，则为正常的指令启动。

- 正常的指令启动时，按照以指令启动时的位置为基准的相对位置进行定位。
- 指令重新执行时，按照以原指令启动时的位置为基准的相对位置进行定位。

通常的指令启动



指令重新执行

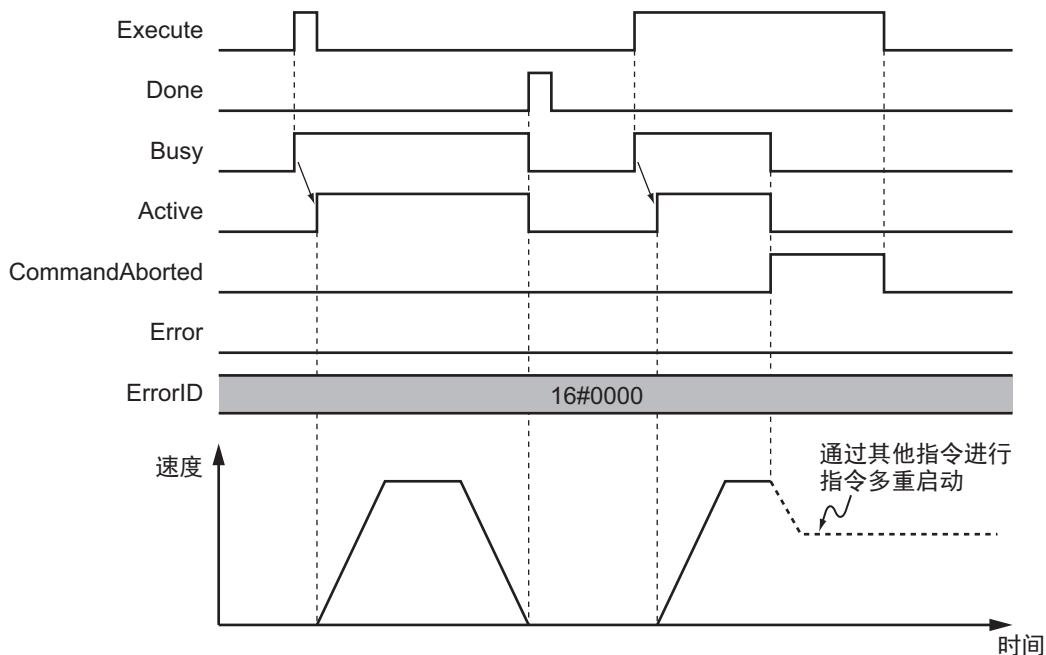


● 位置检查

本指令的位置检查根据轴参数 [位置宽度]、[位置检查时间] 的设定执行。

时序图

- 在 Execute (启动) 上升的同时 Busy (执行中) 变为 TRUE, 在下一周期, Active (控制中) 变为 TRUE。
- 达到 Distance (移动距离), 完成定位后, Done (完成) 变为 TRUE。
- 通过其他指令中止了本指令时, CommandAborted (执行中断) 变为 TRUE, Busy (执行中)、Active (控制中) 变为 FALSE。



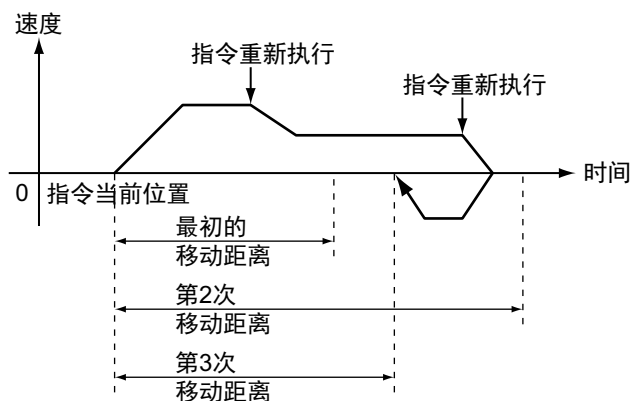
运动指令重新执行

在定位动作中变更输入变量, 并再次将 Execute (启动) 设为 TRUE, 可变更本指令的动作。

通过运动指令重新执行可变更的输入变量包括 Distance (移动距离)、Velocity (目标速度)、Acceleration (加速度)、Deceleration (减速度)。

重新执行时 Distance (移动距离) 的起点不是重新执行时的指令当前位置, 而是第一次启动本指令时的指令当前位置。

在变更 Distance (移动距离) 和 Velocity (目标速度) 的同时, 重新执行运动指令 2 次时的动作示例如下。

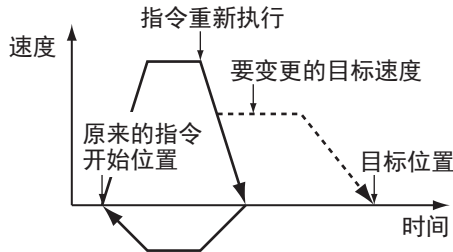




使用注意事项

即使要变更 Distance（移动距离）以外的输入参数，Distance（移动距离）中仍代入与原指令相同的值，然后重新执行指令。

例如，将 Distance（移动距离）的值变更为“0”并重新执行指令时，从第一次启动起的 Distance（移动距离）视为“0”，再重新执行指令。因此，如下图所示，将返回到原来的开始位置。



运动指令重新执行的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 其他指令执行过程中本指令的启动

对当前正在执行的指令启动本指令后，可切换为本指令或缓冲本指令。

每个轴最多可缓冲 1 个。

运动指令多重启动时 Distance（移动距离）的起点为本指令启动、Active（控制中）变为 TRUE 时的指令当前位置。

在 BufferMode（缓冲模式选择）中指定多重启动时本指令的动作。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 切换指令后如果轴的动作方向发生反转，将按照轴参数 [反转时动作] 执行反转动作。
缓冲	当前正在执行的指令正常完成后，自动启动已缓冲的本指令。
共混	将当前正在执行的指令到达目标位置时的速度（中继速度）作为启动速度，连续执行已缓冲的本指令。变更当前正在执行的指令动作，使其以中继速度到达目标位置。中继速度的指定方法有以下 4 种。
低共混	将当前正在执行的目标速度和缓冲中的目标速度中较低的速度作为中继速度。
前共混	将当前正在执行的目标速度作为中继速度。
后共混	将已缓冲的本指令的目标速度作为中继速度。
高共混	将当前正在执行的目标速度和缓冲中的目标速度中较高的速度作为中继速度。

BufferMode（缓冲模式选择）请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

- 本指令执行过程中其他指令的启动

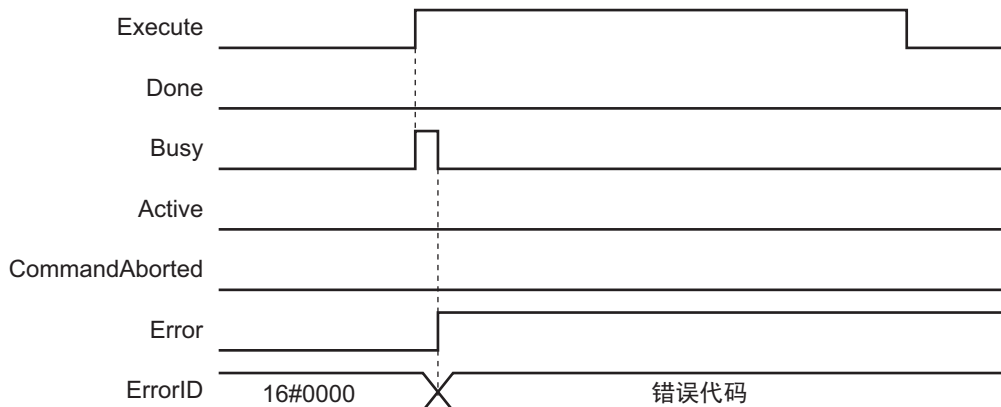
对本指令执行运动指令多重启动时，可进行中断、缓冲、共混。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

- 发生异常时的时序图



- 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

MC_MoveVelocity

使用伺服驱动器的位置控制模式，进行模拟速度控制。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_MoveVelocity	速度控制	FB		<pre>MC_MoveVelocity_instance (Axis := 《参数》, Execute := 《参数》, Velocity := 《参数》, Acceleration := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, Direction := 《参数》, Continuous := 《参数》, BufferMode := 《参数》, InVelocity => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Velocity	目标速度	LREAL	正数或“0”	0	指定目标速度。 单位为 [指令单位 /s]。*1
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度。 单位为 [指令单位 /s ²]。*1
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。 单位为 [指令单位 /s ²]。*1
Jerk	跃度	LREAL	正数或“0”	0	指定跃度。 单位为 [指令单位 /s ³]。*1
Direction	方向选择	_eMC_ DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection 3: _mcCurrentDirection	0 *2	指定旋转方向。 0: 正方向指定 2: 负方向指定 3: 当前方向指定
BufferMode	缓冲 模式选择	_eMC_BUFFER_ MODE	0: _mcAborting 1: _mcBuffered	0 *2	指定运动指令多重启动时的动作。 0: 中断 1: 缓冲
Continuous (Reserved)	持续方向 选择	BOOL	TRUE、FALSE	FALSE	(Reserved)

*1. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》的“单位转换设定”。

*2. 如果是有效范围为枚举型的变量初始值，实际不是数值，而是枚举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
InVelocity	达到目标速度	BOOL	TRUE、FALSE	达到目标速度时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* □ □ 「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
InVelocity	达到目标速度时	<ul style="list-style-type: none"> • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时 • 指令重新执行时变更了目标速度时
Busy	Execute 的上升沿	<ul style="list-style-type: none"> • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时
Active	开始了指令时	<ul style="list-style-type: none"> • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> • 由其他指令进行运动指令多重启动（中断、缓冲），本指令中止时 • 因发生异常，本指令中止时 • 正在发生异常时，启动了本指令 • 正在执行 MC_Stop 指令时，启动了本指令 	<ul style="list-style-type: none"> • Execute 为 TRUE 时，与 Execute 的 FALSE 同时 • Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。*1

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

功能说明

- 使用位置控制进行模拟速度控制。
- 在 Execute（启动）的上升沿开始速度控制的动作。

指令的详情

下面介绍指令的详情。

● Direction（方向选择）

- 在 Direction（方向选择）中指定移动方向。
- Direction（方向选择）为“正方向指定”时，向正方向移动，“负方向指定”时，向负方向移动。
- Direction（方向选择）为“当前方向指定”时，动作因轴是否处于停止状态而不同。轴停止时，轴向上次的移动方向移动。接通电源时或重启时，向正方向移动。
运动指令多重启动时，若在轴正在移动时启动了本指令，将向当前移动的方向移动。



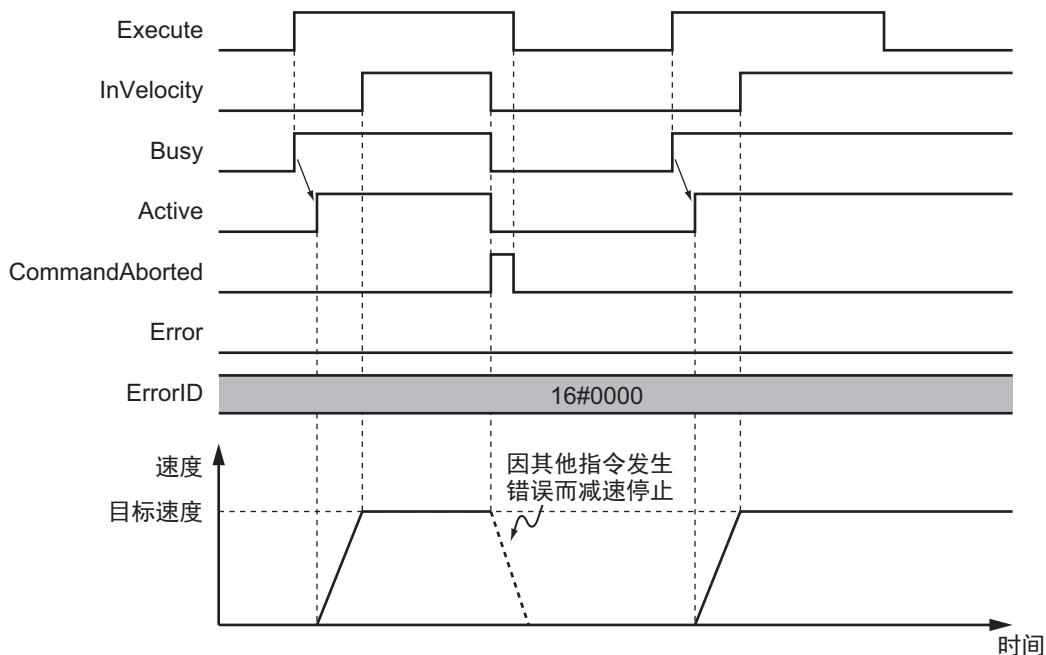
使用注意事项

Direction（方向选择）选择为“3: _mcCurrentDirection（当前方向指定）”时，将沿用前一动作的指令方向进行动作。因此，因指令组合的不同，可能与前一动作的运动指令输入所指示的方向不一致。

使用“3: _mcCurrentDirection（当前方向指定）”时，请通过轴参数 Dir.Posi（正方向指令中）及 Dir.Nega（负方向指令中）确认当前方向。

时序图

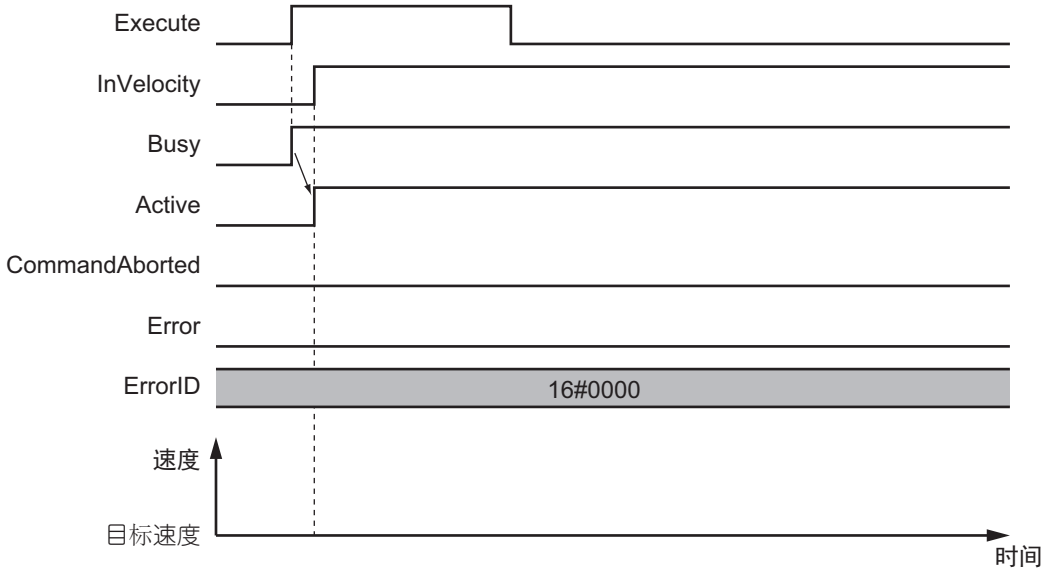
- 在 Execute (启动) 上升的同时 Busy (执行中) 变为 TRUE, 在下一周期, Active (控制中) 变为 TRUE。
- 达到 Velocity (目标速度) 后, InVelocity (达到目标速度) 变为 TRUE。
- 通过其他指令中止了本指令时, CommandAborted (执行中断) 变为 TRUE, Busy (执行中)、Active (控制中)、InVelocity (达到目标速度) 变为 FALSE。



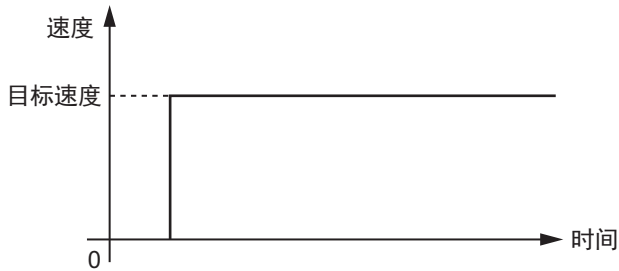
InVelocity (达到目标速度) 表示已达到与本指令的启动及运动指令重新执行相等的速度。因此, InVelocity (达到目标速度) 变为 TRUE 后, 即使因超驰而变更了速度, InVelocity (达到目标速度) 也不会变为 FALSE。

此外, 如果在 InVelocity (达到目标速度) 变为 TRUE 前变更了超驰, InVelocity (达到目标速度) 将在达到变更后的目标速度时变为 TRUE。

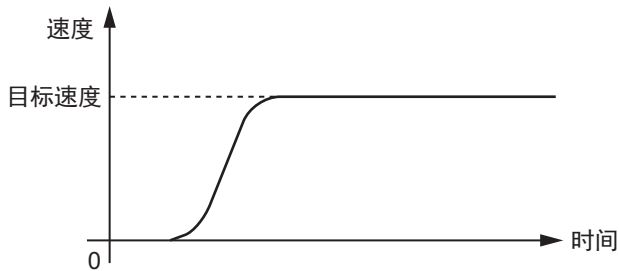
输入变量中可指定 Acceleration（加速度）、Deceleration（减速度）、Jerk（跃度）。
 将 Velocity（目标速度）指定为“0”并启动时，轴不会移动，变为连续动作中。
 将 Velocity（目标速度）设为“0”并启动时的动作示例如下。



将加速度或减速度指定为“0”并启动时，将不进行加减速而达到目标速度。
 加速度为“0”时的动作示例如下。



希望均匀加减速时，需指定 Jerk（跃度）。
 指定了 Jerk（跃度）时的动作示例如下。



Jerk（跃度）的详情请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

运动指令重新执行

在连续动作中变更输入参数，并再次将 Execute（启动）设为 TRUE，可变更本指令的动作。

通过运动指令重新执行可变更的输入变量包括 Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）。

通过运动指令的重新执行变更 Velocity（目标速度）时，InVelocity（达到目标速度）将根据重新执行后设定的新目标速度进行动作。

运动指令重新执行的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 其他指令执行过程中本指令的启动

对当前正在执行的指令启动本指令后，可切换为本指令或缓冲本指令。

每个轴最多可缓冲 1 个。

在 BufferMode（缓冲模式选择）中指定多重启动时本指令的动作。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 切换指令后如果轴的动作方向发生反转，将按照轴参数 [反转时动作] 执行反转动作。
缓冲	当前正在执行的指令正常完成后，自动启动已缓冲的本指令。

BufferMode（缓冲模式选择）请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 本指令执行过程中其他指令的启动

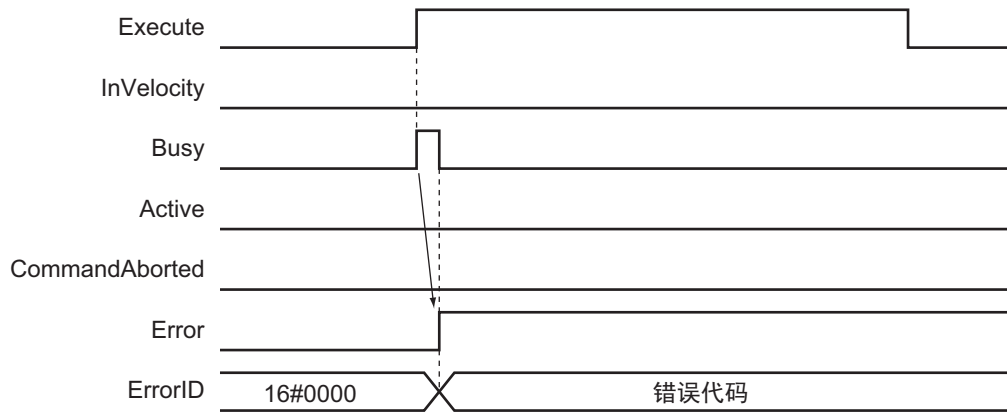
在本指令执行过程中，如需通过其他指令执行运动指令多重启动，需要将其他指令的 BufferMode（缓冲模式选择）指定为中断和缓冲。

指定为缓冲时，本指令的输出变量“InVelocity（达到目标速度）”变为 TRUE 后，多重启动的指令开始动作。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。
参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

示例程序

下面介绍微调器或离心分离机等速度控制相关的示例程序。

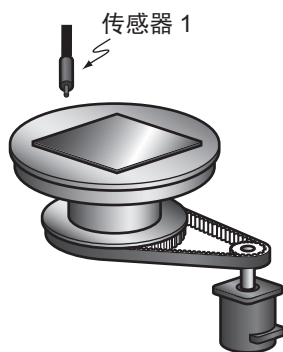
参数设定

下面表示本示例程序中使用的最低限度的设定。

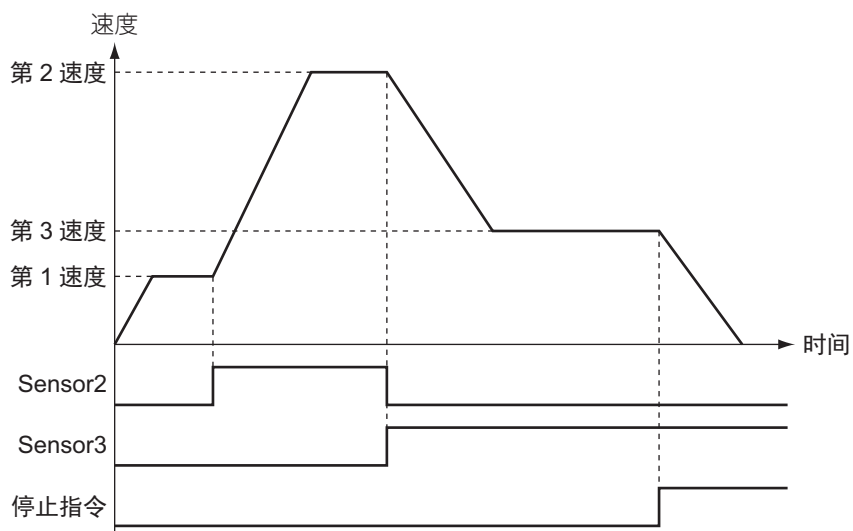
- 设定轴参数
轴种类

轴	轴种类
轴 1	伺服轴

动作示例



- 动作模式



1 速度控制的开始

检测药剂投入的传感器 1 变为“ON”后，轴 1 开始速度控制。

2 变更为第 2 速度

Sensor2（接点）变为“ON”后，设定为超驰 500%，速度发生变更。

3 变更为第 3 速度

Sensor3（接点）变为“ON”后，设定为超驰 200%，速度发生变更。

此外，Sensor2 和 Sensor3 同时变为“ON”时，变为超驰 200%。

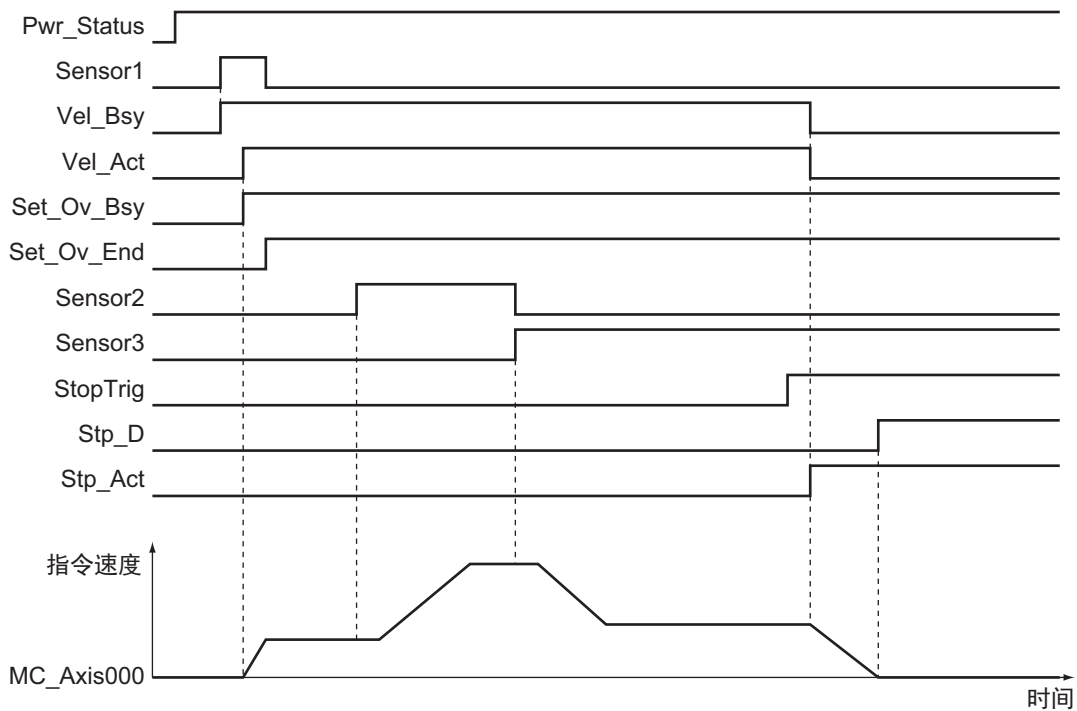
4 速度控制的停止

停止指令（StopTrig）变为“ON”后，减速停止。

梯形图**● 主要变量**

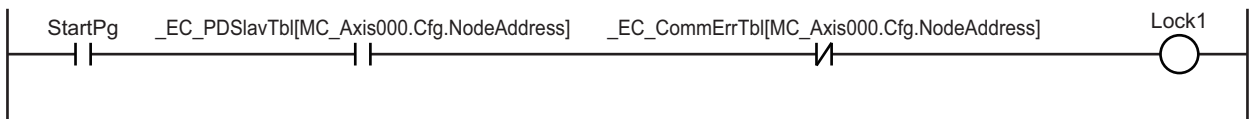
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
Pwr_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR 的输出变量 Status 中的变量。若变为伺服 ON 状态，该变量变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态，则变为伺服 ON 状态。
Vel_Act	BOOL	FALSE	分配到 MC_MoveVelocity 的实例 VEL 的输出 Active 中的变量。若实例 VEL 变为控制中，该变量变为 TRUE。
Set_Ov_Velfct	LREAL	0	超驰值。
StopTrig	BOOL	FALSE	该变量为 TRUE 时，执行 MC_Stop。
Sensor1	BOOL	FALSE	检测到药剂投入后，变为 TRUE。 轴 1 为伺服 ON 状态时，启动 MC_MoveVelocity（速度控制）。
Sensor2	BOOL	FALSE	该变量为 TRUE 时，超驰值设定为 500%。该变量一旦变为 TRUE，将保持状态，Sensor3 变为 TRUE 后，变为 FALSE。
Sensor3	BOOL	FALSE	该变量为 TRUE 时，超驰值设定为 200%。该变量一旦变为 TRUE，将保持状态。

● 时序图

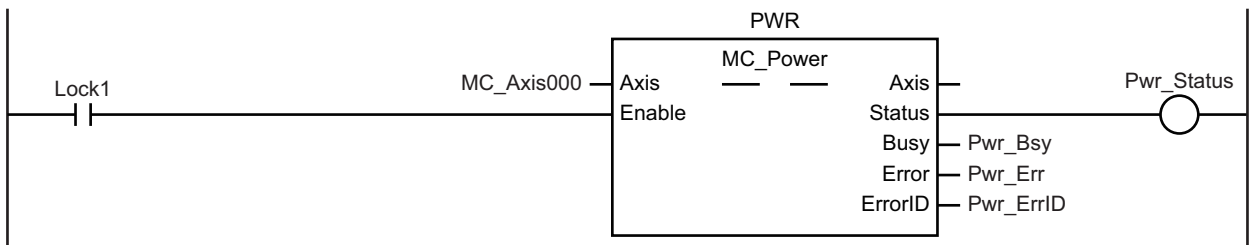


● 示例程序

接点 StartPg 变为 TRUE 后，确认 EtherCAT 通信是否正常进行过程数据通信

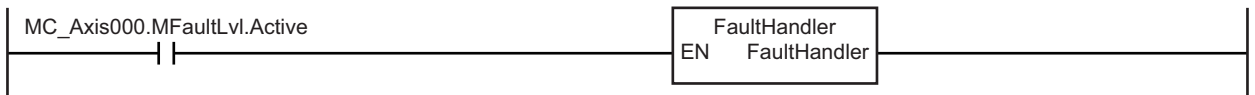


若正在进行过程数据通信，则变为伺服 ON 状态

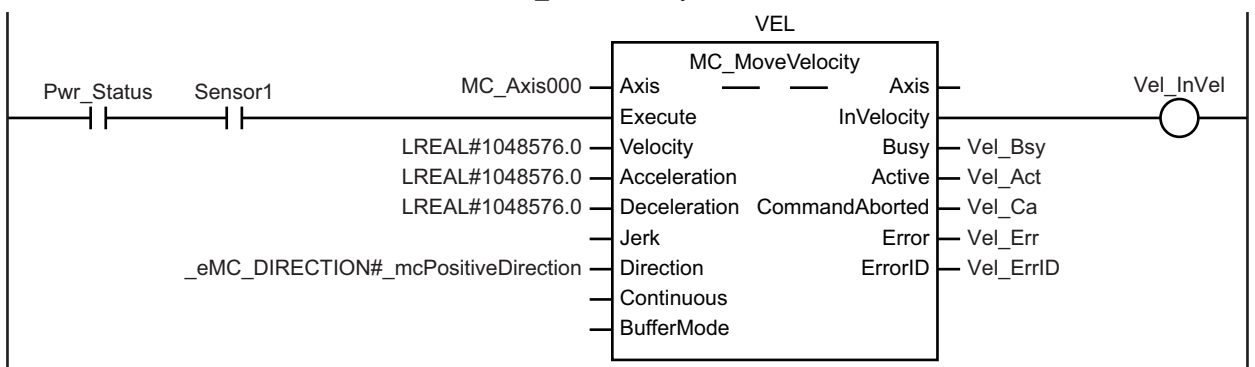


轴 1 发生轻度故障等级的异常时，执行装置发生异常时的处理（FaultHandler）。

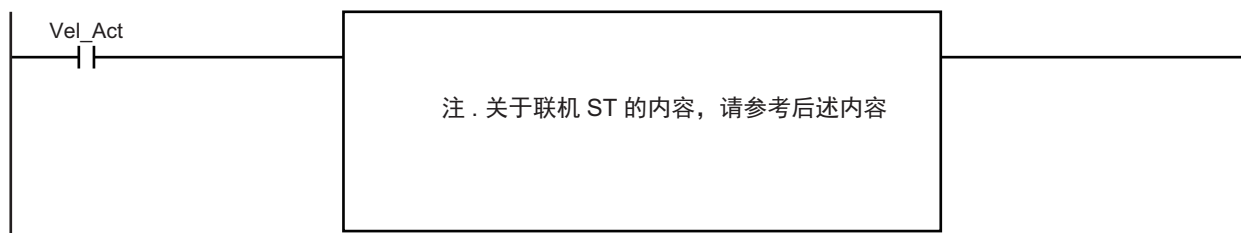
发生异常时的处理（FaultHandler）请根据装置由客户进行编程



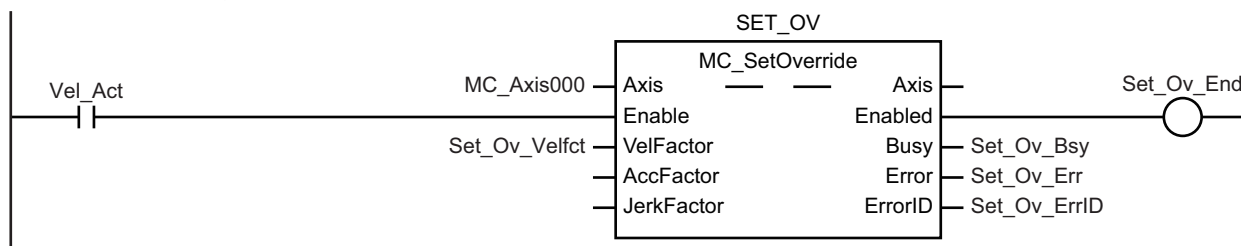
在可运行的状态下，Sensor1 为 TRUE 时启动 MC_MoveVelocity（速度控制）指令



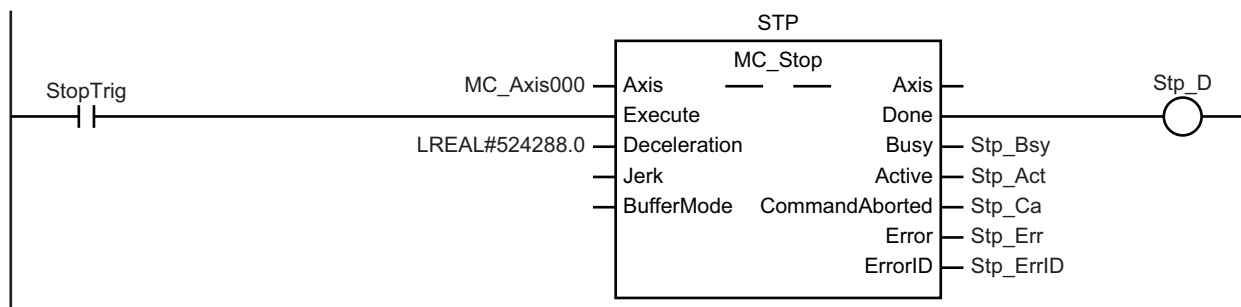
MC_MoveVelocity（速度控制）指令控制过程中，根据接点 Sensor2 和接点 Sensor3 的 ON/OFF，变更超驰值



在 MC_MoveVelocity（速度控制）指令的控制过程中，启动 MC_SetOverride（超驰值设定）指令



停止指令（StopTrig）变为 TRUE 后，启动 MC_Stop（强制停止）指令



联机 ST 的内容

```

IF (Sensor2=FALSE) AND (Sensor3=FALSE) THEN
    Set_Ov_Velfct := LREAL#100.0;
ELSIF (Sensor2=TRUE) AND (Sensor3=FALSE) THEN
    Set_Ov_Velfct := LREAL#500.0;
ELSIF (Sensor2=FALSE) AND (Sensor3=TRUE) THEN
    Set_Ov_Velfct := LREAL#200.0;
ELSE
    Set_Ov_Velfct := LREAL#200.0;
END_IF;

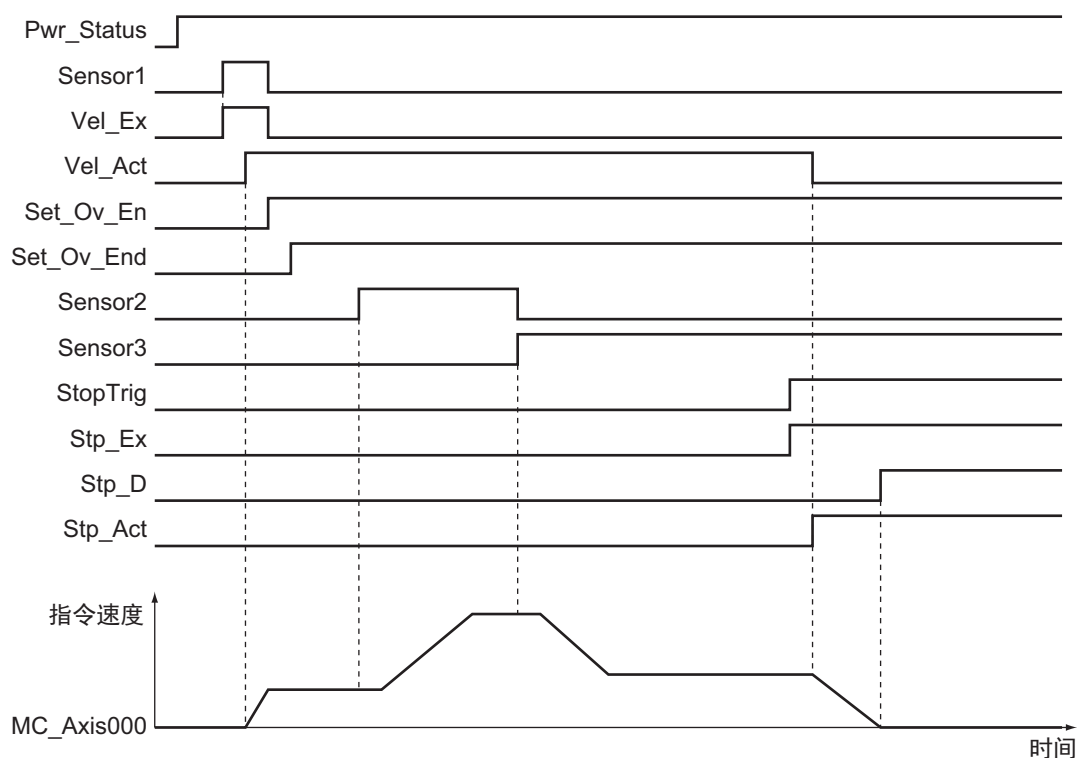
```

结构文本 (ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时, 变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
Pwr_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR 的输出变量 Status 中的变量。若变为伺服 ON 状态, 该变量变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态, 则变为伺服 ON 状态。
Vel_Act	BOOL	FALSE	分配到 MC_MoveVelocity 的实例 VEL 的输出 Active 中的变量。若实例 VEL 变为控制中, 该变量变为 TRUE。
Set_Ov_Velfct	LREAL	0	超驰值。
StopTrig	BOOL	FALSE	该变量为 TRUE 时, 执行 MC_Stop。
Sensor1	BOOL	FALSE	检测到药剂投入后, 变为 TRUE。 轴 1 为伺服 ON 状态时, 启动 MC_MoveVelocity (速度控制) 指令。
Sensor2	BOOL	FALSE	该变量为 TRUE 时, 超驰值设定为 500%。该变量一旦变为 TRUE, 将保持状态, Sensor3 变为 TRUE 后, 变为 FALSE。
Sensor3	BOOL	FALSE	该变量为 TRUE 时, 超驰值设定为 200%。该变量一旦变为 TRUE, 将保持状态。
Vel_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_MoveVelocity 的实例 VEL。
Set_Ov_En	BOOL	FALSE	该变量为 TRUE 期间, 执行 MC_SetOverride 的实例 SET_OV。
Stp_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_Stop 的实例 STP。
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE 时, 设定输入参数。输入参数设定完成后, 变为 TRUE。

● 时序图



● 示例程序

```

// 未设定输入参数时的处理
IF InitFlag = FALSE THEN
  //MC_MoveVelocity 参数
  Vel_Vel := LREAL#1048576.0;
  Vel_Acc := LREAL#1048576.0;
  Vel_Dec := LREAL#1048576.0;
  Vel_Dir := _eMC_DIRECTION#_mcPositiveDirection;

  //MC_SetOverride 参数
  Set_Ov_Velfct := LREAL#100.0;

  //MC_Stop 参数
  Stp_Dec := LREAL#524288.0;

  // 设定输入参数后，将 InitFlag 设为 TRUE
  InitFlag:=TRUE;

END_IF;

// 接点 StartPg 为 TRUE 时，确认 EtherCAT 通信正常通信，
// 轴 1 设为伺服 ON 状态
//EtherCAT 通信未正常通信时，设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDslavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
  Pwr_En:=TRUE;
ELSE
  Pwr_En:=FALSE;
END_IF;

```



```

// 轴 1 发生轻度故障时, 执行异常处理 FaultHandler
// 发生异常时的处理 (FaultHandler) 请根据装置由客户进行编程。
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;
// 轴 1 为伺服 ON 状态, 且接点 Sensor1 为 TRUE 时, 启动 MC_MoveVelocity。
IF (Pwr_Status=TRUE) AND (Sensor1=TRUE) THEN
    Vel_Ex := TRUE;
END_IF;

// MC_MoveVelocity 控制过程中, 根据接点 Sensor2 和接点 Sensor3 的 ON/OFF, 变更超驰值。
IF Vel_Act=TRUE THEN
    IF (Sensor2=FALSE) AND (Sensor3=FALSE) THEN
        Set_Ov_Velfct := LREAL#100.0;
    ELSIF (Sensor2=TRUE) AND (Sensor3=FALSE) THEN
        Set_Ov_Velfct := LREAL#500.0;
    ELSIF (Sensor2=FALSE) AND (Sensor3=TRUE) THEN
        Set_Ov_Velfct := LREAL#200.0;
    ELSE
        Set_Ov_Velfct := LREAL#200.0;
    END_IF;
END_IF;

// MC_MoveVelocity 控制过程中, 启动 MC_SetOverride。
IF Vel_Act=TRUE THEN
    Set_Ov_En := TRUE;
END_IF;

// StopTrig 变为 TRUE 时, 启动 MC_Stop。
IF StopTrig=TRUE THEN
    Stp_Ex := TRUE;
END_IF;

//MC_Power
PWR(
    Axis      := MC_Axis000,
    Enable    := Pwr_En,
    Status    => Pwr_Status,
    Busy      => Pwr_Bsy,
    Error     => Pwr_Err,
    ErrorID   => Pwr_ErrID
);

//MC_MoveVelocity
VEL(
    Axis          := MC_Axis000,
    Execute       := Vel_Ex,
    Velocity      := Vel_Vel,
    Acceleration  := Vel_Acc,
    Deceleration  := Vel_Dec,
    Direction     := Vel_Dir,
    InVelocity    => Vel_Invel,
    Busy          => Vel_Bsy,
    Active        => Vel_Act,
    CommandAborted => Vel_Ca,

```

```
Error          => Vel_Err,  
ErrorID        => Vel_ErrID  
);  
  
//MC_SetOverride  
SET_OV(  
  Axis          := MC_Axis000,  
  Enable        := Set_Ov_En,  
  VelFactor     := Set_Ov_Velfct,  
  AccFactor     := Set_Ov_Accfct,  
  JerkFactor    := Set_Ov_Jfct,  
  Busy          => Set_Ov_Bsy,  
  Enabled       => Set_Ov_End,  
  Error         => Set_Ov_Err,  
  ErrorID      => Set_Ov_ErrID  
);  
  
//MC_Stop  
STP(  
  Axis          := MC_Axis000,  
  Execute       := Stp_Ex,  
  Deceleration  := Stp_Dec,  
  Done          => Stp_D,  
  Busy         => Stp_Bsy,  
  Active        => Stp_Act,  
  CommandAborted => Stp_Ca,  
  Error         => Stp_Err,  
  ErrorID      => Stp_ErrID  
);
```

MC_MoveZeroPosition

将绝对坐标“0”作为目标位置，进行定位并返回原点。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_MoveZeroPosition	高速 原点复位	FB		<pre>MC_MoveZeroPosition_instance (Axis := 《参数》, Execute := 《参数》, Velocity := 《参数》, Acceleration := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, BufferMode := 《参数》, Done => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令
Velocity	目标速度	LREAL	正数	0	指定目标速度。 ^{*1} 单位为 [指令单位 /s]。 ^{*2}
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度。 单位为 [指令单位 /s ²]。 ^{*2}
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。 单位为 [指令单位 /s ²]。 ^{*2}
Jerk	跃度	LREAL	正数或“0”	0	指定跃度。 单位为 [指令单位 /s ³]。 ^{*2}
BufferMode	缓冲 模式选择	_eMC_BUFFER_ MODE	0: _mcAborting 1: _mcBuffered	0 ^{*3}	指定运动指令多重启动时的动作。 0: 中断 1: 缓冲

*1. 请务必设定目标速度。若不设定而直接动作，将发生异常。

*2. 指令单位请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》的“单位转换设定”。

*3. 如果是有效范围为列举型的变量初始值，实际不是数值，而是列举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* □ □ 「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	定位完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	轴已开始移动时	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动（中断），本指令中止时 因发生异常，本指令中止时 正在发生异常时，启动了本指令 正在执行 MC_Stop 指令时，启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*]、_MC1_AX[*]、_MC2_AX[*]）。

功能说明

- 移动到原点位置。
- 输入变量中可指定 Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）、Jerk（跃度）。



使用注意事项

MC_MoveZeroPosition（高速原点复位）指令请在原点确定的状态下执行。若原点未确定，将发生“原点未确定状态下指令启动异常（错误代码：5466 Hex）”。

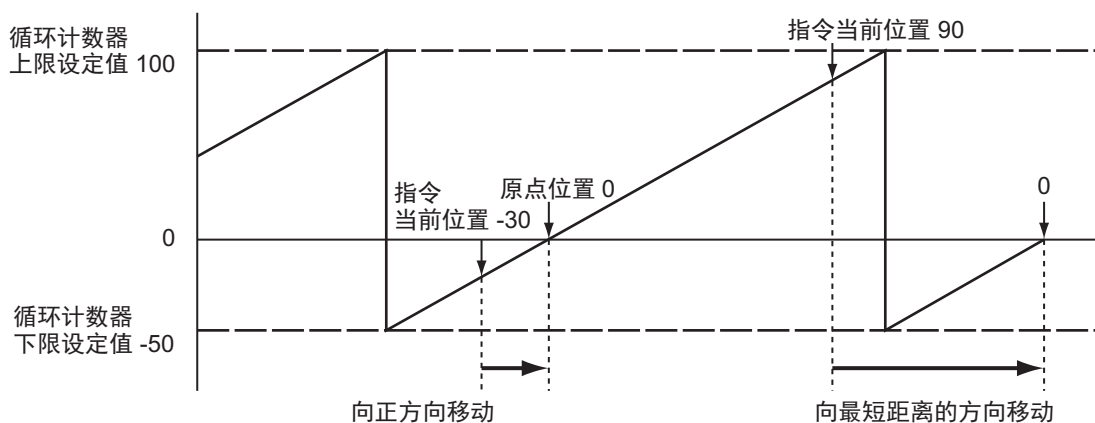
指令的详情

下面介绍指令的详情。

● 方向指定

计数模式为 [旋转模式] 时，向距离原点位置较近的方向进行定位（最短距离）。

与指令当前位置对应的本指令的动作示例如下所示。



使用注意事项

如果在原点位于 [循环计数器上限设定值] 和 [循环计数器下限设定值] 范围外的状态下启动了本指令，将发生“超出目标位置循环计数器范围（错误代码：549C Hex）”。

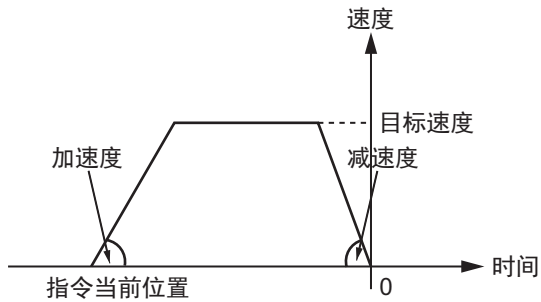
● 超驰

超驰对本指令有效。

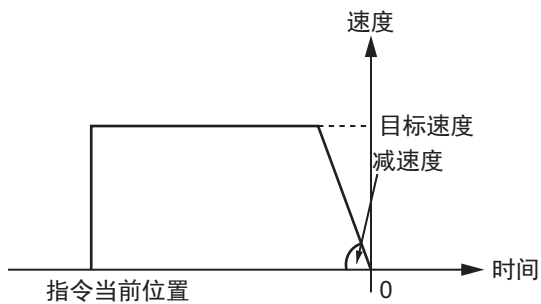
● 位置检查

本指令的位置检查根据轴参数 [位置宽度]、[位置检查时间] 的设置执行。

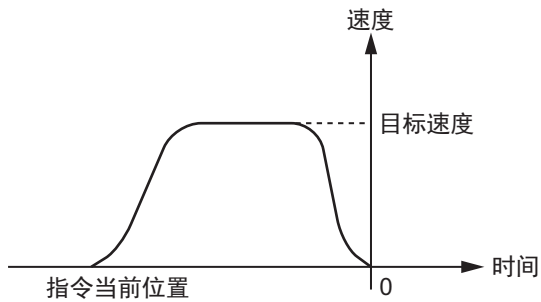
● 动作示例



将加速度或减速度指定为“0”并启动时，将不进行加减速而达到目标速度。
加速度为“0”时的动作示例如下。



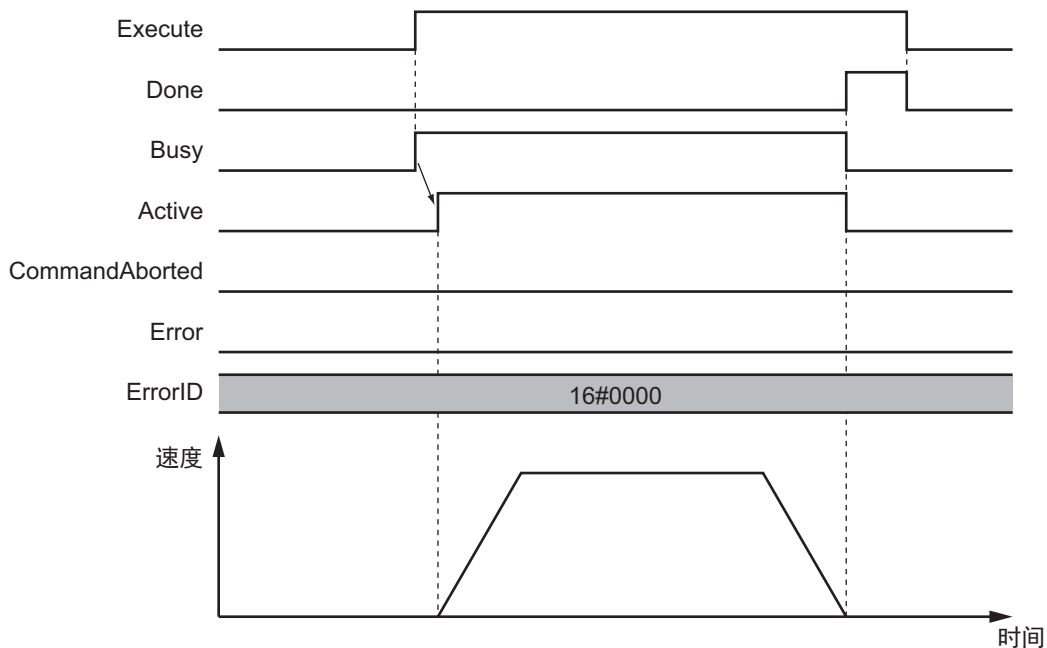
希望均匀加减速时，需指定 Jerk（跃度）。
指定了 Jerk（跃度）时的动作示例如下。



Jerk（跃度）的详情请参考 □□ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

时序图

执行了 MC_MoveZeroPosition（高速原点复位）指令时的时序图如下所示。



指令的中止

即使本指令中止，CommandAborted（执行中断）变为 TRUE，也不会变为原点未确定状态。

运动指令重新执行

在定位动作中变更输入变量，并再次将 Execute（启动）设为 TRUE，可变更本指令的动作。通过运动指令重新执行可变更的输入变量包括 Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）。

运动指令重新执行的详情请参考 [□](#) 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

运动指令多重启动

运动指令多重启动的详情请参考 □□《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 其他指令执行过程中本指令的启动

对当前正在执行的指令启动本指令后，可切换为本指令或缓冲本指令。
每个轴最多可缓冲 1 个。

在 BufferMode（缓冲模式选择）中指定多重启动时本指令的动作。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 切换指令后如果轴的动作方向发生反转，将按照轴参数 [反转时动作] 执行反转动作。
缓冲	当前正在执行的指令正常完成后，自动启动已缓冲的本指令。

BufferMode（缓冲模式选择）请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 本指令执行过程中其他指令的启动

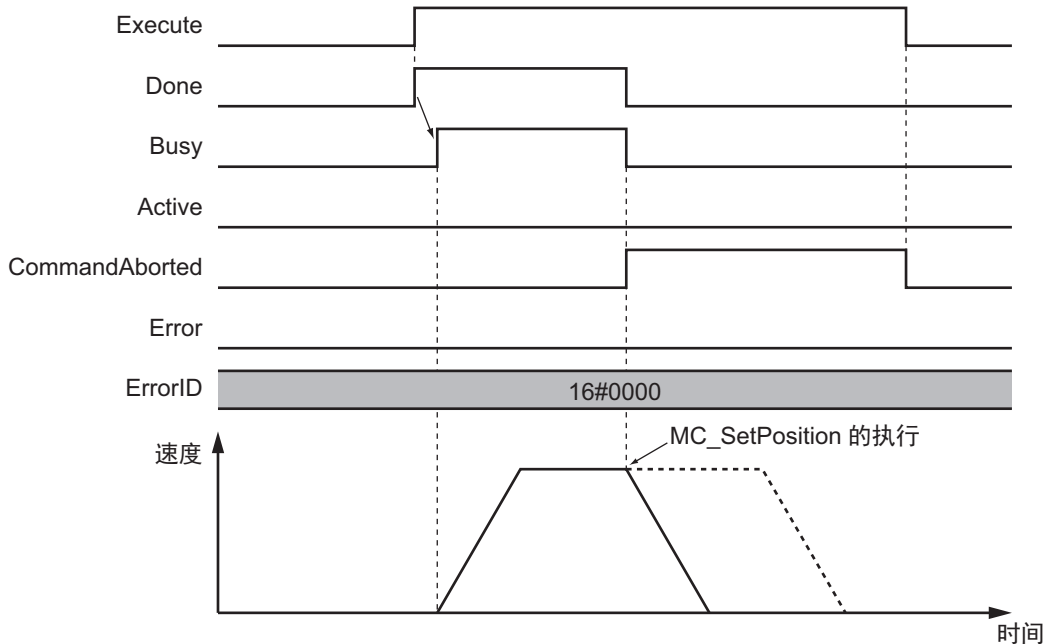
对本指令执行运动指令多重启动时，可进行中断、缓冲。
不可通过共混进行指令多重启动。

正在指令本指令时执行了 MC_SetPosition（当前位置变更）指令时

MC_SetPosition（当前位置变更）指令中，会变为原点未确定状态，因此正在执行本指令时，将发生异常，无法执行。

因执行MC_SetPosition（当前位置变更）指令而变为异常状态时，本指令将减速停止，CommandAborted（执行中断）变为 TRUE。

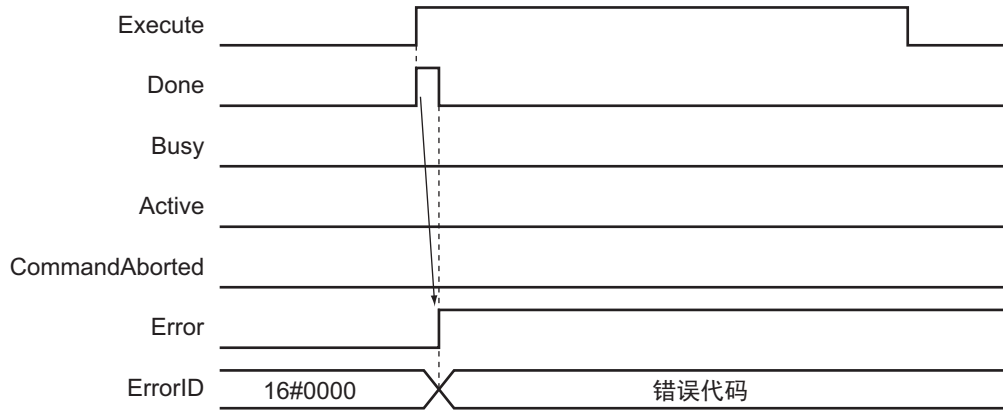
此时，MC_SetPosition（当前位置变更）指令中的输出变量 Error（错误）变为 TRUE。



异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。
参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

MC_MoveFeed

指定基于外部输入的中断输入发生位置起的移动距离，进行定位。
可进行中断固定尺寸定位的控制为绝对值定位、相对值定位、速度控制。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_MoveFeed	中断 固定尺寸定位	FB	<p>MC_MoveFeed_instance</p> <p>MC_MoveFeed</p> <p>Axis — — Axis</p> <p>TriggerInput — — TriggerInput</p> <p>TriggerVariable — — TriggerVariable</p> <p>Execute — — Done</p> <p>WindowOnly — — InFeed</p> <p>FirstPosition — — Busy</p> <p>LastPosition — — Active</p> <p>ReferenceType — — CommandAborted</p> <p>Position — — Error</p> <p>Velocity — — ErrorID</p> <p>Acceleration — —</p> <p>Deceleration — —</p> <p>Jerk — —</p> <p>Direction — —</p> <p>MoveMode — —</p> <p>FeedDistance — —</p> <p>FeedVelocity — —</p> <p>BufferMode — —</p> <p>ErrorDetect — —</p>	<pre> MC_MoveFeed_instance (Axis := 《参数》, TriggerInput := 《参数》, TriggerVariable := 《参数》, Execute := 《参数》, WindowOnly := 《参数》, FirstPosition := 《参数》, LastPosition := 《参数》, ReferenceType := 《参数》, Position := 《参数》, Velocity := 《参数》, Acceleration := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, Direction := 《参数》, MoveMode := 《参数》, FeedDistance := 《参数》, FeedVelocity := 《参数》, BufferMode := 《参数》, ErrorDetect := 《参数》, Done => 《参数》, InFeed => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》); </pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
WindowOnly	窗口有效	BOOL	TRUE、FALSE	FALSE	指定窗口的有效 / 无效。
FirstPosition	开始位置	LREAL	负数、正数、“0”	0	指定锁定变为有效的位置。 单位为 [指令单位]。*1
LastPosition	结束位置	LREAL	负数、正数、“0”	0	指定锁定变为无效的位置。 单位为 [指令单位]。*1
Reference Type	位置种类选择	_eMC_REFERENCE_TYPE	1: _mcFeedback	1 *2	指定位置的类别。 1: 反馈位置（同一任务周期*3 的获取值）
Position	目标位置	LREAL	负数、正数、“0”	0	MoveMode 为 [0: 绝对值定位] 时，指定绝对坐标的目标位置。 MoveMode 为 [1: 相对值定位] 时，指定移动距离。 MoveMode 为 [2: 速度控制] 时，无需指定。 单位为 [指令单位]。*1
Velocity	目标速度	LREAL	正数*4	0	指定目标速度。 单位为 [指令单位 /s]。*1
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度。 单位为 [指令单位 /s ²]。*1
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。 单位为 [指令单位 /s ²]。*1
Jerk	跃度	LREAL	正数或“0”	0	指定跃度。 单位为 [指令单位 /s ³]。*1
Direction	方向选择	_eMC_DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	0 *2	MoveMode 为 [0: 绝对值定位]*5， 计数模式为 [旋转模式] 时，指定旋转方向。 0: 正方向指定 1: 最短距离指定 2: 负方向指定 3: 当前方向指定 4: 无方向指定
MoveMode	移动方法选择	_eMC_MOVE_MODE	0: _mcAbsolute 1: _mcRelative 2: _mcVelocity	0 *2	选择移动方法。 0: 绝对值定位 1: 相对值定位 2: 速度控制

输入变量	名称	数据类型	有效范围	初始值	内容
FeedDistance	固定尺寸距离	LREAL	负数、正数、“0”	0	指定中断输入后的移动距离。向中断输入前的动作方向进行固定尺寸动作时，设定为正数，向其反方向动作时，设定为负数。单位为 [指令单位]。 ^{*1}
FeedVelocity	固定尺寸速度	LREAL	正数	0	指定中断输入后的移动目标速度。单位为 [指令单位 /s]。 ^{*1}
BufferMode	缓冲模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0 ^{*2}	指定运动指令多重启动时的动作。 0: 中断 1: 缓冲
ErrorDetect	错误检测选择	BOOL	TRUE、FALSE	FALSE	无中断输入时，指定是否进行异常检测。 TRUE: 有异常检测 FALSE: 无异常检测

*1. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. 如果是有效范围为列举型的变量初始值，实际不是数值，而是列举值。

*3. 任务为主固定周期任务时，任务周期为主周期，为固定周期任务（执行优先级 5）时，任务周期为固定周期任务（执行优先级 5）的任务周期。

*4. MoveMode 指定为“2: _mcVelocity”时，可以为“0”。

*5. MoveMode 为 [0: 绝对值定位]，计数模式为 [直线模式] 时，移动到指定的位置。

MoveMode 为 [1: 相对值定位] 时，根据 Position 的符号决定移动方向。

MoveMode 为 [2: 速度控制] 时，无论计数器模式如何，“1: _mcShortestWay”及“4: _mcNoDirection”将无法选择。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
InFeed	固定尺寸输送中	BOOL	TRUE、FALSE	接收锁定输入，进行固定尺寸输送时变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*1	发生异常时，输出错误代码。16#0000 表示正常。

*1. □ □ 「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	指令完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	开始了指令时	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
InFeed	通过中断输入, 开始了固定尺寸动作时*1	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动 (中断), 本指令中止时 因发生异常, 本指令中止时 正在发生异常时, 启动了本指令 正在执行 MC_Stop 指令时, 启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

*1. 从中断输入变为“ON”到 InFeed 变为 TRUE 之前, 可能会延迟几个任务周期。任务为主固定周期任务时, 任务周期为主周期, 为固定周期任务 (执行优先级 5) 时, 任务周期为固定周期任务 (执行优先级 5) 的任务周期。

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。*1
TriggerInput	触发输入条件	_sTRIGGER_REF	—	设定触发条件。*2
TriggerVariable	触发变量	BOOL	TRUE、FALSE	在触发条件中指定了控制器模式时, 指定作为触发的输入变量。

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称 (默认“MC_Axis****”)或系统定义变量的轴变量名称 (_MC_AX[*], _MC1_AX[*], _MC2_AX[*])。

*2. 请定义 _sTRIGGER_REF 型用户定义变量。

● _sTRIGGER_REF

成员变量	名称	数据类型	有效范围	功能
Mode	模式	_eMC_ TRIGGER_ MODE	0: _mcDrive 1: _mcController	指定触发的模式。 0: 驱动器模式 1: 控制器模式
LatchID	锁定 ID 选择	_eMC_ TRIGGER_ LATCHID	0: _mcLatch1 1: _mcLatch2	驱动器模式时, 指定使用两个锁定功能中的哪个。 0: 锁定功能 1 1: 锁定功能 2
InputDrive	驱动器触发输入信号	_eMC_ TRIGGER_ INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	指定驱动器模式时的驱动器触发信号。 0: Z 相 1: 外部输入

功能说明

- 在 Execute（启动）的上升沿，根据 MoveMode（移动方法选择）的设定，按绝对值移动、相对值移动或速度控制中的任一移动方法移动。
- 绝对值移动时，在 Position（目标位置）中设定目标位置，相对值移动时，在 Position（目标位置）中设定目标距离。
此外，任一移动方法均按 Velocity（目标速度）进行移动。
- 根据移动过程中外部输入（中断输入）的上升沿，从反馈位置起，按 FeedDistance（固定尺寸距离）中指定的固定尺寸距离，以 FeedVelocity（固定尺寸速度）执行相对定位动作。
- 在绝对值移动或相对值移动指令的中断固定尺寸输送中，达到目标位置前没有中断信号的输入时，在一开始的目标位置停止。
无中断输入并停止时，通过 ErrorDetect（错误检测选择），可指定异常输出的有无。指定为输出异常时，CommandAborted（执行中断）变为 TRUE，Busy（执行中）、Active（控制中）变为 FALSE。
- 使用中断主机时，将 WindowOnly（窗口有效）设为 TRUE，指定 FirstPosition（开始位置）、LastPosition（结束位置）。
根据反馈位置为 FirstPosition（开始位置）到 LastPosition（结束位置）期间最先发生的中断信号，执行中断固定尺寸定位。



使用注意事项

- 中断后的固定尺寸动作作为 FeedDistance（固定尺寸距离）中指定距离的相对移动。
FeedDistance（固定尺寸距离）指定为正数时，向中断输入前相同的方向进行固定尺寸动作，指定为负数时，向反方向进行固定尺寸动作。
- 以固定尺寸动作反转时的加减速遵照轴参数 [反转时动作] 的设定。
- 中断固定尺寸定位后的位置发生溢出 / 下溢时，将发生中断输入时异常。发生溢出 / 下溢后如果有中断输入，也会发生轴异常。



参考

绝对值移动请参考 □□「MC_MoveAbsolute(P.3-47)」，
相对值移动请参考 □□「MC_MoveRelative(P.3-73)」，
速度控制请参考 □□「MC_MoveVelocity(P.3-80)」
WindowOnly（窗口有效）请参考 □□「WindowOnly（窗口有效）(P.3-109)」。

● 数据对象的映射

Mode 选择为 [驱动器模式]，使用 MC_MoveFeed (中断固定尺寸定位) 指令时，请映射以下对象数据。映射在 Sysmac Studio 的轴基本设定 [详细设定] 中进行。

- 锁定功能 (60B8Hex)
- 锁定状态 (60B9Hex)
- 锁定位置 1 (60BAHex)
- 锁定位置 2 (60BCHex)

未设定以上内容时，将发生过程数据对象设定不足 (错误代码: 3461Hex)。

数据对象的映射请参考 □□「2-3 PDO 映射 (P.2-30)」、□□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

使用 NX 系列位置接口单元时请参考 □□《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》的“I/O 入口映射”。

指令的详情

下面介绍指令的详情。

● Axis (轴) 指定

- 在 Axis (轴) 中指定锁定对象的轴。

● 触发输入条件

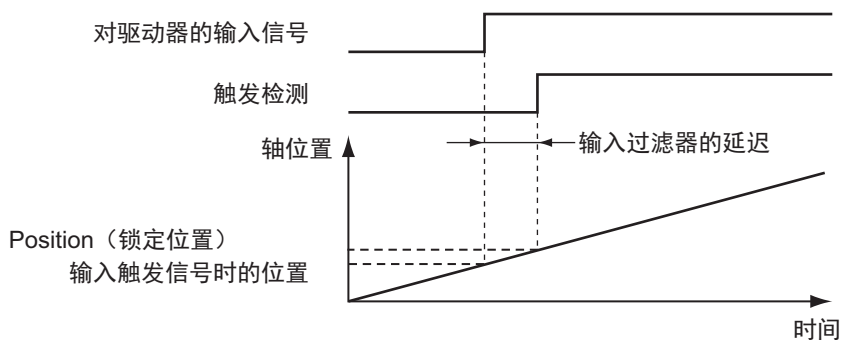
通过 TriggerInput (触发输入条件) 的 Mode、LatchID、InputDrive 指定，选择触发的条件。

Mode

- Mode 的设定包括：在触发中指定伺服驱动器或 NX 系列脉冲输出单元信号的“驱动器模式”；在触发中指定 TriggerVariable (触发变量) 的“控制器模式”。
- 触发信号的上升沿视为发生触发，记录执行锁定指令后最先发生触发 (FALSE → TRUE) 时轴的位置。
- 本指令为 Busy (动作中) 时，即使 Execute (启动) 为 FALSE，TriggerVariable (触发变量) 的变量仍作为触发动作。

驱动器模式

“驱动器模式”通过伺服驱动器的功能进行触发检测和获取当前位置，因此与“控制器模式”相比，锁定的当前位置更准确。



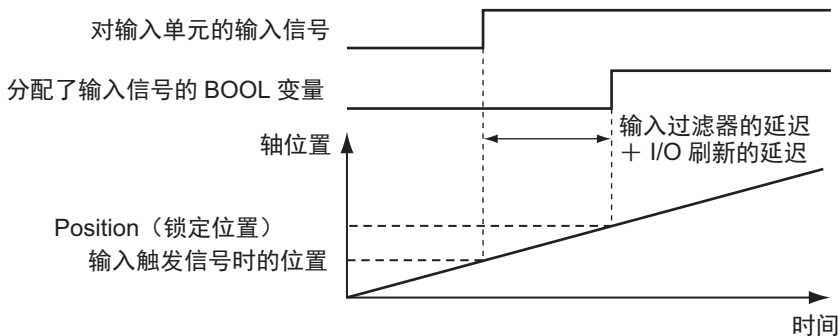


使用注意事项

- 使用驱动器模式时，请务必在要使用的 LatchID（锁定 ID 选择）上连接锁定信号。
- 锁定信号的宽度取决于伺服驱动器或 NX 系列脉冲输出单元的性能等。
- InputDrive 选择为 [驱动器模式]，使用 MC_MoveFeed（中断固定尺寸定位）指令时，请映射以下对象数据。
 锁定功能（60B8Hex）、锁定状态（60B9Hex）、锁定位置 1（60BAHex）、锁定位置 2（60BCHex）
 未设定以上内容时，将发生过程数据对象设定不足（错误代码：3461Hex）。
 对象数据的映射请参考 □「2-3 PDO 映射 (P.2-30)」、□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

控制器模式

- “控制器模式”中，触发可指定为 BOOL 变量。
- 在 TriggerVariable（触发变量）中指定要作为触发的 BOOL 变量。
- 在“控制器模式”中，触发输入信号向 BOOL 变量的反映会因 I/O 刷新的延迟而比“驱动器模式”大幅延迟。



使用注意事项

- 使用控制器模式时，根据任务周期的间隔执行锁定处理。因此，触发变量必须在任务周期的 1 个周期内保持 TRUE。
- 此外，触发变量变为 TRUE 后，在处理 MC 功能模块前，需要 1 个任务周期的时间。
- 任务为主固定周期任务时，这里的任务周期为主周期，为固定周期任务（执行优先度 5）时，任务周期为固定周期任务（执行优先度 5）的任务周期。

LatchID

- 在 LatchID（锁定 ID 选择）中指定使用哪个。1 个轴只能使用其中 1 个。
- 在伺服驱动器或 NX 系列脉冲输出单元中，LatchID（锁定 ID 选择）表示伺服驱动器或 NX 系列脉冲输出单元的锁定回路 1（第 1ch）、锁定回路 2（第 2ch）。

LatchID 请参考 □《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

InputDrive

- 可选择 `_mcEncoderMark` (Z相) 和 `_mcEXT` (外部输入) 作为触发。
- `_mcEncoderMark` (Z相) 中, 将伺服驱动器或 NX 系列脉冲输出单元的 Z 相作为触发使用。
`_mcEXT` (外部输入) 中, 将输入到伺服驱动器或 NX 系列脉冲输出单元的外部触发信号作为触发。
- `_mcEXT` (外部输入) 在欧姆龙生产的伺服驱动器 G5 系列中, 为设定的 Ext1、Ext2、Ext3 之一。设定通过 Sysmac Studio 进行。
驱动器中设定的 2 个触发可以是相同的设定。
- 使用 NX 系列脉冲输出单元时, 请参考 □ 《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

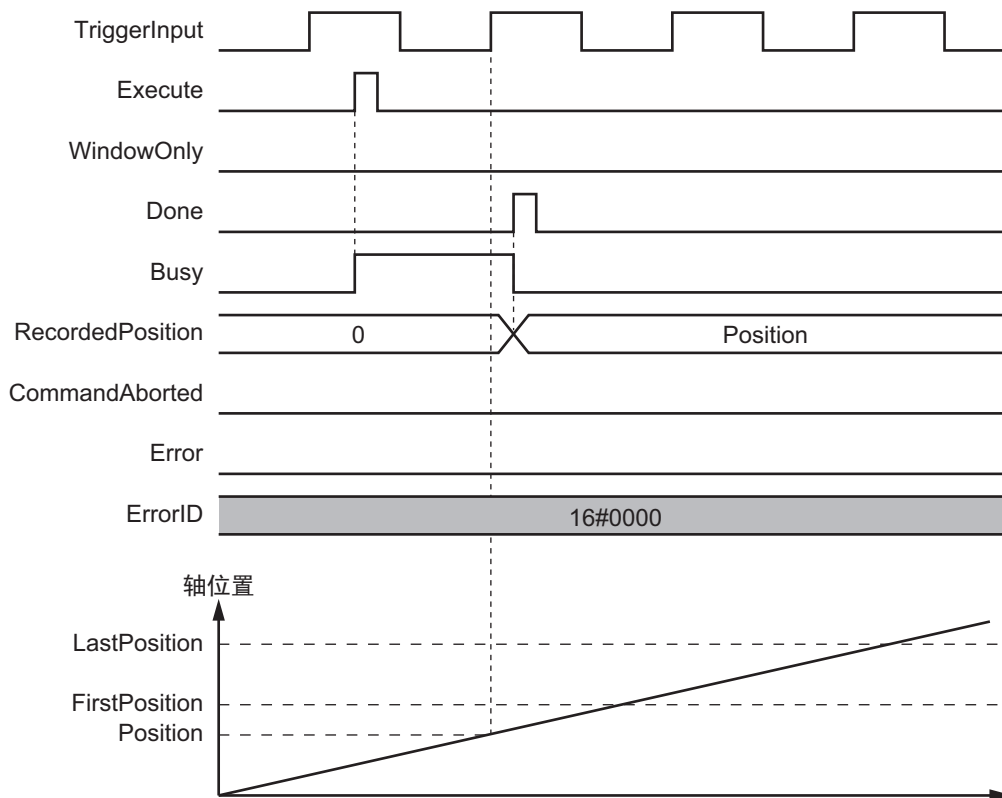
● WindowOnly (窗口有效)

- 在 WindowOnly (窗口有效) 中指定窗口的有效 / 无效。
- 指定为 Disable (无效) 时, 在所有的轴位置上检测触发。
- 指定为 Enable (有效) 时, 仅检测轴位置为 FirstPosition (开始位置) 和 LastPosition (结束位置) 范围内的触发。

不同 WindowOnly (窗口有效) 指定时的动作如时序图所示。

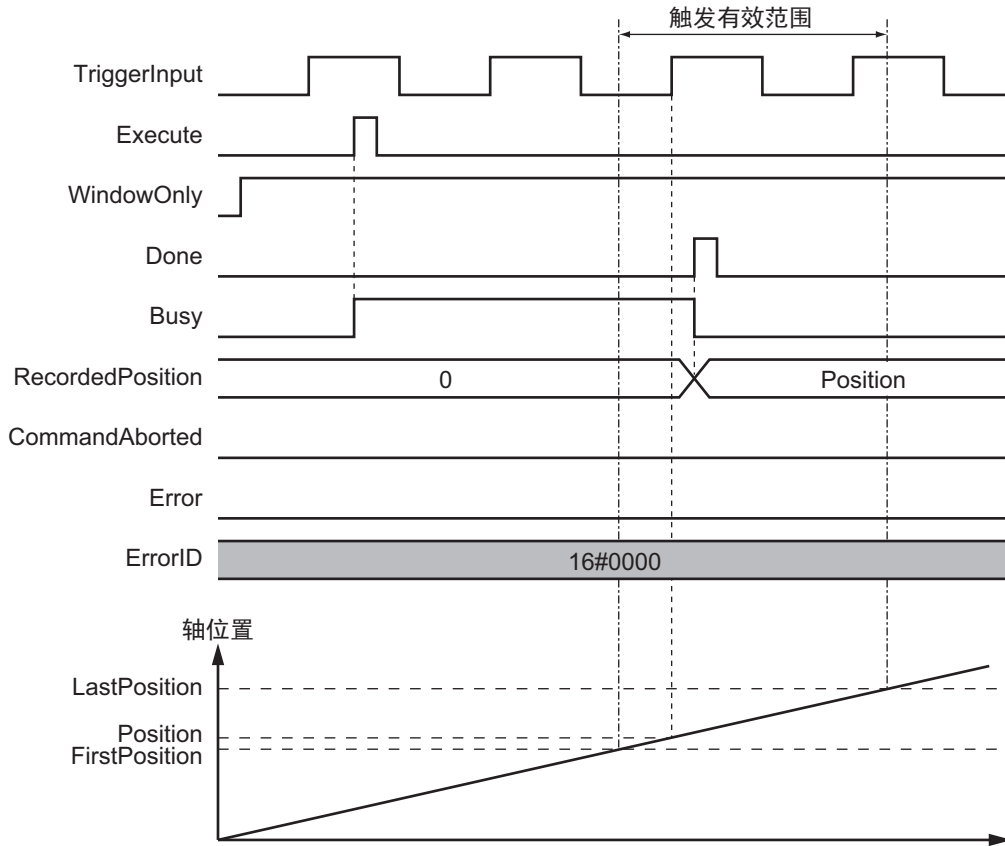
WindowOnly 为 Disable 时

Execute (启动) 变为 TRUE 后, 将最先发生触发时的轴位置作为固定尺寸距离的基准位置。



WindowOnly 为 Enable 时

仅在窗口范围内，检测触发输入，获取轴位置。



使用注意事项

- WindowOnly（窗口有效）从 FALSE 变为 TRUE 的瞬间和锁定功能动作前无法锁定。
- 锁定功能开始动作需要一定的时间，因此 WindowOnly（窗口有效）的有效范围非常短时，将无法锁定。可锁定的有效范围极限值取决于伺服驱动器、编码器输入终端、位置接口单元等的性能及 EtherCAT 通信。

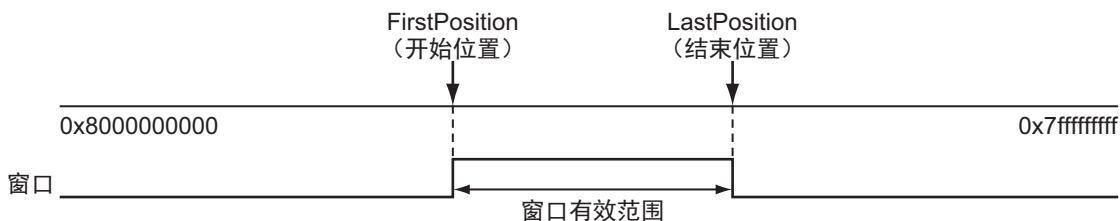
此外，因计数模式的不同，FirstPosition（开始位置）和 LastPosition（结束位置）的范围如下所示。

直线模式

- $\text{FirstPosition (开始位置)} \leq \text{窗口范围} \leq \text{LastPosition (结束位置)}$
- 指定为 $\text{FirstPosition (开始位置)} > \text{LastPosition (结束位置)}$ 时，将发生异常。
- 指定为超出 [直线模式] 的位置范围时，也会发生异常。
- 此外，FirstPosition（开始位置）和 LastPosition（结束位置）为 LREAL 型变量，因此请勿指定为 $\text{FirstPosition (开始位置)} = \text{LastPosition (结束位置)}$ 。

LREAL 型请参考 □「实数（REAL 型、LREAL 型）的处理 (P.1-14)」。

[直线模式] 时的窗口有效范围如下图所示。

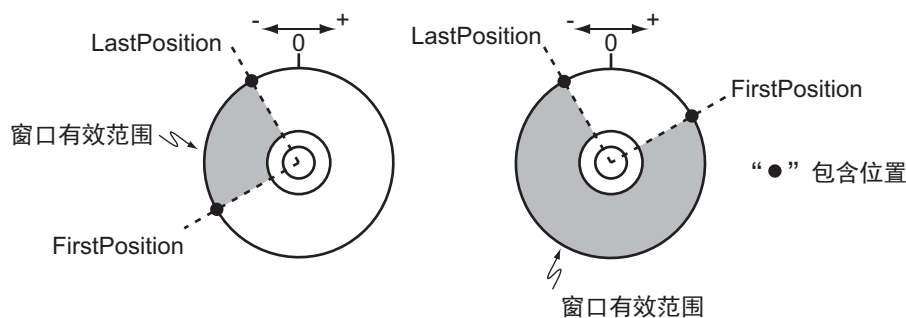


(注) FirstPosition (开始位置) 和 LastPosition (结束位置) 包含于窗口有效范围内。

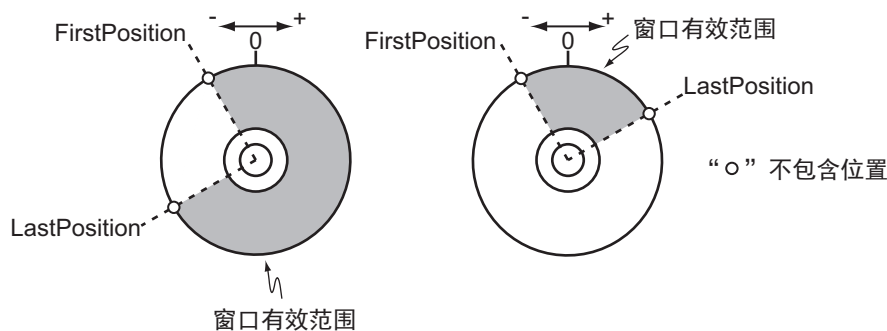
旋转模式

- 可指定为 FirstPosition (开始位置) \leq LastPosition (结束位置) 及 FirstPosition (开始位置) $>$ LastPosition (结束位置)。
- FirstPosition (开始位置) $>$ LastPosition (结束位置) 时, 将跨循环计数器的上下限位置。
- 若指定的位置超出循环计数器上下限范围, 将发生异常。

FirstPosition \leq LastPosition



FirstPosition $>$ LastPosition



● ReferenceType (位置种类选择)

位置的种类如下所示。

- `_mcFeedback`: 同一任务周期的获取值
使用同一任务周期内获取的主轴的反馈位置。
任务为主固定周期任务时, 这里的任务周期为主周期, 为固定周期任务 (执行优先度 5) 时, 任务周期为固定周期任务 (执行优先度 5) 的任务周期。

● FeedDistance（固定尺寸距离）

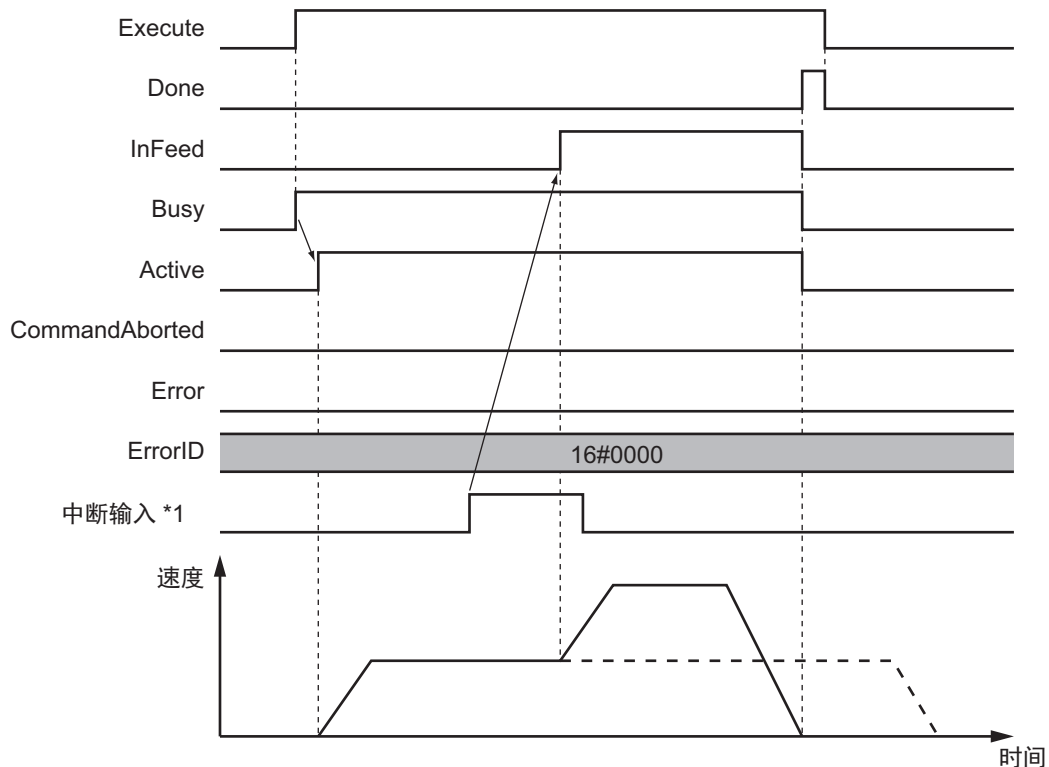
向与中断输入前相同的方向进行固定尺寸动作时，在 FeedDistance（固定尺寸距离）中指定正值。向与中断输入前相反的方向进行固定尺寸动作时，在 FeedDistance（固定尺寸距离）中指定负值。

例如，中断输入前向负方向动作时，若 FeedDistance（固定尺寸距离）指定为正值，则固定尺寸动作向负方向进行。若 FeedDistance（固定尺寸距离）指定为负值，则固定尺寸动作向正方向进行。

时序图

- 在 Execute（启动）上升的同时 Busy（执行中）变为 TRUE，在下一周期，Active（控制中）变为 TRUE。
- 中断输入后，InFeed（固定尺寸输送中）变为 TRUE，达到 FeedDistance（固定尺寸距离），完成定位后，Done（完成）变为 TRUE。
- 通过其他指令中断了本指令时，CommandAborted（执行中断）变为 TRUE，Busy（执行中）、Active（控制中）变为 FALSE。

● MoveMode（移动方法选择）为 _mcAbsolute、_mcRelative 时



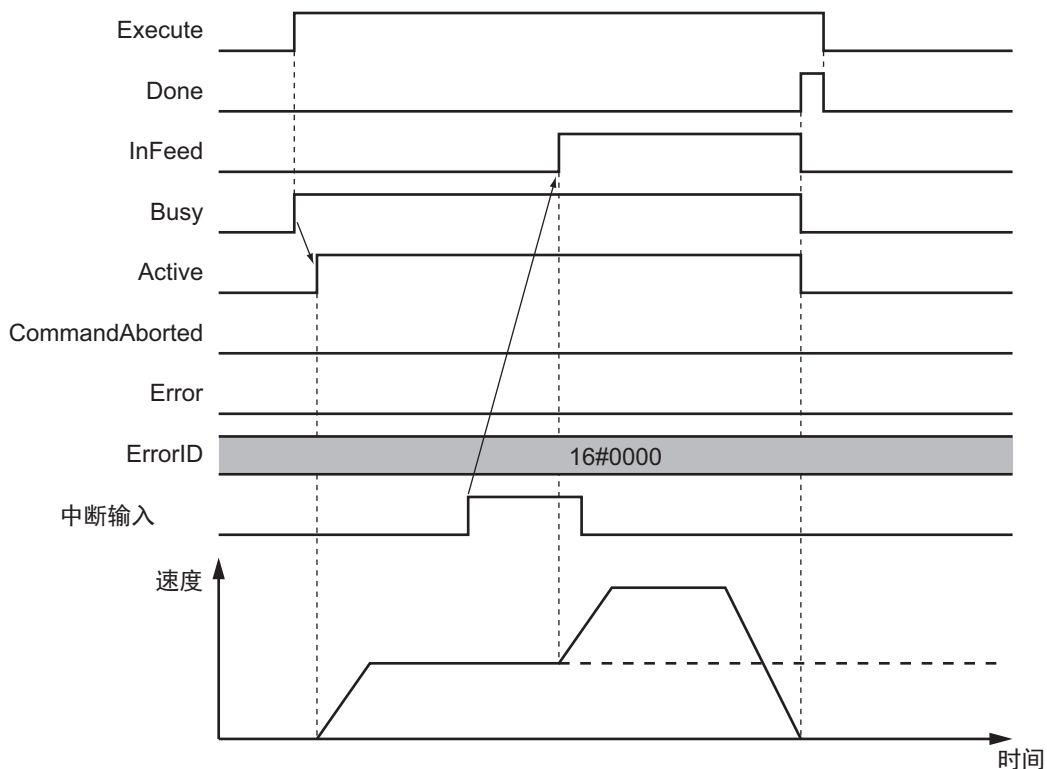
*1 从中断信号变为“ON”到 InFeed 变为 TRUE 之前，可能会延迟几个任务周期。



参考

对中断固定尺寸定位的加速及减速遵照输入变量 Acceleration（加速度）、Deceleration（减速度）。

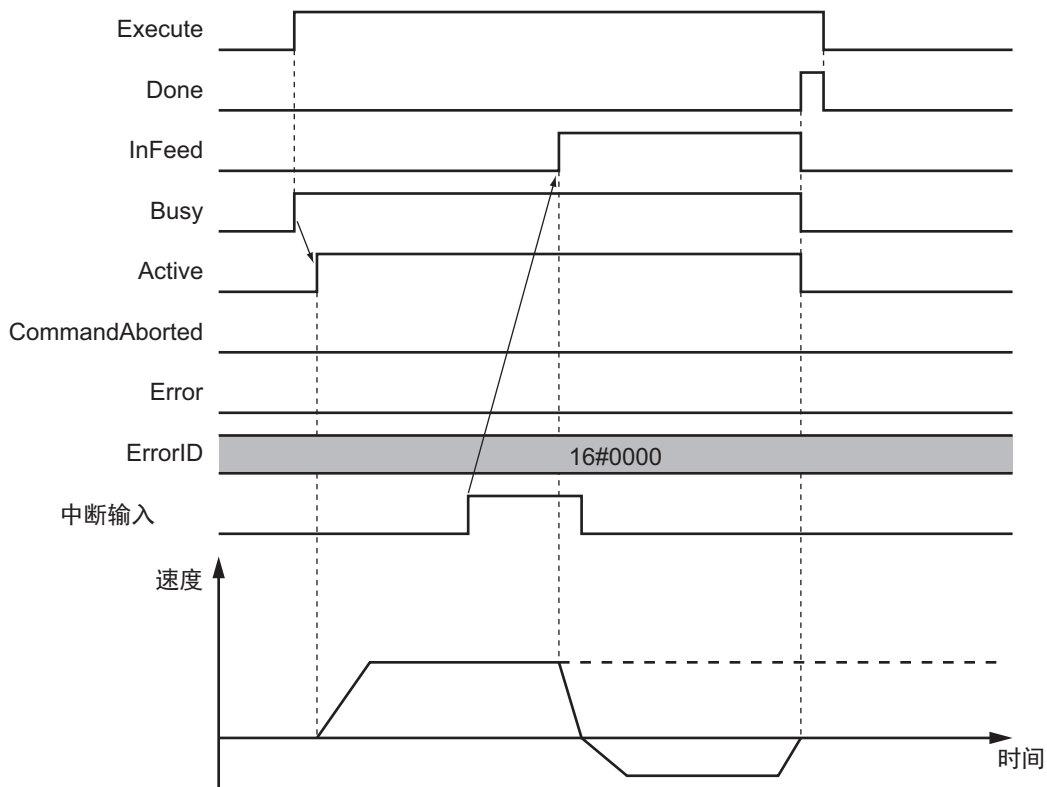
● MoveMode（移动方法选择）为 _mcVelocity 时



● 轴参数的反转时动作

中断后的固定尺寸定位反转时，遵照轴参数 [反转时动作] 进行反转。

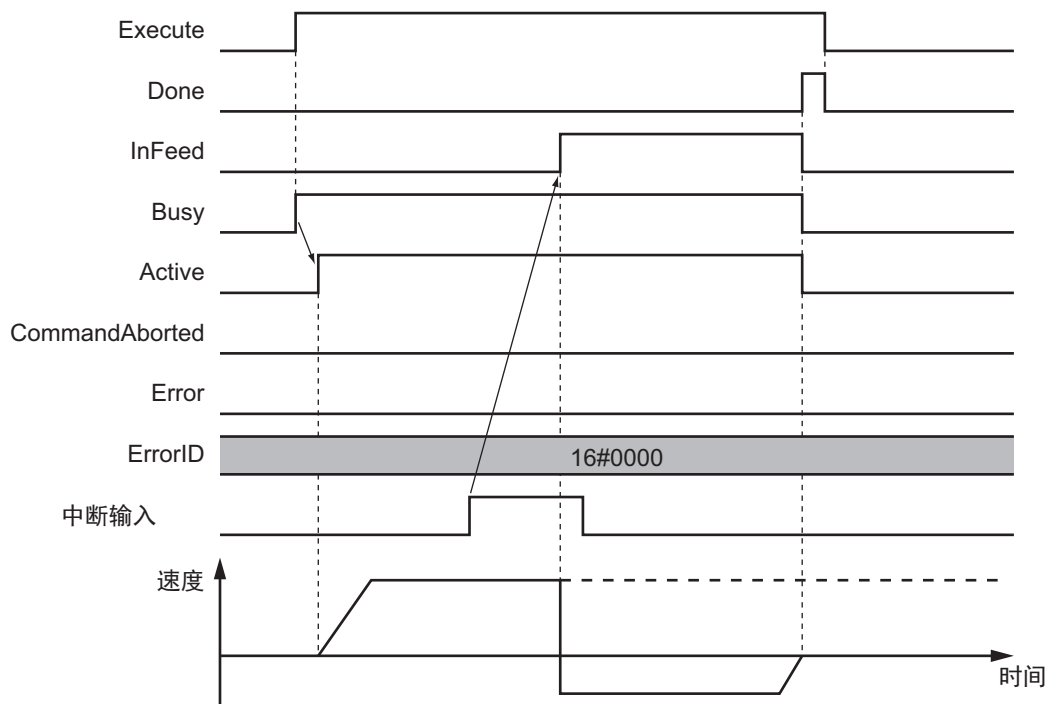
[0: 以反转时减速停止反转] 时



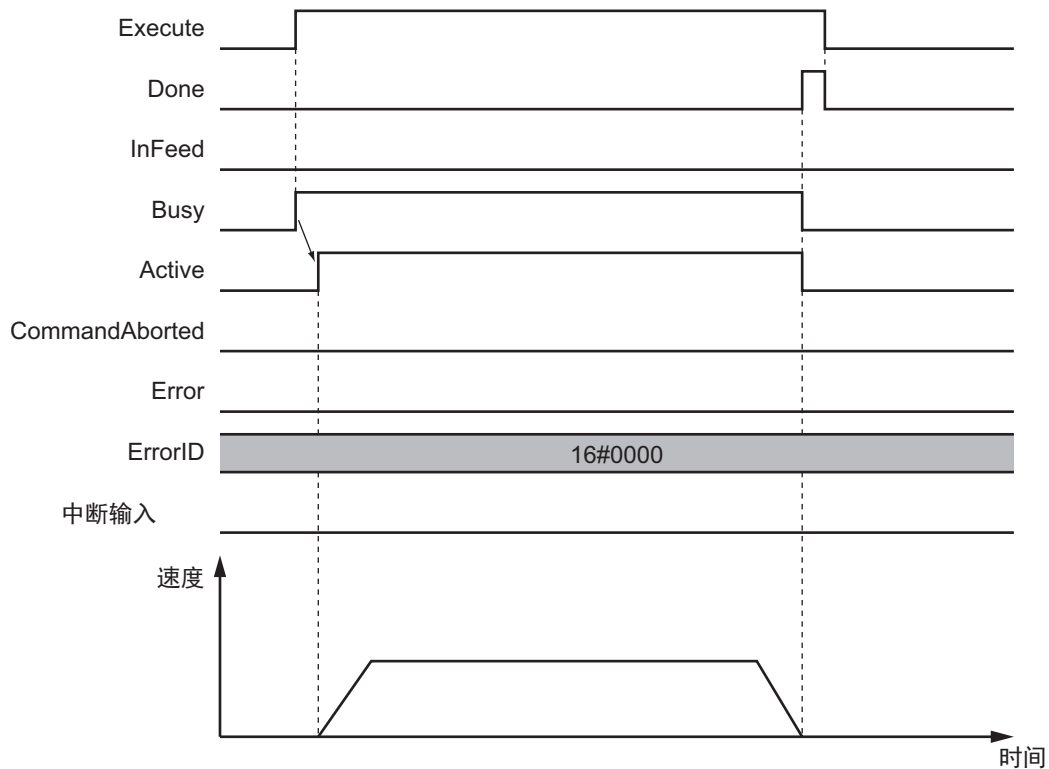
参考

中断输入时的反转减速速度遵照输入变量 Deceleration（减速度）。

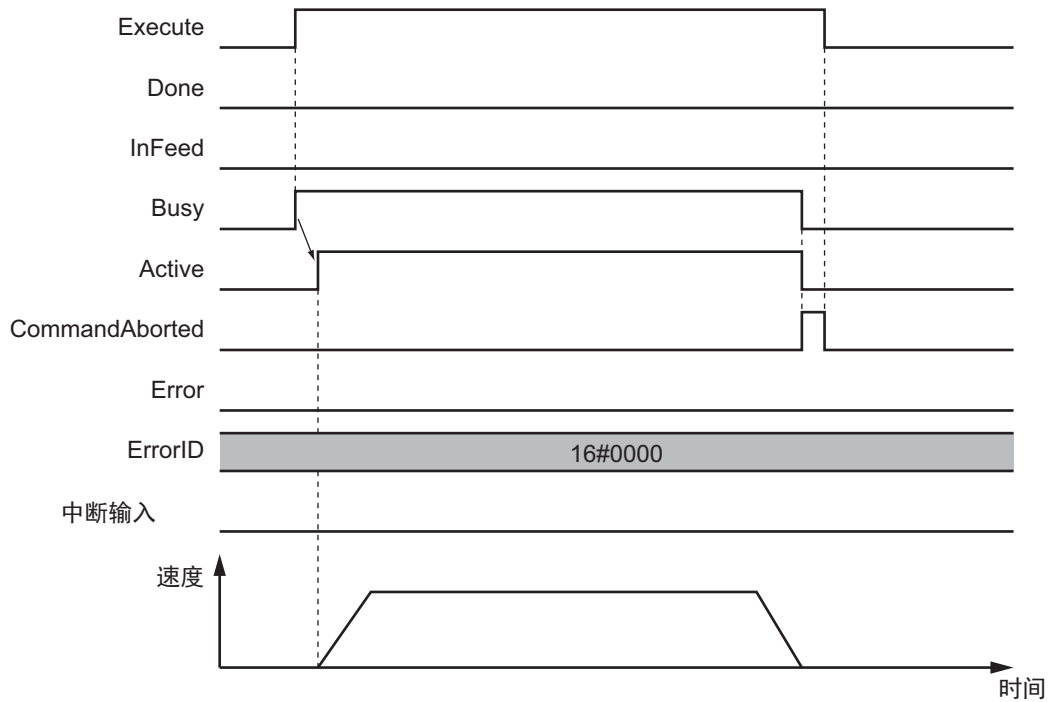
[1: 以反转时立即停止反转] 时

● MoveMode (移动方法选择) 为 `_mcAbsolute`, 无中断输入时

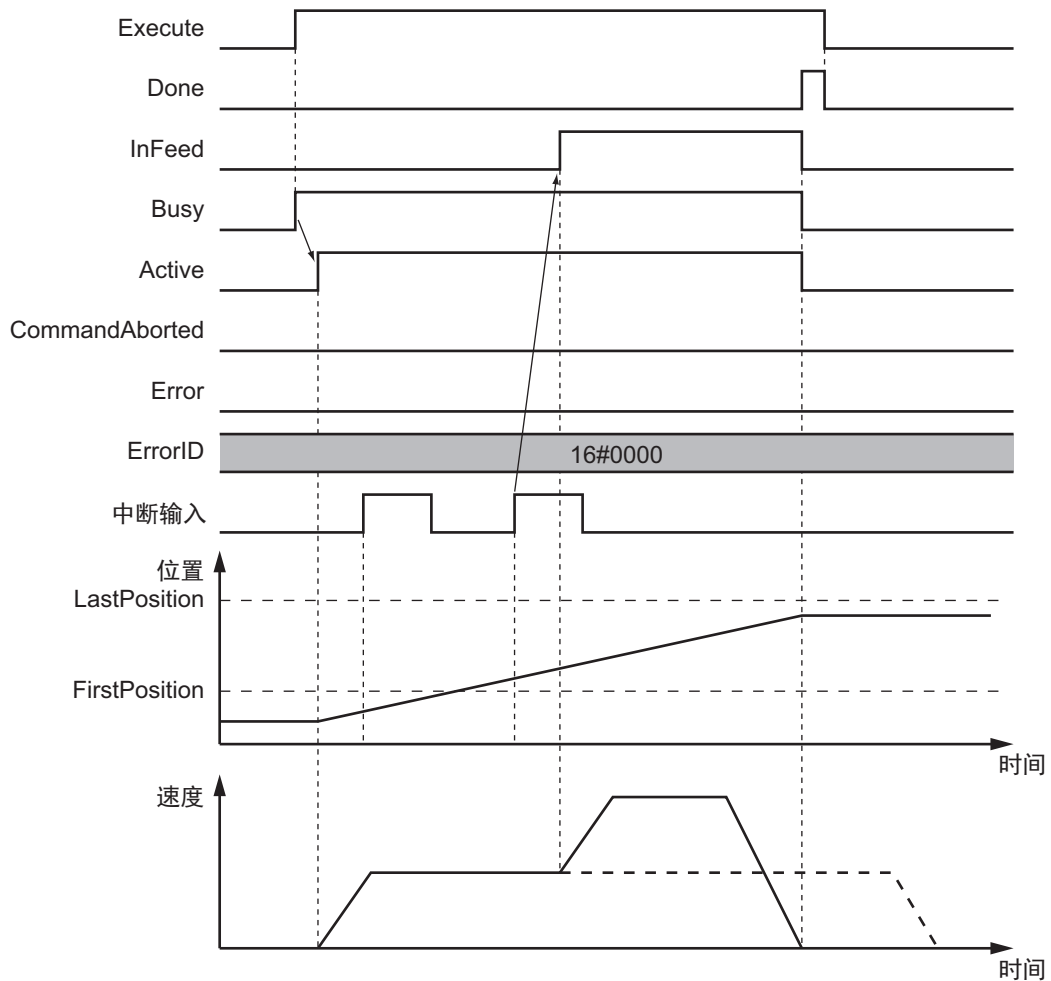
[ErrorDetect (错误检测选择): 无异常检测], 无中断输入时



[ErrorDetect (错误检测选择): 有异常检测], 无中断输入时



● WindowsOnly (窗口有效) 为 [有效] 时



运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

运动指令多重启动的详情请参考 □□《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 其他指令执行过程中本指令的启动

对当前正在执行的指令启动本指令后，可切换为本指令或缓冲本指令。

每个轴最多可缓冲 1 个。

在 BufferMode（缓冲模式选择）中指定多重启动时本指令的动作。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 切换指令后如果轴的动作方向发生反转，将按照轴参数 [反转时动作] 执行反转动作。
缓冲	当前正在执行的指令正常完成后，自动启动已缓冲的本指令。

BufferMode（缓冲模式选择）请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 本指令执行过程中其他指令的启动

正在执行本指令时，可通过中断进行多重启动。

通过缓冲、共混进行多重启动时，

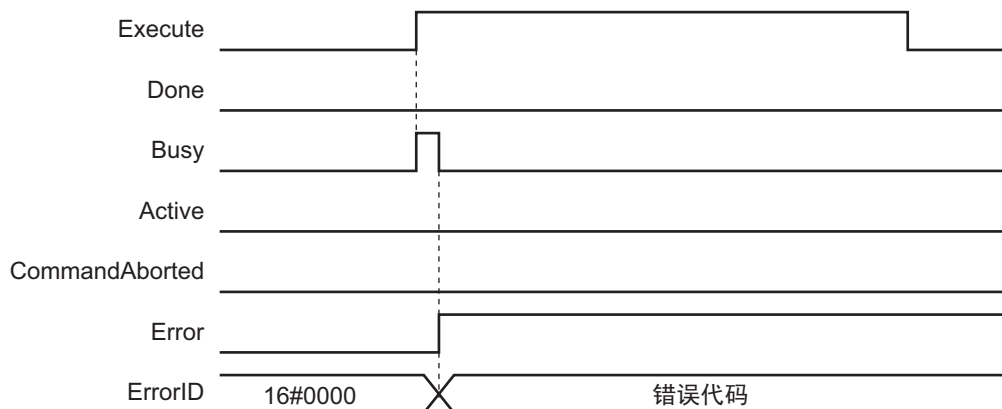
- 执行多重启动的指令的 Error（错误）将变为 TRUE。同时，ErrorID（错误代码）中将输出“运动控制指令不可多重启动（错误代码：543C Hex）”。
- 本指令的处理中止，CommandAborted（执行中断）变为 TRUE。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。
参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图

发生异常时，本指令中使用的中断输入用锁定将变为无效。



● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

示例程序

下面介绍从速度控制变为中断固定尺寸定位的控制相关的示例程序。

参数设定

下面表示本示例程序中使用的最低限度的设定。

● 设定轴参数 轴种类

轴	轴种类
轴 1	伺服轴
轴 2	伺服轴

计数器模式

轴	计数器模式
轴 1	旋转模式
轴 2	直线模式

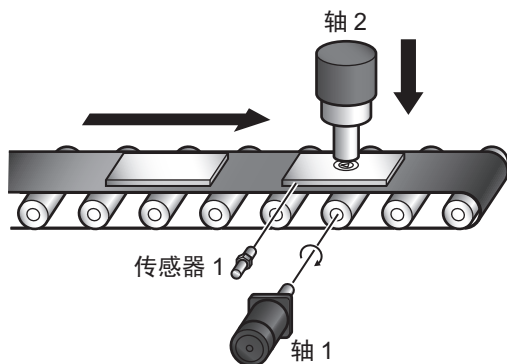
循环计数器

轴	上限值	下限值
轴 1	360	0

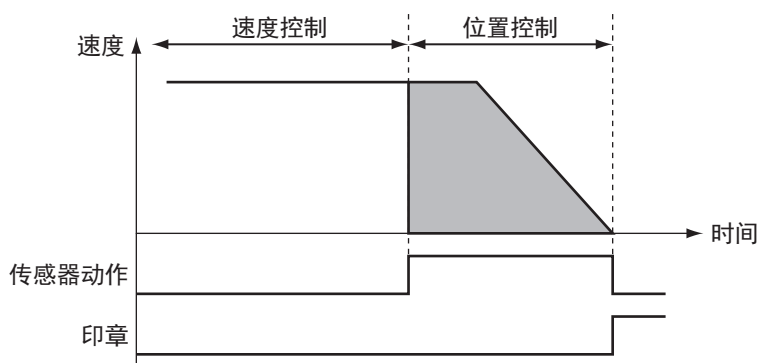
坐标单位选择

轴	坐标单位
轴 1	degree
轴 2	mm

动作示例



● 动作模式



1 输送带动作

使皮带输送机动作的轴 1 按中断固定尺寸定位前的动作进行速度控制。

2 固定尺寸输送

传感器 1 与锁定功能 1 连接。

传感器 1 变为“ON”后，切换为固定尺寸定位，在规定的位置停止。

3 盖戳

定位完成后，图章的轴 2 通过绝对定位垂直下降，进行盖章。

盖章后，轴 2 返回到原点位置。

为了在绝对定位的动作完成的同时，返回到原点位置，需要通过缓冲启动 MC_MoveZeroPosition（高速原点复位）的 BufferMode（缓冲模式选择）。

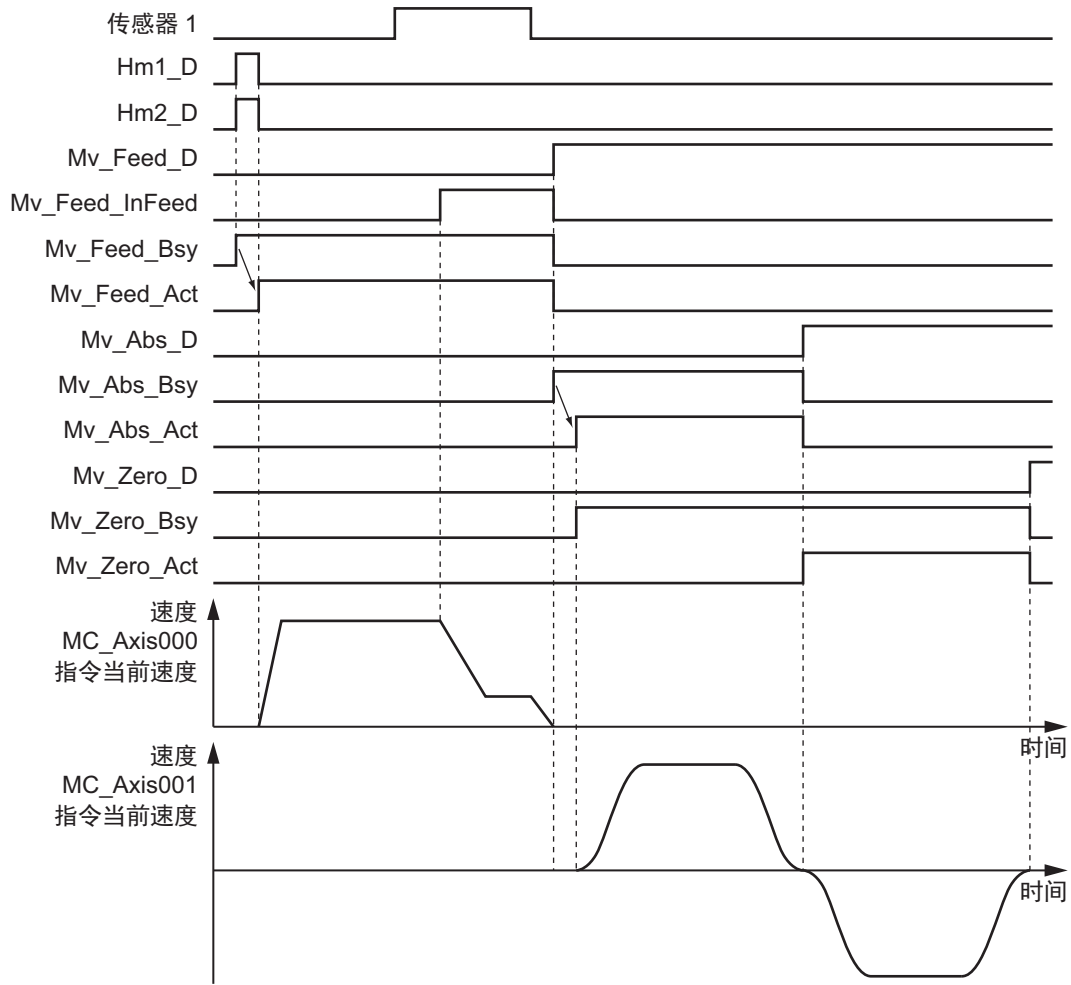
前一指令的输出 Active 为 TRUE 时，进行指令多重启动。

梯形图

● 主要变量

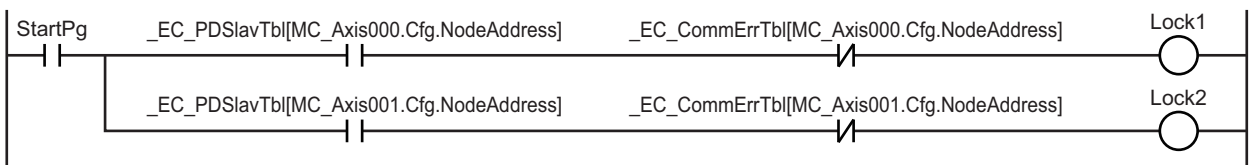
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
MC_Axis001	_sAXIS_REF	—	轴 2 的轴变量。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴 2 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis001.Details.Homed	BOOL	FALSE	轴 2 处于原点确定状态时变为 TRUE。
TrigRef	_sTRIGGER_REF	—	中断输入的指定变量。 在本程序中，使用伺服驱动器的锁定功能 1。检测外部输入（传感器 1）的上升沿，执行固定尺寸移动。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态，则变为伺服 ON 状态。
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE 时，设定输入参数。输入参数设定完成后，变为 TRUE。

● 时序图

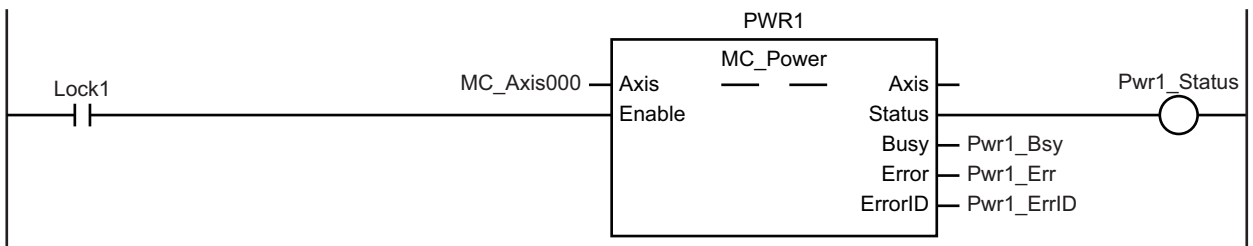


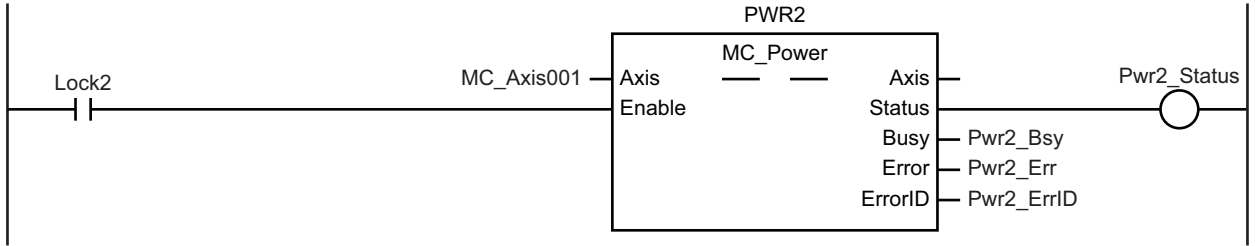
● 示例程序

接点 StartPg 变为 TRUE 后，确认各轴的 EtherCAT 通信是否正常进行过程数据通信

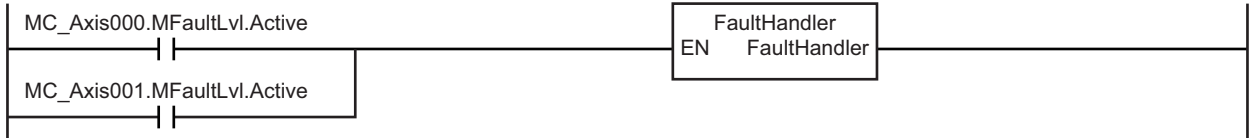


若正在进行过程数据通信，则将各轴设为伺服 ON 状态

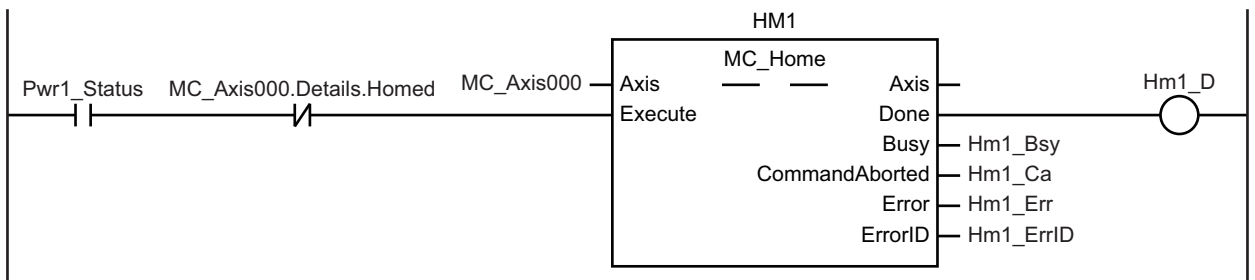




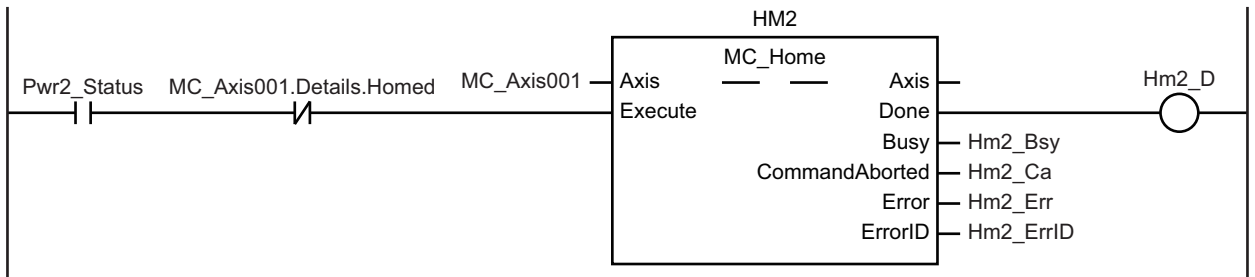
轴构成发生轻度故障等级的异常时，执行装置发生异常时的处理（FaultHandler）。发生异常时的处理（FaultHandler）请根据装置由客户进行编程



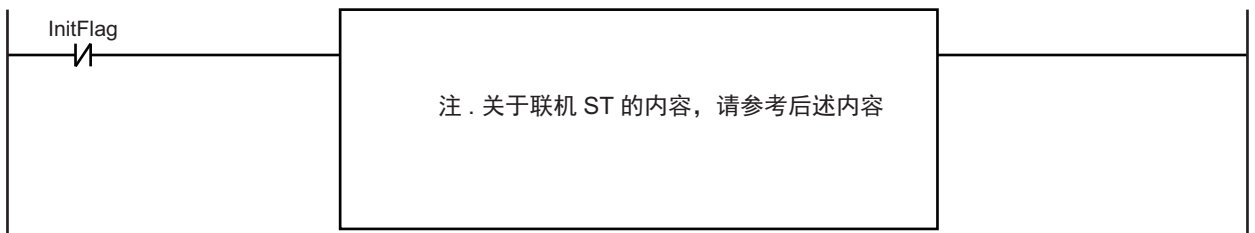
轴 1 变为伺服 ON 状态且处于原点未确定状态时，执行原点复位



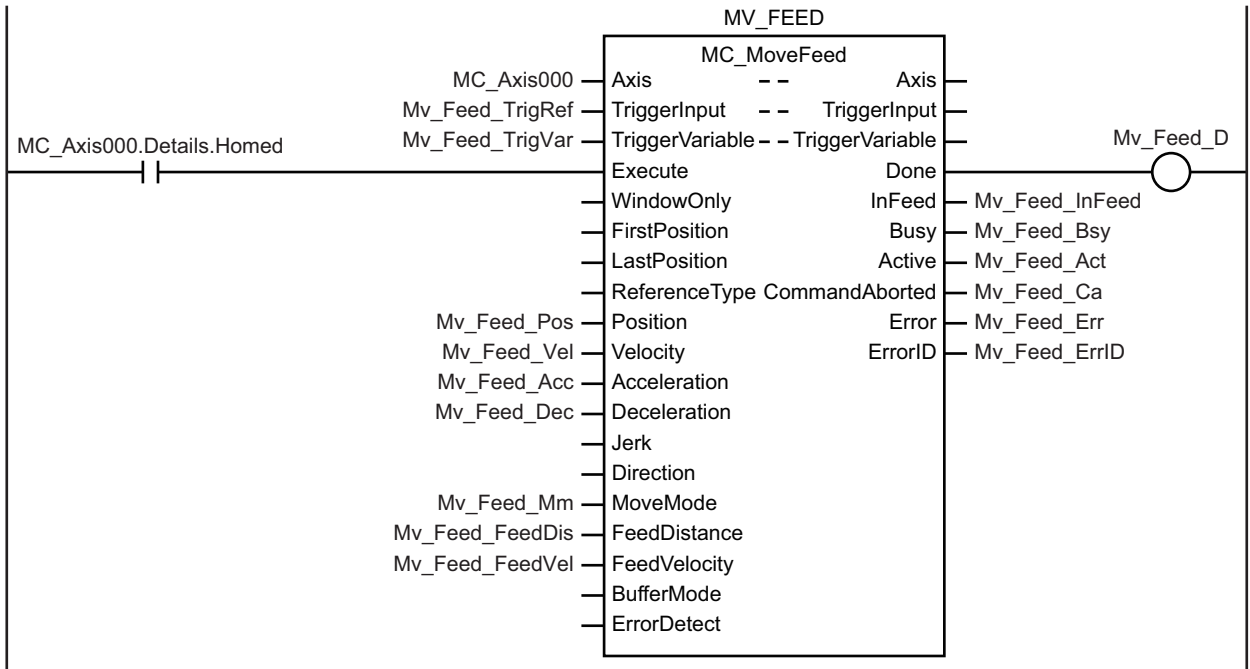
轴 2 变为伺服 ON 状态且处于原点未确定状态时，执行原点复位



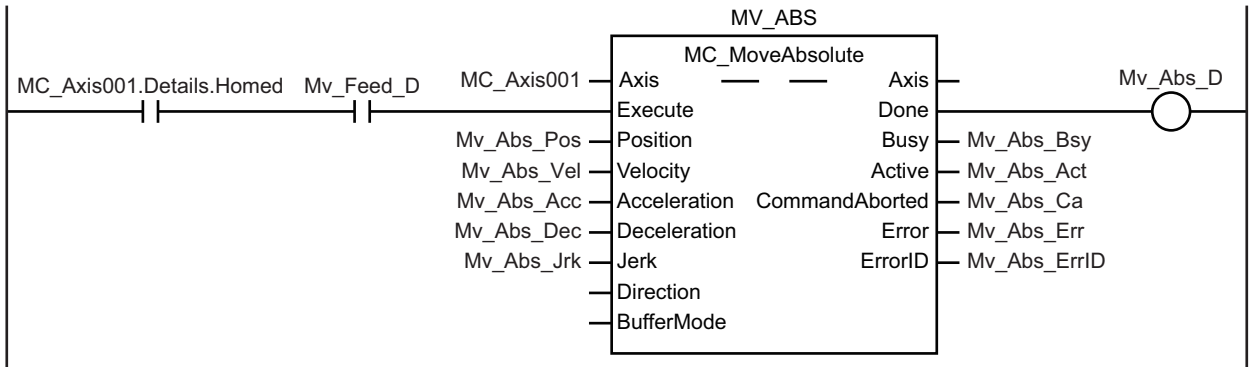
设定中断固定尺寸定位、绝对值定位、高速原点复位的参数



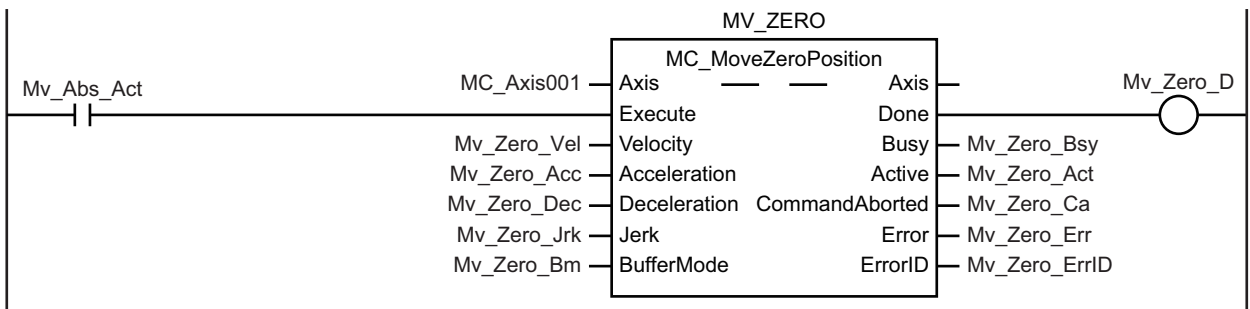
确认轴 1 处于原点确定状态后，对轴 1 执行中断固定尺寸



轴 2 变为原点确定状态，且轴 1 的中断固定尺寸定位完成时，执行轴 2 的绝对值定位



绝对值定位完成后，执行高速原点复位，向原点移动



联机 ST 的内容

```
//MV_FEED 参数
Mv_Feed_TrigRef.Mode      := _eMC_TRIGGER_MODE#_mcDrive;
Mv_Feed_TrigRef.LatchID  := _eMC_TRIGGER_LATCH_ID#_mcLatch1;
Mv_Feed_TrigRef.InputDrive:= _eMC_TRIGGER_INPUT_DRIVE#_mcEXT;
Mv_Feed_TrigVar          := FALSE;
Mv_Feed_Pos              := LREAL#2000.0;
Mv_Feed_Vel              := LREAL#1000.0;
Mv_Feed_Acc              := LREAL#10000.0;
Mv_Feed_Dec              := LREAL#10000.0;
Mv_Feed_Mm               := _eMC_MOVE_MODE#_mcVelocity;
Mv_Feed_FeedDis          := LREAL#500.0;
Mv_Feed_FeedVel          := LREAL#500.0;
```

```
//MV_ABS 参数
Mv_Abs_Pos   := LREAL#1000.0;
Mv_Abs_Vel   := LREAL#500.0;
Mv_Abs_Acc   := LREAL#10000.0;
Mv_Abs_Dec   := LREAL#10000.0;
Mv_Abs_Jrk   := LREAL#10000.0;
```

```
//MV_ZERO 参数
Mv_Zero_Vel  := LREAL#500.0;
Mv_Zero_Acc  := LREAL#10000.0;
Mv_Zero_Dec  := LREAL#10000.0;
Mv_Zero_Jrk  := LREAL#10000.0;
Mv_Zero_Bm   := _eMC_BUFFER_MODE#_mcBuffered;
```

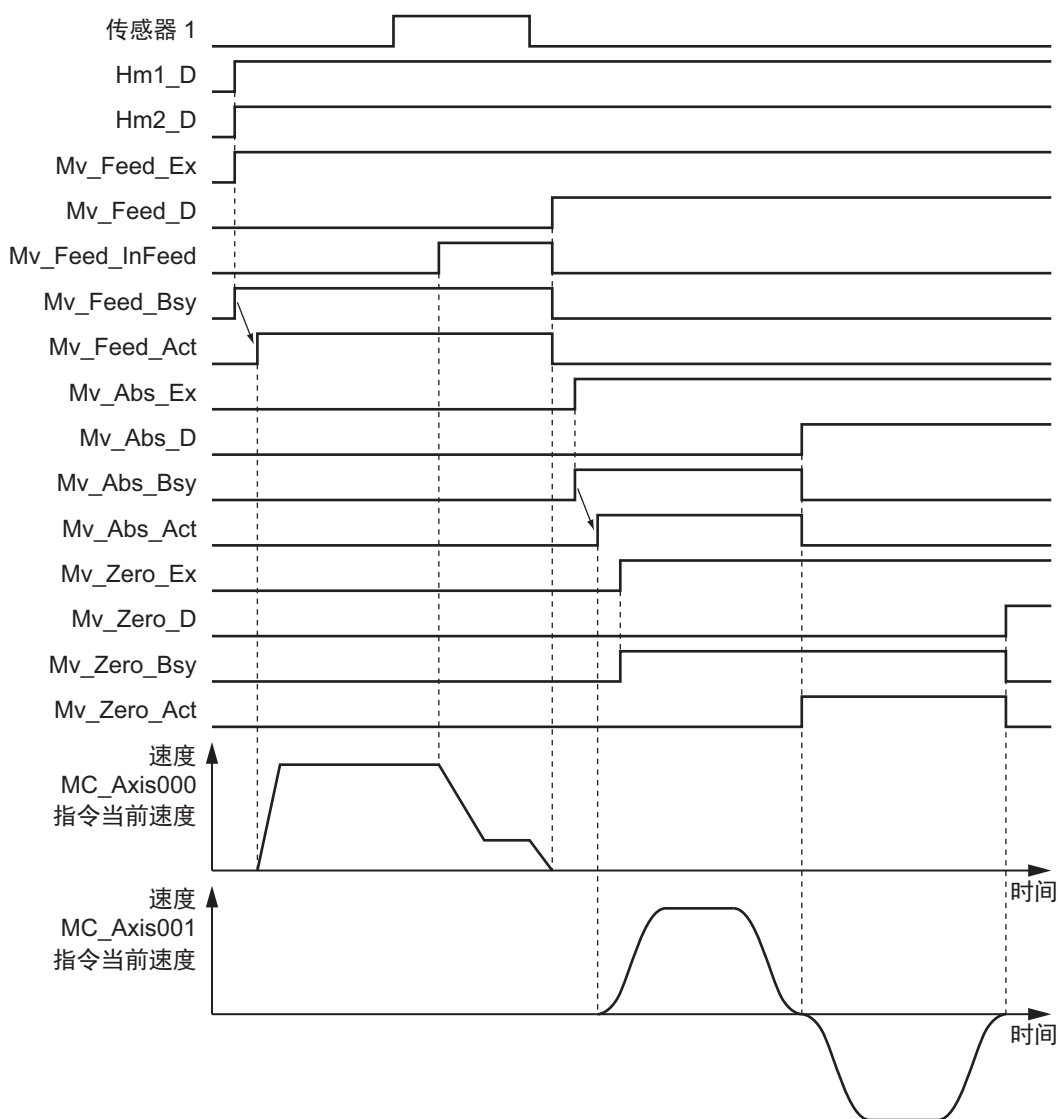
```
// 设定输入参数后，将 InitFlag 设为 TRUE
InitFlag := TRUE;
```

结构文本 (ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时, 变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
MC_Axis001	_sAXIS_REF	—	轴 2 的轴变量。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴 2 发生轻度故障等级的异常时, 变为 TRUE。
MC_Axis001.Details.Homed	BOOL	FALSE	轴 2 处于原点确定状态时变为 TRUE。
TrigRef	_sTRIGGER_REF	—	中断输入的指定变量。 在本程序中, 使用伺服驱动器的锁定功能 1。检测外部输入 (传感器 1) 的上升沿, 执行固定尺寸移动。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态, 则变为伺服 ON 状态。
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE 时, 设定输入参数。输入参数设定完成后, 变为 TRUE。
Hm1_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_Home 的实例 HM1。
Hm2_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_Home 的实例 HM2。
Mv_Feed_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_MoveFeed 的实例 MV_FEED。
Mv_Abs_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_MoveAbsolute 的实例 MV_ABS。
Mv_Zero_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_MoveZeroPosition 的实例 MV_ZERO。

● 时序图



● 示例程序

```
// 未设定输入参数时的处理
IF InitFlag=FALSE THEN
```

```
//MV_FEED 参数
Mv_Feed_Trigger.Mode      := _eMC_TRIGGER_MODE#_mcDrive;
Mv_Feed_Trigger.LatchID  := _eMC_TRIGGER_LATCH_ID#_mcLatch1;
Mv_Feed_Trigger.InputDrive:= _eMC_TRIGGER_INPUT_DRIVE#_mcEXT;
Mv_Feed_Trigger.Var      := FALSE;
Mv_Feed_Pos              := LREAL#2000.0;
Mv_Feed_Vel              := LREAL#1000.0;
Mv_Feed_Acc              := LREAL#10000.0;
Mv_Feed_Dec              := LREAL#10000.0;
Mv_Feed_Mm               := _eMC_MOVE_MODE#_mcVelocity;
Mv_Feed_FeedDis          := LREAL#500.0;
Mv_Feed_FeedVel          := LREAL#500.0;
```

```
//MV_ABS 参数
Mv_Abs_Pos := LREAL#1000.0;
Mv_Abs_Vel := LREAL#500.0;
Mv_Abs_Acc := LREAL#10000.0;
Mv_Abs_Dec := LREAL#10000.0;
```

```

Mv_Abs_Jrk    := LREAL#10000.0;

//MV_ZERO 参数
Mv_Zero_Vel   := LREAL#500.0;
Mv_Zero_Acc   := LREAL#10000.0;
Mv_Zero_Dec   := LREAL#10000.0;
Mv_Zero_Jrk   := LREAL#10000.0;
Mv_Zero_Bm    := _eMC_BUFFER_MODE#_mcBuffered;

// 设定输入参数后, 将 InitFlag 设为 TRUE
InitFlag := TRUE;

END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 1 设为伺服 ON 状态。
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
  Pwr1_En:=TRUE;    // 将轴 1 设为伺服 ON
ELSE
  Pwr1_En:=FALSE;   // 将轴 1 设为伺服 OFF
END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 2 设为伺服 ON 状态。
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
  Pwr2_En:=TRUE;    // 将轴 2 设为伺服 ON
ELSE
  Pwr2_En:=FALSE;   // 将轴 2 设为伺服 OFF
END_IF;

// 发生轻度故障等级异常时的处理
// 发生异常时的处理 (FaultHandler) 请根据装置由客户进行编程。
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
  FaultHandler();
END_IF;

// 轴 1 为伺服 ON 状态且处于原点未确定状态时, 执行轴 1 的原点复位
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
  Hm1_Ex:=TRUE;
END_IF;

// 轴 2 为伺服 ON 状态且处于原点未确定状态时, 执行轴 2 的原点复位
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
  Hm2_Ex:=TRUE;
END_IF;

// 轴 1 变为原点确定后, 执行 MC_MoveFeed
IF MC_Axis000.Details.Homed=TRUE THEN
  Mv_Feed_Ex:=TRUE;
END_IF;

```

```

// 轴 2 变为原点确定状态，且轴 1 的中断固定尺寸定位完成时，执行轴 2 的绝对定位。
IF (MC_Axis001.Details.Homed=TRUE) AND (Mv_Feed_D=TRUE) THEN
  Mv_Abs_Ex := TRUE;
END_IF;

// 开始 MC_MoveAbsolute 后，多重启动 MC_MoveZeroPosition
IF Mv_Abs_Act=TRUE THEN
  Mv_Zero_Ex := TRUE;
END_IF;

// 轴 1 的 MC_Power
PWR1(
  Axis      := MC_Axis000,
  Enable    := Pwr1_En,
  Status    => Pwr1_Status,
  Busy      => Pwr1_Bsy,
  Error     => Pwr1_Err,
  ErrorID   => Pwr1_ErrID
);

// 轴 2 的 MC_Power
PWR2(
  Axis      := MC_Axis001,
  Enable    := Pwr2_En,
  Status    => Pwr2_Status,
  Busy      => Pwr2_Bsy,
  Error     => Pwr2_Err,
  ErrorID   => Pwr2_ErrID
);

// 轴 1 的 MC_Home
HM1(
  Axis          := MC_Axis000,
  Execute       := Hm1_Ex,
  Done          => Hm1_D,
  Busy          => Hm1_Bsy,
  CommandAborted => Hm1_Ca,
  Error         => Hm1_Err,
  ErrorID       => Hm1_ErrID
);

// 轴 2 的 MC_Home
HM2(
  Axis          := MC_Axis001,
  Execute       := Hm2_Ex,
  Done          => Hm2_D,
  Busy          => Hm2_Bsy,
  CommandAborted => Hm2_Ca,
  Error         => Hm2_Err,
  ErrorID       => Hm2_ErrID
);

```

```

//MC_MoveFeed
MV_FEED(
  Axis           := MC_Axis000,
  TriggerInput   := Mv_Feed_TrigRef,
  TriggerVariable := Mv_Feed_TrigVar,
  Execute        := Mv_Feed_Ex,
  Position       := Mv_Feed_Pos,
  Velocity       := Mv_Feed_Vel,
  Acceleration   := Mv_Feed_Acc,
  Deceleration   := Mv_Feed_Dec,
  MoveMode       := Mv_Feed_Mm,
  FeedDistance   := Mv_Feed_FeedDis,
  FeedVelocity   := Mv_Feed_FeedVel,
  Done           => Mv_Feed_D,
  InFeed        => Mv_Feed_InFeed,
  Busy          => Mv_Feed_Bsy,
  Active        => Mv_Feed_Act,
  CommandAborted => Mv_Feed_Ca,
  Error         => Mv_Feed_Err,
  ErrorID       => Mv_Feed_ErrID
);

```

```

//MC_MoveAbsolute
MV_ABS(
  Axis           := MC_Axis001,
  Execute        := Mv_Abs_Ex,
  Position       := Mv_Abs_Pos,
  Velocity       := Mv_Abs_Vel,
  Acceleration   := Mv_Abs_Acc,
  Deceleration   := Mv_Abs_Dec,
  Jerk           := Mv_Abs_Jrk,
  Done          => Mv_Abs_D,
  Busy          => Mv_Abs_Bsy,
  Active        => Mv_Abs_Act,
  CommandAborted => Mv_Abs_Ca,
  Error         => Mv_Abs_Err,
  ErrorID       => Mv_Abs_ErrID
);

```

```

//MC_MoveZeroPosition
MV_ZERO(
  Axis           := MC_Axis001,
  Execute        := Mv_Zero_Ex,
  Velocity       := Mv_Zero_Vel,
  Acceleration   := Mv_Zero_Acc,
  Deceleration   := Mv_Zero_Dec,
  Jerk           := Mv_Zero_Jrk,
  BufferMode      := Mv_Zero_Bm,
  Done          => Mv_Zero_D,
  Busy          => Mv_Zero_Bsy,
  Active        => Mv_Zero_Act,
  CommandAborted => Mv_Zero_Ca,
  Error         => Mv_Zero_Err,
  ErrorID       => Mv_Zero_ErrID
);

```

MC_Stop

将轴减速停止。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_Stop	强制停止	FB		<pre>MC_Stop_instance (Axis := 《参数》, Execute := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, BufferMode := 《参数》, Done => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。 单位为 [指令单位 /s ²]。*1
Jerk	跃度	LREAL	正数或“0”	0	指定跃度。 单位为 [指令单位 /s ³]。*1
BufferMode	缓冲 模式选择	_eMC_BUFFER_ MODE	0: _mcAborting	0 *2	指定运动指令多重启动时的动作。 0: 中断

*1. 指令单位请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. 如果是有效范围为枚举型的变量初始值，实际不是数值，而是枚举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*1	发生异常时，输出错误代码。16#0000 表示正常。

*1. □□「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	完成减速停止，速度达到“0”时	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	开始了指令时	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他 MC_Stop（强制停止）指令进行指令多重启动（中断），本指令中断时 因发生异常，本指令中止时 正在发生异常时，启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。*1

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*]、_MC1_AX[*]、_MC2_AX[*]）。

功能说明

- 将轴从当前速度减速控制至速度“0”。
- 在 Execute（启动）的上升沿开始减速停止的动作。
- 因 MC_Stop（强制停止）指令的启动，正在动作的指令变为 CommandAborted（执行中断）。



使用注意事项

表示轴状态的 Status.ErrorStop（错误减速停止中）为 TRUE 时，无法启动本指令。
若要在错误减速停止过程中停止轴的动作，请使用 MC_ImmediateStop（立即停止）指令。

指令的详情

下面介绍指令的详情。

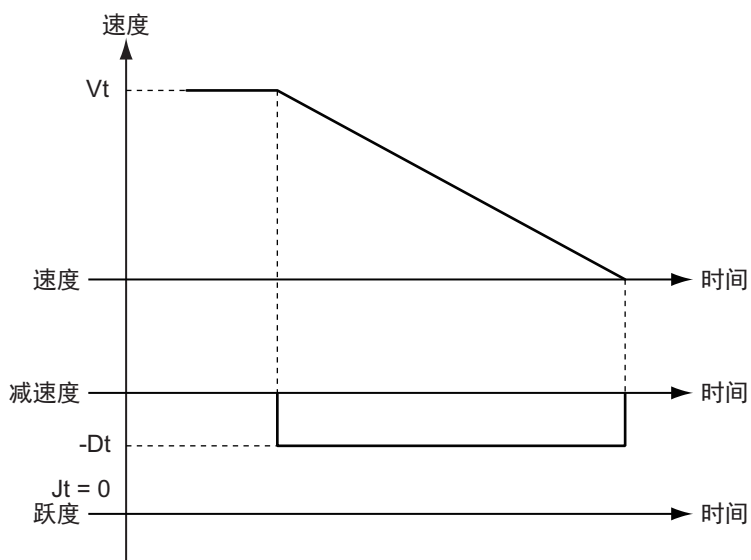
● Deceleration（减速度）、Jerk（跃度）的指定

减速停止时的减速度、跃度在输入变量 Deceleration（减速度）、Jerk（跃度）中指定。

跃度指定为“0”时和指定为“0”以外时的减速度、速度的关系如下所示。

跃度为“0”时

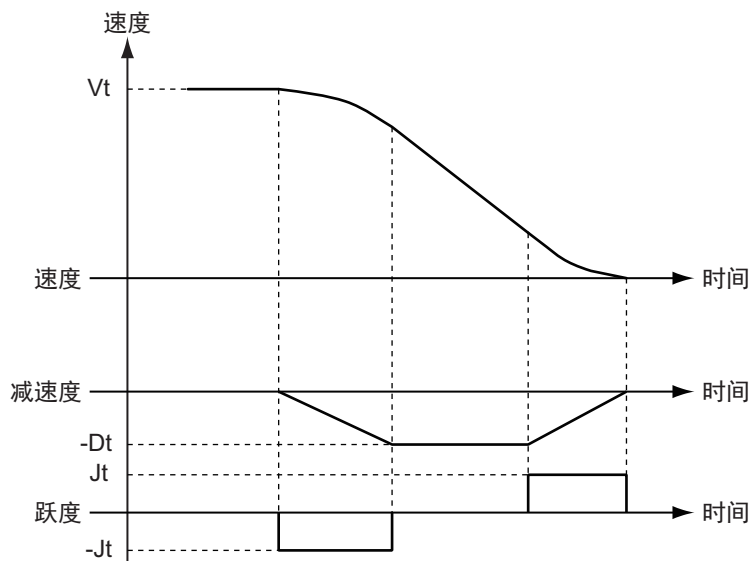
用减速度 Dt 创建速度的指令值。



V_t : 减速启动时的速度、 Dt : 减速度的指定值、 J_t : 跃度的指定值

跃度为“0”以外时

将减速度的上限作为 Dt ，以当前速度为基点，创建速度的指令值。



V_t : 减速启动时的速度、 D_t : 减速度的指定值、 J_t : 跃度的指定值



参考

- 若减速度指定为“0”，则为立即停止指令。不减速，立即停止。仅在减速度指定为“0”时，无论 [加减速超限] 的设置如何，均执行立即停止。

● BufferMode（缓冲模式选择）的指定

指定前一轴动作和本次动作的连接方法。

本指令的 BufferMode（缓冲模式选择）为未来扩展用的保留参数。

为以下 1 种。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，执行本指令。

BufferMode（缓冲模式选择）请参考 □《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

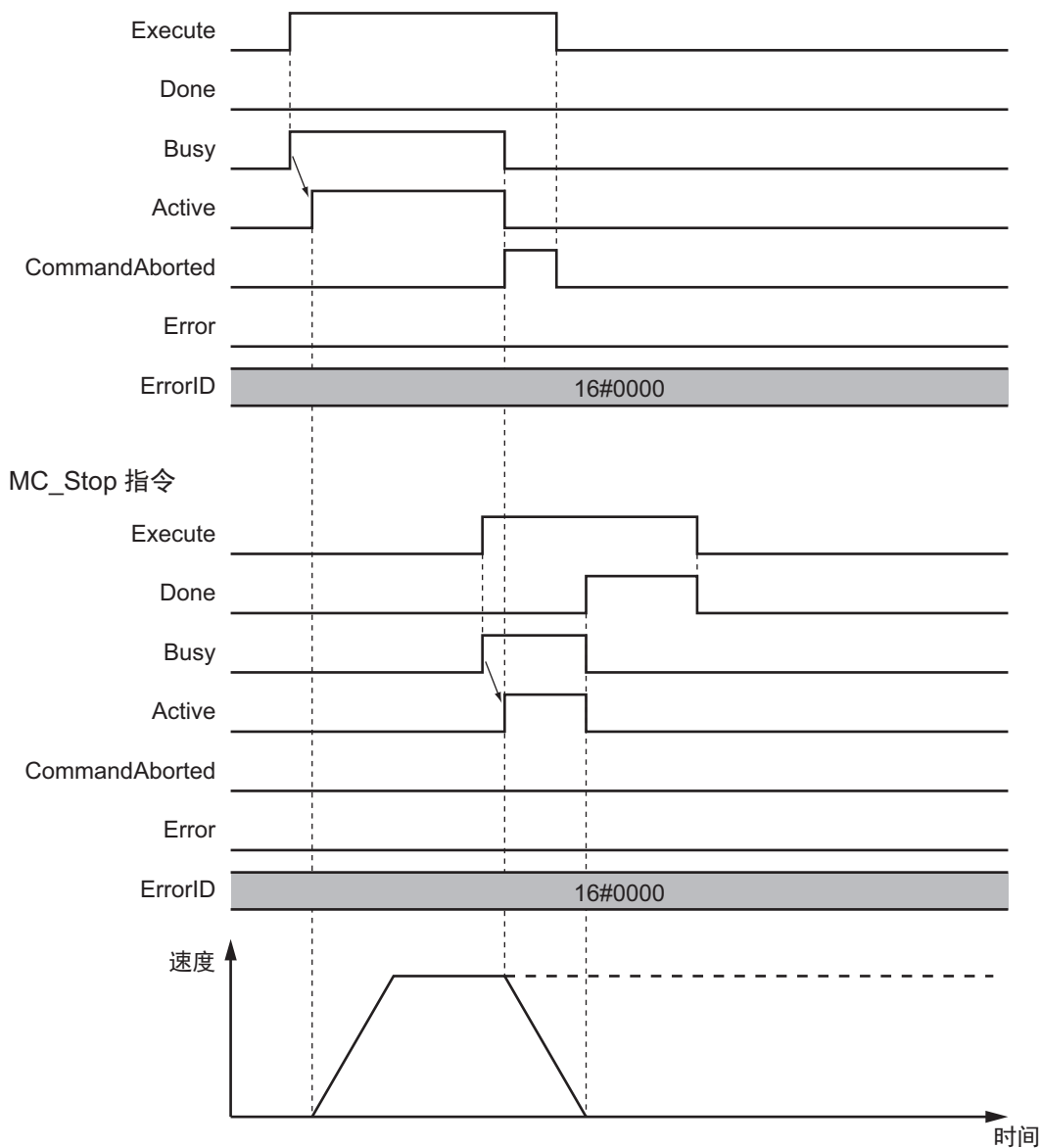
● 位置检查

通过本指令停止时，不进行位置检查。

时序图

- 在 Execute (启动) 上升的同时 Busy (执行中) 变为 TRUE, 在下一周期, Active (控制中) 变为 TRUE。
- 达到速度 (0) 后, Done (完成) 变为 TRUE。
- 通过其他指令中断了本指令时, CommandAborted (执行中断) 变为 TRUE, Busy (执行中)、Active (控制中) 变为 FALSE。

轴的启动指令

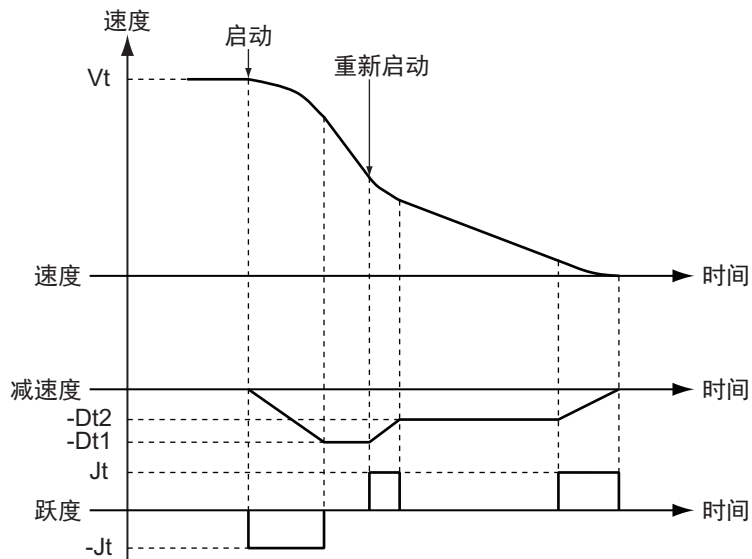


运动指令重新执行

正在执行本指令时，如果再次启动 Execute（启动），将变更 Deceleration（减速度）。
运动指令的重新执行中，不会变更 Jerk（跃度）。

● 跃度为“0”以外时

以当前速度、当前减速度为基点，将变更后的减速度上限作为 Dt2，创建速度的指令值。



V_t : 减速启动时的速度、 Dt : 减速度的指定值、 J_t : 跃度的指定值

运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 其他指令执行过程中本指令的启动

轴组动作中

指定的轴正在执行轴组动作时，若执行 MC_Stop（强制停止）指令，轴将发生异常。
轴组也变为异常，轴组动作停止。

轴变量的 Status.ErrorStop（错误减速停止中）为 TRUE 时

轴发生异常期间，轴变量 Status.ErrorStop（错误减速停止中）变为 TRUE。

若在错误减速停止中，启动了 MC_Stop（强制停止）指令，将无法启动，CommandAborted（执行中断）变为 TRUE。若要停止正在发生异常的轴，请使用 MC_ImmediateStop（立即停止）指令。

MC_ResetFollowingError（偏差计数器复位）指令执行中

如果在执行 MC_ResetFollowingError（偏差计数器复位）指令时启动 MC_Stop（强制停止）指令，MC_ResetFollowingError（偏差计数器复位）指令的 CommandAborted（执行中断）将变为 TRUE，执行 MC_Stop（强制停止）指令。但是，不会适用本指令的 Deceleration（减速度），立即停止。

● 本指令执行过程中其他指令的启动

- 除了通过 MC_Stop (强制停止) 指令进行轴的减速停止外, 启动了以下指令时, 轴变量 Status.Stopping (减速停止中) 变为 TRUE。
 - MC_ResetFollowingError (偏差计数器复位) 指令
 - MC_TouchProbe (外部锁定有效) 指令的立即停止中
 - 其他 MC_Stop (强制停止) 指令的 Execute (启动) 至少有 1 个为 TRUE 时
- 对于 Status.Stopping (减速停止中) 为 TRUE 的轴, 如下所示。
 - 如果进行单轴定位、连续动作、同步动作、手动动作, 这些指令的 CommandAborted (执行中断) 将变为 TRUE。
 - 如果在执行 MC_Stop (强制停止) 指令的过程中启动了 MC_ResetFollowingError (偏差计数器复位) 指令, MC_Stop (强制停止) 指令的 Done (完成) 将变为 TRUE, 执行 MC_ResetFollowingError (偏差计数器复位) 指令。
 - MC_Stop (强制停止) 指令之间可以多重启动。执行中的 MC_Stop (强制停止) 指令的 Done (完成) 变为 TRUE。
- 在启动了 MC_Stop (强制停止) 指令的状态下, 变为以下状态时, MC_Stop (强制停止) 指令的 Done (完成) 将变为 TRUE。
 - 将 MC_Power (可运行) 的输入变量“Enable (有效)”设为 FALSE 时 (设为伺服 OFF 时)
 - 在 MC_TouchProbe (外部锁定有效) 指令的输入变量“StopMode (停止方法选择)”选择为 [1: _mclImmediateStop] 的状态下, 触发条件成立, 欧姆龙生产的伺服驱动器 G5 系列立即停止时

异常

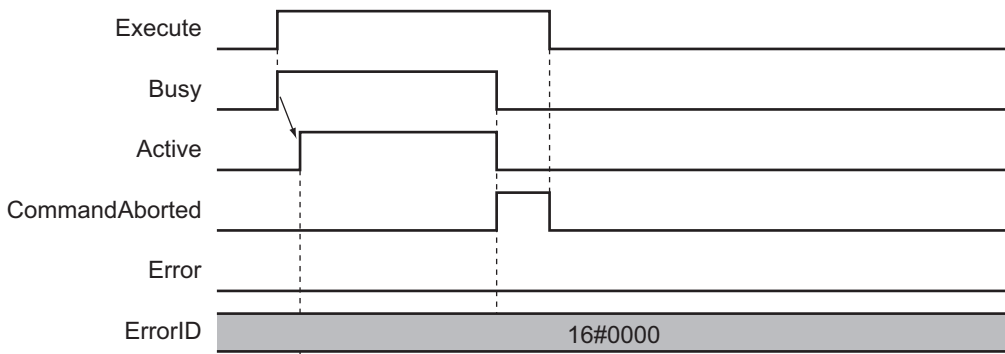
执行本指令的过程中发生轴等异常时，将停止动作。
 通过轴参数指定停止方法，可从立即停止、减速停止、伺服 OFF 中选择。
 停止方法指定为减速停止时，继续进行减速停止动作。

轴参数的停止方法设定请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

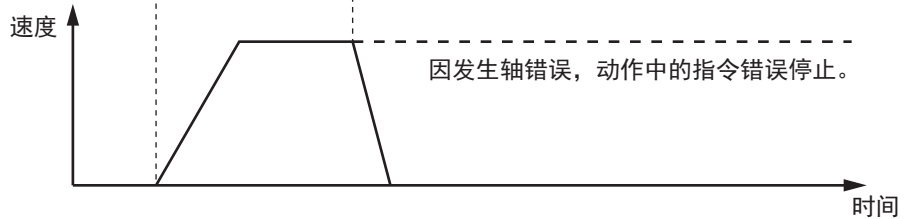
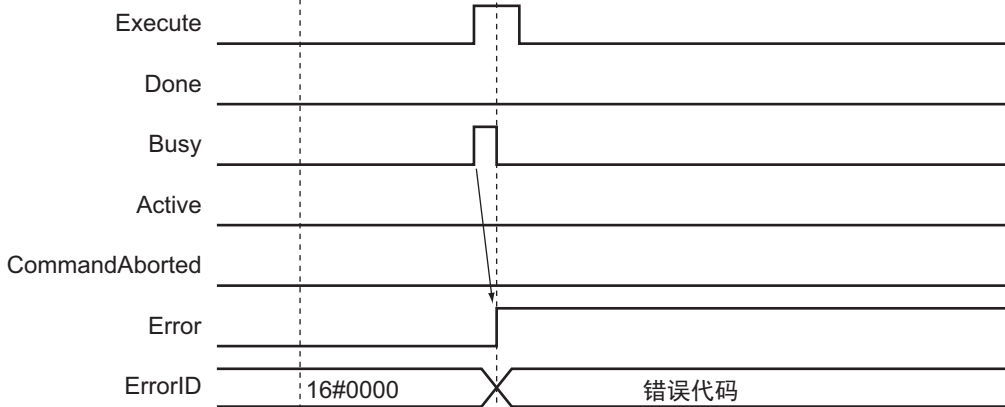
● 发生异常时的时序图

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。
 参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

MC_MoveVelocity



MC_Stop

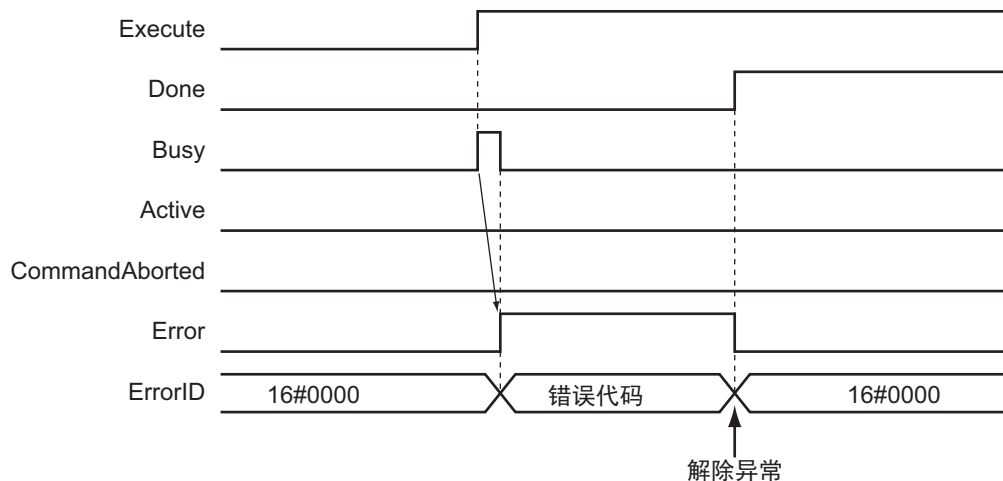


● 异常解除时的动作

在本指令发生异常的状态下，如果在 Execute（启动）为 TRUE 时解除异常，Error（错误）变为 FALSE，Done（完成）变为 TRUE。轴变量 Status.Stopping（减速停止中）正在正常执行减速停止时，同样变为 TRUE。

请在轴停止后再解除异常。轴动作过程中无法解除异常。

MC_Stop



● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

MC_ImmediateStop

与轴状态无关，按照输入变量“StopMode（停止方法选择）”的停止方法停止。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_ImmediateStop	立即停止	FB		<pre>MC_ImmediateStop_instance (Axis := 《参数》, Execute := 《参数》, StopMode := 《参数》, Done => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
StopMode	停止方法选择	_eMC_STOP_MODE	1: _mcImmediateStop 2: _mcImmediateStopFEReset 3: _mcFreeRunStop	1 ^{*1}	选择停止方法。 1: 立即停止 2: 立即停止，进行偏差计数器复位 3: 立即停止，变为伺服 OFF

*1. 如果是有效范围为枚举型的变量初始值，实际不是数值，而是枚举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* □□「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	完成减速停止时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他 MC_Stop (强制停止) 指令进行运动指令多重启动 (中断), 本指令中止时 因发生异常, 本指令中止时 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。*1

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称 (默认“MC_Axis****”)或系统定义变量的轴变量名称 (_MC_AX[*], _MC1_AX[*], _MC2_AX[*])。

功能说明

- 本指令可在任何状态下执行。
例如, 因异常而正在减速停止时, 可以立即停止。
MC_Stop (强制停止) 指令时, 在 Status.ErrorStop (错误减速停止中) 为 TRUE 的状态下无法使用。
但 MC_ImmediateStop (立即停止) 指令可以使用。
- 若执行指令, 将按照 StopMode (停止方法选择) 的指定立即停止, 动作中的指令变为 CommandAborted (执行中断)。
- 若执行本指令, 轴状态的 Status.ErrorStop (错误减速停止中) 将变为 TRUE, 发生“立即停止指令启动 (错误代码: 5485 Hex)”。



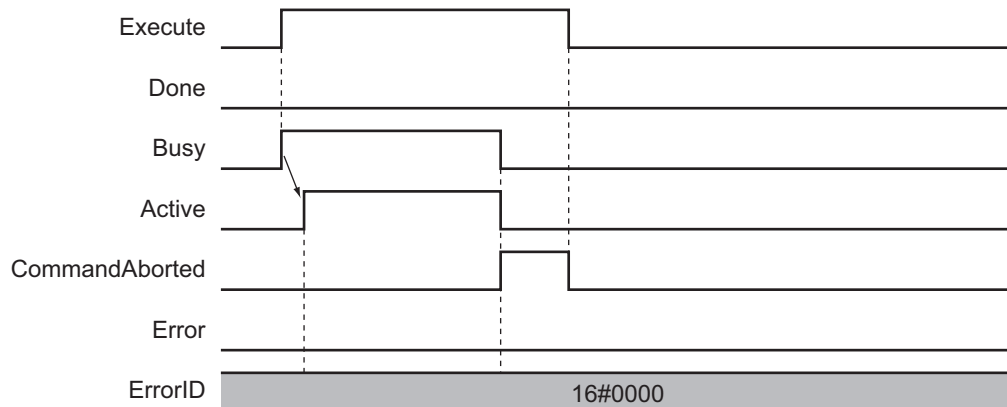
使用注意事项

关于主轴的注意事项, 请参考 □□「同步控制的主轴及辅助轴的注意事项 (P.1-6)」。

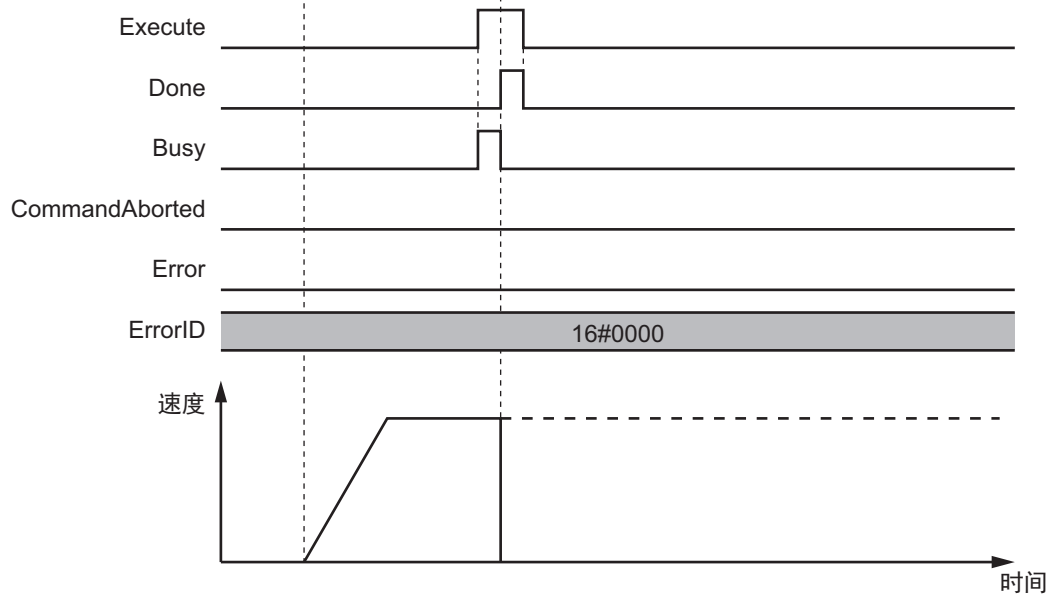
时序图

- 在 Execute（启动）上升的同时 Busy（执行中）变为 TRUE。
- 立即停止指令处理完成后，Done（完成）变为 TRUE。

MC_Move 指令



MC_ImmediateStop 指令



运动指令重新执行

- 无法重新执行本指令。
若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 其他指令执行过程中本指令的启动

轴组动作中

指定的轴正在执行轴组动作时，若执行指令，轴将发生异常，并立即停止。
轴组也变为异常，轴组动作停止。

轴变量的 Status.Stopping（减速停止中）为 TRUE 时

以下状态时，轴变量 Status.Stopping（减速停止中）将变为 TRUE。

- 轴正在根据 MC_Stop（强制停止）指令减速时
- MC_ResetFollowingError（偏差计数器复位）指令执行中
- MC_TouchProbe（外部锁定有效）指令的立即停止中
- MC_Stop（强制停止）指令的 Execute 因多重启动而至少有 1 个为 TRUE 时

对 Status.Stopping（减速停止中）为 TRUE 的轴也可执行本指令。

若启动本指令，以下执行中的指令的 CommandAborted（执行中断）将变为 TRUE。

- MC_Stop（强制停止）指令
- MC_ResetFollowingError（偏差计数器复位）指令
- 立即停止中的 MC_TouchProbe（外部锁定有效）指令

轴变量的 Status.ErrorStop（错误减速停止中）为 TRUE 时

轴发生异常期间，轴状态的 Status.ErrorStop（错误减速停止中）变为 TRUE。
即使因异常而正在减速停止，也可执行本指令。

异常

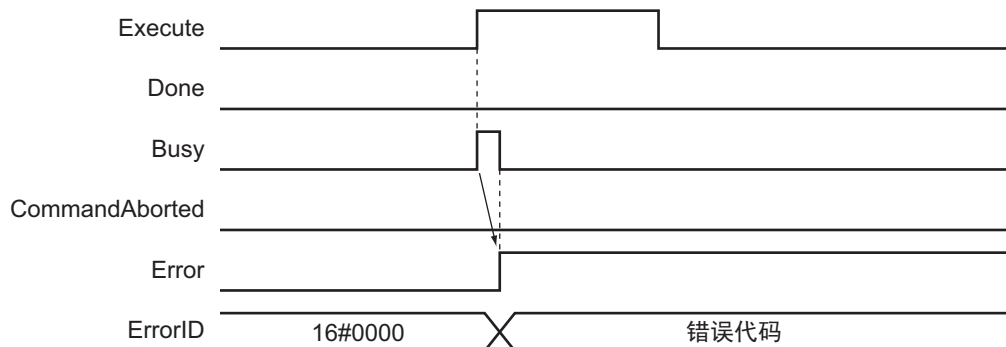
执行本指令的过程中发生轴等异常时，将继续执行立即停止动作。

轴参数的停止方法设定请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 发生异常时的时序图

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴立即停止。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。



● 错误代码

指令中发生的异常请参考 □ 「A-1 错误代码一览 (P.A-2)」。

MC_SetPosition

将轴的指令当前位置和反馈当前位置变更为任意值。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_SetPosition	当前位置变更	FB	<p>The diagram shows a block named 'MC_SetPosition_instance' containing a sub-block 'MC_SetPosition'. The sub-block has several inputs on the left: Axis, Execute, Position, ReferenceType, Relative, and ExecutionMode. It has several outputs on the right: Done, Busy, CommandAborted, Error, and ErrorID.</p>	<pre>MC_SetPosition_instance (Axis := 《参数》, Execute := 《参数》, Position := 《参数》, ReferenceType := 《参数》, Relative := 《参数》, ExcutionMode := 《参数》, Done => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Position	目标位置	LREAL	负数、正数、“0”	0	指定绝对坐标的目标位置。 单位为 [指令单位]。 ^{*1}
ReferenceType	位置种类选择	_eMC_ REFERENCE_ TYPE	0: _mcCommand 1: _mcFeedback	0 ^{*2}	指定位置的类别。 0: 指令位置 (伺服轴、虚拟伺服轴) 1: 反馈位置 (编码器轴、虚拟编码器轴)
Relative (Reserved)	相对位置选择	BOOL	TRUE、FALSE	FALSE	(Reserved)
Execution Mode (Reserved)	执行模式 选择	_eMC_ EXECUTION_ MODE	0: _mclImmediately	0 ^{*2}	(Reserved)

*1. 指令单位请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. 如果是有效范围为列举型的变量初始值，实际不是数值，而是列举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*1	发生异常时，输出错误代码。16#0000 表示正常。

*1. □□「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	指令当前位置和反馈当前位置的变更完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 因发生异常，本指令中止时 正在发生异常时，启动了本指令 正在执行 MC_Stop 指令时，启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。*1

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis***”）或系统定义变量的轴变量名称（_MC_AX[*]、_MC1_AX[*]、_MC2_AX[*]）。

功能说明

- 将伺服轴的指令当前位置变更为指定的目标位置。
此外，对编码器轴执行后，反馈当前位置将变更为指定的目标位置。
- 目标位置以绝对坐标的位置指定。
- 在变更指令当前位置时，会同时变更反馈当前位置，但变更时会保持当时的位置偏差。对指令伺服轴执行后，将在保持当前位置和反馈当前位置差的状态下进行变更。
因此，执行本指令后，轴的反馈当前位置为以下计算式所示的值。

变更后的反馈当前位置 = 目标位置 - 变更前的位置偏差

- 对伺服轴执行了反馈位置时，或对编码器轴指定了指令位置时，将发生位置种类选择异常。
- 计数模式为 [旋转模式] 时，目标位置在循环计数器下限值以上、循环计数器上限值以下的范围内设定。
如果目标位置超出该范围，将发生循环计数器异常。
- 计数模式为 [直线模式] 时，即使目标位置超出软件限位的范围，也可设定。
- 本指令在轴停止和轴动作的状态下均可使用。



使用注意事项

本指令完成后，指定的轴将变为原点未确定状态。
因此，本指令完成后，将无法使用以下功能和指令，敬请注意。

- 软件限位功能
- MC_MoveZeroPosition（高速原点复位）指令
- 多轴协调动作指令（直线插补和圆弧插补）

● ReferenceType（位置种类选择）

- 使用伺服轴或虚拟伺服轴时，请选择“0: _mcCommand（指令位置）”。
- 使用编码器轴或虚拟编码器轴时，请选择“1: _mcFeedback（反馈位置）”。

● 轴种类和位置种类的关系

可监视的轴种类和要监视的位置种类的关系如下所示。

轴种类	ReferenceType	
	_mcCommand	_mcFeedback
伺服轴	○	×
编码器轴	× *1	○
虚拟伺服轴	○	×
虚拟编码器轴	× *1	○

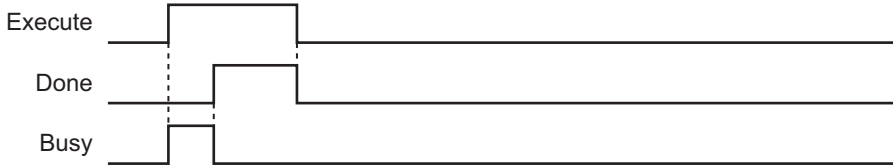
*1. 指令启动时，将发生“超出位置种类选择范围（错误代码：5430 Hex）”。

时序图

● 在轴停止的状态下使用本指令时

在 Execute（启动）的上升沿开始当前位置变更。

在 Execute（启动）上升的同时 Busy（执行中）变为 TRUE，完成当前位置的变更后，Done（完成）变为 TRUE。



● 在轴正在动作的状态下使用本指令时

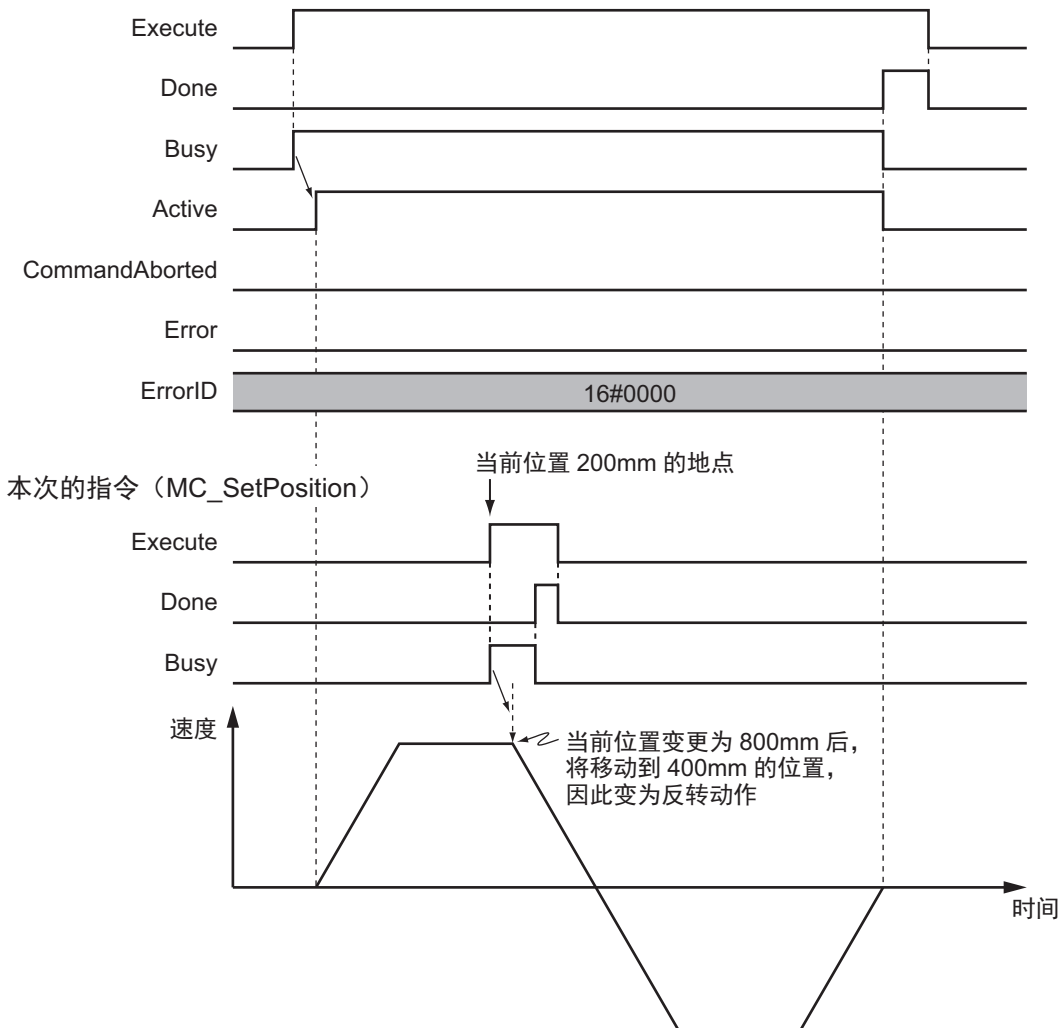
若在向绝对位置执行定位时执行了本指令，因位置的变更，动作的目标值也会变更。

MC_MoveAbsolute（绝对值定位）指令、向 400（mm）移动的过程中，当前位置 200（mm）变更为 800（mm）时的轴动作和时序图如下所示。

此时，当前值变为 800（mm），目标值为 400（mm），因此轴将反转。

如下图所示，即使变更当前位置，MC_MoveAbsolute（绝对值定位）指令仍将从新的当前位置定位到 MC_MoveAbsolute（绝对值定位）指令中指定的目标位置，然后 Done（完成）变为 TRUE。

前一指令（Ex. MC_MoveAbsolute）



**参考**

- MC_MoveRelative（相对值定位）指令中或 MC_MoveVelocity（速度控制）指令中执行了本指令时，当前位置将发生变更。但是，不会影响 MC_MoveRelative（相对值定位）指令或 MC_MoveVelocity（速度控制）指令的动作。
- 有缓冲的指令时，切换缓冲时，将对变更后的位置进行定位。

运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

- **其他指令执行过程中本指令的启动**

MC_SetPosition（当前位置变更）指令不可对正在执行以下指令的轴执行。

若执行，将发生多重启动异常。

MC_MoveJog（点动进给）指令	MC_MoveLink（梯形模式凸轮）指令
MC_Home（原点复位）指令	MC_CombineAxes（加减法定位）指令
MC_HomeWithParameter（参数指定原点复位）指令	MC_MoveZeroPosition（高速原点复位）指令
MC_CamIn（凸轮动作开始）指令	MC_TorqueControl（扭矩控制）指令
MC_GearIn（齿轮动作开始）指令	MC_SyncMoveVelocity（周期性同步速度控制）指令
MC_GearInPos（位置指定齿轮动作）指令	

**使用注意事项****单元版本 1.09 以下的 CPU 单元时**

MC_SetPosition（当前位置变更）指令请勿对 MC_GearIn（齿轮动作开始）指令等同步指令的 Master（主轴）使用。

如果对 Master（主轴）使用，在 Master（主轴）的指令当前位置和反馈当前位置发生变化时，Slave（从轴）将判断为 Master（主轴）发生移动，将根据 Master（主轴）的移动量执行动作。因此，Slave（从轴）的动作会发生剧烈变化，或凸轮动作结束。

不对 Master（主轴）使用 MC_SetPosition（当前位置变更）指令时，请先解除 Master（主轴）和 Slave（从轴）的关系后再执行。

要解除 Master（主轴）和 Slave（从轴）的关系，请执行 MC_GearOut（齿轮动作解除）指令等。

关于主轴的注意事项，请参考 □ 「同步控制的主轴及辅助轴的注意事项 (P.1-6)」。

● 本指令执行过程中其他指令的启动

正在执行 MC_SetPosition（当前位置变更）指令时，若执行其他 MC_SetPosition（当前位置变更）指令，将优先执行后面的 MC_SetPosition（当前位置变更）指令。

此时，刚刚正在执行的 MC_SetPosition（当前位置变更）指令的 Done（完成）将变为 TRUE，但尚未变更为前面设定的位置，敬请注意。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。



参考

若在轴动作过程中执行了本指令，当前动作中的轴指令将视为重新执行，再次计算定位所需的数据。

此时，若发生异常，本指令不会变为异常，当前动作中的轴指令变为异常。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

MC_SetOverride

变更轴的目标速度。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_SetOverride	超驰值设定	FB		<pre>MC_SetOverride_instance (Axis := 《参数》, Enable := 《参数》, VelFactor := 《参数》, AccFactor := 《参数》, JerkFactor := 《参数》, Enabled => 《参数》, Busy => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、 FALSE	FALSE	若设定为 TRUE，则超驰变为有效。 若设定为 FALSE，则超驰恢复为“100%”。
VelFactor	速度 超驰值	LREAL	0 ~ 500	100	指定速度的超驰值。 超驰值的有效范围为“0.01 ~ 500.00”。 “500.00 以上”视为“500”，“0.01 以下（包括负数）”视为“0.01”。 仅在指定为“0”时作为“0”动作。 单位为 [%]。
AccFactor (Reserved)	加减速速度 超驰值	LREAL	0 ~ 500	100	(Reserved)
JerkFactor (Reserved)	跃度 超驰值	LREAL	0 ~ 500	100	(Reserved)

输出变量

输出变量	名称	数据类型	有效范围	内容
Enabled	有效	BOOL	TRUE、 FALSE	控制过程中变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。 16#0000 表示正常。

* □「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Enabled	开始了本指令的执行时	<ul style="list-style-type: none"> • Enable 为 FALSE 时，1 个周期后 • Error 变为 TRUE 时
Busy	Enable 的上升沿	<ul style="list-style-type: none"> • Enable 变为 FALSE 时 • Error 变为 TRUE 时
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

功能说明

- 变更与轴的目标速度对应的超驰。变更超驰后，可变更动作中的目标速度。
- 可变更超驰的指令为以下指令。

MC_Move（定位）指令	MC_MoveJog（点动进给）指令
MC_MoveAbsolute（绝对值定位）指令	MC_MoveFeed（中断固定尺寸定位）指令
MC_MoveRelative（相对值定位）指令	MC_MoveZeroPosition（高速原点复位）指令
MC_MoveVelocity（速度控制）指令	

- 新的目标速度如下所示。

变更后的目标速度 = 当前执行中指令的目标速度 × 超驰

- 超驰的单位为 [%]。“100”以“100%”表示。
- 变更后的目标速度超过轴参数 [最高速度] 时，为最高速度。
- 轴将针对变更后的目标速度，进行加速或减速。
- 超驰指定为“0”时，目标速度变为“0”，轴的动作减速，速度降至“0”后，变为动作中。
要在保持动作状态的同时暂停轴动作时，将超驰设为“0”。
此时，轴变量 Status.Discrete、Status.Continuous 不会变化。
- 超驰值在 Enable（有效）为 TRUE 的状态下，将在执行指令时始终反映。
- 超驰值将在对动作指令停止时的启动、运动指令重新执行、运动指令多重启动时，应用到设定的新目标速度中。
- Enable（有效）为 FALSE 时，超驰恢复为 100 (=100%)。
- 执行 MC_SetOverride（超驰值设定）指令的过程中发生轴异常时，MC_SetOverride（超驰值设定）指令的 Enabled（有效）将继续保持 TRUE。



使用注意事项

若将本指令的 Enable（有效）设为 FALSE，本指令的 Enabled（有效）及 Busy（执行中）将变为 FALSE。

此时，速度将向超驰“100%”加速或减速。



参考

对其他指令的影响

本指令在临时变更各指令的目标速度时使用。

因此，输入中不含目标速度的指令或周期性同步速度控制等每个周期变更目标速度的指令中，无法使用本功能。

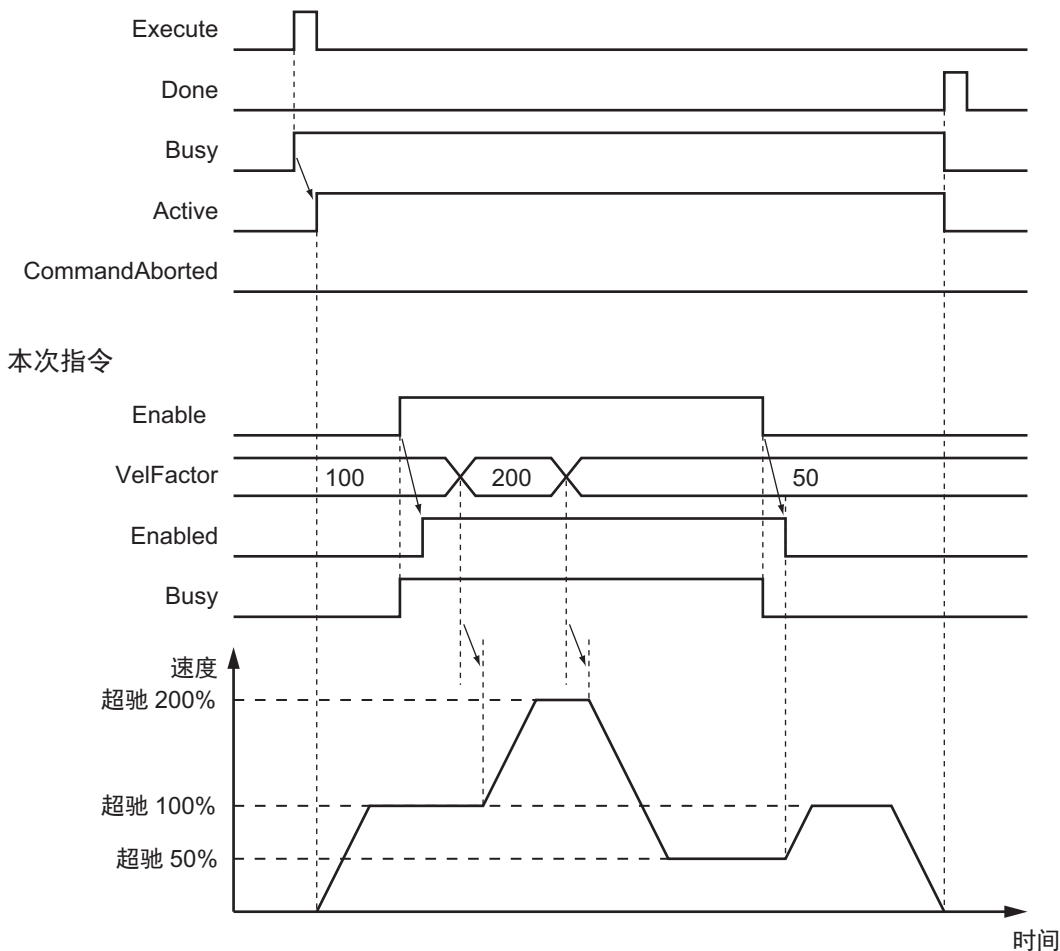
但是，对于无法使用本功能的指令，即使将 MC_SetOverride（超驰值设定）指令设为有效，Enabled（有效）将继续保持 TRUE。

时序图

● MC_MoveAbsolute（绝对值定位）指令的超驰

在 MC_MoveAbsolute（绝对值定位）指令中使用超驰指令时的时序图如下所示。

前一指令（MC_MoveAbsolute）

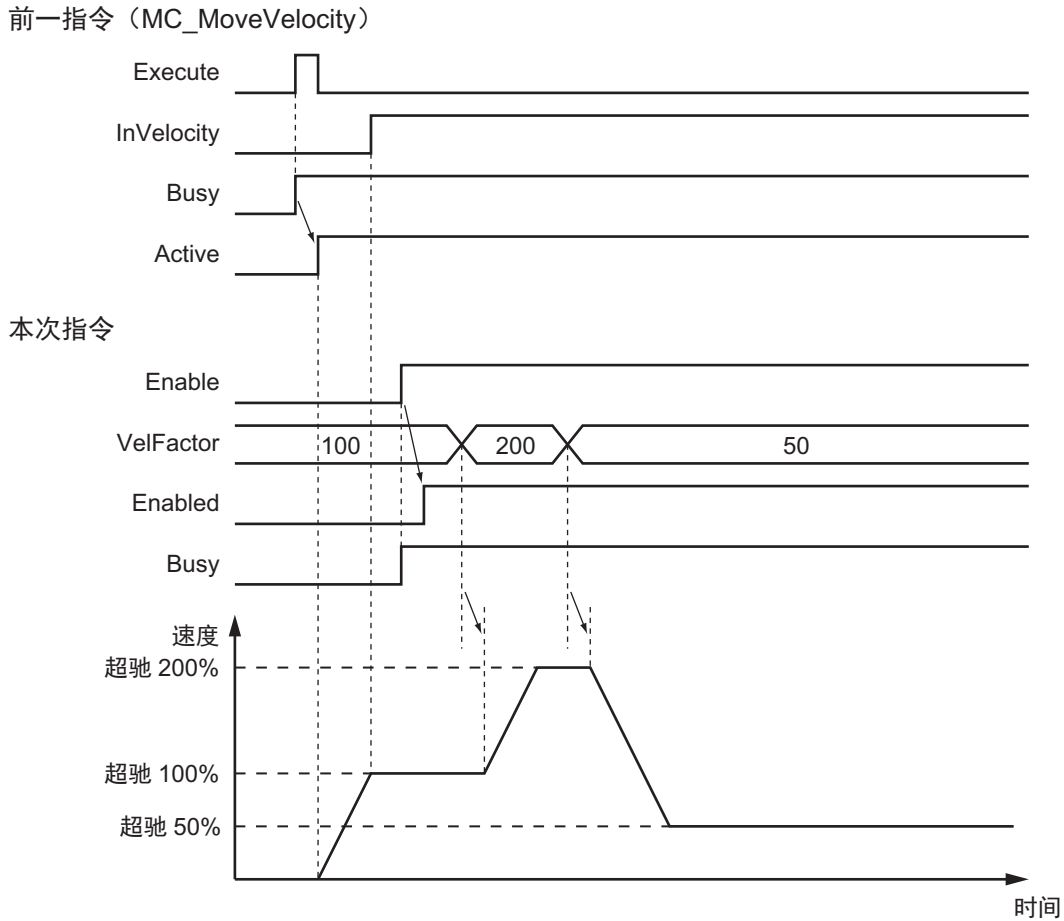


MC_SetOverride（超驰值设定）指令变为无效后，将恢复为超驰 100% 的目标速度。

● 对 MC_MoveVelocity（速度控制）指令的超驰

在 MC_MoveVelocity（速度控制）指令中使用超驰指令时，时序图如下所示。

InVelocity（达到目标速度）变为 TRUE 后，即使变更速度，InVelocity（达到目标速度）也保持 TRUE。



运动指令重新执行

输入为“Enable 型”的指令没有运动指令重新执行。

运动指令多重启动

运动指令多重启动的详情请参考 □□《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 本指令的多重启动

对正在执行 MC_SetOverride（超驰值设定）指令的轴启动了其他实例的 MC_SetOverride（超驰值设定）指令时，后面执行的实例将优先处理。

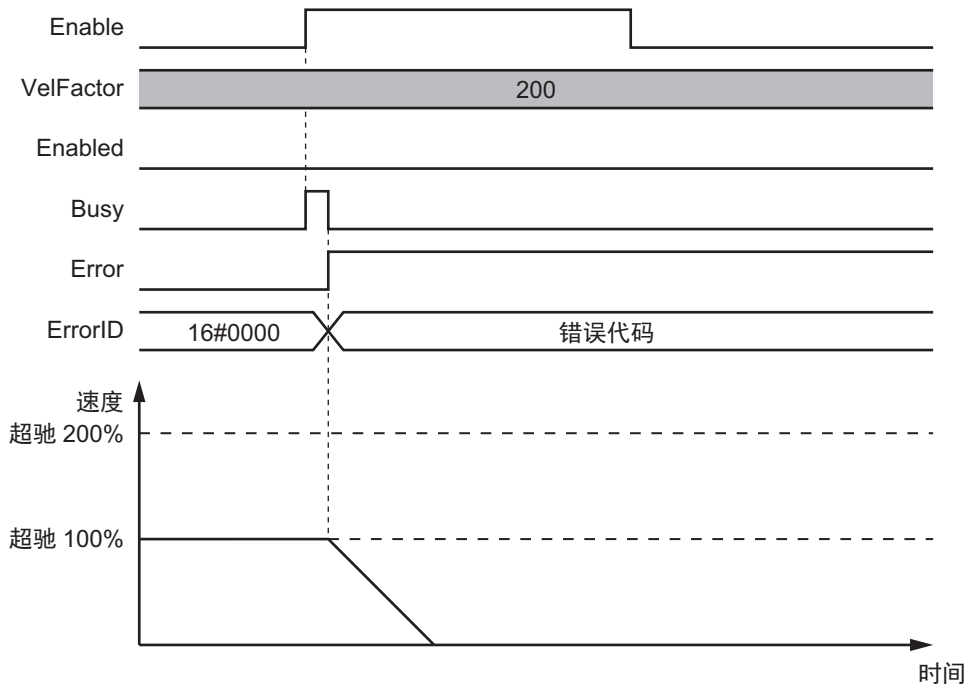
Enabled（有效）时，两者指令均变为 TRUE。

具体动作为，后面执行的实例的超驰值变为有效。若将后面执行的实例的 Enable（有效）设为 FALSE，超驰将变为无效。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，发生轻度故障时，轴停止。
 参照输出到 ErrorID（错误代码）的值，可找出异常的原因。
 排除异常原因后，Error（错误）将变为 FALSE。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 [「A-1 错误代码一览 \(P.A-2\)」](#)。

MC_ResetFollowingError

重置指令当前位置和反馈当前位置的偏差。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_ResetFollowingError	偏差计数器复位	FB		<pre>MC_ResetFollowingError_instance (Axis := 《参数》, Execute := 《参数》, BufferMode := 《参数》, Done => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
BufferMode	缓冲模式选择	_eMC_BUFFER_MODE	0: _mcAborting	0 *1	指定运动指令多重启动时的动作。 0: 中断

*1. 如果是有效范围为枚举型的变量初始值，实际不是数值，而是列举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*1	发生异常时，输出错误代码。16#0000 表示正常。

*1. □□「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	偏差计数器复位完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	开始了指令时	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他 MC_ResetFollowingError 指令进行运动指令多重启动 (中断), 本指令中断时 因发生异常, 本指令中止时 正在发生异常时, 启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

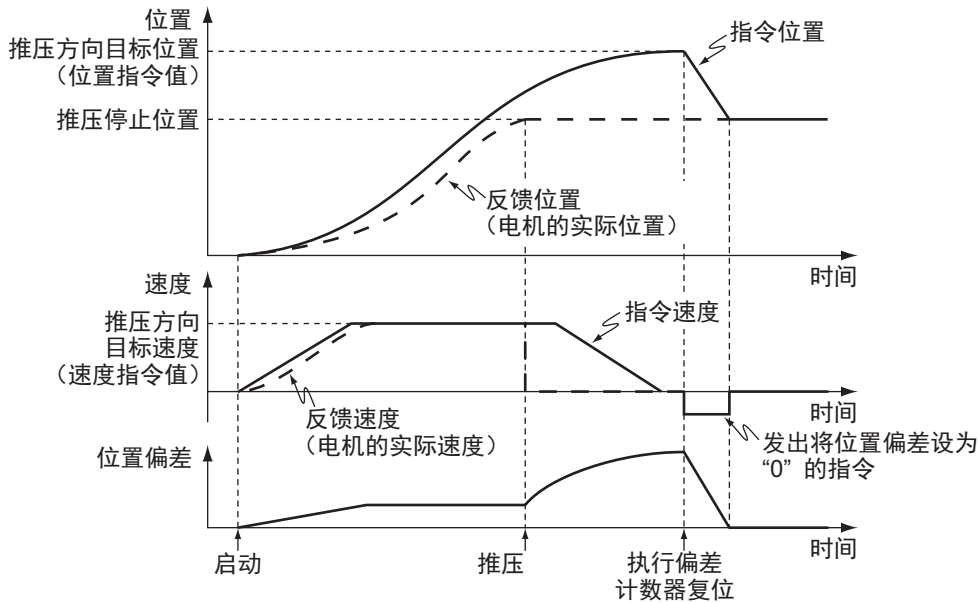
输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。*1

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称 (默认 "MC_Axis****") 或系统定义变量的轴变量名称 (_MC_AX[*], _MC1_AX[*], _MC2_AX[*])。

功能说明

- 本指令的功能是在周期性同步位置模式下，将MC功能模块的指令当前位置和反馈当前位置的偏差设为“0”。
- 检测到 Execute（启动）的上升沿后，将此时的反馈当前位置作为新的目标位置，赋予指令位置。例如，如下图所示，在发生位置偏差的推压动作时，若启动本指令，将赋予反方向的位置指令，位置偏差设为“0”。
发生位置偏差的指令中，CommandAborted（指令中断）变为 TRUE，中止指令执行。



- 将位置偏差设为“0”时，使用轴参数中设定的最高速度赋予位置指令。不会应用最大加速度、最大减速度。
- 完成向新目标位置的指令时，输出变量“Done（完成）”变为 TRUE。
- 本指令会向发生位置偏差的动作的反方向发出指令，但不会应用轴参数“反转时动作”。



使用注意事项

- 请在轴速度较小的状态下启动本指令。
本指令将赋予与前指令（推压方向）相反方向的指令值。因此，若在轴速度较大的状态下启动本指令，可能发生机械冲撞。
- 在垂直轴等长期保持扭矩的轴中使用本指令时，请在确认不会因本指令的启动而发生扭矩不足后，再使用。
- 使用 NX 系列脉冲输出单元时，脉冲输出单元上连接的伺服驱动器中的偏差不会重置。详情请参考 □□《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。
- 对同步控制的主轴使用本指令时
在同步控制的主轴上，且将指令位置作为同步数据使用时，若对主轴启动本指令，从轴将根据齿轮比或凸轮数据变量反转，敬请注意。
关于主轴的注意事项，请参考 □□「同步控制的主轴及辅助轴的注意事项 (P.1-6)」。

指令的详情

下面介绍指令的详情。

● 可使用的轴和启动条件

- 在伺服轴或虚拟伺服轴中，可在以下情况下使用。
 - 单轴的位置控制中
MC_MoveVelocity（速度控制）指令
 - 同步控制中
- 编码器轴、虚拟编码器轴中，启动时会发生异常。

● 轴变量的状态

轴变量的状态 Status.Stopping（减速停止中）变为 TRUE。

● 控制模式切换中本指令的启动

正在执行 MC_TorqueControl（扭矩控制）指令或 MC_SyncMoveVelocity（周期性同步速度控制）指令时，若在执行控制模式变为位置控制模式的指令，且实际切换为位置控制模式之前，启动本指令，因 CPU 单元版本的不同，偏差变为“0”的时间如下所示。

单元版本 1.10 以上的 CPU 单元时

切换为位置控制模式后，立即将指令当前位置和反馈当前位置的偏差设为“0”。

单元版本 1.09 以下的 CPU 单元时

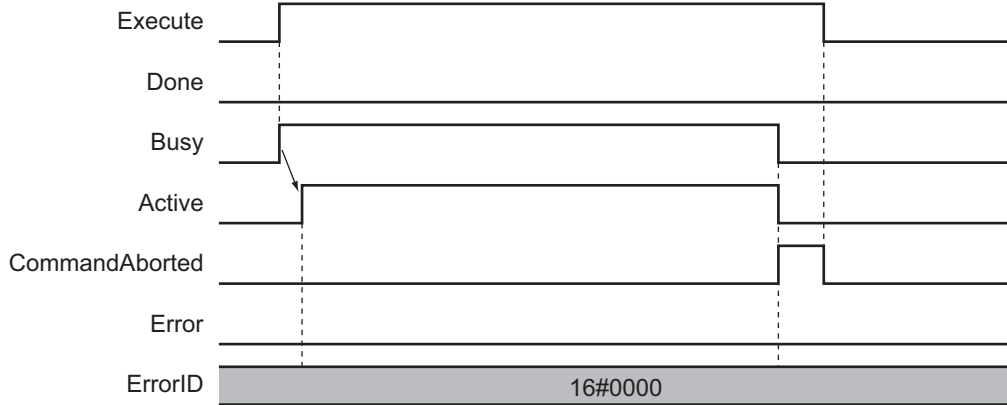
启动本指令时，将指令当前位置和反馈当前位置的偏差设为“0”。

控制模式切换的详情请参考 □□「MC_TorqueControl(P.3-305)」及
□□「MC_SyncMoveVelocity(P.3-360)」。

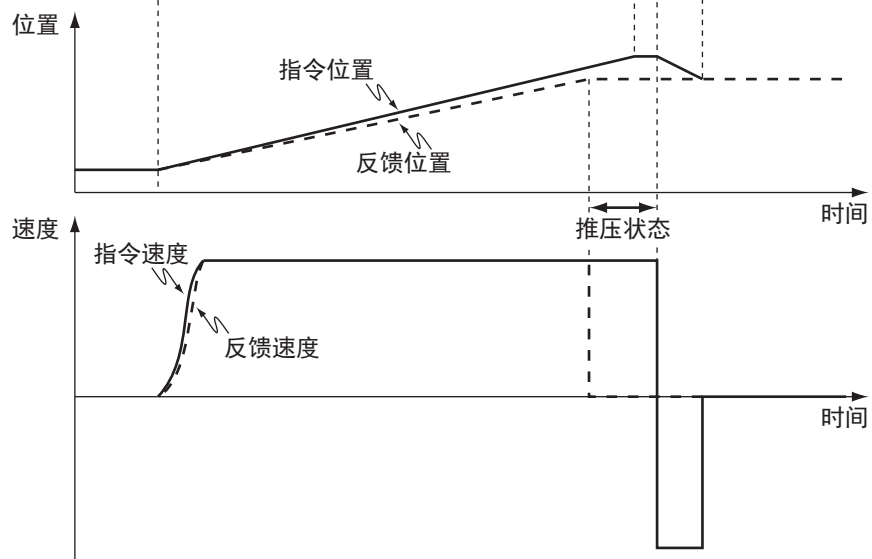
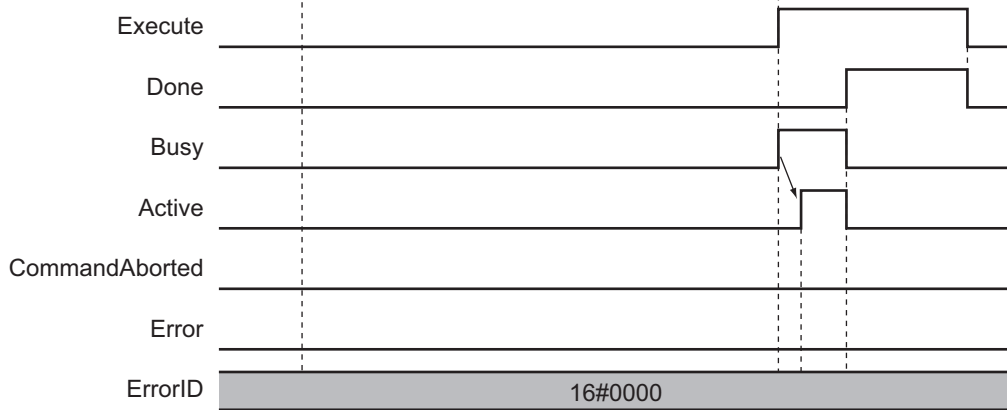
时序图

启动 MC_MoveAbsolute（绝对值定位）指令后，在推压状态下启动了本指令时，时序图如下所示。

MC_MoveAbsolute 指令



MC_ResetFollowingError 指令



运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 其他指令执行过程中本指令的启动

对当前正在执行的指令启动本指令后，可切换为本指令或缓冲本指令。

每个轴最多可缓冲 1 个。

在 BufferMode（缓冲模式选择）中指定多重启动时本指令的动作。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。

BufferMode（缓冲模式选择）请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

与 MC_Stop（强制停止）指令的关系

如果在执行 MC_Stop（强制停止）指令的过程中启动了 MC_ResetFollowingError（偏差计数器复位）指令，MC_Stop（强制停止）指令的 Done 将变为 TRUE，执行 MC_ResetFollowingError（偏差计数器复位）指令。

● 本指令执行过程中其他指令的启动

正在执行本指令时，变为减速停止中。

因此，请在减速停止过程中启动可多重启动的指令。

如果是无法启动的指令，本指令的 CommandAborted（执行中断）将变为 TRUE。

与 MC_Stop（强制停止）指令的关系

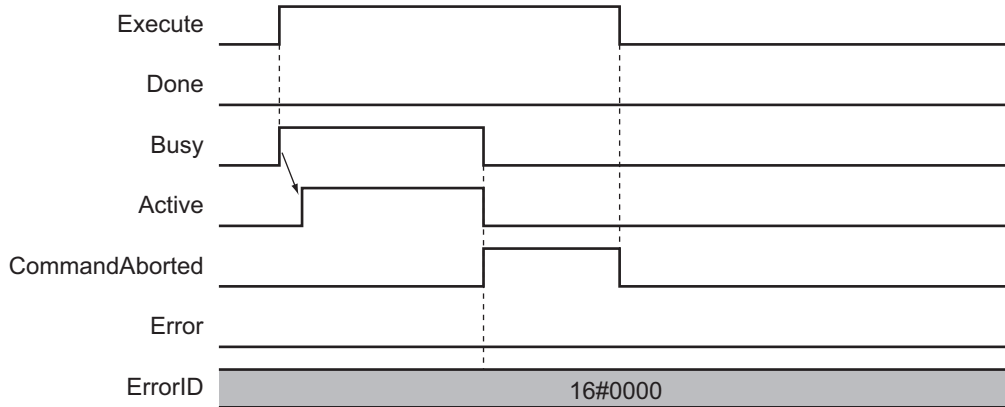
如果在执行 MC_ResetFollowingError（偏差计数器复位）指令时启动了 MC_Stop（强制停止）指令，MC_ResetFollowingError（偏差计数器复位）指令的 CommandAborted（执行中断）将变为 TRUE，执行 MC_Stop（强制停止）指令。但是，不会适用 MC_Stop（强制停止）指令的减速度，立即停止。

异常

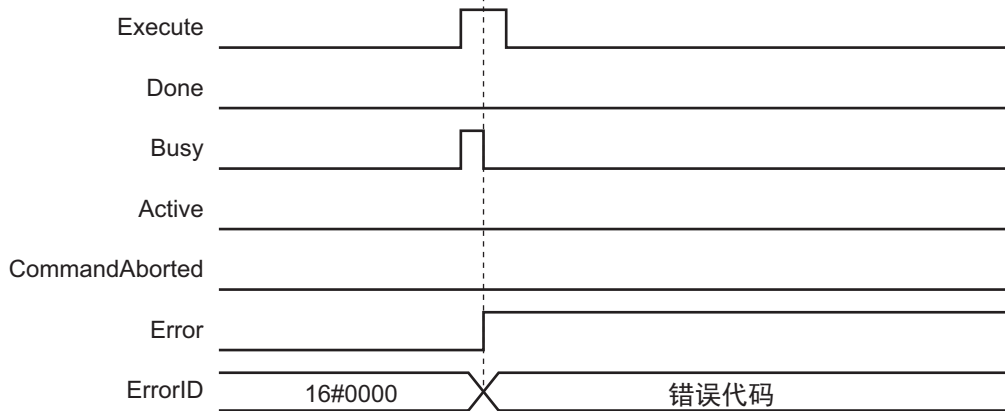
执行本指令的过程中发生异常时，将立即停止，CommandAborted（执行中断）变为 TRUE。此外，如果在启动本指令时检测到异常，Error（错误）将变为 TRUE。参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

启动 MC_MoveAbsolute（绝对值定位）指令后，在推压状态下启动本指令，但发生异常时，时序图如下所示。

MC_MoveAbsolute 指令



MC_ResetFollowingError 指令



● 错误代码

指令中发生的异常请参考 □□「A-1 错误代码一览 (P.A-2)」。

MC_CamIn

使用指定的凸轮表开始凸轮动作。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_CamIn	凸轮动作开始	FB		<pre> MC_CamIn_instance (Master := 《参数》, Slave := 《参数》, CamTable := 《参数》, Execute := 《参数》, Periodic := 《参数》, StartMode := 《参数》, StartPosition := 《参数》, MasterStartDistance := 《参数》, MasterScaling := 《参数》, SlaveScaling := 《参数》, MasterOffset := 《参数》, SlaveOffset := 《参数》, ReferenceType := 《参数》, Direction := 《参数》, CamTransition := 《参数》, BufferMode := 《参数》, InCam => 《参数》, InSync => 《参数》, EndOfProfile => 《参数》, Index => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》); </pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Periodic	重复模式	BOOL	TRUE、FALSE	FALSE	指定是否重复执行指定的凸轮表，或只执行 1 次。 TRUE：重复 FALSE：不重复
StartMode	开始位置方式指定	_eMC_START_MODE	0: _mcAbsolutePosition 1: _mcRelativePosition	0 *1	指定使用 MasterStartDistance(主轴追踪距离) 的坐标系。 0: 绝对位置 1: 相对位置
StartPosition	凸轮表开始位置	LREAL	负数、正数、“0”	0	以主轴的绝对位置指定凸轮表的起点（相位=0）。 单位为 [指令单位]。*2

输入变量	名称	数据类型	有效范围	初始值	内容
MasterStart Distance	主轴追踪距离	LREAL	负数、正数、“0”	0	指定从轴开始凸轮动作时主轴的位置。 StartMode（开始位置方式指定）指定为[绝对定位]时，指定主轴的绝对位置，指定为[相对定位]时，指定从StartPosition（凸轮表开始位置）起的相对量。 单位为[指令单位]。 ^{*2}
Master Scaling	主轴系数	LREAL	正值 (>0.0)	1.0	按指定的比例，放大/缩小主轴的相位。
SlaveScaling	从轴系数	LREAL	正值 (>0.0)	1.0	按指定的比例，放大/缩小从轴的相位。
MasterOffset	主轴偏置	LREAL	负数、正数、“0”	0	按指定的偏置值，错开主轴的相位。
SlaveOffset	从轴偏置	LREAL	负数、正数、“0”	0	按指定的偏置值，错开从轴的相位。
Reference Type	位置种类选择	_eMC_REFERENCE_TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 ^{*1}	指定主轴的位置种类。 0: 指令位置（前一任务周期 ^{*3} 中的计算值） 1: 反馈位置（同一任务周期 ^{*3} 中的获取值） 2: 指令位置（同一任务周期 ^{*3} 中的计算值）
Direction	方向选择	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection 4: _mcNoDirection	4 ^{*1}	仅在主轴按指定方向动作时，从轴执行凸轮动作。如果主轴按指定方向的反方向动作，从轴将停止凸轮动作。 0: 正方向指定 2: 负方向指定 4: 无方向指定
Cam Transition (Reserved)	凸轮转换选择	_eMC_CAM_TRANSITION	0: _mcCTNone	0 ^{*1}	(Reserved)
BufferMode	缓冲模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow ^{*4} 3: _mcBlendingPrevious ^{*4} 4: _mcBlendingNext ^{*4} 5: _mcBlendingHigh ^{*4}	0 ^{*1}	指定运动指令多重启动时的动作。 0: 中断 1: 缓冲 2: 低共混 3: 前共混 4: 后共混 5: 高共混

*1. 如果是有效范围为列举型的变量初始值，实际不是数值，而是列举值。

*2. 指令单位请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》的“单位转换设定”。

*3. 任务为主固定周期任务时，任务周期为主周期，为固定周期任务（执行优先级 5）时，任务周期为固定周期任务（执行优先级 5）的任务周期。

*4. 可在 CPU 单元版本 1.06 以上和 Sysmac Studio Ver.1.07 以上的组合中使用。无论指定为 4 中共混中的哪种，动作均相同。

输出变量

输出变量	名称	数据类型	有效范围	单位	内容
InCam	凸轮动作中	BOOL	TRUE、FALSE	—	执行了凸轮表的起点时变为 TRUE。
InSync	同步中	BOOL	TRUE、FALSE	—	与主轴同步后变为 TRUE。
EndOfProfile	凸轮周期完成	BOOL	TRUE、FALSE	—	执行了凸轮表的终点时变为 TRUE。
Index	索引	UINT	0 或正值	—	输出凸轮数据的索引号。*1
Busy	执行中	BOOL	TRUE、FALSE	—	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	—	控制过程中变为 TRUE。
Command Aborted	执行中断	BOOL	TRUE、FALSE	—	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	—	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*2	—	发生异常时，输出错误代码。16#0000 表示正常。

*1. InCam（凸轮动作中）为 FALSE 期间，输出为“0”。

*2. □「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
InSync	从轴开始了凸轮动作时	<ul style="list-style-type: none"> • Periodic 为 FALSE, EndOfProfile 变为 TRUE 时 • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时
EndOfProfile	将凸轮表终点的相位和位移作为指令位置输出的周期	EndOfProfile 变为 TRUE 的下 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> • Periodic 为 FALSE, EndOfProfile 变为 TRUE 时 • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时
InCam	主轴通过 StartPosition（凸轮表开始位置）时	<ul style="list-style-type: none"> • Periodic 为 FALSE, EndOfProfile 变为 TRUE 时 • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时
Active	开始了指令时	<ul style="list-style-type: none"> • Periodic 为 FALSE, EndOfProfile 变为 TRUE 时 • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> • 由其他指令进行运动指令多重启动（中断），本指令中止时 • 因发生异常，本指令中止时 • 正在发生异常时，启动了本指令 • 正在执行 MC_Stop 指令时，启动了本指令 • 启动了 MC_CamOut 指令时 	<ul style="list-style-type: none"> • Execute 为 TRUE 时，与 Execute 的 FALSE 同时 • Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Master	主轴	_sAXIS_REF	—	指定主轴。 ^{*1}
Slave	从轴	_sAXIS_REF	—	指定从轴。 ^{*1}
CamTable	凸轮表	ARRAY[0..N] OF _sMC_CAM_REF	—	将凸轮数据结构体 _sMC_CAM_REF 型的排列变量指定为凸轮表。 ^{*2}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

*2. 排列要素 [N] 通过 Sysmac Studio 自动设定。指定在 Sysmac Studio 的凸轮编辑器中创建的凸轮数据变量。



使用注意事项

若主轴和从轴指定为相同的轴，将发生轻度故障等级的“主轴从轴相同（错误代码：5436 Hex）”异常。

功能说明

- 本指令可执行相位（主轴）和位移（从轴）按凸轮表同步的凸轮动作。
- 本指令中指定的凸轮表需要事先用凸轮编辑器创建，然后下载到控制器本体中。
- 在 Execute（启动）的上升沿启动本指令。



使用注意事项

凸轮表需要使用 Sysmac Studio 的凸轮编辑器中创建的凸轮数据变量。



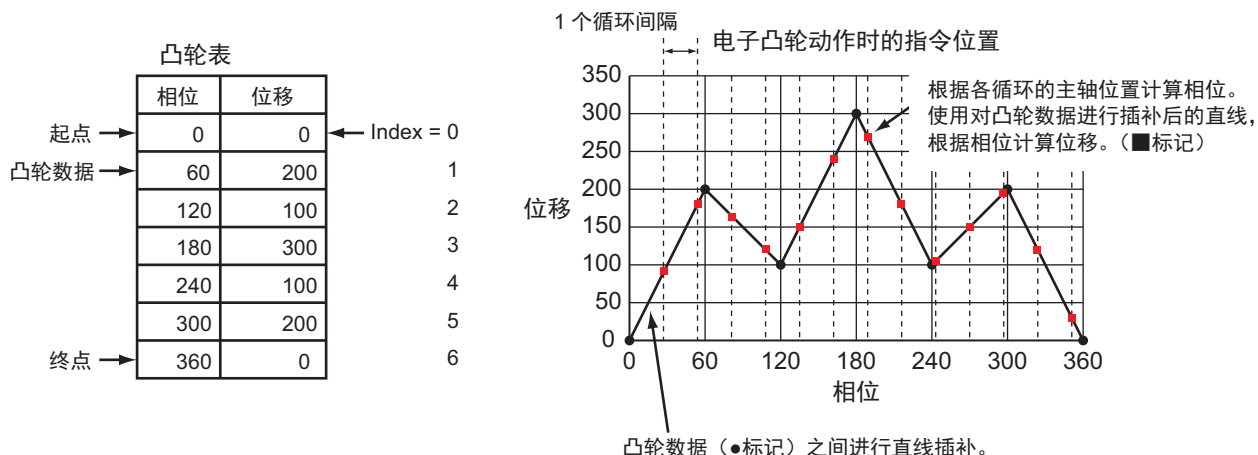
参考

“下载”时使用 Sysmac Studio 的“同步”功能。

凸轮表请参考 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

凸轮表对相位和位移均赋予从起点“0.0”开始的相对量。此外，对主轴、从轴的指令位置，在 2 个凸轮数据间进行直线插补，计算与相位（主轴）对应的位移（从轴）。

如果凸轮数据数较少，相位间隔将变大，凸轮动作的精度会下降。反之，如果凸轮数据数较多，相位间隔将变小，凸轮动作的精度会提高。



使用注意事项

- 在单元版本 1.09 以下的 CPU 单元中使用本指令时，请勿对 Master（主轴）执行 MC_SetPosition（当前位置变更）。若对 Master（主轴）执行 MC_SetPosition（当前位置变更）指令，Slave（从轴）可能发生剧烈追踪的危险。
对 Master（主轴）使用 MC_SetPosition（当前位置变更）指令时，请先解除 Master（主轴）和 Slave（从轴）的关系后再执行。
关于主轴的注意事项，请参考 □「同步控制的主轴及辅助轴的注意事项 (P.1-6)」。
- 本指令会对启动时相位是否按升序排列进行检查。相位不是按升序排列时，将检测为异常。
启动本指令后，若凸轮数据发生变更，确认相位是否按升序排列时，请执行 MC_SetCamTableProperty（凸轮表属性更新）指令。
要在凸轮动作中变更相位时，请在变更前确认按升序排列后，再变更。若不是按升序排列，凸轮动作可能会在中途停止。
- 凸轮数据变量为全局变量，可从多个任务参照或变更凸轮数据变量的值。从多个任务变更凸轮数据变量的值时，编写的程序应确保多个任务的重写处理不会发生冲突。
- 使用“全局变量的任务间排他控制”功能对凸轮数据变量进行排他控制时，请勿用排他对象的凸轮数据变量启动运动指令。将发生轻度故障等级的异常“凸轮表指定不正确（事件代码：5439 Hex）”。

指令的详情

下面介绍指令的详情。

● 指令启动条件

本指令在主轴停止中、位置控制中、速度控制中、同步控制中的任一状态下均可启动。
从轴请参考 □□「运动指令重新执行 (P.3-184)」、□□「运动指令多重启动 (P.3-185)」。

● 软件限位

凸轮动作中从轴超出软件限位时，将发生异常。

● 凸轮数据变量

凸轮数据变量需要对凸轮数据结构体进行排列声明后使用。凸轮数据结构体已事先进行如下类型声明。

```

TYPE
  (* 凸轮数据结构体 *)
  _sMC_CAM_REF :
  STRUCT
    Phase      : REAL;      (* 相位 *)
    Distance   : REAL;      (* 位移 *)
  END_STRUCT;
END_TYPE

```

在 Sysmac Studio 上创建凸轮数据变量。

任意决定凸轮数据变量的名称 —— 凸轮表名称。

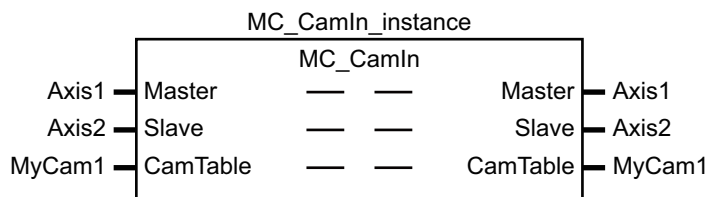
例如，凸轮表名称为“MyCam1”、凸轮数据数为 1000 个时，Sysmac Studio 中将自动进行如下变量声明。
如下自动进行变量声明。

```

VAR
  (* 凸轮表 *)
  MyCam1      : ARRAY [0..999] OF _sMC_CAM_REF;
END_VAR

```

在本指令中指定为“MyCam1”时，记述如下。假设主轴为 Axis1，从轴为 Axis2。



指定的凸轮表在控制器中不存在时，将检测为异常。此外，可在多个轴中指定同一凸轮表。

凸轮数据变量的值可通过用户程序重写。但是，重写值会在控制器电源 OFF、从 Sysmac Studio 下载时消失。

接通 CPU 单元电源后，以及下载完成后，将恢复为从 Sysmac Studio 下载的值。若要保存重写的值，请使用 MC_SaveCamTable（凸轮表保存）指令。

在 CPU 单元的动作模式变更中，将保持重写后的凸轮数据变量值。



参考

- 凸轮数据变量为非网络公开。
例如，“MyCam1[10].Phase”或“MyCam1[10].Distance”可通过 Sysmac Studio 的 Watch View 查看，但不可经由 EIP 从其他控制器访问。
- “下载”时使用 Sysmac Studio 的“同步”功能。

● 凸轮动作的开始

指令启动后，等待主轴达到 StartPosition（凸轮表开始位置）。

主轴通过 StartPosition（凸轮表开始位置）后，执行凸轮表的起点，输出变量 InCam（凸轮动作中）变为 TRUE。

凸轮表的相位和位移是将起点视为 0 的相对量，因此各相位中各轴的绝对位置为将凸轮表起点上各轴绝对位置视为起点的相对值。例如，主轴的计数模式为 0 ~ 360° 的旋转模式，StartPosition（凸轮表开始位置）=60。如下所示，主轴的绝对位置为凸轮表的相位加上 StartPosition 得到的值，从轴的绝对位置为凸轮表的位移加上凸轮表起点上从轴的绝对位置得到的值。

凸轮表			各轴的绝对位置	
	相位	位移	主轴	从轴
起点 →	0	0	60	0 + 凸轮表起点上的从轴绝对位置
凸轮数据 →	60	200	120	200 + 凸轮表起点上的从轴绝对位置
	120	100	180	100 + 凸轮表起点上的从轴绝对位置
	180	300	240	300 + 凸轮表起点上的从轴绝对位置
	240	100	300	100 + 凸轮表起点上的从轴绝对位置
	300	200	360	200 + 凸轮表起点上的从轴绝对位置
终点 →	360	0	60	0 + 凸轮表起点上的从轴绝对位置

StartPosition=60

同时，通过 MasterStartDistance（主轴追踪距离）后，将开始从轴的凸轮动作，输出变量 InSync（同步中）变为 TRUE。

MasterStartDistance（主轴追踪距离）以绝对位置或与 StartPosition（凸轮表开始位置）的相对距离指定。绝对位置或相对位置在 StartMode（开始位置指定方式）中设定。

例 1) 不同 MasterStartDistance 对应的不同从轴动作

本示例中，使用相同的凸轮表和相同的主轴。

凸轮表的设定如下。

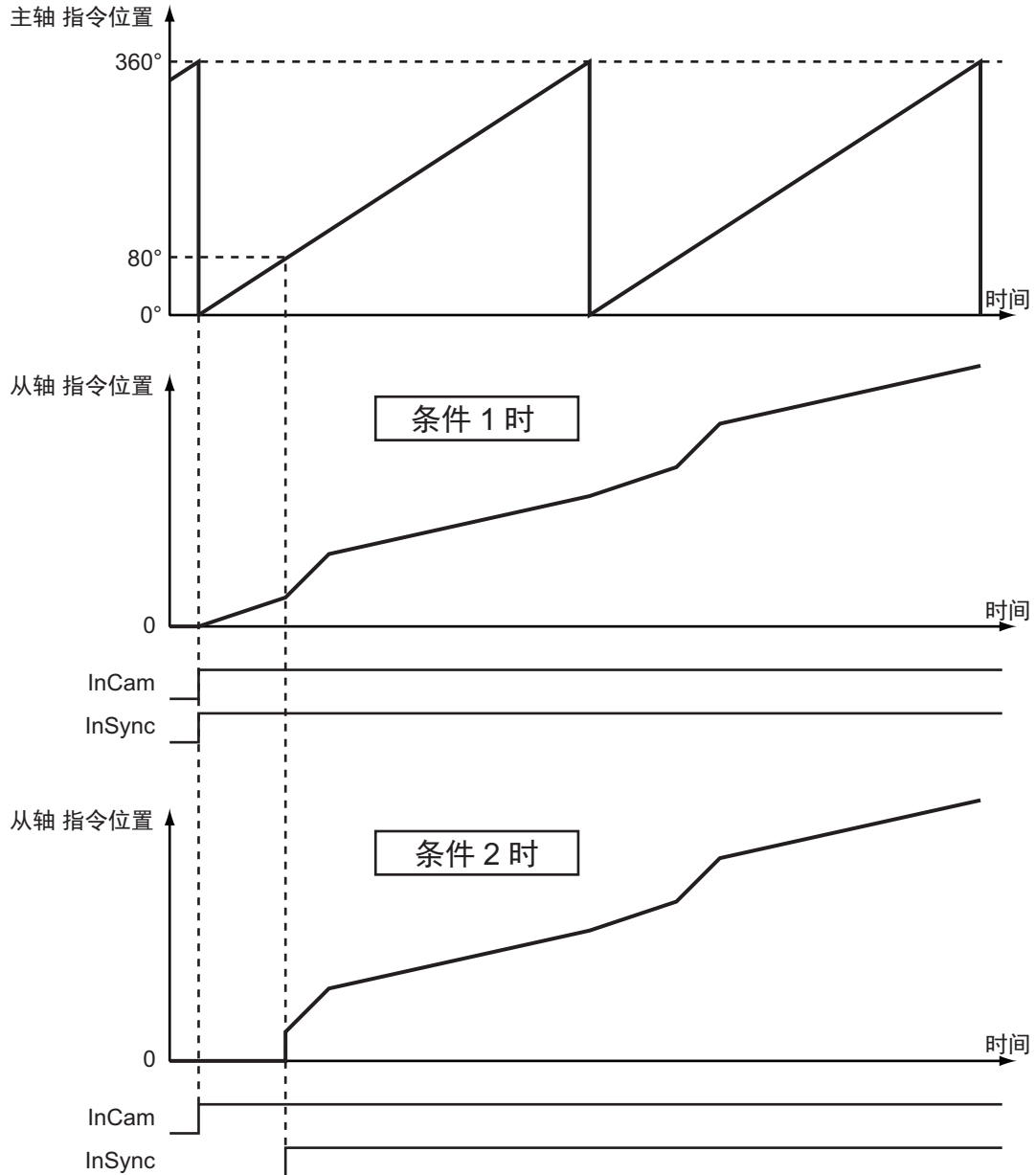
主轴	从轴	凸轮曲线	连接速度	连接加速度	相位间隔宽度
0.000	0.000	—	—	—	—
80.000	80.000	直线	360.000	0.000	0.010
120.000	200.000	直线	1080.000	0.000	0.010
360.000	360.000	直线	240.000	0.000	0.010

凸轮动作的开始条件如下。

输入变量	条件 1	条件 2
Periodic（重复模式）	TRUE（重复）	TRUE（重复）
StartMode（开始位置方式指定）	_mcRelativePosition（相对位置）	_mcRelativePosition（相对位置）
StartPosition（凸轮表开始位置）	0	0
MasterStartDistance（主轴追踪距离）	0	80

条件 1 中，主轴通过 0° 后，输出变量 InCam（凸轮动作中）和输出变量 InSync（同步中）均变为 TRUE，从轴开始凸轮动作。

条件 2 中，主轴通过 0° 后，输出变量 InCam（凸轮动作中），再通过 80°*¹ 后，输出变量 InSync（同步中）均变为 TRUE，从轴开始凸轮动作。条件 2 时，会在凸轮表的中途开始凸轮动作，因此从轴会剧烈加速。

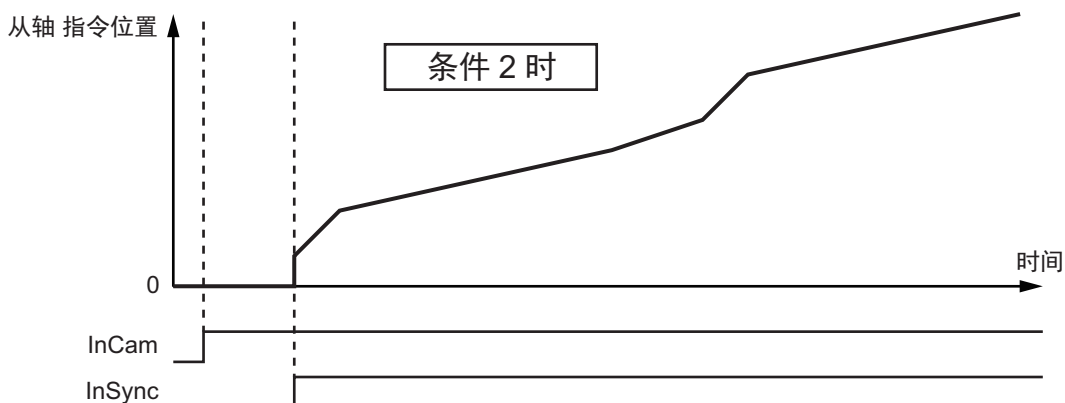


*1. StartMode 为 _mcRelativePosition，因此 StartPosition + MasterStartDistance=80° 为凸轮动作开始位置。

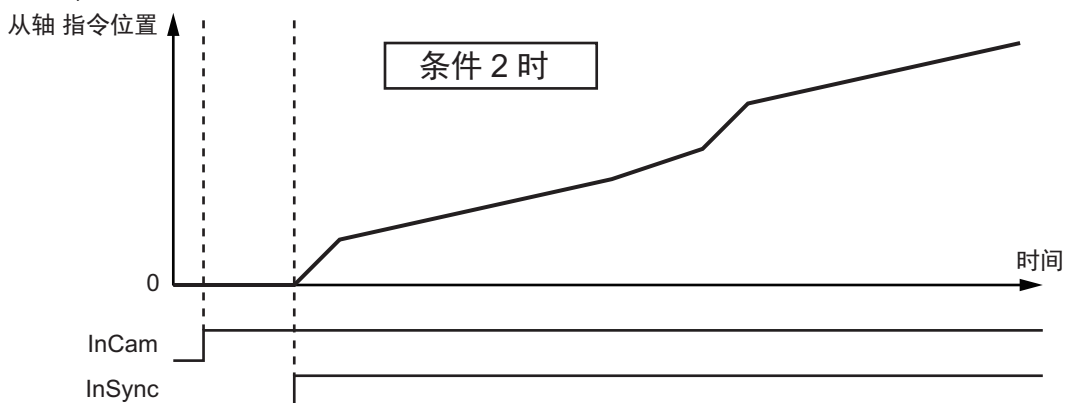


参考

条件 2 时，若设定为 SlaveOffset = 0，从轴将如下图所示剧烈加速。



设定为 SlaveOffset = -80 后，从轴将从 MasterStartDistance（主轴追踪距离）的位移开始同步，不会剧烈加速。



例 2) 不同 StartPosition 和 MasterStartDistance 对应的不同从轴动作

凸轮表的设定与前述示例相同。凸轮动作的开始条件如下。

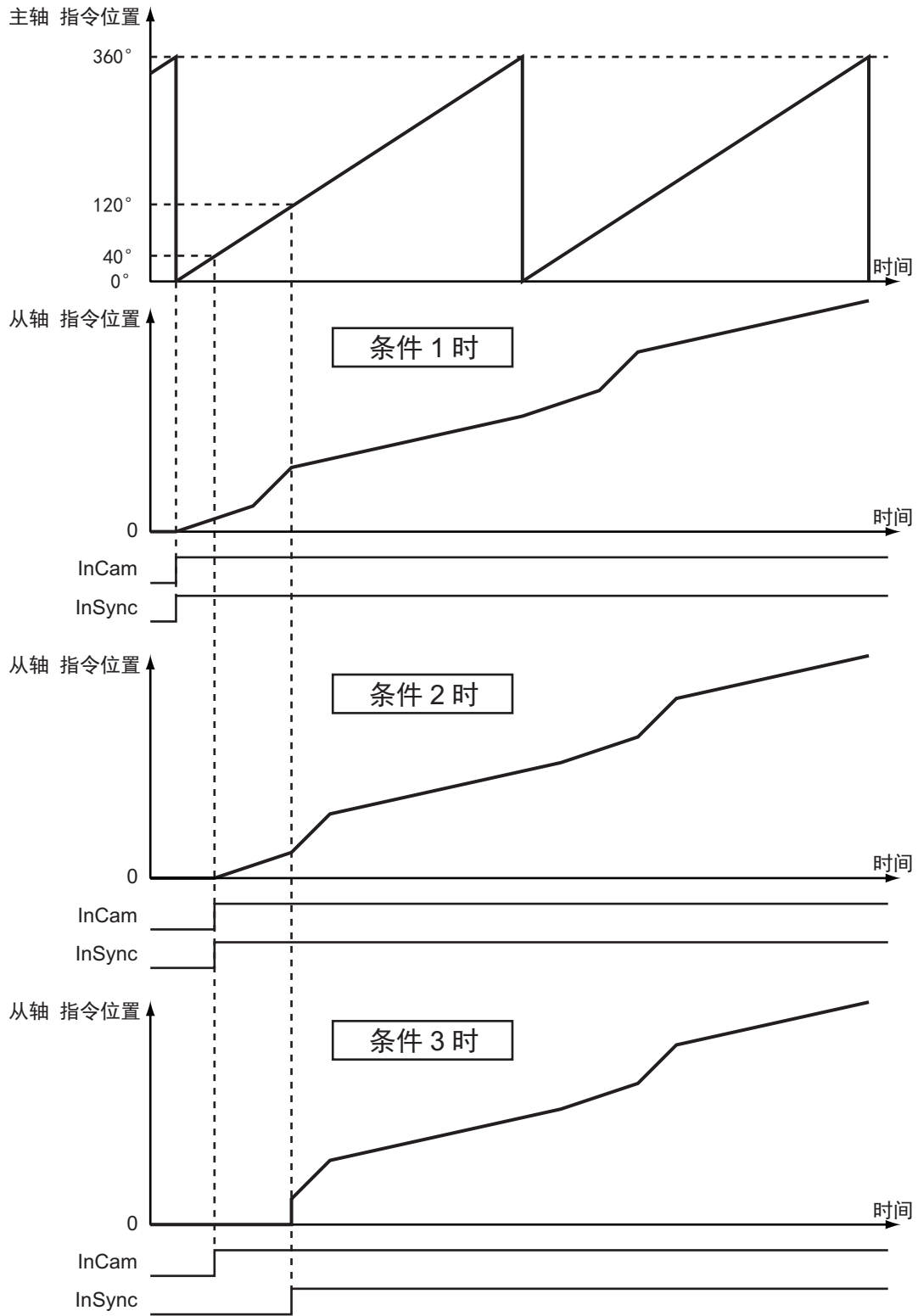
输入变量	条件 1	条件 2	条件 3
Periodic（重复模式）	TRUE（重复）	TRUE（重复）	TRUE（重复）
StartMode（开始位置方式指定）	_mcRelativePosition （相对位置）	_mcRelativePosition （相对位置）	_mcRelativePosition （相对位置）
StartPosition（凸轮表开始位置）	0	40	40
MasterStartDistance（主轴追踪距离）	0	0	80

条件 1 中，主轴通过 0° 后，输出变量 InCam（凸轮动作中）和输出变量 InSync（同步中）均变为 TRUE，从轴开始凸轮动作。

条件 2 中，主轴通过 StartPosition（凸轮表开始位置）中指定的 40° 后，输出变量 InCam（凸轮动作中）和输出变量 InSync（同步中）均变为 TRUE，从轴开始凸轮动作。

条件 3 中，主轴通过 40° 后，输出变量 InCam（凸轮动作中），再通过 $120^{\circ*1}$ 后，输出变量 InSync（同步中）均变为 TRUE，从轴开始凸轮动作。

*1. StartMode 为 _mcRelativePosition，因此 StartPosition + MasterStartDistance = 120° 为凸轮动作开始位置。



例 3) 不同 StartMode (开始位置方式指定) 对应的从轴凸轮动作开始的不同

利用 StartMode (开始位置方式指定), 可以指定将 MasterStartDistance (主轴追踪距离) 中指定的值作为绝对位置或相对位置。

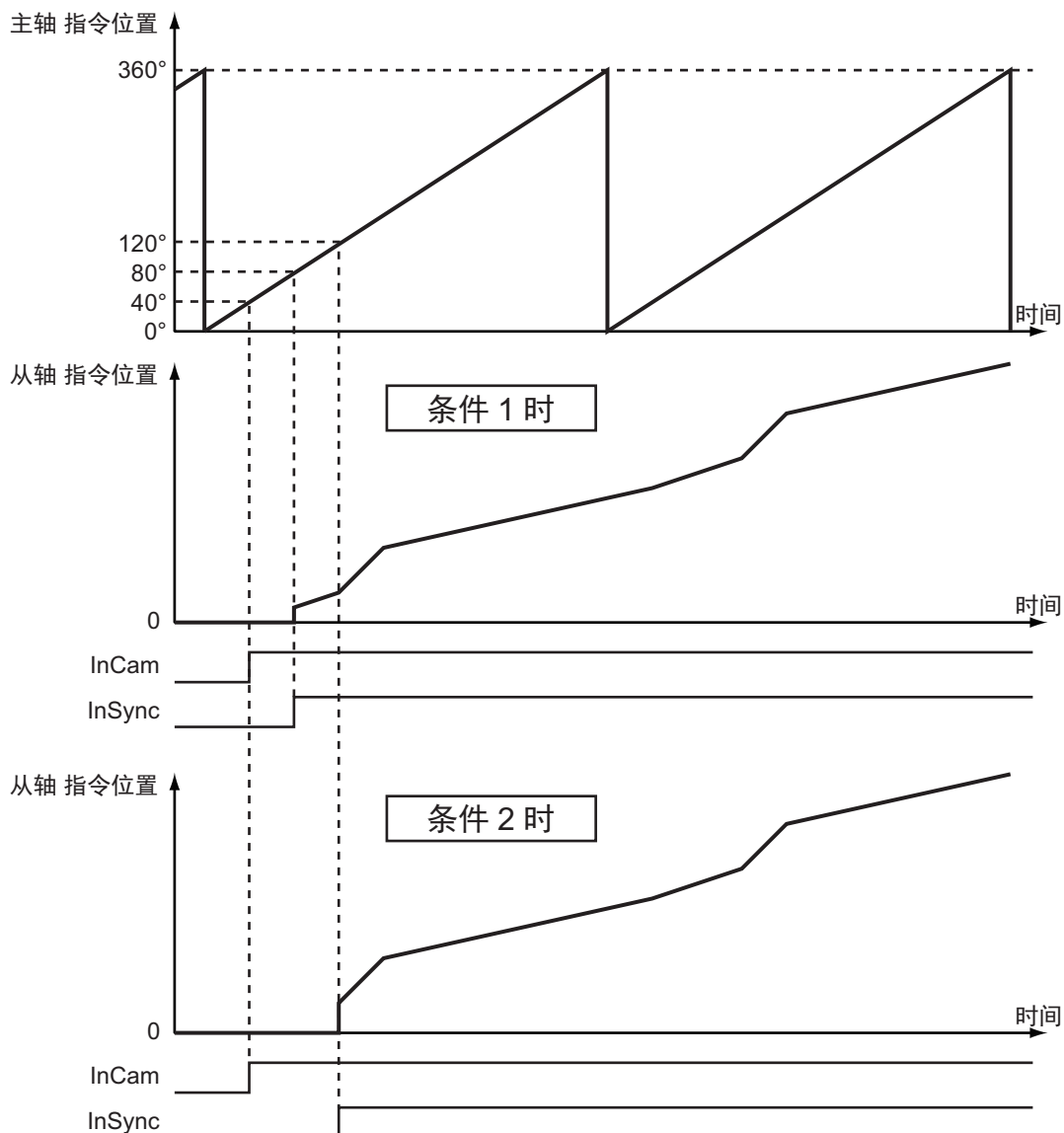
下面介绍不同 StartMode (开始位置方式指定) 对应的从轴凸轮动作开始的不同。凸轮表的设定与前述示例相同。

凸轮动作的开始条件如下。

输入变量	条件 1	条件 2
Periodic (重复模式)	TRUE (重复)	TRUE (重复)
StartMode (开始位置方式指定)	_mcAbsolutePosition (绝对位置)	_mcRelativePosition (相对位置)
StartPosition (凸轮表开始位置)	40	40
MasterStartDistance (主轴追踪距离)	80	80

无论是条件 1 还是条件 2, 当主轴通过 40° 后, 输出变量 InCam (凸轮动作中) 变为 TRUE。条件 1 中, 由于 StartMode (开始位置方式指定) 为 _mcAbsolutePosition (绝对位置), 因此主轴通过 80° 后, 输出变量 InSync (同步中) 变为 TRUE, 从轴开始凸轮动作。

条件 2 中, 由于 StartMode (开始位置方式指定) 为 _mcRelativePosition (相对位置), 因此主轴通过 120° ($=40^{\circ} + 80^{\circ}$) 后, 输出变量 InSync (同步中) 变为 TRUE, 从轴开始凸轮动作。





使用注意事项

要进行凸轮动作，需要使用 Sysmac Studio 的凸轮编辑器，创建凸轮轮廓，再下载到 CPU 单元中。

“下载”时使用 Sysmac Studio 的“同步”功能。

● Periodic（重复模式）

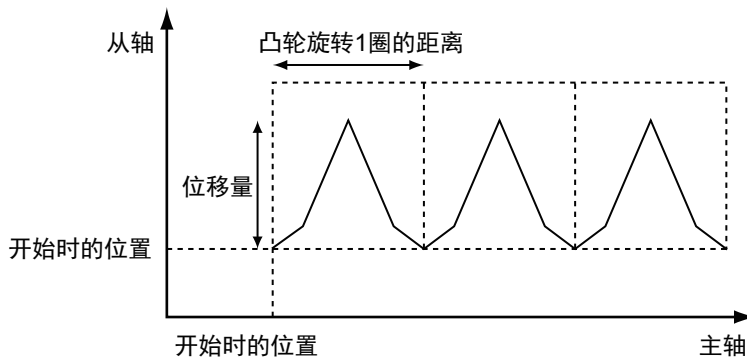
Periodic（重复模式）指定为 TRUE [重复] 时，将从凸轮表的起点到终点，重复执行凸轮动作。

指定为 FALSE [不重复] 时，执行至凸轮表的终点后，结束凸轮动作。

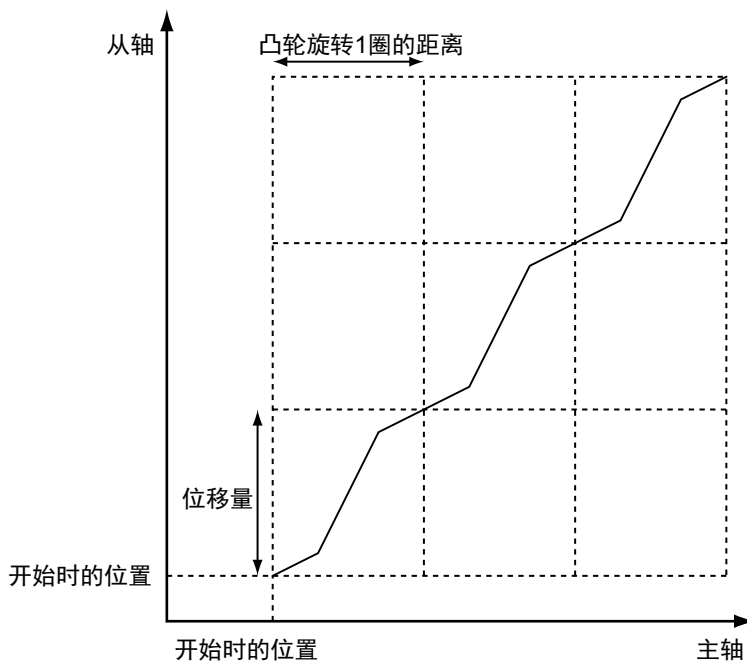
TRUE [重复] 时，在凸轮表的起点和终点上，如果从轴的行程位置相同，将作为往复凸轮动作（参考往复凸轮动作）。在起点和终点上，如果从轴的行程位置不同，将作为传送凸轮动作（参考传送凸轮动作）。

此外，下图中，横轴表示主轴位置，纵轴表示从轴位置。

往复凸轮动作



传送凸轮动作



● EndOfProfile（凸轮周期完成）

在凸轮表的终点上，在输出“使用所定义相位、位移的凸轮动作的指令值”的周期内，EndOfProfile（凸轮周期完成）在 1 个周期内为 TRUE。

StartPosition（凸轮表开始位置）设定为主轴的绝对位置，凸轮表为与该位置相对的表。

EndOfProfile（凸轮周期完成）为表示凸轮表终端的输出。

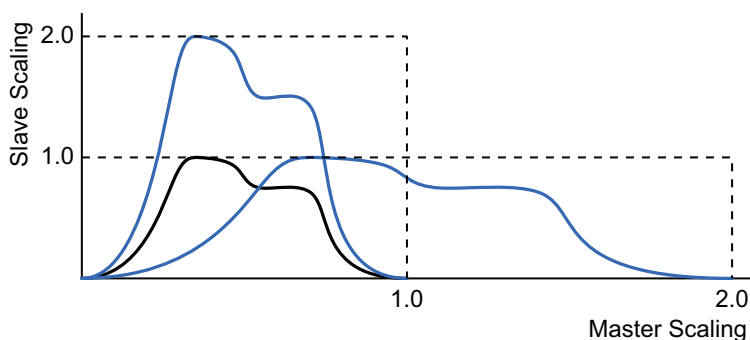
● 凸轮动作的结束

要中途结束凸轮动作时，使用 MC_CamOut（凸轮动作解除）指令或 MC_Stop（强制停止）指令。

● 系数（比例缩放）

可对指定凸轮表的主轴相位、从轴相位分别按指定比例进行放大和缩小。

主轴和从轴可分别设定倍率。

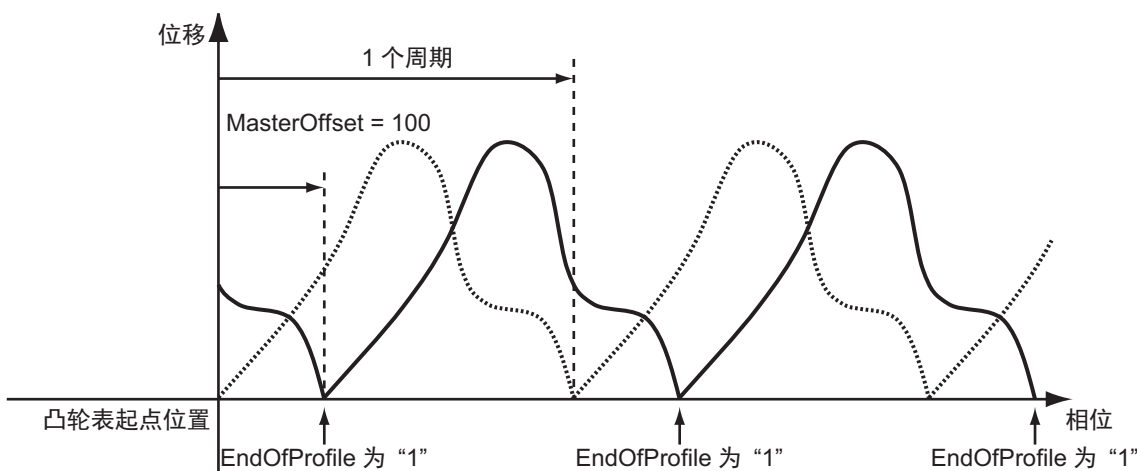


● 偏置

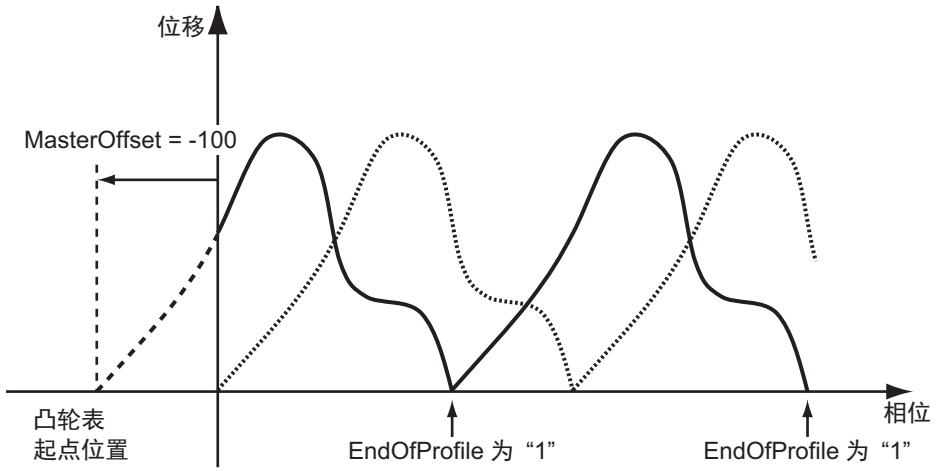
对指定的凸轮表仅按偏置量移动相位和位移的功能。

可对主轴相位、从轴位移分别指定偏置量。

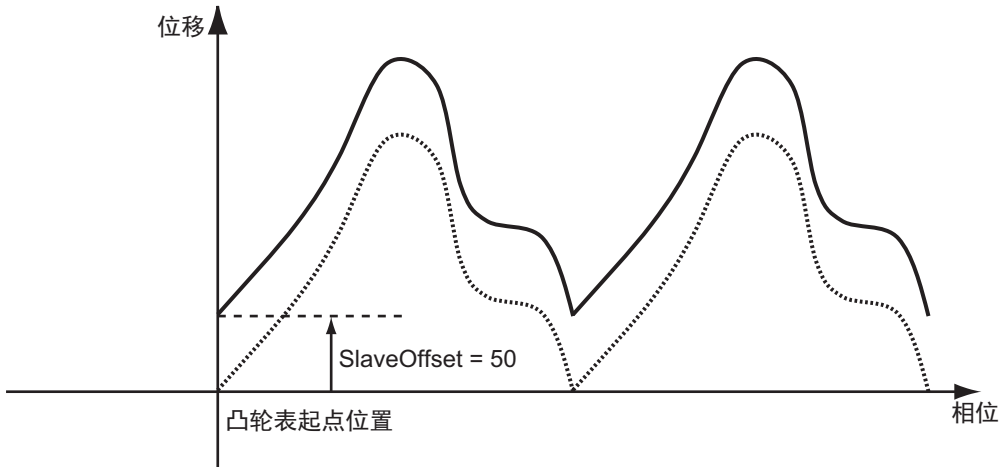
MasterOffset（主轴偏置）> 0 时



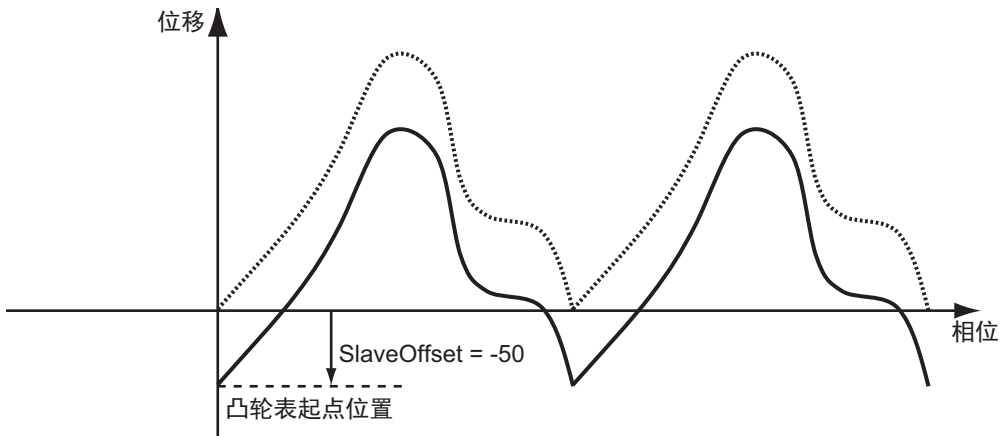
MasterOffset (主轴偏置) < 0 时



SlaveOffset (从轴偏置) > 0 时



SlaveOffset (从轴偏置) < 0 时



● ReferenceType（位置种类选择）

从轴要同步的主轴位置种类可从以下项目中选择。

- `_mcCommand`: 指令位置（前一任务周期中的计算值）
在当前周期，使用前一任务周期中计算的主轴指令位置。
使用计算从轴指令位置之前的固定周期任务中计算得到的主轴指令位置。
- `_mcFeedback`: 同一任务周期的获取值
使用同一任务周期内获取的主轴的反馈位置。
- `_mcLatestCommand`: 指令位置（同一任务周期中的计算值）
使用同一任务周期中计算得到的主轴指令位置。
可使用比 `_mcCommand` 更新的信息。但是，主轴编号必须设定为比从轴编号更新的编号。
如果从轴的轴编号比主轴新，本指令的 Error（错误）将变为 TRUE。ErrorID（错误代码）中将输出“主轴 / 从轴 轴编号非升序（错误代码：5438 Hex）”。



使用注意事项

任务为主固定周期任务时，这里的任务周期为主周期，为固定周期任务（执行优先级 5）时，任务周期为固定周期任务（执行优先级 5）的任务周期。同样，固定周期任务表示主固定周期任务或固定周期任务（执行优先级 5）。



参考

选择同一任务周期内计算得到的指令位置时，与前一任务周期中的计算值相比，同步精度更高。
但是，使用主轴和从轴时，必须保证“Master（主轴）中设定的运动控制系统变量的轴编号 < Slave（从轴）中设定的运动控制系统变量的轴编号”。

● 轴种类和位置种类的关系

可监视的轴种类和要监视的位置种类的关系如下所示。

轴种类	ReferecneType	
	<code>_mcCommand</code> 或 <code>_mcLatestCommand</code>	<code>_mcFeedback</code>
伺服轴	○	○
编码器轴	x ^{*1}	○
虚拟伺服轴	○	○
虚拟编码器轴	x ^{*1}	○

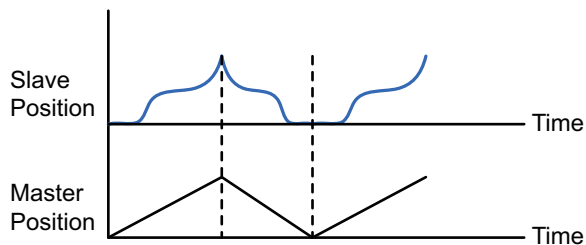
*1. 指令启动时，将发生“超出位置种类选择范围（错误代码：5430 Hex）”。

● Direction（方向选择）

仅在主轴的移动方向与 Direction（方向选择）一致时，可以开始从轴的凸轮动作。
Direction（方向选择）变为有效的时间为 InSync（同步中）中。

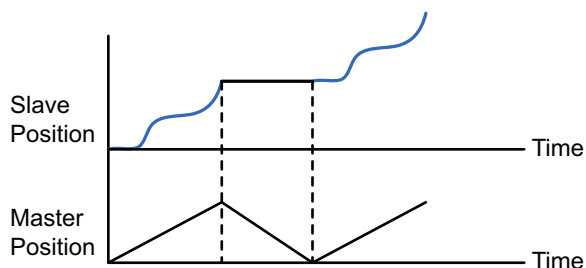
_mcNoDirection（无方向指定）

无论主轴正在向正方向还是负方向移动，均开始凸轮动作。



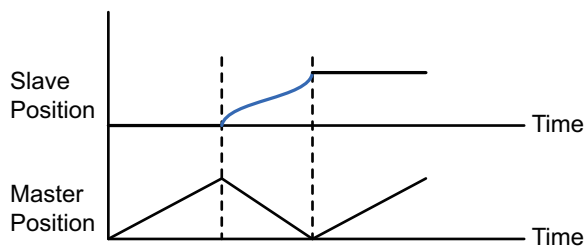
_mcPositiveDirection（正方向指定）

主轴正在向正方向移动时，开始凸轮动作。



_mcNegativeDirection（负方向指定）

主轴正在向负方向移动时，开始凸轮动作。



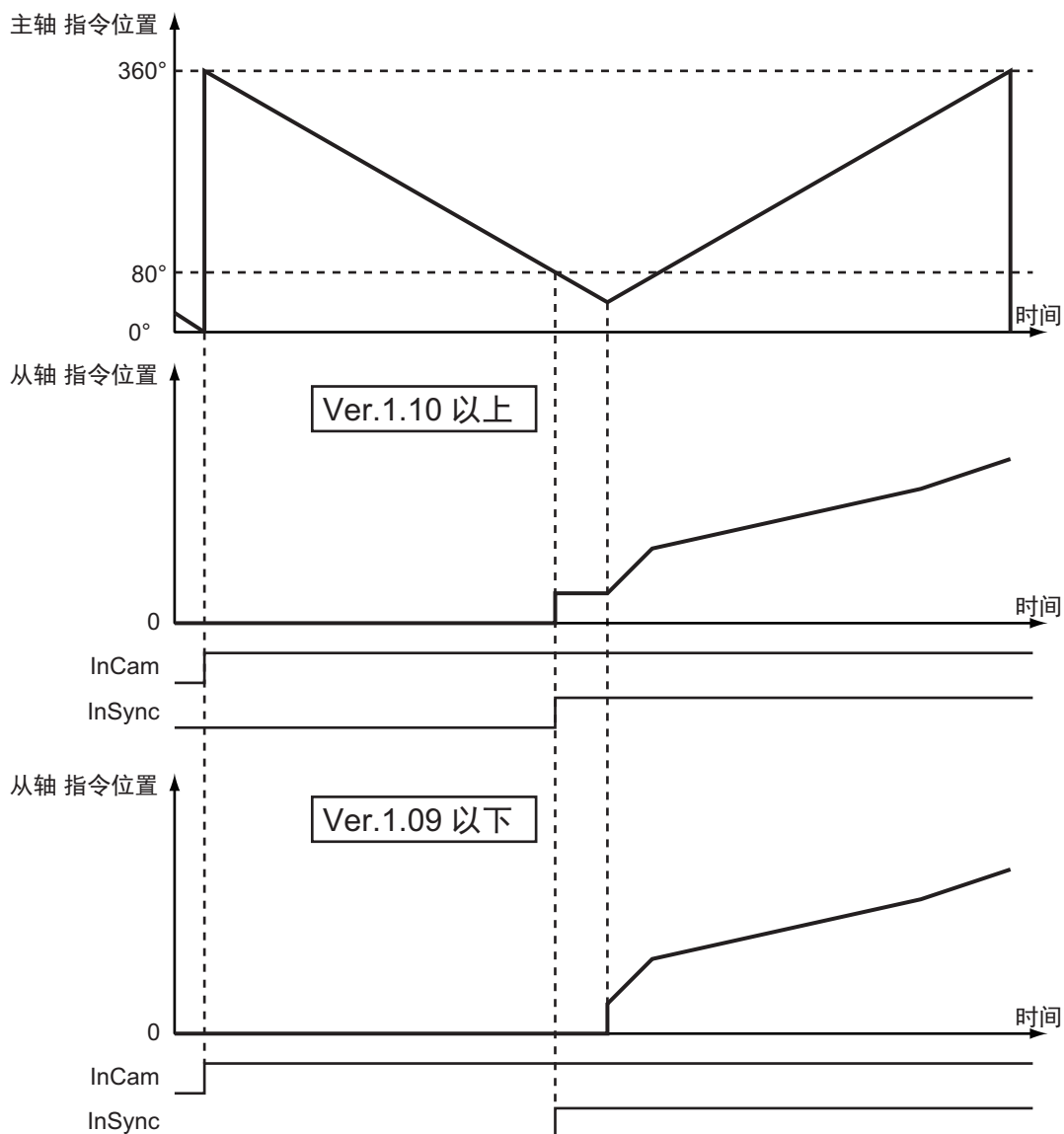
版本相关信息

主轴向 Direction（方向选择）的反方向移动的过程中，若超过 MasterStartDistance（主轴追踪距离），InSync（同步中）变为 TRUE，因 CPU 单元的版本不同，动作如下。

- 单元版本 1.10 以上的 CPU 单元时
InSync（同步中）变为 TRUE 时，从轴向凸轮表的相位对应的位移移动。然后，如果主轴向与 Direction（方向选择）一致的方向移动，则从轴开始凸轮动作。
- 单元版本 1.09 以下的 CPU 单元时
InSync（同步中）变为 TRUE 后，如果主轴向与 Direction（方向选择）一致的方向移动，则从轴开始凸轮动作。

示例如下。凸轮表的设定与前述示例相同。
凸轮动作的开始条件如下。

输入变量	条件
StartMode（开始位置方式指定）	_mcAbsolutePosition（绝对位置）
Direction（方向选择）	_mcPositiveDirection（正方向指定）
StartPosition（凸轮表开始位置）	0
MasterStartDistance（主轴追踪距离）	80



● BufferMode（缓冲模式选择）

指定前一轴动作和本次动作的连接方法。

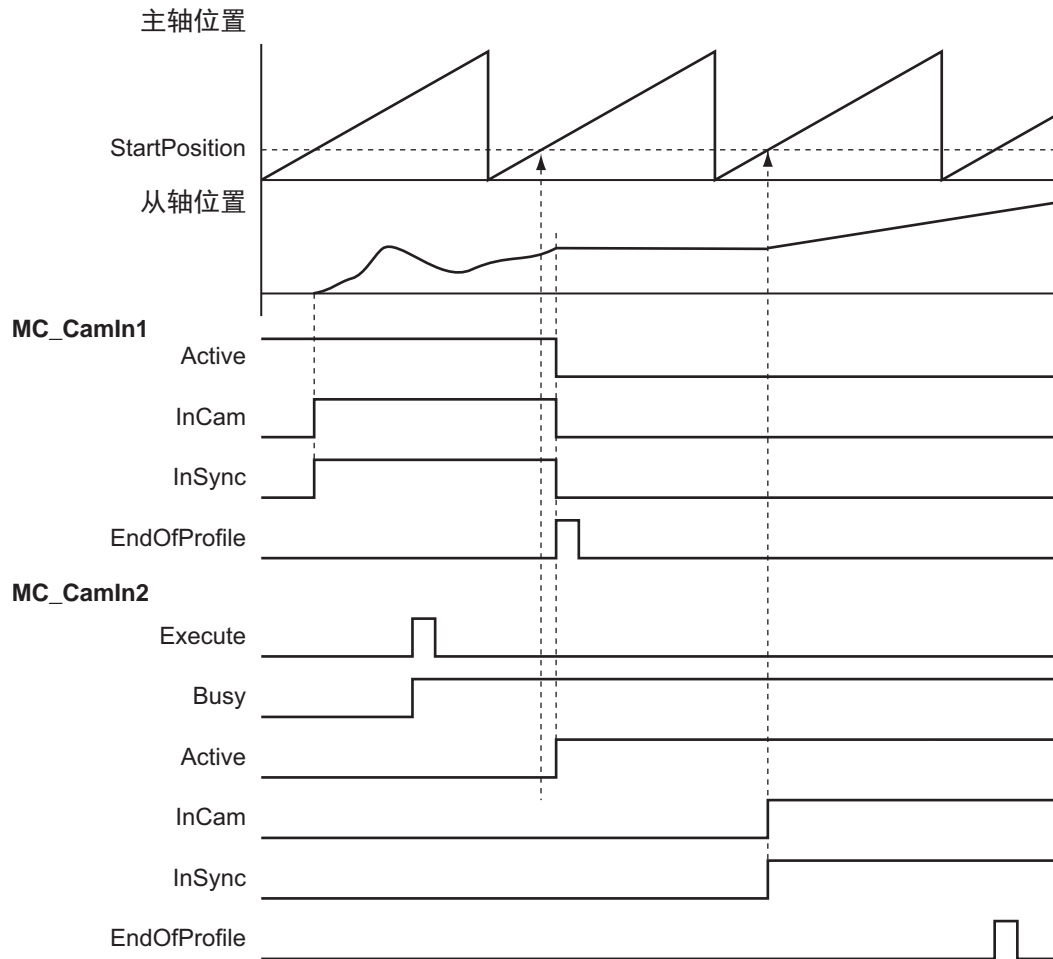
有以下 6 种。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 主轴通过 StartPosition（凸轮表开始位置）后，如果通过 MasterStartDistance（主轴追踪距离），将开始从轴的凸轮动作，输出变量 InSync（同步中）变为 TRUE。 已缓冲的本指令的输出变量 InSync（同步中）变为 TRUE 前，从轴保持停止。 凸轮起点为 StartPosition（凸轮表开始位置）。
缓冲	从当前正在执行的指令正常完成后的周期起，自动启动已缓冲的本指令。主轴通过 StartPosition（凸轮表开始位置）后，如果通过 MasterStartDistance（主轴追踪距离），将开始从轴的凸轮动作，输出变量 InSync（同步中）变为 TRUE。 已缓冲的本指令的输出变量 InSync（同步中）变为 TRUE 前，从轴保持停止。 凸轮起点为 StartPosition（凸轮表开始位置）。
共混*1	从当前正在执行的指令正常完成后的周期起，已缓冲的本指令的输出变量 InSync（同步中）将变为 TRUE，从轴不会停止，直接开始凸轮动作。
低共混	指定了已缓冲的本指令的 StartPosition（凸轮表开始位置）、
前共混	MasterStartDistance（主轴追踪距离）时，无论指定值如何，在指令启动时，
后共混	从轴将同时开始凸轮动作。
高共混	凸轮表起点为当前执行中指令的结束位置。

*1. 可在 CPU 单元版本 1.06 以上和 Sysmac Studio Ver.1.07 以上的组合中使用。无论指定为 4 中共混中的哪种，动作均相同。

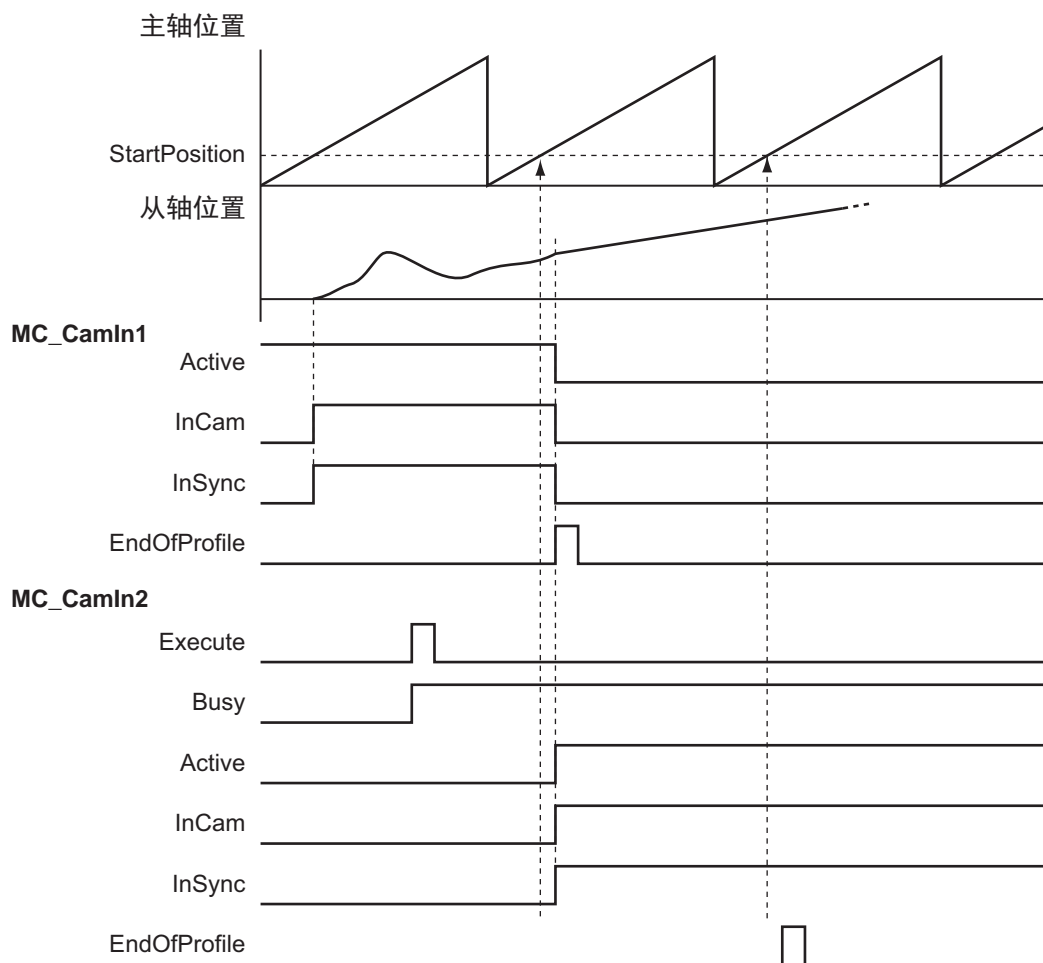
● 指定为缓冲并多重启动时

在执行MC_CamIn1的过程中，指定为缓冲并多重启动MC_CamIn2。MC_CamIn1结束，在MC_CamIn2的Active（控制中）变为TRUE的下一StartPosition（凸轮表开始位置）上，InSync（同步中）变为TRUE，开始凸轮动作。



● 指定为共混并多重启动时

在执行 MC_CamIn1 的过程中，指定为共混并多重启动 MC_CamIn2。在 MC_CamIn1 结束的同一周期内，MC_CamIn2 的 InSync（同步中）变为 TRUE，开始凸轮动作。



BufferMode（缓冲模式选择）请参考 □《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 索引

在计算主轴、从轴的指令位置时使用的 2 个凸轮数据中，将凸轮数据索引编号较小者在输出参数“Index（索引）”中输出。

该值在通过凸轮编辑器或用户程序对凸轮数据进行微调时使用。

● 位置检查

本指令不进行位置检查。

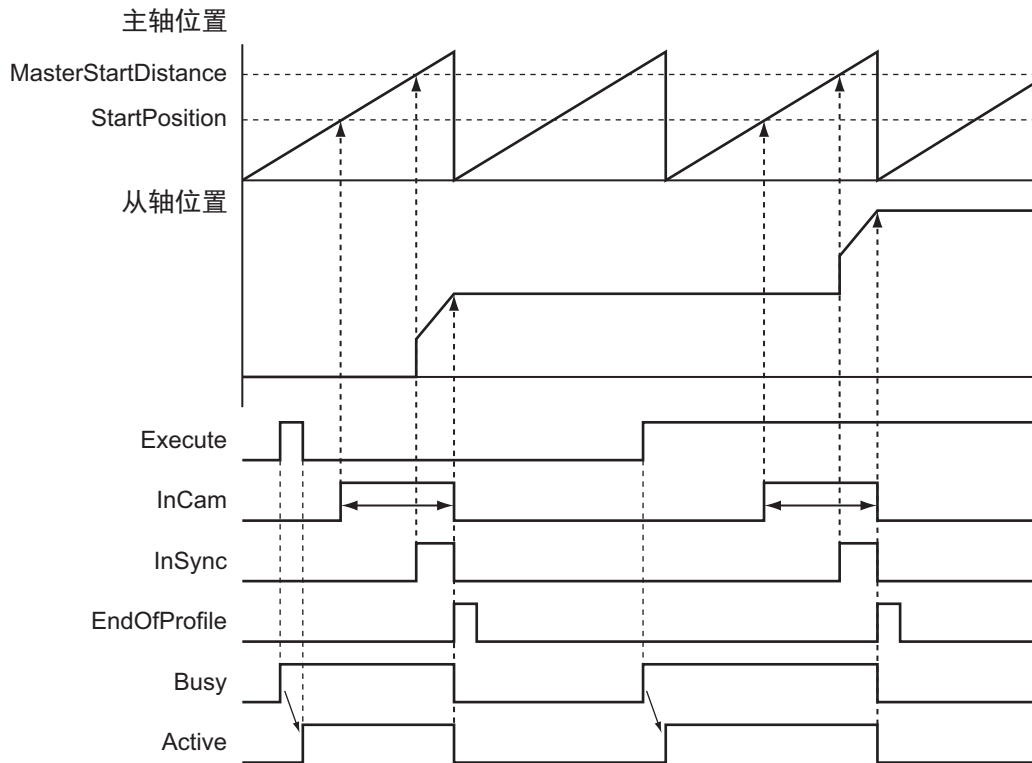
● 超驰

本指令不可通过 MC_SetOverride（超驰值设定）指令进行超驰。

时序图

● 不重复时

MC_CamIn（凸轮动作开始）指令的 Periodic（重复模式）为 FALSE [不重复] 时，如下图所示。



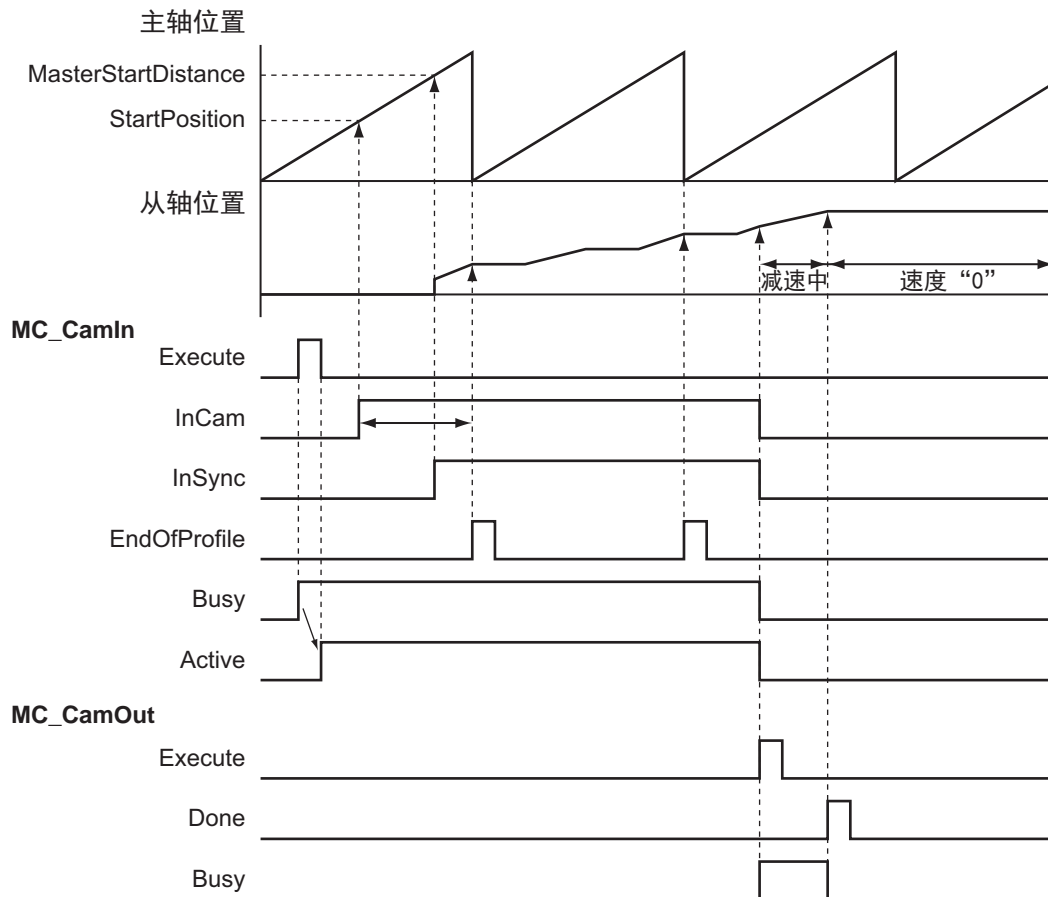
● 有重复时

重复执行凸轮动作。

使用 MC_CamOut（凸轮动作解除）指令解除凸轮时，从轴将减速至“0”。

不会影响主轴的动作。

MC_CamIn（凸轮动作开始）指令的 Periodic（重复模式）为 TRUE [重复] 以及执行了 MC_CamOut（凸轮动作解除）指令时，如下图所示。

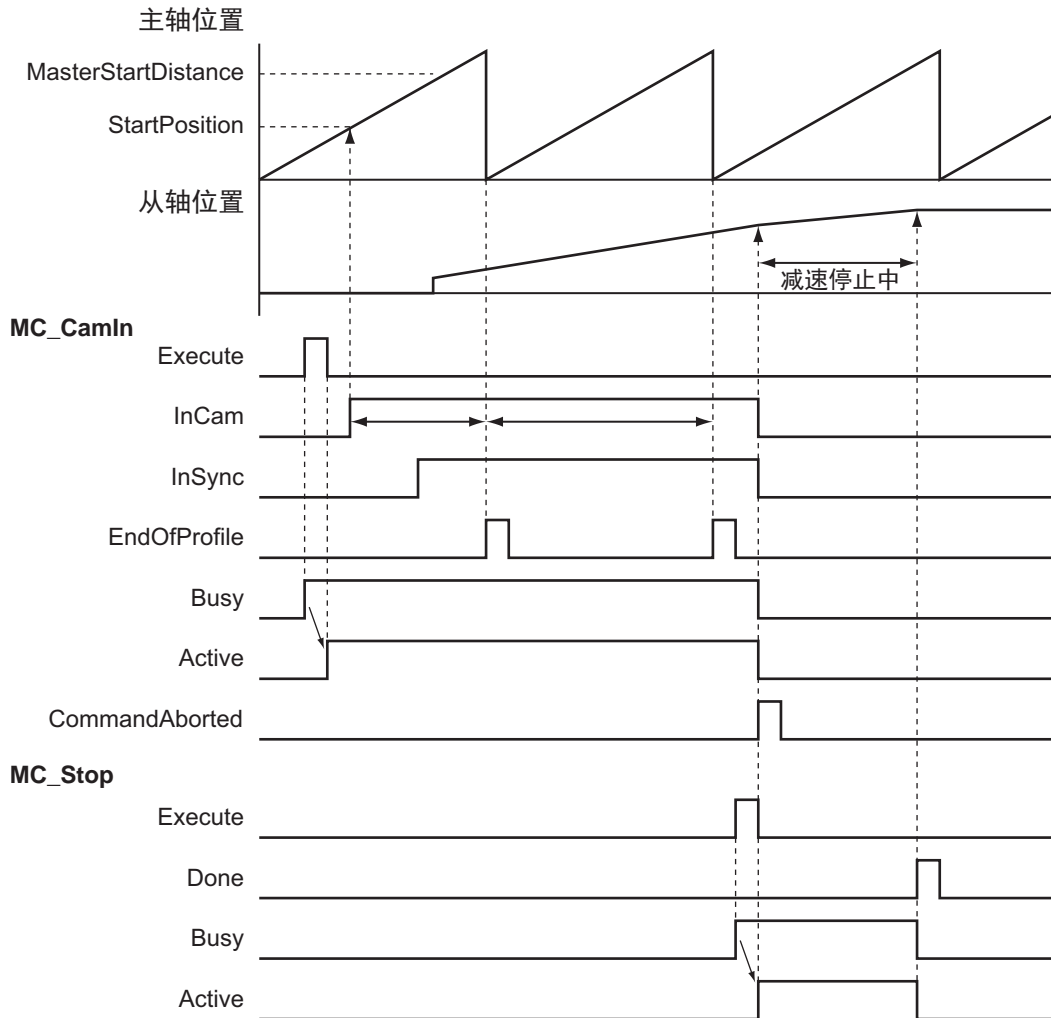


● MC_Stop（强制停止）指令时

在凸轮动作过程中对主轴启动了 MC_Stop（强制停止）指令时，将维持主轴和从轴的同步关系。

在凸轮动作过程中对从轴启动了 MC_Stop（强制停止）指令时，将结束主轴和从轴的同步关系。

MC_CamIn（凸轮动作开始）指令的 Periodic（重复模式）为 TRUE [重复] 以及对从轴执行了 MC_Stop（强制停止）指令时，如下所示。



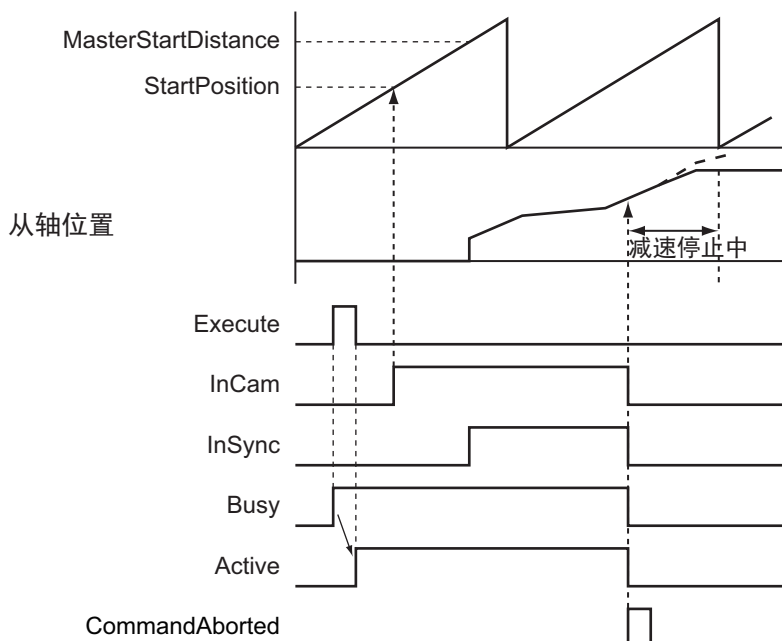
指令的中止

执行本指令的过程中从轴发生轴异常时，从轴将按轴的最大减速度停止。对从轴启动了 MC_Stop（强制停止）指令时，CommandAborted（执行中断）变为 TRUE，从轴按 MC_Stop（强制停止）指令中指定的减速度停止。

轴异常发生原因的查询方法请参考 □「A-1 错误代码一览 (P.A-2)」。

此外，主轴发生轴异常时，凸轮动作将继续。

主轴位置



运动指令重新执行

切换动作中的凸轮表示，可以重新执行本指令。要变更凸轮表的切换以外的条件时，请对本指令执行指令多重启动。

指令重新执行时，InCam（凸轮动作中）、InSync（同步中）将保持重新执行前的状态。在 InSync（同步中）过程中重新执行时，将从比根据当前位置计算得到的相位更大的相位开始凸轮动作。

根据当前位置计算得到的相位是指对凸轮数据进行直线插补时计算得到的相位。

此外，重新执行对象仅限凸轮表。



使用注意事项

在凸轮动作过程中，通过指令重新执行切换凸轮表时，重新执行前后，从轴的速度或加速度可能发生剧烈变化。

可能对机械造成影响，重新启动时需注意。

运动指令重新执行的详情请参考 □《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 重新执行方法

重新执行方法请参考 □「示例程序 1(P.3-187)」及 □「示例程序 2(P.3-198)」。

运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 本指令执行过程中其他指令的启动

执行本指令的过程中，不可多重启动使用同一 CamTable（凸轮表）的 MC_GenerateCamTable（凸轮表生成）指令。

此外，对本指令执行运动指令多重启动时，请指定从轴。

在执行本指令的过程中多重启动其他指令时，根据缓冲设定，有以下限制。

- 中断并多重启动其他指令时，将立即中断凸轮动作，开始下一动作。
- 缓冲并多重启动其他指令时，在 EndOfProfile（凸轮周期完成）变为 TRUE 时，输出下一动作的指令值。

● 本指令相互共混的多重启动

- 仅在本指令之间多重启动时，可指定共混。执行其他指令的过程中，不可以共混的方式多重启动本指令。
- Master（主轴）和 ReferenceType（位置种类选择）请指定为与当前执行中的指令相同的值。若指定为不同的值，将发生“运动指令不可多重启动”。
- 从当前正在执行的指令正常完成后的周期起，已缓冲的本指令的输出变量 InSync（同步中）将变为 TRUE，从轴不会停止，直接开始凸轮动作。
- 指定了 StartPosition（凸轮表开始位置）、MasterStartDistance（主轴追踪距离）时，无论指定值如何，在指令启动时，从轴将同时开始凸轮动作。凸轮表起点为当前执行中指令的结束位置。无论指定为 4 中共混中的哪种，动作均相同。

● 其他指令执行过程中本指令的启动

执行 MC_GenerateCamTable（凸轮表生成）的过程中，不可多重启动使用同一 CamTable（凸轮表）的本指令。

● 对主轴或从轴的补偿

对同步控制中的主轴或从轴进行相位补偿的指令有以下 2 个。

- MC_Phasing（主轴相对值相位补偿）
- MC_SyncOffsetPosition（周期性同步位置偏置补偿）

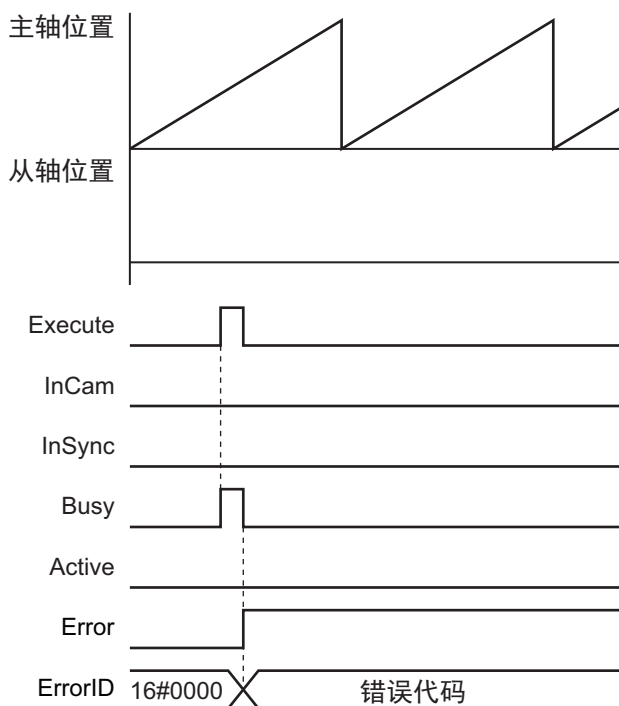
在 MC_CamIn（凸轮动作开始）指令以外的同步控制指令中，因多重启动导致动作中的同步控制指令变为 CommandAborted（执行中断）时，执行相位补偿的指令将同时变为 CommandAborted（执行中断）。以共混方式多重启动 MC_CamIn（凸轮动作开始）指令时，执行相位补偿的指令不会变为 CommandAborted（执行中断），将继续处理。

异常

如果在启动本指令时检测到异常，Error（错误）将变为 TRUE。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

示例程序 1

下面介绍有反复的凸轮动作时的示例程序。
本示例中，主轴为轴 1、从轴为轴 2。



参考

保留 (Reserved) 处理的输入变量不可指定为初始值以外的值。
本示例中，分配了包含各指令定义的初始值的变量，但编程时无需分配变量或参数。

参数设定

下面表示本示例程序中使用的最低限度的设定。

● 设定轴参数

轴种类

轴	轴种类
轴 1	伺服轴 (主轴)
轴 2	伺服轴 (从轴)

计数器模式

轴	计数器模式
轴 1	旋转模式
轴 2	直线模式

循环计数器

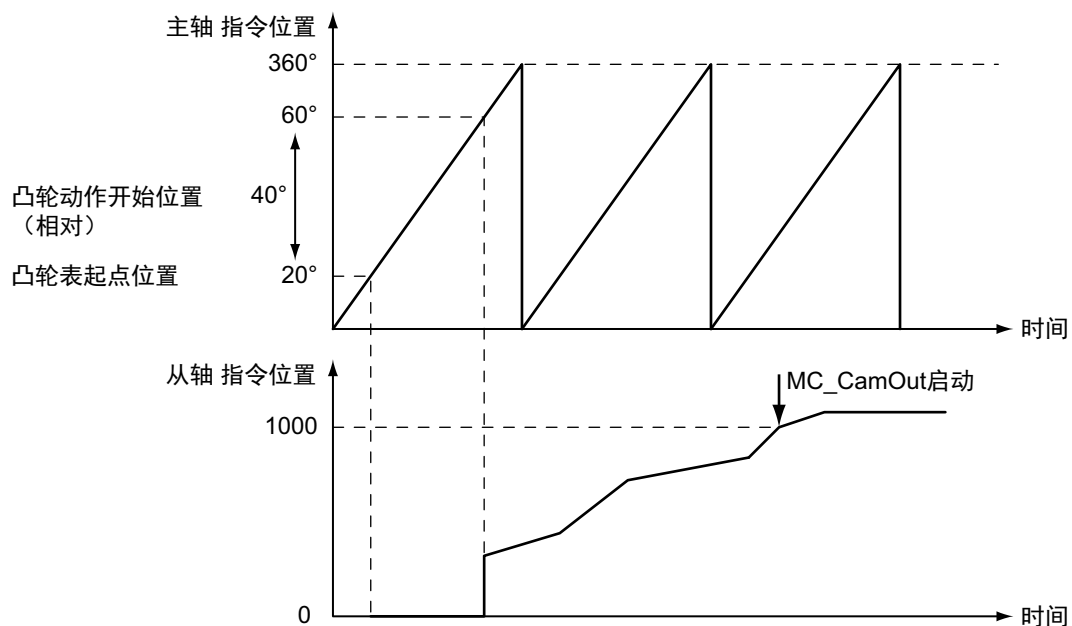
轴	上限值	下限值
轴 1	360	0

坐标单位选择

轴	坐标单位
轴 1	degree
轴 2	mm

动作示例

● 动作模式



1 凸轮动作的开始

将凸轮表起点位置（相位 0 的位置）视为 20°，主轴来到与它的相对角度为 40° 的角度时，从轴开始凸轮动作。凸轮动作以重复动作执行。

将 Periodic（重复模式）设为 TRUE，则变为重复动作。

2 凸轮动作的解除

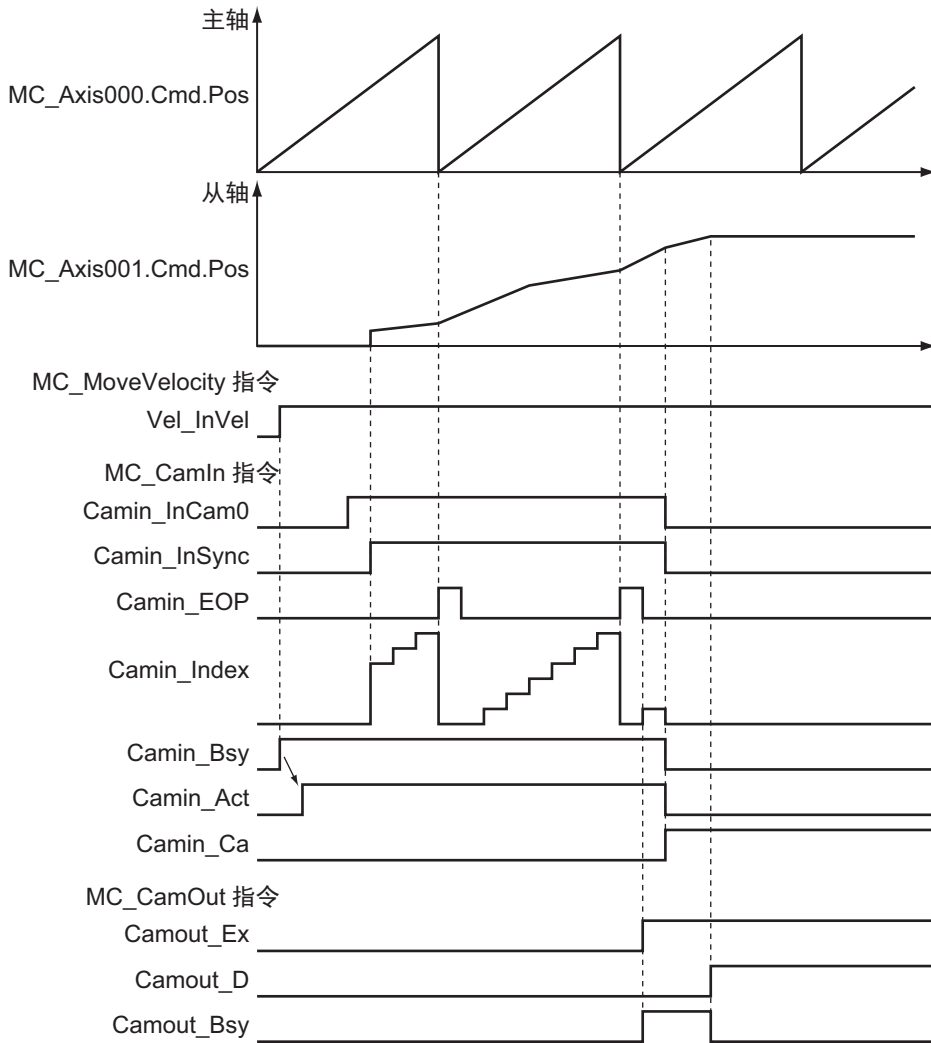
从轴的反馈位置 (MC_Axis001.Act.Pos) 超出 1000.0 后，解除凸轮动作，按减速度 (DecRate2) 停止从轴。

梯形图

● 主要变量

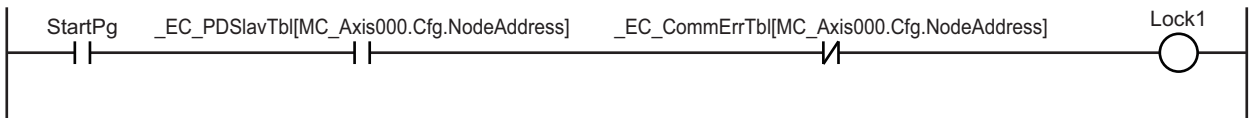
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	主轴轴 1 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
MC_Axis001	_sAXIS_REF	—	从轴轴 2 的轴变量。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴 2 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis001.Details.Homed	BOOL	FALSE	轴 2 处于原点确定状态时变为 TRUE。
Pwr1_S	BOOL	FALSE	分配到 MC_Power 的实例 PWR1 的输出 Status 中的变量。若变为伺服 ON 状态，该变量变为 TRUE。
Pwr2_S	BOOL	FALSE	分配到 MC_Power 的实例 PWR2 的输出 Status 中的变量。若变为伺服 ON 状态，该变量变为 TRUE。
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	—	凸轮数据变量。 排列的要素数 ARRAY[0..N] 在凸轮编辑器中设定。本示例中为 0 ~ 360，排列的要素数因凸轮编辑器的设定不同而异。
DecRate2	LREAL	10000.0	设定执行 MC_CamOut 时的减速度。
Vel_InVel	BOOL	FALSE	分配到 MC_MoveVelocity 的实例 VEL 的输出 InVelocity 中的变量。达到目标速度时，该变量变为 TRUE。
Camin_InCam0	BOOL	FALSE	分配到 MC_CamIn 的实例 CAMIN 的输出 InCam 中的变量。正在执行凸轮动作时，该变量变为 TRUE。
Camout_Ex	BOOL	FALSE	该变量变为 TRUE 时，将启动 MC_CamOut 的实例 CAMOUT。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态，则变为伺服 ON 状态。

● 时序图

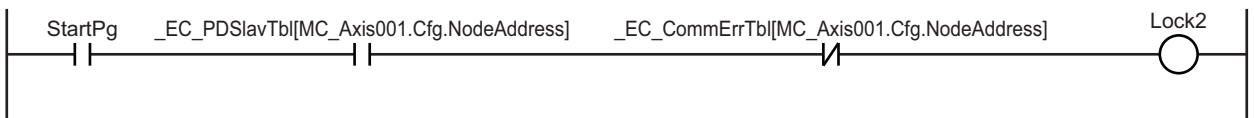


● 示例程序

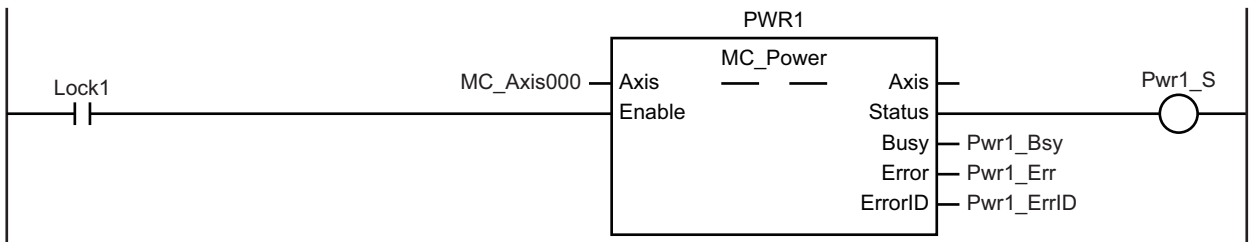
接点 StartPg 变为 TRUE 后，确认轴 1 的 EtherCAT 通信是否正常进行过程数据通信



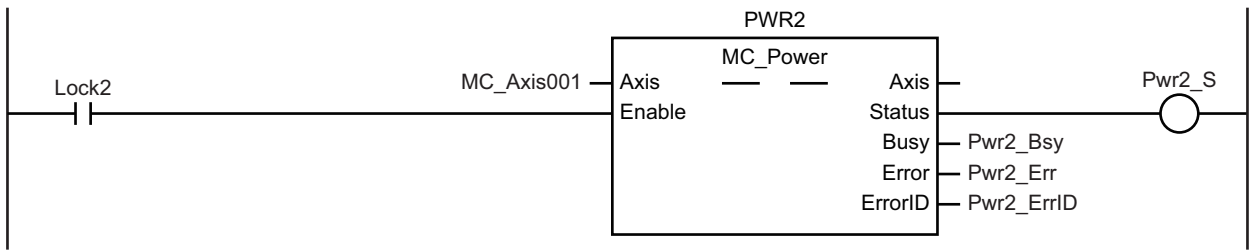
接点 StartPg 变为 TRUE 后，确认轴 2 的 EtherCAT 通信是否正常进行过程数据通信



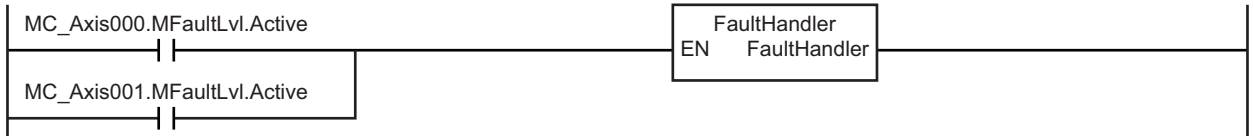
若轴 1 正在进行过程数据通信，则变为伺服 ON 状态



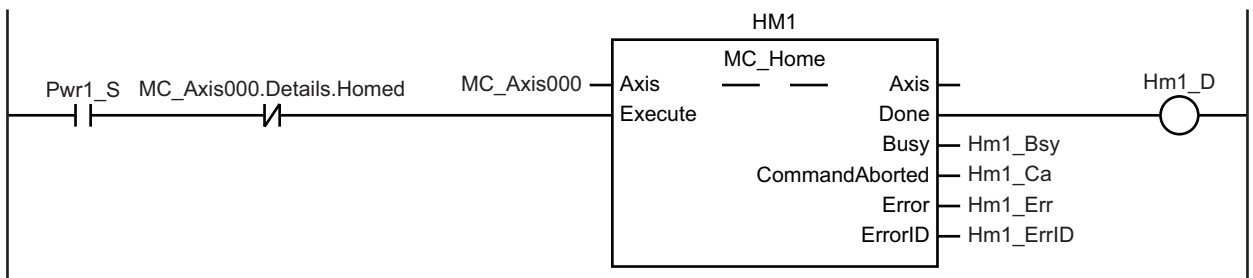
若轴 2 正在进行过程数据通信，则变为伺服 ON 状态



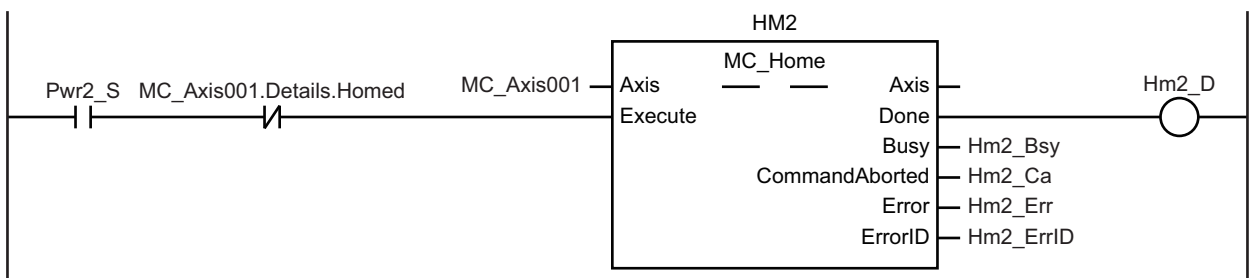
轴构成发生轻度故障等级的异常时，执行装置发生异常时的处理（FaultHandler）。发生异常时的处理（FaultHandler）请根据装置由客户进行编程



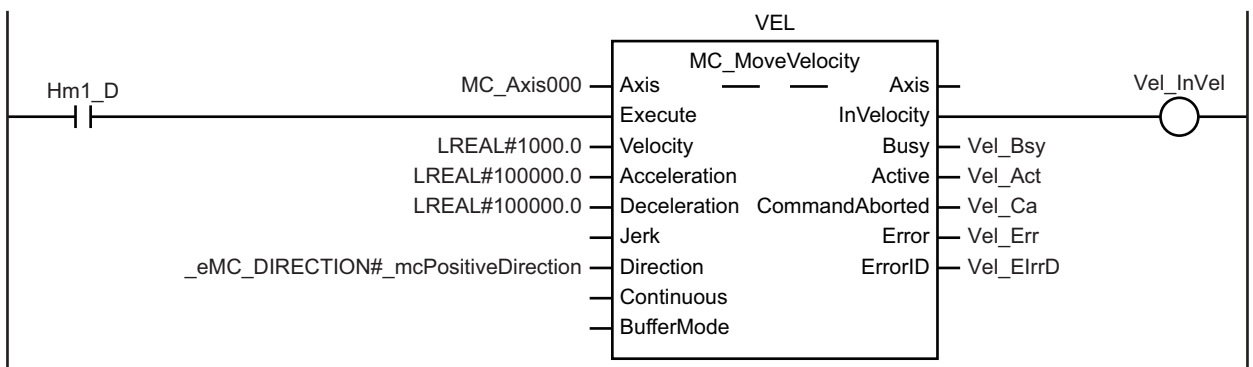
轴 1 为伺服 ON 状态且处于原点未确定状态时，执行原点复位，确定原点



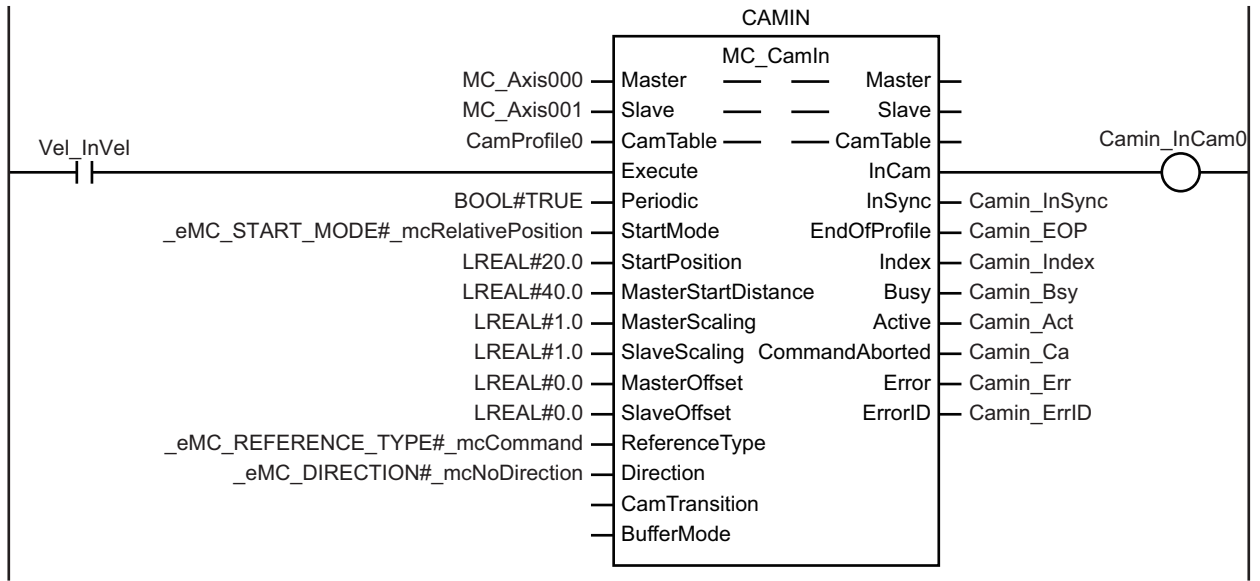
轴 2 为伺服 ON 状态且处于原点未确定状态时，执行原点复位，确定原点



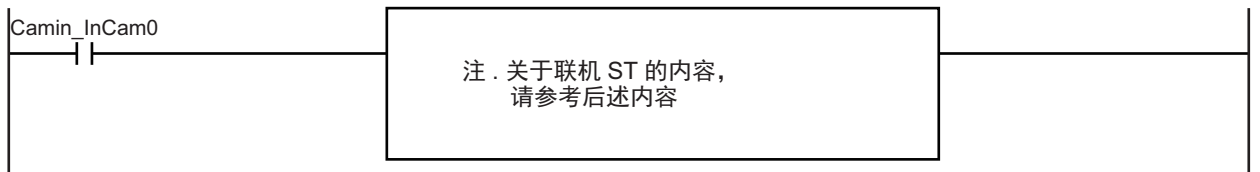
在轴 1 为原点复位完成的状态下，执行 MC_MoveVelocity（速度控制）指令



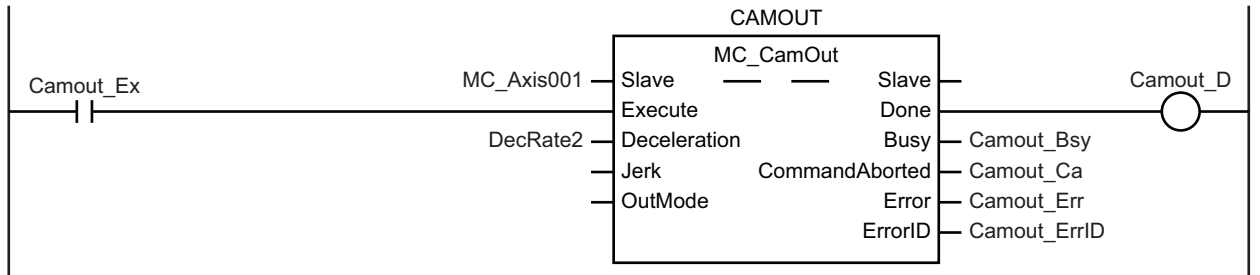
MC_MoveVelocity (速度控制) 指令的 Vel_InVel 变为 TRUE 后, 启动 MC_CamIn (凸轮动作开始) 指令



CamIn_InCam0 为 TRUE, MC_Axis001.Act.Pos 超过 1000 时, 将 CamOut_Ex 设为 TRUE



将 Camout_Ex 设为 TRUE, 启动 MC_CamOut (凸轮动作解除) 指令。
指定 DecRate2 作为减速度



联机 ST 的内容

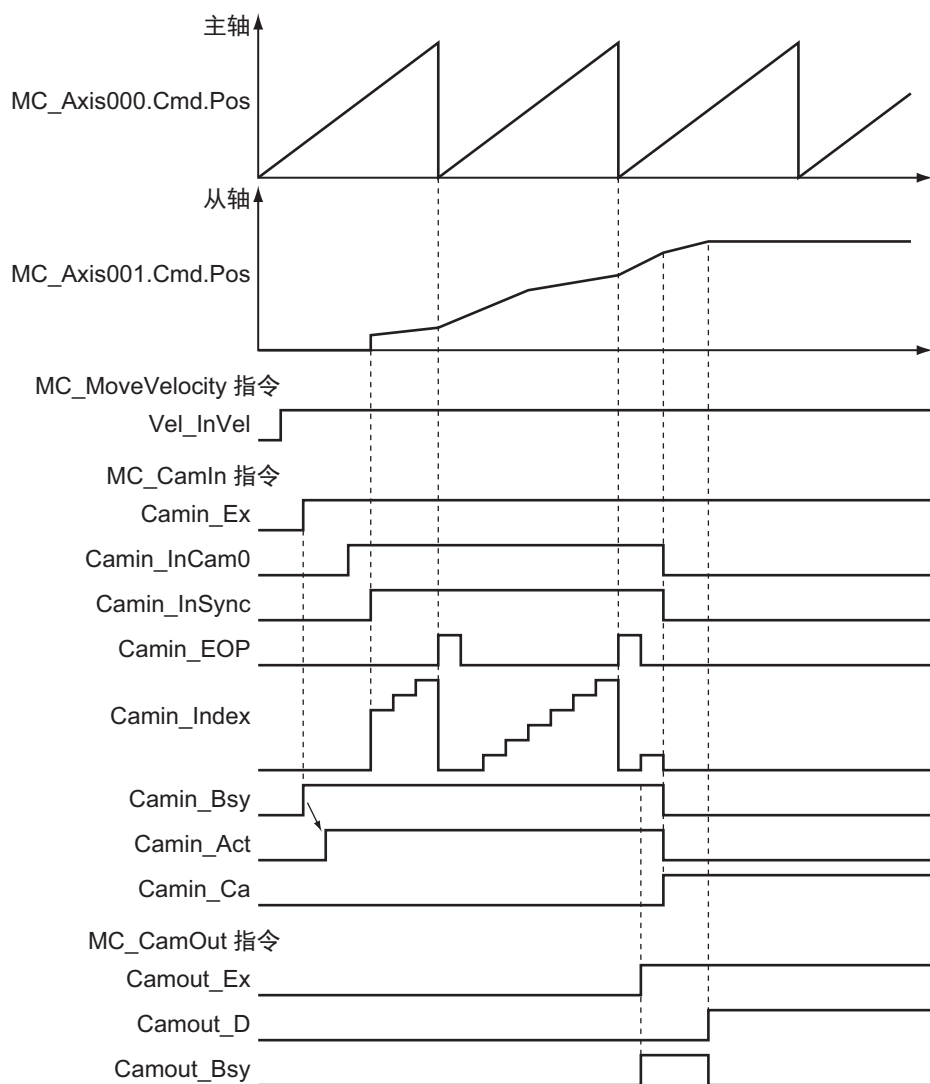
```
IF MC_Axis001.Act.Pos>LREAL#1000.0 THEN
  Camout_Ex := TRUE;
END_IF;
```

结构文本 (ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	主轴轴 1 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时, 变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
MC_Axis001	_sAXIS_REF	—	从轴轴 2 的轴变量。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴 2 发生轻度故障等级的异常时, 变为 TRUE。
MC_Axis001.Details.Homed	BOOL	FALSE	轴 2 处于原点确定状态时变为 TRUE。
Pwr1_S	BOOL	FALSE	分配到 MC_Power 的实例 PWR1 的输出 Status 中的变量。若变为伺服 ON 状态, 该变量变为 TRUE。
Pwr2_S	BOOL	FALSE	分配到 MC_Power 的实例 PWR2 的输出 Status 中的变量。若变为伺服 ON 状态, 该变量变为 TRUE。
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	—	凸轮数据变量。 排列的要素数 ARRAY[0..N] 在凸轮编辑器中设定。本示例中为 0 ~ 360, 排列的要素数因凸轮编辑器的设定不同而异。
DecRate2	LREAL	10000.0	设定执行 MC_CamOut 时的减速度。
Vel_InVel	BOOL	FALSE	分配到 MC_MoveVelocity 的实例 VEL 的输出 InVelocity 中的变量。达到目标速度时, 该变量变为 TRUE。
Camin_InCam0	BOOL	FALSE	分配到 MC_CamIn 的实例 CAMIN 的输出 InCam 中的变量。正在执行凸轮动作时, 该变量变为 TRUE。
Camout_Ex	BOOL	FALSE	该变量变为 TRUE 时, 将启动 MC_CamOut 的实例 CAMOUT。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态, 则变为伺服 ON 状态。
Camin_Ex	BOOL	FALSE	若在 Vel_InVel 的上升沿设为 TRUE, 则将 MC_CamIn 的实例 CAMIN 的 Execute 变为 TRUE。
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE 时, 设定输入参数。输入参数设定完成后, 变为 TRUE。

● 时序图



● 示例程序

```
// 未设定输入参数时的处理
IF InitFlag=FALSE THEN
```

```
//MC_CamIn 参数
Camin_EM := TRUE; // 重复模式
Camin_StMode := _eMC_START_MODE#_mcRelativePosition;
Camin_StPos := LREAL#20.0; // 起点的主轴绝对定位
Camin_MStDis := LREAL#40.0; // 开始凸轮动作的主轴位置
Camin_MSc := LREAL#1.0; // 主轴放大
Camin_SSc := LREAL#1.0; // 从轴放大
Camin_MO := LREAL#0.0; // 主轴偏置
Camin_SO := LREAL#0.0; // 从轴偏置
Camin_RT := _eMC_REFERENCE_TYPE#_mcCommand; // 位置种类选择
Camin_Dir := _eMC_DIRECTION#_mcNoDirection; // 方向选择

//MC_MoveVelocity 参数
Vel_Vel := LREAL#1000.0;
Vel_Acc := LREAL#100000.0;
Vel_Dec := LREAL#100000.0;
Vel_Dir := _eMC_DIRECTION#_mcPositiveDirection;

//MC_CamOut 参数
```

```

Camout_Dec := DecRate2;

// 设定输入参数后, 将 InitFlag 设为 TRUE
InitFlag := TRUE;

END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 1 设为伺服 ON 状态
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
  Pwr1_En:=TRUE;
ELSE
  Pwr1_En:=FALSE;
END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 2 设为伺服 ON 状态
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
  Pwr2_En:=TRUE;
ELSE
  Pwr2_En:=FALSE;
END_IF;

// 轴 1 ~ 轴 2 发生轻度故障时, 执行异常处理。
// 发生异常时的处理 (FaultHandler) 请根据装置由客户进行编程。
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
  FaultHandler();
END_IF;

// 轴 1 为伺服 ON 状态且处于原点未确定状态时, 执行轴 1 的原点复位
IF (Pwr1_S=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
  Hm1_Ex:=TRUE;
END_IF;

// 轴 2 为伺服 ON 状态且处于原点未确定状态时, 执行轴 2 的原点复位
IF (Pwr2_S=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
  Hm2_Ex:=TRUE;
END_IF;

// 在轴 1 为原点复位完成的状态下, 执行 MC_MoveVelocity
IF Hm1_D=TRUE THEN
  Vel_Ex := TRUE;
END_IF;

// MC_MoveVelocity 的 InVel=TRUE 时启动 CamIn
IF Vel_InVel=TRUE THEN
  Camin_Ex := TRUE;
END_IF;

```

```
// Camin_InCam0=TRUE 且 MC_Axis001.Act.Pos>1000 时, 启动 CamOut。
IF (Camin_InCam0=TRUE) AND (MC_Axis001.Act.Pos>LREAL#1000.0) THEN
  Camout_Ex := TRUE;
END_IF;
```

```
// 轴 1 的 MC_Power
PWR1(
  Axis      := MC_Axis000,
  Enable    := Pwr1_En,
  Status    => Pwr1_S,
  Busy      => Pwr1_Bsy,
  Error     => Pwr1_Err,
  ErrorID   => Pwr1_ErrID
);
```

```
// 轴 2 的 MC_Power
PWR2(
  Axis      := MC_Axis001,
  Enable    := Pwr2_En,
  Status    => Pwr2_S,
  Busy      => Pwr2_Bsy,
  Error     => Pwr2_Err,
  ErrorID   => Pwr2_ErrID
);
```

```
// 轴 1 的 MC_Home
HM1(
  Axis          := MC_Axis000,
  Execute       := Hm1_Ex,
  Done          => Hm1_D,
  Busy          => Hm1_Bsy,
  CommandAborted => Hm1_Ca,
  Error         => Hm1_Err,
  ErrorID       => Hm1_ErrID
);
```

```
// 轴 2 的 MC_Home
HM2(
  Axis          := MC_Axis001,
  Execute       := Hm2_Ex,
  Done          => Hm2_D,
  Busy          => Hm2_Bsy,
  CommandAborted => Hm2_Ca,
  Error         => Hm2_Err,
  ErrorID       => Hm2_ErrID
);
```

```
//MC_MoveVelocity
VEL(
  Axis          := MC_Axis000,
  Execute       := Vel_Ex,
  Velocity      := Vel_Vel,
  Acceleration  := Vel_Acc,
  Deceleration  := Vel_Dec,
```



```

Direction      := Vel_Dir,
InVelocity     => Vel_InVel,
Busy           => Vel_Bsy,
Active         => Vel_Act,
CommandAborted => Vel_Ca,
Error          => Vel_Err,
ErrorID        => Vel_ErrID
);

//MC_CamIn
CAMIN(
  Master      := MC_Axis000,
  Slave       := MC_Axis001,
  CamTable    := CamProfile0,
  Execute     := Camin_Ex,
  Periodic    := Camin_EM,
  StartMode   := Camin_StMode,
  StartPosition := Camin_StPos,
  MasterStartDistance := Camin_MStDis,
  MasterScaling := Camin_MSc,
  SlaveScaling := Camin_SSc,
  MasterOffset := Camin_MO,
  SlaveOffset := Camin_SO,
  ReferenceType := Camin_RT,
  Direction   := Camin_Dir,
  CamTransition := Camin_CT,
  BufferMode   := Camin_BM,
  InCam       => Camin_InCam0,
  InSync      => Camin_InSync,
  EndOfProfile => Camin_EOP,
  Index       => Camin_Index,
  Busy        => Camin_Bsy,
  Active      => Camin_Act,
  CommandAborted => Camin_Ca,
  Error       => Camin_Err,
  ErrorID     => Camin_ErrID
);

//MC_CamOut
CAMOUT(
  Slave      := MC_Axis001,
  Execute    := Camout_Ex,
  Deceleration := Camout_Dec,
  Done       => Camout_D,
  Busy       => Camout_Bsy,
  CommandAborted => Camout_Ca,
  Error      => Camout_Err,
  ErrorID    => Camout_ErrID
);

```

示例程序 2

下面介绍液体填充凸轮动作时的示例程序。



参考

保留 (Reserved) 处理的输入变量不可指定为初始值以外的值。
本示例中不指定参数。

参数设定

下面表示本示例程序中使用的最低限度的设定。

● 设定轴参数 轴种类

轴	轴种类
轴 1	伺服轴 (主轴)
轴 2	伺服轴 (从轴)
轴 3	伺服轴 (从轴)
轴 4	伺服轴 (从轴)
轴 5	伺服轴 (从轴)

计数器模式

轴	计数器模式
轴 1	旋转模式
轴 2	旋转模式
轴 3	旋转模式
轴 4	旋转模式
轴 5	旋转模式

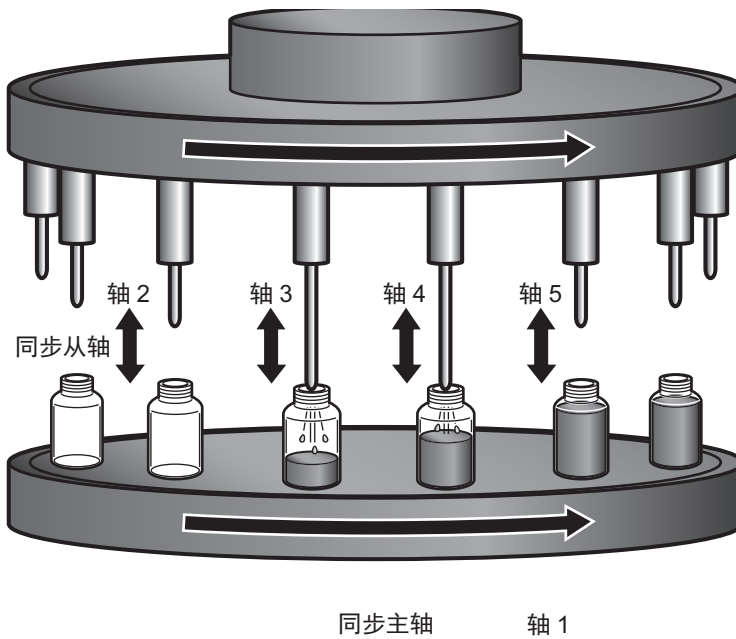
循环计数器

轴	上限值	下限值
轴 1	360	0
轴 2	360	0
轴 3	360	0
轴 4	360	0
轴 5	360	0

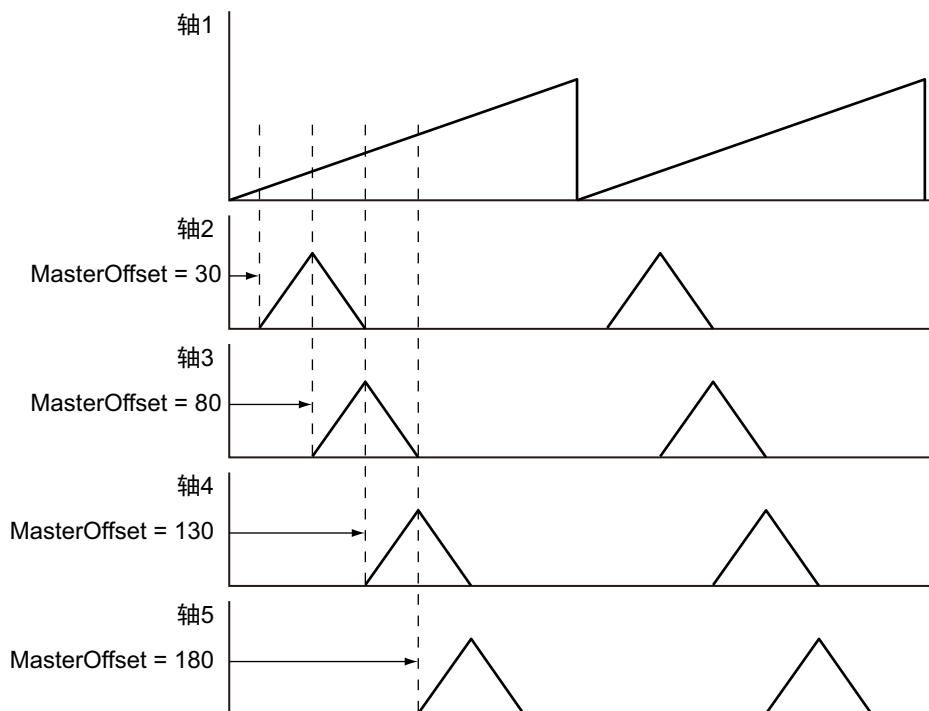
坐标单位选择

轴	坐标单位
轴 1	degree
轴 2	degree
轴 3	degree
轴 4	degree
轴 5	degree

动作示例



● 动作模式



1 凸轮动作开始

与主轴（轴 1）同步，从轴（轴 2 ~ 轴 5）执行凸轮动作。各轴以 50degree 为单位移动相位，开始凸轮动作。

2 重复

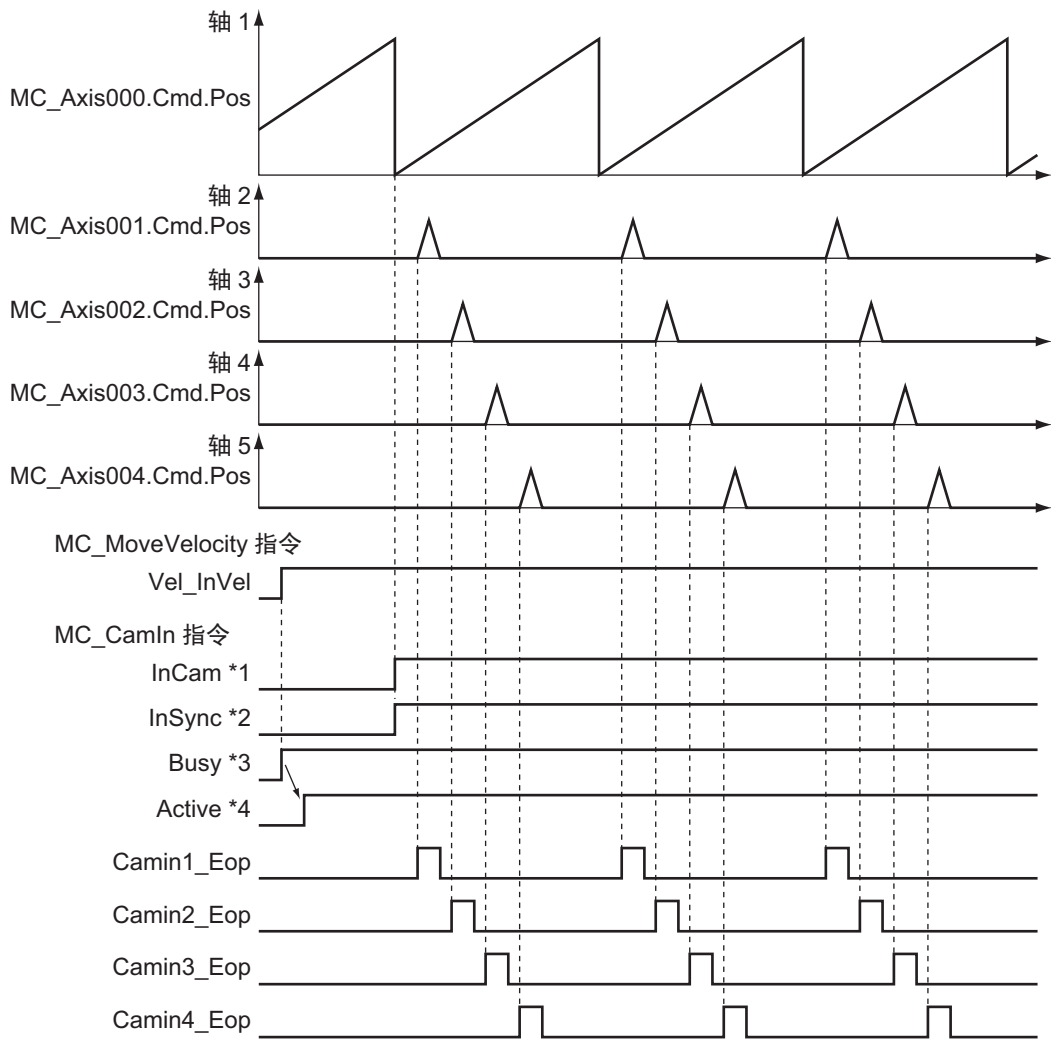
各轴重复执行指定的凸轮动作。

梯形图

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	主轴轴 1 的轴变量。
MC_Axis001	_sAXIS_REF	—	从轴轴 2 的轴变量。
MC_Axis002	_sAXIS_REF	—	从轴轴 3 的轴变量。
MC_Axis003	_sAXIS_REF	—	从轴轴 4 的轴变量。
MC_Axis004	_sAXIS_REF	—	从轴轴 5 的轴变量。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态，则变为伺服 ON 状态。
Vel_InVel	BOOL	FALSE	分配到 MC_MoveVelocity 的实例 VEL 的输出 InVelocity 中的变量。达到目标速度时，该变量变为 TRUE。
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	—	凸轮数据变量。 将 MC_CamIn 的实例 CAMIN1 ~ CAMIN4 的输入 CamTable 指定为本变量。 排列的要素数 ARRAY[0..N] 在凸轮编辑器中设定。

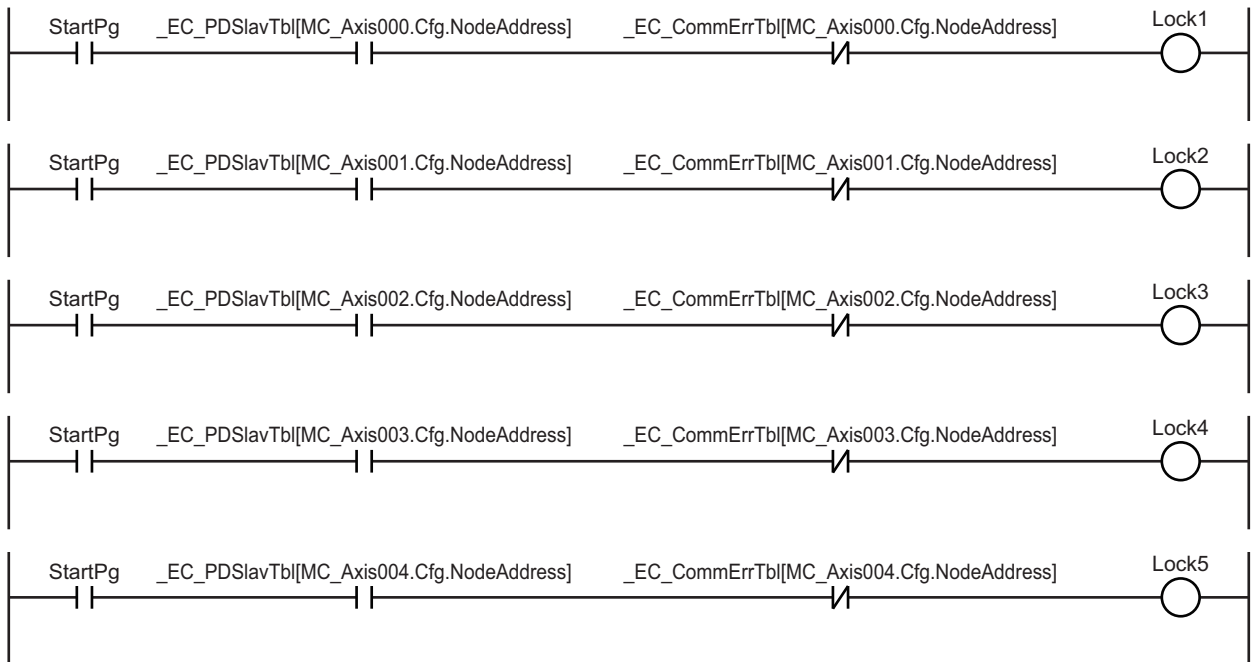
● 时序图



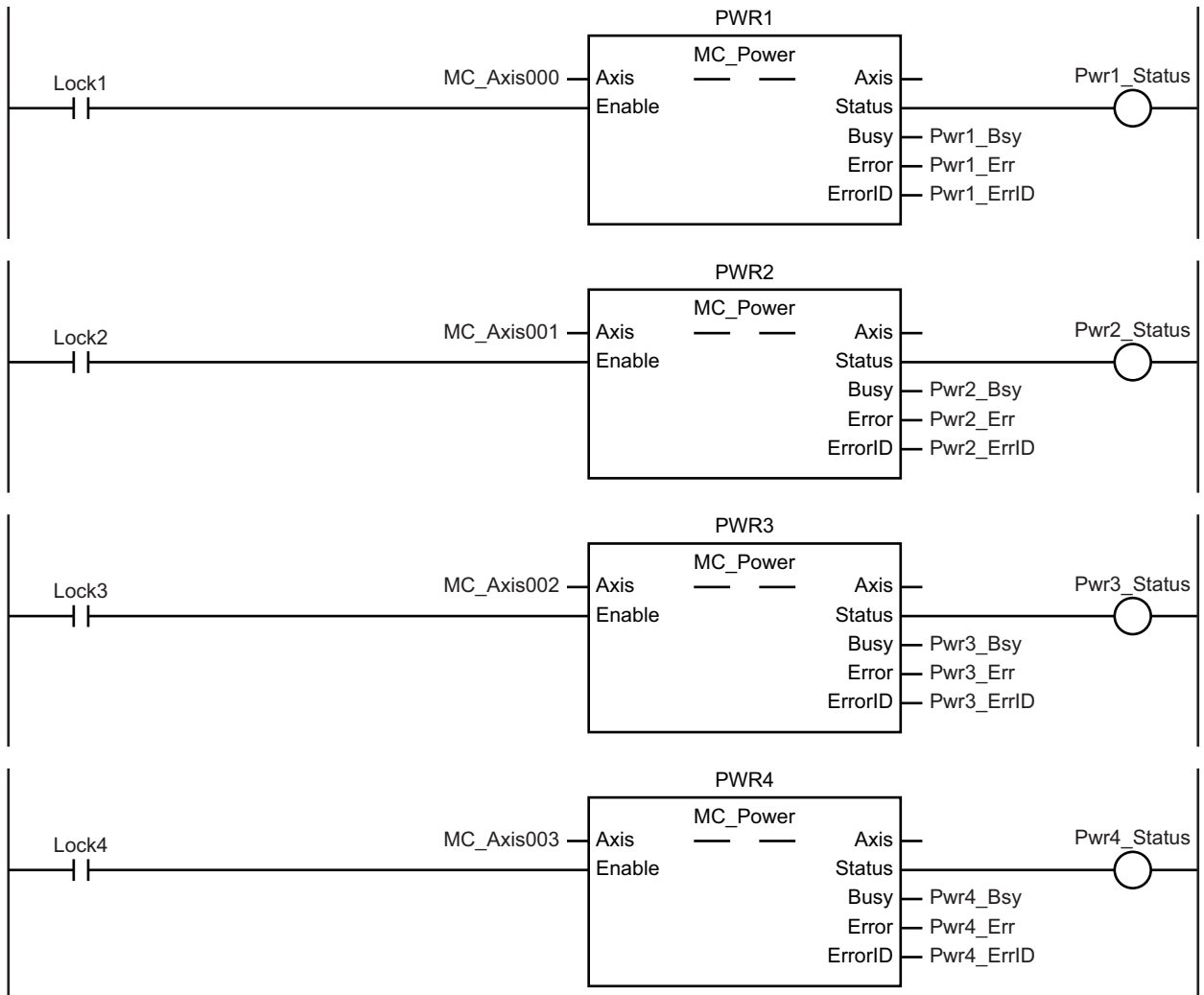
- *1 Camin1_InCam, Camin2_InCam, Camin3_InCam, Camin4_InCam 的时序相同。
- *2 Camin1_InSync, Camin2_InSync, Camin3_InSync, Camin4_InSync 的时序相同。
本示例中，MasterStartDistance 为“0”，因此主轴从“0°”开始，InSync 为 TRUE。
- *3 Camin1_Bsy, Camin2_Bsy, Camin3_Bsy, Camin4_Bsy 的时序相同。
- *4 Camin1_Act, Camin2_Act, Camin3_Act, Camin4_Act 的时序相同。

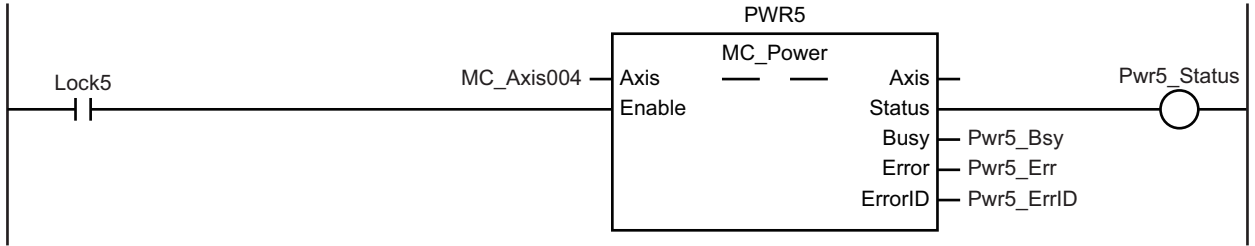
● 示例程序

接点 StartPg 变为 TRUE 后，确认各轴的 EtherCAT 通信是否正常进行过程数据通信

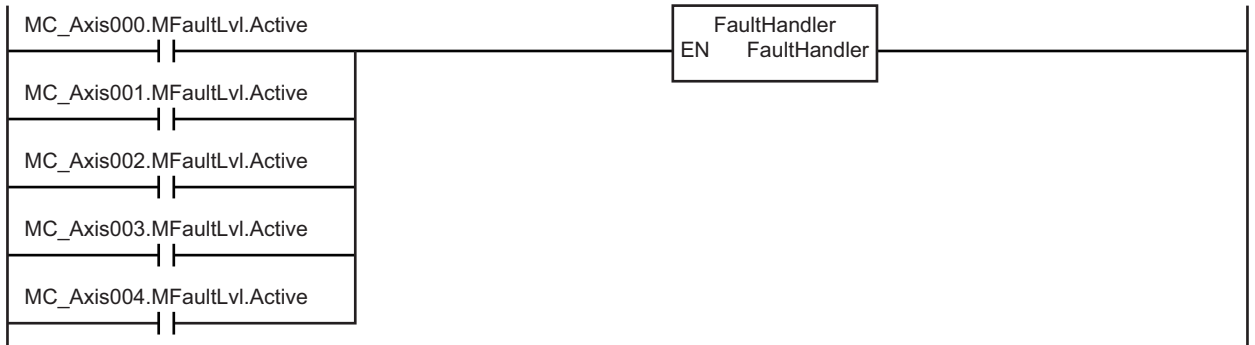


若正在进行过程数据通信，则将各轴设为伺服 ON 状态

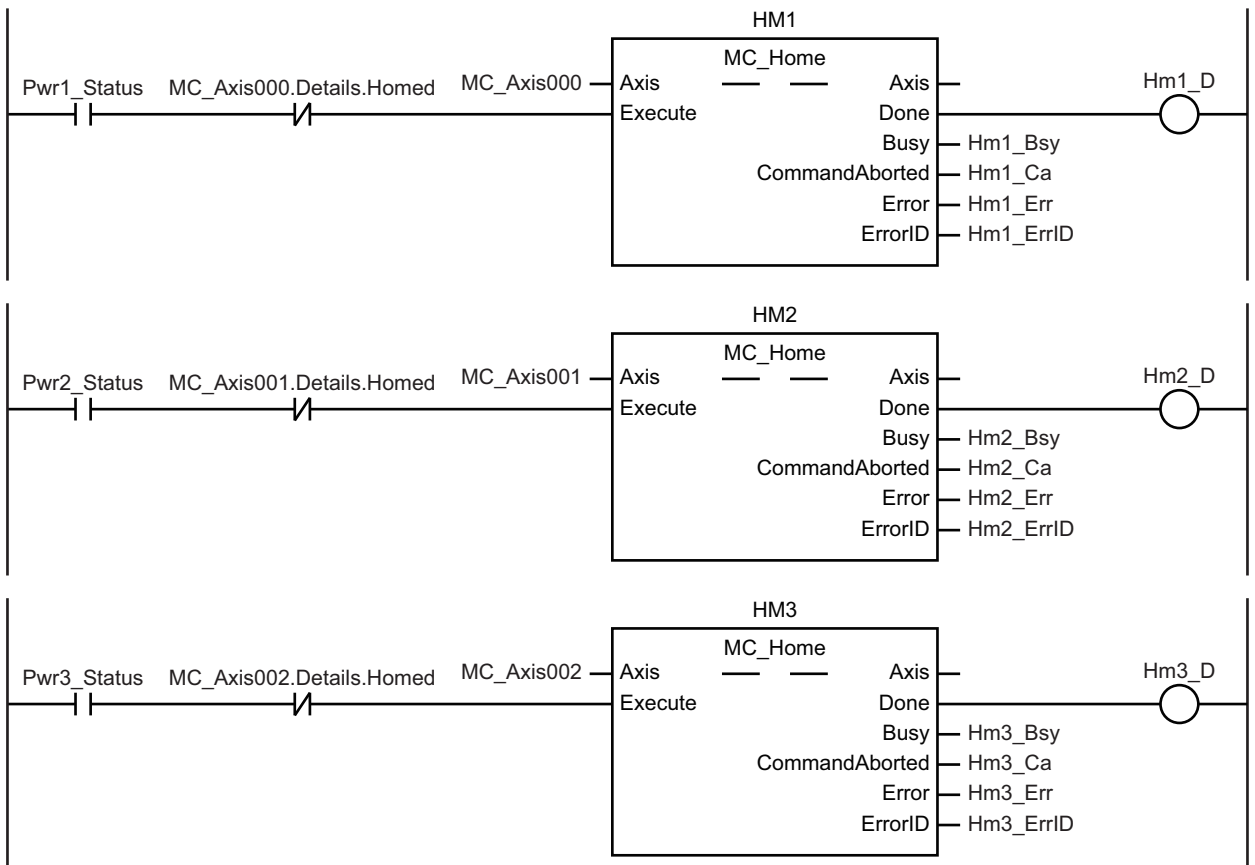


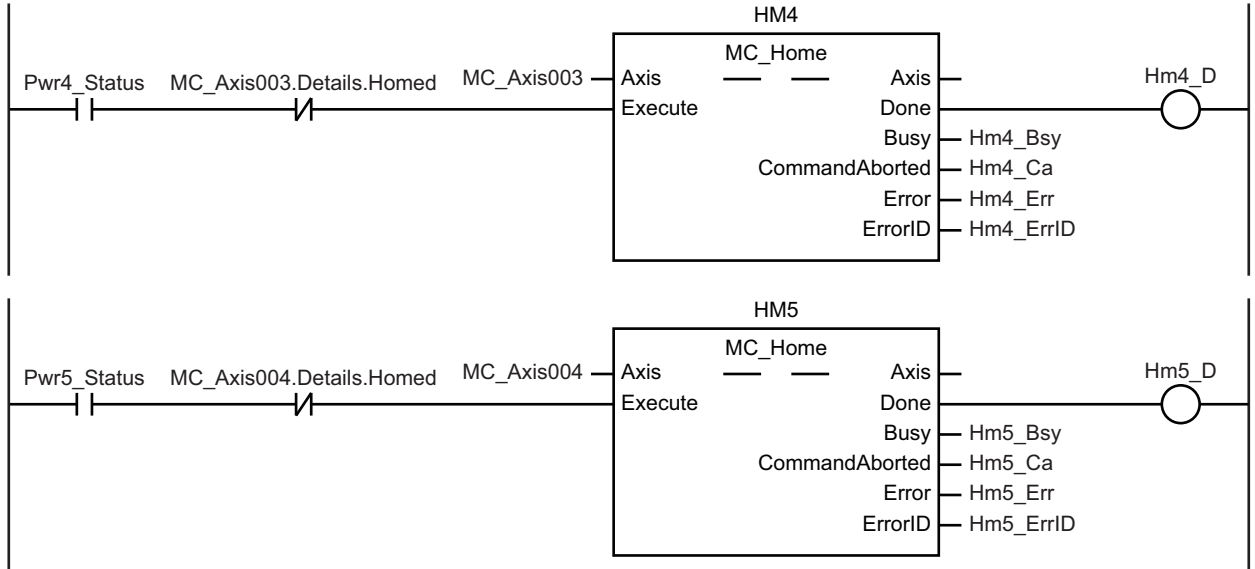


轴构成发生轻度故障等级的异常时，执行装置发生异常时的处理（FaultHandler）。
发生异常时的处理（FaultHandler）请根据装置由客户进行编程

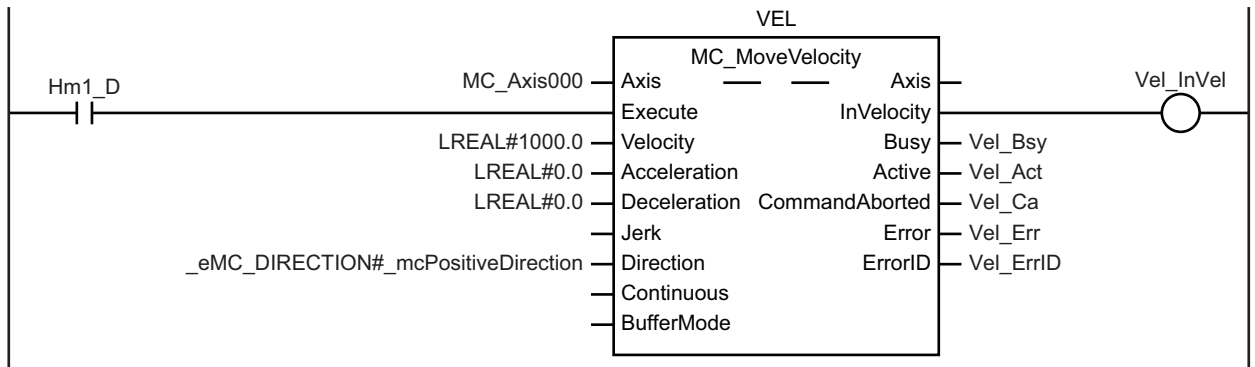


轴 1 ~ 轴 5 为伺服 ON 状态且处于原点未确定状态时，各轴分别执行原点复位

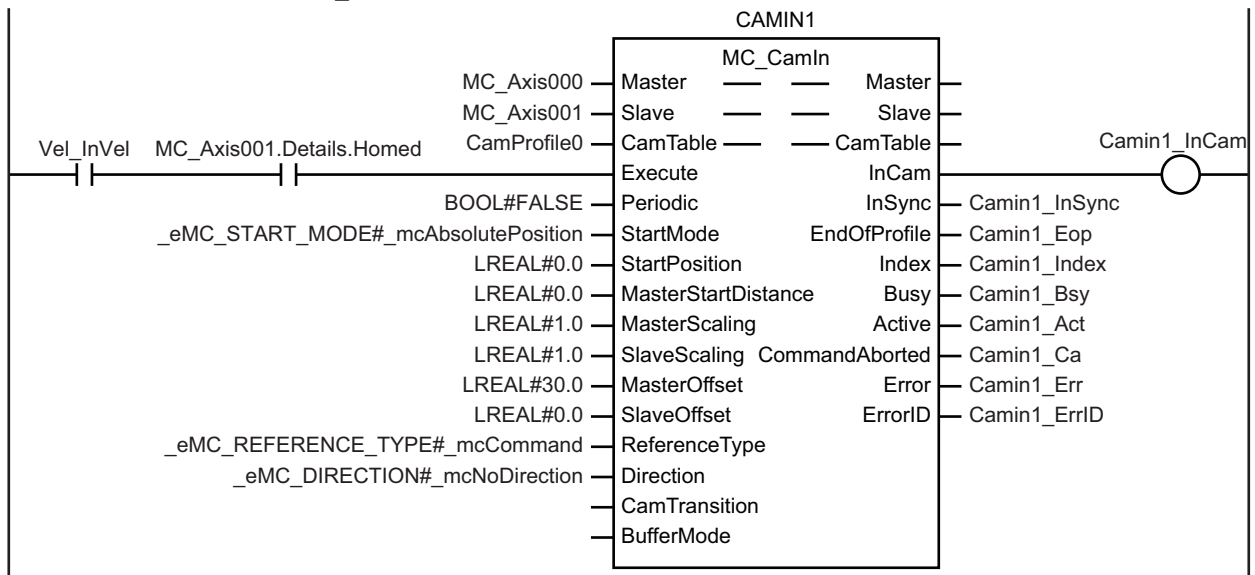




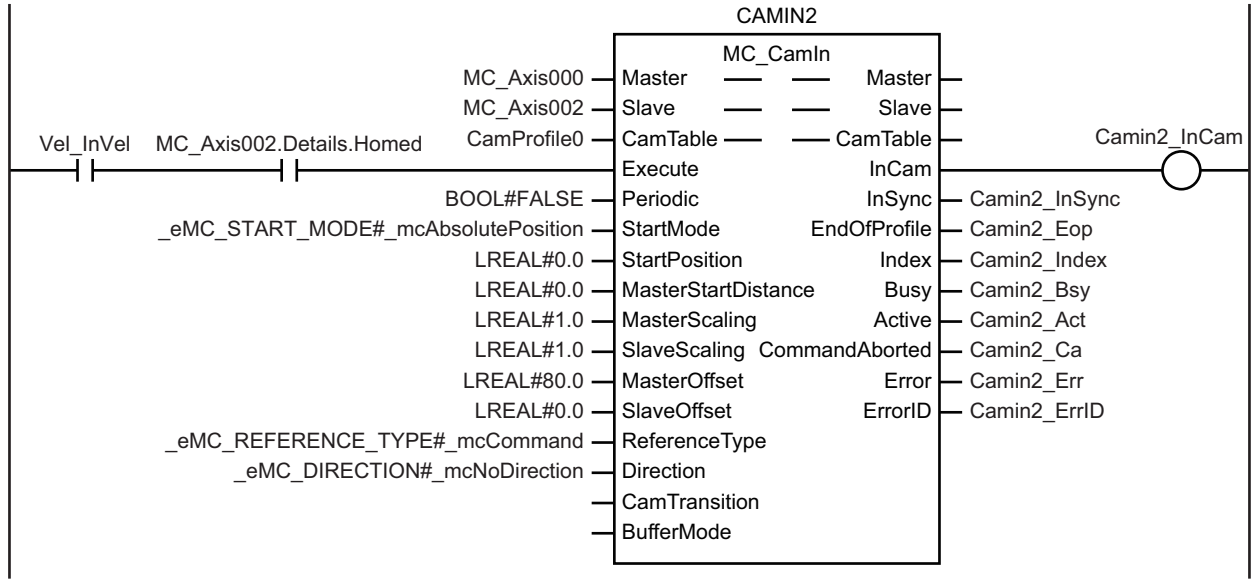
轴 1 完成原点复位后，启动 MC_MoveVelocity（速度控制）指令



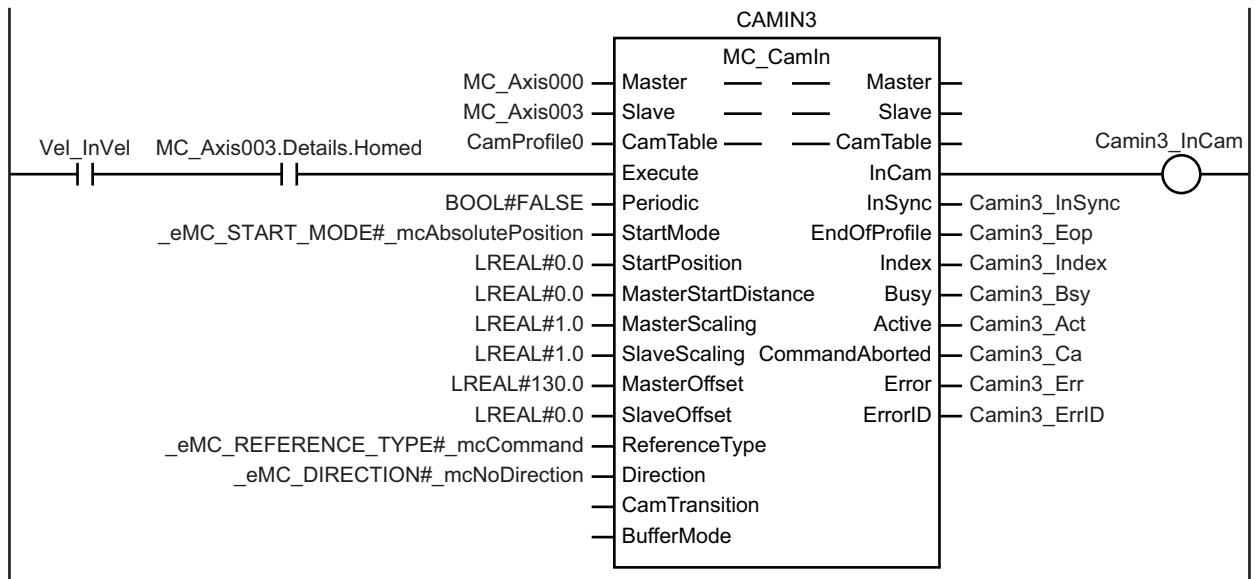
MC_MoveVelocity（速度控制）指令的 Vel_InVel 为 TRUE，且轴 2 完成原点复位后，执行从轴（轴 2）中设定为 MC_CamIn（凸轮动作开始）指令



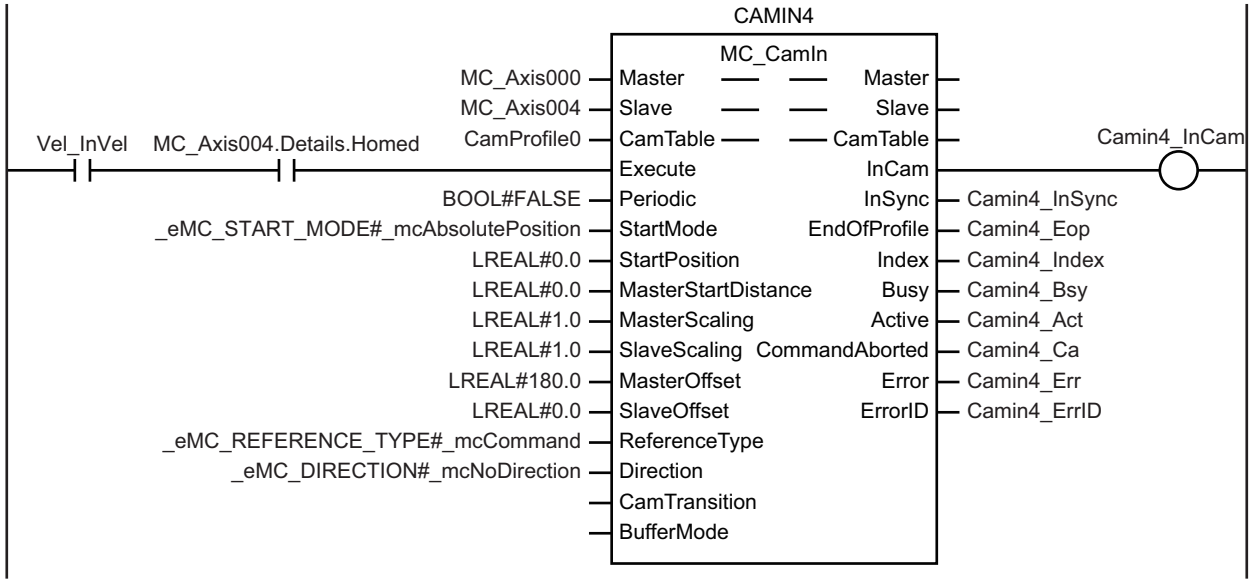
MC MoveVelocity（速度控制）指令的 Vel_InVel 为 TRUE，且轴 3 完成原点复位后，
执行从轴（轴 3）中设定为 MC_CamIn（凸轮动作开始）指令



MC MoveVelocity（速度控制）指令的 Vel_InVel 为 TRUE，且轴 4 完成原点复位后，
执行从轴（轴 4）中设定为 MC_CamIn（凸轮动作开始）指令



MC_MoveVelocity（速度控制）指令的 Vel_InVel 为 TRUE，且轴 5 完成原点复位后，执行从轴（轴 5）中设定为 MC_CamIn（凸轮动作开始）指令

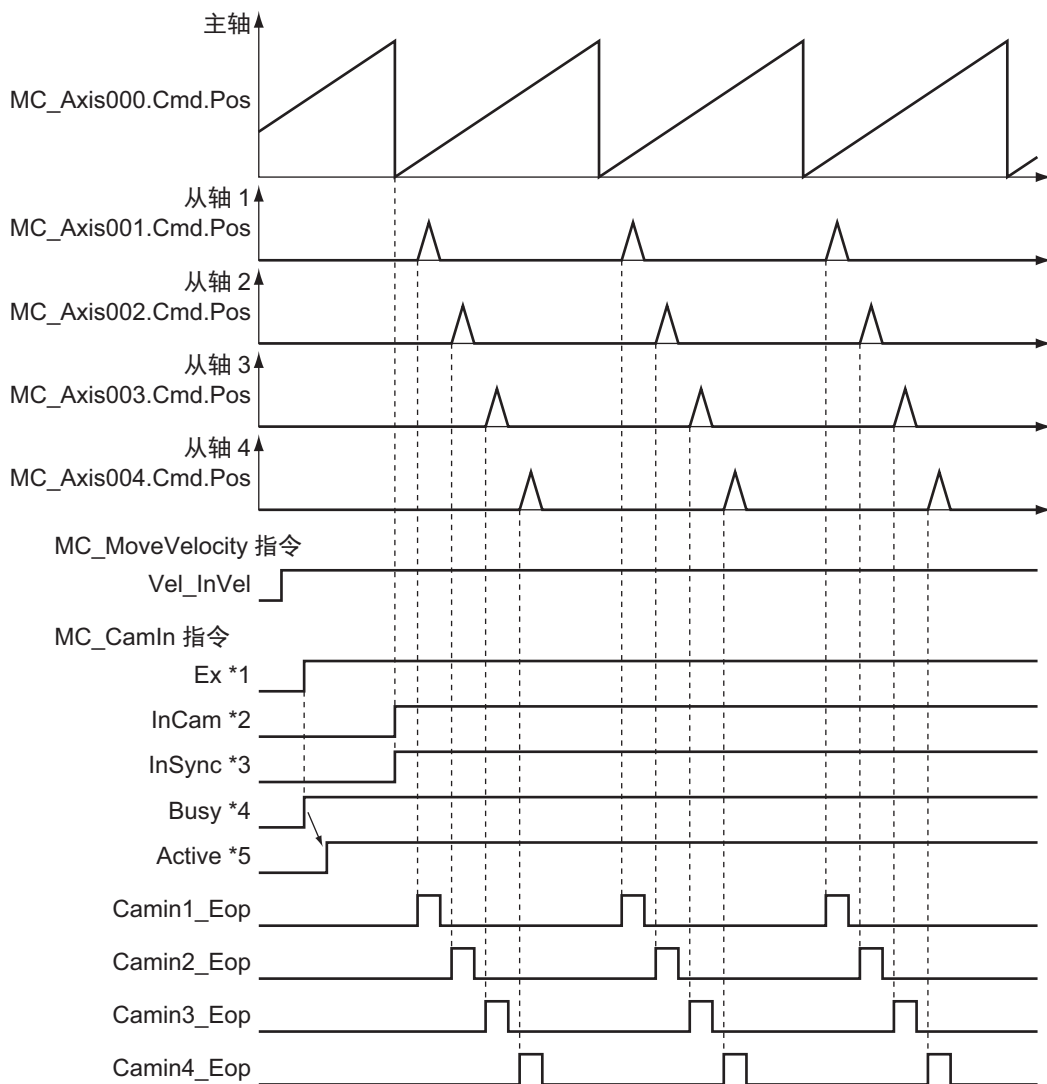


结构文本（ST）

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	主轴轴 1 的轴变量。
MC_Axis001	_sAXIS_REF	—	从轴轴 2 的轴变量。
MC_Axis002	_sAXIS_REF	—	从轴轴 3 的轴变量。
MC_Axis003	_sAXIS_REF	—	从轴轴 4 的轴变量。
MC_Axis004	_sAXIS_REF	—	从轴轴 5 的轴变量。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态，则变为伺服 ON 状态。
Vel_InVel	BOOL	FALSE	分配到 MC_MoveVelocity 的实例 VEL 的输出 InVelocity 中的变量。达到目标速度时，该变量变为 TRUE。
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	—	凸轮数据变量。将 MC_CamIn 的实例 CAMIN1 ~ CAMIN4 的输入 CamTable 指定为本变量。排列的要素数 ARRAY[0..N] 在凸轮编辑器中设定。
Camin1_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时，启动 MC_CamIn 的实例 CAMIN1 ~ CAMIN4。
Camin2_Ex	BOOL	FALSE	
Camin3_Ex	BOOL	FALSE	
Camin4_Ex	BOOL	FALSE	
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE 时，设定输入参数。 输入参数设定完成后，变为 TRUE。

● 时序图



- *1 Camin1_InCam, Camin2_InCam, Camin3_InCam, Camin4_InCam 的时序相同。
- *2 Camin1_InSync, Camin2_InSync, Camin3_InSync, Camin4_InSync 的时序相同。
本示例中，MasterStartDistance 为“0”，因此主轴从“0°”开始，InSync 为 TRUE。
- *3 Camin1_Bsy, Camin2_Bsy, Camin3_Bsy, Camin4_Bsy 的时序相同。
- *4 Camin1_Act, Camin2_Act, Camin3_Act, Camin4_Act 的时序相同。

● 示例程序

```
// 未设定输入参数时的处理
IF InitFlag=FALSE THEN
```

```
  //MC_MoveVelocity Input Parameter
  Vel_Vel := LREAL#1000.0;
  Vel_Acc := LREAL#0.0;
  Vel_Dec := LREAL#0.0;
  Vel_Dir := _eMC_DIRECTION#_mcPositiveDirection;
```

```
  //MC_CamIn Input Parameter
  Camin1_Em := TRUE;
  Camin1_Sm := _eMC_START_MODE#_mcAbsolutePosition;
  Camin1_Sp := LREAL#0.0;
  Camin1_Msd := LREAL#0.0;
  Camin1_Ms := LREAL#1.0;
  Camin1_Ss := LREAL#1.0;
```

```

Camin1_Mo := LREAL#30.0;
Camin1_So := LREAL#0.0;
Camin1_Rt := _eMC_REFERENCE_TYPE#_mcCommand;
Camin1_Dir := _eMC_DIRECTION#_mcNoDirection;

Camin2_Em := TRUE;
Camin2_Sm := _eMC_START_MODE#_mcAbsolutePosition;
Camin2_Sp := LREAL#0.0;
Camin2_Msd := LREAL#0.0;
Camin2_Ms := LREAL#1.0;
Camin2_Ss := LREAL#1.0;
Camin2_Mo := LREAL#80.0;
Camin2_So := LREAL#0.0;
Camin2_Rt := _eMC_REFERENCE_TYPE#_mcCommand;
Camin2_Dir := _eMC_DIRECTION#_mcNoDirection;

Camin3_Em := TRUE;
Camin3_Sm := _eMC_START_MODE#_mcAbsolutePosition;
Camin3_Sp := LREAL#0.0;
Camin3_Msd := LREAL#0.0;
Camin3_Ms := LREAL#1.0;
Camin3_Ss := LREAL#1.0;
Camin3_Mo := LREAL#130.0;
Camin3_So := LREAL#0.0;
Camin3_Rt := _eMC_REFERENCE_TYPE#_mcCommand;
Camin3_Dir := _eMC_DIRECTION#_mcNoDirection;

Camin4_Em := TRUE;
Camin4_Sm := _eMC_START_MODE#_mcAbsolutePosition;
Camin4_Sp := LREAL#0.0;
Camin4_Msd := LREAL#0.0;
Camin4_Ms := LREAL#1.0;
Camin4_Ss := LREAL#1.0;
Camin4_Mo := LREAL#180.0;
Camin4_So := LREAL#0.0;
Camin4_Rt := _eMC_REFERENCE_TYPE#_mcCommand;
Camin4_Dir := _eMC_DIRECTION#_mcNoDirection;

// 设定输入参数后, 将 InitFlag 设为 TRUE
InitFlag := TRUE;

END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 1 设为伺服 ON 状态
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDslavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
  Pwr1_En:=TRUE;
ELSE
  Pwr1_En:=FALSE;
END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 2 设为伺服 ON 状态
//EtherCAT 通信未正常通信时, 设为伺服 OFF。

```

```

IF (StartPg=TRUE)
  AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
  Pwr2_En:=TRUE;
ELSE
  Pwr2_En:=FALSE;
END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 3 设为伺服 ON 状态
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbl[MC_Axis002.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis002.Cfg.NodeAddress]=FALSE) THEN
  Pwr3_En:=TRUE;
ELSE
  Pwr3_En:=FALSE;
END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 4 设为伺服 ON 状态
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbl[MC_Axis003.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis003.Cfg.NodeAddress]=FALSE) THEN
  Pwr4_En:=TRUE;
ELSE
  Pwr4_En:=FALSE;
END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 5 设为伺服 ON 状态
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbl[MC_Axis004.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis004.Cfg.NodeAddress]=FALSE) THEN
  Pwr5_En:=TRUE;
ELSE
  Pwr5_En:=FALSE;
END_IF;

// 轴 1 ~轴 5 发生轻度故障时, 执行异常处理 FaultHandler。
// 发生异常时的处理 (FaultHandler) 请根据装置由客户进行编程。
IF (MC_Axis000.MFaultLvl.Active=TRUE)
OR (MC_Axis001.MFaultLvl.Active=TRUE)
OR (MC_Axis002.MFaultLvl.Active=TRUE)
OR (MC_Axis003.MFaultLvl.Active=TRUE)
OR (MC_Axis004.MFaultLvl.Active=TRUE) THEN
  FaultHandler();
END_IF;

// 轴 1 为伺服 ON 状态且原点未确定时, 执行轴 1 的原点复位
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
  Hm1_Ex:=TRUE;
END_IF;

// 轴 2 为伺服 ON 状态且原点未确定时, 执行轴 2 的原点复位
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
  Hm2_Ex:=TRUE;

```

```

END_IF;

// 轴 3 为伺服 ON 状态且原点未确定时, 执行轴 3 的原点复位
IF (Pwr3_Status=TRUE) AND (MC_Axis002.Details.Homed=FALSE) THEN
    Hm3_Ex:=TRUE;
END_IF;

// 轴 4 为伺服 ON 状态且原点未确定时, 执行轴 4 的原点复位
IF (Pwr4_Status=TRUE) AND (MC_Axis003.Details.Homed=FALSE) THEN
    Hm4_Ex:=TRUE;
END_IF;

// 轴 5 为伺服 ON 状态且原点未确定时, 执行轴 5 的原点复位
IF (Pwr5_Status=TRUE) AND (MC_Axis004.Details.Homed=FALSE) THEN
    Hm5_Ex:=TRUE;
END_IF;

// 在轴 1 为原点复位完成后, 执行 MC_MoveVelocity
IF Hm1_D=TRUE THEN
    Vel_Ex:=TRUE;
END_IF;

// 轴 2 为原点确定, 且 MC_MoveVelocity 的 Vel_InVel 为 TRUE 时
// 从轴启动轴 2 中设定的 MC_CamIn
IF (Vel_InVel=TRUE) AND (MC_Axis001.Details.Homed=TRUE) THEN
    Camin1_Ex := TRUE;
END_IF;

// 轴 3 为原点确定, 且 MC_MoveVelocity 的 Vel_InVel 为 TRUE 时
// 从轴启动轴 3 中设定的 MC_CamIn
IF (Vel_InVel=TRUE) AND (MC_Axis002.Details.Homed=TRUE) THEN
    Camin2_Ex := TRUE;
END_IF;

// 轴 4 为原点确定, 且 MC_MoveVelocity 的 Vel_InVel 为 TRUE 时
// 从轴启动轴 4 中设定的 MC_CamIn
IF (Vel_InVel=TRUE) AND (MC_Axis003.Details.Homed=TRUE) THEN
    Camin3_Ex := TRUE;
END_IF;

// 轴 5 为原点确定, 且 MC_MoveVelocity 的 Vel_InVel 为 TRUE 时
// 从轴启动轴 5 中设定的 MC_CamIn
IF (Vel_InVel=TRUE) AND (MC_Axis004.Details.Homed=TRUE) THEN
    Camin4_Ex := TRUE;
END_IF;

// 轴 1 的 MC_Power
PWR1(
    Axis      := MC_Axis000,
    Enable    := Pwr1_En,
    Status    => Pwr1_Status,
    Busy      => Pwr1_Bsy,
    Error     => Pwr1_Err,
    ErrorID   => Pwr1_ErrID
);

// 轴 2 的 MC_Power
PWR2(

```

```

    Axis      := MC_Axis001,
    Enable    := Pwr2_En,
    Status    => Pwr2_Status,
    Busy      => Pwr2_Bsy,
    Error     => Pwr2_Err,
    ErrorID   => Pwr2_ErrID
);

// 轴 3 的 MC_Power
PWR3(
    Axis      := MC_Axis002,
    Enable    := Pwr3_En,
    Status    => Pwr3_Status,
    Busy      => Pwr3_Bsy,
    Error     => Pwr3_Err,
    ErrorID   => Pwr3_ErrID
);

// 轴 4 的 MC_Power
PWR4(
    Axis      := MC_Axis003,
    Enable    := Pwr4_En,
    Status    => Pwr4_Status,
    Busy      => Pwr4_Bsy,
    Error     => Pwr4_Err,
    ErrorID   => Pwr4_ErrID
);

// 轴 5 的 MC_Power
PWR5(
    Axis      := MC_Axis004,
    Enable    := Pwr5_En,
    Status    => Pwr5_Status,
    Busy      => Pwr5_Bsy,
    Error     => Pwr5_Err,
    ErrorID   => Pwr5_ErrID
);

// 轴 1 的 MC_Home
HM1(
    Axis          := MC_Axis000,
    Execute       := Hm1_Ex,
    Done          => Hm1_D,
    Busy          => Hm1_Bsy,
    CommandAborted => Hm1_Ca,
    Error         => Hm1_Err,
    ErrorID       => Hm1_ErrID
);

// 轴 2 的 MC_Home
HM2(
    Axis          := MC_Axis001,
    Execute       := Hm2_Ex,
    Done          => Hm2_D,
    Busy          => Hm2_Bsy,
    CommandAborted => Hm2_Ca,
    Error         => Hm2_Err,
    ErrorID       => Hm2_ErrID
);

```

```

// 轴 3 的 MC_Home
HM3(
  Axis           := MC_Axis002,
  Execute        := Hm3_Ex,
  Done           => Hm3_D,
  Busy           => Hm3_Bsy,
  CommandAborted => Hm3_Ca,
  Error          => Hm3_Err,
  ErrorID        => Hm3_ErrID
);

// 轴 4 的 MC_Home
HM4(
  Axis           := MC_Axis003,
  Execute        := Hm4_Ex,
  Done           => Hm4_D,
  Busy           => Hm4_Bsy,
  CommandAborted => Hm4_Ca,
  Error          => Hm4_Err,
  ErrorID        => Hm4_ErrID
);

// 轴 5 的 MC_Home
HM5(
  Axis           := MC_Axis004,
  Execute        := Hm5_Ex,
  Done           => Hm5_D,
  Busy           => Hm5_Bsy,
  CommandAborted => Hm5_Ca,
  Error          => Hm5_Err,
  ErrorID        => Hm5_ErrID
);

//MC_MoveVelocity
VEL(
  Axis           := MC_Axis000,
  Execute        := Vel_Ex,
  Velocity       := Vel_Vel,
  Acceleration   := Vel_Acc,
  Deceleration   := Vel_Dec,
  Direction      := Vel_Dir,
  InVelocity     => Vel_InVel,
  Busy           => Vel_Bsy,
  Active         => Vel_Act,
  CommandAborted => Vel_Ca,
  Error          => Vel_Err,
  ErrorID        => Vel_ErrID
);

//MC_CamIn
CAMIN1(
  Master         := MC_Axis000,
  Slave          := MC_Axis001,
  CamTable       := CamProfile0,
  Execute        := Camin1_Ex,
  Periodic       := Camin1_Em,
  StartMode      := Camin1_Sm,
  StartPosition  := Camin1_Sp,
  MasterStartDistance := Camin1_Msd,

```



```

MasterScaling      := Camin1_Ms,
SlaveScaling       := Camin1_Ss,
MasterOffset       := Camin1_Mo,
SlaveOffset        := Camin1_So,
ReferenceType      := Camin1_Rt,
Direction          := Camin1_Dir,
InCam              => Camin1_InCam,
InSync             => Camin1_InSync,
EndOfProfile       => Camin1_Eop,
Index              => Camin1_Index,
Busy               => Camin1_Bsy,
Active             => Camin1_Act,
CommandAborted     => Camin1_Ca,
Error              => Camin1_Err,
ErrorID            => Camin1_ErrID
);

```

```

CAMIN2(
  Master           := MC_Axis000,
  Slave           := MC_Axis002,
  CamTable        := CamProfile0,
  Execute         := Camin2_Ex,
  Periodic        := Camin2_Em,
  StartMode       := Camin2_Sm,
  StartPosition   := Camin2_Sp,
  MasterStartDistance := Camin2_Msd,
  MasterScaling   := Camin2_Ms,
  SlaveScaling    := Camin2_Ss,
  MasterOffset    := Camin2_Mo,
  SlaveOffset     := Camin2_So,
  ReferenceType   := Camin2_Rt,
  Direction       := Camin2_Dir,
  InCam           => Camin2_InCam,
  InSync          => Camin2_InSync,
  EndOfProfile    => Camin2_Eop,
  Index           => Camin2_Index,
  Busy            => Camin2_Bsy,
  Active          => Camin2_Act,
  CommandAborted  => Camin2_Ca,
  Error           => Camin2_Err,
  ErrorID         => Camin2_ErrID
);

```

```

CAMIN3(
  Master           := MC_Axis000,
  Slave           := MC_Axis003,
  CamTable        := CamProfile0,
  Execute         := Camin3_Ex,
  Periodic        := Camin3_Em,
  StartMode       := Camin3_Sm,
  StartPosition   := Camin3_Sp,
  MasterStartDistance := Camin3_Msd,
  MasterScaling   := Camin3_Ms,
  SlaveScaling    := Camin3_Ss,
  MasterOffset    := Camin3_Mo,
  SlaveOffset     := Camin3_So,
  ReferenceType   := Camin3_Rt,
  Direction       := Camin3_Dir,
  InCam           => Camin3_InCam,

```

```

InSync           => Camin3_InSync,
EndOfProfile     => Camin3_Eop,
Index            => Camin3_Index,
Busy             => Camin3_Bsy,
Active           => Camin3_Act,
CommandAborted  => Camin3_Ca,
Error            => Camin3_Err,
ErrorID         => Camin3_ErrID
);

```

```

CAMIN4(
  Master           := MC_Axis000,
  Slave           := MC_Axis004,
  CamTable        := CamProfile0,
  Execute         := Camin4_Ex,
  Periodic        := Camin4_Em,
  StartMode       := Camin4_Sm,
  StartPosition   := Camin4_Sp,
  MasterStartDistance := Camin4_Msd,
  MasterScaling   := Camin4_Ms,
  SlaveScaling    := Camin4_Ss,
  MasterOffset    := Camin4_Mo,
  SlaveOffset     := Camin4_So,
  ReferenceType   := Camin4_Rt,
  Direction       := Camin4_Dir,
  InCam           => Camin4_InCam,
  InSync          => Camin4_InSync,
  EndOfProfile     => Camin4_Eop,
  Index           => Camin4_Index,
  Busy            => Camin4_Bsy,
  Active          => Camin4_Act,
  CommandAborted  => Camin4_Ca,
  Error           => Camin4_Err,
  ErrorID         => Camin4_ErrID
);

```

MC_CamOut

完成输入参数中指定的轴的凸轮动作。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_CamOut	凸轮动作解除	FB		<pre>MC_CamOut_instance (Slave := 《参数》, Execute := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, OutMode := 《参数》, Done => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Deceleration	减速度	LREAL	正值 或“0”	0	指定减速度。 单位为 [指令单位 /s ²]。*1
Jerk (Reserved)	跃度	LREAL	0	0	(Reserved)
OutMode (Reserved)	同步解除 模式选择	_eMC_OUT_ MODE	0: _mcStop	0*2	(Reserved)

*1. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. 如果是有效范围为枚举型的变量初始值，实际不是数值，而是枚举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、 FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、 FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* □ □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	达到速度“0”时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动（中断），本指令中止时 因发生异常，本指令中止时 正在发生异常时，启动了本指令 正在执行 MC_Stop 指令时，启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Slave	从轴	_sAXIS_REF	—	指定从轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

功能说明

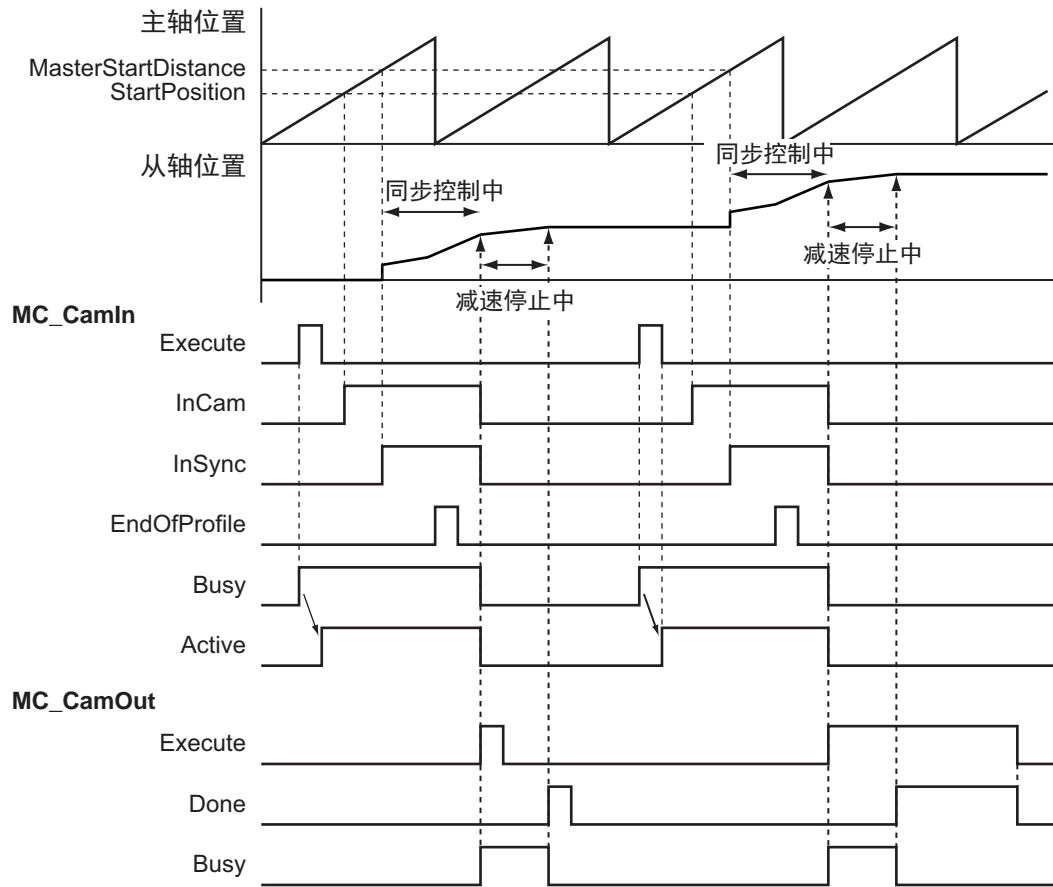
- 本指令可解除从轴的凸轮动作。
- 若将 Execute（启动）设为 TRUE，将按照 Deceleration（减速度）向速度“0”减速。
- 指令速度变为“0”，则表示完成。
- 对未执行凸轮动作的轴启动了本指令时，将发生异常。



使用注意事项

- 凸轮数据变量为全局变量，可从多个任务参照或变更凸轮数据变量的值。从多个任务变更凸轮数据变量的值时，编写的程序应确保多个任务的重写处理不会发生冲突。
- 使用“全局变量的任务间排他控制”功能对凸轮数据变量进行排他控制时，请勿用排他对象的凸轮数据变量启动运动指令。将发生轻度故障等级的异常“凸轮表指定不正确（事件代码：5439 Hex）”。

时序图

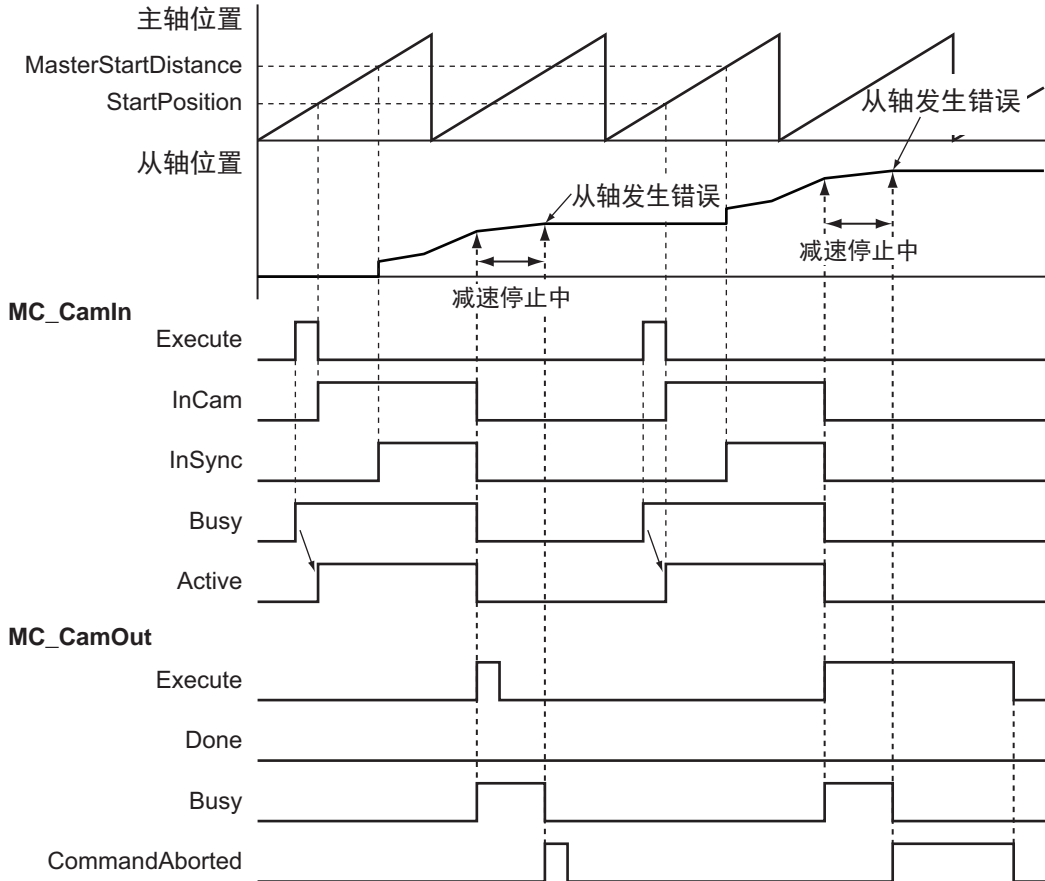


指令的中止

执行本指令的过程中发生异常时，CommandAborted（执行中断）将变为 TRUE，Busy（动作中）变为 FALSE。

此外，轴将按照本指令的 Deceleration（减速度）进行减速。

轴异常请参考 □「A-1 错误代码一览 (P.A-2)」。



运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

● 其他指令执行过程中本指令的启动

执行 MC_CamIn（凸轮动作开始）指令的过程中启动了本指令时，MC_CamIn（凸轮动作开始）指令变为 CommandAborted（执行中断），本指令变为 Busy（执行中）。

不是在执行 MC_CamIn（凸轮动作开始）指令的过程中启动本指令时，将发生 Error（错误）。

● 本指令执行过程中其他指令的启动

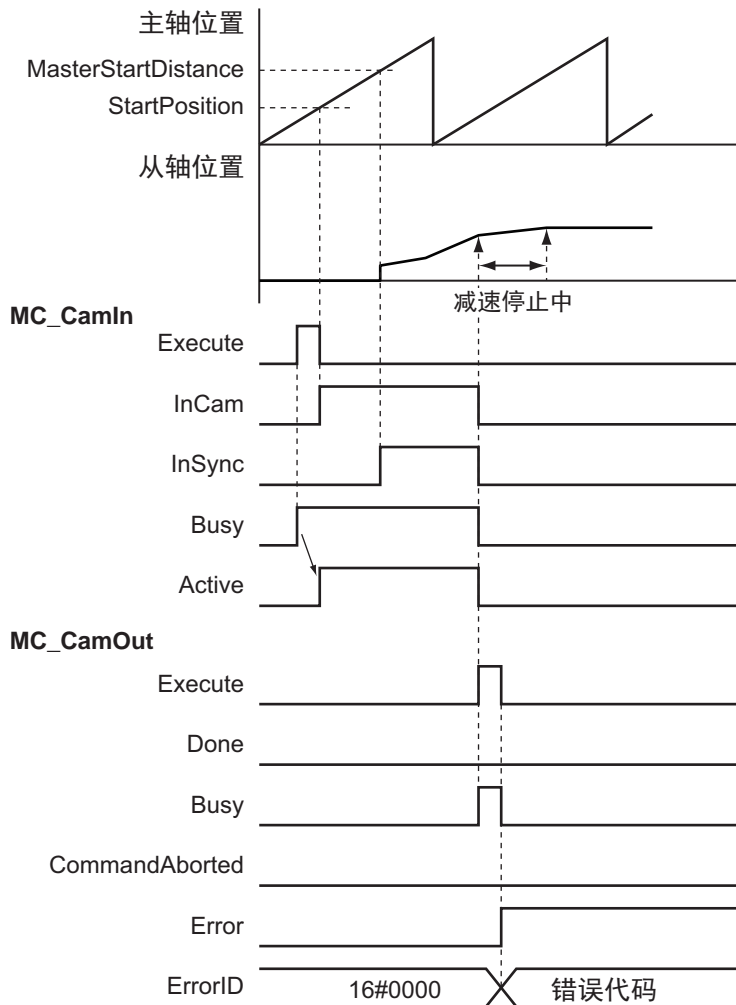
对本指令执行运动指令多重启动时，请指定从轴。

对本指令执行运动指令多重启动时，可进行中断、缓冲。

异常

如果在启动本指令时发生异常，Error（错误）将变为 TRUE。
参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 [□「A-1 错误代码一览 \(P.A-2\)」](#)。

MC_GearIn

设定主轴与从轴间的齿轮比，进行齿轮动作。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_GearIn	齿轮动作开始	FB		<pre>MC_GearIn_instance (Master := 《参数》, Slave := 《参数》, Execute := 《参数》, RatioNumerator := 《参数》, RatioDenominator := 《参数》, ReferenceType := 《参数》, Acceleration := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, BufferMode := 《参数》, InGear => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Ratio Numerator	齿轮比分子	DINT*1	正数或负数 *1	10000	指定主轴和从轴间电子齿轮的分子。
Ratio Denominator	齿轮比分母	UDINT*2	正数	10000	指定主轴和从轴间电子齿轮的分母。
Reference Type*3	位置种类选择	_eMC_REFERENCETYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 *4	指定位置的类别。 0: 指令位置（前一任务周期*5中的计算值） 1: 反馈位置（同一任务周期*5中的获取值） 2: 指令位置（同一任务周期*5中的计算值）
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度。 单位为 [指令单位 /s ²] *6
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。 单位为 [指令单位 /s ²] *3
Jerk (Reserved)	跃度	LREAL	0	0	(Reserved)
BufferMode	缓冲模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0 *2	指定运动指令多重启动时的动作。 0: 中断 1: 缓冲

- *1. 可在 CPU 单元版本 1.02 以上和 Sysmac Studio Ver.1.03 以上的组合中使用。在更低版本的组合中，数据类型为“UINT”，有效范围为“正数”。
- *2. 可在 CPU 单元版本 1.02 以上和 Sysmac Studio Ver.1.03 以上的组合中使用。在更低版本的组合中，数据类型为“UINT”。
- *3. 使用 _mcLatestCommand 时，使用的主轴和从轴必须满足以下关系。
Master（主轴）中设定的运动控制系统变量的轴编号 < Slave（从轴）中设定的运动控制系统变量的轴编号
- *4. 如果是有效范围为列举型的变量初始值，实际不是数值，而是列举值。
- *5. 任务为主固定周期任务时，任务周期为主周期，为固定周期任务（执行优先度 5）时，任务周期为固定周期任务（执行优先度 5）的任务周期。
- *6. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》的“单位转换设定”。

输出变量

输出变量	名称	数据类型	有效范围	内容
InGear	达到齿轮比	BOOL	TRUE、FALSE	从轴达到目标速度时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* □ □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
InGear	以下关系成立时 <ul style="list-style-type: none"> 加速中：从轴的速度 \geq 主轴的速度 \times 齿轮比 减速中：从轴的速度 \leq 主轴的速度 \times 齿轮比 	<ul style="list-style-type: none"> Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	开始了指令时	<ul style="list-style-type: none"> Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动（中断），本指令中止时 因发生异常，本指令中止时 正在发生异常时，启动了本指令 正在执行 MC_Stop 指令时，启动了本指令 启动了 MC_GearOut 指令时 	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Master	主轴	_sAXIS_REF	—	指定主轴。 ^{*1}
Slave	从轴	_sAXIS_REF	—	指定从轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis***”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

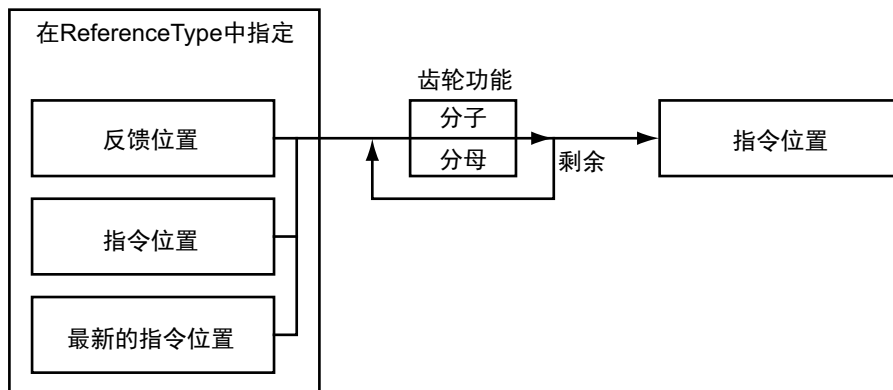


使用注意事项

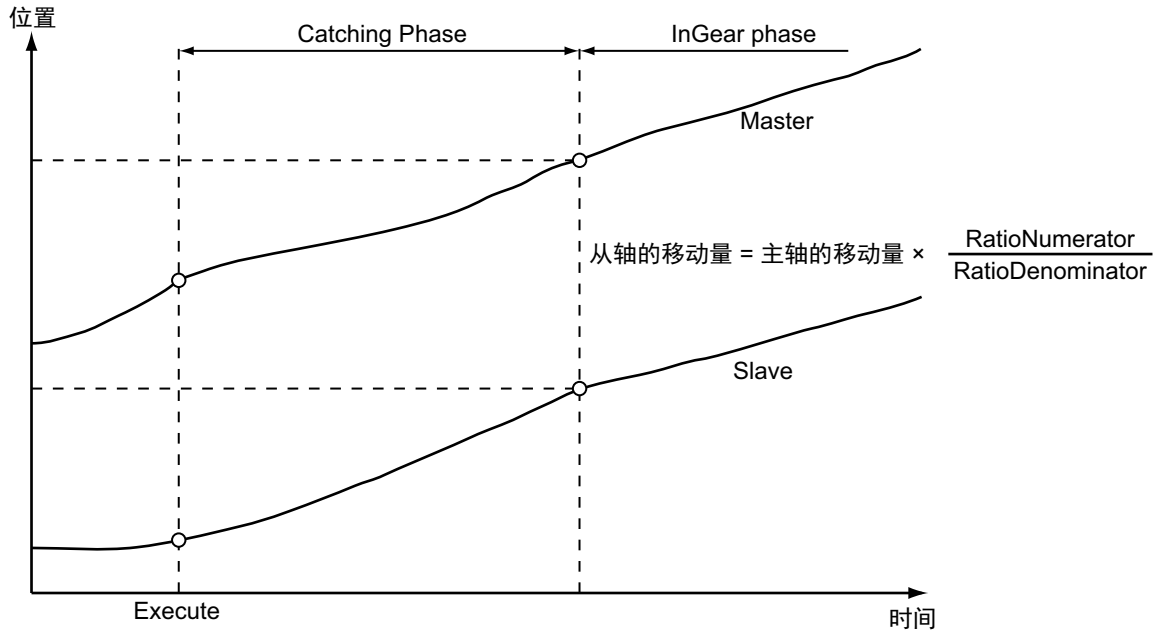
若主轴和从轴指定为相同的轴，将发生轻度故障等级的“主轴从轴相同（错误代码：5436 Hex）”异常。

功能说明

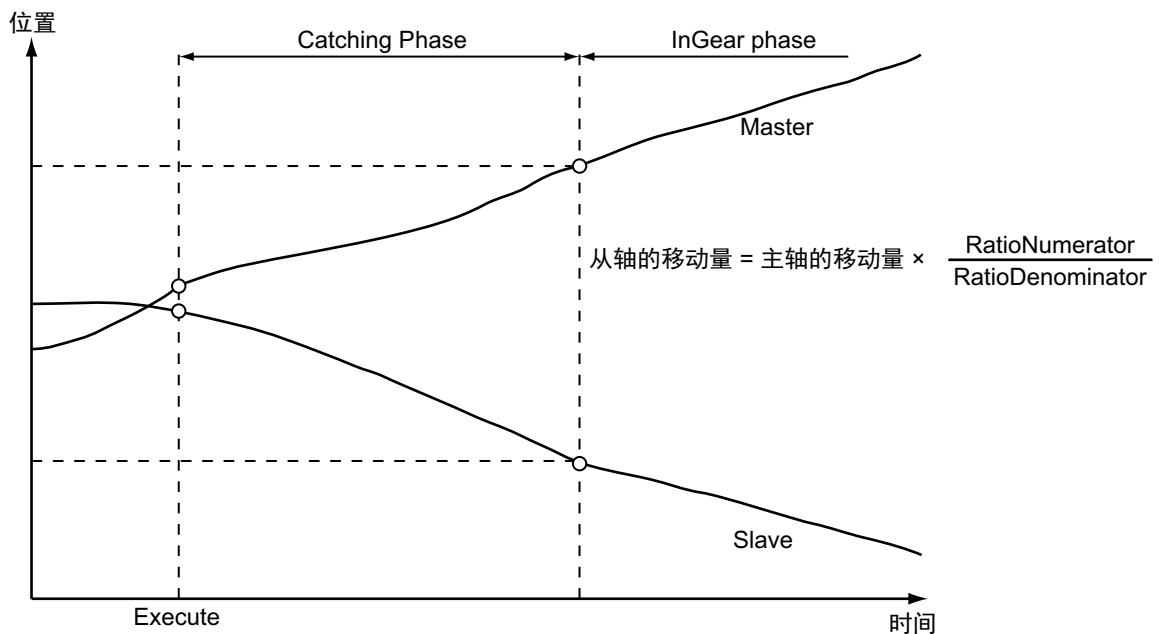
- 在 Slave（从轴）中指定动作对象的轴，指定 RatioNumerator（齿轮比分子）、RatioDenominator（齿轮比分母）、ReferenceType（位置种类选择）、Acceleration（加速度）、Deceleration（减速度），执行齿轮动作。
- Master（主轴）中可指定指令位置、反馈位置、最新指令位置。



- 开始动作后，Slave（从轴）将以 Master（主轴）速度乘以齿轮比得到的速度作为目标速度，执行加速减速动作。
- 达到目标速度前视为 Catching phase（追赶中），达到后称为 InGear phase（齿轮同步中）。
- 齿轮比为正数时，Slave（从轴）与 Master（主轴）的移动方向相同。



- 齿轮比为负数时，Slave（从轴）与 Master（主轴）的移动方向相反。



- 在 Execute（启动）的上升沿开始电子齿轮的动作。

**使用注意事项**

在单元版本 1.09 以下的 CPU 单元中使用本指令时，请勿对 Master（主轴）执行 MC_SetPosition（当前位置变更）。若对 Master（主轴）执行 MC_SetPosition（当前位置变更）指令，Slave（从轴）可能发生剧烈追踪的危险。

不对 Master（主轴）使用 MC_SetPosition（当前位置变更）指令时，请先解除 Master（主轴）和 Slave（从轴）的关系后再执行。

关于主轴的注意事项，请参考 □「同步控制的主轴及辅助轴的注意事项 (P.1-6)」。

● ReferenceType（位置种类选择）

从以下项目中选择位置种类。

- `_mcCommand`: 指令位置（前一任务周期中的计算值）
在当前周期，使用前一任务周期中计算的主轴指令位置。
使用计算从轴指令位置之前的固定周期任务中计算得到的主轴指令位置。
- `_mcFeedback`: 同一任务周期的获取值
使用同一任务周期内获取的主轴的反馈位置。
- `_mcLatestCommand`: 指令位置（同一任务周期中的计算值）
使用同一任务周期中计算得到的主轴指令位置。
可使用比 `_mcCommand` 更新的信息。但是，主轴编号必须设定为比从轴编号更新的编号。
如果从轴的轴编号比主轴新，本指令的 Error（错误）将变为 TRUE。ErrorID（错误代码）中将输出“主轴 / 从轴 轴编号非升序（错误代码：5438 Hex）”。

**使用注意事项**

任务为主固定周期任务时，这里的任务周期为主周期，为固定周期任务（执行优先度 5）时，任务周期为固定周期任务（执行优先度 5）的任务周期。同样，固定周期任务表示主固定周期任务或固定周期任务（执行优先度 5）。

**参考**

选择同一任务周期内计算得到的指令位置时，与前一任务周期中的计算值相比，同步精度更高。

但是，使用主轴和从轴时，必须保证“Master（主轴）中设定的运动控制系统变量的轴编号 < Slave（从轴）中设定的运动控制系统变量的轴编号”。

● 轴种类和位置种类的关系

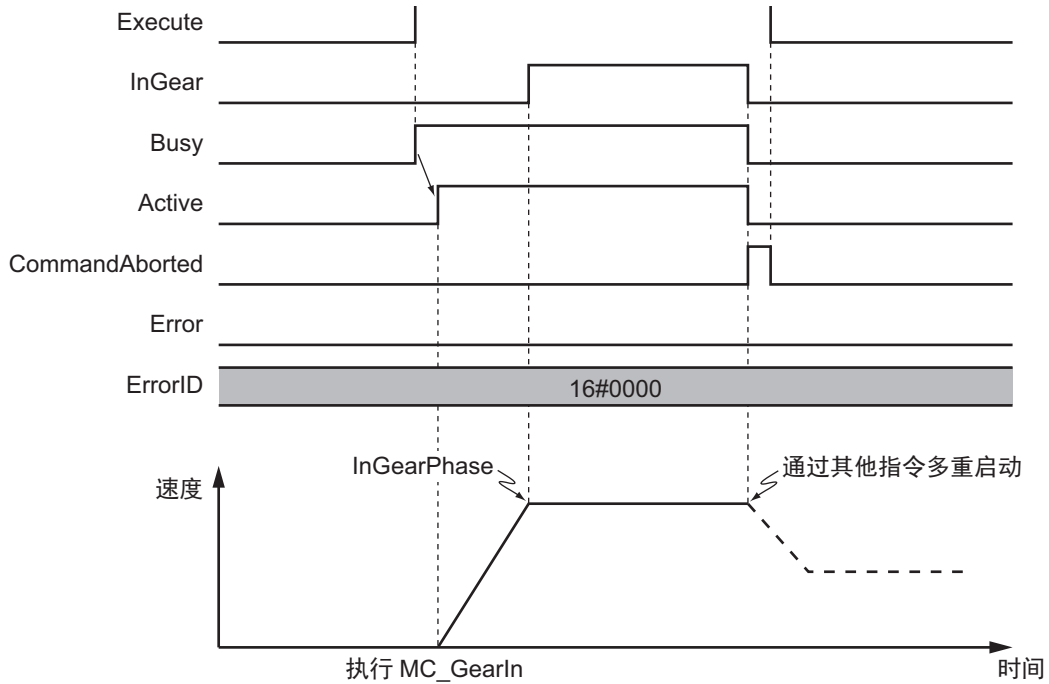
可监视的轴种类和要监视的位置种类的关系如下所示。

轴种类	ReferenceType	
	<code>_mcCommand</code> 或 <code>_mcLatestCommand</code>	<code>_mcFeedback</code>
伺服轴	○	○
编码器轴	x ^{*1}	○
虚拟伺服轴	○	○
虚拟编码器轴	x ^{*1}	○

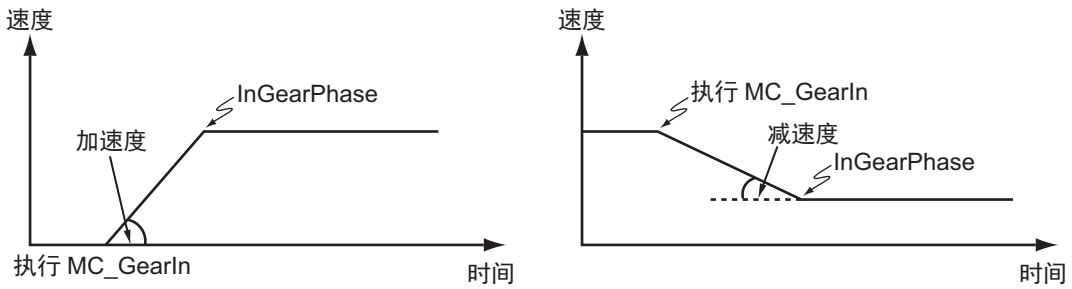
*1. 指令启动时，将发生“超出位置种类选择范围（错误代码：5430 Hex）”。

时序图

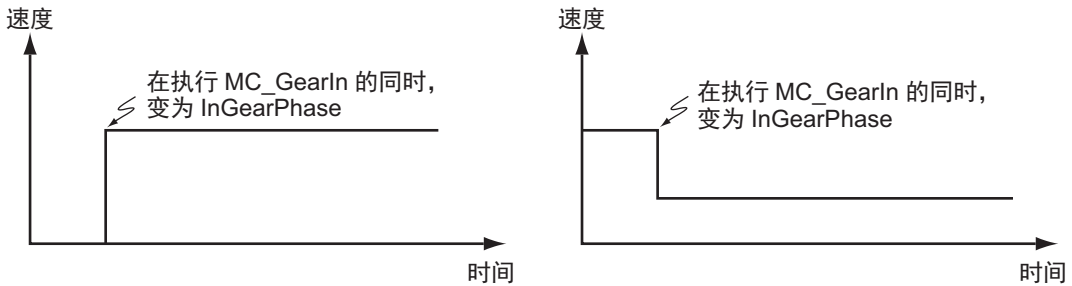
- 在 Execute (启动) 上升的同时 Busy (执行中) 变为 TRUE, 在下一周期, Active (控制中) 变为 TRUE。
- 达到目标速度后, InGear (达到齿轮比) 变为 TRUE。
- 通过其他指令中断了本指令时, CommandAborted (执行中断) 变为 TRUE, Busy (执行中)、Active (控制中)、InGear (达到齿轮比) 变为 FALSE。
- 要中途结束电子齿轮动作时, 使用 MC_GearOut (齿轮动作解除) 指令或 MC_Stop (强制停止) 指令。



输入变量中可指定 Acceleration（加速度）、Deceleration（减速度）。



将 Acceleration（加速度）或 Deceleration（减速度）指定为“0”并启动时，将不进行加减速而达到目标速度。

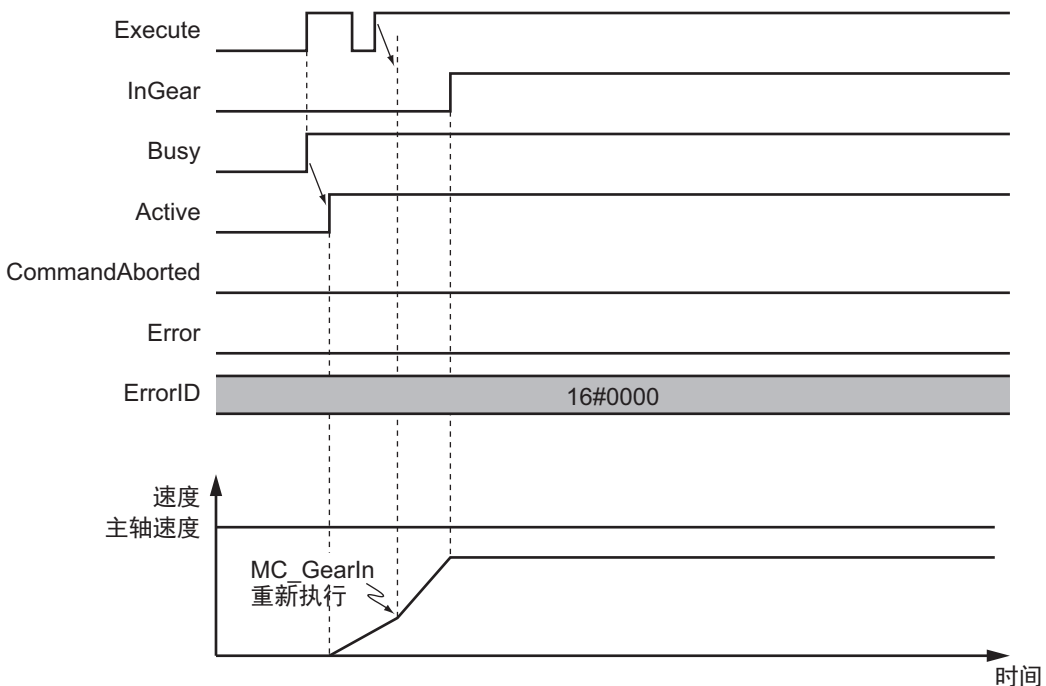


运动指令重新执行

在定位动作中变更输入变量，并再次将 Execute（启动）设为 TRUE，可变更本指令的动作。通过运动指令重新执行可变更的输入变量包括 RatioNumerator（齿轮比分子）、RatioDenominator（齿轮比分母）、Acceleration（加速度）、Deceleration（减速度）。

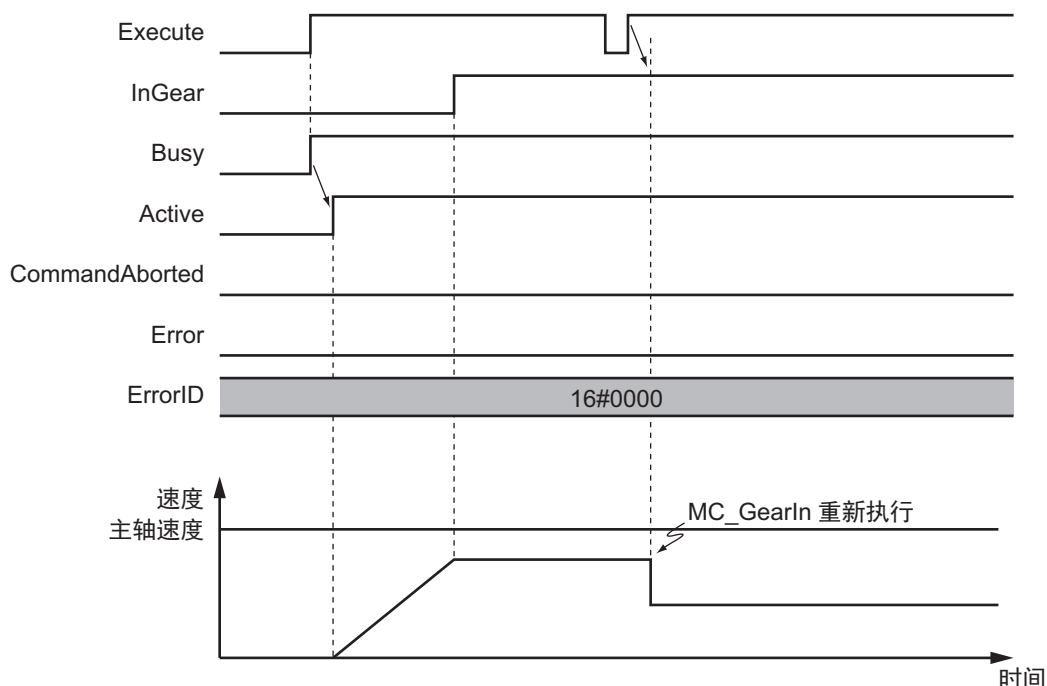
运动指令重新执行的详情请参考 □《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

下面为在 Catching phase（追赶中）中重新执行，并变更了 Acceleration（加速度）时的时序图，以供参考。



在 InGear phase（齿轮同步中）中重新执行，并变更了 RatioNumerator（齿轮比分子）、RatioDenominator（齿轮比分母）时的时序图如下所示。

动作与 Acceleration（加速度）和 Deceleration（减速度）均设定为“0”时相同。



运动指令多重启动

运动指令多重启动的详情请参考 [□](#) 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 其他指令执行过程中本指令的启动

对当前正在执行的指令启动本指令后，可切换为本指令或缓冲本指令。

每个轴最多可缓冲 1 个。

在 BufferMode（缓冲模式选择）中指定指令多重启动时本指令的动作。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 切换指令后如果轴的动作方向发生反转，将按照轴参数 [反转时动作] 执行反转动作。
缓冲	当前正在执行的指令正常完成后，自动启动已缓冲的本指令。

BufferMode（缓冲模式选择）请参考 [□](#) 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 本指令执行过程中其他指令的启动

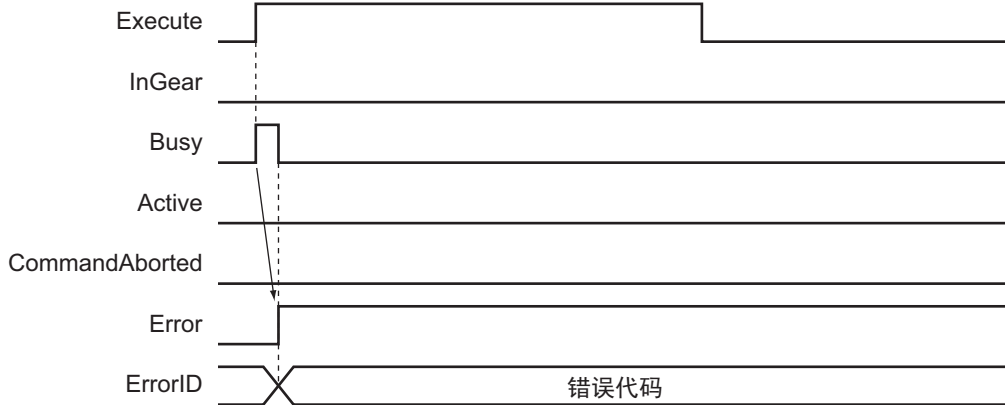
针对从轴，可通过中断进行指令多重启动。

此时，将中止齿轮动作，开始多重启动时的动作。

不可通过中断以外的方式进行指令多重启动。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。
参照输出到 ErrorID（错误代码）的值，可找出异常的原因。



参考

- 同步控制过程中，不会受主轴的异常状态影响。解除主轴的异常状态，主轴开始正常动作后，从轴将继续电子齿轮动作。
- 本指令启动时以及本指令执行过程中，如果从轴发生异常，不会对主轴造成影响。

● 错误代码

指令中发生的异常请参考 □□「A-1 错误代码一览 (P.A-2)」。

示例程序

下面介绍与齿轮比成比例动作时的示例程序。



参考

保留 (Reserved) 处理的输入变量不可指定为初始值以外的值。
本示例中不指定参数。

参数设定

下面表示本示例程序中使用的最低限度的设定。

● 设定轴参数

轴种类

轴	轴种类
轴 1	伺服轴 (主轴)
轴 2	伺服轴 (从轴)
轴 3	伺服轴 (从轴)

计数器模式

轴	计数器模式
轴 1	旋转模式
轴 2	旋转模式
轴 3	旋转模式

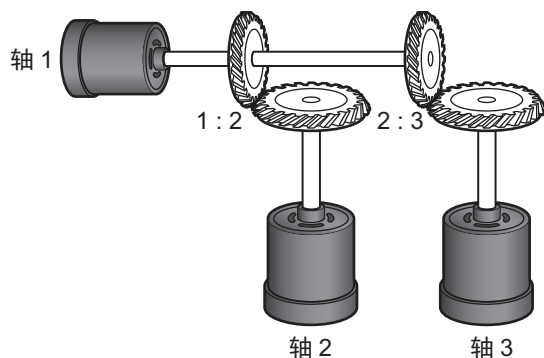
循环计数器

轴	上限值	下限值
轴 1	360	0
轴 2	360	0
轴 3	360	0

坐标单位选择

轴	坐标单位
轴 1	mm
轴 2	mm
轴 3	mm

动作示例



1 主轴的开始

主轴（轴 1）为实际伺服轴，根据速度控制动作。

2 从轴的启动

主轴的反馈速度达到目标速度后，针对主轴的反馈位置，对从轴（轴 2）以齿轮比 1:2、对从轴（轴 3）以齿轮比 2:3 执行齿轮动作。

3 从轴的停止

从轴的反馈位置（MC_Axis000.Act.Pos）超出 1000.0 后，解除轴 2 的齿轮动作，按减速度 DecRate 停止轴 2。此时，轴 3 将继续执行齿轮动作。

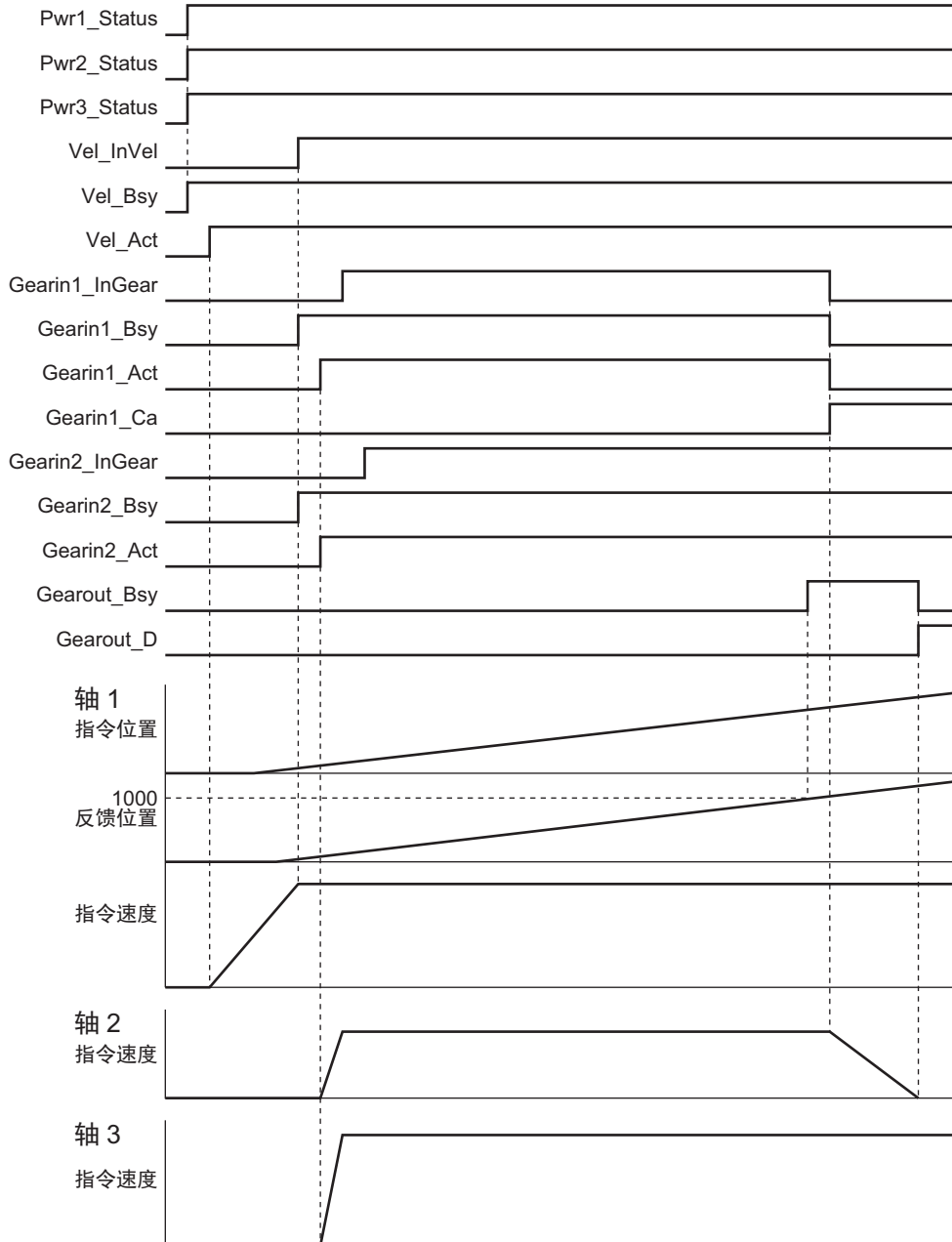
梯形图

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	主轴轴 1 的轴变量。
MC_Axis000.Act.Pos	LREAL	—	表示轴 1 的反馈当前位置。
MC_Axis001	_sAXIS_REF	—	从轴轴 2 的轴变量。
MC_Axis002	_sAXIS_REF	—	从轴轴 3 的轴变量。
Pwr1_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR1 的输出变量 Status 中的变量。若变为伺服 ON 状态，该变量变为 TRUE。
Pwr2_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR2 的输出变量 Status 中的变量。若变为伺服 ON 状态，该变量变为 TRUE。
Pwr3_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR3 的输出变量 Status 中的变量。若变为伺服 ON 状态，该变量变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态，则变为伺服 ON 状态。
Vel_InVel	BOOL	FALSE	分配到 MC_MoveVelocity 的实例 VEL 的输出 InVelocity 中的变量。达到目标速度时，该变量变为 TRUE。
Gearin1_Act	BOOL	FALSE	分配到 MC_GearIn 的实例 GEARIN1 的输出 Active 中的变量。若 GEARIN1 变为控制中，该变量变为 TRUE。

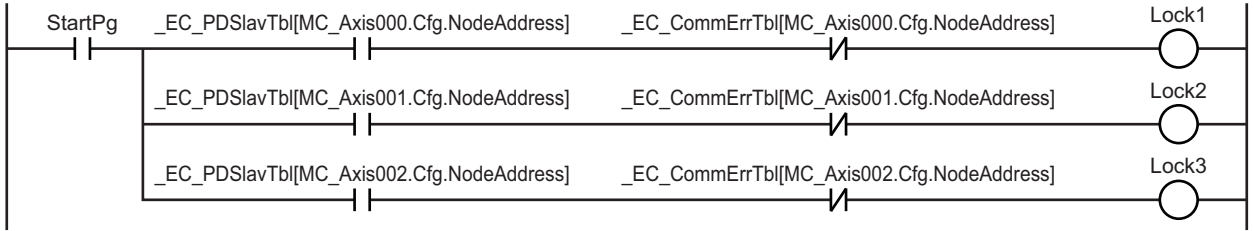
名称	数据类型	初始值	注释
Gearout_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_GearOut 的实例 GEAROUT。

● 时序图

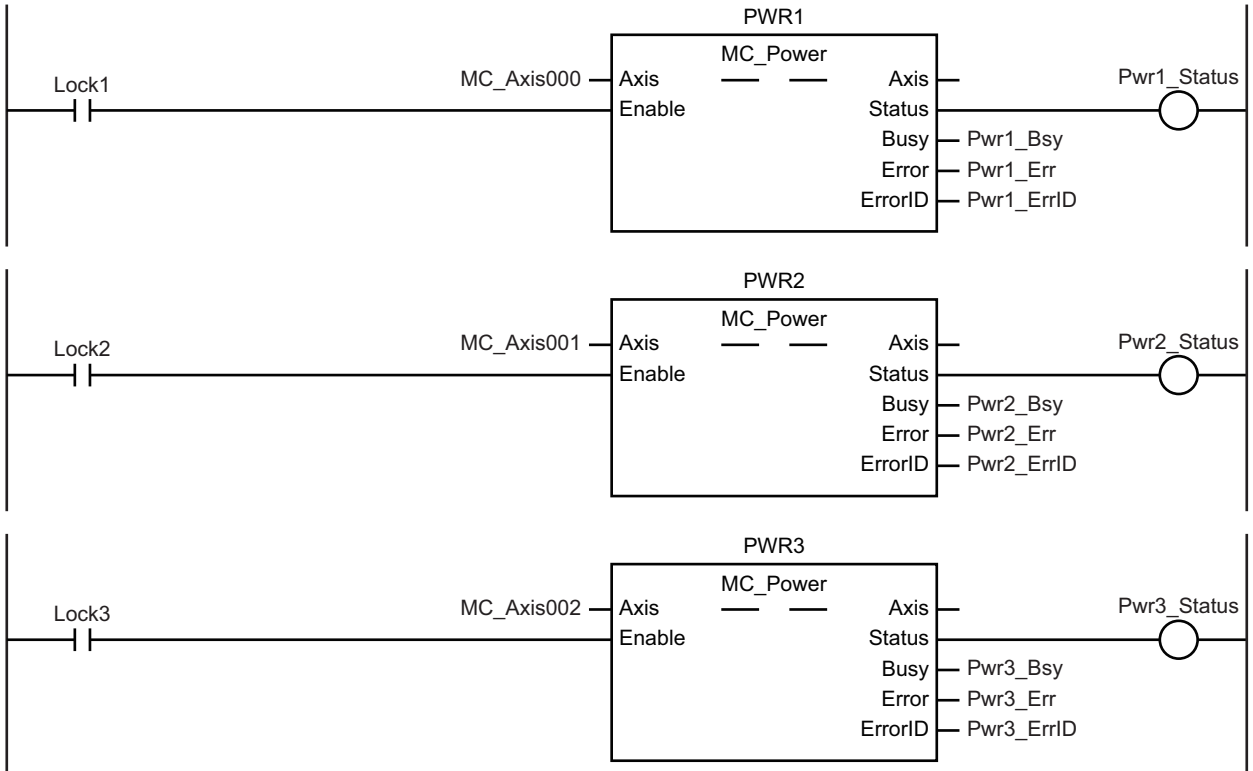


● 示例程序

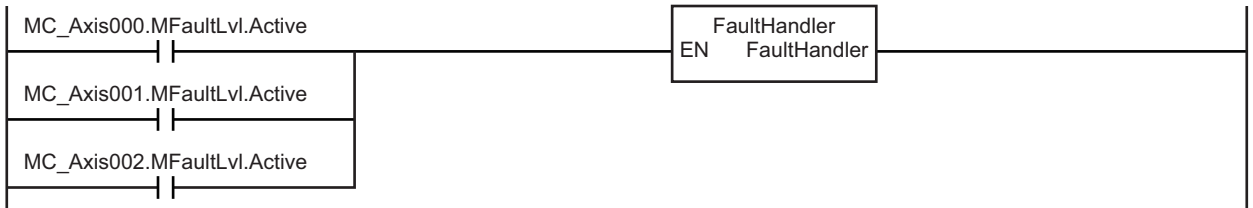
接点 StartPg 变为 TRUE 后，确认各轴的 EtherCAT 通信是否正常进行过程数据通信



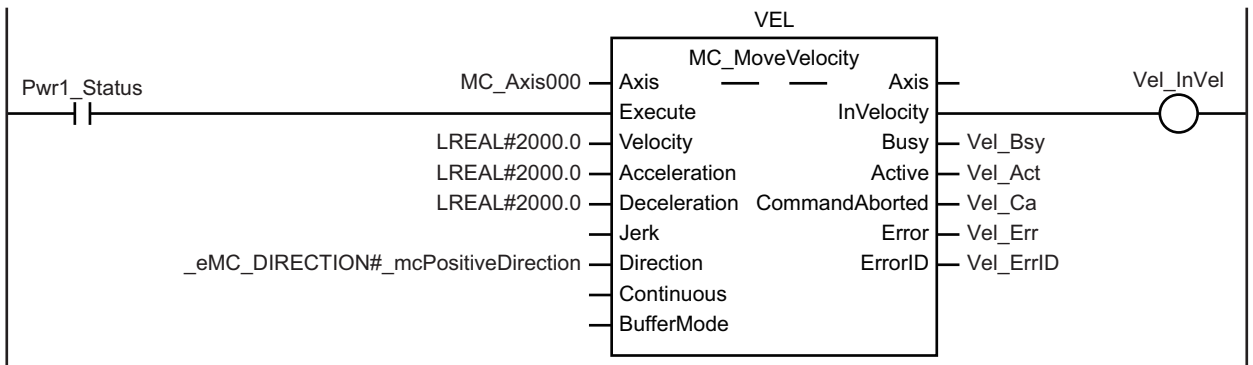
若正在进行过程数据通信，则将各轴设为伺服 ON 状态



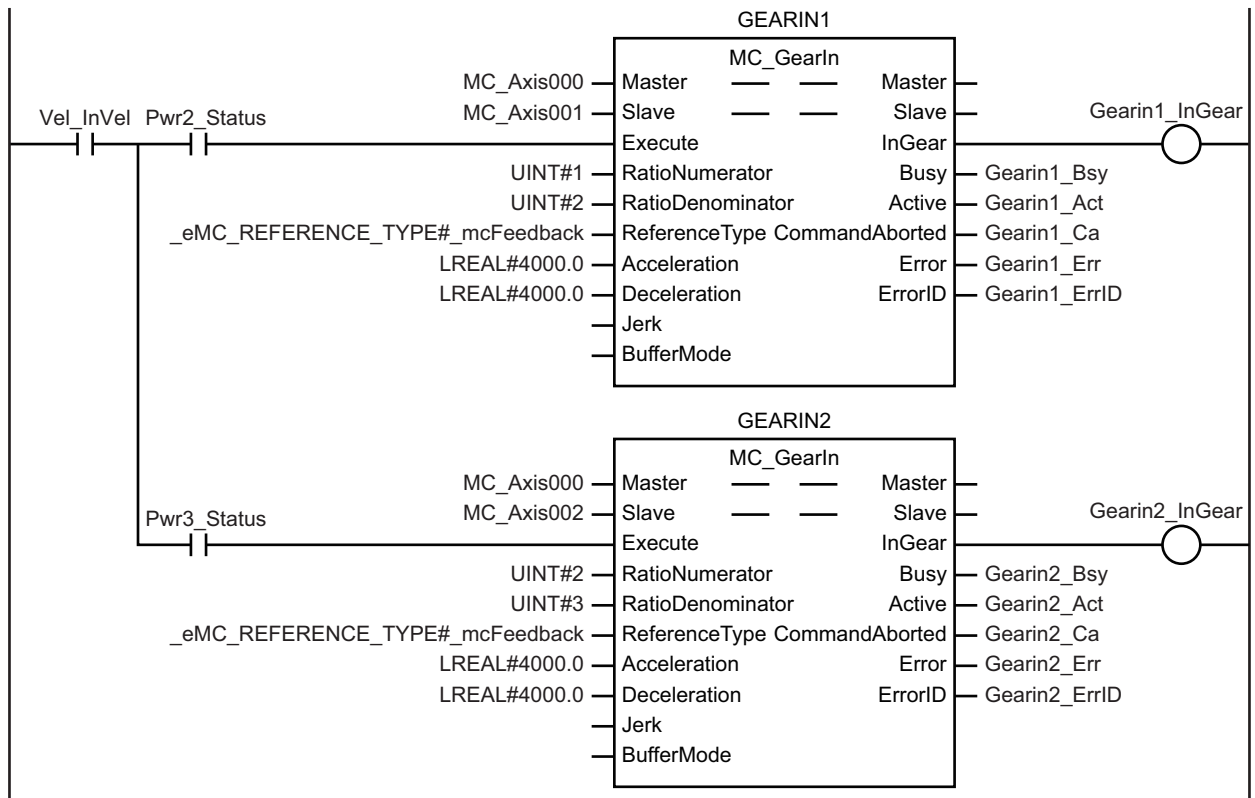
轴构成发生轻度故障等级的异常时，执行装置发生异常时的处理（FaultHandler）。发生异常时的处理（FaultHandler）请根据装置由客户进行编程



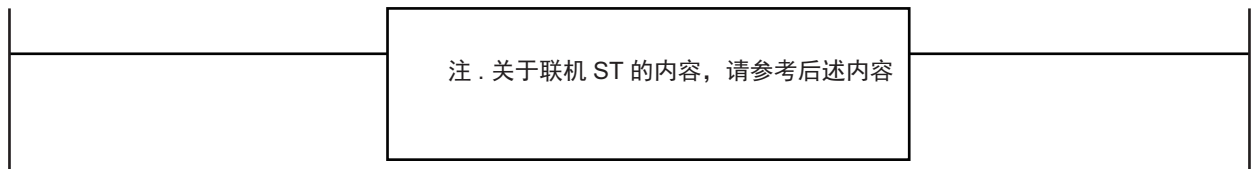
主轴（轴 1）变为可运行状态后，启动 MC_MoveVelocity（速度控制）指令



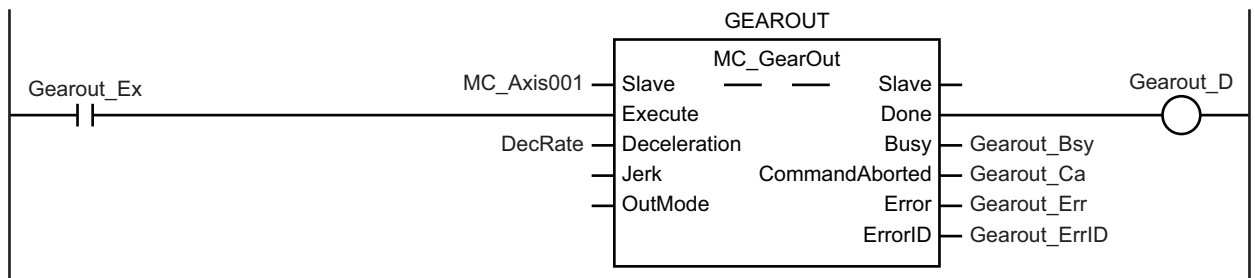
主轴（轴 1）的反馈速度达到目标速度时，对从轴（轴 2）以齿轮比 1:2、对从轴（轴 3）以齿轮比 2:3 开始齿轮动作



从轴（轴 2）为齿轮动作中，且主轴的反馈位置为 1000.0 以上时，将 Gearout_Ex 设为 TRUE



Gearout_Ex 变为 TRUE 后，解除从轴（轴 2）的齿轮动作，减速停止



联机 ST 的内容

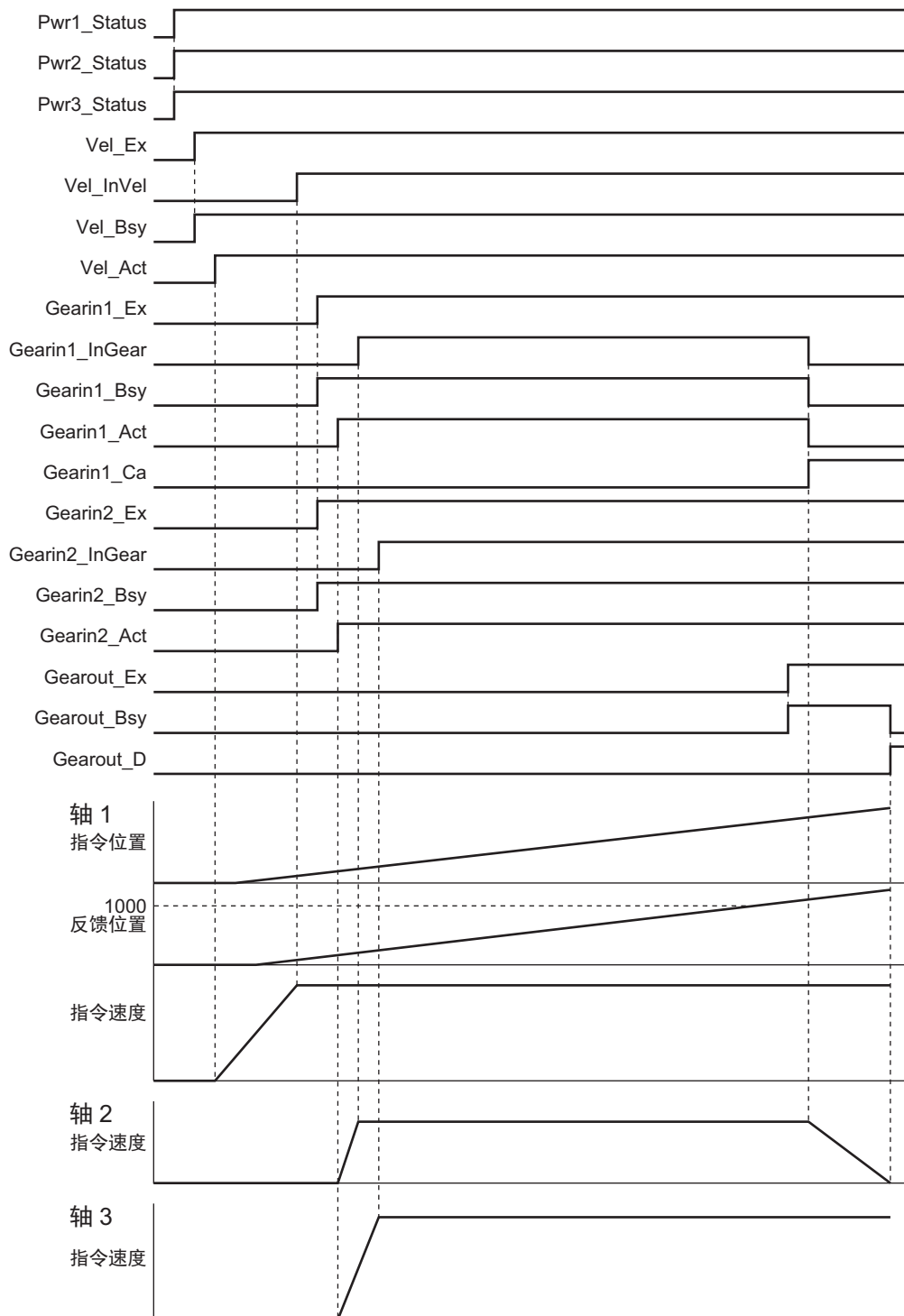
```
IF (Gearin1_Act=TRUE) AND (MC_Axis000.Act.Pos>=LREAL#1000.0) THEN
  Gearout_Ex := TRUE;
END_IF;
```

结构文本 (ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	主轴轴 1 的轴变量。
MC_Axis000.Act.Pos	LREAL	—	表示轴 1 的反馈当前位置。
MC_Axis001	_sAXIS_REF	—	从轴轴 2 的轴变量。
MC_Axis002	_sAXIS_REF	—	从轴轴 3 的轴变量。
Pwr1_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR1 的输出变量 Status 中的变量。若变为伺服 ON 状态, 该变量变为 TRUE。
Pwr2_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR2 的输出变量 Status 中的变量。若变为伺服 ON 状态, 该变量变为 TRUE。
Pwr3_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR3 的输出变量 Status 中的变量。若变为伺服 ON 状态, 该变量变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态, 则变为伺服 ON 状态。
Vel_InVel	BOOL	FALSE	分配到 MC_MoveVelocity 的实例 VEL 的输出 InVelocity 中的变量。达到目标速度时, 该变量变为 TRUE。
Gearin1_Act	BOOL	FALSE	分配到 MC_GearIn 的实例 GEARIN1 的输出 Active 中的变量。若 GEARIN1 变为控制中, 该变量变为 TRUE。
Gearout_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_GearOut 的实例 GEAROUT。
Vel_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_MoveVelocity 的实例 VEL。
Gearin1_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_GearIn 的实例 GEARIN1。
Gearin2_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_GearIn 的实例 GEARIN2。
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE 时, 设定输入参数。输入参数设定完成后, 变为 TRUE。

● 时序图



● 示例程序

```

// 未设定输入参数时的处理
IF InitFlag = FALSE THEN
  //MC_MoveVelocity 参数
  Vel_Vel := LREAL#2000.0;
  Vel_Acc := LREAL#2000.0;
  Vel_Dec := LREAL#2000.0;
  Vel_Dir := _eMC_DIRECTION#_mcPositiveDirection;

  //MC_GearIn1 参数
  Gearin1_Rn := UINT#1;
  Gearin1_Rd := UINT#2;
  Gearin1_Rt := _eMC_REFERENCE_TYPE#_mcFeedback;
  Gearin1_Acc := LREAL#4000.0;
  Gearin1_Dec := LREAL#4000.0;

  //MC_GearIn2 参数
  Gearin2_Rn := UINT#2;
  Gearin2_Rd := UINT#3;
  Gearin2_Rt := _eMC_REFERENCE_TYPE#_mcFeedback;
  Gearin2_Acc := LREAL#4000.0;
  Gearin2_Dec := LREAL#4000.0;

  //MC_GearOut 参数
  DecRate := LREAL#200.0;
  Gearout_Dec := DecRate;

  // 设定输入参数后，将 InitFlag 设为 TRUE
  InitFlag:=TRUE;

END_IF;

//StartPg 为 TRUE 时，确认 EtherCAT 通信正常通信，
// 轴 1 设为伺服 ON 状态
//EtherCAT 通信未正常通信时，设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
  Pwr1_En:=TRUE;
ELSE
  Pwr1_En:=FALSE;
END_IF;

//StartPg 为 TRUE 时，确认 EtherCAT 通信正常通信，
// 轴 2 设为伺服 ON 状态
//EtherCAT 通信未正常通信时，设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
  Pwr2_En:=TRUE;
ELSE
  Pwr2_En:=FALSE;
END_IF;

//StartPg 为 TRUE 时，确认 EtherCAT 通信正常通信，
// 轴 3 设为伺服 ON 状态
//EtherCAT 通信未正常通信时，设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbl[MC_Axis002.Cfg.NodeAddress]=TRUE)

```



```

AND (_EC_CommErrTbl[MC_Axis002.Cfg.NodeAddress]=FALSE) THEN
  Pwr3_En:=TRUE;
ELSE
  Pwr3_En:=FALSE;
END_IF;

// 轴 1 ~轴 3 发生轻度故障时, 执行异常处理 FaultHandler。
// 发生异常时的处理 (FaultHandler) 请根据装置由客户进行编程。
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) OR
(MC_Axis002.MFaultLvl.Active=TRUE) THEN
  FaultHandler();
END_IF;

// 轴 1 为伺服 ON 状态时, 启动 MC_MoveVelocity。
IF Pwr1_Status=TRUE THEN
  Vel_Ex := TRUE;
END_IF;

// MC_MoveVelocity 的 InVelocity 为 TRUE, 且轴 2 为伺服 ON 状态时, 启动主轴设定为轴 1、从轴
// 设定为轴 2 的 MC_GearIn

IF (Vel_InVel=TRUE) AND (Pwr2_Status=TRUE) THEN
  Gearin1_Ex := TRUE;
END_IF;

// MC_MoveVelocity 的 InVelocity 为 TRUE, 且轴 3 为伺服 ON 状态时, 启动主轴设定为轴 1、从轴
// 设定为轴 3 的 MC_GearIn
IF (Vel_InVel=TRUE) AND (Pwr3_Status=TRUE) THEN
  Gearin2_Ex := TRUE;
END_IF;

// 轴 2 为齿轮动作中, 且轴 1 的反馈位置为 1000.0 以上时,
// 从轴设定为轴 2, 启动 GearOut
IF (Gearin1_Act=TRUE) AND (MC_Axis000.Act.Pos>=LREAL#1000.0) THEN
  Gearout_Ex := TRUE;
END_IF;

// 轴 1 的 MC_Power
PWR1(
  Axis      := MC_Axis000,
  Enable    := Pwr1_En,
  Status    => Pwr1_Status,
  Busy      => Pwr1_Bsy,
  Error     => Pwr1_Err,
  ErrorID   => Pwr1_ErrID
);

// 轴 2 的 MC_Power
PWR2(
  Axis      := MC_Axis001,
  Enable    := Pwr2_En,
  Status    => Pwr2_Status,
  Busy      => Pwr2_Bsy,
  Error     => Pwr2_Err,
  ErrorID   => Pwr2_ErrID
);

```

```

// 轴 3 的 MC_Power
PWR3(
    Axis      := MC_Axis002,
    Enable    := Pwr3_En,
    Status    => Pwr3_Status,
    Busy      => Pwr3_Bsy,
    Error     => Pwr3_Err,
    ErrorID   => Pwr3_ErrID
);

//MC_MoveVelocity
VEL(
    Axis          := MC_Axis000,
    Execute       := Vel_Ex,
    Velocity      := Vel_Vel,
    Acceleration  := Vel_Acc,
    Deceleration  := Vel_Dec,
    Direction     := Vel_Dir,
    InVelocity    => Vel_InVel,
    Busy          => Vel_Bsy,
    Active        => Vel_Act,
    CommandAborted => Vel_Ca,
    Error        => Vel_Err,
    ErrorID      => Vel_ErrID
);

// 以轴 1 为主轴、轴 2 为从轴的 MC_GearIn
GEARIN1(
    Master        := MC_Axis000,
    Slave        := MC_Axis001,
    Execute       := Gearin1_Ex,
    RatioNumerator := Gearin1_Rn,
    RatioDenominator := Gearin1_Rd,
    ReferenceType := Gearin1_Rt,
    Acceleration  := Gearin1_Acc,
    Deceleration  := Gearin1_Dec,
    InGear       => Gearin1_InGear,
    Busy         => Gearin1_Bsy,
    Active       => Gearin1_Act,
    CommandAborted => Gearin1_CA,
    Error        => Gearin1_Err,
    ErrorID      => Gearin1_ErrID
);

// 以轴 1 为主轴、轴 3 为从轴的 MC_GearIn
GEARIN2(
    Master        := MC_Axis000,
    Slave        := MC_Axis002,
    Execute       := Gearin2_Ex,
    RatioNumerator := Gearin2_Rn,
    RatioDenominator := Gearin2_Rd,
    ReferenceType := Gearin2_Rt,
    Acceleration  := Gearin2_Acc,
    Deceleration  := Gearin2_Dec,
    InGear       => Gearin2_InGear,
    Busy         => Gearin2_Bsy,
    Active       => Gearin2_Act,
    CommandAborted => Gearin2_CA,

```

```
Error          => Gearin2_Err,  
ErrorID        => Gearin2_ErrID  
);  
  
//MC_GearOut  
GEAROUT(  
  Slave          := MC_Axis001,  
  Execute        := Gearout_Ex,  
  Deceleration   := Gearout_Dec,  
  Done           => Gearout_D,  
  Busy           => Gearout_Bsy,  
  CommandAborted => Gearout_CA,  
  Error          => Gearout_Err,  
  ErrorID        => Gearout_ErrID  
);
```

MC_GearInPos

设定主轴与从轴间的齿轮比，进行电子齿轮动作。
指定开始同步时主轴及从轴的位置。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_GearInPOS	位置指定 齿轮动作	FB		<pre> MC_GearInPos_instance (Master := 《参数》, Slave := 《参数》, Execute := 《参数》, RatioNumerator := 《参数》, RatioDenominator := 《参数》, ReferenceType := 《参数》, MasterSyncPosition := 《参数》, SlaveSyncPosition := 《参数》, Velocity := 《参数》, Acceleration := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, BufferMode := 《参数》, StartSync => 《参数》, InSync => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》); </pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Ratio Numerator	齿轮比分子	DINT ^{*1}	正数或负数 ^{*1}	10000	指定主轴和从轴间电子齿轮的分子。
Ratio Denominator	齿轮比分母	UDINT ^{*2}	正数	10000	指定主轴和从轴间电子齿轮的分母。
Reference Type ^{*3}	位置种类选择	_eMC_REFERENCE_TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 ^{*4}	指定位置的类别。 0: 指令位置（前一任务周期 ^{*5} 中的计算值） 1: 反馈位置（同一任务周期 ^{*5} 中的获取值） 2: 指令位置（同一任务周期 ^{*5} 中的计算值）
MasterSyncPosition	主轴同步位置	LREAL	负数、正数、“0”	0	指定绝对坐标的主轴同步位置。单位为 [指令单位]。 ^{*6}
SlaveSyncPosition	从轴同步位置	LREAL	负数、正数、“0”	0	指定绝对坐标的从轴同步位置。单位为 [指令单位]。 ^{*3}

输入变量	名称	数据类型	有效范围	初始值	内容
Velocity	目标速度	LREAL	正数	0	指定目标速度。请务必设定目标速度。若不设定而直接动作，将发生异常。 单位为 [指令单位 /s]。 ^{*3}
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度。 单位为 [指令单位 /s ²]。 ^{*3}
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。 单位为 [指令单位 /s ²]。 ^{*3}
Jerk (Reserved)	跃度	LREAL	0	0	(Reserved)
BufferMode (Reserved)	缓冲 模式选择	_eMC_ BUFFER_ MODE	0: _mcAborting	0 ^{*2}	(Reserved)

- *1. 可在 CPU 单元版本 1.02 以上和 Sysmac Studio Ver.1.03 以上的组合中使用。在更低版本的组合中，数据类型为“UINT”，有效范围为“正数”。
- *2. 可在 CPU 单元版本 1.02 以上和 Sysmac Studio Ver.1.03 以上的组合中使用。在更低版本的组合中，数据类型为“UINT”。
- *3. 使用 _mcLatestCommand 时，使用的主轴和从轴必须满足以下关系。
Master（主轴）中设定的运动控制系统变量的轴编号 < Slave（从轴）中设定的运动控制系统变量的轴编号
- *4. 如果是有效范围为枚举型的变量初始值，实际不是数值，而是列举值。
- *5. 任务为主固定周期任务时，任务周期为主周期，为固定周期任务（执行优先度 5）时，任务周期为固定周期任务（执行优先度 5）的任务周期。
- *6. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》的“单位转换设定”。

输出变量

输出变量	名称	数据类型	有效范围	内容
StartSync	追踪中	BOOL	TRUE、 FALSE	为了同步而开始加减速动作时变为 TRUE。
InSync	同步中	BOOL	TRUE、 FALSE	从轴达到从轴同步位置时，变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、 FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、 FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

- * □ □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
StartSync	轴已开始移动时	<ul style="list-style-type: none"> • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时
InSync	从轴的位置达到 SlaveSyncPosition 时	<ul style="list-style-type: none"> • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时
Busy	Execute 的上升沿	<ul style="list-style-type: none"> • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时
Active	收到指令时	<ul style="list-style-type: none"> • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> • 由其他指令进行运动指令多重启动（中断），本指令中止时 • 因发生异常，本指令中止时 • 正在发生异常时，启动了本指令 • 正在执行 MC_Stop 指令时，启动了本指令 • 启动了 MC_GearOut 指令时 	<ul style="list-style-type: none"> • Execute 为 TRUE 时，与 Execute 的 FALSE 同时 • Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Master	主轴	_sAXIS_REF	—	指定主轴。 ^{*1}
Slave	从轴	_sAXIS_REF	—	指定从轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis***”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

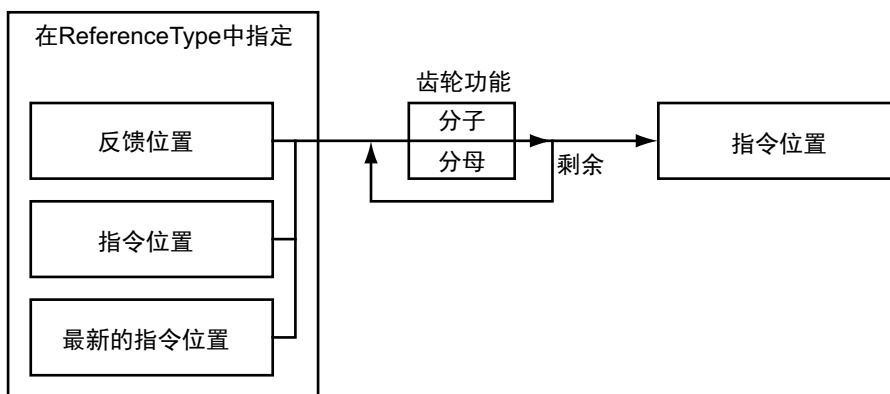


使用注意事项

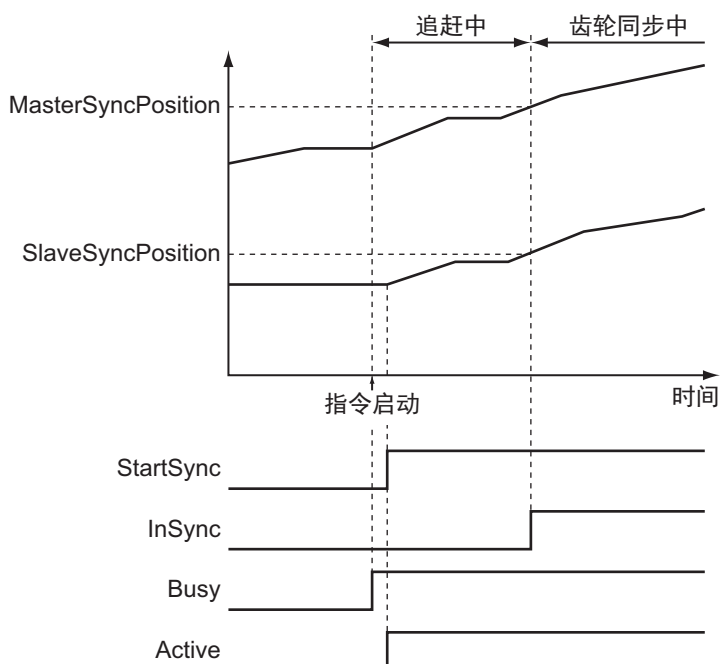
若主轴和从轴指定为相同的轴，将发生轻度故障等级的“主轴从轴相同（错误代码：5436 Hex）”异常。

功能说明

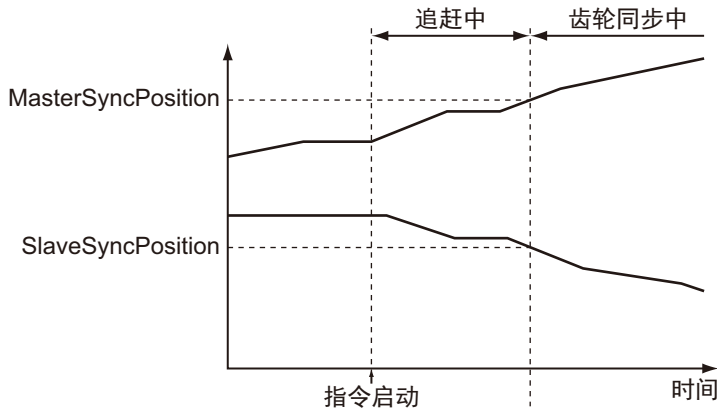
- 在 Slave（从轴）中指定动作对象的轴，指定 RatioNumerator（齿轮比分子）、RatioDenominator（齿轮比分母）、ReferenceType（位置种类选择）、Acceleration（加速度）、Deceleration（减速度），执行齿轮动作。
- Master（主轴）中可指定指令位置、反馈位置、最新指令位置。



- 动作开始后，Slave（从轴）将通过追赶动作与 Master（主轴同步），同时执行加减速动作。
- 从轴到达从轴同步位置前视为 Catching phase（追赶中），到达后称为 InSync phase（齿轮同步中）。无论在哪个区间，都同步到主轴的位置。
- 输入变量 Velocity（目标速度）为 Catching phase（追赶中）时的目标速度。
- 从轴向与开始动作时的主轴相同的方向动作。
动作开始时，如果主轴输入为“0”，将发生异常。
此外，主轴向正方向动作，且 SlaveSyncPosition（从轴同步位置）小于启动时的从轴位置时，从轴将执行反转动作。
同样，主轴向负方向动作，且 SlaveSyncPosition（从轴同步位置）大于启动时的从轴位置时，从轴也执行反转动作。
- 主轴的速度按周期大幅变动时，从轴的速度也会变得不稳定。
- 齿轮比为正数时，Slave（从轴）与 Master（主轴）的移动方向相同。



- 齿轮比为负数时，Slave（从轴）与 Master（主轴）的移动方向相反。

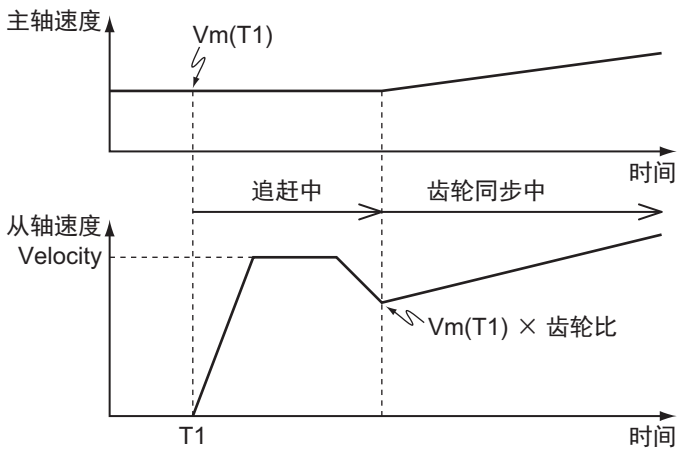


使用注意事项

在单元版本 1.09 以下的 CPU 单元中使用本指令时，请勿对 Master（主轴）执行 MC_SetPosition（当前位置变更）。若对 Master（主轴）执行 MC_SetPosition（当前位置变更）指令，Slave（从轴）可能发生剧烈追踪的危险。
 不对 Master（主轴）使用 MC_SetPosition（当前位置变更）指令时，请先解除 Master（主轴）和 Slave（从轴）的关系后再执行。

- MC 功能模块可按以下 3 种速度，使用 Acceleration（加速度）及 Deceleration（减速度）来计算进行直线加减速的速度曲线。
 - 将开始追赶动作时的 Slave（从轴）速度设为初始速度
 - 将开始追赶动作时的 Master（主轴）速度乘以齿轮比得到的速度设为最终速度
 - 将 Velocity（目标速度）设为目标速度

追赶中的移动量较短时，可能无法达到目标速度。



为了在 MasterSyncPosition（主轴同步位置）、SlaveSyncPosition（从轴同步位置）上 Slave（从轴）能追上 Master（主轴），Velocity（目标速度）需要满足以下条件。

$$\text{Velocity} > \frac{\text{MC_GearInPos 启动时的主轴速度} \times \text{齿轮比分子}}{\text{齿轮比分母}}$$

作为主轴速度使用的信息因位置种类 ReferenceType 的设定不同而异。

_mcCommand、_mcLatestCommand 设定时 : 使用指令当前速度。
_mcFeedback 设定时 : 使用反馈当前位置。

在 MasterSyncPosition (主轴同步位置)、SlaveSyncPosition (从轴同步位置), Slave (从轴) 无法追上 Master (主轴) 时, 将发生“位置指定齿轮动作目标速度不足 (错误代码: 5447 Hex)”。

- 计数模式为旋转模式时, 因 CPU 单元的版本不同, 有以下差别。
 - 单元版本 1.10 以上的 CPU 单元时
 主轴计数模式为旋转模式时, 可指定为超出循环计数器上下限设定值范围的 MasterSyncPosition (主轴同步位置)。此时, 主轴当前位置和主轴同步位置的关系与在 MC_MoveAbsolute (绝对值定位) 指令中将 Direction (方向选择) 指定为“无方向指定”时相同。MC_MoveAbsolute (绝对值定位) 指令请参考 □□「MC_MoveAbsolute(P.3-47)」。
 同样, 从轴计数模式为旋转模式时, 可指定为超出循环计数器上下限设定值范围的 SlaveSyncPosition (从轴同步位置)。
 - 单元版本 1.09 以下的 CPU 单元时
 计数模式为旋转模式时, 请设定为循环计数器在 1 周内开始同步动作。



使用注意事项

关于主轴的注意事项, 请参考 □□「同步控制的主轴及辅助轴的注意事项 (P.1-6)」。

● ReferenceType (位置种类选择)

从以下项目中选择位置种类。

- _mcCommand: 指令位置 (前一任务周期中的计算值)
 在当前周期, 使用前一任务周期中计算的主轴指令位置。
 使用计算从轴指令位置之前的固定周期任务中计算得到的主轴指令位置。
- _mcFeedback: 同一任务周期的获取值
 使用同一任务周期内获取的主轴的反馈位置。
- _mcLatestCommand: 指令位置 (同一任务周期中的计算值)
 使用同一任务周期中计算得到的主轴指令位置。
 可使用比 _mcCommand 更新的信息。但是, 主轴编号必须设定为比从轴编号更新的编号。
 如果从轴的轴编号比主轴新, 本指令的 Error (错误) 将变为 TRUE。ErrorID (错误代码) 中将输出“主轴 / 从轴 轴编号非升序 (错误代码: 5438 Hex)”。



使用注意事项

任务为主固定周期任务时, 这里的任务周期为主周期, 为固定周期任务 (执行优先度 5) 时, 任务周期为固定周期任务 (执行优先度 5) 的任务周期。同样, 固定周期任务表示主固定周期任务或固定周期任务 (执行优先度 5)。



参考

选择同一任务周期内计算得到的指令位置时，与前一任务周期中的计算值相比，同步精度更高。

但是，使用主轴和从轴时，必须保证“Master（主轴）中设定的运动控制系统变量的轴编号 < Slave（从轴）中设定的运动控制系统变量的轴编号”。

● 轴种类和位置种类的关系

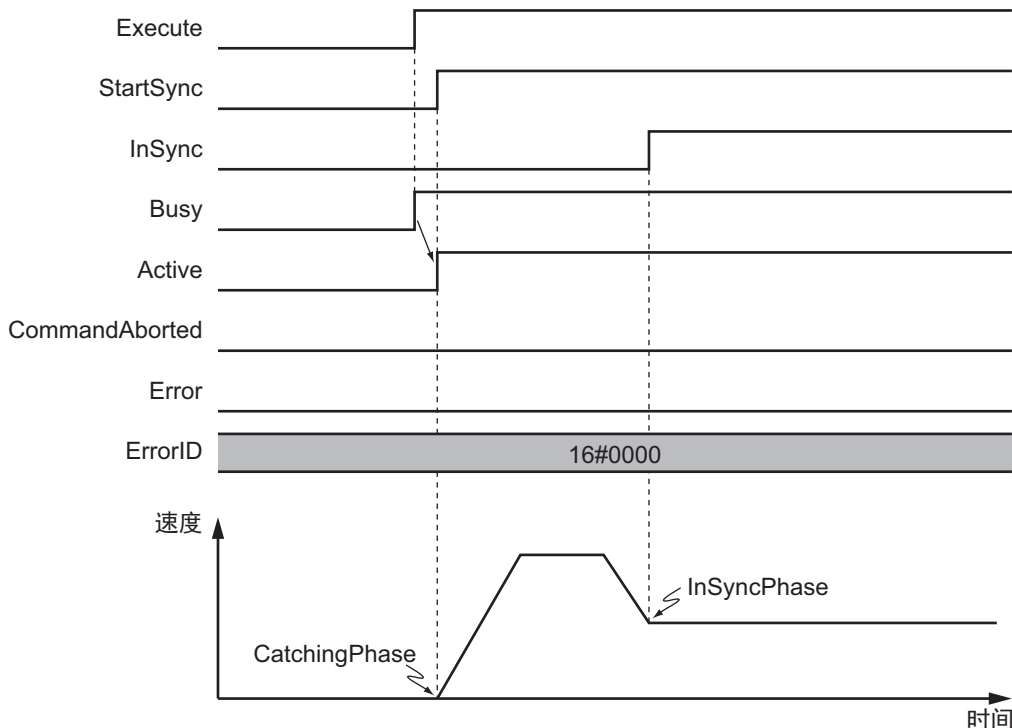
可监视的轴种类和要监视的位置种类的关系如下所示。

轴种类	ReferenceType	
	_mcCommand 或 _mcLatestCommand	_mcFeedback
伺服轴	○	○
编码器轴	x ^{*1}	○
虚拟伺服轴	○	○
虚拟编码器轴	x ^{*1}	○

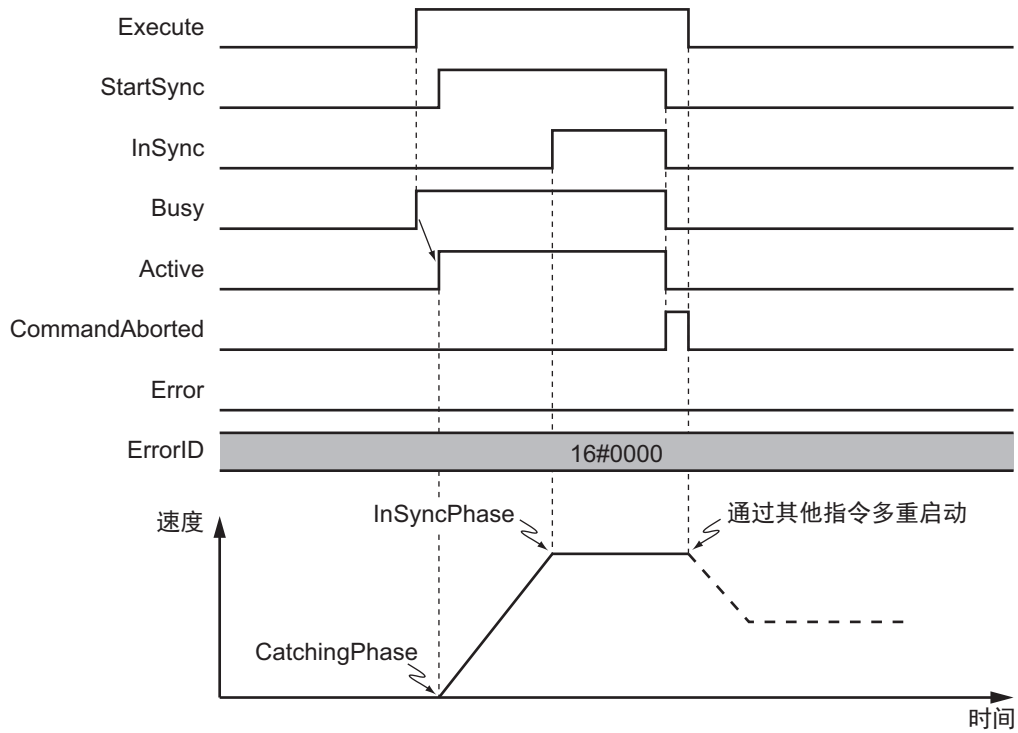
*1. 指令启动时，将发生“超出位置种类选择范围（错误代码：5430 Hex）”。

时序图

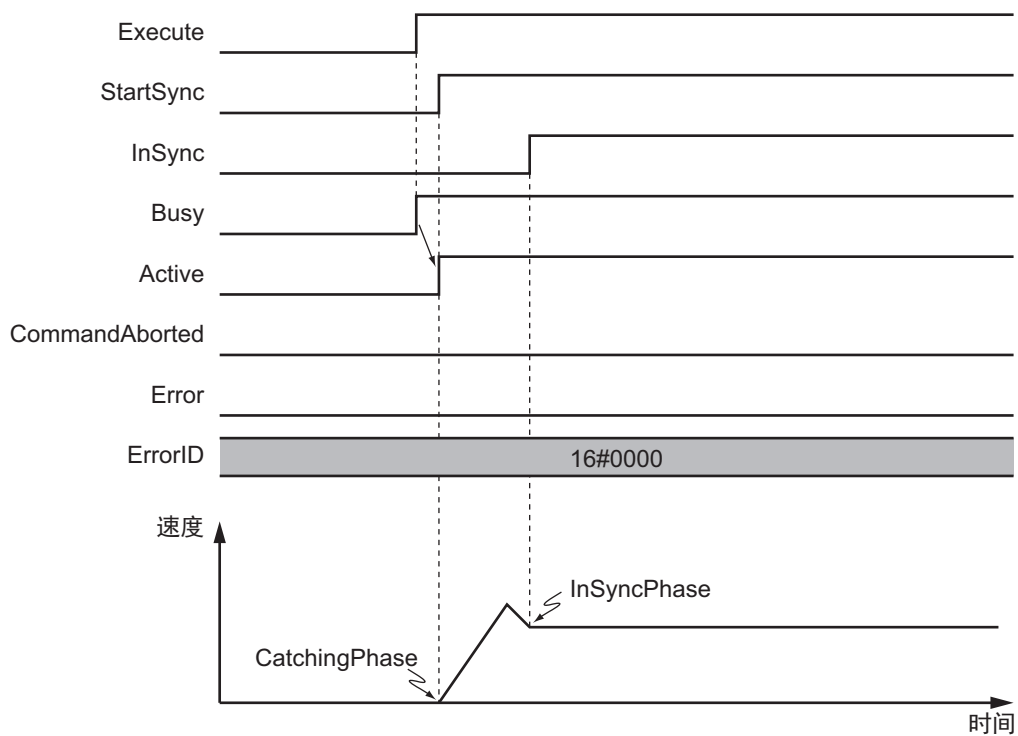
- 在 Execute（启动）的上升沿开始电子齿轮的动作。
- 在 Execute（启动）上升的同时 Busy（执行中）变为 TRUE。开始动作后，Active（控制中）和 StartSync（追踪中）变为 TRUE，Slave（从轴）开始齿轮动作。
- 到达 MasterSyncPosition（主轴同步位置）、SlaveSyncPosition（从轴同步位置）后，InSync（同步中）变为 TRUE。
- 通过其他指令中断了本指令时，CommandAborted（执行中断）变为 TRUE，Busy（执行中）、Active（控制中）、StartSync（追踪中）、InSync（同步中）变为 FALSE。



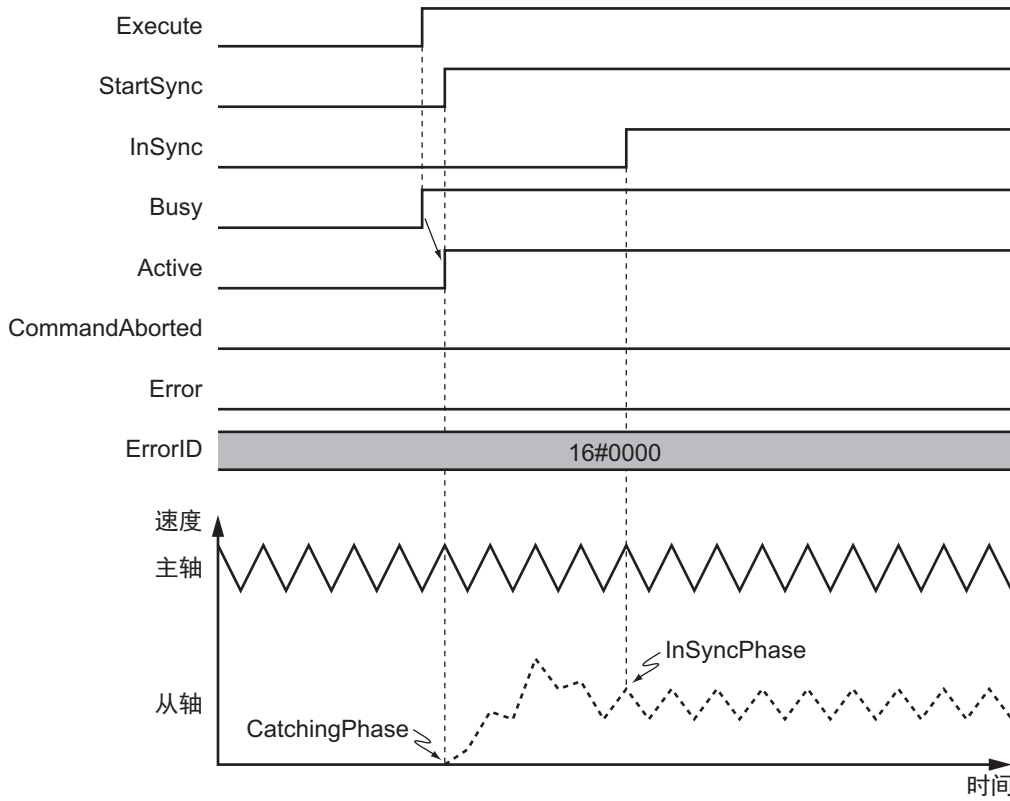
通过其他指令中断了本指令时，如下所示。



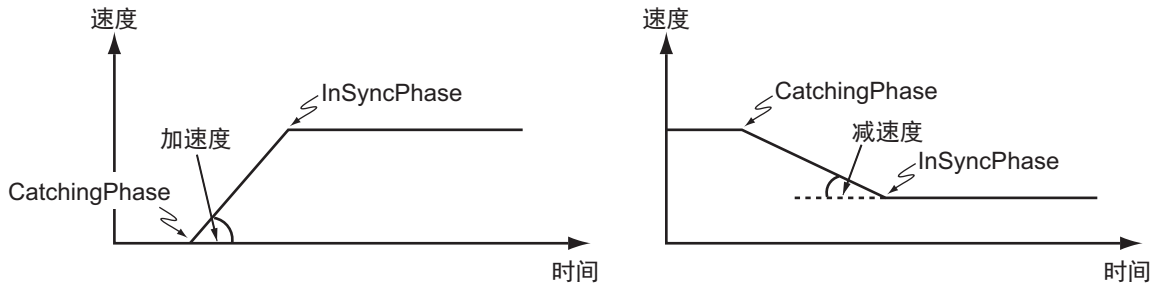
根据 SlaveSyncPosition 的设定，可能无法达到目标速度。此时的示例如下。



此外，到达 InSync phase 前，也会追踪主轴位置。此时的示例如下。



输入变量中可指定 Acceleration（加速度）、Deceleration（减速度）。电子齿轮的动作示例如下所示。



运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

包括异常停止中、减速停止中的主轴轴指令状态不会影响本指令的执行。

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 本指令执行过程中其他指令的启动

在 BufferMode (缓冲模式选择) 中指定多重启动时本指令的动作。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 切换指令后如果轴的动作方向发生反转，将按照轴参数 [反转时动作] 执行反转动作。

BufferMode (缓冲模式选择) 请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。



使用注意事项

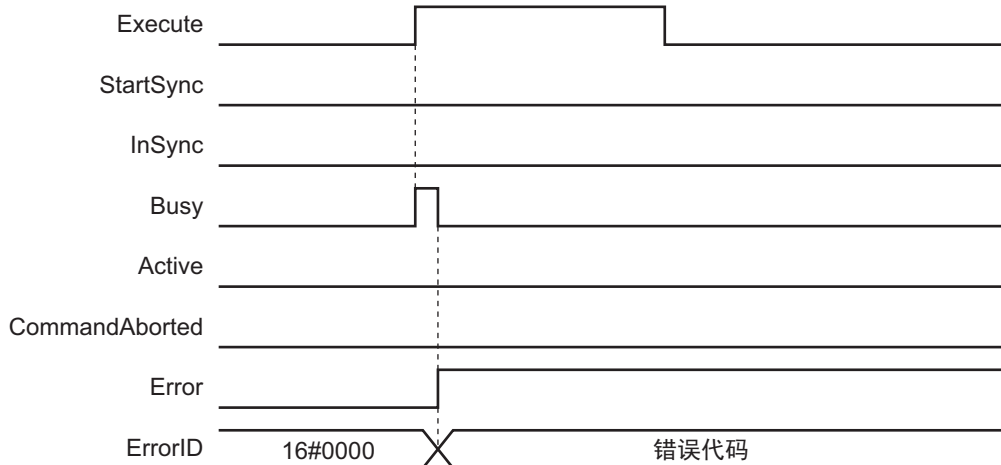
若以 BufferMode (缓冲模式选择) 的“0: 中断”以外的方式启动其他指令，其他指令将发生异常。

异常

如果在启动本指令时发生异常，Error (错误) 将变为 TRUE。

参照输出到 ErrorID (错误代码) 的值，可找出异常的原因。

● 发生异常时的时序图



同步控制过程中，不会受主轴的异常状态影响。解除主轴的异常状态，主轴动作时，从轴将继续电子齿轮动作。

本指令启动时以及本指令执行过程中，如果从轴发生异常，不会对主轴造成影响。

● 错误代码

指令中发生的异常请参考 □ 「A-1 错误代码一览 (P.A-2)」。

示例程序

下面介绍主轴的同步位置为 Pos1、从轴的同步位置为 Pos2 时的示例程序。

参数设定

下面表示本示例程序中使用的最低限度的设定。

● 设定轴参数 轴种类

轴	轴种类
轴 1	伺服轴（主轴）
轴 2	伺服轴（从轴）

计数器模式

轴	计数器模式
轴 1	旋转模式
轴 2	旋转模式

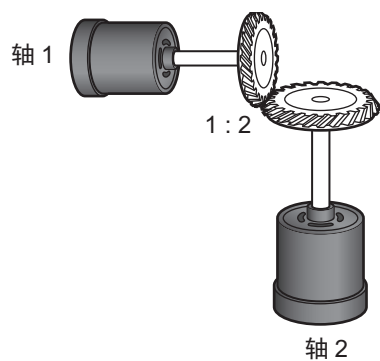
循环计数器

轴	上限值	下限值
轴 1	360	0
轴 2	360	0

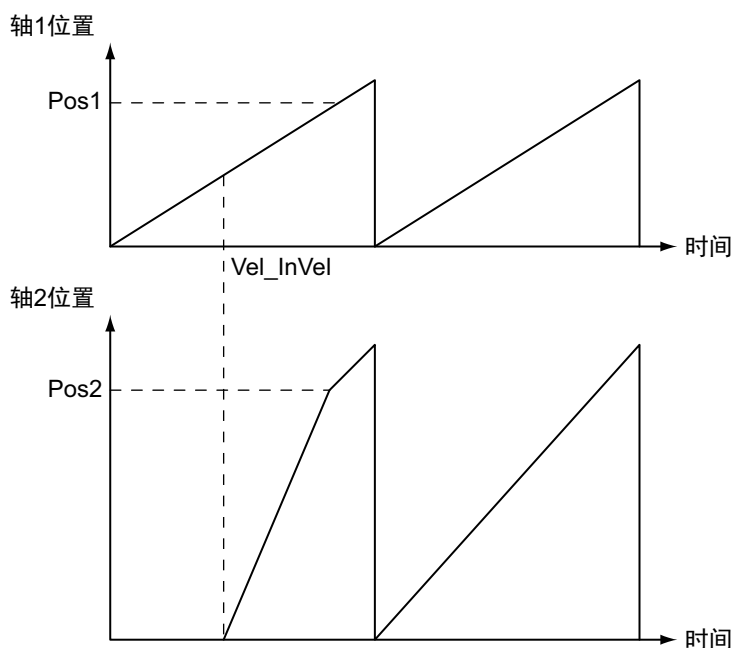
坐标单位选择

轴	坐标单位
轴 1	degree
轴 2	degree

动作示例



● 动作模式



1 主轴的开始

主轴（轴 1）为实际伺服轴，根据速度控制动作。

2 主轴达到目标速度

主轴的指令速度达到目标速度后，主轴的 InVelocity（达到目标速度）变为 TRUE。

3 从轴的启动

主轴的 InVelocity（达到目标速度）变为 TRUE 后，针对主轴的反馈位置，对从轴（轴 2）以齿轮比 1:2 执行齿轮动作。

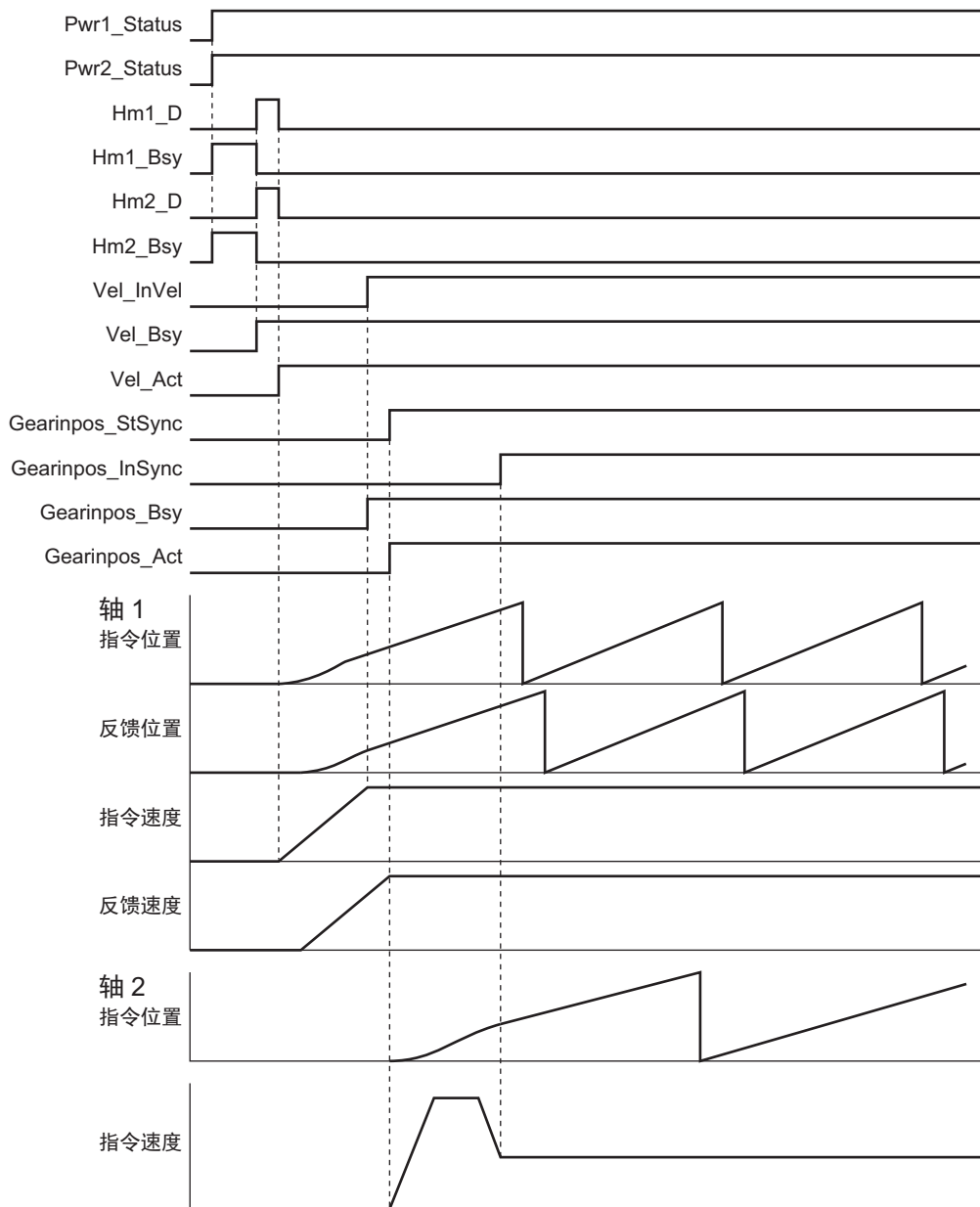
主轴的同步位置为 Pos1、从轴为 Pos2。

梯形图

● 主要变量

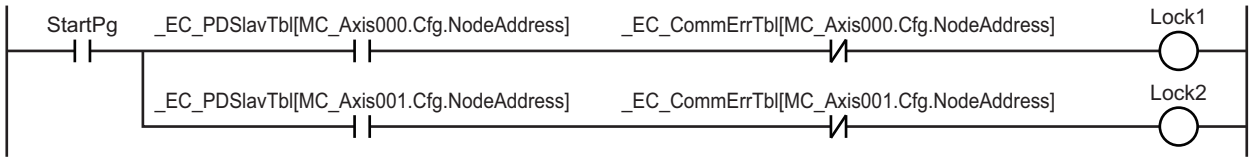
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	主轴轴 1 的轴变量。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis001	_sAXIS_REF	—	从轴轴 2 的轴变量。
MC_Axis001.Details.Homed	BOOL	FALSE	轴 2 处于原点确定状态时变为 TRUE。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴 2 发生轻度故障等级的异常时，变为 TRUE。
Pwr1_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR1 的输出变量 Status 中的变量。若变为伺服 ON 状态，该变量变为 TRUE。
Pwr2_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR2 的输出变量 Status 中的变量。若变为伺服 ON 状态，该变量变为 TRUE。
Hm1_D	BOOL	FALSE	分配到 MC_Home 的实例 HM1 的输出 Done 中的变量。
Hm2_D	BOOL	FALSE	分配到 MC_Home 的实例 HM2 的输出 Done 中的变量。
Vel_InVel	BOOL	FALSE	分配到 MC_MoveVelocity 的实例 VEL 的输出 InVelocity 中的变量。达到目标速度时，该变量变为 TRUE。
Pos1	LREAL	—	表示主轴同步位置的变量。
Pos2	LREAL	—	表示从轴同步位置的变量。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态，则变为伺服 ON 状态。

● 时序图

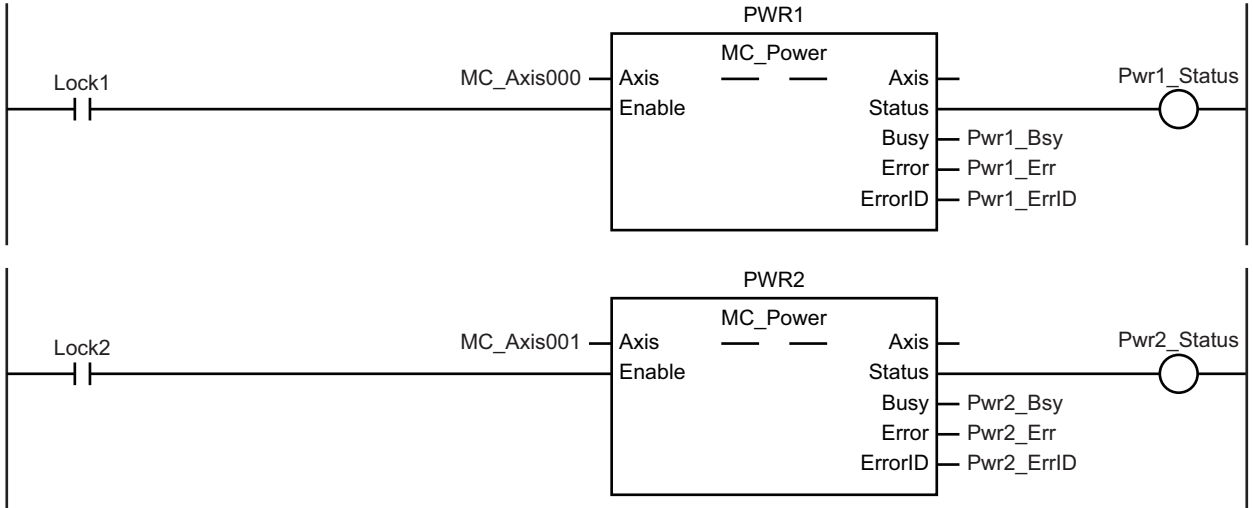


● 示例程序

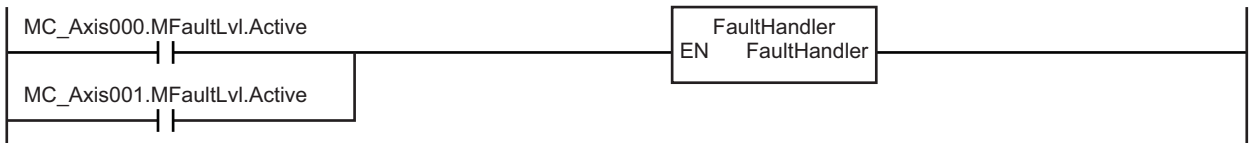
接点 StartPg 变为 TRUE 后，确认各轴的 EtherCAT 通信是否正常进行过程数据通信



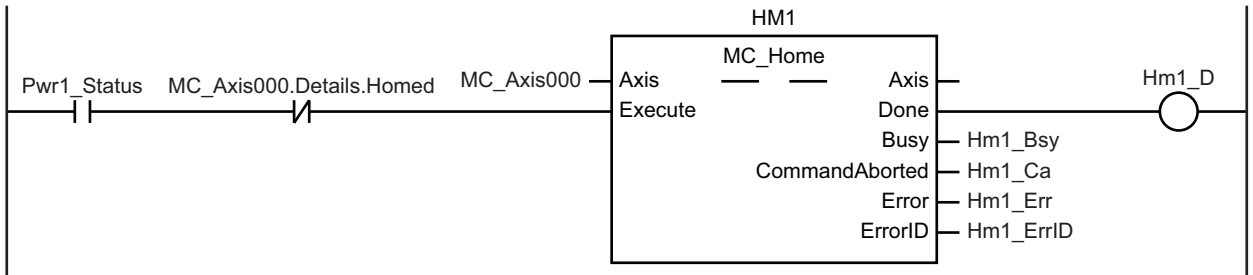
若正在进行过程数据通信，则将各轴设为伺服 ON 状态



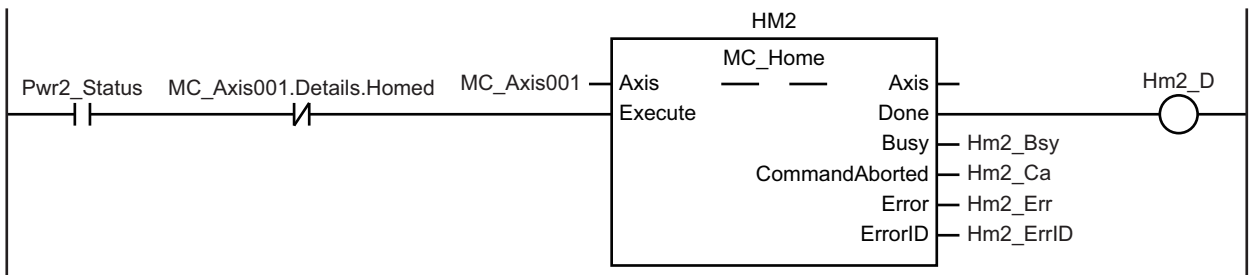
轴构成发生轻度故障等级的异常时，执行装置发生异常时的处理（FaultHandler）。发生异常时的处理（FaultHandler）请根据装置由客户进行编程



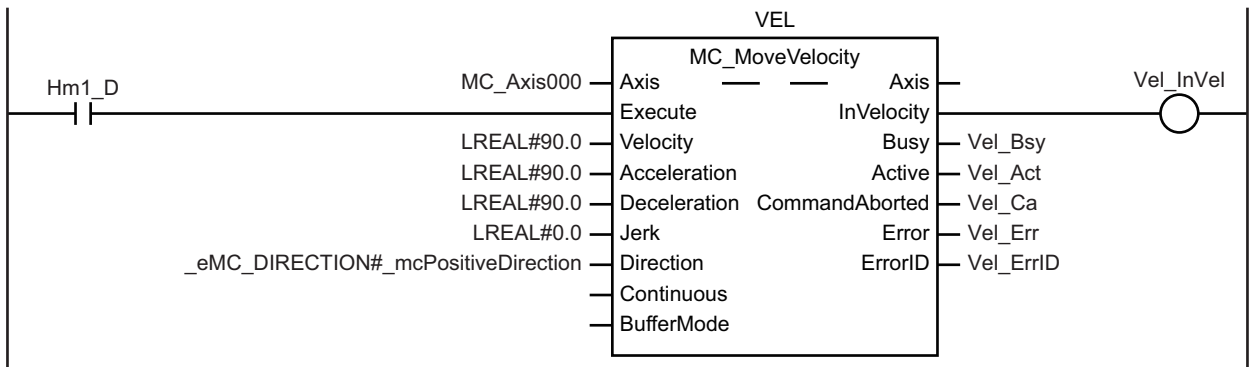
主轴（轴 1）为伺服 ON 状态且处于原点未确定状态时，执行原点复位，确定原点



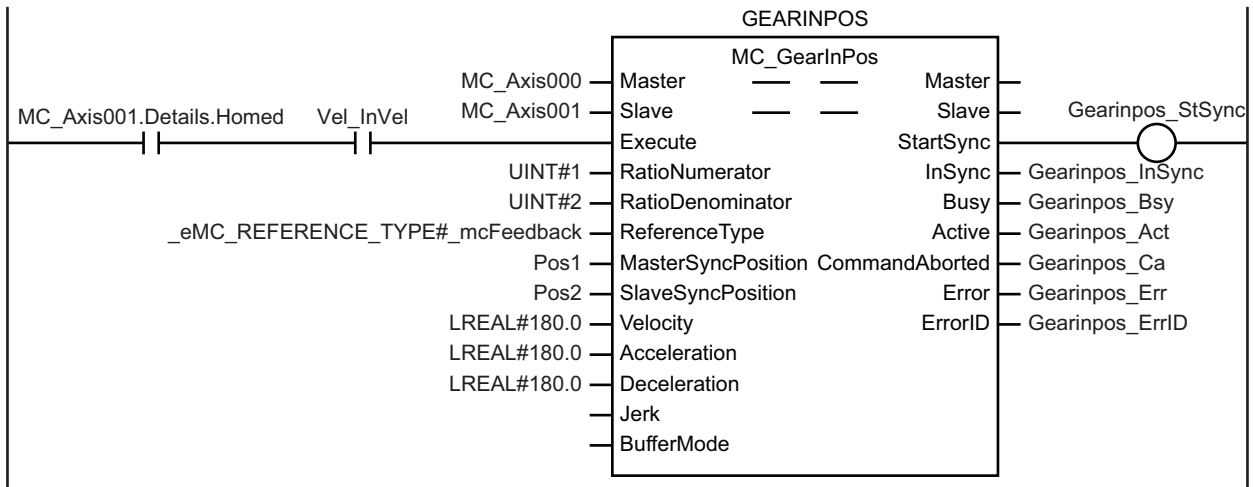
从轴（轴 2）为伺服 ON 状态且处于原点未确定状态时，执行原点复位，确定原点



主轴（轴 1）完成原点复位后，启动 MC_MoveVelocity（速度控制）指令



从轴（轴 2）完成原点复位后，MC_MoveVelocity 的 Vel_InVel 变为 TRUE，启动 MC_GearInPos（位置指定齿轮动作），开始齿轮动作



MC_GearInPos

3

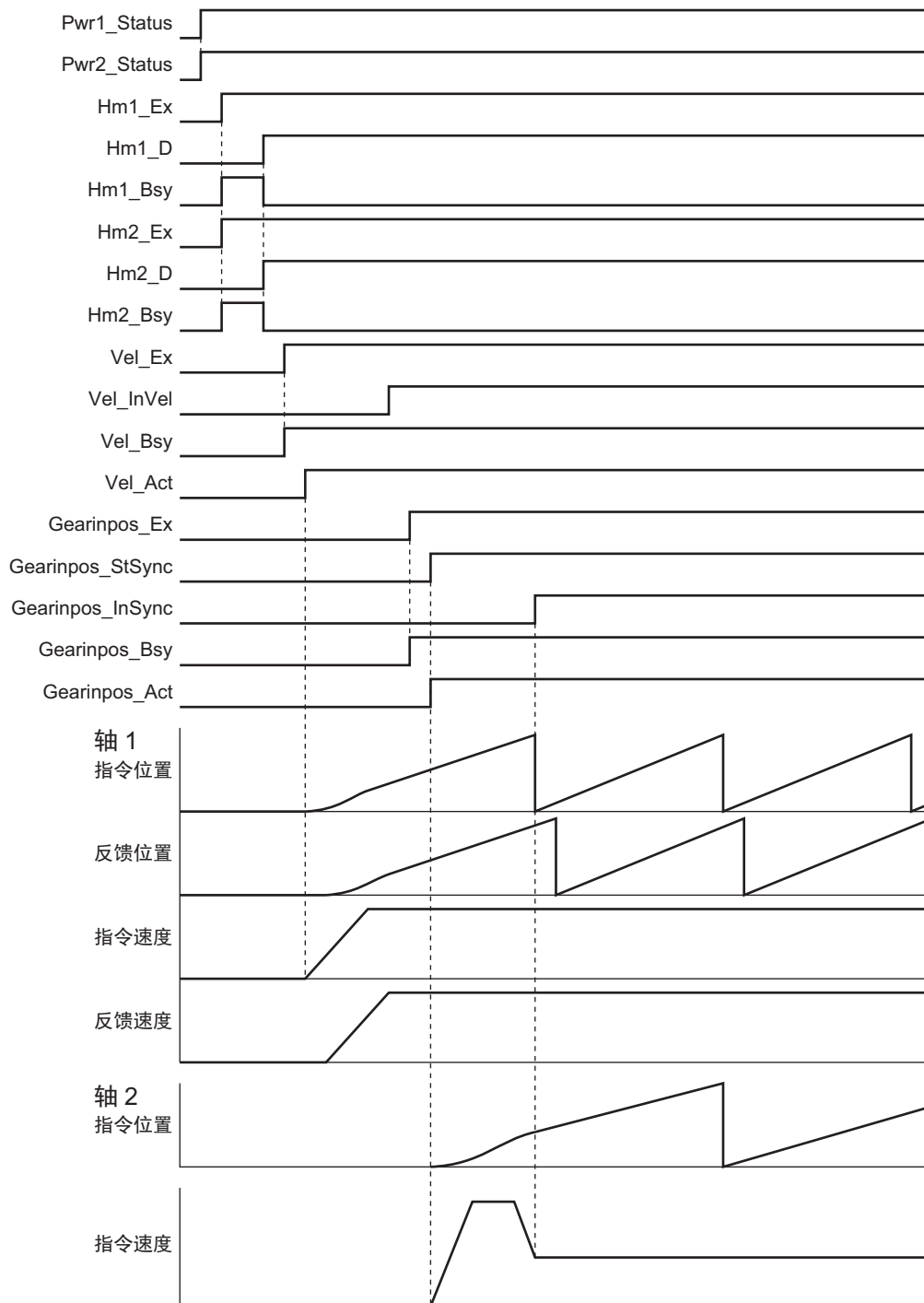
示例程序

结构文本 (ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	主轴轴 1 的轴变量。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时, 变为 TRUE。
MC_Axis001	_sAXIS_REF	—	从轴轴 2 的轴变量。
MC_Axis001.Details.Homed	BOOL	FALSE	轴 2 处于原点确定状态时变为 TRUE。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴 2 发生轻度故障等级的异常时, 变为 TRUE。
Pwr1_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR1 的输出变量 Status 中的变量。若变为伺服 ON 状态, 该变量变为 TRUE。
Pwr2_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR2 的输出变量 Status 中的变量。若变为伺服 ON 状态, 该变量变为 TRUE。
Hm1_D	BOOL	FALSE	分配到 MC_Home 的实例 HM1 的输出 Done 中的变量。
Hm2_D	BOOL	FALSE	分配到 MC_Home 的实例 HM2 的输出 Done 中的变量。
Vel_InVel	BOOL	FALSE	分配到 MC_MoveVelocity 的实例 VEL 的输出 InVelocity 中的变量。达到目标速度时, 该变量变为 TRUE。
Pos1	LREAL	—	表示主轴同步位置的变量。
Pos2	LREAL	—	表示从轴同步位置的变量。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态, 则变为伺服 ON 状态。
Gearinpos_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_GearInPos 的实例 GEARIN1。
Vel_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_MoveVelocity 的实例 VEL。
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE 时, 设定输入参数。输入参数设定完成后, 变为 TRUE。

● 时序图



● 示例程序

```

// 未设定输入参数时的处理
IF InitFlag = FALSE THEN
  // MC_MoveVelocity 参数
  Vel_Vel := LREAL#90.0;
  Vel_Acc := LREAL#90.0;
  Vel_Dec := LREAL#90.0;
  Vel_Dir := _eMC_DIRECTION#_mcPositiveDirection;

  // MC_GearInPos 参数
  Pos1 := LREAL#300.0;
  Pos2 := LREAL#200.0;
  Gearinpos_Rn := UINT#1;
  Gearinpos_Rd := UINT#2;
  Gearinpos_Rt := _eMC_REFERENCE_TYPE#_mcFeedback;
  Gearinpos_Mtpos := Pos1;
  Gearinpos_Svpos := Pos2;
  Gearinpos_Vel := LREAL#180.0;
  Gearinpos_Acc := LREAL#180.0;
  Gearinpos_Dec := LREAL#180.0;

  // 设定输入参数后, 将 InitFlag 设为 TRUE
  InitFlag:=TRUE;

END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 1 设为伺服 ON 状态
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
  Pwr1_En:=TRUE;
ELSE
  Pwr1_En:=FALSE;
END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 2 设为伺服 ON 状态
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
  Pwr2_En:=TRUE;
ELSE
  Pwr2_En:=FALSE;
END_IF;

// 轴 1 ~轴 2 发生轻度故障时, 执行异常处理 FaultHandler。
// 发生异常时的处理 (FaultHandler) 请根据装置由客户进行编程。
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
  FaultHandler();
END_IF;

// 轴 1 为可运行状态且处于原点未确定状态时, 执行原点复位
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
  Hm1_Ex:=TRUE;

```

```

END_IF;

// 轴 2 为可运行状态且处于原点未确定状态时, 执行原点复位
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
  Hm2_Ex:=TRUE;
END_IF;
// 轴 1 完成原点复位后, 启动 MC_MoveVelocity
IF Hm1_D=TRUE THEN
  Vel_Ex := TRUE;
END_IF;

// 轴 2 完成原点复位后, MC_MoveVelocity 的 Vel_InVel 变为 TRUE, 启动 MC_GearInPos
IF (MC_Axis001.Details.Homed=TRUE) AND (Vel_InVel=TRUE) THEN
  Gearinpos_Ex := TRUE;
END_IF;

// 轴 1 的 MC_Power
PWR1(
  Axis      := MC_Axis000,
  Enable    := Pwr1_En,
  Status    => Pwr1_Status,
  Busy      => Pwr1_Bsy,
  Error     => Pwr1_Err,
  ErrorID   => Pwr1_ErrID
);
// 轴 2 的 MC_Power
PWR2(
  Axis      := MC_Axis001,
  Enable    := Pwr2_En,
  Status    => Pwr2_Status,
  Busy      => Pwr2_Bsy,
  Error     => Pwr2_Err,
  ErrorID   => Pwr2_ErrID
);

// 轴 1 的 MC_Home
HM1(
  Axis          := MC_Axis000,
  Execute       := Hm1_Ex,
  Done          => Hm1_D,
  Busy          => Hm1_Bsy,
  CommandAborted => Hm1_Ca,
  Error         => Hm1_Err,
  ErrorID       => Hm1_ErrID
);
// 轴 2 的 MC_Home
HM2(
  Axis          := MC_Axis001,
  Execute       := Hm2_Ex,
  Done          => Hm2_D,
  Busy          => Hm2_Bsy,
  CommandAborted => Hm2_Ca,
  Error         => Hm2_Err,
  ErrorID       => Hm2_ErrID
);

//MC_MoveVelocity

```

```

VEL(
    Axis           := MC_Axis000,
    Execute        := Vel_Ex,
    Velocity       := Vel_Vel,
    Acceleration   := Vel_Acc,
    Deceleration   := Vel_Dec,
    Direction      := Vel_Dir,
    InVelocity     => Vel_InVel,
    Busy           => Vel_Bsy,
    Active         => Vel_Act,
    CommandAborted => Vel_Ca,
    Error          => Vel_Err,
    ErrorID        => Vel_ErrID
);

//MC_GearInPos
GEARINPOS(
    Master           := MC_Axis000,
    Slave           := MC_Axis001,
    Execute         := Gearinpos_Ex,
    RatioNumerator  := Gearinpos_Rn,
    RatioDenominator := Gearinpos_Rd,
    ReferenceType   := Gearinpos_Rt,
    MasterSyncPosition := Gearinpos_Mtpos,
    SlaveSyncPosition := Gearinpos_Svpos,
    Velocity        := Gearinpos_Vel,
    Acceleration    := Gearinpos_Acc,
    Deceleration    := Gearinpos_Dec,
    StartSync      => Gearinpos_StSync,
    InSync         => Gearinpos_InSync,
    Busy           => Gearinpos_Bsy,
    Active         => Gearinpos_Act,
    CommandAborted => Gearinpos_Ca,
    Error          => Gearinpos_Err,
    ErrorID        => Gearinpos_ErrID
);

```


MC_GearOut

中止正在执行的 MC_GearIn（凸轮动作开始）指令、MC_GearInPos（位置指定齿轮动作）指令。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_GearOut	齿轮动作解除	FB		<pre>MC_GearOut_instance (Slave := 《参数》, Execute := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, OutMode := 《参数》, Done => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。 单位为 [指令单位 /s ²]。 *1
Jerk (Reserved)	跃度	LREAL	0	0	(Reserved)
OutMode (Reserved)	同步解除 模式选择	_eMC_OUT_ MODE	0: _mcStop	0 *2	(Reserved)

*1. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》的“单位转换设定”。

*2. 如果是有效范围为枚举型的变量初始值，实际不是数值，而是枚举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、 FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、 FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* □ □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	指令完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动 (中断), 本指令中止时 因发生异常, 本指令中止时 正在发生异常时, 启动了本指令 正在执行 MC_Stop 指令时, 启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Slave	从轴	_sAXIS_REF	—	指定从轴。 ^{*1}

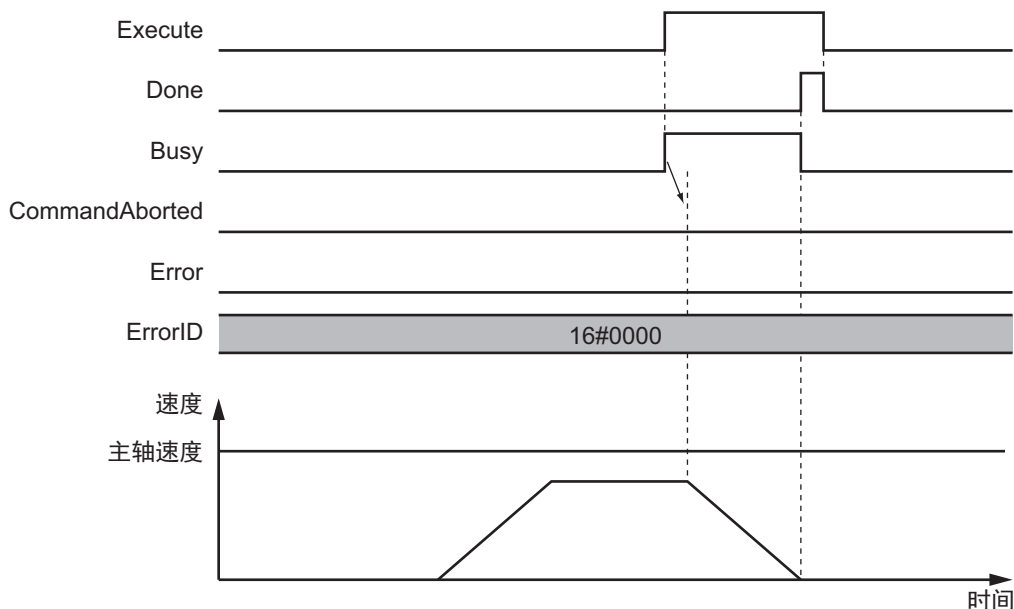
*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称 (默认 "MC_Axis****") 或系统定义变量的轴变量名称 (_MC_AX[*], _MC1_AX[*], _MC2_AX[*])。

功能说明

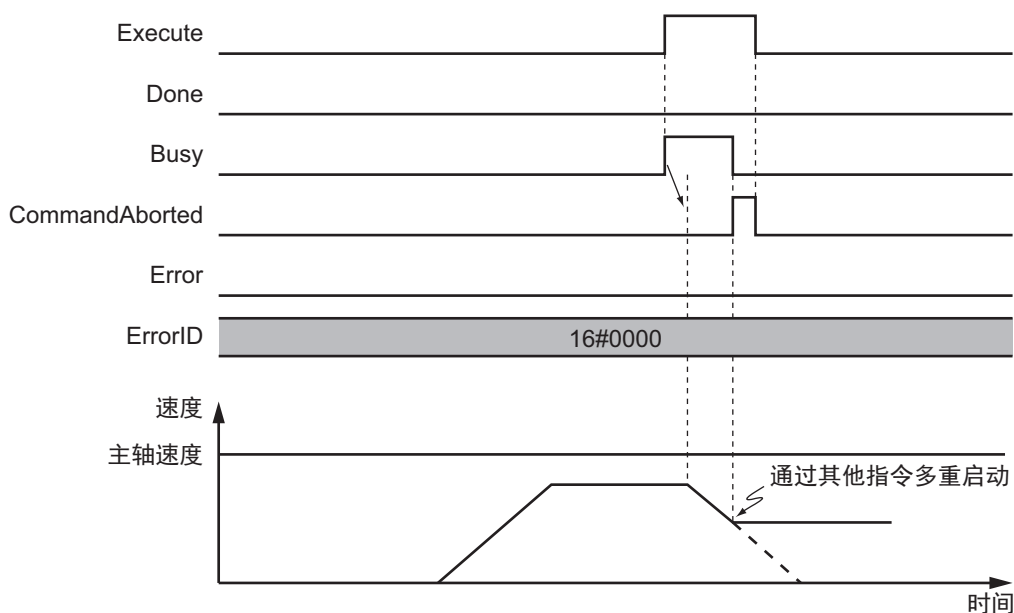
- 在 Slave（从轴）中指定动作对象的轴，并指定 Deceleration（减速度），中止正在执行的 MC_GearIn（凸轮动作开始）指令、MC_GearInPos（位置指定齿轮动作）指令。
- 本指令不会对 MC_GearIn（凸轮动作开始）指令、MC_GearInPos（位置指定齿轮动作）指令的主轴动作造成影响。

时序图

- 在 Execute（启动）上升的同时 Busy（执行中）变为 TRUE。
- 达到目标速度后，Done（完成）变为 TRUE。
- 通过其他指令中断了本指令时，CommandAborted（执行中断）变为 TRUE，Busy（执行中）变为 FALSE。



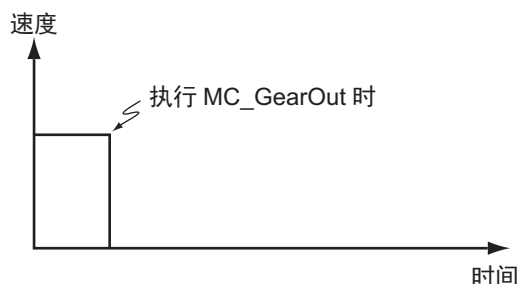
● 通过其他指令中断了本指令时



● 将减速度指定为“0”并启动时

将减速度指定为“0”并启动时，将不进行减速而直接停止。

减速度为“0”时的动作示例如下。



运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 其他指令执行过程中本指令的启动

执行 MC_GearIn（齿轮动作开始）指令或 MC_GearInPos（位置指定齿轮动作）指令的过程中，启动了本指令时，MC_GearIn（齿轮动作开始）指令或 MC_GearInPos（位置指定齿轮动作）指令将变为 CommandAborted（执行中断），本指令变为 Busy（执行中）。

执行 MC_GearIn（齿轮动作开始）指令或 MC_GearInPos（位置指定齿轮动作）指令以外的指令时，若启动了本指令，本指令将变为 Error（错误）。

● 本指令执行过程中其他指令的启动

对本指令执行运动指令多重启动时，请指定从轴。

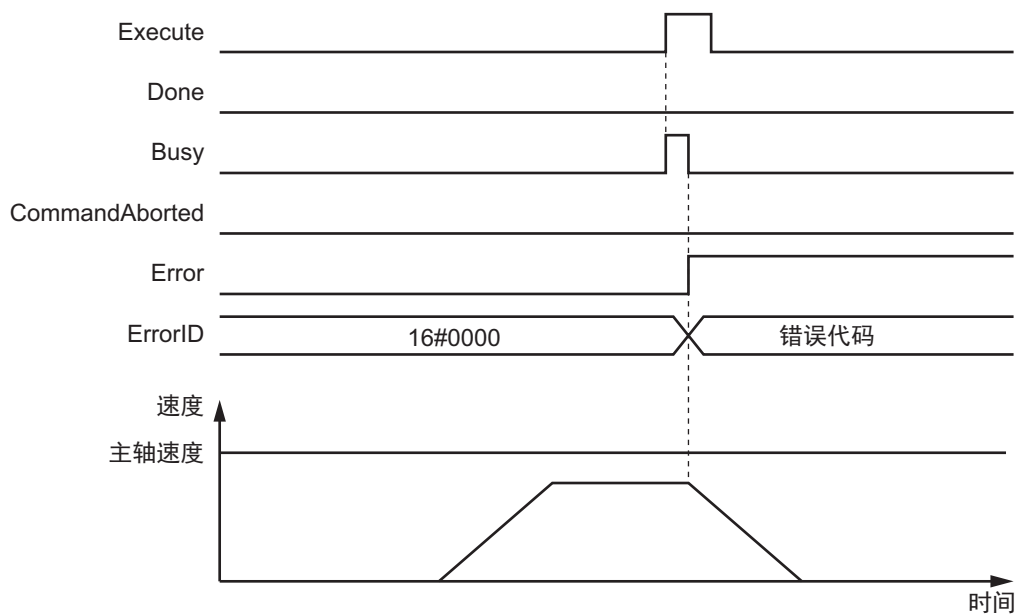
对本指令执行运动指令多重启动时，可进行中断、缓冲。

异常

如果在启动本指令时发生异常，Error（错误）将变为 TRUE。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

MC_MoveLink

与指定的主轴同步进行定位。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_MoveLink	梯形模式 凸轮	FB		<pre> MC_MoveLink_instance (Master := 《参数》, Slave := 《参数》, TriggerInput := 《参数》, TriggerVariable := 《参数》, Execute := 《参数》, ReferenceType := 《参数》, SlaveDistance := 《参数》, MasterDistance := 《参数》, MasterDistanceInACC := 《参数》, MasterDistanceInDEC := 《参数》, LinkOption := 《参数》, MasterStartDistance := 《参数》, BufferMode := 《参数》, Done => 《参数》, InSync => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》); </pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Reference Type ^{*1}	位置种类选择	_eMC_REFERENCE_TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 ^{*2}	指定位置的类别。 0: 指令位置（前一任务周期 ^{*3} 中的计算值） 1: 反馈位置（同一任务周期 ^{*3} 中的获取值） 2: 指令位置（同一任务周期 ^{*3} 中的计算值）
Slave Distance	从轴移动距离	LREAL	负数、正数、“0”	0	指定从轴的移动距离。 单位为 [指令单位]。 ^{*4}
Master Distance	主轴移动距离	LREAL	正数或“0”	0	以无符号的绝对值指定主轴的移动距离。主轴的移动方向为正负时均有效。 单位为 [指令单位]。 ^{*4}

输入变量	名称	数据类型	有效范围	初始值	内容
MasterDistancelnACC	主轴加速移动距离	LREAL	正数或“0”	0	以无符号的绝对值指定从轴加速时主轴的移动距离。主轴的移动方向为正负时均有效。 单位为 [指令单位]。*3
MasterDistancelnDEC	主轴减速移动距离	LREAL	正数或“0”	0	以无符号的绝对值指定从轴减速时主轴的移动距离。主轴的移动方向为正负时均有效。 单位为 [指令单位]。*3
LinkOption	同步开始条件选择	_eMC_LINKOPTION	0: _mcCommandExecution 1: _mcTriggerDetection 2: _mcMasterReach	0*2	指定从轴与主轴同步的条件。 0: 启动时 1: 检测到触发时 2: 主轴到达主轴追踪距离时
MasterStartDistance	主轴追踪距离	LREAL	负数、正数、“0”	0	指定从轴开始追踪主轴时绝对坐标的主轴位置。 单位为 [指令单位]。*3
BufferMode	缓冲模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0*2	指定运动指令多重启动时的动作。 0: 中断 1: 缓冲

*1. 使用 _mcLatestCommand 时，使用的主轴和从轴必须满足以下关系。

Master（主轴）中设定的运动控制系统变量的轴编号 < Slave（从轴）中设定的运动控制系统变量的轴编号

*2. 如果是有效范围为列举型的变量初始值，实际不是数值，而是列举值。

*3. 任务为主固定周期任务时，任务周期为主周期，为固定周期任务（执行优先级 5）时，任务周期为固定周期任务（执行优先级 5）的任务周期。

*4. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》的“单位转换设定”。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
InSync	同步中	BOOL	TRUE、FALSE	开始同步动作时变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常

* □ □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	指令完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
InSync	同步开始条件成立时	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	收到指令时	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动 (中断), 本指令中止时 因发生异常, 本指令中止时 正在发生异常时, 启动了本指令 正在执行 MC_Stop 指令时, 启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Master	主轴	_sAXIS_REF	—	指定主轴。 ^{*1}
Slave	从轴	_sAXIS_REF	—	指定从轴。 ^{*1}
TriggerInput	触发输入条件	_sTRIGGER_REF	—	设定触发条件。 ^{*2}
TriggerVariable	触发变量	BOOL	TRUE、FALSE	在触发条件中指定了控制器模式时, 指定作为触发的输入变量。

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称 (默认 "MC_Axis****") 或系统定义变量的轴变量名称 (_MC_AX[*], _MC1_AX[*], _MC2_AX[*])。

*2. 请定义 _sTRIGGER_REF 型用户定义变量。

● _sTRIGGER_REF

成员变量	名称	数据类型	有效范围	功能
Mode	模式	_eMC_TRIGGER_MODE	0: _mcDrive 1: _mcController	指定触发的模式。 0: 驱动器模式 1: 控制器模式
LatchID	锁定 ID 选择	_eMC_TRIGGER_LATCH_ID	0: _mcLatch1 1: _mcLatch2	驱动器模式时, 指定使用两个锁定功能中的哪个。 0: 锁定功能 1 1: 锁定功能 2
InputDrive	驱动器触发输入信号	_eMC_TRIGGER_INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	指定驱动器模式时的伺服驱动器触发信号。 0: Z 相 1: 外部输入

功能说明

- 从轴与指定的主轴同步进行动作。
- 电子凸轮的一种，但相对于主轴，从轴按梯形曲线执行动作。
- 若要停止正在根据本指令动作的轴，请使用 MC_Stop（强制停止）指令。



使用注意事项

在单元版本 1.09 以下的 CPU 单元中使用本指令时，请勿对 Master（主轴）执行 MC_SetPosition（当前位置变更）。

若对 Master（主轴）执行 MC_SetPosition（当前位置变更）指令，Slave（从轴）可能发生剧烈追踪的危险。

不对 Master（主轴）使用 MC_SetPosition（当前位置变更）指令时，请先解除 Master（主轴）和 Slave（从轴）的关系后再执行。

关于主轴的注意事项，请参考 □□「同步控制的主轴及辅助轴的注意事项 (P.1-6)」。

● 数据对象的映射

在 LinkOption（同步开始条件选择）中选择 [_mcTriggerDetection]，Mode 选择为 [驱动器模式]，使用 MC_MoveLink（梯形模式凸轮）指令时，请映射以下对象数据。

映射在 Sysmac Studio 的轴基本设定 [详细设定] 中进行。

- 锁定功能（60B8Hex）
- 锁定状态（60B9Hex）
- 锁定位置 1（60BAHex）
- 锁定位置 2（60BCHex）

未设定以上内容时，将发生过程数据对象设定不足（错误代码：3461Hex）。

数据对象的映射请参考 □□「2-3 PDO 映射 (P.2-30)」，□□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

使用 NX 系列位置接口单元时请参考 □□《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》的“I/O 入口映射”。

指令的详情

下面介绍指令的详情。

● Master（主轴）指定

在 Master（主轴）中指定主轴。

● Slave（从轴）指定

在 Slave（从轴）中指定从轴。

● TriggerInput（触发输入条件）、TriggerVariable（触发变量）

同步开始条件指定为 _mcTriggerDetectoin 时，指定开始同步的输入信号。

输入信号的选择、发生触发的时序请参考 □□「MC_TouchProbe(P.3-329)」。

TriggerInput（触发输入条件）的 Mode（模式）指定为驱动器模式时，使用从轴的驱动器输入。

指定为控制器模式时，将 TriggerVariable（触发变量）作为触发信号使用。

● ReferenceType（位置种类选择）

从以下项目中选择位置种类。

- `_mcCommand`: 指令位置（前一任务周期中的计算值）
在当前周期，使用前一任务周期中计算的主轴指令位置。
使用计算从轴指令位置之前的固定周期任务中计算得到的主轴指令位置。
- `_mcFeedback`: 同一任务周期的获取值
使用同一任务周期内获取的主轴的反馈位置。
- `_mcLatestCommand`: 指令位置（同一任务周期中的计算值）
使用同一任务周期中计算得到的主轴指令位置。
可使用比 `_mcCommand` 更新的信息。但是，主轴编号必须设定为比从轴编号更新的编号。
如果从轴的轴编号比主轴新，本指令的 Error（错误）将变为 TRUE。ErrorID（错误代码）中将输出“主轴 / 从轴 轴编号非升序（错误代码：5438 Hex）”。



使用注意事项

任务为主固定周期任务时，这里的任务周期为主周期，为固定周期任务（执行优先级 5）时，任务周期为固定周期任务（执行优先级 5）的任务周期。同样，固定周期任务表示主固定周期任务或固定周期任务（执行优先级 5）。



参考

选择同一任务周期内计算得到的指令位置时，与前一任务周期中的计算值相比，同步精度更高。
但是，使用主轴和从轴时，必须保证“Master（主轴）中设定的运动控制系统变量的轴编号 < Slave（从轴）中设定的运动控制系统变量的轴编号”。

● 轴种类和位置种类的关系

可监视的轴种类和要监视的位置种类的关系如下所示。

轴种类	ReferenceType	
	<code>_mcCommand</code> 或 <code>_mcLatestCommand</code>	<code>_mcFeedback</code>
伺服轴	○	○
编码器轴	x ^{*1}	○
虚拟伺服轴	○	○
虚拟编码器轴	x ^{*1}	○

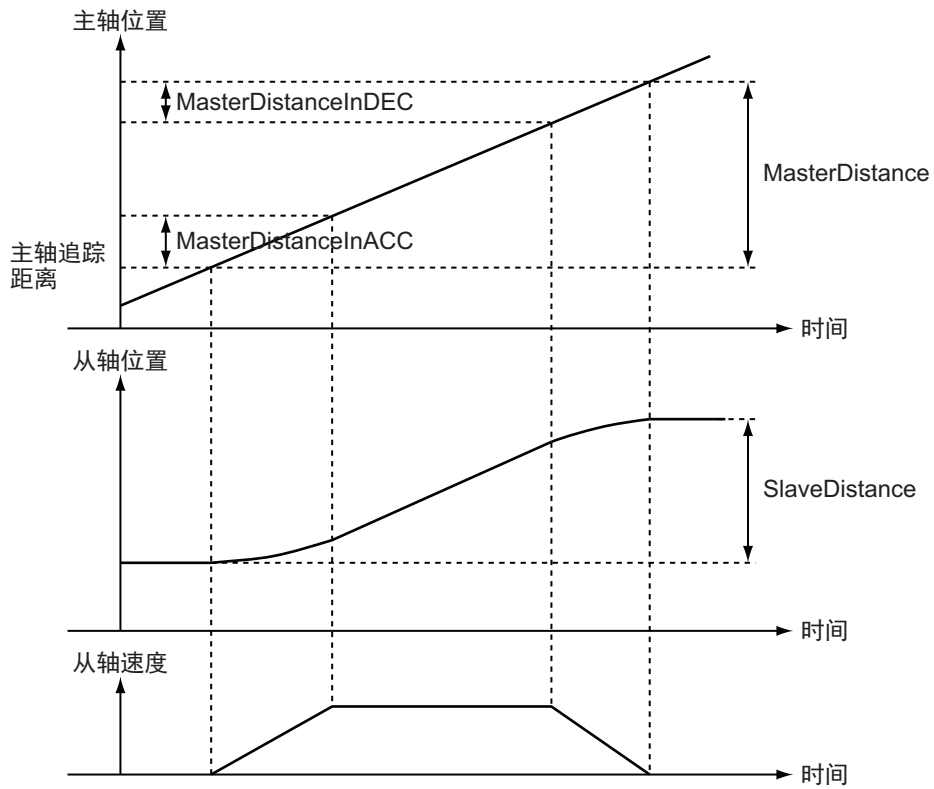
*1. 指令启动时，将发生“超出位置种类选择范围（错误代码：5430 Hex）”。

● SlaveDistance（从轴移动距离）、MasterDistance（主轴移动距离）、MasterDistanceInACC（主轴加速移动距离）、MasterDistanceInDEC（主轴减速移动距离）

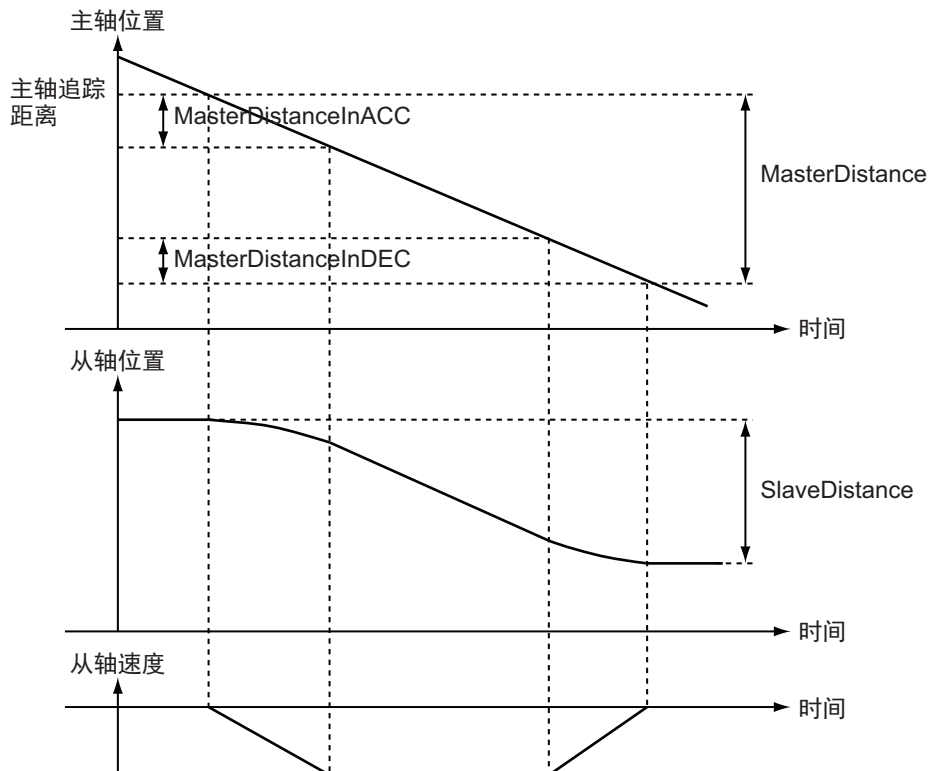
如下图所示，从轴的速度和位置根据主轴和从轴的移动量之比决定。

下图的主轴追踪距离表示同步开始条件变为有效时的位置。

主轴向正方向移动时



主轴向负方向移动时



主轴移动量和从轴移动量的关系如下表所示。

区间	主轴移动量和从轴移动量的关系	
加速	主轴	主轴加速移动距离
	从轴	$\text{从轴移动距离} \times \frac{\text{主轴加速移动距离}}{2} + (\text{主轴移动距离} - \text{主轴加速移动距离} - \text{主轴减速移动距离}) + \frac{\text{主轴减速移动距离}}{2}$
等速	主轴	主轴移动距离 - 主轴加速移动距离 - 主轴减速移动距离
	从轴	从轴移动距离 - 上述加速时的从轴移动距离 - 下述减速时的从轴移动距离
减速	主轴	主轴减速移动距离
	从轴	$\text{从轴移动距离} \times \frac{\text{主轴减速移动距离}}{2} + (\text{主轴移动距离} - \text{主轴加速移动距离} - \text{主轴减速移动距离}) + \frac{\text{主轴加速移动距离}}{2}$

主轴的等速区间为负数时，将发生等速移动量异常并停止。

此外，要以与主轴相同的速度输送从轴时，从轴移动量请指定为以下值。

$$\text{从轴移动距离} = \frac{\text{主轴加速移动距离}}{2} + (\text{主轴移动距离} - \text{主轴加速移动距离} - \text{主轴减速移动距离}) + \frac{\text{主轴减速移动距离}}{2}$$



使用注意事项

主轴的计数器模式为旋转模式时，MasterDistance（主轴移动距离）请指定为主轴循环计数器1圈内的值。

● LinkOption（同步开始条件选择）

指定从轴与主轴同步的条件。

- 指令开始时
本指令启动时，从下一周期开始，从轴与主轴同步进行定位。
- 检测到触发时
从输入触发指定中指定的输入信号变为有效后的下一周期开始，从轴与主轴同步进行定位。
- 主轴到达主轴追踪距离时
本指令启动过程中，主轴达到主轴追踪距离时，从下一周期开始，从轴同步进行定位。
此外，在主轴停止于主轴追踪距离的状态下启动本指令时，从下一周期开始，同步进行定位。



使用注意事项

LinkOption（同步开始条件选择）选择为 [_mcTriggerDetection]，Mode 选择为 [驱动器模式] 时，需要进行对象数据的映射。

请对以下对象进行设定。

- 锁定功能（60B8Hex）
- 锁定状态（60B9Hex）
- 锁定位置1（60BAHex）
- 锁定位置2（60BCHex）

未设定以上内容时，将发生过程数据对象设定不足（错误代码：3461Hex）。对象数据的映射请参考 □□「2-3 PDO 映射 (P.2-30)」、□□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● MasterStartDistance（主轴追踪距离）

以绝对位置指定从轴开始同步时主轴的位置。

单元版本 1.10 以上的 CPU 单元时

主轴计数模式为旋转模式时，可指定为超出循环计数器上下限设定值范围的 MasterStartDistance（主轴追踪距离）。此时，主轴当前位置和主轴追踪距离的关系与在 MC_MoveAbsolute（绝对值定位）指令中将 Direction（方向选择）指定为“无方向指定”时相同。

MC_MoveAbsolute（绝对值定位）指令请参考 □□「MC_MoveAbsolute(P.3-47)」。

单元版本 1.09 以下的 CPU 单元时

主轴计数模式为旋转模式时，请在循环计数器上下限设定值范围内指定 MasterStartDistance（主轴追踪距离）。若指定为超出范围的值，执行指令时，将输出“超出凸轮动作（主轴追踪）开始位置设定范围（错误代码：547B Hex）”。

● BufferMode（缓冲模式选择）

指定前一轴动作和本次动作的连接方法。

有以下 2 种。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。
缓冲	当前正在执行的指令正常完成后，自动启动已缓冲的本指令。

BufferMode（缓冲模式选择）请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 位置检查

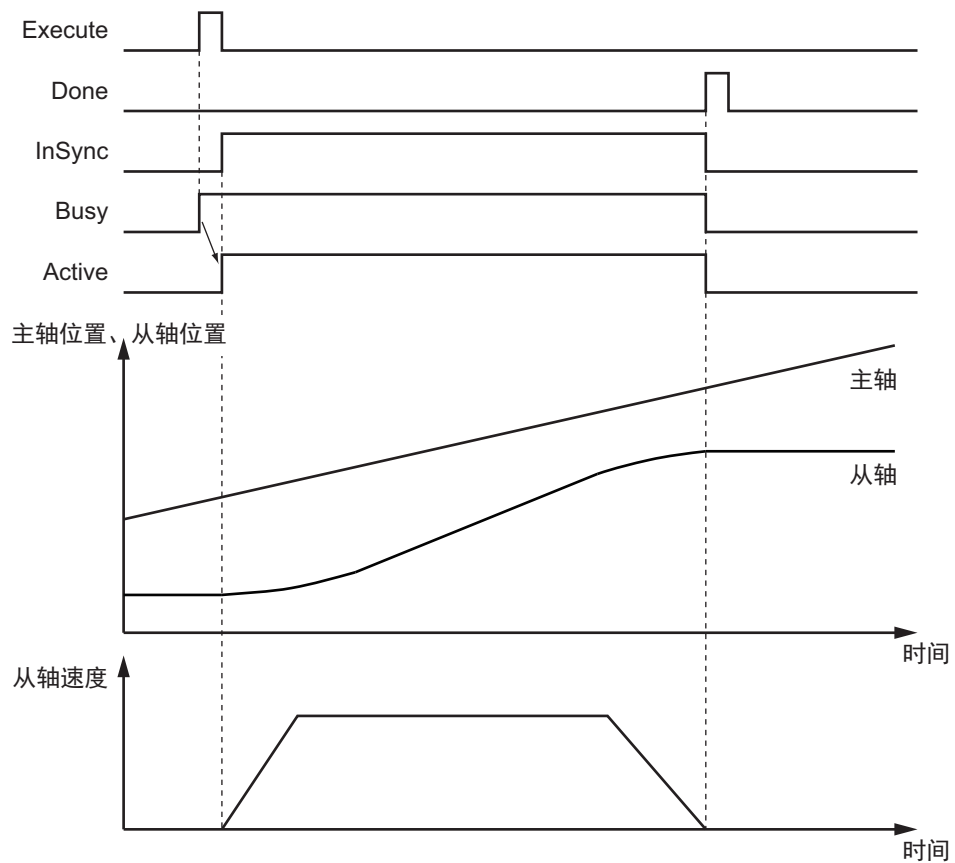
在从轴的动作完成时，执行位置检查。

本指令的位置检查根据轴参数 [位置宽度]、[位置检查时间] 的设定执行。

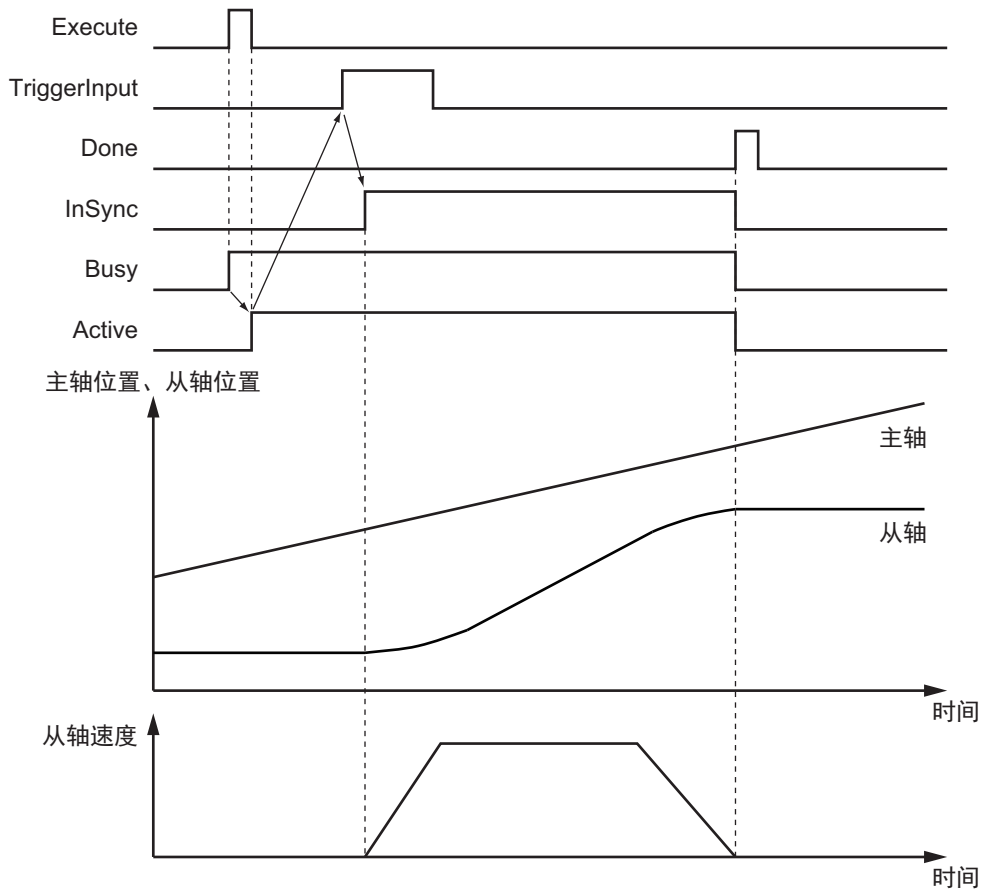
时序图

● 启动本指令时

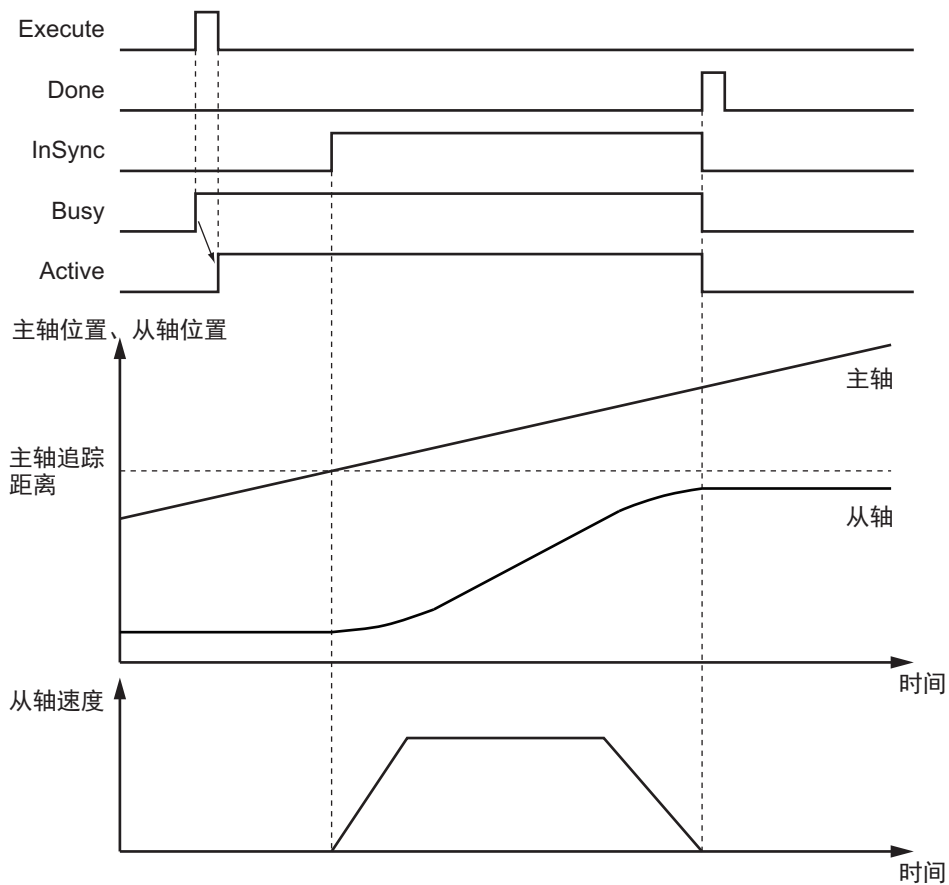
同步开始条件指定=指令开始时



同步开始条件指定 = 检测到触发时



同步开始条件指定 = 主轴到达主轴追踪距离时



- 在前一个动作中，以中断的方式启动 BufferMode（缓冲模式选择）时

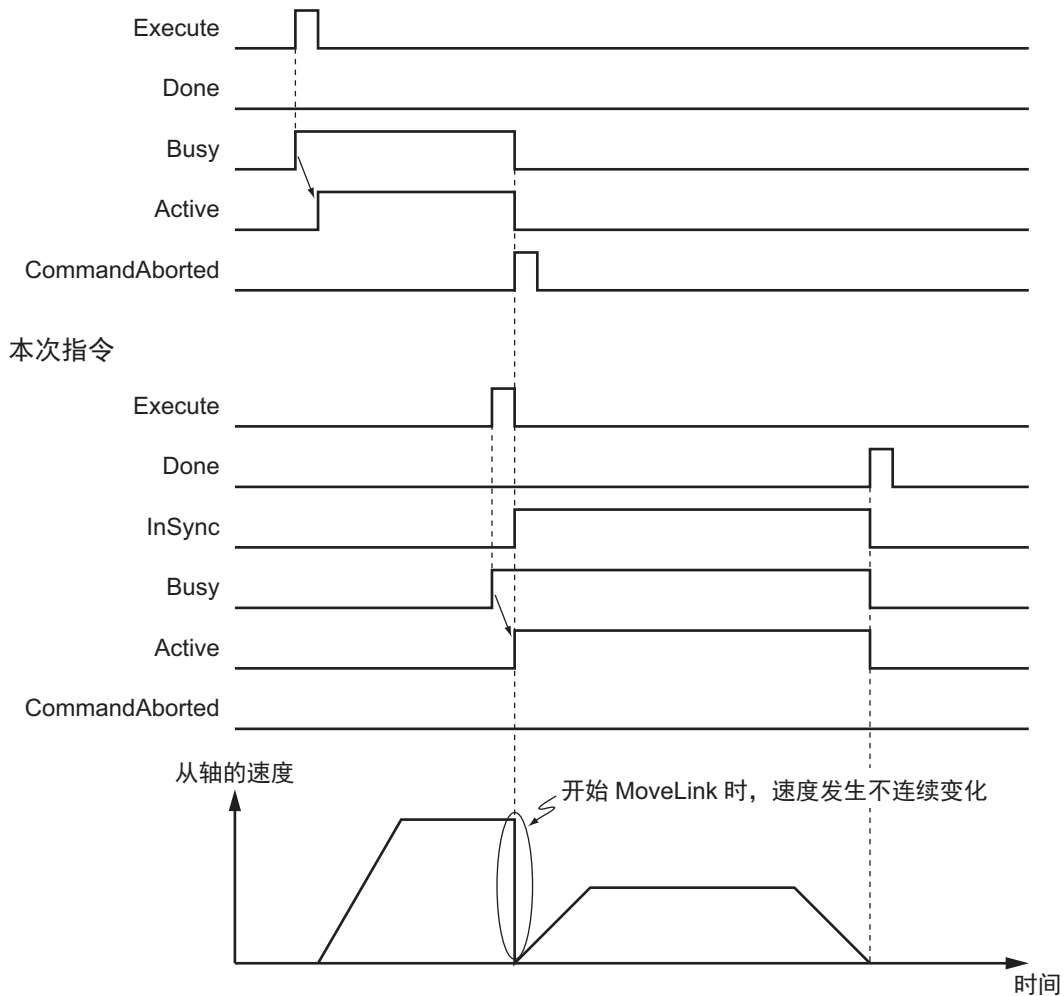
下面介绍前一动作正在执行 MC_MoveAbsolute（绝对值定位）指令时，在同步开始条件指定中指定为指令开始，并执行本指令的情况。



使用注意事项

如下图所示，开始本指令时，从轴的速度为不连续，敬请注意。

前一指令为 MC_MoveAbsolute（绝对值定位）时



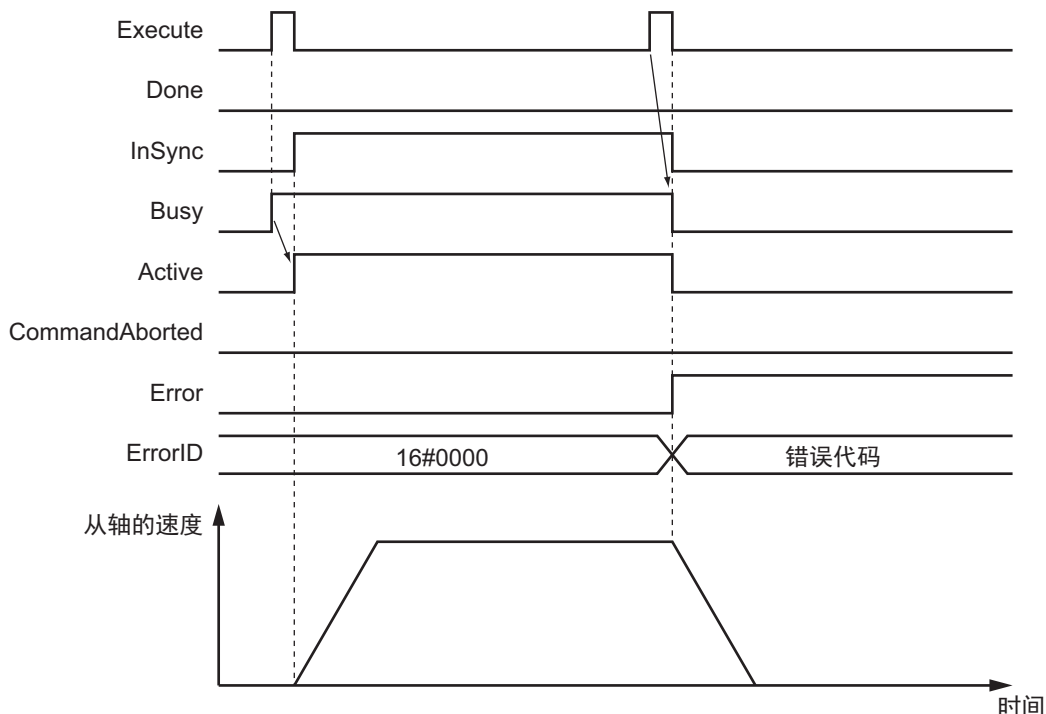
- 在前一个动作中，以缓冲的方式启动 BufferMode（缓冲模式选择）时

前一动作完成后，执行本指令。

运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”，该轴的动作停止。



运动指令多重启动

指令多重启动的详情请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 本指令执行过程中其他指令的启动

对本指令执行运动指令多重启动时，可进行中断、缓冲。

不可通过共混进行指令多重启动。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

发生异常后的时序图请参考 □ 「运动指令重新执行 (P.3-277)」。

● 错误代码

指令中发生的异常请参考 □ 「A-1 错误代码一览 (P.A-2)」。

示例程序

下面介绍裁切控制相关的示例程序。

参数设定

下面表示本示例程序中使用的最低限度的设定。

● 设定轴参数 轴种类

轴	轴种类
轴 1	伺服轴 (主轴)
轴 2	伺服轴 (从轴)

计数器模式

轴	计数器模式
轴 1	旋转模式
轴 2	直线模式

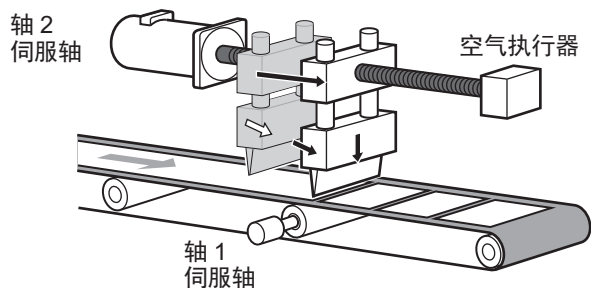
循环计数器

轴	上限值	下限值
轴 1	360	0

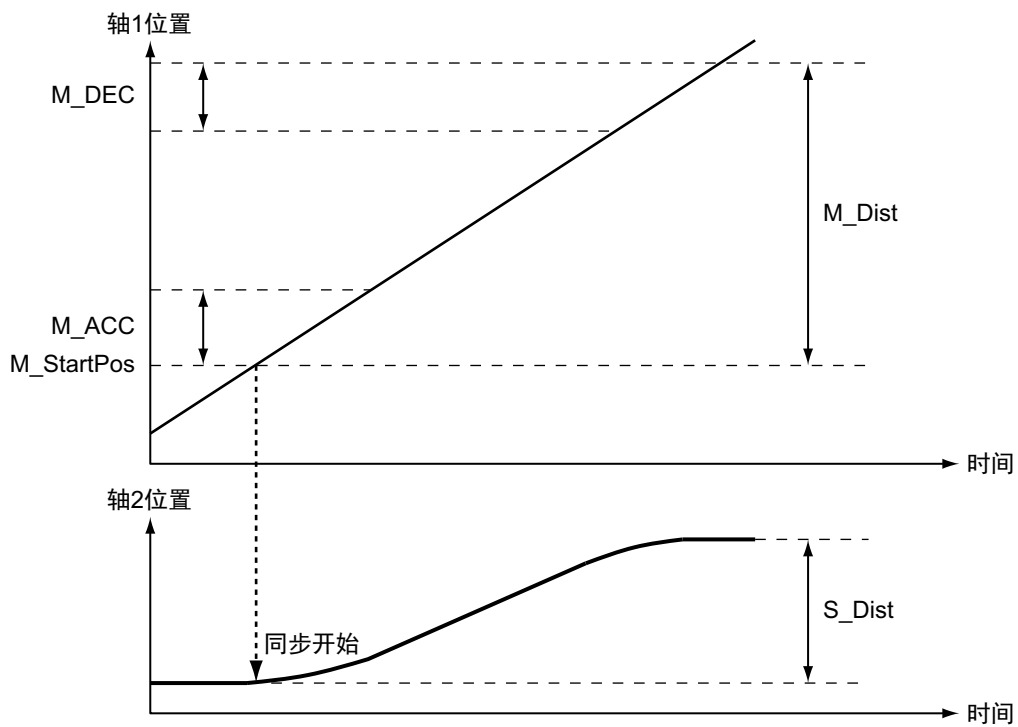
坐标单位选择

轴	坐标单位
轴 1	degree
轴 2	mm

动作示例



● 动作模式



1 主轴的开始

将皮带输送机上安装的轴 1 作为主轴，反馈位置。

2 从轴的启动

按水平方向动作的滚珠丝杆上安装的轴 2 与轴 1 同步进行动作。

3 利用切刀裁切

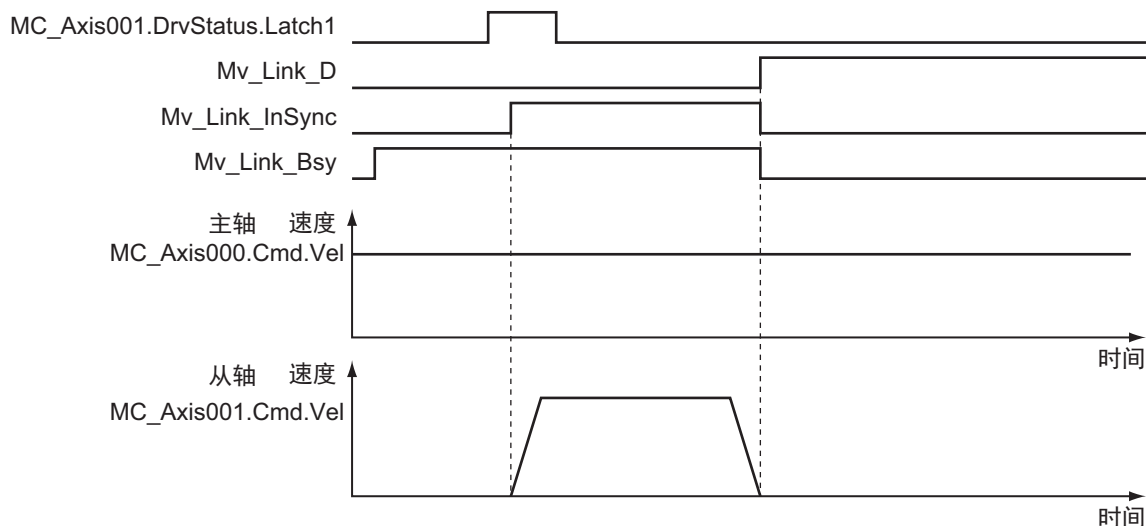
轴 2 正在同步时，空气执行器变为“ON”，空气执行器上连接的切刀垂直下降，进行工件的裁切。

梯形图

● 主要变量

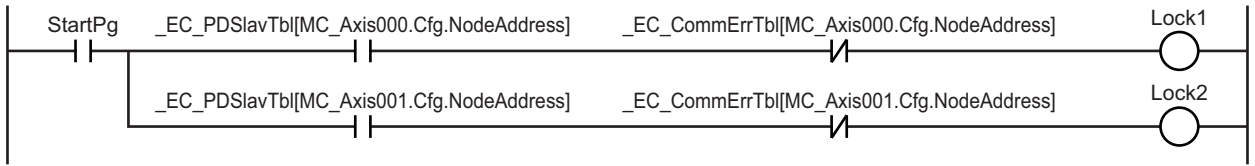
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	主轴轴 1 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
MC_Axis001	_sAXIS_REF	—	从轴轴 2 的轴变量。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴 2 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis001.Details.Homed	BOOL	FALSE	轴 2 处于原点确定状态时变为 TRUE。
Pwr1_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR1 的输出变量 Status 中的变量。若变为伺服 ON 状态，该变量变为 TRUE。
Pwr2_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR2 的输出变量 Status 中的变量。若变为伺服 ON 状态，该变量变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态，则变为伺服 ON 状态。
Actuator	BOOL	FALSE	轴 1 和轴 2 同步后，变为 TRUE。 Actuator 为 TRUE 时，切刀垂直下降。
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE 时，设定输入参数。输入参数设定完成后，变为 TRUE。

● 时序图

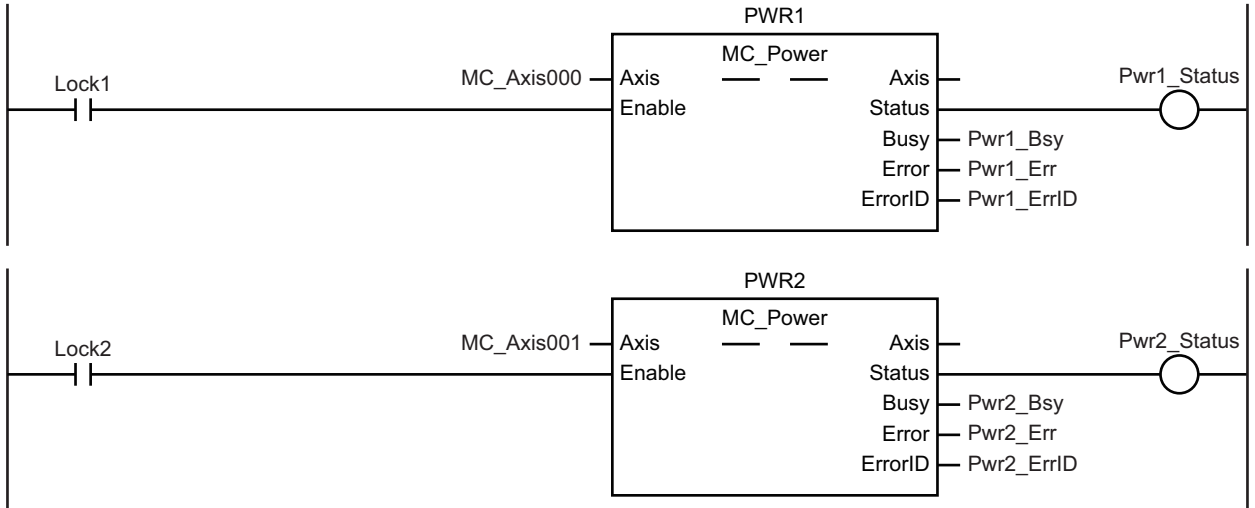


● 示例程序

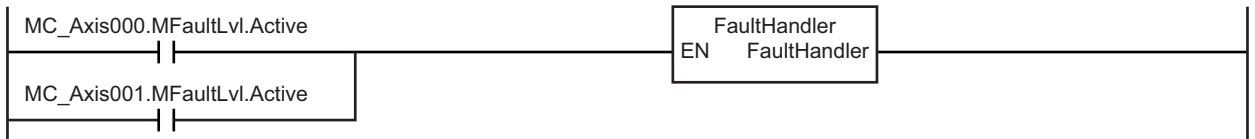
接点 StartPg 变为 TRUE 后，确认各轴的 EtherCAT 通信是否正常进行过程数据通信



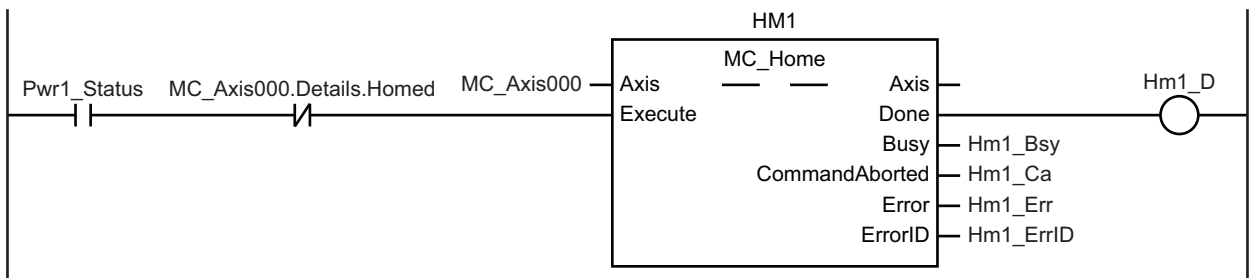
若正在进行过程数据通信，则将各轴设为伺服 ON 状态



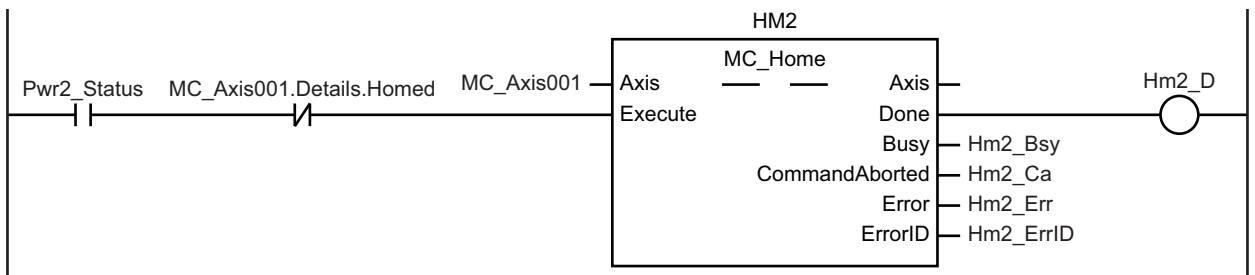
轴构成发生轻度故障等级的异常时，执行装置发生异常时的处理（FaultHandler）。发生异常时的处理（FaultHandler）请根据装置由客户进行编程



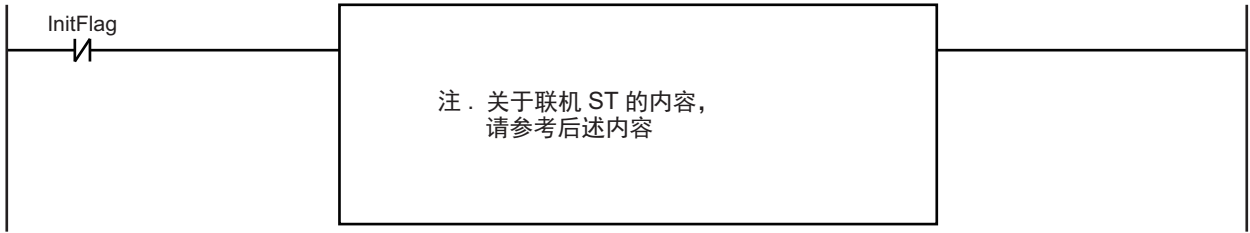
轴 1 为伺服 ON 状态且处于原点未确定状态时，执行原点复位，确定原点



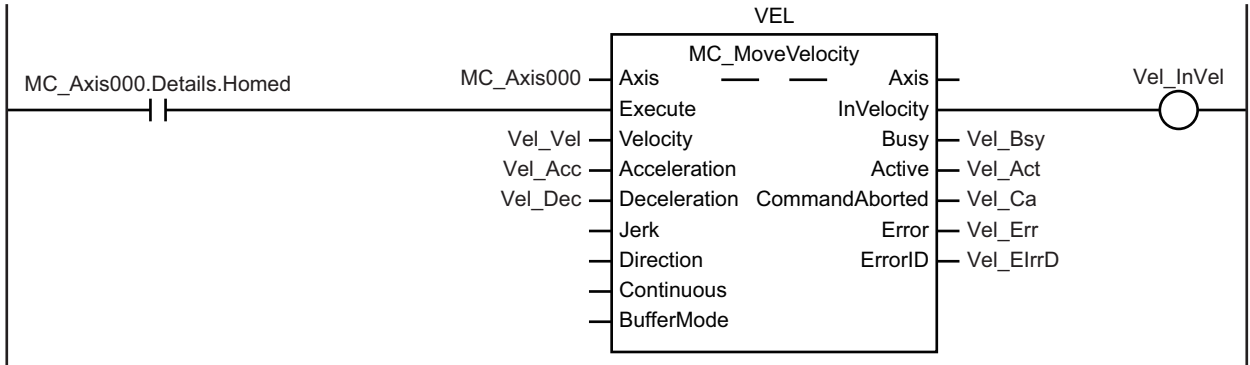
轴 2 为伺服 ON 状态且处于原点未确定状态时，执行原点复位，确定原点



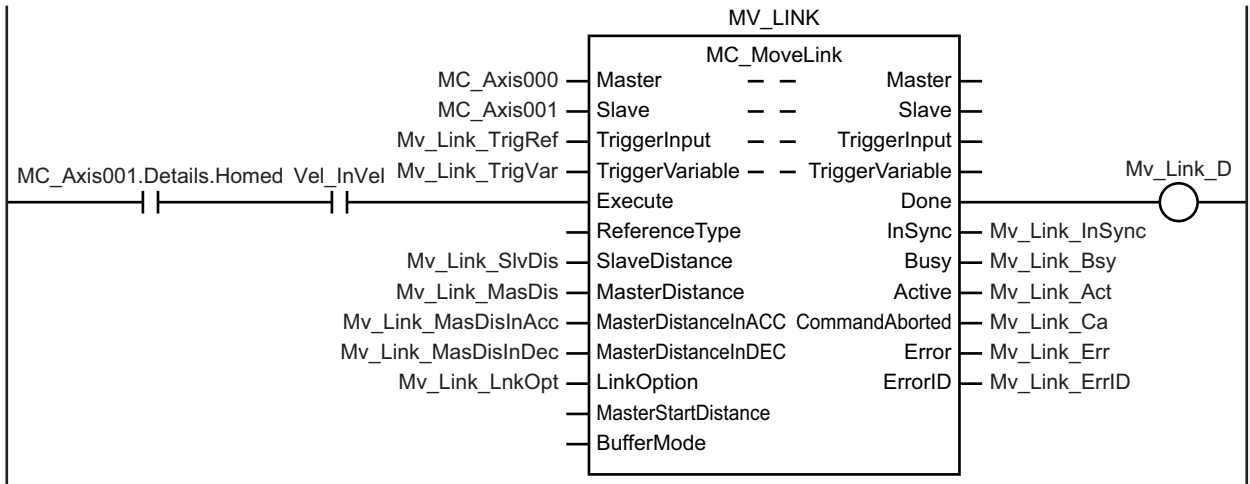
设定 MC_MoveVelocity (速度控制) 指令和 MC_MoveLink (梯形模式凸轮) 指令的参数



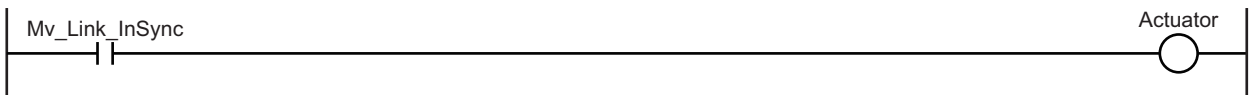
轴 1 变为原点确定状态后, 执行 MC_MoveVelocity (速度控制) 指令



从轴 (轴 2) 变为原点确定状态后, 执行 MC_MoveLink (梯形模式凸轮) 指令



同步过程中将 Actuator 设为 ON



联机 ST 的内容

```
//MC_MoveVelocity 参数
Vel_Vel := LREAL#1000.0;
Vel_Acc := LREAL#0.0;
Vel_Dec := LREAL#0.0;

//MC_MoveLink 参数
Mv_Link_TrigRef.Mode      := _eMC_TRIGGER_MODE#_mcDrive;
Mv_Link_TrigRef.LatchID   := _eMC_TRIGGER_LATCH_ID#_mcLatch1;
Mv_Link_TrigRef.InputDrive := _eMC_TRIGGER_INPUT_DRIVE#_mcEXT;
Mv_Link_TrigVar           := FALSE;
Mv_Link_SlvDis            := LREAL#1000.0;
Mv_Link_MasDis            := LREAL#1000.0;
Mv_Link_MasDisInAcc       := LREAL#100.0;
Mv_Link_MasDisInDec       := LREAL#100.0;
Mv_Link_LnkOpt            := _eMC_LINKOPTION#_mcTriggerDetection;

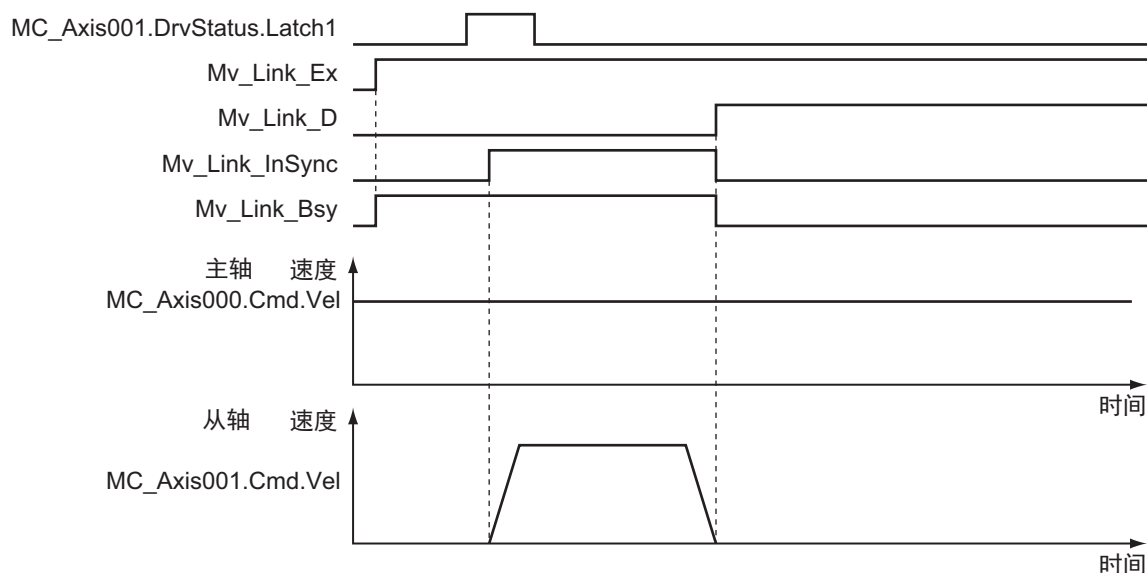
// 设定输入参数后，将 InitFlag 设为 TRUE
InitFlag := TRUE;
```

结构文本 (ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	主轴轴 1 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时, 变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
MC_Axis001	_sAXIS_REF	—	从轴轴 2 的轴变量。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴 2 发生轻度故障等级的异常时, 变为 TRUE。
MC_Axis001.Details.Homed	BOOL	FALSE	轴 2 处于原点确定状态时变为 TRUE。
Pwr1_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR1 的输出变量 Status 中的变量。若变为伺服 ON 状态, 该变量变为 TRUE。
Pwr2_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR2 的输出变量 Status 中的变量。若变为伺服 ON 状态, 该变量变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态, 则变为伺服 ON 状态。
Actuator	BOOL	FALSE	轴 1 和轴 2 同步后, 变为 TRUE。 Actuator 为 TRUE 时, 切刀垂直下降。
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE 时, 设定输入参数。输入参数设定完成后, 变为 TRUE。

● 时序图



● 示例程序

```

// 未设定输入参数时的处理
IF InitFlag=FALSE THEN

    //MC_MoveVelocity 参数
    Vel_Vel := LREAL#1000.0;
    Vel_Acc := LREAL#0.0;
    Vel_Dec := LREAL#0.0;

    //MC_MoveLink 参数
    Mv_Link_TrigRef.Mode      := _eMC_TRIGGER_MODE#_mcDrive;
    Mv_Link_TrigRef.LatchID   := _eMC_TRIGGER_LATCH_ID#_mcLatch1;
    Mv_Link_TrigRef.InputDrive := _eMC_TRIGGER_INPUT_DRIVE#_mcEXT;
    Mv_Link_TrigVar           := FALSE;
    Mv_Link_SlvDis            := LREAL#1000.0;
    Mv_Link_MasDis            := LREAL#1000.0;
    Mv_Link_MasDisInAcc       := LREAL#100.0;
    Mv_Link_MasDisInDec       := LREAL#100.0;
    Mv_Link_LnkOpt            := _eMC_LINKOPTION#_mcTriggerDetection;

    // 设定输入参数后, 将 InitFlag 设为 TRUE
    InitFlag := TRUE;

END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 1 设为伺服 ON 状态。
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
    AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
    AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
    Pwr1_En:=TRUE;    // 将轴 1 设为伺服 ON
ELSE
    Pwr1_En:=FALSE;  // 将轴 1 设为伺服 OFF
END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 2 设为伺服 ON 状态。
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
    AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
    AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
    Pwr2_En:=TRUE;    // 将轴 2 设为伺服 ON
ELSE
    Pwr2_En:=FALSE;  // 将轴 2 设为伺服 OFF
END_IF;

// 发生轻度故障等级异常时的处理
// 发生异常时的处理 (FaultHandler) 请根据装置由客户进行编程。
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;

// 轴 1 为伺服 ON 状态且处于原点未确定状态时, 执行原点复位
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

```

```

// 轴 2 为伺服 ON 状态且处于原点未确定状态时, 执行原点复位
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
  Hm2_Ex:=TRUE;
END_IF;

// 轴 1 变为原点确定后, 在轴 1 上执行 MC_MoveVelocity
IF MC_Axis000.Details.Homed=TRUE THEN
  Vel_Ex:=TRUE;
END_IF;

// 确认轴 2 为原点确定状态, 且轴 1 达到目标速度后, 在轴 2 上执行 MC_MoveLink
IF (MC_Axis001.Details.Homed=TRUE) AND (Vel_InVel=TRUE) THEN
  Mv_Link_Ex:=TRUE;
END_IF;

// 轴 1 和轴 2 同步后, 执行器变为 ON
IF Mv_Link_InSync=TRUE THEN
  Actuator:=TRUE;
ELSE
  Actuator:=FALSE;
END_IF;

// MC_Power1
PWR1(
  Axis      := MC_Axis000,
  Enable    := Pwr1_En,
  Status    => Pwr1_Status,
  Busy      => Pwr1_Bsy,
  Error     => Pwr1_Err,
  ErrorID   => Pwr1_ErrID
);

// MC_Power2
PWR2(
  Axis      := MC_Axis001,
  Enable    := Pwr2_En,
  Status    => Pwr2_Status,
  Busy      => Pwr2_Bsy,
  Error     => Pwr2_Err,
  ErrorID   => Pwr2_ErrID
);

// MC_Home1
HM1(
  Axis              := MC_Axis000,
  Execute           := Hm1_Ex,
  Done              => Hm1_D,
  Busy              => Hm1_Bsy,
  CommandAborted   => Hm1_Ca,
  Error             => Hm1_Err,
  ErrorID           => Hm1_ErrID
);

// MC_Home2
HM2(
  Axis              := MC_Axis001,

```

```

Execute      := Hm2_Ex,
Done         => Hm2_D,
Busy        => Hm2_Bsy,
CommandAborted => Hm2_Ca,
Error       => Hm2_Err,
ErrorID     => Hm2_ErrID
);

// MC_MoveVelocity
VEL(
  Axis          := MC_Axis000,
  Execute       := Vel_Ex,
  Velocity      := Vel_Vel,
  Acceleration  := Vel_Acc,
  Deceleration  := Vel_Dec,
  InVelocity    => Vel_InVel,
  Busy          => Vel_Bsy,
  Active        => Vel_Act,
  CommandAborted => Vel_Ca,
  Error         => Vel_Err,
  ErrorID       => Vel_ErrID
);

// MC_MoveLink
MV_LINK(
  Master          := MC_Axis000,
  Slave           := MC_Axis001,
  TriggerInput    := Mv_Link_TrigRef,
  TriggerVariable := Mv_Link_TrigVar,
  Execute         := Mv_Link_Ex,
  SlaveDistance  := Mv_Link_SlvDis,
  MasterDistance  := Mv_Link_MasDis,
  MasterDistanceInAcc := Mv_Link_MasDisInAcc,
  MasterDistanceInDec := Mv_Link_MasDisInDec,
  LinkOption      := Mv_Link_LnkOpt,
  Done            => Mv_Link_D,
  InSync          => Mv_Link_InSync,
  Busy            => Mv_Link_Bsy,
  Active          => Mv_Link_Act,
  CommandAborted => Mv_Link_Ca,
  Error           => Mv_Link_Err,
  ErrorID         => Mv_Link_ErrID
);

```

MC_CombineAxes

将 2 个轴的位置相加或相减得到的值，作为指令位置输出。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_CombineAxes	加减运算 定位	FB		<pre> MC_CombineAxes_instance (Master := 《参数》, Auxiliary := 《参数》, Slave := 《参数》, Execute := 《参数》, CombineMode := 《参数》, RatioNumeratorMaster := 《参数》, RatioDenominatorMaster := 《参数》, RatioNumeratorAuxiliary := 《参数》, RatioDenominatorAuxiliary := 《参数》, ReferenceTypeMaster := 《参数》, ReferenceTypeAuxiliary := 《参数》, BufferMode := 《参数》, InCombination => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》); </pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
CombineMode	加减法 选择	_eMC _COMBINE_ MODE	0: _mcAddAxes 1: _mcSubAxes	0 ^{*1}	指定合成方法。 0: 加法 1: 减法
Ratio Numerator Master (Reserved)	主轴 齿轮比分子	DINT ^{*2}	正数或负数 ^{*2}	10000	(Reserved)
Ratio Denominator Master (Reserved)	主轴 齿轮比分母	UDINT ^{*3}	正数	10000	(Reserved)
Ratio Numerator Auxiliary (Reserved)	辅助轴 齿轮比分子	DINT ^{*2}	正数或负数 ^{*2}	10000	(Reserved)
Ratio Denominator Auxiliary (Reserved)	辅助轴 齿轮比分母	UDINT ^{*3}	正数	10000	(Reserved)

输入变量	名称	数据类型	有效范围	初始值	内容
Reference TypeMaster	主轴位置种类选择	_eMC_REFERENCE_TYPE	1: _mcFeedback 2: _mcLatestCommand	2 ^{*1}	指定主轴的位置种类。 1: 反馈位置 (同一任务周期 ^{*4} 的获取值) 2: 指令位置 (同一任务周期 ^{*4} 中的计算值)
Reference TypeAuxiliary	辅助轴位置种类选择	_eMC_REFERENCE_TYPE	1: _mcFeedback 2: _mcLatestCommand	2 ^{*1}	指定辅助轴的位置种类。 1: 反馈位置 (同一任务周期 ^{*4} 中的获取值) 2: 指令位置 (同一任务周期 ^{*4} 中的计算值)
BufferMode	缓冲模式选择	_eMC_BUFFER_MODE	0: _mcAborting	0 ^{*1}	指定运动指令多重启动时的动作。 0: 中断

*1. 如果是有效范围为枚举型的变量初始值, 实际不是数值, 而是列举值。

*2. 可在 CPU 单元版本 1.02 以上和 Sysmac Studio Ver.1.03 以上的组合中使用。在更低版本的组合中, 数据类型为“UINT”, 有效范围为“正数”。

*3. 可在 CPU 单元版本 1.02 以上和 Sysmac Studio Ver.1.03 以上的组合中使用。在更低版本的组合中, 数据类型为“UINT”。

*4. 任务为主固定周期任务时, 任务周期为主周期, 为固定周期任务 (执行优先度 5) 时, 任务周期为固定周期任务 (执行优先度 5) 的任务周期。

输出变量

输出变量	名称	数据类型	有效范围	内容
InCombination	加减运算中	BOOL	TRUE、FALSE	加减运算过程中变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时, 输出错误代码。16#0000 表示正常。

* 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
InCombination	开始了加减法定位时	<ul style="list-style-type: none"> Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	开始了指令时	<ul style="list-style-type: none"> Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动 (中断), 本指令中止时 因发生异常, 本指令中止时 正在发生异常时, 启动了本指令 正在执行 MC_Stop 指令时, 启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Master	主轴	_sAXIS_REF	—	指定主轴。 ^{*1}
Auxiliary	辅助轴	_sAXIS_REF	—	指定辅助轴。 ^{*1}
Slave	从轴	_sAXIS_REF	—	指定从轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。



使用注意事项

若主轴、从轴、辅助轴中的任意一个指定为相同的轴，将发生以下轻度故障等级的异常。

- “主轴从轴相同（错误代码：5436 Hex）”异常
- “主轴辅助轴相同（错误代码：5437 Hex）”异常
- “辅助轴从轴相同（错误代码：548E Hex）”异常

功能说明

- 本指令在 Execute（启动）的上升沿开始加减运算定位动作。



使用注意事项

在单元版本 1.09 以下的 CPU 单元中使用本指令时，请勿对 Master（主轴）执行 MC_SetPosition（当前位置变更）。若对 Master（主轴）执行 MC_SetPosition（当前位置变更）指令，Slave（从轴）可能发生剧烈追踪的危险。

不对 Master（主轴）使用 MC_SetPosition（当前位置变更）指令时，请先解除 Master（主轴）和 Slave（从轴）的关系后再执行。

关于主轴的注意事项，请参考 □「同步控制的主轴及辅助轴的注意事项 (P.1-6)」。

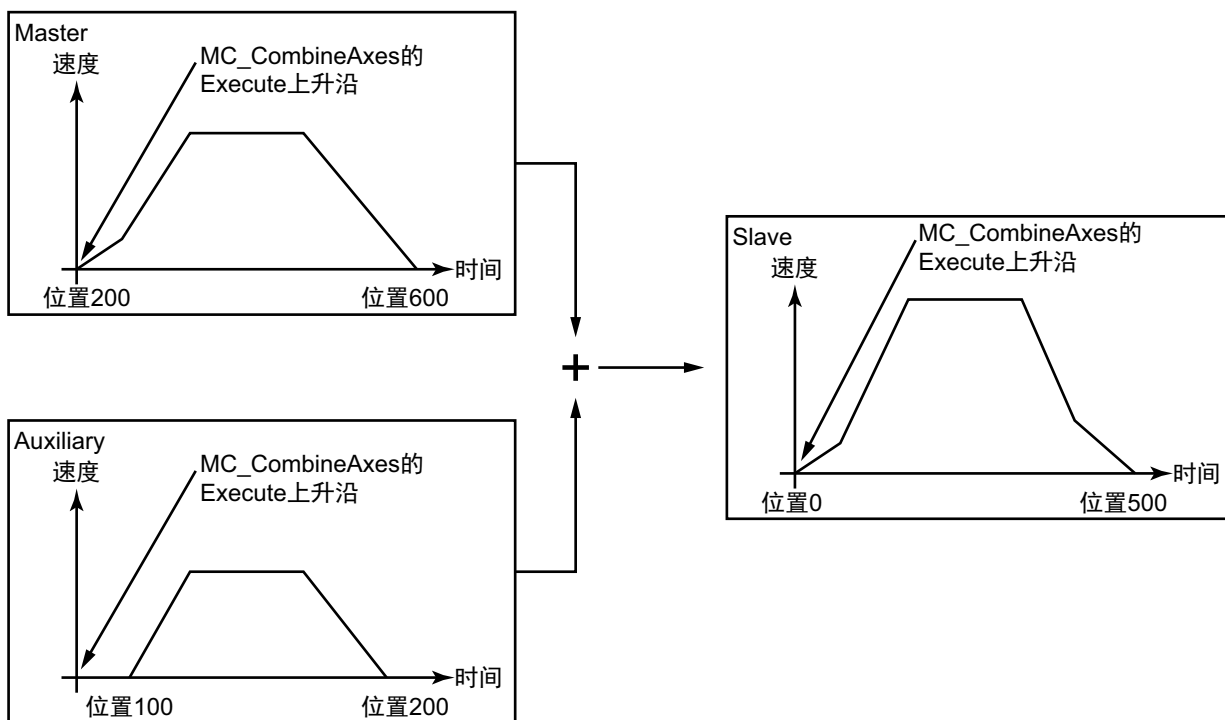
此外，Auxiliary（辅助轴）与 Master（主轴）相同。

指令的详情

将开始时的 Master（主轴）位置相对值和 Auxiliary（辅助轴）位置相对值相加或相减得到的相对值作为 Slave（从轴）的指令位置输出。

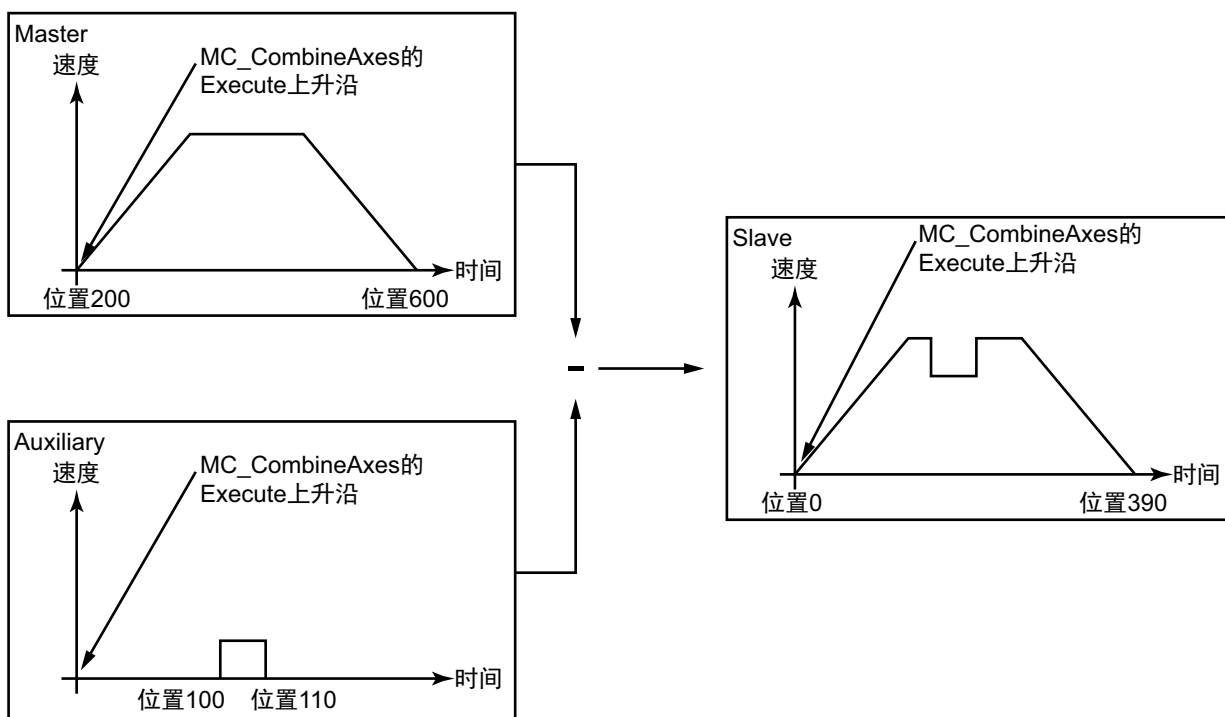
● CombineMode 为 `_mcAddAxes` 时

Slave（从轴）位置 = Master（主轴）位置（相对位置）+ Auxiliary（辅助轴）位置（相对位置）



● CombineMode 为 `_mcSubAxes` 时

Slave（从轴）位置 = Master（主轴）位置（相对位置）- Auxiliary（辅助轴）位置（相对位置）



- 位置的加法和减法与轴参数的轴坐标单位选择无关，始终作为数字进行相加或相减。

- 要结束本指令，请启动 MC_Stop（强制停止）指令。



使用注意事项

受 Master（主轴）和 Auxiliary（辅助轴）的值影响，Slave（从轴）的移动量、速度、加速度可能发生剧烈变化。
使用时请务必注意。

● 位置检查

本指令不进行位置检查。

● 超驰

本指令不可通过 MC_SetOverride（超驰值设定）指令进行超驰。

● ReferenceType（位置种类选择）

从以下项目中选择位置种类。

- `_mcFeedback`: 同一任务周期的获取值
使用同一任务周期内获取的主轴的反馈位置。
- `_mcLatestCommand`: 指令位置（同一任务周期中的计算值）
使用同一任务周期中计算得到的主轴指令位置。
选择了 `_mcLatestCommand` 时，主轴编号必须设定为比辅助轴编号更新的编号。
如果从轴的轴编号比主轴或辅助轴新，本指令的 Error（错误）将变为 TRUE。ErrorID（错误代码）中将输出“主轴 / 从轴 轴编号非升序（错误代码：5438 Hex）”。
此外，主轴和辅助轴之间没有轴编号相关的限制。



使用注意事项

任务为主固定周期任务时，这里的任务周期为主周期，为固定周期任务（执行优先度 5）时，任务周期为固定周期任务（执行优先度 5）的任务周期。同样，固定周期任务表示主固定周期任务或固定周期任务（执行优先度 5）。

● 轴种类和位置种类的关系

可监视的轴种类和要监视的位置种类的关系如下所示。

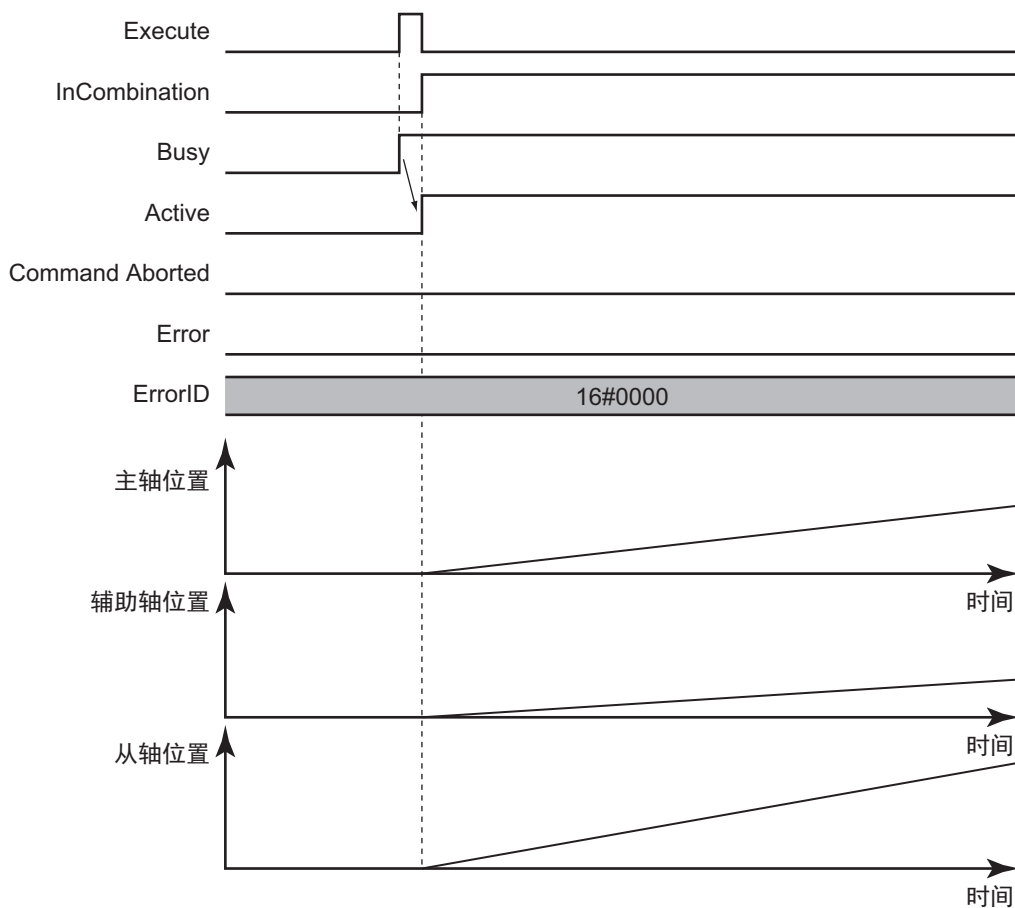
轴种类	ReferenceType	
	<code>_mcFeedback</code>	<code>_mcLatestCommand</code>
伺服轴	○	○
编码器轴	○	x ^{*1}
虚拟伺服轴	○	○
虚拟编码器轴	○	x ^{*1}

*1. 指令启动时，将发生“超出位置种类选择范围（错误代码：5430 Hex）”。

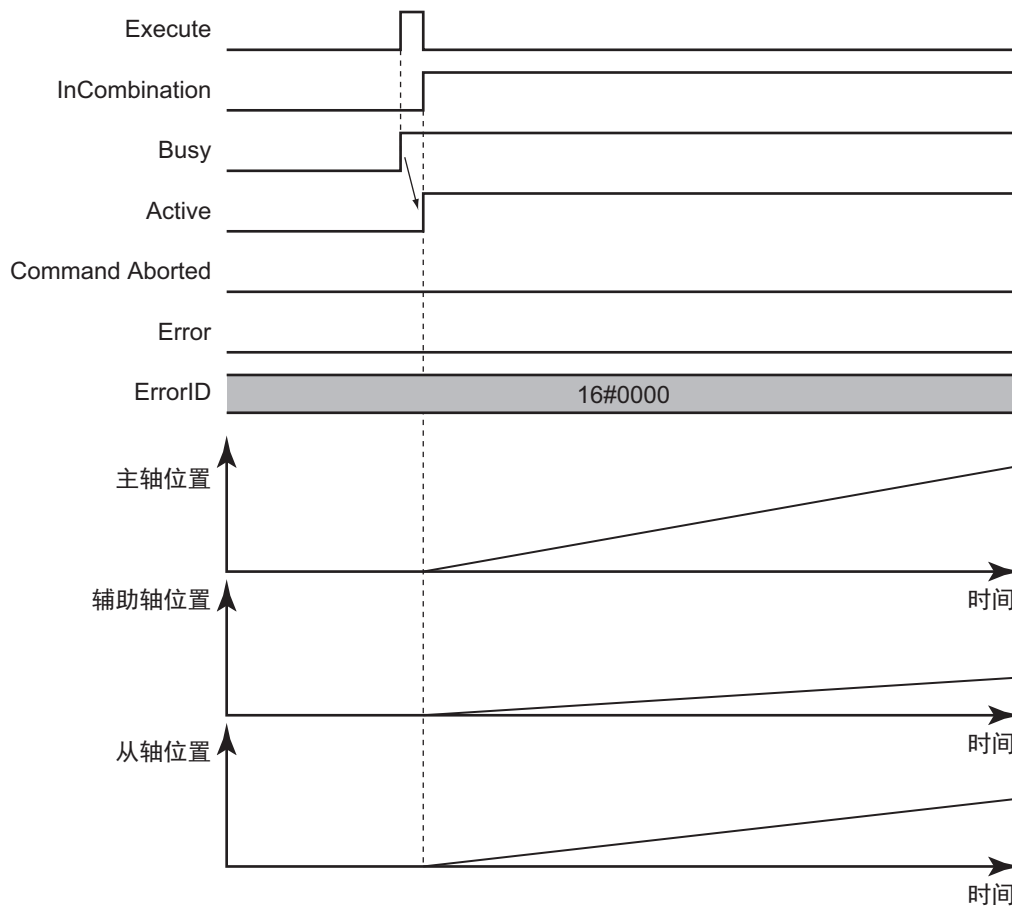
时序图

- 在 Execute (启动) 上升的同时 Busy (执行中) 变为 TRUE, 在下一周期, Active (控制中) 变为 TRUE。
- 在开始加减法输出的周期, InCombination (加减运算中) 变为 TRUE。
- 通过其他指令中断了本指令时, CommandAborted (执行中断) 变为 TRUE, Busy (执行中)、Active (控制中)、InCombination (加减运算中) 变为 FALSE。

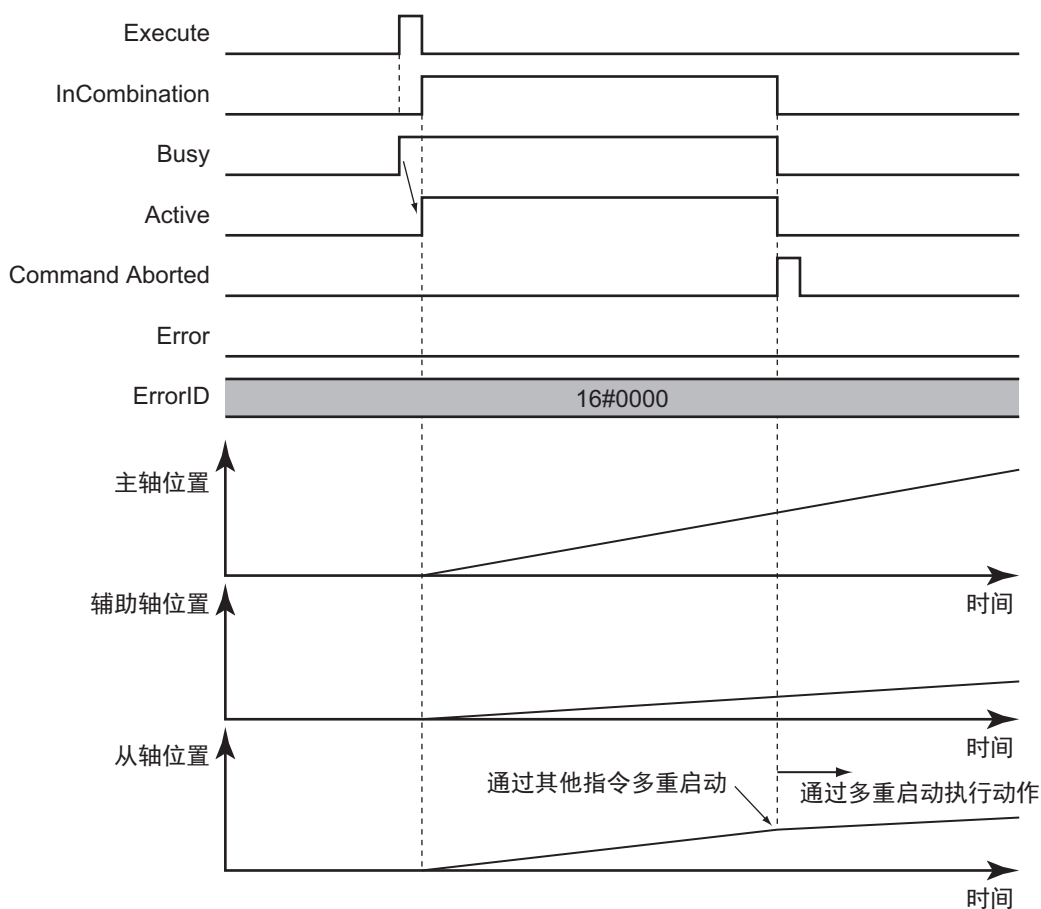
● CombineMode 为 _mcAddAxes 时



● CombineMode 为 _mcSubAxes 时



● 中断时



运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

运动指令多重启动的详情请参考 [□](#) 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 本指令执行过程中其他指令的启动

对本指令执行指令多重启动时，请指定从轴。

在执行本指令的过程中多重启动其他指令时，根据 BufferMode（缓冲模式选择），有以下限制。

- 执行本指令的过程中，通过其他指令多重启动时，可进行中断。
不可通过缓冲、共混进行指令多重启动。

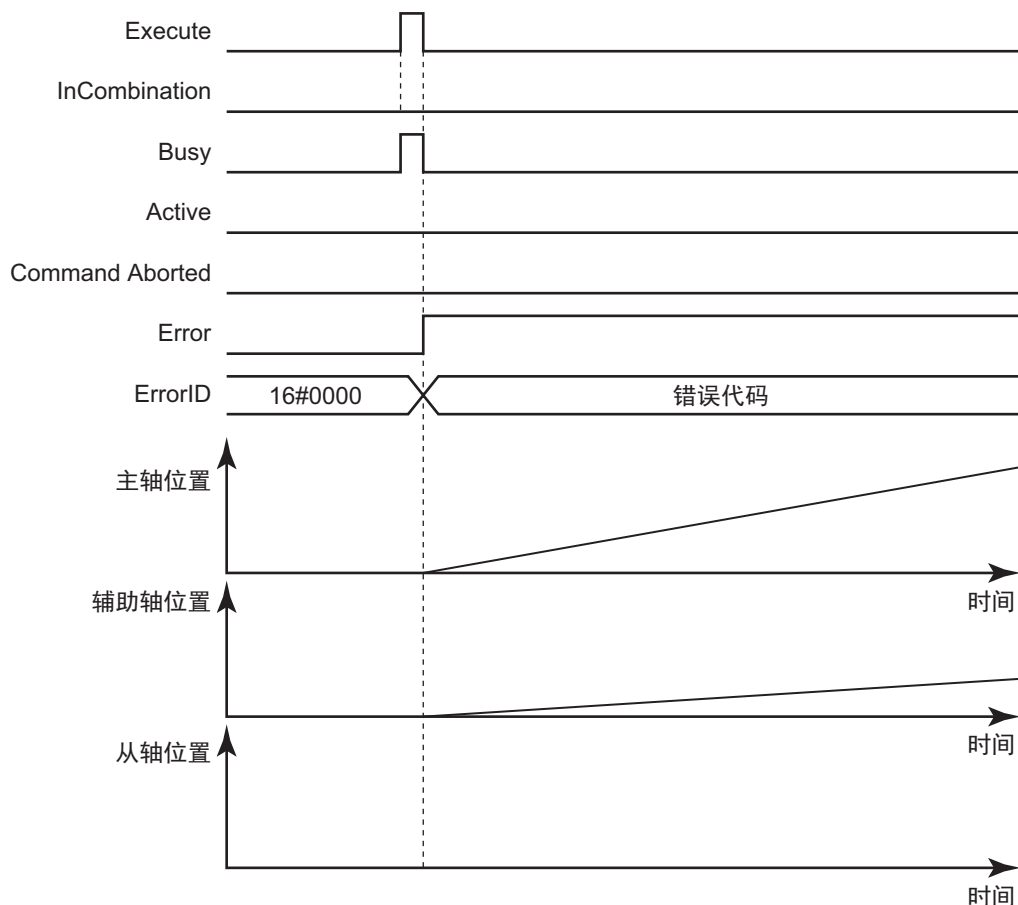
异常

● 发生异常时的时序图

启动本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。

减速停止时的减速度为轴参数中的最大减速度。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。



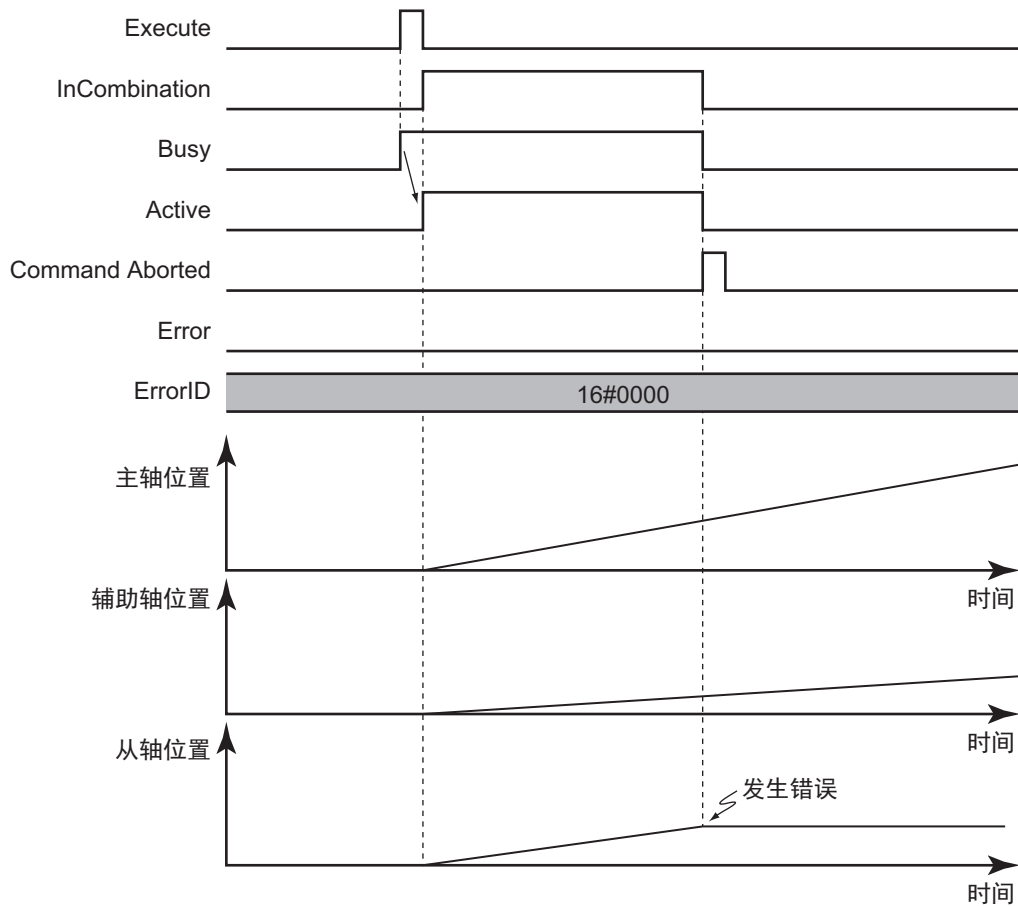
参考

- 本指令不会受 Master（主轴）、Auxiliary（辅助轴）的异常状态影响。
- 解除异常状态，Master（主轴）、Auxiliary（辅助轴）正在动作时，Slave（从轴）继续执行加减法定位动作。此外，本指令启动时以及本指令执行过程中，如果从轴发生异常，本指令将中断，但不会对 Master（主轴）、Auxiliary（辅助轴）造成影响。

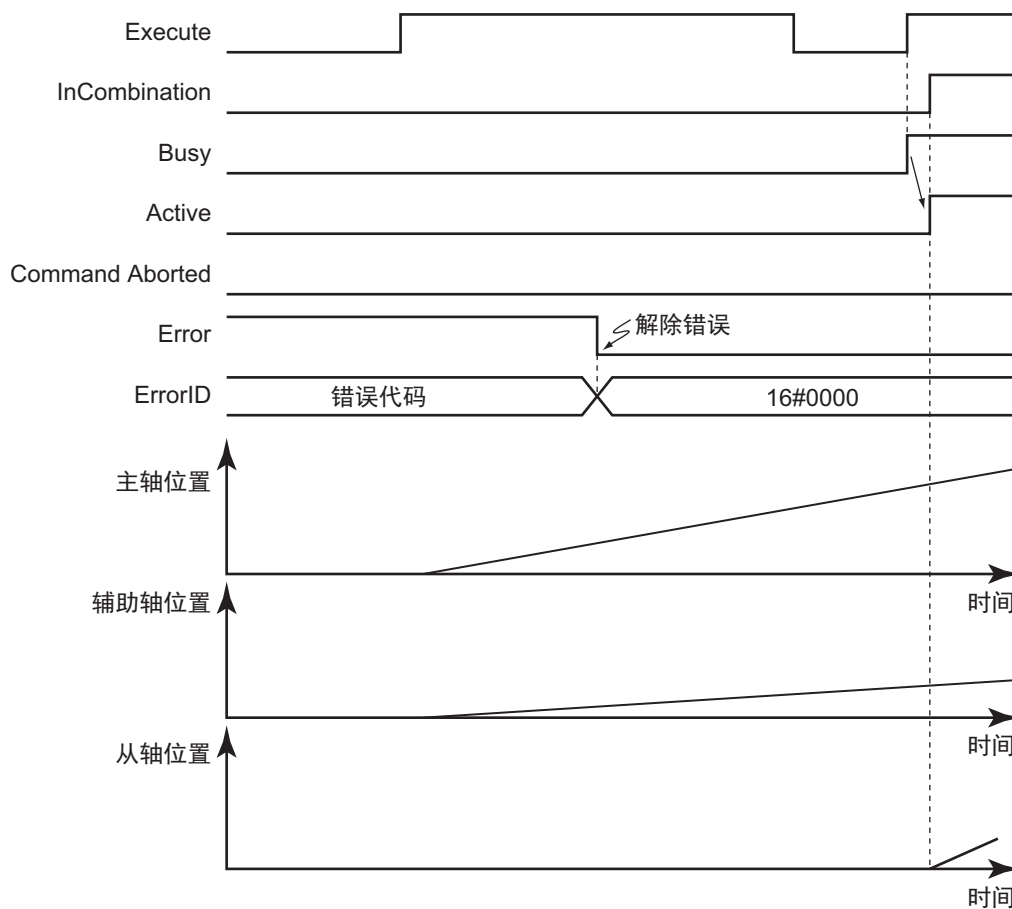
执行本指令的过程中发生轻度故障等级的异常时，CommandAborted（执行中断）将变为 TRUE，轴停止。

减速停止时的减速度为轴参数中的最大减速度。

参照输出到 Slave（从轴）的轴变量 MFalutLvl.Code 的值，可找出异常的原因。



解除本指令的异常后，在下一 Execute（启动）上升前，本指令不会启动。



● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

MC_Phasing

执行同步控制中的主轴相位补偿。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_Phasing	主轴相对值 相位补偿	FB		<pre>MC_Phasing_instance (Master := 《参数》, Slave := 《参数》, Execute := 《参数》, PhaseShift := 《参数》, Velocity := 《参数》, Acceleration := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, BufferMode := 《参数》, Done => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
PhaseShift	相位补偿量	LREAL	负数、正数、“0”	0	指定主轴的相位补偿量。 单位为 [指令单位]。 ^{*1}
Velocity	目标速度	LREAL	正数	0	指定相位补偿量的目标速度（补偿速度）。 ^{*2} 单位为 [指令单位 /s]。 ^{*1}
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度。 单位为 [指令单位 /s ²]。 ^{*1}
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。 单位为 [指令单位 /s ²]。 ^{*1}
Jerk	跃度	LREAL	正数或“0”	0	指定跃度。 单位为 [指令单位 /s ³]。 ^{*1}
BufferMode	缓冲 模式选择	_eMC_ BUFFER_ MODE	0: _mcAborting	0 ^{*3}	指定运动指令多重启动时的动作。 0: 中断

*1. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》的“单位转换设定”。

*2. 请务必设定目标速度。若不设定而直接动作，将发生超出目标速度范围（错误代码：5422Hex）。

*3. 如果是有效范围为列举型的变量初始值，实际不是数值，而是列举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

*  请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	相位补偿完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	开始了相位补偿时	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动（中断），本指令中止时 因发生异常，本指令中止时 正在发生异常时，启动了本指令 正在执行 MC_Stop 指令时，启动了本指令 正在执行的同步控制指令完成时 	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Master	主轴	_sAXIS_REF	—	指定主轴。 ^{*1}
Slave	从轴	_sAXIS_REF	—	指定从轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis***”）或系统定义变量的轴变量名称（_MC_AX[*]、_MC1_AX[*]、_MC2_AX[*]）。

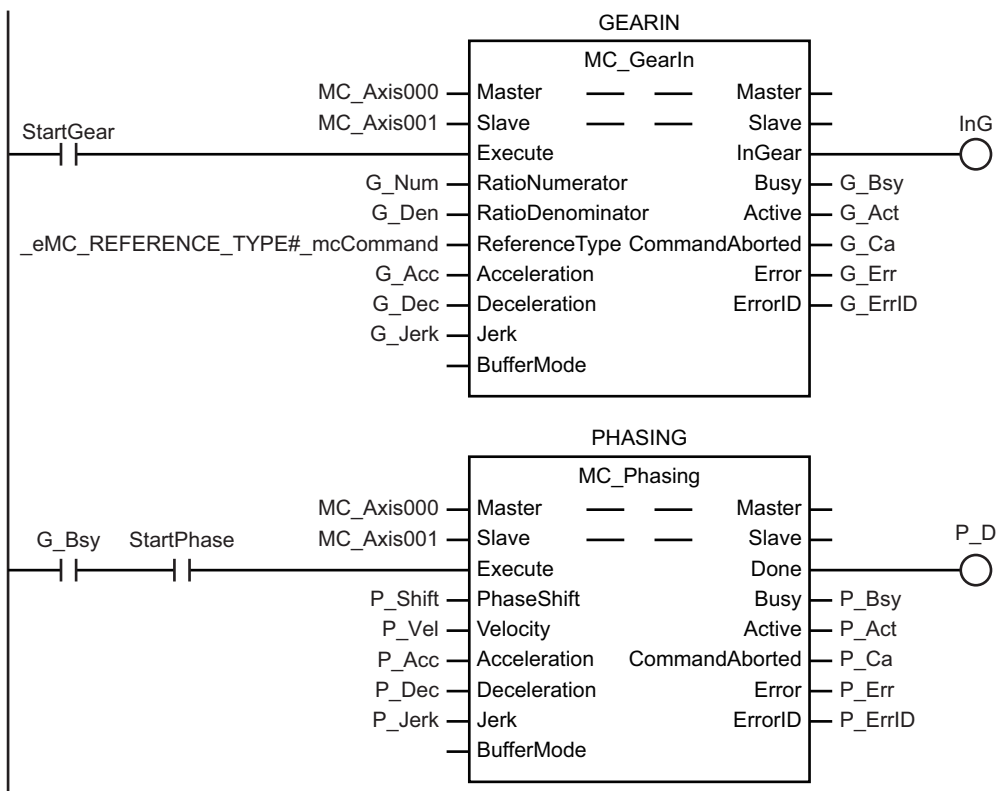


使用注意事项

若主轴和从轴指定为相同的轴，将发生轻度故障等级的“主轴从轴相同（错误代码：5436 Hex）”异常。

功能说明

- 除正在执行 MC_CombineAxes（加减法定位）指令外，若在单轴同步控制中启动本指令，将根据设定的 PhaseShift（相位补偿量）、Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）补偿主轴的相位。
- 主轴的指令当前位置及反馈当前位置无变化，以相对量对主轴的指令当前位置及反馈当前位置进行补偿后得到的值即为“主轴的相位”。
从轴与补偿后的“主轴相位”同步。
- 达到 PhaseShift（相位补偿量）后，Done（完成）变为 TRUE。
- 正在执行的同步控制指令完成后，补偿结束。再次执行了同步控制指令时，不会影响之前的补偿量。
- 主轴相位补偿为有效的同步控制指令包括 MC_CamIn（开始凸轮动作）指令、MC_GearIn（开始齿轮动作）指令、MC_GearInPos（位置指定齿轮动作）指令、MC_MoveLink（梯模式凸轮）指令 4 个。
- 在用户程序中记载本指令时，请如下图所示，在同步控制指令后面记述 MC_Phasing（主轴相对值相位补偿）指令。



使用注意事项

关于主轴的注意事项，请参考 □□「同步控制的主轴及辅助轴的注意事项 (P.1-6)」。

指令的详情

下面介绍指令的详情。

● Master（主轴）、Slave（从轴）的指定

Master（主轴）和 Slave（从轴）用轴变量名称指定轴。

若指定为不是正在执行同步控制指令的 Master（主轴）、Slave（从轴）组合，将发生轴指定异常。

● PhaseShift（相位补偿量）

在 PhaseShift（相位补偿量）中指定与 Slave（从轴）相对的 Master（主轴）的相位补偿量。

相位补偿量以相对量指定。

● Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）、Jerk（跃度）

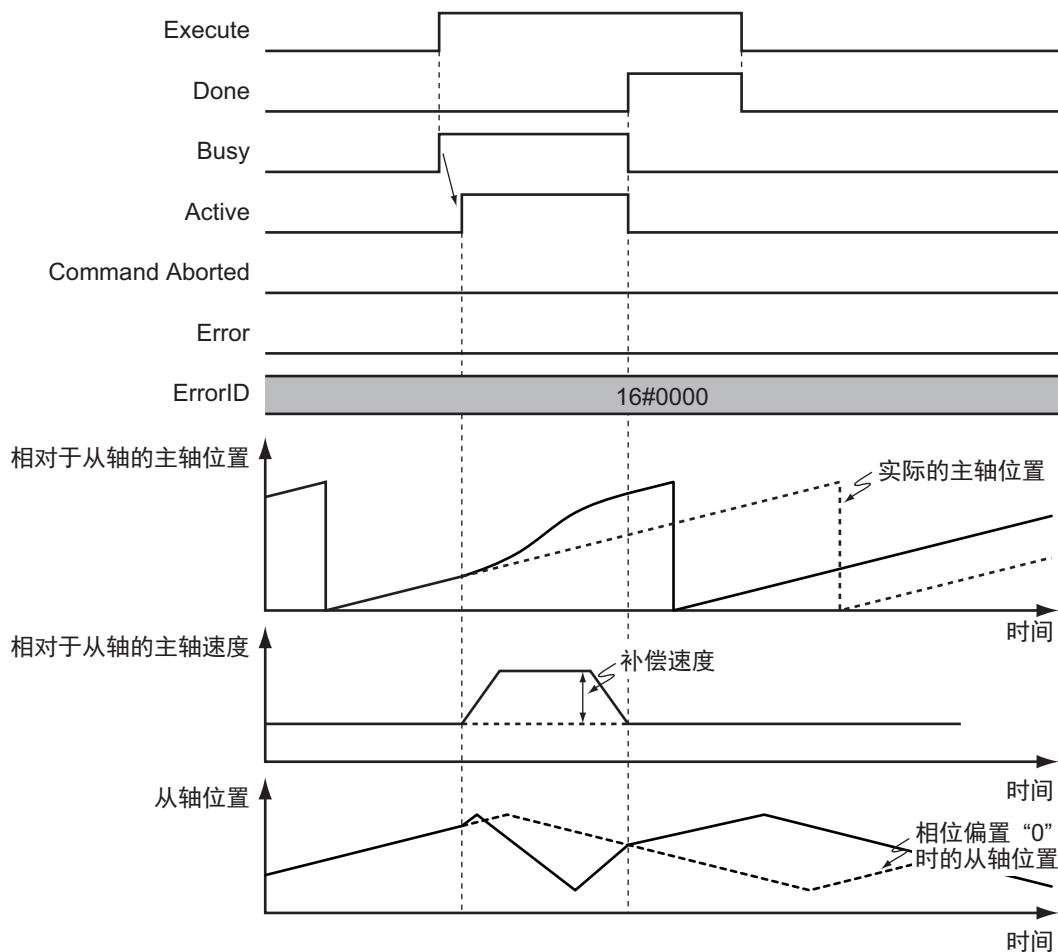
相位补偿量的目标速度（补偿速度）、加速度、减速度、跃度在 Velocity、Acceleration、Deceleration、Jerk 中指定。

与 Slave（从轴）相对的 Master（主轴）目标速度（补偿速度）为主轴速度的相对速度。

与 Slave（从轴）相对的 Master（主轴）目标速度（补偿速度）因相位补偿量、加速度、减速度、跃度不同，如下所示。

例) 电子凸轮的主轴相对值相位补偿时

与 Slave（从轴）相对的 Master（主轴）补偿速度为主轴速度的相对速度。



- 若相位补偿量指定为“0”，则 Master（主轴）的相位补偿量视为“0”，正常结束。
- 若目标速度（补偿速度）指定为“0”，将视为超出范围，Slave（从轴）发生轴异常。
- 指定的目标速度（补偿速度）和主轴速度的相加速度可以指定为超出 Master（主轴）最高速度的值。



参考

执行 MC_Phasing（主轴相对值相位补偿）指令后，已同步的 Slave（从轴）将进行异常检测。

因此，启动指令时，不会对本指令的 Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）设定进行异常检测。

● BufferMode（缓冲模式选择）

指定前一轴动作和本次动作的连接方法。

有以下 1 种。

缓冲模式选择	说明
中断	若重新执行本指令，将立即通过本指令开始补偿。

通过多重启动执行的反转时动作遵照主轴中设定的 [反转时动作选择]。

BufferMode（缓冲模式选择）请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 位置检查

补偿值输出结束，则表示相位补偿动作完成，不进行位置检查。

运动指令重新执行

正在执行本指令时，如果再次启动指令，将变更 PhaseShift（相位补偿量）、Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）。

变更动作与相对定位相同。

运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 本指令执行过程中其他指令的启动

在 Slave（从轴）上执行本指令的过程中，通过其他指令多重启动时，可进行中断。

不可通过缓冲、共混进行指令多重启动。

● 本指令的多重启动

指定的从轴可在执行 MC_Phasing（主轴相对值相位补偿）指令的过程中，启动 MC_Phasing（主轴相对值相位补偿）指令。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，从轴停止。

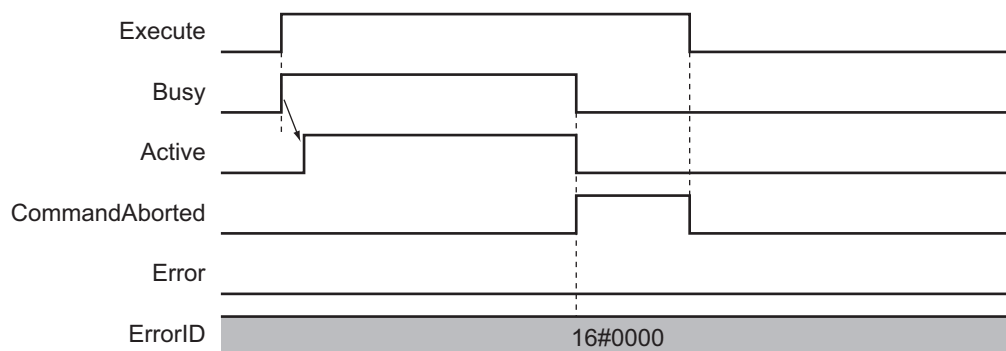
参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

Slave（从轴）因发生异常而变为同步状态解除时，MC_Phasing（主轴相对值相位补偿）指令动作将变为 CommandAborted（执行中断）。

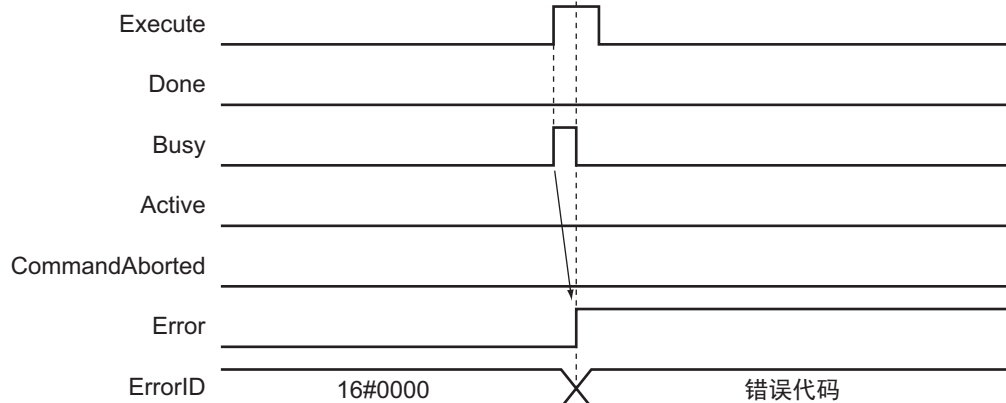
此外，主轴的异常状态不会影响本指令的动作。

● 发生异常时的时序图

MC_CamIn（凸轮动作开始）指令



MC_Phasing（主轴相对值相位补偿）指令



● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

MC_TorqueControl

使用伺服驱动器的扭矩控制模式进行扭矩控制。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_TorqueControl	扭矩控制	FB		<pre>MC_TorqueControl_instance (Axis := 《参数》, Execute := 《参数》, Torque := 《参数》, TorqueRamp := 《参数》, Velocity := 《参数》, Direction := 《参数》, BufferMode := 《参数》, InTorque => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Torque	目标扭矩	LREAL	0 ~ 1000.0	300.0	以 [0.1%] 为单位指定输出到伺服驱动器的目标扭矩。将额定扭矩视为“100.0%”，按比例指定。 ^{*1} 单位为 [%]。
TorqueRamp	扭矩指示灯	LREAL	正数或“0”	0	指定从当前扭矩到输出目标扭矩为止的扭矩变化率。单位为 [%/s]。
Velocity	限制速度	LREAL	正数或“0”	0	指定目标速度。单位为 [指令单位/s]。 ^{*2}
Direction	方向选择	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection	0 ^{*3}	指定目标扭矩的方向。 0: 正方向指定 2: 负方向指定
BufferMode	缓冲模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0 ^{*3}	指定运动指令多重启动时的动作。 0: 中断 1: 缓冲

*1. 指定为超出“1000.0%”的值时，视为“1000.0%”。此外，指定为负值时，视为“0.0%”。

*2. 指令单位请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》的“单位转换设定”。

*3. 如果是有效范围为列举型的变量初始值，实际不是数值，而是列举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
InTorque	达到目标扭矩	BOOL	TRUE、FALSE	达到目标扭矩时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
InTorque	正在输出目标扭矩	<ul style="list-style-type: none"> • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时 • 指令重新执行时变更了目标扭矩时
Busy	Execute 的上升沿	<ul style="list-style-type: none"> • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时
Active	开始了扭矩指令值的输出时	<ul style="list-style-type: none"> • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> • 由其他指令进行运动指令多重启动（中断），本指令中止时 • 因发生异常，本指令中止时 • 正在发生异常时，启动了本指令 • 正在执行 MC_Stop 指令时，启动了本指令 	<ul style="list-style-type: none"> • Execute 为 TRUE 时，与 Execute 的 FALSE 同时 • Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis***”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

功能说明

- 直接指定扭矩指令值，控制伺服电机的输出扭矩。
- 以 [0.1%] 为单位指定目标扭矩。指定了小数点第 2 位以下时，小数点第 2 位四舍五入。
- 若要停止本指令的执行，请使用 MC_Stop（强制停止）指令。
- 扭矩控制利用伺服驱动器的扭矩控制模式功能实现。
- 切换控制模式之前，保持之前的控制模式。

例

位置控制切换为扭矩控制时：伺服驱动器变为扭矩控制前，保持位置控制

扭矩控制切换为位置控制时：伺服驱动器变为位置控制前，保持扭矩控制

● 数据对象的映射

使用 MC_TorqueControl（扭矩控制）指令时，请在 Sysmac Studio 的轴基本设定的 [详细设定] 中，映射以下对象数据。

- 目标扭矩（6071Hex）
- 操作模式（6060Hex）
- 反馈扭矩（6077Hex）
- 操作模式显示（6061Hex）

未设定以上内容时，将发生过程数据对象设定不足（错误代码：3461Hex）。

数据对象的映射请参考 □□「2-3 PDO 映射 (P.2-30)」，□□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。



参考

- NX 系列脉冲输出单元中无法使用本指令。

指令的详情

下面介绍本指令的详情。

● Axis（轴）指定

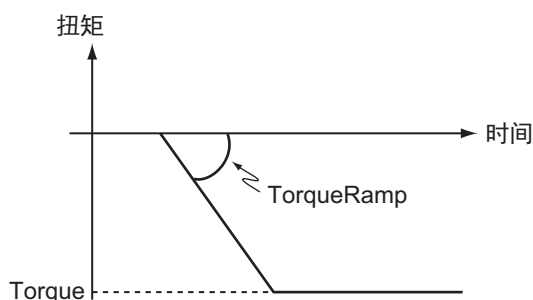
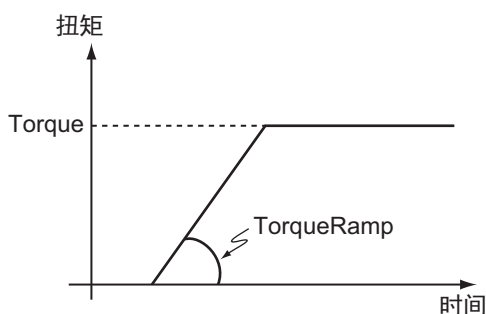
在 Axis（轴）中指定要进行扭矩控制的轴。

● TorqueRamp（扭矩斜率）

指定从当前指定的指令扭矩到输出目标扭矩为止的斜率。

例 1) 方向指定 = 正方向时

例 2) 方向指定 = 负方向时



**使用注意事项**


目标扭矩请设定为小于电机最大扭矩的值。
设定为超出电机最大扭矩时的动作取决于伺服驱动器。

● Velocity（限制速度）

对扭矩限制时的轴最大速度进行限制。

轴速度达到该限制速度值时，伺服驱动器将减小扭矩，降低轴的速度。

限制速度值的功能利用伺服驱动器的功能实现。

详情请参考  《各伺服驱动器手册的扭矩限制功能》。

**使用注意事项**

- 扭矩限制时，轴的速度会变快。为确保安全，请务必设定 Velocity（速度限制）。
- 使用欧姆龙生产的伺服驱动器 G5 系列时，请务必将伺服驱动器的“速度限制选择（3317 Hex）”设定为“1：EtherCAT 通信的速度限制”。
若未设定，速度限制值将不起作用。此外，即使极限输入信号变为 ON，轴也不会停止。
- 速度限制值使用过程数据 607F Hex。
使用欧姆龙生产的伺服驱动器 G5 系列时，请在 Sysmac Studio 的轴参数设定画面的“详细设定”中设定为使用速度限制值（607F Hex）。
此外，在其他公司的伺服驱动器上使用速度限制功能时，请参考其他公司伺服驱动器的手册。

● Direction（方向选择）

指定输出目标扭矩的方向。


要向轴的正方向输出时，指定为正方向，要向轴的负方向输出时，指定为负方向。

● BufferMode（缓冲模式选择）

指定前一轴动作和本次动作的连接方法。

有以下 2 种。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。
缓冲	当前正在执行的指令正常完成后，自动启动已缓冲的本指令。

BufferMode（缓冲模式选择）请参考  《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 扭矩控制中的轴停止

使用欧姆龙生产的伺服驱动器 G5 系列时，若在执行本指令的过程中启动了 MC_Stop（强制停止）指令，将不会应用 MC_Stop（强制停止）指令的减速度，而是立即停止。

此外，即使发生减速停止处理异常，也作为立即停止处理。

● 扭矩控制中的指令当前位置和反馈当前位置

正在利用本指令进行扭矩控制时，可通过运动控制系统变量获取的当前位置如下所示。

- 反馈当前位置 : 将从伺服驱动器获取的值按齿轮比转换得到的值。
- 指令当前位置 : 与 1 个周期前的反馈当前位置相同的值。

● 可使用的轴和启动条件

- 在伺服轴上，MC_Power（可运行）指令的Enable（有效）为TRUE（伺服ON状态）时，可启动本指令。
- 虚拟伺服轴始终可接收。但是，不可执行伺服驱动器的控制模式切换处理。
- 编码器轴、虚拟编码器轴中，启动时会发生异常。

● 伺服 OFF 时的应对

MC_Power（可运行）的输出变量“Status（可运行）”变为FALSE时，MC功能模块将执行切换为CSP模式的处理。

但是，欧姆龙生产的伺服驱动器G5系列在伺服OFF状态下，不会接收来自MC功能模块的控制模式切换。

● 轴变量的状态

轴变量的状态Status.Continuous（连续动作中）变为TRUE。

同时，轴变量DrvStatus（伺服驱动器状态）的CST（周期性同步扭矩（CST）模式中）变为TRUE。

● 原点状态

保持原点确定状态。

● 软件限位

应用软件限位。

即使在轴参数中选择了[对指令位置有效。减速停止]或[对指令位置有效。滞留脉冲停止]，仍会应用。

● 计数模式为[直线模式]时

溢出或下溢时，将视为没有目标位置的動作。

● 反转时动作

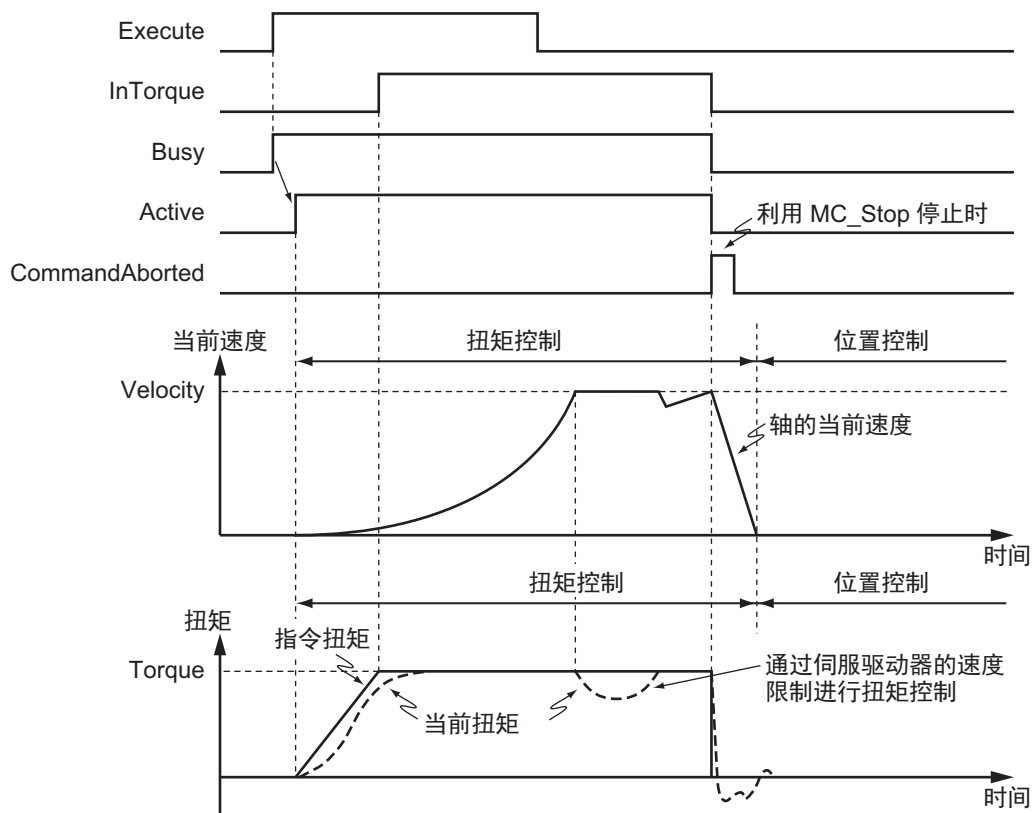
进行多重启动时如果扭矩指令值反转，无论轴参数[反转时动作选择]的设定内容如何，都以本指令的TorqueRamp（扭矩斜率）动作。

此外，通过多重启动反转时的动作如下所示。

- 本指令动作过程中，通过多重启动使用CSP的指令，指令位置发生反转时，将按照轴参数[反转时动作选择]执行反转动作。
- 使用CSP或CSV的指令正在动作时，通过多重启动本指令使扭矩指令值发生反转时，将按照TorqueRamp（扭矩斜率）反转扭矩指令。
- 本指令正在动作时，通过多重启动本指令使扭矩指令值发生反转时，将按照TorqueRamp（扭矩斜率）反转扭矩指令。

时序图

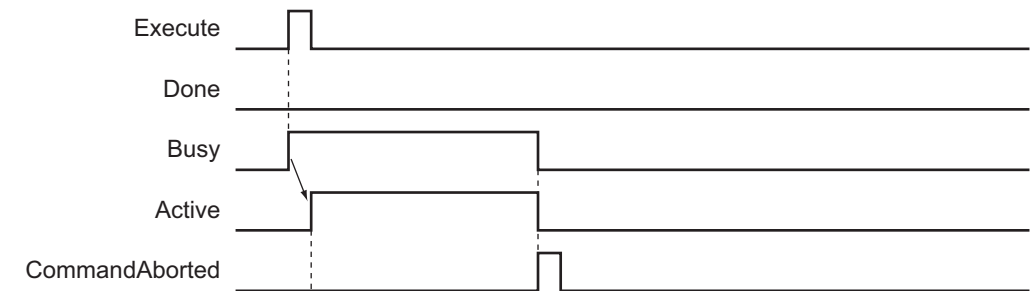
启动本指令并停止时



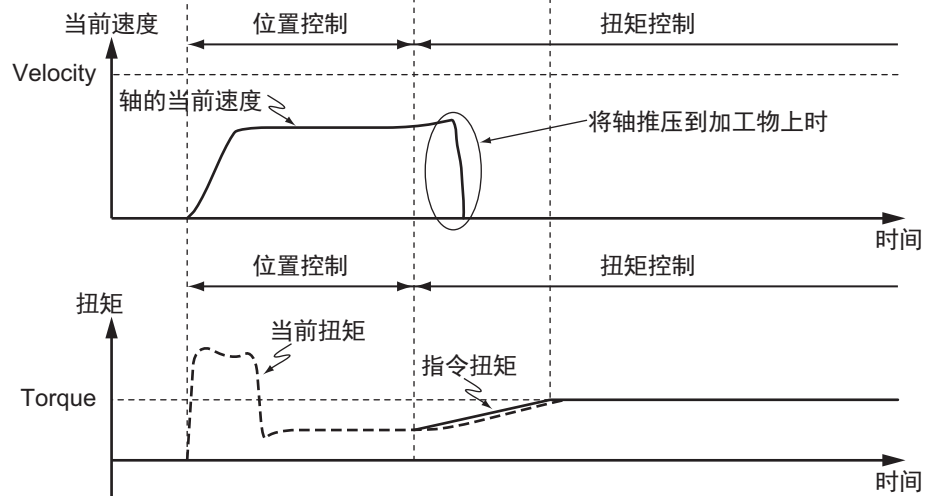
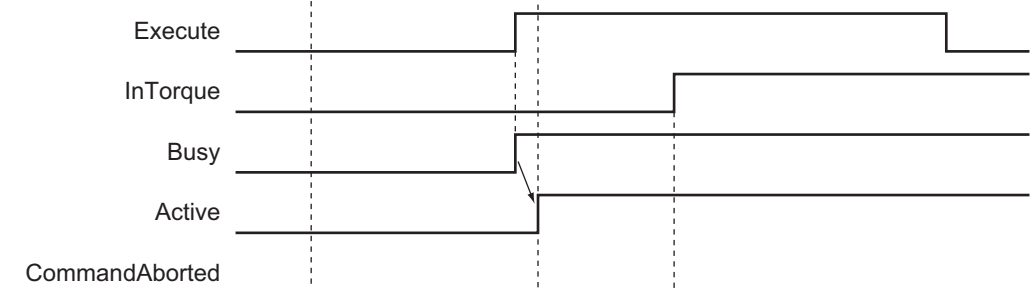
在前一动作过程中中止再启动时

下面以本指令执行过程中，轴发生停止的推压应用为例进行说明。

前一位置控制指令



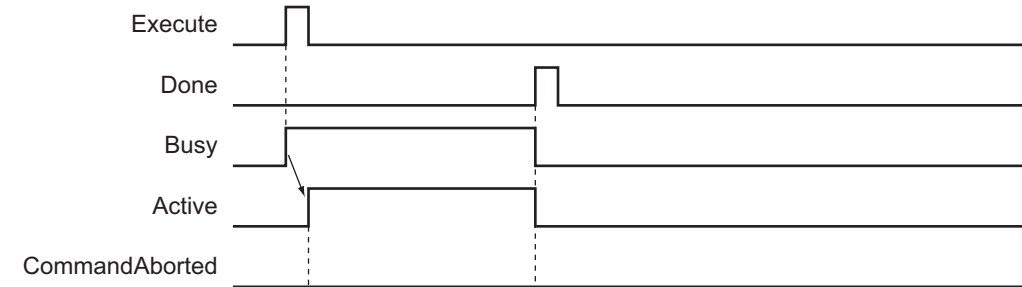
本次指令



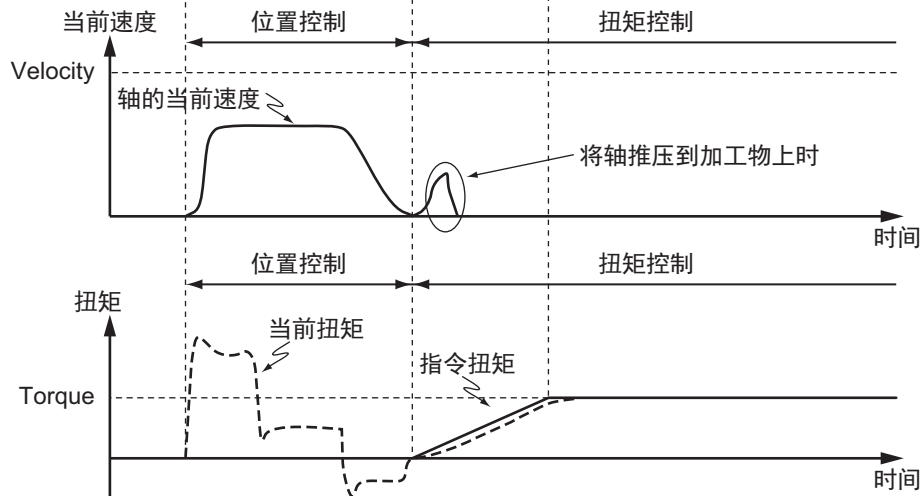
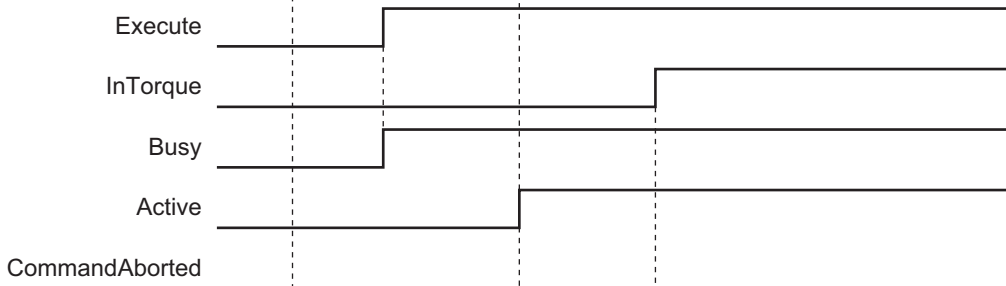
在前一动作过程中缓冲再启动时

下面以本指令执行过程中，轴发生停止的推压应用为例进行说明。

前一位置控制指令



本次指令



● 控制模式的切换

- 正在执行 MC_MoveAbsolute（绝对值定位）指令、MC_MoveRelative（相对值定位）指令等位置控制指令时，如果执行 MC_TorqueControl（扭矩控制）指令，动作将根据 BufferMode（缓冲模式选择）的设定而变化。
BufferMode（缓冲模式选择）为中断时，在执行之时切换为扭矩控制。缓冲时，在前一动作完成之时，切换为扭矩控制。
- 正在执行 MC_TorqueControl（扭矩控制）指令时，如果用 MC_MoveAbsolute（绝对值定位）指令等其他指令中止或轴发生异常，将立即切换为位置控制。
- 执行指令时 Active（控制中）会切换，但伺服驱动器侧的控制模式发生切换需要几个周期。切换为控制模式所需的时间取决于伺服驱动器。

控制模式切换判定

用欧姆龙生产的伺服驱动器 G5 系列停止轴时，MC 功能模块将速度限制值（607F Hex）视为“0”。然后，在固定周期任务中，反馈当前速度连续 3 次满足以下判定式时，将切换为 CSP。

$$\text{反馈当前速度} \leq (\text{最高速度}) \times 0.1$$

如果是其他公司的伺服驱动器，在伺服驱动器的控制模式从 CST 切换为 CSP 时，将同时在切换处理时的反馈当前位置进行伺服锁定。



使用注意事项

这里的固定周期任务表示主固定周期任务或固定周期任务（执行优先级 5）。

控制模式切换失败

MC 功能模块对伺服驱动器发出控制模式切换指令并经过 1 秒后，如果伺服驱动器仍未完成切换，将发生“伺服驱动器控制模式切换异常（错误代码：7439 Hex）”，变为伺服 OFF（自由运行停止）。

“伺服驱动器控制模式切换异常（错误代码：7439 Hex）”请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

此外，速度限制值设为“0”并经过 10 秒后，如果切换判定式仍未成立，与上述一样，将变为伺服 OFF。

控制模式切换动作示例

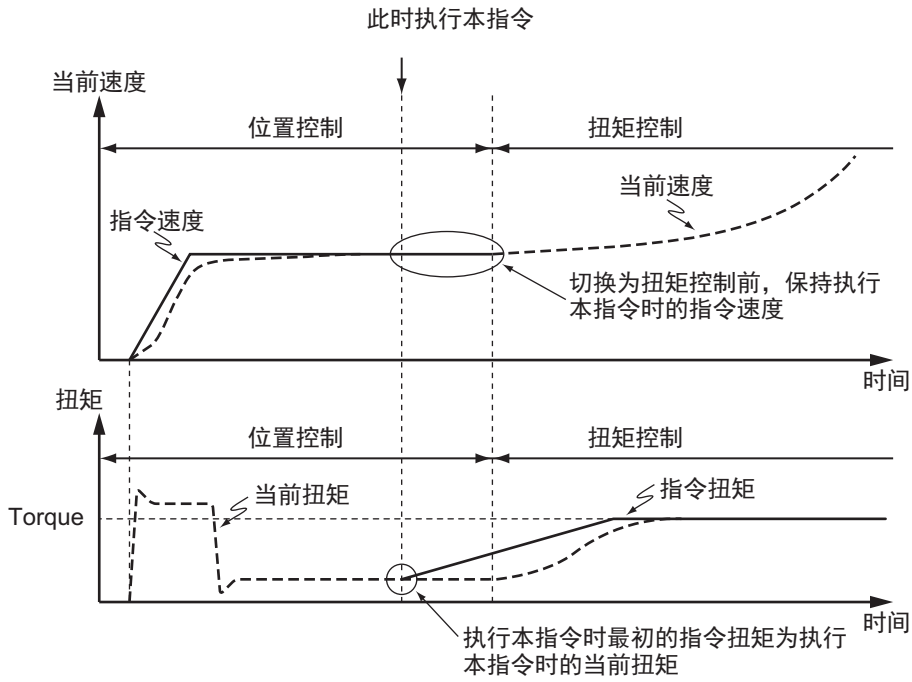
下面以轴动作过程中切换时为例，介绍控制模式切换前指令扭矩和指令速度的关系。



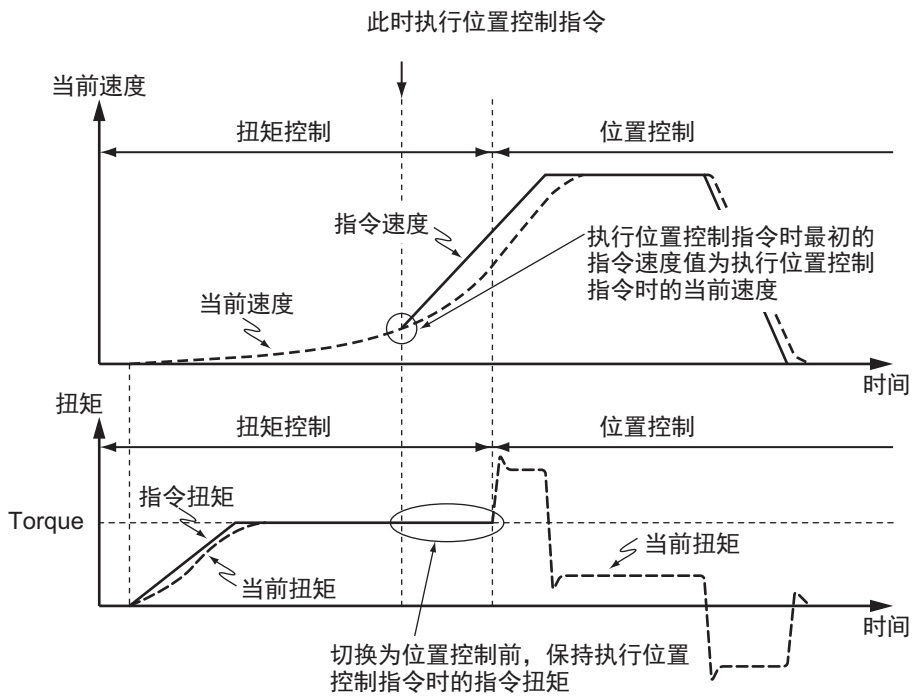
使用注意事项

轴正在动作时，若在伺服驱动器侧切换控制模式，在某些伺服驱动器上，可能发生异常。

从位置控制切换为扭矩控制时



从扭矩控制切换为位置控制时



运动指令重新执行

在扭矩控制动作中变更输入参数，并再次将 Execute（启动）设为 TRUE，可变更本指令的动作。通过运动指令重新执行可变更的输入变量包括 Torque（目标扭矩）、TorqueRamp（扭矩斜率）、Velocity（限制速度）。

通过运动指令的重新执行变更 Torque（目标扭矩）时，InTorque（达到目标扭矩）将根据重新执行后设定的新目标扭矩进行动作。

运动指令重新执行的详情请参考 □ □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

运动指令多重启动

运动指令多重启动的详情请参考 □ □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 其他指令执行过程中本指令的启动

对当前正在执行的指令启动本指令后，可切换为本指令或缓冲本指令。

每个轴最多可缓冲 1 个。

在 BufferMode（缓冲模式选择）中指定多重启动时本指令的动作。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。
缓冲	当前正在执行的指令正常完成后，自动启动已缓冲的本指令。

BufferMode（缓冲模式选择）请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

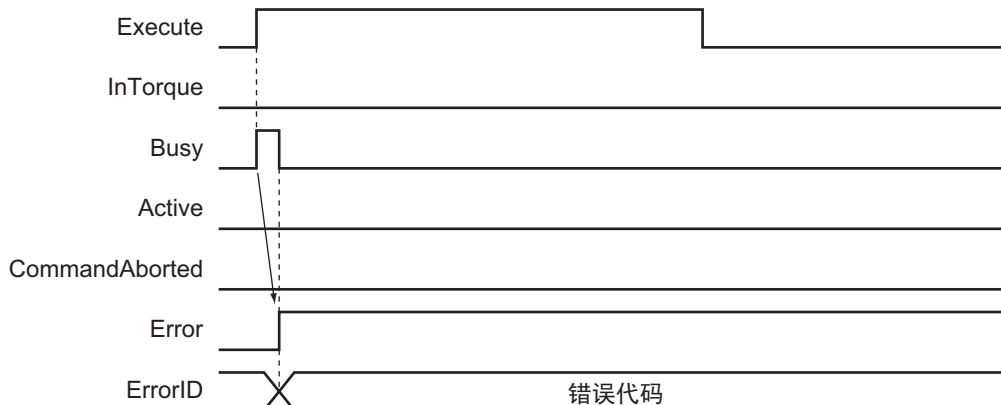
● 本指令执行过程中其他指令的启动

在本指令执行过程中，如需通过其他指令执行多重启动，需要将其他指令的 BufferMode（缓冲模式选择）指定为中断或缓冲。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。



● 错误代码

指令中发生的异常请参考 □ □ 「A-1 错误代码一览 (P.A-2)」。

MC_SetTorqueLimit

使用伺服驱动器的扭矩限制功能，限制伺服驱动器的输出扭矩。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_SetTorqueLimit	扭矩限制	FB		<pre>MC_SetTorqueLimit_instance (Axis := 《参数》, Enable := 《参数》, PositiveEnable := 《参数》, PositiveValue := 《参数》, NegativeEnable := 《参数》, NegativeValue := 《参数》, Enabled => 《参数》, Busy => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、FALSE	FALSE	TRUE 时，执行指令。
PositiveEnable	正方向有效	BOOL	TRUE、FALSE	FALSE	TRUE：正方向的扭矩限制变为有效。 FALSE：正方向的扭矩限制变为无效。
PositiveValue	正方向扭矩限制值	LREAL	0.1 ~ 1000.0 或“0.0”	300.0	以 0.1% 为单位指定正方向的扭矩限制值。 如果输入的值超出轴参数中的 [正方向扭矩限制上限值]，将变为 [正方向扭矩限制上限值]。 指定为“0”及负数时，作为“0”动作。
NegativeEnable	负方向有效	BOOL	TRUE、FALSE	FALSE	TRUE：负方向的扭矩限制变为有效。 FALSE：负方向的扭矩限制变为无效。
NegativeValue	负方向扭矩限制值	LREAL	0.1 ~ 1000.0 或“0.0”	300.0	以 0.1% 为单位指定负方向的扭矩限制值。 如果输入的值超出轴参数中的 [负方向扭矩限制上限值]，将变为 [负方向扭矩限制上限值]。 指定为“0”及负数时，作为“0”动作。

输出变量

输出变量	名称	数据类型	有效范围	内容
Enabled	有效	BOOL	TRUE、FALSE	控制过程中变为 TRUE。 ^{*1}
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*2	发生异常时，输出错误代码。16#0000 表示正常。

*1. Enabled（有效）表示本指令的状态，并非表示伺服驱动器的扭矩限制状态。

*2. □□ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Enabled	<ul style="list-style-type: none"> Enable 变为 TRUE 时 正在执行 MC_Power 指令时 	<ul style="list-style-type: none"> Enable 变为 FALSE 的下 1 个周期后 Error 变为 TRUE 时 MC_Power 指令的 Enable 变为 FALSE 时
Busy	Enable 变为 TRUE 时	<ul style="list-style-type: none"> Error 变为 TRUE 时 Enable 变为 FALSE 时
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

功能说明

- 本指令可对伺服驱动器设定扭矩限制值。
- 在 Enable（有效）为 TRUE 的状态下，
若将 PositiveEnable（正方向有效）设为 TRUE，将根据 PositiveValue（正方向扭矩限制值）进行限制。
若将 NegativeEnable（负方向有效）设为 TRUE，将根据 NegativeValue（负方向扭矩限制值）进行限制。
- 将 PositiveEnable（正方向有效）设为 FALSE 时，伺服驱动器中将设定轴参数 [正方向扭矩限制上限值]。
同样，将 NegativeEnable（负方向有效）设为 FALSE 时，伺服驱动器中将设定轴参数 [负方向扭矩限制上限值]。
- 将本指令的 Enable（有效）设为 FALSE 时，伺服驱动器中将设定 [正方向扭矩限制上限值]、[负方向扭矩限制上限值]。同时，Busy（执行中）、Enabled（有效）变为 FALSE。
- 以 0.1% 为单位对电机扭矩设定扭矩限制值。指定了小数点第 2 位以下时，小数点第 2 位四舍五入。



使用注意事项

- 轴参数 [正方向扭矩限制上限值]、[负方向扭矩限制上限值] 请设定为所用伺服驱动器中规定的扭矩限制上限值。
- NX 系列脉冲输出单元中无法使用本指令。

● 数据对象的映射

使用 MC_SetTorqueLimit（扭矩限制）指令时，请在 Sysmac Studio 的轴基本设定的 [详细设定] 中，映射以下对象数据。

- 正转侧扭矩限制值 (60E0 Hex)
- 反转侧扭矩限制值 (60E1 Hex)

未设定以上内容时，将发生过程数据对象设定不足（错误代码：3461Hex）。



使用注意事项

- 如果使用其他公司生产的伺服驱动器，设定伺服驱动器时，请确保正转侧扭矩限制值 (60E0 Hex)、反转侧扭矩限制值 (60E1 Hex) 为扭矩限制值。
设定方法请参考 □ □ 《各伺服驱动器的手册》。
- 使用其他公司生产的伺服驱动器，正转侧扭矩限制值、反转侧扭矩限制值无法映射到 PDO 时，无法使用本指令。
此时，请使用其他公司的支持功能或 SDO 通信，设定扭矩限制值。

数据对象的映射请参考 □ □ 「2-3 PDO 映射 (P.2-30)」、□ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 输入参数的重写

Enable（有效）为 TRUE 时，可经常更新以下输入变量的值。

- PositiveEnable（正方向有效）
- NegativeEnable（负方向有效）
- PositiveValue（正方向扭矩限制值）
- NegativeValue（负方向扭矩限制值）

● 与 MC_Home（原点复位）指令、MC_HomeWithParameter（参数指定原点复位）指令的推压动作的关系 （使用欧姆龙生产的伺服驱动器 G5 系列时）

原点复位的 [12: 附近退避、推压时间指定]、[13: 无近原点输入、推压原点输入指定] 将使用伺服驱动器中事先设定的扭矩限制值，向推压方向自动开始扭矩限制。



使用注意事项

使用其他公司生产的伺服驱动器时，MC_Home（原点复位）指令或 MC_HomeWithParameter（参数指定原点复位）指令的自动扭矩限制功能不会启用。

请使用 MC_SetTorqueLimit（扭矩限制）指令、SDO 通信、伺服驱动器的支持工具等，设定为合适的值。



参考

- 原点复位正常完成后，继续进行扭矩限制。
- 启动了向反方向移动的指令时，将自动解除扭矩限制。
原点复位的详情请参考 □ 「MC_Home(P.3-15)」或
□ 「MC_HomeWithParameter(P.3-36)」。

● 欧姆龙生产的伺服驱动器 G5 系列的设定

要使用本指令，需要利用伺服驱动器的支持软件，设定伺服驱动器的扭矩限制选择（3521 Hex）。

- 原点复位推压动作时，对原点检测方向进行扭矩控制，且在其他动作中使用本指令的扭矩限制方向或扭矩限制值时，请使用“6”。

此时，在原点复位的推压动作过程中，本指令的输入变量的内容将被忽略。

- 设定为“4”时，始终适用本指令的输入变量内容。原点复位的推压动作时以及其他动作时，需要设定为合适的扭矩限制值。

		扭矩限制选择（3521 Hex）	
		6（推荐）	4
正方向扭矩限制值	原点复位中 *1	适用第 3 扭矩限制（3525 Hex）。	适用本指令的 PositiveValue（正方向扭矩限制值）和第 1 扭矩限制（3013 Hex）中的较小者。
	原点复位以外	适用本指令的 PositiveValue（正方向扭矩限制值）和第 1 扭矩限制（3013 Hex）中的较小者。	
负方向扭矩限制值	原点复位中 *1	适用第 4 扭矩限制（3526 Hex）。	适用本指令的 NegativeValue（负方向扭矩限制值）和第 2 扭矩限制（3522 Hex）中的较小者。
	原点复位以外	适用本指令的 NegativeValue（负方向扭矩限制值）和第 2 扭矩限制（3522 Hex）中的较小者。	

*1. 扭矩限制自动解除前

详情请参考 □ □ 《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型 用户手册 (SBCE-CN5-365)》、
□ □ 《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型 用户手册 (SBCE-CN5-366)》。

● 与 MC_TorqueControl（扭矩控制）指令的关系

MC_SetTorqueLimit（扭矩限制）指令和 MC_TorqueControl（扭矩控制）指令可组合使用。

● 对属于轴组的轴使用

可对属于有效轴组的轴使用本指令。

● 与 CPU 单元动作模式的关系

运行模式中通过本指令设定的内容在变为程序模式时，仍可保持。

● 可使用的轴和启动条件

- 可在伺服轴中使用本指令。

但是，本指令的输出变量“Enabled（有效）”状态因伺服状态不同而异。

	伺服 ON 状态	伺服 OFF 状态
Enable（有效）的状态	TRUE	FALSE ^{*1}

*1. 本指令的 Enabled（有效）为 FALSE 时，伺服驱动器的扭矩限制不会启用。

- 即使是虚拟伺服轴，也可接收指令，但不执行扭矩限制。
- 编码器轴、虚拟编码器轴中，启动时会发生异常。

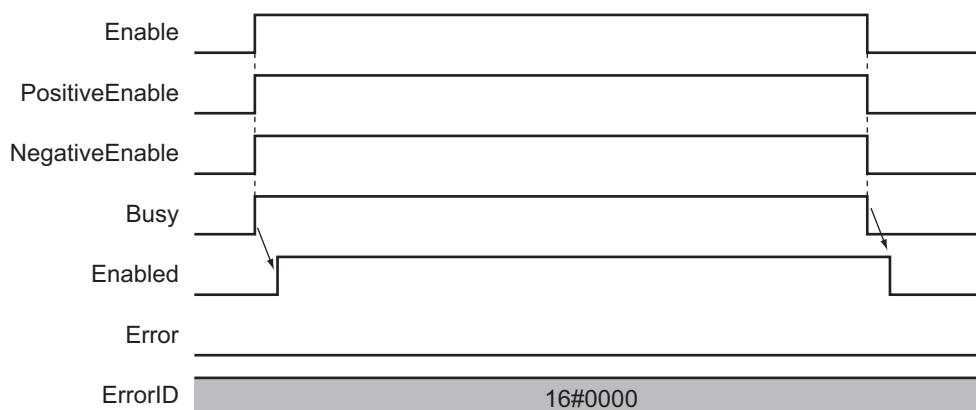
● 轴变量的状态（伺服驱动器状态）

伺服驱动器的内部限制功能运行时，轴变量 DrvStatus（伺服驱动器状态）的 ILA（驱动器内部功能限制中）将变为 TRUE。

本变量为伺服驱动器的扭矩限制、速度限制、驱动禁止输入、软件限位的 OR 输出。

时序图

执行了扭矩限制时的时序图如下。



运动指令重新执行

输入为“Enable 型”的指令没有运动指令重新执行。

运动指令多重启动

运动指令多重启动的详情请参考 [□ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》](#)。

● 本指令的多重启动

对正在执行本指令的轴启动了本指令的其他实例时，后面执行的实例将优先处理。

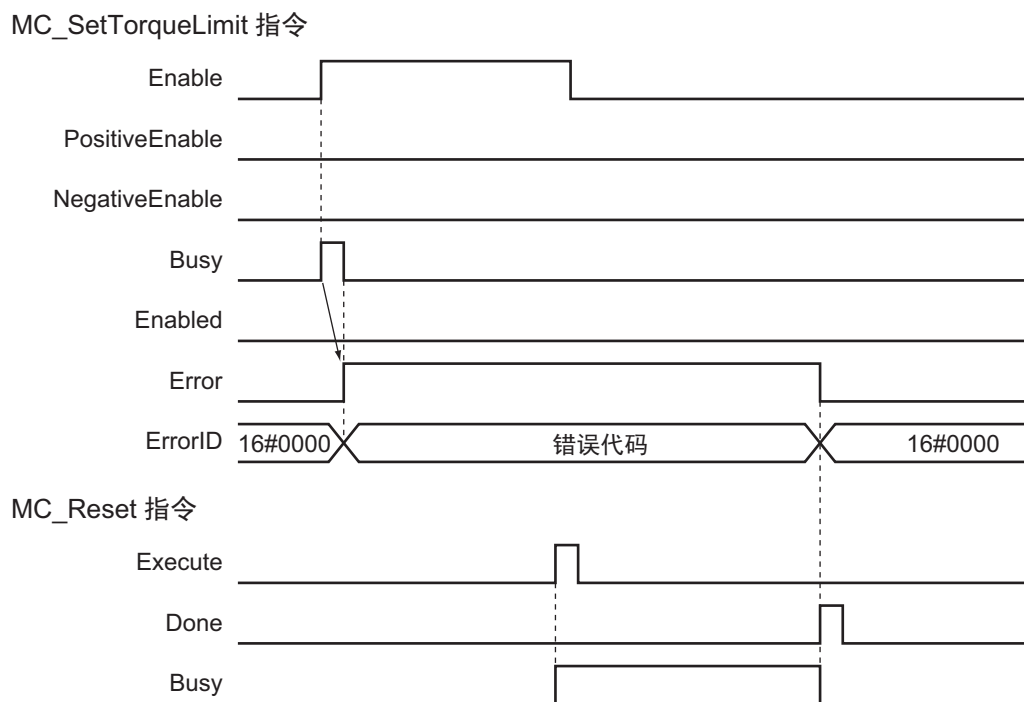
Enabled（有效）时，两者均变为 TRUE。

具体动作为，后面执行的实例的扭矩限制值变为有效。若将后面执行的实例的 Enable（有效）设为 FALSE，扭矩限制将变为无效。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。
参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

MC_ZoneSwitch

判断轴的指令当前位置或反馈当前位置是否存在于指定范围内。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_ZoneSwitch	区域监视	FB		<pre>MC_ZoneSwitch_instance (Axis := 《参数》, Enable := 《参数》, FirstPosition := 《参数》, LastPosition := 《参数》, ReferenceType := 《参数》, Enabled => 《参数》, InZone => 《参数》, Busy => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、FALSE	FALSE	TRUE 时，执行指令。
FirstPosition	开始位置	LREAL	负数、正数、“0”	0	指定区域范围的开始位置。 ^{*1} 单位为 [指令单位]。 ^{*2}
LastPosition	结束位置	LREAL	负数、正数、“0”	0	指定区域范围的结束位置。 ^{*3} 单位为 [指令单位]。 ^{*1}
ReferenceType	位置种类选择	_eMC_ REFERENCE_ TYPE	0: _mcCommand 1: _mcFeedback	0 ^{*4}	指定要监视的轴信息。 0: 指令位置（前一任务周期 ^{*5} 中的计算 值） 1: 反馈位置（同一任务周期 ^{*5} 中的获 取值）

*1. 请指定为小于结束位置的值。

*2. 指令单位请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》的“单位转换设定”。

*3. 请指定为大于开始位置的值。

*4. 如果是有效范围为列举型的变量初始值，实际不是数值，而是列举值。

*5. 任务为主固定周期任务时，任务周期为主周期，为固定周期任务（执行优先级 5）时，任务周期为固定周期任务（执行优先级 5）的任务周期。

输出变量

输出变量	名称	数据类型	有效范围	内容
Enabled	有效	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
InZone	区域内	BOOL	TRUE、FALSE	轴位置在区域范围内时变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* 请参见「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Enabled	Enable 变为 TRUE 时	Enable 变为 FALSE 时
InZone	已进入区域内时	<ul style="list-style-type: none"> 超出区域时 Enable 变为 FALSE 时
Busy	Enabled 的上升沿	<ul style="list-style-type: none"> Error 变为 TRUE 时 Enable 变为 FALSE 时
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_SAXIS_REF	—	指定轴。*1

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis***”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

功能说明

- 在本指令为 Enable（有效）的状态下，轴的指令位置或反馈当前位置存在于 FirstPosition（开始位置）以上、LastPosition（结束位置）以下的区域内时，输出变量的 InZone（区域内）将变为 TRUE。在 ReferenceType（位置种类选择）中，可将监视对象的轴信息设定为指令位置或反馈位置中的任意一个。
- 可在所有轴种类中进行区域监视。
- 在 Enable（有效）为 TRUE 的状态下，重写了 FirstPosition（开始位置）或 LastPosition（结束位置）时，将从进行重写的周期开始应用新的值。
- 对 1 个轴可设定多个区域，即使区域重复，亦可设定。此外，超出软件限位范围外时，也可设定区域。



使用注意事项

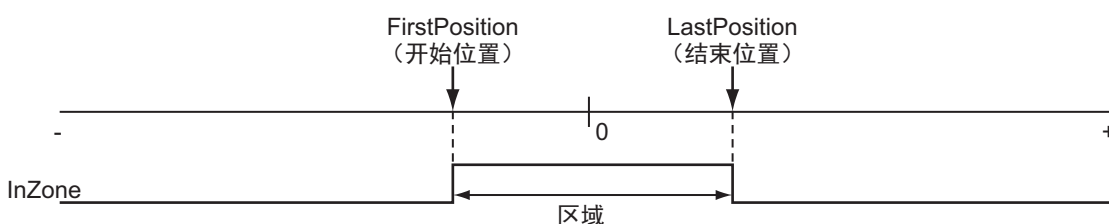
FirstPosition（开始位置）、LastPosition（结束位置）包含无法除尽的除法运算结果时，受取整误差的影响，处理结果可能与预计的不同。

指令的详情

设定 FirstPosition（开始位置）和 LastPosition（结束位置）时，请确保在不同的计数模式下，以下关系成立。关系不成立时，将发生异常。

● 直线模式

请指定为 FirstPosition（开始位置） \leq LastPosition（结束位置）。

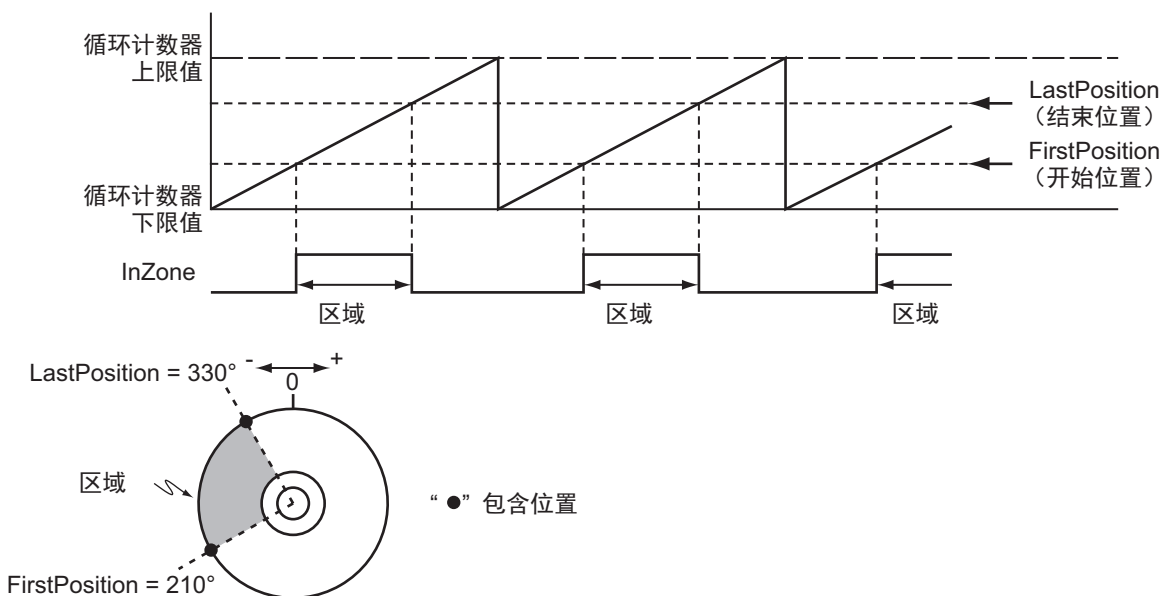


● 旋转模式

[旋转模式] 时，跨或不跨循环计数器的上下限位置时，有以下不同。

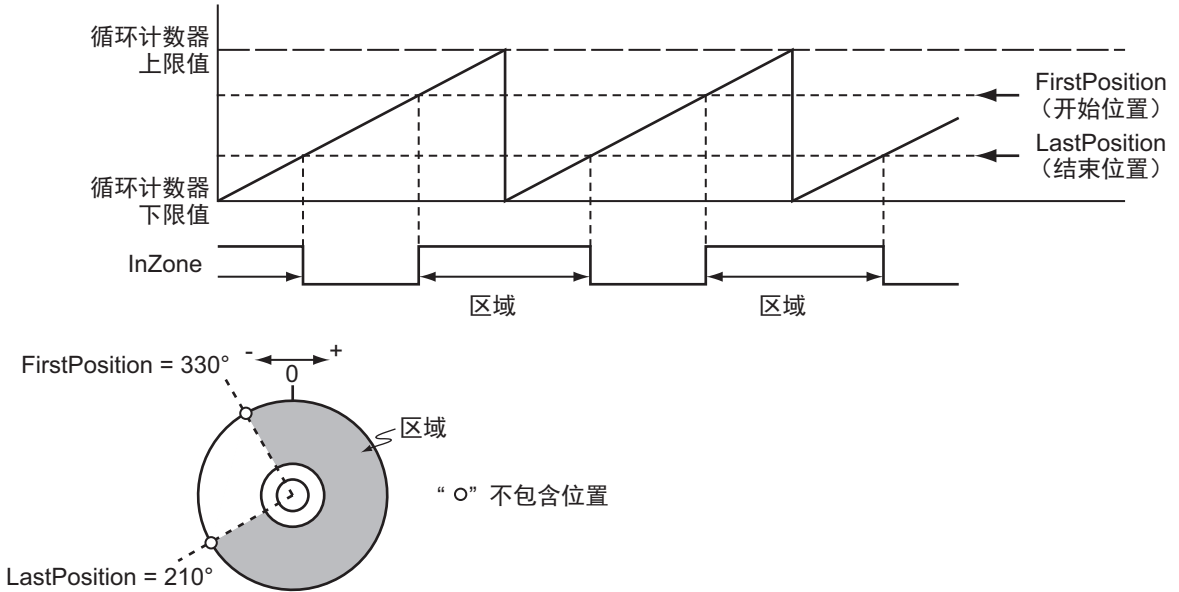
不跨上下限位置时

请指定为 FirstPosition（开始位置） \leq LastPosition（结束位置）。



跨上下限位置时

请指定为 FirstPosition (开始位置) > LastPosition (结束位置)。



● ReferenceType (位置种类选择)

从以下项目中选择位置种类。

- `_mcCommand`: 指令位置 (前一任务周期中的计算值)
 在当前周期, 使用前一任务周期中计算的主轴指令位置。
 使用计算从轴指令位置之前的固定周期任务中计算得到的主轴指令位置。
- `_mcFeedback`: 同一任务周期的获取值
 使用同一任务周期内获取的主轴的反馈位置。



使用注意事项

任务为主固定周期任务时, 这里的任务周期为主周期, 为固定周期任务 (执行优先级 5) 时, 任务周期为固定周期任务 (执行优先级 5) 的任务周期。同样, 固定周期任务表示主固定周期任务或固定周期任务 (执行优先级 5)。

● 轴种类和位置种类的关系

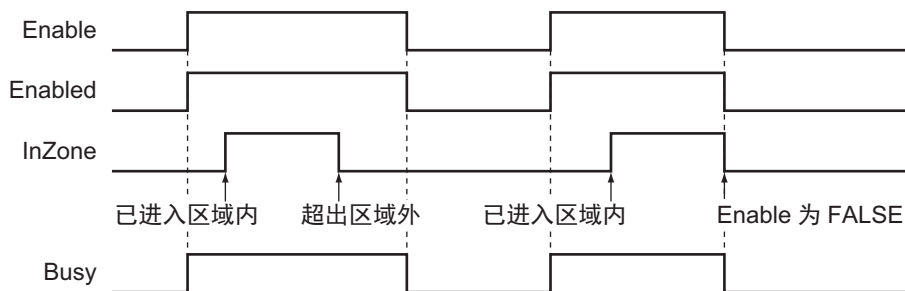
可监视的轴种类和要监视的位置种类的关系如下所示。

轴种类	ReferenceType	
	<code>_mcCommand</code>	<code>_mcFeedback</code>
伺服轴	○	○
编码器轴	x ^{*1}	○
虚拟伺服轴	○	○
虚拟编码器轴	x ^{*1}	○

*1. 指令启动时, 将发生“超出位置种类选择范围 (错误代码: 5430 Hex)”。

时序图

- 动作过程中进入区域内时，以及在区域内 Enable（有效）变为 FALSE 时



- 指令启动前位于区域内，中途超出区域时



运动指令重新执行

输入为“Enable 型”的指令没有运动指令重新执行。

运动指令多重启动

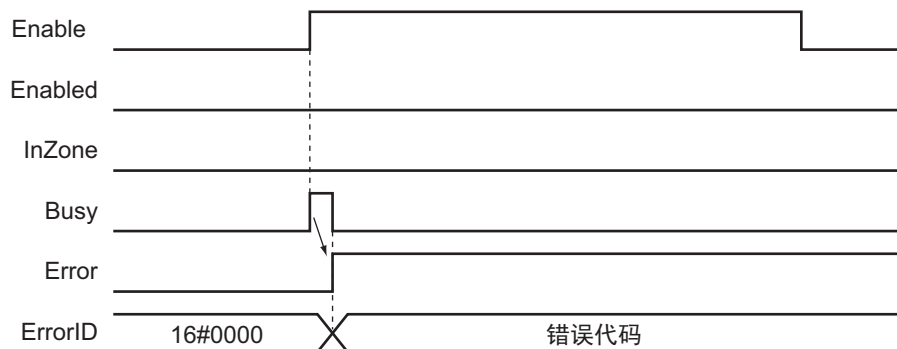
本指令中，各指令单独动作，因此运动指令多重启动没有限制。

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

异常

无法启动本指令时，本指令将发生异常，Error（错误）变为 TRUE。
参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

MC_TouchProbe

发生触发信号，记录轴位置。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_TouchProbe	外部锁定有效	FB		<pre>MC_TouchProbe_instance (Axis := 《参数》, TriggerInput := 《参数》, TriggerVariable := 《参数》, Execute := 《参数》, WindowOnly := 《参数》, FirstPosition := 《参数》, LastPosition := 《参数》, ReferenceType := 《参数》, StopMode := 《参数》, Done => 《参数》, Busy => 《参数》, RecordedPosition => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
WindowOnly	窗口有效	BOOL	TRUE、FALSE	FALSE	指定窗口主机的有效 / 无效。
FirstPosition	开始位置	LREAL	负数、正数、“0”	0	指定开始接收触发的位置。使用 ReferenceType（位置种类选择）中指定的种类。 ^{*1} 单位为 [指令单位]。 ^{*2}
LastPosition	结束位置	LREAL	负数、正数、“0”	0	指定停止接收触发的位置。使用 ReferenceType（位置种类选择）中指定的种类。 ^{*1} 单位为 [指令单位]。 ^{*2}
ReferenceType (Reserved)	位置种类选择	_eMC_ REFERENCE_ TYPE	1: _mcFeedback	1 ^{*3}	(Reserved)
StopMode	停止方法选择	_eMC_STOP_ MODE	1: _mcImmediateStop 4: _mcNonStop	4 ^{*3}	指定停止方法。 1: 立即停止 4: 不停止

*1. 详情请参考 □ 「WindowOnly（窗口有效）(P.3-335)」。

*2. 指令单位请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》的“单位转换设定”。

*3. 如果是有效范围为枚举型的变量初始值，实际不是数值，而是列举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
RecordedPosition	锁定位置	LREAL	负数、正数、“0”	输出锁定的位置。 单位为 [指令单位]。*1
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*2	发生异常时，输出错误代码。16#0000 表示正常。

*1. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. □ □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	<ul style="list-style-type: none"> 发生触发信号后，记录锁定位置，并完成指令时 停止指定有效时，上述情况下，且轴停在锁定位置时 	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动（中断），本指令中止时 因发生异常，本指令中止时 StopMode 为 [_mclImmediateStop]，在执行过程中变更为 CSP 模式以外时*1 从站脱离时 发生从站通信异常时（过程数据通信中除外） 启动了 MC_AbortTrigger 指令时 	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

*1. 使用 NX 系列 CPU 单元时，没有控制模式的限制，因此即使为 CSP 模式以外，也不会发生 CommandAborted。

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。 ^{*1}
TriggerInput	触发输入条件	_sTRIGGER_REF	—	设定触发条件。 ^{*2}
TriggerVariable	触发变量	BOOL	TRUE、FALSE	在触发条件中指定了控制器模式时，指定作为触发的输入变量。

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

*2. 请创建 _sTRIGGER_REF 型用户定义变量。

● _sTRIGGER_REF

成员变量	名称	数据类型	有效范围	功能
Mode	模式	_eMC_TRIGGER_MODE	0: _mcDrive 1: _mcController	指定触发的模式。 0: 驱动器模式 1: 控制器模式
LatchID	锁定 ID 选择	_eMC_TRIGGER_LATCH_ID	0: _mcLatch1 1: _mcLatch2	驱动器模式时，指定使用两个锁定功能中的哪个。 0: 锁定功能 1 1: 锁定功能 2
InputDrive	驱动器触发输入信号	_eMC_TRIGGER_INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	指定驱动器模式时的驱动器触发信号。 0: Z 相 1: 外部输入

功能说明

- 以传感器输入等触发的发生为基点，以位置控制为目的，根据触发信号的发生来记录（锁定）轴位置。
- 根据触发设定，将 Axis（轴）中指定的轴位置输出到 RecordedPosition（锁定位置）。
触发设定中可指定 TriggerInput（触发输入条件）、WindowOnly（窗口有效）、FirstPosition（开始位置）、LastPosition（结束位置）、StopMode（停止方法选择）。
- 在同一 MC_TouchProbe（外部锁定有效）指令的实例中再次记录轴位置前，RecordedPosition（锁定位置）的输出值将保持不变。

● 数据对象的映射

Mode 选择为 [驱动器模式]，使用 MC_TouchProbe（外部锁定有效）指令时，请映射以下对象数据。映射在 Sysmac Studio 的轴基本设定 [详细设定] 中进行。

轴种类为伺服轴时

- 锁定功能（60B8Hex）
- 锁定状态（60B9Hex）
- 锁定位置 1（60BAHex）
- 锁定位置 2（60BCHex）

轴种类为编码器轴时

- 锁定功能 (4020Hex)
- 编码器输入从站软交换 (4020Hex)
- 锁定状态 (4030Hex)
- 锁定位置 1 (4012Hex)
- 锁定位置 2 (4013Hex)
- 编码器输入从站状态 (4030Hex)

未设定以上内容时，将发生过程数据对象设定不足（错误代码：3461Hex）。

数据对象的映射请参考 □ 「2-3 PDO 映射 (P.2-30)」，□ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

使用 NX 系列位置接口单元时请参考 □ 《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》的“I/O 入口映射”。

指令的详情

下面介绍本指令的详情。

● Axis（轴）指定

- 在 Axis（轴）中指定锁定对象的轴。
- 指定的 Axis（轴）正在执行 MC_GroupEnable（多轴协调控制）指令时，MC_TouchProbe（外部锁定有效）指令将变为异常，无法执行。
- 可对各轴指定 LatchID（锁定 ID 选择），最多同时执行 2 个 MC_TouchProbe（外部锁定有效）指令。
- 指定用 MC_AbortTrigger（外部锁定无效）指令中止的锁定时，也会使用 LatchID（锁定 ID 选择）。



参考

- 可使用欧姆龙生产的 GX 系列 EtherCAT 编码器输入从站 GX-EC02 □ □ 中连接的编码器轴，利用本指令进行锁定。
- 也可使用 NX 系列脉冲输出单元，利用本指令进行锁定。详情请参考 □ 《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

● 触发输入条件

通过 TriggerInput（触发输入条件）的 Mode、LatchID、InputDrive 指定，选择触发的条件。

Mode

- Mode 的设定包括：在触发中指定伺服驱动器信号的“驱动器模式”；在触发中指定 TriggerVariable（触发变量）的“控制器模式”。
- 触发信号的上升沿视为发生触发，记录执行锁定指令后最先发生触发（FALSE → TRUE）时轴的位置。
- 本指令为 Busy（动作中）时，即使 Execute（启动）为 FALSE，TriggerVariable（触发变量）的变量仍作为触发动作。

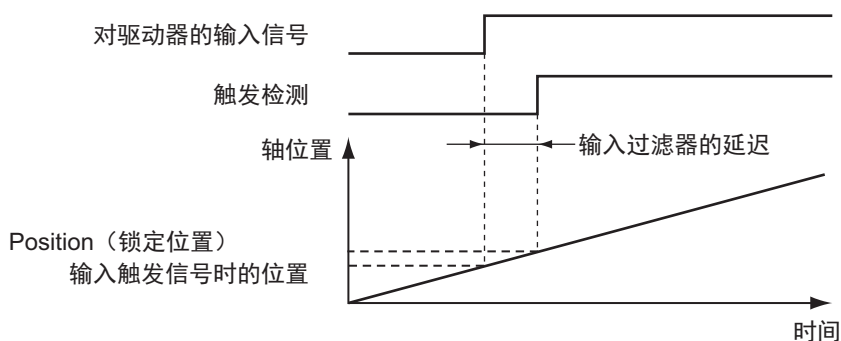


参考

使用欧姆龙生产的 GX 系列 EtherCAT 编码器输入从站 GX-EC02 □□时，Mode 请设定为 [_mcDrive（驱动模式）]。

驱动器模式

“驱动器模式”通过伺服驱动器的功能进行触发检测和获取当前位置，因此与“控制器模式”相比，锁定的当前位置更准确。



使用注意事项

- 使用驱动器模式时，请务必在要使用的 LatchID（锁定 ID 选择）上连接锁定信号。
- 锁定信号的宽度取决于伺服驱动器的性能等。



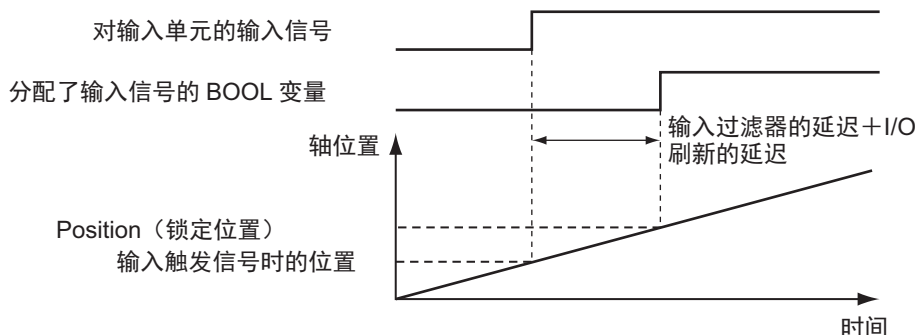
参考

使用欧姆龙生产的 GX 系列 EtherCAT 编码器输入从站 GX-EC02 □□时，InputDrive 请指定为 _mcEXT（外部输入）。

欧姆龙生产的 GX 系列 EtherCAT 编码器输入从站 GX-EC02 □□中没有利用 Z 相进行的保持功能。因此，指定了 _mcEncoderMark（Z 相）时，启动本指令时将发生异常。启动时 Error（错误）变为 TRUE，ErrorID（错误代码）中将输出“过程数据对象设定不足（错误代码：3461 Hex）”。

控制器模式

- “控制器模式”中，触发可指定为 BOOL 变量。
- 在 TriggerVariable（触发变量）中指定要作为触发的 BOOL 变量。
- 在“控制器模式”中，触发输入信号向 BOOL 变量的反映会因 I/O 刷新的延迟而比“驱动器模式”大幅延迟。



使用注意事项

使用控制器模式时，根据任务周期的间隔执行锁定处理。因此，触发变量必须在任务周期的 1 个周期内保持 TRUE。

此外，触发变量变为 TRUE 后，在处理 MC 功能模块前，需要 1 个任务周期的时间。

任务为主固定周期任务时，这里的任务周期为主周期，为固定周期任务（执行优先级 5）时，任务周期为固定周期任务（执行优先级 5）的任务周期。

LatchID

- 对 1 个轴最多可同时执行 2 个 MC_TouchProbe（外部锁定有效）指令。
在 LatchID（锁定 ID 选择）中指定使用哪个。
- 对同一个轴指定的锁定 ID 已经在执行时，后面执行的指令变为有效。之前的指令将变为 CommandAborted（执行中断）。
- 在伺服驱动器中，LatchID（锁定 ID 选择）表示伺服驱动器的锁定回路 1（第 1ch）、锁定回路 2（第 2ch）。

LatchID 请参考 [□□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》](#)。



参考

使用欧姆龙生产的 GX 系列 EtherCAT 编码器输入从站 GX-EC02 [□□](#)时，对应如下。

列举值	编码器输入终端的信号名称
_mcLatch1	外部锁定输入 A
_mcLatch2	外部锁定输入 B

InputDrive

- 可选择 `_mcEncoderMark` (Z相) 和 `_mcEXT` (外部输入) 作为触发。
- `_mcEncoderMark` (Z相) 中, 将伺服驱动器等的 Z 相用于触发。
`_mcEXT` (外部输入) 中, 将伺服驱动器中输入的外部触发信号用于触发。
- `_mcEXT` (外部输入) 在欧姆龙生产的伺服驱动器 G5 系列中, 为设定的 `Ext1`、`Ext2`、`Ext3` 之一。设定通过 Sysmac Studio 进行。
驱动器中设定的 2 个触发可以是相同的设定。

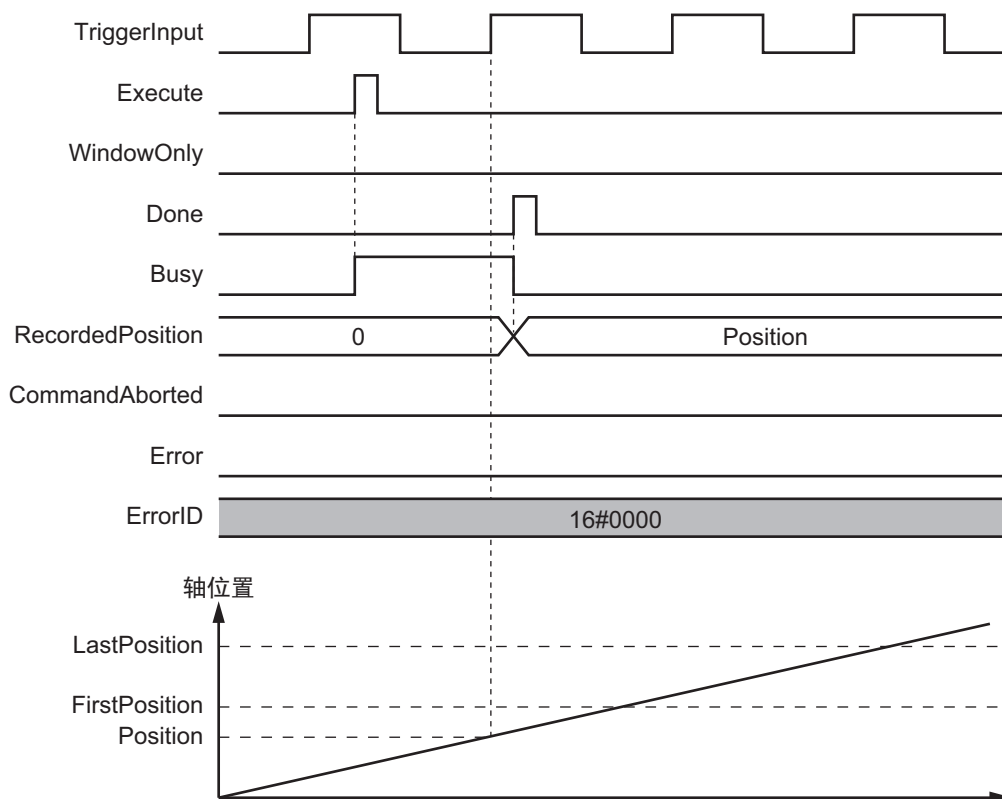
● WindowOnly (窗口有效)

- 在 WindowOnly (窗口有效) 中指定窗口的有效 / 无效。
- 指定为 Disable (无效) 时, 在所有的轴位置上检测触发。
- 指定为 Enable (有效) 时, 仅检测轴位置为 FirstPosition (开始位置) 和 LastPosition (结束位置) 范围内的触发。

不同 WindowOnly (窗口有效) 指定时的动作如时序图所示。

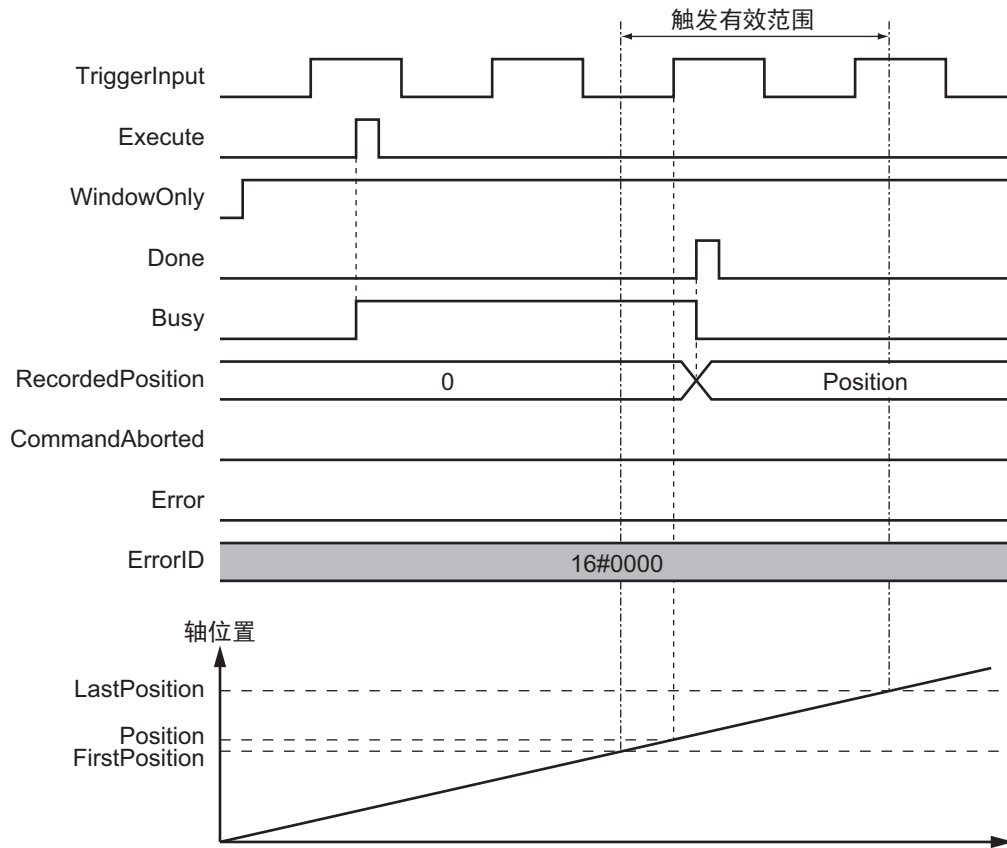
WindowOnly 为 Disable 时

Execute (启动) 变为 TRUE 后, 将最先发生触发时的轴位置输出到 RecordedPosition (锁定位置)。



WindowOnly 为 Enable 时

仅在窗口范围内，检测触发输入，获取轴位置。



使用注意事项

- WindowOnly（窗口有效）从 FALSE 变为 TRUE 的瞬间和锁定功能动作前无法锁定。
- 锁定功能开始动作需要一定的时间，因此 WindowOnly（窗口有效）的有效范围非常短时，将无法锁定。可锁定的有效范围极限值取决于伺服驱动器、编码器输入终端、位置接口单元等的性能及 EtherCAT 通信。

此外，因计数模式的不同，FirstPosition（开始位置）和LastPosition（结束位置）的范围如下所示。

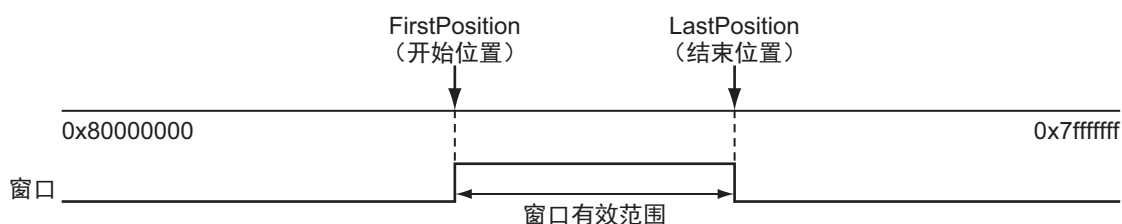
直线模式

- 窗口的有效范围按以下公式表述。

$$\text{FirstPosition (开始位置)} \leq \text{窗口范围} \leq \text{LastPosition (结束位置)}$$

- 指定为 FirstPosition（开始位置）> LastPosition（结束位置）时，将发生异常。
 - 指定为超出 [直线模式] 的位置范围时，也会发生异常。
 - 此外，FirstPosition（开始位置）和 LastPosition（结束位置）为 LREAL 型变量，因此请勿指定为 FirstPosition（开始位置）= LastPosition（结束位置）。
- LREAL 型请参考 □「实数（REAL 型、LREAL 型）的处理 (P.1-14)」。

[直线模式] 时的窗口有效范围如下图所示。

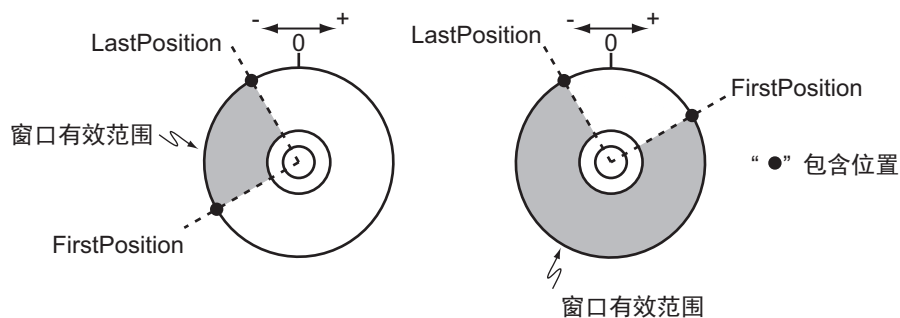


(注) FirstPosition（开始位置）和 LastPosition（结束位置）包含于窗口有效范围内。

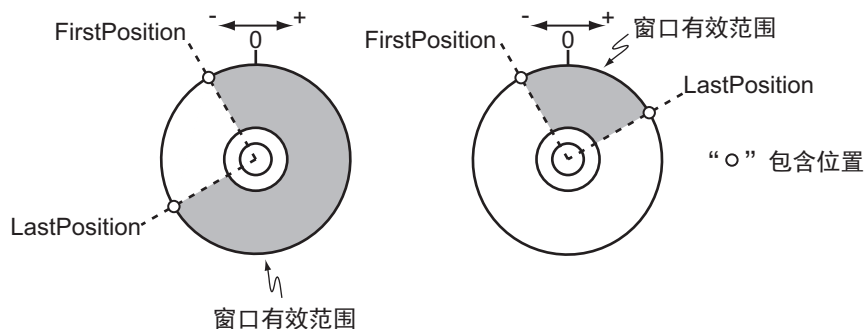
旋转模式

- 可指定为 FirstPosition（开始位置）≤ LastPosition（结束位置）及 FirstPosition（开始位置）> LastPosition（结束位置）。
- FirstPosition（开始位置）> LastPosition（结束位置）时，将跨循环计数器的上下限位置。
- 若指定的位置超出循环计数器上下限范围，将发生异常。

$$\text{FirstPosition} \leq \text{LastPosition}$$



$$\text{FirstPosition} > \text{LastPosition}$$

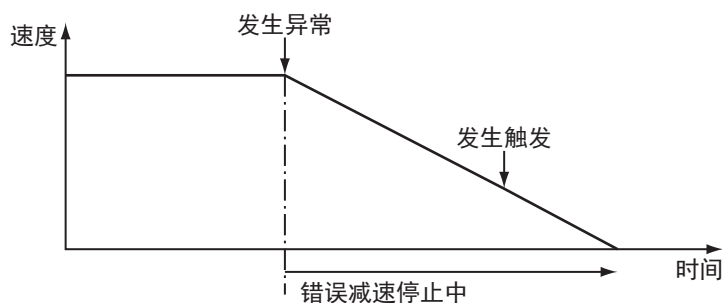


● StopMode（停止方法选择）

- 发生触发时，可对指定的 Axis（轴）指定 StopMode（停止方法选择）。
- `_mcNonStop`（不停止）时，即使发生触发，轴也不会停止。
- `_mclImmediateStop`（立即停止）时，发生触发后，轴在锁定位置立即停止。停止后，正在轴上动作的指令将变为 `CommandAborted`（执行中断）。
- `_mclImmediateStop`（立即停止）时，在轴指令停止之时变为 `Done`（完成）。在锁定位置上停止前，`Busy`（执行中）为 `TRUE`。
- 通过 `_mclImmediateStop`（立即停止）停止时，不进行位置检查。

单元版本 1.10 以上的 CPU 单元时

- 指定为 `_mclImmediateStop`（立即停止）的轴发生轴异常，并在轴完成停止前，若发生触发，轴将继续减速停止。此外，发生触发后，将变为 `CommandAborted`（执行中断）。

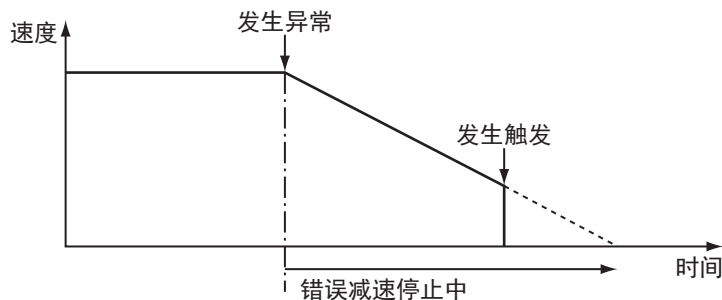


使用注意事项

- `_mclImmediateStop`（立即停止）时，MC 功能模块将执行停止处理。实际上，会停在超出锁定位置的地方，再根据控制器的指示返回到锁定位置，敬请注意。
- 使用 `_mclImmediateStop`（立即停止）时，若指令速度较快，返回到锁定位置的量会变大。设定指令速度时，请注意不可过快。
- 编码器轴请指定为 `_mcNonStop`（不停止）。指定为 `_mclImmediateStop`（立即停止）时，启动本指令时 `Error`（错误）将变为 `TRUE`。同时，`ErrorID`（错误代码）中将输出“外部锁定有效指令不可执行（错误代码：5492 Hex）”。

单元版本 1.09 以下的 CPU 单元时

- 指定为 `_mclImmediateStop`（立即停止）的轴发生轴异常，并在轴完成停止前，若发生触发，将利用欧姆龙生产的伺服驱动器 G5 系列的功能，轴立即停止。此外，发生触发后，将变为 `CommandAborted`（执行中断）。
- 运动控制功能模块在检测到欧姆龙伺服驱动器 G5 系列停止后，立即停止指令。下图的虚线表示未发生触发的目标轨道。





使用注意事项

- `_mclImmediateStop`（立即停止）在 CSP 模式时启用。
 - 在 CSV/CST 模式的状态下，如果指定为 `_mclImmediateStop`（立即停止）并启动，将发生“外部锁定有效指令不可执行（错误代码：5492Hex）”。此外，如果在 `_mclImmediateStop`（立即停止）动作过程中变更为 CSV/CST 模式，将变为 `CommandAborted`（执行中断）。
 - `_mclImmediateStop`（立即停止）会在使用欧姆龙生产的伺服驱动器 G5 系列或 NX 系列脉冲输出单元时，使用伺服驱动器或 NX 系列脉冲输出单元的功能停止。
使用其他公司生产的伺服驱动器时，MC 功能模块将执行停止处理。
任一情况下，实际会停在超出锁定位置的地方，再根据控制器的指示返回到锁定位置，敬请注意。
 - 使用 `_mclImmediateStop`（立即停止）时，若指令速度较快，返回到锁定位置的量会变大。
设定指令速度时，请注意不可过快。
 - 编码器轴请指定为 `_mcNonStop`（不停止）。指定为 `_mclImmediateStop`（立即停止）时，启动本指令时 `Error`（错误）将变为 `TRUE`。同时，`ErrorID`（错误代码）中将输出“外部锁定有效指令不可执行（错误代码：5492Hex）”。
 - 使用 `_mclImmediateStop`（立即停止）时，若在同一控制周期内执行以下 2 个处理，欧姆龙伺服驱动器 G5 系列将变为 `Target Ignore` 状态。
 - 锁定信号 ON 时伺服驱动器的计数器锁定处理
 - `MC_TouchProbe`（外部锁定有效）指令中，`CommandAborted`（执行中断）变为 `TRUE` 时的锁定解除处理
- 确保在锁定信号变为 ON 时，本指令的 `CommandAborted`（执行中断）不会变为 `TRUE`。

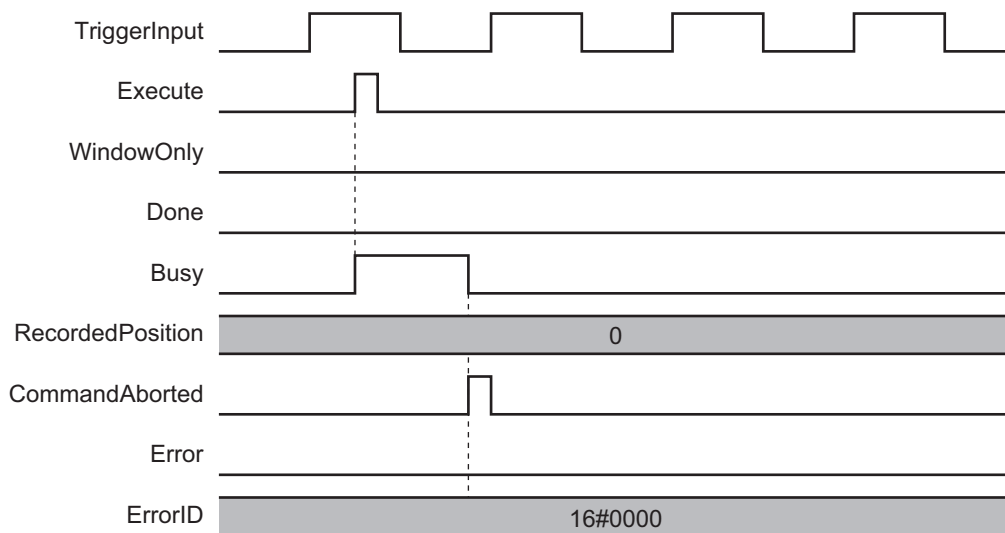
● 轴变量的状态

`StopMode`（停止方法选择）指定为 `_mclImmediateStop`（立即停止）时，如果因发生触发而正在执行停止处理，轴变量 `Status.Stopping`（减速停止中）将变为 `TRUE`。

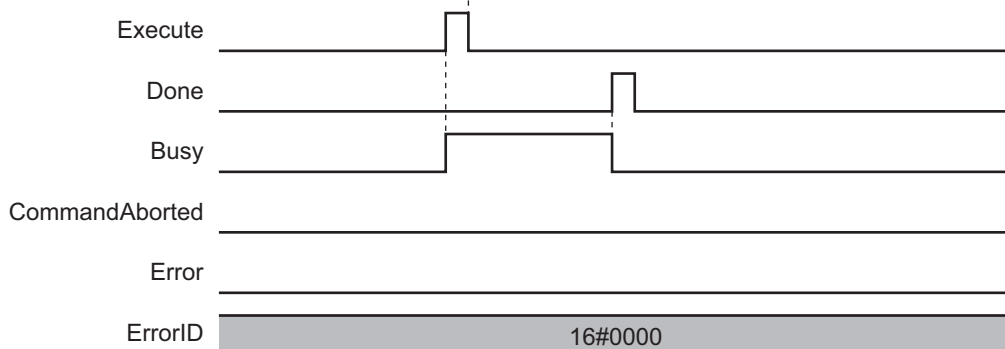
指令的中止

MC_TouchProbe（外部锁定有效）指令可通过 MC_AbortTrigger（外部锁定无效）指令结束。
在 MC_AbortTrigger（外部锁定无效）指令中指定需要中止的 Axis（轴）和 LatchID（锁定 ID 选择）并启动，即可结束。

MC_TouchProbe 指令



MC_AbortTrigger 指令

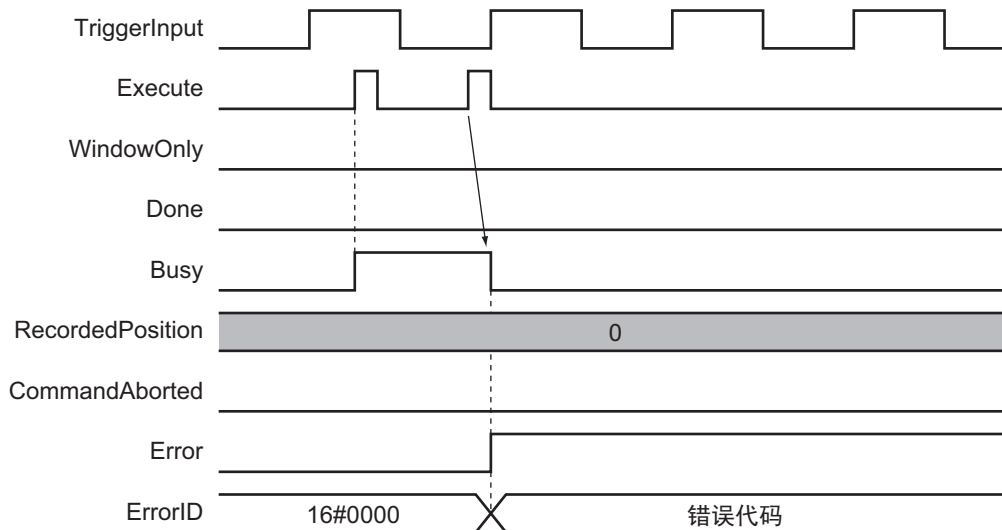


运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

对同一 MC_TouchProbe（外部锁定有效）指令的实例读取 RecordedPosition（锁定位置）且在读取完成前，若再次将 Execute（启动）从 FALSE 设为 TRUE，将发生异常。



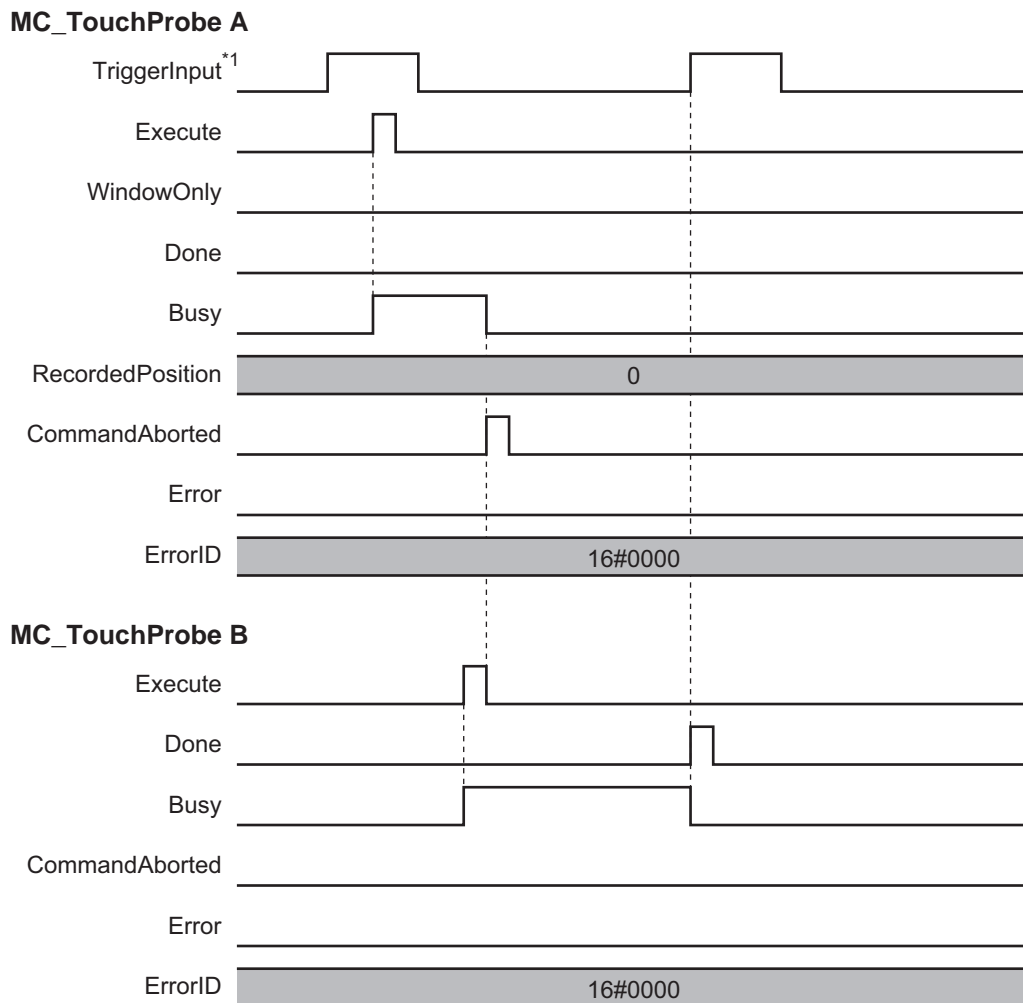
运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 其他指令执行过程中本指令的启动

对同一 Axis (轴) 的 1 个 LatchID (锁定 ID 选择) 只能同时执行 1 个触发设定。

正在执行 MC_TouchProbe (外部锁定有效) 指令时, 若对同一 LatchID (锁定 ID 选择) 执行了其他 MC_TouchProbe (外部锁定有效) 指令的实例, 第 1 个将变为 CommandAborted (执行中断), 第 2 个全新执行。



*1 伺服驱动器等的触发输入信号时。



参考

StopMode (停止方法选择) 指定为 _mclmmediateStop (立即停止) 时, 因发生触发而停止时, 多重启动的指令的 CommandAborted (执行中断) 将变为 TRUE。

异常

执行 MC_TouchProbe（外部锁定有效）指令的过程中发生异常时，Error（错误）将变为 TRUE。
参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 [□「A-1 错误代码一览 \(P.A-2\)」](#)。

示例程序

下面介绍利用外部传感器进行位置提取控制相关的示例程序。

参数设定

下面表示本示例程序中使用的最低限度的设定。

● 轴参数

轴种类

轴	轴种类
轴 1	伺服轴

计数器模式

轴	计数器模式
轴 1	旋转模式

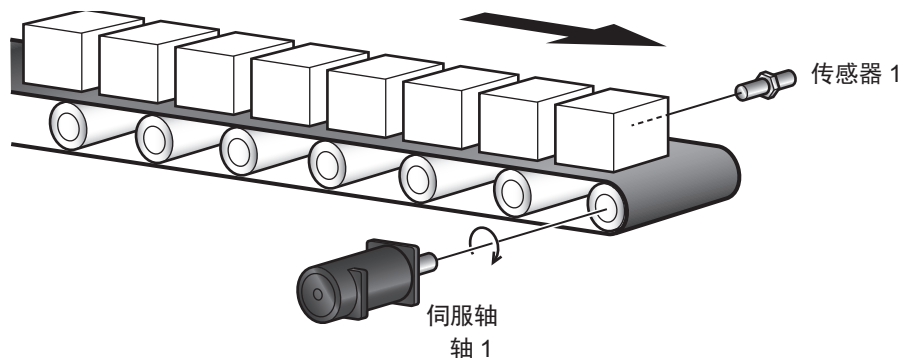
循环计数器

轴	上限值	下限值
轴 1	360	0

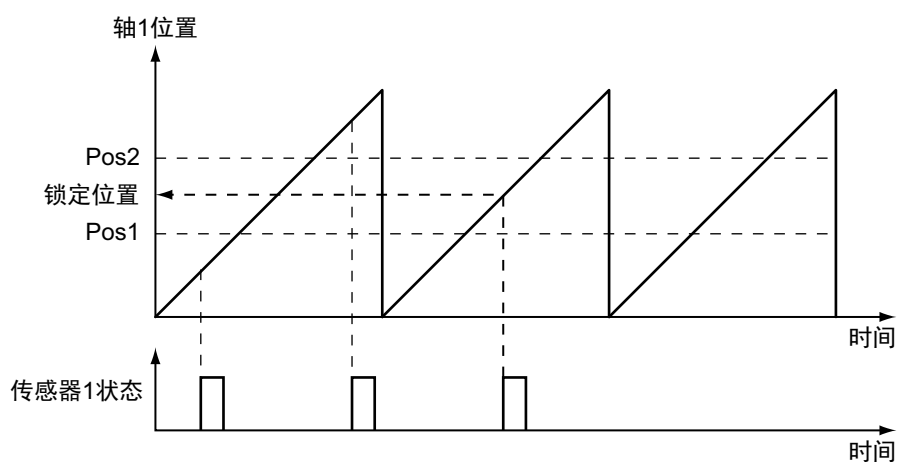
坐标单位选择

轴	坐标单位
轴 1	degree

动作示例



● 动作模式



1 主轴的开始

轴 1 执行速度控制。

2 工件的检测

利用传感器 1 检测工件的有无。

3 位置的获取

在窗口的 TRUE 范围内 (Pos1 ~ Pos2) 检测到工件时, 获取此时轴 1 的锁定位置。

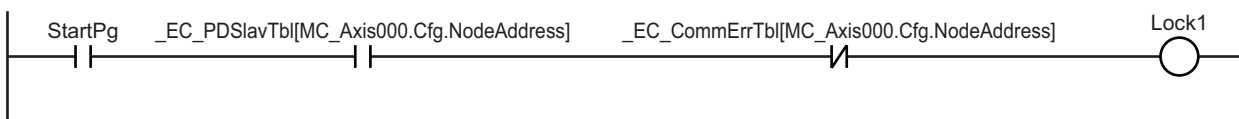
梯形图

● 主要变量

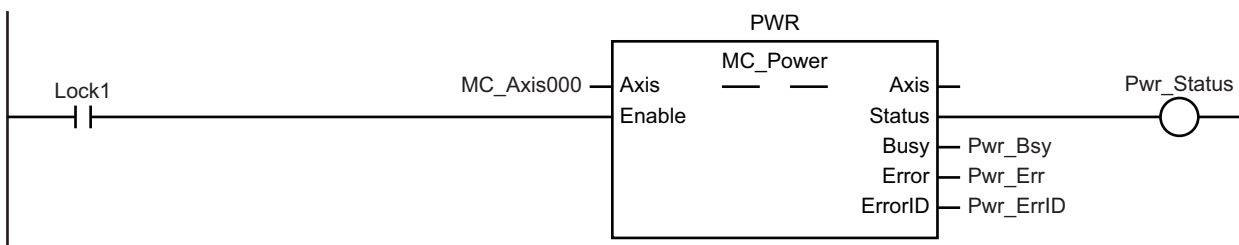
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
Pwr_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR 的输出变量 Status 中的变量。若变为伺服 ON 状态，该变量变为 TRUE。
Pos1	LREAL	—	表示窗口有效的开始位置。
Pos2	LREAL	—	表示窗口有效的结束位置。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态，则变为伺服 ON 状态。
InitFlag	BOOL	FALSE	表示输入参数设定完成。FALSE 时，设定输入参数。输入参数设定完成后，变为 TRUE。

● 示例程序

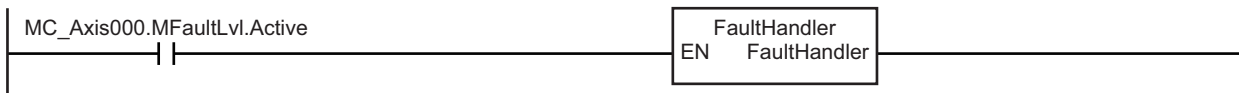
接点 StartPg 变为 TRUE 后，确认 EtherCAT 通信是否正常进行过程数据通信



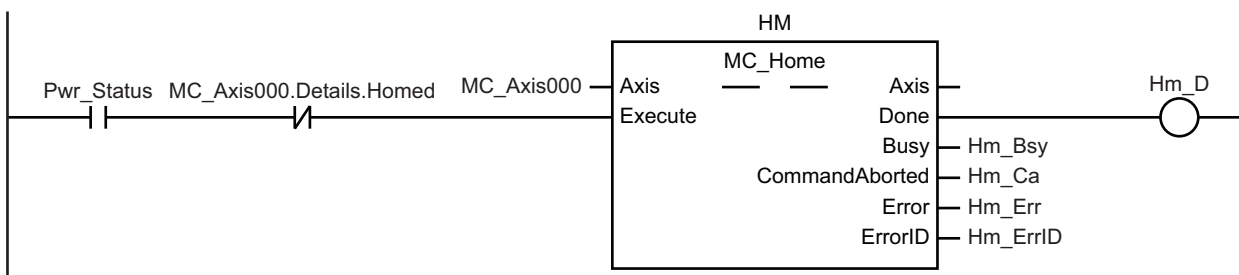
若轴 1 正在进行过程数据通信，则变为伺服 ON 状态



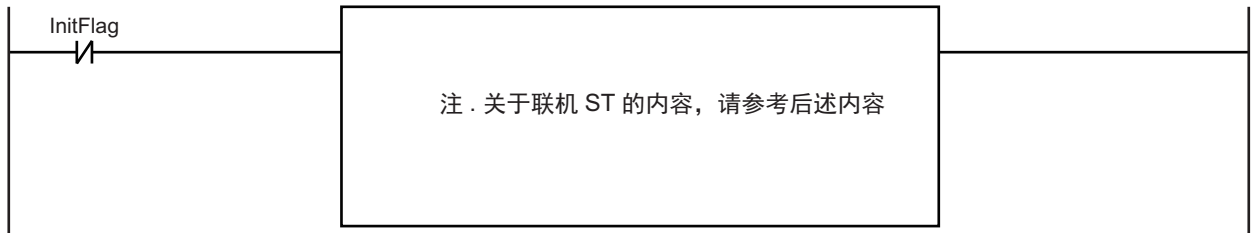
轴 1 发生轻度故障等级的异常时，执行装置发生异常时的处理（FaultHandler）。发生异常时的处理（FaultHandler）请根据装置由客户进行编程



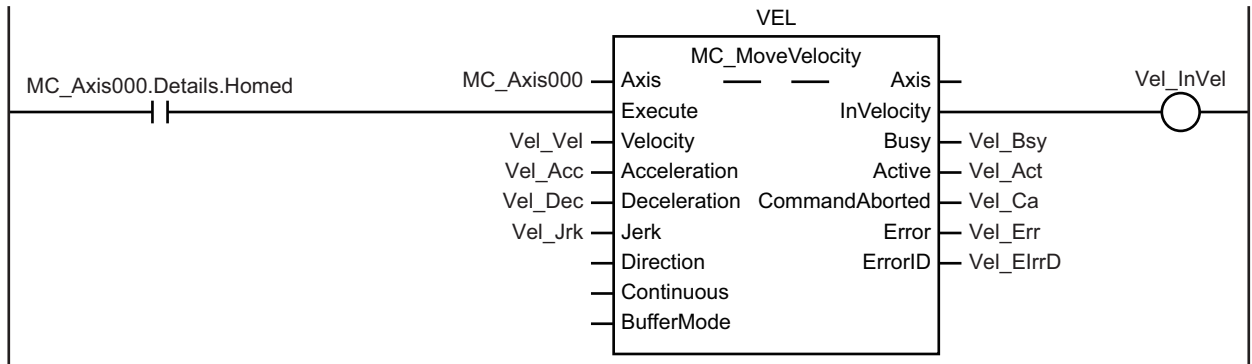
变为伺服 ON 状态且处于原点未确定状态时，执行原点复位



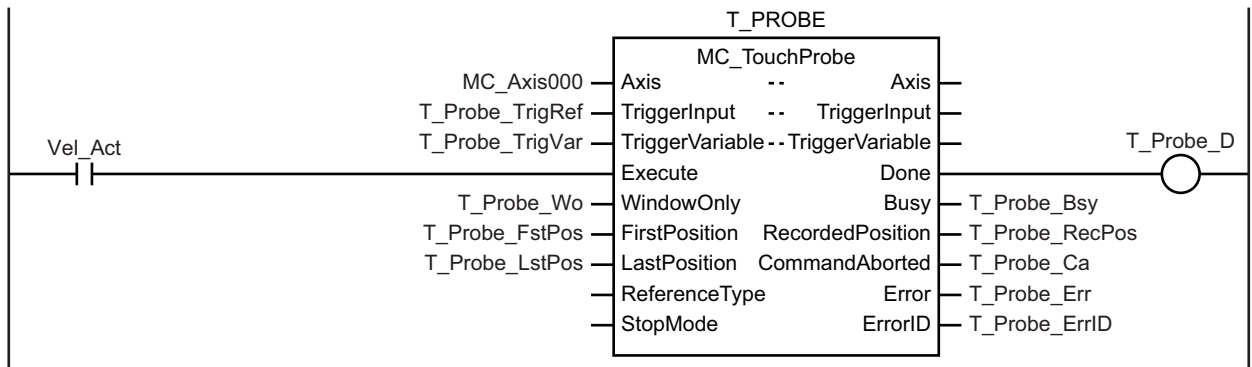
设定 MC_MoveVelocity（速度控制）指令和 MC_TouchProbe（外部锁定有效）指令的参数



轴 1 变为原点确定状态后, 执行 MC_MoveVelocity（速度控制）指令



开始 MC_MoveVelocity（速度控制）指令后, 执行锁定处理



联机 ST 的内容

//MC_MoveVelocity 参数

```
Vel_Vel := LREAL#1000.0;
Vel_Acc := LREAL#1000.0;
Vel_Dec := LREAL#1000.0;
Vel_Jrk := LREAL#1000.0;
```

//MC_TouchProbe 参数

```
T_Probe_TrigRef.Mode := _eMC_TRIGGER_MODE#_mcDrive;
T_Probe_TrigRef.LatchID := _eMC_TRIGGER_LATCH_ID#_mcLatch1;
T_Probe_TrigRef.InputDrive := _eMC_TRIGGER_INPUT_DRIVE#_mcEXT;
T_Probe_TrigVar := FALSE;
T_Probe_Wo := TRUE;
T_Probe_FstPos := LREAL#1000.0;
T_Probe_LstPos := LREAL#2000.0;
```

```
// 设定输入参数后, 将 InitFlag 设为 TRUE
InitFlag:=TRUE;
```

结构文本 (ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时, 变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
Pwr_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR 的输出变量 Status 中的变量。若变为伺服 ON 状态, 该变量变为 TRUE。
Pos1	LREAL	—	表示窗口有效的开始位置。
Pos2	LREAL	—	表示窗口有效的结束位置。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态, 则变为伺服 ON 状态。
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE 时, 设定输入参数。输入参数设定完成后, 变为 TRUE。

● 示例程序

```
// 未设定输入参数时的处理
IF InitFlag=FALSE THEN

    //MC_MoveVelocity 参数
    Vel_Vel := LREAL#1000.0;
    Vel_Acc := LREAL#1000.0;
    Vel_Dec := LREAL#1000.0;
    Vel_Jrk := LREAL#1000.0;

    //MC_TouchProbe 参数
    T_Probe_TrigRef.Mode      := _eMC_TRIGGER_MODE#_mcDrive;
    T_Probe_TrigRef.LatchID   := _eMC_TRIGGER_LATCH_ID#_mcLatch1;
    T_Probe_TrigRef.InputDrive := _eMC_TRIGGER_INPUT_DRIVE#_mcEXT;
    T_Probe_TrigVar          := FALSE;
    T_Probe_Wo               := TRUE;
    T_Probe_FstPos           := LREAL#1000.0;
    T_Probe_LstPos           := LREAL#2000.0;

    // 设定输入参数后, 将 InitFlag 设为 TRUE
    InitFlag:=TRUE;

END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 设为伺服 ON 状态。
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbI[MC_Axis000.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbI[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
  Pwr_En:=TRUE;
ELSE
  Pwr_En:=FALSE;
END_IF;
```

```

// 发生轻度故障等级异常时的处理
// 发生异常时的处理 (FaultHandler) 请根据装置由客户进行编程。
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;

// 伺服 ON 状态且原点未确定时, 执行原点复位
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm_Ex:=TRUE;
END_IF;

// 确定原点后, 执行 MC_MoveVelocity
IF MC_Axis000.Details.Homed=TRUE THEN
    Vel_Ex:=TRUE;
END_IF;

// 启动 MC_MoveVelocity 后, 执行 MC_TouchProbe
IF Vel_Act=TRUE THEN
    T_Probe_Ex:= TRUE;
END_IF;

//MC_Power
PWR(
    Axis      := MC_Axis000,
    Enable    := Pwr_En,
    Status    => Pwr_Status,
    Busy      => Pwr_Bsy,
    Error     => Pwr_Err,
    ErrorID   => Pwr_ErrID
);

//MC_Home
HM(
    Axis          := MC_Axis000,
    Execute       := Hm_Ex,
    Done          => Hm_D,
    Busy          => Hm_Bsy,
    CommandAborted => Hm_Ca,
    Error         => Hm_Err,
    ErrorID       => Hm_ErrID
);

//MC_MoveVelocity
VEL(
    Axis          := MC_Axis000,
    Execute       := Vel_Ex,
    Velocity      := Vel_Vel,
    Acceleration  := Vel_Acc,
    Deceleration  := Vel_Dec,
    Jerk          := Vel_Jrk,
    InVelocity    => Vel_InVel,
    Busy          => Vel_Bsy,
    Active        => Vel_Act,
    CommandAborted => Vel_Ca,
    Error         => Vel_Err,
    ErrorID       => Vel_ErrID
);

```



```
//MC_TouchProbe
T_PROBE(
  Axis           := MC_Axis000,
  TriggerInput   := T_Probe_TrigRef,
  TriggerVariable := T_Probe_TrigVar,
  Execute        := T_Probe_Ex,
  WindowOnly     := T_Probe_Wo,
  FirstPosition  := T_Probe_FstPos,
  LastPosition   := T_Probe_LstPos,
  Done           => T_Probe_D,
  Busy           => T_Probe_Bsy,
  RecordedPosition => T_Probe_RecPos,
  CommandAborted => T_Probe_Ca,
  Error          => T_Probe_Err,
  ErrorID        => T_Probe_ErrID
);
```

MC_AbortTrigger

将执行中的锁定变为无效。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_AbortTrigger	外部锁定无效	FB		<pre>MC_AbortTrigger_instance (Axis := 《参数》, TriggerInput := 《参数》, Execute := 《参数》, Done => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、 FALSE	FALSE	上升时开始指令。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、 FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、 FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* 请参见「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	<ul style="list-style-type: none"> 指定了锁定停止时 对未执行锁定的锁定功能，执行本指令，并结束处理时 	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动（中断），本指令中止时 因发生异常，本指令中止时 	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Error	本指令的启动或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定锁定对象的轴。 ^{*1}
TriggerInput	触发选择	_sTRIGGER_REF	—	选择触发条件。 ^{*2} _sTRIGGER_REF 请参考下表。

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

*2. 请创建 _sTRIGGER_REF 型用户定义变量。

● _sTRIGGER_REF

成员变量	名称	数据类型	有效范围	功能
Mode	模式	_eMC_TRIGGER_MODE	0: _mcDrive 1: _mcController	指定触发的模式。 0: 驱动器模式 1: 控制器模式
LatchID	锁定 ID 选择	_eMC_TRIGGER_LATCH_ID	0: _mcLatch1 1: _mcLatch2	驱动器模式时，指定使用两个锁定功能中的哪个。 0: 锁定功能 1 1: 锁定功能 2
InputDrive	驱动器触发输入信号	_eMC_TRIGGER_INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	指定驱动器模式时的驱动器触发信号。 0: Z 相 1: 外部输入

功能说明

- 中止执行中的锁定。
- 在 MC_AbortTrigger（外部锁定无效）指令中，指定 Axis（轴）和 LatchID（锁定 ID 选择），可指定要中止的锁定。
- 对未执行锁定请求的触发执行了 MC_AbortTrigger（外部锁定无效）指令时，MC_AbortTrigger（外部锁定无效）指令将不执行任何操作，正常结束。
对 MC_TouchProbe（外部锁定有效）指令的执行已经为 Done（完成）的指令执行了 MC_AbortTrigger（外部锁定无效）指令时亦同。

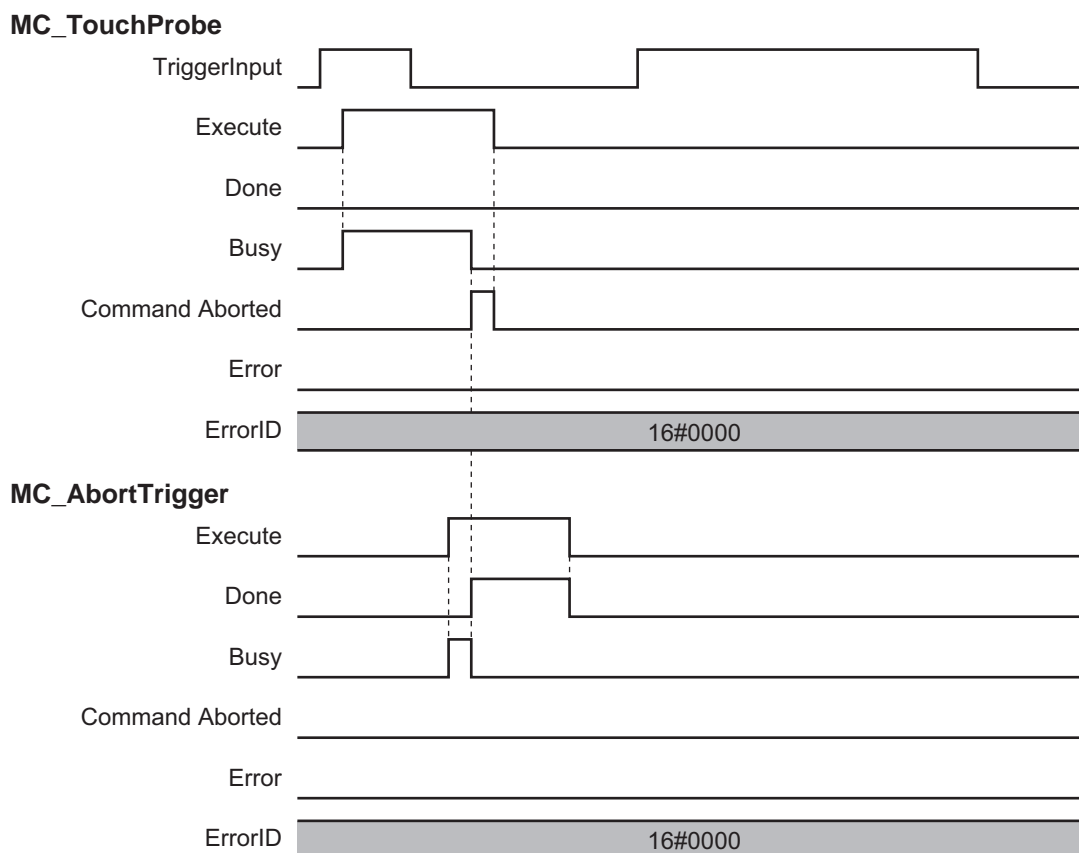


使用注意事项

- 指定了 MC_AbortTrigger（外部锁定无效）指令的 Axis（轴），在 MC_GroupEnable（轴组有效）指令有效时，MC_AbortTrigger（外部锁定无效）指令将变为异常，无法执行。
- 正在执行 MC_Home（原点复位）指令、MC_HomeWithParameter（参数指定原点复位）指令、MC_MoveFeed（中断固定尺寸定位）指令、MC_MoveLink（梯形模式凸轮）指令时，若执行了 MC_AbortTrigger（外部锁定无效）指令，本指令将发生异常。

时序图

- MC_AbortTrigger（外部锁定无效）指令的 Done（完成）在 Execute（启动）的 1 个周期后变为 TRUE。



运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

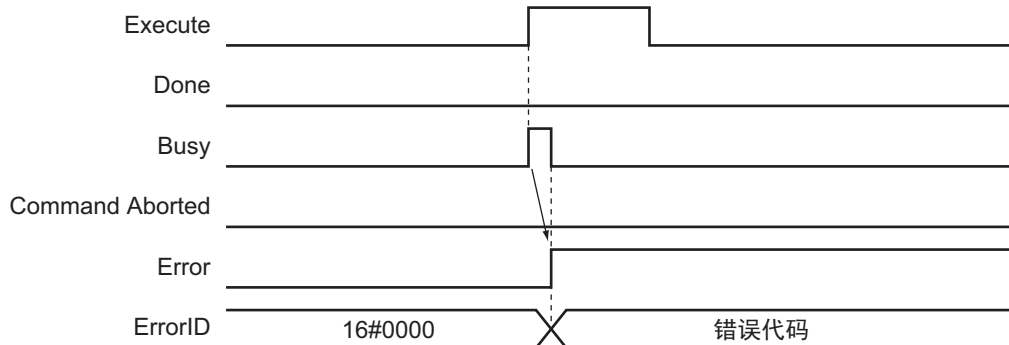
运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

异常

执行 MC_AbortTrigger（外部锁定无效）指令的过程中发生异常时，Error（错误）将变为 TRUE。参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

MC_AbortTrigger



● 错误代码

指令中发生的异常请参考 □ 「A-1 错误代码一览 (P.A-2)」。

MC_AxesObserve

监视指定轴的指令位置或反馈位置的差异量是否超出容许值。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_AxesObserve	轴间偏差监视	FB		<pre>MC_AxesObserve_instance (Master := 《参数》, Slave := 《参数》, Enable := 《参数》, ReferenceType := 《参数》, PermittedDeviation := 《参数》, Enabled => 《参数》, Invalid => 《参数》, Busy => 《参数》, DeviatedValue => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、FALSE	FALSE	TRUE 时，执行指令。
ReferenceType	位置种类选择	_eMC_ RERERENCE_ TYPE	0: _mcCommand 1: _mcFeedback	0 *1	指定位置的类别。 0: 指令位置（前一任务周期 ^{*2} 中的计算值） 1: 反馈位置（同一任务周期 ^{*2} 中的获取值）
Permitted Deviation	轴间偏差容许值	LREAL	正数 或“0”	0	指定容许的主轴和从轴的位置偏差最大值。 单位为 [指令单位]。 ^{*3}

*1. 如果是有效范围为枚举型的变量初始值，实际不是数值，而是枚举值。

*2. 任务为主固定周期任务时，任务周期为主周期，为固定周期任务（执行优先级 5）时，任务周期为固定周期任务（执行优先级 5）的任务周期。

*3. 指令单位请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》的“单位转换设定”。

输出变量

输出变量	名称	数据类型	有效范围	内容
Enabled	有效	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
Invalid	轴间偏差超限	BOOL	TRUE、FALSE	超出轴间偏差容许值时变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
DeviatedValue	轴间偏差值	LREAL	负数、正数、“0”	输出指定主轴和从轴的差。单位为 [指令单位]。*1
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*2	发生异常时，输出错误代码。16#0000 表示正常。

*1. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. □ □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Enabled	Enable 为 TRUE 时	<ul style="list-style-type: none"> • Error 变为 TRUE 时 • Enable 变为 FALSE 时
Invalid	超出轴间偏差容许量时	<ul style="list-style-type: none"> • Error 变为 TRUE 时 • Enable 变为 FALSE 时 • 未超出轴间偏差容许量时
Busy	Enable 的上升沿	<ul style="list-style-type: none"> • Error 变为 TRUE 时 • Enable 变为 FALSE 时
DeviatedValue *1	Enable 为 TRUE 时	
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

*1. Deviated Value 在 Enable 变为 FALSE 时，不会恢复为 FALSE。

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Master	主轴	_sAXIS_REF	—	指定主轴。*1
Slave	从轴	_sAXIS_REF	—	指定从轴。*1

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。



使用注意事项

主轴和从轴请分配到同一任务中。若将不同任务中的轴指定为主轴，将发生“主轴指定不正确（错误代码：5462 Hex）”。



参考

组中所属的轴亦可设定。

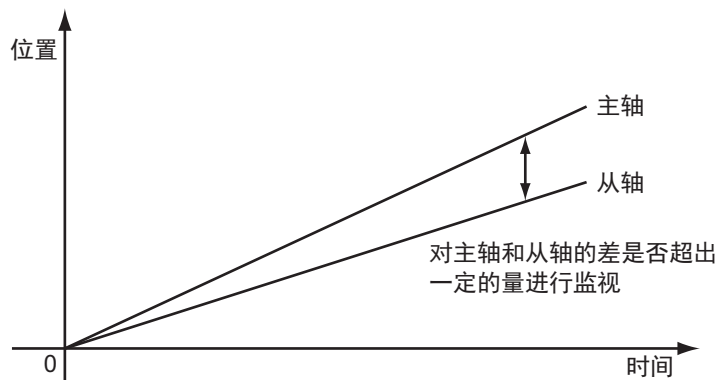
功能说明

- 指定 Master（主轴）和 Slave（从轴）的指令位置或反馈位置的差超出轴间偏差容许值时，Invalid（轴间偏差超限）将变为 TRUE。

即以下条件成立时，Invalid（轴间偏差超限）变为 TRUE。

$$| \text{DeviatedValue (轴间偏差值)} | > \text{PermittedDeviation (轴间偏差容许值)} \text{ 成立时}$$

- 本指令不会影响轴的动作。
- 编程时，请确保按输出变量 Invalid（轴间偏差超限）的状态执行轴停止处理等处理。



使用注意事项

- Master（主轴）和 Slave（从轴）的计数模式请设定为相同的模式。
若指定为不同模式的轴，将作为 [直线模式] 进行比较。
此外，即使都是 [旋转模式]，如果循环计数器的设定范围不同，也会作为 [直线模式] 进行比较。
- PermittedDeviation（轴间偏差容许值）包括除不尽的除法运算结果时，受取整误差的影响，处理结果可能与预计的不同。
- NX 系列 CPU 单元中，Master（主轴）和 Slave（从轴）请分配到同一任务中。如果分配到不同的任务中，Slave（从轴）侧将发生“主轴指定不正确（错误代码：5462 Hex）”。
- NX 系列脉冲输出单元中，由于指令位置 = 反馈当前位置，本指令的监视将无效。

指令的详情

下面介绍指令的详情。

● ReferenceType（位置种类选择）

从轴要同步的主轴位置种类可从以下项目中选择。

- `_mcCommand`: 指令位置（前一任务周期中的计算值）
在当前周期，使用前一任务周期中计算的主轴指令位置。
使用计算从轴指令位置之前的固定周期任务中计算得到的主轴指令位置。
- `_mcFeedback`: 同一任务周期的获取值
使用同一任务周期内获取的主轴的反馈位置。



使用注意事项

任务为主固定周期任务时，这里的任务周期为主周期，为固定周期任务（执行优先度 5）时，任务周期为固定周期任务（执行优先度 5）的任务周期。同样，固定周期任务表示主固定周期任务或固定周期任务（执行优先度 5）。

● 轴种类和位置种类的关系

可监视的轴种类和要监视的位置种类的关系如下所示。

轴种类	ReferenceType	
	_mcCommand	_mcFeedback
伺服轴	○	○
编码器轴	x ^{*1}	○
虚拟伺服轴	○	○
虚拟编码器轴	x ^{*1}	○

*1. 指令启动时，将发生“超出位置种类选择范围（错误代码：5430 Hex）”。

● DeviatedValue（轴间偏差值）的计算方法

DeviatedValue（轴间偏差值）按以下方法计算。

直线模式时

ReferenceType（位置种类选择）为 _mcCommand 时

DeviatedValue（轴间偏差值）= Master（主轴）的指令当前位置 - Slave（从轴）的指令当前位置

ReferenceType（位置种类选择）为 _mcFeedback 时

DeviatedValue（轴间偏差值）= Master（主轴）的反馈当前位置 - Slave（从轴）的反馈当前位置

旋转模式时

ReferenceType（位置种类选择）为 _mcCommand 时

DeviatedValue（轴间偏差值）为循环计数器的范围内，Master（主轴）的指令当前位置和 Slave（从轴）的指令当前位置中较短的距离。

此时，DeviatedValue（轴间偏差值）的符号从较短的距离看：

- Master（主轴）的指令当前位置 \geq Slave（从轴）指令当前位置，则为“+”
- Master（主轴）的指令当前位置 $<$ Slave（从轴）指令当前位置，则为“-”。

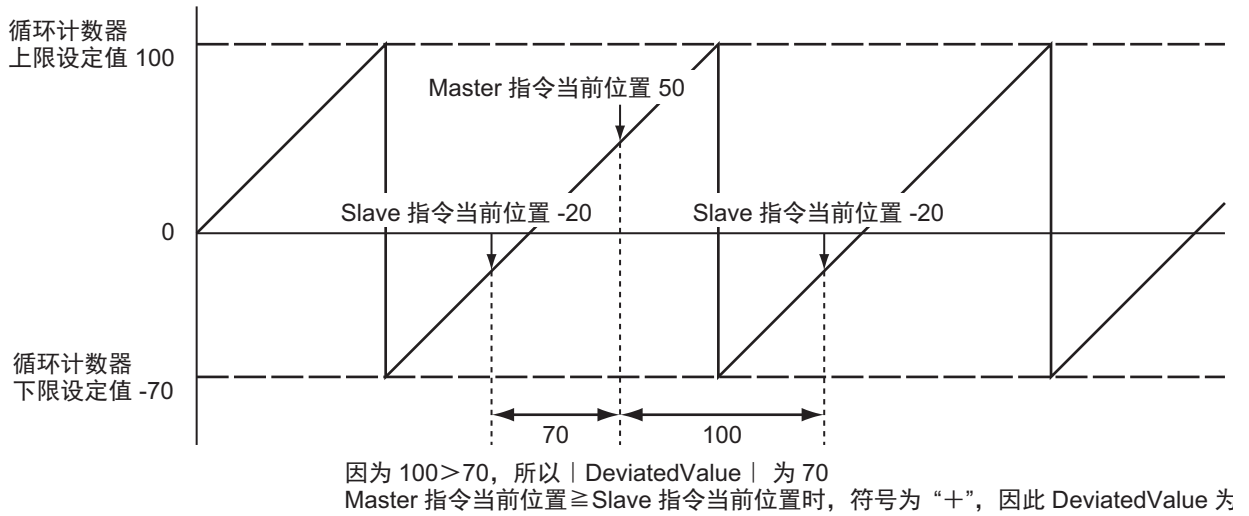
ReferenceType（位置种类选择）为 _mcFeedback 时

DeviatedValue（轴间偏差值）为循环计数器的范围内，Master（主轴）的反馈当前位置和 Slave（从轴）的反馈当前位置中较短的距离。

此时，DeviatedValue（轴间偏差值）的符号从较短的距离看：

- Master（主轴）的反馈当前位置 \geq Slave（从轴）的反馈当前位置，则为“+”
- Master（主轴）的反馈当前位置 $<$ Slave（从轴）的反馈当前位置，则为“-”。

[旋转模式] 下且 ReferenceType (位置种类选择) 为 _mcCommand 时, DeviatedValue (轴间偏差值) 的计算示例

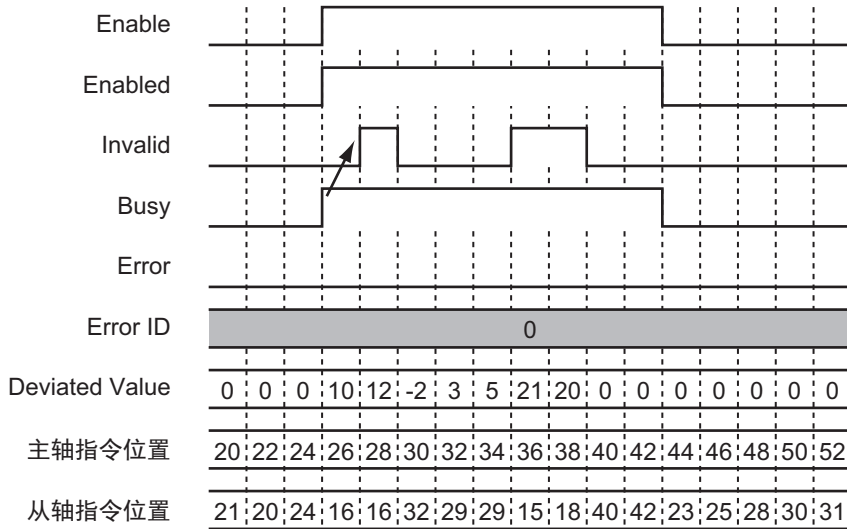


[旋转模式] 下且 ReferenceType (位置种类选择) 为 _mcFeedback 时, 上图的 “指令当前位置” 即为 “反馈当前位置”。

时序图

本指令的异常不会影响轴及组的动作。

PermittedDeviation (轴间偏差容许值) 设定为 “10.0” 时, 时序图如下所示。



运动指令重新执行

输入为 “Enable 型” 的指令没有运动指令重新执行。

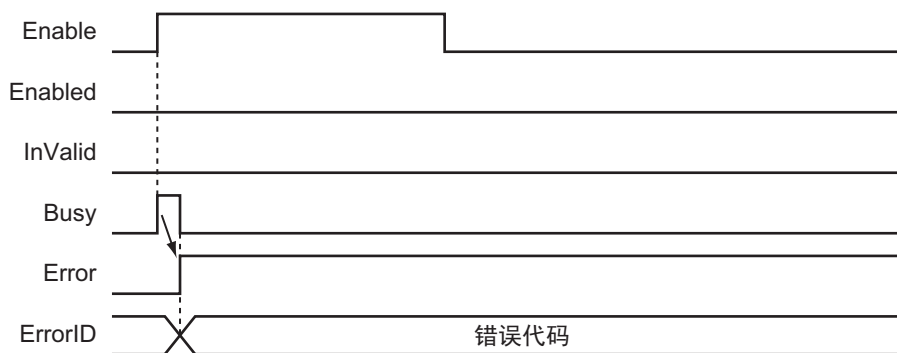
运动指令多重启动

- 无多重启动的限制。

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

异常

- 执行本指令的过程中发生异常时，Error（错误）将变为 TRUE。
- 参照输出到 ErrorID（错误代码）的值，可找出异常的原因。
- 本指令的异常信息将输出到 MC 共通运动变量的轻度故障中。



● 错误代码

指令中发生的异常请参考 □ 「A-1 错误代码一览 (P.A-2)」。

MC_SyncMoveVelocity

将目标速度中设定的值，按任务周期，以周期性同步速度模式（CSV）输出到伺服驱动器。



使用注意事项

任务为主固定周期任务时，这里的任务周期为主周期，为固定周期任务（执行优先级 5）时，任务周期为固定周期任务（执行优先级 5）的任务周期。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_SyncMoveVelocity	周期性同步速度控制	FB		<pre>MC_SyncMoveVelocity_instance (Axis := 《参数》, Execute := 《参数》, Velocity := 《参数》, CmdPosMode := 《参数》, BufferMode := 《参数》, InVelocity => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Velocity	目标速度	LREAL	负数、正数、“0”	0	指定目标速度。 0: 将速度指令值设为“0” 正数: 向正方向移动 负数: 向负方向移动 单位为 [指令单位 /s]。 ^{*1}
CmdPosMode	指令当前位置计数选择	_eMC_CMDPOS_MODE	0: _mcCount	0 ^{*2}	0: 使用反馈当前位置，更新指令当前位置。 保持原点确定状态。
BufferMode	缓冲模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0 ^{*2}	指定运动指令多重启动时的动作。 0: 中断 1: 缓冲

*1. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》的“单位转换设定”。

*2. 如果是有效范围为枚举型的变量初始值，实际不是数值，而是枚举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
InVelocity	达到目标速度	BOOL	TRUE、FALSE	指令速度达到目标速度时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
InVelocity	达到目标速度时	<ul style="list-style-type: none"> • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时
Busy	Execute 的上升沿	<ul style="list-style-type: none"> • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时
Active	开始了指令时	<ul style="list-style-type: none"> • Error 变为 TRUE 时 • CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> • 通过其他指令中止了本指令时 • 其他指令上发生异常，中止了本指令时 • 正在发生轴异常时，启动了本指令 • 正在执行 MC_Stop 指令时，启动了本指令 	<ul style="list-style-type: none"> • Execute 为 TRUE 时，与 Execute 的 FALSE 同时 • Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

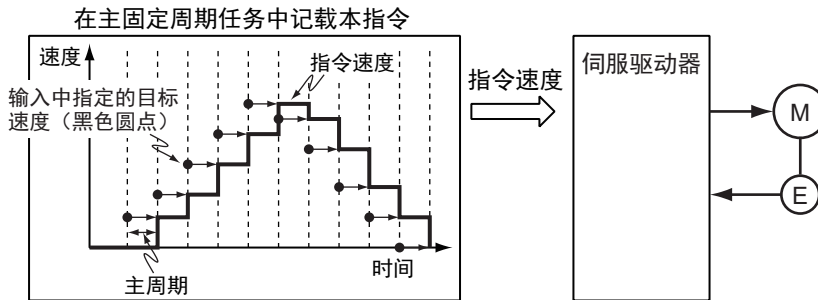
输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。*1

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

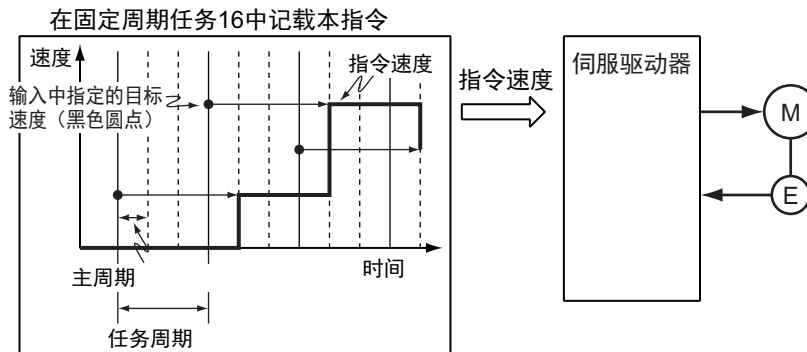
功能说明

- 本指令可将来自用户程序的目标速度，按任务周期，以周期性同步速度模式（CSV）输出到伺服驱动器。
- 在 Execute（启动）的上升沿，在切换伺服驱动器控制模式的同时，输出指令速度。
- 在主固定周期任务或固定周期任务（执行优先级 5）中记述本指令时，将在下一任务周期达到目标速度。

在主固定周期任务中记载本指令的示例如下。固定周期任务（执行优先级 5）中亦同。



- 在固定周期任务 16 中记述本指令时，将在下一固定周期任务中达到目标速度。



使用注意事项

将本指令作为同步控制的主轴使用时的注意事项请参考 □「同步控制的主轴及辅助轴的注意事项 (P.1-6)」。



参考

- 使用 MC_SyncMoveVelocity（周期性同步速度控制）指令时，MC_SetOverride（超驰值设定）指令不会启用。
- NX 系列脉冲输出单元中无法使用本指令。

● 数据对象的映射

使用 MC_SyncMoveVelocity（周期性同步速度控制）指令时，请在 Sysmac Studio 的轴基本设定的 [详细设定] 中，映射以下对象数据。

- 目标速度（60FFHex）
- 操作模式（6060Hex）
- 操作模式显示（6061Hex）

未设定以上内容时，将发生过程数据对象设定不足（错误代码：3461Hex）。

数据对象的映射请参考 □□「2-3 PDO 映射 (P.2-30)」、□□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

指令的详情

下面介绍指令的详情。

● Velocity（目标速度）

输入变量的 Velocity（目标速度）可设定为以“0”为基准的 LREAL 型。

若设定为正数，则向正方向移动，若设定为负数，则向负方向移动。

若设定为“0”，则指令速度为“0”，但 Status.Continuous（连续动作中）保持 TRUE。

每个周期，可通过用户程序设定 Velocity（目标速度）。

指定了与上一周期不同的目标速度时，将适用新的目标速度。与上一周期的值相同时，按上一周期的目标速度动作。



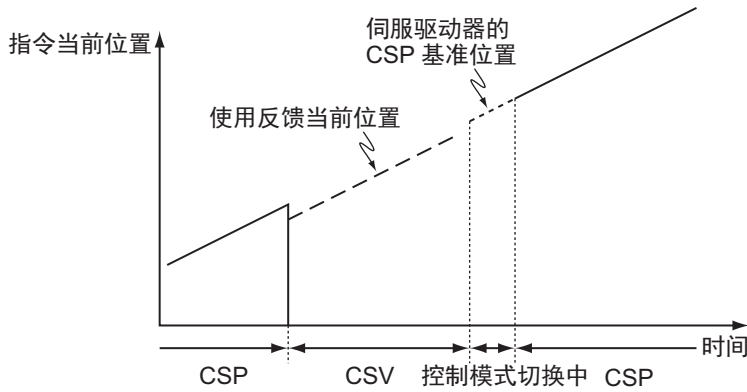
使用注意事项

- 设定目标速度时，请设定为合适的值，确保不会因速度变化而对装置的机械结构造成过大的负担。
- 将使用本指令的轴作为同步控制的主轴使用时，因主轴的目标速度设定不同，从轴可能发生剧烈动作，敬请注意。
- 切换控制模式时，指令当前位置可能会剧烈变化，敬请注意。

● 指令当前位置

若输入变量 CmdPosMode（指令当前位置计数选择）选择为 _mcCount，启动本指令时，指令当前位置变为 1 个周期前的反馈当前位置，本指令结束前，使用反馈当前位置。

在欧姆龙生产的伺服驱动器 G5 系列中，正在切换为 CSP 时，将通过 PDO 发送预先映射的“CSP 基准位置”。完成从 CSV 到 CSP 的切换处理之前，将该基准位置作为指令当前位置使用。完成向 CSP 的切换时，指令当前位置 = 指令位置。



● 使用欧姆龙生产的伺服驱动器 G5 系列时

使用控制模式切换过程中的伺服驱动器“CSP 基准位置”时，请将 CSP 用参照位置（4020 Hex）映射到过程数据中。

在 Sysmac Studio 的 PDO 编辑画面将 CSP 用参照位置（4020 Hex）映射到过程数据中。然后，在轴基本设定的 [详细设定] 中，映射 MC 功能模块的“CSP 切换用基准位置”和 CSP 用参照位置（4020 Hex）。



使用注意事项

将“CSP 用参照位置（4020 Hex）”映射到 PDO 时，请将主周期或固定周期任务（执行优先级 5）的任务周期设定为 1ms 以上。

若主周期或任务周期设定为小于 1ms 的值，伺服驱动器 G5 系列上将发生异常。

详情请参考 [□□《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型 用户手册（SBCE-CN5-365）》](#)。



参考

在欧姆龙产伺服驱动器 G5 系列的直线电机型中，不支持 CSP 用参照位置（4020 Hex）。

● 使用欧姆龙生产的伺服驱动器 G5 系列以外的驱动器时

启动本指令时，或在执行本指令的过程中多重启动了使用 CSP 的指令时，因伺服驱动器的不同，可能在控制模式切换处理时发生异常。

此时，请在轴停止的状态下（速度“0”）启动本指令，或多重启动使用 CSP 的指令。

● 使用 NX 系列位置接口单元时

NX 系列位置接口单元上无法使用本指令。

● 停止处理

下面介绍停止轴动作时的控制模式或指令速度。

进行减速停止时，将以本指令的目标速度为初始速度，使用各停止方法的减速度进行减速停止。

利用 MC_ImmediateStop（立即停止）指令停止

指令速度设为“0”。以下切换判定式成立时，切换为 CSP。

利用 MC_Stop（强制停止）指令停止

指令的减速度，指令速度设为“0”。以下切换判定式成立时，切换为 CSP。

因发生轻度故障等级的异常而停止

各异常规定的减速度，指令速度设为“0”。以下切换判定式成立时，切换为 CSP。

因发生全部停止故障等级及部分停止故障等级的异常而停止

指令速度设为“0”。以下切换判定式成立时，切换为 CSP。

但是，因异常等级不同，可能无法正常执行控制模式的切换，而以 CSV 停止。

因伺服 OFF 而停止

按照规定的方法，将指令速度设为“0”。不执行控制模式的切换处理。

CPU 单元的动作模式切换为程序模式时的停止

按照规定的方法，将指令速度设为“0”。以下切换判定式成立时，切换为 CSP。

● 切换判定式

是否可切换控制模式，取决于伺服驱动器的规格。

停止指令的启动或因发生异常而正在进行停止处理时，为切实切换控制模式，请在确认伺服电机已完全减速后，在切换到 CSP。

指令速度变为“0”后，在固定周期任务中，反馈当前速度连续 3 次满足以下判定式时，将切换为 CSP。

$$\text{反馈当前速度} \leq (\text{最高速度}) \times 0.1$$



使用注意事项

这里的固定周期任务表示主固定周期任务或固定周期任务（执行优先度 5）。

● 恢复为周期性同步位置模式

受停止的影响，可能无法正常切换为 CSP。

例如，运动控制功能模块发生部分停止故障等级的异常时，可能发生以上现象。

因此，MC_Power（可运行）指令的输出变量“Status（可运行）”变为 FALSE 时，将执行切换为 CSP 的处理。

● 控制模式切换失败时的动作

对伺服驱动器发出控制模式切换指令并经过 1 秒后，如果伺服驱动器仍未完成切换，将发生“伺服驱动器控制模式切换异常（错误代码：7439 Hex）”。

此外，对伺服驱动器发出控制模式切换指令，指令速度变为“0”并经过 10 秒后，如果切换判定式仍未成立，与上述相同，将发生“伺服驱动器控制模式切换异常（错误代码：7439 Hex）”。

若发生“伺服驱动器的控制模式切换异常（错误代码：7439 Hex）”，则指令速度变为“0”，伺服 OFF（自由运行停止）。

“伺服驱动器控制模式切换异常（错误代码：7439 Hex）”请参考 □□ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 可使用的轴和启动条件

- 可在伺服轴中使用本指令。
使用时，请将 MC_Power（可运行）指令的 Enable（有效）设为 TRUE（伺服 ON 状态）。
- 虚拟伺服轴始终可接收。
但是，不可执行伺服驱动器的控制模式切换处理。
- 编码器轴、虚拟编码器轴中，启动时会发生异常。

● 轴变量的状态

轴变量的状态 Status.Continuous（连续动作中）变为 TRUE。

此外，要确认伺服驱动器中设定的控制模式时，请确认轴变量 DrvStatus（伺服驱动器状态）。下面表示伺服驱动器状态。

变量名称	数据类型	名称	说明
CSP	BOOL	周期性同步位置 (CSP) 模式中	伺服 ON 且 CSP 模式时，变为 TRUE。
CSV	BOOL	周期性同步速度 (CSV) 模式中	伺服 ON 且 CSV 模式时，变为 TRUE。
CST	BOOL	周期性同步扭矩 (CST) 模式中	伺服 ON 且 CST 模式时，变为 TRUE。

● 原点状态

CmdPosMode（指令当前位置计数选择）为 _mcCount 时，继续保持原点确定状态，无变化。

● 超驰

超驰对本指令无效。

● 软件限位

软件限位对本指令有效。

在轴参数中选择了以下设定时，也有效。

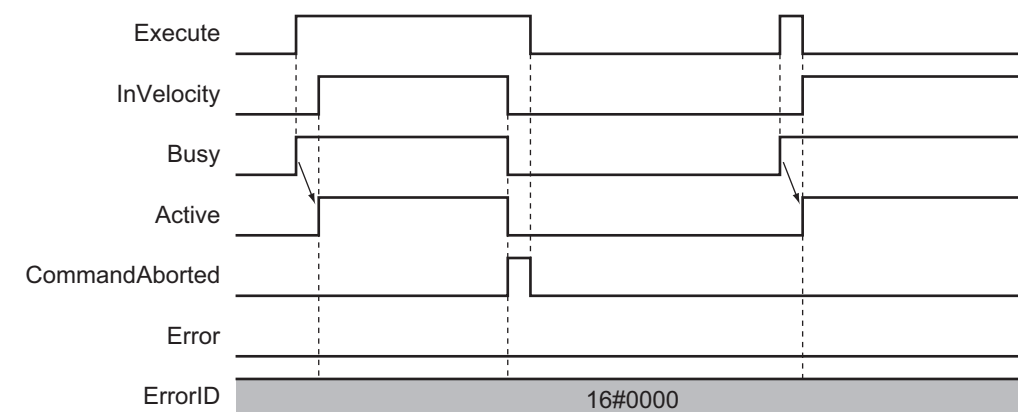
- 对指令位置有效。减速停止。
- 对指令位置有效。进行滞留脉冲停止。

时序图

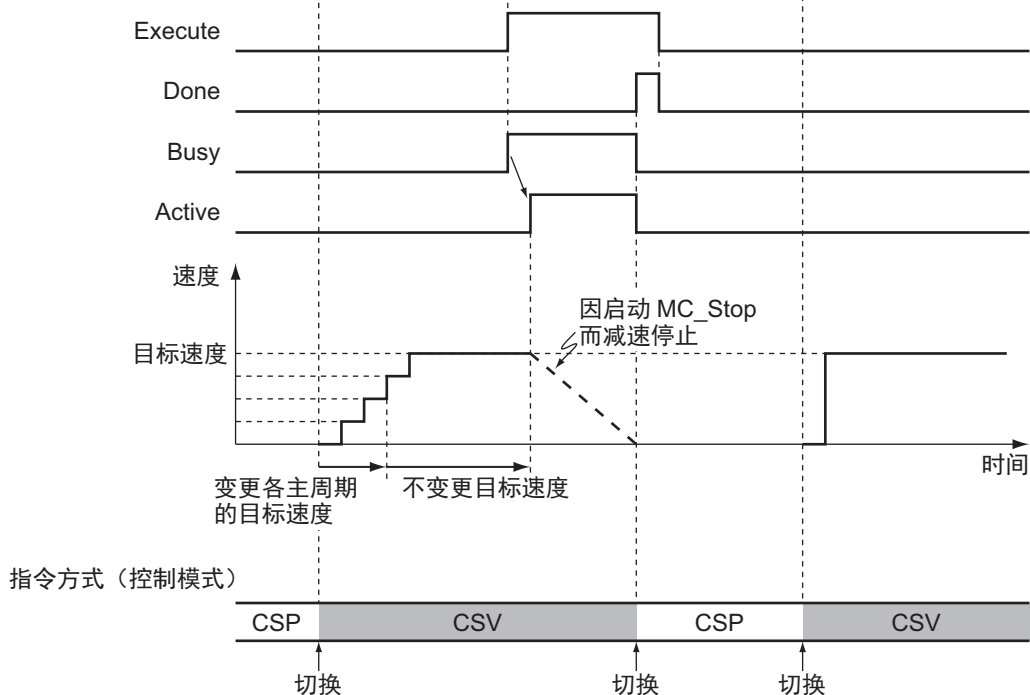
- 在 Execute (启动) 上升的同时 Busy (执行中) 变为 TRUE, 在下一周期, Active (控制中) 变为 TRUE。
- 指令速度达到 Velocity (目标速度) 后, InVelocity (达到目标速度) 变为 TRUE。
- 通过其他指令中止了本指令时, CommandAborted (执行中断) 变为 TRUE, Busy (执行中)、Active (控制中)、InVelocity (达到目标速度) 变为 FALSE。
- 要结束本指令, 请启动 MC_Stop (强制停止) 指令。

在主固定周期任务中记载本指令时如下所示。

MC_SyncMoveVelocity 指令



MC_Stop 指令



参考

MC 功能模块将对伺服驱动器按上述时序图发出控制模式指令。但是，反映到伺服驱动器的时间因各伺服驱动器的规格不同而异。

运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

运动指令多重启动的详情请参考 □□《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 其他指令执行过程中本指令的启动

与 MC_MoveVelocity（速度控制）指令相同，支持 BufferMode（缓冲模式选择）的中断和缓冲。此外，多重启动的指令开始处理时，执行控制模式的切换。

BufferMode（缓冲模式选择）请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 本指令执行过程中其他指令的启动

与 MC_MoveVelocity（速度控制）指令相同，支持 BufferMode（缓冲模式选择）的中断和缓冲。缓冲会在 InVelocity（达到目标速度）变为 TRUE 后，启动其他指令。

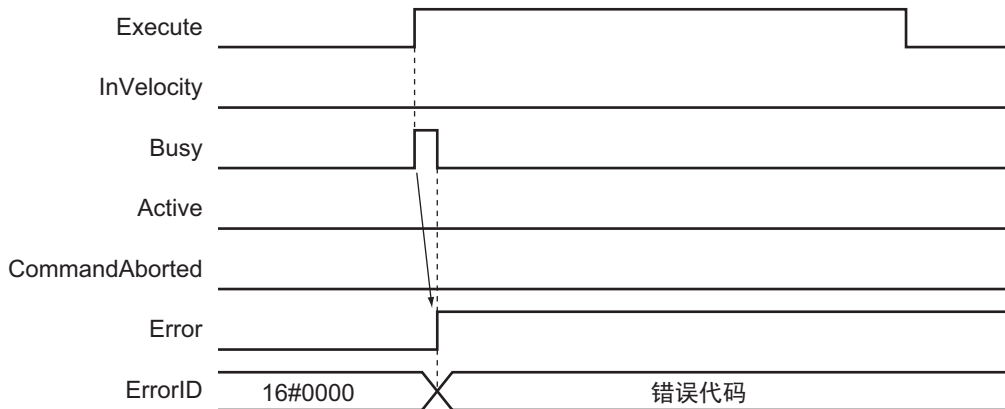
此外，多重启动的指令开始处理时，执行控制模式的切换。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 □□「A-1 错误代码一览 (P.A-2)」。

MC_SyncMoveAbsolute

周期性输出轴中指定的目标位置。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_SyncMove Absolute	周期性同步绝对定位	FB		<pre>MC_SyncMoveAbsolute_instance (Axis := 《参数》, Execute := 《参数》, Position := 《参数》, Direction := 《参数》, BufferMode := 《参数》, InPosition => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

版本相关信息

本指令可在 CPU 单元版本 1.03 以上和 Sysmac Studio Ver.1.04 以上的组合中使用。

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Position	目标位置	LREAL	负数、正数、“0”	0	指定绝对坐标的目标位置。 单位为 [指令单位]。 ^{*1}
Direction	方向选择	_eMC_DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	1 ^{*2}	计数模式为 [旋转模式] 时，指定旋转方向。 ^{*3} 0: 正方向指定 1: 最短距离指定 2: 负方向指定 3: 当前方向指定 4: 无方向指定
BufferMode	缓冲模式选择 ^{*2}	_eMC_BUFFER_MODE	0: _mcAborting	0 ^{*3}	指定运动指令多重启动时的动作。 0: 中断

*1. 指令单位请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. 如果是有效范围为枚举型的变量初始值，实际不是数值，而是列举值。

*3. 旋转方向的指令内容请参考 □□「Direction (方向选择) (P.3-51)」。

输出变量

输出变量	名称	数据类型	有效范围	内容
InPosition	位置	BOOL	TRUE、FALSE	反馈当前位置进入目标位置的位置宽度内时变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*1	发生异常时，输出错误代码。16#0000 表示正常。

*1. □□ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
InPosition	反馈当前位置进入目标位置的位置宽度内时	<ul style="list-style-type: none"> 反馈当前位置超出位置宽度时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	开始了指令时	<ul style="list-style-type: none"> Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动，本指令中止时 其他指令上发生异常，中止了本指令时 正在发生轴异常时，启动了本指令 正在执行 MC_Stop 指令时，启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

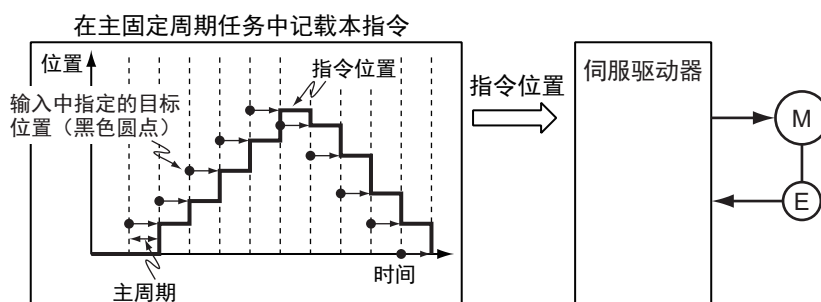
输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。*1

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

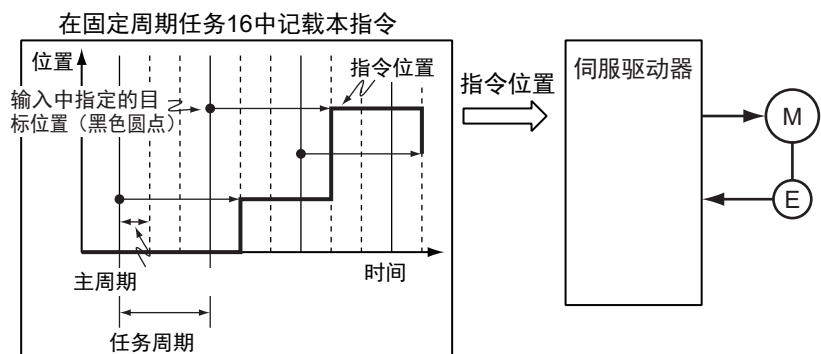
功能说明

- 本指令可将来自用户程序的目标位置，按任务周期，以周期性同步位置模式（CSP）输出到伺服驱动器。目标位置以绝对位置指定。
- 速度以轴参数的 [最高速度] 为上限。不会应用 [最大加速度] 和 [最大减速度]。
- 在主固定周期任务或固定周期任务（执行优先级 5）中记述本指令时，输入中指定的目标位置将在下一任务周期输出到伺服驱动器。

在主固定周期任务中记载本指令的示例如下。固定周期任务（执行优先级 5）中亦同。



- 在固定周期任务 16 中记述本指令时，输入中指定的目标位置将在下一固定周期任务中输出到伺服驱动器。



使用注意事项

指定目标位置时，请确保与指示的目标位置相应的移动量不会超出轴参数的 [最高速度]。如果指定了超出 [最高速度] 的目标位置，将变为指令速度饱和，输出 [最高速度] 中限制的移动量。不足指示的目标位置的移动量指示将在下一周期后输出。此时，轴控制状态的 Details.VelLimit（指令速度饱和）将变为 TRUE。

指令的详情

下面介绍指令的详情。

● 位置检查

不更新 Position（目标位置）时，目标位置和反馈位置的宽度进入轴参数 [位置宽度] 的范围内后，InPosition（位置）变为 TRUE。

InPosition（位置）为 TRUE 时，即使变更目标位置，该周期中将保持 TRUE，在下一周期变为 FALSE。轴参数的 [位置检查时间] 设定变为无效。

● 停止处理

下面介绍停止轴动作时的动作。

停止动作时，应使用 MC_Stop（强制停止）指令或 MC_ImmediateStop（立即停止）指令。若启动这些指令，本指令的 CommandAborted（执行中断）将变为 TRUE。

利用 MC_Stop（强制停止）指令停止

立即停止。

利用 MC_ImmediateStop（立即停止）指令停止

按照轴参数设定 [立即停止输入停止方法] 的指定执行立即停止。

● 因发生异常而停止

发生使轴停止的异常时，无论设定如何，都“立即停止”。

● 可使用的轴

- 可在伺服轴中使用本指令。
使用时，请将 MC_Power（可运行）指令的 Enable（有效）设为 TRUE（伺服 ON 状态）。
- 虚拟伺服轴始终可接收。
- 编码器轴、虚拟编码器轴中，启动时会发生异常。

● 轴变量的状态

轴变量的轴状态 Status.Discrete（定位动作中）变为 TRUE。

不会影响轴控制状态。

● 超驰

超驰对本指令无效。

● 当前位置变更

利用 MC_SetPosition（当前位置变更）指令变更了当前位置时，输出从变更后的当前位置到 Position（目标位置）中指定的目标位置的移动量。

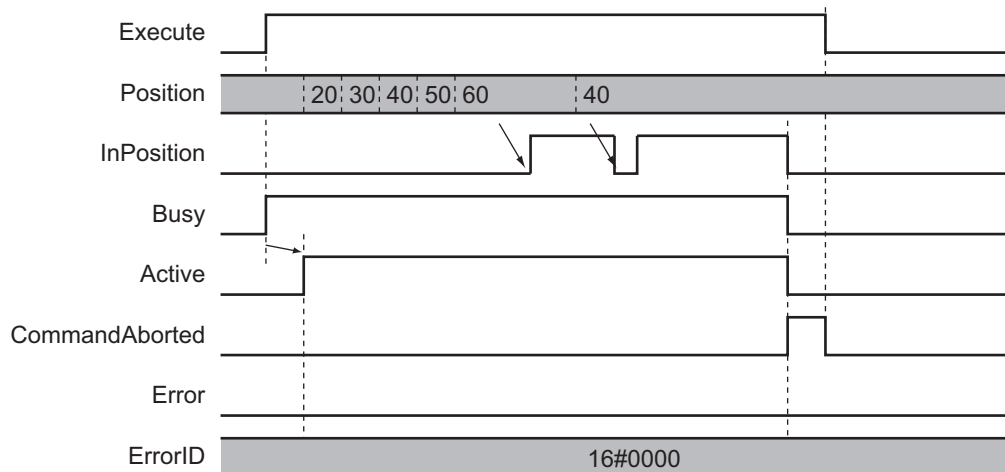
变更当前位置前 InPosition（位置）为 TRUE 时，InPosition（位置）将在当前位置变更后的下一周期变为 FALSE。

时序图

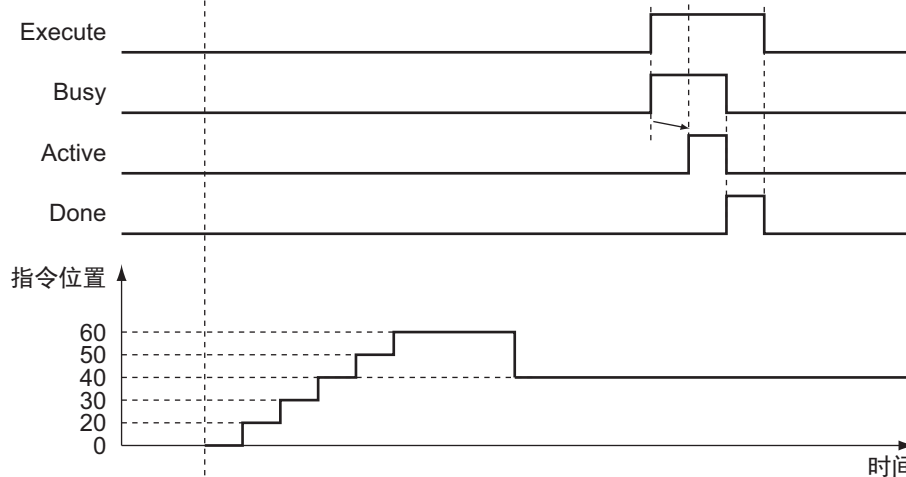
- 在 Execute (启动) 上升的同时 Busy (执行中) 变为 TRUE, 在下一周期, Active (控制中) 变为 TRUE。
- 反馈当前位置进入 Position (目标位置) 的位置宽度内时, InPosition (位置) 变为 TRUE。
- 通过其他指令中止了本指令时, CommandAborted (执行中断) 变为 TRUE, Busy (执行中)、Active (控制中)、InPosition (位置) 变为 FALSE。
- 要结束本指令, 请启动 MC_Stop (强制停止) 指令。

在主固定周期任务中记载本指令时如下所示。

MC_SyncMoveAbsolute 指令



MC_Stop 指令



参考

MC 功能模块将对伺服驱动器按上述时序图发出控制模式指令。但是, 反映到伺服驱动器的时间因各伺服驱动器的规格不同而异。

运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

运动指令多重启动的详情请参考 □□《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 其他指令执行过程中本指令的启动

对当前正在执行的指令启动本指令后，可切换为本指令或缓冲本指令。

每个轴最多可缓冲 1 个。

在 BufferMode（缓冲模式选择）中指定多重启动时本指令的动作。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 切换指令后如果轴的动作方向发生反转，将按照轴参数 [反转时动作] 执行反转动作。

BufferMode（缓冲模式选择）请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 本指令执行过程中其他指令的启动

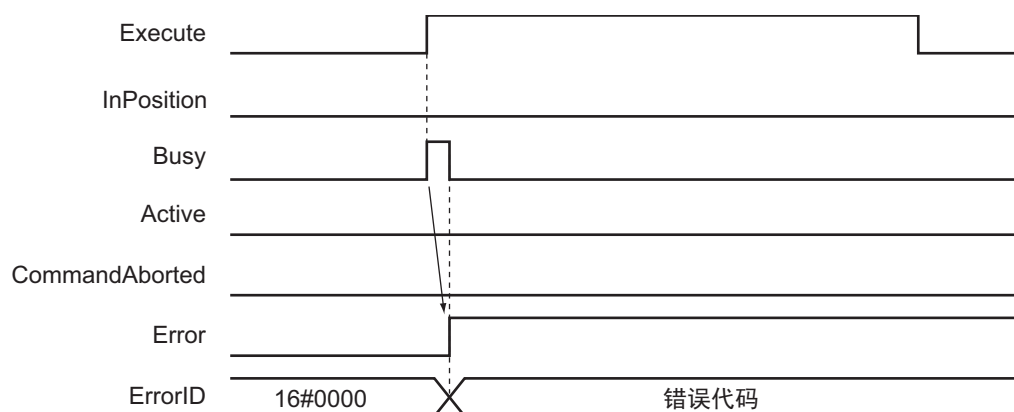
对本指令执行运动指令多重启动时，可进行中断。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 □□「A-1 错误代码一览 (P.A-2)」。

MC_Reset

解除轴异常。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_Reset	轴错误 复位	FB		<pre>MC_Reset_instance (Axis := 《参数》, Execute := 《参数》, Done => 《参数》, Busy => 《参数》, Failure => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、 FALSE	FALSE	上升时开始指令。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、 FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
Failure	错误结束	BOOL	TRUE、 FALSE	未正常执行指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	异常解除处理正常结束时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 Failure 变为 TRUE 时
Failure	<ul style="list-style-type: none"> 轴因为异常而减速停止的过程中, 执行了异常解除时 因发生轴共通异常而发生轴异常的过程中, 执行了异常解除时 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis***”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

功能说明

- 在 Execute（启动）的上升沿，开始 Axis（轴）中指定的轴异常解除处理。
异常解除处理是指轴的异常解除，以及驱动器上发生异常时，驱动器错误的重置。
此外，无论轴的运行状态如何，都将执行异常解除处理。
- 本指令可对所有轴种类使用。
- 进行发生异常的轴执行异常解除处理。
- 对于发生驱动器错误的轴，执行驱动器错误重置处理后，执行异常解除处理。
在清除驱动器错误或轴参数 [驱动器错误重置监视时间] 的时间内，驱动器错误重置处理将继续执行。
- 因错误减速停止中而导致轴减速停止中时，若执行了本指令，将不执行，Failure（错误结束）变为 TRUE。这是为了在轴停止前，使异常无法解除。
此外，因发生 MC 共通异常而导致的轴异常也无法通过本指令解除，因此 Failure（错误结束）将变为 TRUE。
MC 共通异常包括“MC 共通部分停止故障”及“MC 共通轻度故障”。
- 解除对象为 Execute（启动）上升时发生的异常。解除过程中发生的异常不会解除。



使用注意事项

- 本指令的异常解除处理可能跨多个控制周期。
- 发生“MC 共通 部分停止故障”及“MC 共通 轻度故障”的共通异常或轴正在动作时，本指令的 Failure（错误结束）将变为 TRUE。排除异常原因后，请在变为 Done（完成）前，执行重试处理。
- 排除异常发生原因时，请在确认轴完全停止后再操作。
要确认轴是否完全停止，需要确认轴变量 Act.Vel（反馈当前速度）已变为“0”。
- 对欧姆龙生产的伺服驱动器 G5 系列使用本指令时，请进行排他处理，避免同时执行 ResetECError（EtherCAT 异常解除）指令。
- 对 NX 系列脉冲输出单元使用本指令时，脉冲输出单元上连接的驱动器的异常状态不会解除。详情请参考 □《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。



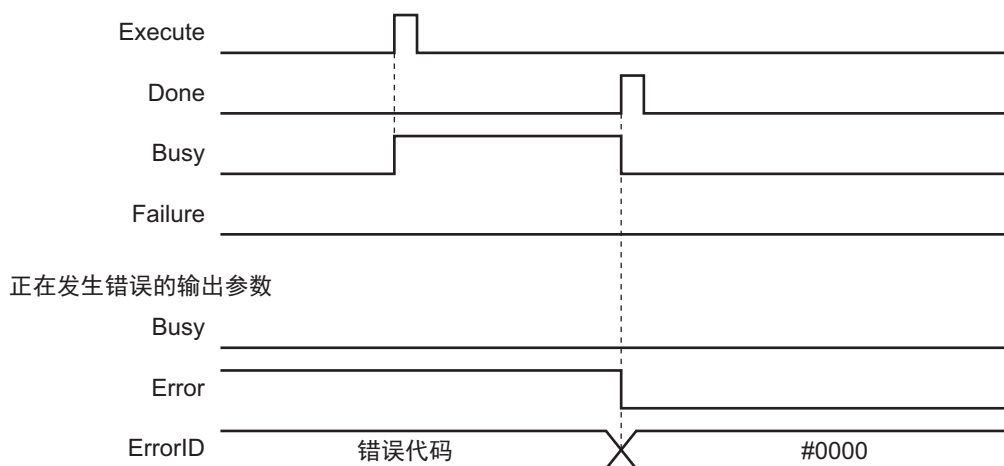
参考

以下异常无法通过本指令解除。

- 轴共通异常 整体：请执行 ResetMcError（全部错误重置）指令。
- 轴组共通异常 整体：请执行 MC_GroupReset（轴组错误重置）指令。

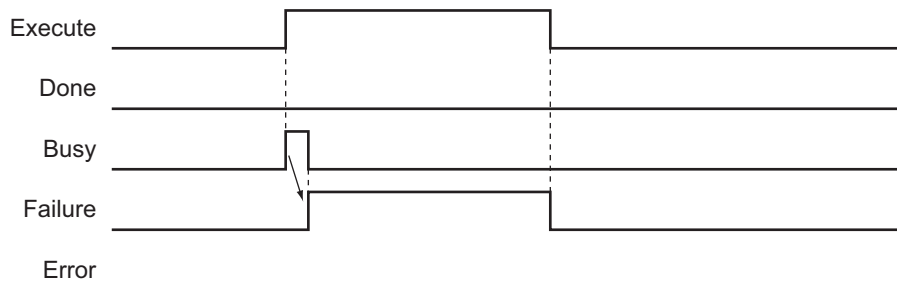
此外，即使执行本指令，从站通信异常等网络相关的异常原因也不会解除。请执行 ResetECATError 指令。

时序图



指令的中止

无法解除因异常而导致的轴减速停止异常或因发生轴共通异常而导致的轴异常时，中断指令。



正在发生错误的输出参数



错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

MC_ChangeAxisUse

临时切换轴参数中的 [轴使用]。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_ChangeAxisUse	轴使用变更	FB		<pre>MC_ChangeAxisUse_instance (Axis := 《参数》, Execute := 《参数》, AxisUse := 《参数》, Done => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>



使用注意事项

通过本指令切换的值不会保存到 CPU 单元的非易失性存储器中。因此，控制器的电源“OFF”及开始下载和 MC 功能模块的重新启动处理时，切换的值将消失，恢复为 Sysmac Studio 中设定的值。

要保存到非易失性存储器时，请使用 Sysmac Studio 传送参数。



参考

- “下载”时使用 Sysmac Studio 的“同步”功能。
- 使用本指令的应用示例请参考 [《NJ/NX 系列 CPU 单元 用户手册 软件篇 \(SBCA-CN5-359\)》](#)。



版本相关信息

本指令可在 CPU 单元版本 1.04 以上和 Sysmac Studio Ver.1.05 以上的组合中使用。

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
AxisUse	轴使用	_eMC_AXIS_USE	1:_mcUnusedAxis 2:_mcUsedAxis	1*1	指定使用轴或未使用轴。 1: 未使用轴 2: 使用轴

*1. 如果是有效范围为列举型的变量初始值，实际不是数值，而是列举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* □□ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	指令完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	其他指令上发生异常，中止了本指令时	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*]、_MC1_AX[*]、_MC2_AX[*]）。

功能说明

- 在 Execute（启动）的上升沿，切换为 AxisUse（轴使用）中指定的轴使用。
- 指令完成执行的同时，轴变量 `_MC_AX[*].Cfg.AxEnable`（轴使用）将切换为指定的轴使用。
- 可切换的轴为轴参数的 [轴使用] 设定为 [未使用轴（可切换为使用轴）] 或 [使用轴] 的轴。请勿将设定为 [未使用轴（不可切换为使用轴）] 的轴切换为 [使用轴]。
- 在原点确定和原点未确定的任一状态下，均可执行指令。若在原点确定时切换为未使用轴，将变为原点未确定。
- 若将使用绝对编码器的轴从使用轴切换为未使用轴，CPU 单元电源 OFF 时，CPU 单元内置的电池备份存储器中保存的 [绝对编码器原点位置偏置] 为切换为未使用轴前的值。



使用注意事项

- 轴编号超出各型号 CPU 单元的最大控制轴数的轴无法切换为使用轴。此外，可切换为使用轴的实轴数最多为使用实轴最大数。

规格项目	NX701-	
	17 □□	16 □□
可切换为使用轴的轴编号	0 ~ 255	0 ~ 127
最大使用实轴数	256 轴	128 轴

规格项目	NJ501-			NJ301-		NJ101-	
	□ 5 □□	□ 4 □□	□ 3 □□	12 □□	11 □□	10 □□	90 □□
可切换为使用轴的轴编号	0 ~ 63	0 ~ 31	0 ~ 15	0 ~ 14 ^{*1}	0 ~ 14 ^{*2}	0 ~ 5	无
最大使用实轴数	64 轴	32 轴	16 轴	8 轴	4 轴	2 轴	0 轴

*1. 单元版本 1.05 以下的 CPU 单元中，为“0 ~ 7”。

*2. 单元版本 1.05 以下的 CPU 单元中，为“0 ~ 3”。

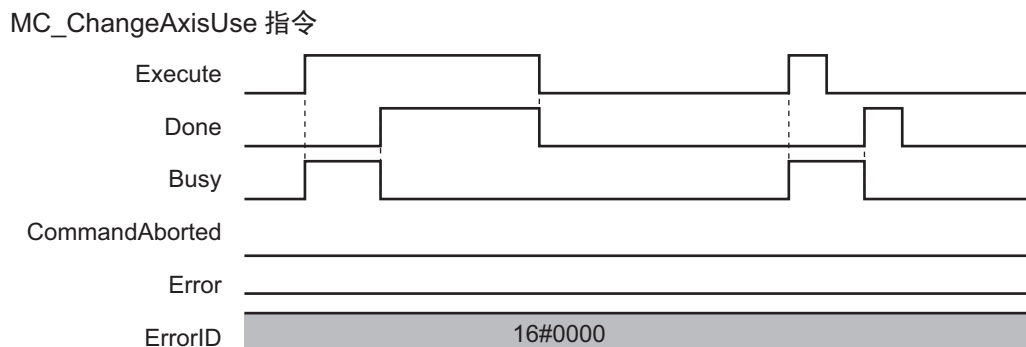
- 仅在轴变量 `_MC_AX[*].Status.Disabled`（轴无效）或 `_MC_AX[*].Status.Standstill`（停止中）为 TRUE 时可以执行指令。若在 FALSE 时执行，将发生异常。
- 若在轴变量 `_MC_AX[*].Details.VelLimit`（指令速度饱和）为 TRUE 时执行，将发生异常。
- 对轴使用为 `_mcUnusedAxis`（未使用轴）的轴，通过 `MC_write`（MC 设定写入）指令重写轴参数时，视为无效。请将轴使用切换为 `_mcUsedAxis`（使用轴）后，再执行 `MC_write`（MC 设定写入）指令。
- 若对轴组（包含通过本指令切换为未使用轴的轴）执行 `MC_GroupEnable`（轴组有效）指令，将发生异常。
- 对使用轴执行指令的过程中发生异常时，将发生轴异常，对未使用轴执行指令的过程中发生异常时，将发生 MC 共通异常。
- 通过本指令切换轴使用后，执行其他运动控制指令时，请在确认本指令的输出变量“Done（完成）”为 TRUE 后再执行。



参考

NX 系列 CPU 单元时，以 `_MC_AX[*]` 开始的变量名称可能是 `_MC1_AX[*]`、`_MC2_AX[*]`。

时序图



运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

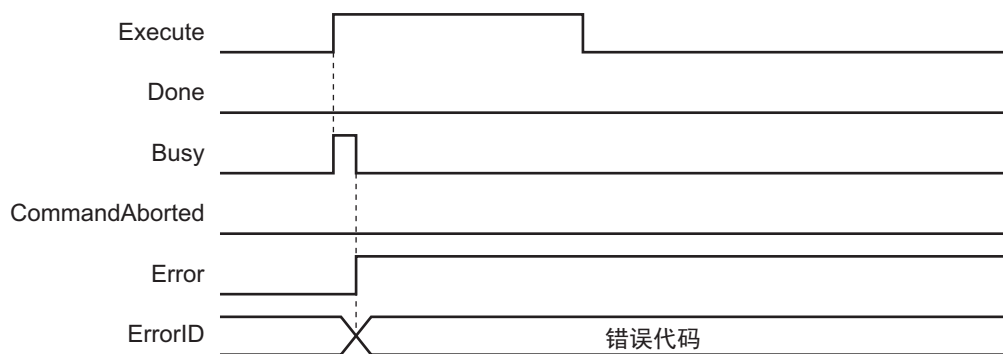
运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

异常

如果在启动本指令时发生异常，Error（错误）将变为 TRUE，轴使用不会切换。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 □ 「A-1 错误代码一览 (P.A-2)」。



使用注意事项

对使用轴执行指令的过程中发生异常时，将发生轴异常，对未使用轴执行指令的过程中发生异常时，将发生 MC 共通异常。

MC_DigitalCamSwitch

根据轴的位置将数字输出设为 ON 或 OFF。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_DigitalCamSwitch	数字凸轮开关有效	FB		<pre>MC_DigitalCamSwitch_instance (Axis := 《参数》, Switches := 《参数》, Outputs := 《参数》, TrackOptions := 《参数》, Enable := 《参数》, EnableMask := 《参数》, ValueSource := 《参数》, InOperation => 《参数》, Busy => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>



安全要点

- 请务必在很低速度的轴上使用 MC_DigitalCamSwitch（数字凸轮开关有效）指令。
- 请在确认 MC_DigitalCamSwitch（数字凸轮开关有效）指令的 InOperation（操作中）变为 TRUE 后，使用 NX_AryDOutTimeStamp（时间戳数字输出排列写入）指令。



使用注意事项

- 本指令对分配有 NX 系列位置接口单元的轴有效。可使用的 NX 单元为 NX-EC0 □□□、NX-ECS □□□。
- 本指令请务必与 NX_AryDOutTimeStamp 指令及支持时间戳方式的数字输出单元组合使用。



版本相关信息

本指令可在 CPU 单元版本 1.06 以上和 Sysmac Studio Ver.1.07 以上的组合中使用。

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、FALSE	FALSE	TRUE 时，执行指令。FALSE 时，不更新“Outputs”的值。 ^{*1}
EnableMask	轨道有效	WORD	16#0000 ~ FFFF	16#0000	指定各轨道的有效或无效。最多 16 个轨道，第 0 位的值为轨道编号 0 的有效无效，第 15 位为轨道编号 15 的有效无效。 0: 无效 ^{*2} 1: 有效
ValueSource (Reserved)	输入信息	_sMC_SOURCE	—	—	(Reserved)

*1. Enable 为 FALSE 时，保持 Outputs（输出信号）的值。此外，NX_AryDOutTimeStamp 指令的 Enable 为 FALSE 时，数字输出单元的数字输出变为 OFF。

*2. EnableMask 的位列为“0”时，相应轨道编号的 Outputs（输出信号）将变为 OFF。

输出变量

输出变量	名称	数据类型	有效范围	内容
InOperation	动作中	BOOL	TRUE、FALSE	正在输出有效的输出信号时变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* □□ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
InOperation	Enable 变为 TRUE 时	<ul style="list-style-type: none"> • Enable 变为 FALSE 时 • Error 变为 TRUE 时
Busy	Enable 的上升沿	<ul style="list-style-type: none"> • Enable 变为 FALSE 时 • Error 变为 TRUE 时
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定要参照位置的轴。 ^{*1}
Switches	开关	ARRAY[0..255] OF _sCAMSWITCH_REF ^{*2}	—	将开关结构体 _sCAMSWITCH_REF 型排列变量作为开关的 ON/OFF 模式数据指定。排列要素编号表示开关编号。
Outputs	输出信号	ARRAY[0..15] OF _sOUTPUT_REF ^{*2 *3}	—	指定输出信号 _sOUTPUT_REF 型的排列变量，作为根据开关的 ON/OFF 模式数据计算得到的数字输出 ON/OFF 时刻的输出目标。排列要素编号表示轨道编号。将该排列变量指定为 NX_AryDOutTimeStamp 指令的输入输出变量，可对实际的数字输出进行 ON/OFF 控制。
TrackOptions	轨道选项	ARRAY[0..15] OF _sTRACK_REF ^{*2 *3}	—	将轨道选项结构体 _sTRACK_REF 型排列变量作为开关的动作条件指定。排列要素编号表示轨道编号。

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

*2. 排列要素的开始编号为“0”以外时，将发生异常。此外，2 维以上的排列时，也将发生异常。

*3. Outputs 的要素数和 TrackOptions 的要素数不一致时，将发生异常。

功能说明

- 本指令会把轴反馈位置到达“Switches”中设定的位置的时刻，作为数字输出进行 ON/OFF 的时刻输出。
- 本指令请务必与 NX_AryDOutTimeStamp 指令及支持时间戳方式的数字输出单元组合使用。
- NX_AryDOutTimeStamp 指令将在指定的时间戳执行指定数字输出的 ON/OFF。使用时间戳方式，不会受控制处理时间的影响，可在任意时间点控制数字输出信号的 ON/OFF。



使用注意事项

- 仅凭本指令无法控制数字输出的 ON/OFF。
- 本指令需要使用 NX 系列编码器输入单元，且变化时刻获取功能处于运行状态。以下情况下，变化时刻获取功能不运行。
 - 使用不支持变化时刻获取功能的编码器输入或使用伺服驱动器时
 - NX 系列编码器输入单元的对象 6010Hex（时间戳）未分配到 PDO 时
 - EtherCAT 耦合器单元的 [DC 有效] 设定为 [无效] 时
- 即使不更新时间戳，本指令也不会发生异常。此时，仍会计算 ON/OFF 时刻，但可能与预计值不同。请通过 Sysmac Studio 的 MC 检视表或监视窗口，确认轴变量的成员“TimeStamp（时间戳）”已更新后，再使用本指令。

- NX_AryDOutTimeStamp 指令及支持时间戳方式的数字输出单元组合使用时，最小 ON/OFF 宽度与任务周期的值及转速的值成比例。
例如，轮盘转 1 圈为 360°、转速为 800r/min、任务周期为 500μs 时，最小 ON/OFF 宽度为 5°。仅将任务周期变更为 1000μs 时，最小 ON/OFF 宽度为 10°。
- 指定开关结构体型变量 FirstOnPosition、LastOnPosition 及 Duration 的值时，请确保数字输出的 ON/OFF 宽度大于最小 ON/OFF 宽度。若小于最小 ON/OFF 宽度，可能实际的数字输出不会变为 ON/OFF。
- 本指令可根据轴的当前位置和当前速度，算出到达指令位置的时间戳。计算得到的时间戳精度受编码器分辨率及轴的转速影响。编码器分辨率较低或轴的转速较低时，误差会变大。最大误差的参考值可按以下公式计算。

时间戳的最大误差 (s)

$$= 180 / [\text{编码器分辨率 (脉冲 / 圈)} \times \text{转速 (r/min)}]$$

下面表示与编码器分辨率和转速对应的时间戳最大误差的示例。

编码器分辨率 (脉冲 / 圈)	转速 (r/min)	计算得到的时间戳的最大误差 (μs)
3600	400	±125.0
	800	±62.5
131072	400	±3.4
	800	±1.7

此外，若轴发生剧烈加减速，算出的误差可能会更大，因此请在轴保持恒定速度的状态下使用。

请充分进行动作验证，确保安全。

- 指定了未使用轴时或正在执行 MC 调试时，若 Enable 为 TRUE，则 Busy 变为 TRUE，InOperation 和 Error 变为 FALSE。
- 请勿配置 2 个以上同名的实例。若配置，可能发生意外输出。

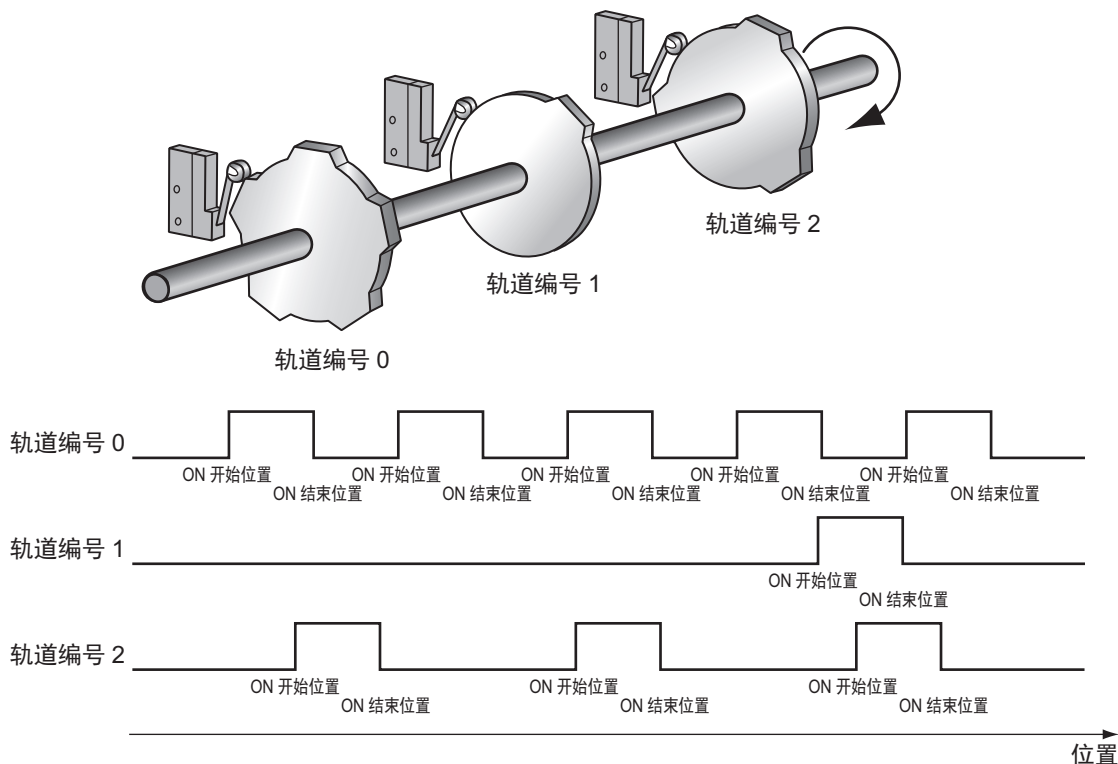


参考

NX_AryDOutTimeStamp 指令请参考 《NJ/NX 系列 指令基准手册 基本篇 (SBCA-CN5-360)》。

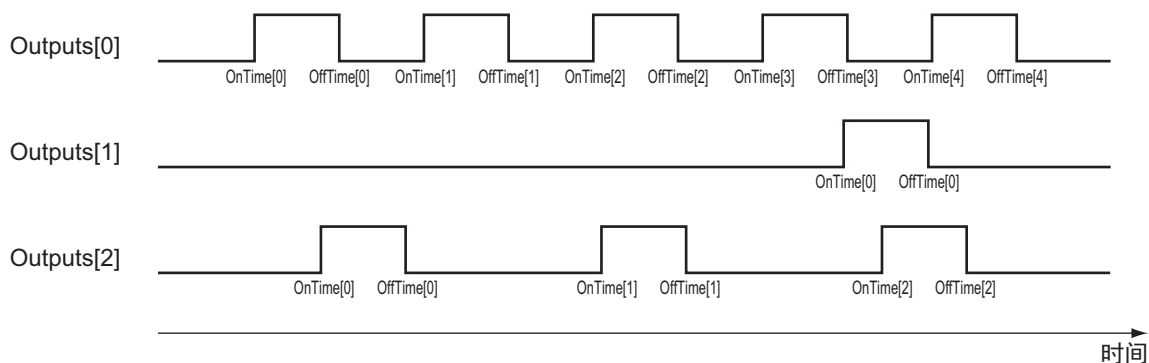
指令的详情

本指令通过控制器的程序，实现根据凸轮的旋转角度来控制传感器输出信号的机械式凸轮开关的功能。1 个轨道相当于 1 个凸轮。



TrackNumber (轨道编号) 相当于凸轮的编号。FirstOnPosition (ON 开始位置)、LastOnPosition (ON 结束位置) 相当于凸轮的形状。

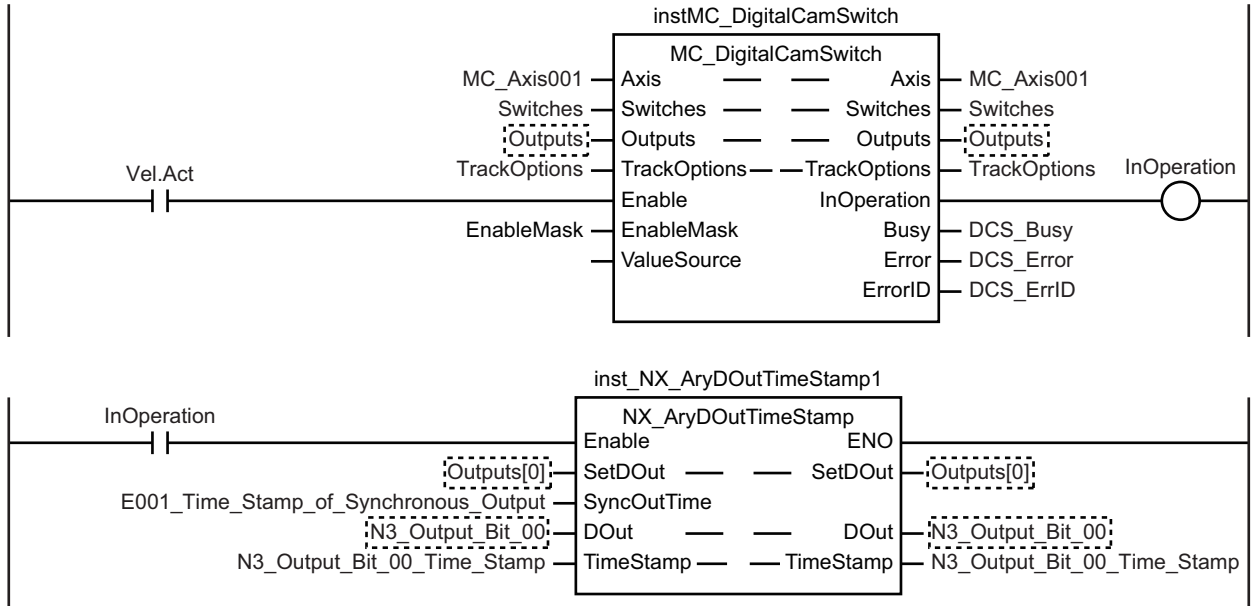
MC_DigitalCamSwitch 指令将计算到达设定的 ON 开始位置 / ON 结束位置的 OnTime (ON 时刻) / OffTime (OFF 时刻)，并保存到 Outputs (输出信号) 中指定的参数中。



NX_AryDOutTimeStamp 指令将根据 SetDOut（输出脉冲）中指定的参数的 OnTime（ON 时刻） / OffTime（OFF 时刻），控制实际数字输出的 ON/OFF。

在 SetDOut（输出脉冲）中指定 MC_DigitalCamSwitch 指令的 Outputs（输出信号）参数中指定的排列变量的要素。

DOut（DOut 单元输出接点）中，请指定为支持时间戳方式的、数字输出单元的输出接点上分配的设备变量，作为实际的数字输出指定。



下面介绍指令的变量。

● **Enable（有效）**

- Enable（有效）为 TRUE 期间，执行指令。FALSE 时，不更新“Outputs”的值。

● **EnableMask（轨道有效）**

- Enable（有效）为 TRUE 时，通过 EnableMask（轨道有效），可指定各轨道的有效或无效。要将数字输出单元的输设为 OFF 时，请将对象轨道设定为无效。
- 第 0 位表示轨道编号 0，第 15 位表示轨道编号 15。若将各位的值设为“1”，则对象轨道变为有效，若设为“0”，则变为无效。若将位的值从“1”变为“0”，则对象轨道的数字输出变为 OFF。
- EnableMask 中指定的值将反映到对应轨道编号的 EnableOut 中。

● 开关结构体（_sCAMSWITCH_REF 型）

开关结构体（_sCAMSWITCH_REF 型）是指指定输出信号的 ON/OFF 模式所需的结构体数据类型。本指令中，可指定最多 256 种 ON/OFF 模式，作为排列变量。1 个轨道中，最多可指定 16 种 ON/OFF 模式。下面介绍开关结构体的成员。

变量	名称	数据类型	有效范围	初始值	内容
TrackNumber	轨道编号	UINT	0 ~ 15 ^{*1}	0	指定对象轨道编号。
FirstOnPosition	ON 开始位置	LREAL	负数、正数、“0”	0	指定输出变为 ON 的位置。 ^{*2}
LastOnPosition	ON 结束位置	LREAL	负数、正数、“0”	0	指定输出变为 OFF 的位置。 ² 开关模式选择指定为位置基准时，变为有效。
AxisDirection	轴方向选择	_eMC_DIRECTION	0:_mcPositiveDirection 2:_mcNegativeDirection 4:_mcNoDirection	0 ^{*3}	指定旋转方向。 0: 正方向指定 2: 负方向指定 4: 无方向指定（双向）
CamSwitchMode	开关模式选择	_eMC_SWITCH_MODE	0:_mcSwitchDisable 1:_mcPositionBased 2:_mcTimeBased	0 ^{*3}	指定开关模式。 0: 开关无效 1: 位置基准 ^{*4} 2: 时间基准 ^{*5}
Duration	ON 时间	TIME	正数、T#0s	T#0s	指定输出变为 ON 的时间。开关模式选择指定为时间基准时，变为有效。

*1. 最大可指定为 Outputs（输出信号）中指定变量的要素编号。

*2. 变为指令单位。指令单位以 [mm]、[μm]、[nm]、[degree]、[inch] 或 [pulse] 表示。

计数模式为 [直线模式] 时，设定范围为转换为脉冲单位后的带符号整数型 40 位（0x8000000000 ~ 7FFFFFFF）。

计数模式为 [旋转模式] 时，设定范围在循环计数器下限值以上、循环计数器上限值以下。

*3. 如果是有效范围为枚举型的变量，实际初始值不是数值，而是列举值。

*4. 指定为 1:_mcPositionBased（位置基准）时，根据 FirstOnPosition（ON 开始位置）和 LastOnPosition（ON 结束位置）的值动作。与 Duration（ON 时间）的值无关。

*5. 指定为 2:_mcTimeBased（时间基准）时，根据 FirstOnPosition（ON 开始位置）和 Duration（ON 时间）的值动作。与 LastOnPosition（ON 结束位置）的值无关。

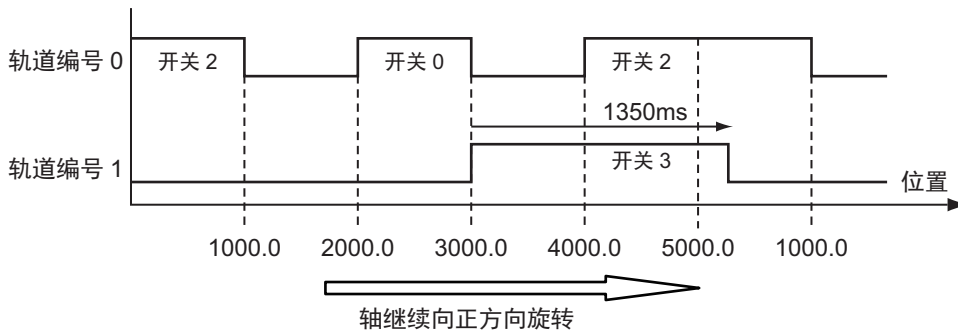
设定例

变量	名称	开关 0	开关 1	开关 2	开关 3	...	开关 255
TrackNumber	轨道编号	0	0	0	1		
FirstOnPosition	ON 开始位置	2000.0	2500.0	4000.0	3000.0		
LastOnPosition	ON 结束位置	3000.0	3000.0	1000.0	_ ^{*1}		
AxisDirection	轴方向选择	正方向指定	负方向指定	无方向指定	无方向指定		
CamSwitchMode	开关模式选择	位置基准	位置基准	位置基准	时间基准		
Duration	ON 时间	_ ^{*2}	_ ^{*2}	_ ^{*2}	T#1350ms		

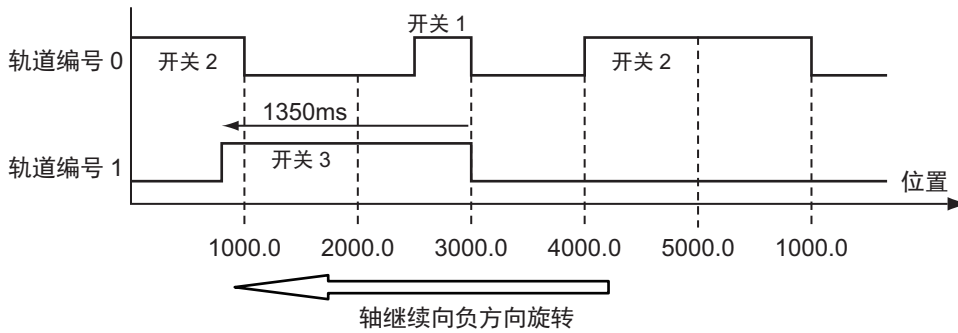
*1. 开关模式选择为时间基准时，根据 ON 开始位置和 ON 时间动作。与 ON 结束位置无关。

*2. 开关模式选择为位置基准时，根据 ON 开始位置和 ON 结束位置动作。与 ON 时间无关。

轴持续向正方向旋转时的动作如下所示。轴参数的计数模式为旋转模式，循环计数器的范围为0~5000。



轴持续向负方向旋转时的动作如下所示。轴参数的设定同上。



● 输出信号结构体（_sOUTPUT_REF 型）

输出信号结构体（_sOUTPUT_REF 型）为结构体数据类型，是指表示根据开关 ON/OFF 模式数据计算得到的数字信号的 ON/OFF 时刻。本指令最多可处理 16 个要素，作为 Outputs（输出信号）的排列变量。Outputs（输出信号）的排列要素编号表示轨道编号。

下面介绍输出信号结构体的成员。

变量	名称	数据类型	有效范围	内容
EnableOut	输出有效	BOOL	TRUE、FALSE	表示该轨道编号的输出有效 / 无效。将反映与 EnableMask 相同编号的位对应的值。 ^{*1} TRUE：该轨道编号的输出为有效 FALSE：该轨道编号的输出为无效
OnTime	ON 时刻	ARRAY[0..15] OF ULINT	正数、“0”	表示数字输出变为 ON 的时间戳。表示以 NX 系列编码器输入单元中的时刻为基准的时间戳。每个任务周期都会更新值。单位为 [ns]。
OffTime	OFF 时刻	ARRAY[0..15] OF ULINT	正数、“0”	表示数字输出变为 OFF 的时间戳。表示以 NX 系列编码器输入单元中的时刻为基准的时间戳。每个任务周期都会更新值。单位为 [ns]。

*1. EnableMask 的 i 位的值将反映到 Outputs[i].EnableOut 中。

● 轨道选项结构体（_sTRACK_REF 型）

轨道选项结构体（_sTRACK_REF 型）为指定开关的动作条件所需的结构体数据类型。本指令中，可指定最多 16 种输出的条件，作为排列变量。

TrackOptions 中指定变量的要素数和 Outputs 中指定变量的要素数请指定为相同的要素数。

下面介绍轨道选项结构体的成员。

变量	名称	数据类型	有效范围	初始值	内容
OnCompensation	ON 时间补偿	TIME	T#-1s ~ T#1s ^{*1}	T#0s	对输出变为 ON 的时间进行补偿。正数时，延迟时间，负数时，提早时间。
OffCompensation	OFF 时间补偿	TIME	T#-1s ~ T#1s ^{*1}	T#0s	对输出变为 OFF 的时间进行补偿。正数时，延迟时间，负数时，提早时间。

*1 计数模式为旋转模式时，根据 CPU 单元的版本，有以下限制。

- 在单元版本 1.09 以上的 CPU 单元中，若设定为超出轴 ± 半圈范围的值，InOperation（操作中）将变为 FALSE。
- 在单元版本 1.08 以下的 CPU 单元中，有效范围为“T#-1s ~ T#0s”。

OnCompensation（ON 时间补偿）及 OffCompensation（OFF 时间补偿）用于补偿机械的微小动作延迟或偏移等。

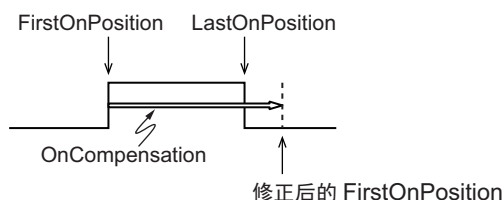


安全要点

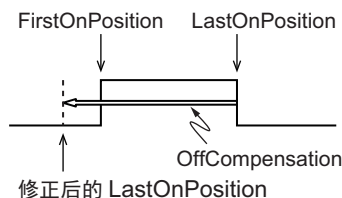
请务必在很定速度的轴上使用 MC_DigitalCamSwitch（数字凸轮开关有效）指令。

若将计数模式设为旋转模式，使用 OnCompensation（ON 时间补偿）或 OffCompensation（OFF 时间补偿），轴速度发生急剧变化，将发生以下动作。

- OnCompensation（ON 时间补偿）或 OffCompensation（OFF 时间补偿）的值超出轴旋转半圈的时间时，InOperation（操作中）将变为 FALSE。
- OnCompensation（ON 时间补偿）的值超出 LastOnPosition（ON 结束位置）时，输出时间将变得不确定。



- OffCompensation（OFF 时间补偿）的值超出 FirstOnPosition（ON 开始位置）时，输出时间将变得不确定。





使用注意事项

将计数模式设为旋转模式，使用 OnCompensation（ON 时间补偿）或 OffCompensation（OFF 时间补偿）时，设定时请确保 FirstOnPosition（ON 开始位置）和 LastOnPosition（ON 结束位置）的位置关系不会反转。

输出时序变得不确定。

动作请参考前述的“安全要点”。



版本相关信息

计数模式为旋转模式时，根据 CPU 单元的版本，有以下限制。

在单元版本 1.09 以上和 1.08 以下的 CPU 单元中，有效范围不同。将 CPU 单元升级到 1.09 以上时，请注意有效范围。

单元版本 1.09 以上的 CPU 单元

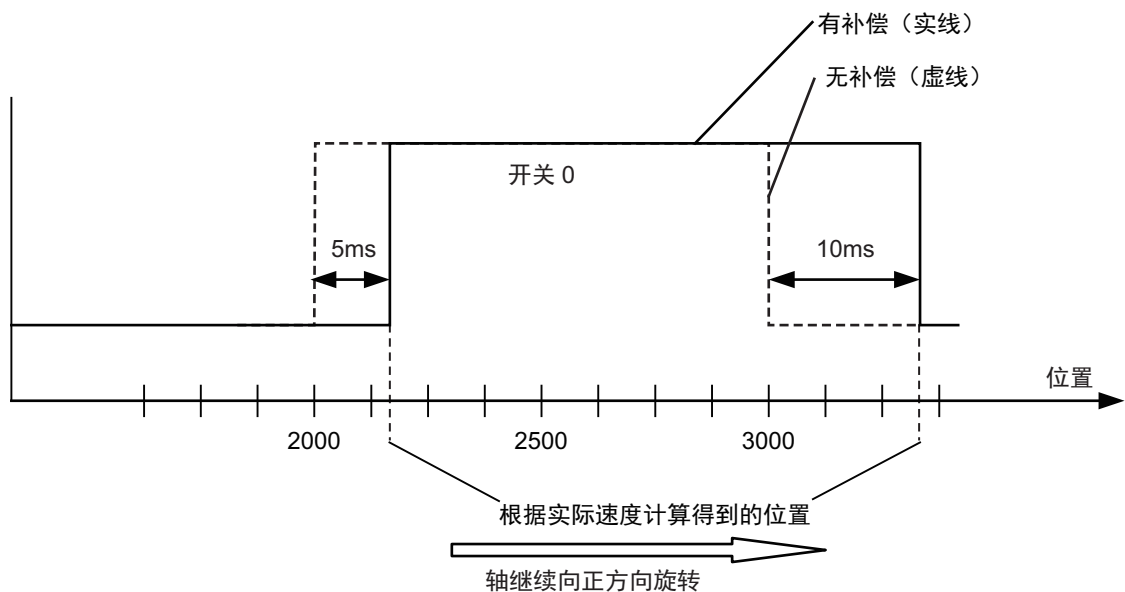
有效范围为“ $T\#-1s \sim T\#1s$ ”，设定时有以下限制。

- 请在不超出轴旋转半圈的时间范围内设定。
例如，以 500rpm 旋转时，每圈的时间为 120ms。因为是半圈的时间，所以 OnCompensation（ON 时间补偿）、OffCompensation（OFF 时间补偿）请设定为 -60ms 到 60ms 之间的值。
- 若设定为超出轴旋转半圈的时间，InOperation（操作中）将变为 FALSE，同时 EnableOut（输出有效）变为 FALSE。请务必在确认 InOperation（操作中）的状态后再使用。
- 因上述原因导致 InOperation（操作中）变为 FALSE 后，若将值恢复为正常范围内，InOperation（操作中）将变为 TRUE。

单元版本 1.08 以下的 CPU 单元

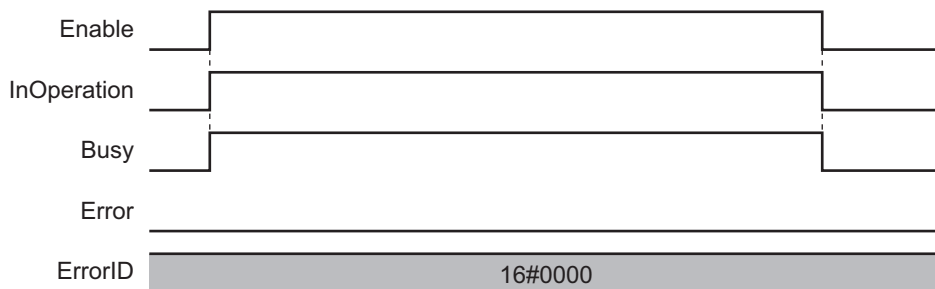
- 有效范围为“ $T\#-1s \sim T\#0s$ ”。

在前述的“设定例 (P.3-382)”中，将 OnCompensation（ON 时间补偿）设定为 $T\#5ms$ ，OffCompensation（OFF 时间补偿）设定为 $T\#10ms$ 时的动作如下所示。



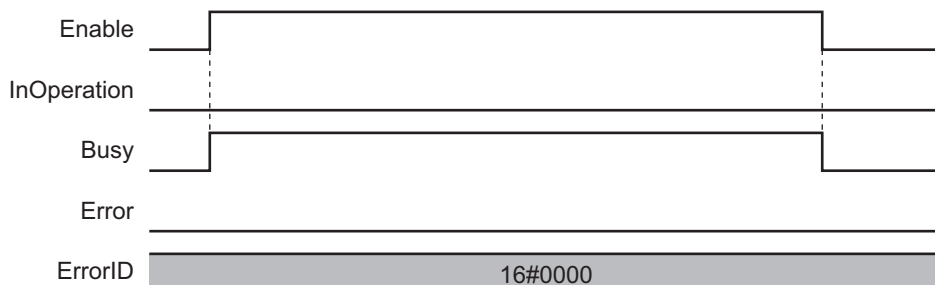
时序图

执行了本指令时的时序图如下所示。



表示指定了未使用轴时或正在执行 MC 调试时的时序图。

单元版本 1.09 以上的 CPU 单元且计数模式为旋转模式时，若将 OnCompensation（ON 时间补偿）或 OffCompensation（OFF 时间补偿）设定为超出轴旋转半圈的时间，时序图相同。



此时，Outputs 的 EnableOut 将变为 FALSE。

运动指令重新执行

输入为“Enable 型”的指令没有运动指令重新执行。

运动指令多重启动

本指令中，各指令实例单独动作，因此运动指令多重启动没有限制。

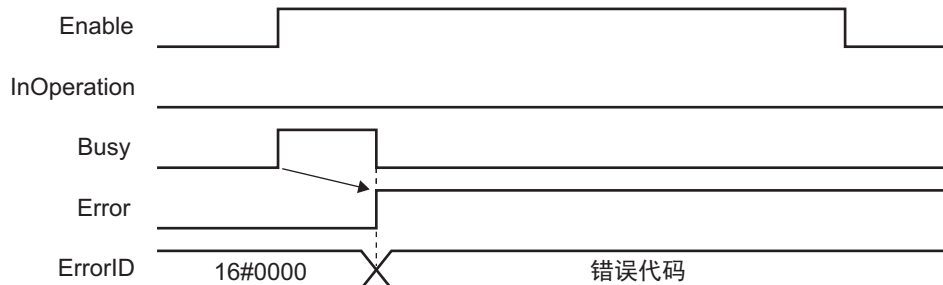
运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

异常

无法启动本指令时，本指令将发生异常，Error（错误）变为 TRUE。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



此时，Outputs 的 EnableOut 将变为 FALSE。

● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

示例程序

下面介绍前述「设定例 (P.3-389)」的示例程序。

使用设备

下面表示本示例程序中使用的设备。

设备	型号
EtherCAT 耦合器单元	NX-ECC201(Ver.1.1) ^{*1}
脉冲输出单元	NX-PG0122 ^{*2}
增量编码器输入单元	NX-EC0122 ^{*3}
数字输出单元	NX-OD2154 ^{*4}

- *1. 节点地址为“1”，设备名称为“E001”。
- *2. NX 单元编号为“1”，设备名称为“N1”。分配到轴 1。
- *3. NX 单元编号为“2”，设备名称为“N2”。分配到轴 2。
- *4. NX 单元编号为“3”，设备名称为“N3”。

参数设定

下面表示本示例程序中使用的最低限度的设定。

● 轴参数

轴种类

轴	轴种类
轴 1	伺服轴
轴 2	编码器轴

计数器模式

轴	计数器模式
轴 1	旋转模式
轴 2	旋转模式

循环计数器

轴	上限值	下限值
轴 1	5000	0
轴 2	5000	0

坐标单位选择

轴	坐标单位
轴 1	pulse
轴 2	pulse

梯形图

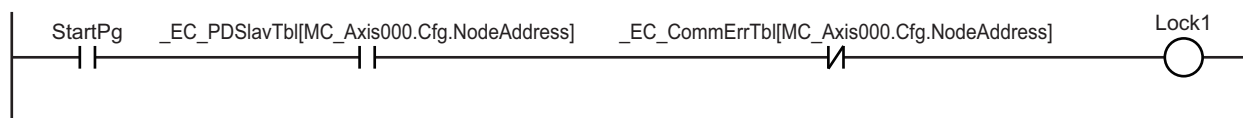
● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis001	_sAXIS_REF	—	轴 2 的轴变量。
E001_Time_Stamp_of_Synchronous_Output	ULINT	—	设备变量。 ^{*1}
N3_Output_Bit_00	BOOL	—	设备变量。
N3_Output_Bit_00_Time_Stamp	ULINT	—	设备变量。
N3_Output_Bit_01	BOOL	—	设备变量。
N3_Output_Bit_01_Time_Stamp	ULINT	—	设备变量。
Switches	ARRAY[0..3] OF _sCAMSWITCH_REF	—	输入输出变量“Switches”的输入参数。 要素编号与开关编号对应。
Outputs	ARRAY[0..1] OF _sOUTPUT_REF	—	输入输出变量“Outputs”的输入参数。 要素编号与轨道编号对应。
TrackOptions	ARRAY[0..1] OF _sTRACK_REF	—	输入输出变量“TrackOptions”的输入参数。 要素编号与轨道编号对应。
EnableMask	WORD	16#0003	输入变量“EnableMask”的输入参数。 轨道 0 和轨道 1 为有效。

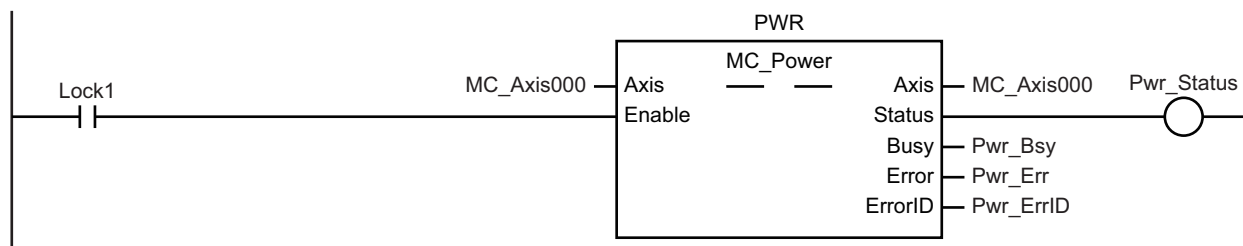
*1. 需要在 EtherCAT 耦合器单元的 I/O 入口上事先添加 0x200A:02(Time Stamp of Synchronous Output)。

● 示例程序

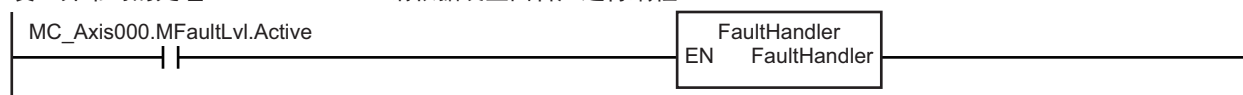
接点 StartPg 变为 TRUE 后，确认轴 1 的 EtherCAT 通信是否正常进行过程数据通信



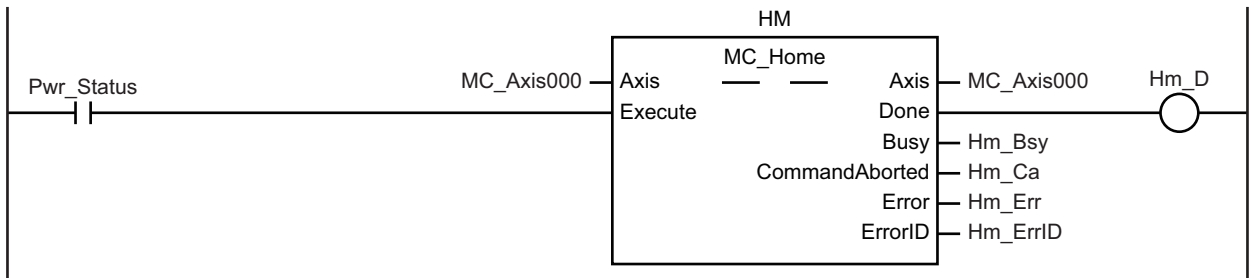
若轴 1 正在进行过程数据通信，则变为伺服 ON 状态



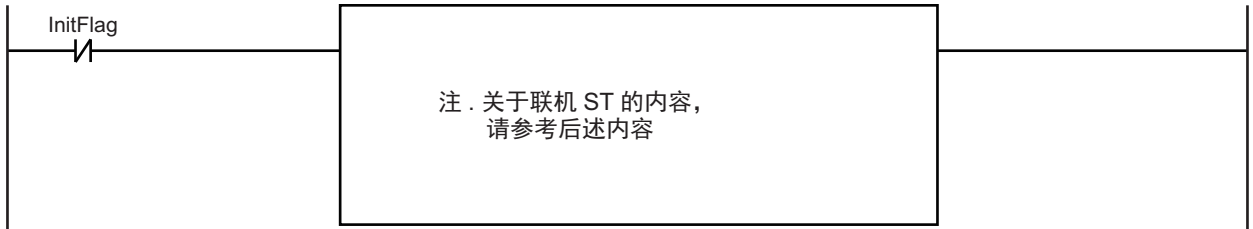
轴构成发生轻度故障等级的异常时，执行装置发生异常时的处理（FaultHandler）。
发生异常时的处理（FaultHandler）请根据装置由客户进行编程



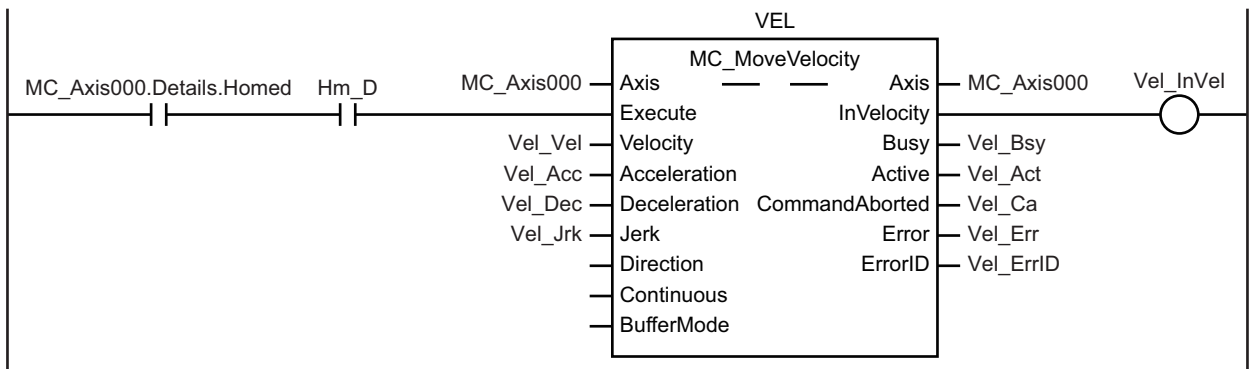
轴 1 为伺服 ON 状态且处于原点未确定状态时，执行原点复位，确定原点



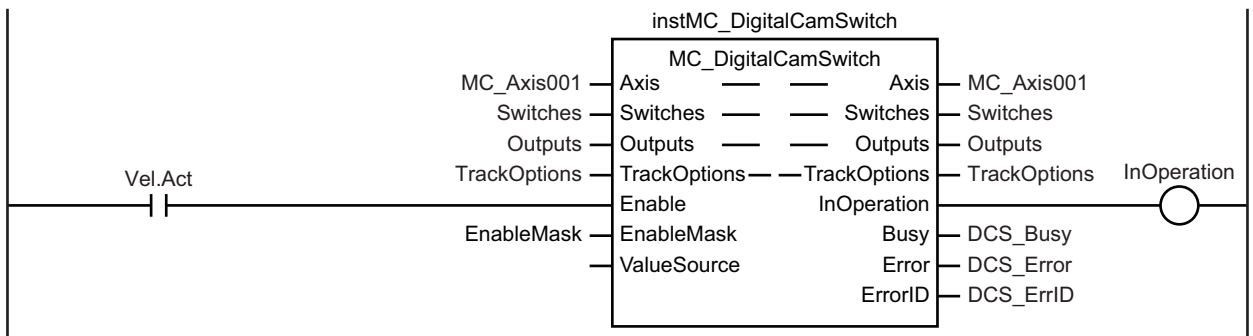
设定 MC_MoveVelocity (速度控制) 指令的参数和 MC_DigitalCamSwitch (数字凸轮开关有效) 指令的 Switches 参数



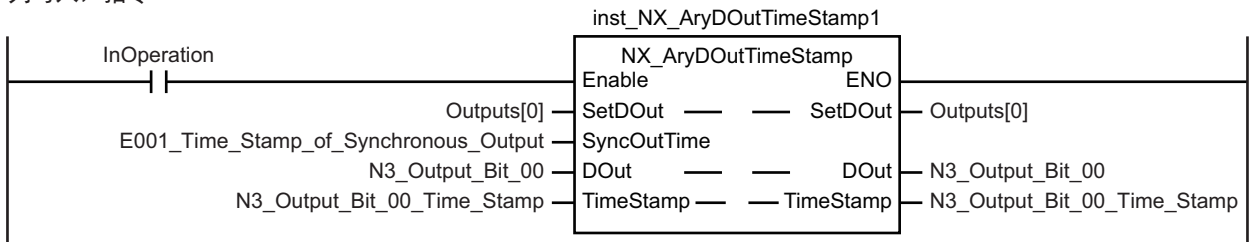
轴 1 变为原点确定状态后，执行 MC_MoveVelocity (速度控制) 指令

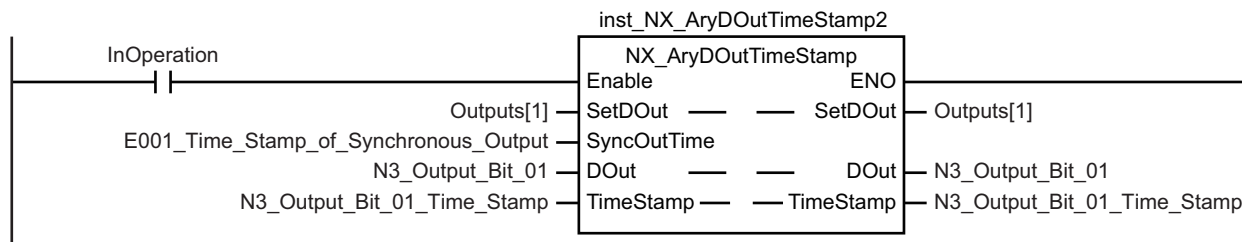


执行轴 1 的 MC_MoveVelocity (速度控制) 指令后，执行轴 2 的 MC_DigitalCamSwitch (数字凸轮开关有效) 指令



轴 2 的 MC_DigitalCamSwitch (数字凸轮开关有效) 指令动作后，执行 NX_AryDOutTimeStamp (时间戳数字输出排列写入) 指令





联机 ST 的内容

//MC_MoveVelocity 参数

```

Vel_Vel := LREAL#1000.0;
Vel_Acc := LREAL#0.0;
Vel_Dec := LREAL#0.0;
Vel_Jrk := LREAL#1000.0;
InitFlag := BOOL#TRUE;
  
```

//MC_DigitalCamSwitch 参数

```

Switches[0].TrackNumber := UINT#0;
Switches[0].FirstOnPosition := LREAL#2000.0;
Switches[0].LastOnPosition := LREAL#3000.0;
Switches[0].AxisDirection := _eMC_DIRECTION#_mcPositiveDirection;
Switches[0].CamSwitchMode := _eMC_SWITCH_MODE#_mcPositionBased;
  
```

```

Switches[1].TrackNumber := UINT#0;
Switches[1].FirstOnPosition := LREAL#2500.0;
Switches[1].LastOnPosition := LREAL#3000.0;
Switches[1].AxisDirection := _eMC_DIRECTION#_mcNegativeDirection;
Switches[1].CamSwitchMode := _eMC_SWITCH_MODE#_mcPositionBased;
  
```

```

Switches[2].TrackNumber := UINT#0;
Switches[2].FirstOnPosition := LREAL#4000.0;
Switches[2].LastOnPosition := LREAL#1000.0;
Switches[2].AxisDirection := _eMC_DIRECTION#_mcNoDirection;
Switches[2].CamSwitchMode := _eMC_SWITCH_MODE#_mcPositionBased;
  
```

```

Switches[3].TrackNumber := UINT#1;
Switches[3].FirstOnPosition := LREAL#3000.0;
Switches[3].Duration := T#1350ms;
Switches[3].AxisDirection := _eMC_DIRECTION#_mcNoDirection;
Switches[3].CamSwitchMode := _eMC_SWITCH_MODE#_mcTimeBased;
  
```

结构文本 (ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis001	_sAXIS_REF	—	轴 2 的轴变量。
E001_Time_Stamp_of_Synchronous_Output	ULINT	—	设备变量。 ^{*1}
N3_Output_Bit_00	BOOL	—	设备变量。
N3_Output_Bit_00_Time_Stamp	ULINT	—	设备变量。
N3_Output_Bit_01	BOOL	—	设备变量。
N3_Output_Bit_01_Time_Stamp	ULINT	—	设备变量。
Pwr_En	BOOL	—	分配到 MC_Power 的实例 PWR 的输入输出变量 Enable 中的变量。
Switches	ARRAY[0..3] OF _sCAMSWITCH_REF	—	输入输出变量“Switches”的输入参数。 要素编号与开关编号对应。
Outputs	ARRAY[0..1] OF _sOUTPUT_REF	—	输入输出变量“Outputs”的输入参数。 要素编号与轨道编号对应。
TrackOptions	ARRAY[0..1] OF _sTRACK_REF	—	输入输出变量“TrackOptions”的输入参数。 要素编号与轨道编号对应。
EnableMask	WORD	16#0003	输入变量“EnableMask”的输入参数。 轨道 0 和轨道 1 为有效。
Hm_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时， 启动 MC_Home 的实例 HM。
Vel_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时， 启动 MC_MoveVelocity 的实例 VEL。

*1. 需要在 EtherCAT 耦合器单元的 I/O 入口上事先添加 0x200A:02(Time Stamp of Synchronous Output)。

● 示例程序

```
// 未设定输入参数时的处理
IF InitFlag=FALSE THEN

    //MC_MoveVelocity 参数
    Vel_Vel      := LREAL#1000.0;
    Vel_Acc      := LREAL#0.0;
    Vel_Dec      := LREAL#0.0;
    Vel_Jrk      := LREAL#1000.0;

    //MC_DigitalCamSwitch 参数
    Switches[0].TrackNumber := UINT#0;
    Switches[0].FirstOnPosition := LREAL#2000.0;
    Switches[0].LastOnPosition := LREAL#3000.0;
```

```

Switches[0].AxisDirection      := _eMC_DIRECTION#_mcPositiveDirection;
Switches[0].CamSwitchMode     := _eMC_SWITCH_MODE#_mcPositionBased;
Switches[1].TrackNumber       := UINT#0;
Switches[1].FirstOnPosition   := LREAL#2500.0;
Switches[1].LastOnPosition    := LREAL#3000.0;
Switches[1].AxisDirection     := _eMC_DIRECTION#_mcNegativeDirection;
Switches[1].CamSwitchMode     := _eMC_SWITCH_MODE#_mcPositionBased;
Switches[2].TrackNumber       := UINT#0;
Switches[2].FirstOnPosition   := LREAL#4000.0;
Switches[2].LastOnPosition    := LREAL#1000.0;
Switches[2].AxisDirection     := _eMC_DIRECTION#_mcNoDirection;
Switches[2].CamSwitchMode     := _eMC_SWITCH_MODE#_mcPositionBased;
Switches[3].TrackNumber       := UINT#1;
Switches[3].FirstOnPosition   := LREAL#3000.0;
Switches[3].Duration          := T#1350ms;
Switches[3].AxisDirection     := _eMC_DIRECTION#_mcNoDirection;
Switches[3].CamSwitchMode     := _eMC_SWITCH_MODE#_mcTimeBased;

// 设定输入参数后, 将 InitFlag 设为 TRUE。
InitFlag:=TRUE;

END_IF;

// 接点 StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 1 设为伺服 ON 状态。
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_RegSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
  Pwr_En:=TRUE;
ELSE
  Pwr_En:=FALSE;
END_IF;

// 轴 1 发生轻度故障时, 执行异常处理 FaultHandler。
// 发生异常时的处理 (FaultHandler) 请根据装置由客户进行编程。
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
  FaultHandler();
END_IF;

// 伺服 ON 状态且原点未确定时, 执行原点复位。
IF (Pwr_Status=TRUE) THEN
  Hm_Ex:=TRUE;
END_IF;

// 确定原点后, 执行 MC_MoveVelocity。
IF MC_Axis000.Details.Homed=TRUE AND Hm_D=TRUE THEN
  Vel_Ex:=TRUE;
END_IF;

//MC_Power
PWR(
  Axis      := MC_Axis000,
  Enable    := Pwr_En,
  Status    => Pwr_Status,
  Busy      => Pwr_Bsy,
  Error     => Pwr_Err,
  ErrorID   => Pwr_ErrID
);

```

```

//MC_Home
HM(
  Axis           := MC_Axis000,
  Execute        := Hm_Ex,
  Done           => Hm_D,
  Busy           => Hm_Bsy,
  CommandAborted => Hm_Ca,
  Error          => Hm_Err,
  ErrorID        => Hm_ErrID
);

//MC_MoveVelocity
VEL(
  Axis           := MC_Axis000,
  Execute        := Vel_Ex,
  Velocity       := Vel_Vel,
  Acceleration   := Vel_Acc,
  Deceleration   := Vel_Dec,
  Jerk           := Vel_Jrk,
  InVelocity     => Vel_InVel,
  Busy           => Vel_Bsy,
  Active         => Vel_Act,
  CommandAborted => Vel_Ca,
  Error          => Vel_Err,
  ErrorID        => Vel_ErrID
);

//MC_DigitalCamSwitch
instMC_DigitalCamSwitch(
  Axis           := MC_Axis001,
  Switches       := Switches,
  Outputs        := Outputs,
  TrackOptions   := TrackOptions,
  Enable         := Vel_Act,
  EnableMask     := EnableMask,
  InOperation    => InOperation,
  Busy           => DCS_Busy,
  Error          => DCS_Error,
  ErrorID        => DCS_ErrorID );

inst_NX_AryDOutTimeStamp1(
  Enable         := Vel_Act,
  SetDOut        := Outputs[0],
  SyncOutTime    := E001_Time_Stamp_of_Synchronous_Output,
  DOut           := N3_Output_Bit_00,
  TimeStamp      := N3_Output_Bit_00_Time_Stamp);

inst_NX_AryDOutTimeStamp2(
  Enable         := Vel_Act,
  SetDOut        := Outputs[1],
  SyncOutTime    := E001_Time_Stamp_of_Synchronous_Output,
  DOut           := N3_Output_Bit_01,
  TimeStamp      := N3_Output_Bit_01_Time_Stamp);

```

MC_TimeStampToPos

计算指定时间戳的轴位置。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_TimeStampToPos	时间戳→轴位置 计算	FB		<pre>MC_TimeStampToPos_instance (Axis := 《参数》, Enable := 《参数》, TimeStamp := 《参数》, ValueSource := 《参数》, Enabled => 《参数》, Busy => 《参数》, CalcPosition => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>



使用注意事项

本指令对分配有 NX 系列位置接口单元的轴有效。
可使用的 NX 单元为 NX-EC0 □□□、NX-ECS □□□。



版本相关信息

本指令可在 CPU 单元版本 1.06 以上和 Sysmac Studio Ver.1.07 以上的组合中使用。

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、FALSE	FALSE	TRUE 时，执行指令。
TimeStamp	时间戳	ULINT	正数或“0”	0	指定用于计算位置的时间戳。 指定以支持时间戳方式的数字输入单元或编码器输入单元中的时刻为基准的时间戳。单位为 [ns]。
ValueSource (Reserved)	输入信息	_sMC_S OURCE	—	—	(Reserved)

输出变量

输出变量	名称	数据类型	有效范围	内容
Enabled	有效	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
CalcPosition	计算位置	LREAL	负数、正数、“0”	输出指定时间戳的位置。 单位为 [指令单位]。 ^{*1}
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	^{*2}	发生异常时，输出错误代码。16#0000 表示正常。

*1. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. □ □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Enabled	Enable 变为 TRUE 时	<ul style="list-style-type: none"> • Enable 变为 FALSE 时 • Error 变为 TRUE 时
Busy	Enable 的上升沿	<ul style="list-style-type: none"> • Enable 变为 FALSE 时 • Error 变为 TRUE 时
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

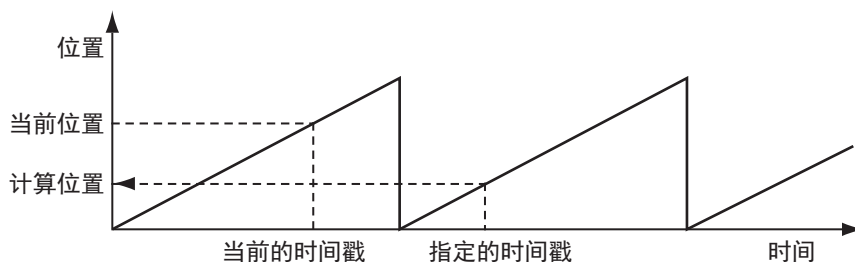
输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。 ^{*1 *2}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

*2. 请指定分配了 NX 系列编码器输入单元的编码器轴。

功能说明

- 本指令可根据轴的反馈当前位置和时间戳，计算输入变量中指定时间戳的反馈当前位置。
- 轴中请指定 NX 系列编码器输入单元。
- 轴参数的计数模式为直线模式时，计算得到的位置为溢出或下溢时，计算值为溢出或下溢的值。此时，不会发生异常。





使用注意事项

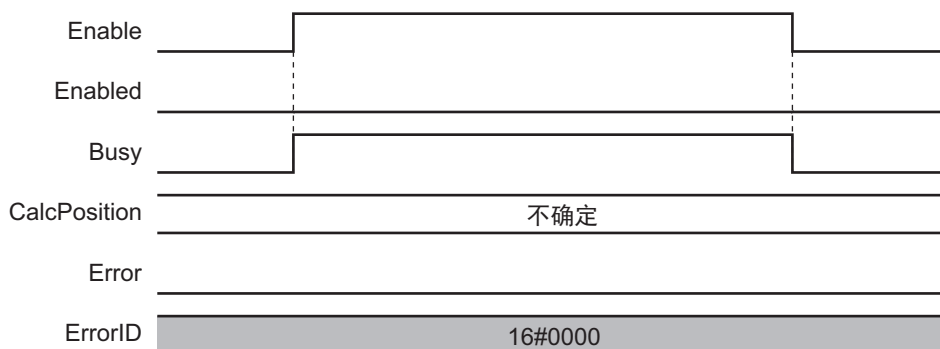
- 本指令需要使用 NX 系列编码器输入单元，且变化时刻获取功能处于运行状态。
以下情况下，变化时刻获取功能不运行。
 - 使用不支持变化时刻获取功能的编码器输入或使用伺服驱动器时
 - NX 系列编码器输入单元的对象 6010Hex（时间戳）未分配到 PDO 时
 - EtherCAT 耦合器单元的 [DC 有效] 设定为 [无效] 时
 - 即使不更新时间戳，本指令也不会发生异常。此时，会计算位置，但不是指定时间戳的位置。
请通过 Sysmac Studio 的 MC 检视表或监视窗口，确认轴变量的成员“TimeStamp（时间戳）”已更新后，再使用本指令。
 - 本指令可根据轴的当前位置和当前速度，算出在指定时间戳时的位置。若轴发生剧烈加减速，算出的误差可能会更大，因此请在轴保持恒定速度的状态下使用。
请充分进行动作验证，确保安全。
 - 指定了未使用轴时或正在执行 MC 调试时，若 Enable 为 TRUE，则 Busy 变为 TRUE，Enabled 和 Error 变为 FALSE。
 - 请勿配置 2 个以上同名的实例。若配置，可能发生意外输出。
-

时序图

执行了本指令时的时序图如下所示。



表示指定了未使用轴时或正在执行 MC 调试时的时序图。



运动指令重新执行

输入为“Enable 型”的指令没有运动指令重新执行。

运动指令多重启动

本指令中，各指令单独动作，因此运动指令多重启动没有限制。

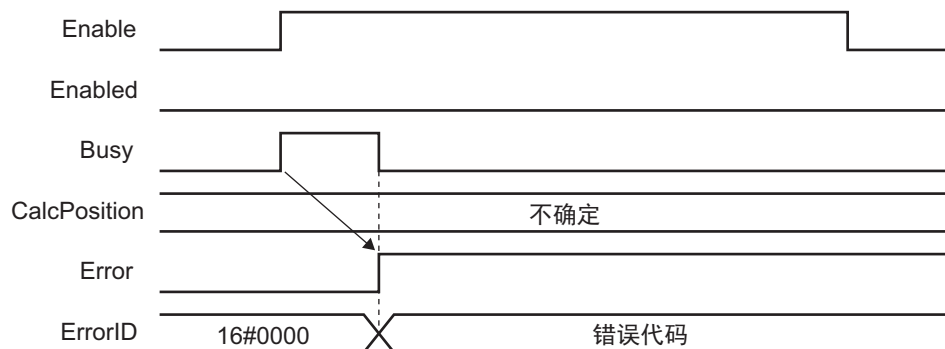
运动指令多重启动的详情请参考 □□ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

异常

无法启动本指令时，本指令将发生异常，Error（错误）变为 TRUE。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

示例程序

以下示例程序为计算皮带输送机上移动的工件间距离的程序。

使用设备

下面表示本示例程序中使用的设备。

设备	型号
EtherCAT 耦合器单元	NX-ECC201(Ver.1.1) ^{*1}
脉冲输出单元	NX-PG0122 ^{*2}
增量编码器输入单元	NX-EC0122 ^{*3}
数字输入单元	NX-ID3344 ^{*4}

*1. 节点地址为“1”，设备名称为“E001”。

*2. NX 单元编号为“1”，设备名称为“N1”。分配到轴 1。

*3. NX 单元编号为“2”，设备名称为“N2”。分配到轴 2。

*4. NX 单元编号为“3”，设备名称为“N3”。

参数设定

下面表示本示例程序中使用的最低限度的设定。

● 轴参数

轴种类

轴	轴种类
轴 1	伺服轴
轴 2	编码器轴

计数器模式

轴	计数器模式
轴 1	旋转模式
轴 2	旋转模式

循环计数器

轴	上限值	下限值
轴 1	360	0
轴 2	360	0

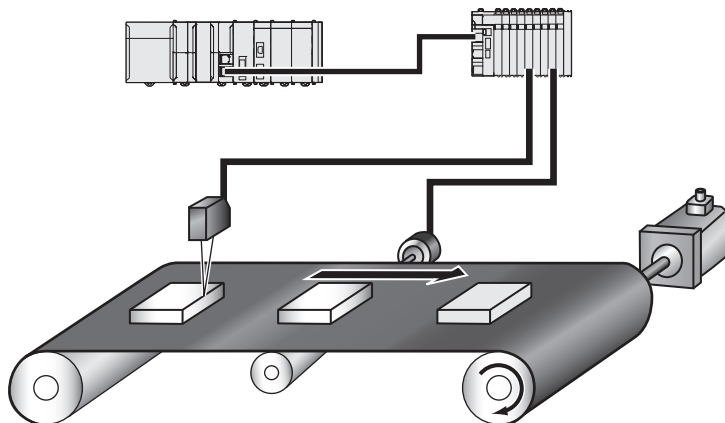
坐标单位选择

轴	坐标单位
轴 1	degree
轴 2	degree

动作示例

传感器检测到工件后，传感器输出变为 ON，工件通过后，传感器输出变为 OFF。传感器检测到下一个工件后，传感器输出再次变为 ON。

根据传感器输出变为 ON 时的时间戳，计算编码器输入的位置。两者位置的差即为工件间的距离。



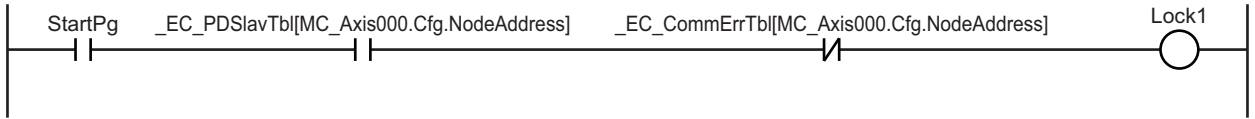
梯形图

● 主要变量

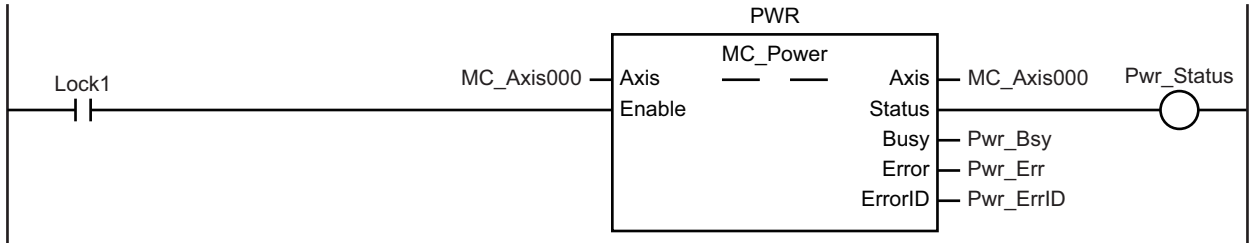
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis001	_sAXIS_REF	—	轴 2 的轴变量。
N3_Input_Bit_00	BOOL	—	设备变量。
N3_Input_Bit_00_Time_Stamp	ULINT	—	设备变量。
Position	ARRAY[0..1] OF LREAL	—	保存算出位置。
Count	ARRAY[0..1] OF ULINT	—	保存多次旋转数。
FirstPoint	UINT	—	处理用的变量。
LastPoint	UINT	—	处理用的变量。
Distance	LREAL	—	工件间的距离。

● 示例程序

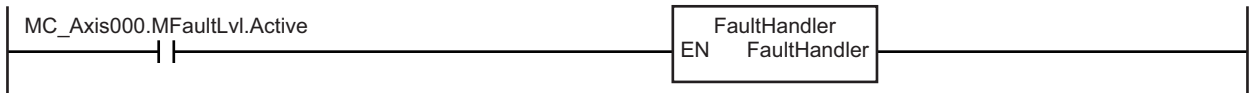
接点 StartPg 变为 TRUE 后，确认轴 1 的 EtherCAT 通信是否正常进行过程数据通信



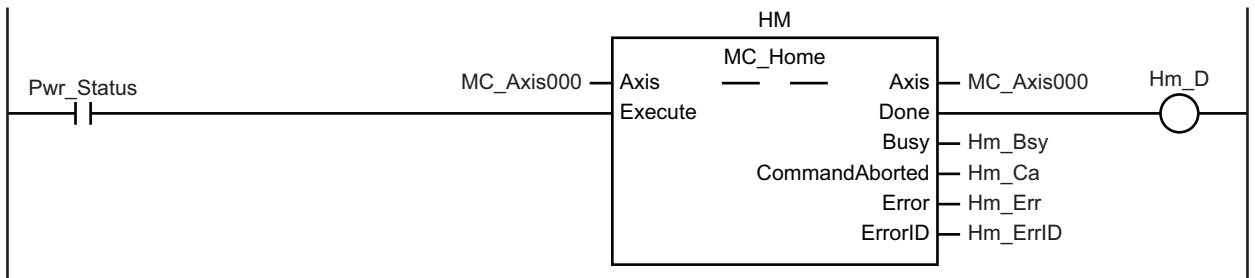
若轴 1 正在进行过程数据通信，则变为伺服 ON 状态



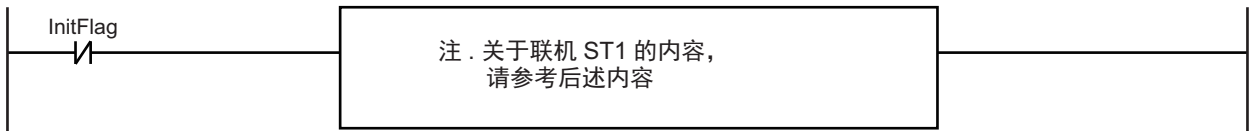
轴构成发生轻度故障等级的异常时，执行装置发生异常时的处理（FaultHandler）。发生异常时的处理（FaultHandler）请根据装置由客户进行编程



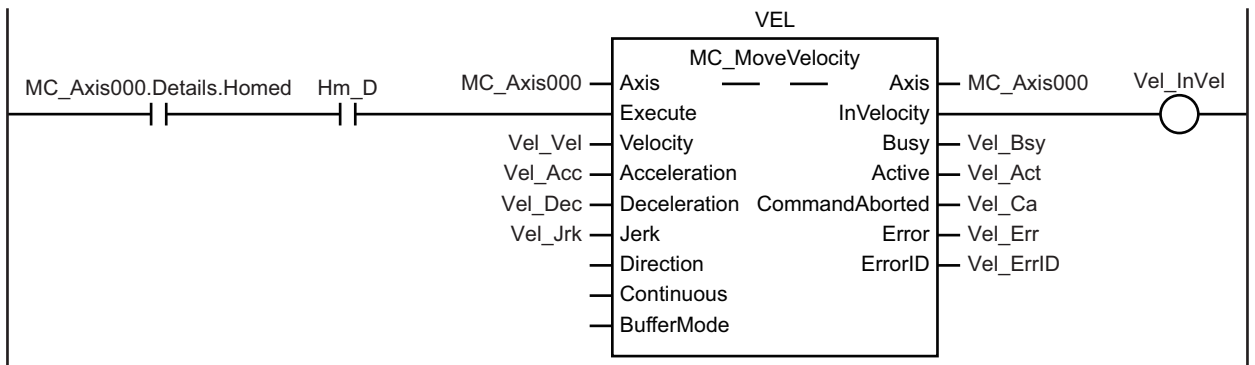
轴 1 为伺服 ON 状态且处于原点未确定状态时，执行原点复位，确定原点



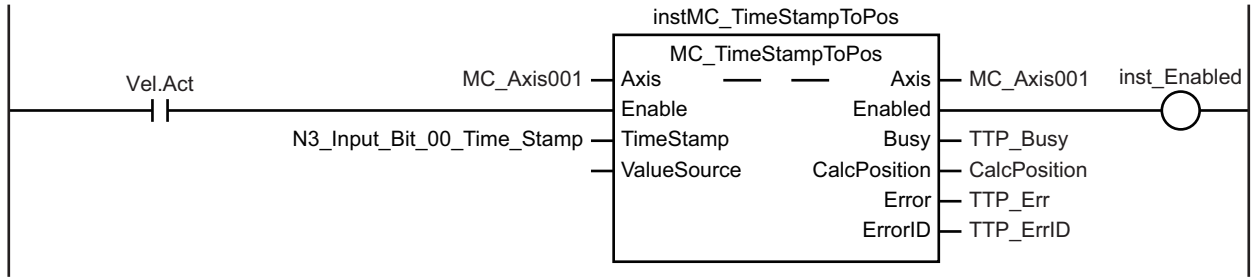
设定 MC_MoveVelocity（速度控制）指令的参数



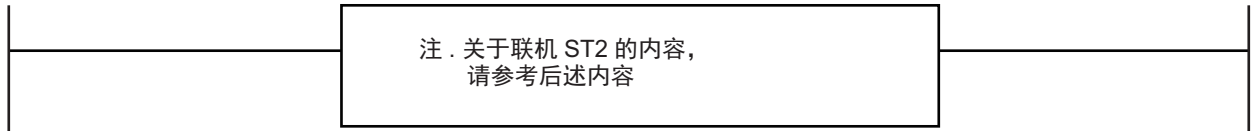
轴 1 变为原点确定状态后，执行 MC_MoveVelocity（速度控制）指令



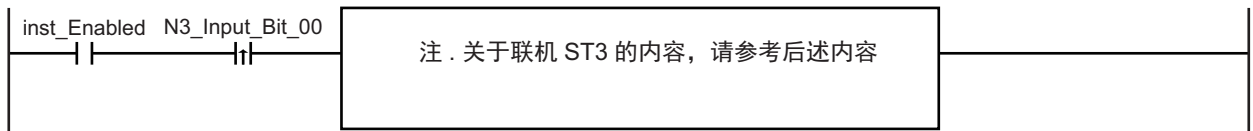
执行轴 1 的 MC_MoveVelocity (速度控制) 指令后, 执行轴 2 的 MC_TimeStampToPos (时间戳→轴位置计算) 指令



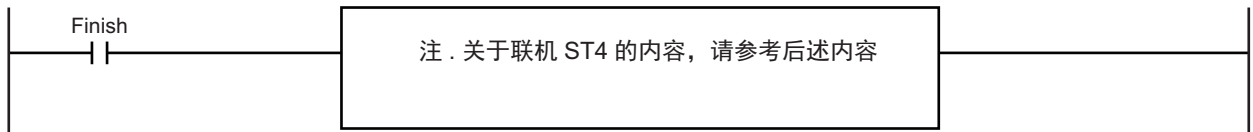
对编码器轴的多次旋转数进行计数 (本次值 < 上次值时, 判断为超出循环计数器的上限值)



传感器输出的 ON 上升沿时获取 CalcPosition (计算位置) (Position[0] 和 Position[1] 相互保存)



检测第 2 个工件后, 计算工件间的距离



联机 ST1 的内容

```
//MC_MoveVelocity 参数
Vel_Vel := LREAL#1000.0;
Vel_Acc := LREAL#0.0;
Vel_Dec := LREAL#0.0;
Vel_Jrk := LREAL#1000.0;
InitFlag := BOOL#TRUE;
```

联机 ST2 的内容

```
IF MC_Axis001.Act.Pos < PreAxis001ActPos THEN
  Inc(RotaryCount);
END_IF;
PreAxis001ActPos := MC_Axis001.Act.Pos;
```

联机 ST3 的内容

```
IF Index < UINT#2 THEN
  Position[Index] := CalcPosition;
  Count[Index] := RotaryCount;
  Index := Index + UINT#1;
END_IF;
IF Index >= UINT#2 THEN
  Finish := BOOL#TRUE;
  Index := UINT#0;
END_IF;
```

联机 ST4 的内容

```
// 第 1 次比较: 工件 1=Position[0]、工件 2=Position[1]
// 第 2 次比较: 工件 2=Position[1]、工件 3=Position[0]
// 第 3 次比较: 工件 3=Position[0]、工件 4=Position[1]
//...
FirstPoint := (Index+UINT#1) MOD UINT#2;
LastPoint := Index;
DiffCount := Count[1] - Count[0];
Distance := (ABS( DiffCount) -LINT#1)* 360.0 +
            (360.0 + Position[FirstPoint] - Position[LastPoint]);
```

结构文本 (ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis001	_sAXIS_REF	—	轴 2 的轴变量。
N3_Input_Bit_00	BOOL	—	设备变量。
N3_Input_Bit_00_Time_Stamp	ULINT	—	设备变量。
Hm_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_Home 的实例 HM。
Vel_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_MoveVelocity 的实例 VEL。
PreN3InputBit00	BOOL	—	设备变量 N3_Input_Bit_00 的上次值。
Position	ARRAY[0..1] OF LREAL	—	保存算出位置。
Count	ARRAY[0..1] OF ULINT	—	保存多次旋转数。
FirstPoint	UINT	—	处理用的变量。
LastPoint	UINT	—	处理用的变量。
Distance	LREAL	—	工件间的距离。

● 示例程序

```

// 未设定输入参数时的处理
IF InitFlag=FALSE THEN

    //MC_MoveVelocity 参数
    Vel_Vel := LREAL#1000.0;
    Vel_Acc := LREAL#0.0;
    Vel_Dec := LREAL#0.0;
    Vel_Jrk := LREAL#1000.0;
    // 设定输入参数后, 将 InitFlag 设为 TRUE
    InitFlag:=TRUE;

END_IF;

// 接点 StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 1 设为伺服 ON 状态。
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
    AND (_EC_RegSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
    AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
    Pwr_En:=TRUE;
ELSE
    Pwr_En:=FALSE;
END_IF;

// 轴 1 发生轻度故障时, 执行异常处理 FaultHandler。
// 发生异常时的处理 (FaultHandler) 请根据装置由客户进行编程。
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;

// 伺服 ON 状态且原点未确定时, 执行原点复位。

```



```

IF (Pwr_Status=TRUE) THEN
  Hm_Ex:=TRUE;
END_IF;

// 确定原点后, 执行 MC_MoveVelocity。
IF MC_Axis000.Details.Homed=TRUE AND Hm_D=TRUE THEN
  Vel_Ex:=TRUE;
END_IF;

// 对编码器轴的多次旋转数进行计数。
// (本次值 < 上次值时, 判断为超出循环计数器的上限值)
IF MC_Axis001.Act.Pos<PreAxis001ActPos THEN
  Inc(RotaryCount);
END_IF;
PreAxis001ActPos := MC_Axis001.Act.Pos;

//MC_TimeStampToPos
instMC_TimeStampToPos(
  Axis           := MC_Axis001,
  Enable         := Vel_Ex,
  TimeStamp      := TimeStamp,
  Enabled        => inst_Enabled,
  Busy           => TSTP_Busy,
  CalcPosition   => CalcPosition,
  Error          => TSTP_Error,
  ErrorID        => TSTP_ErrorID );

// 传感器输出的 ON 上升沿时获取 CalcPosition (计算位置)。
// (Position[0] 和 Position[1] 相互保存)
IF inst_Enabled THEN
  IF PreN3InputBit00=FALSE AND N3_Input_Bit_00=TRUE THEN
    IF Index < UINT#2 THEN
      Position[Index] := CalcPosition;
      Count[Index] := RotaryCount;
      Index := Index + UINT#1;
    END_IF;
    IF Index >= UINT#2 THEN
      Finish := BOOL#TRUE;
      Index := UINT#0;
    END_IF;
  END_IF;
END_IF;
PreN3InputBit00 := N3_Input_Bit_00;

// 检测第 2 个工件后, 计算工件间的距离。
// 第 1 次比较: 工件 1=Position[0]、工件 2=Position[1]
// 第 2 次比较: 工件 2=Position[1]、工件 3=Position[0]
// 第 3 次比较: 工件 3=Position[0]、工件 4=Position[1]
//...
IF Finish THEN
  FirstPoint := (Index+UINT#1) MOD UINT#2;
  LastPoint :=Index;
  DiffCount := Count[1] - Count[0];
  Distance := (ABS( DiffCount) -LINT#1)* 360.0 +
              (360.0 + Position[FirstPoint] - Position[LastPoint]);
END_IF;

//MC_Power

```

```
PWR(  
  Axis           := MC_Axis000,  
  Enable         := Pwr_En,  
  Status        => Pwr_Status,  
  Busy          => Pwr_Bsy,  
  Error         => Pwr_Err,  
  ErrorID       => Pwr_ErrID  
);
```

```
//MC_Home  
HM(  
  Axis           := MC_Axis000,  
  Execute        := Hm_Ex,  
  Done          => Hm_D,  
  Busy          => Hm_Bsy,  
  CommandAborted => Hm_Ca,  
  Error         => Hm_Err,  
  ErrorID       => Hm_ErrID  
);
```

```
//MC_MoveVelocity  
VEL(  
  Axis           := MC_Axis000,  
  Execute        := Vel_Ex,  
  Velocity       := Vel_Vel,  
  Acceleration   := Vel_Acc,  
  Deceleration   := Vel_Dec,  
  Jerk           := Vel_Jrk,  
  InVelocity     => Vel_InVel,  
  Busy          => Vel_Bsy,  
  Active         => Vel_Act,  
  CommandAborted => Vel_Ca,  
  Error         => Vel_Err,  
  ErrorID       => Vel_ErrID  
);
```

MC_PeriodicSyncVariables

按固定周期执行轴变量的任务间同步。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_PeriodicSyncVariables	轴变量任务间固定周期同步	FB		<pre>MC_PeriodicSyncVariables_instance (Axis := 《参数》, Enable := 《参数》, ExecID := 《参数》, Enabled => 《参数》, Busy => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>



使用注意事项

本指令不可在 NJ 系列 CPU 单元中使用。

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
ExecID	执行 ID	UINT	2	2	指定变量同步指定目标的 ID。 2: 固定周期任务（执行优先级 5）

输出变量

输出变量	名称	数据类型	有效范围	内容
Enabled	有效	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*1	发生异常时，输出错误代码。16#0000 表示正常。

*1. □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Enabled	Enable 变为 TRUE 时	<ul style="list-style-type: none"> Enable 为 FALSE 时，1 个周期后 Error 变为 TRUE 时
Busy	Enable 的上升沿	<ul style="list-style-type: none"> Enable 变为 FALSE 时 Error 变为 TRUE 时
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。 ^{*1}

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis***”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

功能说明

- 若将 Enable（有效）设为 TRUE，主固定周期任务和固定周期任务（执行优先级 5）之间，将周期性同步轴变量。
- 同步后的轴变量将输出到指定目标任务的轴变量区域（系统变量）中。



使用注意事项

- 本指令仅支持从主固定周期任务到固定周期任务（执行优先级 5）的变量同步。
- ExecID（执行 ID）中指定的值和该轴所分配的任务相同时，不会发生异常，本指令保持 Busy（执行中）。



参考

若通过用户程序对不同任务所控制的轴的轴参数作为外部变量声明进行参照，因用户程序的执行开始位置不同，参照哪个时间点更新的轴参数将变得不确定。
对于电子凸轮等控制的轴有主从关系的用途，用不同的任务控制时，请使用 MC_PeriodicSyncVariables（轴变量任务间固定周期同步）指令。

指令的详情

下面介绍指令的详情。

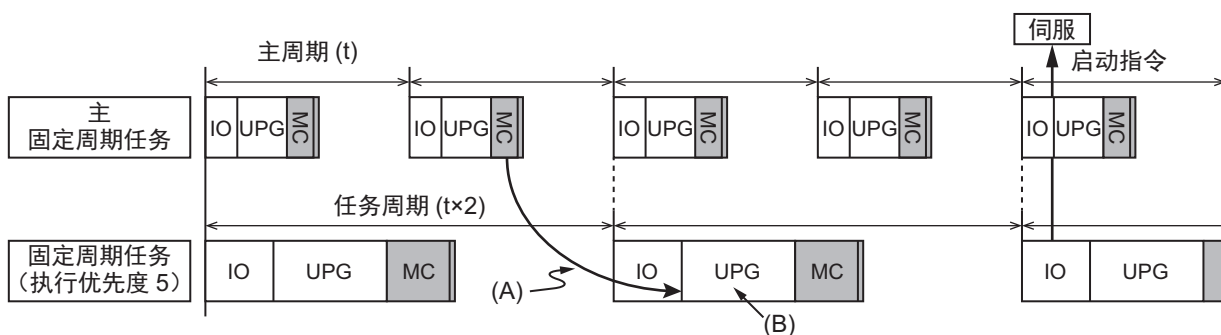
● 同步时序

主固定周期任务控制的轴的变量值可通过固定周期任务（执行优先级 5）的用户程序参照。

正在执行固定周期任务（执行优先级 5）的用户程序时，主固定周期任务的轴变量值不会重写。

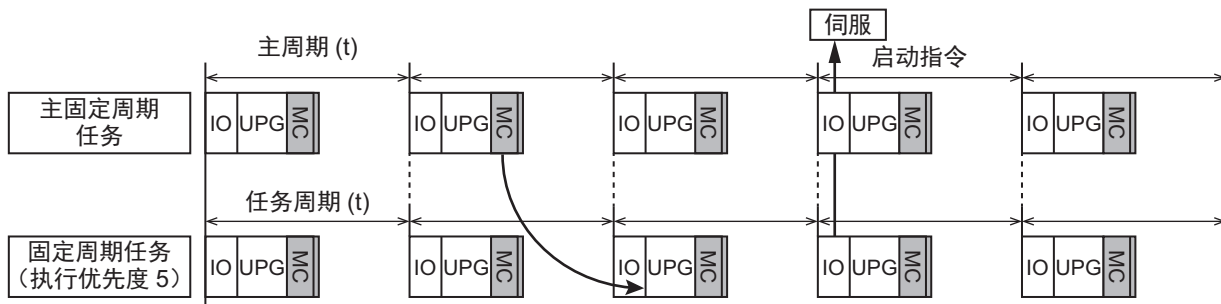
固定周期任务（执行优先级 5）的用户程序将反映主固定周期任务和固定周期任务（执行优先级 5）的任务周期起点重叠前的主固定周期任务的执行结果。

主固定周期任务的轴变量值反映到固定周期任务（执行优先级 5）的时间如下所示。



符号	说明
(A)	轴变量的更新。 无论固定周期任务（执行优先级 5）的用户程序执行开始位置如何，都将反映任务周期起点重叠前的主固定周期任务的执行结果。
(B)	用户程序正在执行时，轴变量的值不会发生重写。

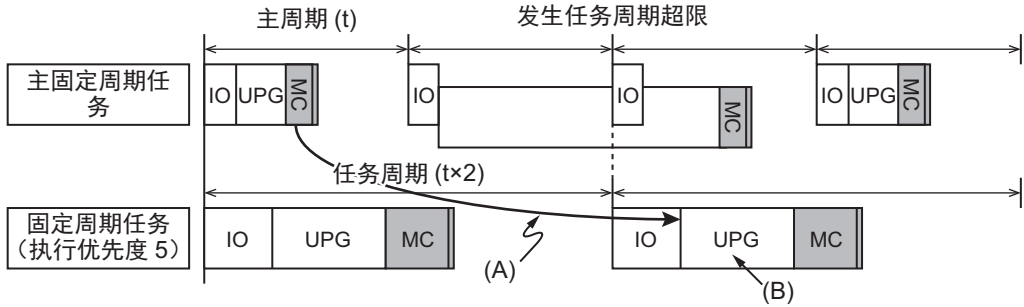
任务周期相同时也一样。





参考

发生任务周期超限时，不会反应任务周期起点重叠前的主固定周期任务的执行结果，而是再前一个周期的执行结果。



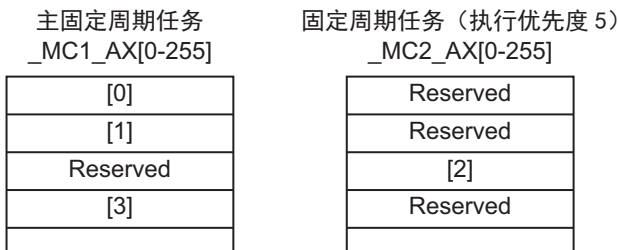
符号	说明
(A)	轴变量的更新。 发生任务周期超限时，反映前一个周期的值。
(B)	用户程序正在执行时，轴变量的值不会发生重写。

● 变量的输出区域

同步后的轴变量将输出到指定目标任务的轴变量区域（系统变量）中。

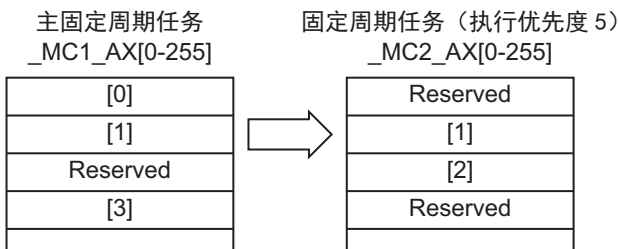
例如，如下设定轴参数“运动控制”时，轴变量（系统变量）的更新对象为下图的区域。

轴编号	运动控制（分配任务）
轴 0	主固定周期任务
轴 1	主固定周期任务
轴 2	固定周期任务（执行优先级 5）
轴 3	主固定周期任务



(注) Reserved 除基本设定外，为初始值。

若将 MC_PeriodicSyncVariables（轴变量任务间周期同步）指令的“Axis（轴）”指定为“轴 1”、“ExecID（执行 ID）”指定为“10#2”并执行，将更新 `_MC2_Ax[1]` 的区域。



运动指令重新执行

输入为“Enable 型”的指令没有运动指令重新执行。

运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

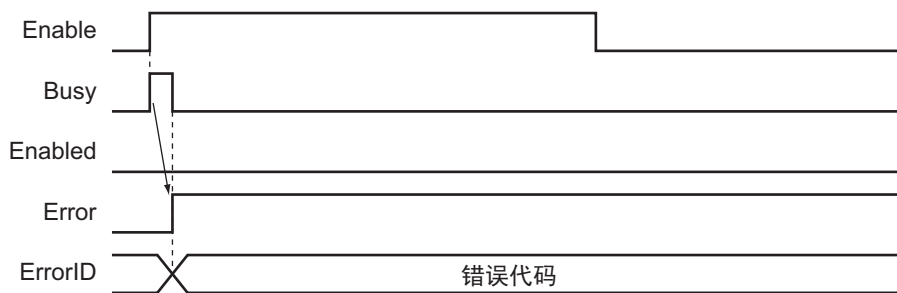
● 本指令的多重启动

对正在执行 MC_PeriodicSyncVariables（轴变量任务间固定周期同步）指令的轴，启动了其他实例的 MC_PeriodicSyncVariables（轴变量任务间固定周期同步）指令时，两个指令都会执行。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，不写入参数。将保持执行指令前的值。参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



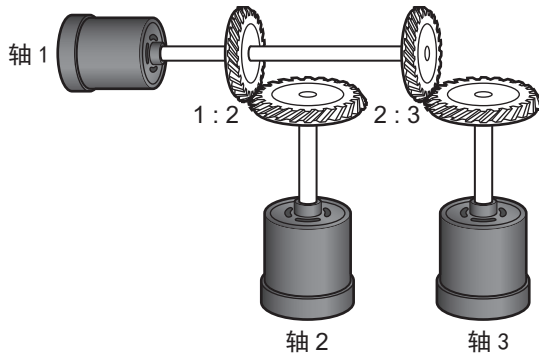
● 错误代码

指令中发生的异常请参考 □ 「A-1 错误代码一览 (P.A-2)」。

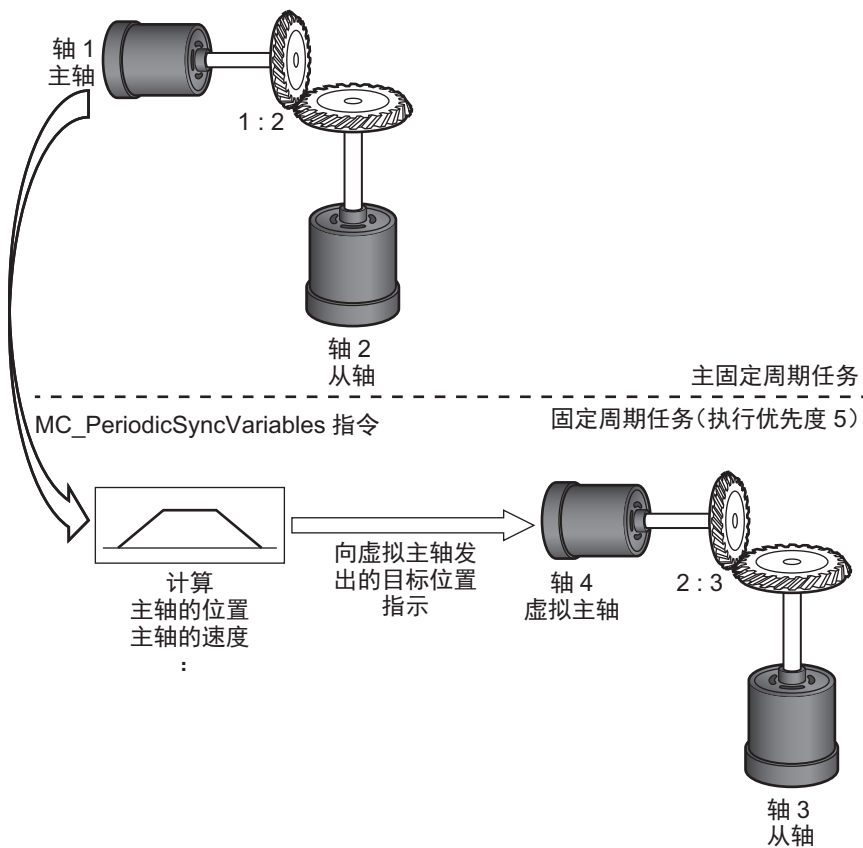
示例程序

具体构成：针对主轴（轴 1），将需要高速高精度的从轴（轴 2）分配到主固定周期任务中，将无精度要求的从轴（轴 3）分配到固定周期任务（执行优先级 5）中进行控制。同时，主轴（轴 1）分配到主固定周期任务中。

● 物理的轴构成



● 逻辑的轴构成



分别在主固定周期任务和固定周期任务（执行优先级 5）中编入程序，实现上述应用的动作。

主固定周期任务的程序

- 在主固定周期任务中，对主固定周期任务中分配的主轴 (Primary_MasterAxis)，执行 MC_PeriodicSyncVariables（轴变量固定周期同步）指令。
- 执行 MC_PeriodicSyncVariables（轴变量任务间固定周期同步）指令，对固定周期任务（执行优先级 5）周期性更新主轴的轴变量。
- 程序中使用的主要变量如下。

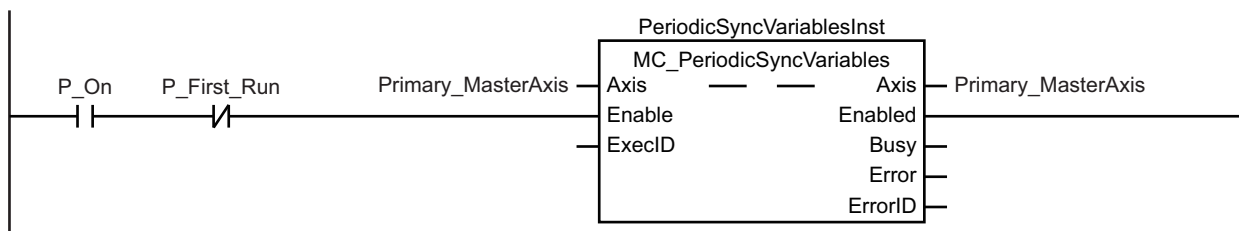
名称	数据类型	注释
Primary_MasterAxis	_sAXIS_REF	主轴（轴 1）。轴变量为 _MC1_AX[0]。 分配到主固定周期任务 *1

*1. 主轴请务必分配到主固定周期任务中。

● 梯形图

主固定周期任务的梯形图程序如下所示。

在主固定周期任务中执行 MC_PeriodicSyncVariables（轴变量固定周期同步）指令。



固定周期任务（执行优先级 5）的程序

- 执行 MC_PeriodicSyncVariables（轴变量任务间固定周期同步）指令，在固定周期任务（执行优先级 5）的程序中，可参照按周期更新的主轴 (Primary_MasterAxis) 的轴变量。
- 根据主轴的信息，对固定周期任务（执行优先级 5）中分配的虚拟主轴 (Secondary_VirtualMasterAxis)，执行 MC_SyncMoveAbsolute（周期性同步绝对定位）指令，追踪主轴。
- 对从轴 (Secondary_SlaveAxis) 指定虚拟主轴 (Secondary_VirtualMasterAxis)。
- 通过执行 MC_GearIn（齿轮动作开始）指令等同步控制指令，可实现任务周期中分配的轴之间的同步控制功能。
- 程序中使用的主要变量如下。

名称	数据类型	注释
Secondary_VirtualMasterAxis	_sAXIS_REF	虚拟主轴。分配到固定周期任务（执行优先级 5）
Secondary_SlaveAxis	_sAXIS_REF	从轴（轴 3）。分配到固定周期任务（执行优先级 5）
Primary_MasterAxis *1	_sAXIS_REF	主轴。轴变量为 _MC2_AX[4]。 利用 MC_PeriodicSyncVariables（轴变量任务间固定周期同步）指令，周期性同步。
PhaseShift	LREAL	相位补偿量
VirtualMasterAxisTargetPosition	LREAL	虚拟主轴的各任务周期的目标位置
SyncExecute	BOOL	从轴的同步控制开始信号

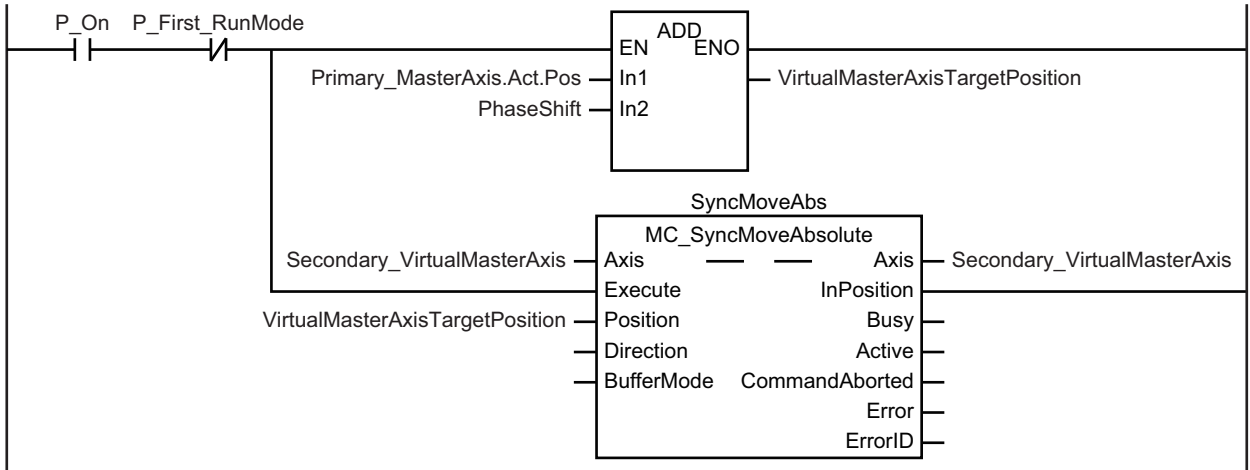
*1. 本示例中，定义并使用了主固定周期任务的主轴和同名的内部变量“Primary_MasterAxis”。

● 梯形图

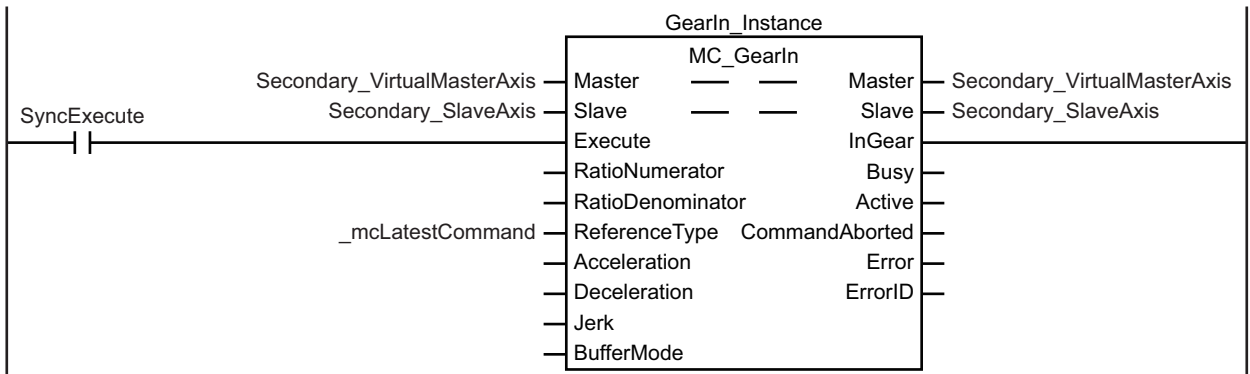
固定周期任务（执行优先级 5）的梯形图程序如下所示。

Primary_MasterAxis 的当前位置和相位补偿量 (PhaseShift) 相加，作为 Secondary_VirtualMasterAxis 的指令位置 (VirtualMasterAxisTargetPosition)。

使用 MC_SyncMoveAbsolute（周期性同步绝对定位）指令，使 Secondary_VirtualMasterAxis 始终追踪 Primary_MasterAxis。



将 Secondary_VirtualMasterAxis 作为虚拟主轴，在 Secondary_SlaveAxis 中执行同步控制指令。



MC_SyncOffsetPosition

执行同步控制中的从轴位置补偿。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_SyncOffsetPosition	周期性同步位置偏置补偿	FB		<pre>MC_SyncOffsetPosition_instance (Axis := 《参数》, Execute := 《参数》, OffsetPosition := 《参数》, BufferMode := 《参数》, MoveMode := 《参数》, OutputtedOffsetPosition => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>



版本相关信息

本指令可在 CPU 单元版本 1.10 以上和 Sysmac Studio Ver.1.12 以上的组合中使用。

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
OffsetPosition	位置偏置	LREAL	负数、正数、“0”	0	指定要加到指令当前位置的位置偏置。单位为 [指令单位]。 ^{*1}
BufferMode	缓冲模式选择	_eMC_BUFFER_MODE	0: _mcAborting	0 ^{*2}	指定运动指令多重启动时的动作。 0: 中断
MoveMode	移动方法选择	_eMC_MOVE_MODE	1: _mcRelative	0 ^{*2}	选择移动方法。 1: 相对值定位

*1. 指令单位请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. 如果是有效范围为枚举型的变量初始值，实际不是数值，而是列举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
OutputtedOffset Position	位置偏置输出值	LREAL	负数、正数、“0”	指定已加到指令当前位置的位置偏置。 Active 为 TRUE 时更新值。 CommandAborted 或 Error 为 TRUE 时，停止更新，保持值。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*1	发生异常时，输出错误代码。16#0000 表示正常。

*1. □□ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	开始了指令时	<ul style="list-style-type: none"> Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 正在执行的同步控制指令停止时 由其他指令进行运动指令多重启动，本指令中止时 其他指令上发生异常，中止了本指令时 正在发生轴异常时，启动了本指令 正在执行 MC_Stop 指令时，启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

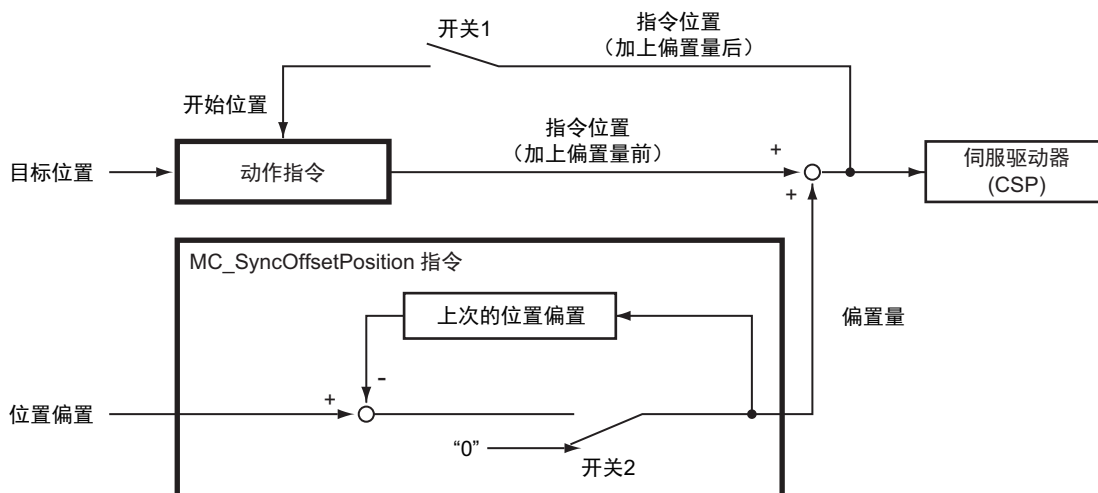
输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定轴。*1 *2

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis***”）或系统定义变量的轴变量名称（_MC_AX[*]、_MC1_AX[*]、_MC2_AX[*]）。

*2. 请指定分配了 NX 系列编码器输入单元的编码器轴。

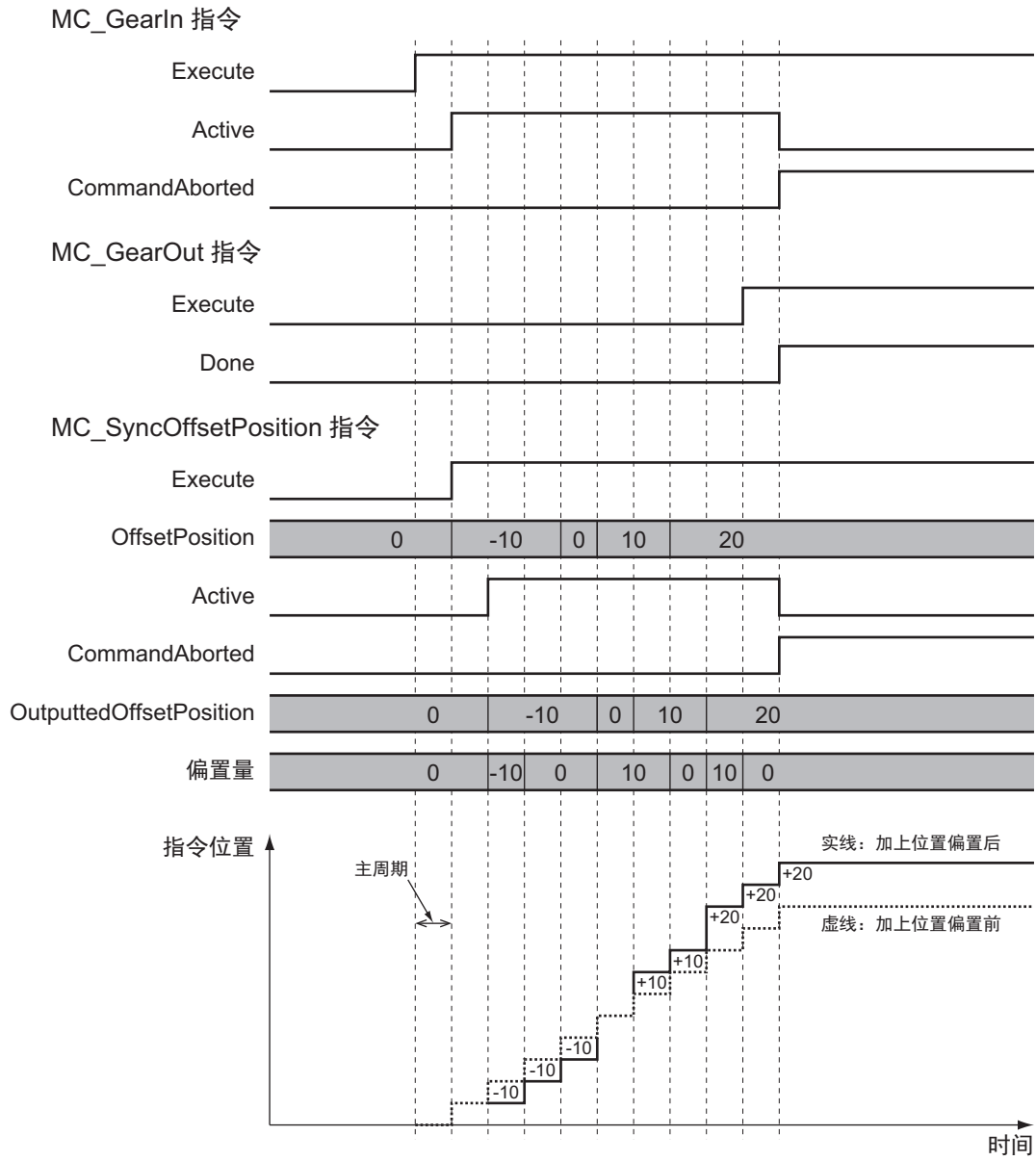
功能说明

- 本指令可针对同步控制中的从轴，将根据输入变量 OffsetPosition（位置偏置）的值计算得到的偏置量加上指令当前位置，输出到伺服驱动器中。
- 输出变量 Active（控制中）为 TRUE 期间，可变更输入变量 OffsetPosition（位置偏置）的值。
- 本指令的 Active（控制中）为 TRUE 期间，OffsetPosition（位置偏置）将本指令启动之时作为起点。详情请参考「重复执行本指令的启动和停止时 (P.3-427)」。



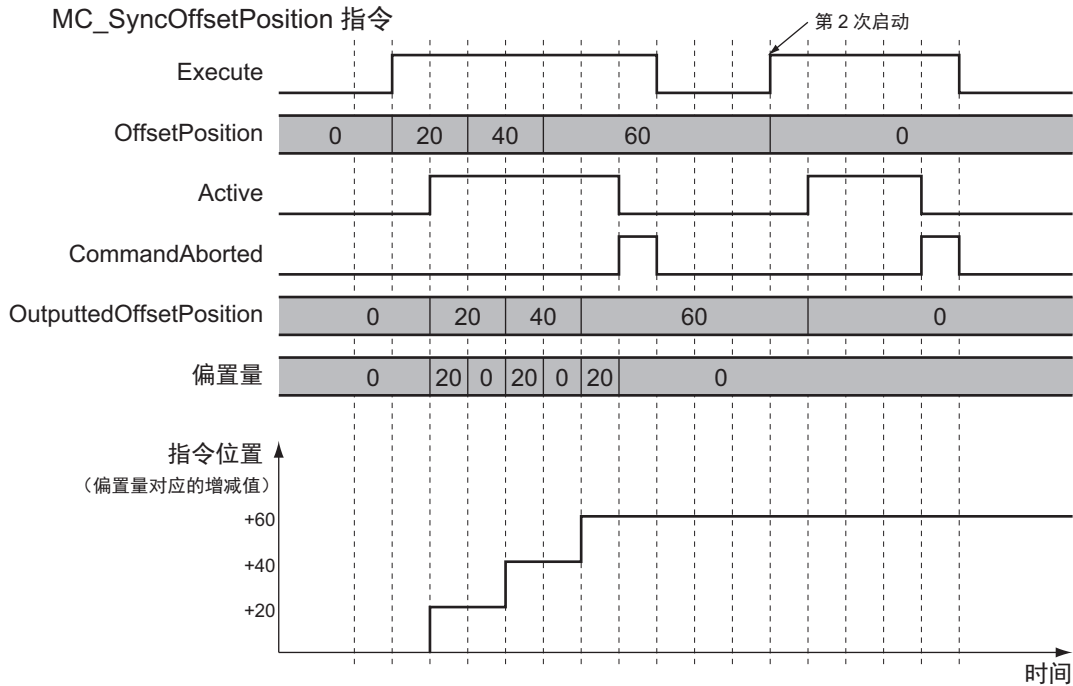
- 开关 1 仅在动作指令 Execute（启动）的上升时，变为 ON 一次。
 - 开关 2 在 MC_SyncOffsetPosition 指令的 Active（控制中）为 TRUE 时变为 ON。
 - 开关 2 为 OFF 时，加到指令位置上的偏置量为“0”。
- 本指令仅可对同步控制中的从轴执行。
 - 启动本指令后，若对象同步控制指令的 Done（完成）、CommandAborted（执行中断）或 Error（错误）变为 TRUE，则本指令的 CommandAborted（执行中断）变为 TRUE。
 - 本指令的 CommandAborted（执行中断）或 Error（错误）变为 TRUE 时，指令位置保持之前相加后的偏置值。

- 正在主固定周期任务中执行 MC_GearIn (齿轮动作开始) 指令时执行本指令, 然后 MC_GearIn (齿轮动作开始) 指令的 CommandAborted (执行中断) 变为 TRUE 时, 时序图如下所示。



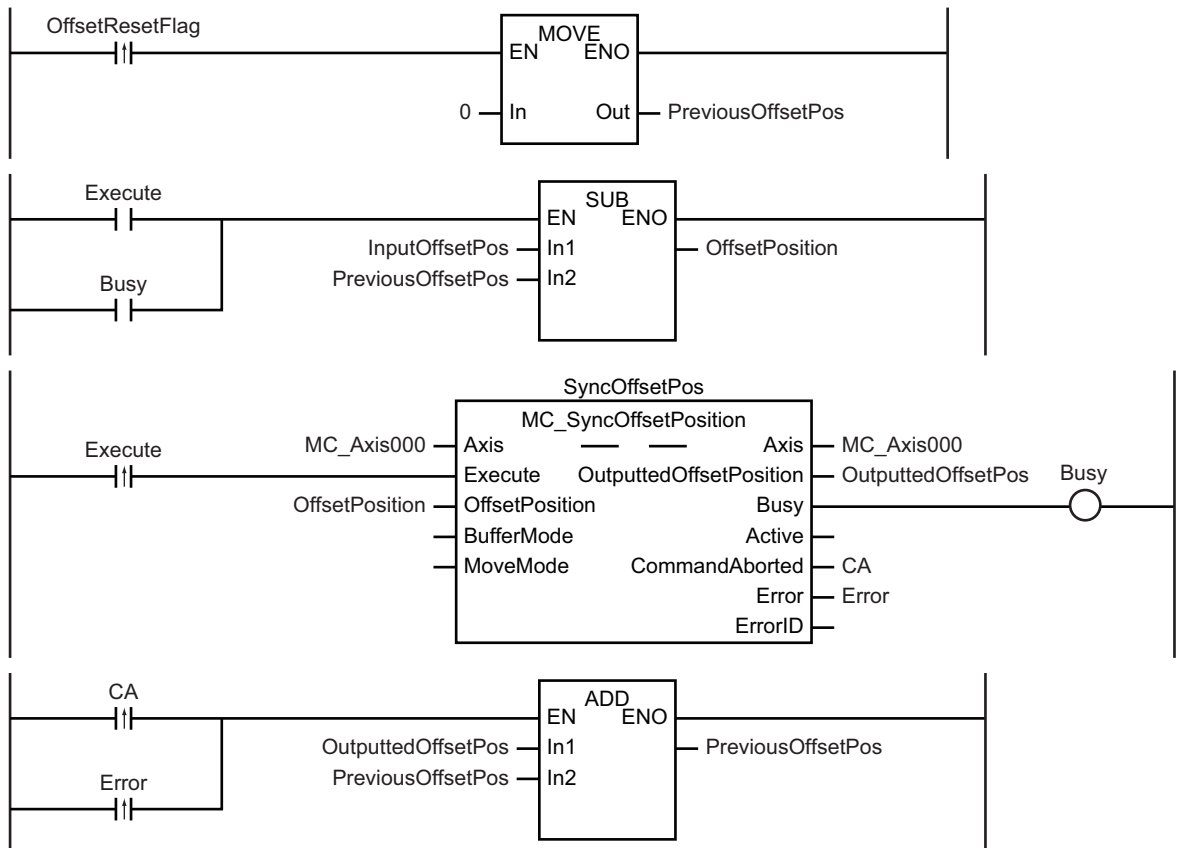
● 重复执行本指令的启动和停止时

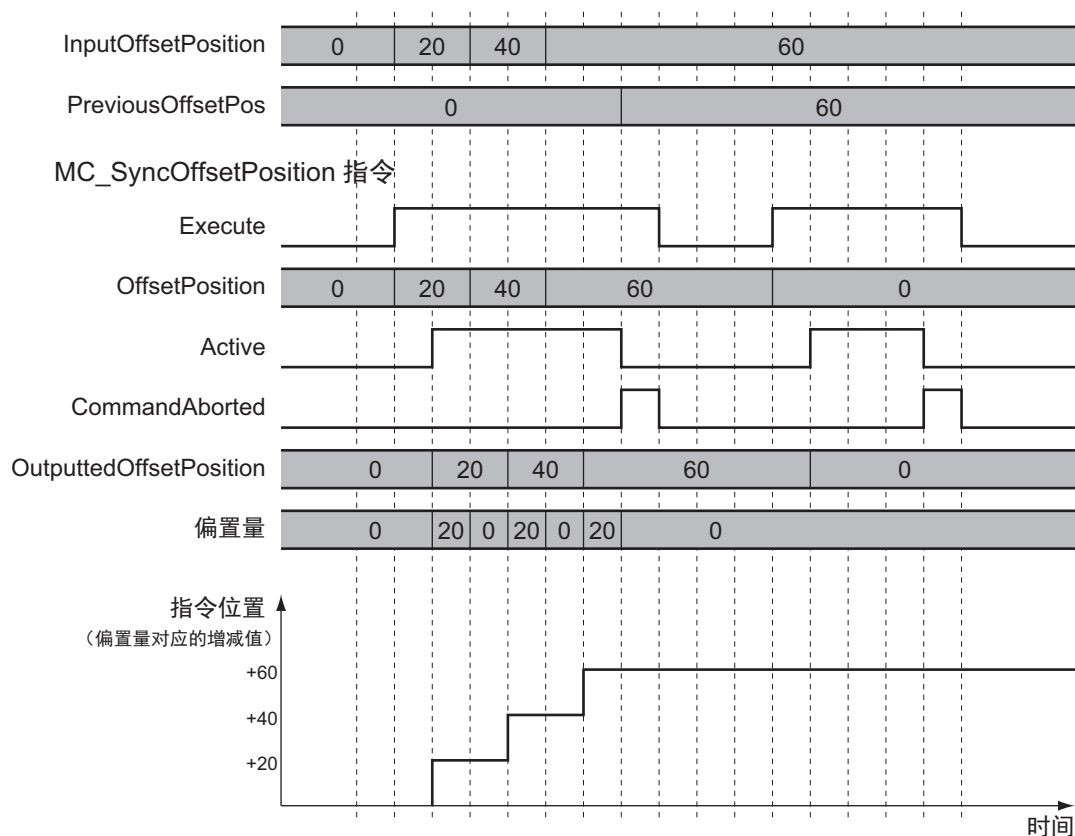
若在本指令中断执行后再次执行，输入变量 OffsetPosition（位置偏置）的基点将变为“0”。第2次启动时，若将 OffsetPosition（位置偏置）指定为“0”，则加到指令位置上的偏置量为“0”。



若在本指令中断执行后再次执行，要以与上次指令相同的基点发出指令时，需要使用输出变量 OutputtedOffsetPosition（位置偏置输出值）。

如以下示例程序所示，通过 OutputtedOffsetPosition（位置偏置输出值）参照执行中断时的位置偏置，从本指令中指定的输入参数的变量中减去。





● 可执行的轴状态

除 Axis（轴）中指定的轴状态为 MC_CombineAxes（加减法定位）指令执行中外，本指令仅在 Status.Synchronized（同步动作中）为 TRUE 时可以执行。若在其他状态下执行，将发生异常。

● 指令位置的处理

轴的指令位置按加上位置偏置后的值进行管理。因此，因位置偏置中设定的值不同，可能发生与以下项目相关的异常。

- 轴参数的动作设定
 - 最高速度
 - 最大加速度
 - 最大减速度
 - 速度警告值
 - 加速度警告值
 - 减速度警告值
 - 位置检查时间
- 轴参数的极限设定
 - 软件限位功能
 - 位置偏差超限值
 - 位置偏差警告值
- 指令位置溢出
- 指令位置下溢

时序图

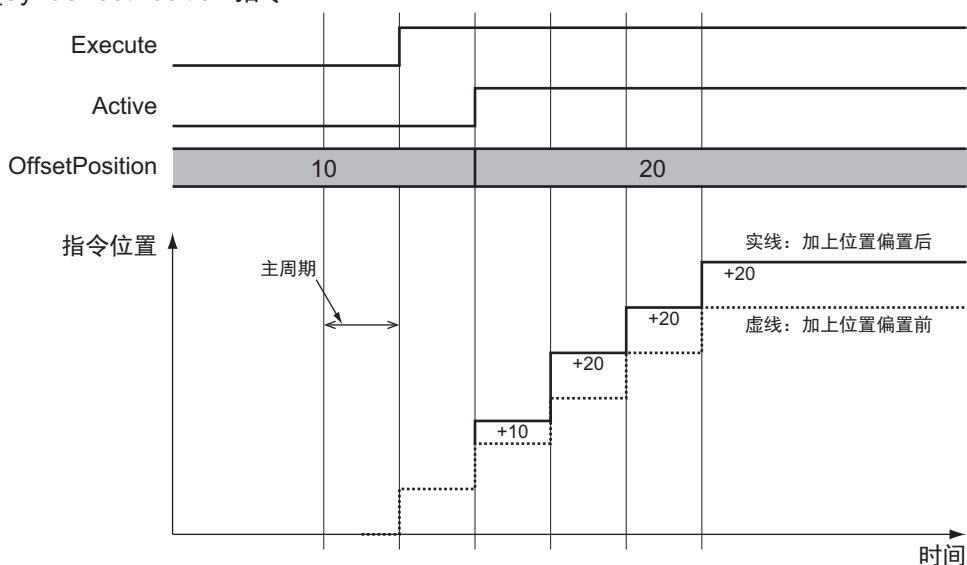
执行了本指令时的位置偏置反映时间如下所示。

● 在主固定周期任务或固定周期任务（执行优先级 5）中记述本指令时

输入中指定的位置偏置将在下一任务周期输出到伺服驱动器。

在主固定周期任务中记述本指令的示例如下。固定周期任务（执行优先级 5）中亦同。

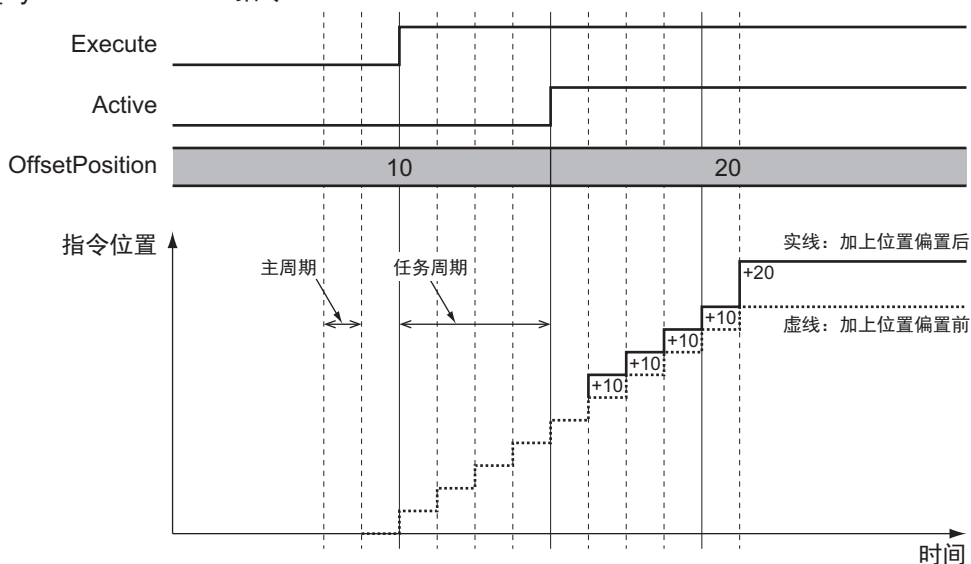
MC_SyncOffsetPosition 指令



● 在固定周期任务（执行优先级 16）中记述本指令时

输入中指定的位置偏置将在下一固定周期任务（执行优先级 16）+ 主周期的 1 个周期后，输出到伺服驱动器。

MC_SyncOffsetPosition 指令



运动指令重新执行

无法重新执行本指令。

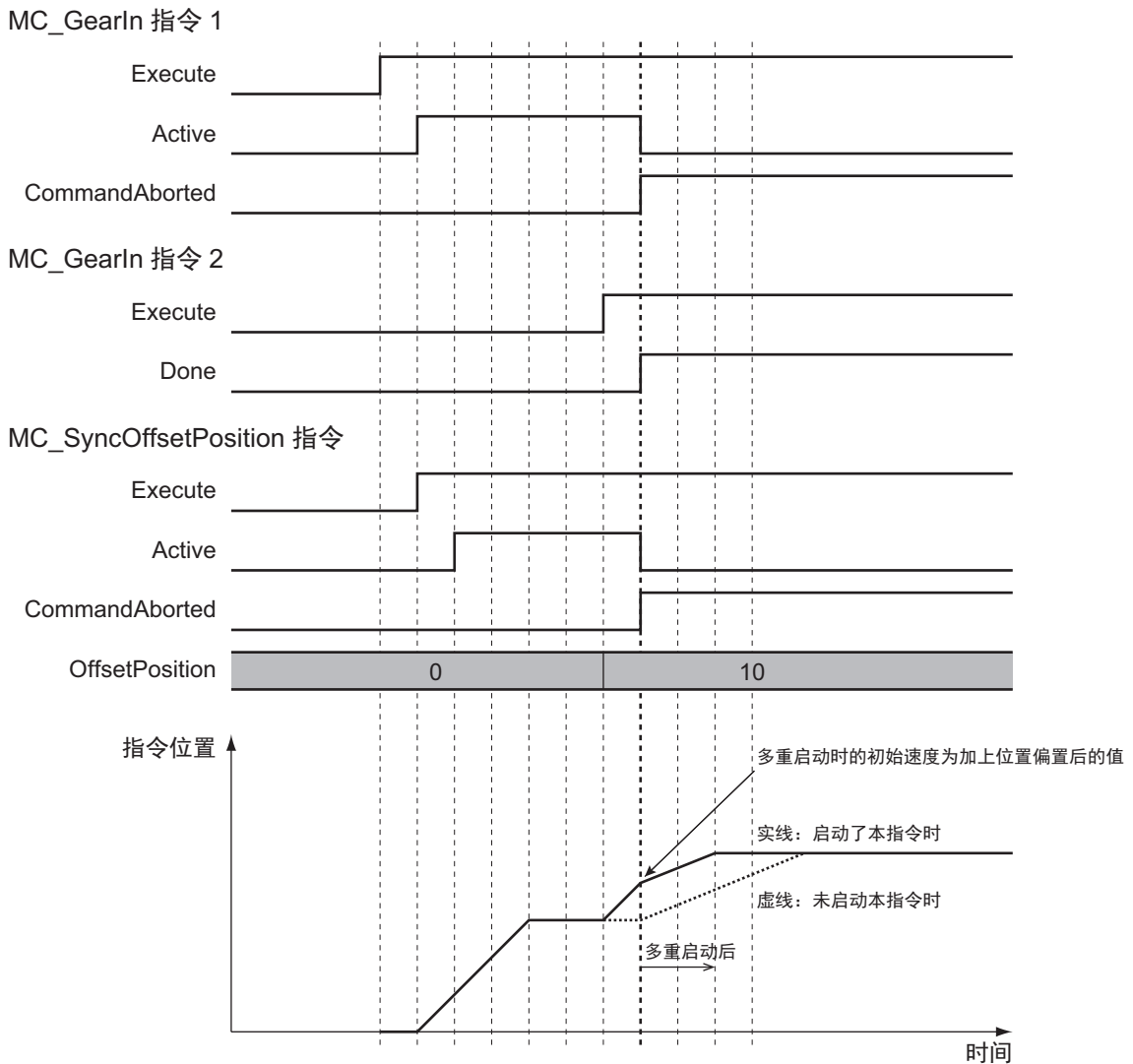
若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 对象同步控制指令的多重启动

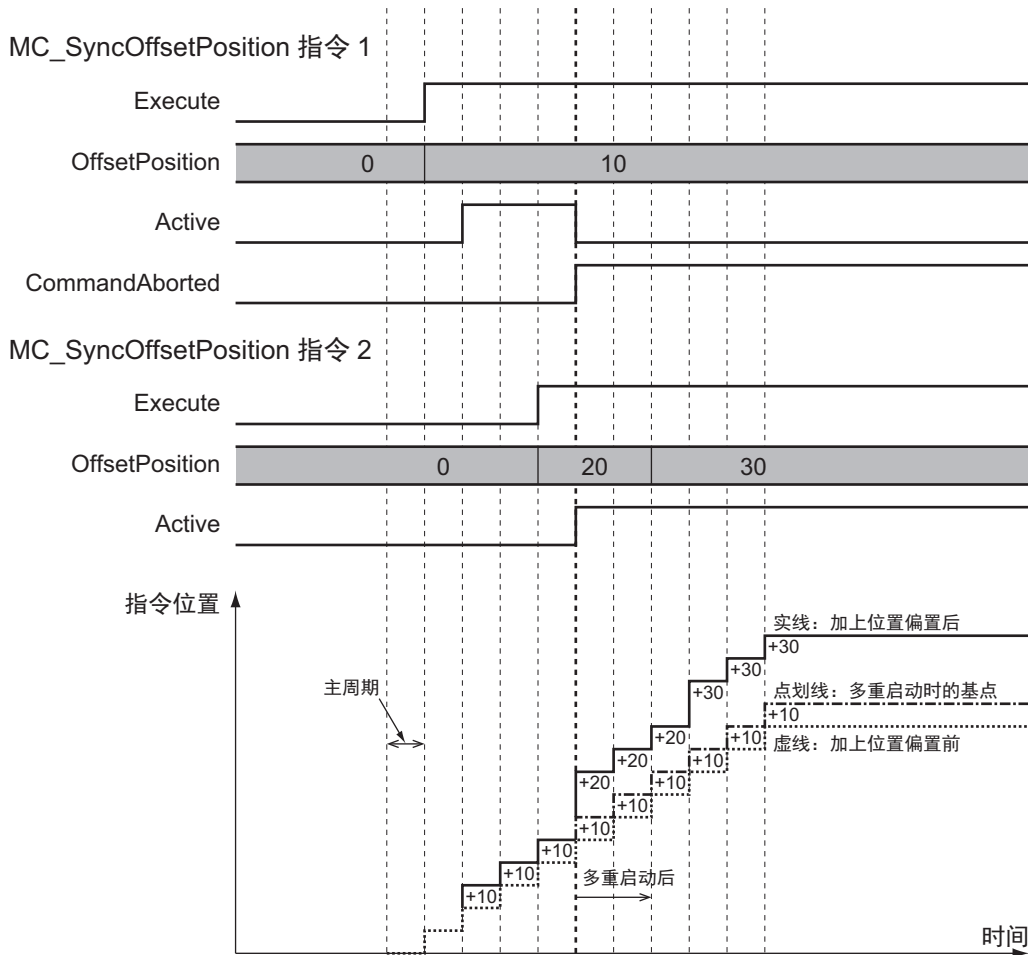
对本指令的对象同步控制指令执行多重启动时，本指令的 CommandAborted (执行中断) 将变为 TRUE。但是，以共混方式多重启动 MC_CamIn (凸轮动作开始) 指令时，本指令的 CommandAborted (执行中断) 保持 FALSE，继续执行处理。此时的初始速度为加上 OffsetPosition (位置偏置) 后的值。对象同步控制指令正在通过 MC_GearIn (齿轮动作开始) 指令执行本指令时，若多重启动了其他实例的 MC_GearIn (齿轮动作开始) 指令，时序图如下所示。



● 本指令的多重启动

本指令多重启动时，执行中实例的 CommandAborted（执行中断）将变为 TRUE，执行后来执行的实例。后面执行的实例基点为加上先执行的实例中断前 1 个周期输出的位置偏差后的位置。本指令的多重启动不会导致其他指令执行中断。

本指令之间多重启动时的时序图如下所示。

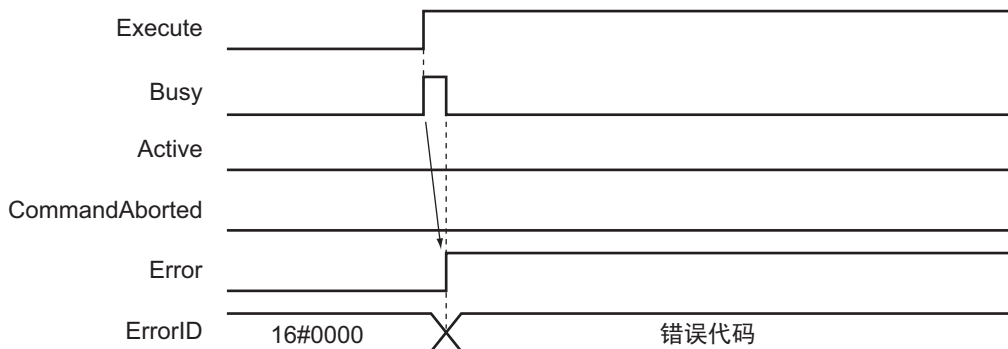


异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

4

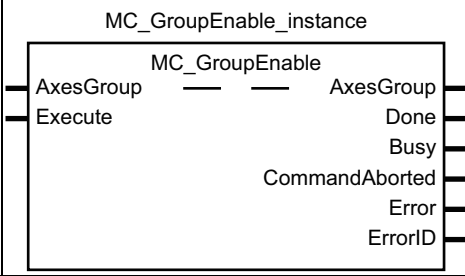
轴组指令

下面介绍 MC 功能模块执行多轴协调控制所需的指令。

MC_GroupEnable	4-2
MC_GroupDisable	4-6
MC_MoveLinear	4-11
MC_MoveLinearAbsolute	4-37
MC_MoveLinearRelative	4-40
MC_MoveCircular2D	4-43
MC_GroupStop	4-67
MC_GroupImmediateStop	4-75
MC_GroupSetOverride	4-79
MC_GroupReadPosition	4-83
MC_ChangeAxesInGroup	4-87
MC_GroupSyncMoveAbsolute	4-91
MC_GroupReset	4-97

MC_GroupEnable

将轴组设为有效。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_GroupEnable	轴组有效	FB		<pre>MC_GroupEnable_instance (AxesGroup := 《参数》, Execute := 《参数》, Done => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、 FALSE	FALSE	上升时开始指令。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、 FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、 FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	指令完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	本指令中止时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
AxesGroup	轴组	_sGROUP_REF	—	指定轴组。 ^{*1}

*1. 请使用 Sysmac Studio 的轴组基本设定画面中创建的用户定义变量的轴组变量名称（默认“MC_Group***”）或系统定义变量的轴组变量名称（_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*]）。

功能说明

介绍本指令的功能前, 先介绍轴组的状态。

- 轴组有 GroupEnable（组有效）和 GroupDisable（组无效）两种状态。

状态	说明
GroupEnable (组有效)	多轴协调控制为有效状态。 除 MC_ChangeAxesInGroup（轴组构成轴写入）指令外, 可执行所有多轴协调指令。
GroupDisable (轴组无效)	多轴协调控制为无效状态。 多轴协调控制指令只能执行以下指令。 <ul style="list-style-type: none"> MC_GroupEnable（轴组有效）指令 MC_GroupDisable（轴组无效）指令 MC_GroupReset（轴组错误复位）指令 MC_GroupSetOverride（轴组超驰值设定）指令 MC_GroupReadPosition（轴组位置获取）指令 MC_ChangeAxesInGroup（轴组构成轴写入）指令

- 要执行多轴协调控制, 轴组的状态需要为 GroupEnable（轴组有效）。

监视运动控制系统变量的轴组变量，可确认轴组的有效、无效。

下面以 `_MC_GRP[0-63]` 为例，介绍轴组变量。

`_MC1_GRP[0-63]`、`_MC2_GRP[0-63]` 亦同。

变量名称	名称	说明
<code>_MC_GRP*.Status.Ready</code>	轴组有效	轴组处于停止状态且已根据轴组指令完成启动准备时变为 TRUE。
<code>_MC_GRP*.Status.Disabled</code>	轴组无效	轴组无效、处于停止状态时变为 TRUE。

(注) “`_MC_GRP*.`” 的 “*” 中填入 0 ~ 63 的数字。

基本功能

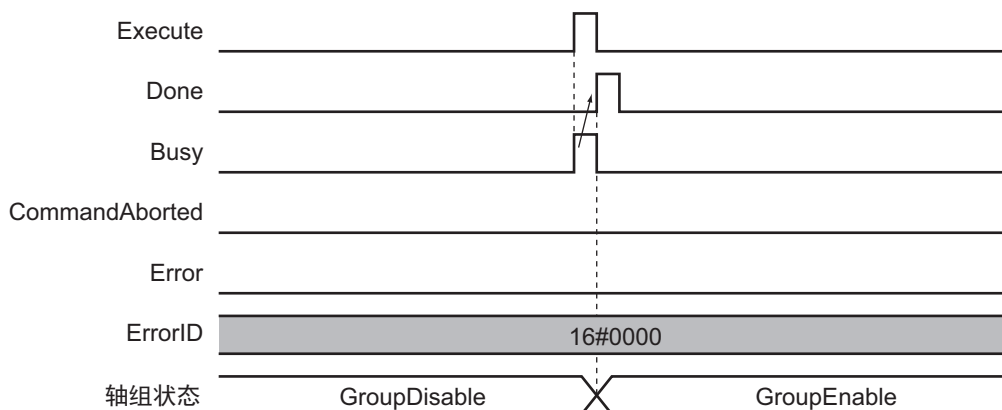
- `MC_GroupEnable` (轴组有效) 指令可将 `AxesGroup` (轴组) 中指定的轴组状态变为 `GroupEnable` (组有效)。
- 设为 `GroupEnable` (组有效) 的状态后，轴组将可以执行所有多轴协调指令。
- 轴组中可指定的轴种类仅为“伺服轴”、“虚拟伺服轴”，包含其他轴种类时，将发生异常。
- 属于轴组的所有轴均需处于停止状态。
轴是否处于停止状态，可根据轴变量 `Status.Disabled` (轴无效) 或 `Status.Standstill` (停止中) 是否为 TRUE 来判断，为 TRUE，则处于停止状态。
- 若轴属于其他轴组，且轴组为有效，将不执行 `MC_GroupEnable` (轴组有效) 指令，并发生异常。
- 轴组变为有效后，轴组所属轴的状态将变为“多轴协调动作中”。
轴变量 `Status.Coordinated` (多轴协调动作中) 变为 TRUE。
- 轴组变为无效的条件为 `MC_GroupDisable` (轴组无效) 指令的执行、因程序模式转移而停止运行及 MC 调试开始时。



使用注意事项

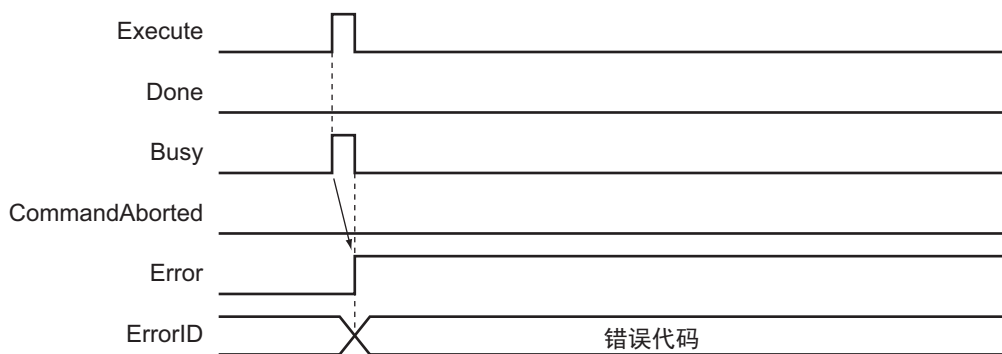
- 要使用轴组，需要通过 Sysmac Studio 创建轴组，然后将设定下载到 CPU 单元中。程序中无法变更轴组的轴。
在 CPU 单元版本 1.01 以上和 Sysmac Studio Ver.1.02 以上的组合中，可使用 `MC_ChangeAxesInGroup` (轴组构成轴写入) 指令，临时进行变更。
- “下载”时使用 Sysmac Studio 的“同步”功能。

时序图



异常

执行 MC_GroupEnable（轴组有效）指令的过程中发生异常时，Error（错误）将变为 TRUE。参照输出到 ErrorID（错误代码）的值，可找出异常的原因。



参考

即使轴组发生异常，组中的各个轴也不会变为异常。

● 错误代码

指令中发生的异常请参考 □ 「A-1 错误代码一览 (P.A-2)」。

MC_GroupDisable

将轴组设为无效。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_GroupDisable	轴组 无效	FB		<pre>MC_GroupDisable_instance (AxesGroup := 《参数》, Execute := 《参数》, Done => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、 FALSE	FALSE	上升时开始指令。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、 FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、 FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	轴组的状态变更为 GroupDisable 时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	没有变为 TRUE 的条件。(Reserved)	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

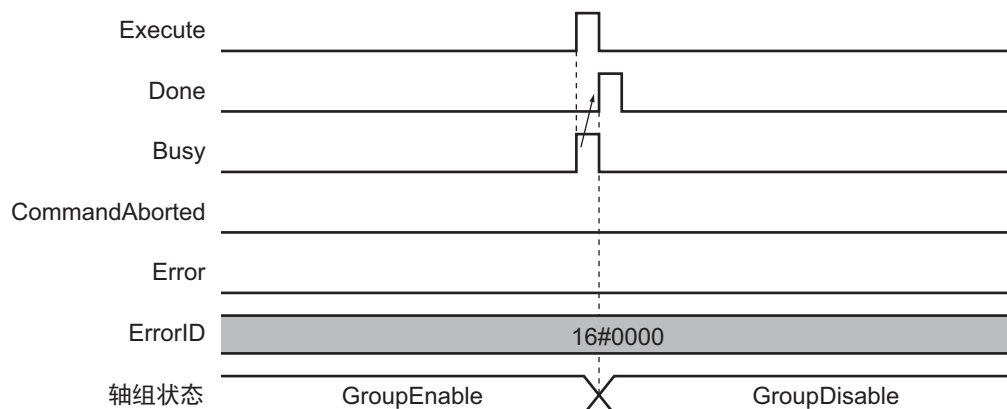
输入输出变量	名称	数据类型	有效范围	内容
AxesGroup	轴组	_sGROUP_REF	—	指定轴组。 ^{*1}

*1. 请使用 Sysmac Studio 的轴组基本设定画面中创建的用户定义变量的轴组变量名称（默认“MC_Group***”）或系统定义变量的轴组变量名称（_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*]）。

功能说明

- 本指令可将轴组状态设为无效。
轴组的状态说明请参考 □「功能说明 (P.4-3)」。
- MC_GroupDisable（轴组无效）指令可将 AxesGroup（轴组）中指定的轴组状态变为 GroupDisable（轴组有效）。
设为 GroupDisable（组无效）的状态后，轴组将无法接收多轴协调指令。
- 将轴组状态设为 GroupDisable（组无效）时，指定 AxesGroup（轴组）的缓冲指令将清除。
- 轴组在运行停止时也无效。
轴组变为无效后，属于轴组的轴的状态将从 Status.Coordinated（多轴协调动作中）为 TRUE 的状态变为各轴的状态。
要知道各轴的状态，请确认轴变量 Status（轴状态）。

时序图

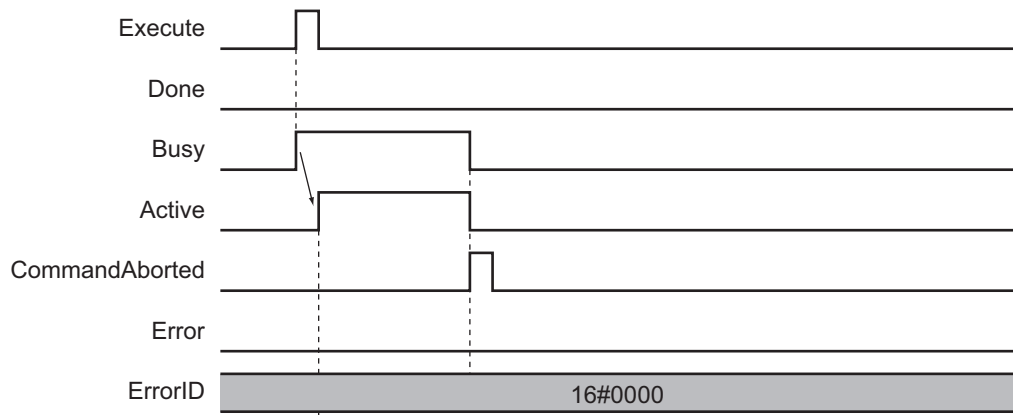


对多轴协调控制中的 AxesGroup（轴组）执行了 MC_GroupDisable（轴组无效）指令时，多轴协调控制将 CommandAborted（执行中断）。

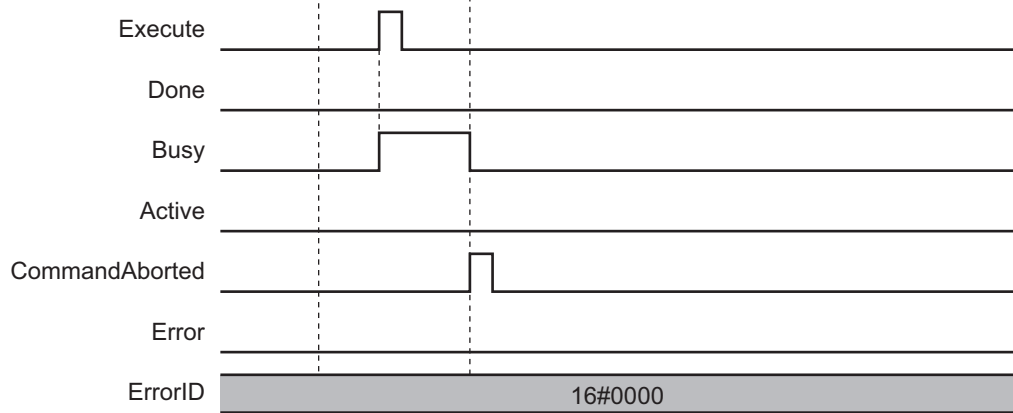
此时，缓冲的多轴协调控制指令将同样 CommandAborted（执行中断）。

动作中的轴将以各轴的最大减速度减速停止。

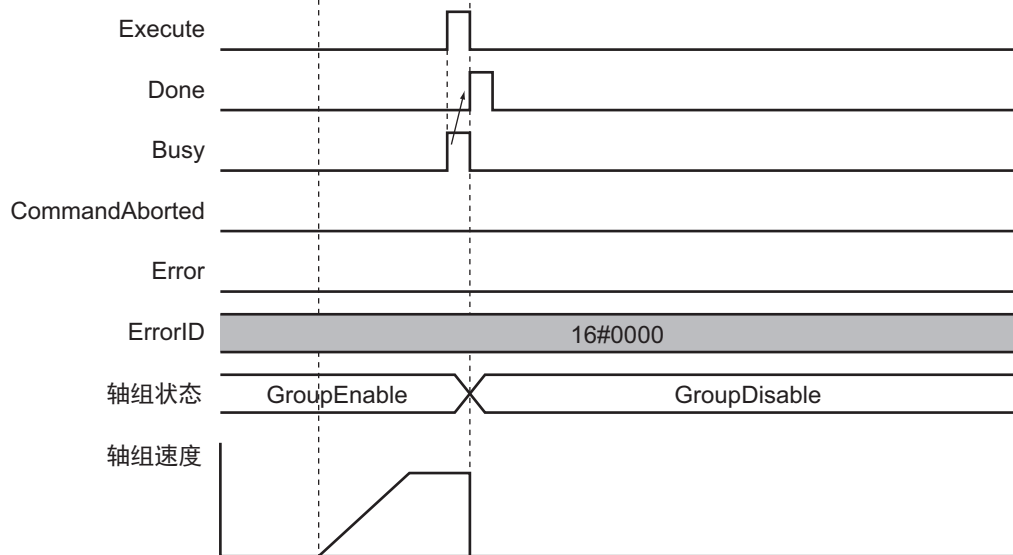
多轴协调控制 1 (Ex. MC_MoveLinearAbsolute)



多轴协调控制 2 (Ex. MC_MoveLinearAbsolute.Buffered)

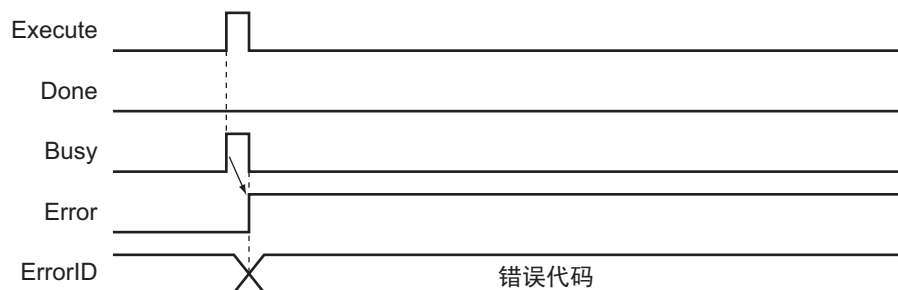


MC_GroupDisable



异常

执行 MC_GroupDisable（轴组无效）指令的过程中发生异常时，Error（错误）将变为 TRUE。
参照输出到 ErrorID（错误代码）的值，可找出异常的原因。



● 错误代码

指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

MC_MoveLinear

进行直线插补。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_MoveLinear	直线插补	FB		<pre> MC_MoveLinear_instance (AxesGroup := 《参数》, Execute := 《参数》, Position := 《参数》, Velocity := 《参数》, Acceleration := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, CoordSystem := 《参数》, BufferMode := 《参数》, TransitionMode := 《参数》, MoveMode := 《参数》, Done => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》); </pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Position	目标位置	ARRAY [0..3] OF LREAL	负数、正数、“0”	0	指定直线插补的目标位置。 单位为 [指令单位]。*1
Velocity*2	目标速度	LREAL	正数	0	指定目标速度。 单位为 [指令单位 /s]。*1
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度。 单位为 [指令单位 /s ²]。*1
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。 单位为 [指令单位 /s ²]。*1
Jerk	跃度	LREAL	正数或“0”	0	指定跃度。 单位为 [指令单位 /s ³]。*1
CoordSystem	坐标系	_eMC_ COORD_ SYSTEM	0: _mcACS	0*3	指定坐标系。 0: 轴坐标系 (ACS)

输入变量	名称	数据类型	有效范围	初始值	内容
BufferMode	缓冲模式选择	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	指定运动指令多重启动时的动作。 0: 中断 1: 缓冲 2: 低共混 3: 前共混 4: 后共混 5: 高共混
Transition Mode	转换模式 (变化模式)	_eMC_ TRANSITION _MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 *3	指定动作的路径。 0: 转换无效 10: 重叠
MoveMode	移动方法选择	_eMC_MOVE _MODE	0: _mcAbsolute 1: _mcRelative	0 *3	选择移动方法。 0: 绝对值定位 1: 相对值定位

*1. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. 请务必设定目标速度。若不设定而直接动作，将发生异常。

*3. 如果是有效范围为列举型的变量初始值，实际不是数值，而是列举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、 FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、 FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、 FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* □ □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	定位完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	轴发生移动时	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动 (中断), 本指令中止时 因发生异常, 本指令中止时 正在发生异常时, 启动了本指令 正在执行 MC_GroupStop 指令时, 启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
AxesGroup	轴组	_sGROUP_REF	—	指定轴组。 ^{*1}

*1. 请使用 Sysmac Studio 的轴组基本设定画面中创建的用户定义变量的轴组变量名称 (默认“MC_Group***”)或系统定义变量的轴组变量名称 (_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])。

功能说明

- 进行 2 轴～ 4 轴的直线插补。



使用注意事项

- 轴组构成轴中有任一轴为原点未确定时，将发生“原点未确定状态下指令启动异常（错误代码：5466 Hex）”。
- 进行直线插补时，属于该轴组的逻辑轴中，任意一个的极限输入为“ON”时，指令将无法启动。

指令的详情

下面介绍本指令的详情。

● 直线插补的步骤

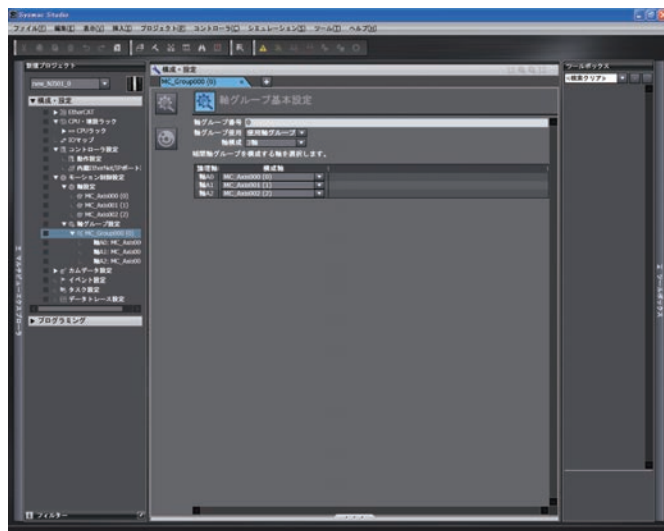
直线插补按以下步骤执行。

1 登录要插补的轴组

- 确定要插补的轴组。
轴组是指 `_MC_GRP[*]` 或 `_MC1_GRP[*]`、`_MC2_GRP[*]`。
- 在该轴组变量的轴构成中指定轴构成。可指定的轴数为 2～4 轴。
- 在该轴组变量的轴选择中，指定要进行插补的轴组合。
- 轴不使用轴编号（轴 0～轴 255），而使用逻辑轴（轴 A0～轴 A3）。
- 在轴选择中，从前往后分别对逻辑轴 A0～A3 指定轴编号 0～255。

逻辑轴	轴编号
轴 A0	轴 0～轴 255
轴 A1	轴 0～轴 255
轴 A2	轴 0～轴 255
轴 A3	轴 0～轴 255

- 在 Sysmac Studio 的轴组设定中，选择要使用的“轴构成”，将轴编号分配到逻辑轴。例如，对拥有 3 个轴的轴组“MC_Group000”，登录 MC_Axis000、MC_Axis001、MC_Axis002 的示例如下。





使用注意事项

轴组中包含编码器轴或虚拟编码器轴时，将发生“轴组种类不合适（错误代码：543D Hex）”，动作结束。
请务必选择伺服轴或虚拟伺服轴。

2 轴组的有效化

- 将轴组的各构成轴设为伺服 ON 状态，并使其变为原点确定状态。
- 执行 MC_GroupEnable（轴组有效）指令，将登录的轴组设为有效。

直线插补指令变为可以使用。

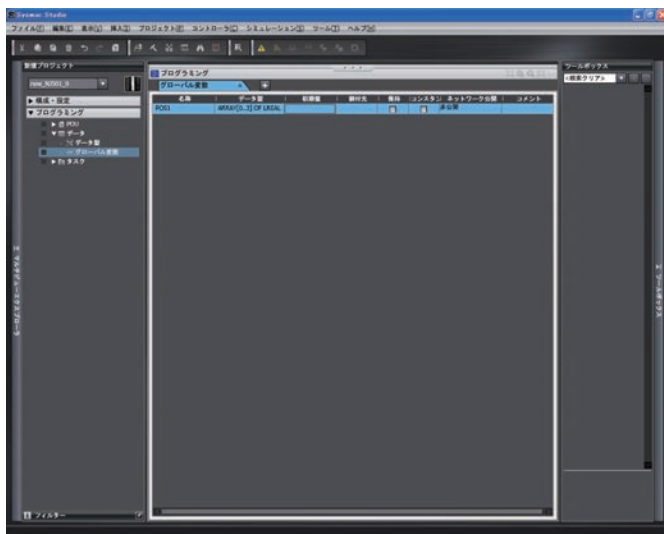
● Position（目标位置）

- 在 Position（目标位置）中指定与轴组变量的轴选择中指定的轴组数量对应的目标位置。
- 代入到 Position（A0, A1, A2, A3）的变量需要在 Sysmac Studio 上创建 1×4 的排列变量。变量名称可任意命名。

在该排列中代入各轴的目标位置。

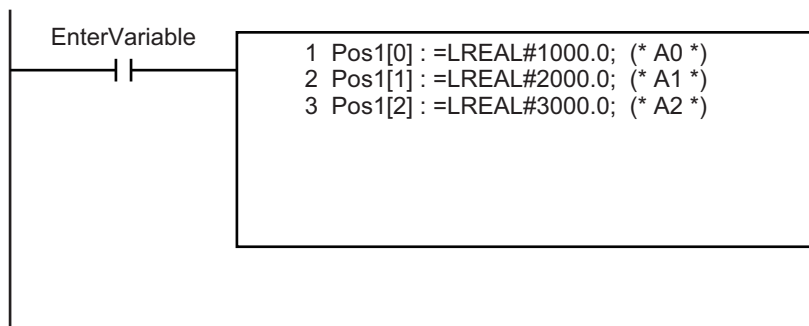
属于所用轴组的轴数少于 4 个时，也必须声明 1×4 个排列。

通过 Sysmac Studio 声明排列变量“Pos1”时的示例如下。



此外，使用联机 ST 在“Pos1”中代入目标位置的示例如下。

图中，A0～A2 的各轴中代入了目标位置（1000.0, 2000.0, 3000.0）。



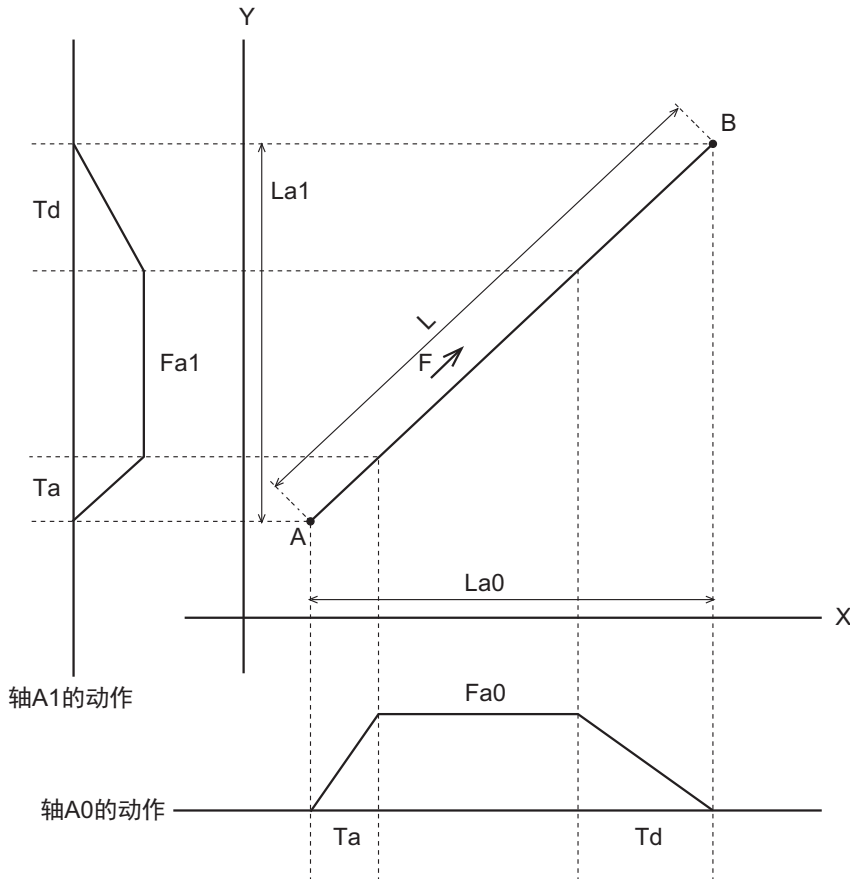
- 将计数模式选择设定为[选择模式]的轴指定为插补轴时，若指定绝对位置，目标值与 Direction（方向选择）为[无方向指定]时相同。

详情请参考 □「Direction（方向选择）(P.3-51)」。

● **Velocity (目标速度)、Acceleration (加速度)、Deceleration (减速度)、Jerk (跃度)**

- 直线插补时的插补速度、加速度、减速度、跃度在 Velocity、Acceleration、Deceleration、Jerk 中指定。
- 直线插补时，将直线分解到各轴后进行动作。

作为示例，如果要表示使用 2 个轴从 A 点到 B 点进行直线插补，则如下图所示。



4 个轴的直线插补时，各轴的目标速度根据插补速度和各轴的移动距离不同，具体如下所示。

- F : 指定的插补传送速度
 - Fa0 : 将 F 展开到轴 A0 时的插补传送速度
 - Fa1 : 将 F 展开到轴 A1 时的插补传送速度
 - Fa2 : 将 F 展开到轴 A2 时的插补传送速度
 - Fa3 : 将 F 展开到轴 A3 时的插补传送速度
 - Ta : 插补加速时间
 - Td : 插补减速时间
 - L : 指定轨迹的移动量
 - La0、La1、La2、La3: 轴 A0、轴 A1、轴 A2、轴 A3 的移动量
- 以上情况下，L, Fa0, Fa1, Fa2, Fa3 可用以下公式表示。

$$Fa0 = F \times \frac{La0}{L}$$

$$Fa1 = F \times \frac{La1}{L}$$

$$Fa2 = F \times \frac{La2}{L}$$

$$Fa3 = F \times \frac{La3}{L}$$

$$L = \sqrt{La0^2 + La1^2 + La2^2 + La3^2}$$

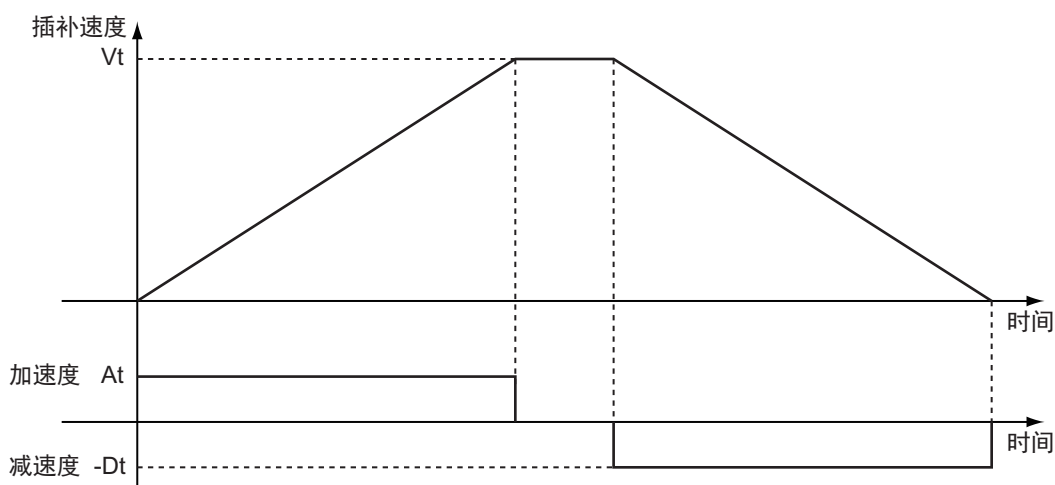
Velocity（目标速度）

- 若 Velocity（目标速度）指定为“0”，将发生插补速度指定异常。若指定轴组的轴为动作中，则所有轴停止。
- 计算得到的 Velocity（目标速度）Fa0～Fa3 中，任意一个超出各轴的最高速度时，将自动调整 Velocity（目标速度），使任意轴都能以最高速度动作。

Jerk（跃度）

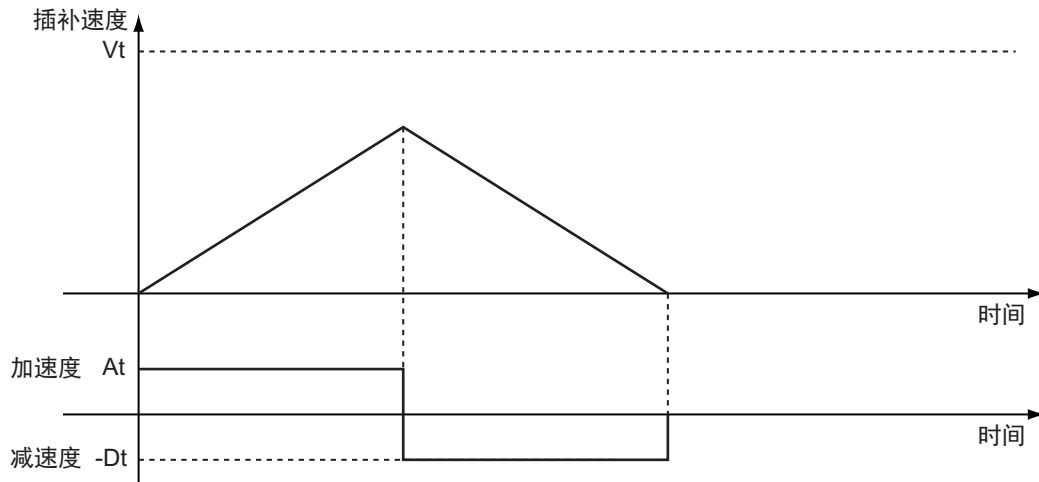
Jerk（跃度）指定为“0”时和指定为“0”以外时的 Acceleration（加速度）、Deceleration（减速度）、Velocity（目标速度）的关系如下所示。

- Jerk（跃度）为“0”时
用加速度 At、减速度 Dt 创建速度的指令值。



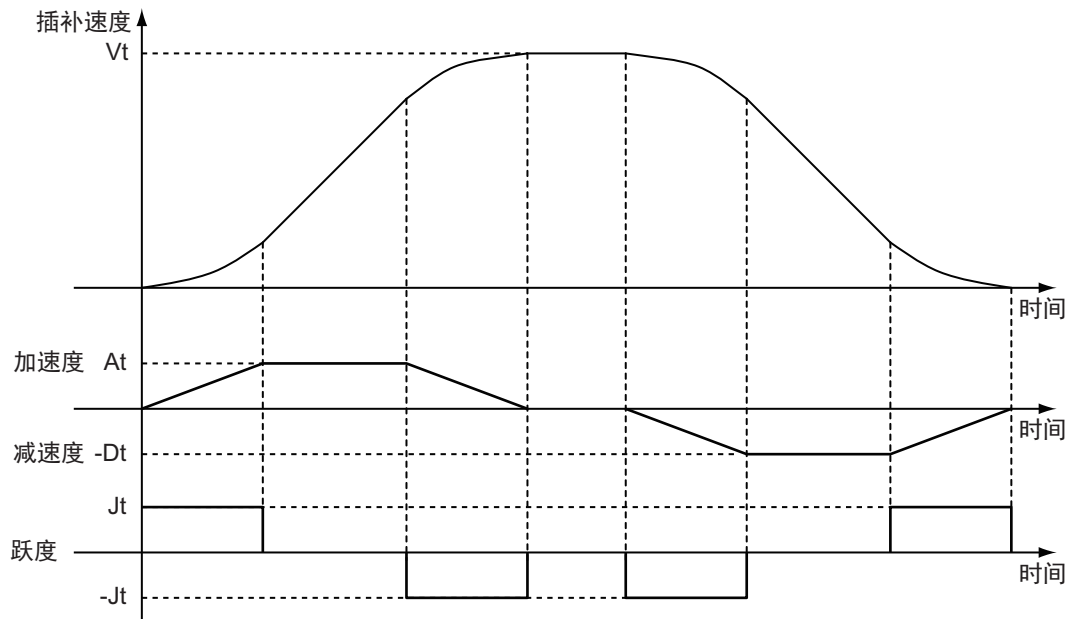
Vt: 插补速度的指定值、At: 加速度的指定值、Dt: 减速度的指定值

- Jerk (跃度) 为“0”且移动距离较短时
插补速度不会达到指定值 V_t (目标速度)。



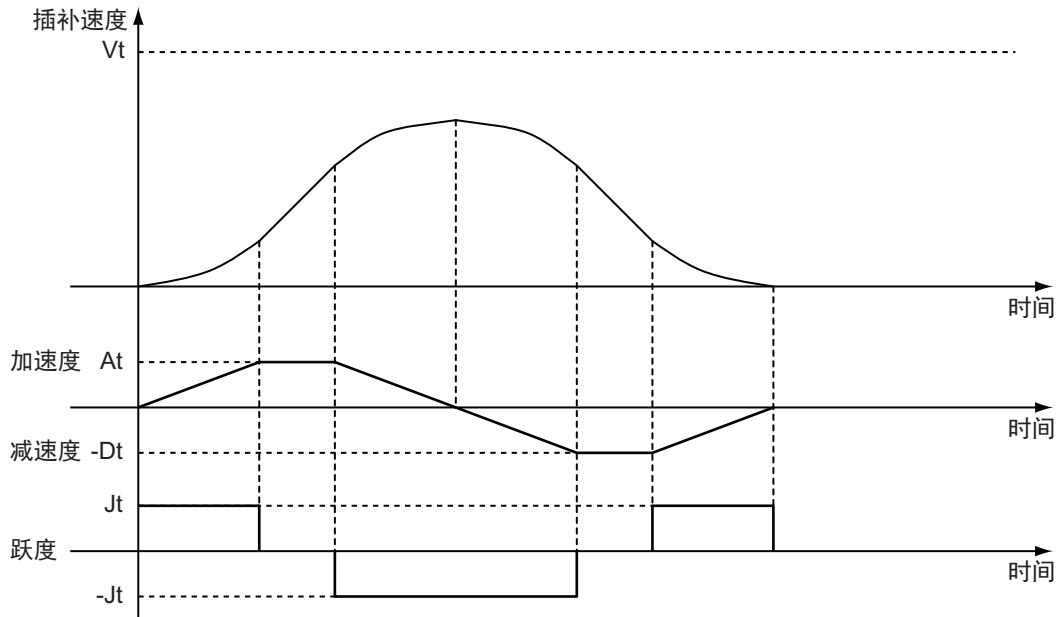
V_t : 插补速度的指定值、 A_t : 加速度的指定值、 D_t : 减速度的指定值

- Jerk (跃度) 为“0”以外时
用加速度的上限 A_t 、减速度的上限 D_t 创建速度的指令值。



V_t : 插补速度的指定值、 A_t : 加速度的指定值、 D_t : 减速度的指定值、 J_t : 跃度的指令值

- Jerk（跃度）为“0”以外且移动距离较短时插补速度不会达到指定值 V_t （目标速度）。



V_t : 插补速度的指定值、 A_t : 加速度的指定值、 D_t : 减速度的指定值、 J_t : 跃度的指令值



参考

- 若 Acceleration（加速度）指定为“0”，将立即按指定的 Velocity（目标速度）动作。
- 若 Deceleration（减速度）指定为“0”，将立即停止。但是，在缓冲模式中指定了共混时，将不停止，在下一个动作时变为指定的目标速度。
详情请参考 □□「BufferMode（缓冲模式选择）(P.4-20)」。
- Acceleration（加速度）或 Deceleration（减速度）指定为“0”时，跃度的设定将变为无效。

● CoordSystem（坐标系）

- 指定进行直线插补的坐标系。
- 仅支持由 2 个以上的轴构成的轴坐标系（ACS）。

● BufferMode（缓冲模式选择）

- 指定前一插补动作和本次插补动作的连接方法。
- 有以下 6 种。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 切换指令后如果轴的动作方向发生反转，将按照轴参数 [反转时动作] 执行反转动作。
缓冲	当前正在执行的指令正常完成后，自动启动已缓冲的本指令。
共混	将当前正在执行的指令到达目标位置时的速度（中继速度）作为启动速度，连续执行已缓冲的本指令。变更当前正在执行的指令动作，使其以中继速度到达目标位置。中继速度的指定方法有以下 4 种。 此外，作为共混的选项，有转换指定（后述）。
低共混	将当前正在执行的目标速度和缓冲中的目标速度中较低的速度作为中继速度。
前共混	将当前正在执行的目标速度作为中继速度。
后共混	将已缓冲的本指令的目标速度作为中继速度。
高共混	将当前正在执行的目标速度和缓冲中的目标速度中较高的速度作为中继速度。

BufferMode（缓冲模式选择）请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● TransitionMode（转换模式）

- 指定前一插补动作和本次插补动作中创建的轨迹的连接方法。
- TransitionMode（转换模式）中，指定为 `_mcTMNone`（转换无效）或 `_mcTMCornerSuperimposed`（重叠）中的一个。
- BufferMode（缓冲模式选择）中指定了共混时，TransitionMode（转换模式）将变为有效。
- 不指定共混，且转换指定为 `_mcTMNone`（转换无效）以外时，将发生异常。

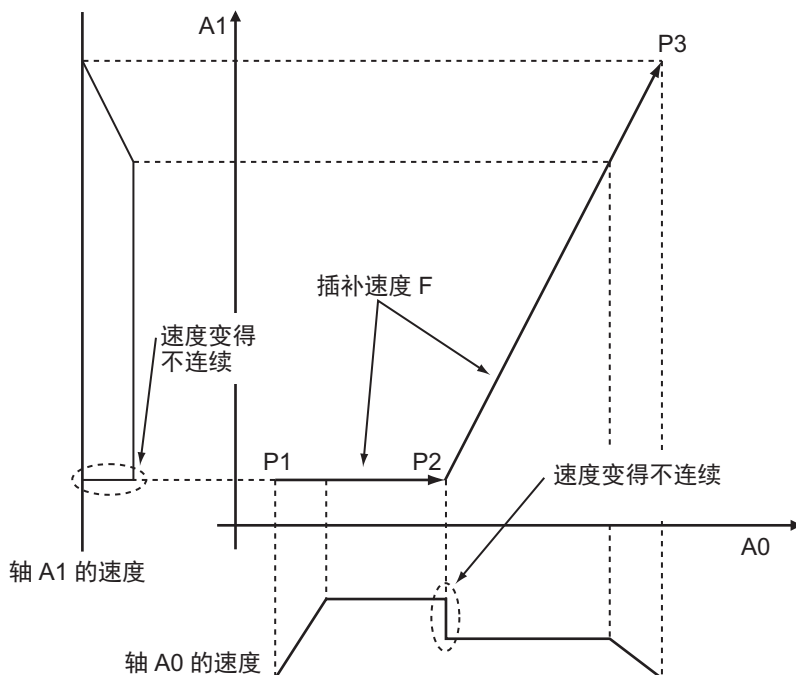
转换无效

由于以轨迹为优先来创建速度指令值，所以各轴的速度指令值可能在每个动作的切换点发生剧烈变化。

动作示例

P1 到 P2、P2 到 P3 的 Velocity（目标速度）、BufferMode（缓冲模式选择）、TransitionMode（转换模式）如下。

- P1 到 P2 的动作: Velocity = F、BufferMode = 中断、TransitionMode = _mcTMNone（转换无效）
- P2 到 P3 的动作: Velocity = F、BufferMode = 后共混、TransitionMode = _mcTMNone（转换无效）
- 从位置 P1 开始移动，通过位置 P2，向位置 P3 以直线插补的方式动作。
- 通过位置 P2 时，维持直线插补的速度 F。因此，如图所示，在位置 P2 上，速度变得不连续。



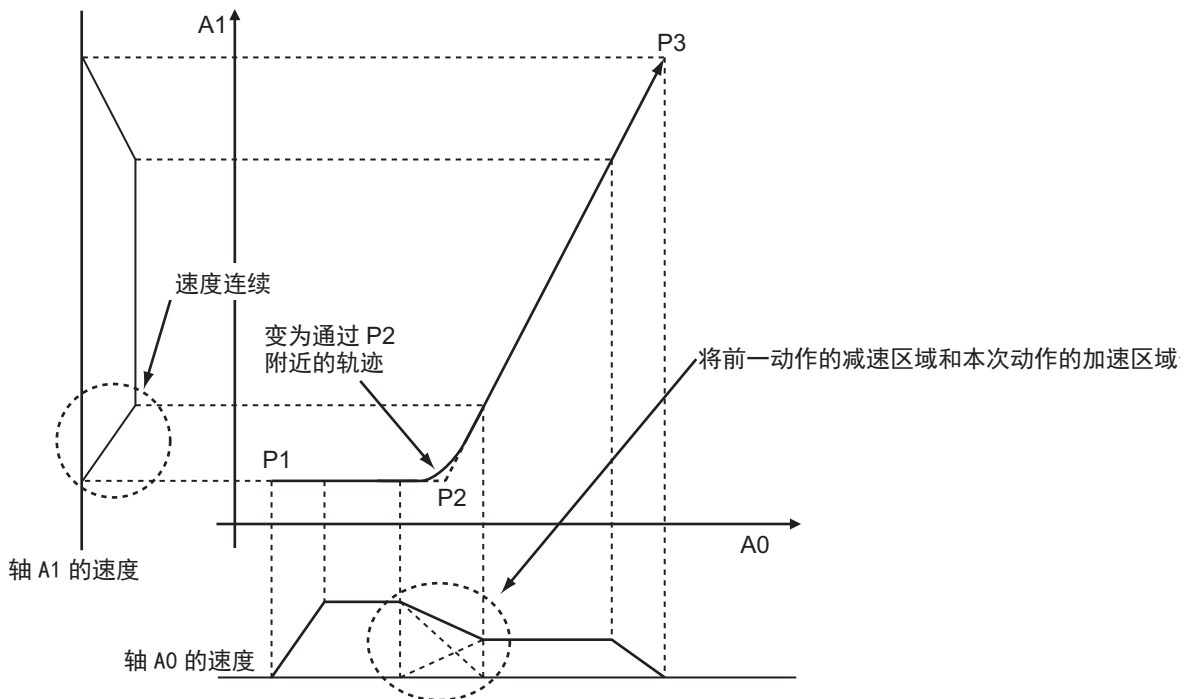
重叠

要使各轴的指令速度变得连续并移动时使用。

动作示例

P1 到 P2、P2 到 P3 的 Velocity（目标速度）、BufferMode（缓冲模式选择）、TransitionMode（转换模式）如下。

- P1 到 P2 的动作: Velocity = F、BufferMode = 中断、TransitionMode = `_mcTMNone`（转换无效）
- P2 到 P3 的动作: Velocity = F、BufferMode = 后共混、TransitionMode = `_mcTMCornerSuperimposed`（重叠）
- 从位置 P1 开始移动，通过位置 P2 附近，向位置 P3 以直线插补的方式动作。
- 为使各轴的指令速度变得连续，将前一动作的减速区域和本次动作的加速区域合成，创建速度指令。因此，本次动作的加速时间与前一动作的减速时间相同。



合成的轨迹变为通过 P2 附近的轨迹。

从 P2 到该轨迹的距离：

- 插补速度变快或前一指令的减速度变小时，将变长
- 插补速度变慢或前一指令的减速度变大时，将变短

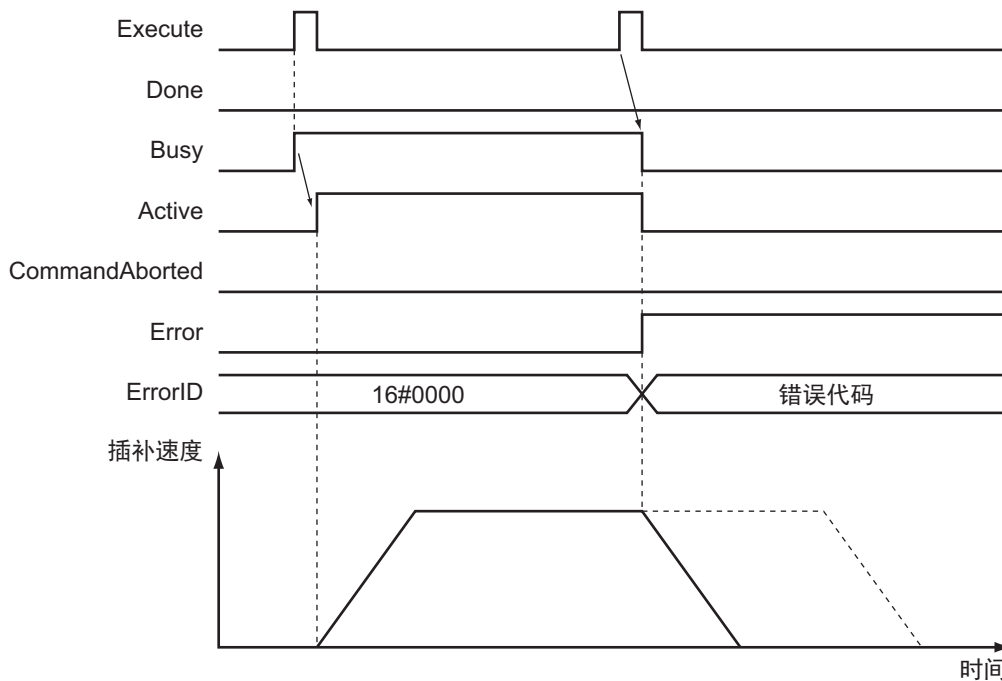


参考

在重叠的区间内，Jerk（跃度）的指定无效。

运动指令重新执行

无法重新执行本指令。若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”，直线插补动作中的所有轴停止。



运动指令多重启动

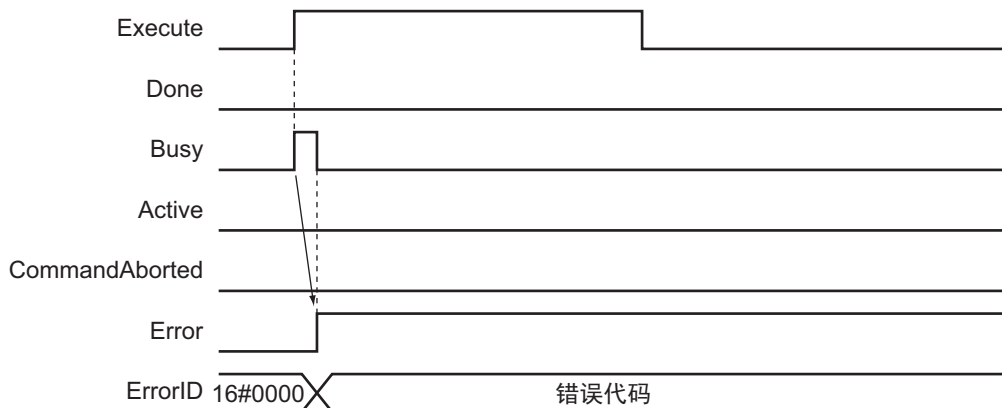
正在执行本指令时，可使用的指令有限制。

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。



● 错误代码

指令中发生的异常请参考 □ 「A-1 错误代码一览 (P.A-2)」。

示例程序

下面介绍通过反复进行多重启动来直线插补时的示例程序。

参数设定

下面表示本示例程序中使用的最低限度的设定。

● 设定轴参数

轴种类

轴	轴种类
轴 1	伺服轴
轴 2	伺服轴

计数器模式

轴	计数器模式
轴 1	直线模式
轴 2	直线模式

坐标单位选择

轴	坐标单位
轴 1	mm
轴 2	mm

● 设定轴组参数

轴构成

为 [2 轴]。

轴选择

为 [轴 1]、[轴 2]。

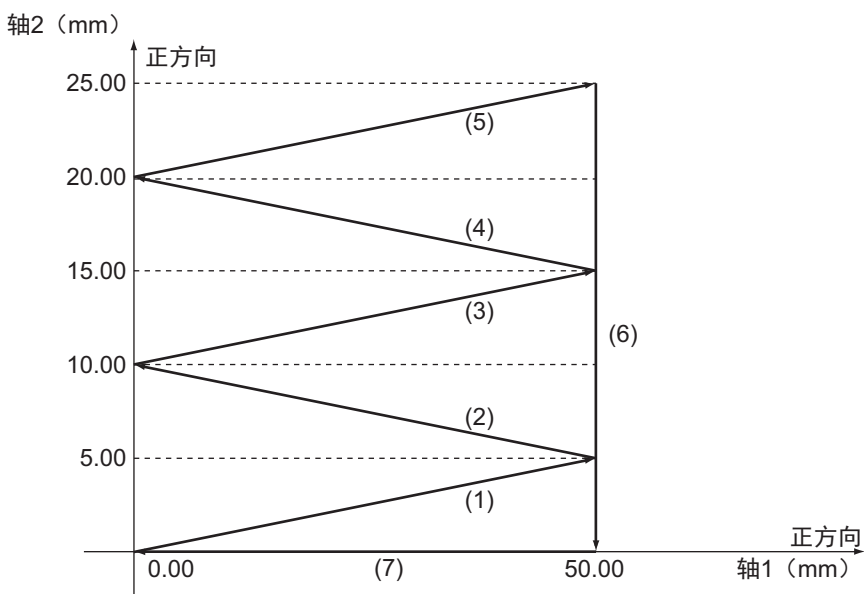
动作示例

自动执行直线插补动作，返回到原点 after 停止，以上动作示例如下。

直线插补 (1) 正在动作时，多重启动直线插补 (2) ~ (7)。此时的缓冲模式选择为缓冲。

本示例中，直线插补 (1) 的输出变量“Active（控制中）”变为 TRUE 时，执行 (2) ~ (7) 的指令多重启动。在多轴协调动作中，最多可多重启动 7 个指令。

● 动作模式



使用直线插补，按（轴 1、轴 2）= (50.00mm, 5.00mm) → (0.00mm, 10.00mm) → (50.00mm, 15.00mm) → (0.00mm, 20.00mm) → (50.00mm, 25.00mm) → (50.00mm, 0.00mm) → (0.00mm, 0.00mm) 的顺序进行定位，然后停止。

梯形图

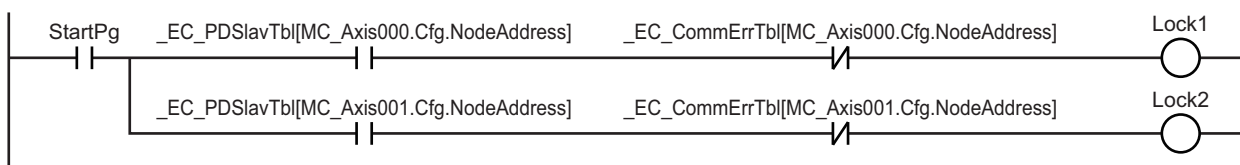
● 主要变量

名称	数据类型	初始值	注释
MC_Group000	_sGROUP_REF	—	轴组 0 的轴组变量。
MC_Group000.MFaultLvl.Active	BOOL	FALSE	轴组 0 发生轻度故障等级的异常时，变为 TRUE。
MC_Group000.Status.Disabled	BOOL	FALSE	轴组 0 为无效时变为 TRUE。
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis001	_sAXIS_REF	—	轴 2 的轴变量。
MC_Axis001.Details.Homed	BOOL	FALSE	轴 2 处于原点确定状态时变为 TRUE。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴 2 发生轻度故障等级的异常时，变为 TRUE。

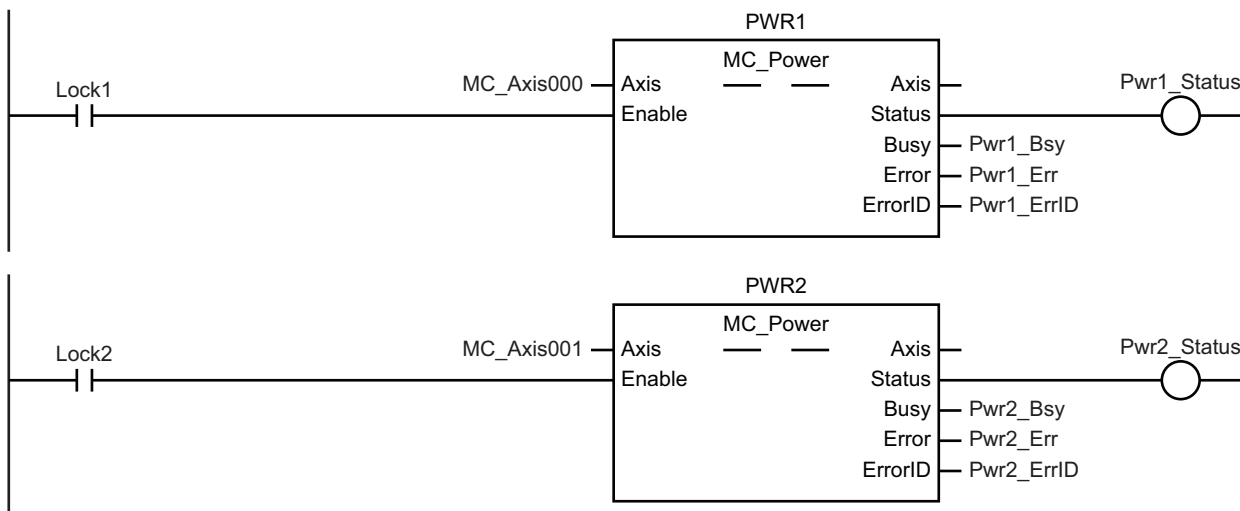
名称	数据类型	初始值	注释
Pwr1_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR1 的输出 Status 中的变量。若变为伺服 ON 状态，则变为 TRUE。
Pwr2_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR2 的输出 Status 中的变量。若变为伺服 ON 状态，则变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态，则轴组中所属的各轴变为伺服 ON 状态。
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE 时，设定输入参数。输入参数设定完成后，变为 TRUE。

● 示例程序

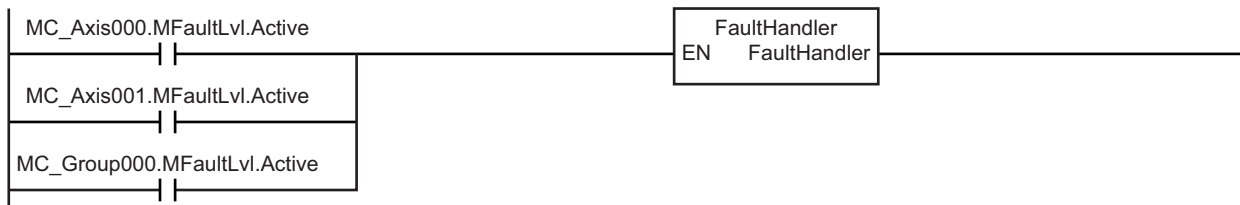
接点 StartPg 变为 TRUE 后，确认各轴的 EtherCAT 通信是否正常进行过程数据通信



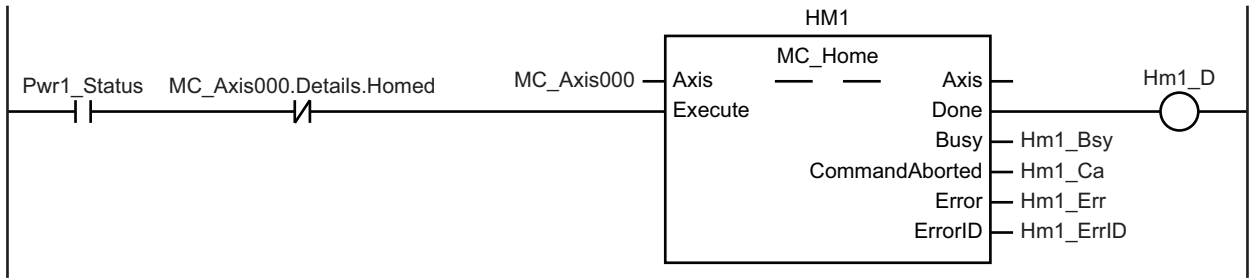
若正在进行过程数据通信，则将各轴设为伺服 ON 状态



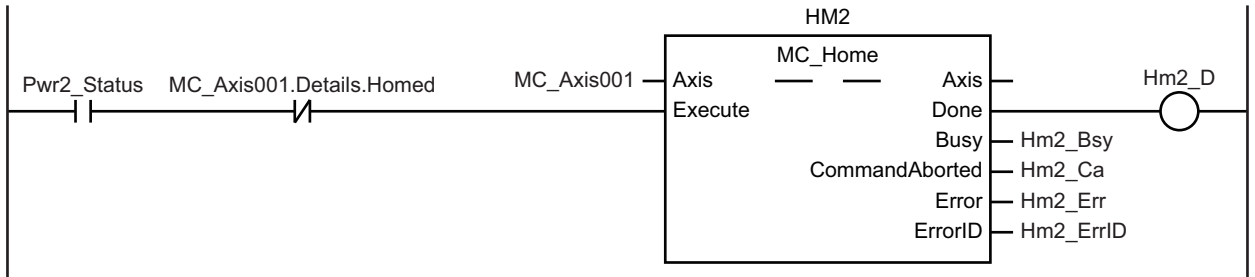
轴构成发生轻度故障等级的异常时，执行装置发生异常时的处理（FaultHandler）。
发生异常时的处理（FaultHandler）请根据装置由客户进行编程



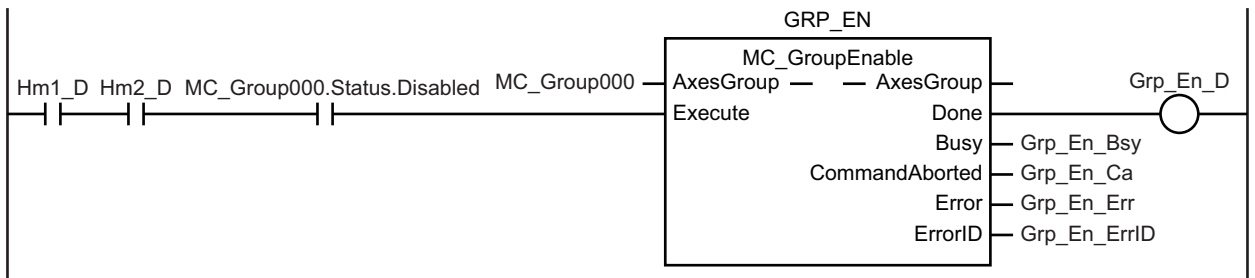
轴 1 为伺服 ON 状态且处于原点未确定状态时，执行原点复位，确定原点



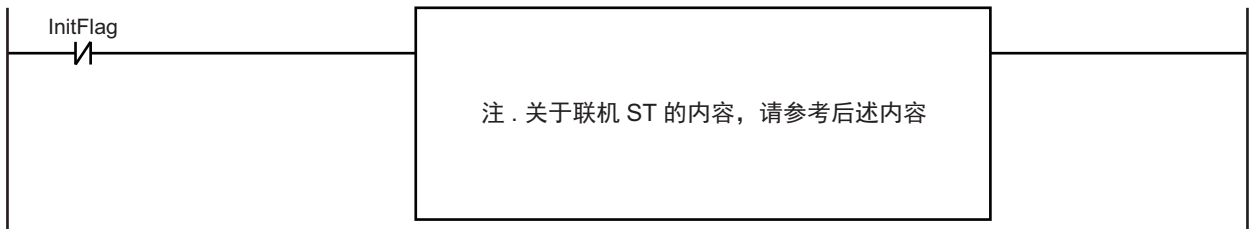
轴 2 为伺服 ON 状态且处于原点未确定状态时，执行原点复位，确定原点



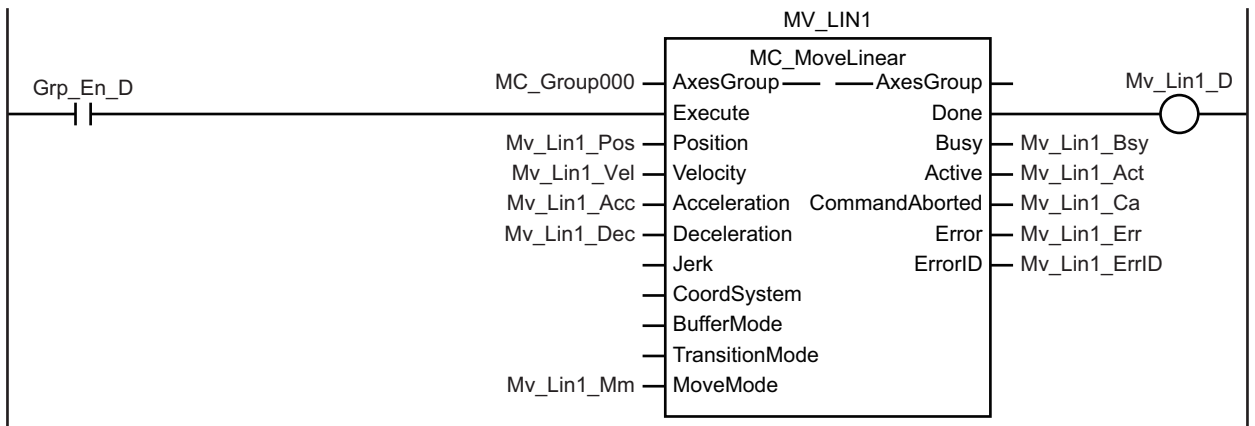
确定轴 1 和轴 2 的原点后，将轴组设为有效



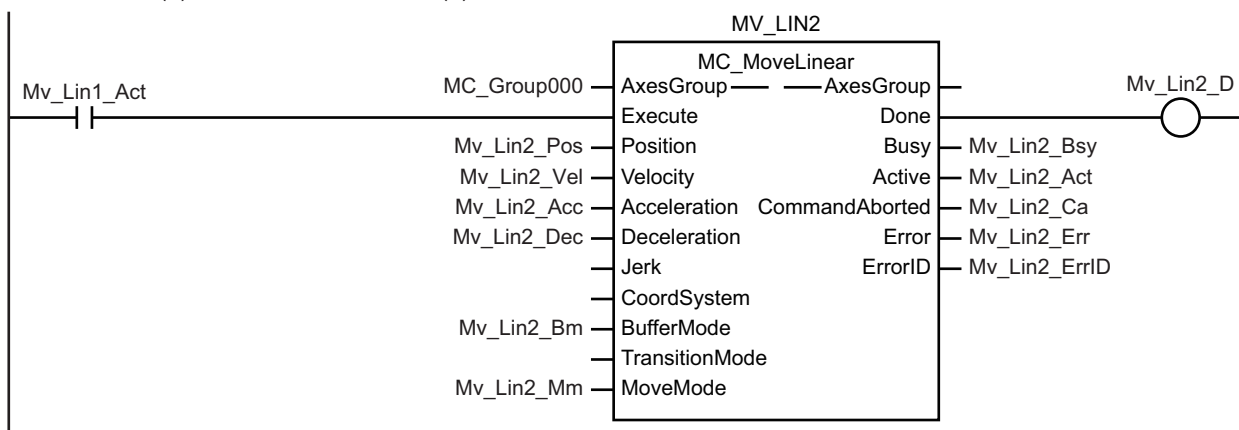
设定直线插补的参数



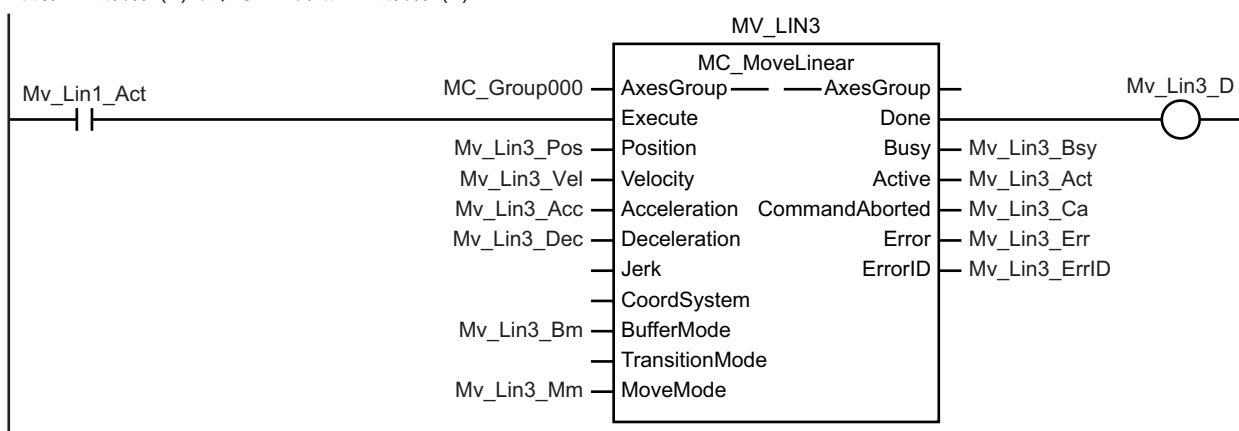
确认轴组变为有效，执行直线插补 (1)



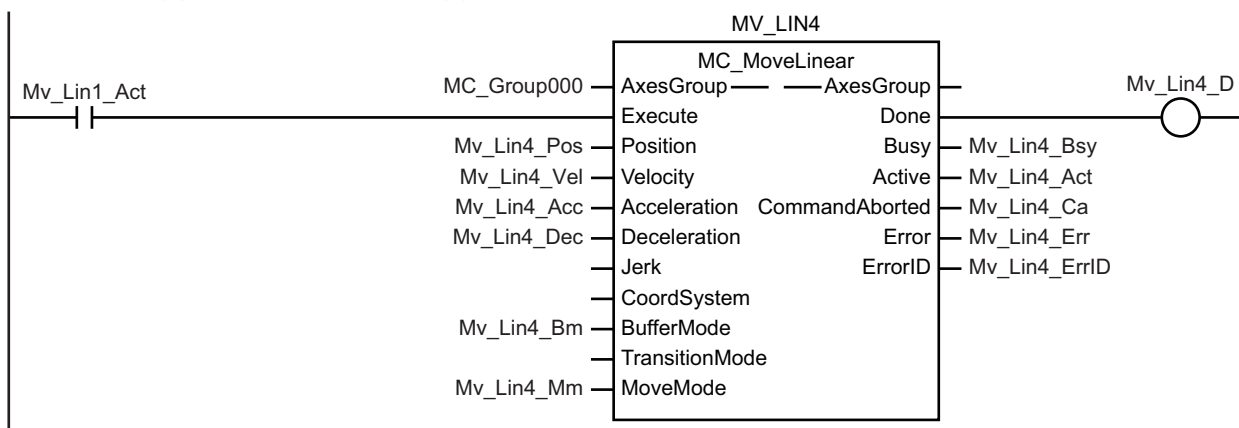
开始直线插补 (1) 后，多重启动直线插补 (2)



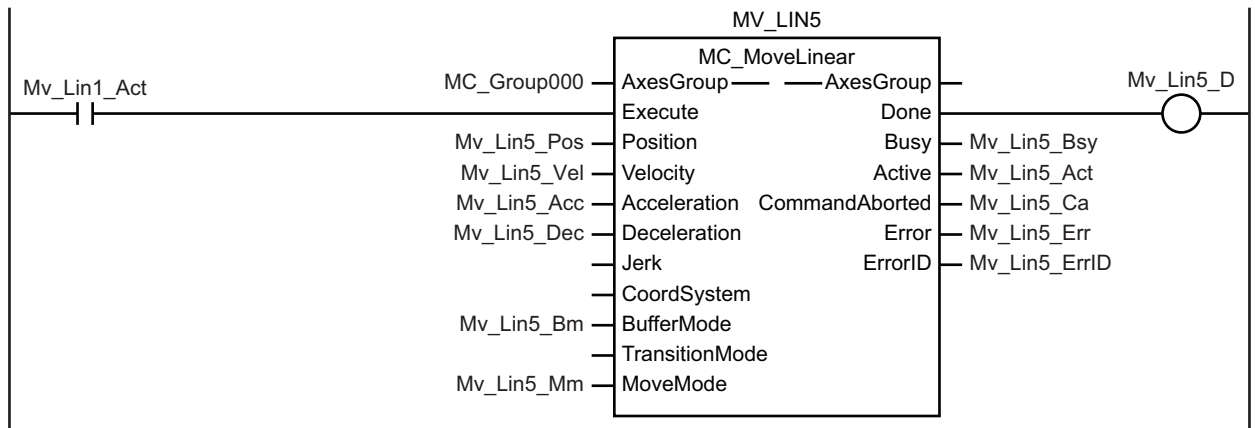
开始直线插补 (1) 后，多重启动直线插补 (3)



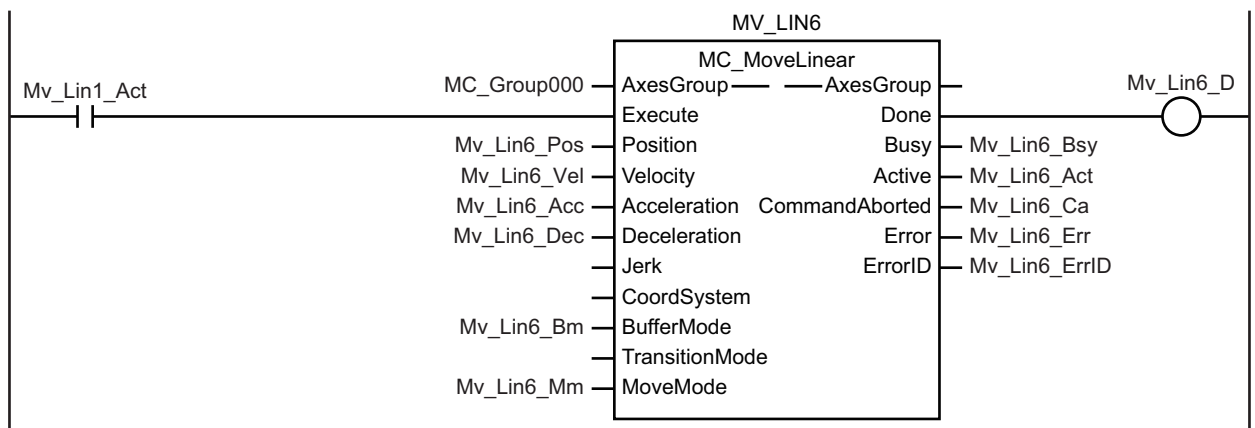
开始直线插补 (1) 后，多重启动直线插补 (4)



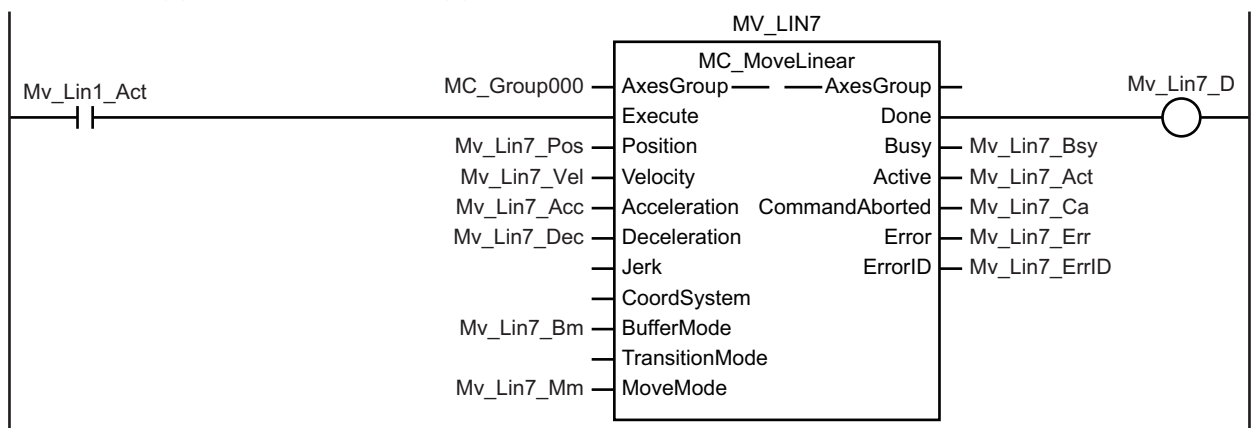
开始直线插补 (1) 后, 多重启动直线插补 (5)



开始直线插补 (1) 后, 多重启动直线插补 (6)



开始直线插补 (1) 后, 多重启动直线插补 (7)



联机 ST 的内容

```
//MV_LIN1 参数
Mv_Lin1_Pos[0] := LREAL#50.0;
Mv_Lin1_Pos[1] := LREAL#5.0;
Mv_Lin1_Vel   := LREAL#100.0;
Mv_Lin1_Acc   := LREAL#100.0;
Mv_Lin1_Dec   := LREAL#100.0;
Mv_Lin1_Mm    := _eMC_MOVE_MODE#_mcAbsolute;
```

```
//MV_LIN2 参数
Mv_Lin2_Pos[0] := LREAL#0.0;
Mv_Lin2_Pos[1] := LREAL#10.0;
Mv_Lin2_Vel   := LREAL#100.0;
Mv_Lin2_Acc   := LREAL#100.0;
```

```

Mv_Lin2_Dec      := LREAL#100.0;
Mv_Lin2_Bm      := _eMC_BUFFER_MODE#_mcBuffered;
Mv_Lin2_Mm      := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN3 参数
Mv_Lin3_Pos[0]  := LREAL#50.0;
Mv_Lin3_Pos[1]  := LREAL#15.0;
Mv_Lin3_Vel     := LREAL#100.0;
Mv_Lin3_Acc     := LREAL#100.0;
Mv_Lin3_Dec     := LREAL#100.0;
Mv_Lin3_Bm      := _eMC_BUFFER_MODE#_mcBuffered;
Mv_Lin3_Mm      := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN4 参数
Mv_Lin4_Pos[0]  := LREAL#0.0;
Mv_Lin4_Pos[1]  := LREAL#20.0;
Mv_Lin4_Vel     := LREAL#100.0;
Mv_Lin4_Acc     := LREAL#100.0;
Mv_Lin4_Dec     := LREAL#100.0;
Mv_Lin4_Bm      := _eMC_BUFFER_MODE#_mcBuffered;
Mv_Lin4_Mm      := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN5 参数
Mv_Lin5_Pos[0]  := LREAL#50.0;
Mv_Lin5_Pos[1]  := LREAL#25.0;
Mv_Lin5_Vel     := LREAL#100.0;
Mv_Lin5_Acc     := LREAL#100.0;
Mv_Lin5_Dec     := LREAL#100.0;
Mv_Lin5_Bm      := _eMC_BUFFER_MODE#_mcBuffered;
Mv_Lin5_Mm      := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN6 参数
Mv_Lin6_Pos[0]  := LREAL#50.0;
Mv_Lin6_Pos[1]  := LREAL#0.0;
Mv_Lin6_Vel     := LREAL#100.0;
Mv_Lin6_Acc     := LREAL#100.0;
Mv_Lin6_Dec     := LREAL#100.0;
Mv_Lin6_Bm      := _eMC_BUFFER_MODE#_mcBuffered;
Mv_Lin6_Mm      := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN7 参数
Mv_Lin7_Pos[0]  := LREAL#0.0;
Mv_Lin7_Pos[1]  := LREAL#0.0;
Mv_Lin7_Vel     := LREAL#100.0;
Mv_Lin7_Acc     := LREAL#100.0;
Mv_Lin7_Dec     := LREAL#100.0;
Mv_Lin7_Bm      := _eMC_BUFFER_MODE#_mcBuffered;
Mv_Lin7_Mm      := _eMC_MOVE_MODE#_mcAbsolute;

// 设定输入参数后, 将 InitFlag 设为 TRUE
InitFlag := TRUE;

```

结构文本 (ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Group000	_sGROUP_REF	—	轴组 0 的轴组变量。
MC_Group000.MFaultLvl.Active	BOOL	FALSE	轴组 0 发生轻度故障等级的异常时, 变为 TRUE。
MC_Group000.Status.Disabled	BOOL	FALSE	轴组 0 为无效时变为 TRUE。
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时, 变为 TRUE。
MC_Axis001	_sAXIS_REF	—	轴 2 的轴变量。
MC_Axis001.Details.Homed	BOOL	FALSE	轴 2 处于原点确定状态时变为 TRUE。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴 2 发生轻度故障等级的异常时, 变为 TRUE。
Pwr1_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR1 的输出 Status 中的变量。若变为伺服 ON 状态, 则变为 TRUE。
Pwr2_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR2 的输出 Status 中的变量。若变为伺服 ON 状态, 则变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态, 则轴组中所属的各轴变为伺服 ON 状态。
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE 时, 设定输入参数。输入参数设定完成后, 变为 TRUE。

● 示例程序

```
// 未设定输入参数时的处理
IF InitFlag=FALSE THEN

    //MV_LIN1 参数
    Mv_Lin1_Pos[0] := LREAL#50.0;
    Mv_Lin1_Pos[1] := LREAL#5.0;
    Mv_Lin1_Vel   := LREAL#100.0;
    Mv_Lin1_Acc   := LREAL#100.0;
    Mv_Lin1_Dec   := LREAL#100.0;
    Mv_Lin1_Mm    := _eMC_MOVE_MODE#_mcAbsolute;

    //MV_LIN2 参数
    Mv_Lin2_Pos[0] := LREAL#0.0;
    Mv_Lin2_Pos[1] := LREAL#10.0;
    Mv_Lin2_Vel   := LREAL#100.0;
    Mv_Lin2_Acc   := LREAL#100.0;
    Mv_Lin2_Dec   := LREAL#100.0;
    Mv_Lin2_Bm    := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Lin2_Mm    := _eMC_MOVE_MODE#_mcAbsolute;

    //MV_LIN3 参数
    Mv_Lin3_Pos[0] := LREAL#50.0;
    Mv_Lin3_Pos[1] := LREAL#15.0;
```

```

Mv_Lin3_Vel      := LREAL#100.0;
Mv_Lin3_Acc      := LREAL#100.0;
Mv_Lin3_Dec      := LREAL#100.0;
Mv_Lin3_Bm       := _eMC_BUFFER_MODE#_mcBuffered;
Mv_Lin3_Mm       := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN4 参数
Mv_Lin4_Pos[0]   := LREAL#0.0;
Mv_Lin4_Pos[1]   := LREAL#20.0;
Mv_Lin4_Vel      := LREAL#100.0;
Mv_Lin4_Acc      := LREAL#100.0;
Mv_Lin4_Dec      := LREAL#100.0;
Mv_Lin4_Bm       := _eMC_BUFFER_MODE#_mcBuffered;
Mv_Lin4_Mm       := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN5 参数
Mv_Lin5_Pos[0]   := LREAL#50.0;
Mv_Lin5_Pos[1]   := LREAL#25.0;
Mv_Lin5_Vel      := LREAL#100.0;
Mv_Lin5_Acc      := LREAL#100.0;
Mv_Lin5_Dec      := LREAL#100.0;
Mv_Lin5_Bm       := _eMC_BUFFER_MODE#_mcBuffered;
Mv_Lin5_Mm       := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN6 参数
Mv_Lin6_Pos[0]   := LREAL#50.0;
Mv_Lin6_Pos[1]   := LREAL#0.0;
Mv_Lin6_Vel      := LREAL#100.0;
Mv_Lin6_Acc      := LREAL#100.0;
Mv_Lin6_Dec      := LREAL#100.0;
Mv_Lin6_Bm       := _eMC_BUFFER_MODE#_mcBuffered;
Mv_Lin6_Mm       := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN7 参数
Mv_Lin7_Pos[0]   := LREAL#0.0;
Mv_Lin7_Pos[1]   := LREAL#0.0;
Mv_Lin7_Vel      := LREAL#100.0;
Mv_Lin7_Acc      := LREAL#100.0;
Mv_Lin7_Dec      := LREAL#100.0;
Mv_Lin7_Bm       := _eMC_BUFFER_MODE#_mcBuffered;
Mv_Lin7_Mm       := _eMC_MOVE_MODE#_mcAbsolute;

// 设定输入参数后, 将 InitFlag 设为 TRUE
InitFlag := TRUE;

END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 1 设为伺服 ON 状态。
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
  Pwr1_En:=TRUE;      // 将轴 1 设为伺服 ON
ELSE
  Pwr1_En:=FALSE;    // 将轴 1 设为伺服 OFF
END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 2 设为伺服 ON 状态。
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
  AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)

```

```

AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
  Pwr2_En:=TRUE;      // 将轴 2 设为伺服 ON
ELSE
  Pwr2_En:=FALSE;    // 将轴 2 设为伺服 OFF
END_IF;

// 发生轻度故障等级异常时的处理
// 发生异常时的处理 (FaultHandler) 内容请根据装置由客户进行编程。
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) OR
(MC_Group000.MFaultLvl.Active=TRUE)THEN
  FaultHandler();
END_IF;

// 轴 1 为伺服 ON 状态且处于原点未确定状态时, 执行原点确定
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
  Hm1_Ex:=TRUE;
END_IF;

// 轴 2 为伺服 ON 状态且处于原点未确定状态时, 执行原点确定
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
  Hm2_Ex:=TRUE;
END_IF;

// 若轴 1 和轴 2 为原点确定状态且轴组 0 为无效, 则将轴组 0 设为有效
IF (Hm1_D=TRUE) AND (Hm2_D=TRUE) AND (MC_Group000.Status.Disabled=TRUE) THEN
  Grp_En_Ex:= TRUE;
END_IF;

// 轴组有效 MC_GroupEnable 完成后, 执行直线插补 (1)
IF Grp_En_D=TRUE THEN
  Mv_Lin1_Ex:=TRUE;
END_IF;

// 直线插补 (1) 的输出 Active 变为 TRUE 后, 多重启动直线插补 (2) ~ (7)
IF Mv_Lin1_Act=TRUE THEN
  Mv_Lin2_Ex:=TRUE;
  Mv_Lin3_Ex:=TRUE;
  Mv_Lin4_Ex:=TRUE;
  Mv_Lin5_Ex:=TRUE;
  Mv_Lin6_Ex:=TRUE;
  Mv_Lin7_Ex:=TRUE;
END_IF;

// 轴 1 的 MC_Power
PWR1(
  Axis      := MC_Axis000,
  Enable    := Pwr1_En,
  Status    => Pwr1_Status,
  Busy      => Pwr1_Bsy,
  Error     => Pwr1_Err,
  ErrorID   => Pwr1_ErrID
);

// 轴 2 的 MC_Power
PWR2(
  Axis      := MC_Axis001,
  Enable    := Pwr2_En,
  Status    => Pwr2_Status,
  Busy      => Pwr2_Bsy,
  Error     => Pwr2_Err,
  ErrorID   => Pwr2_ErrID
);

```

```

// 轴 1 的 MC_Home
HM1(
  Axis           := MC_Axis000,
  Execute        := Hm1_Ex,
  Done           => Hm1_D,
  Busy           => Hm1_Bsy,
  CommandAborted => Hm1_Ca,
  Error          => Hm1_Err,
  ErrorID        => Hm1_ErrID
);

// 轴 2 的 MC_Home
HM2(
  Axis           := MC_Axis001,
  Execute        := Hm2_Ex,
  Done           => Hm2_D,
  Busy           => Hm2_Bsy,
  CommandAborted => Hm2_Ca,
  Error          => Hm2_Err,
  ErrorID        => Hm2_ErrID
);

// 将轴组 0 设为有效
GRP_EN(
  AxesGroup      := MC_Group000,
  Execute        := Grp_En_Ex,
  Done           => Grp_En_D,
  Busy           => Grp_En_Bsy,
  CommandAborted => Grp_En_Ca,
  Error          => Grp_En_Err,
  ErrorID        => Grp_En_ErrID
);

// 直线插补 (1)
MV_LIN1(
  AxesGroup      := MC_Group000,
  Execute        := Mv_Lin1_Ex,
  Position       := Mv_Lin1_Pos,
  Velocity       := Mv_Lin1_Vel,
  Acceleration   := Mv_Lin1_Acc,
  Deceleration   := Mv_Lin1_Dec,
  MoveMode       := Mv_Lin1_Mm,
  Done           => Mv_Lin1_D,
  Busy           => Mv_Lin1_Bsy,
  Active         => Mv_Lin1_Act,
  CommandAborted => Mv_Lin1_Ca,
  Error          => Mv_Lin1_Err,
  ErrorID        => Mv_Lin1_ErrID
);

// 直线插补 (2)
MV_LIN2(
  AxesGroup      := MC_Group000,
  Execute        := Mv_Lin2_Ex,
  Position       := Mv_Lin2_Pos,
  Velocity       := Mv_Lin2_Vel,
  Acceleration   := Mv_Lin2_Acc,
  Deceleration   := Mv_Lin2_Dec,
  BufferMode      := Mv_Lin2_Bm,
  MoveMode       := Mv_Lin2_Mm,
  Done           => Mv_Lin2_D,
  Busy           => Mv_Lin2_Bsy,
  Active         => Mv_Lin2_Act,
  CommandAborted => Mv_Lin2_Ca,

```

```

        Error          => Mv_Lin2_Err,
        ErrorID        => Mv_Lin2_ErrID
    );

// 直线插补 (3)
MV_LIN3(
    AxesGroup          := MC_Group000,
    Execute            := Mv_Lin3_Ex,
    Position           := Mv_Lin3_Pos,
    Velocity           := Mv_Lin3_Vel,
    Acceleration       := Mv_Lin3_Acc,
    Deceleration       := Mv_Lin3_Dec,
    BufferMode          := Mv_Lin3_Bm,
    MoveMode           := Mv_Lin3_Mm,
    Done               => Mv_Lin3_D,
    Busy               => Mv_Lin3_Bsy,
    Active             => Mv_Lin3_Act,
    CommandAborted    => Mv_Lin3_Ca,
    Error              => Mv_Lin3_Err,
    ErrorID            => Mv_Lin3_ErrID
);

// 直线插补 (4)
MV_LIN4(
    AxesGroup          := MC_Group000,
    Execute            := Mv_Lin4_Ex,
    Position           := Mv_Lin4_Pos,
    Velocity           := Mv_Lin4_Vel,
    Acceleration       := Mv_Lin4_Acc,
    Deceleration       := Mv_Lin4_Dec,
    BufferMode          := Mv_Lin4_Bm,
    MoveMode           := Mv_Lin4_Mm,
    Done               => Mv_Lin4_D,
    Busy               => Mv_Lin4_Bsy,
    Active             => Mv_Lin4_Act,
    CommandAborted    => Mv_Lin4_Ca,
    Error              => Mv_Lin4_Err,
    ErrorID            => Mv_Lin4_ErrID
);

// 直线插补 (5)
MV_LIN5(
    AxesGroup          := MC_Group000,
    Execute            := Mv_Lin5_Ex,
    Position           := Mv_Lin5_Pos,
    Velocity           := Mv_Lin5_Vel,
    Acceleration       := Mv_Lin5_Acc,
    Deceleration       := Mv_Lin5_Dec,
    BufferMode          := Mv_Lin5_Bm,
    MoveMode           := Mv_Lin5_Mm,
    Done               => Mv_Lin5_D,
    Busy               => Mv_Lin5_Bsy,
    Active             => Mv_Lin5_Act,
    CommandAborted    => Mv_Lin5_Ca,
    Error              => Mv_Lin5_Err,
    ErrorID            => Mv_Lin5_ErrID
);

// 直线插补 (6)
MV_LIN6(
    AxesGroup          := MC_Group000,
    Execute            := Mv_Lin6_Ex,
    Position           := Mv_Lin6_Pos,
    Velocity           := Mv_Lin6_Vel,
    Acceleration       := Mv_Lin6_Acc,

```

```
Deceleration      := Mv_Lin6_Dec,
BufferMode        := Mv_Lin6_Bm,
MoveMode          := Mv_Lin6_Mm,
Done              => Mv_Lin6_D,
Busy              => Mv_Lin6_Bsy,
Active            => Mv_Lin6_Act,
CommandAborted   => Mv_Lin6_Ca,
Error             => Mv_Lin6_Err,
ErrorID          => Mv_Lin6_ErrID
);
// 直线插补 (7)
MV_LIN7(
  AxesGroup        := MC_Group000,
  Execute          := Mv_Lin7_Ex,
  Position         := Mv_Lin7_Pos,
  Velocity         := Mv_Lin7_Vel,
  Acceleration     := Mv_Lin7_Acc,
  Deceleration     := Mv_Lin7_Dec,
  BufferMode        := Mv_Lin7_Bm,
  MoveMode         := Mv_Lin7_Mm,
  Done             => Mv_Lin7_D,
  Busy             => Mv_Lin7_Bsy,
  Active           => Mv_Lin7_Act,
  CommandAborted   => Mv_Lin7_Ca,
  Error            => Mv_Lin7_Err,
  ErrorID         => Mv_Lin7_ErrID
);
```


MC_MoveLinearAbsolute

指定绝对位置进行直线插补。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_MoveLinearAbsolute	绝对值 直线插补	FB		<pre> MC_MoveLinearAbsolute_instance (AxesGroup := 《参数》, Execute := 《参数》, Position := 《参数》, Velocity := 《参数》, Acceleration := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, CoordSystem := 《参数》, BufferMode := 《参数》, TransitionMode := 《参数》, Done => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》); </pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Position	目标位置	ARRAY [0..3] OF LREAL	负数、正数、“0”	0	指定直线插补的目标位置。 单位为 [指令单位]。 *1
Velocity *2	目标速度	LREAL	正数	0	指定目标速度。 单位为 [指令单位 /s]。 *1
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度。 单位为 [指令单位 /s ²]。 *1
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。 单位为 [指令单位 /s ²]。 *1
Jerk	跃度	LREAL	正数或“0”	0	指定跃度 单位为 [指令单位 /s ³]。 *1
Coord System	坐标系	_eMC_ COORD_ SYSTEM	0: _mcACS	0 *3	指定坐标系。 0: 轴坐标系 (ACS)

输入变量	名称	数据类型	有效范围	初始值	内容
Buffer Mode	缓冲模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 ^{*3}	指定运动指令多重启动时的动作。 0: 中断 1: 缓冲 2: 低共混 3: 前共混 4: 后共混 5: 高共混
Transition Mode	转换模式(变化模式)	_eMC_TRANSITION_MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 ^{*3}	指定动作的路径。 0: 转换无效 10: 重叠

*1. 指令单位请参考 □□ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. 请务必设定目标速度。若不设定而直接动作，将发生异常。

*3. 如果是有效范围为列举型的变量初始值，实际不是数值，而是列举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* □□ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	定位完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	轴已开始移动时	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动 (中断), 本指令中止时 因发生异常, 本指令中止时 正在发生异常时, 启动了本指令 正在执行 MC_GroupStop 指令时, 启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
AxesGroup	轴组	_sGROUP_REF	—	指定轴组。 ^{*1}

*1. 请使用 Sysmac Studio 的轴组基本设定画面中创建的用户定义变量的轴组变量名称 (默认“MC_Group***”)或系统定义变量的轴组变量名称 (_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])。

功能说明

- 进行 2 轴~ 4 轴的直线插补。
- 目标位置以绝对位置指定。

其他与 MC_MoveLinear (直线插补) 指令的规格相同。

详情请参考 □ 「功能说明 (P.4-14)」。



使用注意事项

- 轴组构成轴中有任一轴为原点未确定时, 将发生“原点未确定状态下指令启动异常 (错误代码: 5466 Hex)”。
- 进行直线插补时, 属于该轴组的逻辑轴中, 任意一个的极限输入为“ON”时, 指令将无法启动。

MC_MoveLinearRelative

指定相对位置进行直线插补。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_MoveLinearRelative	相对值 直线插补	FB		<pre>MC_MoveLinearRelative_instance (AxesGroup := 《参数》, Execute := 《参数》, Distance := 《参数》, Velocity := 《参数》, Acceleration := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, CoordSystem := 《参数》, BufferMode := 《参数》, TransitionMode := 《参数》, Done => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Distance	移动距离	ARRAY [0..3] OF LREAL	负数、正数、“0”	0	指定直线插补的目标位置。 单位为 [指令单位]。 *1
Velocity *2	目标速度	LREAL	正数	0	指定目标速度。 单位为 [指令单位 /s]。 *1
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度。 单位为 [指令单位 /s ²]。 *1
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。 单位为 [指令单位 /s ²]。 *1
Jerk	跃度	LREAL	正数或“0”	0	指定跃度 单位为 [指令单位 /s ³]。 *1
CoordSystem	坐标系	_eMC_ COORD_ SYSTEM	0: _mcACS	0 *3	指定坐标系。 0: 轴坐标系 (ACS)

输入变量	名称	数据类型	有效范围	初始值	内容
BufferMode	缓冲模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	指定运动指令多重启动时的动作。 0: 中断 1: 缓冲 2: 低共混 3: 前共混 4: 后共混 5: 高共混
Transition Mode	转换模式 (变化模式)	_eMC_TRANSITION_MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 *3	指定动作的路径。 0: 转换无效 10: 重叠

*1. 指令单位请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. 请务必设定目标速度。若不设定而直接动作，将发生异常。

*3. 如果是有效范围为列举型的变量初始值，实际不是数值，而是列举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* □□ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	定位完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	轴已开始移动时	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动（中断），本指令中止时 因发生异常，本指令中止时 正在发生异常时，启动了本指令 正在执行 MC_GroupStop 指令时，启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
AxesGroup	轴组	_sGROUP_REF	—	指定轴组。 ^{*1}

*1. 请使用 Sysmac Studio 的轴组基本设定画面中创建的用户定义变量的轴组变量名称（默认“MC_Group***”）或系统定义变量的轴组变量名称（_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*]）。

功能说明

- 进行 2 轴～4 轴的直线插补。
- 目标位置以相对位置指定。

其他与 MC_MoveLinear（直线插补）指令的规格相同。

详情请参考 □□「功能说明 (P.4-14)」。



使用注意事项

- 轴组构成轴中有任一轴为原点未确定时，将发生“原点未确定状态下指令启动异常（错误代码：5466 Hex）”。
- 进行直线插补时，属于该轴组的逻辑轴中，任意一个的极限输入为“ON”时，指令将无法启动。

MC_MoveCircular2D

进行 2 轴圆弧插补。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_MoveCircular2D	2 轴圆弧插补	FB		<pre> MC_MoveCircular2D_instance (AxesGroup := 《参数》, Execute := 《参数》, CircAxes := 《参数》, CircMode := 《参数》, AuxPoint := 《参数》, EndPoint := 《参数》, PathChoice := 《参数》, Velocity := 《参数》, Acceleration := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, CoordSystem := 《参数》, BufferMode := 《参数》, TransitionMode := 《参数》, MoveMode := 《参数》, Done => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》); </pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
CircAxes	圆弧轴指定	ARRAY [0,1] OF UINT	0 ~ 3	0	指定进行圆弧插补的轴。 0: 轴 A0 1: 轴 A1 2: 轴 A2 3: 轴 A3
CircMode	圆弧插补模式	_eMC_CIRC_MODE	0: _mcBorder 1: _mcCenter 2: _mcRadius	0 *1	指定圆弧插补的方法。 0: 通过点指定 1: 中心点指定 2: 半径指定
AuxPoint	辅助点	ARRAY [0,1] OF LREAL	负数、正数、“0”	0	指定通过点位置 / 中心位置 / 半径。 单位为 [指令单位]。*2
EndPoint	终点	ARRAY [0,1] OF LREAL	负数、正数、“0”	0	指定目标位置。 单位为 [指令单位]。*2

输入变量	名称	数据类型	有效范围	初始值	内容
PathChoice	路径选择	_eMC_CIRC_PATHCHOICE	0: _mcCW 1: _mcCCW	0 *1	指定路径方向。 0: CW 1: CCW
Velocity *3	目标速度	LREAL	正数	0	指定目标速度。 单位为 [指令单位 /s]。*2
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度。 单位为 [指令单位 /s ²]。*2
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。 单位为 [指令单位 /s ²]。*2
Jerk	跃度	LREAL	正数或“0”	0	指定跃度 单位为 [指令单位 /s ³]。*2
Coord System	坐标系	_eMC_COORD_SYSTEM	0: _mcACS	0 *1	指定坐标系。 0: 轴坐标系 (ACS)
BufferMode	缓冲模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *1	指定运动指令多重启动时的动作。 0: 中断 1: 缓冲 2: 低共混 3: 前共混 4: 后共混 5: 高共混
Transition Mode	转换模式 (变化模式)	_eMC_TRANSITION_MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 *1	指定动作的路径。 0: 转换无效 10: 重叠
MoveMode	移动方法选择	_eMC_MOVE_MODE	0: _mcAbsolute 1: _mcRelative	0 *1	选择移动方法。 0: 绝对值定位 1: 相对值定位

*1. 如果是有效范围为列举型的变量初始值，实际不是数值，而是列举值。

*2. 指令单位请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*3. 请务必设定目标速度。若不设定而直接动作，将发生异常。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	定位完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	轴已开始移动时	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动（中断），本指令中止时 因发生异常，本指令中止时 正在发生异常时，启动了本指令 正在执行 MC_GroupStop 指令时，启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
AxesGroup	轴组	_sGROUP_REF	—	指定轴组。 ^{*1}

*1. 请使用 Sysmac Studio 的轴组基本设定画面中创建的用户定义变量的轴组变量名称（默认“MC_Group***”）或系统定义变量的轴组变量名称（_MC_GRP[*]、_MC1_GRP[*]、_MC2_GRP[*]）。

功能说明

- 在 2 维平面进行 2 轴圆弧插补。



使用注意事项

- 轴组构成轴中有任一轴为原点未确定时，将发生“原点未确定状态下指令启动异常（错误代码：5466 Hex）”。
- 进行 2 轴圆弧插补时，属于该轴组的逻辑轴中，任意一个的极限输入为“ON”时，指令将无法启动。

指令的详情

下面介绍本指令的详情。

● 圆弧插补的步骤

圆弧插补按以下步骤执行。

1 登录要插补的轴组

- 确定要插补的轴组。
轴组是指 `_MC_GRP[*]` 或 `_MC1_GRP[*]`、`_MC2_GRP[*]`。
- 在该轴组变量的轴构成中指定轴构成。
- 在该轴组变量的轴选择中，指定要进行插补的轴组合。
- 轴不使用轴编号（轴 0 ~ 轴 255），而使用逻辑轴（轴 A0 ~ 轴 A3）。
- 在轴选择中，从前往后分别对逻辑轴 A0 ~ A3 指定轴编号 0 ~ 255。

逻辑轴	轴编号
轴 A0	轴 0 ~ 轴 255
轴 A1	轴 0 ~ 轴 255
轴 A2	轴 0 ~ 轴 255
轴 A3	轴 0 ~ 轴 255

例：轴构成为 2 轴、轴编号 0 ~ 1 指定到轴 A0 ~ A1 时，指定如下。

逻辑轴	轴编号	说明
轴 A0	轴 0	从轴 A0 开始，按轴编号从小到大的顺序指定。
轴 A1	轴 1	



使用注意事项

轴组中包含编码器轴或虚拟编码器轴时，将发生“轴组种类不合适（错误代码：543D Hex）”，动作结束。
请务必选择伺服轴或虚拟伺服轴。

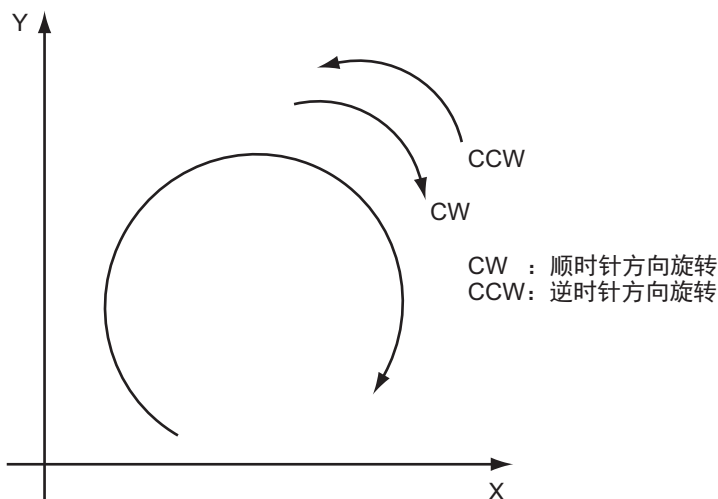
2 轴组的有效化

- 将轴组的各构成轴设为伺服 ON 状态，并使其变为原点确定状态。
- 执行 MC_GroupEnable（轴组有效）指令，将登录的轴组设为有效。

圆弧插补指令变为可以使用。

● CircAxes（圆弧轴指定）

使用圆弧插补的轴为 X 轴、Y 轴。



在 CircAxes（圆弧轴指定）中指定分配到 X 轴、Y 轴的轴。

指定轴时不使用轴编号（轴 0 ~ 轴 255），而以逻辑轴（轴 A0 ~ 轴 A3）指定。



使用注意事项

指定为 X 轴或 Y 轴的轴计数模式请选择“直线模式”。

以 [旋转模式] 启动时，将发生“计数模式设定引起的指令启动异常（错误代码：544A Hex）”。

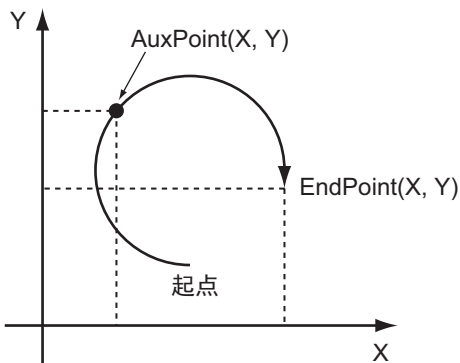
● CircMode（圆弧插补模式）

圆弧插补的方式包括“通过点指定 / 中心点指定 / 半径指定”3种，在 CircMode（圆弧插补模式）中指定。以上述方式指定的位置有“绝对值定位”和“相对值定位”，在 MoveMode（移动方法选择）中指定。

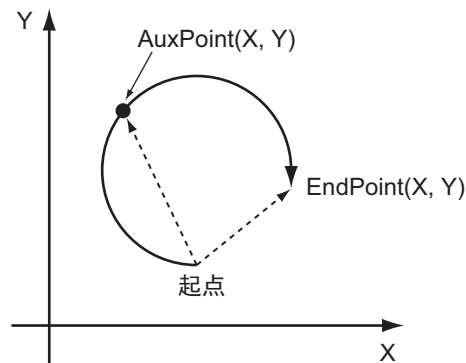
MoveMode	说明
绝对值定位	通过点指定中指定的通过点位置、中心点指定中指定的中心点位置和终点以距离轴坐标系原点的绝对位置指定。
相对值定位	通过点指定中指定的通过点位置、中心点指定中指定的中心点位置和终点以距离起点的相对位置指定。

以通过点指定为例，“绝对值定位”和“相对值定位”的差别如下所示。

圆弧方式：通过点指定<绝对值定位>



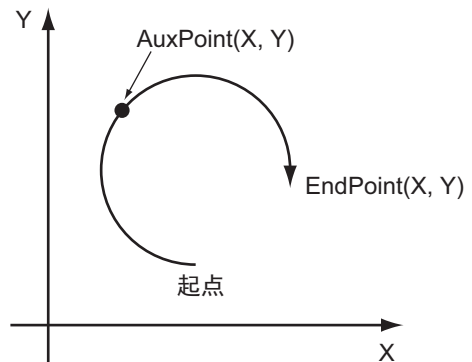
圆弧方式：通过点指定<相对值定位>



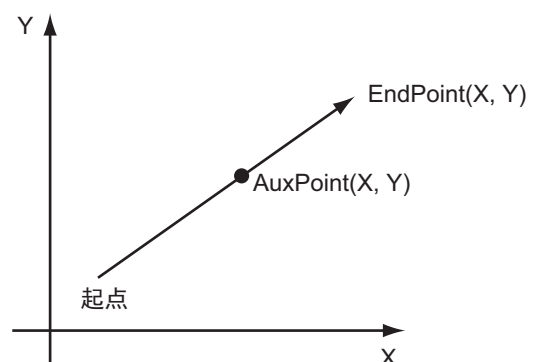
以下说明以 MoveMode（移动方式选择）指定为“绝对值定位”为前提。

通过点指定

将当前位置作为起点，进行连接通过点位置 AuxPoint(X, Y) 和终点 EndPoint(X, Y) 的圆弧插补。



起点、通过点位置和终点在同一直线上时、通过点位置和终点为同一点时或起点和通过点位置为同一点时，从起点到终点之间进行直线插补。

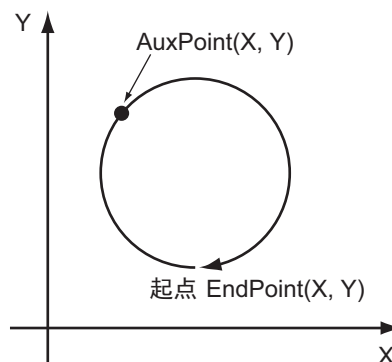




使用注意事项

- 在同一直线上是指: 连接起点和终点的直线与通过点位置的距离在 X 坐标、Y 坐标上均小于 1 个脉冲, 则判断为在同一直线上。
- 起点、通过点位置、终点为同一点时, 将发生异常。
起点、通过点位置、终点为同一点是指指令单位的指定位置相同。
指令单位的指令位置不同, 脉冲转换单位相同时, 不视为同一点, 不会发生异常。

起点和终点为同一点时, 将以连接起点和通过点的线为直径绘制一个圆。此时, 圆弧的旋转方向在 PathChoice (路径选择) 中指定。



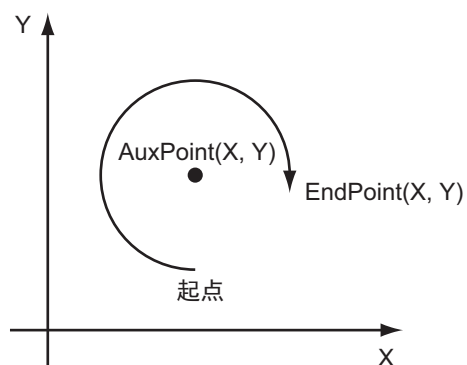
中心点指定

将当前位置作为起点, 用中心位置 AuxPoint(X, Y) 中指定的圆弧, 进行连接终点 EndPoint(X, Y) 的圆弧插补。

圆弧的旋转方向在 PathChoice (路径选择) 中指定。

起点和终点为同一点时, 将绘制一个圆。

指定中心位置到起点的半径和到终点的半径不同时, 使用 2 个半径的平均值进行圆弧插补。此时, 与半径指定时一样, 将计算出中心位置, 并使用该半径和中心位置。



使用注意事项

轴组参数的 [中心点修正容许率] 设定为 “0” 以外、指定的中心点位置超出以下公式的圆时, 将发生 “超出圆弧插补中心点指定位置范围 (错误代码: 5449 Hex)”。

将根据修正后的中心点计算得到的半径视为 “100%”,
超出 “半径 × 中心点指定方式检查方法 [%] / 100” 的圆时

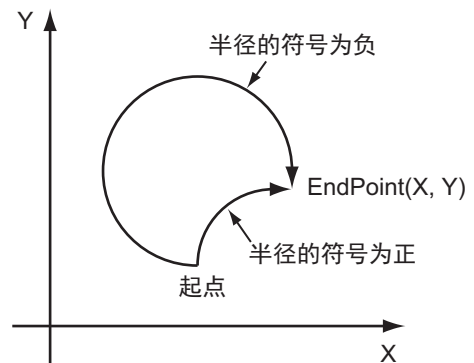
半径指定

将当前位置作为起点，用半径 $AuxPoint(X, Y)$ 中指定的圆弧，进行连接终点 $EndPoint(X, Y)$ 的圆弧插补。半径在 $AuxPoint(X, Y)$ 的第一个要素中指定。第 2 个要素不使用。

例如，半径设为 100 时， $AuxPoint(X, Y)$ 指定为 $AuxPoint(100, 0)$ 。

若半径的符号为负，则绘制圆弧长度较长的圆弧，若半径的符号为正，则绘制圆弧长度较短的圆弧。

圆弧的旋转方向在 $PathChoice$ （路径选择）中指定。



使用注意事项

- 起点和终点为同一点时，将发生圆弧插补起点终点相同异常，该组的所有轴停止动作。
- 指定的半径小于起点和终点距离的 1/2 时，将无法创建圆弧，因此会发生异常。

● Velocity（目标速度）、Acceleration（加速度）、Deceleration（减速度）、Jerk（跃度）

- 圆弧插补时的插补速度、加速度、减速度、跃度在 $Velocity$ 、 $Acceleration$ 、 $Deceleration$ 、 $Jerk$ 中指定。
- 若圆弧插补的插补速度指定为“0”，将发生 $Velocity$ 指定异常，该组的所有轴停止动作。
- 指定的插补速度超出任一轴的最高速度时，动作如下。

只超过 1 个轴时 : 自动调整插补速度，以该轴的最高速度动作。

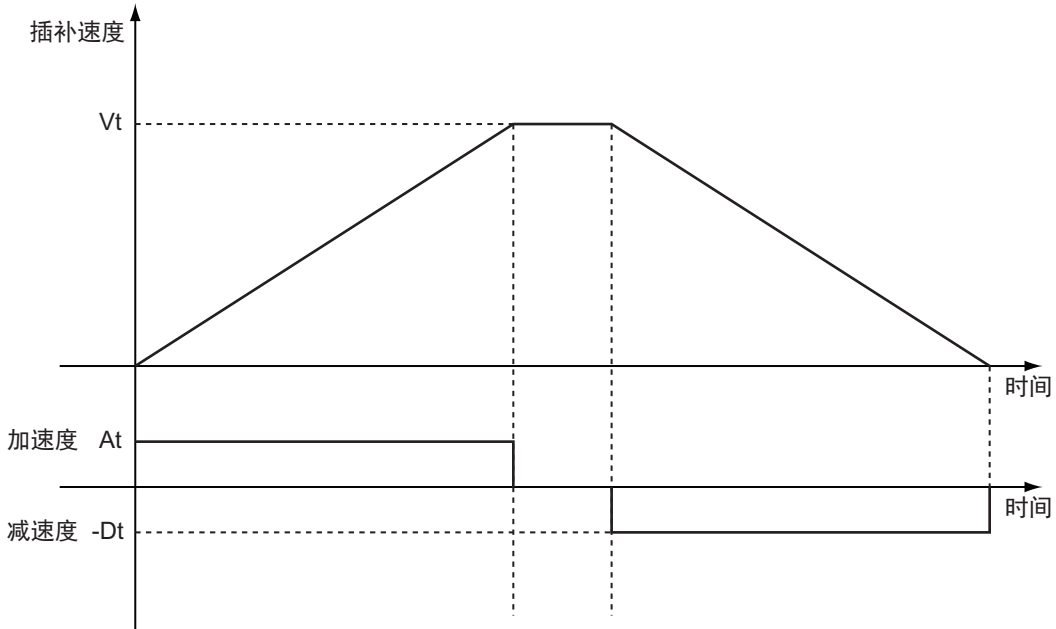
超出 2 个轴时 : 自动调整插补速度，以 2 个轴的最高速度中较慢的速度动作。

Jerk（跃度）

Jerk（跃度）指定为“0”时和指定为“0”以外时的 Acceleration（加速度）、Deceleration（减速度）、Velocity（目标速度）的关系如下所示。

- Jerk（跃度）为“0”时

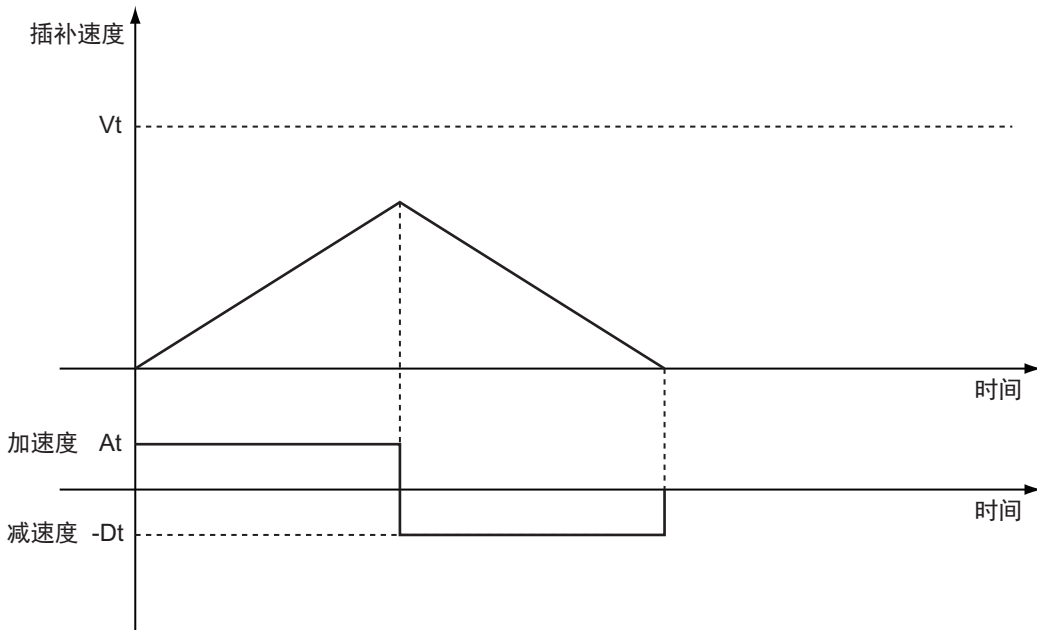
用加速度 A_t 、减速度 D_t 创建速度的指令值。



V_t : 插补速度的指定值、 A_t : 加速度的指定值、 D_t : 减速度的指定值

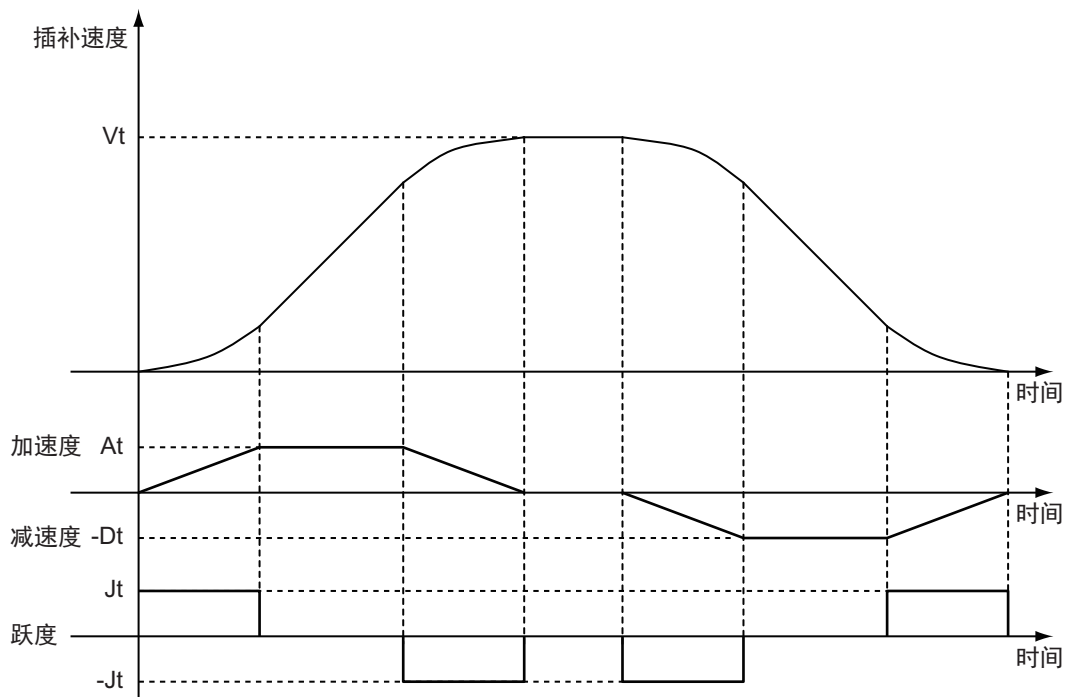
- Jerk（跃度）为“0”且移动距离较短时

插补速度不会达到指定值 V_t （目标速度）。



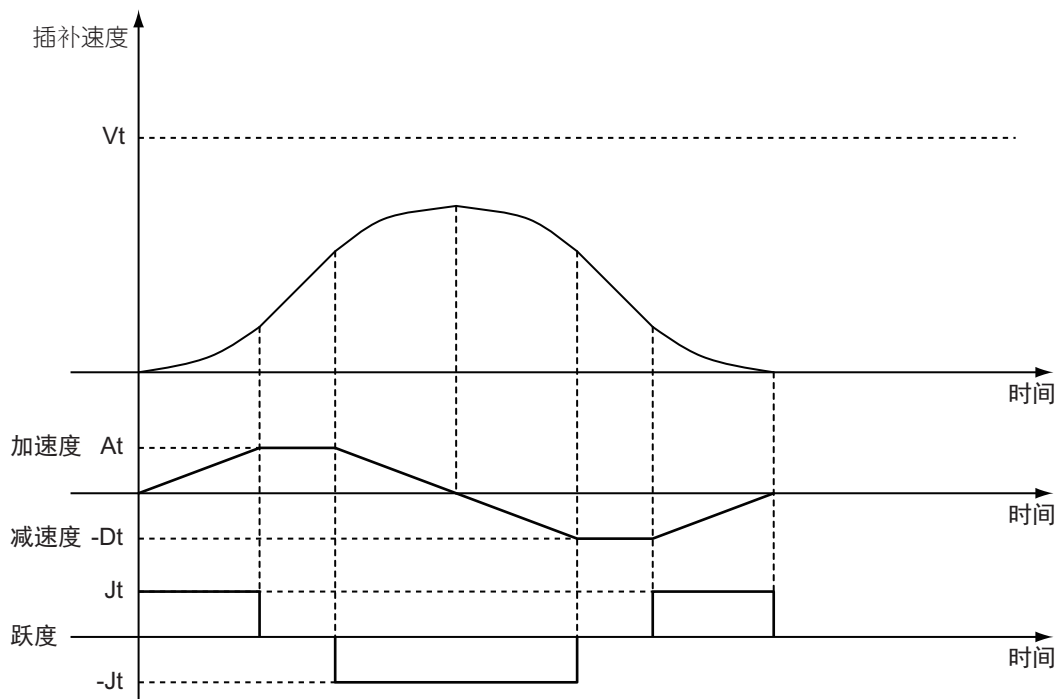
V_t : 插补速度的指定值、 A_t : 加速度的指定值、 D_t : 减速度的指定值

- Jerk（跃度）为“0”以外时
用加速度的上限 A_t 、减速度的上限 D_t 创建速度的指令值。



V_t : 插补速度的指定值、 A_t : 加速度的指定值、 D_t : 减速度的指定值、 J_t : 跃度的指令值

- Jerk（跃度）为“0”以外且移动距离较短时
插补速度不会达到指定值 V_t （目标速度）。



V_t : 插补速度的指定值、 A_t : 加速度的指定值、 D_t : 减速度的指定值、 J_t : 跃度的指令值



参考

- 若 Acceleration（加速度）指定为“0”，将立即按指定的插补速度动作。
- 若 Deceleration（减速度）指定为“0”，将立即停止。但是，在缓冲模式中指定了共混时，将不停止，在下一个动作时立即变为指定的插补速度。详情请参考 □「BufferMode（缓冲模式选择）(P.4-53)」。
- Acceleration（加速度）或 Deceleration（减速度）指定为“0”时，Jerk（跃度）的设置将变为无效。

● CoordSystem（坐标系）

- 指定进行圆弧插补的坐标系。
- 仅支持由 2 个以上的轴构成的轴坐标系（ACS）。

● BufferMode（缓冲模式选择）

- 指定前一插补动作和本次插补动作的连接方法。
- 有以下 6 种。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 切换指令后如果轴的动作方向发生反转，将按照轴参数 [反转时动作] 执行反转动作。
缓冲	当前正在执行的指令正常完成后，自动启动已缓冲的本指令。
共混	将当前正在执行的指令到达目标位置时的速度（中继速度）作为启动速度，连续执行已缓冲的本指令。变更当前正在执行的指令动作，使其以中继速度到达目标位置。中继速度的指定方法有以下 4 种。 此外，作为共混的选项，有转换指定（后述）。
低共混	将当前正在执行的目标速度和缓冲中的目标速度中较低的速度作为中继速度。
前共混	将当前正在执行的目标速度作为中继速度。
后共混	将已缓冲的本指令的目标速度作为中继速度。
高共混	将当前正在执行的目标速度和缓冲中的目标速度中较高的速度作为中继速度。

BufferMode（缓冲模式选择）请参考 □《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● TransitionMode（转换模式）

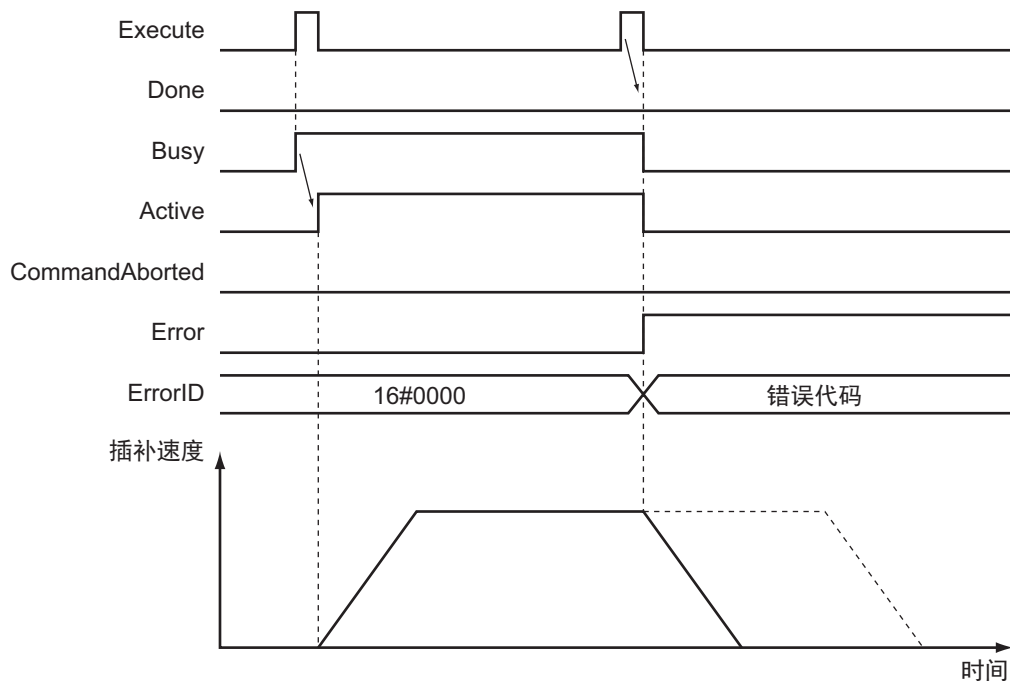
- 指定前一插补动作和本次插补动作中创建的轨迹的连接方法。
- BufferMode（缓冲模式选择）中指定了共混时，TransitionMode（转换模式）将变为有效。
- 不指定共混，且 TransitionMode（转换模式）指定为 _mcTMNone（转换无效）以外时，将发生异常。

详情请参考 □「TransitionMode（转换模式）(P.4-20)」。

运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”，圆弧插补动作中的所有轴停止。



运动指令多重启动

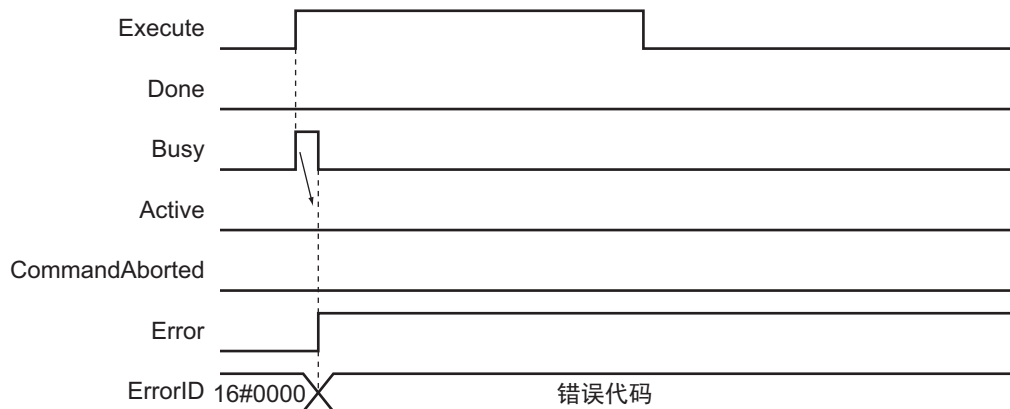
正在执行本指令时，可使用的指令有限制。

运动指令多重启动的详情请参考 □□《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。



● 错误代码

指令中发生的异常请参考 □□「A-1 错误代码一览 (P.A-2)」。

示例程序

下面介绍通过多重启动进行圆弧插补时的示例程序。

参数设定

下面表示本示例程序中使用的最低限度的设定。

- 设定轴参数

轴种类

轴	轴种类
轴 1	伺服轴
轴 2	伺服轴

计数器模式

轴	计数器模式
轴 1	直线模式
轴 2	直线模式

坐标单位选择

轴	坐标单位
轴 1	mm
轴 2	mm

- 设定轴组参数

轴构成

为 [2 轴]。

轴选择

为 [轴 1]、[轴 2]。

动作示例

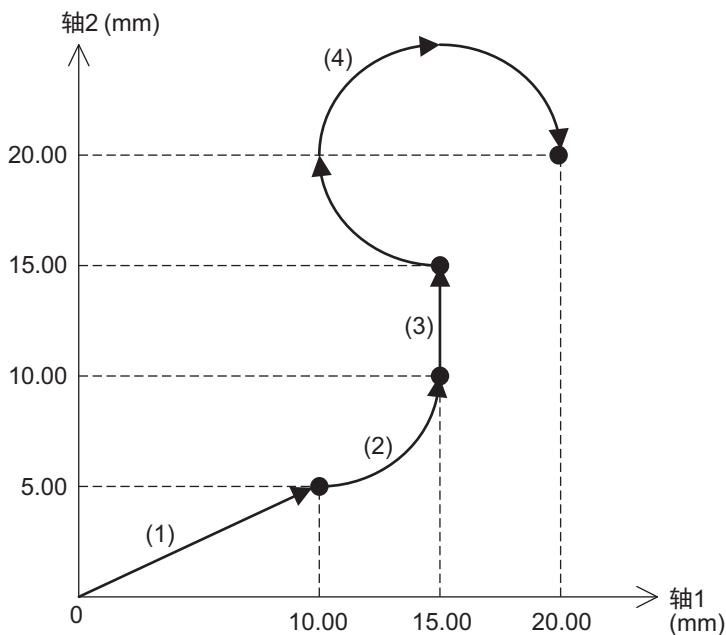
利用直线插补动作和圆弧插补，自动进行定位的动作示例如下。

使用直线插补和圆弧插补，移动到最终目标位置（20.00mm、20.00mm）。

缓冲模式选择指定为缓冲，通过多重启动发出指令。

本示例中，直线插补 (1) 的输出 Active 变为 TRUE 时，执行 (2) ~ (4) 的指令多重启动。在多轴协调动作中，最多可多重启动 7 个指令。

● 动作模式



1 启动

若在原点处将动作开始开关拨到“ON”，将通过直线插补动作移动到（10.00mm,5.00mm）。

2 连续移动

直接通过圆弧插补动作移动到（15.00mm, 10.00mm），通过直线插补动作移动到（15.00mm, 15.00mm），再通过圆弧插补动作定位到（20.00mm, 20.00mm），然后停止。

此时的速度为 10.00mm/s。

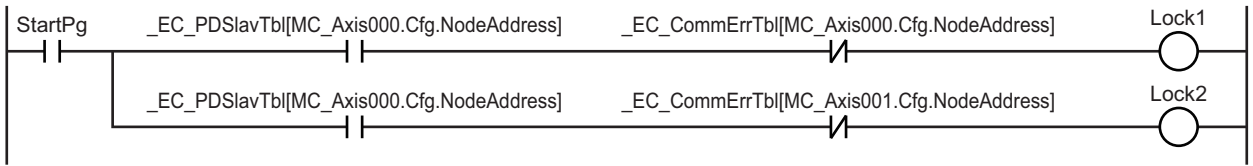
梯形图

● 主要变量

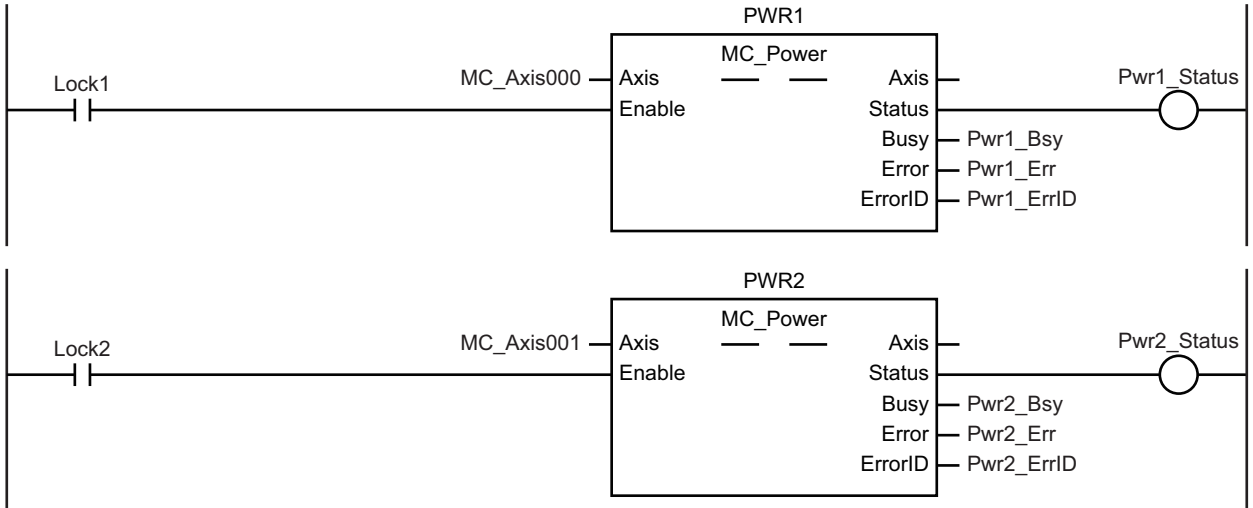
名称	数据类型	初始值	注释
MC_Group000	_sGROUP_REF	—	轴组 0 的轴组变量。
MC_Group000.MFaultLvl.Active	BOOL	FALSE	轴组 0 发生轻度故障等级的异常时，变为 TRUE。
MC_Group000.Status.Disabled	BOOL	FALSE	轴组 0 为无效时变为 TRUE。
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis001	_sAXIS_REF	—	轴 2 的轴变量。
MC_Axis001.Details.Homed	BOOL	FALSE	轴 2 处于原点确定状态时变为 TRUE。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴 2 发生轻度故障等级的异常时，变为 TRUE。
Pwr1_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR1 的输出 Status 中的变量。若变为伺服 ON 状态，则变为 TRUE。
Pwr2_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR2 的输出 Status 中的变量。若变为伺服 ON 状态，则变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态，则轴组中所属的各轴变为伺服 ON 状态。
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE 时，设定输入参数。输入参数设定完成后，变为 TRUE。

● 示例程序

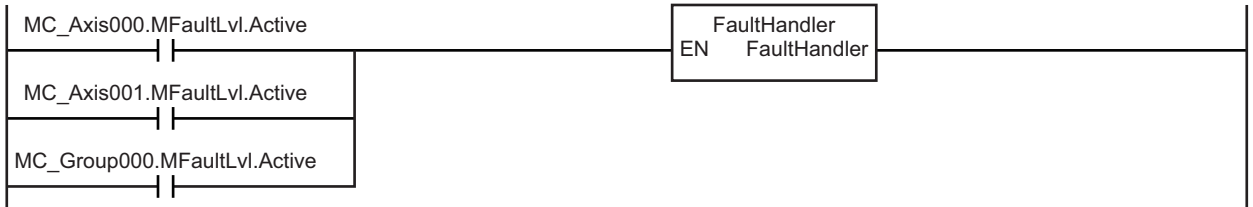
接点 StartPg 变为 TRUE 后，确认各轴的 EtherCAT 通信是否正常进行过程数据通信



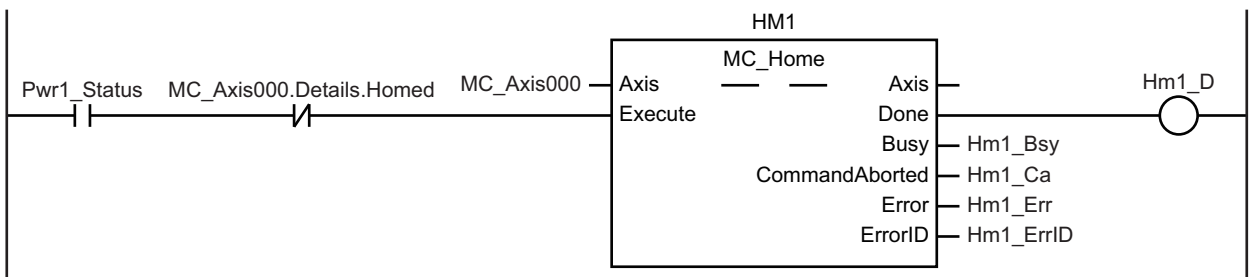
若正在进行过程数据通信，则变为伺服 ON 状态



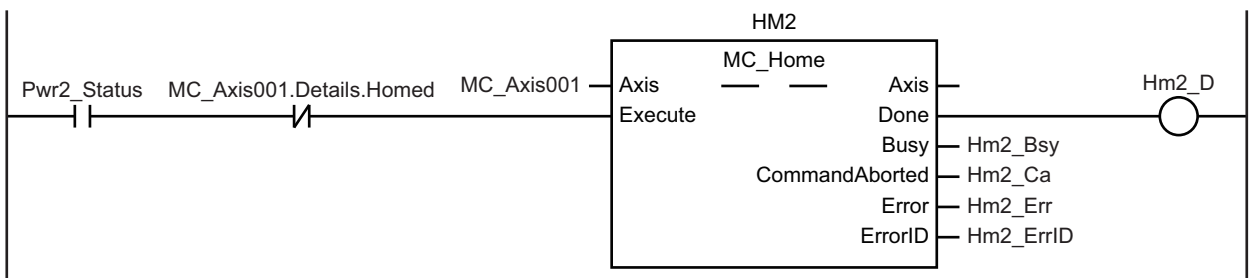
轴构成发生轻度故障等级的异常时，执行装置发生异常时的处理（FaultHandler）。发生异常时的处理（FaultHandler）请根据装置由客户进行编程



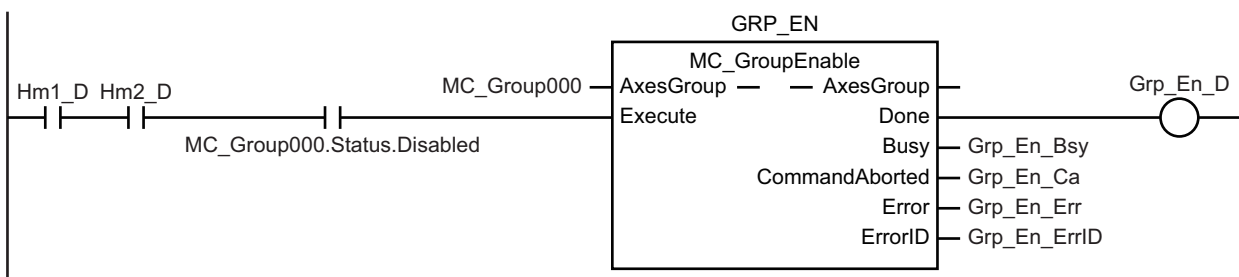
轴 1 变为伺服 ON 状态且处于原点未确定状态时，执行原点复位



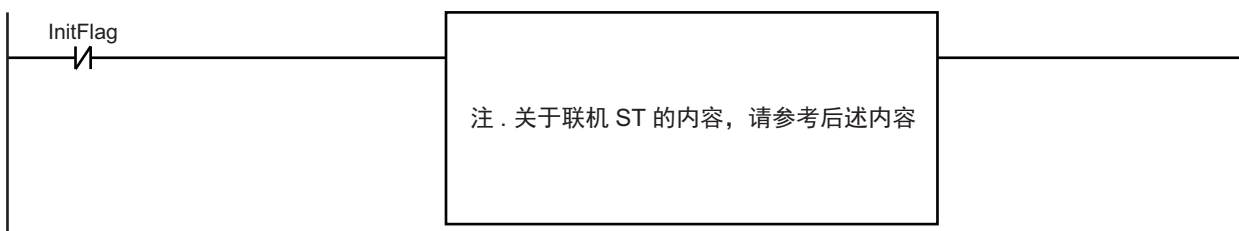
轴 2 变为伺服 ON 状态且处于原点未确定状态时，执行原点复位



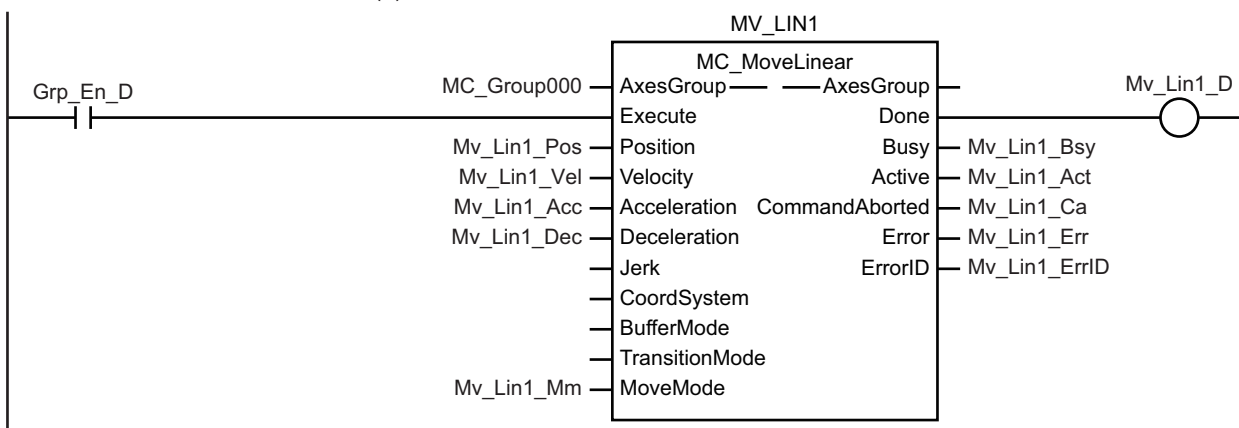
确定轴 1 和轴 2 的原点后，将轴组设为有效



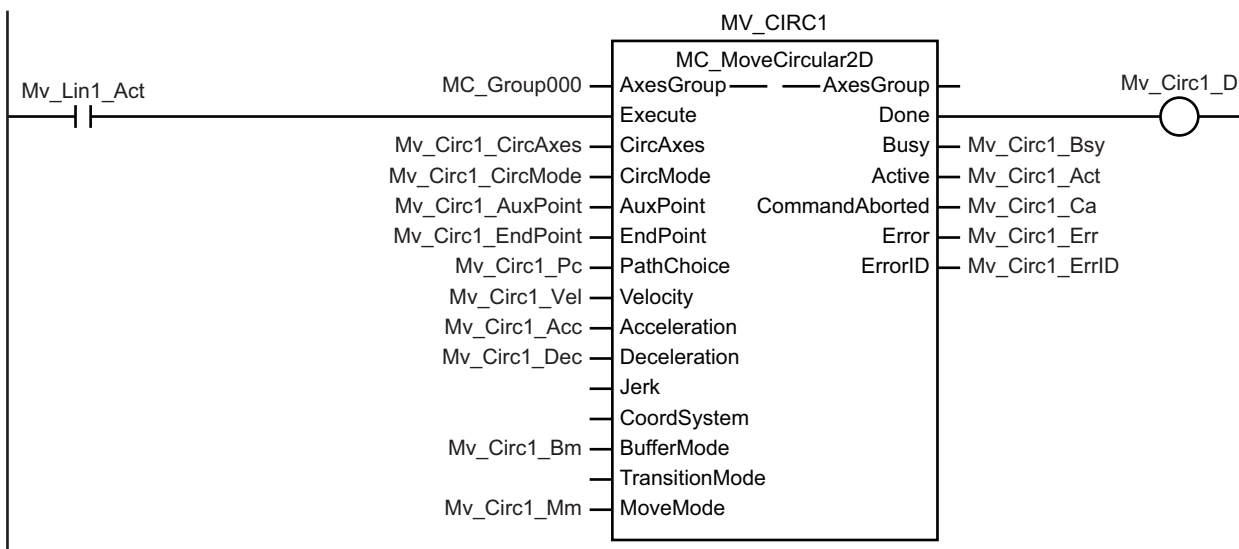
设定直线插补和圆弧插补的参数



确认轴组变为有效，执行直线插补 (1)



启动直线插补 (1) 后，多重启动圆弧插补 (2)

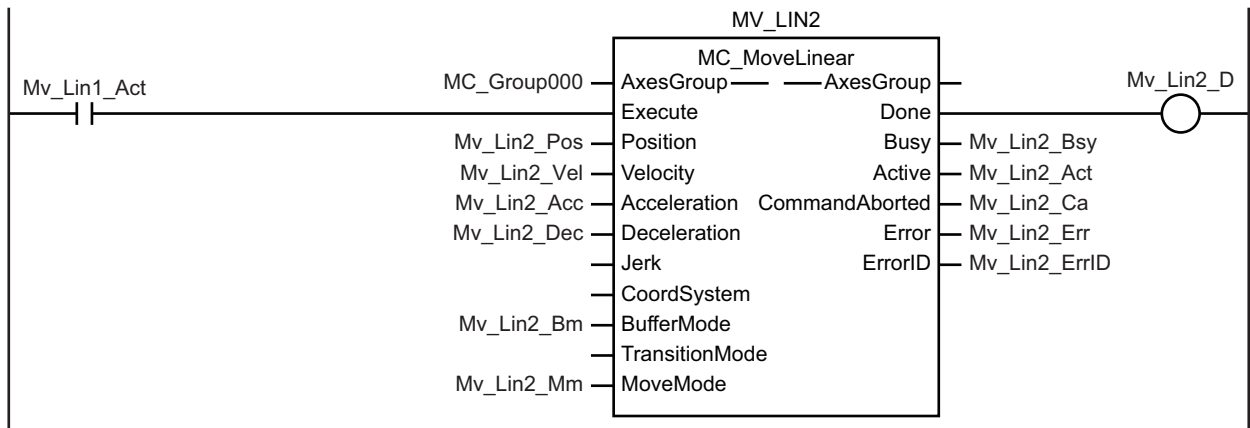


MC_MoveCircular2D

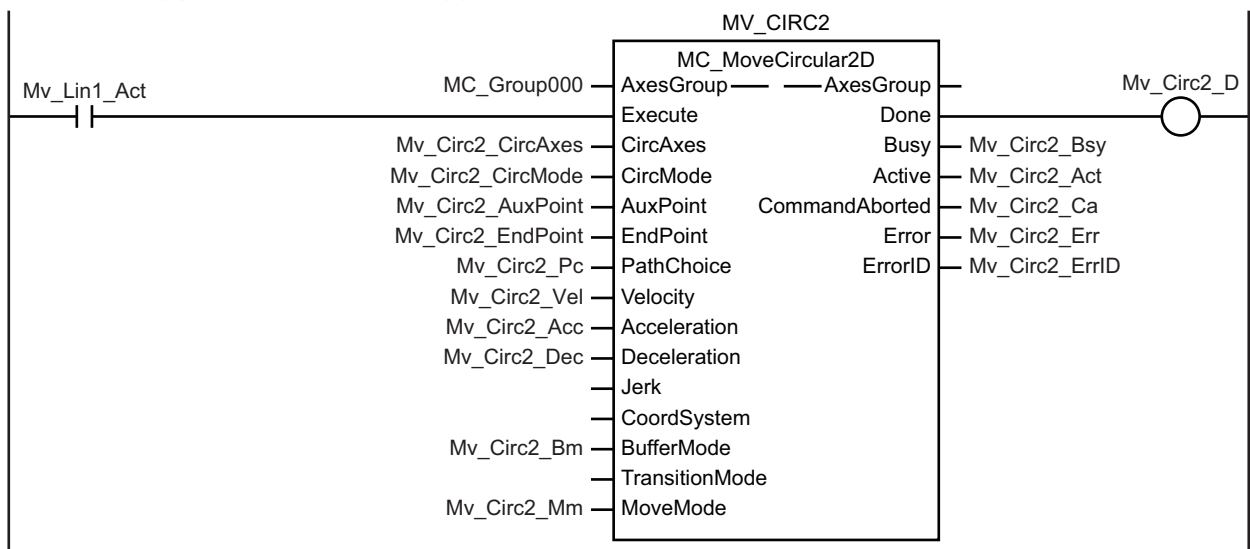
4

示例程序

启动直线插补 (1) 后, 多重启动直线插补 (3)



启动直线插补 (1) 后, 多重启动圆弧插补 (4)



联机 ST 的内容

```
//MV_CIRC1 参数
Mv_Circ1_CircAxes[0] := UINT#0;
Mv_Circ1_CircAxes[1] := UINT#1;
Mv_Circ1_CircMode    := _eMC_CIRC_MODE#_mcRadius;
Mv_Circ1_AuxPoint[0] := LREAL#5.0;
Mv_Circ1_AuxPoint[1] := LREAL#0.0;
Mv_Circ1_EndPoint[0] := LREAL#15.0;
Mv_Circ1_EndPoint[1] := LREAL#10.0;
Mv_Circ1_Pc         := _eMC_CIRC_PATHCHOICE#_mcCCW;
Mv_Circ1_Vel       := LREAL#100.0;
Mv_Circ1_Acc       := LREAL#20.0;
Mv_Circ1_Dec       := LREAL#20.0;
Mv_Circ1_Bm        := _eMC_BUFFER_MODE#_mcBuffered;
Mv_Circ1_Mm        := _eMC_MOVE_MODE#_mcAbsolute;
```

```
//MV_CIRC2 参数
Mv_Circ2_CircAxes[0] := UINT#0;
Mv_Circ2_CircAxes[1] := UINT#1;
Mv_Circ2_CircMode    := _eMC_CIRC_MODE#_mcCenter;
Mv_Circ2_AuxPoint[0] := LREAL#15.0;
Mv_Circ2_AuxPoint[1] := LREAL#20.0;
Mv_Circ2_EndPoint[0] := LREAL#20.0;
Mv_Circ2_EndPoint[1] := LREAL#20.0;
Mv_Circ2_Pc         := _eMC_CIRC_PATHCHOICE#_mcCW;
Mv_Circ2_Vel       := LREAL#100.0;
Mv_Circ2_Acc       := LREAL#20.0;
```



```
Mv_Circ2_Dec := LREAL#20.0;
Mv_Circ2_Bm := _eMC_BUFFER_MODE#_mcBuffered;
Mv_Circ2_Mm := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN1 参数
Mv_Lin1_Pos[0] := LREAL#10.0;
Mv_Lin1_Pos[1] := LREAL#5.0;
Mv_Lin1_Vel := LREAL#100.0;
Mv_Lin1_Acc := LREAL#20.0;
Mv_Lin1_Dec := LREAL#20.0;
Mv_Lin1_Mm := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN2 参数
Mv_Lin2_Pos[0] := LREAL#15.0;
Mv_Lin2_Pos[1] := LREAL#15.0;
Mv_Lin2_Vel := LREAL#100.0;
Mv_Lin2_Acc := LREAL#20.0;
Mv_Lin2_Dec := LREAL#20.0;
Mv_Lin2_Bm := _eMC_BUFFER_MODE#_mcBuffered;
Mv_Lin2_Mm := _eMC_MOVE_MODE#_mcAbsolute;

// 设定输入参数后，将 InitFlag 设为 TRUE
InitFlag := TRUE;
```

结构文本 (ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Group000	_sGROUP_REF	—	轴组 0 的轴组变量。
MC_Group000.MFaultLvl.Active	BOOL	FALSE	轴组 0 发生轻度故障等级的异常时, 变为 TRUE。
MC_Group000.Status.Disabled	BOOL	FALSE	轴组 0 为无效时变为 TRUE。
MC_Axis000	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时, 变为 TRUE。
MC_Axis001	_sAXIS_REF	—	轴 2 的轴变量。
MC_Axis001.Details.Homed	BOOL	FALSE	轴 2 处于原点确定状态时变为 TRUE。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴 2 发生轻度故障等级的异常时, 变为 TRUE。
Pwr1_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR1 的输出 Status 中的变量。若变为伺服 ON 状态, 则变为 TRUE。
Pwr2_Status	BOOL	FALSE	分配到 MC_Power 的实例 PWR2 的输出 Status 中的变量。若变为伺服 ON 状态, 则变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态, 则轴组中所属的各轴变为伺服 ON 状态。
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE 时, 设定输入参数。输入参数设定完成后, 变为 TRUE。
Hm1_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_Home 的实例 HM1。
Hm2_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_Home 的实例 HM2。
Grp_En_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_GroupEnable 的实例 GRP_EN。
Mv_Lin1_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_MoveLinear 的实例 MV_LIN1。
Mv_Lin2_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_MoveLinear 的实例 MV_LIN2。
Mv_Circ1_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_MoveCircular 的实例 MV_CIRC1。
Mv_Circ2_Ex	BOOL	FALSE	该变量从 FALSE 变为 TRUE 时, 启动 MC_MoveCircular 的实例 MV_CIRC2。

● 示例程序

```

// 未设定输入参数时的处理
IF InitFlag=FALSE THEN

    //MV_CIRC1 参数
    Mv_Circ1_CircAxes[0] := UINT#0;
    Mv_Circ1_CircAxes[1] := UINT#1;
    Mv_Circ1_CircMode := _eMC_CIRC_MODE#_mcRadius;
    Mv_Circ1_AuxPoint[0] := LREAL#5.0;
    Mv_Circ1_AuxPoint[1] := LREAL#0.0;
    Mv_Circ1_EndPoint[0] := LREAL#15.0;
    Mv_Circ1_EndPoint[1] := LREAL#10.0;
    Mv_Circ1_Pc := _eMC_CIRC_PATHCHOICE#_mcCCW;
    Mv_Circ1_Vel := LREAL#100.0;
    Mv_Circ1_Acc := LREAL#20.0;
    Mv_Circ1_Dec := LREAL#20.0;
    Mv_Circ1_Bm := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Circ1_Mm := _eMC_MOVE_MODE#_mcAbsolute;

    //MV_CIRC2 参数
    Mv_Circ2_CircAxes[0] := UINT#0;
    Mv_Circ2_CircAxes[1] := UINT#1;
    Mv_Circ2_CircMode := _eMC_CIRC_MODE#_mcCenter;
    Mv_Circ2_AuxPoint[0] := LREAL#15.0;
    Mv_Circ2_AuxPoint[1] := LREAL#20.0;
    Mv_Circ2_EndPoint[0] := LREAL#20.0;
    Mv_Circ2_EndPoint[1] := LREAL#20.0;
    Mv_Circ2_Pc := _eMC_CIRC_PATHCHOICE#_mcCW;
    Mv_Circ2_Vel := LREAL#100.0;
    Mv_Circ2_Acc := LREAL#20.0;
    Mv_Circ2_Dec := LREAL#20.0;
    Mv_Circ2_Bm := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Circ2_Mm := _eMC_MOVE_MODE#_mcAbsolute;

    //MV_LIN1 参数
    Mv_Lin1_Pos[0] := LREAL#10.0;
    Mv_Lin1_Pos[1] := LREAL#5.0;
    Mv_Lin1_Vel := LREAL#100.0;
    Mv_Lin1_Acc := LREAL#20.0;
    Mv_Lin1_Dec := LREAL#20.0;
    Mv_Lin1_Mm := _eMC_MOVE_MODE#_mcAbsolute;

    //MV_LIN2 参数
    Mv_Lin2_Pos[0] := LREAL#15.0;
    Mv_Lin2_Pos[1] := LREAL#15.0;
    Mv_Lin2_Vel := LREAL#100.0;
    Mv_Lin2_Acc := LREAL#20.0;
    Mv_Lin2_Dec := LREAL#20.0;
    Mv_Lin2_Bm := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Lin2_Mm := _eMC_MOVE_MODE#_mcAbsolute;

    // 设定输入参数后, 将 InitFlag 设为 TRUE
    InitFlag := TRUE;

END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 1 设为伺服 ON 状态。
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
    AND (_EC_PDSlavTbI[MC_Axis000.Cfg.NodeAddress]=TRUE)
    AND (_EC_CommErrTbI[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
        Pwr1_En:=TRUE; // 将轴 1 设为伺服 ON

```

```

ELSE
    Pwr1_En:=FALSE;    // 将轴 1 设为伺服 OFF
END_IF;

//StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 2 设为伺服 ON 状态。
//EtherCAT 通信未正常通信时, 设为伺服 OFF。
IF (StartPg=TRUE)
    AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
    AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
        Pwr2_En:=TRUE;    // 将轴 2 设为伺服 ON
    ELSE
        Pwr2_En:=FALSE;    // 将轴 2 设为伺服 OFF
    END_IF;

// 发生轻度故障等级异常时的处理
// 发生异常时的处理 (FaultHandler) 内容请根据装置由客户进行编程。
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) OR
(MC_Group000.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;

// 轴 1 为伺服锁定状态且处于原点未确定状态时, 执行原点复位。
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

// 轴 2 为伺服锁定状态且处于原点未确定状态时, 执行原点复位。
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;

// 若轴 1 和轴 2 完成原点复位且轴组 0 为无效, 则将轴组 0 设为有效
IF (Hm1_D=TRUE) AND (Hm2_D=TRUE) AND (MC_Group000.Status.Disabled=TRUE) THEN
    Grp_En_Ex:= TRUE;
END_IF;

// 轴组 0 变为有效后, 启动直线插补 (1)
IF Grp_En_D=TRUE THEN
    Mv_Lin1_Ex:=TRUE;
END_IF;

// 直线插补 (1) 的输出 Active 变为 TRUE 后, 多重启动后面的指令
IF Mv_Lin1_Act=TRUE THEN
    Mv_Circ1_Ex:=TRUE;
    Mv_Lin2_Ex:=TRUE;
    Mv_Circ2_Ex:=TRUE;
END_IF;

// 轴 1 的 MC_Power
PWR1(
    Axis      := MC_Axis000,
    Enable    := Pwr1_En,
    Status    => Pwr1_Status,
    Busy      => Pwr1_Bsy,
    Error     => Pwr1_Err,
    ErrorID   => Pwr1_ErrID
);

// 轴 2 的 MC_Power
PWR2(
    Axis      := MC_Axis001,
    Enable    := Pwr2_En,
    Status    => Pwr2_Status,

```

```

    Busy      => Pwr2_Bsy,
    Error     => Pwr2_Err,
    ErrorID   => Pwr2_ErrID
);

// 轴 1 的 MC_Home
HM1(
    Axis      := MC_Axis000,
    Execute   := Hm1_Ex,
    Done      => Hm1_D,
    Busy      => Hm1_Bsy,
    CommandAborted => Hm1_Ca,
    Error     => Hm1_Err,
    ErrorID   => Hm1_ErrID
);

// 轴 2 的 MC_Home
HM2(
    Axis      := MC_Axis001,
    Execute   := Hm2_Ex,
    Done      => Hm2_D,
    Busy      => Hm2_Bsy,
    CommandAborted => Hm2_Ca,
    Error     => Hm2_Err,
    ErrorID   => Hm2_ErrID
);

// 轴组 0 有效
GRP_EN(
    AxesGroup := MC_Group000,
    Execute   := Grp_En_Ex,
    Done      => Grp_En_D,
    Busy      => Grp_En_Bsy,
    CommandAborted => Grp_En_Ca,
    Error     => Grp_En_Err,
    ErrorID   => Grp_En_ErrID
);

// 直线插补 (1)
MV_LIN1(
    AxesGroup := MC_Group000,
    Execute   := Mv_Lin1_Ex,
    Position  := Mv_Lin1_Pos,
    Velocity  := Mv_Lin1_Vel,
    Acceleration := Mv_Lin1_Acc,
    Deceleration := Mv_Lin1_Dec,
    MoveMode  := Mv_Lin1_Mm,
    Done      => Mv_Lin1_D,
    Busy      => Mv_Lin1_Bsy,
    Active    => Mv_Lin1_Act,
    CommandAborted => Mv_Lin1_Ca,
    Error     => Mv_Lin1_Err,
    ErrorID   => Mv_Lin1_ErrID
);

// 圆弧插补 (2)
MV_CIRC1(
    AxesGroup := MC_Group000,
    Execute   := Mv_Circ1_Ex,
    CircAxes  := Mv_Circ1_CircAxes,
    CircMode  := Mv_Circ1_CircMode,
    AuxPoint  := Mv_Circ1_AuxPoint,
    EndPoint  := Mv_Circ1_EndPoint,
    PathChoice := Mv_Circ1_Pc,

```

```

Velocity          := Mv_Circ1_Vel,
Acceleration      := Mv_Circ1_Acc,
Deceleration     := Mv_Circ1_Dec,
BufferMode       := Mv_Circ1_Bm,
MoveMode         := Mv_Circ1_Mm,
Done             => Mv_Circ1_D,
Busy             => Mv_Circ1_Bsy,
Active           => Mv_Circ1_Act,
CommandAborted  => Mv_Circ1_Ca,
Error            => Mv_Circ1_Err,
ErrorID         => Mv_Circ1_ErrID
);

// 直线插补 (3)
MV_LIN2(
  AxesGroup       := MC_Group000,
  Execute         := Mv_Lin2_Ex,
  Position        := Mv_Lin2_Pos,
  Velocity        := Mv_Lin2_Vel,
  Acceleration    := Mv_Lin2_Acc,
  Deceleration    := Mv_Lin2_Dec,
  BufferMode       := Mv_Lin2_Bm,
  MoveMode        := Mv_Lin2_Mm,
  Done            => Mv_Lin2_D,
  Busy            => Mv_Lin2_Bsy,
  Active          => Mv_Lin2_Act,
  CommandAborted => Mv_Lin2_Ca,
  Error           => Mv_Lin2_Err,
  ErrorID        => Mv_Lin2_ErrID
);

// 圆弧插补 (4)
MV_CIRC2(
  AxesGroup       := MC_Group000,
  Execute         := Mv_Circ2_Ex,
  CircAxes        := Mv_Circ2_CircAxes,
  CircMode        := Mv_Circ2_CircMode,
  AuxPoint        := Mv_Circ2_AuxPoint,
  EndPoint        := Mv_Circ2_EndPoint,
  PathChoice      := Mv_Circ2_Pc,
  Velocity        := Mv_Circ2_Vel,
  Acceleration    := Mv_Circ2_Acc,
  Deceleration    := Mv_Circ2_Dec,
  BufferMode       := Mv_Circ2_Bm,
  MoveMode        := Mv_Circ2_Mm,
  Done            => Mv_Circ2_D,
  Busy            => Mv_Circ2_Bsy,
  Active          => Mv_Circ2_Act,
  CommandAborted => Mv_Circ2_Ca,
  Error           => Mv_Circ2_Err,
  ErrorID        => Mv_Circ2_ErrID
);

```

MC_GroupStop

使插补动作中的所有轴减速停止。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_GroupStop	轴组 强制停止	FB		<pre>MC_GroupStop_instance (AxesGroup := 《参数》, Execute := 《参数》, Deceleration := 《参数》, Jerk := 《参数》, BufferMode := 《参数》, Done => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。 单位为 [指令单位 /s ²]。*1
Jerk	跃度	LREAL	正数或“0”	0	指定跃度 单位为 [指令单位 /s ³]。*1
BufferMode	缓冲 模式选择	_eMC_ BUFFER_ MODE	0: _mcAborting	0 *2	指定运动指令多重启动时的动作。 0: 中断

*1. 指令单位请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. 如果是有效范围为枚举型的变量初始值，实际不是数值，而是列举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* 请参见「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	指令完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	开始了指令时	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动（中断），本指令中止时 因发生异常，本指令中止时 正在发生异常时，启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
AxesGroup	轴组	_sGROUP_REF	—	指定轴组。 ^{*1}

*1. 请使用 Sysmac Studio 的轴组基本设定画面中创建的用户定义变量的轴组变量名称（默认“MC_Group****”）或系统定义变量的轴组变量名称（_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*]）。

功能说明

- 本指令可停止正在通过轴组指令动作的 AxesGroup(轴组) 中指定的轴组下所有的轴, 轴组指令变为无效状态。
停止轴时, 以 Deceleration(减速度) 减速停止的指令为 MC_MoveLinear (直线插补) 指令、MC_MoveLinearAbsolute (绝对值直线插补) 指令、MC_MoveLinearRelative (相对值直线插补) 指令、MC_MoveCircular2D (2 轴圆弧插补) 指令。
停止轴时, 与 Deceleration(减速度) 的设定值无关, 立即停止的指令为 MC_GroupSyncMoveAbsolute (轴组周期性同步绝对位置控制) 指令。
- 因本指令的启动, 正在执行插补动作的指令将变为 CommandAborted (执行中断)。
- 若在执行插补指令的过程中执行本指令, 将在直线插补或圆弧插补的轨迹上减速停止。
- 在 Execute (启动) 的上升沿开始减速停止的动作。



使用注意事项

轴组变量 Status.ErrorStop (轴组错误减速停止中) 为 TRUE 时, 无法启动本指令。
若要在错误减速停止过程中停止轴组的动作, 请使用 MC_GroupImmediateStop (立即停止) 指令。

指令的详情

下面介绍本指令的详情。

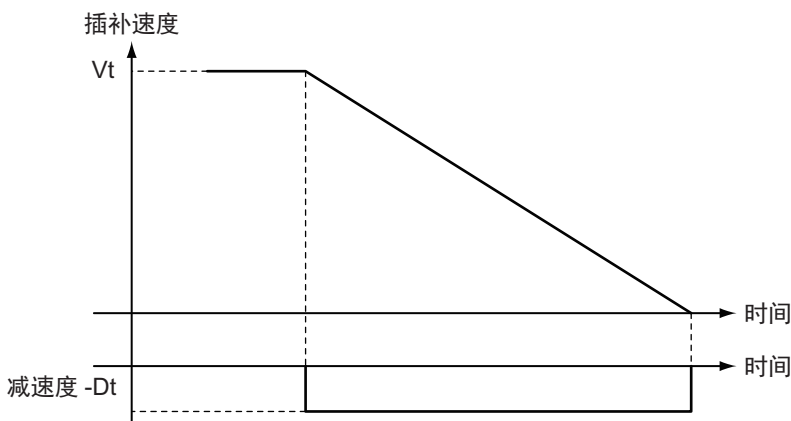
● Deceleration (减速度)、Jerk (跃度)

减速停止时的减速度、跃度在输入变量 Deceleration (减速度)、Jerk (跃度) 中指定。
执行了本指令时, 将对插补速度使用这里指定的 Deceleration (减速度)、Jerk (跃度)。

Jerk (跃度)

Jerk (跃度) 指定为 “0” 时和指定为 “0” 以外时的减速度、插补速度的关系如下所示。

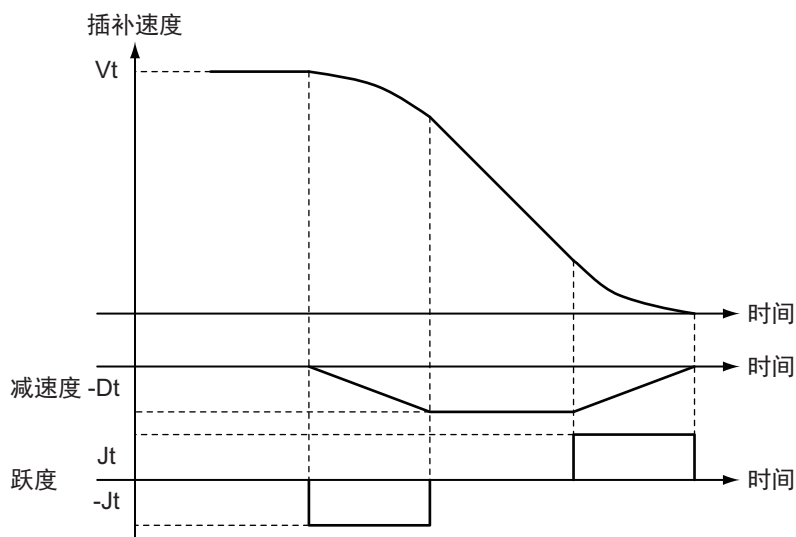
- Jerk (跃度) 为 “0” 时
用减速度 Dt 创建速度的指令值。



V_t : 减速启动时的速度、 Dt : 减速度的指定值

- Jerk（跃度）为“0”以外时

将减速度的上限作为 Dt ，以当前速度为基点，创建速度的指令值。



V_t : 减速启动时的速度、 Dt : 减速度的指定值、 J_t : 跃度的指定值



参考

- 若减速度指定为“0”，则为立即停止指令，立即停止。
- 仅在减速度指定为“0”时，无论轴参数 [加减速超限] 的设定如何，均执行立即停止。

● BufferMode（缓冲模式选择）

指定前一插补动作和本次插补动作的连接方法。

有以下 1 种。

缓冲模式选择	说明
中断	立即中止当前正在执行的指令，执行本指令。

BufferMode（缓冲模式选择）请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 位置检查

通过本指令停止时，不进行位置检查。

时序图

- 在 Execute (启动) 上升的同时 Busy (执行中) 变为 TRUE, 在下一周期, Active (控制中) 变为 TRUE。
- 达到速度 (0) 后, Done (完成) 变为 TRUE。
- 通过其他指令中断了本指令时, CommandAborted (执行中断) 变为 TRUE, Busy (执行中)、Active (控制中) 变为 FALSE。

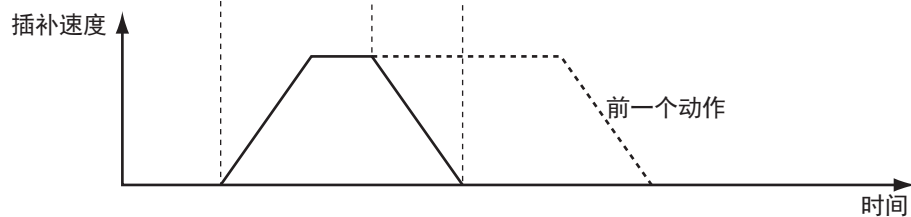
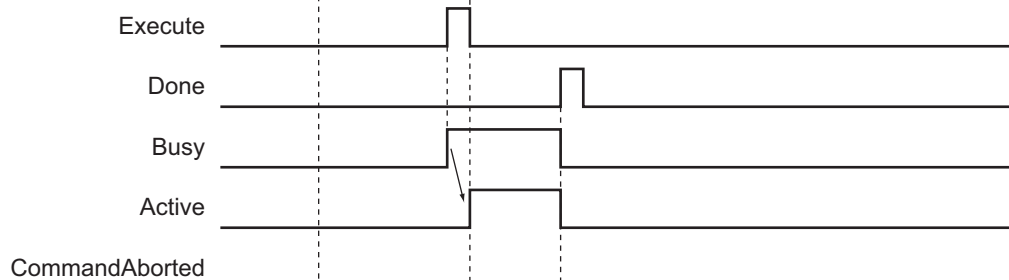
直线插补过程中减速停止时的时序图如下所示。

因本指令的启动, 正在执行插补动作的指令将变为 CommandAborted (执行中断)。

前一指令 (MC_MoveLinear)



本次指令

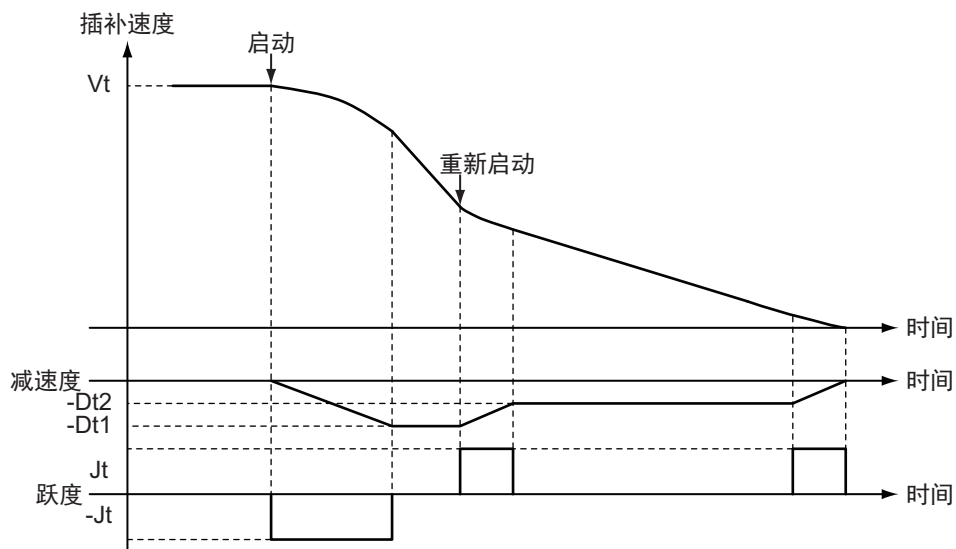


运动指令重新执行

正在执行本指令时，如果再次启动 Execute（启动），将变更减速度。
运动指令的重新执行中，不会变更 Jerk（跃度）。

● Jerk（跃度）为“0”以外时

以当前速度、当前减速度为基点，将变更后的减速度上限作为 $Dt2$ ，创建速度的指令值。



V_t : 减速启动时的速度、 Dt : 减速度的指定值、 J_t : 跃度的指定值

运动指令多重启动

本指令的执行及正在执行本指令时可使用的状态和指令有限制。

运动指令多重启动的详情请参考 □□《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 其他指令执行过程中本指令的启动 轴组无效中

若对无效的轴组执行 MC_GroupStop（轴组强制停止）指令，轴组将发生异常。对轴无影响。

轴组变量的 Status.Stopping（轴组减速停止中）为 TRUE 时

以下状态时，轴变量 Status.Stopping（减速停止中）将变为 TRUE。

- 轴组正在根据 MC_GroupStop（轴组强制停止）指令减速时
- MC_GroupStop（轴组强制停止）指令的 Execute 因多重启动而至少有 1 个为 TRUE 时

若对减速停止状态的轴组启动了以下指令，所启动指令的 CommandAborted（执行中断）将变为 TRUE。

- MC_MoveLinear（直线插补）指令
- MC_MoveLinearAbsolute（绝对值直线插补）指令
- MC_MoveLinearRelative（相对值直线插补）指令
- MC_MoveCircular2D（2轴圆弧插补）指令

正在执行 MC_GroupStop（轴组强制停止）指令时，可多重启动 MC_GroupStop（轴组强制停止）指令。执行中的 MC_GroupStop（轴组强制停止）指令的 Done（完成）变为 TRUE。

轴组变量的 Status.ErrorStop（轴组错误减速停止中）为 TRUE 时

轴组发生异常期间，轴组的 Status.ErrorStop（错误减速停止中）变为 TRUE。

若在 Status.ErrorStop（错误减速停止中）为 TRUE 时启动 MC_GroupStop（轴组强制停止）指令，CommandAborted（执行中断）变为 TRUE。

此时，请使用 MC_GroupImmediateStop（轴组立即停止）指令。

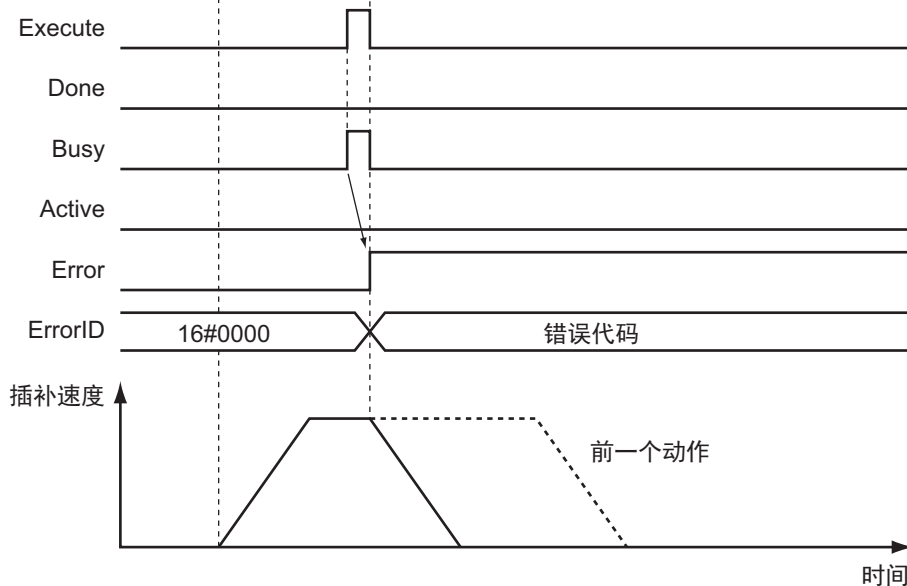
异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。
参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

前一指令（MC_MoveLinear）



本次指令

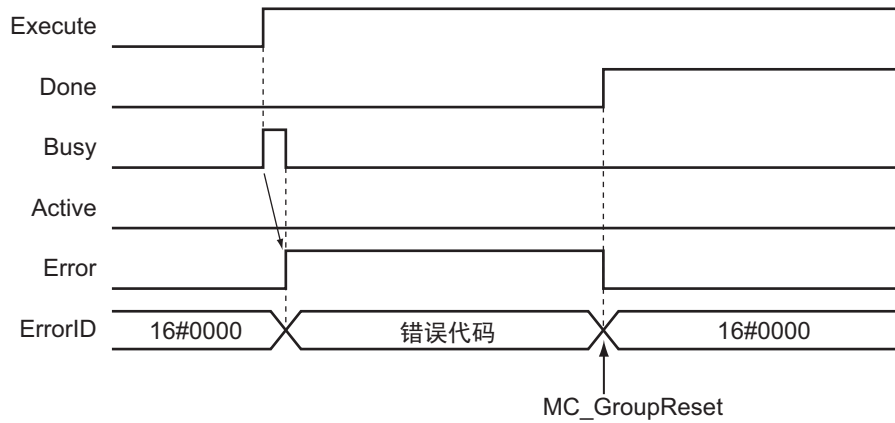


● 异常解除时的动作

在本指令发生异常的状态下，如果在 Execute（启动）为 TRUE 时解除异常，Error（错误）变为 FALSE，Done（完成）变为 TRUE，与减速停止的正常执行时相同，轴状态的 Status.Stopping（减速停止中）变为 TRUE。

请在轴组停止后再解除异常。

轴组动作过程中无法解除异常。



● 错误代码

本指令中发生的异常请参考 □□「A-1 错误代码一览 (P.A-2)」。

MC_GroupImmediateStop

使插补动作中的所有轴立即停止。

若对象轴组为有效，无论轴状态如何，都按照轴参数 [立即停止输入停止方法] 的停止方法，停止所有构成轴。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_GroupImmediateStop	轴组立即停止	FB		<pre>MC_GroupImmediateStop_instance (AxesGroup := 《参数》, Execute := 《参数》, Done => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	完成立即停止时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他 MC_GroupStop (轴组强制停止) 指令进行运动指令多重启动 (中断), 本指令中止时 因发生异常, 本指令中止时 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
AxesGroup	轴组	_sGROUP_REF	—	指定轴组。 *1

*1. 请使用 Sysmac Studio 的轴组基本设定画面中创建的用户定义变量的轴组变量名称 (默认“MC_Group**”) 或系统定义变量的轴组变量名称 (_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])。

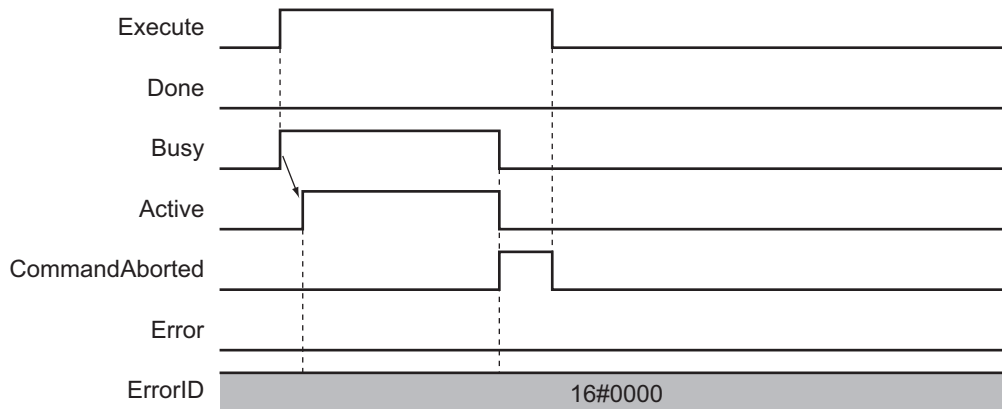
功能说明

- 本指令可对正在通过轴组指令动作的 AxesGroup (轴组) 中指定的轴组下所有的轴, 以及因 MC_GroupStop (轴组强制停止) 指令或发生异常而处于停止状态的轴组执行。
- 若执行指令, 将按照各轴的参数设定 [立即停止输入停止方法] 的指定立即停止, 动作中的指令变为 CommandAborted (执行中断)。
- 若执行本指令, 轴组状态的 Status.ErrorStop (错误减速停止中) 将变为 TRUE, 发生“轴组立即停止指令启动 (错误代码: 5486 Hex) ”。
“轴组立即停止指令启动 (错误代码: 5486 Hex) ”请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363) 》。

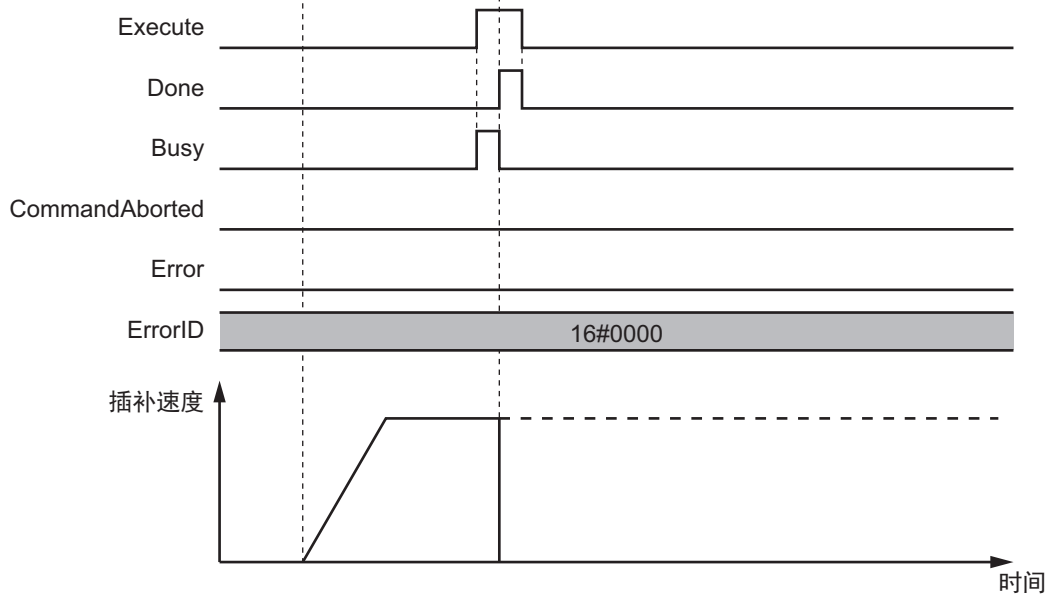
时序图

- 在 Execute（启动）上升的同时 Busy（执行中）变为 TRUE。
- 本指令处理完成后，Done（完成）变为 TRUE。

MC_MoveLiner 指令



MC_GroupImmediateStop 指令



运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”，该轴的动作停止。

运动指令多重启动

运动指令多重启动的详情请参考 □□《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 其他指令执行过程中本指令的启动

轴组无效中

若对无效的轴组执行本指令，轴组将发生异常。对轴无影响。

轴组变量的 Status.Stopping（减速停止中）为 TRUE 时

以下状态时，轴组变量 Status.Stopping（减速停止中）将变为 TRUE。

- 轴组正在根据 MC_GroupStop（轴组强制停止）指令减速时
- MC_GroupStop（轴组强制停止）指令的 Execute 因多重启动而至少有 1 个为 TRUE 时

对于减速停止状态的轴组也可通过本指令多重启动。

若启动本指令，执行中的 MC_GroupStop（轴组强制停止）指令的 CommandAborted（执行中断）将变为 TRUE。

轴组变量的 Status.ErrorStop（错误减速停止中）为 TRUE 时

轴组发生异常期间，轴组的状态 Status.ErrorStop（错误减速停止中）变为 TRUE。

即使因轴组错误而正在减速停止，也可执行本指令。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 错误代码

本指令中发生的异常请参考 □□「A-1 错误代码一览 (P.A-2)」。

MC_GroupSetOverride

变更插补动作中的合成目标速度。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_GroupSetOverride	轴组 超驰值设定	FB		<pre>MC_GroupSetOverride_instance (AxesGroup := 《参数》, Enable := 《参数》, VelFactor := 《参数》, AccFactor := 《参数》, JerkFactor := 《参数》, Enabled => 《参数》, Busy => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>


变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、 FALSE	FALSE	若设定为 TRUE，则超驰变为有效。 若设定为 FALSE，则超驰恢复为“100%”。
VelFactor	速度 超驰值	LREAL	0 ~ 500	100	指定速度的超驰值。 超驰值的有效范围为“0.01 ~ 500.00”。 “500.00 以上”视为“500”，“0.01 以下（包括负数）”视为“0.01”。 仅在指定为“0”时作为“0”动作。 单位为 [%]。
AccFactor (Reserved)	加减速速度 超驰值	LREAL	0 ~ 500	100	(Reserved)
JerkFactor (Reserved)	跃度 超驰值	LREAL	0 ~ 500	100	(Reserved)

输出变量

输出变量	名称	数据类型	有效范围	内容
Enabled	有效	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。 16#0000 表示正常。

*  请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Enabled	开始了本指令的执行时	<ul style="list-style-type: none"> • Enable 为 FALSE 时，1 个周期后 • Error 变为 TRUE 时
Busy	Enabled 的上升沿	<ul style="list-style-type: none"> • Error 变为 TRUE 时
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
AxesGroup	轴组	_sGROUP_REF	—	指定轴组。*1

*1. 请使用 Sysmac Studio 的轴组基本设定画面中创建的用户定义变量的轴组变量名称（默认“MC_Group***”）或系统定义变量的轴组变量名称（_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*]）。

功能说明

- 变更与根据以下指令动作的组的插补目标速度对应的超驰。

变更超驰后，可变更动作中的目标速度。

对象指令为以下指令。

MC_MoveLinear（直线插补）指令	MC_MoveLinearRelative（相对值直线插补）指令
MC_MoveLinearAbsolute（绝对值直线插补）指令	MC_MoveCircular2D（2轴圆弧插补）指令

- 新的目标速度如下所示。
变更后的目标速度 = 当前执行中指令的插补速度 × 超驰 [%]
- 超驰的单位为 [%]。“100”以“100%”表示。
- 变更后的插补速度超过轴组变量的最高插补速度时，为最高插补速度。
- 轴将针对变更后的插补速度，进行加速或减速。
- 超驰指定为“0”时，目标速度变为“0”，组的动作减速停止，速度降至“0”后，变为动作中。动作过程中，轴组变量 Status.Moving（动作中）变为 TRUE。
要在保持动作状态的同时暂停轴动作时，将超驰设为“0”。

- Enable（有效）为 FALSE 时，超驰恢复为 100（=100%）。
- 执行 MC_GroupSetOverride（轴组超驰值设定）指令的过程中发生组异常时，MC_GroupSetOverride（轴组超驰值设定）指令将继续保持 Enabled（有效）。



使用注意事项

若将本指令的 Enable（有效）设为 FALSE，本指令的 Enabled（有效）及 Busy（执行中）将变为 FALSE。
此时，速度将向超驰“100%”加速或减速。



参考

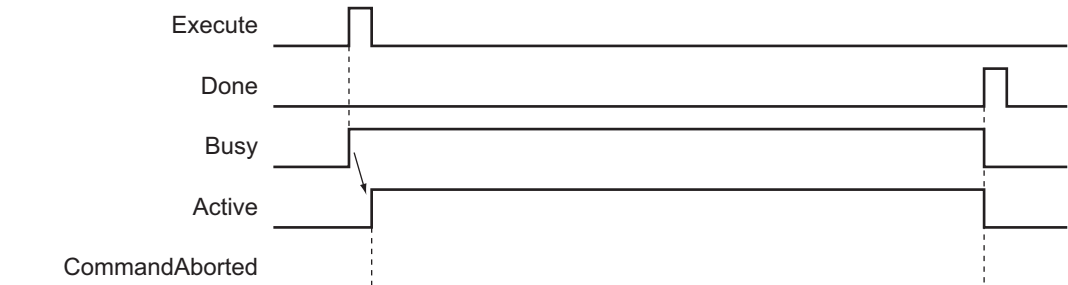
对其他指令的影响

本指令在临时变更各指令的目标速度时使用。
因此，输入中不含目标速度的指令中，无法使用本功能。
但是，对于无法使用本功能的指令，即使将 MC_GroupSetOverride（轴组超驰值设定）指令设为有效，Enabled（有效）将继续保持 TRUE。

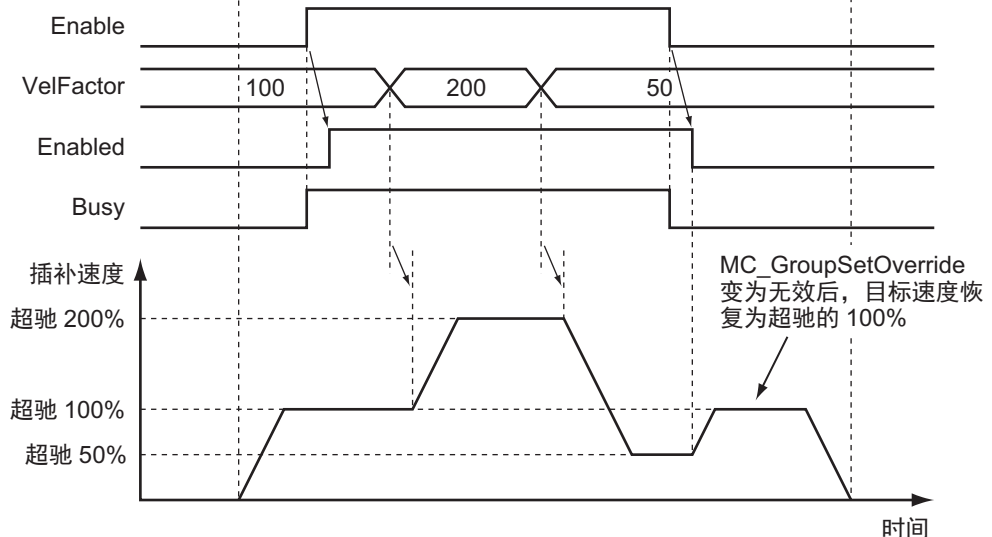
时序图

● 在 MC_MoveLinear（直线插补）指令中使用本指令时

前一指令（MC_MoveLinear）



本次指令



运动指令重新执行

输入为“Enable 型”的指令没有运动指令重新执行。

运动指令多重启动

运动指令多重启动的详情请参考 □□《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 本指令执行过程中其他指令的启动

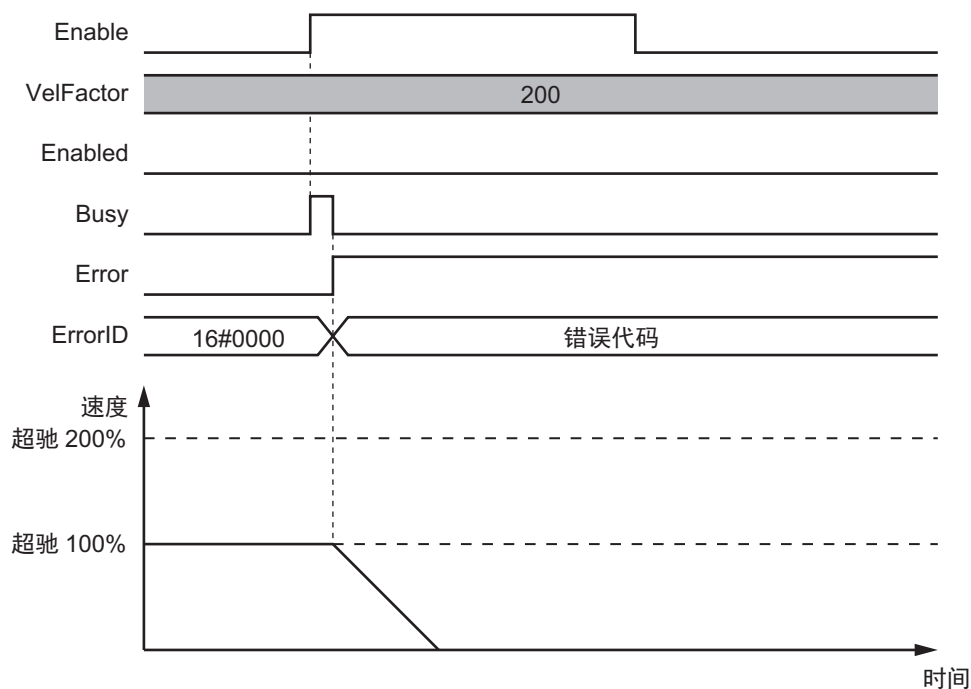
正在执行 MC_GroupSetOverride (轴组超驰值设定) 指令时, 若对同一轴组启动了其他实例的 MC_GroupSetOverride (轴组超驰值设定) 指令, 后面执行的实例将优先处理。

Enabled (有效) 时, 两者均变为 TRUE。

具体动作为, 后面执行的实例的超驰值变为有效。若将后面执行的实例的 Enable (有效) 设为 FALSE, 超驰将变为无效。

异常

执行本指令的过程中发生异常时, Error (错误) 将变为 TRUE, 发生轴组轻度故障时, 轴组停止。参照输出到 ErrorID (错误代码) 的值, 可找出异常的原因。



● 错误代码

指令中发生的异常请参考 □□「A-1 错误代码一览 (P.A-2)」。

MC_GroupReadPosition

获取轴组指令的当前位置和反馈当前位置。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_GroupReadPosition	轴组位置获取	FB		<pre>MC_GroupReadPosition_instance (AxesGroup := 《参数》, Enable := 《参数》, CoordSystem := 《参数》, Valid => 《参数》, Busy => 《参数》, Error => 《参数》, ErrorID => 《参数》, CommandPosition => 《参数》, ActualPosition => 《参数》);</pre>

版本相关信息

本指令可在 CPU 单元版本 1.01 以上和 Sysmac Studio Ver.1.02 以上的组合中使用。

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Enable	有效	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
CoordSystem	坐标系	_eMC_COORD _SYSTEM	0: _mcACS	0 *1	指定坐标系。 0: 轴坐标系 (ACS)

*1. 如果是有效范围为枚举型的变量初始值，实际不是数值，而是列举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
Valid	有效	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*1	发生异常时，输出错误代码。16#0000 表示正常。
CommandPosition	指令当前位置	ARRAY [0..3] OF LREAL	负数、正数、“0”	输出指令位置的当前值。 单位为 [指令单位]。*2
ActualPosition	反馈当前位置	ARRAY [0..3] OF LREAL	负数、正数、“0”	输出反馈位置的当前值。 单位为 [指令单位]。*1

*1. 请参考「A-1 错误代码一览 (P.A-2)」。

*2. 指令单位请参考 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

(注) 1 轴组的轴构成为 2 轴或 3 轴时，空构成轴的 CommandPosition (指令当前位置) 或 ActualPosition (反馈当前位置) 的值将变得不确定。

2 Enable (有效) 变为 FALSE 后，CommandPosition (指令当前位置) 或 ActualPosition (反馈当前位置) 的值将变得不确定。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Valid	Enable 的上升沿	<ul style="list-style-type: none"> • Enable 变为 FALSE 时 • Error 变为 TRUE 时
Busy	Enable 的上升沿	Error 变为 TRUE 时
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
AxesGroup	轴组	_SGROUP_REF	—	指定轴组。*1

*1. 请使用 Sysmac Studio 的轴组基本设定画面中创建的用户定义变量的轴组变量名称 (默认“MC_Group****”) 或系统定义变量的轴组变量名称 (_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])。

功能说明

- Valid（有效）为 TRUE 时，在每个控制周期获取轴组构成轴的指令当前位置和反馈当前位置。
- 各类轴的可使用当前位置的种类如下所示。

轴种类	当前位置的种类	
	指令当前位置	反馈当前位置
伺服轴	可使用	可使用
虚拟伺服轴	可使用	可使用 ^{*1}
编码器轴	不可使用	可使用
虚拟编码器轴	不可使用	可使用

*1. 在虚拟伺服轴中，“反馈位置 = 指令位置”。

在 MC 功能模块内部，以双精度实数型处理，因此可能包括运算误差。

- 原点未确定状态下，也可执行本指令。
- 无论轴组为有效还是无效，均可使用本指令。



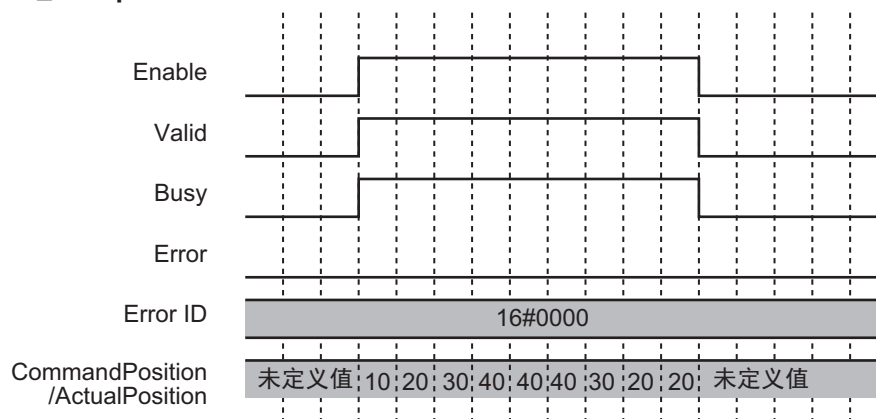
使用注意事项

若在不是分配了轴组的任务中使用本指令，请在使用的任务中，对 Axes（轴组构成轴）中指定的构成轴对应的轴变量进行外部变量声明，然后再使用。

时序图

- 若将 Enable（有效）设为 TRUE，Busy（执行中）和 Valid（有效）将变为 TRUE。
- 若将 Enable（有效）设为 FALSE，Busy（执行中）和 Valid（有效）将变为 FALSE。

MC_GroupReadPosition



运动指令重新执行

输入为“Enable 型”的指令没有运动指令重新执行。

运动指令多重启动

本指令中，各指令单独动作，因此多重启动没有限制。

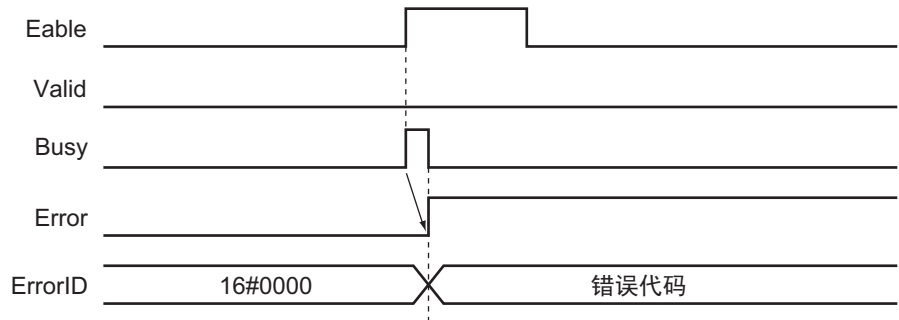
运动指令多重启动的详情请参考 □□《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

MC_GroupReadPosition



● 错误代码

指令中发生的异常请参考 □□「A-1 错误代码一览 (P.A-2)」。

MC_ChangeAxesInGroup

暂时改写轴组参数中的 [构成轴]。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_ChangeAxesInGroup	轴组构成轴写入	FB		<pre>MC_ChangeAxesInGroup_instance (AxesGroup := 《参数》, Axes := 《参数》, Execute := 《参数》, Done => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>



使用注意事项

通过本指令重写的值不会保存到 CPU 单元的非易失性存储器中。因此，控制器的电源“OFF”及开始下载和 MC 功能模块的重新启动处理时，重写的值将消失，恢复为 Sysmac Studio 中设定的值。

要保存到非易失性存储器时，请使用 Sysmac Studio 传送参数。



参考

“下载”时使用 Sysmac Studio 的“同步”功能。



版本相关信息

本指令可在 CPU 单元版本 1.01 以上和 Sysmac Studio Ver.1.02 以上的组合中使用。

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*1	发生异常时，输出错误代码。16#0000 表示正常。

*1. □□ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	指令完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	其他指令上发生异常，中止了本指令时	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

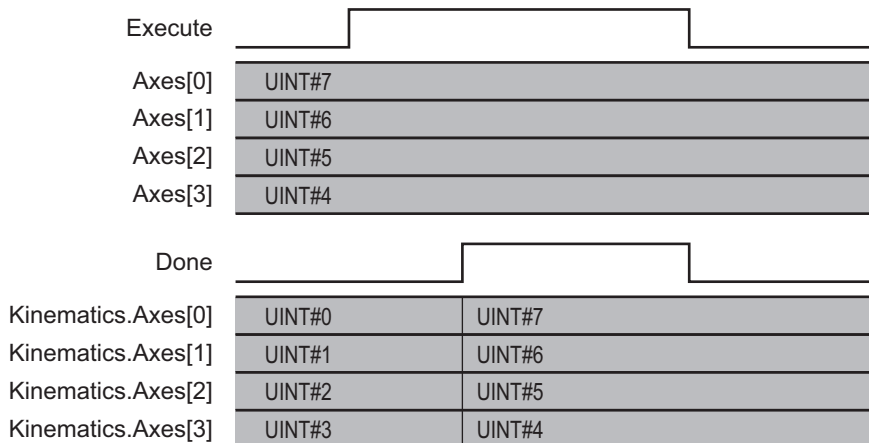
输入输出变量	名称	数据类型	有效范围	内容
AxesGroup	轴组	_sGROUP_REF	—	指定重写对象的轴组。*1
Axes	轴组构成轴	ARRAY [0..3] OF UINT	—	指定重写后构成轴的轴编号。*2

*1. 请使用 Sysmac Studio 的轴组基本设定画面中创建的用户定义变量的轴组变量名称（默认“MC_Group***”）或系统定义变量的轴组变量名称（_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*]）。

*2. 构成轴请设定为伺服轴或虚拟伺服轴。此外，重写后的构成轴数应与写入前的构成轴数保持一致。

功能说明

- 在 Execute（启动）的上升沿，在通过 AxesGroup（轴组）指定的轴组的轴参数中，写入 Axes（轴组构成轴）中指定的构成轴。
- 指令执行完成后，轴组变量 Kinematics（运动学转换设定）的 Axis[] 将重写为指定的构成轴。
- 本指令无法重写轴构成（构成轴数）。例如，对轴构成为 3 轴的轴组执行本指令，轴构成仍为 3 轴。
- 将构成轴的轴编号从“0, 1, 2, 3”变更为“7, 6, 5, 4”时的动作如下所示。



- 无论原点是确定还是未确定，都可执行本指令。
- 仅在轴组为无效时可以执行本指令。



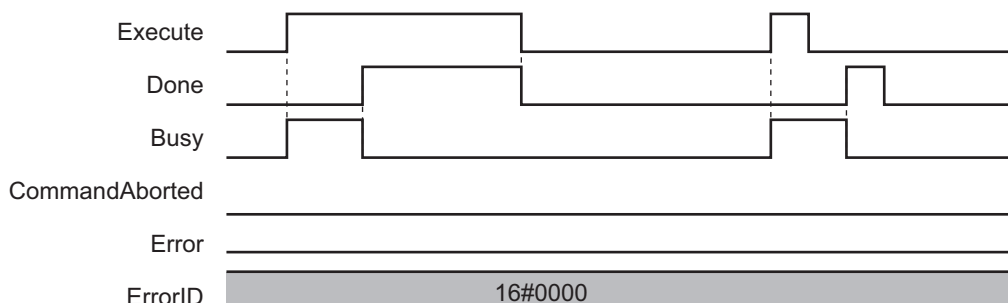
使用注意事项

- 通过本指令重写的值不会保存到 CPU 单元的非易失性存储器中。因此，控制器的电源“OFF”及开始下载和 MC 功能模块的重新启动处理时，重写的值将消失，恢复为 Sysmac Studio 中设定的值。
要保存到非易失性存储器时，请使用 Sysmac Studio 传送参数。
- 使用 NX 系列 CPU 单元时，Axes（轴组构成轴）中指定的所有构成轴与执行指令的对象轴组需要分配到相同的任务中。若指定了分配到不同任务中的轴，将发生“超出轴组构成轴设定范围（错误代码：549D Hex）”。

时序图

- 若将 Execute（启动）设为 TRUE，Busy（执行中）将变为 TRUE。
- 构成轴的重写完成后，Done（完成）变为 TRUE。

MC_ChangeAxesInGroup 指令



运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

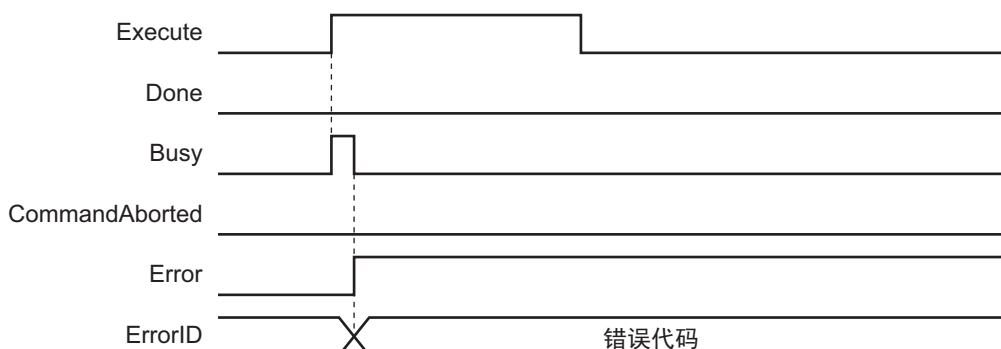
异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE。

此时，将不写入参数，保持执行指令前的值。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 □ 「A-1 错误代码一览 (P.A-2)」。

MC_GroupSyncMoveAbsolute

周期性输出各轴中指定的目标位置。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_GroupSyncMoveAbsolute	轴组周期性同步绝对位置控制	FB		<pre>MC_GroupSyncMoveAbsolute_instance (AxesGroup := 《参数》, Execute := 《参数》, Position := 《参数》, CoordSystem := 《参数》, BufferMode := 《参数》, InPosition => 《参数》, Busy => 《参数》, Active => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

版本相关信息

本指令可在 CPU 单元版本 1.01 以上和 Sysmac Studio Ver.1.02 以上的组合中使用。

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Position	目标位置	ARRAY [0..3] OF LREAL	负数、正数、“0”	0	指定绝对坐标的目标位置。单位为 [指令单位]。 ^{*1}
CoordSystem	坐标系	_eMC_COORD_SYSTEM	0: _mcACS	0 ^{*2}	指定坐标系。 0: 轴坐标系 (ACS)
BufferMode	缓冲模式选择	_eMC_BUFFER_MODE	0: _mcAborting	0	指定运动指令多重启动时的动作。 0: 中断

*1. 指令单位请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

*2. 如果是有效范围为枚举型的变量初始值，实际不是数值，而是列举值。

输出变量

输出变量	名称	数据类型	有效范围	内容
InPosition	位置	BOOL	TRUE、FALSE	所有构成轴的反馈当前位置进入目标位置的位置宽度内时变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
Active	控制中	BOOL	TRUE、FALSE	控制过程中变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
InPosition	所有构成轴的反馈当前位置进入目标位置的位置宽度内时	<ul style="list-style-type: none"> 反馈当前位置超出位置宽度时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
Active	开始了指令时	<ul style="list-style-type: none"> Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	<ul style="list-style-type: none"> 由其他指令进行运动指令多重启动，本指令中止时 其他指令上发生异常，中止了本指令时 正在发生轴异常时，启动了本指令 正在执行 MC_GroupStop 指令时，启动了本指令 	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
AxesGroup	轴组	_sGROUP_REF	—	指定轴组。 ^{*1}

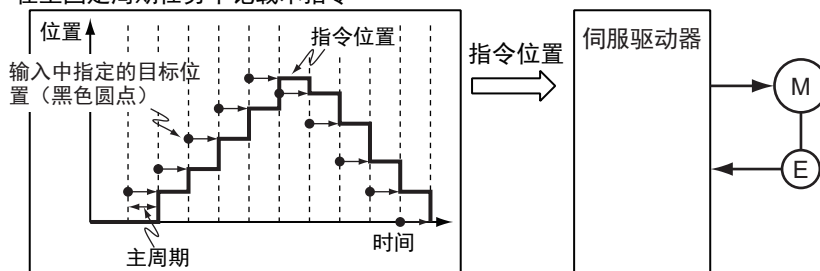
*1. 请使用 Sysmac Studio 的轴组基本设定画面中创建的用户定义变量的轴组变量名称（默认“MC_Group****”）或系统定义变量的轴组变量名称（_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*]）。

功能说明

- 本指令可将来自用户程序的目标位置，按任务周期，以周期性同步位置模式（CSP）输出到伺服驱动器。目标位置以绝对位置指定。
- 速度以轴参数的 [最高速度] 为上限。不会应用最大加速度和最大减速度。
- 在主固定周期任务或固定周期任务（执行优先级 5）中记述本指令时，输入中指定的目标位置将在下一任务周期输出到伺服驱动器。

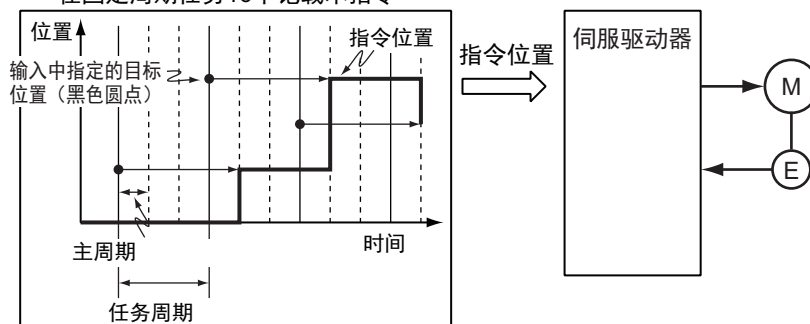
在主固定周期任务中记载本指令的示例如下。固定周期任务（执行优先级 5）中亦同。

在主固定周期任务中记载本指令



- 在固定周期任务 16 中记述本指令时，输入中指定的目标位置将在下一固定周期任务中输出到伺服驱动器。

在固定周期任务 16 中记载本指令



使用注意事项

- 轴组构成轴中有任一轴为原点未确定时，将发生“原点未确定状态下指令启动异常（错误代码：5466 Hex）”。
- 指定目标位置时，请确保与指示的目标位置相应的移动量不会超出轴参数的 [最高速度]。如果指定了超出 [最高速度] 的目标位置，将变为指令速度饱和，输出 [最高速度] 中限制的移动量。不足指示的目标位置的移动量指示将在下一周期后输出。此时，轴控制状态的 Details.VelLimit（指令速度饱和）将变为 TRUE。

指令的详情

下面介绍指令的详情。

● 位置检查

不更新 Position（目标位置）时，目标位置和反馈位置的宽度进入轴参数 [位置宽度] 的范围内后，InPosition（位置）变为 TRUE。

InPosition（位置）为 TRUE 时，即使变更目标位置，该周期中将保持 TRUE，在下一周期变为 FALSE。轴参数的 [位置检查时间] 设定变为无效。

● 停止处理

下面介绍停止轴组动作时的动作。

停止动作时，应使用 MC_GroupStop（轴组强制停止）指令或 MC_GroupImmediateStop（轴组立即停止）指令。若启动这些指令，本指令的 CommandAborted（执行中断）将变为 TRUE。

通过 MC_GroupStop（轴组强制停止）指令停止

立即停止。

通过 MC_GroupImmediateStop（轴组立即停止）指令停止

按照各轴的轴参数设定 [立即停止输入停止方法] 的指定执行立即停止。

● 因发生异常而停止

发生使轴停止的异常时，无论设定如何，都“立即停止”。

● 可使用的轴

- 可在伺服轴中使用本指令。
- 使用时，请将 MC_Power（可运行）指令的 Enable（有效）设为 TRUE（伺服 ON 状态）。
- 虚拟伺服轴始终可接收。

● 启动条件

- 请将轴参数 [计数模式] 设为直线模式。
- 请将所有构成轴设为原点确定状态。
- 请利用 MC_GroupEnable（轴组有效）指令将轴组设为有效。

● 轴变量的状态

轴组变量的轴组状态 Status.Moving（动作中）变为 TRUE。
不会影响轴组控制状态。

● 超驰

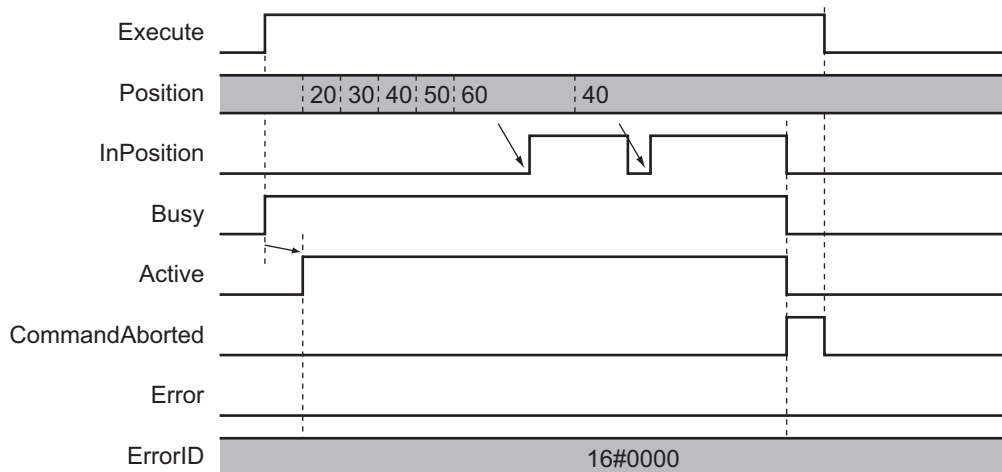
超驰对本指令无效。

时序图

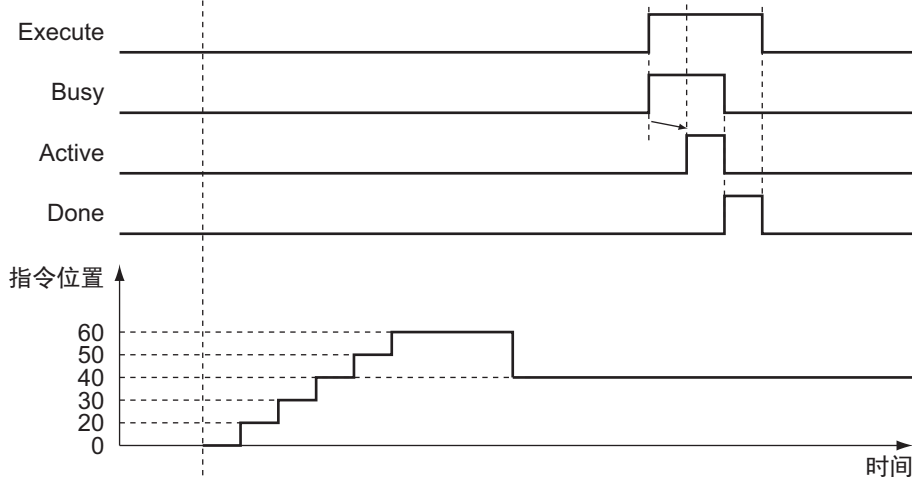
- 在 Execute (启动) 上升的同时 Busy (执行中) 变为 TRUE, 在下一周期, Active (控制中) 变为 TRUE。
- 所有构成轴的反馈当前位置进入 Position (目标位置) 的位置宽度内时, InPosition (位置) 变为 TRUE。
- 通过其他指令中止了本指令时, CommandAborted (执行中断) 变为 TRUE, Busy (执行中)、Active (控制中)、InPosition (位置) 变为 FALSE。
- 要结束本指令, 请启动 MC_GroupStop (轴组强制停止) 指令。

在主固定周期任务中记载本指令时如下所示。

MC_GroupSyncMoveAbsolute 指令



MC_GroupStop 指令



参考

MC 功能模块将对伺服驱动器按上述时序图发出控制模式指令。但是, 反映到伺服驱动器的时间因各伺服驱动器的规格不同而异。

运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

运动指令多重启动的详情请参考 □□《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 其他指令执行过程中本指令的启动

本指令的 BufferMode（缓冲模式选择）仅支持中断。

● 本指令执行过程中其他指令的启动

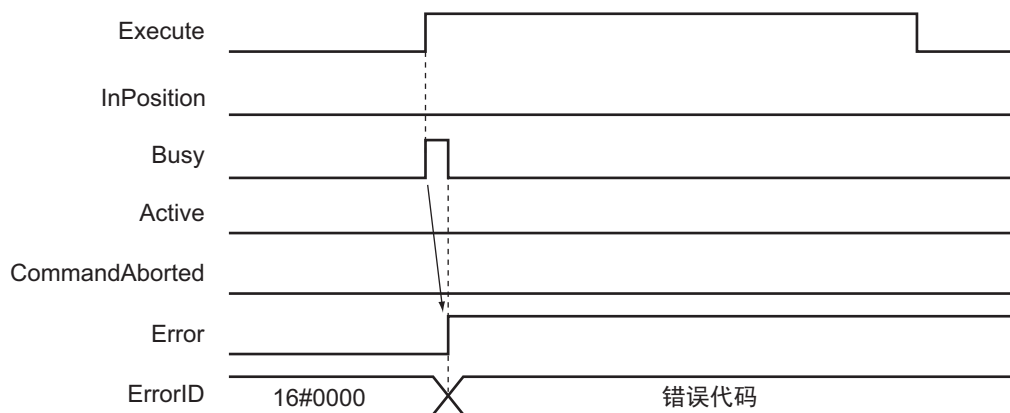
对本指令执行运动指令多重启动时，只能进行中断。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，轴停止。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 □□「A-1 错误代码一览 (P.A-2)」。

MC_GroupReset

解除轴组及轴异常。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_GroupReset	轴组 错误复位	FB		<pre>MC_GroupReset_instance (AxesGroup := 《参数》, Execute := 《参数》, Done => 《参数》, Busy => 《参数》, Failure => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

MC_GroupReset

4

变量

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、 FALSE	FALSE	上升时开始指令。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、 FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
Failure	错误结束	BOOL	TRUE、 FALSE	未正常执行指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	异常解除处理正常结束时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 Failure 变为 TRUE 时
Failure	<ul style="list-style-type: none"> 轴或轴组因为异常而减速停止的过程中, 执行了本指令时 因发生轴共通异常而发生轴组异常的过程中, 执行了本指令时 	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
AxesGroup	轴组	_sGROUP_REF	—	指定轴组。*1

*1. 请使用 Sysmac Studio 的轴组基本设定画面中创建的用户定义变量的轴组变量名称（默认“MC_Group***”）或系统定义变量的轴组变量名称（_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*]）。

功能说明

- 在 Execute（启动）的上升沿，解除 GroupEnable（组有效）状态的 AxesGroup（轴组）中指定的轴组异常及轴组中所属轴的异常。
可解除的异常为轴或轴组中发生的“轻度故障”等级异常、“监视信息”等级的异常，还可解除驱动器错误重置。
- 无论轴的状态为伺服 ON 还是伺服 OFF，都将执行异常解除处理。
- 对于发生驱动器错误的轴，执行驱动器错误重置处理后，执行异常解除处理。
- 在清除驱动器错误或轴参数 [驱动器错误重置监视时间] 的时间内，驱动器错误重置处理将继续执行。驱动器错误重置处理将同时对属于轴组的轴执行。
- 解除对象为 Execute（启动）上升时发生的异常。解除过程中发生的异常不会解除。
- 在轴组因错误而减速停止的过程中，若执行了本指令，将不执行，Failure（错误结束）变为 TRUE。
这是为了在轴停止前，使异常无法解除。
此外，因发生轴共通异常而导致的轴异常也无法通过本指令解除，因此 Failure（错误结束）将变为 TRUE。



使用注意事项

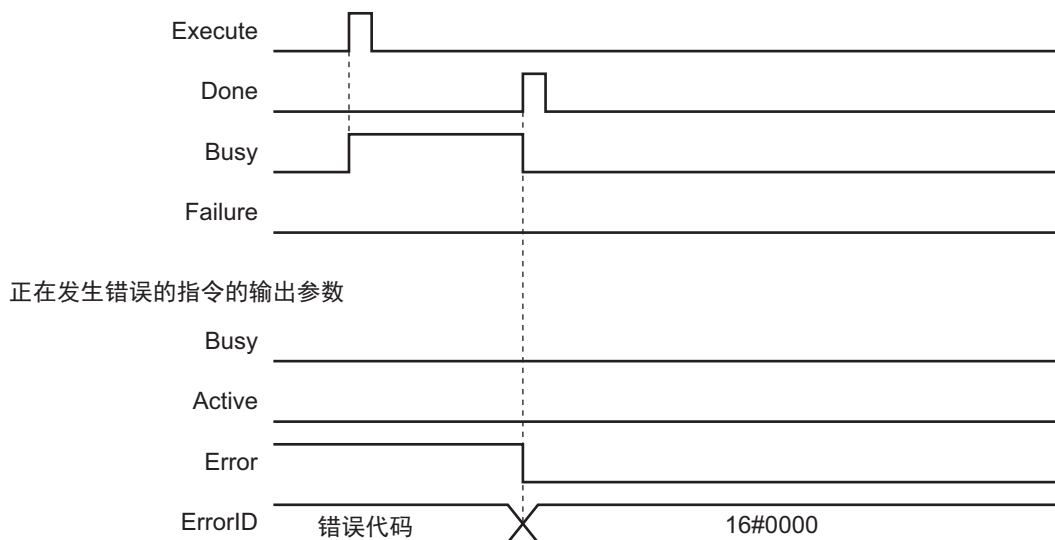
- 本指令的异常解除处理可能跨多个控制周期。
- 启动本指令后轴仍继续动作时，本指令的输出变量“Failure（错误结束）”将变为 TRUE。排除异常原因后，请在变为 Done（完成）变为 TRUE 前，执行重试处理。
- 排除异常发生原因时，请在确认各轴完全停止后再操作。
- 对欧姆龙生产的伺服驱动器 G5 系列使用本指令时，请进行排他处理，避免同时执行 ResetEError（EtherCAT 异常解除）指令。
- 对 NX 系列脉冲输出单元使用本指令时，脉冲输出单元上连接的驱动器的异常状态不会解除。详情请参考 □ 《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。



参考

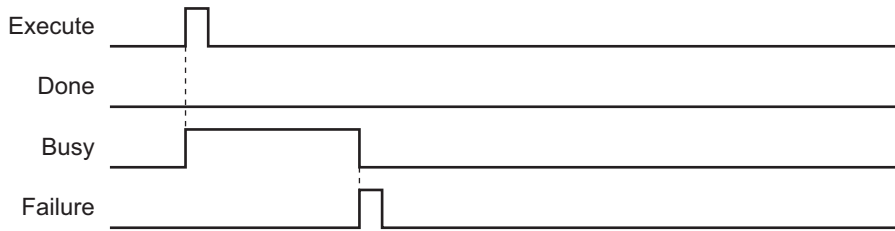
- 轴异常仅在轴组有效时可以解除。
- 以下异常无法通过本指令解除。
轴共通异常 整体
要解除异常，请执行 ResetMError（全部错误重置）指令。
- 对未发生异常的轴组执行了本指令时，指令将正常结束，状态继续保持。

时序图



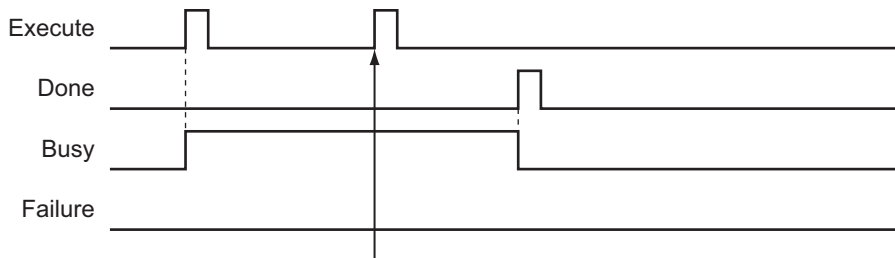
指令的中止

无法解除因异常而导致的轴、轴组减速停止中或因发生轴共通异常而导致的轴组异常时，中断指令。



运动指令重新执行

在本指令的执行过程中再次启动了 Execute（启动）时，将忽略重新执行的指令，继续执行异常解除处理。



不执行重新执行的指令，继续进行执行中的处理

运动指令多重启动

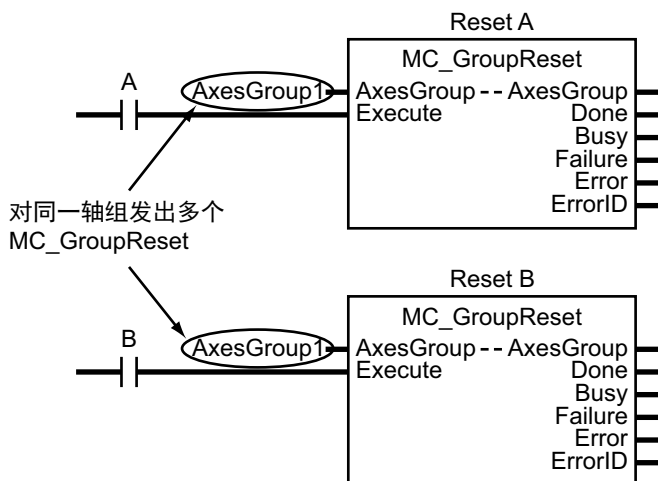
运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● 本指令执行过程中其他指令的启动

对同一轴组执行了其他实例的 MC_GroupReset (轴组错误重置) 指令时, 将分别执行两个指令。

发生从站类异常时, 可能会等到轴设定的“驱动器错误重置监视时间”。经过时间也会按指令实例分别计时。

执行本指令的过程中, 对属于指定轴组的轴执行了 MC_Reset (轴错误重置) 指令时, 也会分别执行两个指令。



错误代码

指令中发生的异常请参考 □ 「A-1 错误代码一览 (P.A-2)」。

5

共通指令

下面介绍轴和轴组共通的指令。

MC_SetCamTableProperty	5-2
MC_SaveCamTable	5-8
MC_Write	5-13
MC_GenerateCamTable	5-18
MC_WriteAxisParameter	5-45
MC_ReadAxisParameter	5-58

MC_SetCamTableProperty

更新输入输出参数中指定的凸轮表的终点索引。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_SetCamTableProperty	凸轮表属性更新	FB		<pre>MC_SetCamTableProperty_instance (CamTable := 《参数》, Execute := 《参数》, Done => 《参数》, EndPointIndex => 《参数》, MaxDataNumber => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、 FALSE	FALSE	上升时开始指令。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、 FALSE	指令执行完成时，变为 TRUE。
EndPointIndex	终点索引	UINT	正数或“0”	输出凸轮表的终点索引。
MaxDataNumber	最大凸轮数据数	UINT	正数	输出最大凸轮数据数。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、 FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	凸轮表的重写完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	发生要中止本指令的原因时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
CamTable	凸轮表	ARRAY[0..N] OF _sMC_CAM_REF	—	将凸轮数据结构体 _sMC_CAM_REF 型的排列变量指定为凸轮表。*1

*1. 排列要素 [N] 通过 Sysmac Studio 自动设定。指定在 Sysmac Studio 的凸轮编辑器中创建的凸轮数据变量。

功能说明

- 本指令可更新输入输出参数中指定的凸轮表的终点索引。
- 终点是指从起点开始搜索凸轮表时, 配置在相位首次为“0”的凸轮数据的前一个数据。
- 将终点的排列编号输出到 EndPointIndex (终点索引)。
- 检测到相位“0”后, 后面将作为无效凸轮数据处理, 因此将忽略相位 / 位移的值。
- 最大凸轮数据数表示凸轮数据结构体排列的最大要素数。
最大凸轮数据数为通过 Sysmac Studio 对结构体排列进行声明时指定的值。
- 通过用户程序变更了凸轮数据的终点索引时, 需要更新终点。此时, 请使用本指令, 更新有效凸轮数据数。

凸轮表请参考 □ □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。



使用注意事项

- 正在搜索凸轮表时，若在找到相位“0”前检测到相位不是按升序排列，将发生异常。
- 最大凸轮数据数不可通过用户程序变更。
- 变更凸轮数据的终点或重写凸轮数据的值后，请执行本指令。
变更了终点索引或相位不是升序时，凸轮动作或 MC_CamIn（凸轮动作开始）指令的 EndOfProfile（凸轮周期完成）将无法执行预期动作。
- 在本指令的处理过程中变更了凸轮表数据时，将无法正确执行更新处理。要通过程序变更凸轮数据时，请在本指令完成后再执行。
- 凸轮数据变量为全局变量，可从多个任务参照或变更凸轮数据变量的值。从多个任务变更凸轮数据变量的值时，编写的程序应确保多个任务的重写处理不会发生冲突。
- 使用“全局变量的任务间排他控制”功能对凸轮数据变量进行排他控制时，请勿用排他对象的凸轮数据变量启动运动指令。将发生轻度故障等级的异常“凸轮表指定不正确（事件代码：5439 Hex）”。

指令的详情

例如，以下凸轮表中，EndPointIndex（终点索引）为“999”、“MaxDataNumber（最大凸轮数据数）”为“5000”。

凸轮数据结构体排列

	相位	位移	
MyCam1 [0]	0	0	起点
.	.	.	
.	.	.	有效数据
.	.	.	
MyCam1 [997]	359.8	2	终点
MyCam1 [998]	359.9	1	
MyCam1 [999]	360.0	0	无效数据
MyCam1 [1000]	0	0	
.	.	.	无效数据
.	.	.	
MyCam1 [4999]	0	0	

最大数据数：5000

以下表示凸轮数据的重写和 EndPointIndex（终点索引）的关系。

若在 MyCam1[1000] 之后的相位为“0”的凸轮表中启动本指令，EndPointIndex（终点索引）将变为“999”。
若将 MyCam1[997] 的相位重写为“0”后启动本指令，EndPointIndex（终点索引）将变为“996”。

凸轮数据结构体排列	相位	位移	相位	位移	
MyCam1 [0]	0	0	0	0	
.	
.	
.	
MyCam1 [995]	359.6	5	359.6	5	
MyCam1 [996]	359.7	4	360.0	0	终点
MyCam1 [997]	359.8	2	0	0	无效数据
MyCam1 [998]	359.9	1	359.9	1	
MyCam1 [999]	360.0	0	360.0	0	
MyCam1 [1000]	0	0	0	0	
.	
.	
.	
MyCam1 [4999]	0	0	0	0	

此外，若将 MyCam1[1000] 到 MyCam1[4997] 之间的相位重写为“0”以外的数值后启动本指令，EndPointIndex（终点索引）将变为“4997”。

凸轮数据结构体排列	相位	位移	相位	位移	
MyCam1 [0]	0	0	0	0	
.	
.	
.	
MyCam1 [998]	359.9	1	100.3	20.3	
MyCam1 [999]	360.0	0	100.4	20.4	终点
MyCam1 [1000]	0	0	100.5	20.5	无效数据
.	
.	
.	
MyCam1 [4996]	0	0	359.99	0.01	
MyCam1 [4997]	0	0	360.00	0.0	终点
MyCam1 [4998]	0	0	0	0	无效数据
MyCam1 [4999]	0	0	0	0	

将 MyCam1[1000] 到 MyCam1[4997] 之间的相位重写为“0”以外，但未启动本指令时，凸轮动作将在 MyCam1[0] 到 MyCam1[999] 之间执行。

凸轮表会发生重写，但 EndPointIndex（终点索引）无变化。

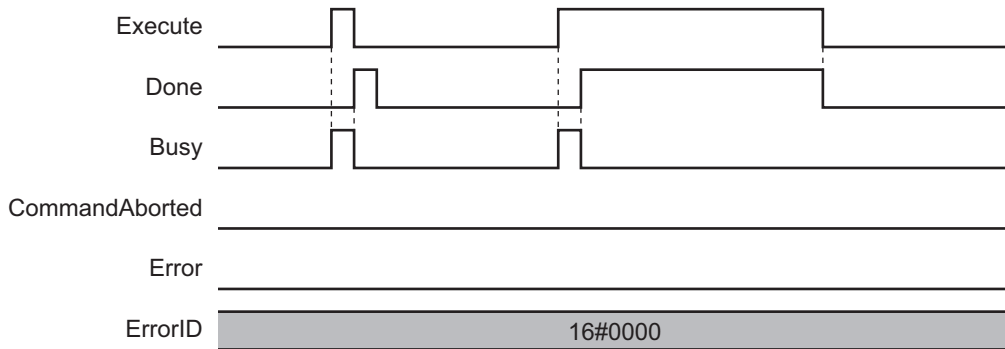
凸轮表的数据结构详情请参考 □ 「MC_CamIn(P.3-161)」。

时序图

下面表示本指令的两种启动方法。无论是哪种启动方法，动作均表示正常结束。

第 1 次启动表示 Execute（启动）设为 TRUE，完成前 Execute（启动）变为 False 的情况。

第 2 次启动表示保持 Execute（启动）的情况。



运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

● 其他指令执行过程中本指令的启动

与本指令使用相同 CamTable（凸轮表）的 MC_SaveCamTable（凸轮表保存）指令、MC_SetCamTableProperty（凸轮表属性更新）指令、MC_GenerateCamTable（凸轮表生成）指令已经启动时，无法多重启动本指令。

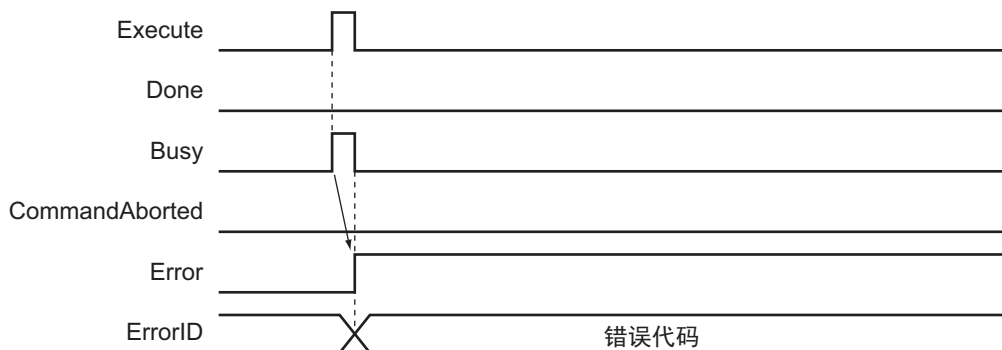
● 本指令执行过程中其他指令的启动

本指令执行过程中，无法多重启动使用相同 CamTable（凸轮表）的 MC_SaveCamTable（凸轮表保存）指令、MC_SetCamTableProperty（凸轮表属性更新）指令、MC_GenerateCamTable（凸轮表生成）指令。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。



● 错误代码

本指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

MC_SaveCamTable

将输入参数中指定的凸轮表保存至非易失性存储器中。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_SaveCamTable	凸轮表保存	FB		<pre>MC_SaveCamTable_instance (CamTable := 《参数》, Execute := 《参数》, Done => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、 FALSE	FALSE	上升时开始指令。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、 FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、 FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	凸轮表的保存完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	发生要中止本指令的原因时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
CamTable	凸轮表	ARRAY[0..N] OF _sMC_CAM_REF	—	将凸轮数据结构体 _sMC_CAM_REF 型的排列变量指定为凸轮表。 ^{*1}

*1. 排列要素 [N] 通过 Sysmac Studio 自动设定。指定在 Sysmac Studio 的凸轮编辑器中创建的凸轮数据变量。

功能说明

- 本指令可将输入输出变量中指定的凸轮表保存至非易失性存储器中。
- 正在执行凸轮表的保存处理时，运动控制系统变量 `_MC_COM.Status.CamTableBusy`（凸轮表文件保存执行中 / 待机中）将变为 TRUE。

凸轮表的数据规格请参考 □「MC_CamIn(P.3-161)」。



使用注意事项

- 重写凸轮数据后，请在将控制器的电源设为“OFF”前，通过本指令进行保存。若不保存直接将控制器电源设为“OFF”，重写的数据将消失。
- 在本指令的处理过程中变更了凸轮表数据时，将无法正确执行更新处理。通过程序变更凸轮表时，请勿在执行本指令的过程中向凸轮表执行写入。
- 本指令的处理时间会明显比其他指令长。此外，受 CPU 单元的运算负载量影响，处理时间可能会大幅变化。因此，以本指令的完成为触发启动下一指令时，请注意下一指令的启动时间。
- 本指令执行过程中请勿将控制器的电源设为“OFF”。若设为“OFF”，将无法正确保存数据。此外，非易失性存储器中保存的凸轮表可能损坏。
- 在本指令处理过程中，无法使用凸轮数据的下载及上传、在线开始操作、在线编辑、数据追踪开始等操作。
此外，若在凸轮数据的上传和下载、在线编辑等过程中执行本指令，将发生凸轮表保存指令不可执行（错误代码：743CHex）。请执行重试处理，直至凸轮数据正确保存。
同时，“下载”和“上传”时使用 Sysmac Studio 的“同步”功能。
- 非易失性存储器的写入次数有上限。若频繁执行本指令，非易失性存储器的寿命会变短，因此请以最小限度使用本指令。
- 凸轮数据变量为全局变量，可从多个任务参照或变更凸轮数据变量的值。从多个任务变更凸轮数据变量的值时，编写的程序应确保多个任务的重写处理不会发生冲突。
- 使用“全局变量的任务间排他控制”功能对凸轮数据变量进行排他控制时，请勿用排他对象的凸轮数据变量启动运动指令。将发生轻度故障等级的异常“凸轮表指定不正确（事件代码：5439 Hex）”。
- 正在反映在线编辑(*)时，请勿执行本指令。若执行，可能无法正确反映在线编辑的变更。
* 在 CPU 单元版本 1.04 以上和 Sysmac Studio Ver.1.05 以上的组合中，在反映确认窗口中选择 [是] 到反映确认窗口关闭为止的时间是指关闭内置非易失性存储器备份中窗口为止的时间。

● 与 CPU 单元动作模式的关系

正在通过本指令保存凸轮数据时，CPU 单元的动作模式切换为程序模式后，继续进行凸轮数据的保存处理。

要在程序模式中确认是否正在执行凸轮数据保存处理时，请将 Sysmac Studio 在线连接，对运动控制系统变量的 `_MC_COM.Status.CamTableBusy` 进行监视。

● 通过在线编辑删除指令时

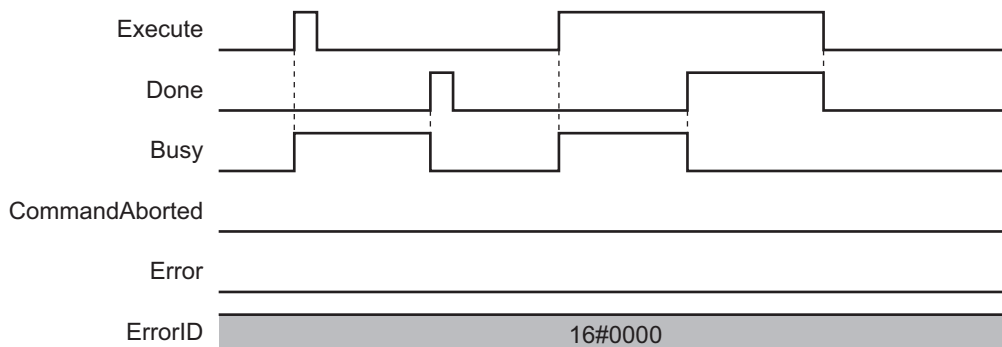
正在通过本指令保存凸轮数据时，即使通过在线编辑删除本指令，也会继续执行凸轮数据保存处理。

时序图

下面表示本指令的两种启动方法。无论是哪种启动方法，动作均表示正常结束。

第 1 次启动表示 Execute（启动）设为 TRUE，完成前 Execute（启动）变为 False 的情况。

第 2 次启动表示保持 Execute（启动）的情况。



运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。

运动指令多重启动

运动指令多重启动的详情请参考 [□](#) 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

● 其他指令执行过程中本指令的启动

与本指令使用相同 CamTable（凸轮表）的 MC_SetCamTableProperty（凸轮表属性更新）指令已经启动时，无法多重启动本指令。

此外，正在执行 MC_SaveCamTable（凸轮表保存）时，无法多重启动本指令。

● 本指令执行过程中其他指令的启动

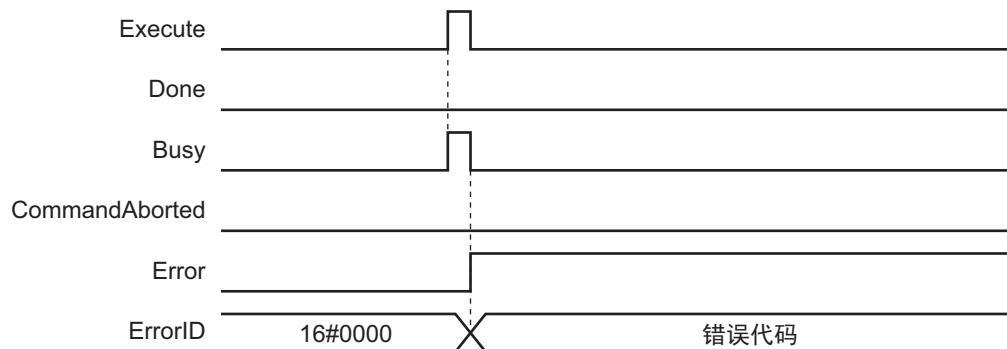
执行本指令的过程中，不可多重启动使用同一 CamTable（凸轮表）的其他指令的 MC_SetCamTableProperty（凸轮表属性更新）指令。

此外，正在执行本指令时，无法多重启动 MC_SaveCamTable（凸轮表保存）指令。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。



● 错误代码

本指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

MC_Write

重写 MC 功能模块运动控制参数设定的一部分。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_Write	MC 设定写入	FB		<pre> MC_Write_instance (Target := 《参数》, SettingValue := 《参数》, Execute := 《参数》, ParameterNumber := 《参数》, Done => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》); </pre>



使用注意事项

通过本指令重写的值不会保存到 CPU 单元的非易失性存储器中。因此，控制器的电源“OFF”及开始下载和 MC 功能模块的重新启动处理时，重写的值将消失，恢复为 Sysmac Studio 中设定的值。

要保存到非易失性存储器时，请使用 Sysmac Studio 传送参数。



参考

“下载”时使用 Sysmac Studio 的“同步”功能。

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、FALSE	FALSE	上升时开始指令。
Parameter Number	参数编号	_eMC_PARA METER_NUM MBER	0:_mcChkVel 1:_mcChkAcc 2:_mcChkDec 3:_mcPosiChkTrq 4:_mcNegaChkTrq 5:_mcFELmt 6:_mcChkFELmt 7:_mcSwLmtMode 8:_mcPosiSwLmt 9:_mcNegaSwLmt 10:_mcInPosTime 11:_mcInPosRange*1 12:_mcStartVel*2	0*3	指定写入目标的参数。 0: 速度警告值 / 插补速度警告值 1: 加速度警告值 / 插补加速度警告值 2: 减速度警告值 / 插补减速度警告值 3: 正方向扭矩警告值 4: 负方向扭矩警告值 5: 位置偏差超限值 6: 位置偏差警告值 7: 软件限位功能 8: 正方向软件限位 9: 负方向软件限位 10: 位置检查时间 11: 位置宽度 12: 启动速度

*1. 可在 CPU 单元版本 1.01 以上和 Sysmac Studio Ver.1.02 以上的组合中指定。

*2. 可在 CPU 单元版本 1.10 以上和 Sysmac Studio Ver.1.12 以上的组合中指定。

*3. 如果是有效范围为列举型的变量初始值，实际不是数值，而是列举值。

● 参数的数据类型和有效范围

参数	数据类型	有效范围	备注
0 : 速度警告值 / 插补速度警告值	UINT	0 ~ 100	单位为 [%]。
1 : 加速度警告值 / 插补加速度警告值	UINT	0 ~ 100	单位为 [%]。
2 : 减速度警告值 / 插补减速度警告值	UINT	0 ~ 100	单位为 [%]。
3 : 正方向扭矩警告值	UINT	0 ~ 1000	单位为 [%]。
4 : 负方向扭矩警告值	UINT	0 ~ 1000	单位为 [%]。
5 : 位置偏差超限值	LREAL	正数	单位为 [指令单位]。*1
6 : 位置偏差警告值	LREAL	正数	设为小于位置偏差超限值。 单位为 [指令单位]。*1
7 : 软件限位功能	_eMC_SWLMT_MODE	0:_mcNonSwLmt 1:_mcCmdDeceleration Stop 2:_mcCmdImmediateStop 3:_mcActDeceleration Stop 4:_mcActImmediateStop	0: 软件限位功能的无效 1: 对指令位置有效, 减速停止 2: 对指令位置有效, 立即停止 (滞留脉冲停止) 3: 对反馈位置有效, 减速停止 4: 对反馈位置有效, 立即停止 (滞留脉冲停止)
8 : 正方向软件限位	LREAL	负数、正数、“0”	单位为 [指令单位]。*1
9 : 负方向软件限位	LREAL	负数、正数、“0”	单位为 [指令单位]。*1
10 : 位置检查时间	UINT	0 ~ 10000	单位为 [ms]。
11 : 位置宽度	LREAL	正数、“0”	单位为 [指令单位]。*1
12 : 启动速度	LREAL	正数、“0”	单位为 [指令单位 /s]。*1

*1. 指令单位请参考 □□《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》的“单位转换设定”。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、FALSE	指令执行完成时, 变为 TRUE。
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时, 输出错误代码。16#0000 表示正常。

* □□ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	指令完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	其他指令上发生异常, 中止了本指令时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Target	写入对象	_sAXIS_REF 或 _sGROUP_REF	—	指定作为写入对象的轴或轴组。*1
SettingValue	设定值	取决于指定变量的类型 *2	—	指定要写入的值。 有效范围遵照 ParameterNumber (参数编号) 中指定的运动控制参数。初始值为“0”。

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称 (默认“MC_Axis****”)或轴组基本设定画面中创建的用户定义变量的轴组变量名称 (默认“MC_Group****”)。任何情况下, 系统定义变量的变量名称 (_MC_AX[*], _MC1_AX[*], _MC2_AX[*], _MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*]) 均可使用。

*2. 关于变量的数据类型, 请参考 □□「参数的数据类型和有效范围 (P.5-14)」。

● 输入输出变量的反映时序

输入输出变量	写入时序
SettingValue	Done 的上升时

功能说明

- 在 Execute（启动）的上升沿，在 Target（写入对象）及 ParameterNumber（参数编号）中指定的运动控制系统变量中写入 SettingValue（设定值）。
- 输入变量中指定的参数将在轴变量 Status.Standstill（停止中）为 TRUE 时，或轴组变量 Status.Standby（停止中）为 TRUE 并开始动作时，以及中断指定下的多重启动时应用。
因此，在重新执行及中断指定以外的多重启动时，若要继续动作，将不应用。



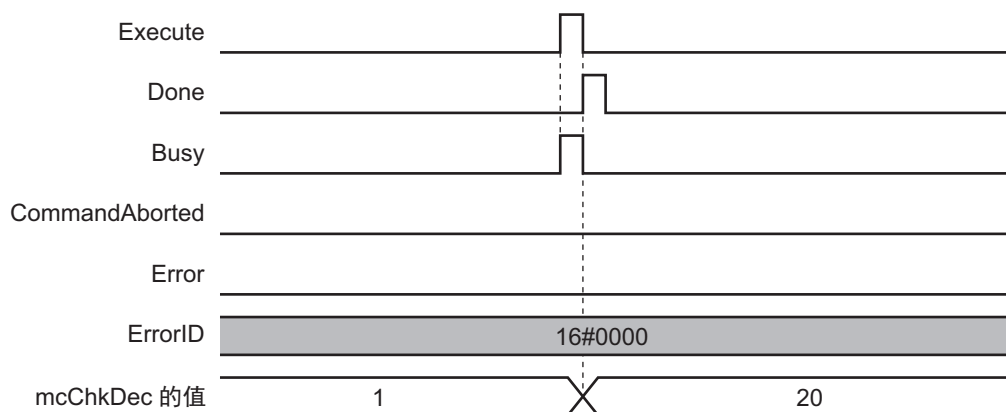
使用注意事项

通过本指令重写的值不会保存到 CPU 单元的非易失性存储器中。因此，控制器的电源“OFF”及开始下载和 MC 功能模块的重新启动处理时，重写的值将消失，恢复为 Sysmac Studio 中设定的值。

要保存到非易失性存储器时，请使用 Sysmac Studio 传送参数。

时序图

在轴参数设定的 _mcChkDec（减速度警告值）中写入输入“20”时，时序图如下。



运动指令重新执行

本指令的 Busy（执行中）为 TRUE 时，若有相同实例的 Execute（启动）上升，将重新执行指令。此时，将重写为后面的 Execute（启动）上升时的 Target（写入对象）、ParameterNumber（参数编号）和 SettingValue（设定值）。

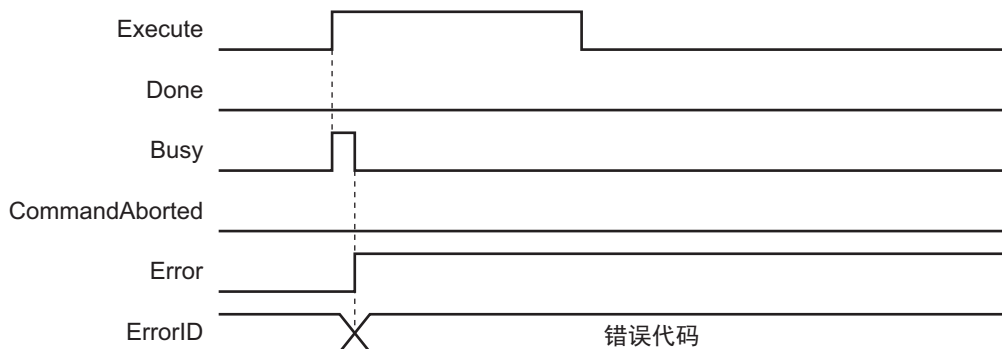
运动指令多重启动

运动指令多重启动的详情请参考 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，不写入参数。将保持执行指令前的值。参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

● 发生异常时的时序图



● 错误代码

指令中发生的异常请参考 [□「A-1 错误代码一览 \(P.A-2\)」](#)。

MC_GenerateCamTable

根据输入输出参数中指定的凸轮属性和凸轮节点生成凸轮表。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_GenerateCamTable	凸轮表生成	FB		<pre>MC_GenerateCamTable_instance (CamTable := 《参数》, CamProperty := 《参数》, CamNodes := 《参数》, Execute := 《参数》, Done => 《参数》, EndPointIndex => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》, ErrorParameterCode => 《参数》, ErrorNodePointIndex => 《参数》);</pre>



使用注意事项

本指令中指定的凸轮表需要事先用 Sysmac Studio 的凸轮编辑器创建，然后下载到 CPU 单元中。“下载”时使用 Sysmac Studio 的“同步”功能。



版本相关信息

本指令可在 CPU 单元版本 1.08 以上和 Sysmac Studio Ver.1.09 以上的组合中使用。

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、 FALSE	FALSE	上升时开始指令。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、 FALSE	指令执行完成时，变为 TRUE。
EndPointIndex	终点索引	UINT	正数或“0”	指令执行完成时，输出凸轮表的终点索引。

输出变量	名称	数据类型	有效范围	内容
Busy	执行中	BOOL	TRUE、FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。
ErrorParameterCode	参数详细代码	WORD	*	有些错误代码存在附属信息。存在时，输出发生异常的参数之详细代码。
ErrorNodePointIndex	节点要素编号	UINT	*	有些错误代码存在附属信息。存在时，输出发生异常的节点要素编号。

* □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	凸轮表的生成完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	发生要中止本指令的原因时	<ul style="list-style-type: none"> Execute 为 TRUE 时，与 Execute 的 FALSE 同时 Execute 为 FALSE 时，1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
CamTable	凸轮表	ARRAY[0..N] OF _sMC_CAM_REF	—	将凸轮数据结构体 _sMC_CAM_REF 型的排列变量指定为凸轮表。 ^{*1} 指定在 Sysmac Studio 的凸轮编辑器中创建的凸轮数据变量。
CamProperty	凸轮属性	_sMC_CAM_PROPERTY	—	指定凸轮属性结构体 _sMC_CAM_PROPERTY 型的变量。 请指定 _sMC_CAM_PROPERTY 型用户定义变量或 Sysmac studio 中创建的凸轮属性变量。
CamNodes	凸轮节点	ARRAY[0..N] OF _sMC_CAM_NODE	—	指定凸轮节点结构体 _sMC_CAM_NODE 型的排列变量。 请指定 _sMC_CAM_NODE 型用户定义变量 ^{*2} 或 Sysmac studio 中创建的凸轮节点变量。

*1. 排列要素 [N] 通过 Sysmac Studio 自动设定。

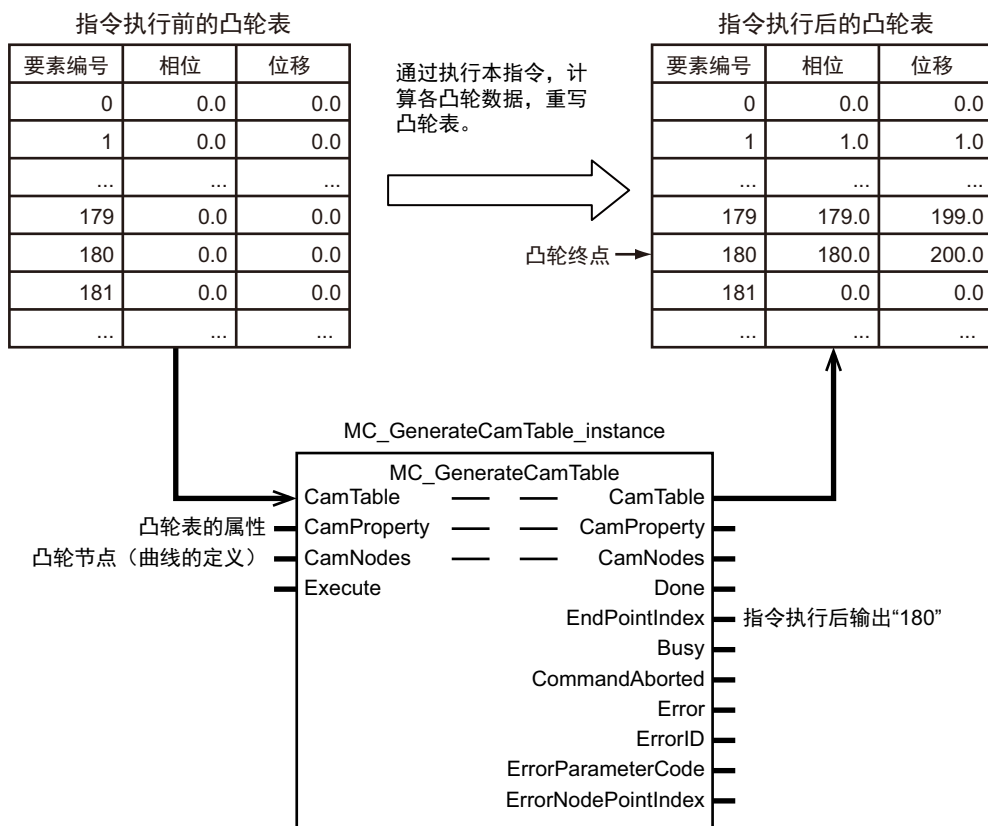
*2. 用户定义变量时，应创建为排列变量的要素开始编号为 0，排列要素 [N] 小于 358 以下。

功能说明

- 本指令可在 Execute（启动）的上升沿，根据输入输出参数中指定的 CamProperty（凸轮属性）和 CamNodes（凸轮节点）的值，计算凸轮数据。
- 将输入输出变量中指定的 CamTable（凸轮表）重写为计算得到的凸轮数据值。
- CamProperty（凸轮属性）和 CamNodes（凸轮节点）的各项目与 Sysmac Studio 的凸轮编辑器的各设定项目一致。
- 凸轮表重写完成后，更新凸轮表的终点索引，将凸轮终点的要素编号输出到 EndPointIndex（终点索引）中。

本指令完成后，不需要执行 MC_SetCamTableProperty（凸轮表属性更新）指令。

- 正在执行凸轮表的生成处理时，运动控制系统变量的 MC 共通变量中的“凸轮表生成执行中（_MC_COM.Status.GenerateCamBusy）”将变为 TRUE。





使用注意事项

- 指定凸轮表的最大凸轮数据数请设定为大于本指令生成的凸轮表的凸轮数据数。
- 在Sysmac Studio的凸轮编辑器中，曲线的形状可指定为“自由曲线”，但本指令中不可指定“自由曲线”。
- 本指令的处理时间会明显比其他指令长。此外，受 CPU 单元的运算负载量影响，处理时间可能会大幅变化。因此，以本指令的完成为触发启动下一指令时，请注意下一指令的启动时间。
- 即使 Sysmac Studio 的凸轮编辑器中的设定值和本指令中的设定项目相同，因内部运算处理的不同，生成的凸轮数据值可能会不同。
- 凸轮数据变量为全局变量，可从多个任务参照或变更凸轮数据变量的值。从多个任务变更凸轮数据变量的值时，编写的程序应确保多个任务的重写处理不会发生冲突。
- 使用“全局变量的任务间排他控制”功能对凸轮数据变量进行排他控制时，请勿用排他对象的凸轮数据变量启动运动指令。将发生轻度故障等级的异常“凸轮表指定不正确（事件代码：5439 Hex）”。
- 将本指令的输入输出变量 CamProperty（凸轮属性）及 CamNodes（凸轮节点）中指定的变量作为用户定义变量创建时，变量保持属性的初始值为“不保持”。变更变量的值并切换为程序模式后或重新接通电源再次使用时，请将变量的保持属性设定为“保持”。Sysmac studio 上创建的凸轮属性变量及凸轮节点变量的保持属性固定为“保持”。
- 通过本指令生成的凸轮数据变量不会保存到 CPU 单元的非易失性存储器中。要保存到非易失性存储器时，请执行 MC_SaveCamTable（凸轮表保存）指令。
- 正在执行本指令时，请勿变更 CamNodes（凸轮节点）中指定的排列变量的值。本指令可能发生意外动作。
- 若 CamNodes（凸轮节点）的排列大小过大，指令执行时间将变长，可能超出任务周期。此时，将发生任务周期超限（错误代码：6001 Hex），因此请调整排列大小或变更任务周期。
- 正在执行本指令时，即使从运行模式切换为程序模式，仍将继续执行凸轮表的生成处理。若在该状态下再次切换为运行模式，并执行以凸轮表为对象的指令，将发生轻度故障等级的异常“运动指令不可多重启动（错误代码：543C Hex）”。



参考

- 凸轮表的最大凸轮数据数可使用 Sysmac Studio，通过全局变量的数据类型确认，或在用户程序中执行 SizeOfAry（获取排列要素数）指令来确认。
- 通过本指令生成的凸轮表的凸轮数据数可按以下公式计算。

$$T_{cd} = \sum_{k=0}^{n-1} \frac{M_k - M_{k-1}}{P_k} + 1$$

T_{cd} : 凸轮表的凸轮数据数

k : 凸轮节点变量的排列要素编号

M_k : 凸轮节点变量的排列要素编号 k 的 Phase（主轴相位）
（但是， M_{-1} 的 Phase（主轴相位）为“0”）

P_k : 凸轮节点变量的排列要素编号 k 的 PhasePitch（相位间隔宽度）

n : 节点点数

注） Phase（主轴相位）/ PhasePitch（相位间隔宽度）无法除尽时，
请将小数第 1 位取整。

- 通过本指令生成的凸轮表的凸轮数据数请设定为 65535 以下。
- 生成的凸轮表的凸轮数据数可通过 EndPointIndex（终点索引）的值确认。可在显示器上显示 EndPointIndex（终点索引）的值，用于改善凸轮表的分辨率或接点的追加等调整。
- 凸轮表的凸轮数据详情请参考 MC_CamIn（凸轮动作开始）指令。
- 终点索引的详情请参考 MC_SetCamTableProperty（凸轮表属性更新）指令。
- CamProperty（凸轮属性）和 CamNodes（凸轮节点）的各设定项目请参考 □《Sysmac Studio Version 1 操作手册（SBCE-CN5-362K 以后）》。
- 曲线的形状、连接速度和连接加速度的关系请参考 □《Sysmac Studio Version 1 操作手册（SBCE-CN5-362K 以后）》。

● 与 CPU 单元动作模式的关系

正在通过本指令生成凸轮表时，CPU 单元的动作模式切换为程序模式后，仍继续进行凸轮表的生成处理。

● 通过在线编辑删除指令时

正在执行本指令时，即使通过在线编辑删除本指令，也会继续执行凸轮表生成处理。

● 执行 MC 调试时

正在通过本指令生成凸轮表时，若执行 MC 调试，本指令的输出变量“CommandAborted（执行中断）”将变为 TRUE，继续执行凸轮表的生成处理。

指令的详情

下面介绍指令的详情。

● 凸轮属性结构体（_sMC_CAM_PROPERTY 型）

凸轮属性结构体（_sMC_CAM_PROPERTY 型）是指指定属性所需的结构体数据类型。各成员相当于 Sysmac Studio 的凸轮编辑器中设定的部分“凸轮表的属性”项目。

本指令中，作为输入输出变量的 CamProperty（凸轮属性）中指定的变量使用。

下面介绍凸轮属性结构体的成员。

变量	名称	数据类型	有效范围	内容
InitVel	初始速度	REAL	负数 ^{*1} 、正数 ^{*1} 、“0”	指定开始节点（相位 =“0”、位移 =“0”）时的动作开始时速度。 开始节点后面的凸轮节点上指定的曲线形状为 3 次曲线或 5 次曲线时，初始速度为有效。 单位为 [指令单位 /s]。
InitAcc	初始加速度	REAL	负数 ^{*1} 、正数 ^{*1} 、“0”	指定开始节点（相位 =“0”、位移 =“0”）时的动作开始时加速度。 开始节点后面的凸轮节点上指定的曲线形状为 5 次曲线时，初始加速度为有效。 单位为 [指令单位 /s ²]。
CycleTime	周期时间	REAL	正数 ^{*1}	指定 1 个凸轮动作周期的时间。 单位为 [s]。

*1. 绝对值请指定为 0.001 以上。小数点后第 4 位采用四舍五入。

● 凸轮节点结构体（_sMC_CAM_NODE 型）

凸轮节点结构体（_sMC_CAM_NODE 型）是指指定曲线的定义所需的结构体数据类型。各成员相当于 Sysmac Studio 的凸轮编辑器中设定的“凸轮节点”项目。

	主轴	从轴	力△曲线	接统速度	接统加速度	位相间隔幅
开始节点	0.000	0.000				
CamNodes[0]	1.000	1.000	5次曲线	<input checked="" type="checkbox"/> 100.000	<input checked="" type="checkbox"/> 10.000	0.010
CamNodes[1]	2.000	2.000	3次曲线	<input checked="" type="checkbox"/> 100.000	<input type="checkbox"/> 1176.000	0.010

本指令中，作为输入输出变量的 CamNodes（凸轮节点）中指定的排列变量使用。

凸轮节点的排列变量要素称为“节点”。排列要素的要素数请设定为大于所设定的节点数。

凸轮轮廓曲线开始点（相位 =“0”、位移 =“0”）的节点称为“开始节点”。在排列变量中按要素编号的顺序指定除开始节点外的节点。

若 Phase（主轴相位）指定为 0.001 以上的正数，将变为有效节点。若指定为“0”，后面的节点将变为无效。但是，若将要素编号为“0”的 Phase（主轴相位）指定为“0”，将发生异常。

下面为有效节点数 5、排列变量的要素数 10 时的示例。

要素编号	Phase（主轴相位）
0	正数
...	...
4	正数
5	0
6	任意
...	...
9	任意

请按排列编号的升序指定，使 Phase（主轴相位）的值为升序。（上次值 < 本次值）

下面介绍凸轮结构体的成员。

变量	名称	数据类型	有效范围	内容
Phase	主轴相位	REAL	正数 ^{*1} *2、“0”	指定在节点下主轴相位的值。 单位为 [指令单位]。
Distance	从轴位移	REAL	负数 ^{*1} *2、正数 ^{*1} *2、“0”	指定在节点下从轴位移的值。 单位为 [指令单位]。
Curve	曲线形状	_eMC_CAM_CURVE	0: _mcConstantLine 1: _mcStraightLine 2: _mcParabolic 3: _mcModifiedConstantVel 4: _mcModifiedTrapezoid 5: _mcModifiedSine 6: _mcCycloidal 7: _mcTrapeclloid 8: _mcReverseTrapeclloid 9: _mcSimpleHarmonic 10: _mcDoubleHarmonic 11: _mcReverseDoubleHarmonic 12: _mcNC2Curve 13: _mcPolynomic3 14: _mcPolynomic5	指定到节点为止的凸轮曲线形状。 0: 直线（保持位移） ^{*3} 1: 直线 2: 等加速度 3: 变形等速度 4: 变形梯形 5: 变形正弦 6: 摆线 7: Trapeclloid 8: 反 Trapeclloid 9: 单弦 10: 复弦 11: 反复弦 12: NC2 曲线 13: 3 次曲线 14: 5 次曲线
ConnectingVelEnable	连接速度有效	BOOL	TRUE、FALSE	若设为 TRUE，指定的曲线形状为 3 次曲线或 5 次曲线时，指定的连接速度将变为有效。 若设为 FALSE，则连接速度变为无效。
ConnectingVel	连接速度	REAL	负数 ^{*1} 、正数 ^{*1} 、“0”	指定的曲线形状为 3 次曲线或 5 次曲线时，可指定与后续曲线的连接部的速度。 要平滑地连接曲线和曲线时指定。 单位为 [指令单位 /s]。
ConnectingAccEnable	连接加速度有效	BOOL	TRUE、FALSE	若设为 TRUE，指定的曲线形状为 5 次曲线时，指定的连接加速度将变为有效。 若设为 FALSE，则连接加速度变为无效。
ConnectingAcc	连接加速度	REAL	负数 ^{*1} 、正数 ^{*1} 、“0”	指定的曲线形状为 5 次曲线时，可指定与后续曲线的连接部的加速度。 要平滑地连接曲线和曲线时指定。 单位为 [指令单位 /s ²]。
PhasePitch	相位间隔宽度	REAL	正数 ^{*1}	按指定的间隔宽度分割节点间的相位。 ^{*4} 单位为 [指令单位]。

*1. 绝对值请指定为 0.001 以上。小数点后第 4 位采用四舍五入。

*2. Phase（主轴相位）和 Distance（从轴位移）的有效位数为 7 位。若输入 8 位以上，第 8 位开始将舍去。舍去后，若与其他 Phase（主轴相位）的值相同，将发生“凸轮节点主轴相位非升序（错误代码：5740Hex）”。请在有效位数 7 位的范围内按升序输入值。

*3. 若指定为直线（保持位移），Distance（从轴位移）将变为无效，以与前一个节点中指定的值相同的值进行处理。排列的要素编号为 0 时，若指定为直线（保持位移），Distance（从轴位移）将作为 0 进行处理。

*4. 按节点生成的凸轮数据总计请指定为 65535 以下。

● 凸轮表的生成示例

下面介绍用本指令生成凸轮表的示例。

CamNodes（凸轮节点）中指定的排列要素数为 4、Curve（曲线形状）为“_mcStraightLine（直线）”。ConnectingVel（连接速度）、ConnectingAcc（连接加速度）及 CamProperty（凸轮属性）的 InitVel（初始速度）和 InitAcc（初始加速度）将变为无效，省略记载。Phase（主轴相位）和 Distance（从轴位移）的值如下表所示。

CamNodes（凸轮节点） 的排列变量要素编号	Phase （主轴相位）	Distance （从轴位移）	Curve （曲线形状）	PhasePitch （相位间隔宽度）
0	180.000	180.000	_mcStraightLine	0.100
1	360.000	0.000	_mcStraightLine	0.100
2	0.00	-	-	-
3	-	-	-	-

事先通过 Sysmac studio 的凸轮编辑器创建的凸轮数据变量的要素数为 4000，相位和位移的值全部为“未定义值”。

执行本指令前，凸轮表的凸轮数据变量如下表所示。

要素编号	Phase（相位）	Distance（位移）
0	（未定义值）	（未定义值）
1	（未定义值）	（未定义值）
...		
1799	（未定义值）	（未定义值）
1800	（未定义值）	（未定义值）
1801	（未定义值）	（未定义值）
...		
3599	（未定义值）	（未定义值）
3600	（未定义值）	（未定义值）
3601	（未定义值）	（未定义值）
...		
3999	（未定义值）	（未定义值）

下面依次介绍因执行本指令而发生的凸轮数据变量的变更过程。变更处以背景填充的方式表示。

若执行本指令，凸轮数据变量的要素编号“0”的相位和位移将重写为“0”。

要素编号	Phase（相位）	Distance（位移）
0	0.0	0.0
1	（未定义值）	（未定义值）
...		
1799	（未定义值）	（未定义值）
1800	（未定义值）	（未定义值）
1801	（未定义值）	（未定义值）
...		
3599	（未定义值）	（未定义值）
3600	（未定义值）	（未定义值）
3601	（未定义值）	（未定义值）
...		
3999	（未定义值）	（未定义值）

然后按照 CamNodes（凸轮节点）的要素编号“0”的指定值，计算从开始节点到节点为止的凸轮数据数及各凸轮数据的相位和位移。凸轮数据数为“1800”，因此将重写凸轮数据变量的要素编号“1”到“1800”为止的相位和位移。

要素编号	Phase（相位）	Distance（位移）
0	0.0	0.0
1	0.1	0.1
...		
1799	179.9	179.9
1800	180.0	180.0
1801	（未定义值）	（未定义值）
...		
3599	（未定义值）	（未定义值）
3600	（未定义值）	（未定义值）
3601	（未定义值）	（未定义值）
...		
3999	（未定义值）	（未定义值）

同样，按照 CamNodes（凸轮节点）的要素编号“1”的指定值，计算节点之间的凸轮数据数及各凸轮数据的相位和位移。凸轮数据数为“1800”，因此将重写凸轮数据变量的要素编号“1801”到“3600”为止的相位和位移。

要素编号	Phase（相位）	Distance（位移）
0	0.0	0.0
1	0.1	0.1
...		
1799	179.9	179.9
1800	180.0	180.0
1801	180.1	179.9
...		
3599	359.9	0.1
3600	360.0	0.0
3601	（未定义值）	（未定义值）
...		
3999	（未定义值）	（未定义值）

然后，由于 CamNodes（凸轮节点）的要素编号“2”为无效节点，所以结束计算。为使凸轮数据变量的要素编号“3601”以后的凸轮数据变为无效，将相位的值重写为“0”。

要素编号	Phase（相位）	Distance（位移）
0	0.0	0.0
1	0.1	0.1
...		
1799	179.9	179.9
1800	180.0	180.0
1801	180.1	179.9
...		
3599	359.9	0.1
3600	360.0	0.0
3601	0.0	（未定义值）
...		
3999	（未定义值）	（未定义值）

但是，在 Sysmac studio 的凸轮编辑器上创建的凸轮数据变量的要素数超过 3601 时，因为不存在无效凸轮数据，所以相位的值不会重写为“0”。

要素编号	Phase (相位)	Distance (位移)
0	0.0	0.0
1	0.1	0.1
...		
1799	179.9	179.9
1800	180.0	180.0
1801	180.1	179.9
...		
3599	359.9	0.1
3600	360.0	0.0

将最后重写的凸轮数据变量的要素编号作为凸轮表终点索引的值，输出到本指令的输出变量 EndPointIndex (终点索引) 中。本示例中输出为“3600”。

至此，本指令的执行完成。

● 发生凸轮表生成位移溢出时

通过本指令计算得到的凸轮数据的 Distance (位移) 值超出 REAL 型的有效范围时，将发生“凸轮表生成位移溢出 (错误代码: 5742 Hex)”。此时，相应的 Distance (位移) 值不会更新，并中止凸轮表的生成处理。

凸轮表生成位移溢出只在 Curve (曲线形状) 指定为 3 次曲线或 5 次曲线时可能发生。发生该异常时，请参考后面记载的 3 次曲线的位移计算方法或 5 次曲线的位移计算方法，修改输入输出变量 CamProperty (凸轮属性) 及 CamNodes (凸轮节点) 的值。

● 3 次曲线的位移计算方法

CamNodes (凸轮节点) 中指定的排列变量的要素编号为“m”。

要素编号为“m”的 Curve (曲线形状) 指定为 3 次曲线时，要素编号“m-1”的 Phase 为主轴相位的初始值 P_{init} 。同样，Distance 为从轴位移的初始值 d_{init} 。m=0 时， P_{init} 和 d_{init} 视为“0”进行计算。此外，要素编号“m”的 Phase 为主轴相位的终端值 P_{final} 。同样，要素编号“m”的 Distance 为从轴位移的终端值 d_{final} 。

若从 d_{init} 开始第 n 个凸轮数据的 Distance (位移) 为 $d(n)$ ，则 $d(n)$ 的计算式如下。

$0 \leq n < N$ 时

$$d(n) = d_{init} + (d_{final} - d_{init}) \cdot \sum_{i=1}^3 a_i \cdot \left(\frac{pitch}{(P_{final} - P_{init})} \cdot n \right)^i$$

n=N 时

$$d(n) = d_{final}$$

但是，

$(P_{final} - P_{phase}) / pitch$ 为整数时

$$N = \frac{P_{final} - P_{phase}}{pitch}$$

$(P_{final} - P_{phase}) / pitch$ 不是整数时

$$N = \text{floor}\left(\frac{P_{final} - P_{phase}}{\text{pitch}}\right) + 1$$

$$a_1 = \frac{V_{init} \cdot T}{(d_{final} - d_{init})}$$

$$a_2 = 3 - (2 \cdot V_{init} + V_{fin}) \cdot \frac{T}{(d_{final} - d_{init})}$$

$$a_3 = (V_{init} + V_{fin}) \cdot \frac{T}{(d_{final} - d_{init})} - 2$$

$$T = \text{CycleTime} \cdot \frac{(P_{final} - P_{init})}{P_{max}}$$

pitch: CamNodes (凸轮节点) 的要素编号“m”的 PhasePitch (相位间隔宽度)

CycleTime: CamProperty (凸轮属性) 的 CycleTime (周期时间)

P_{max} : CamNodes (凸轮节点) 的有效节点下 Phase (主轴相位) 的最大值

V_{init} : 凸轮轮廓曲线的初始速度 [指令单位 /s]

但是,

m=0 时

$V_{init} = \text{CamProperty}$ (凸轮属性) 的 InitVel (初始速度)

m ≥ 1 时, 且 CamNodes (凸轮节点) 的要素编号“m-1”的 Curve (曲线形状) 为直线 (_mcStraightLine) 时

$$V_{init} = \frac{(\text{要素编号}m-1\text{的}d_{final} - \text{要素编号}m-1\text{的}d_{init})}{(\text{要素编号}m-1\text{的}P_{final} - \text{要素编号}m-1\text{的}P_{init})} \cdot \frac{P_{max}}{\text{CycleTime}}$$

m ≥ 1 时, 且 CamNodes (凸轮节点) 的要素编号“m-1”的 Curve (曲线形状) 为 3 次曲线或 5 次曲线时

- CamNodes (凸轮节点) 的要素编号“m-1”的 ConnectingVelEnable (连接速度有效) 为 TRUE

$V_{init} = \text{CamNodes}$ (凸轮节点) 的要素编号“m-1”的 ConnectingVel (连接速度)

- CamNodes (凸轮节点) 的要素编号“m-1”的 ConnectingVelEnable (连接速度有效) 为 FALSE

$V_{init} = 0$

m ≥ 1, 且上述以外时

$V_{init} = 0$

V_{fin} : 凸轮曲线的终端速度 [指令单位 /s]

但是,

CamNodes (凸轮节点) 的要素编号“m”的 ConnectingVelEnable (连接速度有效) 为 TRUE 时

$V_{fin} = \text{CamNodes}$ (凸轮节点) 的要素编号“m”的 ConnectingVel (连接速度)

CamNodes（凸轮节点）的要素编号“m”的 ConnectingVelEnable（连接速度有效）为 FALSE 时

- CamNodes（凸轮节点）的要素编号“m+1”的 Curve（曲线形状）为直线（_mcStraightLine）

$$V_{fin} = \frac{(\text{要素编号}m+1\text{的}d_{final} - \text{要素编号}m+1\text{的}d_{init})}{(\text{要素编号}m+1\text{的}P_{final} - \text{要素编号}m+1\text{的}P_{init})} \cdot \frac{P_{max}}{CycleTime}$$

- CamNodes（凸轮节点）的要素编号“m+1”的 Curve（曲线形状）为直线（_mcStraightLine）以外

$$V_{fin} = 0$$

● 5 次曲线的位移计算方法

CamNodes（凸轮节点）中指定的排列变量的要素编号为“m”。

要素编号为“m”的 Curve（曲线形状）指定为 5 次曲线时，要素编号“m-1”的 Phase 为主轴相位的初始值 P_{init} 。同样，Distance 为从轴位移的初始值 d_{init} 。m=0 时， P_{init} 和 d_{init} 视为“0”进行计算。此外，

要素编号“m”的 Phase 为主轴相位的终端值 P_{final} 。同样，要素编号“m”的 Distance 为从轴位移的终端值 d_{final} 。

若从 d_{init} 开始第 n 个凸轮数据的 Distance（位移）为 $d(n)$ ，则 $d(n)$ 的计算式如下。

$0 \leq n < N$ 时

$$d(n) = d_{init} + (d_{final} - d_{init}) \cdot \sum_{i=1}^5 a_i \cdot \left(\frac{pitch}{(P_{final} - P_{init})} \cdot n \right)^i$$

n=N 时

$$d(n) = d_{final}$$

但是，

$(P_{final} - P_{phase}) / pitch$ 为整数时

$$N = \frac{P_{final} - P_{phase}}{pitch}$$

$(P_{final} - P_{phase}) / pitch$ 不是整数时

$$N = \text{floor}\left(\frac{P_{final} - P_{phase}}{pitch}\right) + 1$$

$$a_1 = \frac{V_{init} \cdot T}{(d_{final} - d_{init})}$$

$$a_2 = \frac{A_{init} \cdot T^2}{2 \cdot (d_{final} - d_{init})}$$

$$a_3 = \frac{-\left((3A_{init} - A_{fin}) \cdot \frac{T^2}{(d_{final} - d_{init})} + (8V_{fin} + 12V_{init}) \cdot \frac{T}{(d_{final} - d_{init})} - 20\right)}{2}$$

$$a_4 = \left((1.5A_{init} - A_{fin}) \cdot \frac{T^2}{(d_{final} - d_{init})} + (7V_{fin} + 8V_{init}) \cdot \frac{T}{(d_{final} - d_{init})} - 15 \right)$$

$$a_5 = \frac{-\left((A_{init} - A_{fin}) \cdot \frac{T^2}{(d_{final} - d_{init})} + (6V_{fin} + 6V_{init}) \cdot \frac{T}{(d_{final} - d_{init})} - 12 \right)}{2}$$

$$T = CycleTime \cdot \frac{(P_{final} - P_{init})}{P_{max}}$$

pitch: CamNodes (凸轮节点) 的要素编号“m”的 PhasePitch (相位间隔宽度)

CycleTime: CamProperty (凸轮属性) 的 CycleTime (周期时间)

P_{max} : CamNodes (凸轮节点) 的有效节点下 Phase (主轴相位) 的最大值

A_{init} : 凸轮轮廓曲线的初始加速度 [指令单位 /s²]

但是,

m=0 时

$A_{init} = CamProperty$ (凸轮属性) 的 InitAcc (初始加速度)

m ≥ 1 时, 且 CamNodes (凸轮节点) 的要素编号“m-1”的 Curve (曲线形状) 为 5 次曲线时

- CamNodes (凸轮节点) 的要素编号“m-1”的 ConnectingVelEnable (连接速度有效) 为 TRUE

$A_{init} = CamNodes$ (凸轮节点) 的要素编号“m-1”的 ConnectingAcc (连接加速度)

- CamNodes (凸轮节点) 的要素编号“m-1”的 ConnectingVelEnable (连接速度有效) 为 FALSE

$A_{init} = 0$

m ≥ 1 时, 且 CamNodes (凸轮节点) 的要素编号“m-1”的 Curve (曲线形状) 为 5 次曲线以外时

$A_{init} = 0$

A_{fin} : 凸轮曲线的终端加速度 [指令单位 /s²]

但是,

CamNodes (凸轮节点) 的要素编号“m”的 ConnectingAccEnable (连接加速度有效) 为 TRUE 时

$A_{fin} = CamNodes$ (凸轮节点) 的要素编号“m”的 ConnectingAcc (连接加速度)

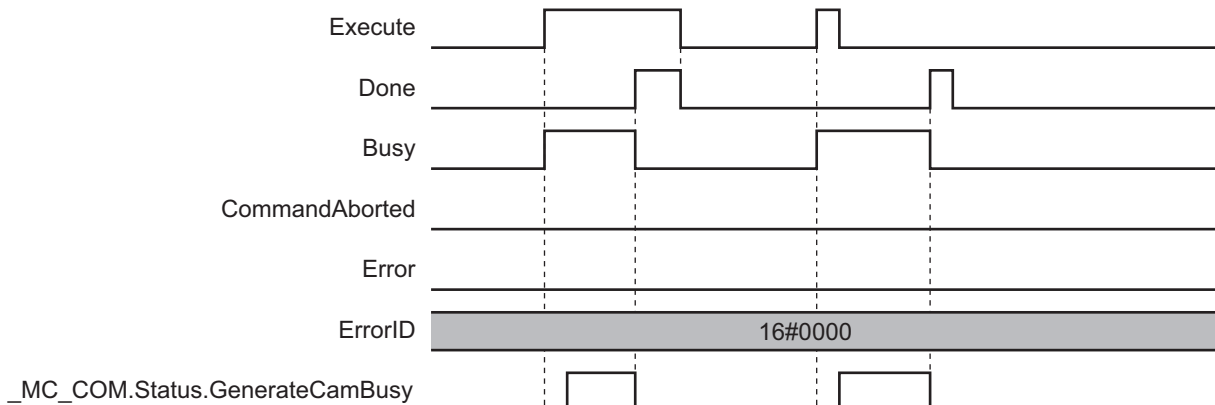
CamNodes (凸轮节点) 的要素编号“m”的 ConnectingAccEnable (连接加速度有效) 为 FALSE 时

$A_{fin} = 0$

V_{init} : 凸轮轮廓曲线的初始速度 [指令单位/s] 及 V_{fin} : 凸轮曲线的终端速度 [指令单位/s] 请参考「3次曲线的位移计算方法 (P.5-27)」。

时序图

在 Execute（执行）上升的同时，Busy（执行中）变为 TRUE，到下一个周期，“凸轮表生成执行中（_MC_COM.Status.GenerateCamBusy）”变为 TRUE。



运动指令重新执行

无法重新执行本指令。

若重新执行，将发生“运动指令不可重新执行（错误代码：543B Hex）”。但是，执行中的凸轮表生成处理将继续执行。

运动指令多重启动

● 其他指令执行过程中本指令的启动

与本指令使用相同 CamTable（凸轮表）的 MC_CamIn（凸轮动作开始）指令、MC_SaveCamTable（凸轮表保存）指令及 MC_SetCamTableProperty（凸轮表属性更新）指令已经启动时，无法多重启动本指令。

● 本指令执行过程中其他指令的启动

执行本指令的过程中，不可多重启动使用同一 CamTable（凸轮表）的 MC_CamIn（凸轮动作开始）指令、MC_SaveCamTable（凸轮表保存）指令及 MC_SetCamTableProperty（凸轮表属性更新）指令。

● 本指令执行过程中本指令的启动

本指令执行过程中，不可启动本指令的其他实例。此外，“凸轮表生成执行中（_MC_COM.Status.GenerateCamBusy）”为 TRUE 时，无法启动本指令。使用“凸轮表生成执行中（_MC_COM.Status.GenerateCamBusy）”或本指令的输出变量进行排他处理。

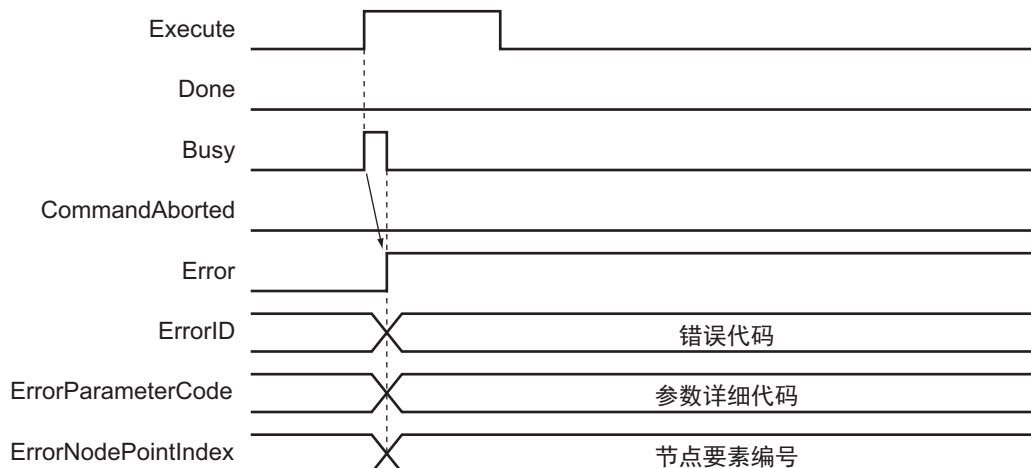
在同一周期内启动本指令的多个实例时，该周期内，“凸轮表生成执行中（_MC_COM.Status.GenerateCamBusy）”为 FALSE。请使用其他实例的 Busy（执行中）进行排他处理。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。

此外，有些 ErrorID（错误代码）时，可能会将附属信息输出到 ErrorParameterCode（参数详细代码）和 ErrorNodePointIndex（节点要素编号）中。



即使在执行本指令的过程中发生异常，CamTable（凸轮表）中指定的凸轮数据变量仍保持本指令执行前的值。

但是，以下情况下不保持凸轮数据变量的值。

- 启动 Execute（执行）后，变更了 CamNodes（凸轮节点）中指定的参数时
- 通过本指令计算得到的凸轮数据的 Distance（位移）值超出 REAL 型的有效范围时

凸轮数据变量的值未保持时，凸轮表将变为错误状态。若指定错误状态的凸轮表，并执行 MC_CamIn（凸轮动作开始）指令、MC_SaveCamTable（凸轮表保存）指令及 MC_SetCamTableProperty（凸轮表属性更新）指令，将发生异常。

要解除凸轮表的错误状态，请先解除指令中发生的异常，再执行以下任一操作。

- 对错误状态的凸轮表再次执行本指令，完成凸轮表的生成
- 通过 Sysmac Studio 下载凸轮表
- 重新接通 CPU 单元的电源

程序模式或 MC 调试过程中，若执行中的本指令发生异常，ErrorID（错误代码）中不输出错误代码。此时，请通过事件日志功能或运动控制系统变量的 MC 共通轻度故障代码（_MC_COM.MFaultLvl.Code），确认异常的原因。

● 错误代码

本指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。

示例程序

下面为重写 Sysmac Studio 的凸轮编辑器中所创建凸轮表的凸轮数据变量的示例程序。

参数设定

下面表示本示例程序中使用的最低限度的设定。

● 轴参数

轴种类

轴	轴种类
轴 1	伺服轴
轴 2	伺服轴

计数器模式

轴	计数器模式
轴 1	旋转模式
轴 2	旋转模式

循环计数器

轴	上限值	下限值
轴 1	360	0
轴 2	360	0

坐标单位选择

轴	坐标单位
轴 1	degree
轴 2	degree

重写内容

介绍通过 Sysmac Studio 的凸轮编辑器事先创建的凸轮属性变量和凸轮节点变量的初始值，以及通过本示例程序重写的值。

● 凸轮属性

凸轮属性变量的初始值如下所示。

InitVel (初始速度)	InitAcc (初始加速度)	CycleTime (周期时间)
0.000	0.000	1.000

本示例程序中，将 CycleTime（周期时间）变更为“0.8”。

InitVel (初始速度)	InitAcc (初始加速度)	CycleTime (周期时间)
0.000	0.000	0.800

● 凸轮节点

凸轮节点变量的初始值如下所示。

要素 编号	Phase (主轴相 位)	Distance (从轴位 移)	Curve (曲线形状)	ConnectingV elEnable (连接速度有 效)	Connectin gVel (连接速 度)	Connecti ngAccEn able (连接加速 度有效)	ConnectingA cc (连接加速度)	PhasePitch (相位间隔 宽度)
0	180.000	-100.000	_mcModifiedSine	FALSE	0.000	FALSE	0.000	0.010
1	360.000	0.000	_mcPolynomial5	FALSE	0.000	FALSE	0.000	0.010
2	0.0	0.0	_mcConstantLine	FALSE	0.000	FALSE	0.0	0.0
3	0.0	0.0	_mcConstantLine	FALSE	0.000	FALSE	0.0	0.0

通过本示例程序，变更要素编号“0”的曲线形状，在要素编号“1”上添加节点。

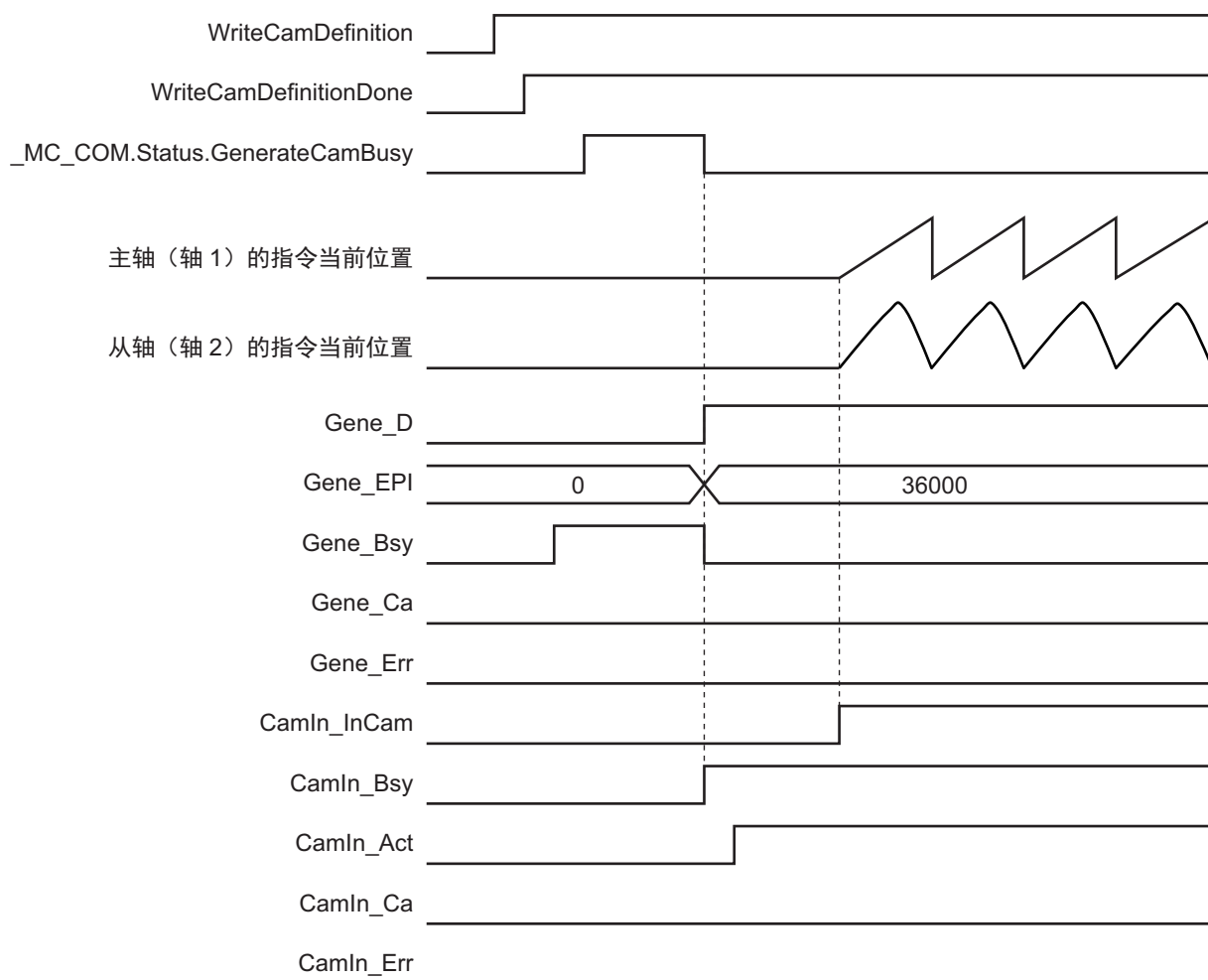
要素 编号	Phase (主轴相 位)	Distance (从轴位 移)	Curve (曲线形状)	ConnectingV elEnable (连接速度有 效)	Connectin gVel (连接速 度)	Connecti ngAccEn able (连接加速 度有效)	ConnectingA cc (连接加速度)	PhasePitch (相位间隔 宽度)
0	180.000	-100.000	_mcPolynomial5	FALSE	0.000	FALSE	0.000	0.010
1	200.000	-102.000	_mcStraightLine	FALSE	0.000	FALSE	0.000	0.010
2	360.000	0.000	_mcPolynomial5	FALSE	0.000	FALSE	0.000	0.010
3	0.0	0.0	_mcConstantLine	FALSE	0.000	FALSE	0.0	0.0

梯形图

● 主要变量

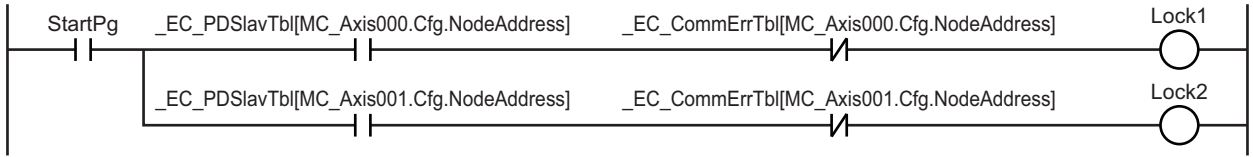
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	主轴轴 1 的轴变量。
MC_Axis001	_sAXIS_REF	—	从轴轴 2 的轴变量。
CamProfile0	ARRAY[0..36000] OF _sMC_CAM_REF	—	凸轮数据变量。 事先在 Sysmac Studio 的凸轮编辑器中创建。
Pwr1_Status	BOOL	—	分配到 MC_Power 的实例 PWR1 的输出 Status 中的变量。若变为伺服 ON 状态，该变量变为 TRUE。
Pwr2_Status	BOOL	—	分配到 MC_Power 的实例 PWR2 的输出 Status 中的变量。若变为伺服 ON 状态，该变量变为 TRUE。
StartPg	BOOL	—	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态，则变为伺服 ON 状态。
WriteCamDefinition	BOOL	—	该变量为 TRUE 时，重写凸轮属性变量和凸轮节点变量的值。
CamProperty0	_sMC_CAM_PROPERTY	—	凸轮属性变量。
CamNode0	ARRAY[0..3] OF _sMC_CAM_NODE	—	凸轮节点变量。
_MC_COM.Status.GenerateCamBusy	BOOL	—	运动控制系统变量。正在生成凸轮表时变为 TRUE。

● 时序图

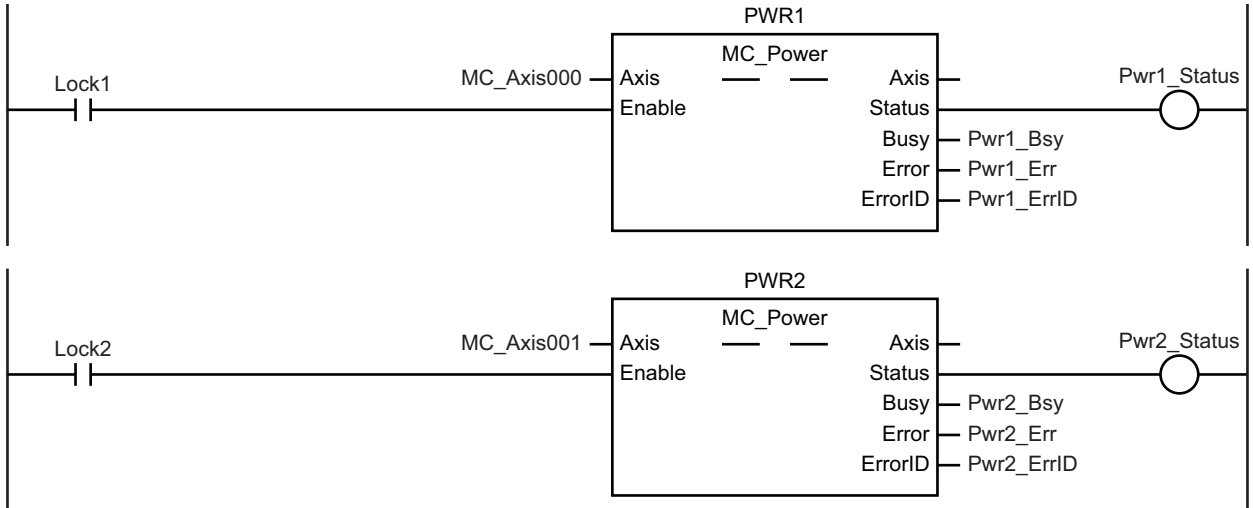


● 示例程序

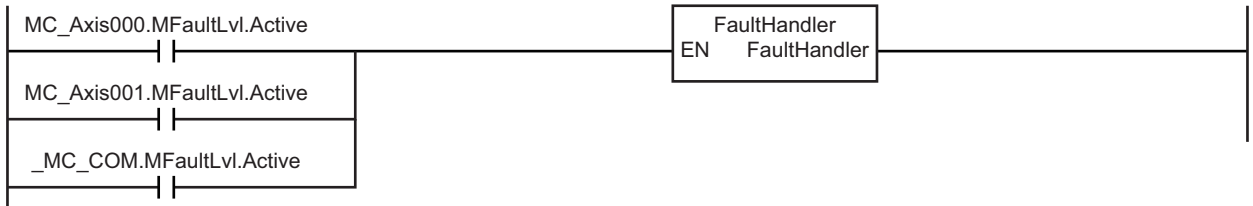
接点 StartPg 变为 TRUE 后，确认各轴的 EtherCAT 通信是否正常进行过程数据通信



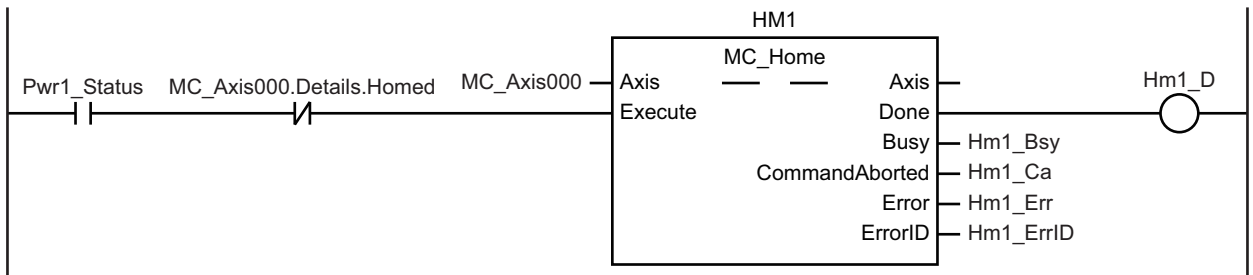
若正在进行过程数据通信，则将各轴设为伺服 ON 状态



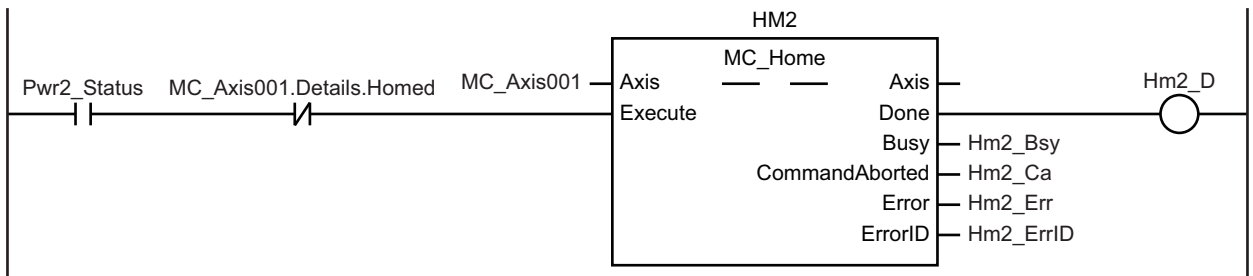
发生轴或 MC 共通的轻度故障等级的异常时，执行装置发生异常时的处理（FaultHandler）。发生异常时的处理（FaultHandler）请根据装置由客户进行编程



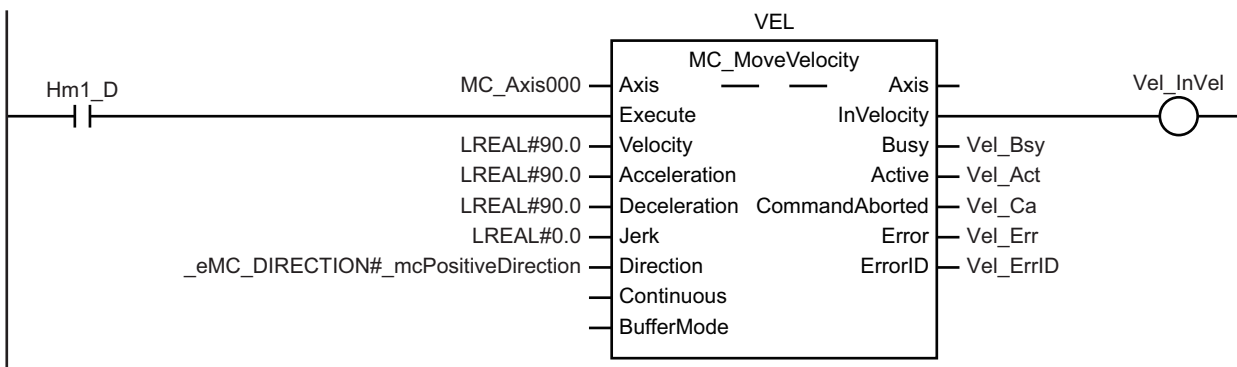
主轴（轴 1）为伺服 ON 状态且处于原点未确定状态时，执行原点复位，确定原点



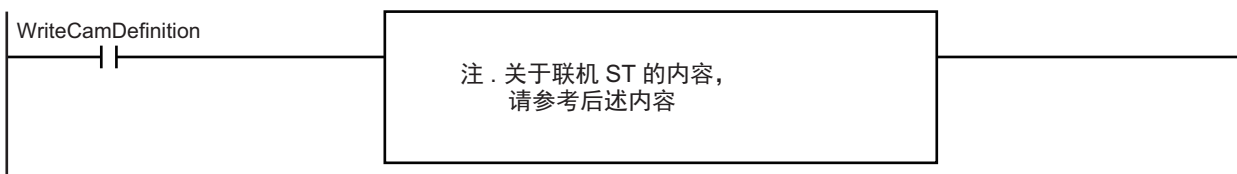
从轴（轴 2）为伺服 ON 状态且处于原点未确定状态时，执行原点复位，确定原点



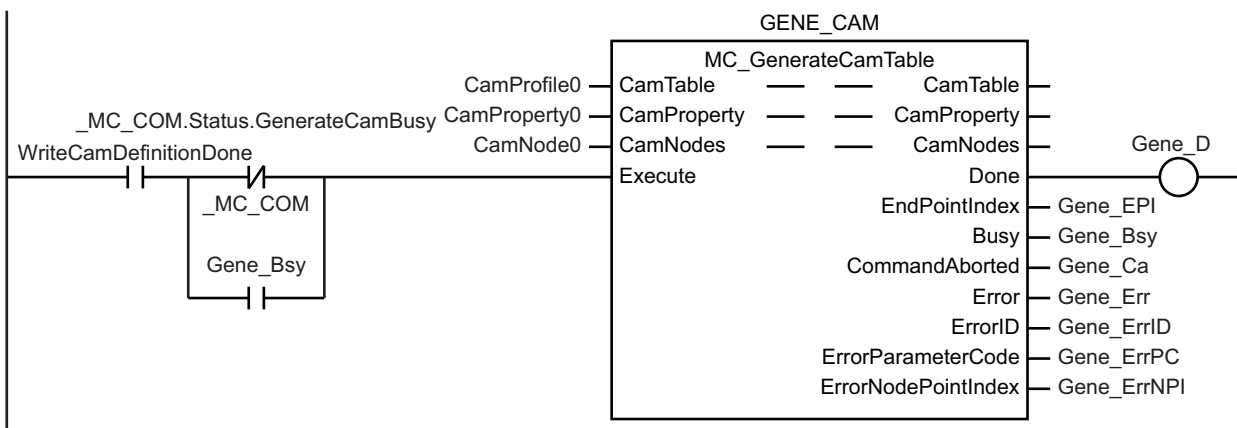
主轴（轴 1）完成原点复位后，启动 MC_MoveVelocity（速度控制）指令



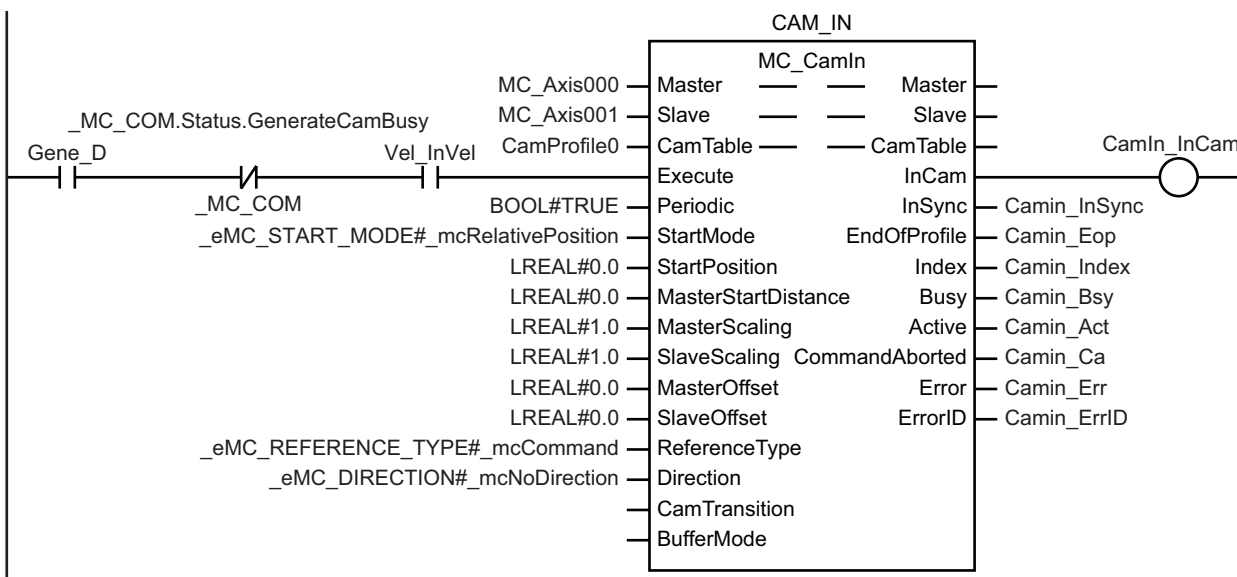
WriteCamDefinition 为 TRUE 且 MC_GenerateCamTable（凸轮表生成）指令为未执行状态时，重写凸轮属性和凸轮节点。重写后，WriteCamDefinitionDone 变为 TRUE



WriteCamDefinitionDone 为 TRUE 且不是凸轮表生成处理中时，启动 MC_GenerateCamTable（凸轮表生成）指令



完成凸轮表生成处理，主轴（轴 1）达到目标速度后，启动 MC_CamIn（凸轮动作开始）指令



联机 ST 的内容

```

CamProperty0.CycleTime           := REAL#0.800;
CamNode0[0].Curve                 := _eMC_CAM_CURVE#_mcPolynomic5;
CamNode0[0].ConnectingVelEnable  := FALSE;
CamNode0[0].ConnectingVel        := REAL#0.000;
CamNode0[0].ConnectingAccEnable  := FALSE;
CamNode0[0].ConnectingAcc        := REAL#0.000;
CamNode0[1].Phase                 := REAL#200.000;
CamNode0[1].Distance              := REAL#-102.000;
CamNode0[1].Curve                 := _eMC_CAM_CURVE#_mcStraightLine;
CamNode0[1].ConnectingVelEnable  := FALSE;
CamNode0[1].ConnectingVel        := REAL#0.000;
CamNode0[1].ConnectingAccEnable  := FALSE;
CamNode0[1].ConnectingAcc        := REAL#0.000;
CamNode0[2].Phase                 := REAL#360.000;
CamNode0[2].Distance              := REAL#0.000;
CamNode0[2].Curve                 := _eMC_CAM_CURVE#_mcPolynomic5;
CamNode0[2].ConnectingVelEnable  := FALSE;
CamNode0[2].ConnectingVel        := REAL#0.000;
CamNode0[2].ConnectingAccEnable  := FALSE;
CamNode0[2].ConnectingAcc        := REAL#0.000;
CamNode0[2].PhasePitch           := REAL#0.010;
CamNode0[3].Phase                 := REAL#0.000;
WriteCamDefinitionDone            := TRUE;

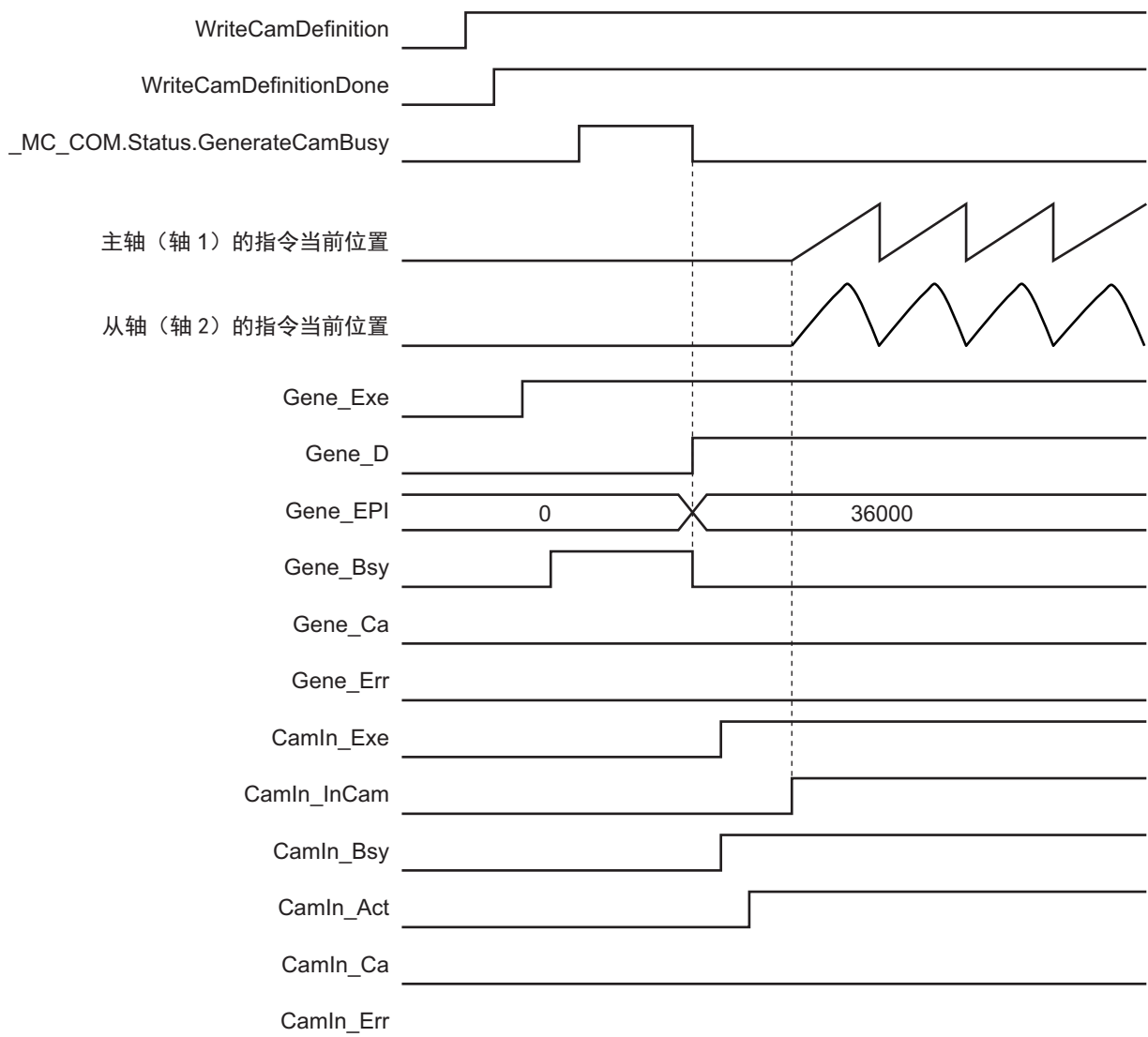
```

结构文本 (ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	主轴轴 1 的轴变量。
MC_Axis001	_sAXIS_REF	—	从轴轴 2 的轴变量。
CamProfile0	ARRAY[0..36000] OF _sMC_CAM_REF	—	凸轮数据变量。 事先在 Sysmac Studio 的凸轮编辑器中创建。
Pwr1_Status	BOOL	—	分配到 MC_Power 的实例 PWR1 的输出 Status 中的变量。若变为伺服 ON 状态, 该变量变为 TRUE。
Pwr2_Status	BOOL	—	分配到 MC_Power 的实例 PWR2 的输出 Status 中的变量。若变为伺服 ON 状态, 该变量变为 TRUE。
StartPg	BOOL	—	若该变量为 TRUE 且 EtherCAT 的过程数据通信为建立状态, 则变为伺服 ON 状态。
WriteCamDefinition	BOOL	—	该变量为 TRUE 时, 重写凸轮属性变量和凸轮节点变量的值。
CamProperty0	_sMC_CAM_PROPERTY	—	凸轮属性变量。
CamNode0	ARRAY[0..3] OF _sMC_CAM_NODE	—	凸轮节点变量。
_MC_COM.Status.GenerateCamBusy	BOOL	—	运动控制系统变量。正在生成凸轮表时变为 TRUE。

● 时序图



● 示例程序

```

// StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 1 设为伺服 ON 状态
// EtherCAT 通信未正常通信时, 设为伺服 OFF
IF (StartPg=TRUE)
  AND (_EC_PDslavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
  Pwr1_En:=TRUE;
ELSE
  Pwr1_En:=FALSE;
END_IF;

// StartPg 为 TRUE 时, 确认 EtherCAT 通信正常通信,
// 轴 2 设为伺服 ON 状态
// EtherCAT 通信未正常通信时, 设为伺服 OFF
IF (StartPg=TRUE)
  AND (_EC_PDslavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
  AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
  Pwr2_En:=TRUE;
ELSE
  Pwr2_En:=FALSE;
END_IF;

// 发生轴或 MC 共通的轻度故障时, 执行异常处理 FaultHandler
// 发生异常时的处理 (FaultHandler) 请根据装置由客户进行编程
IF (MC_Axis000.MFaultLvl.Active=TRUE)
OR (MC_Axis001.MFaultLvl.Active=TRUE)
OR (_MC_COM.MFaultLvl.Active=TRUE) THEN
  FaultHandler();
END_IF;

// 轴 1 为可运行状态且处于原点未确定状态时, 执行原点复位
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
  Hm1_Ex:=TRUE;
END_IF;

// 轴 2 为可运行状态且处于原点未确定状态时, 执行原点复位
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
  Hm2_Ex:=TRUE;
END_IF;

// 轴 1 完成原点复位后, 启动 MC_MoveVelocity
IF Hm1_D=TRUE THEN
  Vel_Ex := TRUE;
END_IF;

// WriteCamDefinition 为 TRUE 且 MC_GenerateCamTable (凸轮表生成) 指令为未执行状态时,
// 重写凸轮属性和凸轮节点。
// 重写后, WriteCamDefinitionDone 变为 TRUE
IF (WriteCamDefinition = TRUE) THEN
  // 凸轮属性的重写
  CamProperty0.CycleTime           := REAL#0.800;
  // 凸轮节点的重写
  CamNode0[0].Curve                 := _eMC_CAM_CURVE#_mcPolynomic5;
  CamNode0[0].ConnectingVelEnable   := FALSE;
  CamNode0[0].ConnectingVel         := REAL#0.000;
  CamNode0[0].ConnectingAccEnable   := FALSE;
  CamNode0[0].ConnectingAcc         := REAL#0.000;
  CamNode0[1].Phase                 := REAL#200.000;
  CamNode0[1].Distance              := REAL#-102.000;
  CamNode0[1].Curve                 := _eMC_CAM_CURVE#_mcStraightLine;
  CamNode0[1].ConnectingVelEnable   := FALSE;

```

```

CamNode0[1].ConnectingVel           := REAL#0.000;
CamNode0[1].ConnectingAccEnable     := FALSE;
CamNode0[1].ConnectingAcc           := REAL#0.000;
CamNode0[2].Phase                   := REAL#360.000;
CamNode0[2].Distance                := REAL#0.000;
CamNode0[2].Curve                   := _eMC_CAM_CURVE#_mcPolynomic5;
CamNode0[2].ConnectingVelEnable     := FALSE;
CamNode0[2].ConnectingVel           := REAL#0.000;
CamNode0[2].ConnectingAccEnable     := FALSE;
CamNode0[2].ConnectingAcc           := REAL#0.000;
CamNode0[2].PhasePitch              := REAL#0.010;
CamNode0[3].Phase                   := REAL#0.000;
// 将 WriteCamDefinitionDone 设为 TRUE
WriteCamDefinitionDone              := TRUE;
END_IF;

// WriteCamDefinitionDone 为 TRUE 且不是凸轮表生成处理中时, 启动 MC_GenerateCamTable
// (凸轮表生成) 指令
IF(WriteCamDefinitionDone = TRUE)
AND(_MC_COM.Status.GenerateCamBusy = FALSE) THEN
  Gene_Exe := TRUE;
END_IF;

// 完成凸轮表生成处理, 主轴 (轴 1) 达到目标速度后, 启动 MC_CamIn (凸轮动作开始) 指令
IF (Gene_D=TRUE)
AND(_MC_COM.Status.GenerateCamBusy = FALSE)
AND (Vel_InVel=TRUE) THEN
  Camin_Ex := TRUE;
END_IF;

// MC_GenerateCamTable
GENE_CAM(
  CamTable           := CamProfile0,
  CamProperty        := CamProperty0,
  CamNodes           := CamNode0,
  execute            := Gene_Exe,
  Done               => Gene_D,
  EndPointIndex      => Gene_EPI,
  Busy               => Gene_Bsy,
  CommandAborted     => Gene_CA,
  Error              => Gene_Err,
  ErrorID            => Gene_ErrID,
  ErrorParameterCode => Gene_ErrPC,
  ErrorNodePointIndex => Gene_ErrNPI
);

// 主轴 (轴 1) 的 MC_Power
PWR1(
  Axis       := MC_Axis000,
  Enable     := Pwr1_En,
  Status     => Pwr1_Status,
  Busy       => Pwr1_Bsy,
  Error      => Pwr1_Err,
  ErrorID    => Pwr1_ErrID
);

// 从轴 (轴 2) 的 MC_Power
PWR2(
  Axis       := MC_Axis001,
  Enable     := Pwr2_En,
  Status     => Pwr2_Status,
  Busy       => Pwr2_Bsy,
  Error      => Pwr2_Err,
  ErrorID    => Pwr2_ErrID
);

```

```

// 主轴（轴 1）的 MC_Home
HM1(
  Axis           := MC_Axis000,
  Execute        := Hm1_Ex,
  Done           => Hm1_D,
  Busy           => Hm1_Bsy,
  CommandAborted => Hm1_Ca,
  Error          => Hm1_Err,
  ErrorID       => Hm1_ErrID
);

// 从轴（轴 2）的 MC_Home
HM2(
  Axis           := MC_Axis001,
  Execute        := Hm2_Ex,
  Done           => Hm2_D,
  Busy           => Hm2_Bsy,
  CommandAborted => Hm2_Ca,
  Error          => Hm2_Err,
  ErrorID       => Hm2_ErrID
);

//MC_MoveVelocity
VEL(
  Axis           := MC_Axis000,
  Execute        := Vel_Ex,
  Velocity       := Vel_Vel,
  Acceleration   := Vel_Acc,
  Deceleration   := Vel_Dec,
  Direction      := Vel_Dir,
  InVelocity     => Vel_InVel,
  Busy           => Vel_Bsy,
  Active         => Vel_Act,
  CommandAborted => Vel_Ca,
  Error          => Vel_Err,
  ErrorID       => Vel_ErrID
);

//MC_CamIn
CAM_IN(
  Master         := MC_Axis000,
  Slave          := MC_Axis001,
  CamTable       := CamProfile0,
  Execute        := Camin_Ex,
  Periodic       := Camin_P,
  StartMode      := Camin_Sm,
  StartPosition  := Camin_Sp,
  MasterStartDistance := Camin_Msd,
  MasterScaling  := Camin_Ms,
  SlaveScaling   := Camin_Ss,
  MasterOffset   := Camin_Mo,
  SlaveOffset    := Camin_So,
  ReferenceType  := Camin_Rt,
  Direction      := Camin_Dir,
  InCam          => Camin_InCam,
  InSync         => Camin_InSync,
  EndOfProfile   => Camin_Eop,
  Index          => Camin_Index,
  Busy           => Camin_Bsy,
  Active         => Camin_Act,
  CommandAborted => Camin_Ca,
  Error          => Camin_Err,
  ErrorID       => Camin_ErrID
);

```

MC_WriteAxisParameter

重写 MC 功能模块的轴参数设定。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_WriteAxisParameter	轴参数写入	FB		<pre>MC_WriteAxisParameter_instance (Axis := 《参数》, AxisParameter := 《参数》, Execute := 《参数》, Done => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》, ErrorParameterCode => 《参数》);</pre>



版本相关信息

本指令可在 CPU 单元版本 1.08 以上和 Sysmac Studio Ver.1.09 以上的组合中使用。

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、 FALSE	FALSE	上升时开始指令。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、 FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、 FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。
ErrorParameterCode	参数详细代码	WORD	*	有些错误代码存在附属信息。存在时，输出发生异常的参数之详细代码。

* □ 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	指令完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	因发生异常, 本指令中止时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定重写对象的轴。*1
AxisParameter	轴参数	_sAXIS_PARAM	—	指定要重写的值。*2

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

*2. 请定义 _sAXIS_PARAM 型用户定义变量。

功能说明

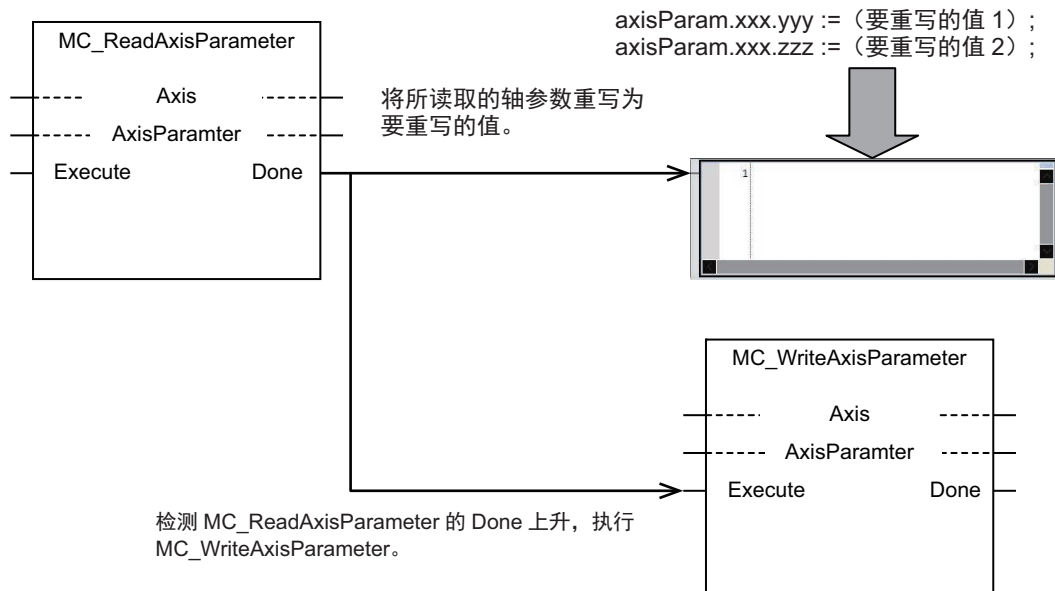
- 本指令可在 Execute（启动）的上升沿，将 AxisParameter（轴参数）中指定的值写入指定 Axis（轴）的轴参数中。
- 仅在指定轴的轴使用为未使用轴时可以写入。若在其他情况下执行，将发生错误，不会写入轴参数。将保持执行指令前的值。
- 若在通过 MC_ChangeAxisUse（轴使用变更）指令切换轴使用后执行本指令，请在确认 MC_ChangeAxisUse（轴使用变更）指令的输出变量“Done（完成）”为 TRUE 后再执行。
- AxisParameter（轴参数）为错误值时，或轴参数之间不匹配时，执行时将发生错误，不会写入轴参数。将保持执行指令前的值。

轴参数的可设定范围检查内容及轴参数间的匹配检查请参考 □ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇（SBCE-CN5-363）》。

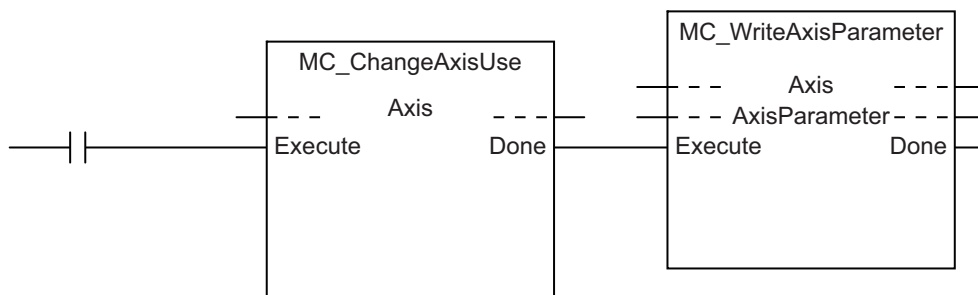


使用注意事项

- 通过本指令重写的值不会保存到 CPU 单元的非易失性存储器中。因此，CPU 单元的电源 OFF、开始下载和 MC 功能模块的重新启动处理时，重写的值将消失，恢复为 Sysmac Studio 中设定的值。
- 通过本指令重写的值无法通过 Sysmac Studio 上传。
- 本指令不仅是对要写入的轴参数，还要对写入对象轴的轴参数进行参数设定。对象参数请参考 □ 「写入或读取对象的轴参数 (P.5-55)」。不需要变更的轴参数请设定为通过 Sysmac Studio 设定的值或通过 MC_ReadAxisParameter（轴参数读取）指令读取的值。
- 本指令和 MC_ReadAxisParameter（轴参数读取）指令配合使用时的执行示意图如下所示。



- 下面表示本指令和 MC_ChangeAxisUse（轴使用变更）指令配合使用时的示例。请在确认 MC_ChangeAxisUse（轴使用变更）指令已变为 Done（完成）后再执行。



- 若对编码器种类设定为“绝对编码器(ABS)”的轴，使用本指令变更“单位转换设定”或“位置计数设定”，机械的物理位置和 MC 功能模块上的轴位置关系会发生变化。此时，请通过 MC_ChangeAxisUse（轴使用变更）指令变更为使用轴后，在执行原点复位指令，再次确定原点。
变更为使用轴后，如果发生“无法计算绝对编码器当前位置（错误代码：6458 Hex）”，请在解除轴异常后，执行原点复位指令，再次确定原点。
- 轴参数的 Reserved 变量请指定为“0”以外的值。

指令的详情

下面介绍指令的详情。

● sAXIS_PARAM

sAXIS_PARAM 型是指指定轴参数的值所需的结构体数据类型。由不同类型轴参数的结构体数据类型构成。本指令中，作为输入输出变量的 AxisParameter（轴参数）中指定的变量使用。

下面介绍各成员。

成员变量	参数名称	数据类型	内容
UnitConversion	单位转换设定	_sAXIS_UNIT_CONVERSION_SETTINGS	对单位转换设定的各成员变量，指定要重写的值。
Operation	动作设定	_sAXIS_OPERATION_SETTINGS	对动作设定的各成员变量，指定要重写的值。
OtherOperation	扩展动作设定	_sAXIS_OTHER_OPERATION_SETTINGS	对扩展动作设定的各成员变量，指定要重写的值。
Limit	限位设定	_sAXIS_LIMIT_SETTINGS	对限位设定的各成员变量，指定要重写的值。
PosCount	位置计数设定	_sAXIS_POSITION_COUNT_SETTINGS	对位置计数设定的各成员变量，指定要重写的值。
Homing	原点复位设定	_sAXIS_HOMING_SETTINGS	对原点复位设定的各成员变量，指定要重写的值。
Reserved	（保留区域）	ARRAY[0..255] OF BYTE	—

● sAXIS_UNIT_CONVERSION_SETTINGS (单位转换设定)

sAXIS_UNIT_CONVERSION_SETTINGS 型是指指定轴参数的单位转换设定值所需的结构体数据型。

下面介绍各成员。

成员变量	参数名称	数据类型	有效范围	内容
UnitDisplay	显示单位	_eMC_UNITS	0: _mcPls 1: _mcMm 2: _mcUm 3: _mcNm 4: _mcDeg 5: _mcInch	设定指令位置的单位。 0: pulse 1: mm 2: μm 3: nm 4: degree 5: inch
CmdPlsCountPerMotorRotation	电机每转 1 圈的脉冲数	UDINT	1 ~ 4294967295	根据编码器分辨率, 设定指令位置的电机每转 1 圈的脉冲数。 根据电子齿轮比, 从指令值转换为脉冲量。
WorkTravelDistancePerMotorRotation	电机每转 1 圈的移动量	LREAL	0.000000001 ~ 4294967295	设定指令位置的电机每转 1 圈的工件移动量。
ReducerUse ^{*1}	减速机使用	BOOL	TRUE、FALSE	指定是否使用减速机设定。 TRUE: 使用 FALSE: 不使用
WorkTravelDistancePerWorkSideRotation ^{*1 *2 *3}	工件每转 1 圈的移动量	LREAL	双精度实数型的正数	设定每转 1 圈后工件的移动量。
WorkSideGearRatio ^{*1 *2}	工件侧齿轮比	UDINT	1 ~ 4294967295	设定工件侧的齿轮比。
MotorSideGearRatio ^{*1 *2}	电机侧齿轮比	UDINT	1 ~ 4294967295	设定电机侧的齿轮比。
Reserved	(保留区域)	ARRAY [0..7] OF BYTE ^{*4}	—	—

*1. 可在 CPU 单元版本 1.11 以上和 Sysmac Studio Ver.1.15 以上的组合中使用。

*2. 将减速机设定为“使用”时, 为有效。

*3. 仅当计数器模式为直线模式时可设定。为旋转模式时, 根据循环计数器上限设定值及循环计数器下限设定值计算。

*4. 单元版本 1.10 以下的 CPU 单元中, 为 ARRAY [0..31] OF BYTE。

● _sAXIS_OPERATION_SETTINGS (动作设定)

_sAXIS_OPERATION_SETTINGS 型是指指定轴参数的动作设定值所需的结构体数据类型。

下面介绍各成员。

成员变量	参数名称	数据类型	有效范围	内容
MaxVel	最高速度	LREAL	双精度实数型的正数	设定各轴的最高速度。 设定时请勿超出所用电机的最大转速。
StartVel	启动速度	LREAL	0.0 ~ 双精度实数型的正上限值	设定各轴的启动速度。 设定时请勿超出最高速度。
MaxJogVel	点动最高速度	LREAL	双精度实数型的正数	设定各轴的点动最高速度。 设定时请勿超出最高速度。
MaxAcc	最大加速度	LREAL	双精度实数型的正数、0	设定轴动作指令时各轴的最大加速度。 若设定为“0”，将没有加速度的限制。
MaxDec	最大减速度	LREAL	双精度实数型的正数、0	设定轴动作指令时各轴的最大减速度。 若设定为“0”，将没有减速度的限制。
AccDecOver	加减速超限	_eMC_ ACCDECO VER	0: _mcAccDecOverBuffer 1: _mcAccDecOverRapid 2: _mcAccDecOverError Stop	在轴的加减速控制中，为了优先停止在目标位置，会发生减速超限，导致超出最大加减速速度，指定这种情况下的动作。 0: 剧烈加减速（将共混切换为缓冲）*1 1: 剧烈加减速 2: 异常停止*2
ReverseMode	反转时动作	_eMC_ REVERSE _MODE	0: _mcReverseMode DecelerationStop 1: _mcReverseMode ImmediateStop	指定指令多重启动时、指令重新启动时、中断固定尺寸定位时的反转时动作。 0: 减速停止 1: 立即停止
VelWarningVal	速度警告值	UINT	0、1 ~ 100	设定相对于最高速度的比例，作为各轴输出速度警告的标准。 若设定为“0”，将不输出速度警告。
AccWarningVal	加速度警告值	UINT	0、1 ~ 100	设定相对于最高加速度的比例，作为各轴输出加速度警告的标准。 若设定为“0”，将不输出加速度警告。
DecWarningVal	减速度警告值	UINT	0、1 ~ 100	设定相对于最高减速度的比例，作为各轴输出减速度警告的标准。 若设定为“0”，将不输出减速度警告。
PosiTrq WarningVal	正方向扭矩警告值	UINT	0、1 ~ 1000	设定扭矩指令值，作为各轴输出正方向扭矩警告的标准。 若设定为“0”，将不输出正方向扭矩警告。
NegaTrq WarningVal	负方向扭矩警告值	UINT	0、1 ~ 1000	设定扭矩指令值，作为各轴输出负方向扭矩警告的标准。 若设定为“0”，将不输出负方向扭矩警告。

成员变量	参数名称	数据类型	有效范围	内容
InPosRange	位置宽度	LREAL	双精度实数型的正数、0	设定定位完成宽度
InPosCheck Time	位置检查时间	UINT	0、1 ~ 10000	以毫秒为单位设定完成定位的检查时间。 若设定为“0”，仅在原点复位中确定原点时继续执行定位完成检查。不是原点复位时，不执行定位完成检查。
ActVelFilter TimeConstant	反馈速度过滤器 时间常数	UINT	0、1 ~ 100	以毫秒为单位设定计算反馈速度移动平均值的时间。 若设定为“0”，将不计算移动平均值。
ZeroPosRange	原点位置范围	LREAL	双精度实数型的正数、0	设定原点位置检测宽度。
Reserved	(保留区域)	ARRAY [0..31] OF BYTE	—	—

*1. 单元版本 1.10 以上的 CPU 单元中，无法将共混切换为缓冲。详情请参考 □□ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

*2. 单元版本 1.10 以上的 CPU 单元中，发生共混动作时，将视为发生异常而停止。详情请参考 □□ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

● _sAXIS_OTHER_OPERATION_SETTINGS (扩展动作设定)

_sAXIS_OTHER_OPERATION_SETTINGS 型是指指定轴参数的扩展动作设定值所需的结构体数据类型。
下面介绍各成员。

成员变量	参数名称	数据类型	有效范围	内容
ImmediateStop InputStopMode	立即停止输入停止方法	_eMC_ST OP_MODE	1: _mcImmediateStop 2: _mcImmediateStopFE Reset 3: _mcFreeRunStop	设定立即停止输入有效时 MC 功能模块的停止方法。 1: 立即停止指令值 2: 立即停止指令值，同时执行偏差计数器复位 3: 立即停止指令值，同时执行伺服 OFF
LimitInputStopMode	极限输入停止方法	_eMC_ST OP_MODE	0: _mcDecelerationStop 1: _mcImmediateStop 2: _mcImmediateStopFE Reset 3: _mcFreeRunStop	设定正方向极限输入或负方向极限输入有效时 MC 功能模块的停止方法。 0: 减速停止指令值 1: 立即停止指令值 2: 立即停止指令值，同时执行偏差计数器复位 3: 立即停止指令值，同时执行伺服 OFF
DriveErrorReset MonitoringTime	驱动器错误重置 监视时间	UINT	1 ~ 1000	设定执行驱动器错误重置时的监视时间。(单位: ms) 若超出监视时间，即使驱动器的异常未重置，也将结束重置处理。
MaxPosiTrq Limit	正方向扭矩限制 上限值	LREAL	0.0 ~ 1000.0	设定正方向扭矩限制值的上限。
MaxNegaTrq Limit	负方向扭矩限制 上限值	LREAL	0.0 ~ 1000.0	设定负方向扭矩限制值的上限。

成员变量	参数名称	数据类型	有效范围	内容
ImmediateStop InputLogic Inversion	立即停止输入逻辑反转	BOOL	TRUE、FALSE	设定立即停止输入信号的逻辑反转。 TRUE: 反转 FALSE: 不反转
PosiLimitInput LogicInversion	正方向极限输入逻辑反转	BOOL	TRUE、FALSE	设定正方向极限输入信号的逻辑反转。 TRUE: 反转 FALSE: 不反转
NegaLimitInputLo gicInversion	负方向极限输入逻辑反转	BOOL	TRUE、FALSE	设定负方向极限输入信号的逻辑反转。 TRUE: 反转 FALSE: 不反转
HomeProximity InputLogic Inversion	近原点输入逻辑反转	BOOL	TRUE、FALSE	设定近原点输入信号的逻辑反转。 TRUE: 反转 FALSE: 不反转
Reserved	(保留区域)	ARRAY [0..31] OF BYTE	—	—

● sAXIS_LIMIT_SETTINGS (限位设定)

sAXIS_LIMIT_SETTINGS 型是指指定轴参数的限位设定值所需的结构体数据类型。

下面介绍各成员。

成员变量	参数名称	数据类型	有效范围	内容
SwLimitMode	软件限位功能	_eMC_ SWLMT_M ODE	0: _mcNonSwLmt 1: _mcCmdDeceleration Stop 2: _mcCmdImmediate Stop 3: _mcActDeceleration Stop 4: _mcActImmediateStop	选择软件限位的功能。 0: 无效 1: 对指令位置有效, 减速停止 2: 对指令位置有效, 立即停止 3: 对反馈位置有效, 减速停止 4: 对反馈位置有效, 立即停止
PosiSwLimit	正方向软件限位	LREAL	双精度实数型的负数、正数、0	设定正方向的软件限位。
NegaSwLimit	负方向软件限位	LREAL	双精度实数型的负数、正数、0	设定负方向的软件限位。
FollowingError OverVal	位置偏差超限值	LREAL	双精度实数型的正数、0	设定位置偏差超限的阈值。 若设定为“0”, 位置偏差超限检查为无效。
FollowingErrorWa rningVal	位置偏差警告值	LREAL	双精度实数型的正数、0	设定位置偏差警告的阈值。 若设定为“0”, 位置偏差警告检查为无效。
Reserved	(保留区域)	ARRAY [0..31] OF BYTE	—	—

● _sAXIS_POSITION_COUNT_SETTINGS（位置计数设定）

_sAXIS_POSITION_COUNT_SETTINGS 型是指指定轴参数的位置计数设定值所需的结构体数据类型。

下面介绍各成员。

成员变量	参数名称	数据类型	有效范围	内容
CountMode	计数模式	_eMC_COUNT_MODE	0: _mcCountModeLinear 1: _mcCountModeRotary	设定位置的计数模式。 0: 直线模式（长度有限） 1: 旋转模式（长度无限）
ModuloMaxPosVal	循环计数器上限设定值	LREAL	双精度实数型的负数、正数、0	设定计数模式为旋转模式时循环计数器的上限值。
ModuloMinPosVal	循环计数器下限设定值	LREAL	双精度实数型的负数、正数、0	设定计数模式为旋转模式时循环计数器的下限值。
Reserved	（保留区域）	ARRAY [0..31] OF BYTE	—	—

● _sAXIS_HOMING_SETTINGS（原点复位设定）

_sAXIS_HOMING_SETTINGS 型是指指定轴参数的原点复位设定值所需的结构体数据类型。

下面介绍各成员。

成员变量	参数名称	数据类型	有效范围	内容
Mode	原点复位动作	_eMC_HOMING_MODE	0: _mcHomeSwTurnHomeSwOff 1: _mcHomeSwTurnHomeSwOn 4: _mcHomeSwOff 5: _mcHomeSwOn 8: _mcLimitInputOff 9: _mcHomeSwTurnHomeMask 11: _mcLimitInputOnly 12: _mcHomeSwTurnHoldingTime 13: _mcNoHomeSwHoldingHomeInput 14: _mcHomePreset	设定原点复位的动作方法。 0: 附近退避、近原点输入 OFF 指定 1: 附近退避、近原点输入 ON 指定 4: 近原点输入 OFF 指定 5: 近原点输入 ON 指定 8: 极限输入 OFF 指定 9: 附近退避、原点输入掩码距离指定 11: 仅极限输入 12: 附近退避、推压时间指定 13: 无近原点输入、推压原点输入指定 14: 原点预设
HomeInputSignal	原点输入信号	_eMC_HOME_INPUT	0: _mcZPhase 1: _mcExternalSignal	选择原点输入信号。 0: 使用 Z 相输入 1: 使用外部原点输入
StartDirection	原点复位开始方向	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection	设定原点复位启动时的开始方向。 0: 正方向 2: 负方向
HomeInputDetectionDirection	原点检测方向	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection	设定原点复位的原点输入检测方向。 0: 正方向 2: 负方向
PosiLimitInputMode	正方向极限输入时动作	_eMC_LIMIT_REVERSE_MODE	0: _mcErrorStop 1: _mcRevImmediateStop 2: _mcRevDecelerationStop	设定在原点复位过程中将正方向极限输入设为“ON”时的停止方法。 0: 不反转、轴异常停止（停止方法遵照极限输入停止方法） 1: 反转、立即停止 2: 反转、减速停止

成员变量	参数名称	数据类型	有效范围	内容
NegaLimitInputMode	负方向极限输入时动作	_eMC_ LIMIT_ REVERSE _MODE	0: _mcErrorStop 1: _mcRevImmediateStop 2: _mcRevDeceleration Stop	设定在原点复位过程中将负方向极限输入设为“ON”时的停止方法。 0: 不反转、轴异常停止 (停止方法遵照极限输入停止方法) 1: 反转、立即停止 2: 反转、减速停止
Vel	原点复位速度	LREAL	双精度实数型的正数	设定原点复位时的速度。
ApproachVel	原点复位接近速度	LREAL	双精度实数型的正数	设定近原点输入变为“ON”后的速度。
Acc	原点复位加速度	LREAL	双精度实数型的正数、0	指定原点复位时的加速度。 若设定为“0”，则不加速，达到原点复位速度等目标速度。
Dec	原点复位减速度	LREAL	双精度实数型的正数、0	指定原点复位时的减速度。 若设定为“0”，则不减速，达到原点复位接近速度等目标速度。
Jerk	原点复位跃度	LREAL	双精度实数型的正数、0	指定原点复位时的跃度。 若设定为“0”，将没有跃度。
HomeInputMaskDistance	原点输入掩码距离	LREAL	双精度实数型的正数、0	设定原点复位动作模式为“附近退避、原点输入掩码距离指定”时的原点输入掩码距离。
HomeOffset	原点位置偏置	LREAL	双精度实数型的负数、正数、0	原点复位完成时按设定的值预设当前位置。
HoldingTime	原点复位推压时间	UINT	0 ~ 10000	以毫秒为单位设定原点复位动作模式为“附近退避、推压时间指定”时的推压时间。
CompensationVal	原点复位修正值	LREAL	双精度实数型的负数、正数、0	设定原点复位的原点确定后的原点复位修正量。
CompensationVel	原点复位修正速度	LREAL	双精度实数型的正数	设定原点复位修正时的速度。
Reserved	(保留区域)	ARRAY [0..31] OF BYTE	—	—

写入或读取对象的轴参数

本指令的写入对象轴参数如下。

MC_ReadAxisParameter（轴参数读取）指令的读取对象轴参数亦同。

轴参数分类	轴参数名称	○：写入 / 读取对象 ○：非写入 / 读取对象
轴基本设定	轴编号	×
	轴使用	×
	轴种类	×
	输入设备 / 输出设备	×
单位转换设定	显示单位	○
	电机每转 1 圈的脉冲数	○
	电机每转 1 圈的移动量	○
	减速机使用 *1	○
	工件侧每转 1 圈的移动量 *1	○ *2
	工件侧齿轮比 *1	○
	电机侧齿轮比 *1	○
动作设定	最高速度	○ *3
	启动速度	○ *3
	点动最高速度	○ *3
	最大加速度	○ *3
	最大减速度	○ *3
	加减速超限	○ *3
	反转时动作	○ *3
	速度警告值	○ *3
	加速度警告值	○ *3
	减速度警告值	○ *3
	正方向扭矩警告值	○ *3
	负方向扭矩警告值	○ *3
	位置宽度	○ *3
	位置检查时间	○ *3
	反馈速度过滤器时间常数	○
	原点位置范围	○ *3
	扩展动作设定	立即停止输入停止方法
极限输入停止方法		○ *3
驱动器错误重置监视时间		○ *3
正方向扭矩限制上限值		○ *3
负方向扭矩限制下限值		○ *3
立即停止输入逻辑反转		○ *3
正方向极限输入逻辑反转		○ *3
负方向极限输入逻辑反转		○ *3
近原点输入逻辑反转		○ *3
限位设定		软件限位功能
	正方向软件限位	○ *3
	负方向软件限位	○ *3
	位置偏差超限值	○ *3
	位置偏差警告值	○ *3

轴参数分类	轴参数名称	○：写入 / 读取对象 ○：非写入 / 读取对象
位置计数设定	计数模式	○
	循环计数器上限设定值	○
	循环计数器下限设定值	○
	编码器种类	×
伺服驱动器设定	循环计数器上限设定值	×
	循环计数器下限设定值	×
	PDS 状态控制方式 ^{*4}	×
原点复位设定	原点复位动作	○ ^{*3}
	原点输入信号	○ ^{*3}
	原点复位开始方向	○ ^{*3}
	原点检测方向	○ ^{*3}
	正方向极限输入时动作	○ ^{*3}
	负方向极限输入时动作	○ ^{*3}
	原点复位速度	○ ^{*3}
	原点复位接近速度	○ ^{*3}
	原点复位加速度	○ ^{*3}
	原点复位减速度	○ ^{*3}
	原点复位跃度	○ ^{*3}
	原点输入掩码距离	○ ^{*3}
	原点位置偏置	○ ^{*3}
	原点复位推压时间	○ ^{*3}
	原点复位修正值	○ ^{*3}
	原点复位修正速度	○ ^{*3}

*1. 可在 CPU 单元版本 1.11 以上和 Sysmac Studio Ver.1.15 以上的组合中使用。

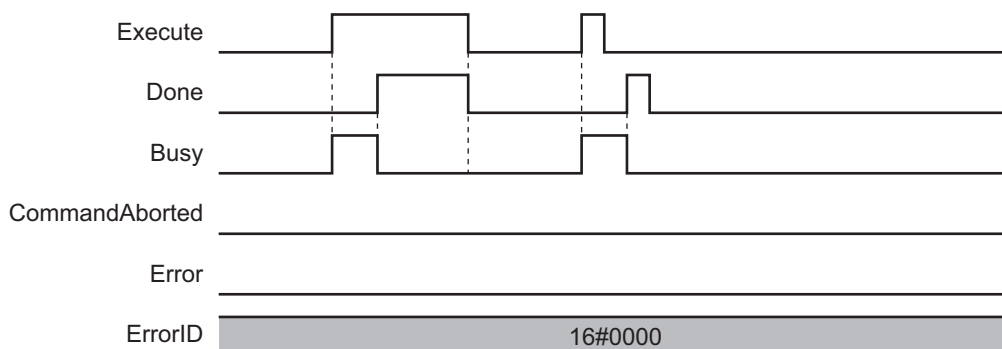
*2. 写入对象轴的“减速机使用”为“使用”且“计数模式”为“旋转模式”时，不会写入“工件每转 1 圈的移动量”的值。| 循环计数器上限值 - 循环计数器下限值 | 的值作为“工件每转 1 圈的移动量”处理。此外，读取对象轴的“减速机使用”为“使用”且“计数模式”为“旋转模式”时，“工件每转 1 圈的移动量”的值将读取为 | 循环计数器上限值 - 循环计数器下限值 | 的值。

*3. 写入对象轴的轴种类为编码器轴或虚拟编码器轴时，不会写入到轴参数。此外，读取对象轴的轴种类为编码器轴或虚拟编码器轴时，将读取轴参数的初始值。轴参数的初始值请参考 □□ 《NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

*4. 可在 CPU 单元版本 1.10 以上和 Sysmac Studio Ver.1.12 以上的组合中指定。

时序图

执行了 MC_WriteAxisParameter（轴参数写入）指令时的时序图如下所示。



运动指令重新执行

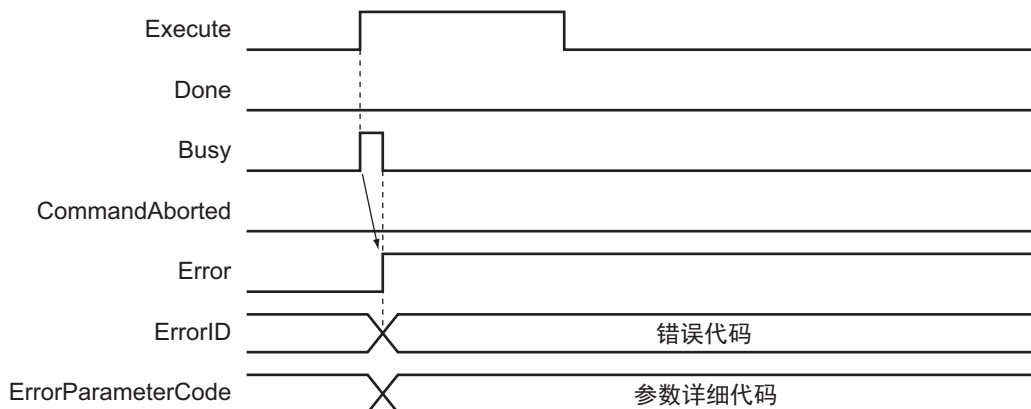
运动指令重新执行的详情请参考 □ □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

运动指令多重启动

运动指令多重启动的详情请参考 □ □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇（SBCE-CN5-363）》。

异常

执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，不写入参数。将保持执行指令前的值。参照输出到 ErrorID（错误代码）的值，可找出异常的原因。此外，有些 ErrorID（错误代码）时，可能会将附属信息输出到 ErrorParameterCode（参数详细代码）中。



● 错误代码

本指令中发生的异常请参考 □ □ 「A-1 错误代码一览 (P.A-2)」。

MC_ReadAxisParameter

读取 MC 功能模块的轴参数设定。

指令	名称	FB/ FUN	图形表现	ST 表现
MC_ReadAxisParameter	轴参数读取	FB		<pre>MC_ReadAxisParameter (Axis := 《参数》, AxisParameter := 《参数》, Execute := 《参数》, Done => 《参数》, Busy => 《参数》, CommandAborted => 《参数》, Error => 《参数》, ErrorID => 《参数》);</pre>



版本相关信息

本指令可在 CPU 单元版本 1.08 以上和 Sysmac Studio Ver.1.09 以上的组合中使用。

变量

输入变量

输入变量	名称	数据类型	有效范围	初始值	内容
Execute	启动	BOOL	TRUE、 FALSE	FALSE	上升时开始指令。

输出变量

输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE、 FALSE	指令执行完成时，变为 TRUE。
Busy	执行中	BOOL	TRUE、 FALSE	收到指令时变为 TRUE。
CommandAborted	执行中断	BOOL	TRUE、 FALSE	中止了指令时变为 TRUE。
Error	错误	BOOL	TRUE、 FALSE	发生异常时变为 TRUE。
ErrorID	错误代码	WORD	*	发生异常时，输出错误代码。16#0000 表示正常。

* 请参考「A-1 错误代码一览 (P.A-2)」。

● 输出变量的反映时序

变量	变为 TRUE 时	变为 FALSE 时
Done	指令完成时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Busy	Execute 的上升沿	<ul style="list-style-type: none"> Done 变为 TRUE 时 Error 变为 TRUE 时 CommandAborted 变为 TRUE 时
CommandAborted	因发生异常, 本指令中止时	<ul style="list-style-type: none"> Execute 为 TRUE 时, 与 Execute 的 FALSE 同时 Execute 为 FALSE 时, 1 个周期后
Error	本指令的启动条件或输入参数中包含异常原因时	解除了异常时

输入输出变量

输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	—	指定读取对象的轴。*1
AxisParameter	轴参数	_sAXIS_PARAM	—	保存读取的值。*2

*1. 请使用 Sysmac Studio 的轴基本设定画面中创建的用户定义变量的轴变量名称（默认“MC_Axis****”）或系统定义变量的轴变量名称（_MC_AX[*], _MC1_AX[*], _MC2_AX[*]）。

*2. 请定义 _sAXIS_PARAM 型用户定义变量。

● _sAXIS_PARAM

成员变量	参数名称	数据类型 *1	功能
UnitConversion	单位转换设定	_sAXIS_UNIT_CONVERSION_SETTINGS	读取的单位转换设定值将保存到各成员变量中。
Operation	动作设定	_sAXIS_OPERATION_SETTINGS	读取的动作设定值将保存到各成员变量中。
OtherOperation	扩展动作设定	_sAXIS_OTHER_OPERATION_SETTINGS	读取的扩展动作设定值将保存到各成员变量中。
Limit	限位设定	_sAXIS_LIMIT_SETTINGS	读取的限位设定值将保存到各成员变量中。
PosCount	位置计数设定	_sAXIS_POSITION_COUNT_SETTINGS	读取的位置计数设定值将保存到各成员变量中。
Homing	原点复位设定	_sAXIS_HOMING_SETTINGS	读取的原点复位设定值将保存到各成员变量中。
Reserved	(保留区域)	ARRAY[0..255] OF BYTE	—

*1. 数据类型的详情请参考 □□「指令的详情 (P.5-48)」。

功能说明

- 在 Execute（启动）的上升沿，将指定 Axis（轴）的轴参数读取到 AxisParameter（轴参数）中。
- 无论轴参数 Cfg.AxEnable（轴使用）的状态如何，本指令始终可读取轴参数。



使用注意事项

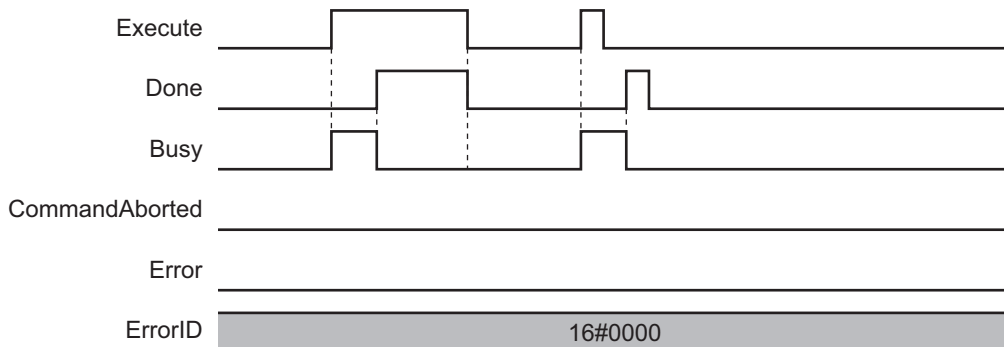
- 可通过本指令读取的值为执行本指令时变为有效的轴参数，并非 CPU 单元的非易失性存储器中保存的值。
例如，使用 MC_Write（MC 设定写入）指令写入轴参数时，将读取所写入的轴参数。
- 本指令和 MC_WriteAxisParameter（轴参数写入）指令配合使用时，请在确认 MC_WriteAxisParameter（轴参数写入）指令已变为 Done（完成）后再启动。
- 仅在本指令完成时读取到 AxisParameter（轴参数）中。之后不会读取到 AxisParameter（轴参数）中。因此，已写入到 AxisParameter（轴参数）时，将被覆盖。

读取对象的轴参数

本指令的读取对象轴参数请参考 □ 「写入或读取对象的轴参数 (P.5-55)」。

时序图

执行了 MC_ReadAxisParameter（轴参数读取）指令时的时序图如下所示。



运动指令重新执行

运动指令重新执行的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

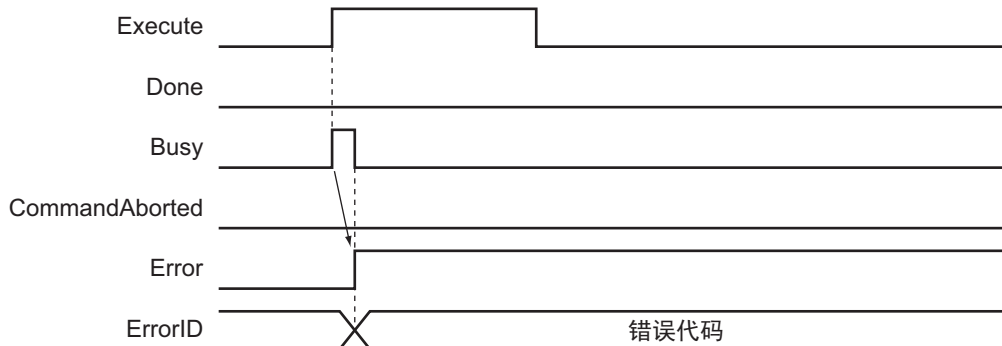
运动指令多重启动

运动指令多重启动的详情请参考 □ 《NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-CN5-363)》。

异常

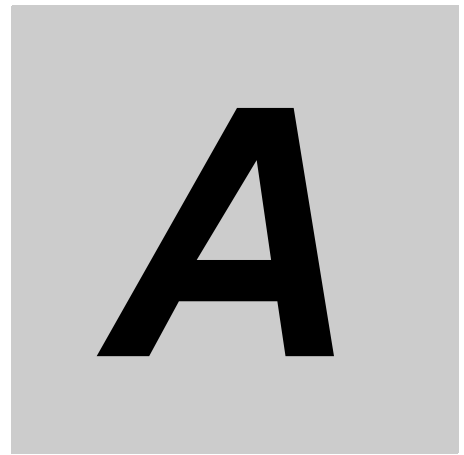
执行本指令的过程中发生异常时，Error（错误）将变为 TRUE，不读取参数。

参照输出到 ErrorID（错误代码）的值，可找出异常的原因。



● 错误代码

本指令中发生的异常请参考 □「A-1 错误代码一览 (P.A-2)」。



附录

指令中发生的错误代码进行说明。

A-1	错误代码一览	A-2
A-2	错误代码详情	A-22
A-2-1	控制器异常说明的解释	A-22
A-2-2	异常说明	A-23
A-3	多重启动可否	A-88
A-3-1	轴的状态、轴组的状态	A-89
A-3-2	状态切换和多重启动的可否	A-91
A-4	版本相关信息	A-96

A

A-1 错误代码一览

下面记载的是执行指令语等时显示的 ErrorID（错误代码）一览表。

下表中事件代码的前四位便是 ErrorID（错误代码）。

错误代码详情，请参考 □「A-2 错误代码详情 (P.A-22)」。

重要程度栏中使用以下简称。

简称	名称
全	全部停止故障
部	部分停止故障
轻	轻度故障
监	监视信息
般	一般信息

事件代码栏的（ ）中表示限定发生该事件的 CPU 单元时，CPU 单元的版本。

关于 NJ/NX 系列的所有事件代码，请参考 □《NJ/NX 系列 故障排除手册（SBCE-CN5-361）》。

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
34610000Hex	过程数据对象设定不足	PDO 映射不正确	<ul style="list-style-type: none"> 没有运动指令所需的 PDO 映射 对不包含支持该指令的对象之目标设备，执行了该指令 对映射了欧姆龙制 EtherCAT 编码器从站 GX-EC02 □□ 的轴，将 Z 相（_mcEncoderMark）指定为触发条件，并启动了运动指令 			○			P.A-23
54200000Hex	超出电子齿轮分子设定范围	运动控制指令的输入变量“RatioNumerator”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-23
54210000Hex	超出电子齿轮分母设定范围	运动控制指令的输入变量“RatioDenominator”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-24
54220000Hex	超出目标速度设定范围	运动控制指令的输入变量“Velocity”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-24

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
54230000Hex	超出加速度设定范围	运动控制指令的输入变量“Acceleration”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-24
54240000Hex	超出减速度设定范围	运动控制指令的输入变量“Deceleration”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-25
54250000Hex	超出跃度设定范围	运动控制指令的输入变量“Jerk”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-25
54270000Hex	超出转矩设定范围	运动控制指令的输入变量“TorqueRamp”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-25
54280000Hex	超出主轴系数设定范围	运动控制指令的输入变量“MasterScaling”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-26
54290000Hex	超出从轴系数设定范围	运动控制指令的输入变量“SlaveScaling”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-26
542A0000Hex	超出固定速度设定范围	运动控制指令的输入变量“FeedVelocity”中指定的参数超出范围	• 保持固定速度（输入变量“FeedVelocity”）的初始值（0）			○			P.A-26
542B0000Hex	超出缓存模式选择范围	运动控制指令的输入变量“BufferMode”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-27
542C0000Hex	超出坐标系选择范围	运动控制指令的输入变量“CoordSystem”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-27
542D0000Hex	超出圆弧插补模式选择范围	运动控制指令的输入变量“CircMode”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-27
542E0000Hex	超出方向选择范围	运动控制指令的输入变量“Direction”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-28

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
542F0000Hex	超出路径选择范围	运动控制指令的输入变量“PathChoice”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-28
54300000Hex	超出位置类型选择范围	运动控制指令的输入变量“ReferenceType”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-28
54310000Hex	超出移动方法选择范围	运动控制指令的输入变量“MoveMode”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-29
54320000Hex	超出转换模式选择范围	运动控制指令的输入变量“TransitionMode”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 将“BufferMode”指定为“_mcAborting”、“_mcBuffered”且将“TransitionMode”指定为“_mcTMCcornerSuperimposed” 			○			P.A-29
54330000Hex	超出持续方向选择范围	变更了运动控制指令的输入变量“Continuous(Reserved)”的值	<ul style="list-style-type: none"> 变更了输入变量“Continuous(Reserved)”的值 			○			P.A-30
54340000Hex	超出加减法方向选择范围	运动控制指令的输入变量“CombineMode”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-30
54350000Hex	超出同步开始条件指定范围	运动控制指令的输入变量“LinkOption”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-30
54360000Hex	主轴从轴相同	运动控制指令的输入变量“Master”和“Slave”中指定的轴相同	<ul style="list-style-type: none"> 指令的输入变量“Master”和“Slave”的参数相同 			○			P.A-31
54370000Hex	主轴辅助轴相同	运动控制指令的输入变量“Master”和“Auxiliary”中指定的轴相同	<ul style="list-style-type: none"> 指令的输入变量“Master”和“Auxiliary”的参数相同 			○			P.A-31
54380000Hex	主轴 / 从轴轴编号非升序	运动控制指令的输入变量“Master”和“Slave”中指定的轴编号不是升序	<ul style="list-style-type: none"> 在指令的输入变量“ReferenceType”中指定“_mcLatestCommand”时，指令的输入变量“Master”和“Slave”的参数不是升序 			○			P.A-31

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
54390000Hex	凸轮表指定不正确	运动控制指令的输入变量“CamTable”中指定的参数超出范围	<ul style="list-style-type: none"> 在指令的输入变量“CamTable”中指定了凸轮数据变量以外的值 			○			P.A-32
543A0000Hex	同步停止中	执行了运动控制的同步控制指令，但并非可执行的条件	<ul style="list-style-type: none"> 执行了 MC_CamOut（解除凸轮动作）指令，但 MC_CamIn（开始凸轮动作）指令不是执行中 执行了 MC_GearOut（解除凸轮动作）指令，但 MC_GearIn（开始凸轮动作）指令、MC_GearInPos（位置指定齿轮动作）指令不是执行中 执行了 MC_Phasing（主轴相对值相位补偿）指令，但 MC_CamIn（开始凸轮动作）指令、MC_GearIn（开始齿轮动作）指令、MC_GearInPos（位置指定齿轮动作）指令、MC_MoveLink（梯模式凸轮）指令不是执行中 			○			P.A-32
543B0000Hex	运动指令不可重新执行	重新执行了无法重新执行的运动控制指令	<ul style="list-style-type: none"> 无法重新执行的运动控制指令的重新执行 			○			P.A-33
543C0000Hex	运动指令不可多重启动	对同一对象（MC 共通 / 轴 / 轴组）执行了多个不可同时执行的功能	<ul style="list-style-type: none"> 对同一对象（MC 共通 / 轴）执行了多个不可同时执行的功能 			○			P.A-33
543D0000Hex	轴种类不合适	对编码器轴执行了动作指令	<ul style="list-style-type: none"> 对编码器轴执行动作指令 			○			P.A-34
543E0000Hex	多轴协调动作中不可启动指令	<ul style="list-style-type: none"> 对多轴协调动作中的轴或轴组执行了动作指令 轴组为有效状态时，执行了不可使用的机器人指令 	<ul style="list-style-type: none"> 对多轴协调动作中的轴或轴组执行动作指令 对轴组为有效状态的轴组执行 MC_SetKinTransform 指令 			○			P.A-34
543F0000Hex	轴组为无效状态时启动多轴协调指令	对轴组为无效状态的轴组，启动了多轴协调指令	<ul style="list-style-type: none"> 对轴组为无效状态的轴组，启动多轴协调指令 对轴组为无效状态的轴组，启动以下指令 MC_MoveTimeAbsolute MC_SyncLinearConveyor MC_SyncOut MC_RobotJog 			○			P.A-35

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
54400000Hex	轴组不可启用	MC_GroupEnable（轴组启用）指令执行失败	<ul style="list-style-type: none"> 执行 MC_GroupEnable（轴组启用）指令时，构成轴中存在未停止的轴 执行 MC_GroupEnable（轴组启用）指令时，构成轴中存在正在执行 MC_TouchProbe（外部锁定有效）指令的轴 			○			P.A-35
54410000Hex	不可运行（伺服 OFF）轴动作指示	对伺服 OFF 中的轴执行了动作指令	<ul style="list-style-type: none"> 对伺服 OFF 中的轴执行动作指令 通过 MC_Home（原点复位）指令或 MC_HomeWithParameter（参数指定原点复位）指令对未建立 EtherCAT 过程数据通信的轴执行了原点复位 			○			P.A-36
54420000Hex	构成轴强制停止中错误	对正在向构成轴执行 MC_Stop（强制停止）指令的轴组执行了动作指令	<ul style="list-style-type: none"> 对正在向构成轴执行 MC_Stop（强制停止）指令的轴组执行了动作指令 			○			P.A-36
54430000Hex	运动指令多重启动数超限	通过缓存模式 Buffered、Blending 缓存的运动控制指令的缓存数超出极限	<ul style="list-style-type: none"> 轴指令中，执行中的指令和缓存中的指令总和超过“2” 轴组指令中，执行中的指令和缓存中的指令总和超过“8” 			○			P.A-37
54440000Hex	移动量不足	定位指令的多重启动 / 重新执行时，无法按指定的减速度或加速度执行指定的动作	<ul style="list-style-type: none"> “加减速超限”设定为“异常停止”时，无法按定位指令的多重启动 / 重新执行中指定的减速度 / 加速度停到目标位置 			○			P.A-37
54450000Hex	达到共混中继速度所需的移动量不足	加减速至中继速度所需的移动量不足	<ul style="list-style-type: none"> “加减速超限”设定为“异常停止”时，将当前指令加减速至中继速度所需的移动量不足 			○			P.A-38
54460000Hex	梯形模式凸轮等速移动量不足	主轴的等速移动量不足“0”	<ul style="list-style-type: none"> MC_MoveLink（梯形模式凸轮）指令中，主轴的等速移动量不足“0” 			○			P.A-38
54470000Hex	位置指定齿轮动作目标速度不足	MC_GearInPos（位置指定齿轮动作）指令中，从轴的“目标速度”较小，因此无法达到需要的速度	<ul style="list-style-type: none"> MC_GearInPos（位置指定齿轮动作）指令中，输入变量“Velocity（目标速度）”的值小于（指令启动时的主轴速度 × 齿轮比） 			○			P.A-38

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
54480000Hex	圆弧插补起点终点相同	MC_MoveCircular2D（2轴圆弧插补）指令中，指定半径指定方式时，起点和终点为相同位置或者指定通过点指定方式时，起点、终点和通过点为相同位置	<ul style="list-style-type: none"> MC_MoveCircular2D（2轴圆弧插补）指令中，指定半径指定方式，起点和终点的位置相同 MC_MoveCircular2D（2轴圆弧插补）指令中，指定通过点指定方式，起点、终点和通过点的位置相同 			○			P.A-39
54490000Hex	超出圆弧插补中心点指定位置范围	MC_MoveCircular2D（2轴圆弧插补）指令中，指定中心点指定方式时，中心点的位置指定超出容许范围	<ul style="list-style-type: none"> MC_MoveCircular2D（2轴圆弧插补）指令中，指定中心点指定方式，起点和中心点的距离、终点和中心点的距离之差超出轴组设定的“中心点修正容许率”中指定的容许范围 			○			P.A-39
544A0000Hex	计数模式设定引起的指令启动异常	对计数模式设定为旋转模式的轴执行了旋转模式下无法使用的指令	<ul style="list-style-type: none"> 在旋转模式下无法使用的指令中，使用计数模式设定为旋转模式的轴 			○			P.A-40
544C0000Hex	超出参数选择范围	运动控制指令的输入变量“ParameterNumber”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-40
544D0000Hex	超出停止方法选择范围	运动控制指令的输入变量“StopMode”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-40
544E0000Hex	超出触发输入条件的锁定ID选择范围	运动控制指令的输入变量“TriggerInput::LatchID”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-41
544F0000Hex	超出MC设定写入的设定范围	运动控制指令的输入变量“SettingValue”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 参数指定和设定值的数据类型不一致 			○			P.A-41
54500000Hex	超出触发输入条件的模式选择范围	运动控制指令的输入变量“TriggerInput::Mode”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-41
54510000Hex	超出触发输入条件的驱动触发输入信号选择范围	运动控制指令的输入变量“TriggerInput::InputDrive”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-42

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
54530000Hex	运动指令不可重新执行（轴指定）	重新执行运动控制指令时，变更了不可变更的输入变量“Axis”的参数	• 重新执行指令时，变更了不可变更的输入变量的参数			○			P.A-42
54540000Hex	运动指令不可重新执行（缓存模式选择）	重新执行运动控制指令时，变更了不可变更的输入变量“BufferMode”的参数	• 重新执行指令时，变更了不可变更的输入变量的参数			○			P.A-42
54550000Hex	运动指令不可重新执行（方向选择）	重新执行运动控制指令时，变更了不可变更的输入变量“Direction”的参数	• 重新执行指令时，变更了不可变更的输入变量的参数			○			P.A-43
54560000Hex	运动指令不可重新执行（重复模式）	重新执行运动控制指令时，变更了不可变更的输入变量“Periodic”的参数	• 重新执行指令时，变更了不可变更的输入变量的参数			○			P.A-43
54570000Hex	运动指令不可重新执行（轴组指定）	重新执行运动控制指令时，变更了不可变更的输入变量“AxesGroup”的参数	• 重新执行指令时，变更了不可变更的输入变量的参数			○			P.A-43
54580000Hex	运动指令不可重新执行（跃动设定）	重新执行运动控制指令时，变更了不可变更的输入变量“Jerk”的参数	• 重新执行指令时，变更了不可变更的输入变量的参数			○			P.A-44
54590000Hex	运动指令不可重新执行（主轴）	重新执行运动控制指令时，变更了不可变更的输入变量“Master”的参数	• 重新执行指令时，变更了不可变更的输入变量的参数			○			P.A-44
545A0000Hex	运动指令不可重新执行（Master Offset）	重新执行运动控制指令时，变更了不可变更的输入变量“MasterOffset”的参数	• 重新执行指令时，变更了不可变更的输入变量的参数			○			P.A-45
545B0000Hex	运动指令不可重新执行（Master Scaling）	重新执行运动控制指令时，变更了不可变更的输入变量“MasterScaling”的参数	• 重新执行指令时，变更了不可变更的输入变量的参数			○			P.A-45

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
545C0000Hex	运动指令不可重新执行（Master Start Distance）	重新执行运动控制指令时，变更了不可变更的输入变量“MasterStart Distance”的参数	• 重新执行指令时，变更了不可变更的输入变量的参数			○			P.A-45
545D0000Hex	运动指令不可重新执行（Continuou s）	重新执行运动控制指令时，变更了不可变更的输入变量“Continuous”的参数	• 重新执行指令时，变更了不可变更的输入变量的参数			○			P.A-46
545E0000Hex	运动指令不可重新执行（Move Mode）	重新执行运动控制指令时，变更了不可变更的输入变量“MoveMode”的参数	• 重新执行指令时，变更了不可变更的输入变量的参数			○			P.A-46
545F0000Hex	辅助轴指定不正确	运动控制指令的输入变量“Auxiliary”中指定的轴不存在	• 指令的输入变量“Auxiliary”中指定的轴为不存在的变量			○			P.A-46
54600000Hex	轴指定不正确	运动控制指令的输入变量“Axis”中指定的轴不存在	• 指令的输入变量“Axis”中指定的轴为不存在的变量			○			P.A-47
54610000Hex	轴组指定不正确	运动控制指令的输入变量“AxesGroup”中指定的轴组不存在或不是使用轴组	• 指令的“AxesGroup”中指定的轴组为不存在的变量 • 指令的“AxesGroup”中指定的轴组未设定为使用轴组			○			P.A-47
54620000Hex	主轴指定不正确	运动控制指令的输入变量“Master”中指定的轴不正确	• 指令的输入变量“Master”中指定的轴为不存在的变量 • MC_Phasing（主轴相对值相位补偿）指令时，输入变量“Master”中指定轴不是同步主轴 • 分配了主轴和从轴的任务不是同一任务			○			P.A-48
54630000Hex	运动指令不可重新执行（Slave Offset）	重新执行运动控制指令时，变更了不可变更的输入变量“SlaveOffset”的参数	• 重新执行指令时，变更了不可变更的输入变量的参数			○			P.A-48
54640000Hex	运动指令不可重新执行（Slave Scaling）	重新执行运动控制指令时，变更了不可变更的输入变量“SlaveScaling”的参数	• 重新执行指令时，变更了不可变更的输入变量的参数			○			P.A-49

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
54650000Hex	运动指令不可重新执行（Start Position）	重新执行运动控制指令时，变更了不可变更的输入变量“StartPosition”的参数	<ul style="list-style-type: none"> 重新执行指令时，变更了不可变更的输入变量的参数 			○			P.A-49
54660000Hex	原点未确定状态下指令启动异常	在原点未确定状态下执行了高速原点复位或插补指令	<ul style="list-style-type: none"> 在原点未确定状态下执行了高速原点复位 对包含原点未确定状态的构成轴执行了插补指令 对包含原点未确定状态的逻辑轴的轴组执行了以下机器人指令 MC_SetKinTransform MC_MoveTimeAbsolute MC_SyncLinearConveyor MC_SyncOut MC_GroupMon MC_RobotJog 			○			P.A-50
54670000Hex	运动指令不可重新执行（位置类型）	重新执行运动控制指令时，变更了不可变更的输入变量“ReferenceType”的参数	<ul style="list-style-type: none"> 重新执行指令时，变更了不可变更的输入变量的参数 			○			P.A-50
54680000Hex	未使用轴指定（主轴）	运动控制指令中指定的主轴为未使用轴	<ul style="list-style-type: none"> 运动控制指令中指定的主轴为未使用轴 			○			P.A-51
54690000Hex	超出开始位置设定范围	运动控制指令的输入变量“FirstPosition”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-51
546A0000Hex	超出结束位置设定范围	运动控制指令的输入变量“LastPosition”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-51
546B0000Hex	开始位置 / 结束位置、大小关系不正确（线性模式）	运动控制指令的输入变量“LastPosition”中指定的参数值小于输入变量“FirstPosition”中指定的参数	<ul style="list-style-type: none"> 计数模式为线性模式时，指令的输入参数中，“LastPosition”的值小于“FirstPosition”的值 			○			P.A-52
546C0000Hex	超出主轴同步位置设定范围	运动控制指令的输入变量“MasterSync Position”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-52

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
546D0000Hex	超出从轴同步位置设定范围	运动控制指令的输入变量“Slave SyncPosition”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-52
546E0000Hex	触发输入条件的锁定 ID 重复	运动控制指令中指定的锁定 ID 在多个指令中重复	<ul style="list-style-type: none"> 在 MC_TouchProbe（外部锁定有效）指令、MC_MoveLink（梯形模式凸轮）指令、MC_MoveFeed（中断固定尺寸定位）指令中同时使用了相同的锁定 ID 试图以 MC_AbortTrigger（外部锁定无效）指令中止 MC_TouchProbe（外部锁定有效）指令以外正在使用的锁定 			○			P.A-53
546F0000Hex	超出跃度超驰值范围	运动控制指令的输入变量“JerkFactor”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-53
54700000Hex	超出加减速超驰值范围	运动控制指令的输入变量“AccFactor”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-54
54710000Hex	超出开始位置方式指定范围	运动控制指令的输入变量“StartMode”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-54
54720000Hex	运动指令不可重新执行（开始位置方式）	重新执行运动控制指令时，变更了不可变更的输入变量“StartMode”的参数	<ul style="list-style-type: none"> 重新执行指令时，变更了不可变更的输入变量的参数 			○			P.A-54
54740000Hex	未使用轴指定（辅助轴）	运动控制指令的输入变量“Auxiliary”中指定的轴为未使用轴	<ul style="list-style-type: none"> 指令的“Auxiliary”中指定的轴为未使用轴 			○			P.A-55
54750000Hex	位置指定齿轮指定值异常	无法以运动控制指令中输入的速度/加速度/减速度进行同步动作	<ul style="list-style-type: none"> 无法以指令中输入的速度/加速度/减速度进行指定的同步动作 			○			P.A-55
54760000Hex	位置指定齿轮主轴速度零	启动运动控制指令时，主轴的速度为“0”	<ul style="list-style-type: none"> 启动指令时，主轴的速度为“0” 			○			P.A-55
54780000Hex	超出目标位置设定范围	运动控制指令的输入变量“Position”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 在旋转模式的轴中，目标位置超出链接设定范围 			○			P.A-56

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
54790000Hex	超出移动距离范围	运动控制指令的输入变量“Distance”中指定的参数超出范围或加上“Distance”的值后目标位置超出范围	<ul style="list-style-type: none"> 将指令的输入参数绝对值转换为脉冲单位时，超出 40 位的范围 在线性模式的轴中，将加上移动距离后的目标位置转换为脉冲单位时，带符号超出 40 位的范围 			○			P.A-56
547A0000Hex	超出凸轮表起点位置设定范围	运动控制指令的输入变量“StartPosition”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-57
547B0000Hex	超出凸轮动作（主轴追踪）开始位置设定范围	运动控制指令的输入变量“MasterStart Distance”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-57
547C0000Hex	圆弧插补半径指定异常	MC_MoveCircular2D（2轴圆弧插补）指令中，指定半径指定方式时，无法以指定的半径创建圆弧轨迹	<ul style="list-style-type: none"> MC_MoveCircular2D（2轴圆弧插补）指令中，指定为半径指定方式，但无法以指定的半径创建圆弧轨迹 			○			P.A-57
547D0000Hex	圆弧插补半径溢出	MC_MoveCircular2D（2轴圆弧插补）指令中，指定为通过点指定方式 / 中心点指定方式时，圆弧半径超出最大值	<ul style="list-style-type: none"> MC_MoveCircular2D（2轴圆弧插补）指令中，指定为通过点指定方式 / 中心点指定方式时，将圆弧半径转换为脉冲单位后，超出 40 位的范围 			○			P.A-58
547E0000Hex	超出圆弧轴指定范围	运动控制指令的输入变量“CircAxes”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 “CircAxes”中指定的轴不在轴组设定的构成轴中 “CircAxes”的2个轴中指定了同一个轴 			○			P.A-58
547F0000Hex	辅助轴 / 从轴、轴编号非升序	运动控制指令中指定的输入变量“Auxiliary”和“Slave”的参数值未按升序排列	<ul style="list-style-type: none"> 指令的输入变量“Auxiliary”和“Slave”的参数非升序 			○			P.A-59
54800000Hex	凸轮表属性更新中数据升序异常	在计算有效数据的过程中，发现相位未按升序排列或者得到计算结果后发现有效数据数为“0”	<ul style="list-style-type: none"> 在计算有效数据的过程中，发现相位未按升序排列 得到计算结果后发现有效数据数为“0” 			○			P.A-59

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
54810000Hex	超出 MC 设定写入的对象范围	运动控制指令的输入变量“Target”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-59
54820000Hex	超出主轴移动距离指定范围	运动控制指令的输入变量“MasterDistance”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-60
54830000Hex	超出主轴加速移动距离指定范围	运动控制指令的输入变量“MasterDistanceIn ACC”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-60
54840000Hex	超出主轴减速移动距离指定范围	运动控制指令的输入变量“MasterDistanceIn DEC”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-60
54870000Hex	超出执行模式选择范围	运动控制指令的输入变量“ExecutionMode”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-61
54880000Hex	超出轴间偏差容许值范围	运动控制指令的输入变量“PermittedDeviation”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-61
54890000Hex	超出通过点位置 / 中心位置 / 半径指定范围	运动控制指令的输入变量“AuxPoint”中指定的参数超出范围	<ul style="list-style-type: none"> 通过点指定或中心指定时，将“AuxPoint”的值转换为脉冲单位时，带符号超出 40 位的范围 半径指定时，将“AuxPoint[0]”的绝对值转换为脉冲单位时，超出 40 位的范围 			○			P.A-61
548A0000Hex	超出终点指定范围	运动控制指令的输入变量“EndPoint”中指定的参数超出范围	<ul style="list-style-type: none"> 将指令的输入参数转换为脉冲单位时，带符号超出 40 位的范围 			○			P.A-62
548B0000Hex	超出从轴移动距离指定范围	运动控制指令的输入变量“SlaveDistance”中指定的参数超出范围	<ul style="list-style-type: none"> 将指令的输入参数转换为脉冲单位时，超出 40 位的范围 			○			P.A-62

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
548C0000Hex	超出相位补偿量范围	运动控制指令的输入变量“PhaseShift”中指定的参数超出范围	• 将指令的输入参数绝对值转换为脉冲单位时，超出 40 位的范围			○			P.A-62
548D0000Hex	超出固定距离范围	运动控制指令的输入变量“FeedDistance”中指定的参数超出范围	• 将指令的输入参数绝对值转换为脉冲单位时，超出 40 位的范围			○			P.A-63
548E0000Hex	辅助轴 / 从轴相同	运动控制指令的输入变量“Auxiliary”和“Slave”中指定的轴相同	• 指令的输入变量“Auxiliary”和“Slave”的参数相同			○			P.A-63
548F0000Hex	超出相对位置选择范围	运动控制指令的输入变量“Relative”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-63
54900000Hex	超出凸轮转换指定选择范围	运动控制指令的输入变量“CamTransition”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-64
54910000Hex	超出同步控制解除模式选择范围	运动控制指令的输入变量“OutMode”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-64
54920000Hex	不可执行外部锁定有效指令	针对编码器轴，将输入变量“StopMode”指定为“_mclImmediateStop（立即停止）”，并执行了驱动器模式的MC_TouchProbe（外部锁定有效）指令	• 针对编码器轴，将输入变量“StopMode”指定为“_mclImmediateStop（立即停止）”，并执行了驱动器模式的MC_TouchProbe（外部锁定有效）指令			○			P.A-64
54930000Hex	超出主轴偏置设定范围	运动控制指令的输入变量“MasterOffset”中指定的参数超出范围	• 将指令的输入参数转换为脉冲单位时，带符号超出 40 位的范围			○			P.A-65
54940000Hex	超出从轴偏置设定范围	运动控制指令的输入变量“SlaveOffset”中指定的参数超出范围	• 将指令的输入参数转换为脉冲单位时，带符号超出 40 位的范围			○			P.A-65
54950000Hex	超出指令当前位置计数选择范围	运动控制指令的输入变量“CmdPosMode”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-65

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
54960000Hex	超出主轴齿轮比分子范围	运动控制指令的输入变量 “RatioNumeratorMaster”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-66
54970000Hex	超出主轴齿轮比分母范围	运动控制指令的输入变量 “RatioDenominatorMaster”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-66
54980000Hex	超出辅助轴齿轮比分子范围	运动控制指令的输入变量 “RatioNumeratorAuxiliary”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-66
54990000Hex	超出辅助轴齿轮比分母范围	运动控制指令的输入变量 “RatioDenominatorAuxiliary”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-67
549A0000Hex	超出主轴位置类型选择范围	运动控制指令的输入变量 “ReferenceTypeMaster”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-67
549B0000Hex	超出辅助轴位置类型选择范围	运动控制指令的输入变量 “ReferenceTypeAuxiliary”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-67
549C0000Hex	超出目标位置链接计数器范围	所执行指令的目标位置超出链接计数器范围，因此无法运行	• 在链接计数器范围不含0的设定下，执行了高速原点复位			○			P.A-68
549D0000Hex (Ver.1.01 以上)	超出轴组构成轴设定范围	运动控制指令的输入变量“Axes”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围 • 分配了轴组之构成轴的任务不是同一任务			○			P.A-68
549E0000Hex (Ver.1.04 以上)	超出轴使用设定范围	运动控制指令的输入变量“AxisUse”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-69
57000000Hex (Ver.1.03 以上)	超出原点复位参数设定范围	运动控制指令的输入变量 “HomingParameter”中指定的参数超出范围	• 指令的输入参数超出输入变量的范围			○			P.A-69

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
57020000Hex (Ver.1.04 以上)	轴未使用切换异常	对非停止中或指令速度已饱和的轴执行了 MC_ChangeAxisUse（轴使用变更）指令	<ul style="list-style-type: none"> 对非停止中或指令速度已饱和的轴执行了 MC_ChangeAxisUse（轴使用变更）指令 			○			P.A-70
57030000Hex (Ver.1.06 以上)	轴使用不可切换	执行了 MC_ChangeAxisUse（轴使用变更）指令，超出使用实轴最大数	<ul style="list-style-type: none"> 执行了 MC_ChangeAxisUse（轴使用变更）指令，超出使用实轴最大数 			○			P.A-70
57200000Hex (Ver.1.04 以上)	切换轴使用时运动控制参数设定异常	切换为使用轴的轴运动控制参数设定不正确	<ul style="list-style-type: none"> 通过 MC_ChangeAxisUse（轴使用变更）指令，将未使用轴切换为使用轴的轴运动控制参数设定不正确 运动控制参数设定的下载过程中断电 非易失性存储器故障或非易失性存储器已到使用寿命 			○			P.A-71
57210000Hex (Ver.1.04 以上)	切换轴使用时必需过程数据对象未设定	切换为使用轴的轴种类中，必需对象未设定	<ul style="list-style-type: none"> 切换为使用轴的轴种类中，必需对象未在 PDO 映射设定中设定 运动控制参数设定的下载过程中断电 非易失性存储器故障或非易失性存储器已到使用寿命 对 [轴使用] 设定为 [未使用轴（不可切换为使用轴）] 的轴，执行了 MC_ChangeAxisUse（轴使用变更）指令 			○			P.A-72
572F0000Hex (Ver.1.06 以上)	运动指令不可多重启动（主轴）	指令多重启动时，变更了不可变更的输入输出变量“Master”	<ul style="list-style-type: none"> 指令多重启动时，变更不可变更的输入输出变量“Master” 			○			P.A-72
57300000Hex (Ver.1.06 以上)	运动指令不可多重启动（位置类型选择）	指令多重启动时，变更了不可变更的输入输出变量“ReferenceType”	<ul style="list-style-type: none"> 指令多重启动时，变更不可变更的输入输出变量“ReferenceType” 			○			P.A-73
573A0000Hex (Ver.1.08 以上)	轴参数不可写入	对未使用轴以外的轴执行了该指令	<ul style="list-style-type: none"> 对使用轴或未创建轴执行了该指令 			○			P.A-73
573B0000Hex (Ver.1.08 以上)	超出轴参数设定范围	运动控制指令的输入变量“AxisParameter”中指定的参数超出有效范围	<ul style="list-style-type: none"> 指令的输入变量“AxisParameter”中指定的参数超出输入变量的范围 			○			P.A-74

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
573C0000Hex (Ver.1.08 以上)	超出凸轮属性设定范围	运动控制指令的输入变量“CamProperty”中指定的参数超出有效范围	<ul style="list-style-type: none"> 指令的输入变量“CamProperty”中指定的参数超出输入变量的范围 			○			P.A-75
573D0000Hex (Ver.1.08 以上)	超出凸轮节点设定范围	运动控制指令的输入变量“CamNodes”中指定的参数超出有效范围	<ul style="list-style-type: none"> 指令的输入变量“CamNodes”中指定的参数超出输入变量的范围 			○			P.A-75
573E0000Hex (Ver.1.08 以上)	凸轮节点类型指定不正确	运动控制指令的输入变量“CamNodes”中指定的参数不是_sMC_CAM_NODE 型排列变量	<ul style="list-style-type: none"> 指令的输入变量“CamNodes”中指定的参数不是_sMC_CAM_NODE 型排列变量 			○			P.A-76
573F0000Hex (Ver.1.08 以上)	凸轮表生成节点数不足	运动控制指令的输入变量“CamNodes”中指定的参数之排列变量中，要素编号 0 的 Phase 值为“0”	<ul style="list-style-type: none"> 指令的输入变量“CamNodes”中指定的参数之排列变量中，要素编号 0 的 Phase（主轴相位）值为“0” 			○			P.A-76
57400000Hex (Ver.1.08 以上)	凸轮节点主轴相位非升序	运动控制指令的输入变量“CamNodes”中指定的参数之排列变量中，Phase 的值未按要素编号顺序升序排列	<ul style="list-style-type: none"> 指令的输入变量“CamNodes”中指定的参数之排列变量中，Phase（主轴相位）的值未按要素编号顺序升序排列。或者有效位数第 8 位及之后的舍去结果不是升序 			○			P.A-76
57410000Hex (Ver.1.08 以上)	凸轮表生成数据数过多	生成的凸轮数据数超出运动控制指令的输入变量“CamTable”中指定的凸轮数据变量的排列要素数	<ul style="list-style-type: none"> 所生成凸轮表的凸轮数据数超出指令的输入变量“CamTable”中指定的凸轮数据变量的排列要素数 			○			P.A-77
57420000Hex (Ver.1.08 以上)	凸轮表生成位移溢出	所生成凸轮表的 Distance 超出 REAL 型可表现的范围	<ul style="list-style-type: none"> 所生成凸轮表的 Distance 超出 REAL 型可表现的范围 			○			P.A-77
57430000Hex (Ver.1.08 以上)	生成中止凸轮表的使用	在该指令的输入变量“CamTable”中指定了中止生成的凸轮数据变量	<ul style="list-style-type: none"> 因 MC_GenerateCamTable（凸轮表生成）指令异常而中止生成的凸轮数据变量，指定为了该指令的输入变量“CamTable” 			○			P.A-78
57490000Hex (Ver.1.10 以上)	超出执行 ID 设定范围	运动控制指令的输入变量“ExecID”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入变量“ExecID”中指定的参数超出输入变量的范围 			○			P.A-78

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
574A0000Hex (Ver.1.10 以上)	超出位置偏置范围	运动控制指令的输入变量“OffsetPosition”中指定的参数超出范围	<ul style="list-style-type: none"> 将位置偏置转换为脉冲单位时，带符号超出 40 位的范围 			○			P.A-79
574B0000Hex (Ver.1.10 以上)	超出 PDS 状态变化指令选择范围	运动控制指令的输入变量“TransitionCmd”中指定的参数超出范围	<ul style="list-style-type: none"> 指令的输入参数超出输入变量的范围 			○			P.A-79
64400000Hex	目标位置正方向软件超限	指定的位置超出正方向软件限位	<ul style="list-style-type: none"> 指令的输入变量“Position”中指定的参数超出正方向软件限位 开始位置超出正方向软件限位的范围，执行了向软件限位范围相反方向动作的指令 设定为通过点指定的 MC_MoveCircular2D（2 轴圆弧插补）指令的输入变量“AuxPoint”中指定的参数超出正方向软件限位范围 			○			P.A-80
64410000Hex	目标位置负方向软件超限	指定位置超出负方向软件限位范围	<ul style="list-style-type: none"> 指令的输入变量“Position”中指定的参数超出负方向软件限位范围 开始位置超出负方向软件限位的范围，执行了向软件限位范围相反方向动作的指令 设定为通过点指定的 MC_MoveCircular2D（2 轴圆弧插补）指令的输入变量“AuxPoint”中指定的参数超出负方向软件限位范围 			○			P.A-81
64420000Hex	正在发生指令位置溢出 / 下溢	正在发生指令位置溢出 / 下溢时，执行了定位、向溢出 / 下溢方向移动的指令或方向不确定的指令	<ul style="list-style-type: none"> 正在发生指令位置溢出/下溢时执行了以下指令 执行了定位指令 执行了向溢出/下溢方向移动的连续控制指令 执行了方向不确定的指令（同步功能、扭矩控制） 			○			P.A-81

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
64430000Hex	正方向极限输入中	在正方向极限输入为“ON”的状态下执行了向正方向移动的指令	<ul style="list-style-type: none"> 在正方向极限输入为“ON”的状态下执行了向正方向移动的指令或在正方向极限输入为“ON”的状态下执行了未指定移动方向的指令在正方向极限输入为“ON”的状态下执行了轴组动作指令 			○			P.A-82
64440000Hex	负方向极限输入中	在负方向极限输入为“ON”的状态下执行了向负方向移动的指令	<ul style="list-style-type: none"> 在负方向极限输入为“ON”的状态下执行了向负方向移动的指令或在负方向极限输入为“ON”的状态下执行了未指定移动方向的指令在负方向极限输入为“ON”的状态下执行了轴组动作指令 			○			P.A-82
74220000Hex	伺服主回路电源关闭状态	在伺服驱动器的主回路电源为“OFF”的状态下执行了伺服 ON	<ul style="list-style-type: none"> 在伺服驱动器的主回路电源为“OFF”的状态下执行了伺服 ON 			○			P.A-83
57220000Hex (Ver.1.06 以上)	正在发生反馈位置溢出 / 下溢	正在发生反馈位置溢出 / 下溢时，启动了无法执行的指令	<ul style="list-style-type: none"> 正在发生反馈位置溢出或下溢时，启动了无法执行的指令 				○		P.A-83
57230000Hex (Ver.1.06 以上)	超出切换结构体的轨道编号设定范围	运动控制指令的输入输出变量“Switches”中指定的“TrackNumber”的值超出范围	<ul style="list-style-type: none"> 指令的输入输出变量中指定的结构体型变量的成员值超出范围 				○		P.A-83
57240000Hex (Ver.1.06 以上)	超出切换结构体的 ON 开始位置设定范围	运动控制指令的输入输出变量“Switches”中指定的“FirstOnPosition”的值超出范围	<ul style="list-style-type: none"> 指令的输入输出变量中指定的结构体型变量的成员值超出范围 				○		P.A-84
57250000Hex (Ver.1.06 以上)	超出切换结构体的 ON 结束位置设定范围	运动控制指令的输入输出变量“Switches”中指定的“LastOnPosition”的值超出范围	<ul style="list-style-type: none"> 指令的输入输出变量中指定的结构体型变量的成员值超出范围 				○		P.A-84
57260000Hex (Ver.1.06 以上)	超出切换结构体的方向选择范围	运动控制指令的输入输出变量“Switches”中指定的“AxisDirection”的值超出范围	<ul style="list-style-type: none"> 指令的输入输出变量中指定的结构体型变量的成员值超出范围 				○		P.A-84

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
57270000Hex (Ver.1.06 以上)	超出切换结构体的切换模式选择范围	运动控制指令的输入输出变量“Switches”中指定的“CamSwitchMode”的值超出范围	• 指令的输入输出变量中指定的结构体型变量的成员值超出范围				○		P.A-85
57280000Hex (Ver.1.06 以上)	超出切换结构体的 ON 时间设定范围	运动控制指令的输入输出变量“Switches”中指定的“Duration”的值超出范围	• 指令的输入输出变量中指定的结构体型变量的成员值超出范围				○		P.A-85
57290000Hex (Ver.1.06 以上)	超出轨道选项结构体的 ON 时刻修正设定范围	运动控制指令的输入输出变量“TrackOptions”中指定的“OnCompensation”的值超出范围	• 指令的输入输出变量中指定的结构体型变量的成员值超出范围				○		P.A-85
572A0000Hex (Ver.1.06 以上)	超出轨道选项结构体的 OFF 时刻修正设定范围	运动控制指令的输入输出变量“TrackOptions”中指定的“OffCompensation”的值超出范围	• 指令的输入输出变量中指定的结构体型变量的成员值超出范围				○		P.A-86
572B0000Hex (Ver.1.06 以上)	超出切换结构体型变量的排列要素数范围	运动控制指令的输入输出变量“Switches”中指定的结构体型变量的排列要素数超出范围	• 指令的输入输出变量中指定的结构体型变量的排列要素数超出范围				○		P.A-86
572C0000Hex (Ver.1.06 以上)	超出输出信号结构体型变量的排列要素数范围	运动控制指令的输入输出变量“Outputs”中指定的结构体型变量的排列要素数超出范围	• 指令的输入输出变量中指定的结构体型变量的排列要素数超出范围				○		P.A-86
572D0000Hex (Ver.1.06 以上)	超出轨道选项结构体型变量的排列要素数范围	运动控制指令的输入输出变量“TrackOptions”中指定的结构体型变量的排列要素数超出范围	• 指令的输入输出变量中指定的结构体型变量的排列要素数超出范围				○		P.A-87
572E0000Hex (Ver.1.06 以上)	输出信号和轨道选项的排列要素数不一致	运动控制指令的输入输出变量“Outputs”和“TrackOptions”中指定的结构体型变量的排列要素数不一致	• 指令的输入输出变量中指定的输出信号结构体型变量和轨道选项结构体型变量的排列要素数不一致				○		P.A-87

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
5731000Hex (Ver.1.06 以上)	超出切换结构体的同一轨道指定数范围	运动控制指令的输入输出变量“Switches”中指定的“TrackNumber”的同一轨道编号数超出范围	<ul style="list-style-type: none"> 指令的输入输出变量中指定的切换结构体变量的“TrackNumber”中，指定的同一轨道编号数超出 1 个轨道上可指定的个数 				○		P.A-87

A-2 错误代码详情

A-2-1 控制器异常说明的解释

[] 内表示表中各项目的含义，用于解释各种异常。

事件名称	[异常（事件）的名称]		事件代码	[异常（事件）的代码]		
内容	[异常（事件）的内容]					
发生位置	[发生异常（事件）的位置]		发生位置详情	[发生位置的详细信息]	检测时序	[检测异常的时序]
异常属性	重要程度	[对控制的影响程度] ^{*1}	恢复方法	[恢复方法] ^{*2}	日志类别	[所保存日志的种类] ^{*3}
产生的影响	用户程序	[用户程序的执行状态] ^{*4}	动作	[与发生异常（事件）时的动作相关的特殊事项]		
LED	内置 EtherNet/IP 端口用 LED、内置 EtherCAT 端口用 LED 的显示状态。仅当发生位置为 EtherCAT 主机功能模块、EtherNet/IP 功能模块时记载					
系统定义变量	变量名称		数据类型		名称	
	[检测异常的系统定义变量、受异常影响的系统定义变量、引起异常的系统定义变量的变量名称和数据类型、名称]					
发生原因和处理措施	发生原因（推测原因）		处理对策		防止再发	
	[异常（时间）的发生原因、处理对策和防止再发方法]					
附属信息	[Sysmac Studio / NS 系列显示器上显示的附属信息内容] ^{*5}					
注意事项 / 备注	[其他注意事项、限制事项、补充说明等]					

*1. 以下任一。

全部停止故障：全部停止故障等级
 部分停止故障：部分停止故障等级
 轻度故障：轻度故障等级
 监视信息
 一般信息

*2. 以下任一。

自动恢复：排除原因后，自动恢复正常。
 异常解除：排除原因并执行异常解除后恢复正常。
 重新接通电源：排除原因并重新接通电源后恢复正常。
 控制器重置：排除原因并重置控制器后恢复正常。
 取决于发生原因：根据发生原因而定。

*3. 以下任一。

系统日志：系统事件日志
 访问日志：访问事件日志

*4. 以下任一。

继续：继续执行用户程序。
 停止：停止执行用户程序。
 开始：开始执行用户程序。

*5. 关于显示器的故障排除应对范围，请参考 □ 《NJ/NX 系列 故障排除手册（SBCE-CN5-361）》的附录。



使用注意事项

单元版本 Ver.1.10 以上的 CPU 单元时，以 `_MC_AX[*]` 开始的变量名称可能是 `_MC1_AX[*]`、`_MC2_AX[*]`。同样，以 `_MC_GRP[*]` 开始的变量名称可能是 `_MC1_GRP[*]`、`_MC2_GRP[*]`。

A-2-2 异常说明

下面记载的是执行指令语等时显示的 ErrorID（错误代码）详情。

下表中事件代码的前四位便是 ErrorID（错误代码）。

事件名称	过程数据对象设定不足		事件代码	34610000Hex	
内容	PDO 映射不正确				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴不可动作	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFAultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理措施	发生原因（推测原因）		处理对策		防止再发
	没有运动指令所需的 PDO 映射		请执行该指令所需的 PDO 映射 关于所需的 PDO 映射，请参考该指令的“功能说明”		请执行使用的指令所需的 PDO 映射 关于各指令所需的 PDO 映射（伺服驱动器的设定），请参考《NJ/NX 系列 用户手册 运动控制篇》
	对不包含支持该指令的对象之目标设备，执行了该指令		存在不支持该指令的对象设备 请参考对象设备的手册，确认是否支持该指令，不得执行不支持的指令		请参考对象设备的手册，将程序编写为不执行不支持的指令
	对映射了欧姆龙制 EtherCAT 编码器从站 GX-EC02 □□的轴，将 Z 相（_mcEncoderMark）指定为触发条件，并启动了运动指令		对映射了欧姆龙产 EtherCAT 编码器从站 GX-EC02 □□的轴，请使用外部输入（_mcEXT）作为触发条件		对映射了欧姆龙产 EtherCAT 编码器从站 GX-EC02 □□的轴，请使用外部输入（_mcEXT）作为触发条件
附属信息	无				
注意事项 / 备注	无				

事件名称	超出电子齿轮分子设定范围		事件代码	54200000Hex	
内容	运动控制指令的输入变量“RatioNumerator”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFAultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理措施	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项 / 备注	无				

事件名称	超出电子齿轮分母设定范围			事件代码	54210000Hex
内容	运动控制指令的输入变量“RatioDenominator”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理措施	发生原因（推测原因）	处理对策		防止再发	
	指令的输入参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项/备注	无				

事件名称	超出目标速度设定范围			事件代码	54220000Hex
内容	运动控制指令的输入变量“Velocity”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
	_MC_GRP[*].MFaultLvl.Active	BOOL		轴组 发生轻度故障	
发生原因和处理措施	发生原因（推测原因）	处理对策		防止再发	
	指令的输入参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项/备注	无				

事件名称	超出加速度设定范围			事件代码	54230000Hex
内容	运动控制指令的输入变量“Acceleration”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
	_MC_GRP[*].MFaultLvl.Active	BOOL		轴组 发生轻度故障	
发生原因和处理措施	发生原因（推测原因）	处理对策		防止再发	
	指令的输入参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项/备注	无				

事件名称	超出减速度设定范围		事件代码	54240000Hex	
内容	运动控制指令的输入变量“Deceleration”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
		_MC_GRP[*].MFaultLvl.Active		BOOL	
				轴组 发生轻度故障	
发生原因和处理措施	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项 / 备注	无				

事件名称	超出跃度设定范围		事件代码	54250000Hex	
内容	运动控制指令的输入变量“Jerk”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
		_MC_GRP[*].MFaultLvl.Active		BOOL	
				轴组 发生轻度故障	
发生原因和处理措施	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项 / 备注	无				

事件名称	超出转矩设定范围		事件代码	54270000Hex	
内容	运动控制指令的输入变量“TorqueRamp”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理措施	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项 / 备注	无				

事件名称	超出主轴系数设定范围			事件代码	54280000Hex
内容	运动控制指令的输入变量“MasterScaling”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统 定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和 处理措施	发生原因（推测原因）	处理对策		防止再发	
	指令的输入参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出从轴系数设定范围			事件代码	54290000Hex
内容	运动控制指令的输入变量“SlaveScaling”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统 定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和 处理措施	发生原因（推测原因）	处理对策		防止再发	
	指令的输入参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出固定速度设定范围			事件代码	542A0000Hex
内容	运动控制指令的输入变量“FeedVelocity”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统 定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和 处理措施	发生原因（推测原因）	处理对策		防止再发	
	保持固定速度（输入变量“FeedVelocity”）的初始值（0）	请将固定速度（输入变量“FeedVelocity”）指定为正值		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出缓存模式选择范围		事件代码	542B0000Hex	
内容	运动控制指令的输入变量“BufferMode”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFAultLvl.Active		BOOL		轴 发生轻度故障
	_MC_GRP[*].MFAultLvl.Active		BOOL		轴组 发生轻度故障
发生原因和处理措施	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项 / 备注	无				

事件名称	超出坐标系选择范围		事件代码	542C0000Hex	
内容	运动控制指令的输入变量“CoordSystem”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴组不可动作 相应轴组正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_GRP[*].MFAultLvl.Active		BOOL		轴组 发生轻度故障
发生原因和处理措施	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项 / 备注	无				

事件名称	超出圆弧插补模式选择范围		事件代码	542D0000Hex	
内容	运动控制指令的输入变量“CircMode”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴组不可动作 相应轴组正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_GRP[*].MFAultLvl.Active		BOOL		轴组 发生轻度故障
发生原因和处理措施	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项 / 备注	无				

事件名称	超出方向选择范围			事件代码	542E0000Hex
内容	运动控制指令的输入变量“Direction”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理措施	发生原因（推测原因）	处理对策		防止再发	
	指令的输入参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出路径选择范围			事件代码	542F0000Hex
内容	运动控制指令的输入变量“PathChoice”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴组不可动作 相应轴组正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active	BOOL		轴组 发生轻度故障	
发生原因和处理措施	发生原因（推测原因）	处理对策		防止再发	
	指令的输入参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出位置类型选择范围			事件代码	54300000Hex
内容	运动控制指令的输入变量“ReferenceType”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	MC 共通 / 轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_COM.MFaultLvl.Active	BOOL		MC 共通 发生轻度故障	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理措施	发生原因（推测原因）	处理对策		防止再发	
	指令的输入参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出移动方法选择范围		事件代码	54310000Hex	
内容	运动控制指令的输入变量“MoveMode”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFAultLvl.Active		BOOL		轴 发生轻度故障
	_MC_GRP[*].MFAultLvl.Active		BOOL		轴组 发生轻度故障
发生原因和处理措施	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项 / 备注	无				

事件名称	超出转换模式选择范围		事件代码	54320000Hex	
内容	运动控制指令的输入变量“TransitionMode”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴组不可动作 相应轴组正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_GRP[*].MFAultLvl.Active		BOOL		轴组 发生轻度故障
发生原因和处理措施	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
	将“BufferMode”指定为“_mcAborting”、“_mcBuffered”且将“TransitionMode”指定为“_mcTMCORnerSuperimposed”		将“BufferMode”指定为“_mcAborting”、“_mcBuffered”时，请将“TransitionMode”指定为“_mcTMNone”，将“TransitionMode”指定为“_mcTMCORnerSuperimposed”时，请将“BufferMode”指定为“_mcBlendingLow”、“_mcBlendingPrevious”、“_mcBlendingNext”、“_mcBlendingHigh”		将“BufferMode”指定为“_mcAborting”、“_mcBuffered”时，请将“TransitionMode”指定为“_mcTMNone”，将“TransitionMode”指定为“_mcTMCORnerSuperimposed”时，请将“BufferMode”指定为“_mcBlendingLow”、“_mcBlendingPrevious”、“_mcBlendingNext”、“_mcBlendingHigh”
附属信息	无				
注意事项 / 备注	无				

事件名称	超出持续方向选择范围			事件代码	54330000Hex
内容	变更了运动控制指令的输入变量“Continuous (Reserved)”的值				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统	变量名称		数据类型	名称	
定义变量	_MC_AX[*].MFaultLvl.Active		BOOL	轴 发生轻度故障	
发生原因和 处理措施	发生原因（推测原因）		处理对策	防止再发	
	变更了输入变量“Continuous (Reserved)”的值		请修正程序，确保不变更输入变量“Continuous (Reserved)”的值	请编写程序，确保不变更输入变量“Continuous (Reserved)”的值	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出加减法方向选择范围			事件代码	54340000Hex
内容	运动控制指令的输入变量“CombineMode”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统	变量名称		数据类型	名称	
定义变量	_MC_AX[*].MFaultLvl.Active		BOOL	轴 发生轻度故障	
发生原因和 处理措施	发生原因（推测原因）		处理对策	防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围	请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出同步开始条件指定范围			事件代码	54350000Hex
内容	运动控制指令的输入变量“LinkOption”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统	变量名称		数据类型	名称	
定义变量	_MC_AX[*].MFaultLvl.Active		BOOL	轴 发生轻度故障	
发生原因和 处理措施	发生原因（推测原因）		处理对策	防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围	请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项 / 备注	无				

事件名称	主轴从轴相同		事件代码	54360000Hex		
内容	运动控制指令的输入变量“Master”和“Slave”中指定的轴相同					
发生位置	运动控制功能模块		发生位置详情	MC 共通 / 轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障	
		_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入变量“Master”和“Slave”的参数相同		请修正参数，确保该指令的输入变量“Master”和“Slave”中指定的轴不同		请在指令的输入变量“Master”和“Slave”中指定不同的轴	
附属信息	无					
注意事项 / 备注	无					

事件名称	主轴辅助轴相同		事件代码	54370000Hex		
内容	运动控制指令的输入变量“Master”和“Auxiliary”中指定的轴相同					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入变量“Master”和“Auxiliary”的参数相同		请修正参数，确保该指令的输入变量“Master”和“Auxiliary”中指定的轴不同		请在指令的输入变量“Master”和“Auxiliary”中指定不同的轴	
附属信息	无					
注意事项 / 备注	无					

事件名称	主轴 / 从轴 轴编号非升序		事件代码	54380000Hex		
内容	运动控制指令的输入变量“Master”和“Slave”中指定的轴编号不是升序					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	在指令的输入变量“ReferenceType”中指定“_mcLatestCommand”时，指令的输入变量“Master”和“Slave”的参数不是升序		请修正参数，确保在指令的输入变量“ReferenceType”中指定“_mcLatestCommand”时，该指令的输入变量“Master”和“Slave”中指定的轴编号为升序或者在主轴位置类型选择中指定“_mcCommand”		指定时，请确保在变量“ReferenceType”中指定“_mcLatestCommand”时，输入变量的主轴和从轴为升序	
附属信息	无					
注意事项 / 备注	无					

事件名称	凸轮表指定不正确			事件代码	54390000Hex
内容	运动控制指令的输入变量“CamTable”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	MC 共通 / 轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	发生位置详情为“MC 共通”时： 无影响 发生位置详情为“轴”时： 相应轴不可动作	
系统定义变量	变量名称		数据类型		名称
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障
		_MC_AX[*].MFaultLvl.Active		BOOL	轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	在指令的输入变量“CamTable”中指定了凸轮数据变量以外的值		请修正参数，使该指令的输入变量“CamTable”中指定的参数为凸轮数据变量		请在指令的输入变量“CamTable”中指定凸轮数据变量
附属信息	无				
注意事项 / 备注	无				

事件名称	同步停止中			事件代码	543A0000Hex
内容	执行了运动控制的同步控制指令，但并非可执行的条件				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	<ul style="list-style-type: none"> • 执行了 MC_CamOut（解除凸轮动作）指令，但 MC_CamIn（开始凸轮动作）指令不是执行中 • 执行了 MC_GearOut（解除凸轮动作）指令，但 MC_GearIn（开始凸轮动作）指令、MC_GearInPos（位置指定齿轮动作）指令不是执行中 • 执行了 MC_Phasing（主轴相对值相位补偿）指令，但 MC_CamIn（开始凸轮动作）指令、MC_GearIn（开始齿轮动作）指令、MC_GearInPos（位置指定齿轮动作）指令、MC_MoveLink（梯模式凸轮）指令不是执行中 		请将程序修正为正在执行该指令时也能执行的条件		请修正为正在执行同步控制指令时也能执行的条件
附属信息	无				
注意事项 / 备注	无				

事件名称	运动指令不可重新执行		事件代码	543B0000Hex			
内容	重新执行了无法重新执行的运动控制指令						
发生位置	运动控制功能模块		发生位置详情	MC 共通 / 轴 / 轴组	检测时序	指令启动时	
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统	
产生的影响	用户程序	继续	动作	发生位置详情为“MC 共通”时： 无影响 发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止			
系统定义变量	变量名称		数据类型		名称		
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障		
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障		
		_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发		
	无法重新执行的运动控制指令的重新执行		请修正程序，确保在该指令的输出变量“Busy”变为 FALSE 前，输入变量“Execute”不启动		使用无法重新执行的指令时，请在输入变量“Execute”的启动条件中加入该指令的输出变量“Busy”为 FALSE 这一条或者在重新执行指令前，先停止指令		
附属信息	无						
注意事项 / 备注	无						

事件名称	运动指令不可多重启动		事件代码	543C0000Hex			
内容	对同一对象（MC 共通 / 轴 / 轴组）执行了多个不可同时执行的功能						
发生位置	运动控制功能模块		发生位置详情	MC 共通 / 轴 / 轴组	检测时序	指令多重启动开始时	
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统	
产生的影响	用户程序	继续	动作	发生位置详情为“MC 共通”时： 无影响 发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止			
系统定义变量	变量名称		数据类型		名称		
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障		
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障		
		_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发		
	对同一对象（MC 共通 / 轴 / 轴组）执行了多个不可同时执行的功能		请确认该指令的多重启动规格，并修正程序，确保不会同时执行不可同时执行的指令		请确认所用指令的多重启动规格，确保不会同时执行不可同时执行的指令		
附属信息	无						
注意事项 / 备注	无						

事件名称	轴种类不合适			事件代码	543D0000Hex
内容	对编码器轴执行了动作指令				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	对编码器轴执行动作指令		请在该指令的轴种类设定中指定伺服轴或虚拟伺服轴，或修正程序，不对编码器轴执行该动作指令		请设定为对伺服轴或虚拟伺服轴执行动作指令
附属信息	无				
注意事项 / 备注	无				

事件名称	多轴协调动作中不可启动指令			事件代码	543E0000Hex
内容	<ul style="list-style-type: none"> 对多轴协调动作中的轴或轴组执行了动作指令 轴组为有效状态时，执行了不可使用的机器人指令 				
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序 指令多重启动开始时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	轴组的减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障
	发生原因（推测原因）		处理对策		防止再发
发生原因和处理	对多轴协调动作中的轴或轴组执行动作指令		请修正程序，使其在该轴或该轴组未处于多轴协调动作时执行该指令		请对未处于多轴协调动作的轴或轴组执行轴动作指令
	对轴组为有效状态的轴组执行 MC_SetKinTransform（运动学转换设定）指令		请修正程序，使其在轴组为无效时执行该指令。		请在轴组为无效时执行该指令。
附属信息	无				
注意事项 / 备注	无				

事件名称	轴组为无效状态时启动多轴协调指令		事件代码	543F0000Hex	
内容	对轴组为无效状态的轴组，启动了多轴协调指令				
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴组不可动作 相应轴组正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_GRP[*].MFAultLvl.Active		BOOL		轴组 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	对轴组为无效状态的轴组，启动多轴协调指令		请修正程序，将轴组设为有效状态后，再执行该指令要将轴组设为有效状态，请执行		请将轴组设为有效后，再执行多轴协调动作指令要将轴组设为有效状态，请执行 MC_GroupEnable（轴组有效）指令
	对轴组为无效状态的轴组，启动以下指令		MC_GroupEnable（轴组有效）指令		
<ul style="list-style-type: none"> MC_MoveTimeAbsolute（时间指定绝对值位置指令）指令 MC_SyncLinearConveyor（输送带同步动作开始）指令 MC_SyncOut（同步动作解除）指令 MC_RobotJog（轴组点动进给）指令 					
附属信息	无				
注意事项/备注	无				

事件名称	轴组不可启用		事件代码	54400000Hex	
内容	MC_GroupEnable（轴组启用）指令执行失败				
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴组不可动作。构成轴的动作继续	
系统定义变量	变量名称		数据类型		名称
	_MC_GRP[*].MFAultLvl.Active		BOOL		轴组 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	执行 MC_GroupEnable（轴组启用）指令时，构成轴中存在未停止的轴		请修正程序，使其在所有构成轴停止时执行 MC_GroupEnable（轴组有效）指令轴变量“Status.Disabled”或“Status.Standstill”为 TRUE 时，该轴处于停止中		请编写程序，使其在所有构成轴停止时执行 MC_GroupEnable（轴组有效）指令轴变量“Status.Disabled”或“Status.Standstill”为 TRUE 时，该轴处于停止中
	执行 MC_GroupEnable（轴组启用）指令时，构成轴中存在正在执行 MC_TouchProbe（外部锁定有效）指令的轴		请修正程序，使其在所有构成轴上未执行 MC_TouchProbe（外部锁定有效）指令时，执行 MC_GroupEnable（轴组有效）指令		请编写程序，使其在所有构成轴上未执行 MC_TouchProbe（外部锁定有效）指令时，执行 MC_GroupEnable（轴组有效）指令
附属信息	无				
注意事项/备注	无				

事件名称	不可运行（伺服 OFF）轴动作指示			事件代码	54410000Hex
内容	对伺服 OFF 中的轴执行了动作指令				
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	动作指令不启动	
系统定义变量	变量名称		数据类型	名称	
	_MC_AX[*].MFaultLvl.Active		BOOL	轴 发生轻度故障	
	_MC_GRP[*].MFaultLvl.Active		BOOL	轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策	防止再发	
	对伺服 OFF 中的轴执行动作指令		请修正程序，使其在伺服 ON 后执行该指令	请在伺服 ON 后执行轴动作指令	
	通过 MC_Home（原点复归）指令或 MC_HomeWithParameter（参数指定原点复位）指令对未建立 EtherCAT 过程数据通信的轴执行了原点复位		EtherCAT 主机的系统定义变量“_EC_PDslavTbl”（过程数据通信中从站表）为 FALSE 时，请排除原因，待“_EC_PDslavTbl”变为 TRUE 后，通过 MC_Home（原点复位）指令或 MC_HomeWithParameter（参数指定原点复位）指令执行原点复位	控制器电源打开或下载后、从站通信异常的解除、脱离、再加入、无效、有效后，若要立即通过 MC_Home（原点复位）指令或 MC_HomeWithParameter（参数指定原点复位）指令执行原点复位，请编写程序，使其在确认 EtherCAT 主机的系统定义变量“_EC_PDslavTbl”（过程数据通信中从站表）已变为 TRUE 后再执行。	
附属信息	附属信息 1：因发生源详情不同而异 轴时：“0” 轴组时：发生异常的逻辑轴编号				
注意事项 / 备注	无				

事件名称	构成轴强制停止中错误			事件代码	54420000Hex
内容	对正在向构成轴执行 MC_Stop（强制停止）指令的轴组执行了动作指令				
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴组不可动作	
系统定义变量	变量名称		数据类型	名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL	轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策	防止再发	
	对正在向构成轴执行 MC_Stop（强制停止）指令的轴组执行了动作指令		请将构成轴强制停止指令的输入变量“Execute”设为 FALSE，停止执行，然后解除异常，再发出动作指令	请将构成轴强制停止指令的输入变量“Execute”设为 FALSE 并结束，然后再执行轴组动作指令	
附属信息	附属信息 1：强制停止的逻辑轴编号				
注意事项 / 备注	无				

事件名称	运动指令多重启动数超限		事件代码	54430000Hex	
内容	通过缓存模式 Buffered、Blending 缓存的运动控制指令的缓存数超出极限				
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
		_MC_GRP[*].MFaultLvl.Active		BOOL	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	轴指令中，执行中的指令和缓存中的指令总和超过“2”		请修正程序，确保在执行该指令时缓存数不会超出上限		轴指令中，执行中的指令和缓存中的指令总和请勿超过“2”
		轴组指令中，执行中的指令和缓存中的指令总和超过“8”		轴组指令中，执行中的指令和缓存中的指令总和请勿超过“8”	
附属信息	无				
注意事项 / 备注	无				

事件名称	移动量不足		事件代码	54440000Hex	
内容	定位指令的多重启动 / 重新执行时，无法按指定的减速度或加速度执行指定的动作				
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
		_MC_GRP[*].MFaultLvl.Active		BOOL	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	“加减速超限”设定为“异常停止”时，无法按定位指令的多重启动 / 重新执行中指定的减速度 / 加速度停到目标位置		请修正程序，确保在该指令的动作规格下，定位动作多重启动 / 重新执行时，以指定的减速度或加速度动作的情况下，不会超过目标位置或者将“加减速超限”的设定变更为“异常停止”以外的值		请先确认所用指令的动作规格，编写程序时，确保不会发生本异常或者将“加减速超限”设定为“异常停止”以外的值
附属信息	无				
注意事项 / 备注	无				

事件名称	达到共混中继速度所需的移动量不足			事件代码	5445000Hex						
内容	加减速至中继速度所需的移动量不足										
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序	指令多重启动开始时					
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统					
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止							
系统定义变量	变量名称		数据类型		名称						
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障						
						_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发						
	“加减速超限”设定为“异常停止”时，将当前指令加减速至中继速度所需的移动量不足		请修正程序，确保在该指令的动作规格下，不会发生移动量不足或者将“加减速超限”的设定变更为“异常停止”以外的值		请先确认所用指令的动作规格，编写程序时，确保不会发生本异常或者将“加减速超限”设定为“异常停止”以外的值						
附属信息	无										
注意事项 / 备注	无										

事件名称	梯形模式凸轮等速移动量不足			事件代码	5446000Hex	
内容	主轴的等速移动量不足“0”					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	MC_MoveLink（梯形模式凸轮）指令中，主轴的等速移动量不足“0”		请修正程序，确保（主轴移动距离 \geq 主轴加速移动距离+主轴减速移动距离）		请先确认所用指令的动作规格，编写程序时，确保不会发生本异常	
附属信息	无					
注意事项 / 备注	无					

事件名称	位置指定齿轮动作目标速度不足			事件代码	5447000Hex	
内容	MC_GearInPos（位置指定齿轮动作）指令中，从轴的“目标速度”较小，因此无法达到需要的速度					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	MC_GearInPos（位置指定齿轮动作）指令中，输入变量“Velocity（目标速度）”的值小于（指令启动时的主轴速度 \times 齿轮比）		请根据该指令的动作规格，将输入变量“Velocity（目标速度）”的值设定为大于（指令启动时的主轴速度 \times 齿轮比）		请先确认所用指令的动作规格，编写程序时，确保不会发生本异常	
附属信息	无					
注意事项 / 备注	无					

事件名称	圆弧插补起点终点相同		事件代码	54480000Hex	
内容	MC_MoveCircular2D（2轴圆弧插补）指令中，指定半径指定方式时，起点和终点为相同位置或者指定通过点指定方式时，起点、终点和通过点为相同位置				
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴组不可动作 相应轴组正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_GRP[*].MFAultLvl.Active	BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	MC_MoveCircular2D（2轴圆弧插补）指令中，指定半径指定方式，起点和终点的位置相同	请修正程序，确保在该指令的起点和终点位置相同时，不会指定为半径指定方式		执行半径指定方式的圆弧插补时，请确保起点和终点不是同一点	
	MC_MoveCircular2D（2轴圆弧插补）指令中，指定通过点指定方式，起点、终点和通过点的位置相同	请修正程序，确保在该指令的起点、终点和通过点位置相同时，不会指定为通过点指定方式		执行通过点指定方式的圆弧插补时，请确保起点、终点和通过点不是同一点	
附属信息	无				
注意事项/备注	无				

事件名称	超出圆弧插补中心点指定位置范围		事件代码	54490000Hex	
内容	MC_MoveCircular2D（2轴圆弧插补）指令中，指定中心点指定方式时，中心点的位置指定超出容许范围				
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴组不可动作 相应轴组正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_GRP[*].MFAultLvl.Active	BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	MC_MoveCircular2D（2轴圆弧插补）指令中，指定中心点指定方式，起点和中心点的距离、终点和中心点的距离之差超出轴组设定的“中心点修正容许率”中指定的容许范围	请修正程序，使设定的中心点能够确保起点和中心点的距离、终点和中心点的距离之差小于轴组设定的“中心点修正容许率”中指定的容许范围		请确保起点和中心点的距离、终点和中心点的距离之差小于轴组设定的“中心点修正容许率”中指定的容许范围	
附属信息	无				
注意事项/备注	无				

事件名称	计数模式设定引起的指令启动异常			事件代码	544A0000Hex	
内容	对计数模式设定为旋转模式的轴执行了旋转模式下无法使用的指令					
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴组不可动作 相应轴组正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	在旋转模式下无法使用的指令中，使用计数模式设定为旋转模式的轴		请将该指令中指定的该轴的计数模式变更为线性模式		请确认可执行所用指令的计数模式，再设定轴的计数模式	
附属信息	无					
注意事项/备注	无					

事件名称	超出参数选择范围			事件代码	544C0000Hex	
内容	运动控制指令的输入变量“ParameterNumber”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	不执行当前指令		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项/备注	无					

事件名称	超出停止方法选择范围			事件代码	544D0000Hex	
内容	运动控制指令的输入变量“StopMode”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项/备注	无					

事件名称	超出触发输入条件的锁定 ID 选择范围			事件代码	544E0000Hex	
内容	运动控制指令的输入变量“TriggerInput::LatchID”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项/备注	无					

事件名称	超出 MC 设定写入的设定范围			事件代码	544F0000Hex	
内容	运动控制指令的输入变量“SettingValue”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	不执行当前指令		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
	参数指定和设定值的数据类型不一致		请将参数指定和设定值的数据类型修正为一致		请统一参数指定和设定值的数据类型	
附属信息	无					
注意事项/备注	无					

事件名称	超出触发输入条件的模式选择范围			事件代码	54500000Hex	
内容	运动控制指令的输入变量“TriggerInput::Mode”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项/备注	无					

事件名称	超出触发输入条件的驱动触发输入信号选择范围		事件代码	54510000Hex	
内容	运动控制指令的输入变量“TriggerInput::InputDrive”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项/备注	无				

事件名称	运动指令不可重新执行（轴指定）		事件代码	54530000Hex	
内容	重新执行运动控制指令时，变更了不可变更的输入变量“Axis”的参数				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令重新执行时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	重新执行指令时，变更了不可变更的输入变量的参数		请修正程序，确保在重新执行该指令时，该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更，在编写程序时，确保不可变更的输入变量在重新执行时不会发生输入参数变化
附属信息	无				
注意事项/备注	无				

事件名称	运动指令不可重新执行（缓存模式选择）		事件代码	54540000Hex	
内容	重新执行运动控制指令时，变更了不可变更的输入变量“BufferMode”的参数				
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序 指令重新执行时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
	_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	重新执行指令时，变更了不可变更的输入变量的参数		请修正程序，确保在重新执行该指令时，该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更，在编写程序时，确保不可变更的输入变量在重新执行时不会发生输入参数变化
附属信息	无				
注意事项/备注	无				

事件名称	运动指令不可重新执行（方向选择）		事件代码	54550000Hex	
内容	重新执行运动控制指令时，变更了不可变更的输入变量“Direction”的参数				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令重新执行时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	重新执行指令时，变更了不可变更的输入变量的参数		请修正程序，确保在重新执行该指令时，该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更，在编写程序时，确保不可变更的输入变量在重新执行时不会发生输入参数变化
附属信息	无				
注意事项/备注	无				

事件名称	运动指令不可重新执行（重复模式）		事件代码	54560000Hex	
内容	重新执行运动控制指令时，变更了不可变更的输入变量“Periodic”的参数				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令重新执行时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	重新执行指令时，变更了不可变更的输入变量的参数		请修正程序，确保在重新执行该指令时，该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更，在编写程序时，确保不可变更的输入变量在重新执行时不会发生输入参数变化
附属信息	无				
注意事项/备注	无				

事件名称	运动指令不可重新执行（轴组指定）		事件代码	54570000Hex	
内容	重新执行运动控制指令时，变更了不可变更的输入变量“AxesGroup”的参数				
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序 指令重新执行时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴组不可动作 相应轴组正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	重新执行指令时，变更了不可变更的输入变量的参数		请修正程序，确保在重新执行该指令时，该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更，在编写程序时，确保不可变更的输入变量在重新执行时不会发生输入参数变化
附属信息	无				
注意事项/备注	无				

事件名称	运动指令不可重新执行（跃动设定）			事件代码	54580000Hex	
内容	重新执行运动控制指令时，变更了不可变更的输入变量“Jerk”的参数					
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序	指令重新执行时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
		_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	重新执行指令时，变更了不可变更的输入变量的参数		请修正程序，确保在重新执行该指令时，该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更，在编写程序时，确保不可变更的输入变量在重新执行时不会发生输入参数变化	
附属信息	无					
注意事项 / 备注	无					

事件名称	运动指令不可重新执行（主轴）			事件代码	54590000Hex	
内容	重新执行运动控制指令时，变更了不可变更的输入变量“Master”的参数					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令重新执行时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	重新执行指令时，变更了不可变更的输入变量的参数		请修正程序，确保在重新执行该指令时，该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更，在编写程序时，确保不可变更的输入变量在重新执行时不会发生输入参数变化	
附属信息	无					
注意事项 / 备注	无					

事件名称	运动指令不可重新执行 (MasterOffset)		事件代码	545A0000Hex	
内容	重新执行运动控制指令时, 变更了不可变更的输入变量 "MasterOffset" 的参数				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时, 则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因 (推测原因)		处理对策		防止再发
	重新执行指令时, 变更了不可变更的输入变量的参数		请修正程序, 确保在重新执行该指令时, 该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更, 在编写程序时, 确保不可变更的输入变量在重新执行时不会发生输入参数变化
附属信息	无				
注意事项 / 备注	无				

事件名称	运动指令不可重新执行 (MasterScaling)		事件代码	545B0000Hex	
内容	重新执行运动控制指令时, 变更了不可变更的输入变量 "MasterScaling" 的参数				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时, 则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因 (推测原因)		处理对策		防止再发
	重新执行指令时, 变更了不可变更的输入变量的参数		请修正程序, 确保在重新执行该指令时, 该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更, 在编写程序时, 确保不可变更的输入变量在重新执行时不会发生输入参数变化
附属信息	无				
注意事项 / 备注	无				

事件名称	运动指令不可重新执行 (MasterStartDistance)		事件代码	545C0000Hex	
内容	重新执行运动控制指令时, 变更了不可变更的输入变量 "MasterStartDistance" 的参数				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时, 则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因 (推测原因)		处理对策		防止再发
	重新执行指令时, 变更了不可变更的输入变量的参数		请修正程序, 确保在重新执行该指令时, 该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更, 在编写程序时, 确保不可变更的输入变量在重新执行时不会发生输入参数变化
附属信息	无				
注意事项 / 备注	无				

事件名称	运动指令不可重新执行 (Continuous)			事件代码	545D0000Hex	
内容	重新执行运动控制指令时, 变更了不可变更的输入变量“Continuous”的参数					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令重新执行时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时, 则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 (推测原因)		处理对策		防止再发	
	重新执行指令时, 变更了不可变更的输入变量的参数		请修正程序, 确保在重新执行该指令时, 该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更, 在编写程序时, 确保不可变更的输入变量在重新执行时不会发生输入参数变化	
附属信息	无					
注意事项/备注	无					

事件名称	运动指令不可重新执行 (MoveMode)			事件代码	545E0000Hex	
内容	重新执行运动控制指令时, 变更了不可变更的输入变量“MoveMode”的参数					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令重新执行时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时, 则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 (推测原因)		处理对策		防止再发	
	重新执行指令时, 变更了不可变更的输入变量的参数		请修正程序, 确保在重新执行该指令时, 该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更, 在编写程序时, 确保不可变更的输入变量在重新执行时不会发生输入参数变化	
附属信息	无					
注意事项/备注	无					

事件名称	辅助轴指定不正确			事件代码	545F0000Hex	
内容	运动控制指令的输入变量“Auxiliary”中指定的轴不存在					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应从轴不可动作。 相应从轴正在动作时, 则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 (推测原因)		处理对策		防止再发	
	指令的输入变量“Auxiliary”中指定的轴为不存在的变量		请修正为该指令中指定的轴为存在的变量		将指令的输入参数指定为变量时, 请指定为存在的变量	
附属信息	无					
注意事项/备注	无					

事件名称	轴指定不正确			事件代码	54600000Hex
内容	运动控制指令的输入变量“Axis”中指定的轴不存在				
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	不执行当前指令	
系统定义变量	变量名称		数据类型		名称
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入变量“Axis”中指定的轴为不存在的变量		请修正为该指令中指定的轴为存在的变量		将指令的输入参数指定为变量时，请指定为存在的变量
附属信息	无				
注意事项/备注	无				

事件名称	轴组指定不正确			事件代码	54610000Hex
内容	运动控制指令的输入变量“AxesGroup”中指定的轴组不存在或不是使用轴组				
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	不执行当前指令	
系统定义变量	变量名称		数据类型		名称
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入变量“AxesGroup”中指定的轴组为不存在的变量		请修正为该指令中指定的轴组为存在的轴组变量		将指令的输入参数指定为变量时，请指定为存在的变量
	指令的输入变量“AxesGroup”中指定的轴组未设定为使用轴组		请将该指令中指定的轴组设定为使用轴组		请将输入变量“AxesGroup”中指定的轴组设定为使用轴组
附属信息	无				
注意事项/备注	无				

事件名称	主轴指定不正确			事件代码	54620000Hex
内容	运动控制指令的输入变量“Master”中指定的轴不正确				
发生位置	运动控制功能模块		发生位置详情	MC 共通 / 轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_COM.MFaultLvl.Active	BOOL		MC 共通 发生轻度故障	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入变量“Master”中指定的轴为不存在的变量		请修正为该指令中指定的轴为存在的变量		将指令的输入参数指定为变量时，请指定为存在的变量
	MC_Phasing（主轴相对值相位补偿）指令时，输入变量“Master”中指定轴不是同步主轴		请将 MC_Phasing（主轴相对值相位补偿）指令的输入变量“Master”中输入的变量修正为同步控制指令的主轴中指定的轴		请将 MC_Phasing（主轴相对值相位补偿）指令的输入变量“Master”中输入的变量设为同步控制指令的主轴中指定的轴
	分配了主轴和从轴的任务不是同一任务		请将该指令的输入变量“Master”和输入变量“Slave”中输入的轴分配到同一任务中		主轴和从轴请指定为分配到同一任务的轴
附属信息	无				
注意事项 / 备注	无				

事件名称	运动指令不可重新执行（SlaveOffset）			事件代码	54630000Hex
内容	重新执行运动控制指令时，变更了不可变更的输入变量“SlaveOffset”				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令重新执行时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	重新执行指令时，变更了不可变更的输入变量的参数		请修正程序，确保在重新执行该指令时，该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更，在编写程序时，确保不可变更的输入变量在重新执行时不会发生输入参数变化
附属信息	无				
注意事项 / 备注	无				

事件名称	运动指令不可重新执行 (SlaveScaling)		事件代码	54640000Hex	
内容	重新执行运动控制指令时，变更了不可变更的输入变量“SlaveScaling”				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因 (推测原因)		处理对策		防止再发
	重新执行指令时，变更了不可变更的输入变量的参数		请修正程序，确保在重新执行该指令时，该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更，在编写程序时，确保不可变更的输入变量在重新执行时不会发生输入参数变化
附属信息	无				
注意事项 / 备注	无				

事件名称	运动指令不可重新执行 (StartPosition)		事件代码	54650000Hex	
内容	重新执行运动控制指令时，变更了不可变更的输入变量“StartPosition”				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因 (推测原因)		处理对策		防止再发
	重新执行指令时，变更了不可变更的输入变量的参数		请修正程序，确保在重新执行该指令时，该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更，在编写程序时，确保不可变更的输入变量在重新执行时不会发生输入参数变化
附属信息	无				
注意事项 / 备注	无				

事件名称	原点未确定状态下指令启动异常			事件代码	54660000Hex		
内容	在零点未确定状态下执行了高速原点复位或插补指令						
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序	指令启动时	
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统	
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止			
系统定义变量	变量名称		数据类型		名称		
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障		
		_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发		
	在零点未确定状态下执行了高速原点复位		请通过原点复位确定原点，再执行该高速原点复位		请通过原点复位，在零点确定的状态下，执行高速原点复位指令		
	对包含零点未确定状态的构成轴执行了插补指令		请通过原点复位确定所有构成轴的原点，再执行该插补指令		请通过原点复位确定所有构成轴的原点，再执行插补指令		
		对包含零点未确定状态的逻辑轴的轴组执行了以下机器人指令。 • MC_SetKinTransform（运动学转换设定）指令 • MC_MoveTimeAbsolute（时间指定绝对值位置指令）指令 • MC_SyncLinearConveyor（输送带同步动作开始）指令 • MC_SyncOut（同步动作解除）指令 • MC_GroupMon（轴组监视）指令 • MC_RobotJog（轴组点动进给）指令					
附属信息	附属信息 1：因发生源详情不同而异 发生源为轴时：“0” 发生源为轴组时：逻辑轴编号						
注意事项 / 备注	执行原点复位后，若执行“当前位置变更”指令，仍会变为零点未确定状态，因此需要重新执行原点复位才能变为零点确定状态						

事件名称	运动指令不可重新执行（位置类型）			事件代码	54670000Hex	
内容	重新执行运动控制指令时，变更了不可变更的输入变量“ReferenceType”					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令重新执行时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	重新执行指令时，变更了不可变更的输入变量的参数		请修正程序，确保在重新执行该指令时，该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更，在编写程序时，确保不可变更的输入变量在重新执行时不会发生输入参数变化	
附属信息	无					
注意事项 / 备注	无					

事件名称	未使用轴指定（主轴）			事件代码	54680000Hex	
内容	运动控制指令中指定的主轴为未使用轴					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	运动控制指令中指定的主轴为未使用轴		请将该指令中指定的主轴修正为使用轴		请将运动控制指令中指定的主轴设为使用轴	
附属信息	无					
注意事项/备注	无					

事件名称	超出开始位置设定范围			事件代码	54690000Hex	
内容	运动控制指令的输入变量“FirstPosition”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项/备注	无					

事件名称	超出结束位置设定范围			事件代码	546A0000Hex	
内容	运动控制指令的输入变量“LastPosition”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项/备注	无					

事件名称	开始位置 / 结束位置、大小关系不正确（线性模式）		事件代码	546B0000Hex	
内容	运动控制指令的输入变量“LastPosition”中指定的参数值小于输入变量“FirstPosition”中指定的参数				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	计数模式为线性模式时，指令的输入参数中，“LastPosition”的值小于“FirstPosition”的值	请将该指令中指定的“LastPosition”的值修正为大于“FirstPosition”的值 或者将计数模式变更为旋转模式		请编写程序，使该指令中指定的“LastPosition”的值大于“FirstPosition”的值或者确认该轴的计数模式为旋转模式	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出主轴同步位置设定范围		事件代码	546C0000Hex	
内容	运动控制指令的输入变量“MasterSyncPosition”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	指令的输入参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出从轴同步位置设定范围		事件代码	546D0000Hex	
内容	运动控制指令的输入变量“SlaveSyncPosition”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	指令的输入参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项 / 备注	无				

事件名称	触发输入条件的锁定 ID 重复		事件代码	546E0000Hex	
内容	运动控制指令中指定的锁定 ID 在多个指令中重复				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	在 MC_TouchProbe（外部锁定有效）指令、MC_MoveLink（梯形模式凸轮）指令、MC_MoveFeed（中断固定尺寸定位）指令中同时使用了相同的锁定 ID		请修正程序，避免相同的锁定 ID 在该指令和其他指令中同时使用请使用不同的锁定 ID 或避免同时执行使用相同锁定 ID 的指令 此外，执行 MC_Home（原点复位）指令或 MC_HomeWithParameter（参数指定原点复位）指令的过程中，锁定功能 1/2 均视为使用中		请勿在 MC_TouchProbe（外部锁定有效）指令、MC_MoveLink（梯形模式凸轮）指令、MC_MoveFeed（中断固定尺寸定位）指令中同时使用相同的锁定 ID
	试图以 MC_AbortTrigger（外部锁定无效）指令中止 MC_TouchProbe（外部锁定有效）指令以外正在使用的锁定		请勿以外部锁定无效指令中止外部锁定有效指令以外正在使用的锁定		请勿对外部锁定有效指令以外正在使用的锁定执行外部锁定无效指令
附属信息	无				
注意事项 / 备注	变更锁定 ID 时，请确认该锁定 ID 没有在其他指令中同时使用				

事件名称	超出跃度超驰值范围		事件代码	546F0000Hex	
内容	运动控制指令的输入变量“JerkFactor”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	附属信息 1：因发生源详情不同而异 发生源为轴时：“0” 发生源为轴组时：逻辑轴编号				
注意事项 / 备注	无				

事件名称	超出加减速度超驰值范围			事件代码	5470000Hex
内容	运动控制指令的输入变量“AccFactor”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	附属信息 1：因发生源详情不同而异 发生源为轴时：“0” 发生源为轴组时：逻辑轴编号				
注意事项 / 备注	无				

事件名称	超出开始位置方式指定范围			事件代码	54710000Hex
内容	运动控制指令的输入变量“StartMode”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项 / 备注	无				

事件名称	运动指令不可重新执行（开始位置方式）			事件代码	54720000Hex
内容	重新执行运动控制指令时，变更了不可变更的输入变量“StartMode”的参数				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令重新执行时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	重新执行指令时，变更了不可变更的输入变量的参数		请修正程序，确保在重新执行该指令时，该输入变量的参数不会变化		请通过手册确认所用运动控制指令的各输入变量是否可在重新执行时变更，在编写程序时，确保不可变更的输入变量在重新执行时不会发生输入参数变化
附属信息	无				
注意事项 / 备注	无				

事件名称	未使用轴指定（辅助轴）		事件代码	5474000Hex	
内容	运动控制指令的输入变量“Auxiliary”中指定的轴为未使用轴				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入变量“Auxiliary”中指定的轴为未使用轴		请将该指令中指定的轴设定为使用轴或者修正参数，确保指定为使用轴		请将指令中指定的轴设为使用轴
附属信息	无				
注意事项/备注	无				

事件名称	位置指定齿轮指定值异常		事件代码	5475000Hex	
内容	无法以运动控制指令中输入的速度 / 加速度 / 减速度进行同步动作				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	无法以指令中输入的速度 / 加速度 / 减速度进行指定的同步动作		请修正程序，确保在MC_GearInPos（位置指定齿轮动作）指令的动作规格下，可执行同步动作		请确认所用指令的处理，设定为可执行同步动作的值
附属信息	无				
注意事项/备注	无				

事件名称	位置指定齿轮主轴速度零		事件代码	5476000Hex	
内容	启动运动控制指令时，主轴的速度为“0”				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	启动指令时，主轴的速度为“0”		请修正程序，确保在启动该指令时，主轴的速度不为“0”		请编写程序，确保在执行指令时，主轴的速度不为“0”
附属信息	无				
注意事项/备注	无				

事件名称	超出目标位置设定范围			事件代码	54780000Hex		
内容	运动控制指令的输入变量“Position”中指定的参数超出范围						
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序	指令启动时	
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统	
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止			
系统定义变量	变量名称		数据类型		名称		
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障		
		_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发		
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围		
		在旋转模式的轴中，目标位置超出链接设定范围		请修正为在旋转模式的轴中，目标位置不会超出链接设定范围		请确保在旋转模式的轴中，目标位置不会超出链接设定范围	
附属信息	附属信息 1：因发生源详情不同而异 发生源为轴时：“0” 发生源为轴组时：相应指令的输入变量“Position”指定的参数超过范围						
注意事项 / 备注	无						

事件名称	超出移动距离范围			事件代码	54790000Hex		
内容	运动控制指令的输入变量“Distance”中指定的参数超出范围或加上“Distance”的值后目标位置超出范围						
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序	指令启动时	
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统	
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止			
系统定义变量	变量名称		数据类型		名称		
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障		
		_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发		
	将指令的输入参数绝对值转换为脉冲单位时，超出 40 位的范围		请修正该指令的输入变量“Distance”中指定的输入参数，使移动距离或目标位置在范围内		请编写程序，确保指令的移动距离或目标位置在范围内		
		在线性模式的轴中，将加上移动距离后的目标位置转换为脉冲单位时，带符号超出 40 位的范围					
附属信息	无						
注意事项 / 备注	无						

事件名称	超出凸轮表起点位置设定范围		事件代码	547A0000Hex	
内容	运动控制指令的输入变量“StartPosition”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项/备注	无				

事件名称	超出凸轮动作（主轴追踪）开始位置设定范围		事件代码	547B0000Hex	
内容	运动控制指令的输入变量“MasterStartDistance”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项/备注	无				

事件名称	圆弧插补半径指定异常		事件代码	547C0000Hex	
内容	MC_MoveCircular2D（2轴圆弧插补）指令中，指定半径指定方式时，无法以指定的半径创建圆弧轨迹				
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴组不可动作 相应轴组正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	MC_MoveCircular2D（2轴圆弧插补）指令中，指定为半径指定方式，但无法以指定的半径创建圆弧轨迹		请修正半径，使圆弧轨迹可以创建		请确认所用指令的处理，设定为可创建圆弧轨迹的半径
附属信息	无				
注意事项/备注	无				

事件名称	圆弧插补半径溢出			事件代码	547D0000Hex	
内容	MC_MoveCircular2D（2轴圆弧插补）指令中，指定为通过点指定方式 / 中心点指定方式时，圆弧半径超出最大值					
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴组不可动作 相应轴组正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	MC_MoveCircular2D（2轴圆弧插补）指令中，指定为通过点指定方式 / 中心点指定方式时，将圆弧半径转换为脉冲单位后，超出 40 位的范围		请修正参数，确保在该指令的动作规格下，将圆弧半径转换为脉冲单位时，不会超出 40 位的范围。 通过点指定时：起点、通过点、终点 中心点指定时：起点、终点、中心点		请确认所用指令的处理并设定输入参数，确保将圆弧半径转换为脉冲单位时，不会超出 40 位的范围	
附属信息	无					
注意事项 / 备注	半径指定方式时，若半径超出最大值，将发生“超出通过点位置 / 中心位置 / 半径指定范围”异常					

事件名称	超出圆弧轴指定范围			事件代码	547E0000Hex	
内容	运动控制指令的输入变量“CircAxes”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴组不可动作 相应轴组正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使输入到该指令的参数超出输入变量的范围	
	“CircAxes”中指定的轴不在轴组设置的构成轴中		请变更设定，使“CircAxes”中指定的轴在轴组构成中		请确保“CircAxes”中指定的轴在轴组构成中	
“CircAxes”的 2 个轴中指定了同一个轴		请将“CircAxes”中指定的 2 个轴修正为不同的轴		请编写程序，使“CircAxes”中指定的 2 个轴为不同的轴		
附属信息	无					
注意事项 / 备注	无					

事件名称	辅助轴 / 从轴、轴编号非升序			事件代码	547F0000Hex	
内容	运动控制指令中指定的输入变量“Auxiliary”和“Slave”的参数值未按升序排列					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入变量“Auxiliary”和“Slave”的参数非升序		请将该指令的“Auxiliary”和“Slave”的输入参数中指定的轴编号修正为升序		请编写程序，使“Auxiliary”和“Slave”中指定的轴编号为升序	
附属信息	无					
注意事项 / 备注	无					

事件名称	凸轮表属性更新中数据升序异常			事件代码	54800000Hex	
内容	在计算有效数据的过程中，发现相位未按升序排列或者得到计算结果后发现有效数据数为“0”					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	在计算有效数据的过程中，发现相位未按升序排列		请修正凸轮数据，使相位为升序		请使凸轮表数据的相位为升序	
	得到计算结果后发现有效数据数为“0”		请修正凸轮表的数据，使其包含“0”以外的相位		请使凸轮表的数据包含“0”以外的相位	
附属信息	无					
注意事项 / 备注	无					

事件名称	超出 MC 设定写入的对象范围			事件代码	54810000Hex	
内容	运动控制指令的输入变量“Target”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项 / 备注	无					

事件名称	超出主轴移动距离指定范围			事件代码	54820000Hex
内容	运动控制指令的输入变量“MasterDistance”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	指令的输入参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出主轴加速移动距离指定范围			事件代码	54830000Hex
内容	运动控制指令的输入变量“MasterDistanceInACC”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	指令的输入参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出主轴减速移动距离指定范围			事件代码	54840000Hex
内容	运动控制指令的输入变量“MasterDistanceInDEC”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	指令的输入参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出执行模式选择范围			事件代码	54870000Hex
内容	运动控制指令的输入变量“ExecutionMode”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项/备注	无				

事件名称	超出轴间偏差容许值范围			事件代码	54880000Hex
内容	运动控制指令的输入变量“PermittedDeviation”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	不执行当前指令	
系统定义变量	变量名称		数据类型		名称
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项/备注	无				

事件名称	超出通过点位置/中心位置/半径指定范围			事件代码	54890000Hex
内容	运动控制指令的输入变量“AuxPoint”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴组不可动作 相应轴组正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	通过点指定或中心指定时，将“AuxPoint”的值转换为脉冲单位时，带符号超出 40 位的范围 半径指定时，将“AuxPoint[0]”的绝对值转换为脉冲单位时，超出 40 位的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项/备注	无				

事件名称	超出终点指定范围			事件代码	548A0000Hex
内容	运动控制指令的输入变量“EndPoint”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴组不可动作 相应轴组正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active	BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	将指令的输入参数转换为脉冲单位时，带符号超出 40 位的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出从轴移动距离指定范围			事件代码	548B0000Hex
内容	运动控制指令的输入变量“SlaveDistance”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	将指令的输入参数转换为脉冲单位时，超出 40 位的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出相位补偿量范围			事件代码	548C0000Hex
内容	运动控制指令的输入变量“PhaseShift”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	将指令的输入参数绝对值转换为脉冲单位时，超出 40 位的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出固定距离范围			事件代码	548D0000Hex	
内容	运动控制指令的输入变量“FeedDistance”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	将指令的输入参数绝对值转换为脉冲单位时，超出 40 位的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项 / 备注	无					

事件名称	辅助轴 / 从轴相同			事件代码	548E0000Hex	
内容	运动控制指令的输入变量“Auxiliary”和“Slave”中指定的轴相同					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入变量“Auxiliary”和“Slave”的参数相同		请修正参数，确保该指令的输入变量“Auxiliary”和“Slave”中指定的轴不同		请在指令输入变量的辅助轴和从轴中指定不同的轴	
附属信息	无					
注意事项 / 备注	无					

事件名称	超出相对位置选择范围			事件代码	548F0000Hex	
内容	运动控制指令的输入变量“Relative”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项 / 备注	无					

事件名称	超出凸轮转换指定选择范围			事件代码	5490000Hex
内容	运动控制指令的输入变量“CamTransition”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	指令的输入参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项/备注	无				

事件名称	超出同步控制解除模式选择范围			事件代码	5491000Hex
内容	运动控制指令的输入变量“OutMode”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	指令的输入参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无				
注意事项/备注	无				

事件名称	不可执行外部锁定有效指令			事件代码	5492000Hex
内容	针对编码器轴，将输入变量“StopMode”指定为“_mclImmediateStop（立即停止）”，并执行了驱动器模式的MC_TouchProbe（外部锁定有效）指令				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	针对编码器轴，将输入变量“StopMode”指定为“_mclImmediateStop（立即停止）”，并执行了驱动器模式的MC_TouchProbe（外部锁定有效）指令	请修正程序，避免将编码器轴的“StopMode”指定为“_mclImmediateStop（立即停止）”		指定为“_mclImmediateStop（立即停止）”，并执行驱动器模式的外部锁定有效指令时，请设为对伺服轴执行	
附属信息	无				
注意事项/备注	无				

事件名称	超出主轴偏置设定范围			事件代码	54930000Hex	
内容	运动控制指令的输入变量“MasterOffset”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	将指令的输入参数转换为脉冲单位时，带符号超出 40 位的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项 / 备注	无					

事件名称	超出从轴偏置设定范围			事件代码	54940000Hex	
内容	运动控制指令的输入变量“SlaveOffset”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作 相应从轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	将指令的输入参数转换为脉冲单位时，带符号超出 40 位的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项 / 备注	无					

事件名称	超出指令当前位置计数选择范围			事件代码	54950000Hex	
内容	运动控制指令的输入变量“CmdPosMode”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项 / 备注	无					

事件名称	超出主轴齿轮比分子范围			事件代码	54960000Hex
内容	运动控制指令的输入变量“RatioNumeratorMaster”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应从轴不可动作。相应从轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项 / 备注	无				

事件名称	超出主轴齿轮比分母范围			事件代码	54970000Hex
内容	运动控制指令的输入变量“RatioDenominatorMaster”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应从轴不可动作。相应从轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项 / 备注	无				

事件名称	超出辅助轴齿轮比分子范围			事件代码	54980000Hex
内容	运动控制指令的输入变量“RatioNumeratorAuxiliary”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应从轴不可动作。相应从轴正在动作时，则减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项 / 备注	无				

事件名称	超出辅助轴齿轮比分母范围			事件代码	54990000Hex	
内容	运动控制指令的输入变量“RatioDenominatorAuxiliary”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应从轴不可动作。相应从轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项/备注	无					

事件名称	超出主轴位置类型选择范围			事件代码	549A0000Hex	
内容	运动控制指令的输入变量“ReferenceTypeMaster”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应从轴不可动作。相应从轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项/备注	无					

事件名称	超出辅助轴位置类型选择范围			事件代码	549B0000Hex	
内容	运动控制指令的输入变量“ReferenceTypeAuxiliary”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应从轴不可动作。相应从轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项/备注	无					

事件名称	超出目标位置链接计数器范围			事件代码	549C0000Hex
内容	所执行指令的目标位置超出链接计数器范围，因此无法运行				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	在链接计数器范围不含 0 的设定下，执行了高速原点复位	在链接计数器范围不含 0 的设定下，无法执行高速原点复位请修正程序，确保不会执行高速原点复位或者将该轴的链接计数器范围变更为包含 0 的设定		在链接计数器范围不含 0 的设定下，无法执行高速原点复位请编写程序，确保不会执行高速原点复位或者将该轴的链接计数器范围设定为包含 0 的值	
附属信息	无				
注意事项 / 备注	无				

事件名称	超出轴组构成轴设定范围			事件代码	549D0000Hex ^{*1}
内容	运动控制指令的输入变量“Axes”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	相应轴组不可动作。相应轴组正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active	BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	指令的输入参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
	分配了轴组之构成轴的任务不是同一任务	请将该指令的输入变量“Axes”中指定的轴全部分配到同一任务中		轴组的构成轴请指定为分配到同一任务的轴	
附属信息	附属信息 1：异常的详情 01Hex：类型指定异常 02Hex：数组的元素数小于构成轴数 03Hex：指定的轴编号重复、指定轴编号的轴种类在对象范围外、指定的轴编号在范围之外				
注意事项 / 备注	无				

*1. 单元版本为 1.01 以上的环境下 CPU 单元发生的事件。

事件名称	超出轴使用设定范围		事件代码	549E0000Hex *1	
内容	运动控制指令的输入变量“AxisUse”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	MC 共通 / 轴	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	无影响	
系统定义变量	变量名称	数据类型		名称	
	_MC_COM.MFaultLvl.Active	BOOL		MC 共通 发生轻度故障	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	无				
注意事项/备注	执行使用轴的指令发生异常时为轴的异常，执行未使用轴的指令发生异常时为 MC 共通异常。				

*1. 单元版本为 1.04 以上的环境下 CPU 单元发生的事件。

事件名称	超出原点复位参数设定范围		事件代码	57000000Hex *1	
内容	运动控制指令的输入变量“HomingParameter”中指定的参数超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	无影响	
系统定义变量	变量名称	数据类型		名称	
	_MC_COM.MFaultLvl.Active	BOOL		MC 共通 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围
附属信息	<p>附属信息 1：异常的详情</p> <p>1：原点复位动作超出范围、2：原点输入信号超出范围、3：原点复位开始方向超出范围、4：原点检测方向超出范围、5：正方向极限输入时动作超出范围、6：负方向极限输入时动作超出范围、7：原点复位速度超出范围、8：原点复位接近速度超出范围、9：原点复位加速度超出范围、10：原点复位减速度超出范围、11：原点复位跃度超出范围、12：原点输入屏蔽距离超出范围、13：原点位置偏置超出范围、14：原点复位接触时间超出范围、15：原点复位补偿值超出范围、16：原点补偿速度超出范围、100：原点输入屏蔽距离在脉冲单位转换后超过 40 位、101：原点输入屏蔽距离超过环长、102：原点复位补偿值在脉冲单位转换后超过 40 位、103：原点复位补偿值超过环长、104：原点位置偏置在脉冲单位转换后超过 40 位、105：原点位置偏置在环范围外、106：原点复位速度超过最高速度、107：原点复位接近速度超过最高速度、108：未满足原点复位接近速度 ≤ 原点复位速度、109：未满足原点补偿速度 ≤ 原点复位速度、110：原点复位加速度超过最大加速度、111：原点复位减速度超过最大减速度</p>				
注意事项/备注	无				

*1. 单元版本为 1.03 以上的环境下 CPU 单元发生的事件。

事件名称	轴未使用切换异常		事件代码	57020000Hex *1	
内容	对非停止中或指令速度已饱和的轴执行了 MC_ChangeAxisUse（轴使用变更）指令				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	无影响	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	对非停止中或指令速度已饱和的轴执行了 MC_ChangeAxisUse（轴使用变更）指令	请执行异常解除，并待轴停止，在指令速度非饱和的状态下，执行 MC_ChangeAxisUse（轴使用变更）指令 轴变量“Status.Disabled”或“Status.Standstill”为 TRUE 时，该轴处于停止中轴变量“Details.VelLimit”为 TRUE 时，该轴处于指令速度饱和中		请在轴停止且指令速度非饱和的状态下，执行 MC_ChangeAxisUse（轴使用变更）指令	
附属信息	无				
注意事项 / 备注	无				

*1. 单元版本为 1.04 以上的环境下 CPU 单元发生的事件。

事件名称	轴使用不可切换		事件代码	57030000Hex *1	
内容	执行了 MC_ChangeAxisUse（轴使用变更）指令，超出使用实轴最大数				
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	无影响	
系统定义变量	变量名称	数据类型		名称	
	_MC_COM.MFaultLvl.Active	BOOL		MC 共通 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	执行了 MC_ChangeAxisUse（轴使用变更）指令，超出使用实轴最大数	请修正程序，确保不会超出所用 CPU 单元的使用实轴最大数		请编写程序，确保不会超出所用 CPU 单元的使用实轴最大数	
附属信息	无				
注意事项 / 备注	无				

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

事件名称	切换轴使用时运动控制参数设定异常		事件代码	57200000Hex *1		
内容	切换为使用轴的轴运动控制参数设定不正确					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称	数据类型		名称		
	_MC_COM.MFaultLvl.Active	BOOL		MC 共通 发生轻度故障		
发生原因和处理	发生原因（推测原因）	处理对策		防止再发		
	通过 MC_ChangeAxisUse（轴使用变更）指令，将未使用轴切换为使用轴的轴运动控制参数设定不正确	请使用 Sysmac Studio，将异常发生轴的 [轴使用] 变更为 [使用轴]，确认异常位置并修正若未发生异常，请重新变更为 [未使用轴] 后下载		请通过 [使用轴] 的设定确认正常动作后，下载为 [未使用轴]		
	运动控制参数设定的下载过程中断电	请从 Sysmac Studio 下载运动控制参数设定		保存参数设定的过程中请勿关闭电源		
	非易失性存储器故障或非易失性存储器已到使用寿命	若采取上述措施后仍无法解除本异常，请更换 CPU 单元		无		
附属信息	无					
注意事项 / 备注	无					

*1. 单元版本为 1.04 以上的环境下 CPU 单元发生的事件。

事件名称	切换轴使用时必需过程数据对象未设定			事件代码	57210000Hex *1
内容	切换为使用轴的轴种类中，必需对象未设定				
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	无影响	
系统定义变量	变量名称		数据类型		名称
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	切换为使用轴的轴种类中，必需对象未在 PDO 映射设定中设定		请在 Sysmac Studio 的 [PDO 映射设定的编辑] 中，对发生异常的轴设定必需对象必需对象详情，请参考「2-3 PDO 映射 (P.2-30)」。		请通过 [使用轴] 的设定确认正常动作后，下载为 [未使用轴]
	运动控制参数设定的下载过程中断电		请从 Sysmac Studio 下载运动控制参数设定		保存参数设定的过程中请勿关闭电源
	非易失性存储器故障或非易失性存储器已到使用寿命		若采取上述措施后仍无法解除本异常，请更换 CPU 单元		无
	对 [轴使用] 设定为 [未使用轴（不可切换为使用轴）] 的轴，执行了 MC_ChangeAxisUse（轴使用变更）指令		请修正程序，确保不会对 [轴使用] 设定为 [未使用轴（不可切换为使用轴）] 的轴，执行 MC_ChangeAxisUse（轴使用变更）指令		请编写程序，确保不会对 [轴使用] 设定为 [未使用轴（不可切换为使用轴）] 的轴，执行 MC_ChangeAxisUse（轴使用变更）指令
附属信息	无				
注意事项 / 备注	无				

*1. 单元版本为 1.04 以上的环境下 CPU 单元发生的事件。

事件名称	运动指令不可多重启动（主轴）			事件代码	572F0000Hex *1
内容	指令多重启动时，变更了不可变更的输入输出变量“Master”				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令多重启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	无影响	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令多重启动时，变更不可变更的输入输出变量“Master”		请修正程序，确保该指令多重启动时，不变更输入输出变量“Master”的值		请编写程序，确保该指令多重启动时，不变更输入输出变量“Master”的值
附属信息	无				
注意事项 / 备注	无				

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

事件名称	运动指令不可多重启动（位置类型选择）		事件代码	57300000Hex *1		
内容	指令多重启动时，变更了不可变更的输入输出变量“ReferenceType”					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令多重启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令多重启动时，变更不可变更的输入输出变量“ReferenceType”		请修正程序，确保该指令多重启动时，不变更输入输出变量“ReferenceType”的值		请编写程序，确保该指令多重启动时，不变更输入输出变量“ReferenceType”的值	
附属信息	无					
注意事项/备注	无					

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

事件名称	轴参数不可写入		事件代码	573A0000Hex *1		
内容	对未使用轴以外的轴执行了该指令					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	对使用轴或未创建轴执行了该指令		请修正程序，确保在 MC_ChangeAxisUse（轴使用变更）指令中，将对象轴变更为未使用后启动该指令		请编写程序，确保该指令启动时，对象轴变为未使用轴	
附属信息	无					
注意事项/备注	无					

*1. 单元版本为 1.08 以上的环境下 CPU 单元发生的事件。

事件名称	超出轴参数设定范围		事件代码	573B0000Hex *1		
内容	运动控制指令的输入变量“AxisParameter”中指定的参数超出有效范围					
发生位置	运动控制功能模块	发生位置详情	MC 共通	检测时序	指令启动时	
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称	数据类型	名称			
	_MC_COM.MFaultLvl.Active	BOOL	MC 共通 发生轻度故障			
发生原因和处理	发生原因（推测原因）	处理对策	防止再发			
	指令的输入变量“AxisParameter”中指定的参数超出输入变量的范围	请修正参数，避免超出该指令的输入变量范围 超出范围的参数或参数间的不匹配问题可通过附属信息确认	请勿使指令的输入参数超出输入变量的范围 输入变量的有效范围请参考 MC_WriteAxisParameter（轴参数写入）指令			
附属信息	<p>附属信息 1: 异常的详情</p> <ul style="list-style-type: none"> 范围检查的详情代码 <p>0000Hex: 显示单位超出范围、0001Hex: 电机单圈脉冲数超出范围、0002Hex: 电机单元移动量超出范围、0003Hex: 工件侧单圈移动量超出范围、0004Hex: 工件侧齿轮比超出范围、0005Hex: 电机侧齿轮比超出范围、0100Hex: 最高速度超出范围、0101Hex: 启动速度超出范围、0102Hex: 点动最高速度超出范围、0103Hex: 最大加速度超出范围、0104Hex: 最大减速度超出范围、0105Hex: 加减速超限超出范围、0106Hex: 反转时动作超出范围、0107Hex: 速度警告值超出范围、0108Hex: 加速度警告值超出范围、0109Hex: 减速度警告值超出范围、010AHex: 正方向扭矩警告值超出范围、010BHex: 负方向扭矩警告值超出范围、010CHex: 位置宽度超出范围、010DHex: 位置检查时间超出范围、010EHex: 反馈速度过滤器时间常数超出范围、010FHex: 原点位置范围超出范围、0200Hex: 立即停止输入停止方法超出范围、0201Hex: 极限输入停止方法超出范围、0202Hex: 驱动器错误重置监视时间超出范围、0203Hex: 正方向扭矩限制上限值超出范围、0204Hex: 负方向扭矩限制上限值超出范围、0300Hex: 软件限位功能超出范围、0301Hex: 正方向软件限位超出范围、0302Hex: 负方向软件限位超出范围、0303Hex: 位置偏差超限值超出范围、0304Hex: 位置偏差警告值超出范围、0400Hex: 计数模式超出范围、0401Hex: 循环计数器上限设定值超出范围、0402Hex: 循环计数器下限设定值超出范围、0500Hex: 原点复位动作超出范围、0501Hex: 原点输入信号超出范围、0502Hex: 原点复位开始方向超出范围、0503Hex: 原点检测方向超出范围、0504Hex: 正方向极限输入时动作超出范围、0505Hex: 负方向极限输入时动作超出范围、0506Hex: 原点复位速度超出范围、0507Hex: 原点复位接近速度超出范围、0508Hex: 原点复位加速度超出范围、0509Hex: 原点复位减速度超出范围、050AHex: 原点复位跃度超出范围、050BHex: 原点输入屏蔽距离超出范围、050CHex: 原点位置偏置超出范围、050DHex: 原点复位接触时间超出范围、050EHex: 原点复位补偿值超出范围、050FHex: 原点复位补偿速度超出范围</p> <ul style="list-style-type: none"> 完整性检查的详情代码 <p>1000Hex: 工件侧单圈移动量 × 工件侧齿轮比 / 电机侧齿轮比的值超出 0.00000001 ~ 231 的范围、1001Hex: 电机单圈脉冲数 × 电机侧齿轮比的值超过 40 位、1100Hex: 脉冲单位转换后的最高速度超过上限值²、1100Hex: 最高速度在脉冲单位转换后超过 500MHz、1101Hex: 启动速度超过最高速度、1102Hex: 点动最高速度超过最高速度、1103Hex: 位置宽度在脉冲单位转换后超过 40 位、1104Hex: 原点位置范围在脉冲单位转换后超过 40 位、1300Hex: 正方向软件限位在脉冲单位转换后超过 40 位、1301Hex: 负方向软件限位在脉冲单位转换后超过 40 位、1302Hex: 未满足正方向软件限位 > 负方向软件限位、1303Hex: 位置偏差超限值在脉冲单位转换后超过 40 位、1304Hex: 未满足位置偏差超限值 ≥ 位置偏差警告值、1400Hex: 循环计数器上限设定值在脉冲单位转换后超过 40 位、1401Hex: 循环计数器下限设定值在脉冲单位转换后超过 40 位、1402Hex: 未满足循环计数器上限设定值 > 循环计数器下限设定值、1403Hex: 循环计数器上限设定值 - 循环计数器下限设定值 在脉冲单位转换后未满足 2 以上、1500Hex: 原点复位速度超过最高速度、1501Hex: 未满足原点复位接近速度 ≤ 原点复位速度、1502Hex: 原点复位加速度超过最大加速度、1503Hex: 原点复位减速度超过最大减速度、1504Hex: 原点输入屏蔽距离在脉冲单位转换后超过 40 位、1505Hex: 原点输入屏蔽距离超过环长、1506Hex: 原点位置偏置在脉冲单位转换后超过 40 位、1507Hex: 原点位置偏置超出环范围、1508Hex: 原点复位补偿值的绝对值在脉冲单位转换后超过 40 位、1509Hex: 原点复位补偿值的绝对值超过环长、150AHex: 未满足原点复位补偿速度 ≤ 最高速度</p> <p>(注) 即使在发生多个异常时, 显示的代码只有 1 个。显示的优先顺序为“范围检查的详情代码”→“完整性检查的详情代码”。</p>					
注意事项 / 备注	无					

*1. 单元版本为 1.08 以上的环境下 CPU 单元发生的事件。

*2. 单元版本 Ver.1.10 以下时最高速度的上限值为 500MHz；单元版本 Ver.1.11 以上时最高速度的上限值为 2,147,483,647Hz。

事件名称	超出凸轮属性设定范围		事件代码	573C0000Hex *1	
内容	运动控制指令的输入变量“CamProperty”中指定的参数超出有效范围				
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	无影响	
系统定义变量	变量名称		数据类型		名称
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入变量“CamProperty”中指定的参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围 超出范围的参数可通过附属信息确认		请勿使指令的输入参数超出输入变量的范围
附属信息	附属信息 1：异常的详情 0000Hex：初始速度超出范围 0001Hex：初始加速度超出范围 0002Hex：周期时间超出范围				
注意事项 / 备注	无				

*1. 单元版本为 1.08 以上的环境下 CPU 单元发生的事件。

事件名称	超出凸轮节点设定范围		事件代码	573D0000Hex *1	
内容	运动控制指令的输入变量“CamNodes”中指定的参数超出有效范围				
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序 指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别 系统
产生的影响	用户程序	继续	动作	无影响	
系统定义变量	变量名称		数据类型		名称
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入变量“CamNodes”中指定的参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围 超出范围的参数可通过附属信息确认		请勿使指令的输入参数超出输入变量的范围
附属信息	附属信息 1：异常的详情 0000Hex：主轴相位超出范围 0001Hex：从轴位移超出范围 0002Hex：曲线形状超出范围 0003Hex：连接速度超出范围 0004Hex：连接加速度超出范围 0005Hex：相位间隔宽度超出范围 附属信息 2：变为异常的节点要素编号				
注意事项 / 备注	无				

*1. 单元版本为 1.08 以上的环境下 CPU 单元发生的事件。

事件名称	凸轮节点类型指定不正确			事件代码	573E0000Hex *1	
内容	运动控制指令的输入变量“CamNodes”中指定的参数不是_sMC_CAM_NODE型排列变量					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入变量“CamNodes”中指定的参数不是_sMC_CAM_NODE型排列变量		请修正程序，使其在该指令的输入变量中指定_sMC_CAM_NODE型排列变量		请编写程序，使其在该指令的输入变量中指定_sMC_CAM_NODE型排列变量	
附属信息	无					
注意事项/备注	无					

*1. 单元版本为 1.08 以上的环境下 CPU 单元发生的事件。

事件名称	凸轮表生成节点数不足			事件代码	573F0000Hex *1	
内容	运动控制指令的输入变量“CamNodes”中指定的参数之排列变量中，要素编号 0 的 Phase 值为“0”					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入变量“CamNodes”中指定的参数之排列变量中，要素编号 0 的 Phase（主轴相位）值为“0”		请修正程序，确保输入变量“CamNodes”中指定的参数之排列变量中，要素编号 0 的 Phase（主轴相位）值不是“0”		请编写程序，确保输入变量“CamNodes”中指定的参数之排列变量中，要素编号 0 的 Phase（主轴相位）值不是“0”	
附属信息	无					
注意事项/备注	无					

*1. 单元版本为 1.08 以上的环境下 CPU 单元发生的事件。

事件名称	凸轮节点主轴相位非升序			事件代码	57400000Hex *1	
内容	运动控制指令的输入变量“CamNodes”中指定的参数之排列变量中，Phase 的值未按要素编号顺序升序排列					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入变量“CamNodes”中指定的参数之排列变量中，Phase（主轴相位）的值未按要素编号顺序升序排列或者有效位数第 8 位及之后的舍去结果不是升序		请修正程序，确保输入变量“CamNodes”中指定的参数之排列变量中，Phase（主轴相位）的值按要素编号顺序升序排列		请编写程序，确保输入变量“CamNodes”中指定的参数之排列变量中，Phase（主轴相位）的值按要素编号顺序升序排列	
附属信息	附属信息 1：变为异常的节点要素编号					
注意事项/备注	无					

*1. 单元版本为 1.08 以上的环境下 CPU 单元发生的事件。

事件名称	凸轮表生成数据数过多			事件代码	57410000Hex *1	
内容	生成的凸轮数据数超出运动控制指令的输入变量“CamTable”中指定的凸轮数据变量的排列要素数					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	所生成凸轮表的凸轮数据数超出指令的输入变量“CamTable”中指定的凸轮数据变量的排列要素数		请修正程序，确保所生成凸轮表的凸轮数据数不会超出指令的输入变量“CamTable”中指定的凸轮数据变量的排列要素数 生成的凸轮表的凸轮数据数请参考 MC_GenerateCamTable（凸轮表生成）指令		请编写程序，确保所生成凸轮表的凸轮数据数不会超出指令的输入变量“CamTable”中指定的凸轮数据变量的排列要素数 生成的凸轮表的凸轮数据数请参考 MC_GenerateCamTable（凸轮表生成）指令	
附属信息	附属信息 1：变为异常的节点要素编号					
注意事项/备注	无					

*1. 单元版本为 1.08 以上的环境下 CPU 单元发生的事件。

事件名称	凸轮表生成位移溢出			事件代码	57420000Hex *1	
内容	所生成凸轮表的 Distance 超出 REAL 型可表现的范围					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	指令启动时 / 指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	所生成凸轮表的 Distance 超出 REAL 型可表现的范围		输入变量“CamNodes”的 Curve（曲线形状）指定为 3 次曲线或 5 次曲线时，请修正 InitVel（初始速度）、ConnectingVel（连接速度）、ConnectingAcc（连接加速度）的值，确保 Distance 不会发生溢出。 Distance 的计算方法请参考 MC_GenerateCamTable（凸轮表生成）指令		输入变量“CamNodes”的 Curve（曲线形状）指定为 3 次曲线或 5 次曲线时，请指定 InitVel（初始速度）、ConnectingVel（连接速度）、ConnectingAcc（连接加速度）的值，确保 Distance 不会发生溢出。 Distance 的计算方法请参考 MC_GenerateCamTable（凸轮表生成）指令	
附属信息	附属信息 1：变为异常的节点要素编号					
注意事项/备注	无					

*1. 单元版本为 1.08 以上的环境下 CPU 单元发生的事件。

事件名称	生成中止凸轮表的使用		事件代码	57430000Hex *1		
内容	在该指令的输入变量“CamTable”中指定了中止生成的凸轮数据变量					
发生位置	运动控制功能模块	发生位置详情	MC 共通 / 轴	检测时序	指令启动时	
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称	数据类型		名称		
	_MC_COM.MFaultLvl.Active	BOOL		MC 共通 发生轻度故障		
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障		
发生原因和处理	发生原因（推测原因）	处理对策		防止再发		
	因 MC_GenerateCamTable（凸轮表生成）指令异常而中止生成的凸轮数据变量，指定为了该指令的输入变量“CamTable”	请确认 MC_GenerateCamTable（凸轮表生成）指令的输出变量 ErrorID（错误代码）、ErrorParameterCode（参数详细代码）、ErrorNodePointIndex（节点要素编号），并修正程序，确保生成正常的凸轮数据变量		请编写程序，确保 MC_GenerateCamTable（凸轮表生成）指令生成正常的凸轮数据变量或者确保仅当 MC_GenerateCamTable（凸轮表生成）指令正常完成时，启动该指令		
附属信息	无					
注意事项 / 备注	无					

*1. 单元版本为 1.08 以上的环境下 CPU 单元发生的事件。

事件名称	超出执行 ID 设定范围		事件代码	57490000Hex *1		
内容	运动控制指令的输入变量“ExecID”中指定的参数超出范围					
发生位置	运动控制功能模块	发生位置详情	轴	检测时序	指令启动时	
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴正在动作时，则减速停止		
系统定义变量	变量名称	数据类型		名称		
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障		
发生原因和处理	发生原因（推测原因）	处理对策		防止再发		
	指令的输入变量“ExecID”中指定的参数超出输入变量的范围	请修正程序，使指令的输入变量“ExecID”中指定的输入参数在范围内		请编写程序，使指令的输入变量“ExecID”中指定的输入参数在范围内		
附属信息	无					
注意事项 / 备注	无					

*1. 单元版本为 1.10 以上的环境下 CPU 单元发生的事件。

事件名称	超出位置偏置范围		事件代码	574A0000Hex *1		
内容	运动控制指令的输入变量“OffsetPosition”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	将指令的输入参数转换为脉冲单位时，带符号超出 40 位的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项 / 备注	无					

*1. 单元版本为 1.10 以上的环境下 CPU 单元发生的事件。

事件名称	超出 PDS 状态变化指令选择范围		事件代码	574B0000Hex *1		
内容	运动控制指令的输入变量“TransitionCmd”中指定的参数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入参数超出输入变量的范围		请修正参数，避免超出该指令的输入变量范围		请勿使指令的输入参数超出输入变量的范围	
附属信息	无					
注意事项 / 备注	无					

*1. 单元版本为 1.10 以上的环境下 CPU 单元发生的事件。

事件名称	目标位置正方向软件超限		事件代码	64400000Hex		
内容	指定的位置超出正方向软件限位					
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作。相应轴正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止		
系统定义变量	变量名称	数据类型		名称		
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障		
	_MC_GRP[*].MFaultLvl.Active	BOOL		轴组 发生轻度故障		
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入变量“Position”中指定的参数超出正方向软件限位		请修正该指令的输入变量“Position”中指定的参数，使其在正方向软件限位的范围内		请确保指令的输入变量“Position”中指定的输入参数在正方向软件限位的范围内	
	开始位置超出正方向软件限位的范围，执行了向软件限位范围相反方向动作的指令		请修正程序，确保该指令下的移动方向在正方向软件限位的范围内		请编写程序，确保在开始位置超出正方向软件限位范围时，移动方向在正方向软件限位的范围内	
	设定为通过点指定的MC_MoveCircular2D（2轴圆弧插补）指令的输入变量“AuxPoint”中指定的参数超出正方向软件限位范围		请修正该指令的输入变量“AuxPoint”中指定的参数，使其在正方向软件限位的范围内		请确保通过点指定的MC_MoveCircular2D（2轴圆弧插补）指令的输入变量“AuxPoint”中指定的参数在正方向软件限位范围内	
附属信息	附属信息 1：因发生源详情不同而异 发生源为轴时：“0” 发生源为轴组时：逻辑轴编号					
注意事项 / 备注	无					

事件名称	目标位置负方向软件超限		事件代码	64410000Hex	
内容	指定位置超出负方向软件限位范围				
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作的相应轴。正在动作时，则减速停止 发生位置详情为“轴组”时： 相应轴组不可动作。相应轴组正在动作时，则减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
	_MC_GRP[*].MFaultLvl.Active	BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	指令的输入变量“Position”中指定的参数超出负方向软件限位范围	请修正该指令的输入变量“Position”中指定的参数，使其在负方向软件限位的范围内		请确保指令的输入变量“Position”中指定的输入参数在负方向软件限位的范围内	
	开始位置超出负方向软件限位的范围，执行了向软件限位范围相反方向动作的指令	请修正程序，确保该指令下的移动方向在负方向软件限位的范围内		请编写程序，确保在开始位置超出负方向软件限位范围时，移动方向在负方向软件限位的范围内	
附属信息	设定为通过点指定的MC_MoveCircular2D（2轴圆弧插补）指令的输入变量“AuxPoint”中指定的参数超出负方向软件限位范围 请修正该指令的输入变量“AuxPoint”中指定的参数，使其在负方向软件限位的范围内 请确保通过点指定的MC_MoveCircular2D（2轴圆弧插补）指令的输入变量“AuxPoint”中指定的参数在负方向软件限位范围内				
附属信息	附属信息 1：因发生源详情不同而异 发生源为轴时：“0” 发生源为轴组时：逻辑轴编号				
注意事项 / 备注	无				

事件名称	正在发生指令位置溢出 / 下溢		事件代码	64420000Hex	
内容	正在发生指令位置溢出 / 下溢时，执行了定位、向溢出 / 下溢方向移动的指令或方向不确定的指令				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴不可动作。	相应轴正在动作时，则减速停止
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	正在发生指令位置溢出 / 下溢时执行了以下指令 <ul style="list-style-type: none"> • 执行了定位指令 • 执行了向溢出 / 下溢方向移动的连续控制指令 • 执行了方向不确定的指令（同步功能、扭矩控制） 	请执行异常解除，通过原点复位或当前位置重置，解除溢出 / 下溢状态		请确保不会发生溢出 / 下溢	
附属信息	无				
注意事项 / 备注	无				

事件名称	正方向极限输入中			事件代码	64430000Hex	
内容	在正方向极限输入为“ON”的状态下执行了向正方向移动的指令					
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作 发生位置详情为“轴组”时： 相应轴组不可动作		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
		_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	在正方向极限输入为“ON”的状态下执行了向正方向移动的指令或在正方向极限输入为“ON”的状态下执行了未指定移动方向的指令在正方向极限输入为“ON”的状态下执行了轴组动作指令		请执行异常解除，向负方向执行复位动作在轴组动作指令的执行过程中发生异常时，请将发生异常的轴组设为无效，再执行上述复位动作若仍然发生本异常，请确认正方向极限信号的连接、正方向极限输入的逻辑设定、启动指令的执行条件，并修正错误。 逻辑设定请确认轴参数和从站端两者的设定		请确认正方向极限信号的连接、正方向极限输入的逻辑设定、启动指令的执行条件没有错误。 逻辑设定请确认轴参数和从站端两者的设定	
附属信息	附属信息 1：因发生源详情不同而异 发生源为轴时：“0” 发生源为轴组时：逻辑轴编号					
注意事项 / 备注	无					

事件名称	负方向极限输入中			事件代码	64440000Hex	
内容	在负方向极限输入为“ON”的状态下执行了向负方向移动的指令					
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	发生位置详情为“轴”时： 相应轴不可动作 发生位置详情为“轴组”时： 相应轴组不可动作		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
		_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	在负方向极限输入为“ON”的状态下执行了向负方向移动的指令或在负方向极限输入为“ON”的状态下执行了未指定移动方向的指令在负方向极限输入为“ON”的状态下执行了轴组动作指令		请执行异常解除，向正方向执行复位动作在轴组动作指令的执行过程中发生异常时，请将发生异常的轴组设为无效，再执行上述复位动作若仍然发生本异常，请确认负方向极限信号的连接、负方向极限输入的逻辑设定、启动指令的执行条件，并修正错误。 逻辑设定请确认轴参数和从站端两者的设定		请确认负方向极限信号的连接、负方向极限输入的逻辑设定、启动指令的执行条件没有错误。 逻辑设定请确认轴参数和从站端两者的设定	
附属信息	附属信息 1：因发生源详情不同而异 发生源为轴时：“0” 发生源为轴组时：逻辑轴编号					
注意事项 / 备注	无					

事件名称	伺服主回路电源关闭状态		事件代码	74220000Hex	
内容	在伺服驱动器的伺服主回路电源为“OFF”的状态下执行了伺服 ON				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别
产生的影响	用户程序	继续	动作	相应轴的伺服“OFF”	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	在伺服驱动器的伺服主回路电源为“OFF”的状态下执行伺服 ON		请将发生本异常的轴的伺服驱动器主回路电源设为“ON”，再打开伺服		请将伺服驱动器的主回路电源设为“ON”，再打开伺服
附属信息	无				
注意事项 / 备注	无				

事件名称	正在发生反馈位置溢出 / 下溢		事件代码	57220000Hex *1	
内容	正在发生反馈位置溢出 / 下溢时，启动了无法执行的指令				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序
异常属性	重要程度	监视信息	恢复方法	—	日志类别
产生的影响	用户程序	继续	动作	无影响	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	正在发生反馈位置溢出或下溢时，启动了无法执行的指令		请执行异常解除，通过“当前位置变更”或“原点复位”解除溢出或下溢状态		请编写程序，确保不会发生溢出或下溢
附属信息	无				
注意事项 / 备注	无				

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

事件名称	超出切换结构体的轨道编号设定范围		事件代码	57230000Hex *1	
内容	运动控制指令的输入输出变量“Switches”中指定的“TrackNumber”的值超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序
异常属性	重要程度	监视信息	恢复方法	—	日志类别
产生的影响	用户程序	继续	动作	无影响	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中
发生原因和处理	发生原因（推测原因）		处理对策		防止再发
	指令的输入输出变量中指定的结构体成员值的成员值超出范围		请修正值，确保该指令的输入输出变量中指定的结构体成员不会超出有效范围		请确保指令的输入输出变量中指定的结构体成员不会超出有效范围
附属信息	无				
注意事项 / 备注	无				

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

事件名称	超出切换结构体的 ON 开始位置设定范围		事件代码	57240000Hex *1		
内容	运动控制指令的输入输出变量“Switches”中指定的“FirstOnPosition”的值超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入输出变量中指定的结构体型变量的成员值超出范围		请修正值，确保该指令的输入输出变量中指定的结构体型变量的成员不会超出有效范围		请确保指令的输入输出变量中指定的结构体型变量的成员不会超出有效范围	
附属信息	无					
注意事项 / 备注	无					

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

事件名称	超出切换结构体的 ON 结束位置设定范围		事件代码	57250000Hex *1		
内容	运动控制指令的输入输出变量“Switches”中指定的“LastOnPosition”的值超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入输出变量中指定的结构体型变量的成员值超出范围		请修正值，确保该指令的输入输出变量中指定的结构体型变量的成员不会超出有效范围		请确保指令的输入输出变量中指定的结构体型变量的成员不会超出有效范围	
附属信息	无					
注意事项 / 备注	无					

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

事件名称	超出切换结构体的方向选择范围		事件代码	57260000Hex *1		
内容	运动控制指令的输入输出变量“Switches”中指定的“AxisDirection”的值超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入输出变量中指定的结构体型变量的成员值超出范围		请修正值，确保该指令的输入输出变量中指定的结构体型变量的成员不会超出有效范围		请确保指令的输入输出变量中指定的结构体型变量的成员不会超出有效范围	
附属信息	无					
注意事项 / 备注	无					

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

事件名称	超出切换结构体的切换模式选择范围		事件代码	57270000Hex *1	
内容	运动控制指令的输入输出变量“Switches”中指定的“CamSwitchMode”的值超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	监视信息	恢复方法	—	日志类别 系统
产生的影响	用户程序	继续	动作	无影响	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].Obsr.Active	BOOL		轴 监视信息发生中	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	指令的输入输出变量中指定的结构体型变量的成员值超出范围	请修正值，确保该指令的输入输出变量中指定的结构体型变量的成员不会超出有效范围		请确保指令的输入输出变量中指定的结构体型变量的成员不会超出有效范围	
附属信息	无				
注意事项/备注	无				

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

事件名称	超出切换结构体的 ON 时间设定范围		事件代码	57280000Hex *1	
内容	运动控制指令的输入输出变量“Switches”中指定的“Duration”的值超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	监视信息	恢复方法	—	日志类别 系统
产生的影响	用户程序	继续	动作	无影响	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].Obsr.Active	BOOL		轴 监视信息发生中	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	指令的输入输出变量中指定的结构体型变量的成员值超出范围	请修正值，确保该指令的输入输出变量中指定的结构体型变量的成员不会超出有效范围		请确保指令的输入输出变量中指定的结构体型变量的成员不会超出有效范围	
附属信息	无				
注意事项/备注	无				

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

事件名称	超出轨道选项结构体的 ON 时刻修正设定范围		事件代码	57290000Hex *1	
内容	运动控制指令的输入输出变量“TrackOptions”中指定的“OnCompensation”的值超出范围				
发生位置	运动控制功能模块		发生位置详情	轴	检测时序 指令启动时
异常属性	重要程度	监视信息	恢复方法	—	日志类别 系统
产生的影响	用户程序	继续	动作	无影响	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].Obsr.Active	BOOL		轴 监视信息发生中	
发生原因和处理	发生原因（推测原因）	处理对策		防止再发	
	指令的输入输出变量中指定的结构体型变量的成员值超出范围	请修正值，确保该指令的输入输出变量中指定的结构体型变量的成员不会超出有效范围		请确保指令的输入输出变量中指定的结构体型变量的成员不会超出有效范围	
附属信息	无				
注意事项/备注	无				

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

事件名称	超出轨道选项结构体的 OFF 时刻修正设定范围		事件代码	572A0000Hex *1		
内容	运动控制指令的输入输出变量“TrackOptions”中指定的“OffCompensation”的值超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入输出变量中指定的结构体变量的成员值超出范围		请修正值，确保该指令的输入输出变量中指定的结构体变量的成员不会超出有效范围		请确保指令的输入输出变量中指定的结构体变量的成员不会超出有效范围	
附属信息	无					
注意事项 / 备注	无					

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

事件名称	超出切换结构体变量的排列要素数范围		事件代码	572B0000Hex *1		
内容	运动控制指令的输入输出变量“Switches”中指定的结构体变量的排列要素数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	监视信息	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入输出变量中指定的结构体变量的排列要素数超出范围		请进行修正，确保该指令的输入输出变量中指定的结构体变量的排列要素数不会超出范围		请确保指令的输入输出变量中指定的结构体变量的排列要素数不会超出范围	
附属信息	无					
注意事项 / 备注	无					

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

事件名称	超出输出信号结构体变量的排列要素数范围		事件代码	572C0000Hex *1		
内容	运动控制指令的输入输出变量“Outputs”中指定的结构体变量的排列要素数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入输出变量中指定的结构体变量的排列要素数超出范围		请进行修正，确保该指令的输入输出变量中指定的结构体变量的排列要素数不会超出范围		请确保指令的输入输出变量中指定的结构体变量的排列要素数不会超出范围	
附属信息	无					
注意事项 / 备注	无					

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

事件名称	超出轨道选项结构体型变量的排列要素数范围		事件代码	572D0000Hex *1		
内容	运动控制指令的输入输出变量“TrackOptions”中指定的结构体型变量的排列要素数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入输出变量中指定的结构体型变量的排列要素数超出范围		请进行修正，确保该指令的输入输出变量中指定的结构体型变量的排列要素数不会超出范围		请确保指令的输入输出变量中指定的结构体型变量的排列要素数不会超出范围	
附属信息	无					
注意事项/备注	无					

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

事件名称	输出信号和轨道选项的排列要素数不一致		事件代码	572E0000Hex *1		
内容	运动控制指令的输入输出变量“Outputs”和“TrackOptions”中指定的结构体型变量的排列要素数不一致					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入输出变量中指定的输出信号结构体型变量和轨道选项结构体型变量的排列要素数不一致		请将该指令的输入输出变量中指定的输出信号结构体型变量和轨道选项结构体型变量的排列要素数修正为一致		请确保指令的输入输出变量中指定的输出信号结构体型变量和轨道选项结构体型变量的排列要素数一致	
附属信息	无					
注意事项/备注	无					

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

事件名称	超出切换结构体的同一轨道指定数范围		事件代码	57310000Hex *1		
内容	运动控制指令的输入输出变量“Switches”中指定的“TrackNumber”的同一轨道编号数超出范围					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	无影响		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令的输入输出变量中指定的切换结构体型变量的“TrackNumber”中，指定的同一轨道编号数超出 1 个轨道上可指定的个数		请修正值，确保“TrackNumber”中指定的同一轨道编号数不会超出 1 个轨道上可指定的个数		请确保“TrackNumber”中指定的同一轨道编号数不会超出 1 个轨道上可指定的个数	
附属信息	无					
注意事项/备注	无					

*1. 单元版本为 1.06 以上的环境下 CPU 单元发生的事件。

A-3 多重启动可否

运动指令多重启动是否可以执行，取决于当前的轴状态或当前的轴组状态以及所执行的指令的种类。下面对这些条件的关系进行说明。

关于各指令的多重启动的详情，请参考 □□ “NJ 系列 CPU 单元用户手册 运动控制篇 (SBCE-CN5-363) ”。



使用注意事项

- 1 个单轴只能缓存 1 个多重启动。若执行 2 个以上指令的多重启动，指令会发生异常。
 - 1 个轴组最多能缓存 7 个多重启动。若执行 8 个以上指令的多重启动，指令会发生异常。
 - 对于根据轴指令动作的轴，不能多重启动轴组指令。对于根据轴组指令动作的轴，不能多重启动轴指令。若执行上述不允许的多重启动，指令会发生异常。
-

A-3-1 轴的状态、轴组的状态

可否进行多重启动会因当前轴状态或当前轴组状态而改变。轴的状态及轴组的状态可通过相应轴的轴变量及轴组变量的值得知。

关于轴的状态、轴组的状态、轴变量、轴组变量的详情，请参考 □ “NJ 系列 CPU 单元用户手册 运动控制篇（SBCE-CN5-363）”。

表示轴状态的轴变量

表示轴状态的轴变量有以下 9 种。这些变量是排他性的，只有其中 1 个为 TRUE。

下面以 `_MC_AX[0-255]` 为例进行说明。`_MC1_AX[0-255]`、`_MC2_AX[0-255]` 亦同。

变量名称	数据类型	名称	功能
<code>_MC_AX[0-255]</code>	<code>_sAXIS_REF</code>	轴变量	
Status	<code>_sAXIS_REF_STA</code>	轴状态	
Disabled	BOOL	轴无效	轴为伺服 OFF、处于停止时变为 TRUE。
Standstill	BOOL	停止中	轴为伺服 ON、处于停止时变为 TRUE。
Descrete	BOOL	定位动作中	正在向目标位置执行定位控制时变为 TRUE。 包括在定位动作中将超驰值设定为“0”后速度变为“0”的状态。
Continuous	BOOL	连续动作中	没有目标位置，连续动作时变为 TRUE。 速度控制或扭矩控制时变为该状态。 包括将目标速度设为“0”后速度为“0”的状态或在连续动作中将超驰值设定为“0”后速度变为“0”的状态。
Synchronized	BOOL	同步动作中	正在执行同步控制时变为 TRUE。 包括切换同步控制指令后的同步等待状态。
Homing	BOOL	原点复位中	正在通过 <code>MC_Home</code> 指令、 <code>MC_HomeWithParameter</code> 指令执行原点复位时变为 TRUE。
Stopping	BOOL	减速停止中	利用 <code>MC_Stop</code> 指令或 <code>MC_TouchProbe</code> 指令停止轴时，轴完全停止前为 TRUE。 包括利用 <code>MC_Stop</code> 指令停止轴后， <code>Execute</code> 为 TRUE 时。 此状态下，无法启动轴动作指令（ <code>CommandAborted=TRUE</code> ）。
ErrorStop	BOOL	错误减速停止中	因 <code>MC_ImmediateStop</code> 指令或轴发生轻度故障（ <code>_MC_AX[*]MFaultLvl.Active=TRUE</code> ），轴停止前及停止时变为 TRUE。 此状态下，无法启动轴动作指令（ <code>CommandAborted=TRUE</code> ）。
Coordinated	BOOL	多轴协调动作中	因多轴协调指令，轴组处于有效时变为 TRUE。

表示轴组状态的轴组变量

表示轴组状态的轴组变量有以下 5 种。这些变量是排他性的，只有其中 1 个为 TRUE。

下面以 `_MC_GRP[0-63]` 为例进行说明。

`_MC1_GRP[0-63]`、`_MC2_GRP[0-63]` 亦同。

变量名称	数据类型	名称	功能
<code>_MC_GRP[0-63]</code>	<code>_sGROUP_REF</code>	轴组变量	
Status	<code>_sGROUP_REF_STA</code>	轴组状态	
Disabled	BOOL	轴组无效	轴组无效、处于停止状态时变为 TRUE。
Standby	BOOL	停止中	轴组动作指令处于停止状态时变为 TRUE。 与轴组构成轴的伺服 ON/OFF 状态无关。
Moving	BOOL	动作中	正在向目标位置执行轴组动作指令时变为 TRUE。 包括位置等待的状态或因超驰而使速度变为“0”的状态。
Stopping	BOOL	减速停止中	利用 <code>MC_GroupStop</code> 指令停止轴组时，轴组完全停止前为 TRUE。 包括利用 <code>MC_GroupStop</code> 指令停止轴组后， <code>Execute</code> 为 TRUE 时。 此状态下，无法启动轴动作指令（ <code>CommandAborted=TRUE</code> ）。
ErrorStop	BOOL	错误减速停止中	因 <code>MC_GroupImmediateStop</code> 指令或轴组发生轻度故障（ <code>_MC_GRP[*].MFaultLvl.Active=TRUE</code> ），轴组停止前及停止时变为 TRUE。 此状态下，无法启动轴组动作指令（ <code>CommandAborted=TRUE</code> ）。

A-3-2 状态切换和多重启动的可否

以下对各轴的状态以及各轴组的状态下是否可以各指令的多重启动进行说明。同时，可以进行各指令的多重启动进行说明。

关于运动控制功能模块的状态切换以及与状态切换有关的运动控制指令的详情，请参考 □ “NJ 系列 CPU 单元用户手册 运动控制篇（SBCE-CN5-363）”。

对伺服轴、虚拟伺服轴的指令的多重启动

指令对象为伺服轴及伺服轴组时的多重启动可否及状态切换目标如下表所示。
多重启动可否以表格单元的颜色来表示。

- 白色 : 可多重启动
- 灰色 : 不可多重启动。发生异常。
- 黄色 : 不可多重启动。CommandAborted（执行中断）变为 TRUE，指令执行无效。

以下是表内的符号所表示的切换目标的状态。
轴的状态

- A : 轴无效（Disabled）
- B : 停止中（Standstill）
- C : 定位动作中（Discrete）
- D : 连续动作中（Continuous）
- E : 同步动作中（Synchronized）
- F : 原点复位中（Homing）
- G : 减速动作中（Stopping）
- H : 错误减速停止中（ErrorStop）
- I : 多轴协调动作中（Coordinated）

轴组的状态

- a : 轴组无效（Disabled）
- b : 停止中（Standby）
- c : 动作中（Moving）
- d : 减速动作中（Stopping）
- e : 错误减速停止中（ErrorStop）

● 轴指令

指令	执行指令前的轴的状态									执行指令前的轴组的状态				
	A	B	C	D	E	F	G	H	I	a	b	c	d	e
	轴无效	停止中	定位动作中	连续动作中	同步动作中	原点复位中	减速停止中	错误减速停止中	多轴协调动作中	轴组无效	停止中	动作中	减速停止中	错误减速停止中
MC_Power(Enable=TRUE)	*1	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_Power(Enable=FALSE)	A	A	A	A	A	A	A	H	I	a	b	*2	d	e
MC_MoveJog	H	D	H	H	H	H	G	H	I	a	e	e	e	e
MC_Home	H	F	H	H	H	H	G	H	I	a	e	e	e	e
MC_HomeWithParameter	H	F	H	H	H	H	G	H	I	a	e	e	e	e
MC_Move	H	C	C	C	C	H	G	H	I	a	e	e	e	e

指令	执行指令前的轴的状态									执行指令前的轴组的状态				
	A	B	C	D	E	F	G	H	I	a	b	c	d	e
	轴无效	停止中	定位动作中	连续动作中	同步动作中	原点复位中	减速停止中	错误减速停止中	多轴协调动作中	轴组无效	停止中	动作中	减速停止中	错误减速停止中
MC_MoveAbsolute	H	C	C	C	C	H	G	H	I	a	e	e	e	e
MC_MoveRelative	H	C	C	C	C	H	G	H	I	a	e	e	e	e
MC_MoveVelocity	H	D	D	D	D	H	G	H	I	a	e	e	e	e
MC_MoveZeroPosition	H	C	C	C	C	H	G	H	I	a	e	e	e	e
MC_MoveFeed	H	*3	*3	*3	*3	H	G	H	I	a	e	e	e	e
MC_Stop	A	G	G	G	G	G	G	H	I	a	e	e	e	e
MC_ImmediateStop	H	H	H	H	H	H	H	H	I	a	e	e	e	e
MC_SetPosition	A	B	F*4	D*5	H	H	G	H	I	a	e	e	e	e
MC_SetOverride	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_ResetFollowingError	H	B	G	G	G	H	G	H	I	a	e	e	e	e
MC_CamIn	H	E	E	E	E	H	G	H	I	a	e	e	e	e
MC_CamOut	H	H	H	H	C*6	H	G	H	I	a	e	e	e	e
MC_GearIn	H	E	E	E	E	H	G	H	I	a	e	e	e	e
MC_GearInPos	H	E	H	H	H	H	G	H	I	a	e	e	e	e
MC_GearOut	H	H	H	H	C*7	H	G	H	I	a	e	e	e	e
MC_MoveLink	H	E	E	E	E	H	G	H	I	a	e	e	e	e
MC_CombineAxes	H	E	E	E	E	H	G	H	I	a	e	e	e	e
MC_Phasing	H	H	H	H	E*8	H	G	H	I	a	e	e	e	e
MC_TorqueControl	H	D	D	D	D	H	G	H	I	a	e	e	e	e
MC_SetTorqueLimit	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_ZoneSwitch	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_TouchProbe	A	B	C	D	E	H	G	H	I	a	e	e	e	e
MC_AbortTrigger	A	B	C*9	D*5	E	H	G	H	I	a	e	e	e	e
MC_AxesObserve	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_SyncMoveVelocity	H	D	D	D	D	H	G	H	I	a	e	e	e	e
MC_SyncMoveAbsolute	H	C	C	C	C	H	G	H	I	a	e	e	e	e
MC_Reset	A	B	C	D	E	F	G	*10	I	a	b	c	d	e
MC_ChangeAxisUse (未使用轴→使用轴)	A	—	—	—	—	—	—	—	—	—	—	—	—	—
MC_ChangeAxisUse (使用轴→未使用轴)	—	*11	H	H	H	H	H	H	H	a	b	c	d	e
MC_DigitalCamSwitch	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_TimeStampToPos	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_PeriodicSyncVariables	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_SyncOffsetPosition	H	H	H	H	E*8	H	G	H	I	a	e	e	e	e

- *1. MC_Stop 的 Execute 为 FALSE 时切换为 B: 停止中。为 TRUE 时切换为 G: 减速停止中。
- *2. 根据最大减速度, 切换为 b: 停止中或 d: 减速停止中。
- *3. 设定有目标位置时, 切换为 C: 定位动作中。若没有设定, 切换为 D: 连续动作中。
- *4. MC_MoveZeroPosition 执行中会发生异常, 状态切换目标为 H: 错误减速停止中。
- *5. MC_TorqueControl 或 MC_SyncMoveVelocity 执行中会发生异常, 状态切换目标为 H: 错误减速停止中。
- *6. MC_CamIn 未执行中会发生异常, 状态切换目标为 H: 错误减速停止中。
- *7. MC_GearIn 或 MC_GearInPos 未执行中会发生异常, 状态切换目标为 H: 错误减速停止中。
- *8. MC_CombineAxes 执行中会发生异常, 状态切换目标为 H: 错误减速停止中。
- *9. MC_AbortTrigger 的 LatchID 与执行中的 MC_MoveFeed 的 LatchID 重复时会发生异常, 状态切换目标为 H 错误减速停止中。

- *10.在不同的条件下，状态切换目标如下变化。
 通过伺服OFF成功解除异常时、或通过伺服OFF使MC_Stop的Execute变为TRUE时，切换到A：轴无效。
 通过伺服 ON 成功解除异常时，切换到 B：停止中。
 通过伺服 ON 使 MC_Stop 的 Execute 变为 TRUE 时，切换到 G：减速停止中。
 与伺服 ON/OFF 无关，异常解除失败时，切换到 H：错误减速停止中。
- *11.轴变量 `_MC_AX[].Details.VelLimit`（指令速度饱和）变为 TRUE 时，切换到 H：错误减速停止中。FALSE 时会切换为未使用轴，因此变为—：轴状态无。



使用注意事项

NX 系列 CPU 单元时，以 `_MC_AX[*]` 开始的变量名称可能是 `_MC1_AX[*]`、`_MC2_AX[*]`。

● 轴组指令

指令	执行指令前的轴的状态									执行指令前的轴组的状态				
	A	B	C	D	E	F	G	H	I	a	b	c	d	e
	轴无效	停止中	定位动作中	连续动作中	同步动作中	原点复位中	减速停止中	错误减速停止中	多轴协调动作中	轴组无效	停止中	动作中	减速停止中	错误减速停止中
MC_GroupEnable	I	I	C	D	E	F	G	H	I	b	b	c	d	e ^{*1}
MC_GroupDisable	A	B	C	D	E	F	G	H	*2	a	a	a	a	e
MC_MoveLinear	A	B	C	D	E	F	G	H	I	e	c	c	d	e
MC_MoveLinearAbsolute	A	B	C	D	E	F	G	H	I	e	c	c	d	e
MC_MoveLinearRelative	A	B	C	D	E	F	G	H	I	e	c	c	d	e
MC_MoveCircular2D	A	B	C	D	E	F	G	H	I	e	c	c	d	e
MC_GroupStop	A	B	C	D	E	F	G	H	I	e	d	d	d	e
MC_GroupImmediateStop	A	B	C	D	E	F	G	H	I	e	e	e	e	e
MC_GroupSetOverride	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_GroupReadPosition	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_ChangeAxesInGroup	A	B	C	D	E	F	G	H	I	a ^{*3}	e	e	e	e ^{*4}
MC_GroupSyncMoveAbsolute	A	B	C	D	E	F	G	H	I	e	c	c	d	e
MC_GroupReset	A	B	C	D	E	F	G	H	I	a	b	c	d	*5

- *1. 伺服 OFF 时，CommandAborted（执行中断）变为 TRUE。
- *2. 在不同的条件下，状态切换目标如下变化。
 伺服 OFF 时若未发生异常，切换为 A：轴无效。
 通过伺服 OFF 使 MC_GroupStop 的 Execute 变为 FALSE 时，切换到 B：停止中。
 通过伺服 ON 使 MC_GroupStop 的 Execute 变为 TRUE 时，切换到 G：减速停止中。
 与伺服 ON/OFF 无关，发生异常时，切换到 H：错误减速停止中。
- *3. MC_GroupEnable 执行中会发生异常，状态切换目标为 e：错误减速停止中。
- *4. 伺服 ON 时发生异常。
- *5. 在不同的条件下，状态切换目标如下变化。
 通过伺服 OFF 成功解除异常时，切换到 a：不启用轴组。
 通过伺服 ON 成功解除异常时，切换到 b：停止中。
 通过伺服 ON 使 MC_GroupStop 的 Execute 变为 TRUE 时，切换到 d：减速停止中。

● 通用指令

通用指令无论轴的状态、轴组的状态如何，均可随时进行多重启动。

此外，执行通用指令不能将轴的状态及轴组的状态切换为其他状态。保持当前状态不变。

对编码器轴、虚拟编码器轴的指令的多重启动

指令对象为编码器轴及编码器轴组时的多重启动可否及状态切换目标请见下表。

多重启动可否以表格单元的颜色来表示。

白色 : 可多重启动

灰色 : 不可多重启动。发生异常。

黄色 : 不可多重启动。CommandAborted（执行中断）变为 TRUE，指令执行无效。

以下是表内的符号所表示的切换目标的状态。

轴的状态

A : 轴无效（Disabled）

H : 错误减速停止中（ErrorStop）

● 轴指令

指令	执行指令前的轴的状态	
	A	H
	轴无效	错误减速停止中
MC_Power	H	H
MC_MoveJog	H	H
MC_Home	H	H
MC_HomeWithParameter	H	H
MC_Move	H	H
MC_MoveAbsolute	H	H
MC_MoveRelative	H	H
MC_MoveVelocity	H	H
MC_MoveZeroPosition	H	H
MC_MoveFeed	H	H
MC_Stop	H	H
MC_ImmediateStop	H	H
MC_SetPosition	A	H
MC_SetOverride	H	H
MC_ResetFollowingError	H	H
MC_CamIn	H	H
MC_CamOut	H	H
MC_GearIn	H	H
MC_GearInPos	H	H
MC_GearOut	H	H

指令	执行指令前的轴的状态	
	A	H
	轴无效	错误减速停止中
MC_MoveLink	H	H
MC_CombineAxes	H	H
MC_Phasing	H	H
MC_TorqueControl	H	H
MC_SetTorqueLimit	H	H
MC_ZoneSwitch	A	H
MC_TouchProbe	A	H
MC_AbortTrigger	A	H
MC_AxesObserve	A	H
MC_SyncMoveVelocity	H	H
MC_SyncMoveAbsolute	H	H
MC_Reset	A	A
MC_ChangeAxisUse (未使用轴→使用轴)	A	H
MC_ChangeAxisUse (使用轴→未使用轴)	—	H
MC_DigitalCamSwitch	A	H
MC_TimeStampToPos	A	H
MC_PeriodicSyncVariables	A	H
MC_SyncOffsetPosition	H	H

● 轴组指令

编码器轴、虚拟编码器轴不能作为轴组构成轴设定。因此，如果对编码器轴、虚拟编码器轴执行轴组指令的多重启动，全部会发生异常。

● 通用指令

通用指令无论轴的状态如何，均可随时进行多重启动。

此外，执行通用指令不能将轴的状态切换为其他状态。保持当前状态不变。

A-4 版本相关信息

下面记载了因 CPU 单元和 Sysmac Studio 的各版本更新，发生变更的信息。

因版本更新而发生规格变更的指令和新建指令一览

指令中，因 CPU 单元、Sysmac Studio 的版本不同，对应情况和规格可能不同。具体内容如下。

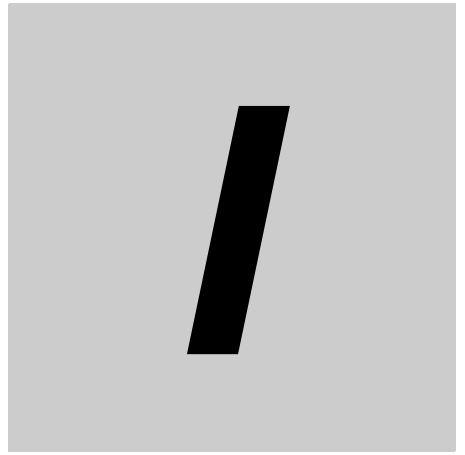
○：该版本新增

△：该版本规格变更

种类	指令	名称	版本		页
			CPU 单元	Sysmac Studio	
共通指令	MC_Write	MC 设定写入	△ Ver.1.01 △ Ver.1.10	△ Ver.1.02 △ Ver.1.12	5-13
	MC_GenerateCam Table	凸轮表生成	○ Ver.1.08	○ Ver.1.09	5-18
	MC_WriteAxis Parameter	轴参数写入	○ Ver.1.08 △ Ver.1.10	○ Ver.1.09 △ Ver.1.12	5-45
	MC_ReadAxis Parameter	轴参数读取			5-58
轴组指令	MC_GroupRead Position	轴组位置获取	○ Ver.1.01	○ Ver.1.02	4-83
	MC_ChangeAxesIn Group	轴组构成轴写入			4-87
	MC_GroupSyncMoveAbsolute	轴组周期性同步绝对位置控制			4-91
轴指令	MC_Power	可运转	△ Ver.1.10	△ Ver.1.12	3-3
	MC_SetPosition	当前位置变更			3-143
	MC_ResetFollowing Error	偏差计数器复位			3-154
	MC_CamIn	凸轮动作开始			3-161
	MC_CombineAxes	加减法定位			3-288
	MC_GearIn	齿轮动作开始	△ Ver.1.02 △ Ver.1.10	△ Ver.1.03 △ Ver.1.12	3-220
	MC_GearInPos	位置指定齿轮动作	△ Ver.1.02 △ Ver.1.10	△ Ver.1.03 △ Ver.1.12	3-240
	MC_MoveLink	梯形模式凸轮	△ Ver.1.10	△ Ver.1.12	3-266
	MC_TouchProbe	外部锁定有效	△ Ver.1.10	△ Ver.1.12	3-329
	MC_HomeWith Parameter	参数指定原点复位	○ Ver.1.03	○ Ver.1.04	3-36
	MC_SyncMove Absolute	周期性同步绝对定位			3-369
	MC_ChangeAxisUse	轴使用变更	○ Ver.1.04	○ Ver.1.05	3-379
	MC_DigitalCamSwitch	数字凸轮开关有效	○ Ver.1.06 △ Ver.1.09	○ Ver.1.07 △ Ver.1.10	3-383
	MC_TimeStampToPos	时间戳→轴位置计算	○ Ver.1.06	○ Ver.1.07	3-402
MC_PeriodicSync Variables	轴变量任务间固定周期同步	○ Ver.1.10	○ Ver.1.13	3-415	
MC_SyncOffsetPosition	周期性同步位置偏置补偿		○ Ver.1.12	3-423	

使用 NX 系列 位置接口单元时

NX 系列位置接口单元可在 CPU 单元版本 1.05 以上和 Sysmac Studio Ver.1.06 以上的组合中使用。
对 NX 系列位置接口单元使用运动控制指令时，存在功能限制的指令和不可使用的指令。详情请参考
□ 《NX 系列 位置接口单元 用户手册（SBCE-CN5-374）》。



索引



数字

2 轴圆弧插补 4-43

C

参数指定原点复位 3-36
 超驰值设定 3-149, 4-79
 齿轮动作 3-240
 齿轮动作解除 3-261
 齿轮动作开始 3-220
 错误代码
 3461Hex A-2, A-23
 5420Hex A-2, A-23
 5421Hex A-2, A-24
 5422Hex A-2, A-24
 5423Hex A-3, A-24
 5424Hex A-3, A-24
 5425Hex A-3, A-25
 5427Hex A-3, A-25
 5428Hex A-3, A-26
 5429Hex A-3, A-26
 542AHex A-3, A-26
 542BHex A-3, A-27
 542CHex A-3, A-27
 542DHex A-3, A-27
 542EHex A-3, A-28
 542FHex A-4, A-28
 5430Hex A-4, A-28
 5431Hex A-4, A-29
 5432Hex A-4, A-29
 5433Hex A-4, A-30
 5434Hex A-4, A-30
 5435Hex A-4, A-30
 5436Hex A-4, A-31
 5437Hex A-4, A-31
 5438Hex A-4, A-31
 5439Hex A-5, A-32
 543AHex A-5, A-32
 543BHex A-5, A-33
 543CHex A-5, A-33
 543DHex A-5, A-34
 543EHex A-5, A-34
 543FHex A-5, A-35
 5440Hex A-6, A-35
 5441Hex A-6, A-36
 5442Hex A-6, A-36
 5443Hex A-6, A-37
 5444Hex A-6, A-37
 5445Hex A-6, A-38
 5446Hex A-6, A-38
 5447Hex A-6, A-38
 5448Hex A-7, A-38
 5449Hex A-7, A-39
 544AHex A-7, A-40
 544CHex A-7, A-40
 544DHex A-7, A-40
 544EHex A-7, A-41
 544FHex A-7, A-41

5450Hex A-7, A-41
 5451Hex A-7, A-42
 5453Hex A-8, A-42
 5454Hex A-8, A-42
 5455Hex A-8, A-42
 5456Hex A-8, A-43
 5457Hex A-8, A-43
 5458Hex A-8, A-44
 5459Hex A-8, A-44
 545AHex A-8, A-45
 545BHex A-8, A-45
 545CHex A-9, A-45
 545DHex A-9, A-46
 545EHex A-9, A-46
 545FHex A-9, A-46
 5460Hex A-9, A-47
 5461Hex A-9, A-47
 5462Hex A-9, A-48
 5463Hex A-9, A-48
 5464Hex A-9, A-49
 5465Hex A-10, A-49
 5466Hex A-10, A-50
 5467Hex A-10, A-50
 5468Hex A-10, A-51
 5469Hex A-10, A-51
 546AHex A-10, A-51
 546BHex A-10, A-52
 546CHex A-10, A-52
 546DHex A-11, A-52
 546EHex A-11, A-53
 546FHex A-11, A-53
 5470Hex A-11, A-54
 5471Hex A-11, A-54
 5472Hex A-11, A-54
 5474Hex A-11, A-55
 5475Hex A-11, A-55
 5476Hex A-11, A-55
 5478Hex A-11, A-56
 5479Hex A-12, A-56
 547AHex A-12, A-57
 547BHex A-12, A-57
 547CHex A-12, A-57
 547DHex A-12, A-58
 547EHex A-12, A-58
 547FHex A-12, A-59
 5480Hex A-12, A-59
 5481Hex A-13, A-59
 5482Hex A-13, A-60
 5483Hex A-13, A-60
 5484Hex A-13, A-60
 5487Hex A-13, A-61
 5488Hex A-13, A-61
 5489Hex A-13, A-61
 548AHex A-13, A-62
 548BHex A-13, A-62
 548CHex A-14, A-62
 548DHex A-14, A-63
 548EHex A-14, A-63
 548FHex A-14, A-63

5490Hex	A-14, A-64
5491Hex	A-14, A-64
5492Hex	A-14, A-64
5493Hex	A-14, A-65
5494Hex	A-14, A-65
5495Hex	A-14, A-65
5496Hex	A-15, A-66
5497Hex	A-15, A-66
5498Hex	A-15, A-66
5499Hex	A-15, A-67
549AHex	A-15, A-67
549BHex	A-15, A-67
549CHex	A-15, A-68
549DHex	A-15, A-68
549EHex	A-15, A-69
5700Hex	A-15, A-69
5702Hex	A-16, A-70
5703Hex	A-16, A-70
5720Hex	A-16, A-71
5721Hex	A-16, A-72
5722Hex	A-19, A-83
5723Hex	A-19, A-83
5724Hex	A-19, A-84
5725Hex	A-19, A-84
5726Hex	A-19, A-84
5727Hex	A-20, A-85
5728Hex	A-20, A-85
5729Hex	A-20, A-85
572AHex	A-20, A-86
572BHex	A-20, A-86
572CHex	A-20, A-86
572DHex	A-20, A-87
572EHex	A-20, A-87
572FHex	A-16, A-72
5730Hex	A-16, A-73
5731Hex	A-21, A-87
573AHex	A-16, A-73
573BHex	A-16, A-74
573CHex	A-17, A-75
573DHex	A-17, A-75
573EHex	A-17, A-76
573FHex	A-17, A-76
5740Hex	A-17, A-76
5741Hex	A-17, A-77
5742Hex	A-17, A-77
5743Hex	A-17, A-78
5749Hex	A-17, A-78
574AHex	A-18, A-79
574BHex	A-18, A-79
6440Hex	A-18, A-80
6441Hex	A-18, A-81
6442Hex	A-18, A-81
6443Hex	A-19, A-82
6444Hex	A-19, A-82
7422Hex	A-19, A-83
错误复位	3-375, 4-97

D

当前位置变更	3-143
点动进给	3-8
电子齿轮	3-240
定位	3-41, 3-47, 3-73, 3-102

G

高速原点复位	3-95
--------	------

J

加减法定位	3-288
绝对值定位	3-47
绝对值直线插补	4-37

K

可运行	3-3
-----	-----

L

立即停止	3-138
------	-------

M

MC_AbortTrigger (外部锁定无效)	3-350
MC_AxesObserve (轴间偏差监视)	3-354
MC_CamIn (凸轮动作开始)	3-161
MC_CamOut (凸轮动作解除)	3-215
MC_ChangeAxesInGroup (轴组构成轴写入)	4-87
MC_ChangeAxisUse (轴使用变更)	3-379
MC_CombineAxes (加减法定位)	3-288
MC_DigitalCamSwitch (数字凸轮开关)	3-383
MC_GearIn (齿轮动作开始)	3-220
MC_GearInPOS (位置指定齿轮动作)	3-240
MC_GearOut (齿轮动作解除)	3-261
MC_GenerateCamTable (凸轮表生成)	5-18
MC_GroupDisable (轴组无效)	4-6
MC_GroupEnable (轴组有效)	4-2
MC_GroupImmediateStop (轴组立即停止)	4-75
MC_GroupReadPosition (轴组位置获取)	4-83
MC_GroupReset (轴组错误复位)	4-97
MC_GroupSetOverride (轴组超驰值设定)	4-79
MC_GroupStop (轴组强制停止)	4-67
MC_GroupSyncMoveAbsolute (轴组周期性同步绝对位置控制)	4-91
_MC_GRP[*].Cfg.ExecID (执行 ID)	2-10
MC_Home (原点复位)	3-15
MC_HomeWithParameter (参数指定原点复位)	3-36
MC_ImmediateStop (立即停止)	3-138
MC_Move (定位)	3-41
MC_MoveAbsolute (绝对值定位)	3-47
MC_MoveCircular2D (2轴圆弧插补)	4-43
MC_MoveFeed (中断固定尺寸定位)	3-102
MC_MoveJog (点动进给)	3-8
MC_MoveLinear (直线插补)	4-11

MC_MoveLinearAbsolute (绝对值直线插补)	4-37
MC_MoveLinearRelative (相对值直线插补)	4-40
MC_MoveLink (梯形模式凸轮)	3-266
MC_MoveRelative (相对值定位)	3-73
MC_MoveVelocity (速度控制)	3-80
MC_MoveZero (高速原点复位)	3-95
MC_PeriodicSyncVariables (轴变量任务间固定周期同步)	3-415
MC_Phasing (主轴相对值相位补偿)	3-299
MC_Power (可运行)	3-3
MC_ReadAxisParameter (轴参数读取)	5-58
MC_Reset (轴错误复位)	3-375
MC_ResetFollowingError (偏差计数器复位)	3-154
MC_SaveCamTable (凸轮表保存)	5-8
MC_SetCamTableProperty (凸轮表属性更新)	5-2
MC_SetOverride (超驰值设定)	3-149
MC_SetPosition (当前位置变更)	3-143
MC_SetTorqueLimit (扭矩限制)	3-316
MC_Stop (强制停止)	3-129
MC_SyncMoveAbsolute (周期性同步绝对定位)	3-369
MC_SyncMoveVelocity (周期性同步速度控制)	3-360
MC_SyncOffsetPosition (周期性同步位置偏置补偿)	3-423
MC 设定写入	5-13
MC_TimeStampToPos (时间戳→轴位置计算)	3-402
MC_TorqueControl (扭矩控制)	3-305
MC_TouchProbe (外部锁定有效)	3-329
MC_Write (MC 设定写入)	5-13
MC_WriteAxisParameter (轴参数写入)	5-45
MC_ZoneSwitch (区域监视)	3-323

N

扭矩控制	3-305
扭矩限制	3-316

P

偏差计数器复位	3-154
---------	-------

Q

强制停止	3-129, 4-67
区域监视	3-323

S

设定写入	5-13
时间戳→轴位置计算	3-402
数字凸轮开关有效	3-383
速度控制	3-80

T

梯形模式凸轮	3-266
凸轮表保存	5-8
凸轮表生成	5-18
凸轮表属性更新	5-2
凸轮动作解除	3-215
凸轮动作开始	3-161

W

外部锁定无效	3-350
外部锁定有效	3-329
位置指定齿轮动作	3-240

X

相对值定位	3-73
相对值直线插补	4-40
相位补偿	3-299

Y

原点复位	3-15
圆弧插补	4-43

Z

直线插补	4-11, 4-37, 4-40
执行 ID	2-10
中断固定尺寸定位	3-102
轴变量任务间固定周期同步	3-415
轴参数读取	5-58
轴参数写入	5-45
轴错误复位	3-375
轴间偏差监视	3-354
周期性同步绝对定位	3-369
周期性同步速度控制	3-360
周期性同步位置偏置补偿	3-423
轴使用变更	3-379
轴组超驰值设定	4-79
轴组错误复位	4-97
轴组构成轴写入	4-87
轴组立即停止	4-75
轴组强制停止	4-67
轴组位置获取	4-83
轴组无效	4-6
轴组有效	4-2
轴组周期性同步绝对位置控制	4-91
主轴相对值相位补偿	3-299

承诺事项

承蒙对欧姆龙株式会社(以下简称“本公司”)产品的一贯厚爱和支持,藉此机会再次深表谢意。

如果未特别约定,无论贵司从何处购买的产品,都将适用本承诺事项中记载的事项。

请在充分了解这些注意事项基础上订购。

1. 定义

本承诺事项中的术语定义如下。

- (1) “本公司产品”:是指“本公司”的FA系统机器、通用控制器、传感器、电子/结构部件。
- (2) “产品目录等”:是指与“本公司产品”有关的欧姆龙综合产品目录、FA系统设备综合产品目录、安全组件综合产品目录、电子/机构部件综合产品目录以及其他产品目录、规格书、使用说明书、操作指南等,包括以电子数据方式提供的资料。
- (3) “使用条件等”:是指在“产品目录等”资料中记载的“本公司产品”的使用条件、额定值、性能、运行环境、操作使用方法、使用时的注意事项、禁止事项以及其他事项。
- (4) “客户用途”:是指客户使用“本公司产品”的方法,包括将“本公司产品”组装或运用到客户生产的部件、电子电路板、机器、设备或系统等产品中。
- (5) “适用性等”:是指在“客户用途”中“本公司产品”的(a)适用性、(b)动作、(c)不侵害第三方知识产权、(d)法规法令的遵守以及(e)满足各种规格标准。

2. 关于记载事项的注意事项

对“产品目录等”中的记载内容,请理解如下要点。

- (1) 额定值及性能值是在单项试验中分别在各条件下获得的值,并不构成对各额定值及性能值的综合条件下获得值的承诺。
- (2) 提供的参考数据仅作为参考,并非可在该范围内一直正常运行的保证。
- (3) 应用示例仅作参考,不构成对“适用性等”的保证。
- (4) 如果因技术改进等原因,“本公司”可能会停止“本公司产品”的生产或变更“本公司产品”的规格。

3. 使用时的注意事项

选用及使用本公司产品时请理解如下要点。

- (1) 除了额定值、性能指标外,使用时还必须遵守“使用条件等”。
- (2) 客户应事先确认“适用性等”,进而再判断是否选用“本公司产品”。“本公司”对“适用性等”不做任何保证。
- (3) 对于“本公司产品”在客户的整个系统中的设计用途,客户应负责事先确认是否已进行了适当配电、安装等事项。
- (4) 使用“本公司产品”时,客户必须采取如下措施:(i)相对额定值及性能指标,必须在留有余量的前提下使用“本公司产品”,并采用冗余设计等安全设计(ii)所采用的安全设计必须确保即使“本公司产品”发生故障时也可将“客户用途”中的危险降到最小程度、(iii)构建随时提示使用者危险的完整安全体系、(iv)针对“本公司产品”及“客户用途”定期实施各项维护保养。
- (5) “本公司产品”是作为应用于一般工业产品的通用产品而设计生产的。如果客户将“本公司产品”用于以下所列用途,则本公司对产品不作任何保证。但“本公司”已表明可用于特殊用途,或已与客户有特殊约定时,另行处理。
 - (a) 必须具备很高安全性的用途(例:核能控制设备、燃烧设备、航空/宇宙设备、铁路设备、升降设备、娱乐设备、医疗设备、安全装置、其他可能危及生命及人身安全的用途)
 - (b) 必须具备很高可靠性的用途(例:燃气、自来水、电力等供应系统、24小时连续运行系统、结算系统、以及其他处理权利、财产的用途等)
 - (c) 具有苛刻条件或严酷环境的用途(例:安装在室外的设备、会受到化学污染的设备、会受到电磁波影响的设备、会受到振动或冲击的设备等)
 - (d) “产品目录等”资料中未记载的条件或环境下的用途
- (6) 除了不适用于上述3.(5)(a)至(d)中记载的用途外,“本产品目录等资料中记载的产品”也不适用于汽车(含二轮车,以下同)。请勿配置到汽车上使用。关于汽车配置用产品,请咨询本公司销售人员。

4. 保修条件

“本公司产品”的保修条件如下。

- (1) 保修期限 自购买之日起1年。(但是,“产品目录等”资料中有明确说明时除外。)
- (2) 保修内容 对于发生故障的“本公司产品”,由“本公司”判断并可选择以下其中之一方式进行保修。
 - (a) 在本公司的维修保养服务点对发生故障的“本公司产品”进行免费修理(但是对于电子、结构部件不提供修理服务。)
 - (b) 对发生故障的“本公司产品”免费提供同等数量的替代品
- (3) 当故障因以下任何一种情形引起时,不属于保修的范围。
 - (a) 将“本公司产品”用于原本设计用途以外的用途
 - (b) 超过“使用条件等”范围的使用
 - (c) 违反本注意事项“3.使用时的注意事项”的使用
 - (d) 非因“本公司”进行的改装、修理导致故障时
 - (e) 非因“本公司”出品的软件导致故障时
 - (f) “本公司”生产时的科学、技术水平无法预见的原因
 - (g) 除上述情形外的其它原因,如“本公司”或“本公司产品”以外的原因(包括天灾等不可抗力)

5. 责任限制

本承诺事项中记载的保修是关于“本公司产品”的全部保证。对于因“本公司产品”而发生的其他损害,“本公司”及“本公司产品”的经销商不负任何责任。

6. 出口管理

客户若将“本公司产品”或技术资料出口或向境外提供时,请遵守中国及各国关于安全保障进出口管理方面的法律、法规。否则,“本公司”有权不予提供“本公司产品”或技术资料。

201611

注:规格如有变更,恕不另行通知。请以最新产品说明书为准。

欧姆龙自动化(中国)有限公司

<http://www.fa.omron.com.cn> 咨询热线:400-820-4535