

机器自动化控制器 NJ/NX系列

指令基准手册 运动篇

NX701-17□□

NX701-16□□

NX1P2-11□□□□

NX1P2-10□□□□

NX1P2-90□□□□

NJ501-□5□□

NJ501-□4□□

NJ501-□3□□

NJ301-12□□

NJ301-11□□

NJ101-10□□

— 预告 —

- 严禁擅自对本手册的部分或全部内容进行影印、复制或转载。
- 因产品改良的关系，本手册记载的产品规格等有时可能会不经预告而变更，恕不事先通知。
- 本手册内容力求尽善尽美，如有不明或错误之处等，烦请联系本公司分部或营业所。届时，请一并告知卷末记载的手册编号。

— 商标 —

- Sysmac为欧姆龙株式会社在日本和其他国家用于欧姆龙工厂自动化产品的商标或注册商标。
- Microsoft、Windows、Windows Vista、Excel、Visual Basic是美国Microsoft Corporation在美国及其它国家的注册商标或商标。
- EtherCAT®是德国Beckhoff Automation GmbH提供许可的注册商标，是获得专利保护的技术。
- ODVA、CIP、CompoNet、DeviceNet、EtherNet/IP是ODVA的商标。
- SD、SDHC标志是SD-3C, LLC的商标。



本手册中记载的其它公司名称、产品名称为各公司的商标或注册商标。

— 著作权 —

屏幕截图的使用已获得微软的许可。

前言

非常感谢您购买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，请同时参阅NJ/NX系列CPU单元的用户手册。

此外，阅读后请将本手册妥善保管于易取处。

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；
- 安装或连接FA设备的人员；
- FA现场管理人员。

此外，编程语言的阅读对象为理解国际标准规格 IEC 61131-3 或国内标准规格 JIS B 3503 的规定内容的人员。

对象产品

本手册以下列产品为对象。

NX系列 CPU单元

- NX701-17□□
- NX701-16□□
- NX1P2-11□□□□
- NX1P2-11□□□□1
- NX1P2-10□□□□
- NX1P2-10□□□□1
- NX1P2-90□□□□
- NX1P2-90□□□□1

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记载在其他手册中。请确认□□“分册构成 (P.2)”及□□“相关手册 (P.25)”。

分册构成

本产品手册按下表分册。请根据目的阅读，充分应用本产品。

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

使用目的	手动													
	基本信息													
	NJ/NX系列 CPU单元 用户手册 硬件篇	NX系列 NX1P2 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	NX系列 NX1P2 CPU单元 用户手册 内置I/O、扩展板功能篇	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 用户手册 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 故障诊断手册
了解NX701的概要	●													
了解NX1P2的概要		●												
了解NJ系列的概要			●											
进行安装、设置、硬件设定														
进行运动控制时						●								
使用EtherCAT时	●	●	●					●						
使用EtherNet/IP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												●		
使用NX1P2的功能时					●									
编写用户程序														
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												●		
进行异常处理时													●	
使用NX1P2的功能时					●									
进行动作确认和调试														
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NX1P2的功能时					●									
了解异常管理功能和故障发生时的处理方法 ^{*1}	△	△	△	△	△		△		△	△	△	△	△	●

使用目的	手动									
	基本信息									
	NJ/NX系列 故障诊断手册	NJ系列 NJ Robotics 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 数据库连接CPU单元 用户手册	NJ/NX系列 CPU单元 内置EtherNet/IP端口用户手册	NJ/NX系列 CPU单元 内置EtherCAT端口 用户手册	NJ/NX系列 指令基准手册 运动篇	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 基本篇	NJ/NX系列 指令基准手册 基本篇
了解维护作业										
进行运动控制时						●				
使用EtherCAT时					●					
使用EtherNet/IP时					●					

*1. 关于异常管理的思路和异常项目的概要，请参阅 □ “NJ/NX系列 故障诊断手册(SBCA-361)”。关于异常详情，请根据异常内容，参阅标有△标志的手册。

手册的阅读方法

页面构成

本手册各页面的构成及符号如下所示。

章标题 → 3 轴指令

节标题 → MC_Power

项标题 → 变量

手册名称

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。

备注、补充、参考页 备注、补充、参考页等各项目通过图标表示。

(注) 本页为用于说明的范例页。与实际内容有所差异。

图标

本资料中使用的图标，含义如下。



安全要点

表示为了产品的安全使用而应当实施或避免的事项。



使用注意事项

表示为了预防产品无法动作、误动作，或者对产品性能、功能产生不良影响而应当实施或避免的事项。



参考

项目内容请根据需要阅读。
对应当了解的信息及使用时可作为参考的相关内容进行说明。



版本相关信息

对CPU单元、Sysmac Studio不同版本的不同性能和功能进行说明。



表示详细信息、相关信息的所在页。

目录构成

		1
1	运动控制指令概要	2
2	变量和指令	3
3	轴指令	4
4	轴组指令	5
5	通用指令	A
A	附录	I
I	索引	

目录

前言	1
分册构成	2
手册的阅读方法	4
目录构成	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
法规与标准	19
版本	21
相关手册	25
手册修订履历	28

第1章 运动控制指令概要

1-1 运动控制指令	1-2
PLCopen® 运动控制用功能块	1-2
运动控制指令概要	1-3
同步控制的主轴及辅轴的注意事项	1-6
1-2 运动控制指令的基本知识	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 变量一览表	2-2
MC 通用变量	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
通用指令	2-26
轴指令	2-27
轴组指令	2-29
2-3 PDO映射	2-30
必需对象	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	3-37
变量	3-37
功能说明	3-40
MC_Move	3-42
变量	3-42
功能说明	3-44
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	4-25
MC_MoveLinearAbsolute	4-38
变量	4-38
功能说明	4-40
MC_MoveLinearRelative	4-41
变量	4-41
功能说明	4-43
MC_MoveCircular2D	4-44
变量	4-44
功能说明	4-47
示例程序	4-56
MC_GroupStop	4-68
变量	4-68
功能说明	4-70
MC_GroupImmediateStop	4-76
变量	4-76
功能说明	4-77
MC_GroupSetOverride	4-80
变量	4-80
功能说明	4-81
MC_GroupReadPosition	4-84
变量	4-84
功能说明	4-86
MC_ChangeAxesInGroup	4-88
变量	4-88

功能说明	4-90
MC_GroupSyncMoveAbsolute	4-92
变量	4-92
功能说明	4-94
MC_GroupReset	4-98
变量	4-98
功能说明	4-99

第5章 通用指令

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	5-57
变量	5-57
功能说明	5-59

附录

A-1 错误代码一览	A-2
A-2 错误代码详情	A-17
A-2-1 控制器异常说明的解释	A-17
A-2-2 异常说明	A-18
A-3 多重启动可否	A-83
A-3-1 轴的状态、轴组的状态	A-84
A-3-2 状态切换和多重启动的可否	A-86
A-4 版本相关信息	A-92

索引

承诺事项

关于“本公司产品”，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

本承诺事项中用语的定义如下所示。

- “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“本公司产品”的生产或变更“本公司产品”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”。
- “本公司产品”是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。但经“本公司”许可后用于以下用途或与“本公司”签订特殊协议的情况除外。
 - (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、会受振动和冲击影响的设备等)
 - (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

“本公司产品”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“本公司产品”原本用途以外的用途
 - (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

安全注意事项请参阅下列手册。

- □ “NX系列 CPU单元 用户手册 硬件篇(SBCA-418)”
- □ “NX系列 NX1P2 CPU单元 用户手册 硬件篇(SBCA-448)”
- □ “NJ系列 CPU单元 用户手册 硬件篇(SBCA-358)”




安全要点

安全要点请参阅下列手册。

-  “NX系列 CPU单元 用户手册 硬件篇(SBCA-418)”
-  “NX系列 NX1P2 CPU单元 用户手册 硬件篇(SBCA-448)”
-  “NJ系列 CPU单元 用户手册 硬件篇(SBCA-358)”

使用注意事项

使用注意事项请参阅下列手册。

-  “NX系列 CPU单元 用户手册 硬件篇(SBCA-418)”
-  “NX系列 NX1P2 CPU单元 用户手册 硬件篇(SBCA-448)”
-  “NJ系列 CPU单元 用户手册 硬件篇(SBCA-358)”

法规与标准

日本国外的使用

对本产品，根据外汇和外国贸易管理法的规定，出口(或提供给非本土企业)需获得出口许可、批准的货物(或技术)时，需依照上述法规获得出口许可、批准(或劳务贸易许可)。

符合EU指令

符合指令

- EMC指令
- 低电压指令

适用途径

● EMC指令

欧姆龙的产品为装入各种机械、制造装置使用的电气设备，为使装入的机械、装置更容易符合EMC标准，产品自身需符合相关EMC标准*。

但客户的机械、装置多种多样，且EMC的性能因装入符合EU指令产品的机械、控制柜的构成、布线状态、配置状态等而异，因此无法确认客户使用状态下的适用性。因此，请客户自行确认机械和装置整体最终的EMC适用性。

* EMC (Electro-Magnetic Compatibility: 电磁环境兼容性) 相关标准中，
与EMS (Electro-Magnetic Susceptibility: 电磁敏感性) 相关的为EN61131-2、
与EMI (Electro-Magnetic Interference: 电磁干扰)相关的为EN61131-2。
此外，Radiated emission依照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。要使客户的机械、装置符合EU指令，需注意以下事项。

- NJ/NX系列请务必安装在控制柜内。
- 与DC电源单元及I/O单元连接的DC电源请使用SELV 规格的电源。
- NJ/NX系列的EU指令符合产品符合EMI相关的通用排放标准，但关于Radiated emission(10m法)，会因使用的控制柜构成、与连接的其它设备间的关系、接线等而异。
因此，使用符合EU指令的NJ/NX系列时，也需客户自行根据机械、装置整体确认是否符合EU指令。

符合KC标准

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)

本设备属于商用电磁波发生设备(ClassA)，旨在用于家庭以外的场所。销售方和用户请注意这一点。

符合船级标准

本产品符合各种船级标准。为符合各船级标准，需设定使用条件，部分安装场所无法使用，因此使用时请务必向本公司营业部咨询。

各船级标准的使用条件(NK、LR)

- 本产品请务必安装在控制柜中。
- 控制柜的开关口等处的间隙请使用衬垫等完全遮蔽。
- 电源线上请连接下列噪声滤波器。

噪声滤波器

厂家	型号
COSEL公司	TAH-06-683

软件许可证与著作权

本产品已安装第三方软件。该软件的相关许可证和著作权请浏览http://www.fa.omron.co.jp/nj_info_j/。

版本

NJ/NX 系列各单元及各 EtherCAT 从站的硬件和软件是通过硬件版本、单元版本等不同编号来进行版本管理。硬件或软件每次变更规格，都将更新硬件版本或单元版本。因此，即使是同一型号的单元和 EtherCAT 从站，如果硬件版本或单元版本不同，配备的功能和性能就会存在差异。

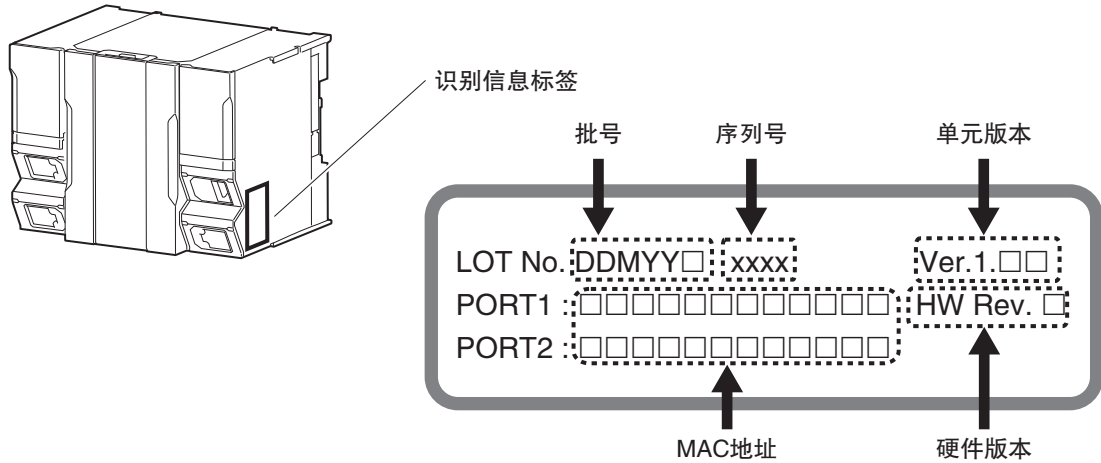
版本确认方法

版本可通过识别信息标签或 Sysmac Studio 进行确认。

通过识别信息标签确认

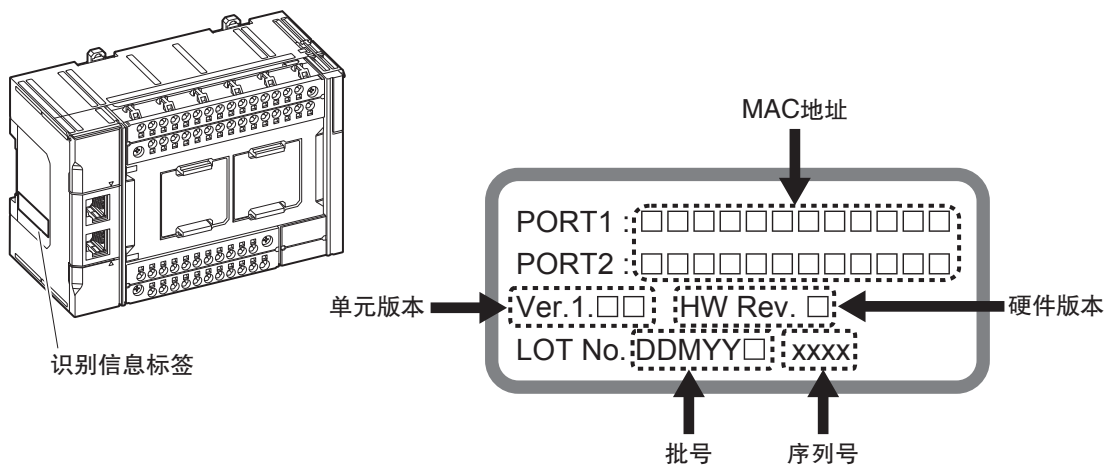
版本可通过产品侧面的识别信息标签进行确认。

NX 系列 CPU 单元 NX701-□□□□ 的识别信息标签如下图所示。



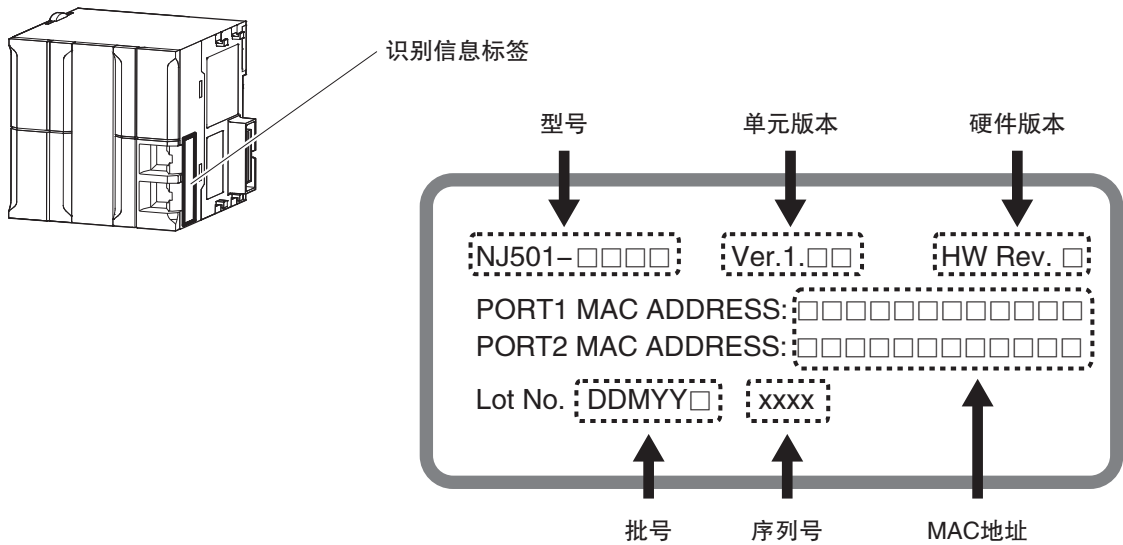
(注) 硬件版本为“无”的单元不显示硬件版本。

NX 系列 CPU 单元 NX1P2-□□□□□□ 的识别信息标签如下图所示。



(注) 硬件版本为“无”的单元不显示硬件版本。

NJ系列CPU单元NJ501-□□□□的识别信息标签如下图所示。



(注) 硬件版本为“无”的单元不显示硬件版本。

基于Sysmac Studio的确认方法

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下的[生产信息]确认。可确认版本的单元为CPU单元、CPU机架上的NX单元及扩展板。

- 1 在多视图浏览器中右击[构成·设定]-[CPU·扩展机架]-[CPU机架]，选择[显示生产信息]。显示[生产信息]对话框。

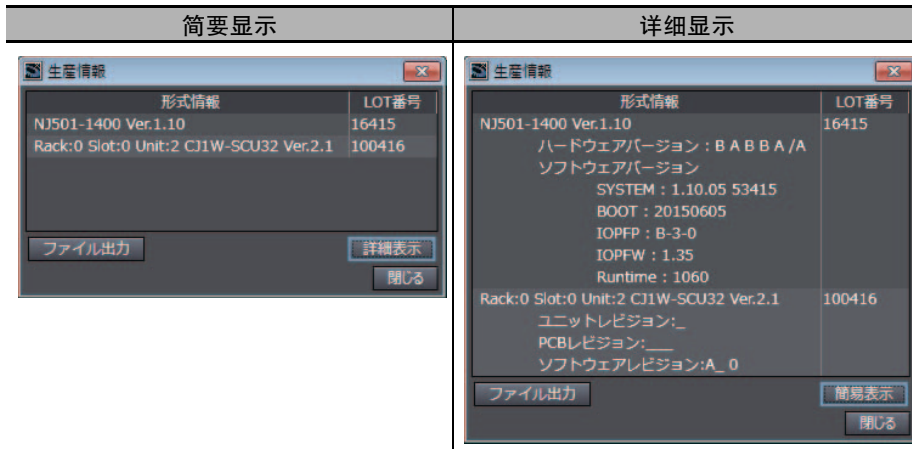
● NJ系列CPU单元的版本确认方法

单元版本可通过在线状态下的[生产信息]确认。但可确认版本的单元只有CPU单元、CJ系列的高功能I/O单元及CJ系列的CPU高功能单元。CJ单元的基本I/O单元的版本无法通过Sysmac Studio进行确认。

- 1 在多视图浏览器中双击[构成·设定]-[CPU·扩展机架]。或者右击[构成·设定]-[CPU·扩展机架]，选择[编辑]。显示单元编辑器。
- 2 右击单元编辑器的空白处，选择[显示生产信息]。显示[生产信息]对话框。

● 生产信息显示内容的切换

- 1 选择[生产信息]对话框右下方的[简要显示]或[详细显示]。
[生产信息]的简要显示和详细显示将会切换。



简要显示和详细显示的显示内容不同。详细显示会显示单元版本、硬件版本及软件版本。简要显示只
显示单元版本。

(注) 硬件版本在硬件版本的右端以 “/” 隔开显示。硬件版本为 “无” 的单元不显示硬件版本。

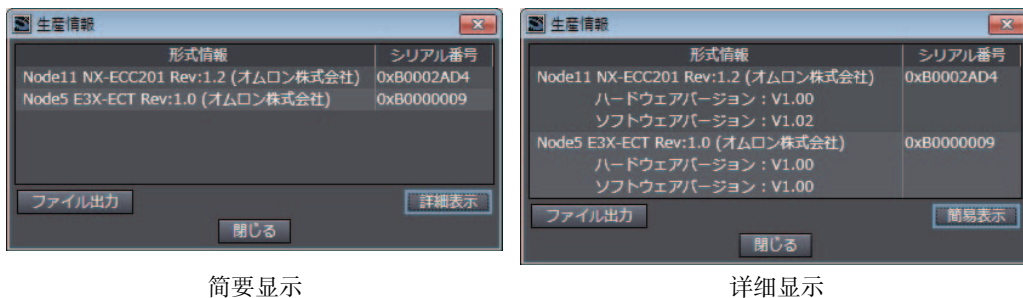
● EtherCAT从站版本确认方法

EtherCAT从站版本可通过在线状态下的[生产信息]确认。确认方法如下所示。

- 1 双击多视图浏览器内的[构成・设定]-[EtherCAT]。或者右击[构成・设定]-[EtherCAT]，
选择[编辑]。
显示EtherCAT构成编辑画面。
- 2 在EtherCAT构成的编辑画面中右击主机，选择[显示生产信息]。
显示生产信息对话框。
显示的单元版本附带 “Rev” 字样。

● 生产信息显示内容的切换

选择[生产信息]对话框右下方的[简要显示]或[详细显示]。
[生产信息]的简要显示和详细显示将会切换。



CPU单元的单元版本和Sysmac Studio版本

配备的功能因 NJ/NX 系列 CPU 单元的单元版本而异。使用版本升级后的新增功能时，需使用对应版本的 Sysmac Studio。

CPU单元的单元版本的种类与Sysmac Studio版本之间的关系，以及单元版本支持的功能一览请参阅

☐ “NJ/NX系列 CPU单元 用户手册 软件篇(SBCA-359)”。

相关手册

与本手册相关的手册如下表所述。请同时参阅。

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-418	NX701-□□□□	希望了解NX701 CPU单元的概要/设计/安装/保养等基本规格时。与硬件相关的信息为主。	对NX701的系统整体概要和CPU单元进行以下内容的说明。 <ul style="list-style-type: none"> • 特长和系统构成 • 概要 • 各部分的名称和功能 • 一般规格 • 设置和接线 • 维护检查
NX系列 NX1P2 CPU单元 用户手册 硬件篇	SBCA-448	NX1P2-□□□□	希望了解NX1P2 CPU单元的概要/设计/安装/保养等基本规格时。与硬件相关的信息为主。	对NX1P2的系统整体概要和CPU单元进行以下内容的说明。 <ul style="list-style-type: none"> • 特长和系统构成 • 概要 • 各部分的名称和功能 • 一般规格 • 安装与接线 • 维护检查
NJ系列 CPU单元 用户手册 硬件篇	SBCA-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。与硬件相关的信息为主。	对NJ系列的系统整体概要和CPU单元进行以下内容的说明。 <ul style="list-style-type: none"> • 特长和系统构成 • 概要 • 各部分的名称和功能 • 一般规格 • 设置和接线 • 维护检查
NJ/NX系列 CPU单元 用户手册 软件篇	SBCA-359	NX701-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	希望了解NJ/NX系列CPU单元的编程/系统调试时。与软件相关的信息为主。	对NJ/NX系列的CPU单元进行以下内容的说明。 <ul style="list-style-type: none"> • CPU单元的动作 • CPU单元的功能 • 初始设定 • 符合IEC 61131-3标准的语言规格和编程
NX系列 NX1P2 CPU单元 用户手册 内置I/O、扩展板 功能篇	SBCA-449	NX1P2-□□□□	希望了解NX系列NX1P2 CPU单元独有功能的详情和NJ/NX系列功能的概要时。	对NX1P2 CPU单元功能中的以下内容进行说明。 <ul style="list-style-type: none"> • 内置I/O • 串行通信扩展板 • 对模拟输入输出用扩展板以及NJ/NX系列CPU单元的以下功能概要进行说明。 • 运动控制功能 • EtherNet/IP通信功能 • EtherCAT通信功能
NJ/NX系列 指令基准手册 基本篇	SBCA-360	NX701-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	希望了解NJ/NX系列的基本指令规格的详情时。	对各指令(IEC 61131-3标准)的详情进行说明。
NJ/NX系列 CPU单元 用户手册 运动控制篇	SBCE-363	NX701-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	希望了解运动控制的设定和编程思路时。	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。
NJ/NX系列 指令基准手册 运动篇	SBCE-364	NX701-□□□□ NX1P2-□□□□ 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。
NJ/NX系列 CPU单元 内置EtherCAT®端口 用户手册	SBCD-358	NX701-□□□□ NX1P2-□□□□ 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、安装进行描述。

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-359	NX701-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	使用NJ/NX系列CPU单元的内置EtherNet/IP端口时。	对内置EtherNet/IP端口进行说明。 对基本设定、标签数据链接及其他功能进行描述。
NJ系列 数据库连接CPU单元 用户手册	SBCA-411	NJ501-1□20 NJ101-□□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 CPU单元 用户手册	SBCA-412	NJ501-1340	在NJ系列中使用GEM服务功能时。	对GEM服务功能进行说明。
NJ系列 NJ Robotics CPU单元 用户手册	SBCA-421	NJ501-4□□□	在NJ系列中进行机器人控制时。	对机器人控制功能进行说明。
NJ/NX系列 故障诊断 手动	SBCA-361	NX701-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	希望了解通过NJ/NX系列检测异常的详情时。	对通过NJ/NX系列系统检测的异常管理的途径和各异常项目进行说明。
Sysmac Studio Version 1 操作手册	SBCA-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-361	NX-ECC□□□□	希望了解NX系列EtherCAT耦合器单元和EtherCAT从站终端的使用方法时。	对由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和构成方法，以及经由EtherCAT对NX单元进行设定、控制、监控的EtherCAT耦合器单元的硬件、设定方法及功能进行说明。
NX系列 数据基准手册	SBCA-410	NX-□□□□□□	希望通过一览表查看NX系列各单元的系统构成所需的数据时。	汇总了NX系列各单元的消耗功率、重量等系统构建所需的数据。
NX系列 NX单元 用户手册	SBCA-407	NX-ID□□□□ NX-IA□□□□ NX-OC□□□□ NX-OD□□□□ NX-MD□□□□	希望了解NX单元的使用方法时。	对NX单元的硬件、设定方法及功能进行说明。 备有以下单元的手册。 数字I/O单元、模拟I/O单元、系统单元、位置接口单元、通信接口单元、负载传感器输入单元、IO-Link主站单元。
	SBCA-408	NX-AD□□□□ NX-DA□□□□		
	SBCA-440	NX-TS□□□□ NX-HB□□□□		
	SBCA-409	NX-PD1□□□□ NX-PF0□□□□ NX-PC0□□□□ NX-TBX01		
	SBCE-374	NX-EC0□□□□ NX-ECS□□□□ NX-PG0□□□□		
	SBCA-422	NX-CIF□□□□		
	SBCA-439	NX-RS□□□□		
	SBCD-370	NX-ILM□□□□		
EtherCAT远程I/O终端 GX系列 EtherCAT从站 用户手册	SBCD-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、设定方法及功能进行说明。
AC伺服电机/驱动器 1S系列 (EtherCAT®通信内置型) 用户手册	SBCE-377	R88M-1□ R88D-1SN□-ECT	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/驱动器的硬件和设定方法、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-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/驱动器的硬件和设定方法、功能进行说明。 G5系列有直线电机型和位置控制限定型。
	SBCE-365	R88M-K□ R88D-KN□-ECT		
	SBCE-366	R88L-EC-□ R88D-KN□ -ECT-L		

手册修订履历

手册的修订记号附加在封面和封底左下角的手册编号的末尾，以字母表示。

例



修订记号

修订记号	修订日	修订内容和修订页
A	2011年7月	初版
B	2012年1月	添加以下轴组指令 <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) 错误修正
C	2012年5月	CPU单元升级为Ver.1.02所伴随的修订 错误修正
D	2012年8月	CPU单元升级为Ver.1.03所伴随的修订
E	2013年2月	CPU单元升级为Ver.1.04所伴随的修订
F	2013年4月	CPU单元升级为Ver.1.05所伴随的修订 错误修正
G	2013年6月	CPU单元升级为Ver.1.06所伴随的修订 错误修正
H	2013年12月	CPU单元升级为Ver.1.08所伴随的修订 错误修正
J	2014年7月	CPU单元升级为Ver.1.09所伴随的修订 错误修正
K	2015年1月	CPU单元升级为Ver.1.10所伴随的修订 错误修正
L	2015年4月	追加NX系列CPU单元NX701-□□□□、NJ系列CPU单元NJ101-□□□□所伴随的修订 错误修正
M	2016年4月	CPU单元升级为Ver.1.11所伴随的修订 错误修正
N	2016年7月	CPU单元升级为Ver.1.12 所伴随的修订 错误修正
P	2016年10月	追加NX 系列 NX1P2 CPU单元所伴随的修订 CPU单元升级为Ver.1.13 所伴随的修订 错误修正

运动控制指令概要

本章对NJ/NX系列CPU单元内置的运动控制指令进行简要说明。

1-1 运动控制指令	1-2
PLCopen®运动控制用功能块	1-2
运动控制指令概要	1-3
同步控制的主轴及辅轴的注意事项	1-6
1-2 运动控制指令的基本知识	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

要从NJ/NX系列的用户程序执行运动控制功能，需要使用作为功能块定义的运动控制指令。

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

运动控制指令有PLCopen[®]定义的指令和MC功能模块独有的指令两种。

下面对PLCopen[®]运动控制用功能块和运动控制指令的概要进行说明。

运动控制指令的详情请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”。

Ver.1.05以上的CPU单元和Ver.1.06以上的Sysmac Studio组合时，可使用NX系列位置接口单元。

使用NX系列位置接口单元时，请参阅 □ “NX系列 位置接口单元 用户手册(SBCE-374)”。

PLCopen[®]运动控制用功能块

PLCopen[®]对运动控制用功能块进行了标准化，制定了IEC 61131-3(JIS B 3503)规格语言程序接口的定义。

除了单轴定位、电子凸轮功能、多轴插补控制等功能，还对启动指令的基本步骤进行了定义。

利用PLCopen[®]运动控制用功能块，可以减小对硬件的依赖，提高用户程序的再利用率。

还能降低培训、支持等的成本。



参考

何谓PLCopen[®]

PLCopen[®]是总部在欧洲的IEC 61131-3推广团体，是一个全球性的会员组织。

IEC 61131-3是PLC编程的国际标准规格。

PLCopen[®]Japan是日本市场的推广委员会，由关注日本市场的会员构成。

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

运动控制指令概要

下面对PLCopen[®]运动控制用功能块技术规格所定义的项目和MC功能模块中的概要进行说明。

运动控制指令的种类

下表所示为运动控制指令的种类和概要。

种类	种类	功能项目	概要
通用指令	通用的管理类指令	凸轮表	控制MC功能模块的通用状态、操作或监视各种数据的指令。
		参数	
轴指令	单轴的动作类指令	单轴位置控制	使轴执行单轴动作的指令。
		单轴速度控制	
		单轴转矩控制	
		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、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	使轴组执行协调动作的指令。
	多轴的管理类指令	多轴协调控制辅助	控制、监视轴组状态的指令。

状态变化

对轴的状态、轴组的状态及指令启动引发的状态变化进行定义。

关于MC功能模块的状态和状态变化，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-363)”。

运动控制指令的启动和状态

将启动指令的变量及表示指令执行状态的变量作为指令的通用规则进行定义。

启动指令的输入变量有Execute和Enable两类。

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

关于MC功能模块的详细规格，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”。



使用注意事项

手册中记载的时序图可能与Sysmac Studio的数据跟踪中显示的时序不同。

关于数据跟踪的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 软件篇(SBCA-359)”。

异常处理

MC功能模块的运动控制通过运动控制指令执行。执行运动控制指令时，检查输入参数的错误及指令处理导致的异常。

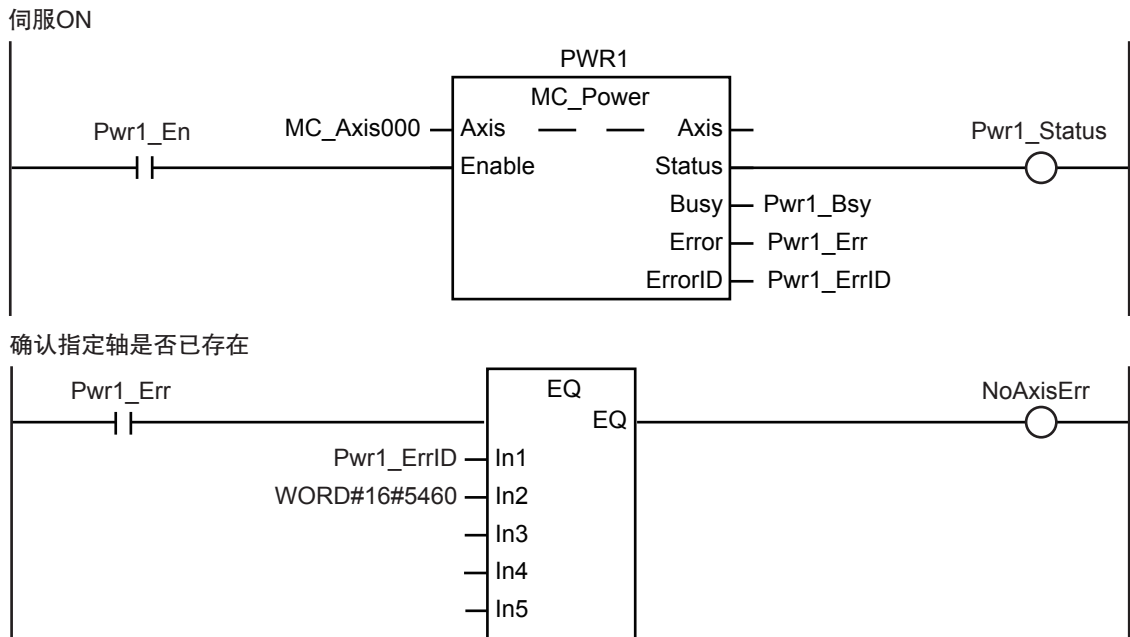
指令有异常时，指令的输出变量Error(错误)变为TRUE，并将异常代码输出到ErrorID(错误代码)。

运动控制指令可通过2种方法编写异常处理程序。

● 按指令进行异常处理

可利用指令的输出变量Error(错误)或ErrorID(错误代码),对每个发生异常的指令进行异常处理。

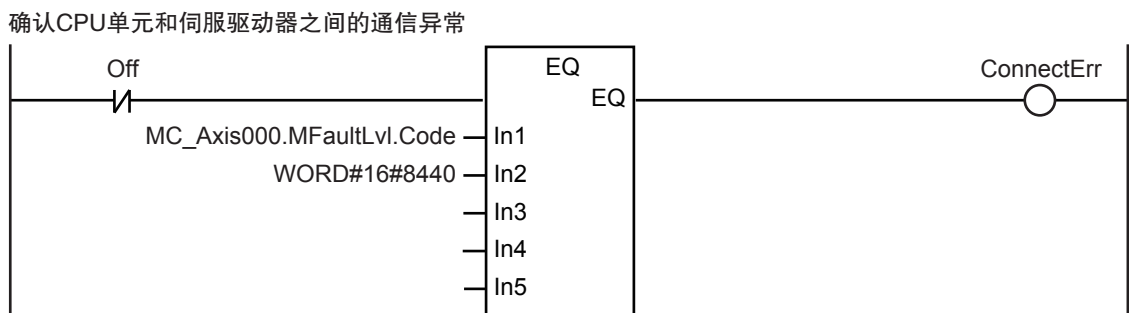
以下是判定实例名称为“PWR1”的指令发生“轴指定错误”的例子。程序编制为“NoAxisErr”变为TRUE时执行异常处理。



● 按种类进行异常处理

可利用运动控制系统变量的异常状态，对每一类异常进行异常处理。

以下是判定“MC_Axis000”的轴发生“从站通信异常”的例子。程序编制为“ConnectErr”变为TRUE时执行异常处理。



执行运动控制指令时输入变量的变更(指令重启)

在执行运动控制指令时变更相同指令实例的输入变量值，再次将Execute变为TRUE，即以变更后的值进行动作。

关于MC功能模块的指令重启，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”。

通过选择缓存模式执行指令多重启动

可以在动作中执行其他指令实例。

可通过设定名为缓存模式选择(BufferMode)的输入来指定动作开始的时间。

缓存模式选择(BufferMode)中有以下模式可选。

- 中断 : 中断(Aborting)
- 等待 : 等待(Buffered)
- 以低速合并 : 低速合并(BlendingLow)
- 以前一个速度合并 : 以前一个速度合并(BlendingPrevious)
- 以后一个速度合并 : 以后一个速度合并(BlendingNext)
- 以高速合并 : 以高速合并(BlendingHigh)

若选择中断模式，则中断其他动作立即执行新指令。

选择其他缓存模式时，会一直等待，直到执行中的指令的输出变量Done及InVelocity变为TRUE。

在等待模式下，执行中的指令动作正常结束(Done = TRUE)后再执行下一个指令。

在合并模式下，不停止且连续执行两个指令的动作。连续执行动作的速度可从四种缓存模式中选择。

MC功能模块中通过缓存模式选择(BufferMode)执行的指令多重启动只表述为指令多重启动进行说明。

关于MC功能模块的指令多重启动，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”。

MC功能模块的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。相关详情请参阅 □ “A-3 多重启动可否(P.A-83)”。

通过运动控制使用的结构体

PLCopen[®]技术资料对运动控制所需的信息作为结构体进行定义。对数据类型的名称及概要作出了定义，但没有对结构体的内容作出定义。

PLCopen[®]中的主要数据类型和MC功能模块使用的数据类型如下所示。

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。 • 触发指定 • 检测模式信息(正、负、两者、边缘、电平、模式识别等)
INPUT_REF	-	包含输入规格相关信息的结构体。有时为虚拟数据。 MC功能模块中未使用。
OUTPUT_REF	_sOUTPUT_REF	与物理性输出相关的结构体。 MC功能模块中，Ver.1.06以上的CPU单元和Ver.1.07以上的Sysmac Studio组合时使用。

如上表所示，MC功能模块使用一部分PLCopen[®]定义的数据类型和MC功能模块自定义的数据类型。

关于MC功能模块使用的数据类型和结构体的定义，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”。

同步控制的主轴及辅轴的注意事项

以下是同步控制的主轴及辅轴在速度发生急剧变化时的注意事项和发生异常的条件。

速度急剧变化时

同步动作开始或动作过程中主轴及辅轴的速度发生急剧变化时，从轴的动作会急剧变化，可能对装置施加过度的力。

在以下条件下，主轴及辅轴的速度有可能发生急剧变化，请予注意。

- 主轴及辅轴执行以下4种指令时
 - MC_ImmediateStop(立即停止)指令
 - MC_SetPosition(变更当前位置)指令
 - MC_ResetFollowingError(偏差计数器复位)指令
 - MC_SyncMoveVelocity(周期同步速度控制)指令
 请正确设置上述指令的输入参数及启动时间，或者在解除同步控制后再启动上述指令，以防止从轴急剧动作。
- 主轴及辅轴的立即停止输入信号、极限停止输入信号变为TRUE时
- 主轴及辅轴从伺服ON状态变为伺服OFF状态时
 - 主轴及辅轴为垂直轴等时，在变为伺服OFF的状态时速度可能会发生急剧变化。
 - 请采取对主轴或辅轴设置制动器、或者在同步控制结束后再变为伺服OFF状态等对策，防止从轴急剧动作。

- 切换伺服驱动器的控制模式时
 请注意指令启动时的速度变化。
 请恰当设置指令的输入参数。



版本相关信息

Ver.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轴的指令当前位置，从轴的动作也不会急剧变化。

发生异常的条件

有NJ/NX系列CPU单元通用的条件和NX701CPU单元独有的条件。

● NJ/NX系列CPU单元通用的条件

开始同步动作时或同步动作过程中，主轴及辅轴发生以下四个条件时，从轴会发生“主轴位置读取异常”或“辅轴位置读取异常”。

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

- EtherCAT的过程数据通信未建立时
- 在“EtherCAT的通信未建立”的状态下发生“从站通信异常”时
- 发生“不能计算绝对值编码器当前位置”的异常时
- 从站脱离时

对从轴执行了同步控制指令多重启动时,如下所示。

- 多重启动时，即使主轴及辅轴发生上述4个条件中的任意一个，也将正常受理多重启动，进入缓冲状态。
- 执行多重启动、缓冲状态的指令开始动作时，只要不发生上述4个条件中的任意一个，便可开始动作。



参考

对主轴及辅轴启动MC_Home或MC_HomeWithParameter、或者使用绝对值编码器的轴执行MC_Power时，从轴会忽视主轴及辅轴的位置变化。
 因此，由于原点确定时的处理，从轴不会急剧动作。

● NX701CPU单元的条件

将主轴及辅轴的任务和从轴的任务分配至不同任务时，从轴将会发生“主轴指定错误(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 运动控制指令的基本知识

下面对使用CPU单元内置的MC功能模块的运动控制指令编写程序的基本规格及限制事项进行说明。

关于运动控制指令的详情，请参阅 □ “第3章 轴指令”、□ “第4章 轴组指令”、□ “第5章 通用指令”。

运动控制指令的名称

MC功能模块的所有运动控制指令开头均带有“MC”字样。

要判断指令为PLCopen®定义的指令还是MC功能模块独有的指令，请参阅 □ “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以下述编程语言使用。

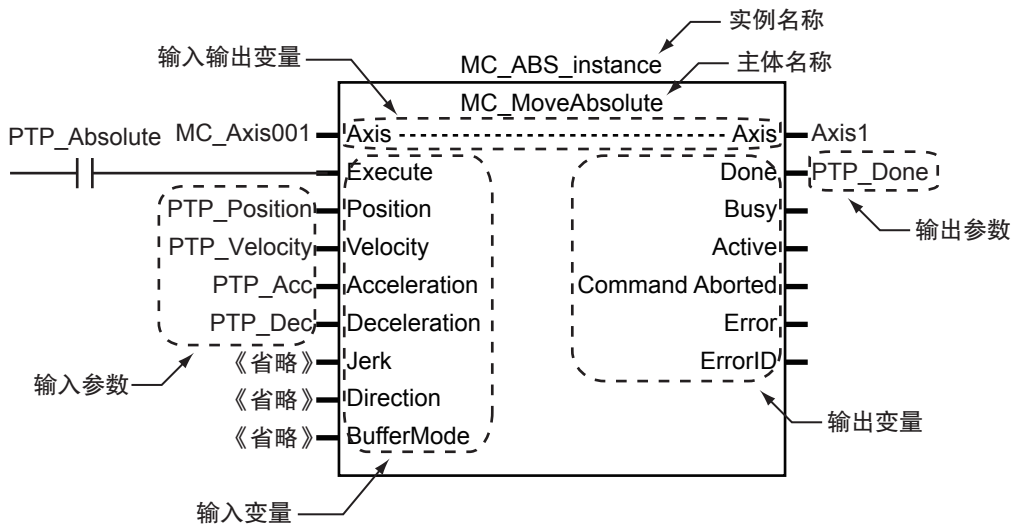
- 梯形图(LD)
- 结构文本(ST)

梯形图(LD)

将运动控制指令的指令实例配置在梯形图上。

可以对每个指令实例设置实例名称。

以下是MC_MoveAbsolute(绝对值定位)指令的记述范例。



- 向输入输出变量Axis指定要控制的伺服驱动器等的轴变量名称。
- 向各输入变量指定目标位置及目标速度等的动作条件。
- 向各输出变量输出指令的状态及伺服驱动器的状态。
- 省略各输入参数时，变为各输入变量的初始值。

结构文本(ST)

指定指令实例名称。

指令变量按照从左上到左下、从右上到右下的顺序记述。

以下是MC_MoveAbsolute(绝对值定位)指令的记述范例。

```
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

以下对运动控制指令可以配置在哪个任务中、以及配置在程序的哪个位置会出现什么样的动作进行说明。

任务种类

运动控制指令可以在原始恒定周期任务、固定周期任务(执行优先度5)及固定周期任务(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，实施编连时会发生异常。

任务种类	可否使用	备注
原始恒定周期任务	可以	<ul style="list-style-type: none"> 不指定轴/轴组的通用指令 针对分配至原始恒定周期任务的轴/轴组的指令
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/轴组的通用指令 针对分配至固定周期任务(执行优先度5)的轴/轴组的指令
固定周期任务(执行优先度16)	可以	<ul style="list-style-type: none"> 不指定轴/轴组的通用指令 针对分配至原始恒定周期任务的轴/轴组的指令
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NX701 CPU单元可使用。

*2 无法在NX1P2 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内使用。

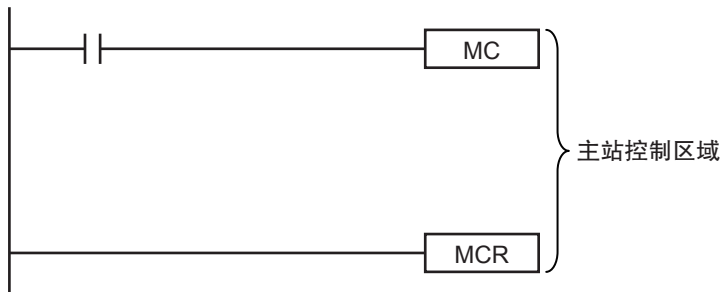


参考

将由多个动作组合而成的工序创建成一个功能块，可以通过程序结构化提高设计效率，改善程序的识别性。

主站控制区域

在梯形图程序中，从MC指令(主站控制开始指令)到MCR指令(主站控制结束指令)的区域就是主站控制区域。

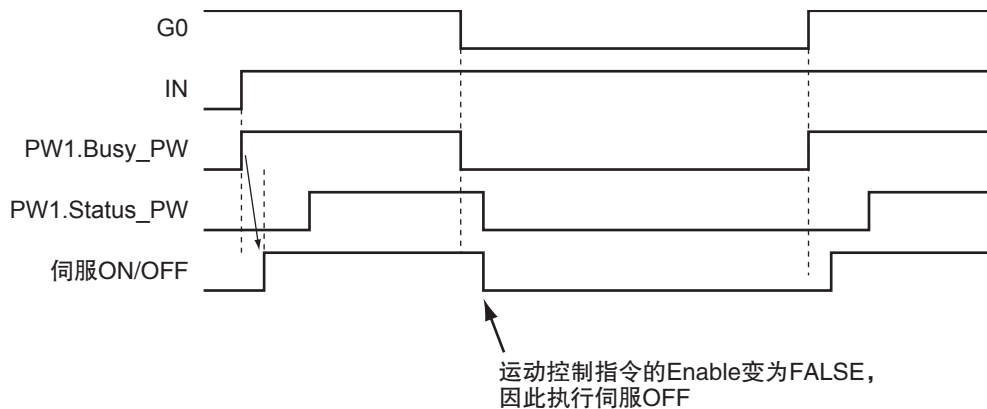
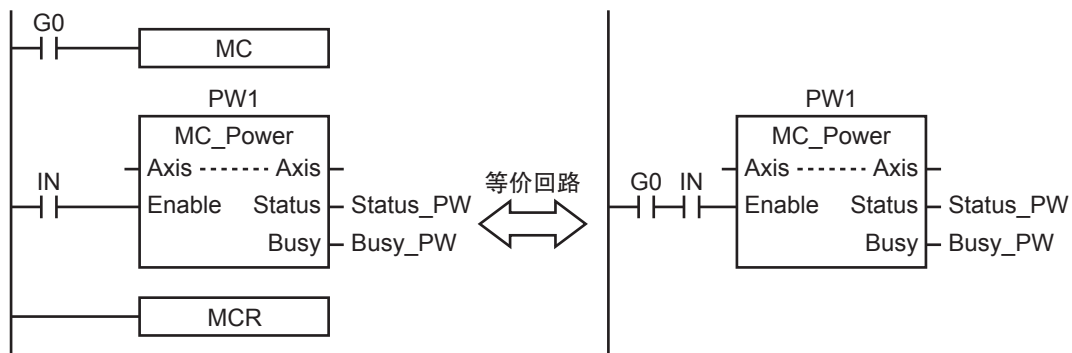


运动控制指令配置到主站控制区域时，当MC的输入条件为FALSE时，执行以下动作。

- 从左母线直接输入到运动控制指令的输入变量Enable或Execute的指令作为FALSE动作。
- 内联ST部分如常动作。
- 运动控制指令的输入变量Enable或Execute作为FALSE动作时，输出参数值也如常更新。

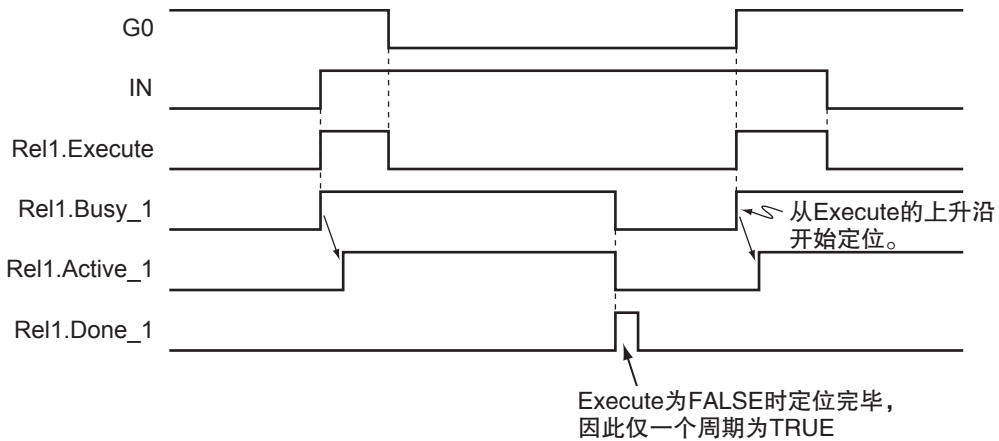
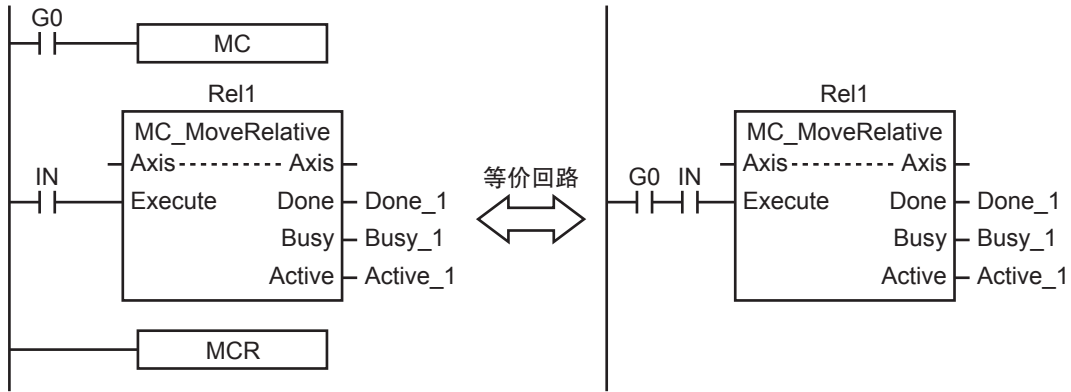
● Enable型运动控制指令时

- 配置到主站控制区域的指令与下图右侧的回路等价。
- G0为TRUE时，如常动作。
- G0为FALSE时，Enable输入与FALSE相同。



● Execute型运动控制指令时

- 配置到主站控制区域的指令与下图右侧的回路等价。
- G0为TRUE时，如常动作。
- G0为FALSE时，Execute输入与FALSE相同。
- G0为TRUE时启动的指令在动作过程中即使G0变为FALSE，也会继续动作直到完成。这种情况下输出参数的值也会如常更新。



使用注意事项

Execute型的运动控制指令是在上升沿开始动作，因此不推荐在主站控制区域内使用。使用时请充分注意动作情况。



参考

- ST程序中，MC指令(主站控制开始指令)的功能无效。所有指令如常动作。
有关MC指令/MCR指令的详情，请参阅 □ “NJ/NX系列 指令基准手册 基本篇(SBCA-360)”。

ST语法指令中

下面对运动控制指令配置到IF指令、CASE指令、WHILE指令、REPEAT指令等的ST语法指令中时的情况进行说明。

若与各ST语法指令的条件表达式的评价结果不一致，则不执行运动控制指令。也不更新指令的输出变量值。

在Execute型指令开始执行之后，即使评价结果不一致，也会继续执行指令直到处理完成。这种情况下，不更新指令的输出变量值。




使用注意事项

条件表达式的评价结果不一致时，由于执行中的Execute型指令的执行状态不明确，因此不建议在ST语法指令中使用。
使用时请充分注意动作情况。



参考

- 要根据条件表达式切换执行的Execute型指令，只需将Execute的输入参数配置到ST语法指令中。Execute型指令自身配置到ST语法指令外。
关于ST语法指令的详请，请参阅  “NJ/NX系列 指令基准手册 基本篇(SBCA-360)”。

实数(REAL型、LREAL型)的使用

实数(REAL型、LREAL型)是浮点型数据格式的数据类型。
下面对其表现方法和运算处理进行说明。

● REAL型、LREAL型的数据大小

REAL型和LREAL型的数据大小不一样。
REAL型的数据大小为32位，LREAL型为64位。

● 浮点形式的数据格式

浮点形式是指通过符号、指数、尾数表示实数的形式。
某个实数以下式表示时，s的值为符号、e的值为指数、f的值为尾数。

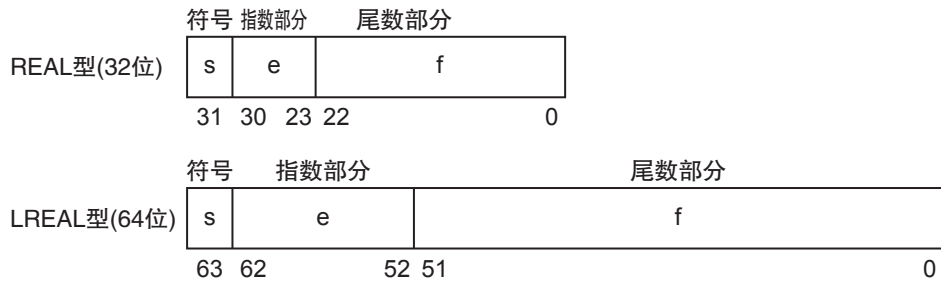
• REAL型

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

• LREAL型

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

浮点形式的数据格式以IEEE754标准为依据。详见下例。



(例) REAL型实数，表示为“-86.625”时

1

符号的设定

由于是负数，因此s = 1。

2

2进制表示

将“86.625”以2进制表示，为“1010110.101”。

3

2进制数的标准化

上述值的标准化表示为“1.010110101 × 2⁶”。

4

指数部的表示

根据前式，e - 127 = 6，因此e = 133。

“133”的2进制表示为“10000101”，这就是指数部的表示。

5

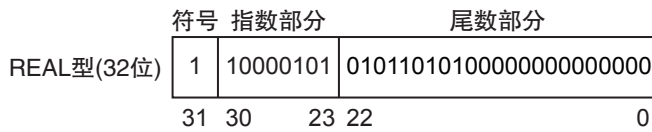
尾数部的表示

“1.010110101”的小数点以后的数值为“010110101”。

要将该小数点后的数值表示为23位，位数不足的部分需要附加0。以23位表示的数值为f部分。

因此，f = 01011010100000000000000。

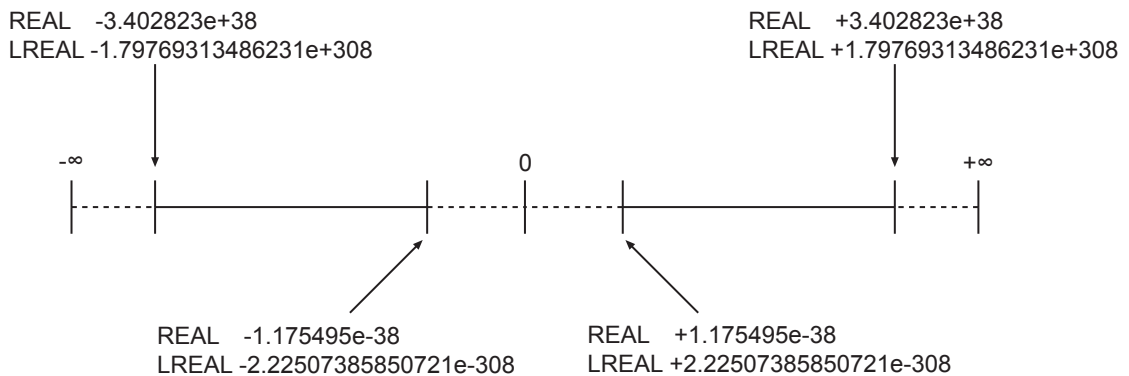
也就是说，“-86.625”可以以下图表示。



● 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”、非数被称为特殊值。

非数是指不视为浮点数的数据，不作为数值处理。

“+0”和“-0”虽然在数学上的含义都为“0”，但运算处理的内容却不同。

详情将于下文阐述。

特殊值的符号s、指数e、尾数f的值请见下表。

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	-	255	0以外
LREAL	$+\infty$	0	2047	0
	$-\infty$	1	2047	0
	+0	0	0	0
	-0	1	0	0
	非数	-	2047	0以外

● 非标准化数

浮点形式无法表示接近于“0”的数值(绝对值非常小的值)。
因此,为扩大接近于“0”的数值的有效范围而设置了非标准化数值。
比起以通常数据格式表示的数值,非标准化数可以表示绝对值很小的值。



参考

以通常数据格式表示的值叫做标准化数。

指数 $e=0$ 、尾数 $f \neq 0$ 的数值被看作非标准化数,其值如下表示。

- 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型实数,表示为“ 0.75×2^{-127} ”时

1

符号的设定

由于是正数,因此 $s=0$ 。

2

2进制表示

将“0.75”以2进制表示,为“0.11”。

3

尾数部的计算

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

4

尾数部的表示

由上式可知: $f = 0110000000000000000000$ 。

也就是说,“ 0.75×2^{-127} ”可以以下图表示。

	符号		指数部分				尾数部分																							
REAL型(32位)	0	00000000	011000000000000000000000000000																											
	31	30	23	22																									0	

与标准化数相比,非标准化数的有效位数较少。因此,标准化数的运算结果转换为非标准化数、或者运算途中的结果转换为非标准化数时,运算结果的有效位数可能达不到标准化数的有效位数。

● 运算处理

浮点形式表示的是近似值，与实际值之间存在误差。值的有效范围也有限制。因此，在运算时应执行以下处理。

取整

实际值超过尾数的有效位数时，按照以下规则对值进行取整。

- 将浮点形式表现的数值中最接近实际值的值作为运算结果。
- 在浮点形式表现的数值中有两个最接近实际值的值时，将尾数最低位位值为“0”的值作为运算结果。
- 有两个最接近实际值的值时，实际值相当于两个近似值的中间值。

溢出、下溢

实际值的绝对值大于以浮点形式表示的值，叫做溢出。相反，若小于以浮点形式表示的值，则叫做下溢。

- 溢出时，如果实际值的符号为正，则运算结果为“ $+\infty$ ”。
- 如果实际值的符号为负，则运算结果为“ $-\infty$ ”。
- 下溢时，如果实际值的符号为正，则运算结果为“+0”。
- 如果实际值的符号为负，则运算结果为“-0”。

特殊值的运算

特殊值的运算应遵守以下规则。

关于特殊值，请参阅“特殊值(P.1-15)”。

- “ $+\infty$ ”和“ $-\infty$ ”的相加结果为非数。
- 符号相同、无穷大之间的减法运算结果为非数。
- “+0”和无穷大、“-0”和无穷大的乘法运算结果为非数。
- “+0”之间的除法运算结果、“-0”之间的除法运算结果、无穷大之间的除法运算结果均为非数。
- “+0”和“-0”的相加结果为“+0”。
- “+0”之间的减法运算结果、“-0”之间的减法运算结果为“+0”。
- 含有非数的四则运算结果为非数。
- 在CMP指令等比较指令中，将“+0”和“-0”判断为相等。
- 在比较指令中，如果比较对象包括非数，则始终判断为不相等。



使用注意事项

MC功能模块中，电子齿轮的设定值及运动控制指令的目标位置等使用浮点(LREAL)的变量。因此，运算结果包含取整误差。

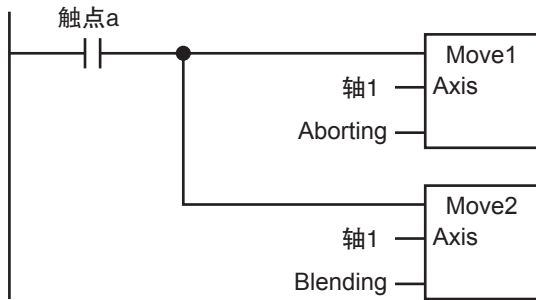
例如，重复执行MC_MoveRelative(相对值定位)指令时，误差会不断累积。

若误差累积造成问题，请将指令单位设置为[pulse]，或者指定类似于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

以下对在同一个任务周期内对同一个轴启动多个运动控制指令时的情况进行说明。

- 下图程序中，触点a “ON” 时，指令实例[Move1]和[Move2]在同一个任务周期内同时启动。
- 程序中记述的指令实际上按从上往下的顺序启动，因此[Move1]先启动，[Move2]在[Move1]动作完成之前启动。
- 这相当于运动控制指令的多重启动。在该程序中，[Move2]相对于[Move1]以Blending进行多重启动。

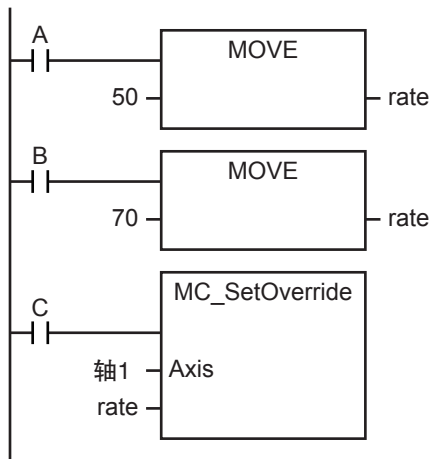


关于各运动控制指令的多重启动的详情，请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-363)”。



参考

若如上所示，同时启动MC_SetOverride(超调值设定)指令，则下侧的超调值生效。通过MC_SetOverride(超调值设定)指令区别使用多个超调值时，建议按以下方式编程。



运动控制指令的在线编辑

Sysmac Studio的在线编辑功能可以对于运动控制指令执行以下操作。

在线编辑操作
删除运动控制指令
添加运动控制指令
添加运动控制指令的输入、输出、输入输出变量
变更运动控制指令的输入、输出、输入输出变量
删除运动控制指令的输入、输出、输入输出变量



使用注意事项

如果在轴动作中删除MC_Stop(强制停止)指令、MC_GroupStop(轴组强制停止)指令等停止轴动作的指令，在某些用户程序内容下，轴的动作可能无法停止。
对运动控制指令执行在线编辑时，请确认安全后再使用。

CPU单元的动作模式切换

NJ/NX系列的CPU单元的动作模式有程序模式和运行模式两种。
下面对切换动作模式时的MC功能模块的动作进行说明。

从运行模式切换到程序模式

- 会中断执行中的运动控制指令。输出变量CommandAborted(执行中断)保持FALSE，但轴的动作与CommandAborted(执行中断)为TRUE时相同。
- 轴在动作中时，以最大减速速度减速并停止。伺服ON/OFF状态保持不变。
- 如果正在通过凸轮表保存指令执行凸轮表保存，则继续进行保存处理。
- 如果正在通过凸轮表生成指令执行凸轮表生成，则会继续执行生成处理。
- 配置到执行优先度为16的固定周期任务中的运动控制指令，在执行此任务的最终指令后再执行上述处理。

从程序模式切换到运行模式

- 运动控制指令的输出变量会被清除。
- 从运行模式变更为程序模式时，轴会减速停止。如果在减速停止过程中返回运行模式，由于运动控制指令的输出变量会被清除，已执行过的运动控制指令的CommandAborted(执行中断)会保持FALSE不变。



参考

- 动作模式切换后想要浏览运动控制指令的输出变量时，请分配输出参数属性为保持属性的变量。
浏览分配的输出参数，便可以查看切换动作模式前的输出变量。
 - 即使切换动作模式，伺服ON/OFF的状态仍会保持不变。
-

2

变量和指令

本章将列出MC功能模块使用的变量和指令的一览表。

2-1 变量一览表	2-2
MC通用变量	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
通用指令	2-26
轴指令	2-27
轴组指令	2-29
2-3 PDO映射	2-30
必需对象	2-30
不同指令的设定对象	2-32

2-1 变量一览表

MC功能模块使用的变量分为两类。

一类是监视轴等的状态及部分参数设定内容的系统定义变量。MC功能模块使用的系统定义变量被称作运动控制系统变量。

还有一类是运动控制指令将指令的自变量作为输入接收、将指令的执行状态作为输出的变量。运动控制指令的输入变量中，有几个是从选项中选择枚举体变量。

本章节将刊登各种变量的一览表、运动控制指令的输入变量的有效范围、以及枚举体变量的一览表。

● 运动控制系统变量

第1层	第2层	第3层	内容
系统定义变量	运动控制系统变量	MC通用变量	可监控MC功能模块的通用状态。
		轴变量	可监控各轴的状态和部分轴参数的设定内容。
		轴组变量	可监控各轴组的状态和部分轴组参数的设定内容。

关于运动控制系统变量的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”。



参考

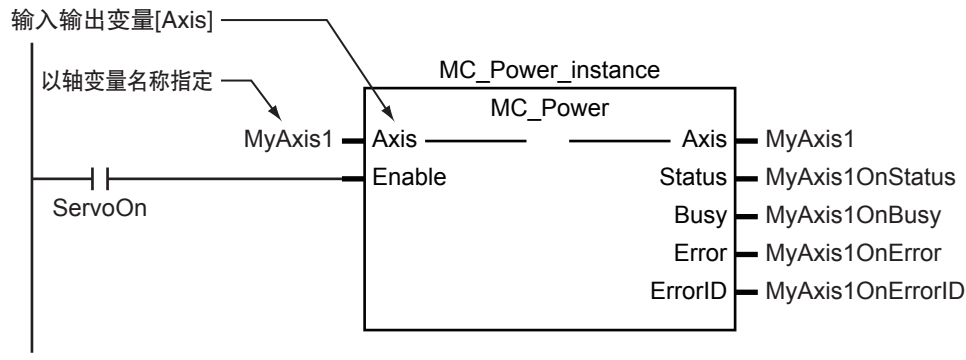
轴变量和轴组变量均为系统定义变量。

这些变量在用户程序中使用，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，除了上述以外，还可使用通过Sysmac Studio创建的变量名称。

通过Sysmac Studio创建的轴及轴组的变量名称可变更为任意名称。

下图是关于系统定义变量的轴变量名称为“_MC_AX[0]”的轴在通过Sysmac Studio添加时的轴变量名称变更为“MyAxis1”时的例子。



● 运动控制指令的变量

种类	概要
输入变量	指令的自变量
输出变量	指令执行状态的监视
输入输出变量	指令执行对象的指定



参考

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

关于MC功能模块使用的数据类型，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-363)”。

MC通用变量

MC通用变量是变量名称为_MC_COM、数据类型为_sCOMMON_REF型的结构体变量。

下面对MC通用变量的构成及其要素进行详细说明。

变量名称	数据类型	名称	功能
_MC_COM	_sCOMMON_REF	MC通用变量	
Status	_sCOMMON_REF_STA	MC通用状态	
RunMode	BOOL	MC运行中	MC功能模块在运行中变为TRUE
TestMode	BOOL	MC试运行中	通过Sysmac Studio试运行时变为TRUE
CamTableBusy	BOOL	凸轮表文件保存执行中/ 待机中	在执行或等待执行凸轮表文件保存时变为TRUE
GenerateCam Busy *1	BOOL	凸轮表生成执行中	执行凸轮表生成时变为TRUE
PFaultLvl	_sMC_REF_EVENT	MC通用 部分停止故障	
Active	BOOL	MC通用 部分停止 故障发生中	MC通用 发生停止故障时变为TRUE
Code	WORD	MC通用 部分停止 故障代码	MC通用 输出停止故障的代码。 与事件代码的前四位相同的值
MFaultLvl	_sMC_REF_EVENT	MC通用 轻度故障	
Active	BOOL	MC通用 轻度故障发生中	MC通用 发生轻度故障时变为TRUE
Code	WORD	MC通用 轻度故障代码	MC通用 输出轻度故障的代码。 与事件代码的前四位相同的值
Obsr	_sMC_REF_EVENT	MC通用 监视信息	
Active	BOOL	MC通用 监视信息发生中	MC通用 发生监视信息时变为TRUE
Code	WORD	MC通用 监视信息代码	MC通用 输出监视信息的代码。 与事件代码的前四位相同的值

*1. Ver.1.08以上的CPU单元和Ver.1.09以上的Sysmac Studio组合时可使用。

轴变量

轴变量是系统定义变量的名称为 `_MC_AX[0-255]`或`_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]` 为例对轴变量的构成及其要素进行详细说明。`_MC1_AX[0-255]`、`_MC2_AX[0-255]` 也是如此。

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	轴启动准备就绪并在停止中的状态下变为TRUE。 <code>_MC_AX[*].Status.Standstill</code> 表示与“TRUE: 停止中”的状态相同。
Disabled	BOOL	轴无效	在轴由于伺服OFF而停止的状态下变为TRUE。 以下轴状态是排他性的，任意一个单独变为TRUE Disabled/ Standstill/ Discrete/ Continuous/ Synchronized/ Homing/ Stopping/ ErrorStop/ Coordinated
Standstill	BOOL	停止中	在轴由于伺服ON而停止的状态下变为TRUE。
Discrete	BOOL	定位动作中	在向目标位置执行定位控制时变为TRUE。 也包括定位动作中将超调值设定为“0”而使速度为“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。 速度控制和转矩控制时变为此状态。 也包括将目标速度设为“0”而使速度为“0”的状态、以及连续动作中将超调值设定为“0”而使速度为“0”的状态。
Synchronized	BOOL	同步动作中	执行同步控制时变为TRUE。 也包括同步控制指令切换后的同步等待状态
Homing	BOOL	原点复位中	通过MC_Home指令或MC_HomeWithParameter指令执行原点复位时变为TRUE
Stopping	BOOL	减速停止中	在通过MC_Stop指令、MC_TouchProbe指令停止轴动作之前的状态下变为TRUE。 也包括通过MC_Stop指令停止轴动作后Execute变为TRUE时。 在减速停止中的状态下不能启动轴动作指令 (CommandAborted = TRUE)
ErrorStop	BOOL	错误减速停止中	执行MC_ImmediateStop指令或正在发生轻度故障(<code>_MC_AX[*].MFaultLvl.Active</code> 为“TRUE: 正在发生轴轻度故障”)时，轴停止前和停止后的状态。 在该状态下不能启动轴动作指令 (CommandAborted = TRUE)。

变量名称	数据类型	名称	功能
Coordinated	BOOL	多轴协调动作中	在通过多轴协调指令使轴组有效的状态下变为TRUE
Details	_sAXIS_REF_DET	轴控制状态 *1	
Idle	BOOL	停止中	除了到位等待，在未进行指令值运算时变为TRUE。*2 Idle与InPosWaiting相互排斥，不会同时变为TRUE。
InPosWaiting	BOOL	到位等待	到位等待时变为TRUE。 通过到位检查执行定位执行到位检查的状态
Homed	BOOL	原点确定	在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变为TRUE。 以下是AND条件 • 原点确定状态 • 反馈当前位置以原点为中心进入“原点位置范围”时 在指令状态下轴动作过程中通过原点位置时也会变为TRUE。
VelLimit *4	BOOL	指令速度饱和	同步控制中，指速度以最高速度进行限制时变为TRUE。
Dir	_sAXIS_REF_DIR	指令方向状态 *5	
Posi	BOOL	正方向指令中	指令指定为正方向时变为TRUE。
Nega	BOOL	负方向指令中	指令指定为负方向时变为TRUE。
DrvStatus	_sAXIS_REF_STA_DRV	伺服驱动器状态 *6	
ServoOn	BOOL	伺服ON	伺服电机通电时变为TRUE
Ready	BOOL	伺服待机	伺服待机状态时变为TRUE
MainPower	BOOL	主电路电源	伺服驱动器的主电路电源通电时变为TRUE
P_OT	BOOL	正方向极限输入	正方向极限输入有效时变为TRUE
N_OT	BOOL	负方向极限输入	负方向极限输入有效时变为TRUE
HomeSw	BOOL	近原点输入	近原点输入有效时变为TRUE
Home	BOOL	原点输入	启用原点输入时变为TRUE。*7
ImdStop	BOOL	即停输入	即停输入有效时变为TRUE
Latch1	BOOL	外部锁定输入1	锁定输入1有效时变为TRUE
Latch2	BOOL	外部锁定输入2	锁定输入2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	通过伺服驱动器的限制功能进行实际限制时变为TRUE。*8
CSP	BOOL	周期同步位置(CSP)模式中	伺服驱动器的伺服ON，在CSP模式下变为TRUE。*9
CSV	BOOL	周期同步速度(CSV)模式中	伺服驱动器侧伺服ON、进入CSV模式时变为TRUE *9

变量名称	数据类型	名称	功能
CST	BOOL	周期同步转矩(CST)模式中	伺服驱动器侧伺服ON、进入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或非位置控制模式时, 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) 正方向移动时附带正号、负方向移动时附带负号。 根据指令当前位置之差算出的速度。伺服OFF或非位置控制模式时, 根据反馈当前位置计算出的速度。
AccDec	LREAL	指令当前加减速速度	输出指令加减速速度的当前值。 (单位: 指令单位/s ²) 根据指令当前速度之差算出的加减速速度。加速时附带正号, 减速时附带负号。 执行中的指令的加减速速度为“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) 加减速速度的绝对值增大时附带正号, 减小时附带负号。 执行中的指令的加减速速度或指令跃度为“0”时变为“0”。
Trq	LREAL	指令当前转矩	输出指令转矩的当前值。(单位: %) 正方向移动时附带正号、负方向移动时附带负号。 非转矩控制模式时, 输出与反馈当前转矩相同的值。
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) 正方向移动时附带正号、负方向移动时附带负号。
Trq	LREAL	反馈当前转矩	输出反馈转矩的当前值。(单位: %) 正方向移动时附带正号、负方向移动时附带负号。
TimeStamp *11	ULINT	时间戳	输出轴当前位置的更新时刻。运行变化时刻获取功能的轴有效。 (单位: ns)
MFaultLvl	_sMC_REF_EVENT	轴 轻度故障	
Active	BOOL	轴 轻度故障发生中	轴发生轻度故障时变为TRUE
Code	WORD	轴 轻度故障代码	输出轴轻度故障的代码。 与事件代码的前四位相同的值

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	发生轴监视信息时变为TRUE
Code	WORD	轴 监视信息代码	输出轴监视信息的代码。 与事件代码的前四位相同的值
Cfg	_sAXIS_REF_CFG	轴基本设定 *12	
AxNo	UINT	轴号	表示轴的逻辑编号。 参考_sAXIS_REF时是为识别轴号而进行参考
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接线 0: _mcServo(伺服轴) 1: _mcEncdr(编码器轴) 2: _mcVirServo(虚拟伺服轴) 3: _mcVirEncdr(虚拟编码器轴)
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: 分配至原始恒定周期任务 2: 分配至固定周期任务(执行优先级5)
Scale	_sAXIS_REF_SCALE	单位转换 *15	
Num	UDINT	电机转1圈的脉冲数	表示指令位置电机转1圈的脉冲数。 根据电子齿轮比将指令值转换为脉冲量
Den	LREAL	电机转1圈的移动量	表示指令位置电机转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(线性模式) 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、同步控制中以及多轴协调动作中也属于正在运算的状态。

*3. 即使为TRUE，以下情况仍需重新确定原点。

变更位置计数设定或单位转换设定时。

在伺服驱动器侧发生异常或进行操作导致丢失绝对位置信息时。例如，编码器电缆断线或绝对值编码器数据清零等。

*4. VelLimit请仅用于同步控制中的从轴。

*5. 表示指令指定的移动方向。

*6. 表示驱动器等的状态。

*7. 通过Sysmac Studio的[轴基本设置]画面中的[高级设置]设定为[数字输入]的[编码器Z相检测]的信号的状态。使用其他公司生产的驱动器可能无法将此信号映射至PDO。请确认所连接的驱动器的手册。

*8. 表示映射至PDO的“状态字(6041Hex)”的位11(启用内部限制功能)的状态。变为TRUE的条件取决于伺服驱动器的规格。请参阅所连接的伺服驱动器的使用手册。欧姆龙制伺服驱动器IS系列及G5系列为转矩限制/速度限制/驱动禁止输入/软件限制的其中之一。

- *9. 根据映射到PDO的“操作模式显示(6061Hex)”的值进行显示。CSP、CSV、CST变为TRUE的条件取决于伺服驱动器的规格。请参阅所连接的伺服驱动器的使用手册。
如“操作模式显示(6061Hex)”未映射至PDO，不同版本CPU单元的显示如下所示。
对于Ver.1.09以下版本的CPU单元，始终为FALSE。
对于Ver.1.10以上版本的CPU单元，如映射至PDO的“状态字(6041Hex)”的状态为Operation Enabled，则为TRUE。
- *10. 如分配给轴的EtherCAT从站或NX单元与CPU单元之间尚未确立过程数据通信，不同版本CPU单元的显示如下所示。
对于Ver.1.09以下版本的CPU单元，轴变量的反馈当前位置及指令当前位置为“0”或下限值。下限值的条件是计数模式为旋转模式时位置范围中不含“0”。
对于Ver.1.10以上版本的CPU单元，轴变量的反馈当前位置及指令当前位置为尚未确立过程数据通信前的反馈当前位置。
- *11. Ver.1.06以上的CPU单元和Ver.1.07以上的Sysmac Studio组合时可使用。
- *12. 表示轴基本设定内容。
- *13. 对于NX系列位置接口单元，表示安装了位置接口单元的EtherCAT耦合器单元的节点地址。
- *14. Ver.1.10以上的CPU单元和Ver.1.12以上的Sysmac Studio组合时可使用。
- *15. 表示电子齿轮比的设定值。
- *16. Ver.1.11以上版本的CPU单元新增的单位转换设定中，减速机使用设定为“使用”时，该参数无效。确认反之有效的参数“工件侧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. 计数模式为线性模式时，表示溢出前的位置。为旋转模式时，表示环计数器上限值。
- *18. 计数模式为线性模式时，表示下溢前的位置。为旋转模式时，表示环计数器下限值。

轴组变量

轴组变量是系统定义变量的名称为_MC_GRP[0-63]或_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]为例对轴组变量的构成及其要素进行详细说明。_MC1_GRP[0-63]、_MC2_GRP[0-63]也是如此。

功能说明中以_MC_AX[*]为例对轴变量进行了说明，_MC1_AX[*]、_MC2_AX[*]也是如此。

变量名称	数据类型	名称	功能
_MC_GRP[0-63]	_sGROUP_REF	轴组变量	
Status	_sGROUP_REF_STA	轴组状态	
Ready	BOOL	启动准备完成	轴组停止、启动准备已经完成时变为TRUE。 以下AND条件即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。 • _MC_GRP[*].Status.Standby(停止中)变为TRUE • 构成轴为伺服ON状态 • 构成轴的_MC_AX[*].Details.Homed(原点确定)为TRUE
Disabled	BOOL	不启用轴组	因禁用轴组而停止时变为TRUE。 以下轴组状态是排他性的，任意一个单独变为TRUE。 Disable/ Standby/ Moving/ Stopping/ ErrorStop

变量名称	数据类型	名称	功能
Standby	BOOL	停止中	轴组动作指令停止时变为TRUE。 与构成轴组的轴的伺服ON/OFF状态无关
Moving	BOOL	动作中	在向目标位置执行轴组动作指令时变为TRUE。 也包括到位等待的状态、以及因超调而使速度为“0”的状态。
Stopping	BOOL	减速停止中	在通过MC_GroupStop指令停止轴组动作之前的状态下变为TRUE。 也包括通过MC_GroupStop指令停止轴动作后Execute变为TRUE时。 在该状态下不能启动轴组动作指令 (CommandAborted = TRUE)
ErrorStop	BOOL	错误减速停止中	通过MC_GroupImmediateStop指令、或者在发生轴组轻度故障时 (_MC_GRP[*].MFAultLvl.Active=TRUE)，轴组停止之前、以及停止后变为TRUE。在此状态下不能启动轴组动作指令(CommandAborted = TRUE)。
Details	_sGROUP_REF_DET	轴组控制状态 *1	
Idle	BOOL	停止中	除了到位等待，在未进行指令值运算时变为TRUE。*2 Idle与InPosWaiting相互排斥，不会同时变为TRUE。
InPosWaiting	BOOL	到位等待	任意一个构成轴到位等待时变为TRUE。 通过到位检查执行定位执行到位检查的状态
Cmd	_sGROUP_REF_CMD_DATA	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之差算出的插补速度。 正方向移动时附带正号、负方向移动时附带负号。 轴组无效时变为“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。 根据指令插补速度之差算出的插补加减速度。 加速时附带正号，减速时附带负号。 禁用轴组或正在执行的轴组动作指令的指令加减速度指定为“0”时变为“0”。
MFAultLvl	_sMC_REF_EVENT	轴组轻度故障	
Active	BOOL	轴组轻度故障发生中	轴组发生轻度故障时变为TRUE
Code	UINT	轴组轻度故障代码	输出轴组轻度故障的代码。 与事件代码的前四位相同的值

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	发生轴组监视信息时变为TRUE
Code	WORD	轴组监视信息代码	输出轴组监控信息的代码。 与事件代码的前四位相同的值
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。 参考_sGROUP_REF时是为识别轴组编号而进行参考
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	表示各分配任务的执行ID。 0: 未分配任务(未创建组) 1: 分配至原始恒定周期任务 2: 分配至固定周期任务(执行优先级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)	表示向轴A0分配的轴号
Axis[1]	UINT	构成轴选择(轴A1)	表示向轴A1分配的轴号
Axis[2]	UINT	构成轴选择(轴A2)	表示向轴A2分配的轴号
Axis[3]	UINT	构成轴选择(轴A3)	表示向轴A3分配的轴号

*1. 表示指令的控制状态。

*2. 以速度“0”动作也属于正在运算的状态。

*3. Ver.1.10以上的CPU单元和Ver.1.12以上的Sysmac Studio组合时可使用。

*4. 表示轴组的运动学转换定义。

运动控制指令的输入变量

以下是运动控制指令的输入变量一览表、有效范围一览表以及枚举体的有效范围一览表。

输入变量一览表

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	在上升沿开始指令。 在Execute的上升沿时，导入其他输入变量。 想要更新输入值时，先变更输入值，然后再度启动Execute。 指令执行完成后也是Execute变为TRUE时输出变量有效。 此后，在Execute的下降沿，Error和ErrorID以外的输出变量无效。 指令执行完成前Execute变为FALSE时，输出变量至少在1个周期内有效
Enable	有效	BOOL	TRUE, FALSE	FALSE	变为TRUE时指令的功能生效，变为FALSE时无效。 Enable为TRUE时，其他输入变量按周期导入。 将Enable变为FALSE时，Error和ErrorID以外的输出变量无效
Positive Enable	正方向有效	BOOL	TRUE, FALSE	FALSE	<ul style="list-style-type: none"> MC_MoveJog指令 设为TRUE，则开始正方向移动。设为FALSE，则结束移动。MC_MoveJog指令的Velocity、Acceleration、Deceleration导入PositiveEnable的上升沿时的值 MC_SetTorqueLimit指令 变为TRUE时正方向的转矩限制有效。变为FALSE时正方向的转矩限制无效。
Negative Enable	负方向有效	BOOL	TRUE, FALSE	FALSE	<ul style="list-style-type: none"> MC_MoveJog指令 设为TRUE，则开始负方向移动。设为FALSE，则结束移动。MC_MoveJog指令的Velocity、Acceleration、Deceleration导入NegativeEnable的上升沿时的值 MC_SetTorqueLimit指令 变为TRUE时负方向的转矩限制有效。变为FALSE时负方向的转矩限制无效。
BufferMode	缓存模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 ^{*1}	指定多重启动运动控制指令时的动作。 0: 中断 1: 缓存 2: 以低速合并 3: 以前一个速度合并 4: 以后一个速度合并 5: 以高速合并
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}
Distance	移动距离	LREAL	负数、正数、“0”	0	指定自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}

2 变量和指令

输入变量	名称	数据类型	有效范围	初始值	内容
VelFactor	速度超调值	LREAL	0 ~ 500	100	指定速度的超调值。 超调值的有效范围为“0.01 ~ 500.00”。 “500.00 以上”作为“500”处理，“0.01 以下(包括负数)”作为“0.01”处理。 只有指定为“0”时才作为“0”动作。 单位为[%]
AccFactor (Reserved)	加减速度超调值	LREAL	0 ~ 500	100	(Reserved)
JerkFactor (Reserved)	跃度超调值	LREAL	0 ~ 500	100	(Reserved)
Reference Type *7	位置类型选择	_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(最近任务周期 *8 下的计算值) 1: 反馈位置(同一任务周期 *8 下的取值) 2: 指令位置(同一任务周期 *8 下的计算值)
Feed Distance	标准距离	LREAL	负数、正数、“0”	0	指定中断输入后的移动距离
FeedVelocity	标准速度	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, 指定是否执行异常检测。 TRUE: 执行异常检测 FALSE: 不执行异常检测
Periodic	重复模式	BOOL	TRUE, FALSE	FALSE	指定是反复执行指定的凸轮表, 还是只执行1次。 TRUE: 重复 FALSE: 不重复
StartMode	起始位置方式 指定	_eMC_ START_MODE	0: _mcAbsolutePosition 1: _mcRelativePosition	0 *1	指定MasterStartDistance(主轴跟踪距离)使用的坐标系。 0: 绝对位置 1: 相对位置
StartPosition	凸轮表起始位置	LREAL	负数、正数、“0”	0	利用主轴的绝对位置指定凸轮表的起点(相位=0)。
MasterStart Distance	主轴跟踪距离	LREAL	负数、正数、“0”	0	指定从轴开始凸轮动作时主轴的位置。 StartMode(起始位置方式指定)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, 指定自StartPosition(凸轮表起始位置)起的相对量。
Master Scaling	主轴系数	LREAL	正值(>0.0)	1.0	以指定的比例放大/缩小主轴的相位
SlaveScaling	从轴系数	LREAL	正值(>0.0)	1.0	以指定的比例放大/缩小从轴的位移
MasterOffset	主轴偏移	LREAL	负数、正数、“0”	0	以指定的偏移值移动主轴的相位
SlaveOffset	从轴偏移	LREAL	负数、正数、“0”	0	以指定的偏移值移动从轴的位移
Cam Transition (Reserved)	凸轮过渡指定	_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: 指定为正方向 1: 指定为附近 2: 指定为负方向 3: 指定为当前方向 4: 无方向指定
Continuous (Reserved)	持续方法选择	BOOL	TRUE, FALSE	FALSE	(Reserved)
Ratio Numerator	齿轮比分子	DINT*9	正数或负数 *9	10000	指定主轴和从轴间的电子齿轮的分子
Ratio Denominator	齿轮比分母	UDINT*10	正数	10000	指定主轴和从轴间的电子齿轮的分母
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

输入变量	名称	数据类型	有效范围	初始值	内容
Master Distance	主轴移动距离	LREAL	正数或“0”	0	指定主轴的移动距离
Master DistanceIn 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 DistanceIn 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: 主轴到达主轴跟踪距离时
Combine Mode	加减法运算方法选择	_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	指定主轴和从轴间的电子齿轮的分子
Ratio DenominatorMaster (Reserved)	主轴齿轮比分母	UDINT ^{*10}	正数	10000	指定主轴和从轴间的电子齿轮的分母
Ratio Numerator Auxiliary (Reserved)	辅轴齿轮比分子	DINT ^{*9}	正数或负数 ^{*9}	10000	指定辅轴和从轴间的电子齿轮的分子
Ratio DenominatorAuxiliary (Reserved)	辅轴齿轮比分母	UDINT ^{*10}	正数	10000	指定辅轴和从轴间的电子齿轮的分母
Reference TypeMaster	主轴位置类型选择	_eMC_REFERENCE_TYPE	1: _mcFeedback 2: _mcLatestCommand	2 ^{*1}	指定主轴的位置类型 1: 反馈位置(同一任务周期 ^{*8} 下的取值) 2: 指令位置(同一任务周期 ^{*8} 下的计算值)
Reference TypeAuxiliary	辅轴位置类型选择	_eMC_REFERENCE_TYPE	1: _mcFeedback 2: _mcLatestCommand	2 ^{*1}	指定辅轴的位置类型 1: 反馈位置(同一任务周期 ^{*8} 下的取值) 2: 指令位置(同一任务周期 ^{*8} 下的计算值)
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	目标转矩	LREAL	0 ~ 1000.0	300.0	以0.1%为单位指定向伺服驱动器输出的目标转矩。 以额定转矩为100%时的比率进行指定。单位为“%”
TorqueRamp	转矩斜度	LREAL	正数或“0”	0	指定从当前转矩到输出目标转矩为止的转矩变化率。 单位为“%/s”
Positive Value	正方向转矩限制值	LREAL	0.1 ~ 1000.0、 或“0.0”	300.0	以0.1%为单位指定正方向的转矩限制值。 输入值超过轴参数的[正方向转矩限制上限值]时, 使用[正方向转矩限制上限值]。 指定为“0”及“负数”时作为“0”动作。
Negative Value	负方向转矩限制值	LREAL	0.1 ~ 1000.0、 或“0.0”	300.0	以0.1%为单位指定负方向的转矩限制值。 输入值超过轴参数的[负方向转矩限制上限值]时, 使用[负方向转矩限制上限值]。 指定为“0”及“负数”时作为“0”动作。
WindowOnly	窗口有效	BOOL	TRUE, FALSE	FALSE	指定窗口掩码有效/无效
FirstPosition	起始位置	LREAL	负数、正数、“0”	0	指定起始位置。
LastPosition	终止位置	LREAL	负数、正数、“0”	0	指定终止位置。
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, 并进行偏差计数器复位 3: 立即停止, 并变为伺服OFF 4: 不停止
Relative (Reserved)	相对位置选择	BOOL	TRUE, FALSE	FALSE	(Reserved)

2 变量和指令

输入变量	名称	数据类型	有效范围	初始值	内容
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	指定可容许的主轴及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。 原点确定状态保持不变
CoordSystem	坐标系	_eMC_COORD_ SYSTEM	0: _mcACS	0 ^{*1}	指定坐标系 0: 轴坐标系(ACS)
Transition Mode	过渡模式 (切换模式)	_eMC_ TRANSITION_ MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 ^{*1}	指定动作的路径 0: 过渡无效 10: 附加角
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。
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}	指定写入目标位置的参数 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: 正方向转矩警告值 4: 负方向转矩警告值 5: 位置偏差超过值 6: 位置偏差警告值 7: 软件限制功能 8: 正方向软件限制 9: 负方向软件限制 10: 到位检查时间 11: 到位宽度 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}	指定要改写的[原点复位动作]。 0: 指定为附近避让、近原点输入OFF 1: 指定为附近避让、近原点输入ON 4: 指定为近原点输入OFF 5: 指定为近原点输入ON 8: 指定为极限输入OFF 9: 指定为附近避让、原点输入屏蔽距离 11: 仅极限输入 12: 指定为附近避让、接触时间 13: 指定为无近原点输入、接触原点输入 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

输入变量	名称	数据类型	有效范围	初始值	内容
EnableMask *15	轨段有效	WORD	16#0000 ~ FFFF	0	按轨段指定有效或无效。最多16个轨段中，第0位的值指定轨段编号0的有效或无效，第15位的值指定轨段编号15的有效或无效。 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	指定需计算位置的时间戳。 指定以时间戳方式对应的数字输入单元和编码器输入单元内的时刻为基准的时间戳。单位为[ns]。
ExecID *16	执行ID	UINT	2	2	指定变量同步指定对象的ID。 2: 固定周期任务(执行优先度5)
OffsetPosition	位置偏置	LREAL	负数、正数、“0”	0	指定在指令当前位置叠加的位置偏置 *6

- *1. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。
- *2. 也有MC_MoveJog指令及MC_MoveVelocity指令等可以设定为“0”的指令。
- *3. 以指令单位/s表示。指令单位有[mm]、[μm]、[nm]、[degree]、[inch]或[pulse]。
- *4. 以指令单位/s²表示。
- *5. 以指令单位/s³表示。
- *6. 以指令单位表示。
- *7. 使用_mcLatestCommand时，必须在满足以下关系的条件下使用主轴、从轴。
Master(主轴)设定的运动控制系统变量的轴号 < Slave(从轴)设定的运动控制系统变量的轴号
- *8. 任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先度5)的周期。
- *9. Ver.1.02以上的CPU单元和Ver.1.03以上的Sysmac Studio组合时可使用。上述以下版本进行组合时，数据类型为“UINT”，有效范围为“正数”。
- *10. Ver.1.02以上的CPU单元和Ver.1.03以上的Sysmac Studio组合时可使用。上述以下版本进行组合时，数据类型为“UINT”。
- *11. Ver.1.01以上的CPU单元和Ver.1.02以上的Sysmac Studio组合时可使用。
- *12. Ver.1.10以上的CPU单元和Ver.1.12以上的Sysmac Studio组合时可使用。
- *13. Ver.1.03以上的CPU单元和Ver.1.04以上的Sysmac Studio组合时可使用。
- *14. Ver.1.04以上的CPU单元和Ver.1.05以上的Sysmac Studio组合时可使用。
- *15. Ver.1.06以上的CPU单元和Ver.1.07以上的Sysmac Studio组合时可使用。
- *16. NX1P2 CPU单元及NJ系列CPU单元中没有。

输入变量的有效范围

下面对运动控制指令的输入变量的有效范围进行说明。各指令的有效范围请参阅各指令的说明。

● BOOL型输入变量

FALSE以外的均视为TRUE。因此，不会发生范围外异常。

● 枚举(ENUM)体输入变量

有效范围外作为异常处理。

● 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范围外数值时的动作如下表所示。



输入变量	名称	有效范围	最大值范围外	最小值范围外(0除外)
Velocity	速度	0或者 ($-1 \leq \text{且} \leq -\text{“最高速度”}$)、 ($1 \leq \text{且} \leq -\text{“最高速度”}$) ^{*1}	正数时：“最高速度” 负数时： $-\text{“最高速度”}$ 为固定值 ^{*2}	正数时：1pulse/s 负数时： -1pulse/s 为固定值 ^{*2}
Acceleration	加速度	0或者 ($0.004 \leq \text{且} \leq \text{“最大加速度”}$) ^{*3}	固定为最大加速度。 但，加速时间短于 ^{*4} 125 μs 时，固定为125 μs。	正数时：固定为 0.004pulse/s ² 。 但，加速时间 ^{*4} 超过250s 时，固定为250s。 负数时会发生异常。
Deceleration	减速度	0或者 ($0.004 \leq \text{且} \leq \text{“最大减速度”}$) ^{*5}	固定为最大减速度。 但，减速时间 ^{*4} 短于125 μs 时，固定为125 μs。	正数时：固定为 0.004pulse/s ² 。 但，减速时间 ^{*4} 超过250s 时，固定为250s。 负数时会发生异常。
Jerk	跃度	0或者 ($0.000016 \leq \text{且}$ $25,600,000,000,000,000 \text{ pulse/s}^3$)	固定为 25,600,000,000,000,000 pulse/s ³ 。 但，加速跃度适用时间 ^{*6} 或减速跃度适用时间 ^{*6} 短于 125 μs时，均固定为125 μs	固定为0.000016 pulse/s ³ 。 但，加速跃度适用时间 ^{*6} 或 减速跃度适用时间 ^{*6} 超过 250s时，均固定为250s。 负数时会发生异常。
Distance	移动距离	($0\text{xFFFFFF0000000001} \leq \text{且}$ $\leq (0\text{x000000FFFFFF})$)	会发生异常。	无最小值范围外。
Position	指令位置	($0\text{xFFFFFF8000000000} \leq \text{且}$ $< (0\text{x0000007FFFFFFF}+1)$) ^{*7}	会发生异常。	无最小值范围外。
VelFactor	超调值	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或者 ($0.000\ 000\ 000\ 000\ 000\ 4 \leq \text{且} \leq \text{“最大插补加速度”}$) ^{*10}	固定为 “最大插补加速度”。 但，插补加速时间短于 ^{*11} 125 μs时，固定为125 μs	正数时：固定为0.000 000 000 000 4 pulse/s ² 。但，插补加 速时间 ^{*11} 超过250s时，固 定为250s。 负数时会发生异常。
Deceleration	插补 减速度	0或者 ($0.000\ 000\ 000\ 000\ 000\ 04 \leq \text{且}$ $\leq \text{“最大插补减速度”}$) ^{*12}	固定为“最大插补减速度”。 但，插补减速时间 ^{*11} 短于 125 μs时，固定为125 μs。	正数时：固定为0.000 000 000 000 4 pulse/s ² 。但，插补减 速时间 ^{*11} 超过250s时，固 定为250s。 负数时会发生异常。

输入变量	名称	有效范围	最大值范围外	最小值范围外 (0除外)
Jerk	插补跃度	0或者 (0.000 000 000 000 001 6 ≤ 且 ≤ 51,200,000,000,000,000 pulse/s ³)	固定为 51,200,000,000,000,000 pulse/s ³ 。 但, 插补加速跃度适用时间 *13或插补减速跃度适用时 间 *13短于125 μs时, 均固 定为125 μs。	固定为0.000 000 000 000 001 6 pulse/s ³ 。 但, 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, 均固定为 250s。 负数时会发生异常

- *1. 可设定的最大值为转换为脉冲单位的值, 如下所示。
对于Ver.1.02以下版本的CPU单元, 为400,000,000[pulse/s],
对于Ver.1.03以上版本的CPU单元, 为500,000,000[pulse/s],
对于Ver.1.11以上版本的CPU单元, 为2,147,483,647[pulse/s]。
- *2. 负数及“0”不在有效范围内时, 若指定为负数及“0”, 则会出现异常。
- *3. 轴参数“最大加速度”的上限为3,200,000,000,000 pulse/s²。
- *4. 按照“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”进行计算。
- *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。
按照“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”来计算。
- *7. 脉冲单位的绝对位置必须在带符号的40bit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”的上限的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. 按照“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”进行计算。
- *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。
按照“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”来计算。

枚举体一览表

以下是运动控制指令的输入变量使用的ENUM体变量一览表。



枚举体输入变量不是通过数值设定，而是通过枚举元素进行设定。

数据类型	有效范围	内容	适应指令的变量 (变量名称)
_eMC_BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	指定多重启动运动控制指令时的动作 0: 中断 1: 缓存 2: 以低速合并 3: 以前一个速度合并 4: 以后一个速度合并 5: 以高速合并	BufferMode (缓存模式选择)
_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/修正	CamTransition (凸轮过渡指定)
_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 (加减法运算方法选择)
_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: 指定为正方向 1: 指定为附近 2: 指定为负方向 3: 指定为当前方向 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: 主轴到达主轴跟踪距离时	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	适应指令的变量 (变量名称)
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	指定写入目标位置的参数 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: 正方向转矩警告值 4: 负方向转矩警告值 5: 位置偏差超过值 6: 位置偏差警告值 7: 软件限制功能 8: 正方向软件限制 9: 负方向软件限制 10: 到位检查时间 11: 到位宽度 12: 启动速度	ParameterNumber (参数编号)
eMC SWLMT_MODE	0: _mcNonSwLmt 1: _mcCmdDecelerationStop 2: _mcCmdImmediateStop 3: _mcActDecelerationStop 4: _mcActImmediateStop	指定软件限制功能有效/无效以及停止方法 0: 软件限制功能无效 1: 有效, 对指令位置执行减速停止 2: 有效, 对指令位置执行立即停止 3: 有效, 对反馈位置执行减速停止 4: 有效, 对反馈位置执行立即停止	SettingValue (设定值)
eMC REFERENCE_ TYPE * ³	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	指定位置类型 0: 指令位置(最近任务周期* ⁴ 下的计算值) 1: 反馈位置(同一任务周期* ⁴ 下的取值) 2: 指令位置(同一任务周期* ⁴ 下的计算值)	ReferenceType (位置类型选择) ReferenceTypeMaster (主轴位置类型选择) ReferenceTypeAuxiliary (辅轴位置类型选择)
eMC START_MODE	0: _mcAbsolutePosition 1: _mcRelativePosition	指定MasterStartDistance(主轴跟踪距离)使用的坐标系。 0: 绝对位置 1: 相对位置	StartMode (起始位置方式指定)
_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, 并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2个锁定功能中的哪一个。 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 原点确定状态保持不变	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: 过渡无效 10: 附加角	TransitionMode (过渡模式(切换模式))
_eMC_TRIGGER_ MODE	0: _mcDrive 1: _mcController	指定触发模式 0: 驱动器模式 1: 控制器模式	Mode(模式)
_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 (驱动触发输入信号)

数据类型	有效范围	内容	适应指令的变量 (变量名称)
_eMC_HOMING_ MODE *5	0:_mcHomeSwTurnHomeSwOff 1:_mcHomeSwTurnHomeSwOn 4:_mcHomeSwOff 5:_mcHomeSwOn 8:_mcLimitInputOff 9:_mcHomeSwTurnHomeMask 11:_mcLimitInputOnly 12:_mcHomeSwTurnHoldingTime 13:_mcNoHomeSwHoldingHomeInput 14:_mcHomePreset	指定要改写的[原点复位动作] 0: 指定为附近避让、近原点输入OFF 1: 指定为附近避让、近原点输入ON 4: 指定为近原点输入OFF 5: 指定为近原点输入ON 8: 指定为极限输入OFF 9: 指定为附近避让、原点输入屏蔽距离 11: 仅极限输入 12: 指定为附近避让、接触时间 13: 指定为无近原点输入、接触原点输入 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	设定原点复位中极限输入“ON”时的停止方法 0: 不反转/轴异常而停止(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_ CURVE *6	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的形状。 0: 直线(位移保持) 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: 逆Trapeclloid 9: 单弦 10: 复弦 11: 逆复弦 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:_mcAccDecOverErrorStop	在轴的加减速控制中, 优先向目标位置停止后, 产生减速超限。指定超过最大加减速度的动作。 0: 提高加减速速度(将合并切换为等待)*7 1: 提高加减速速度 2: 异常停止*8	-
eMC REVERSE_MODE *6	0:_mcReverseModeDecelerationStop 1:_mcReverseModeImmediateStop	指定多重启动指令时、重启指令时、中断标准定位时的反转时动作。 0: 减速停止 1: 立即停止	-
eMC COUNT_MODE *6	0:_mcCountModeLinear 1:_mcCountModeRotary	设定位置的计数模式。 0: 线性模式(有限长) 1: 旋转模式(无限长)	-

数据类型	有效范围	内容	适应指令的变量 (变量名称)
_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: mm 3: nm 4: degree 5: inch	-

- *1. Ver.1.01以上的CPU单元和Ver.1.02以上的Sysmac Studio组合时可指定。
- *2. Ver.1.10以上的CPU单元和Ver.1.12以上的Sysmac Studio组合时可指定。
- *3. 使用_mcLatestCommand时，必须在满足以下关系的条件下使用主轴、从轴。
Master(主轴)设定的运动控制系统变量的轴号 < Slave(从轴)设定的运动控制系统变量的轴号
- *4. 任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先度5)的周期。
- *5. Ver.1.03以上的CPU单元和Ver.1.04以上的Sysmac Studio组合时可指定。
- *6. Ver.1.08以上的CPU单元和Ver.1.09以上的Sysmac Studio组合时可指定。
- *7. Ver.1.10以上的CPU单元，无法将合并切换为等待。详情请参阅  “NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-363)”。
- *8. Ver.1.10以上的CPU单元，合并动作时不会异常停止。详情请参阅  “NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-363)”。

运动控制指令的输出变量

以下是运动控制指令的输出变量一览表。


输出变量	名称	数据类型	有效范围	内容
Done	完成	BOOL	TRUE, FALSE	指令执行完毕时变为TRUE。 此时，输出变量Active、Error、CommandAborted为FALSE。 指令完成时，输入变量Execute为FALSE时，Done至少在一个周期内为TRUE。 Execute为TRUE时，Done则在Execute变为FALSE之前保持TRUE状态。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。 所谓控制中，是指该指令对轴及轴组进行实际控制的时候。此时，输出变量Done、Error、CommandAborted变为FALSE。
Enabled	有效	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	无法启动指令、或执行中发生中断时变为TRUE。 对象轴或轴组发生异常时无法启动指令。同样，减速停止中也无法启动指令。 启动了其他指令、或发生本指令以外的异常时，指令中断。 此时，输出变量Done、Active、Error变为FALSE。输入变量Execute为FALSE时发生中断时，CommandAborted至少在一个周期内为TRUE。 Execute或Enable为TRUE时，CommandAborted则在Execute或Enable变为FALSE之前保持TRUE状态
Error	错误	BOOL	TRUE, FALSE	因输入变量错误或指令处理而发生异常时变为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	非法结束	BOOL	TRUE, FALSE	指令未正常执行时变为TRUE
Status	可运行	BOOL	TRUE, FALSE	进入可运行状态时变为TRUE
EndOfProfile	凸轮周期完成	BOOL	TRUE, FALSE	执行凸轮表终点后变为TRUE
Index	索引	UINT	正数或 “0”	输出凸轮数据的索引编号
StartSync	跟踪中	BOOL	TRUE, FALSE	开始为同步进行加减速动作时变为TRUE
RecordedPosition	锁定位置	LREAL	负数、正数、 “0”	输出锁定的位置 * ³
Invalid	轴间偏差超调	BOOL	TRUE, FALSE	超过轴间偏差容许值时变为TRUE
DeviatedValue	轴间偏差值	LREAL	负数、正数、 “0”	输出指定的主轴和从轴的差异量 * ³
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
InVelocity	达到目标速度	BOOL	TRUE, FALSE	达到目标速度后变为TRUE

输出变量	名称	数据类型	有效范围	内容
InSync	同步中	BOOL	TRUE, FALSE	与主轴同步时、或者从轴到达从轴同步位置时变为TRUE
InGear	齿轮比到达	BOOL	TRUE, FALSE	从轴达到目标速度时变为TRUE
InCombination	加减运算中	BOOL	TRUE, FALSE	加减运算时变为TRUE
InCam	凸轮动作中	BOOL	TRUE, FALSE	执行凸轮表起点后变为TRUE
InTorque	目标转矩到达	BOOL	TRUE, FALSE	达到目标转矩时变为TRUE
InFeed	标准传送中	BOOL	TRUE, FALSE	接受锁定输入后进行标准传送时变为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	到位	BOOL	TRUE, FALSE	所有构成轴的反馈当前位置进入目标位置的到位宽度 内时变为TRUE
InOperation * ⁵	动作中	BOOL	TRUE, FALSE	指令动作中变为TRUE。
CalcPosition * ⁵	计算位置	LREAL	负数、正 数、“0”	输出指定的时间戳的位置。* ³
ErrorParameterCode* ⁶	参数详情代码	WORD	* ⁷	有些错误代码存在附属信息。如果存在该信息，则输出 异常参数的详情代码。
ErrorNodePointIndex * ⁶	节点元素编号	UINT	* ⁷	有些错误代码存在附属信息。如果存在该信息，则输出 异常节点的元素编号。
OutputtedOffset Position * ⁸	位置偏置输出 值	LREAL	负数、正 数、“0”	输出在指令当前位置叠加的位置偏置。 Active为TRUE时更新值。 CommandAborted或Error为TRUE时，停止更新并保持 值。

*1. Error(错误)在执行MC_Reset(轴错误复位)指令、MC_GroupReset(轴组错误复位)指令、ResetMCErrror(所有错误复位)指令之前不会被清零。

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

此外，在Error为TRUE时，不启动运动控制指令。解除异常后，即使Execute为TRUE，由于不是上升沿，因此也不启动指令。Enable型运动控制指令在Enable(有效)为TRUE时启动。


*2. 与事件代码的前四位数字相同。详情请参阅  “A-1 错误代码一览(P.A-2)”。

*3. 以指令单位表示。指令单位有[mm]、[μm]、[nm]、[degree]、[inch]或[pulse]。

*4. Ver.1.01以上的CPU单元和Ver.1.02以上的Sysmac Studio组合时可使用。

*5. Ver.1.06以上的CPU单元和Ver.1.07以上的Sysmac Studio组合时可使用。

*6. Ver.1.08以上的CPU单元和Ver.1.09以上的Sysmac Studio组合时可指定。

*7. 详情请参阅  “A-1 错误代码一览(P.A-2)”。

*8. Ver.1.10以上的CPU单元和Ver.1.12以上的Sysmac Studio组合时可使用。



参考

运行模式变更后想要浏览运动控制指令的输出变量时，请分配输出参数属性为保持属性的变量。


浏览分配的输出参数，便可以查看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

以下是运动控制指令的输入输出变量一览表。

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	辅轴	_sAXIS_REF	-	指定辅轴
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型的数组变量作为凸轮表指定。*1
TriggerInput	触发输入条件	_sTRIGGER_REF	-	设定触发条件
TriggerVariable	触发变量	BOOL	TRUE, FALSE	在触发条件下指定控制器模式时，指定触发的输入变量
Target	写入对象	_sAXIS_REF 或 _sGROUP_REF	-	指定作为写入对象的轴或者轴组
SettingValue	设定值	视指定的变量类型而定	-	指定写入的值。 有效范围根据通过ParameterNumber(参数编号)指定的运动控制参数而定。*2 初始值为“0”
Axes*3	轴组构成轴	ARRAY[0.0.3] OF UINT	-	指定改写后的构成轴的轴号
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型的数组变量指定为开关的ON/OFF模式数据。 数组元素编号表示开关编号。
Outputs*5	输出信号	ARRAY[0..15] OF _sOUTPUT_REF	-	将输出信号_sOUTPUT_REF型的数组变量指定为基于开关的ON/OFF模式数据计算出的数字输出的ON/OFF时刻的输出对象。数组元素编号表示轨段编号。 通过将该数组变量指定为NX_AryDOutTimeStamp指令的输入输出变量，将实际的数字输出设为ON/OFF。
TrackOptions*5	轨段选项	ARRAY[0..15] OF _sTRACK_REF	-	将轨段选项结构体_sTRACK_REF型的数组变量指定为开关的动作条件。数组元素编号表示轨段编号。
CamProperty*6	凸轮属性	_sMC_CAM_PROP ERTY	-	指定凸轮属性结构体_sMC_CAM_PROPERTY型的变量。 请指定通过_sMC_CAM_PROPERTY型的用户定义变量或Sysmac Studio的凸轮编辑器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型的数组变量。 请指定通过_sMC_CAM_NODE型的用户定义变量或*7 Sysmac studio的凸轮编辑器创建的凸轮节点变量。

输入输出变量	名称	数据类型	有效范围	内容
AxisParameter ^{*6}	轴参数	_sAXIS_PARAM	-	改写时指定轴参数。 读取时，将_sAXIS_PARAM型的变量指定为已读取的轴参数的保存位置。

- *1. 数组元素[N]通过Sysmac Studio自动设定。请指定通过Sysmac Studio的凸轮编辑器创建的凸轮数据变量。
- *2. 关于变量的数据类型，请参阅  “参数的数据类型和有效范围(P.5-14)”。
- *3. Ver.1.01以上的CPU单元和Ver.1.02以上的Sysmac Studio组合时可指定。
- *4. Ver.1.03以上的CPU单元和Ver.1.04以上的Sysmac Studio组合时可指定。
- *5. Ver.1.06以上的CPU单元和Ver.1.07以上的Sysmac Studio组合时可指定。
- *6. Ver.1.08以上的CPU单元和Ver.1.09以上的Sysmac Studio组合时可指定。
- *7. 创建用户定义变量时，请确保数组变量的元素开始编号为0，数组元素[N]为358以下。

2-2 指令一览

运动控制指令分为以下3种。

种类	概要
通用指令	MC功能模块通用指令
轴指令	MC功能模块执行单轴控制的指令
轴组指令	MC功能模块执行多轴协调控制的指令

通用指令的详情参阅“第5章 通用指令”，轴指令的详情参阅“第3章 轴指令”，轴组的详情参阅“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，部分指令的功能受限或无法使用。详情请参阅 □□ “NX系列 位置接口单元 用户手册(SBCE-374)”。

通用指令

下面对MC功能模块通用指令进行说明。

指令种类栏中的“管理”指非动作类指令，“动作”指动作类指令。

P : PLCopen®技术规格定义的指令

O : MC功能模块独有的指令

指令	指令名称	概要	指令种类		单轴*1
MC_SetCamTableProperty	凸轮表属性更新	更新通过输入参数指定的凸轮表的终点索引	管理	0	
MC_SaveCamTable	保存凸轮表	将通过输入参数指定的凸轮表保存到非易失性存储器	管理	0	
MC_Write	写入MC设定	改写一部分运动控制参数设定	管理	0	○
MC_GenerateCamTable *2	凸轮表生成	基于通过输入输出参数指定的凸轮属性和凸轮节点生成凸轮表	管理	0	
MC_WriteAxisParameter *2	轴参数写入	改写运动控制参数设定中的轴参数设定	管理	0	○
MC_ReadAxisParameter *2	轴参数读取	读取运动控制参数设定中的轴参数设定	管理	0	○

*1 单轴位置控制轴可使用的指令处加有○标记。

*2 Ver.1.08以上的CPU单元和Ver.1.09以上的Sysmac Studio组合时可使用。

关于通过指令启动进行轴状态的切换，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”。

轴指令

下面对MC功能模块执行单轴控制的指令进行说明。

指令种类栏中的“管理”指非动作类指令，“动作”指动作类指令。

P：PLCopen®技术规格定义的指令

O：MC功能模块独有的指令

指令	指令名称	概要	指令种类	单轴*1
MC_Power	可运行	将驱动器切换为可运行状态	管理 P	○
MC_MoveJog	微动移动	根据指定目标速度执行微动移动。	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*2	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	进行绝对值定位或相对值定位。	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	指定自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”设为目标位置进行定位，返回原点。	动作 O	○
MC_MoveFeed	中断标准定位	指定自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	使轴减速停止	动作 P	○
MC_ImmediateStop	立即停止	与轴的状态无关，根据输入变量“StopMode(停止方法选择)”的停止方法执行停止。	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	超调值设定	变更轴的目标速度。	管理 P	○
MC_ResetFollowingError	偏差计数器复位	对指令位置和反馈位置的偏差进行复位	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	结束通过输入参数指定的轴的凸轮动作	动作 P	
MC_GearIn	齿轮动作开始	设定主轴和从轴间的齿轮比，进行齿轮动作	动作 P	
MC_GearInPos	位置指定齿轮动作	设定主轴和从轴间的齿轮比，进行电子齿轮动作。指定开始同步的主轴及从轴的位置	动作 P	
MC_GearOut	齿轮动作解除	中止执行中的MC_GearIn、MC_GearInPos	动作 P	
MC_MoveLink	梯形模式凸轮	与指定的主轴同步，执行定位	动作 O	
MC_CombineAxes	加减运算定位	将2个轴的指令位置相加、或相减的值作为指令位置输出	动作 O	
MC_Phasing	主轴相对值相位补偿	执行同步控制中主轴的相位补偿	动作 P	
MC_TorqueControl	转矩控制	利用伺服驱动器的转矩控制模式进行转矩控制	动作 P	
MC_SetTorqueLimit	转矩限制	使用伺服驱动器的转矩限制功能，对伺服驱动器的输出转矩进行限制	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在指定范围(区域)内	管理 O	○
MC_TouchProbe	启用外部锁定	根据触发信号的发生记录轴的位置	管理 P	○
MC_AbortTrigger	不启用外部锁定	使执行中的锁定无效	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过了容许值	管理 O	○
MC_SyncMoveVelocity	周期同步速度控制	按照任务周期*3，以周期同步速度模式(CSV)将目标速度设定值输出到伺服驱动器。	动作 O	
MC_SyncMoveAbsolute*2	周期同步绝对定位	按周期输出轴的指定目标位置	动作 O	○
MC_Reset	轴错误复位	解除轴的异常。	管理 P	○
MC_ChangeAxisUse*4	轴使用变更	暂时切换轴参数的[轴使用]。	管理 P	○
MC_DigitalCamSwitch*5	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	○

指令	指令名称	概要	指令种类	单轴 ^{*1}
MC_TimeStampToPos ^{*5}	时间戳→轴位置计算	计算指定的时间戳的轴位置。	管理	0 ○
MC_PeriodicSyncVariables ^{*6}	轴变量任务间固定周期同步	在固定周期内执行轴变量的任务间同步。	管理	0
MC_SyncOffsetPosition ^{*7}	周期性同步位置偏置补偿	执行同步控制中的从轴的位置补偿。	动作	0

*1 单轴位置控制轴可使用的指令处加有○标记。

*2 Ver.1.03以上的CPU单元和Ver.1.04以上的Sysmac Studio组合时可使用。

*3 任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先度5)的周期。

*4 Ver.1.04以上的CPU单元和Ver.1.05以上的Sysmac Studio组合时可使用。

*5 Ver.1.06以上的CPU单元和Ver.1.07以上的Sysmac Studio组合时可使用。

*6 NX1P2 CPU单元及NJ系列CPU单元无法使用。

*7 Ver.1.10以上的CPU单元和Ver.1.12以上的Sysmac Studio组合时可使用。

关于通过指令启动进行轴状态的切换，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-363)”。

关于PLCopen[®]技术规格的依据内容，请参阅依据列表。

依据列表可从PLCopen[®]的Web网站上下载。

轴组指令

下面对MC功能模块执行多轴协调控制的指令进行说明。

指令种类栏中的“组管理”指非动作类指令，“组动作”指动作类指令。

P : PLCopen[®]技术规格定义的指令

O : MC功能模块独有的指令

指令	指令名称	概要	指令种类	
MC_GroupEnable	启用轴组	使轴组有效	组管理	P
MC_GroupDisable	不启用轴组	使轴组无效	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	进行两轴的圆弧插补。	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	插补动作中的所有轴按照通过轴参数选择的方法立即停止	组动作	O
MC_GroupSetOverride	轴组超调值设定	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	获取轴组的指令当前位置和反馈当前位置	组管理	O
MC_ChangeAxesInGroup ^{*1}	写入轴组构成轴	暂时改写轴组参数的[构成轴]。	组管理	O
MC_GroupSyncMoveAbsolute ^{*1}	轴组周期同步绝对位置控制	按周期输出各轴的指定目标位置	组动作	O
MC_GroupReset	轴组错误复位	解除轴组及轴的异常。	组管理	P

*1. Ver.1.01以上的CPU单元和Ver.1.02以上的Sysmac Studio组合时可使用。

关于通过指令启动进行轴组状态的切换，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-363)”。

关于PLCopen[®]技术规格的依据内容，请参阅依据列表。

依据列表可从PLCopen[®]的Web网站上下载。

2-3 PDO映射

要使用运动控制的各功能，必须将使用功能所需的对象映射到过程数据通信中。
预先将必要对象汇集在一起，构成PDO映射。

使用欧姆龙伺服驱动器1S系列R88D-1SN□□□-ECT及G5系列R88D-KN□□□-ECT(Ver.2.1以上)、R88D-KN□□□-ECT-L(Ver.1.1以上)时，PDO映射可直接使用Sysmac Studio默认的映射。

RxPDO [261th接收 PDO映射] (1704Hex)	控制字(6040Hex)、目标位置(607AHex)、目标速度(60FFHex)、目标转矩(6071Hex)、 操作模式(6060Hex)、锁定功能(60B8Hex)、最大曲线速度(607FHex)、 正方向转矩限制值(60E0Hex)、负方向转矩限制值(60E1Hex)
TxPDO [259th发送 PDO映射] (1B02Hex)	错误代码(603FHex)、状态字(6041Hex)、当前位置(6064Hex)、当前转矩(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

在欧姆龙制伺服驱动器G5系列R88D-KN□□□-ECT中进行全闭环控制时，请给RxPDO选择1701 Hex或将对象的总大小设为12位以下的1600Hex(Ver.2.1以上时)。

关于PDO映射的设定方法，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”。

使用NX系列位置接口单元时，请参阅 □□ “NX系列 位置接口单元 用户手册(SBCE-374)” 中的 “I/O入口映射”。

必需对象

必需对象有对于伺服轴的对象和对于编码器轴的对象。
没有设定任何一个必需对象时，会发生必需过程数据对象未设定(错误代码：3460Hex)错误。

伺服轴

要将MC功能模块的指令用于伺服轴，必须设定以下对象。

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
	当前位置	6064Hex

版本相关信息

- Ver.1.09以下版本的CPU单元，如未使用欧姆龙制伺服驱动器1S系列EtherCAT通信内置型及G5系列EtherCAT通信内置型，需映射操作模式(6060Hex)及操作模式显示(6061Hex)。
- Ver.1.10以上版本的CPU单元，根据有无操作模式(6060Hex)及操作模式显示(6061Hex)映射，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"> • 使用CSP*¹、CSV*²、CST*³的指令可执行。 • CSP、CSV、CST以外，将会伺服OFF。 	<ul style="list-style-type: none"> • 使用CSP的指令可执行。执行使用其他控制模式的指令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。向伺服驱动器发送CSP指令。
无操作模式(6060Hex)	<ul style="list-style-type: none"> • 使用CSP的指令可执行。执行使用其他控制模式的指令时发生过程数据对象设定不足(错误代码：3461Hex)。 • CSP以外，将会伺服OFF。 	<ul style="list-style-type: none"> • 使用CSP的指令可执行。执行使用其他控制模式的指令时发生过程数据对象设定不足(错误代码：3461Hex)。

*1. 伺服驱动器的控制模式中的周期性同步位置模式(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性同步转矩模式(CST)。

编码器轴

要将MC功能模块的指令用于编码器轴，必须设定以下对象。

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

不同的指令有各自需要设定的对象。

必须对伺服轴、编码器轴分别进行设定。

各指令没有设定必要的对象时，会发生过程数据对象设定不足(错误代码：3461Hex)错误。

伺服轴

作为伺服轴使用时，各指令有必须设定的对象。

请参考下述一览表，设定必要的对象。

下述一览表中未列举的指令只需设定必需对象，就能执行伺服轴的动作。

● 输出的设定

指令名称	功能名称						
	目标速度 60FFHex	目标转矩 6071Hex	操作模式 6060Hex ^{*1}	锁定功能 60B8Hex	最大曲线速度 607FHex	正方向转矩 限制值 60E0Hex	负方向转矩 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)。只设定其中一个将无法正常动作。

*2. 除了原点复位动作模式“11”、“12”、“14”，都需要设定。

*3. Mode选择[驱动器模式]时。

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

*5. 只在使用欧姆龙制伺服驱动器1S系列EtherCAT通信内置型或G5系列EtherCAT通信内置型时需要设定。

● 输入的设置

指令名称	功能名称				
	当前转矩 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)。只设定其中一个将无法正常动作。

*2. 除了原点复位动作模式“11”、“12”、“14”, 都需要设定。

*3. Mode选择[驱动器模式]时。

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

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

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

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

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

编码器轴

作为编码器轴使用时, 各指令有必须设定的对象。

请参考下述一览表, 设定必要的对象。

下述一览表中未列举的指令只需设定必需对象, 就能执行编码器轴的动作。

● 输出的设定

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

*1. Mode选择[驱动器模式]时。

*2. 使用欧姆龙GX系列编码器从站, Mode选择[驱动器模式]时。

● 输入的设置

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[驱动器模式]时。

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

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

*4. 使用欧姆龙GX系列编码器从站, Mode选择[驱动器模式]时。

3

轴指令

下面对MC功能模块执行单轴控制的指令进行说明。

MC_Power	3-3
MC_MoveJog	3-8
MC_Home	3-15
MC_HomeWithParameter	3-37
MC_Move	3-42
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	设为TRUE, 则进入可运行状态; 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	进入可运行状态时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为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为正常。

*1. □ 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Status	指定轴进入可运行状态时	<ul style="list-style-type: none"> 指定轴的可运行状态已解除时 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含有异常因素时	异常已解除时

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. 请使用在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时, Axis(轴)指定的轴进入可运行状态。
将轴设为可运行状态, 可实现轴控制。
- 将Enable(有效)设为FALSE, 可解除Axis(轴)指定轴的可运行状态。
解除可运行状态后, 轴不接收动作指令, 无法实现轴控制。并且, 解除可运行状态的轴相应的动作指令表现为异常。但是, 即使是解除状态也可以执行MC_Power(可运行)指令、MC_Reset(轴错误复位)指令。
- 也可将动作中的轴切换为解除状态。此时, 中断执行(CommandAborted)动作中的指令, 停止动作指令的输出, 切换为解除状态。
- 使用带绝对值编码器的伺服电机的情况下, 在轴切换为可运行状态时, 如果处于原点未确定状态, 则执行[绝对值编码器原点位置偏置]的补偿, 即可变为原点确定状态。
对于Ver.1.10以上版本的CPU单元, 在EtherCAT过程数据通信从未确立状态切换至确立状态时, 也会变为原点确定状态。
关于[绝对值编码器原点位置偏置], 请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。



使用注意事项

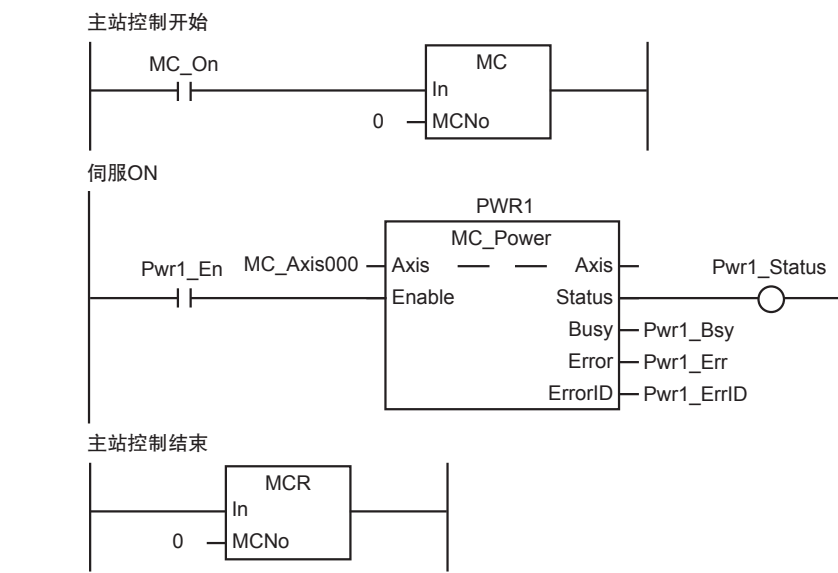
- 本指令适用于轴种类为伺服轴或虚拟伺服轴。用于编码器轴、或虚拟编码器轴时, 表现为异常。
- 对NX系列脉冲输出单元执行本指令时, 本指令的功能存在差别。详情请参阅 □ “NX系列位置接口单元 用户手册(SBCE-374)”。
- 针对同步控制的主轴执行本指令时
在垂直轴等中, 解除主轴的可运行状态时, 主轴位置可能会发生急剧变化。据此, 从轴的动作也可能发生急剧变化。
请对主轴设置制动器、或者在同步控制结束后解除主轴的运行状态, 防止从轴急剧动作。



参考

Execute型的运动控制指令在与Execute输入连接触点的上升沿开始动作，持续动作直至指令控制完毕。即使处于主站控制的复位中，在执行复位前执行指令时，也会持续动作直至指令控制完毕。

要对Execute型的运动控制指令进行联锁时，请如下图所示，在主站控制区域配置MC_Power(可运行)指令。将MC_On设为FALSE，可切实执行伺服OFF。



● 与CPU单元动作模式的关系

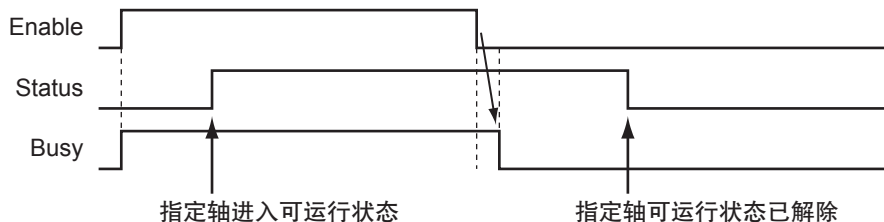
在运行模式下利用本指令进入可运行状态后，即使切换为程序模式，也会持续运行状态。

● 利用在线编辑删除指令时

利用本指令进入可运行状态后，即使在在线编辑中删除本指令，也会持续可运行状态。

时序图

- 将Enable(有效)设为TRUE时，表示正在受理指令的Busy(执行中)变为TRUE。
- 然后，在轴进入可运行状态时，Status(可运行)变为TRUE。
- 将Enable(有效)设为FALSE，Busy(执行中)变为FALSE。Status(可运行)在解除可运行状态时变为FALSE。无论Enable(有效)是TRUE还是FALSE，Status(可运行)均输出轴的可运行状态。



使用注意事项

- 将Enable(有效)设为TRUE后，只要轴侧的处理不结束，Status(可运行)就不会变为TRUE。运行轴时，请务必在确认Status(可运行)已变为TRUE后开始动作。
- 接通控制器电源后开始运行时，请在确认已建立EtherCAT通信后，编写用户程序，以启动运动控制指令。并且，请在用户程序中编入联锁功能，从而在运行中监控EtherCAT通信是否发生异常。

重启运动指令

输入为“Enable型”指令时，不会重启运动指令。

多重启动运动指令

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

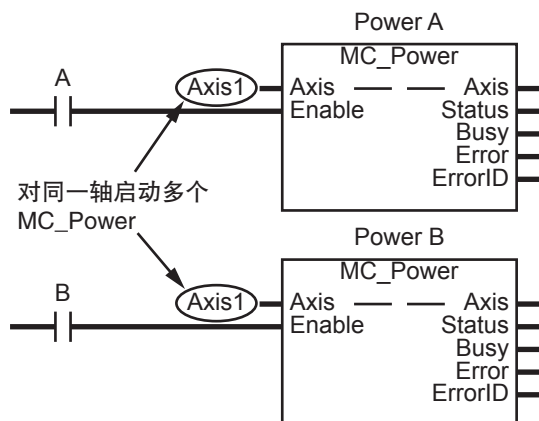
● 多重启动多个本指令

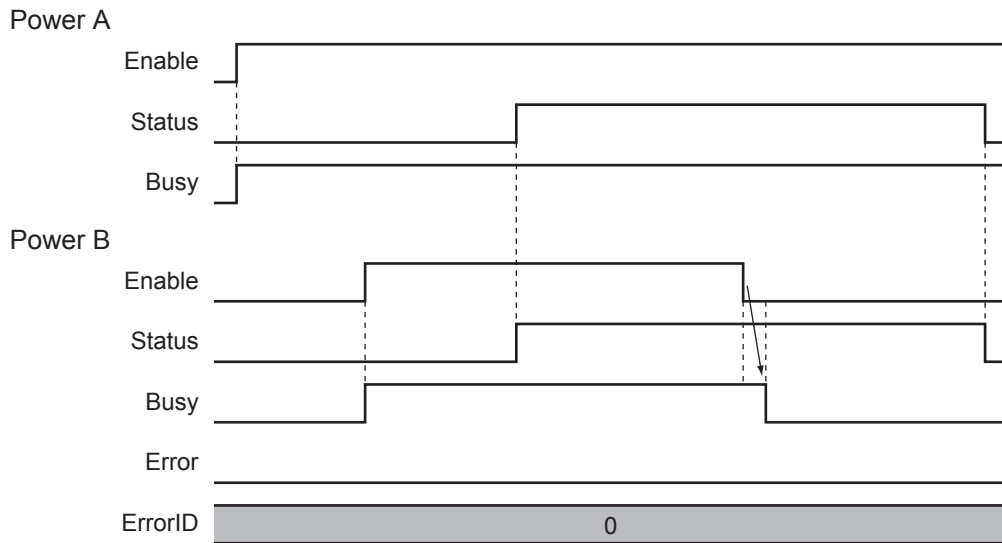


使用注意事项

请勿在正在执行MC_Power(可运行)指令的轴中，编写用于启动其它实例的MC_Power(可运行)指令的程序。原则上1根轴只能设1个MC_Power(可运行)指令。

如果在正在执行MC_Power(可运行)指令的轴中，启动其它实例的MC_Power(可运行)指令，则优先执行后执行的MC_Power(可运行)指令。





错误代码

关于指令发生的异常，请参阅 □□ “A-1 错误代码一览(P.A-2)”。

MC_MoveJog

根据指定目标速度执行微动移动。

指令	名称	FB/ FUN	图形表现	ST表现
MC_MoveJog	微动移动	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	设为TRUE, 则开始正方向移动。设为FALSE, 则结束移动。
NegativeEnable	负方向有效	BOOL	TRUE, FALSE	FALSE	设为TRUE, 则开始负方向移动。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. 关于指令单位, 请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 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	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时, 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为正常。

*1. 请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Busy	PositiveEnable、或NegativeEnable的上升沿	<ul style="list-style-type: none"> 轴已停止时 Error变为TRUE时 CommandAborted变为TRUE时
CommandAborted	<ul style="list-style-type: none"> 利用其它指令多重启动运动指令(中断), 中止本指令时 因发生异常, 中止本指令时 发生异常过程中, 启动本指令时 执行MC_Stop指令中, 启动本指令时 	<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和NegativeEnable两者均为FALSE时, 1个周期后
Error	本指令的启动条件或输入参数中含有异常因素时	异常已解除时

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. 请使用在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[*])。

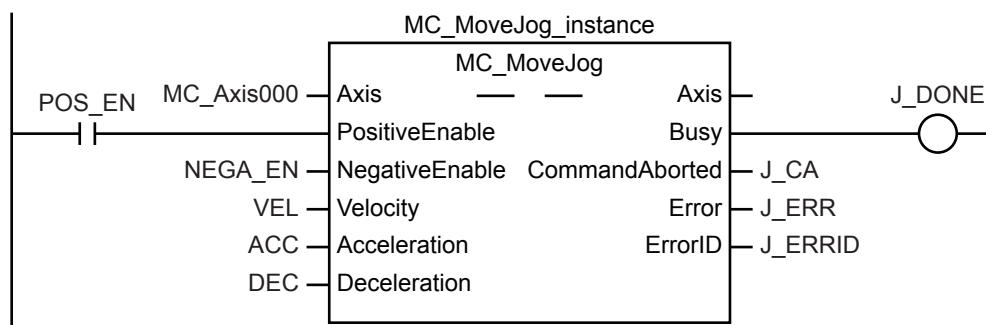
功能说明

- 根据指定Velocity(目标速度)执行微动移动。
- 需要正方向微动移动时, 将PositiveEnable(正方向有效)设为TRUE, 需要负方向微动移动时, 将NegativeEnable(负方向有效)设为TRUE。
- 同时将PositiveEnable(正方向有效)和NegativeEnable(负方向有效)设为TRUE时, PositiveEnable(正方向有效)优先, 正方向微动移动。
- MC_MoveJog(微动移动)指令的指令速度指定值超过轴参数的[微动最高速度]时, 表现为[微动最高速度]。

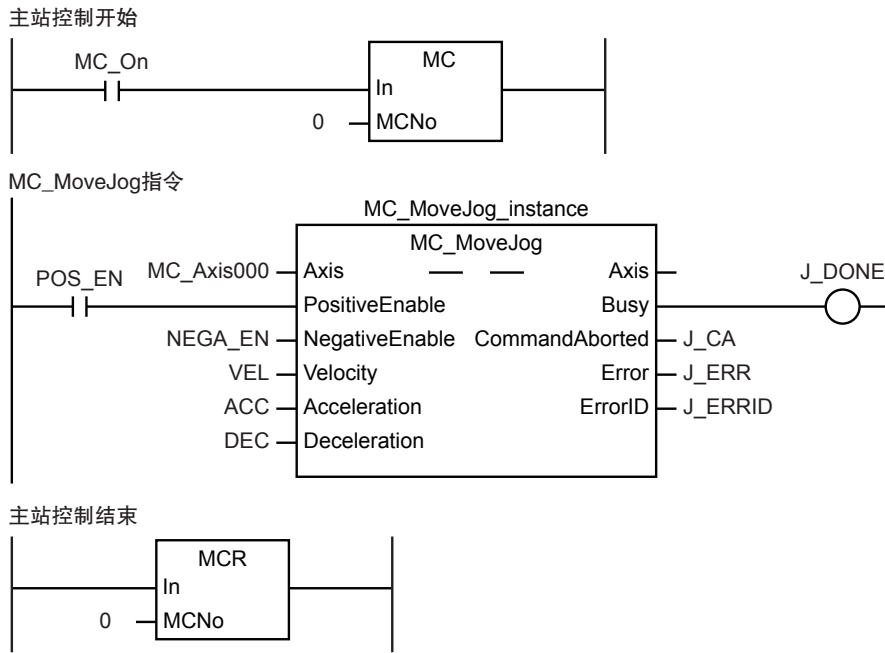


使用注意事项

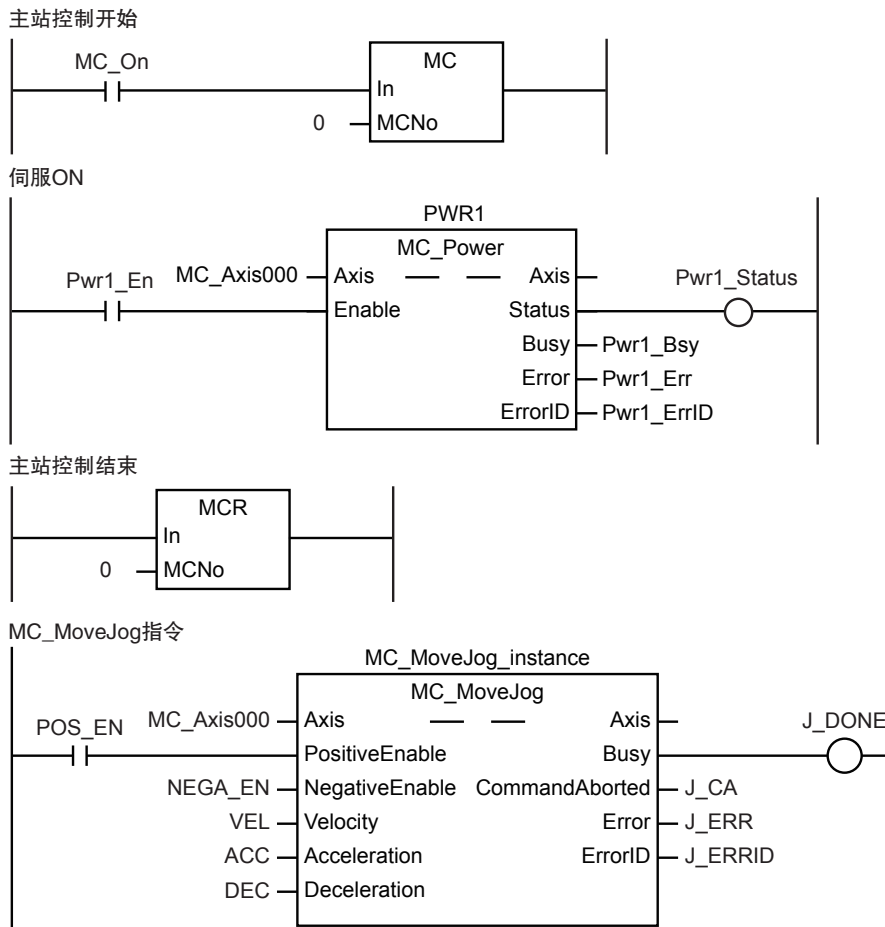
- 编写梯形图程序时, 如下所示, 输入变量“PositiveEnable(正方向有效)”与左母线连接, “NegativeEnable(负方向有效)”必须指定变量。



如果对于MC_MoveJog(微动移动)使用主站控制指令(MC指令)时, 请勿进行如下描述。否则, 主站控制仅对PositiveEnable(正方向有效)发挥作用, 对NegativeEnable(负方向有效)无效。

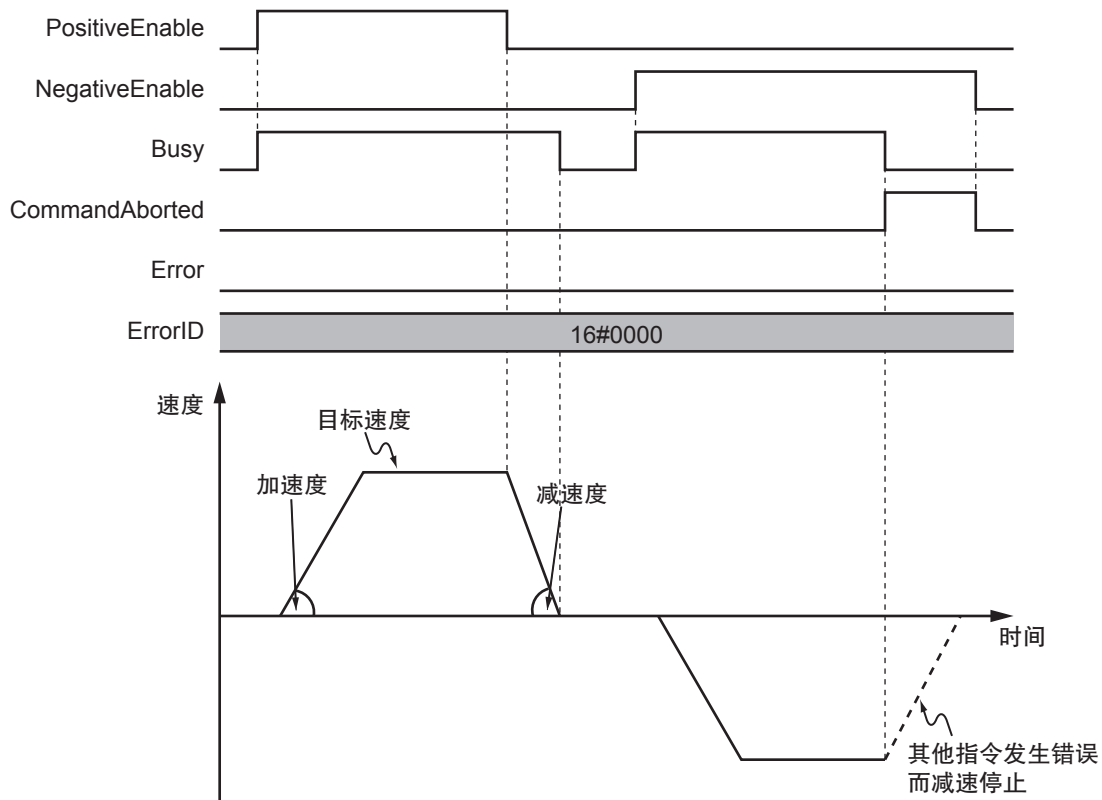


请务必对MC_Power(可运行)使用主站控制指令。



时序图

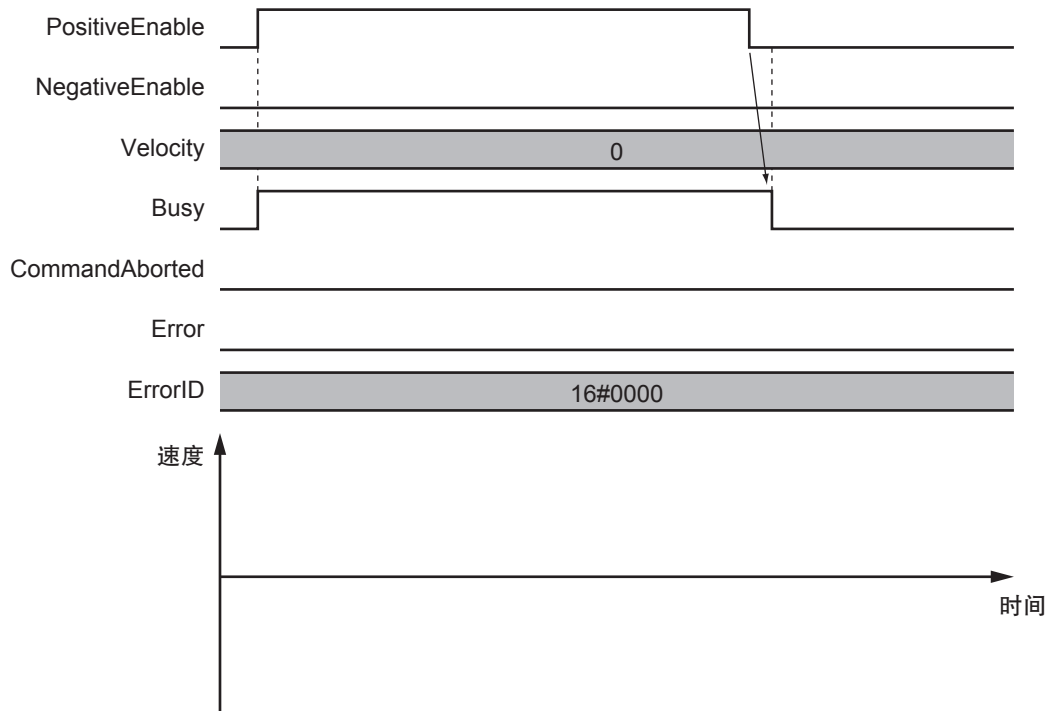
- 在启动PositiveEnable(正方向有效)或NegativeEnable(负方向有效)的同时, Busy(执行中)变为TRUE。
- 在PositiveEnable(正方向有效)或NegativeEnable(负方向有效)的下降沿开始减速并停止轴的同时, Busy(执行中)变为FALSE。
- 利用其它指令中止本指令时, CommandAborted(执行中断)变为TRUE, Busy(执行中)变为FALSE。



可在输入变量中指定Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度)。
 输入变量的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)的上升沿反映到动作中。因此,在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)也无法变更速度。

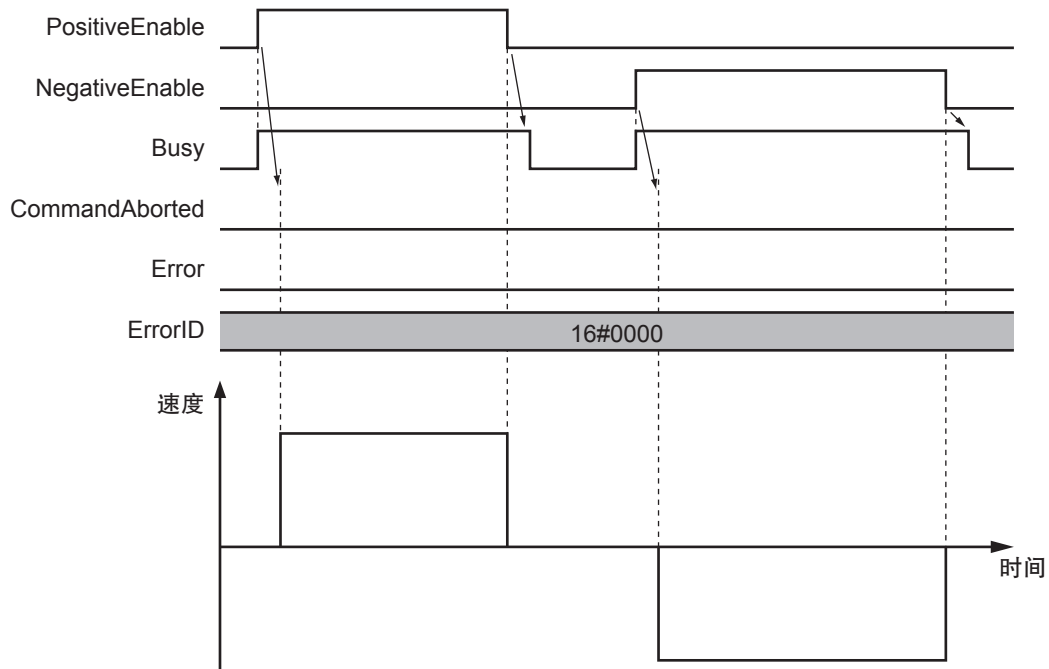
● 目标速度为“0”时的时序图

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



● 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，不作加减速而达到目标速度。Acceleration(加速度)和Deceleration(减速度)两者均为“0”时的动作示例如下所示。



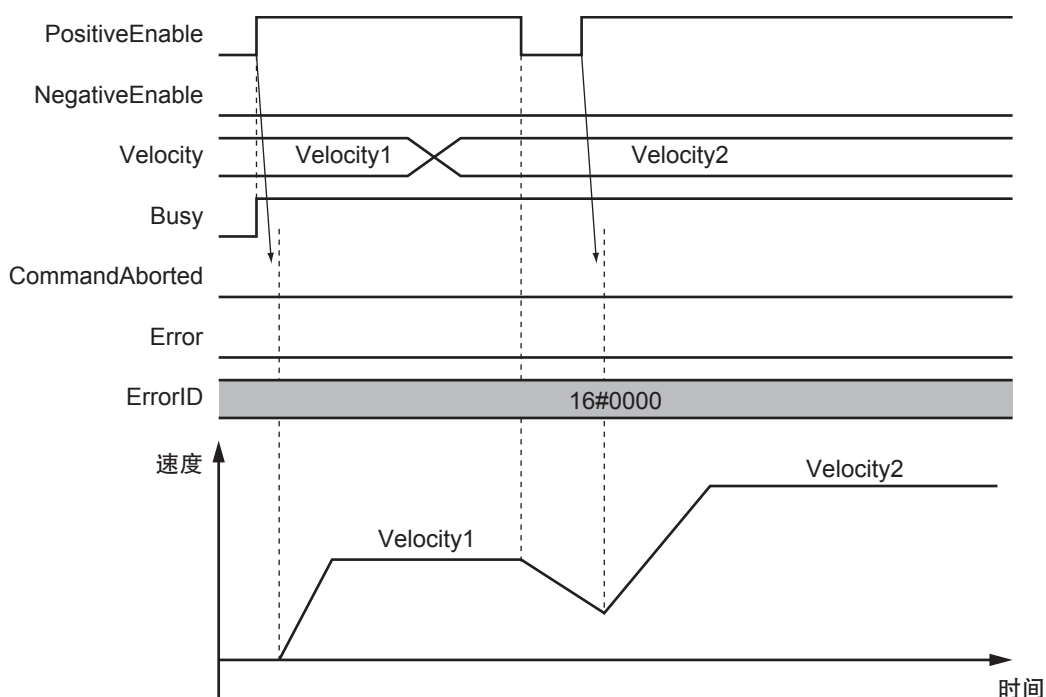
重启运动指令

● 重启同向Enable时

将PositiveEnable(正方向有效)或NegativeEnable(负方向有效)设为FALSE并减速的过程中,再次将PositiveEnable(正方向有效)或NegativeEnable(负方向有效)设为TRUE时,从减速中途进行加减速,向目标速度移动。此时,如果Velocity(目标速度)、Acceleration(加速度)或Deceleration(减速度)已变更,则以新的输入变量进行动作。

并且,由于轴并未停止,因此Busy(执行中)不会变为FALSE。

在减速中将PositiveEnable(正方向有效)设为TRUE时的动作示例如下所示。



● 重启不同方向Enable时

在将PositiveEnable(正方向有效)设为TRUE,正方向进行微动移动时,如果将NegativeEnable(负方向有效)设为TRUE,则方向反转,负方向进行微动移动。

此时,以将NegativeEnable(负方向有效)设为TRUE时的输入变量进行微动移动。输入变量有Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度)等。

反转时的减速度和反转后的加速度取决于将NegativeEnable(负方向有效)设为TRUE时的输入变量。与轴参数的[反转时动作]无关。

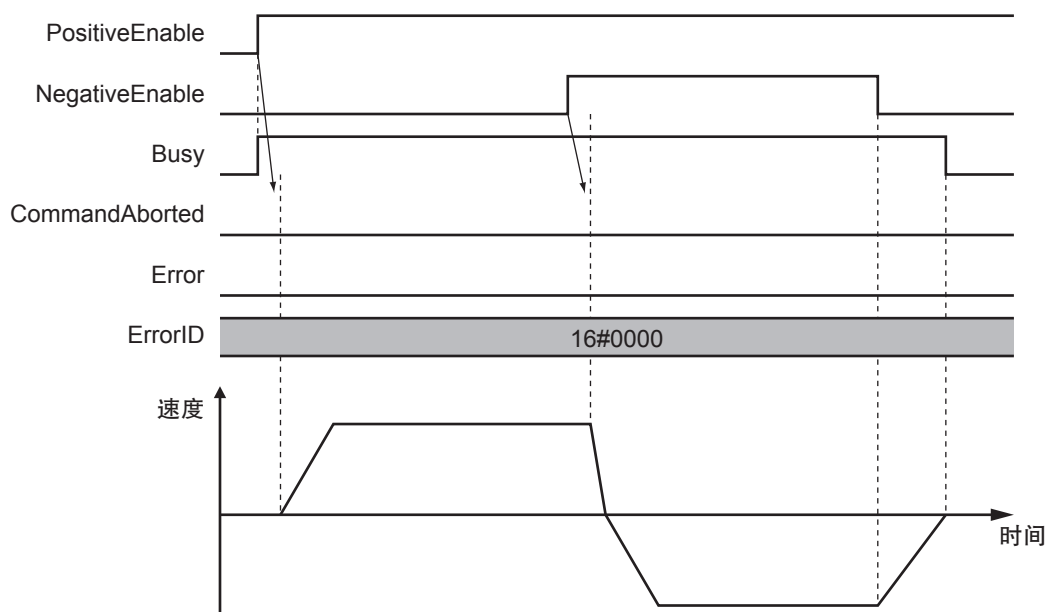
在将NegativeEnable(负方向有效)设为TRUE,负方向进行微动移动时,如果将PositiveEnable(正方向有效)设为TRUE,也会发生同样动作。

在将PositiveEnable(正方向有效)设为TRUE,之后将NegativeEnable(负方向有效)设为TRUE负方向进行微动移动时,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,也不会正方向进行微动移动。

需要正方向进行微动移动时,请先将PositiveEnable(正方向有效)设为FALSE,然后再次设为TRUE。

相反情况下也会变为同样的动作。

将PositiveEnable(正方向有效)设为TRUE后,将NegativeEnable(负方向有效)设为TRUE时的动作示例如下所示。



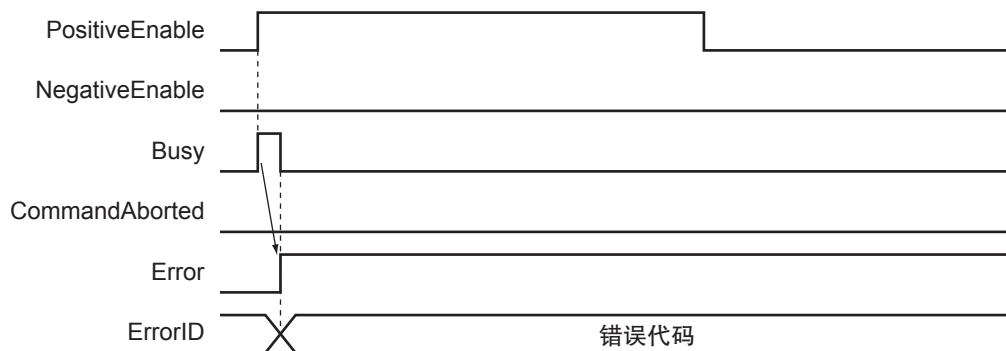
多重启动运动指令

多重启动运动指令的详情,请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

异常

在执行本指令中发生异常时, Error(错误)变为TRUE, 轴停止动作。
可查看ErrorID(错误代码)的输出值, 了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常, 请参阅 □□ “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	<p>The diagram shows a block named 'MC_Home_instance' containing a sub-block 'MC_Home'. The 'MC_Home' block has two input lines on the left labeled 'Axis' and 'Execute', and five output lines on the right labeled 'Done', 'Busy', 'CommandAborted', 'Error', and 'ErrorID'.</p>	<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	在上升沿开始指令。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

*1. □ 请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 执行MC_Stop指令中，启动本指令时 	<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含有异常因素时	异常已解除时

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. 请在使用在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

- 对于Axis(轴)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由轴参数设定。
- 原点复位指令中有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]进行设定。



使用注意事项

有关对于主轴的注意事项，请参阅 □ “同步控制的主轴及辅轴的注意事项(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的轴基本设置的[高级设置]中映射如下对象数据。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)

未进行这些设定时，将发生过程数据对象设定不足(错误代码：3461Hex)错误。

关于数据对象的映射，请参阅 □ “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-363)”或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。使用NX系列位置接口单元时，请参阅 □ “NX系列 位置接口单元 用户手册(SBCE-374)”中的“I/O入口映射”。

欧姆龙制伺服驱动器1S系列的设定

关于在原点复位指令中使用的近原点信号等输入信号，请在欧姆龙制伺服驱动器1S系列中设定。

关于输入信号的设定方法，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”的“与伺服驱动器的连接”及 □ “AC伺服电机/驱动器1S系列EtherCAT通信内置型用户手册(SBCE-377)”的“通用输入信号”。

欧姆龙制伺服驱动器 G5系列的设定

关于在原点复位指令中使用的近原点信号等输入信号，请在欧姆龙制伺服驱动器G5系列中设定。

关于输入信号的设定方法，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”的“与伺服驱动器的连接”及 □ “AC伺服电机/驱动器G5系列EtherCAT通信内置型用户手册(SBCE-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-366)”的“时序输入输出信号”。

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，请参阅 □ “NX系列 位置接口单元 用户手册(SBCE-374)”。

原点复位动作模式

作为确定机械原点的动作，可以从10种动作中选择。

- 指定为附近避让、近原点输入OFF
- 指定为附近避让、近原点输入ON
- 指定为近原点输入OFF
- 指定为近原点输入ON
- 指定为极限输入OFF
- 指定为附近避让、原点输入屏蔽距离
- 仅极限输入
- 指定为附近避让、接触时间
- 指定为无近原点输入、接触原点输入
- 原点预设

各原点复位动作使用的原点复位参数如下表所示。


(○：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	原点输入屏蔽距离	原点位置偏置	原点复位接触时间	原点复位补偿值	原点复位补偿速度
指定为附近避让、近原点输入OFF	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为附近避让、近原点输入ON	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为近原点输入OFF	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为近原点输入ON	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为极限输入OFF	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为附近避让、原点输入屏蔽距离	○	○	○	○	○	○	○	○	○	○	○	○	×	○	○
仅极限输入	×	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为附近避让、接触时间	×	○	○	○	○	○	○	○	○	○	×	○	○	○	○
指定为无近原点输入、接触原点输入	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
原点预设	×	×	×	×	×	×	×	×	×	×	×	○	×	×	×

关于各原点复位模式的动作详情，请参阅  “原点确定的动作(P.3-22)”。

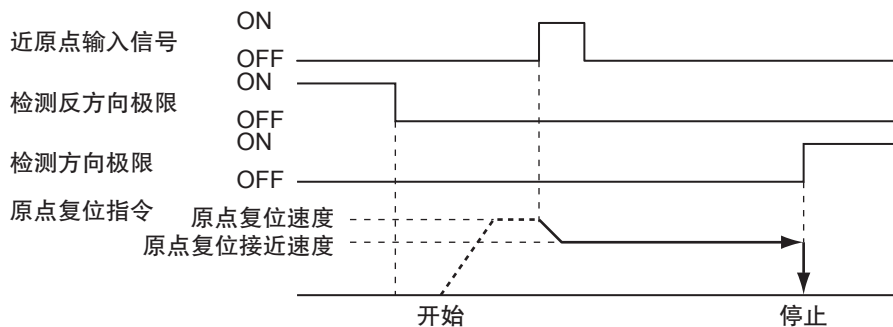


参考

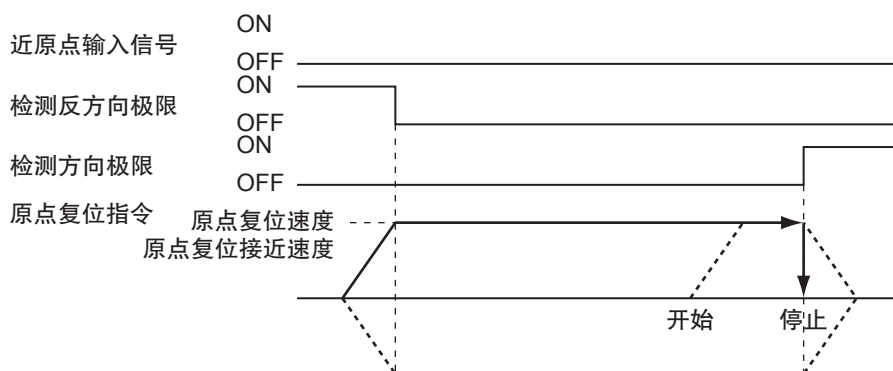
使用NX系列位置接口单元时，无法选择使用接触的原点复位动作模式。详情请参阅  “NX系列位置接口单元用户手册(SBCE-374)”。

正方向极限输入时动作和负方向极限输入时动作

- 在 origin 复位过程中到达动作方向的极限输入时，选择反转后继续 origin 复位，或不反转而紧急停止。反转时，也要选择其停止方法。
- 设定了[反转]时，如果以 origin 复位接近速度将 origin 检测方向的极限信号设为“ON”，将发生异常停止。但是，在不使用附近信号的 origin 复位动作模式[13：指定为无近原点输入、接触原点输入]下，不会发生异常停止。



- 此外，将两个方向的极限输入时动作设定为[反转]时，在 origin 检测方向上从相反侧极限输入到另一侧极限输入进行动作，无法检测到原点时，将发生异常停止。



原点复位开始方向

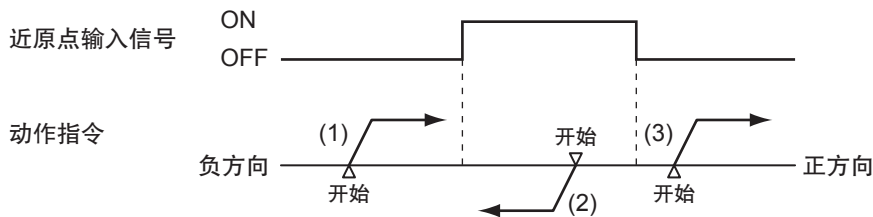
选择启动 origin 复位时开始动作的方向。

但是，在包含基于附近避让的反转动作的 origin 复位动作模式下，在近原点信号“ON”时启动 origin 复位后，沿 origin 检测方向的反方向开始动作。

包含基于附近避让的反转动作的 origin 复位动作模式有如下4种。

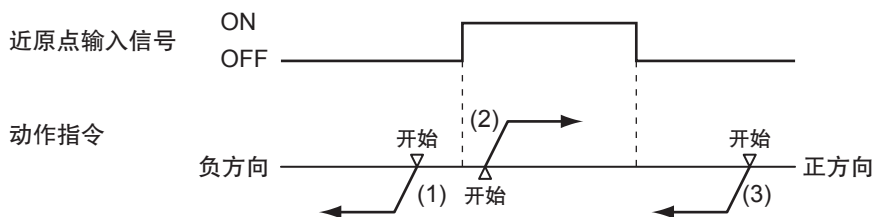
- 0：指定为附近避让、近原点输入OFF
- 1：指定为附近避让、近原点输入ON
- 9：指定为附近避让、原点输入屏蔽距离
- 12：指定为附近避让、接触时间

原点复位开始方向：正
原点检测方向：正



- (1)、(3) : 从近原点信号“OFF”状态启动，因此在原点复位开始方向上开始动作
- (2) : 从近原点信号“ON”状态启动，因此在原点检测方向的反方向上开始动作

原点复位开始方向：负
原点检测方向：负



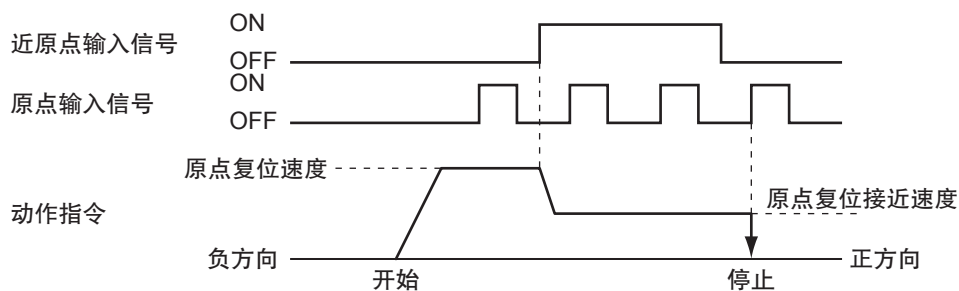
- (1)、(3) : 从近原点信号“OFF”状态启动，因此在原点复位开始方向上开始动作
- (2) : 从近原点信号“ON”状态启动，因此在原点检测方向的反方向上开始动作

原点检测方向

选择原点输入检测方向。

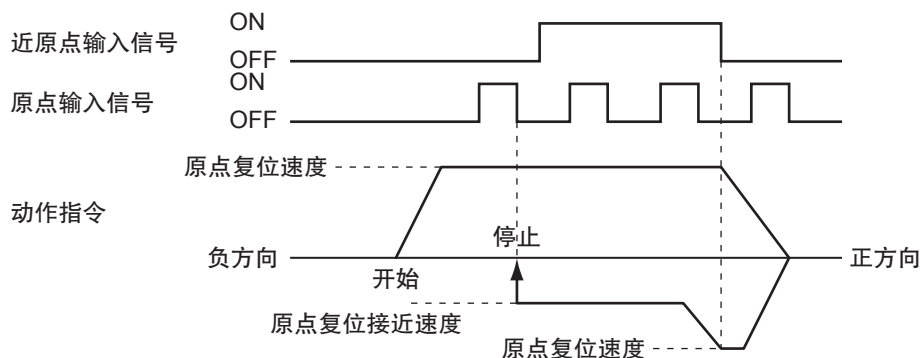
原点检测方向为FALSE(正方向)时的动作如下所示。

动作示例：指定为附近避让、近原点输入OFF



此外，原点检测方向为TRUE(负方向)时的动作如下所示。

动作示例：指定为附近避让、近原点输入OFF



原点输入屏蔽距离

对原点复位动作模式设定为[9：指定为附近避让、原点输入屏蔽距离]时的标准距离进行设定。
动作详情请参阅 □ “9：指定为附近避让、原点输入屏蔽距离的动作(P.3-25)”。

原点复位补偿值

设定原点检测后的原点复位补偿量。
通过[原点复位补偿速度]设定其时的移动速度。
动作详情请参阅 □ “原点补偿(P.3-31)”。

原点位置偏置

预设原点复位动作完成时的当前位置。
[原点位置偏置]的设定值即为当前位置。

原点复位速度

原点复位时的高速速度。

原点复位接近速度

原点复位时的附近速度。

原点复位补偿速度

设定[原点复位补偿值]时的移动速度。
动作详情请参阅 □ “原点补偿(P.3-31)”。

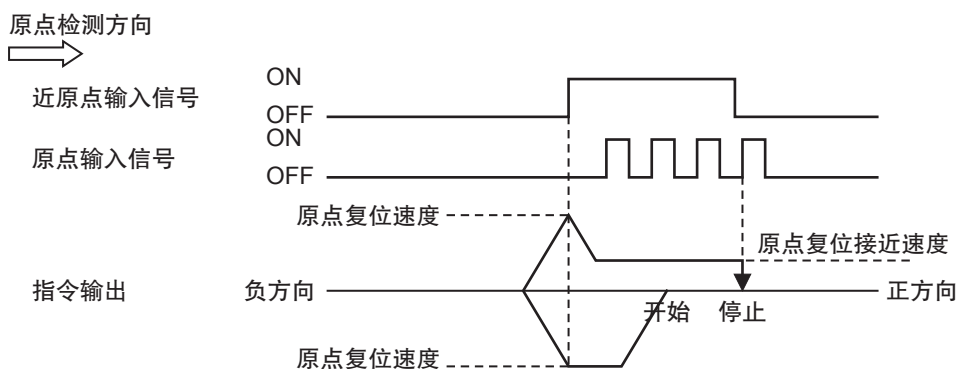
指令详情

● 原点确定的动作

下面对10种的原点复位动作模式进行说明。

0: 指定为附近避让、近原点输入OFF的动作

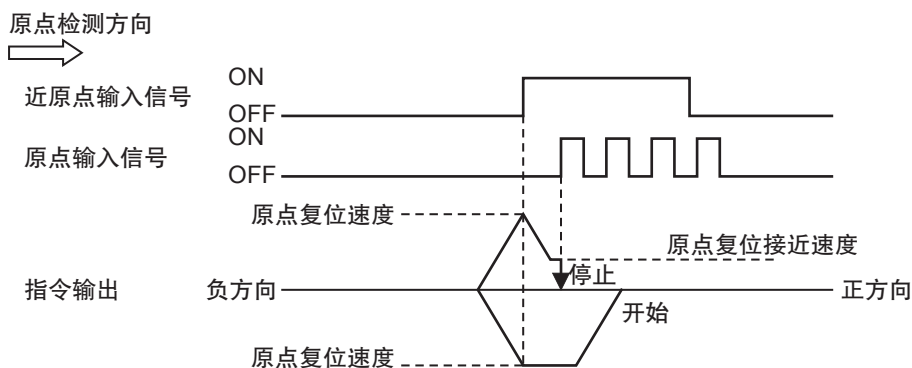
- 1** 以原点复位速度启动，执行近原点输入信号的“OFF” → “ON”，开始减速为原点复位接近速度。
- 2** 达到原点复位接近速度，在近原点输入信号由“ON” → “OFF”后的最初的原点输入时停止动作，确定机械原点。



- 如果在近原点输入信号“ON”的状态下执行原点复位，则在原点检测方向的反方向上以原点复位速度开始动作。在近原点输入信号的“ON”状态下避让后，以原点复位速度，在原点检测方向上开始原点复位动作。
- 开始原点复位后，在低于原点复位接近速度的状态下，近原点输入信号由“OFF” → “ON” → “OFF”后，在原点输入信号“ON”后也可确定机械原点。

1: 指定为附近避让、近原点输入ON的动作

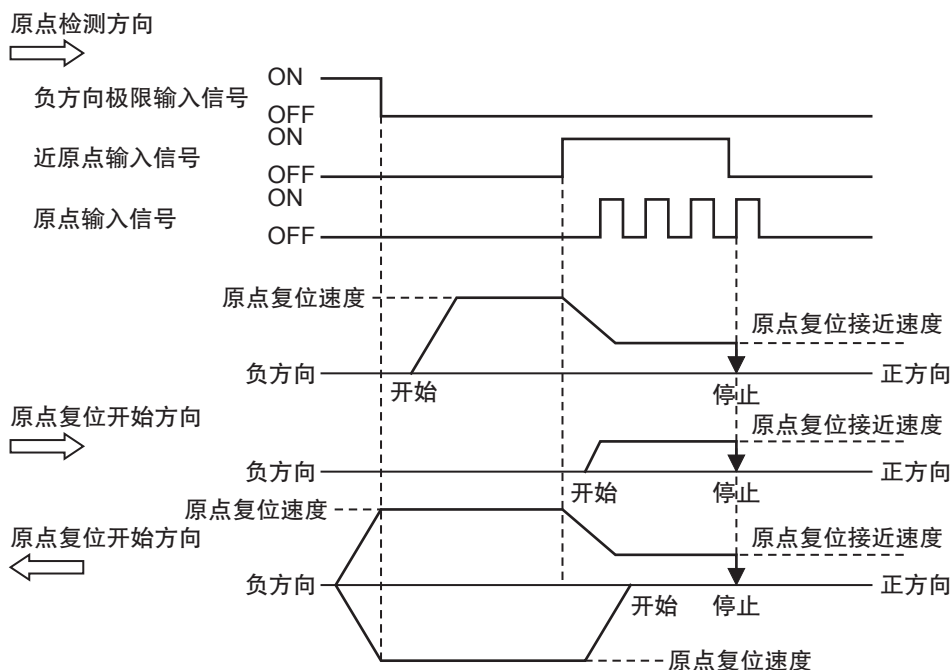
- 1** 以原点复位速度启动，执行近原点输入信号的“OFF” → “ON”，开始减速为原点复位接近速度。
- 2** 在达到原点复位接近速度后的最初的原点输入时停止动作，确定机械原点。
“ON”后近原点输入信号的状态不影响动作。



- 如果在近原点输入信号“ON”的状态下执行原点复位，则在原点检测方向的反方向上以原点复位速度开始动作。在近原点输入信号的“ON”状态下避让后，以原点复位速度，在原点检测方向上开始原点复位动作。
- 开始原点复位后，在低于原点复位接近速度的状态下，近原点输入信号“ON”后，在原点输入信号“ON”后也可确定机械原点。

4: 指定为近原点输入OFF的动作

- 1 以原点复位速度启动，执行近原点输入信号的“OFF” → “ON”，开始减速为原点复位接近速度。
- 2 达到原点复位接近速度，在近原点输入信号由“ON” → “OFF”后的最初的原点输入时停止动作，确定机械原点。



- 如果在近原点输入信号“ON”的状态下执行原点复位，则根据原点复位开始方向的设定，产生如下动作。

原点复位开始方向和原点检测方向相同时

不执行避让动作，以原点复位接近速度，在原点检测方向上开始原点复位动作。

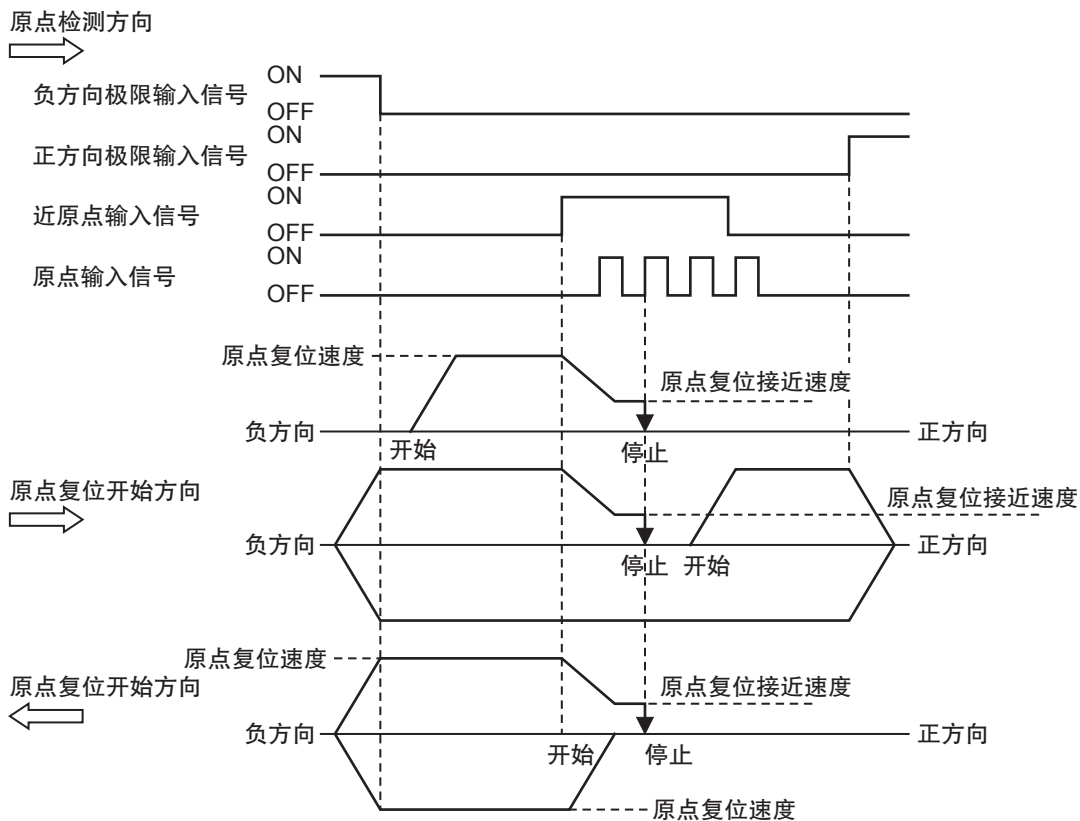
原点复位开始方向和原点检测方向不同时

无论近原点输入信号的状态如何，均以原点复位速度，在原点复位开始方向上开始动作。原点复位开始方向的极限输入时动作设定为“反转”时，在检测到极限输入时反转，在原点检测方向上执行原点复位动作。

- 开始原点复位后，在低于原点复位接近速度的状态下，近原点输入信号由“OFF” → “ON” → “OFF”后，在原点输入信号“ON”后也可确定机械原点。

5: 指定为近原点输入ON的动作

- 1** 以原点复位速度启动，执行近原点输入信号的“OFF” → “ON”，开始减速为原点复位接近速度。
- 2** 在达到原点复位接近速度后的最初的原点输入时停止动作，确定机械原点。
“ON”后近原点输入信号的状态不影响动作。



- 如果在近原点输入信号“ON”的状态下执行原点复位，则根据原点复位开始方向的设定，产生如下动作。

原点复位开始方向和原点检测方向相同时

不执行避让动作，以原点复位速度，在原点检测方向上开始原点复位动作。

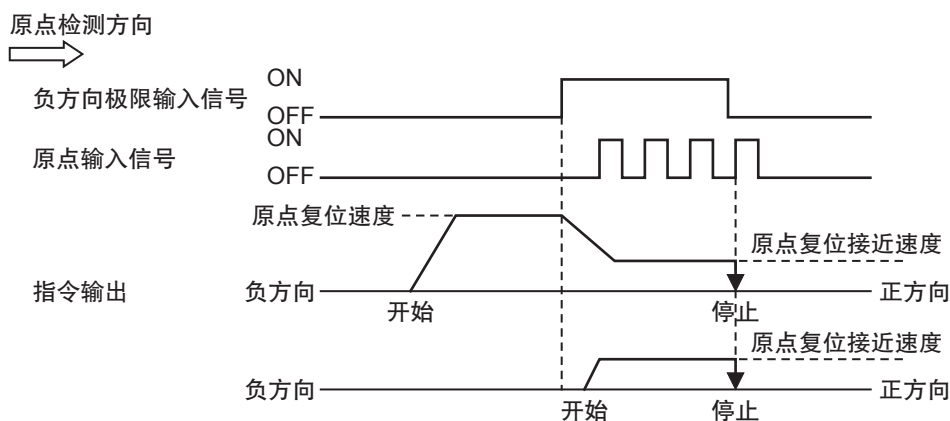
原点复位开始方向和原点检测方向不同时

无论近原点输入信号的状态如何，均以原点复位速度，在原点复位开始方向上开始动作。原点复位开始方向的极限输入时动作设定为“反转”时，在检测到极限输入时反转，在原点检测方向上执行原点复位动作。

- 开始原点复位后，在低于原点复位接近速度的状态下，近原点输入信号由“OFF” → “ON”后，在原点输入信号“ON”后也可确定机械原点。

8: 指定为极限输入OFF的动作

- 1 以原点复位速度启动，原点检测方向和相反方向的极限信号由“OFF”→“ON”，开始减速为原点复位接近速度。
- 2 达到原点复位接近速度，在极限信号由“ON”→“OFF”后的最初的原点输入时停止动作，确定机械原点。



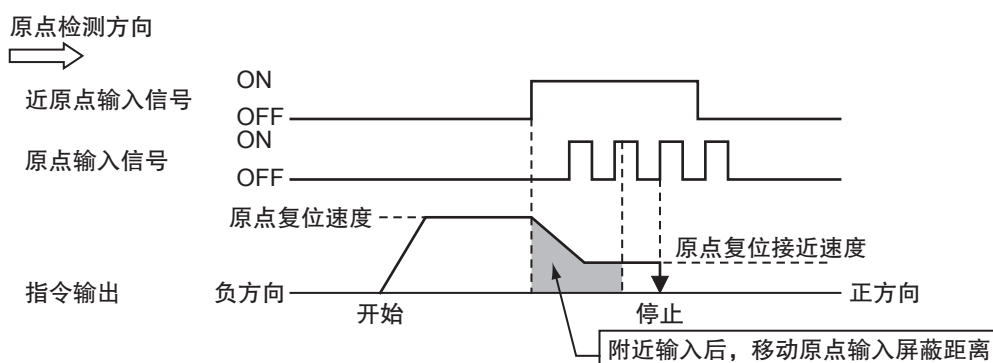
- 如果在原点检测方向反方向的极限输入“ON”的状态下执行原点复位，则以原点复位接近速度，在原点检测方向上执行原点复位动作。
- 开始原点复位后，在低于原点复位接近速度的状态下，检测方向和反方向的极限信号由“OFF”→“ON”→“OFF”后，在原点输入信号“ON”后也可确定机械原点。

9: 指定为附近避让、原点输入屏蔽距离的动作

- 1 以原点复位速度启动，执行近原点输入信号的“OFF”→“ON”，开始减速为原点复位接近速度。
- 2 近原点输入信号由“OFF”→“ON”后，在移动原点输入屏蔽距离完成后，在最初的原点输入时停止动作，确定机械原点。

“ON”后近原点输入信号的状态不影响动作。

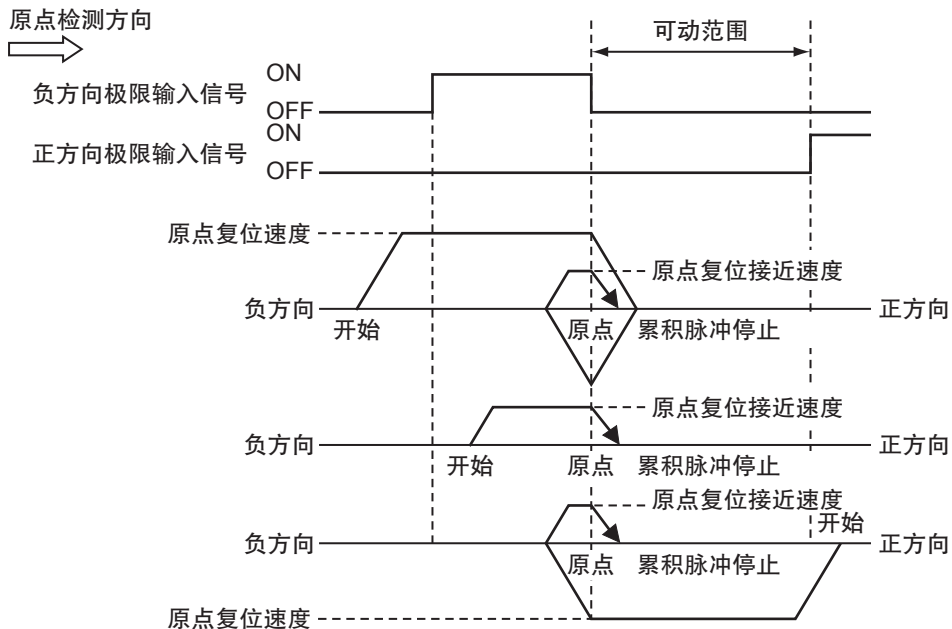
其时，设定移动量较短，在减速到原点复位接近速度之前达到移动量的情况下，启动原点复位时将发生“原点输入屏蔽距离错误(错误代码：742B Hex)”。



- 如果在近原点输入信号“ON”的状态下执行原点复位，则在原点检测方向的反方向上以原点复位速度开始动作。在近原点输入信号的“ON”状态下避让后，以原点复位接近速度，在原点检测方向上开始原点复位动作。
- 开始原点复位后，在低于原点复位接近速度的状态下，近原点输入信号“ON”后，移动原点输入屏蔽距离，在原点输入信号“ON”后也可确定机械原点。

11: 仅极限输入的动作

- 1** 以原点复位速度启动，原点检测反方向的极限信号由“ON”→“OFF”，停止减速。
- 2** 停止动作后，以原点复位速度反转，极限信号由“OFF”→“ON”，停止减速。
- 3** 停止动作后，以原点复位接近速度反转，极限信号由“ON”→“OFF”，将该位置作为原点，立即停止(累积脉冲停止)。但是，不会产生复位到机械原点位置的动作。



- 仅使用极限信号而执行原点复位时，将在原点检测方向上动作时的极限信号由“ON”→“OFF”的变化点作为原点检测位置。
- 在原点检测方向的极限信号“ON”的状态下执行原点复位时，以原点复位速度，在原点检测反方向上开始动作。原点检测反方向的极限信号由“OFF”→“ON”，停止减速。
- 检测到极限信号后，在停止减速前即使将极限信号设为“OFF”，也无法确定机械原点而继续减速。此时，不会发生异常。
- 反转后，或在极限信号输入状态下启动后，在达到原点复位接近速度之前极限信号由“ON”→“OFF”时，确定机械原点。

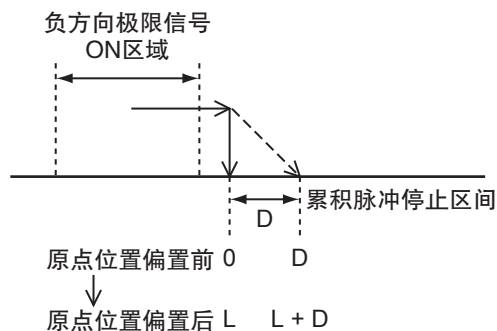
原点复位动作模式11和其它原点复位动作模式的区别

在模式11下，通过MC功能模块的处理，检测“极限信号的ON→OFF”，确定原点。因此，与将原点输入用作外部锁定信号的原点复位动作模式相比，存在如下不同。

请在理解模式11与其它原点复位动作模式的区别的基础上加以使用。

- 检测极限输入时，不使用伺服驱动器等的外部锁定功能，而是通过MC功能模块的处理进行检测。MC功能模块的处理是指原始恒定周期任务以原始周期为间隔，固定周期任务(执行优先度5)以固定周期任务(执行优先度5)的任务周期为间隔执行的处理。因此，确定原点的精度取决于原点复位接近速度和原始恒定周期任务、固定周期任务(执行优先度5)的控制周期。
- 不会产生复位到机械原点的动作。原点补偿量为“0”时，在不同于机械原点的位置停止的状态下结束原点复位处理。
- 原点补偿量设为“0”时，不执行原点补偿动作。原点补偿量为“0”时，如前所述，在不同于机械原点的位置结束原点复位处理。原点补偿量非“0”时，与其它原点复位动作模式相同，将原点补偿量作为机械原点的相对量进行动作。

- 原点位置偏置可变更机械原点位置。如下图所示，停止位置从原点偏移“D”时，原点位置偏置为“L”，如果原点补偿量为“0”，则原点复位处理结束时的位置为“L + D”。

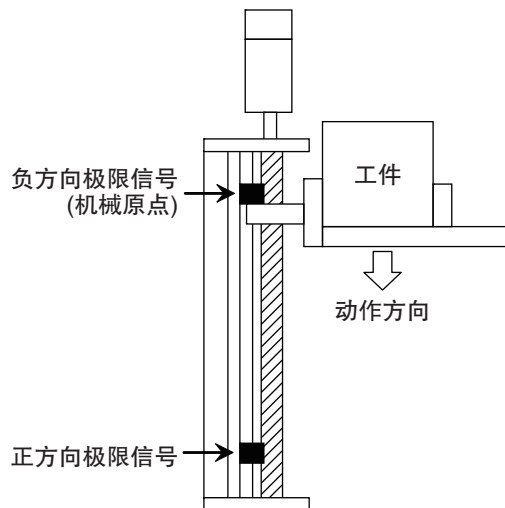


使用注意事项

- 检测到“极限信号ON→OFF”后，在执行累积脉冲停止到开始原点补偿动作期间，可忽略原点检测方向和反方向的极限信号。
- 在补偿量为“0”，确定原点时，要将原点检测反方向的极限输入信号设为“ON”时，请在远离极限信号输入的方向上设定原点补偿量。

原点复位动作模式11的使用示例

如下图所示，没有设置“负方向极限信号”和“原点信号”这两者的空间时，使“负方向极限信号”兼作极限信号和机械原点加以使用。



12: 指定为附近避让、接触时间的动作

1 以原点复位速度启动，执行近原点输入信号的“OFF” → “ON”，开始减速为原点复位接近速度。

在开始减速的同时，开始时间监控。

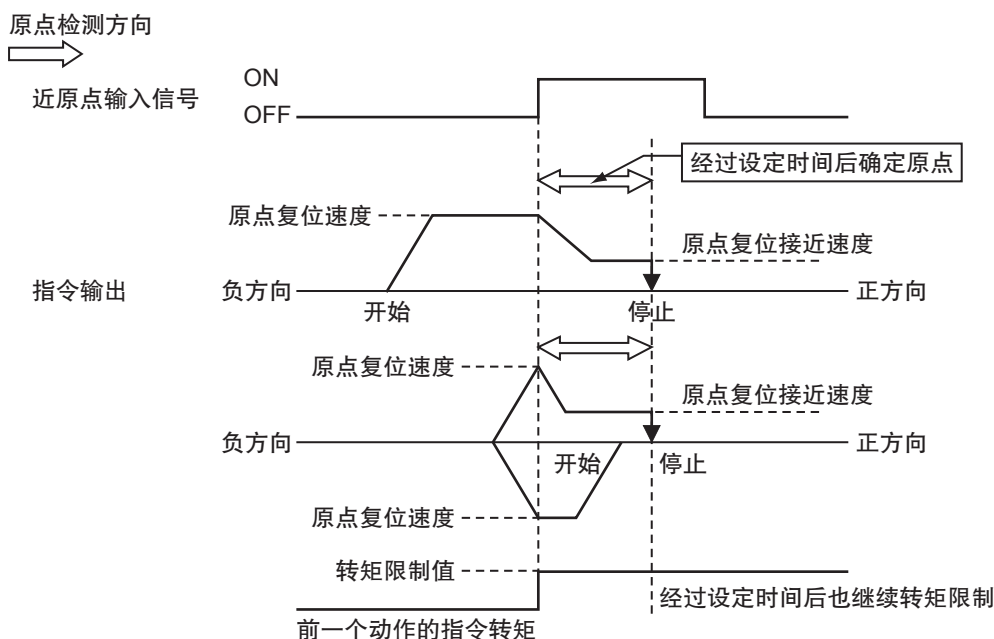
关于开始接触时的转矩限制，欧姆龙制伺服驱动器1S系列及G5系列和其它伺服的处理存在如下不同。

- 1S系列、G5系列 : 自动执行转矩限制。
 上述以外的伺服 : 根据设定执行转矩限制。

关于接触动作，请参阅 □□ “接触动作(使用欧姆龙制伺服驱动器1S系列时)(P.3-31)” 或 □□ “接触动作(使用欧姆龙制伺服驱动器 G5系列时)(P.3-32)”。

2 经过设定时间后，确定机械原点。

“ON”后近原点输入信号的状态不影响动作。



- 如果在近原点输入信号“ON”的状态下执行原点复位，则在原点检测方向的反方向上以原点复位速度开始动作。在近原点输入信号的“ON”状态下避让后，以原点复位接近速度，在原点检测方向上开始原点复位动作。
- 关于转矩限制的解除，欧姆龙制伺服驱动器1S系列及G5系列和其它伺服的处理也不相同。

- 1S系列、G5系列 : 原点复位后，在原点复位的反方向上初次动作时自动“OFF”。
 上述以外的伺服 : 使用转矩限制时，在原点复位后，在原点复位的反方向上初次移动时，请解除转矩限制。对转矩限制改写时，请使用EC_CoESDOWrite(CoE SDO 写入)指令。

- 检测到近原点输入信号后，在达到原点复位接近速度之前，即使经过接触时间也不发生异常，从而确定机械原点。
- 开始原点复位后，在低于原点复位接近速度的状态下，近原点输入信号由“OFF” → “ON”后，在经过接触时间后也可确定机械原点。

13: 指定为无近原点输入、接触原点输入的动作

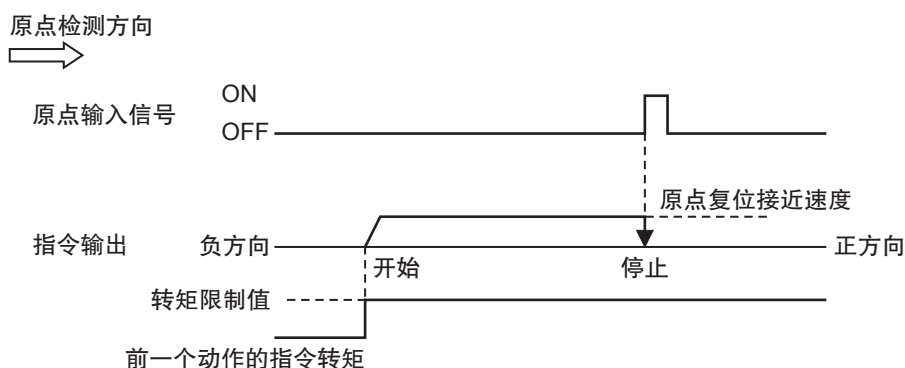
1 以原点复位接近速度启动。

关于开始接触时的转矩限制，欧姆龙制伺服驱动器1S系列及G5系列和其它伺服的处理存在如下不同。

- 1S系列、G5系列 : 自动执行转矩限制。
- 上述以外的伺服 : 根据需要执行转矩限制。

关于接触动作，请参阅 □ “接触动作(使用欧姆龙制伺服驱动器1S系列时)(P.3-31)” 或 □ “接触动作(使用欧姆龙制伺服驱动器 G5系列时)(P.3-32)”。

2 原点输入由“OFF” → “ON”，确定机械原点。



- 关于转矩限制的解除，欧姆龙制伺服驱动器1S系列及G5系列和其它伺服的处理也不相同。

- 1S系列、G5系列 : 原点复位后，在原点复位的反方向上初次动作时自动“OFF”。
- 上述以外的伺服 : 使用转矩限制时，在原点复位后，在原点复位的反方向上初次移动时，请解除转矩限制。对转矩限制改写时，请使用EC_CoESDOWrite(CoE SDO 写入)指令。

- 开始原点复位后，在达到原点复位接近速度之前，原点输入信号“ON”后也可确定机械原点。

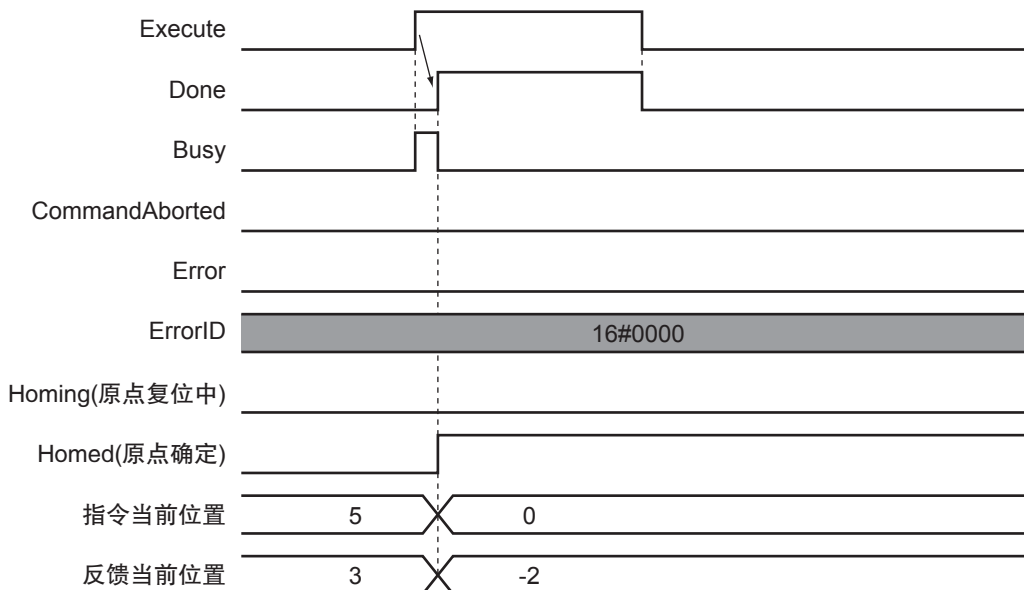


使用注意事项

欧姆龙制伺服驱动器 G5系列 EtherCAT通信内置直线电机型无法将Z相输入映射至PDO。因此，要使用原点复位动作的“13: 指定为无近原点输入、接触原点输入”时，请勿在原点输入信号中选择Z相输入。注：该原点复位动作使用PDO中映射的Z相输入。

14: 原点预设的动作

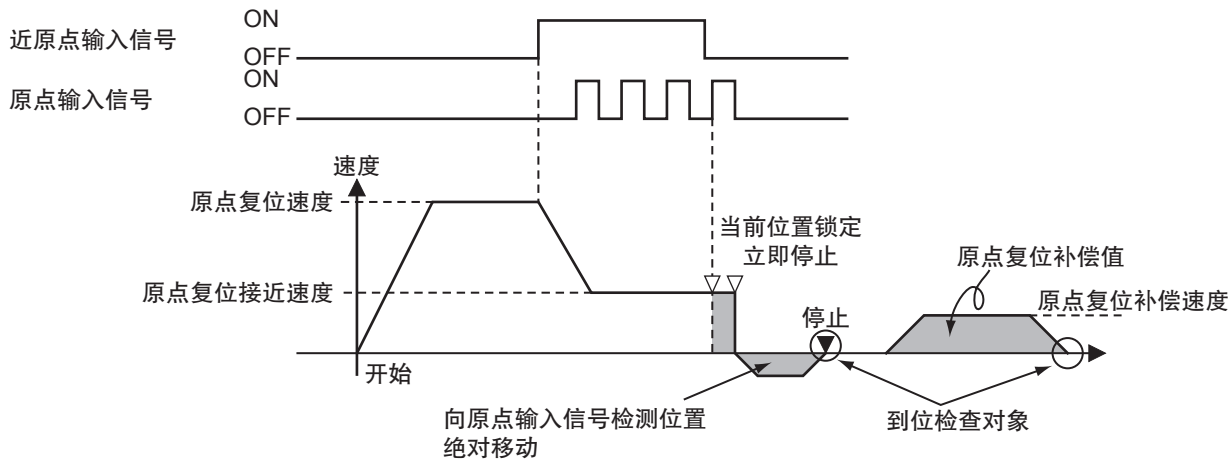
将指令当前位置变更为原点位置偏置，确定机械原点。
并且保持此时的指令当前位置和反馈位置的偏差。



● 确定原点时的停止方法和到位检查

确定原点时，按照如下步骤进行停止。

- 1** 锁定原点输入信号检出时的当前位置，立即停止。
- 2** 立即停止后，以绝对移动的方式反转移到1锁定的位置。



- 执行本指令时，总是等待定位完成。与轴参数“到位检查时间”的设定无关。尤其是，向原点输入信号检测位置绝对移动时、以及将到位检查时间设定为“0”时，将持续等待定位完成。因此，设定时请确保在到位宽度范围内。在原点复位补偿值的动作中，即使将到位检查时间设为“0”，也不会等待到位。此外，可通过运动控制系统变量“`InPosWaiting`(到位等待)”确认是否处于到位等待状态。

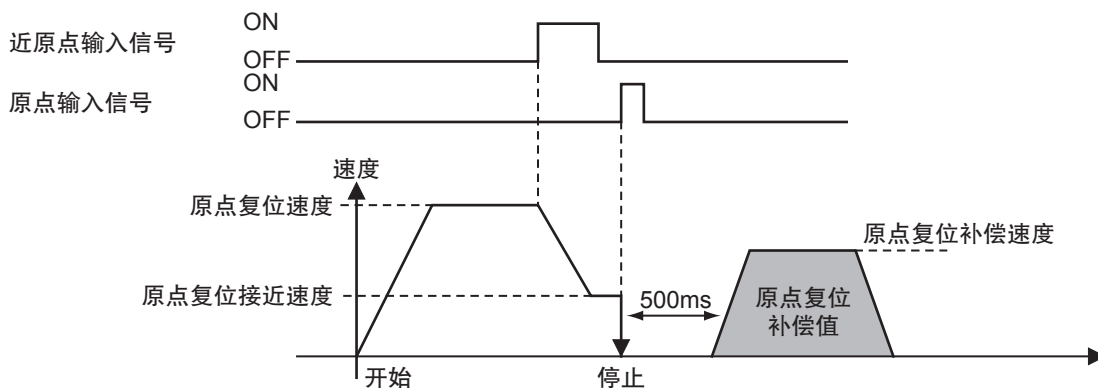
● 原点补偿

设定了原点复位补偿值，检测到原点输入后，可按照原点复位补偿值移动，确定原点。

在机械系统中确定机械原点后，调整工件比较困难。因此，需要微调时，请将最初确定的机械原点的偏移量设为补偿数据。

此时的移动速度即为原点复位补偿速度，在检测到原点输入而停止后经过500ms，再开始移动。

原点复位补偿值的符号表示相对坐标轴的方向，原点复位补偿值为负数时负方向动作。



● 超调

超调对本指令无效。

● 转矩限制的自动控制

使用欧姆龙制伺服驱动器1S系列或G5系列时，在原点复位动作中选择“12: 指定为附近避让、接触时间”或“13: 指定为无近原点输入、接触原点输入”时，在接触方向上自动开始限制转矩。

原点复位时转矩限制的自动控制表现为，在原点复位后也会保持转矩限制状态，在原点检测方向和反方向动作时，转矩限制无效。

此外，在如下场合，可自动解除转矩限制。

- 伺服OFF(伺服解锁)
- 从周期同步位置控制(CSP)切换为其它控制模式时

在原点复位动作过程中，如果动作方向与原点检测方向相反，则解除转矩限制。

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

继续反转并且动作方向变为原点检测方向时，再次将转矩限制设为有效。

● 接触动作(使用欧姆龙制伺服驱动器1S系列时)

选择原点复位的[12:指定为附近避让、接触时间]、[13:指定为无近原点输入、接触原点输入]时，使用伺服驱动器事先设定的转矩限制值，自动在原点检测方向上开始转矩限制。



使用注意事项

使用其他公司制造的伺服驱动器时，基于MC_Home(原点复位)指令的自动转矩限制功能不起作用。

请使用MC_SetTorqueLimit(转矩限制)指令、SDO通信、支持工具等，设定合适值。



参考

- 原点复位正常完成后也将继续转矩限制。
- 启动向反方向移动的指令后，转矩限制自动解除。

欧姆龙制伺服驱动器1S系列的设定

使用接触动作时，必须利用Sysmac Studio设定1S系列伺服驱动器的[转矩限制]的[切换选择](3330Hex-01Hex)。

- 在原点复位的接触动作时，在原点检测方向上进行转矩限制。对于其他动作，在使用MC_SetTorqueLimit(转矩限制)指令的转矩限制方向和转矩限制值时，选择“2”。
这种情况下，在原点复位的接触动作中，MC_SetTorqueLimit(转矩限制)指令输入变量的内容将被忽略。
- 设定为“0”时，始终采用MC_SetTorqueLimit(转矩限制)指令输入变量的内容。原点复位的接触动作时和其他动作时都必须设定合适的转矩限制值。

		[转矩限制]的[切换选择](3330Hex-01Hex)	
		2	0
正方向转矩限制值	原点复位中	采用伺服驱动器的[转矩限制]的[正方向转矩限制值2](3330Hex-05Hex)。	采用MC_SetTorqueLimit(转矩限制)指令的PositiveValue(正方向转矩限制值)。
	非原点复位	采用MC_SetTorqueLimit(转矩限制)指令的PositiveValue(正方向转矩限制值)。	
负方向转矩限制值	原点复位中	采用伺服驱动器的[转矩限制]的[负方向转矩限制值2](3330Hex-06Hex)。	采用MC_SetTorqueLimit(转矩限制)指令的NegativeValue(负方向转矩限制值)。
	非原点复位	采用MC_SetTorqueLimit(转矩限制)指令的NegativeValue(负方向转矩限制值)。	

关于转矩限制，请参阅 □ “MC_SetTorqueLimit(P.3-316)”。

此外，关于1S系列伺服驱动器的设定，请参阅 □ “AC伺服电机/驱动器1S系列EtherCAT通信内置型用户手册(SBCE-377)”。

● 接触动作(使用欧姆龙制伺服驱动器 G5系列时)

选择原点复位的[12:指定为附近避让、接触时间]、[13:指定为无近原点输入、接触原点输入]时，使用伺服驱动器事先设定的转矩限制值，自动在原点检测方向上开始转矩限制。



使用注意事项

使用其他公司制造的伺服驱动器时，基于MC_Home(原点复位)指令的自动转矩限制功能不起作用。

请使用MC_SetTorqueLimit(转矩限制)指令、SDO通信、支持工具等，设定合适值。



参考

- 原点复位正常完成后也将继续转矩限制。
- 启动向反方向移动的指令后，转矩限制自动解除。

欧姆龙制伺服驱动器 G5系列的设定

使用接触动作时，必须利用伺服驱动器的支持软件，设定伺服驱动器G5系列的转矩限制选择(3521 Hex)。

- 在 原点复位的接触动作时，在 原点检测方向上进行转矩限制。对于其他动作，在使用 MC_SetTorqueLimit(转矩限制)指令的转矩限制方向和转矩限制值时，选择“6”。
- 这种情况下，在 原点复位的接触动作中，MC_SetTorqueLimit(转矩限制)指令输入变量的内容将被忽略。
- 设定为“4”时，始终采用MC_SetTorqueLimit(转矩限制)指令输入变量的内容。原点复位的接触动作时和其他动作时都必须设定合适的转矩限制值。

		转矩限制选择(3521 Hex)	
		6(推荐)	4
正方向转矩限制值	原点复位中	采用第3转矩限制(3525 Hex)。	采用MC_SetTorqueLimit(转矩限制)指令的PositiveValue(正方向转矩限制值)与第1转矩限制(3013 Hex)中的较小值。
	非原点复位	采用MC_SetTorqueLimit(转矩限制)指令的PositiveValue(正方向转矩限制值)与第1转矩限制(3013 Hex)中的较小值。	
负方向转矩限制值	原点复位中	采用第4转矩限制(3526 Hex)。	采用MC_SetTorqueLimit(转矩限制)指令的NegativeValue(负方向转矩限制值)与第2转矩限制(3522 Hex)中的较小值。
	非原点复位	采用MC_SetTorqueLimit(转矩限制)指令的NegativeValue(负方向转矩限制值)与第2转矩限制(3522 Hex)中的较小值。	

关于转矩限制，请参阅 □□ “MC_SetTorqueLimit(P.3-316)”。

关于伺服驱动器 G5系列的设定，请参阅 □□ “AC伺服电机/驱动器 G5系列 EtherCAT通信内置型 用户手册(SBCE-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-366)”。

● 接触动作中的位置偏差监控

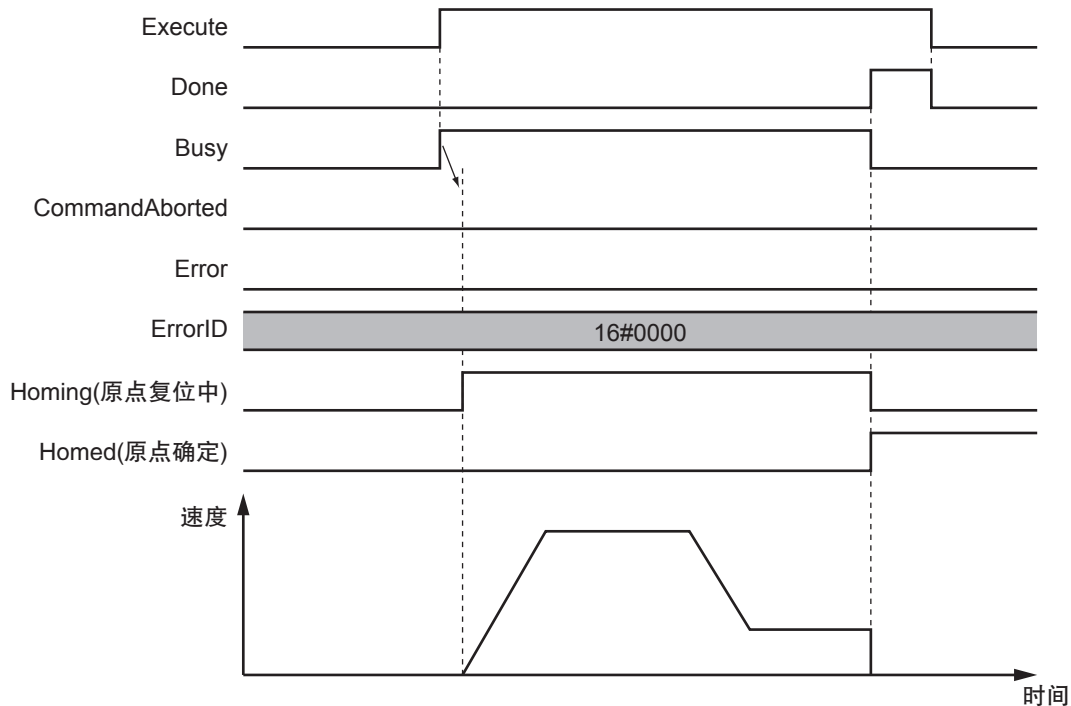
在 原点复位的接触动作中，位置偏差监控功能不会启动。

位置偏差监控功能的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

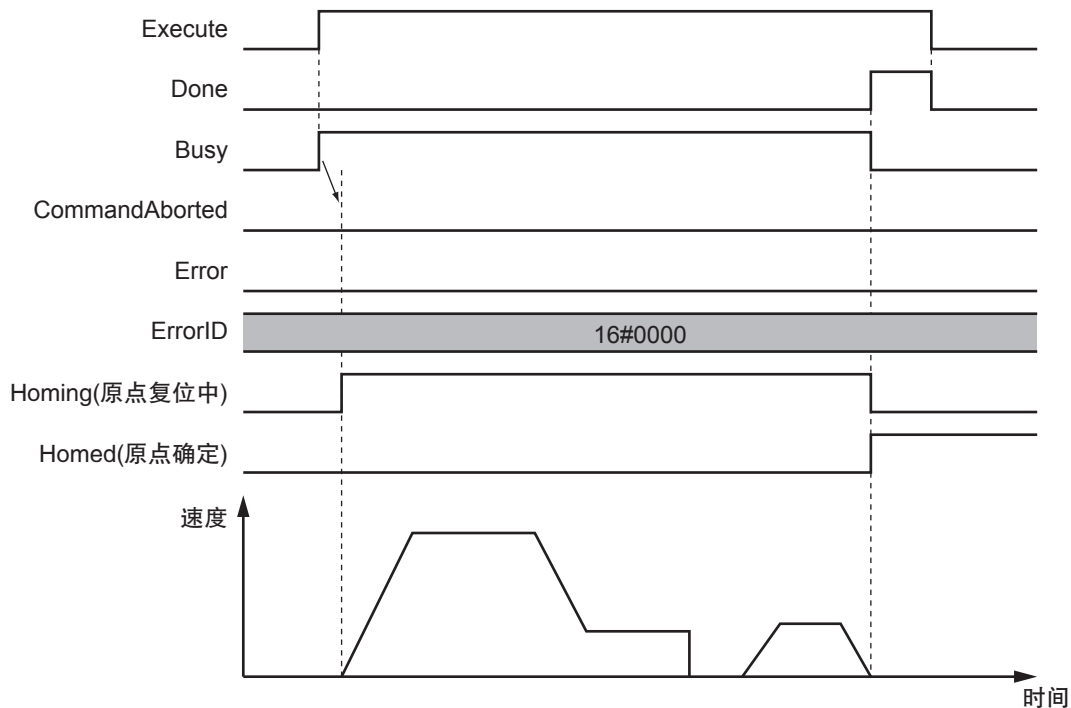
时序图

执行原点复位时的时序图如下所示。

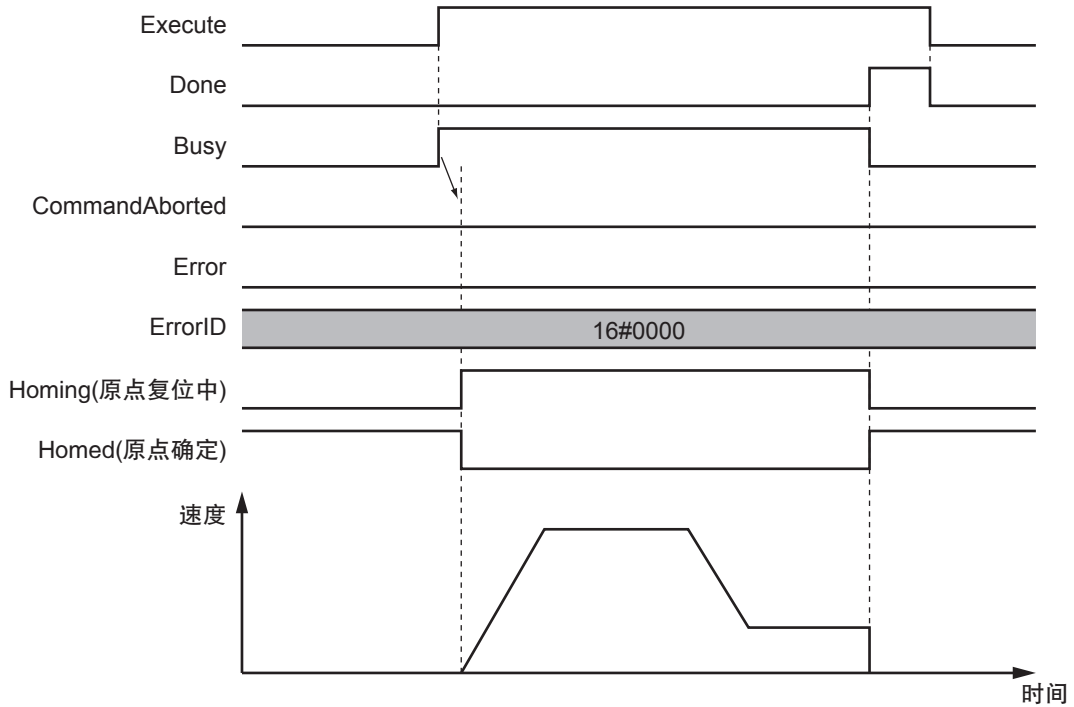
● 无原点补偿时



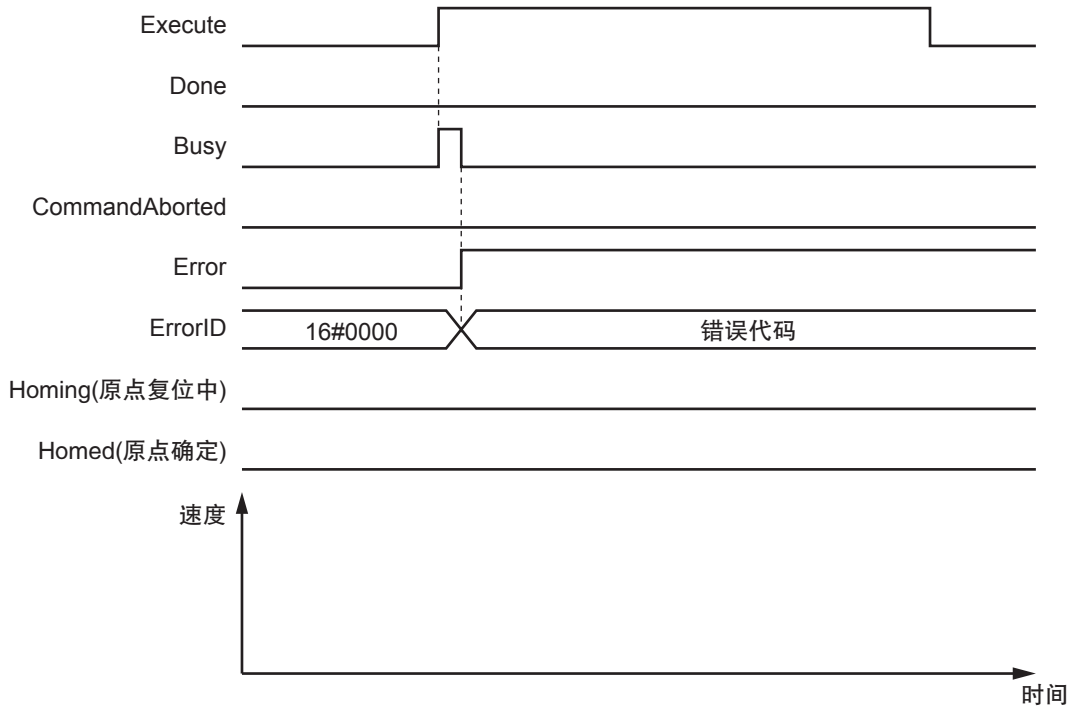
● 有原点补偿时



● 在 origin 确定状态下执行了本指令时



● 在参数错误的状态下执行了本指令时、以及在不可执行运动指令的状态下执行了本指令时



重启运动指令

无法重启本指令。

重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

多重启动运动指令的详情，请参阅 [□](#) “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 [□](#) “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

错误代码

关于指令发生的异常，请参阅 [□](#) “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

Ver.1.03以上的CPU单元和Ver.1.04以上的Sysmac Studio组合时可使用本指令。

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	在上升沿开始指令。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

*1. □ 请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 执行MC_Stop指令中，启动本指令时 	<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含有异常因素时	异常已解除时

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. 请使用在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型的用户定义变量。

● _sHOMING_REF

结构要素变量	名称	数据类型	有效范围	功能
HomingMode	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	指定要改写的原点复位动作。 0: 指定为附近避让、近原点输入OFF 1: 指定为附近避让、近原点输入ON 4: 指定为近原点输入OFF 5: 指定为近原点输入ON 8: 指定为极限输入OFF 9: 指定为附近避让、原点输入屏蔽距离 11: 仅极限输入 12: 指定为附近避让、接触时间 13: 指定为无近原点输入、接触原点输入 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	设定启动原点复位时的开始方向。 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


结构要素变量	名称	数据类型	有效范围	功能
PosiLmtMode	正方向极限输入时动作	_eMC_ LIMIT_ REVERSE_ MODE	0:_mcErrorStop 1:_mcRevImmediateStop 2:_mcRevDeceleration Stop	设定原点复位中正方向极限输入“ON”时的停止方法。 0: 不反转/轴异常而停止(遵照极限输入停止方法) 1: 反转/立即停止 2: 反转/减速停止
NegaLmtMode	负方向极限输入时动作	_eMC_ LIMIT_ REVERSE_ MODE	0:_mcErrorStop 1:_mcRevImmediateStop 2:_mcRevDeceleration Stop	设定原点复位中负方向极限输入“ON”时的停止方法。 0: 不反转/轴异常而停止(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	设定原点复位中的速度。 请设定为最高速度以下, 并且满足原点复位接近速度 \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ON后的速度。 请设定为最高速度以下, 并且满足原点复位接近速度 \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。 设定为“0”时不加速。 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。 设定为“0”时不减速。 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。 设定为“0”时无跃度。 请在脉冲单位40位以内设定。 单位为[指令单位/s ³]。 ^{*2}
Mask	原点输入屏蔽距离	LREAL	正数或“0”	对原点复位动作模式指定为附近避让、原点输入屏蔽距离时原点输入的屏蔽距离进行设定。 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, 按照设定值预设当前位置。 在旋转模式下设定时, 请确保满足环计数器下限值 \leq 原点位置偏置 $<$ 环计数器上限值。 并且, 请在脉冲单位40位以内设定。 单位为[指令单位]。 ^{*2}
PushTime	原点复位接触时间	UINT	0 ~ 10,000	以毫秒为单位, 对原点复位动作模式指定为附近避让、接触时间时的接触时间进行设定。 单位为[ms]。

结构要素变量	名称	数据类型	有效范围	功能
Compensation	原点复位补偿值	LREAL	负数、正数、“0”	设定原点复位的原点确定后的原点复位补偿量。 在旋转模式下设定时，请确保满足 $ \text{原点复位补偿值} < \text{环计数器上限值} - \text{环计数器下限值} $ 。并且，请在脉冲单位40位以内设定。 单位为[指令单位]。 ^{*2}
Compensation Vel	原点复位补偿速度	LREAL	正数	设定原点复位补偿时的速度。 请确保在最高速度以下。 单位为[指令单位/s]。 ^{*2}

- *1. 使用欧姆龙制伺服驱动器1S系列及G5系列时可指定。将分配给伺服驱动器的锁定功能1的输入用作外部原点输入。伺服驱动器1S系列及G5系列的初始状态下，将外部锁定输入1分配给锁定功能1。
详情请参阅  “AC伺服电机/驱动器1S系列EtherCAT通信内置型用户手册(SBCE-377)”、 “AC伺服电机/驱动器G5系列EtherCAT通信内置型用户手册(SBCE-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-366)”。
- *2. 关于指令单位，请参阅  “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”或  “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”的“单位转换设定”。

功能说明

- 对于Axis(轴)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由输入输出变量HomingParameter来设定。
- 执行本指令不会变更轴参数。
- 本指令与MC_Home(原点复位)指令的不同之处仅在于各种参数的设定方法。利用本指令，可由输入输出变量HomingParameter来设定各种参数。利用MC_Home(原点复位)指令，可通过Sysmac Studio的轴参数[原点复位动作]来设定各种参数。

关于各种参数的设定方法以外的指令规格，请参阅  “MC_Home(P.3-15)”。



使用注意事项

本指令指定的原点复位参数仅在执行本指令时有效。依次执行本指令、MC_Home(原点复位)指令后，根据轴参数设定的原点复位参数进行动作。

原点复位动作与原点复位参数的关系

原点复位参数中，根据原点复位动作的设定，存在不使用的参数。不使用的参数并非范围检查和匹配检查的对象。执行本指令时，执行范围检查和匹配检查。

对于原点复位动作的设定，是否使用各原点复位参数，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	原点输入屏蔽距离	原点位置偏置	原点复位接触时间	原点复位补偿值	原点复位补偿速度
指定为附近避让、近原点输入OFF	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为附近避让、近原点输入ON	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为近原点输入OFF	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为近原点输入ON	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为极限输入OFF	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为附近避让、原点输入屏蔽距离	○	○	○	○	○	○	○	○	○	○	○	○	×	○	○
仅极限输入	×	○	○	○	○	○	○	○	○	○	×	○	×	○	○
指定为附近避让、接触时间	×	○	○	○	○	○	○	○	○	○	×	○	○	○	○
指定为无近原点输入、接触原点输入	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
原点预设	×	×	×	×	×	×	×	×	×	×	×	○	×	×	×



参考

使用NX系列位置接口单元时，无法选择使用接触的原点复位动作模式。详情请参阅 □□ “NX系列位置接口单元 用户手册(SBCE-374)”。

重启运动指令

无法重启本指令。

重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

多重启动运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

错误代码

关于指令发生的异常，请参阅 □□ “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	在上升沿开始指令
Position	目标位置	LREAL	负数、正数、“0”	0	对移动方法选择指定绝对定位时，指定绝对坐标的目标位置。 对移动方法选择指定相对定位时，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}
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: 指定为正方向 1: 指定为附近 2: 指定为负方向 3: 指定为当前方向 4: 无方向指定

输入变量	名称	数据类型	有效范围	初始值	内容
BufferMode	缓存模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	指定多重启动运动指令时的动作。 0: 中断 1: 等待 2: 以低速合并 3: 以前一个速度合并 4: 以后一个速度合并 5: 以高速合并
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. 关于指令单位, 请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*2. 请务必设定目标速度。不作设定就执行动作, 将超出目标速度设定范围(错误代码: 5422Hex)。

*3. 有效范围为枚举体的变量, 其实际初始值不是数值, 而是枚举元素。

*4. MoveMode为[1: 相对值定位]时, 以Position的符号确定移动方向。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时, 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为正常。

*1. 请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	定位完成后	<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	开始指令后	<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"> 利用其它指令多重启动运动指令(中断), 中止本指令时 因发生异常, 中止本指令时 发生异常过程中, 启动本指令时 执行MC_Stop指令中, 启动本指令时 	<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含有异常因素时	异常已解除时

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. 请在使用在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。
- 对MoveMode(移动方法选择)指定“绝对值定位”时，产生与MC_MoveAbsolute(绝对值定位)指令相同的动作。同样，指定“相对值定位”时，产生与MC_MoveRelative(相对值定位)指令相同的动作。
- 作为相对值定位指令动作时，不使用Direction(方向选择)。

详情请参阅 □□ “MC_MoveAbsolute(P.3-47)” 或 □□ “MC_MoveRelative(P.3-73)”。

指令详情

下面对本指令详细说明。



使用注意事项

进行绝对值定位时，请在轴参数的[环计数器上限设定值]、以及[环计数器下限设定值]的范围内指定目标位置。
不在[环计数器上限设定值]和[环计数器下限设定值]的范围内指定目标位置时，将发生“超出目标位置设定范围(错误代码：5478Hex)”错误。

● 到位检查

根据轴参数的[到位宽度]、[到位检查时间]的设定，进行本指令的到位检查。

重启运动指令

在定位动作中变更输入变量，再次将Execute(启动)设为TRUE，可变更本指令的动作。重启运动指令可变更的输入变量有Position(目标位置)、Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度)。

重启运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

多重启动运动指令

多重启动运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动

对当前正在执行的指令启动本指令，可切换或缓存到本指令。各轴可缓存到1个。

多重启动时的本指令的动作由BufferMode(缓存模式选择)指定。

缓存模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 轴的动作方向因指令切换而反转时，根据轴参数中的“反转时动作”进行反转。
等待	当前正在执行的指令正常完成后，已缓存的本指令自动启动。
合并	以当前正在执行的指令到达目标位置时的速度(中继速度)为启动速度，连续使已缓存的本指令动作。变更当前正在执行的指令的动作，确保以中继速度到达目标位置。中继速度的指定方法分为如下4种。
以低速合并	当前正在执行的目标速度与已缓存的目标速度中，以速度较低者为中继速度。
以前一个速度合并	以当前正在执行的目标速度为中继速度。
以后一个速度合并	以已缓存的本指令的目标速度为中继速度。
以高速合并	当前正在执行的目标速度与已缓存的目标速度中，以速度较高者为中继速度。

关于BufferMode(缓存模式选择)，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

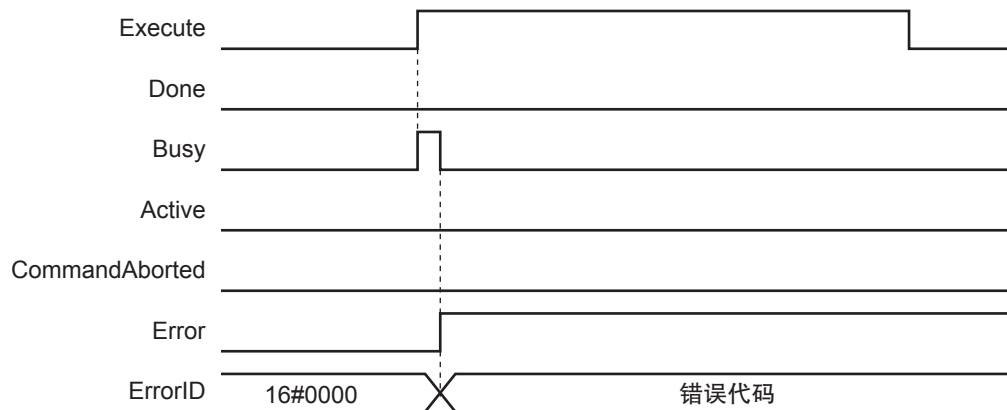
● 本指令执行中的其它指令启动

对本指令进行多重启动运动指令时，可选择中断、等待、合并。

异常

● 发生异常时的时序图

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。



● 错误代码

关于指令发生的异常，请参阅  “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	在上升沿开始指令。
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}
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: 指定为正方向 1: 指定为附近 2: 指定为负方向 3: 指定为当前方向 4: 无方向指定

输入变量	名称	数据类型	有效范围	初始值	内容
BufferMode	缓存模式选择	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	指定多重启动运动指令时的动作。 0: 中断 1: 等待 2: 以低速合并 3: 以前一个速度合并 4: 以后一个速度合并 5: 以高速合并

*1. 关于指令单位，请参阅  “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或  “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*2. 请务必设定目标速度。不作设定就执行动作，将发生异常。

*3. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

*1.  请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	定位完成后	<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	开始指令后	<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"> 利用其它指令多重启动运动指令(中断), 中止本指令时 因发生异常, 中止本指令时 发生异常过程中, 启动本指令时 MC_Stop指令执行中, 启动本指令后 	<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含有异常因素时	异常已解除时

输入输出变量

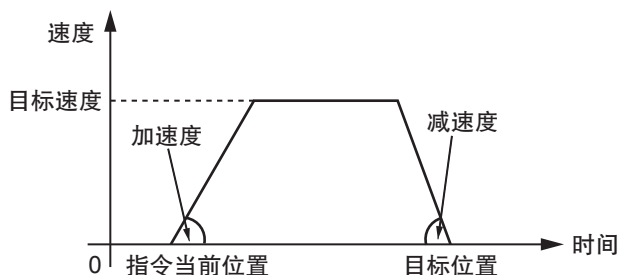
输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	-	指定轴。*1

*1. 请使用在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[*])。

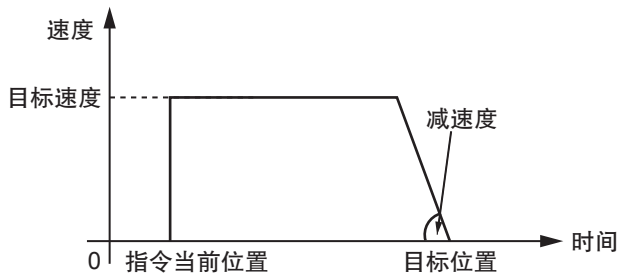
功能说明

- 指定绝对坐标的目标位置, 进行定位。
- 在Execute(启动)的上升沿, 开始绝对定位的动作。
- 即使原点未确定也可执行本指令。
- 可在输入变量中指定Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度)、Jerk(跃度)。

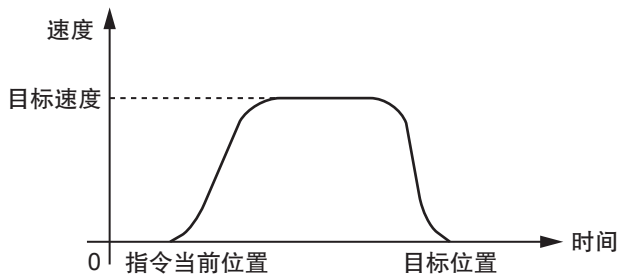
绝对定位的动作示例如下所示。



将加速度、或减速度指定为“0”并启动后，不作加减速而达到目标速度。
加速度为“0”时的动作示例如下所示。



需要平稳加减速时指定Jerk(跃度)。
指定Jerk(跃度)后的动作示例如下所示。



关于Jerk(跃度)的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

与指令当前位置相同，指定Position(目标位置)并启动后，轴不发生移动，但是Done(完成)将变为TRUE。

指令详情

下面对指令详细说明。

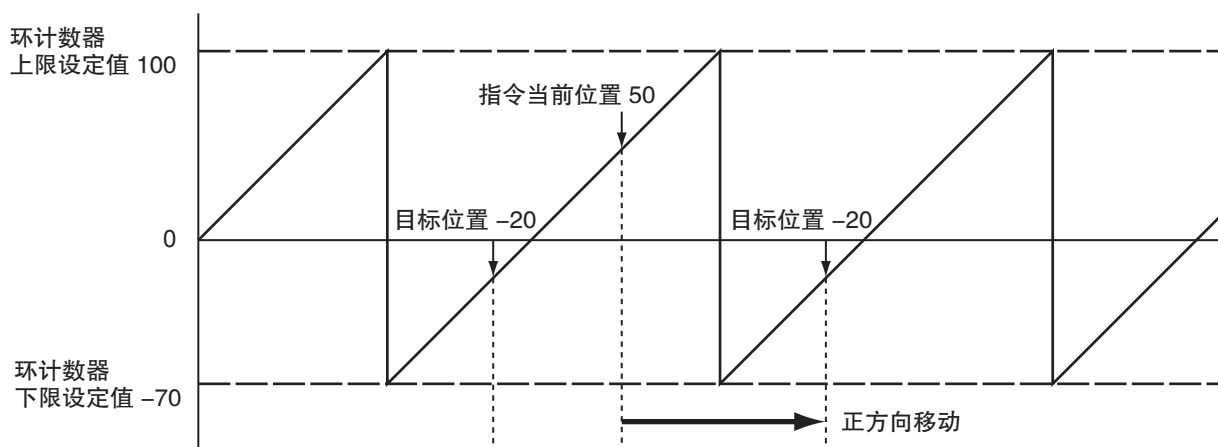
● Direction(方向选择)

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

计数模式为[线性模式]时，忽略Direction(方向选择)。

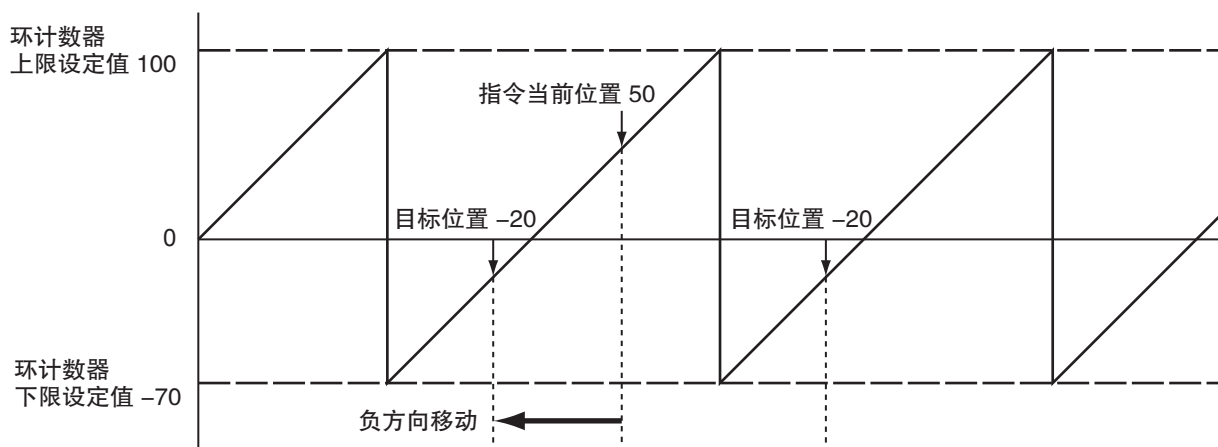
Direction(方向选择)为“指定为正方向”时，迎着正方向目标位置进行定位。

指令位置由“50”到“-20”定位时的动作示例如下所示。



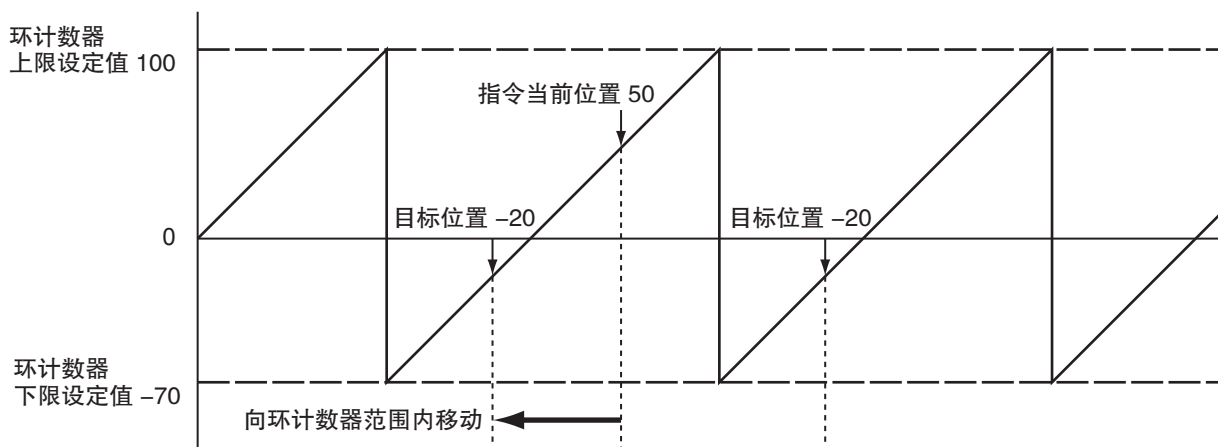
Direction(方向选择)为“指定为负方向”时，迎着负方向目标位置进行定位。

指令位置由“50”到“-20”定位时的动作示例如下所示。



Direction(方向选择)为“无方向指定”时,迎着环计数器范围内的目标位置进行定位。因此,移动方向取决于指令当前位置与目标位置的大小关系。

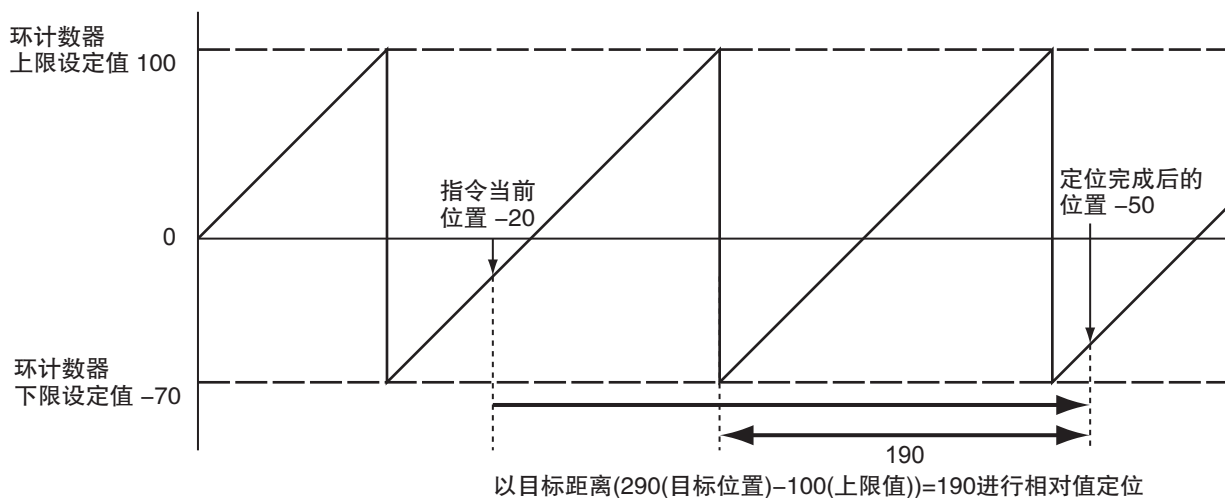
指令位置由“50”到“-20”定位时的动作示例如下所示。



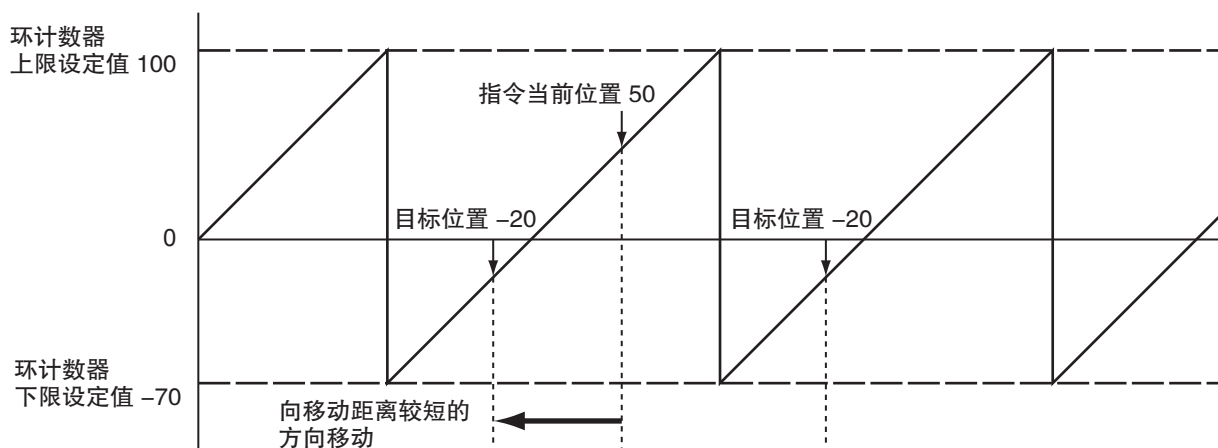
Direction(方向选择)为“无方向指定”时,可指定超出环计数器上下限设定值范围的Position(目标位置)。指定了超过环计数器上限设定值的Position(目标位置)时,将超出设定值的移动量作为超过后的相对量进行定位。因此,可实现环多圈定位。

指定了超过环计数器下限设定值的Position(目标位置)时,同样进行定位。

指令当前位置为“-20”、Position(目标位置)指定为“290”时的动作示例如下所示。



Direction(方向选择)为“指定为附近”时，迎着指令当前位置和目标位置距离较短的方向进行定位。指令位置由“50”到“-20”定位时的动作示例如下所示。



正方向、负方向移动距离都相同时，动作与“指定为当前方向”时相同。



使用注意事项

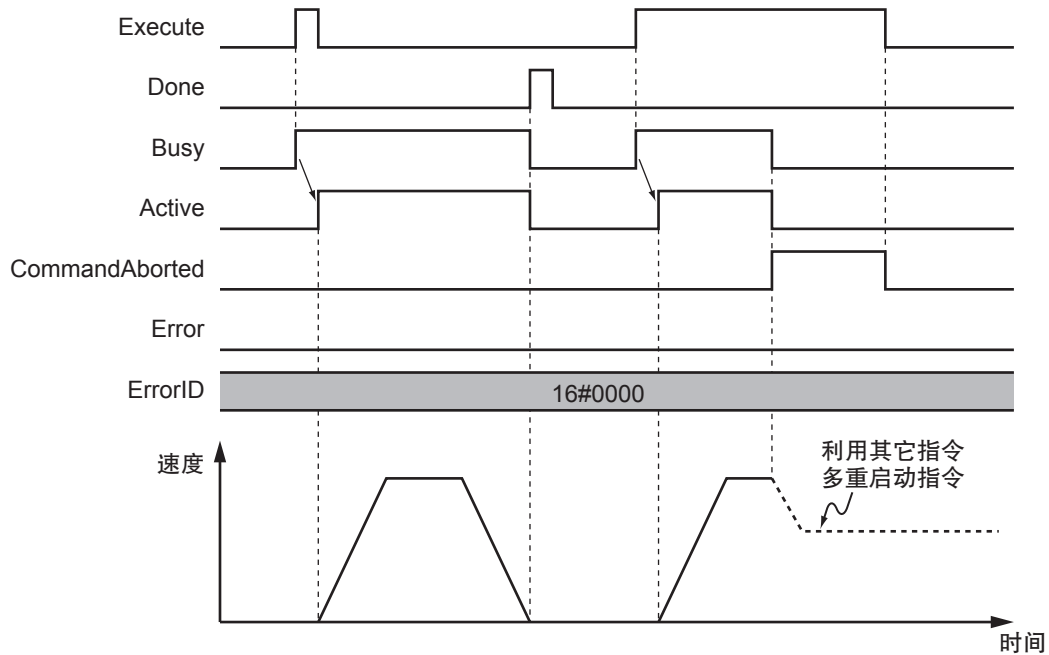
- 进行绝对值定位时，请在低于轴参数的[环计数器上限设定值]、及不低于[环计数器下限设定值]的范围内指定目标位置。否则，将发生“超出目标位置设定范围(错误代码：5478Hex)”错误。但是，Direction(方向选择)为“无方向指定”时，可指定超出环计数器上下限设定值范围的Position(目标位置)。
- Direction(方向选择)选择“3: _mcCurrentDirection(指定为当前方向)”时，沿着前一动作的指令方向进行动作。因此，根据指令组合的不同，可能与前一动作的运动指令的输入发出指令的方向不一致。使用“3: _mcCurrentDirection(指定为当前方向)”时，请通过轴变量的Dir.Posi(正方向指令中)、以及Dir.Nega(负方向指令中)确认当前方向。

● 到位检查

根据轴参数的[到位宽度]、[到位检查时间]的设定，进行本指令的到位检查。

时序图

- 在启动Execute(启动)的同时，Busy(执行中)变为TRUE。在下一个周期Active(控制中)变为TRUE。
- 到达Position(目标位置)、定位完成时，Done(完成)变为TRUE。
- 利用其它指令中止本指令时，CommandAborted(执行中断)变为TRUE，Busy(执行中)、Active(控制中)变为FALSE。



重启运动指令

在定位动作中变更输入变量，再次将Execute(启动)设为TRUE，可变更本指令的动作。重启运动指令可变更的输入变量有Position(目标位置)、Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度)。

重启运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

多重启动运动指令

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动

对当前正在执行的指令启动本指令，可切换或缓存到本指令。

各轴可缓存到1个。

多重启动时的本指令的动作由BufferMode(缓存模式选择)指定。

缓存模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 轴的动作方向因指令切换而反转时，根据轴参数中的“反转时动作”进行反转。
等待	当前正在执行的指令正常完成后，已缓存的本指令自动启动。
合并	以当前正在执行的指令到达目标位置时的速度(中继速度)为启动速度，连续使已缓存的本指令动作。变更当前正在执行的指令的动作，确保以中继速度到达目标位置。中继速度的指定方法分为如下4种。
以低速合并	当前正在执行的目标速度与已缓存的目标速度中，以速度较低者为中继速度。
以前一个速度合并	以当前正在执行的目标速度为中继速度。
以后一个速度合并	以已缓存的本指令的目标速度为中继速度。
以高速合并	当前正在执行的目标速度与已缓存的目标速度中，以速度较高者为中继速度。

关于BufferMode(缓存模式选择)，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

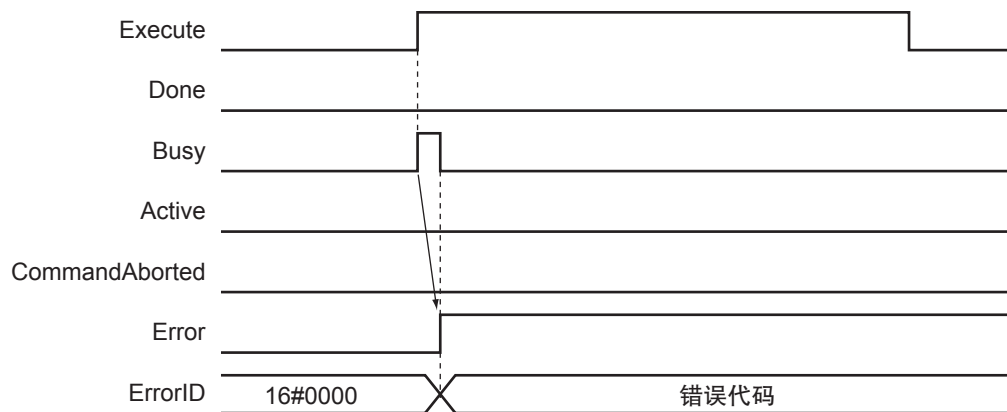
● 本指令执行中的其它指令启动

对本指令进行多重启动运动指令时，可选择中断、等待、合并。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 “A-1 错误代码一览(P.A-2)”。

示例程序1

下面，对基于反复多重启动的定位方法中的示例程序进行说明。

参数设定

该示例程序中使用的最少的必要设定如下所示。

● 轴参数的设定

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	线性模式

坐标单位选择

轴	坐标单位
轴1	mm

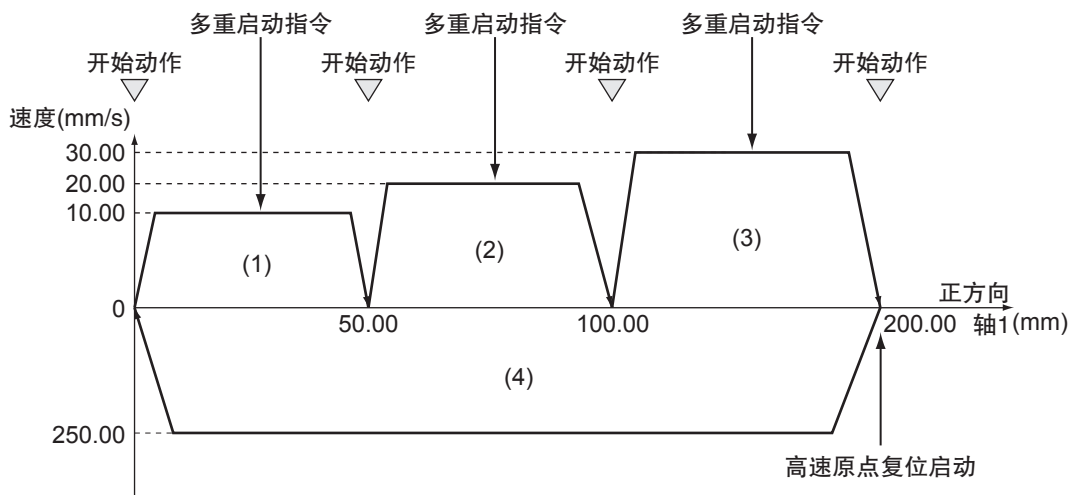
动作示例

利用MC_MoveAbsolute(绝对值定位)指令将BufferMode(缓存模式选择)设为[等待],在多重启动运动指令的同时，移动到最终目标位置。

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

在之前的指令输出变量“Active(控制中)”为TRUE的状态下进行多重启动。只有1个指令可执行单轴动作的指令多重启动。

● 动作模式



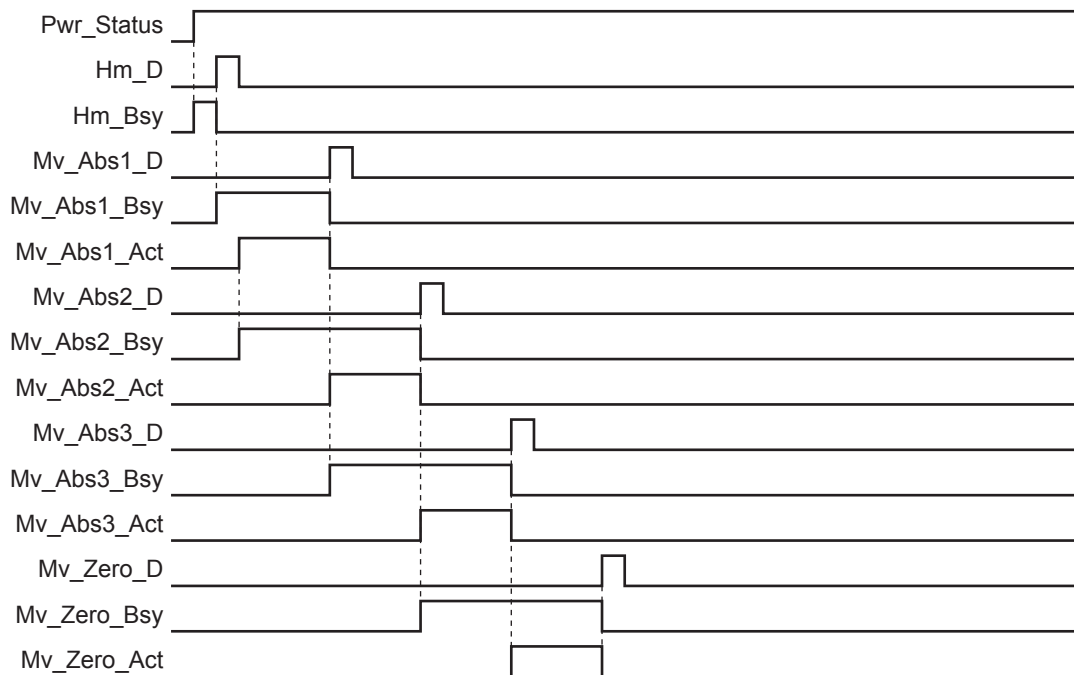
- 1 动作开始开关的“ON”
在原点开启动作开始开关，则轴1正方向50.00mm定位。
- 2 其次，动作开始开关的“ON”
接着，每当开启动作开始开关，轴1正方向100.00mm、200.00mm定位，返回原点并停止动作。

梯形图

● 主要变量

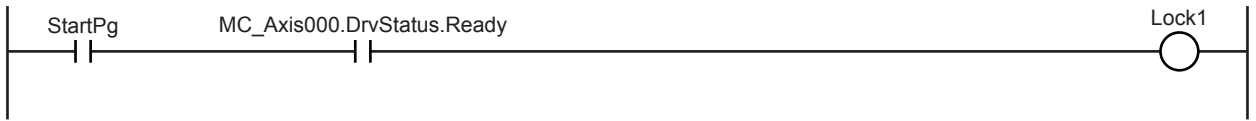
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	轴1的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴1发生轻度故障等级的异常时变为TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴1为原点确定状态时变为TRUE。
Pwr_Status	BOOL	FALSE	分配给MC_Power的实例PWR的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则进入伺服ON状态。

● 时序图

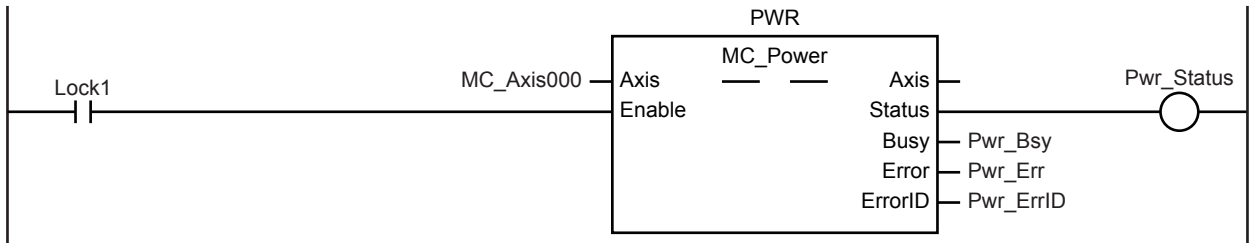


● 示例程序

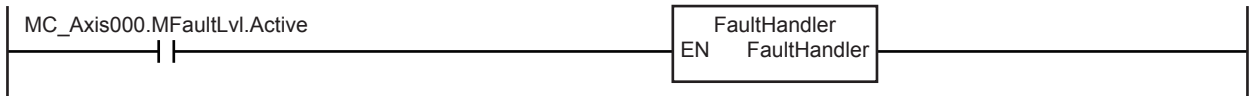
触点StartPg变为TRUE时，确认伺服驱动器处于伺服准备就绪状态



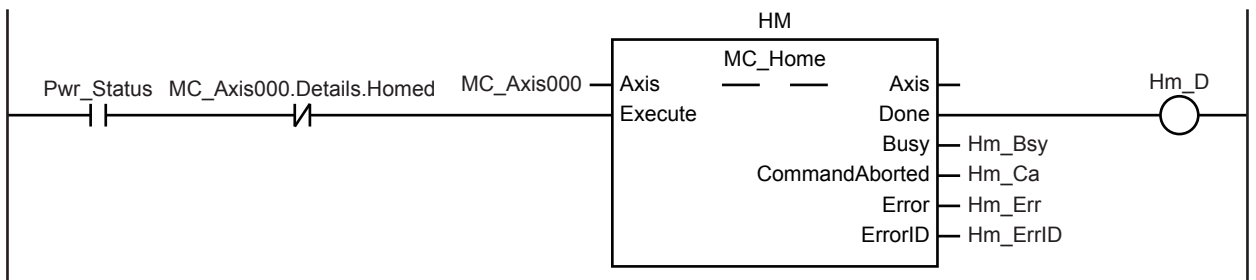
如果处于伺服准备就绪状态，则设为伺服ON状态



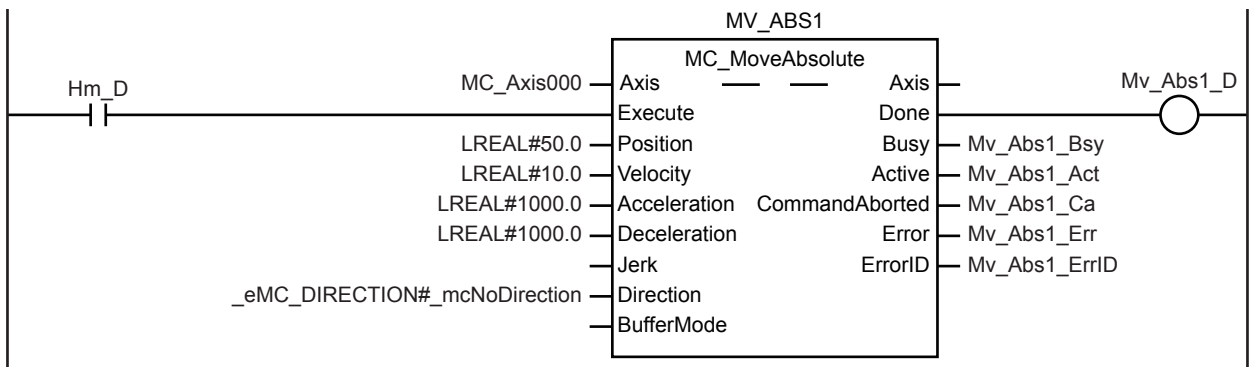
轴1发生轻度故障程度的异常时，执行设备的异常处理(FaultHandler)。发生异常时的处理(FaultHandler)由客户根据装置进行编程



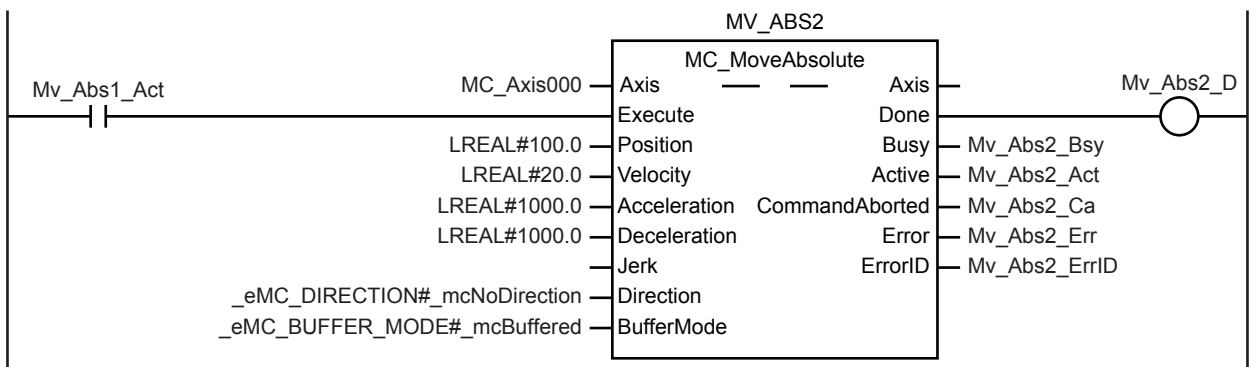
处于伺服ON状态、且原点未确定时，执行原点复位



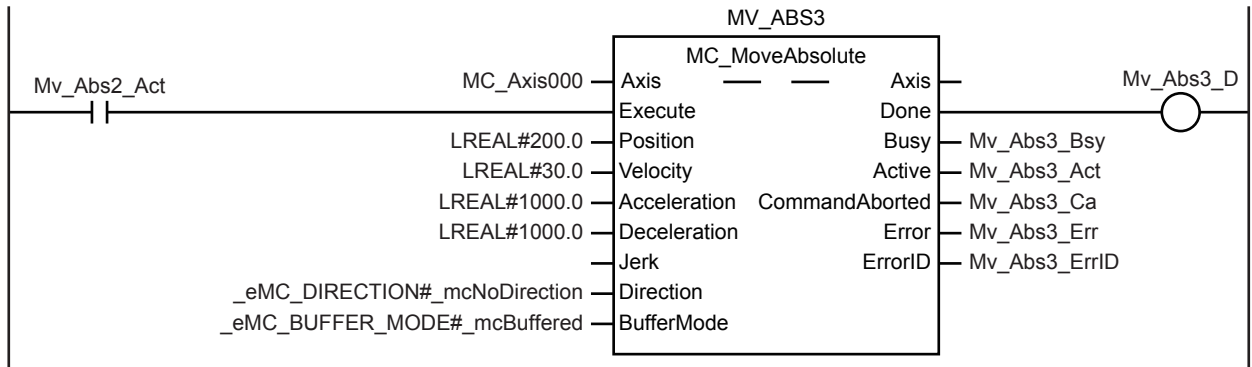
确定原点后，启动绝对值定位(1)



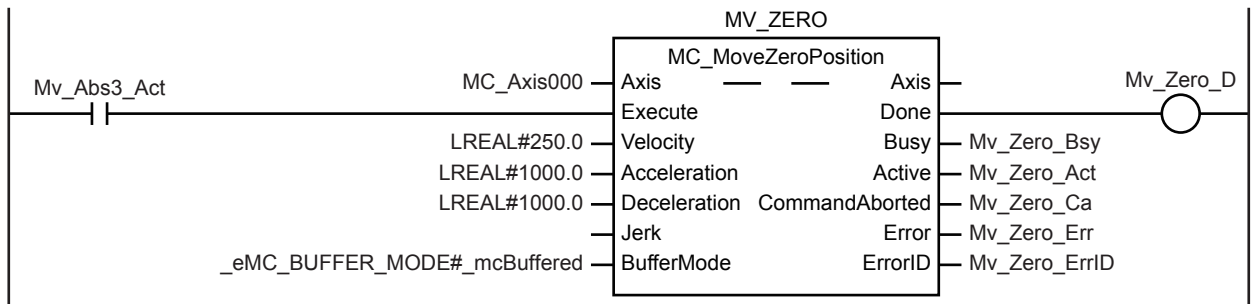
绝对值定位(1)开始后，多重启动绝对值定位(2)



绝对值定位(2)开始后, 多重启动绝对值定位(3)



绝对值定位(3)开始后, 执行高速原点复位

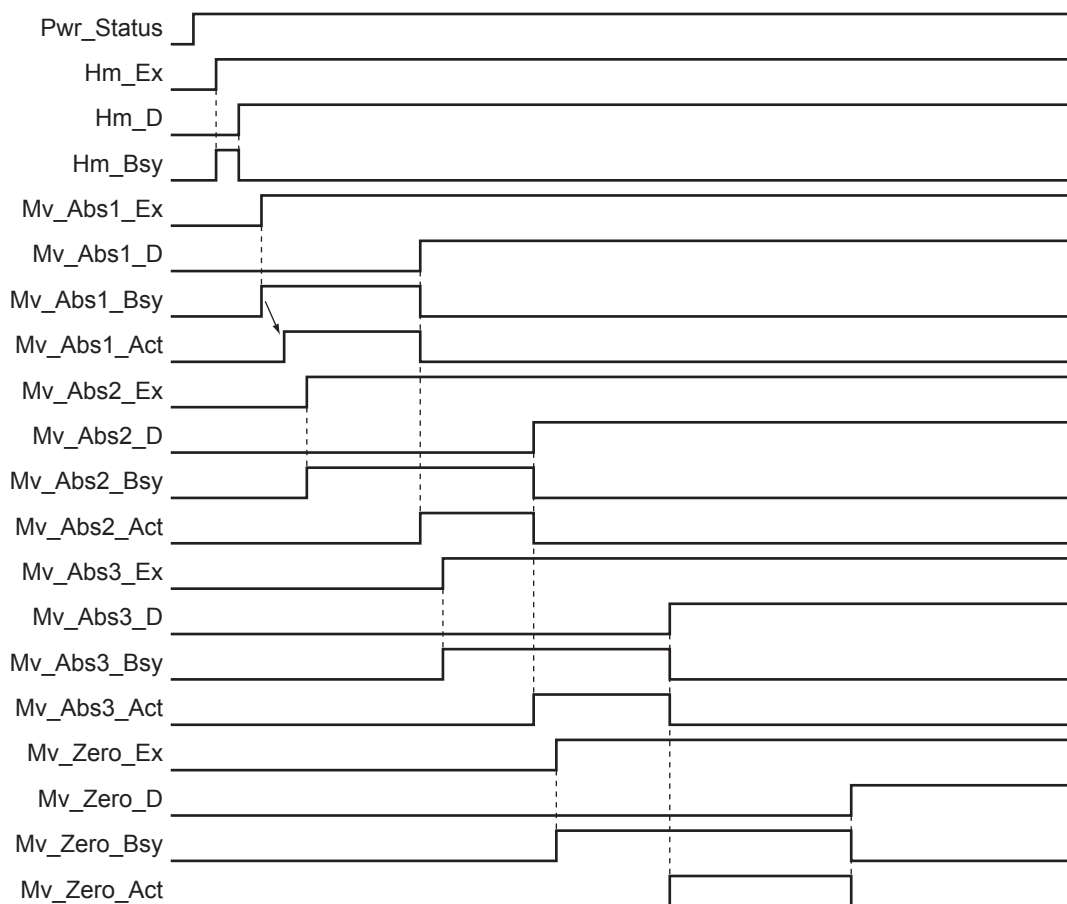


结构文本(ST)

● 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为原点确定状态时变为TRUE。
Pwr_Status	BOOL	FALSE	分配给MC_Power的实例PWR的输出变量Status的变量。进入伺服ON状态时, 该变量变为TRUE。
StartPg	BOOL	FALSE	如果该变量为TRUE, EtherCAT的过程数据通信已建立, 则进入伺服ON状态。
Hm_Ex	BOOL	FALSE	该变量由FALSE→TRUE时, 启动MC_Home的实例HM。
Mv_Abs1_Ex	BOOL	FALSE	该变量由FALSE→TRUE时, 启动MC_MoveAbsolute的实例MV_ABS1。
Mv_Abs2_Ex	BOOL	FALSE	该变量由FALSE→TRUE时, 启动MC_MoveAbsolute的实例MV_ABS2。
Mv_Abs3_Ex	BOOL	FALSE	该变量由FALSE→TRUE时, 启动MC_MoveAbsolute的实例MV_ABS3。
Mv_Zero_Ex	BOOL	FALSE	该变量由FALSE→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;

//确认StartPg为TRUE, 且伺服驱动器处于伺服准备就绪状态,
//则设为伺服ON状态。
//如果未处于伺服准备就绪状态, 则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) 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)由客户根据装置进行编程。
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，为了向目标位置定位，自动识别左右哪边较近进行定位。
下面，对工具更换的附近控制中的示例程序进行说明。

参数设定

该示例程序中使用的最少的必要设定如下所示。

● 轴参数的设定

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

环计数器

轴	上限值	下限值
轴1	360	0

坐标单位选择

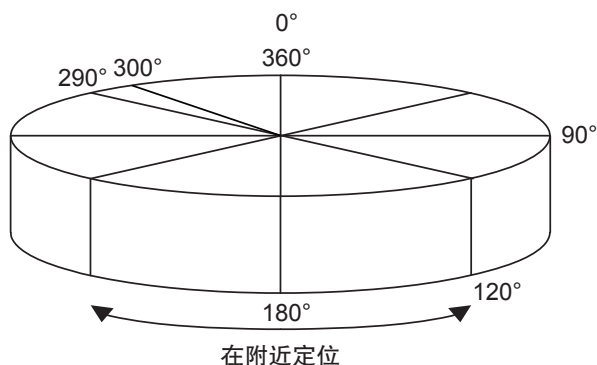
轴	坐标单位
轴1	degree

动作示例

利用绝对值定位指令的多重启动，在0~360°的范围内动作的示例如下所示。此外，如果当前位置超出0~360°的范围，则返回0°。

在之前的指令输出变量“Active(控制中)”为TRUE的状态下进行多重启动。只有1个指令可执行单轴动作的指令多重启动。

本例中，将BufferMode(缓存模式选择)设为等待，使之多重启动。



如果指定原点(0°)、90°、120°、290°等4个任意位置，则移动到该位置。此时的旋转方向表现为较近方向。并且，移动速度为250°/s。

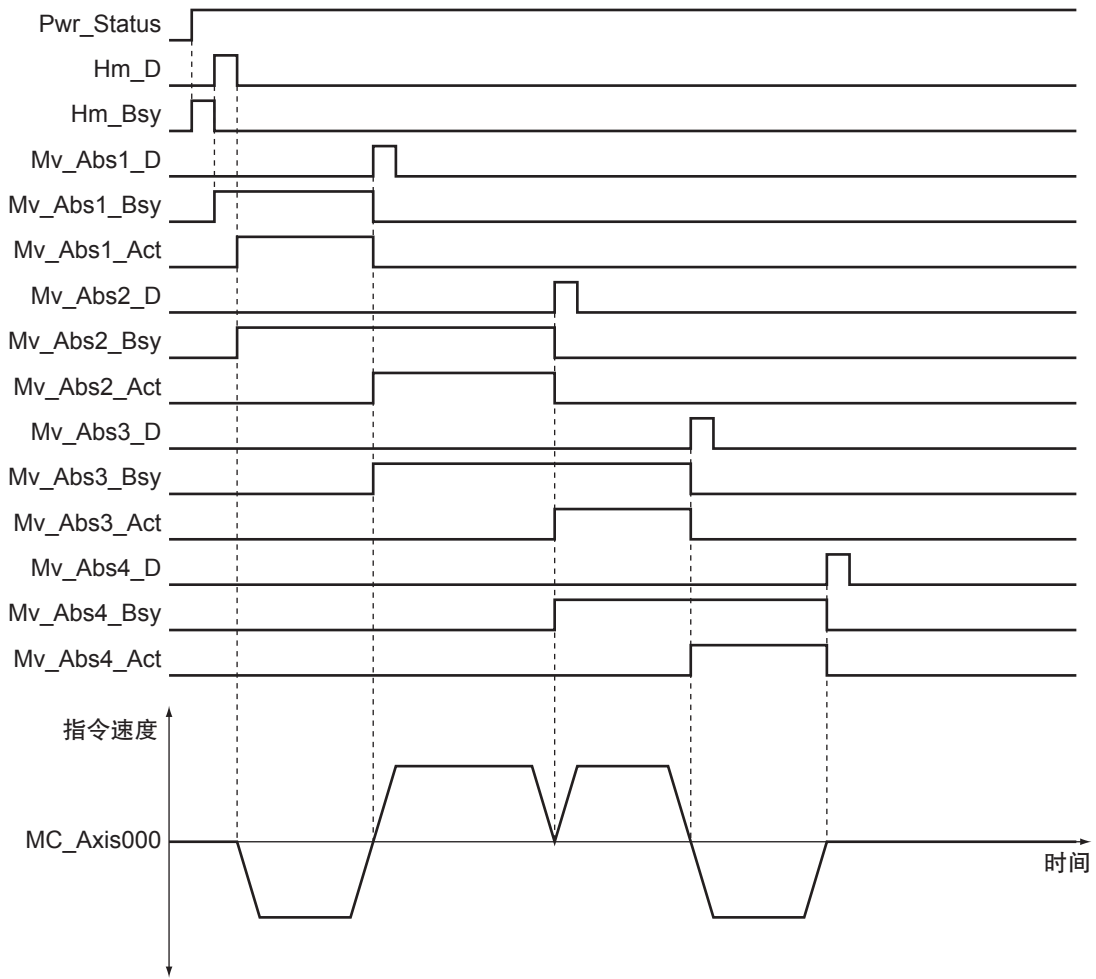
在示例程序中，按照指定为附近定位为290°→90°→120°→原点(0°)。

梯形图

● 主要变量

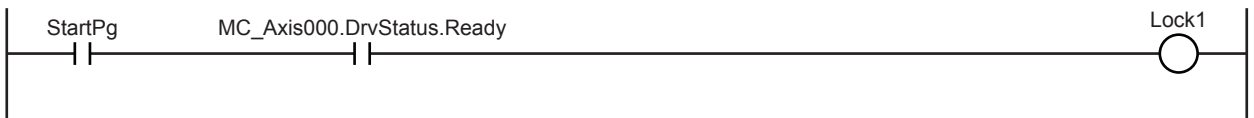
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	轴1的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴1发生轻度故障等级的异常时变为TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴1为原点确定状态时变为TRUE。
Pwr_Status	BOOL	FALSE	分配给MC_Power的实例PWR的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则进入伺服ON状态。

● 时序图

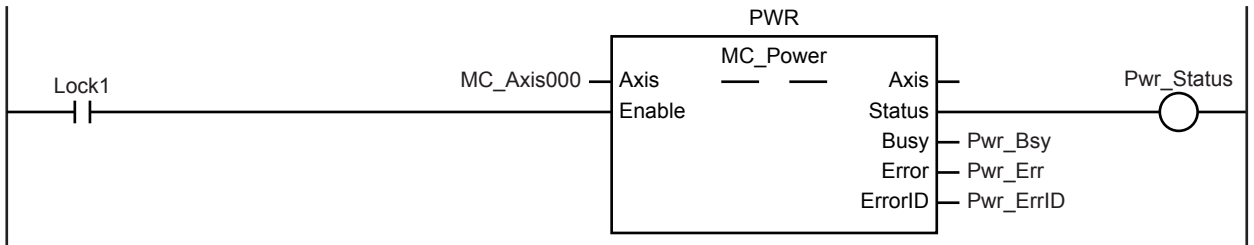


● 示例程序

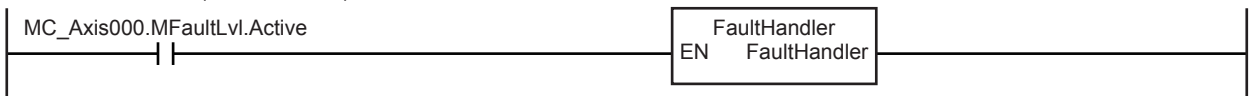
触点StartPg变为TRUE时，确认伺服驱动器处于伺服准备就绪状态



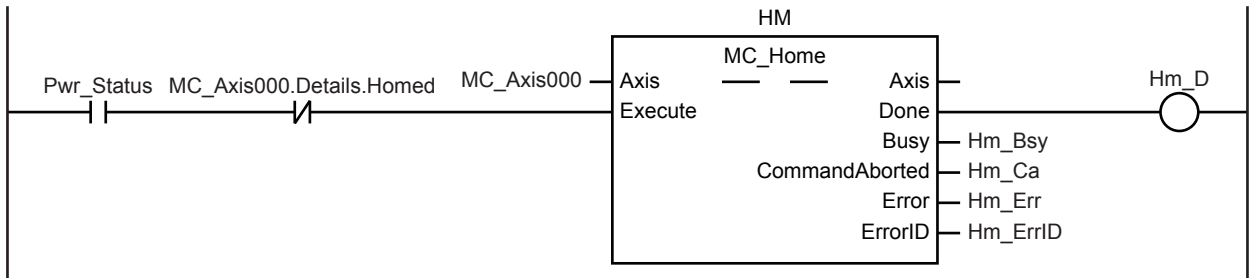
如果处于伺服准备就绪状态，则设为伺服ON状态



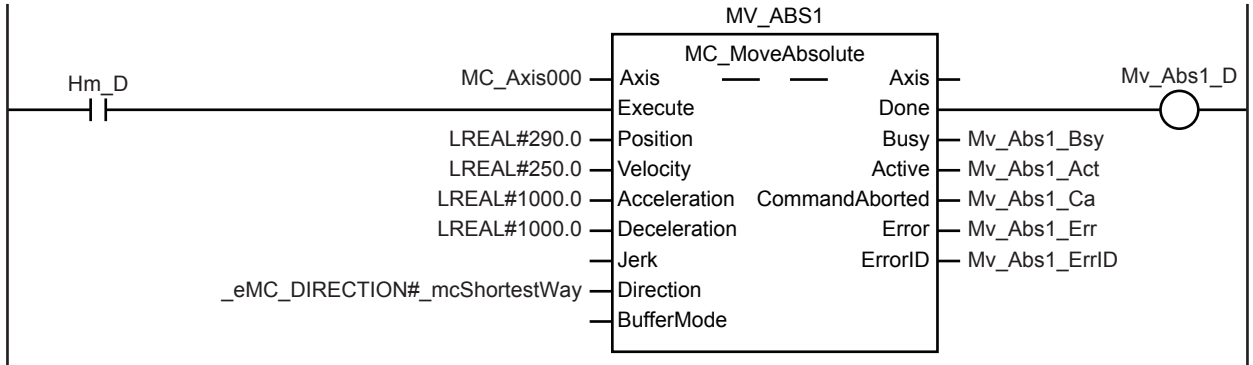
轴1发生轻度故障程度的异常时，执行设备的异常处理(FaultHandler)。发生异常时的处理(FaultHandler)由客户根据装置进行编程



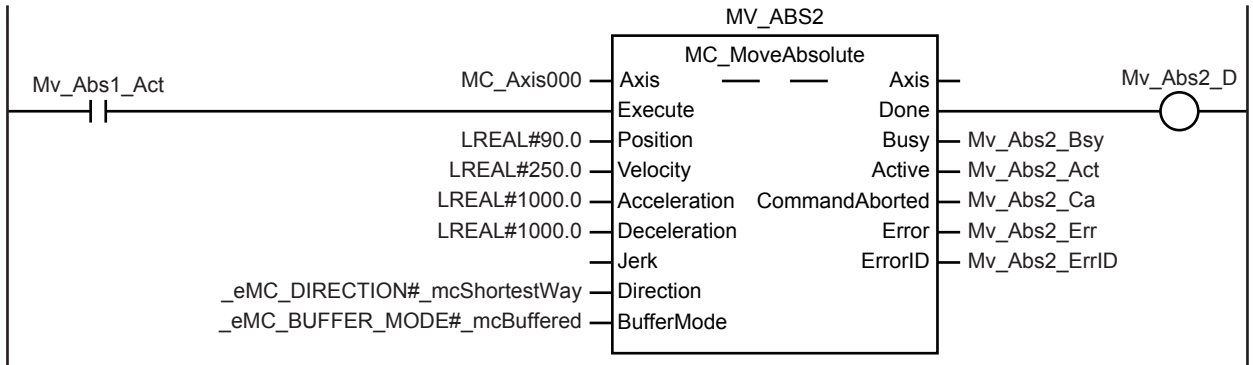
处于伺服ON状态、且原点未确定时，执行原点复位



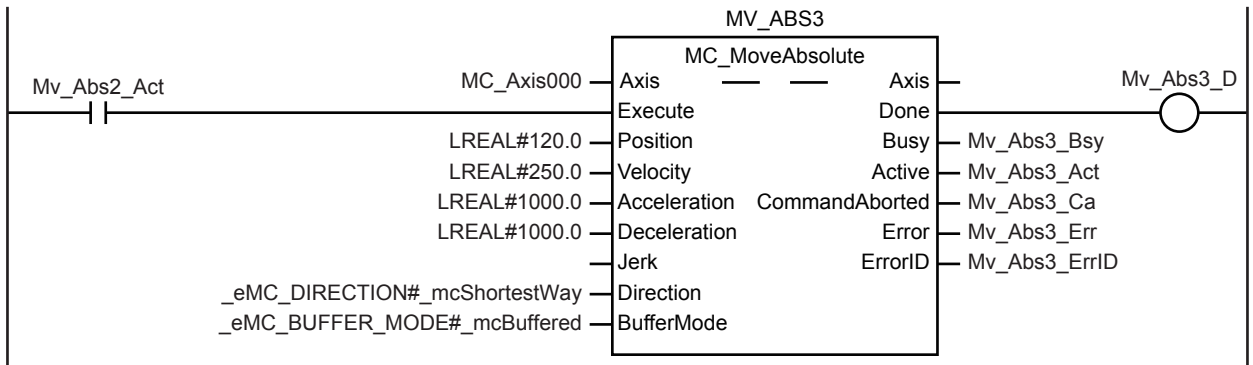
确定原点后，执行绝对值定位(1)，移动至290.0°。动作方向指定为附近



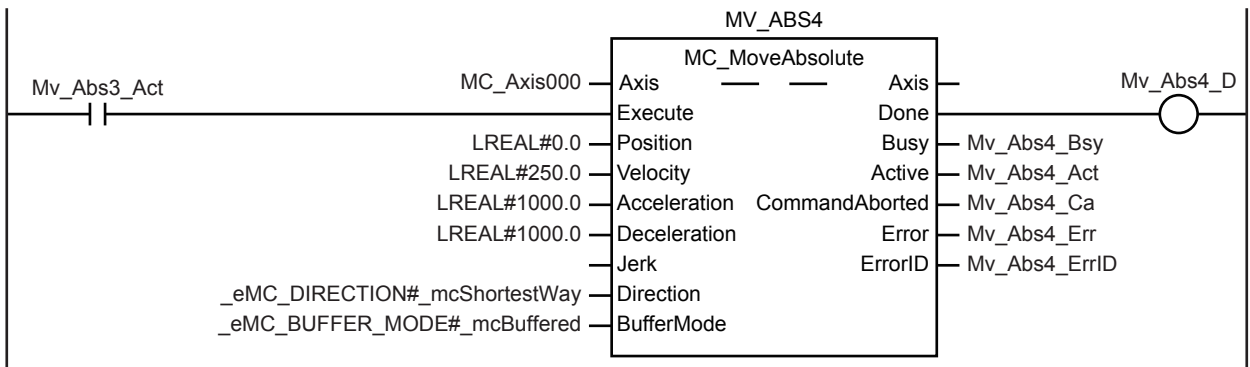
多重启动绝对值定位(2)，从290.0°移动至90.0°。动作方向指定为附近



多重启动绝对值定位(3)，从90.0°移动至120.0°。动作方向指定为附近



多重启动绝对值定位(4)，从120.0°移动至0.0°。动作方向指定为附近

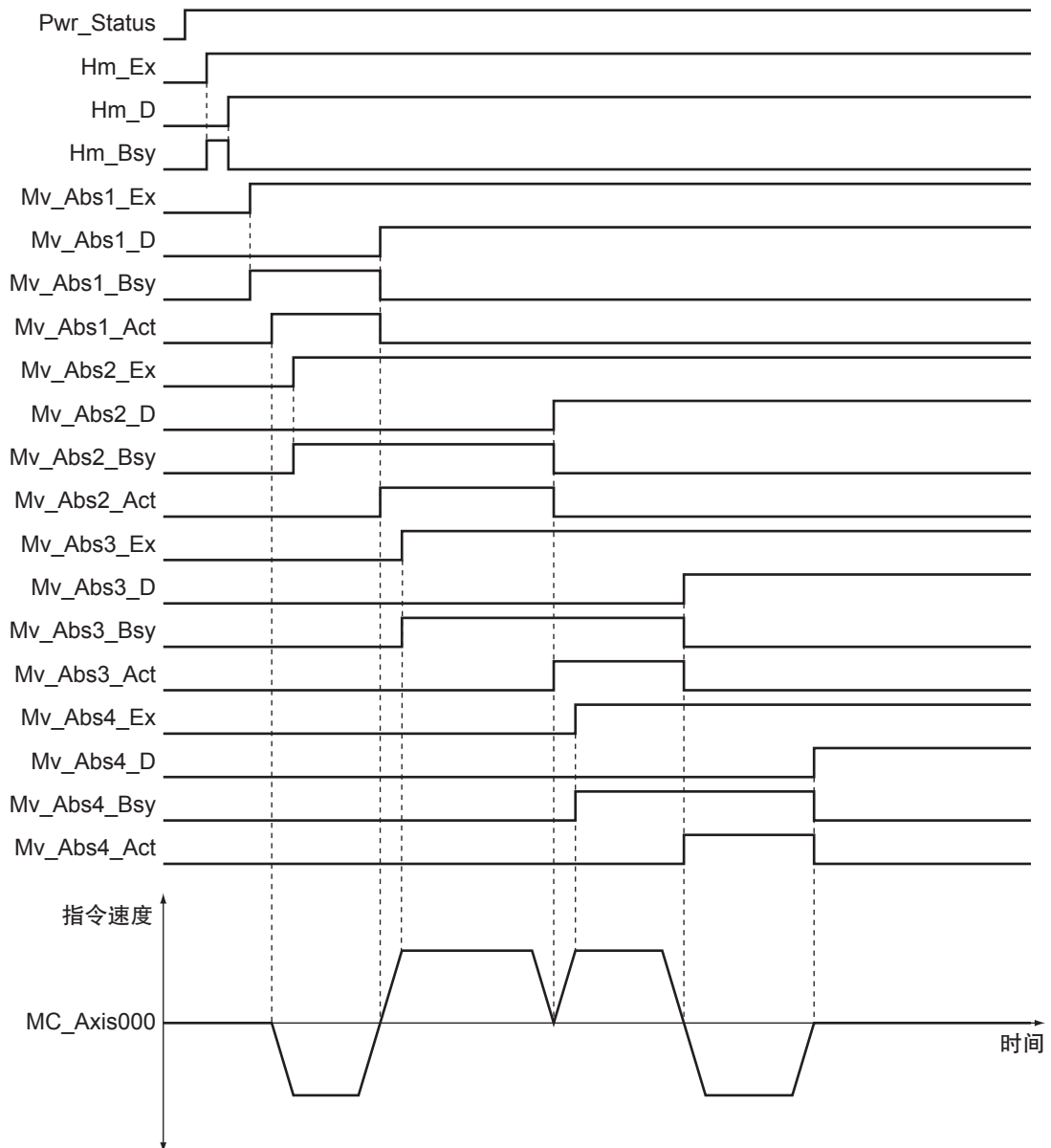


结构文本(ST)

● 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为原点确定状态时变为TRUE。
Pwr_Status	BOOL	FALSE	分配给MC_Power的实例PWR的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则进入伺服ON状态。
Hm_Ex	BOOL	FALSE	该变量由FALSE→TRUE时，启动MC_Home的实例HM。
Mv_Abs1_Ex	BOOL	FALSE	该变量由FALSE→TRUE时，启动MC_MoveAbsolute的实例MV_ABS1。
Mv_Abs2_Ex	BOOL	FALSE	该变量由FALSE→TRUE时，启动MC_MoveAbsolute的实例MV_ABS2。
Mv_Abs3_Ex	BOOL	FALSE	该变量由FALSE→TRUE时，启动MC_MoveAbsolute的实例MV_ABS3。
Mv_Abs4_Ex	BOOL	FALSE	该变量由FALSE→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;

//确认StartPg为TRUE，且伺服驱动器处于伺服准备就绪状态，
//则设为伺服ON状态。
//如果未处于伺服准备就绪状态，则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) 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)由客户根据装置进行编程。
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

指定自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	在上升沿开始指令。
Distance	移动距离	LREAL	负数、正数、“0”	0	指定自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	缓存模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 ^{*3}	指定多重启动运动指令时的动作。 0: 中断 1: 等待 2: 以低速合并 3: 以前一个速度合并 4: 以后一个速度合并 5: 以高速合并

*1. 关于指令单位，请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*2. 请务必设定目标速度。不作设定就执行动作，将发生异常。

*3. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

*1. 请参见“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	定位完成后	<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	开始指令后	<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"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 执行MC_Stop指令中，启动本指令时 	<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含有异常因素时	异常已解除时

输入输出变量

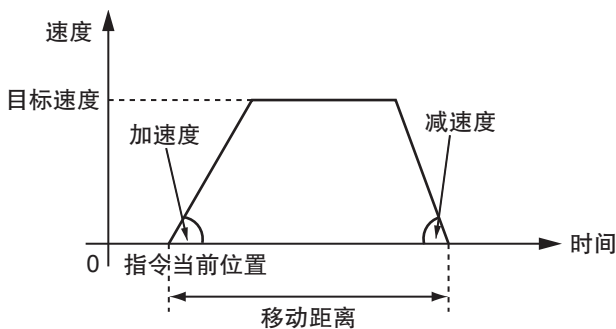
输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	-	指定轴。*1

*1. 请在使用在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[*])。

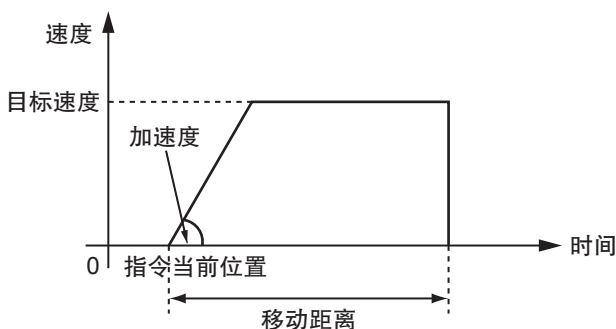
功能说明

- 指定自指令当前位置起的移动距离，进行定位。
- 在Execute(启动)的上升沿，开始相对定位的动作。
- 可在输入变量中指定Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度)、Jerk(跃度)。

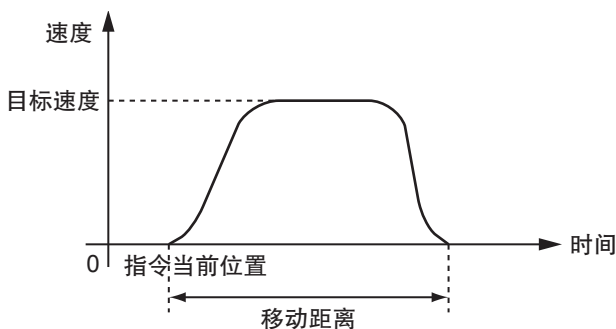
相对定位的动作示例如下所示。



将加速度、或减速度指定为“0”并启动后，不作加减速而达到目标速度。减速度为“0”时的动作示例如下所示。



需要平稳加减速时指定Jerk(跃度)。
指定Jerk(跃度)后的动作示例如下所示。

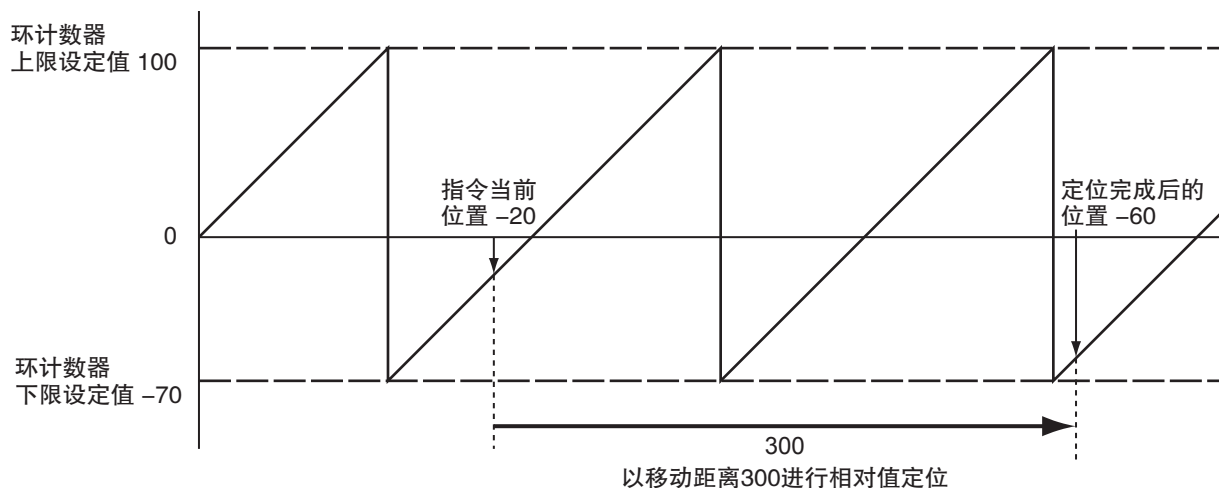


关于Jerk(跃度)的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

指令详情

计数模式为[旋转模式]时，指定超出轴参数的[环计数器下限设定值]~[环计数器上限设定值]的相对距离范围的Distance(移动距离)，可实现环多圈定位。

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



将Distance(移动距离)指定为“0”并启动后，轴不发生移动，但是Done(完成)将变为TRUE。



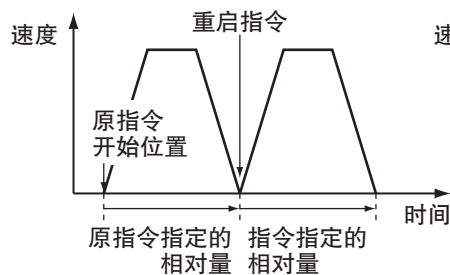
使用注意事项

定位完成前重启相对值定位时，请注意以下几点。

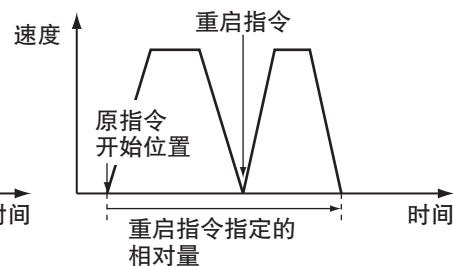
如果在MC功能模块重启前定位已完成，则表现为常规的指令启动。

- 常规指令启动时，在以指令启动时的位置为基准的相对位置进行定位。
- 重启指令时，在以原指令启动时的位置为基准的相对位置进行定位。

常规指令启动



重启指令

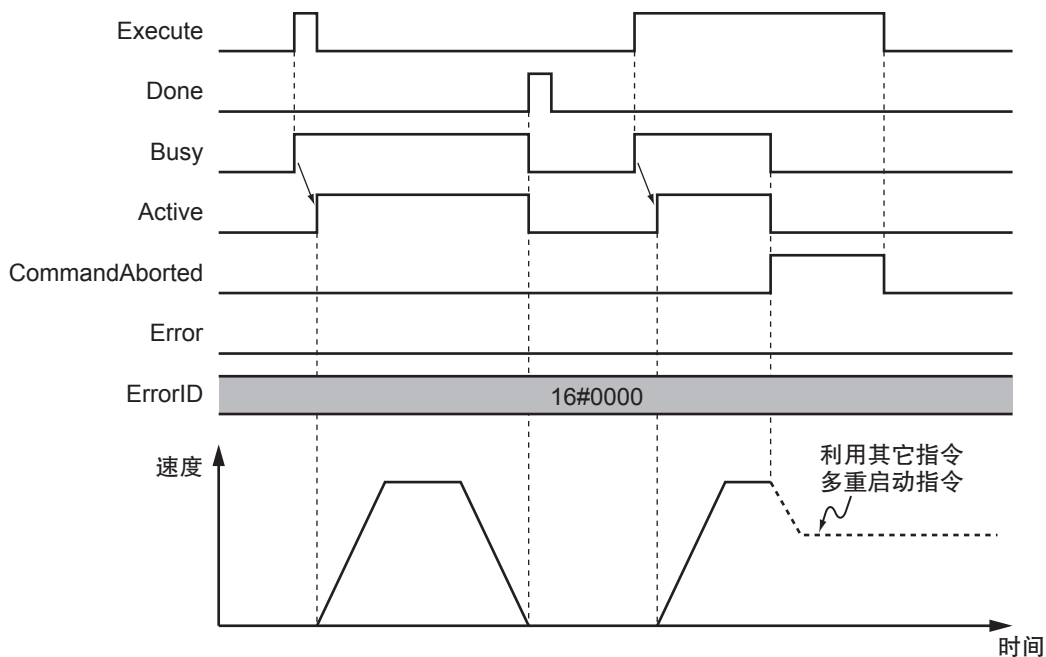


● 到位检查

根据轴参数的[到位宽度]、[到位检查时间]的设定，进行本指令的到位检查。

时序图

- 在启动Execute(启动)的同时, Busy(执行中)变为TRUE。在下一个周期Active(控制中)变为TRUE。
- 达到Distance(移动距离)、定位完成时, Done(完成)变为TRUE。
- 利用其它指令中止本指令时, CommandAborted(执行中断)变为TRUE, Busy(执行中)、Active(控制中)变为FALSE。

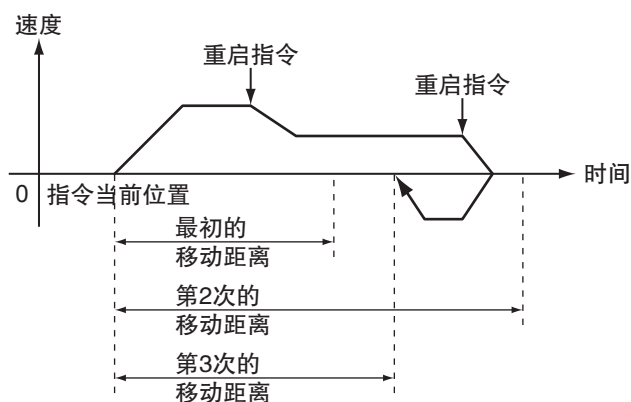


重启运动指令

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

重启运动指令可变更的输入变量有Distance(移动距离)、Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度)。

重启时的Distance(移动距离)的起点, 并非重启时的指令当前位置, 而是最初启动本指令的指令当前位置。在变更Distance(移动距离)和Velocity(目标速度)的同时两次重启运动指令时的动作示例如下所示。

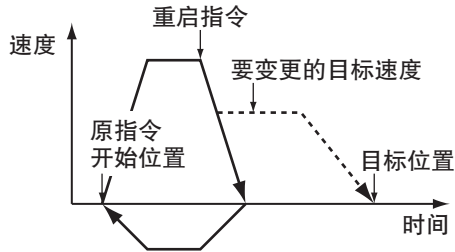




使用注意事项

即使需要变更Distance(移动距离)以外的输入参数时,也必须在Distance(移动距离)中代入与原指令相同的数值,然后重启指令。

例如,如果将Distance(移动距离)的值变更为“0”并重启指令,则将初次启动时的Distance(移动距离)设为“0”并重启指令。因此返回到原来的开始位置。如下图所示。



重启运动指令的详情,请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

多重启动运动指令

多重启动运动指令的详情,请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动

对当前正在执行的指令启动本指令,可切换或缓存到本指令。

各轴可缓存到1个。

多重启动运动指令时的Distance(移动距离)的起点为,启动本指令,Active(控制中)变为TRUE时的指令当前位置。

多重启动时的本指令的动作由BufferMode(缓存模式选择)指定。

缓存模式选择	说明
中断	立即中止当前正在执行的指令,切换为本指令。 轴的动作方向因指令切换而反转时,根据轴参数中的“反转时动作”进行反转。
等待	当前正在执行的指令正常完成后,已缓存的本指令自动启动。
合并	以当前正在执行的指令到达目标位置时的速度(中继速度)为启动速度,连续使已缓存的本指令动作。变更当前正在执行的指令的动作,确保以中继速度到达目标位置。中继速度的指定方法分为如下4种。
以低速合并	当前正在执行的目标速度与已缓存的目标速度中,以速度较低者为中继速度。
以前一个速度合并	以当前正在执行的目标速度为中继速度。
以后一个速度合并	以已缓存的本指令的目标速度为中继速度。
以高速合并	当前正在执行的目标速度与已缓存的目标速度中,以速度较高者为中继速度。

关于BufferMode(缓存模式选择),请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

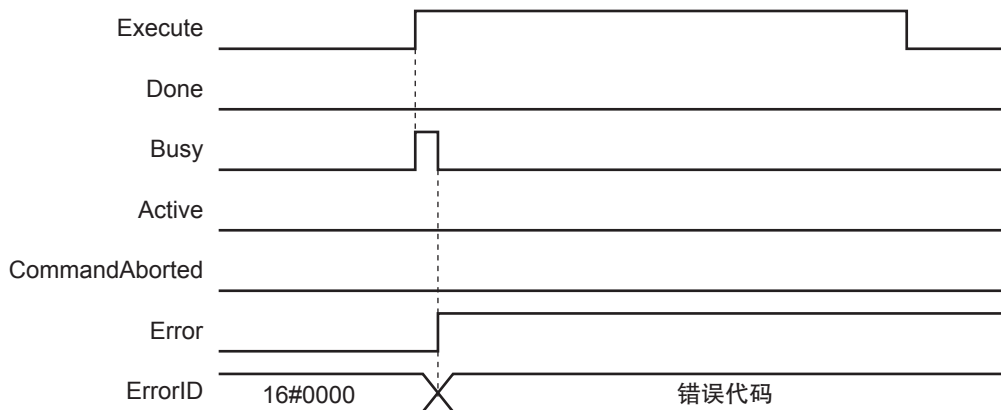
● 本指令执行中的其它指令启动

对本指令进行多重启动运动指令时,可选择中断、等待、合并。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 [□ “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	在上升沿开始指令。
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: 指定为正方向 2: 指定为负方向 3: 指定为当前方向
BufferMode	缓存 模式选择	_eMC_BUFFER_ MODE	0: _mcAborting 1: _mcBuffered	0 *2	指定多重启动运动指令时的动作。 0: 中断 1: 等待 (Reserved)
Continuous (Reserved)	持续方法 选择	BOOL	TRUE, FALSE	FALSE	

*1. 关于指令单位，请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*2. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	达到目标速度后变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

*  请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
InVelocity	达到目标速度后	<ul style="list-style-type: none"> • Error变为TRUE时 • CommandAborted变为TRUE时 • 重启指令变更目标速度后
Busy	Execute的上升沿	<ul style="list-style-type: none"> • Error变为TRUE时 • CommandAborted变为TRUE时
Active	开始指令后	<ul style="list-style-type: none"> • Error变为TRUE时 • CommandAborted变为TRUE时
CommandAborted	<ul style="list-style-type: none"> • 利用其它指令多重启动运动指令(中断、等待), 中止本指令时 • 因发生异常, 中止本指令时 • 发生异常过程中, 启动本指令时 • 执行MC_Stop指令中, 启动本指令时 	<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含有异常因素时	异常已解除时

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. 请在使用在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

- 进行基于位置控制的模拟速度控制。
- 在Execute(启动)的上升沿,开始速度控制的动作。

指令详情

下面对指令详细说明。

● Direction(方向选择)

- 通过Direction(方向选择)指定移动方向。
- Direction(方向选择)为“指定为正方向”时正方向移动;为“指定为负方向”时负方向移动。
- Direction(方向选择)为“指定为当前方向”时,动作因轴是否停止而不同。轴已停止时,轴沿着上次的移动方向进行移动。接通电源或重启电源时正方向移动。
多重启动运动指令使轴移动的过程中启动本指令时,沿当前移动的方向移动。



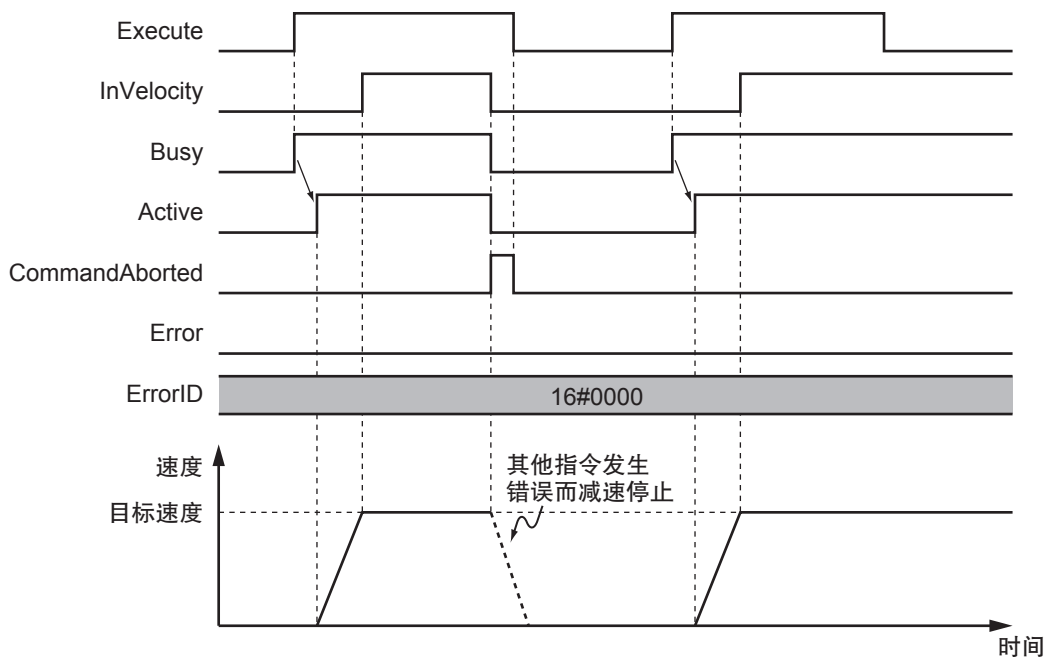
使用注意事项

Direction(方向选择)选择“3: _mcCurrentDirection(指定为当前方向)”时,沿着前一动作的指令方向进行动作。因此,根据指令组合的不同,可能与前一动作的运动指令的输入发出指令的方向不一致。

使用“3: _mcCurrentDirection(指定为当前方向)”时,请通过轴变量的Dir.Posi(正方向指令中)、以及Dir.Nega(负方向指令中)确认当前方向。

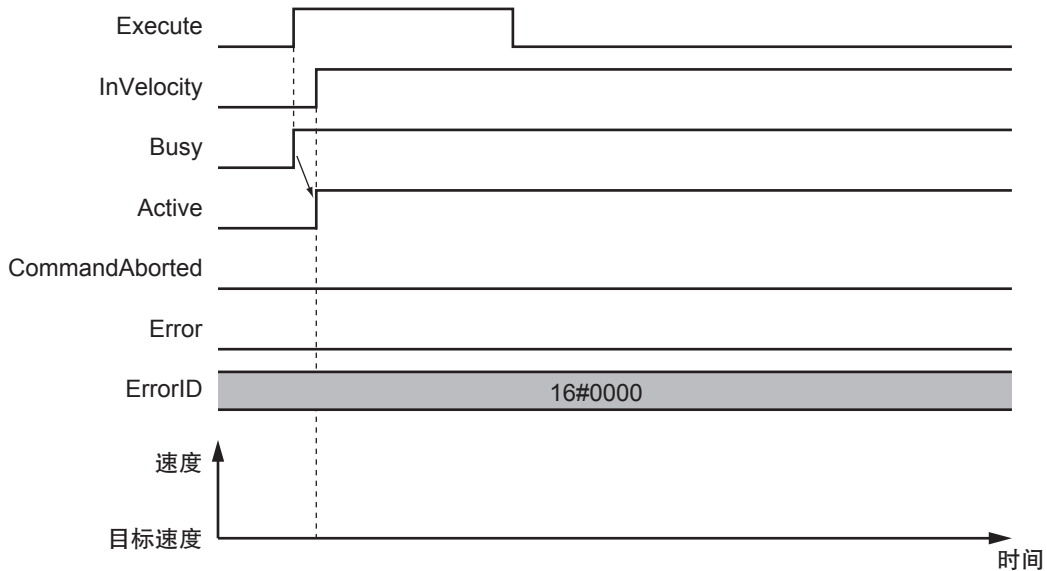
时序图

- 在启动Execute(启动)的同时, Busy(执行中)变为TRUE。在下一个周期Active(控制中)变为TRUE。
- 达到Velocity(目标速度)时, InVelocity(达到目标速度)变为TRUE。
- 利用其它指令中止本指令时, 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。

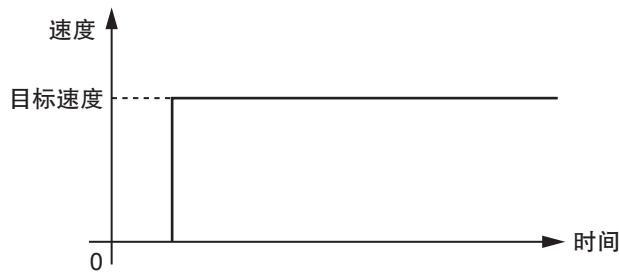


InVelocity(达到目标速度)是表示对于启动本指令和重启运动指令达到等速的输出。因此, InVelocity(达到目标速度)变为TRUE后, 即使利用超调来变更速度, InVelocity(达到目标速度)也不会变为FALSE。并且, 在InVelocity(达到目标速度)变为TRUE之前已变更超调时, 如果已达到变更目标速度, 则InVelocity(达到目标速度)变为TRUE。

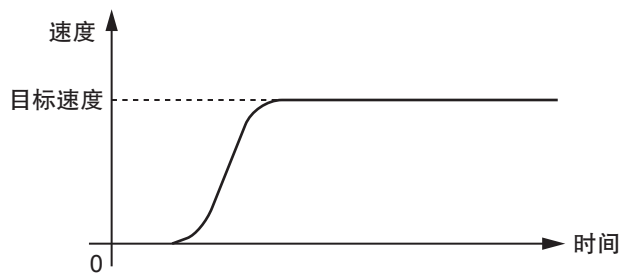
可在输入变量中指定Acceleration(加速度)、Deceleration(减速度)、Jerk(跃度)。
 将Velocity(目标速度)指定为“0”并启动时，轴不会移动，但处于连续动作中。
 将Velocity(目标速度)设为“0”并启动时的动作示例如下所示。



将加速度、或减速度指定为“0”并启动后，不作加减速而达到目标速度。
 加速度为“0”时的动作示例如下所示。



需要平稳加减速时指定Jerk(跃度)。
 指定Jerk(跃度)后的动作示例如下所示。



关于Jerk(跃度)的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

重启运动指令

在连续动作中变更输入参数，再次将Execute(启动)设为TRUE，可变更本指令的动作。
 重启运动指令可变更的输入变量有Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度)。
 重启运动指令变更Velocity(目标速度)时，InVelocity(达到目标速度)针对重启而设定的新目标速度进行动作。

重启运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

多重启动运动指令

多重启动运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动

对当前正在执行的指令启动本指令，可切换或缓存到本指令。
 各轴可缓存到1个。
 多重启动时的本指令的动作由BufferMode(缓存模式选择)指定。

缓存模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 轴的动作方向因指令切换而反转时，根据轴参数中的“反转时动作”进行反转。
等待	当前正在执行的指令正常完成后，已缓存的本指令自动启动。

关于BufferMode(缓存模式选择)，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

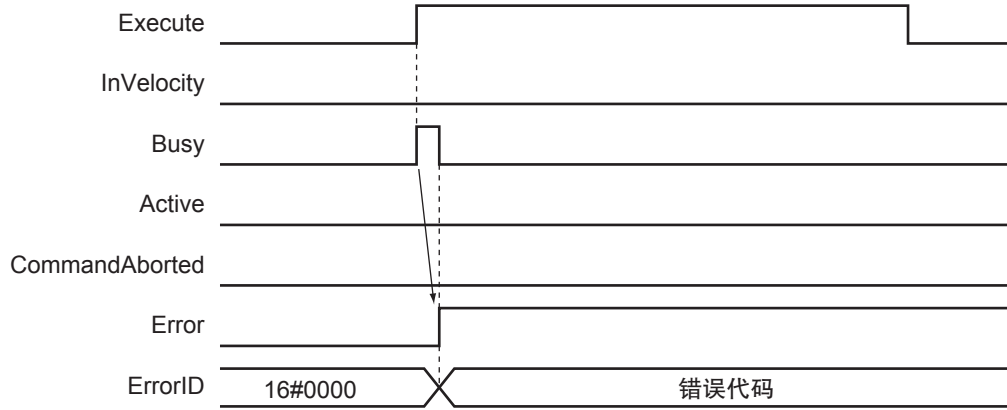
● 本指令执行中的其它指令启动

只有对其他指令的BufferMode(缓存模式选择)指定中断和等待时，才能在执行本指令时，利用其它指令多重启动运动指令。
 指定等待时，如果本指令的输出变量“InVelocity(达到目标速度)”变为TRUE，则开始多重启动指令的动作。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 □ “A-1 错误代码一览(P.A-2)”。

示例程序

下面，对转盘或离心分离器之类的速度控制中的示例程序进行说明。

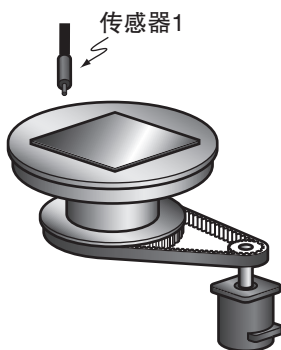
参数设定

该示例程序中使用的最少的必要设定如下所示。

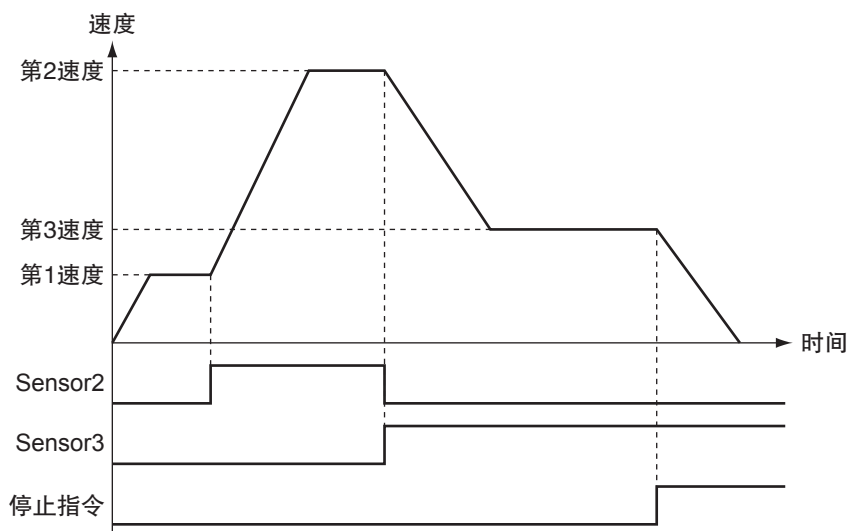
- 轴参数的设定
轴种类

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

将对药液投入进行检测的传感器1设为“ON”时，轴1开始速度控制。

2 变更为第2速度

将Sensor2(触点)设为“ON”时，设定为超调500%，速度得以变更。

3 变更为第3速度

将Sensor3(触点)设为“ON”时，设定为超调200%，速度得以变更。

将Sensor2和Sensor3同时设为“ON”时，则为超调200%。

4 速度控制的停止

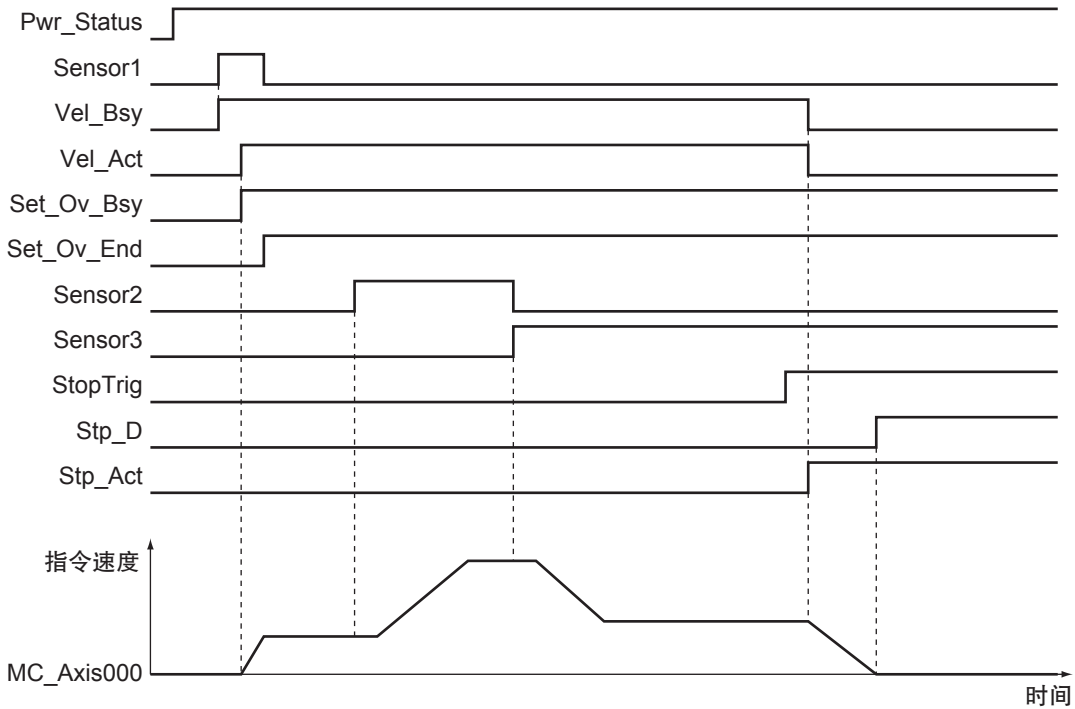
将停止指令(StopTrig)设为“ON”时，停止减速。

梯形图

● 主要变量

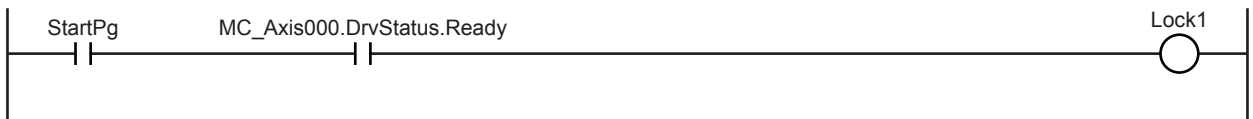
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	轴1的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴1发生轻度故障等级的异常时变为TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴1为原点确定状态时变为TRUE。
Pwr_Status	BOOL	FALSE	分配给MC_Power的实例PWR的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则进入伺服ON状态。
Vel_Act	BOOL	FALSE	分配给MC_MoveVelocity的实例VEL的输出Active的变量。实例VEL处于控制中时，该变量变为TRUE。
Set_Ov_Velfct	LREAL	0	超调值。
StopTrig	BOOL	FALSE	该变量为TRUE时，执行MC_Stop。
Sensor1	BOOL	FALSE	检测到药液投入时变为TRUE。 轴1处于伺服ON状态时，启动MC_MoveVelocity(速度控制)。
Sensor2	BOOL	FALSE	该变量为TRUE时，将超调值设定为500%。该变量变为TRUE时保持状态；Sensor3变为TRUE时，则该变量变为FALSE。
Sensor3	BOOL	FALSE	该变量为TRUE时，将超调值设定为200%。该变量变为TRUE时保持状态。

● 时序图

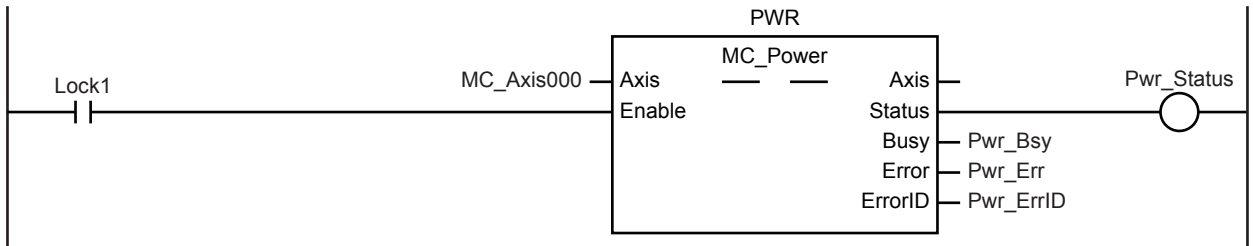


● 示例程序

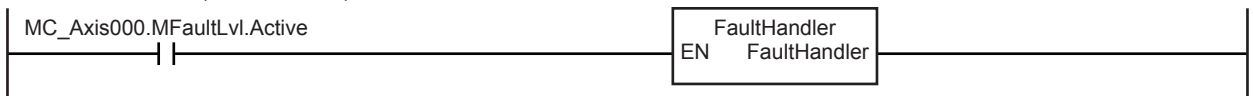
触点StartPg变为TRUE时，确认伺服驱动器处于伺服准备就绪状态



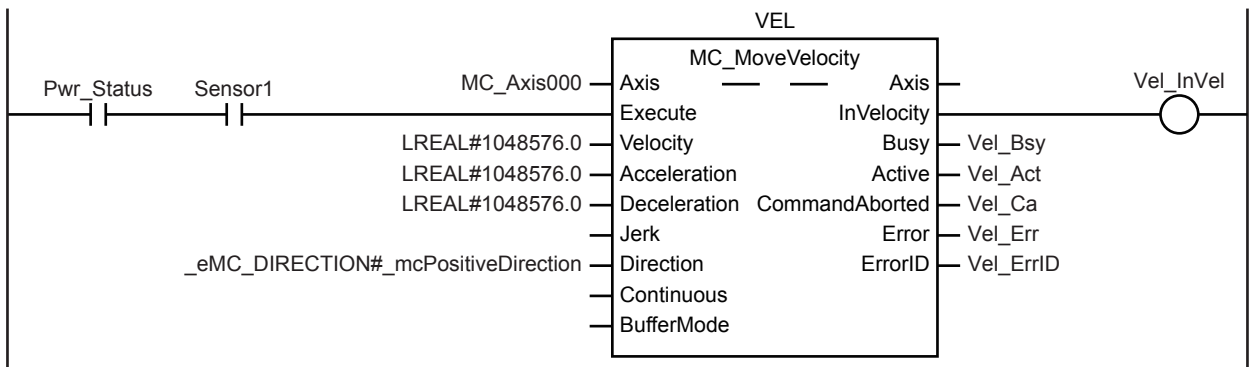
如果处于伺服准备就绪状态，则设为伺服ON状态



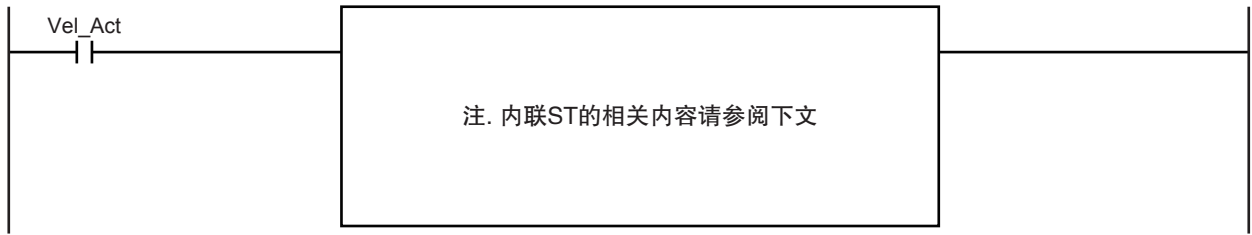
轴1发生轻度故障程度的异常时，执行设备的异常处理(FaultHandler)。发生异常时的处理(FaultHandler)由客户根据装置进行编程



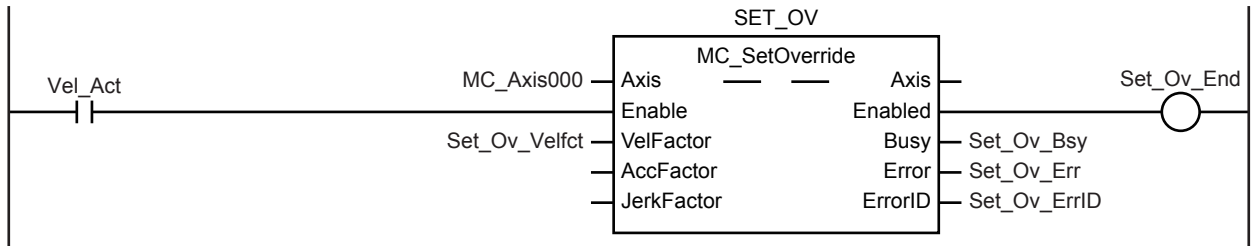
在可运行状态下，Sensor1为TRUE时，启动MC_MoveVelocity(速度控制)指令



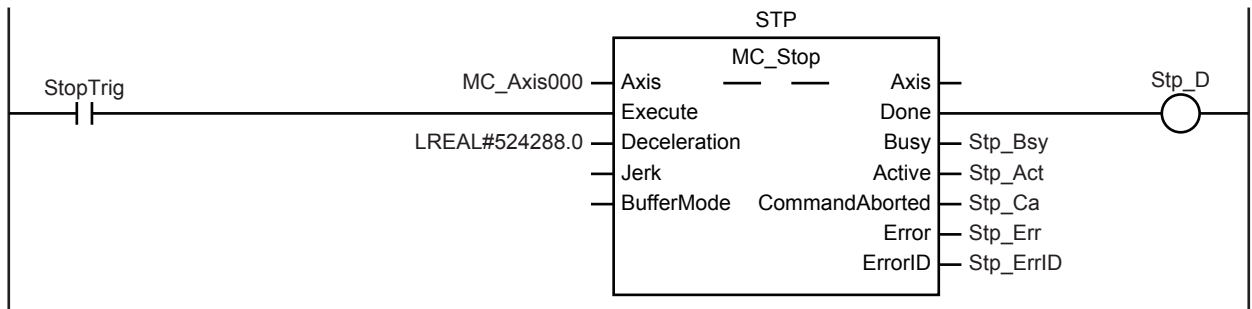
在MC_MoveVelocity(速度控制)指令控制中，根据触点Sensor2和触点Sensor3的ON/OFF，变更超调值



在MC_MoveVelocity(速度控制)指令的控制中，启动MC_SetOverride(超调值设定)指令



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



内联ST的内容

```

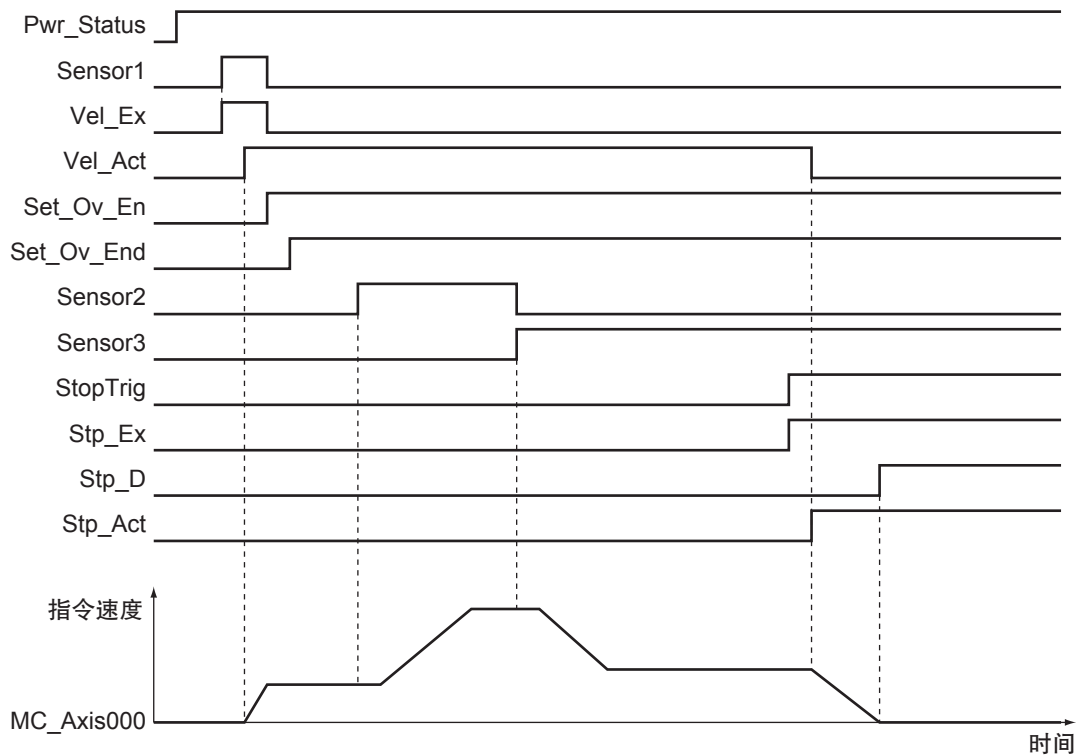
IF (Sensor2=FALSE) AND (Sensor3=FALSE) THEN
    Set_Ov_Velfct := LREAL#100.0;
ELSIF (Sensor2=TRUE) AND (Sensor3=FALSE) THEN
    Set_Ov_Velfct := LREAL#500.0;
ELSIF (Sensor2=FALSE) AND (Sensor3=TRUE) THEN
    Set_Ov_Velfct := LREAL#200.0;
ELSE
    Set_Ov_Velfct := LREAL#200.0;
END_IF;
    
```

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为原点确定状态时变为TRUE。
Pwr_Status	BOOL	FALSE	分配给MC_Power的实例PWR的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则进入伺服ON状态。
Vel_Act	BOOL	FALSE	分配给MC_MoveVelocity的实例VEL的输出Active的变量。实例VEL处于控制中时，该变量变为TRUE。
Set_Ov_Velfct	LREAL	0	超调值。
StopTrig	BOOL	FALSE	该变量为TRUE时，执行MC_Stop。
Sensor1	BOOL	FALSE	检测到药液投入时变为TRUE。 轴1处于伺服ON状态时，启动MC_MoveVelocity(速度控制)指令。
Sensor2	BOOL	FALSE	该变量为TRUE时，将超调值设定为500%。该变量变为TRUE时保持状态；Sensor3变为TRUE时，则该变量变为FALSE。
Sensor3	BOOL	FALSE	该变量为TRUE时，将超调值设定为200%。该变量变为TRUE时保持状态。
Vel_Ex	BOOL	FALSE	该变量由FALSE→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	该变量由FALSE→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;

//StartPg为TRUE时，确认伺服驱动器处于伺服准备就绪状态，
//将轴1设为伺服ON状态
//如果未处于伺服准备就绪状态，则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
  Pwr_En:=TRUE;
ELSE
  Pwr_En:=FALSE;
END_IF;

```



```

// 轴1发生轻度故障后，执行异常时处理FaultHandler。
//发生异常时的处理(FaultHandler)由客户根据装置进行编程。
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;
// 轴1处于伺服ON状态，且触点Sensor1为TRUE时，启动MC_MoveVelocity。
IF (Pwr_Status=TRUE) AND (Sensor1=TRUE) THEN
    Vel_Ex := TRUE;
END_IF;

// 在MC_MoveVelocity控制中，根据触点Sensor2和触点Sensor3的ON/OFF，变更超调值。
IF Vel_Act=TRUE THEN
    IF (Sensor2=FALSE) AND (Sensor3=FALSE) THEN
        Set_Ov_Velfet := LREAL#100.0;
    ELSEIF (Sensor2=TRUE) AND (Sensor3=FALSE) THEN
        Set_Ov_Velfet := LREAL#500.0;
    ELSEIF (Sensor2=FALSE) AND (Sensor3=TRUE) THEN
        Set_Ov_Velfet := LREAL#200.0;
    ELSE
        Set_Ov_Velfet := LREAL#200.0;
    END_IF;
END_IF;

// MC_MoveVelocity控制中启动MC_SetOverride。
IF Vel_Act=TRUE THEN
    Set_Ov_En := TRUE;
END_IF;

// StopTrig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”设为目标位置进行定位，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	在上升沿开始指令
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	缓存 模式选择	_eMC_BUFFER_ MODE	0: _mcAborting 1: _mcBuffered	0 ^{*3}	指定多重启动运动指令时的动作。 0: 中断 1: 等待

*1. 请务必设定目标速度。不作设定就执行动作，将发生异常。

*2. 关于指令单位，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*3. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

* □ 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	定位完成后	<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	轴开始移动后	<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"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 执行MC_Stop指令中，启动本指令时 	<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含有异常因素时	异常已解除时

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. 请在使用在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

- 向原点位置移动。
- 可在输入变量中指定Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度)、Jerk(跃度)。



使用注意事项

在原点已确定的状态下执行MC_MoveZeroPosition(高速原点复位)指令。如果原点未确定，则发生“原点未确定状态下的指令启动异常(错误代码：5466 Hex)”。

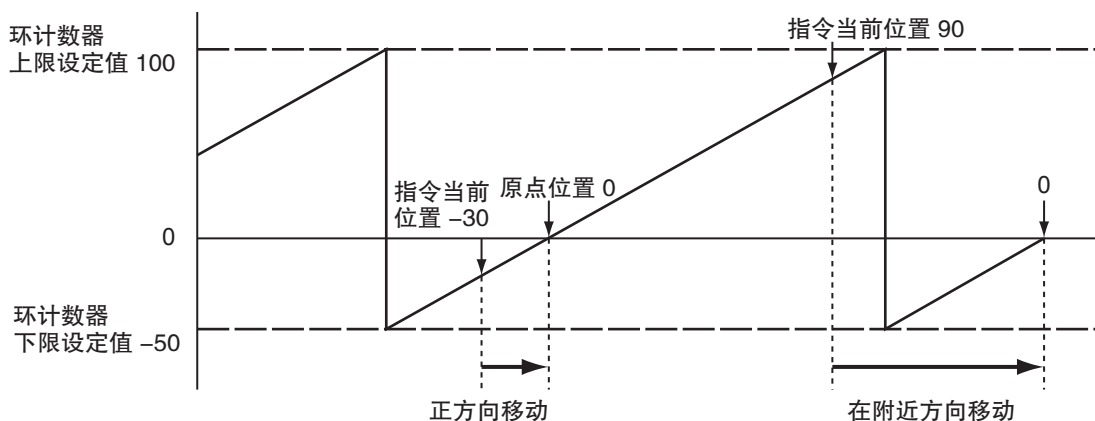
指令详情

下面对指令详细说明。

● 方向指定

计数模式为[旋转模式]时，迎着到原点位置的距离较短的方向进行定位(附近)。

指令当前位置相应的本指令的动作示例如下所示。



使用注意事项

在轴参数的[环计数器上限设定值]和[环计数器下限设定值]的范围外存在原点的状态下启动本指令时，发生“超过目标位置环计数器范围(错误代码：549C Hex)”。

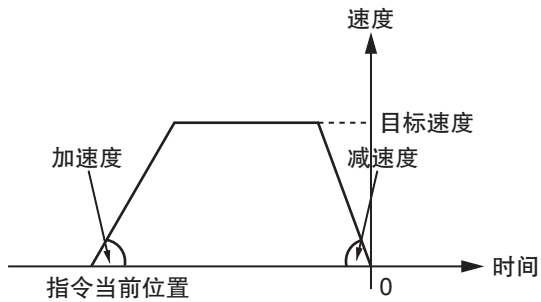
● 超调

超调对本指令生效。

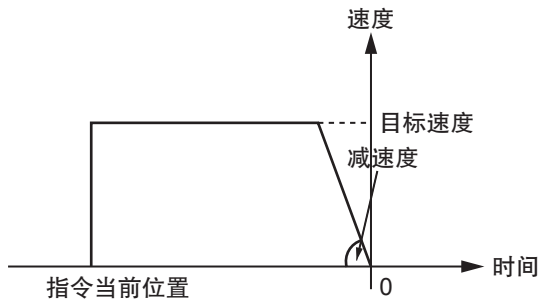
● 到位检查

根据轴参数的[到位宽度]、[到位检查时间]的设定，进行本指令的到位检查。

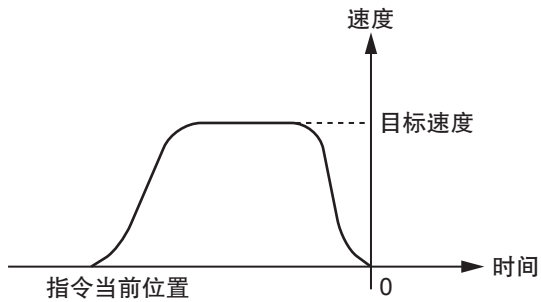
● 动作示例



将加速度、或减速度指定为“0”并启动后，不作加减速而达到目标速度。
加速度为“0”时的动作示例如下所示。



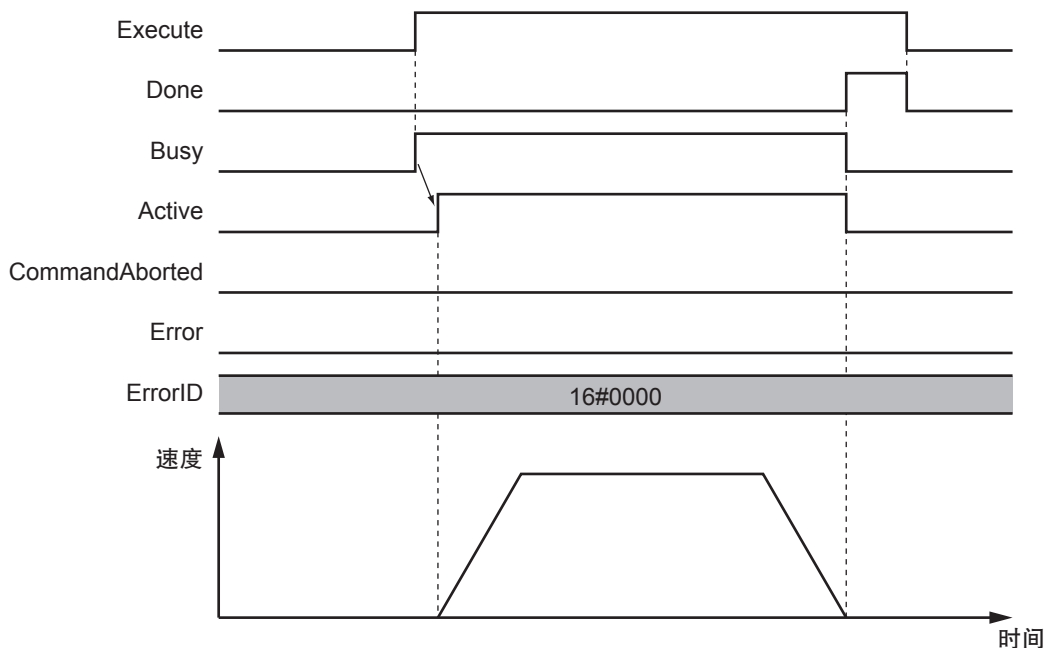
需要平稳加减速时指定Jerk(跃度)。
指定Jerk(跃度)后的动作示例如下所示。



关于Jerk(跃度)的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

时序图

执行MC_MoveZeroPosition(高速原点复位)指令后的时序图如下所示。



指令的中止

本指令中止，CommandAborted(执行中断)变为TRUE时，也不会变成原点未确定状态。

重启运动指令

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

重启运动指令可变更的输入变量有Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度)。

重启运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

多重启动运动指令

多重启动运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动

对当前正在执行的指令启动本指令，可切换或缓存到本指令。

各轴可缓存到1个。

多重启动时的本指令的动作由BufferMode(缓存模式选择)指定。

缓存模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 轴的动作方向因指令切换而反转时，根据轴参数中的“反转时动作”进行反转。
等待	当前正在执行的指令正常完成后，已缓存的本指令自动启动。

关于BufferMode(缓存模式选择)，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 本指令执行中的其它指令启动

对本指令进行多重启动运动指令时，可选择中断、等待。

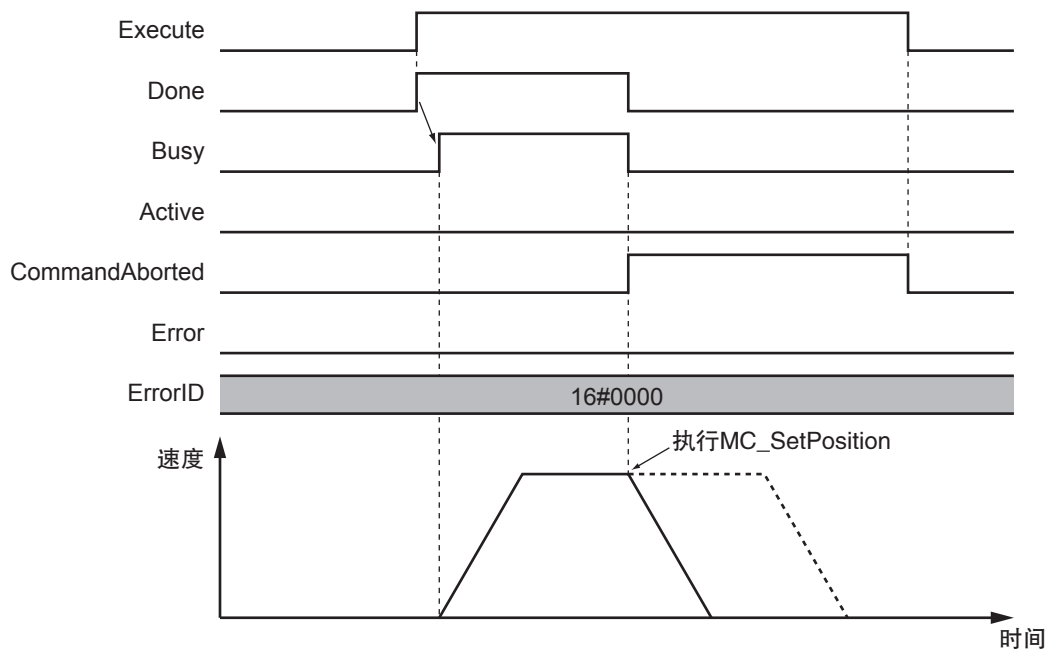
无法实现基于合并的指令多重启动。

在本指令执行中执行MC_SetPosition(当前位置变更)指令后

由于MC_SetPosition(当前位置变更)指令中原点未确定，因此在本指令执行中执行该指令将发生异常。也就是说无法执行。

执行MC_SetPosition(当前位置变更)指令发生异常时，本指令停止减速，CommandAborted(执行中断)变为TRUE。

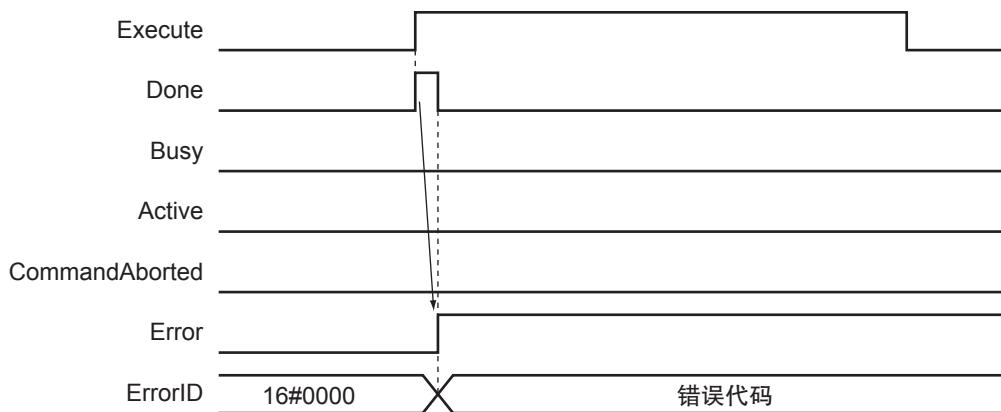
此时，MC_SetPosition(当前位置变更)指令中，输出变量的Error(错误)变为TRUE。



异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 [□ “A-1 错误代码一览\(P.A-2\)”](#)。

MC_MoveFeed

指定自外部输入的中断输入发生位置起的移动距离，进行定位。
可实现中断标准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	中断标准定位	FB	<p>MC_MoveFeed_instance</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	在上升沿开始指令。
WindowOnly	窗口有效	BOOL	TRUE, FALSE	FALSE	指定窗口的有效/无效。
FirstPosition	起始位置	LREAL	负数、正数、“0”	0	指定锁定有效的位置。 单位为[指令单位]。 ^{*1}
LastPosition	终止位置	LREAL	负数、正数、“0”	0	指定锁定无效的位置。 单位为[指令单位]。 ^{*1}
ReferenceType	位置类型选择	_eMC_ REFERENCE_ TYPE	1: _mcFeedback	1 * ²	指定位置类型。 1: 反馈位置(同一任务周期 ^{*3} 下的取值)
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: 指定为正方向 1: 指定为附近 2: 指定为负方向 3: 指定为当前方向 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
FeedDistance	标准距离	LREAL	负数、正数、“0”	0	指定中断输入后的移动距离。 沿着与中断输入前动作方向相 同的方向使之按照标准距离动作时, 设为正数; 反方向动作时, 设为 负数。 单位为[指令单位]。 ^{*1}
FeedVelocity	标准速度	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	缓存模式选择	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered	0 * ²	指定多重启动运动指令时的动作。 0: 中断 1: 等待

输入变量	名称	数据类型	有效范围	初始值	内容
ErrorDetect	错误检测选择	BOOL	TRUE, FALSE	FALSE	无中断输入时, 指定是否执行异常检测。 TRUE: 执行异常检测 FALSE: 不执行异常检测

- *1. 关于指令单位, 请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的 “单位转换设定”。
- *2. 有效范围为枚举体的变量, 其实际初始值不是数值, 而是枚举元素。
- *3. 任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先度5)的周期。
- *4. MoveMode中指定 “2:_mcVelocity” 时, 也可以为 “0”。
- *5. MoveMode为[0: 绝对值定位], 计数模式为[线性模式]时, 移动到指定位置。
MoveMode为[1: 相对值定位]时, 移动方向由Position的符号决定。
MoveMode为[2: 速度控制]时, 无论计数器模式如何, 均无法选择 “1:_mcShortestWay” 和 “4:_mcNoDirection”。

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	指令执行完毕时变为TRUE。
InFeed	标准传送中	BOOL	TRUE, FALSE	接收锁定输入, 标准传送中变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时, 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为正常。

- *1. □ 请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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	开始指令后	<ul style="list-style-type: none"> Done变为TRUE时 Error变为TRUE时 CommandAborted变为TRUE时
InFeed	中断输入后，开始标准距离动作时*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"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 执行MC_Stop指令中，启动本指令时 	<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含有异常因素时	异常已解除时

*1. 从中断输入“ON”到InFeed变为TRUE之前，可能延迟数个任务周期。任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先级5)的周期。

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	在触发条件下指定控制器模式时，指定触发的输入变量。

*1. 请使用在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型的用户定义变量。

● _sTRIGGER_REF

结构要素变量	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	在驱动器模式下，指定使用2个锁定功能中的哪一个。 0: 锁定功能1 1: 锁定功能2
InputDrive	驱动触发 输入信号	_eMC_ TRIGGER_ INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	在驱动器模式下指定驱动器的触发信号。 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, 按照绝对值移动、相对值移动或速度控制中的某一移动方法进行移动。
- 采用绝对值移动时, 在Position(目标位置)中设定目标位置; 采用相对值移动时, 在Position(目标位置)中设定目标距离。
无论何种移动方法, 均以Velocity(目标速度)进行移动动作。
- 移动过程中, 在外部输入(中断输入)的上升沿进行相对定位动作。以FeedVelocity(标准速度), 从反馈位置起, 移动FeedDistance(标准距离)指定的标准距离。
- 利用绝对值移动或相对值移动指令进行中断标准传送, 在到达目标位置前未输入中断信号时, 在当初的目标位置停止动作。
无中断输入而停止动作时, 通过ErrorDetect(错误检测选择), 可指定有无异常输出。指定异常输出时, CommandAborted(执行中断)变为TRUE, Busy(执行中)、Active(控制中)变为FALSE。
- 使用中断屏蔽时, 将WindowOnly(窗口有效)设为TRUE, 指定FirstPositon(起始位置)、LastPositon(终止位置)。
通过反馈位置从FirstPositon(起始位置)到LastPosition(终止位置)之间发生的最初中断信号, 执行中断标准定位。



使用注意事项

- 中断后的标准距离动作表现为, 按照FeedDistance(标准距离)指定的距离进行相对移动。对FeedDistance(标准距离)指定正数时, 沿与中断输入前相同的方向进行标准距离动作; 指定负数时, 沿相反方向进行标准距离动作。
- 以标准距离动作反转时的加减速遵从轴参数[反转时动作]的设定。
- 中断标准定位后的位置为溢出/下溢时, 发生中断输入时间异常。发生溢出/下溢后, 即使输入中断也同样发生轴异常。



参考

绝对值移动参阅 □ “MC_MoveAbsolute(P.3-47)”、
 相对值移动参阅 □ “MC_MoveRelative(P.3-73)”、
 速度控制参阅 □ “MC_MoveVelocity(P.3-80)”、
 WindowOnly(窗口有效)参阅 □ “WindowOnly(窗口有效)(P.3-109)”。

● 数据对象的映射

在Mode中选择[驱动器模式]，使用MC_MoveFeed(中断标准定位)指令时，请映射如下对象数据。请在Sysmac Studio的轴基本设置的[高级设置]中进行映射。

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

未进行这些设定时，将发生过程数据对象设定不足(错误代码：3461Hex)错误。

关于数据对象的映射，请参阅 □□ “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-363)”或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。使用NX系列位置接口单元时，请参阅 □□ “NX系列 位置接口单元 用户手册(SBCE-374)”中的“I/O入口映射”。

指令详情

下面对指令详细说明。

● Axis(轴)指定

- 通过Axis(轴)指定锁定对象轴。

● 触发输入条件

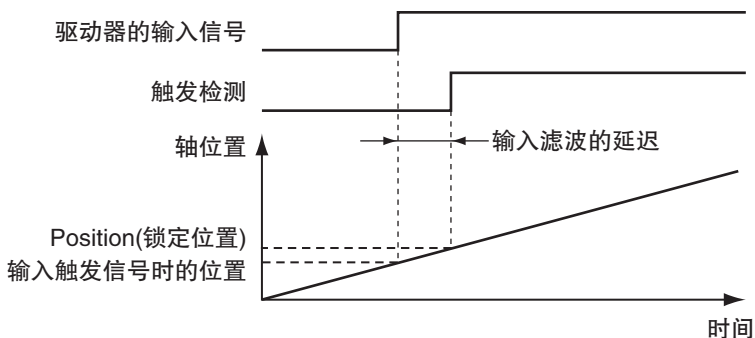
通过TriggerInput(触发输入条件)的Mode、LatchID、InputDrive的指定，选择触发条件。

Mode

- Mode的设定有两种可选，在触发中指定伺服驱动器或NX系列输出单元信号的“驱动器模式”、在触发中指定TriggerVariable(触发变量)的“控制器模式”。
- 将触发信号的上升沿视为发生触发，对执行锁定指令后最初发生触发(FALSE→TRUE)时轴的位置进行记录。
- 本指令为Busy(动作中)中时，即使Execute(启动)变为FALSE，也会将TriggerVariable(触发变量)的变化作为触发进行动作。

驱动器模式

“驱动器模式”是利用伺服驱动器的功能检测触发、获取当前位置，因此，相比“控制器模式”，锁定的当前位置更准确。





使用注意事项

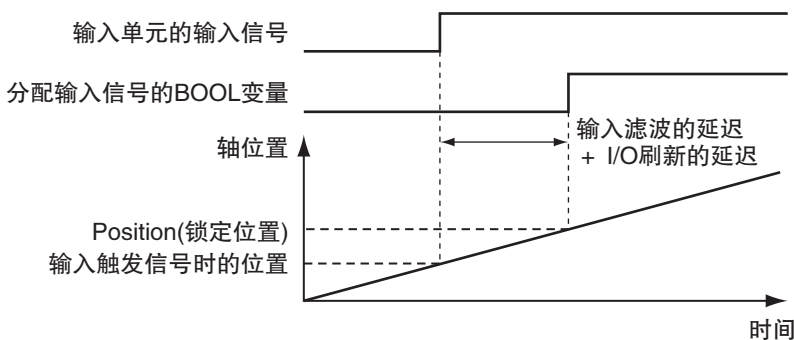
- 使用驱动器模式时，请务必在要使用的LatchID(锁定ID选择)上连接锁定信号。
- 锁定信号的宽度取决于伺服驱动器或NX系列脉冲输出单元的性能等。
- 在InputDrive中选择[驱动器模式]，使用MC_MoveFeed(中断标准定位)指令时，请映射如下对象数据。

锁定功能(60B8Hex)、锁定状态(60B9Hex)、锁定位置1(60BAHex)、锁定位置2(60BCHex)未进行这些设定时，将发生过程数据对象设定不足(错误代码：3461Hex)错误。

关于对象数据的映射，请参阅 □ “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-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

控制器模式

- 在“控制器模式”下，可对触发指定BOOL变量。
- 在TriggerVariable(触发变量)中指定要触发的BOOL变量。
- 与“驱动器模式”相比，在“控制器模式”下，由于I/O刷新的延迟，触发输入信号的BOOL变量的反映会产生较大延迟。



使用注意事项

使用控制器模式时，按照任务周期的间隔进行锁定处理。因此，触发变量必须在任务周期的一个周期内为TRUE。


并且，在触发变量变为TRUE后，MC功能模块执行处理之前，需要任务周期的一个周期的时间。这里的任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先级5)的周期。

LatchID

- 通过LatchID(锁定ID选择)指定使用哪一个。对于1根轴只能使用某一个。
- 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电路1(第1ch)、锁定电路2(第2ch)。

关于LatchID，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

InputDrive

- 作为触发，可选择_mcEncoderMark(Z相)或_mcEXT(外部输入)。
- _mcEncoderMark(Z相)中，将伺服驱动器或NX系列脉冲输出单元的Z相用作触发。
_mcEXT(外部输入)中，将伺服驱动器或NX系列脉冲输出单元中输入的外部触发信号用作触发。
- 欧姆龙制伺服驱动器 1S 系列中，_mcEXT(外部输入)为设定的Ext1 或 Ext2。G5 系列中，为设定的Ext1、Ext2、Ext3三者之一。通过Sysmac Studio进行设定。
驱动器中设定的两个触发可以相同。
- 使用NX系列 脉冲输出单元时，请参阅  “NX系列 位置接口单元 用户手册(SBCE-374)”。

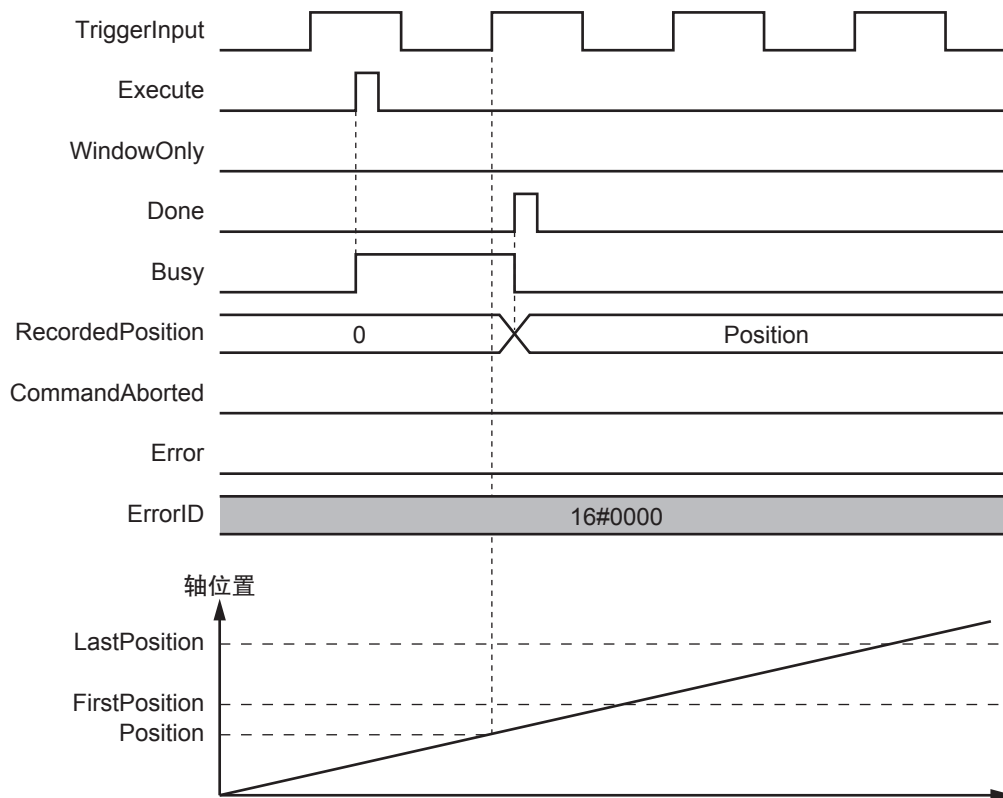
● WindowOnly(窗口有效)

- 在WindowOnly(窗口有效)中，指定窗口的有效/无效。
- 指定Disable(无效)时，在所有轴位置检出触发。
- 指定Enable(有效)时，仅轴位置在FirstPosition(起始位置)和LastPosition(终止位置)的范围内时检出触发。

WindowOnly(窗口有效)的指定不同，动作也不同。如时序图所示。

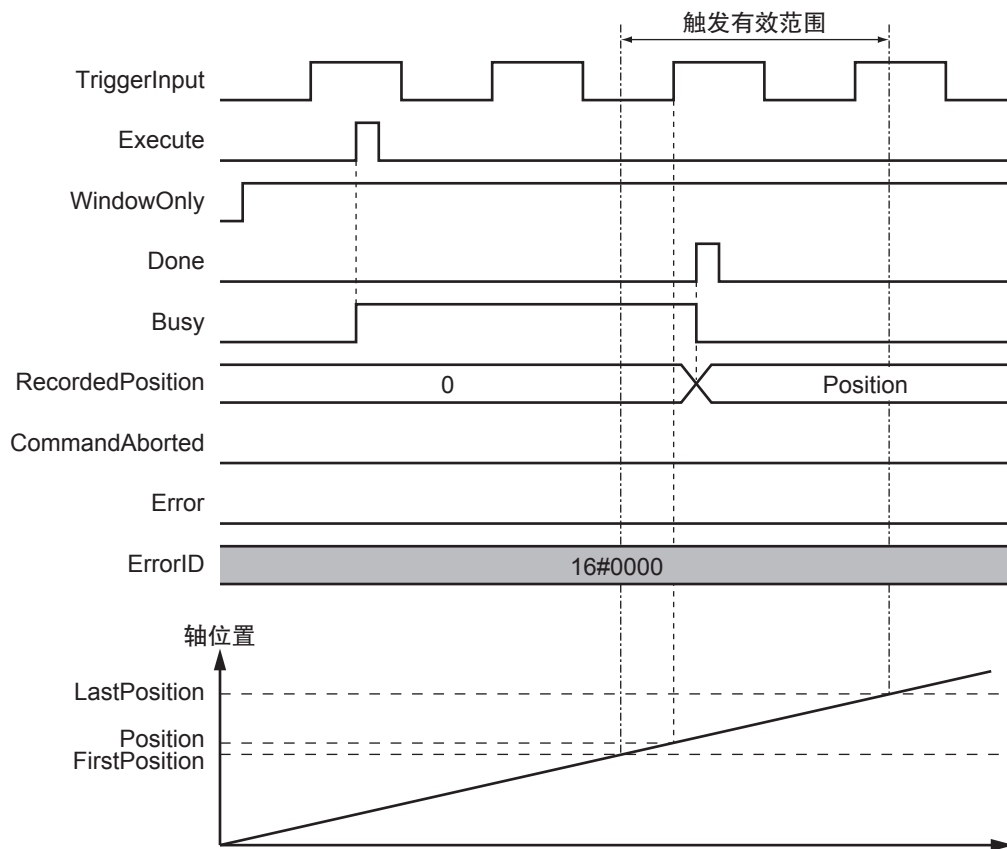
WindowOnly为Disable时

将Execute(启动)变为TRUE后最初发生触发时的轴位置作为标准距离的基准位置。



WindowOnly为Enable时

仅在窗口的范围内检出触发输入，获取轴位置。



使用注意事项

- WindowOnly(窗口有效)由FALSE变化为TRUE的瞬间以及锁定功能启动之间的时间无法锁定。
- 锁定功能启动需要时间，因此，WindowOnly(窗口有效)的有效范围极短时无法锁定。可锁定有效范围的临界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和EtherCAT的通信。

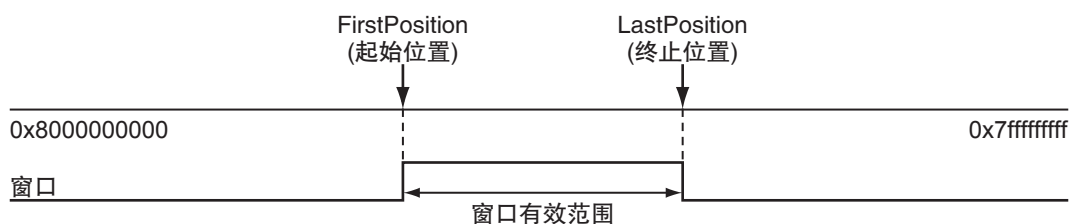
不同计数模式的FirstPosition(起始位置)和LastPosition(终止位置)的范围如下所示。

线性模式

- $\text{FirstPosition(起始位置)} \leq \text{窗口范围} \leq \text{LastPosition(终止位置)}$
- 指定 $\text{FirstPosition(起始位置)} > \text{LastPosition(终止位置)}$ 时，会发生异常。
- 超过[线性模式]的位置范围指定时，也会发生异常。
- 此外，由于FirstPosition(起始位置)和LastPosition(终止位置)为LREAL型变量，因此，请勿指定 $\text{FirstPosition(起始位置)} = \text{LastPosition(终止位置)}$ 。

关于LREAL型，请参阅  “实数(REAL型、LREAL型)的使用(P.1-14)”。

[线性模式]时的窗口有效范围如下图所示。

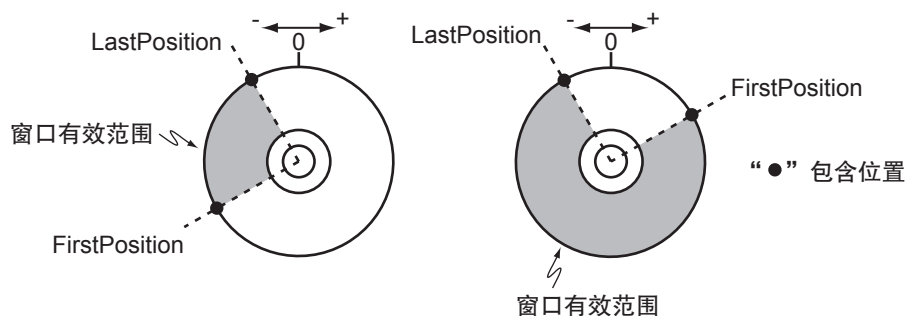


(注) FirstPosition(起始位置)和LastPosition(终止位置)包含在窗口有效范围中。

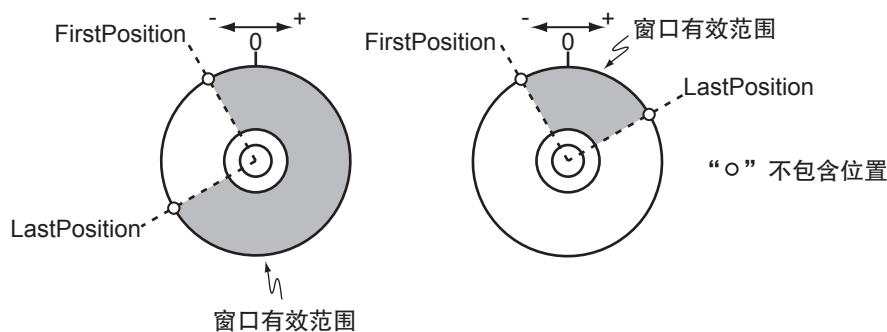
旋转模式

- FirstPosition(起始位置) \leq LastPosition(终止位置)和FirstPosition(起始位置) $>$ LastPosition(终止位置)两者均可指定。
- FirstPosition(起始位置) $>$ LastPosition(终止位置)时, 设定值跨越环计数器的上下限位置。
- 超过环计数器上下限范围指定时, 会发生异常。

FirstPosition \leq LastPosition



FirstPosition $>$ LastPosition



● ReferenceType(位置类型选择)

位置类型如下所示。

- `_mcFeedback`: 同一任务周期下的取值
使用同一任务周期时获取的主轴反馈位置。
这里的任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先级5)的周期。

● FeedDistance(标准距离)

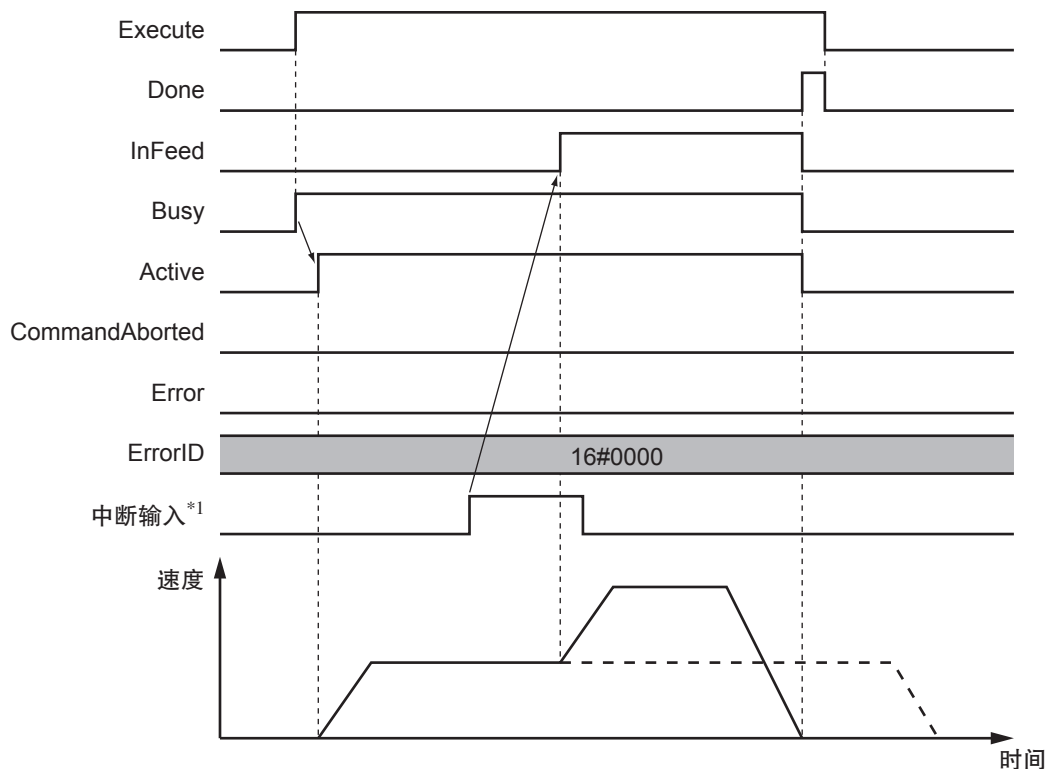
沿与中断输入前相同方向执行标准距离动作时，对FeedDistance(标准距离)指定正值。沿与中断输入前相反方向执行标准距离动作时，对FeedDistance(标准距离)指定负值。

例如，中断输入前动作表现为负方向动作时，如果对FeedDistance(标准距离)指定正值，则标准距离动作表现为负方向动作。如果对FeedDistance(标准距离)指定负值，则标准距离动作表现为正方向动作。

时序图

- 在启动Execute(启动)的同时，Busy(执行中)变为TRUE。在下一个周期Active(控制中)变为TRUE。
- 中断输入后，InFeed(标准传送中)变为TRUE、到达FeedDistance(标准距离)，定位完成时，Done(完成)变为TRUE。
- 利用其它指令中断本指令时，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



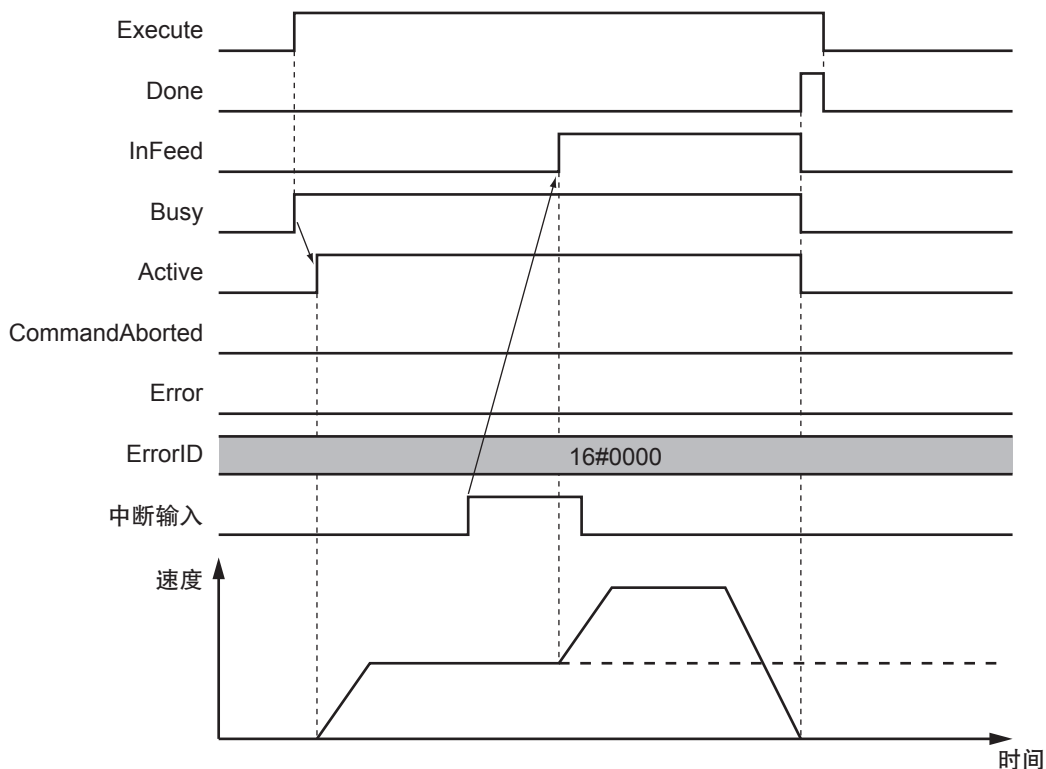
*1 从中断信号“ON”到InFeed变为TRUE之前，可能延迟多个控制周期。



参考

对中断标准定位的加速和减速取决于输入变量的Acceleration(加速度)、Deceleration(减速度)。

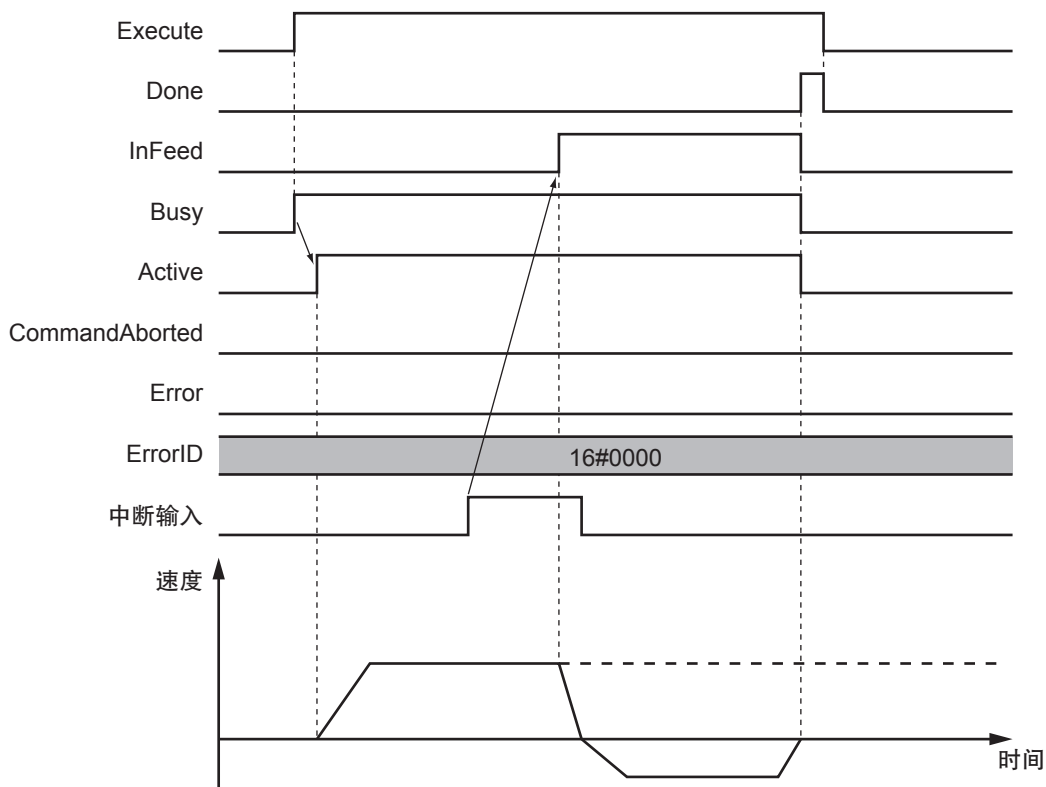
● MoveMode(移动方法选择)为_mcVelocity时



● 轴参数的反转时动作

中断后的标准定位反转时，根据轴参数的[反转时动作]进行反转。

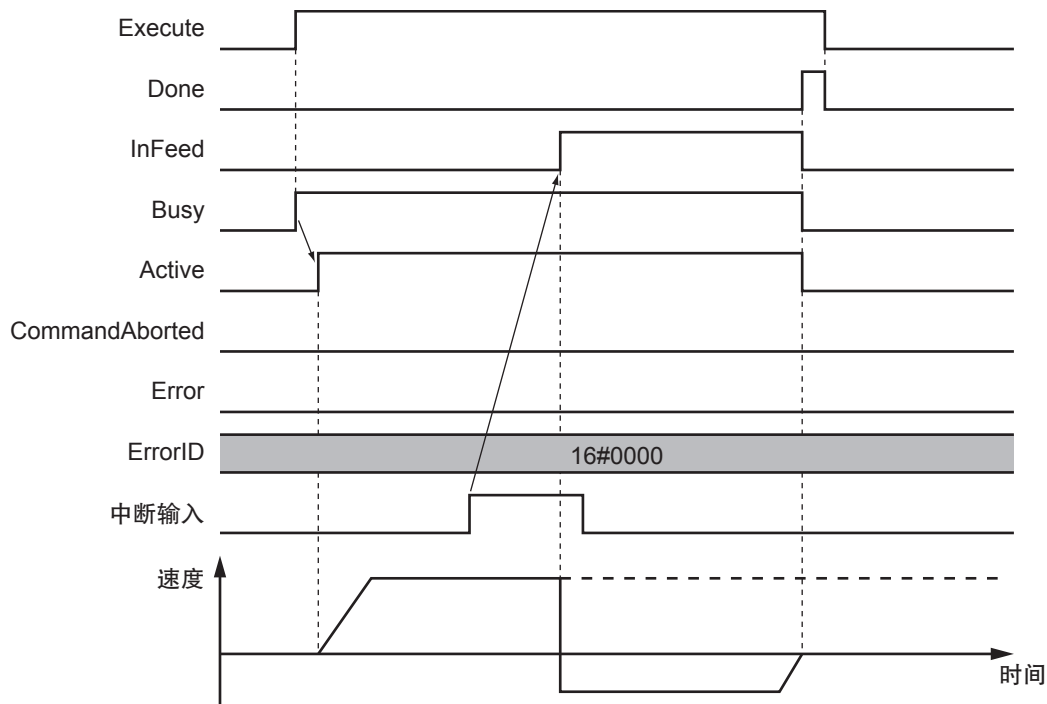
[0: 以反转时减速停止反转时]时



参考

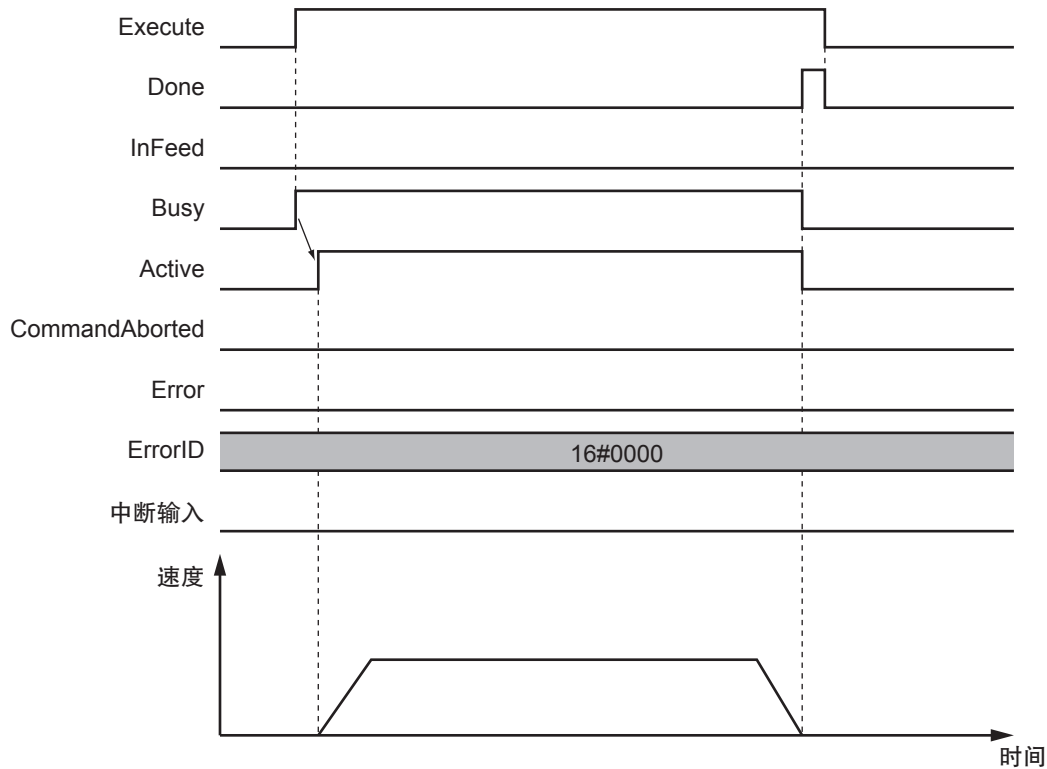
中断输入时反转的减速度取决于输入变量的Deceleration(减速度)。

[1: 以反转时立即停止反转时]时

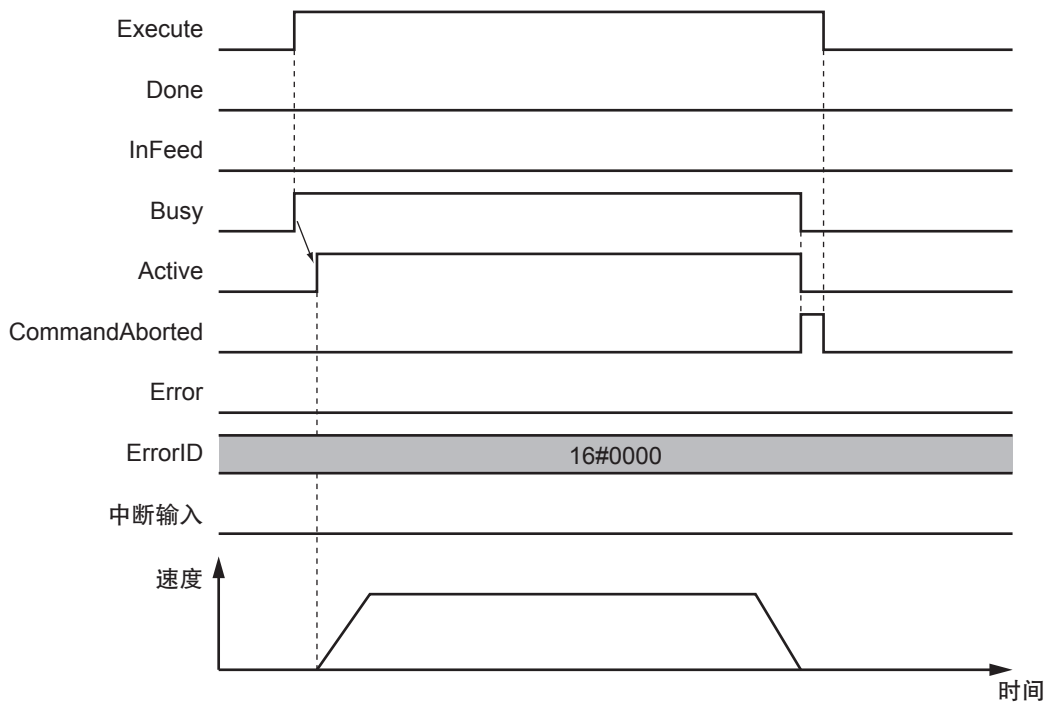


- MoveMode(移动方法选择)为_mcAbsolute, 无中断输入时

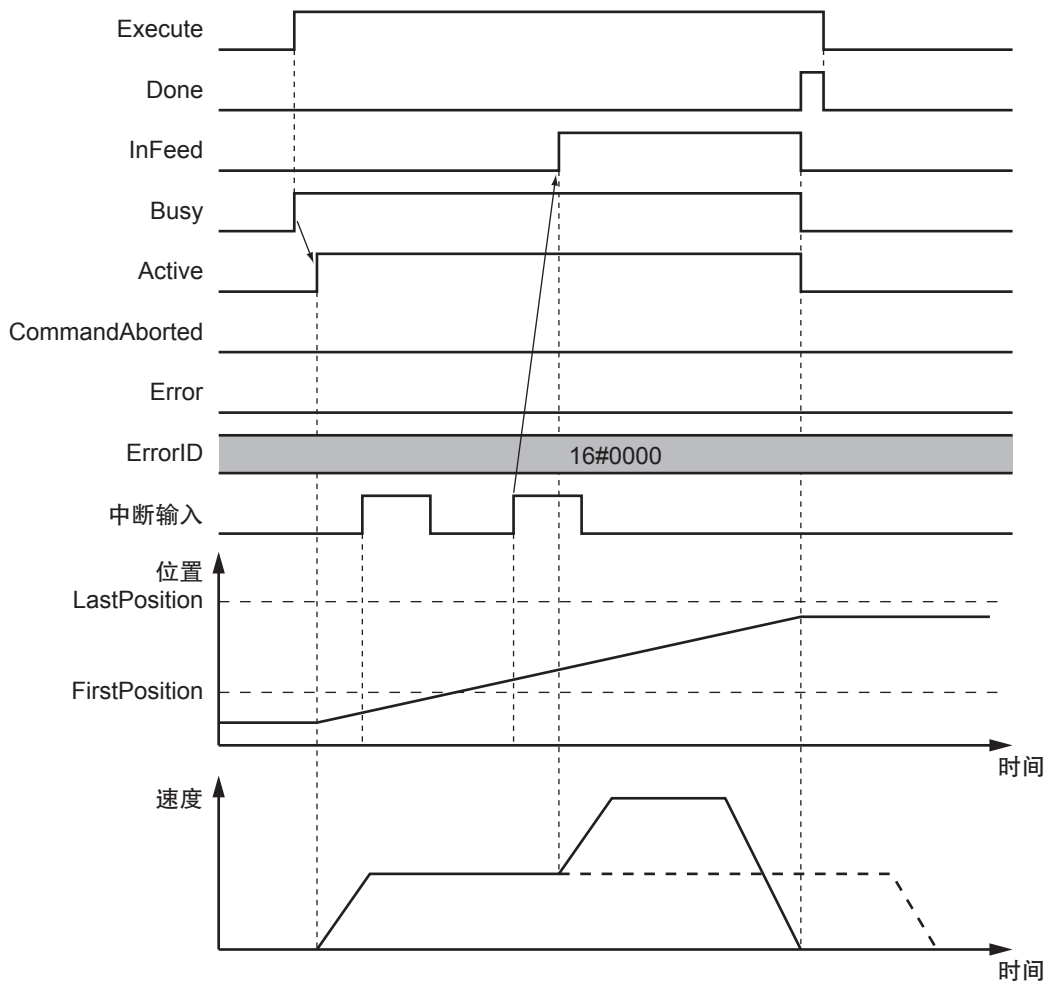
[ErrorDetect(错误检测选择): 不执行异常检测], 无中断输入时



[ErrorDetect(错误检测选择): 执行异常检测], 无中断输入时



● WindowsOnly(窗口有效)为[有效]时



重启运动指令

无法重启本指令。

重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

多重启动运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动

对当前正在执行的指令启动本指令，可切换或缓存到本指令。

各轴可缓存到1个。

多重启动时的本指令的动作由BufferMode(缓存模式选择)指定。

缓存模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 轴的动作方向因指令切换而反转时，根据轴参数中的“反转时动作”进行反转。
等待	当前正在执行的指令正常完成后，已缓存的本指令自动启动。

关于BufferMode(缓存模式选择)，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 本指令执行中的其它指令启动

本指令执行中，可能以中断模式多重启动。

选择等待、合并多重启动时，

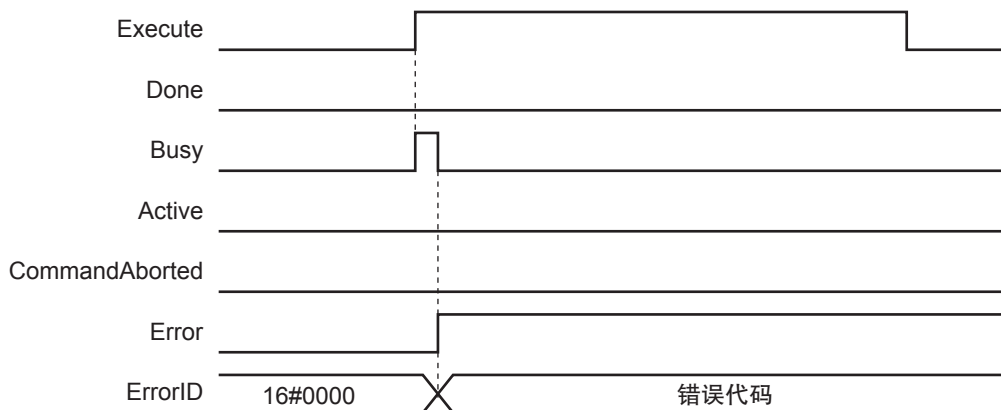
- 执行多重启动指令时，Error(错误)变为TRUE。同时，ErrorID(错误代码)输出“无法多重启动运动指令(错误代码：543C Hex)”。
- 本指令处理中止，CommandAborted(执行中断)变为TRUE。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图

发生异常时，本指令使用的中断输入用锁定无效。



● 错误代码

关于指令发生的异常，请参阅 [□](#) “A-1 错误代码一览(P.A-2)”。

示例程序

下面，对从速度控制变为中断标准定位的控制中的示例程序进行说明。

参数设定

该示例程序中使用的最少的必要设定如下所示。

● 轴参数的设定

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	线性模式

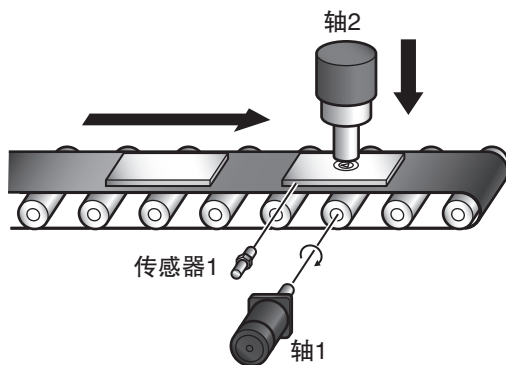
环计数器

轴	上限值	下限值
轴1	360	0

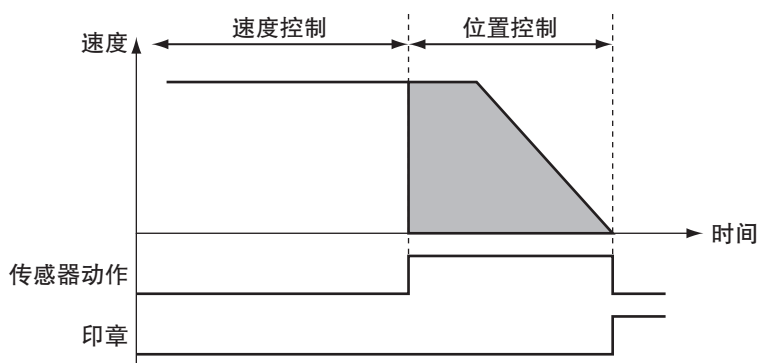
坐标单位选择

轴	坐标单位
轴1	degree
轴2	mm

动作示例



● 动作模式



- 1 输送机动作**
驱动皮带输送机的轴1，以中断标准定位前的动作进行速度控制。
- 2 标准传送**
传感器1连接锁定功能1。
传感器1“ON”时，切换为标准定位，在规定位置停止。
- 3 印章的盖印**
定位完成后，印章的轴2因绝对定位而垂直下降，进行盖印。
盖印后，轴2返回原点位置。

在绝对定位动作完成的同时返回原点位置，因此，以等待启动MC_MoveZeroPosition(高速原点复位)的Buffer Mode(缓存模式选择)。

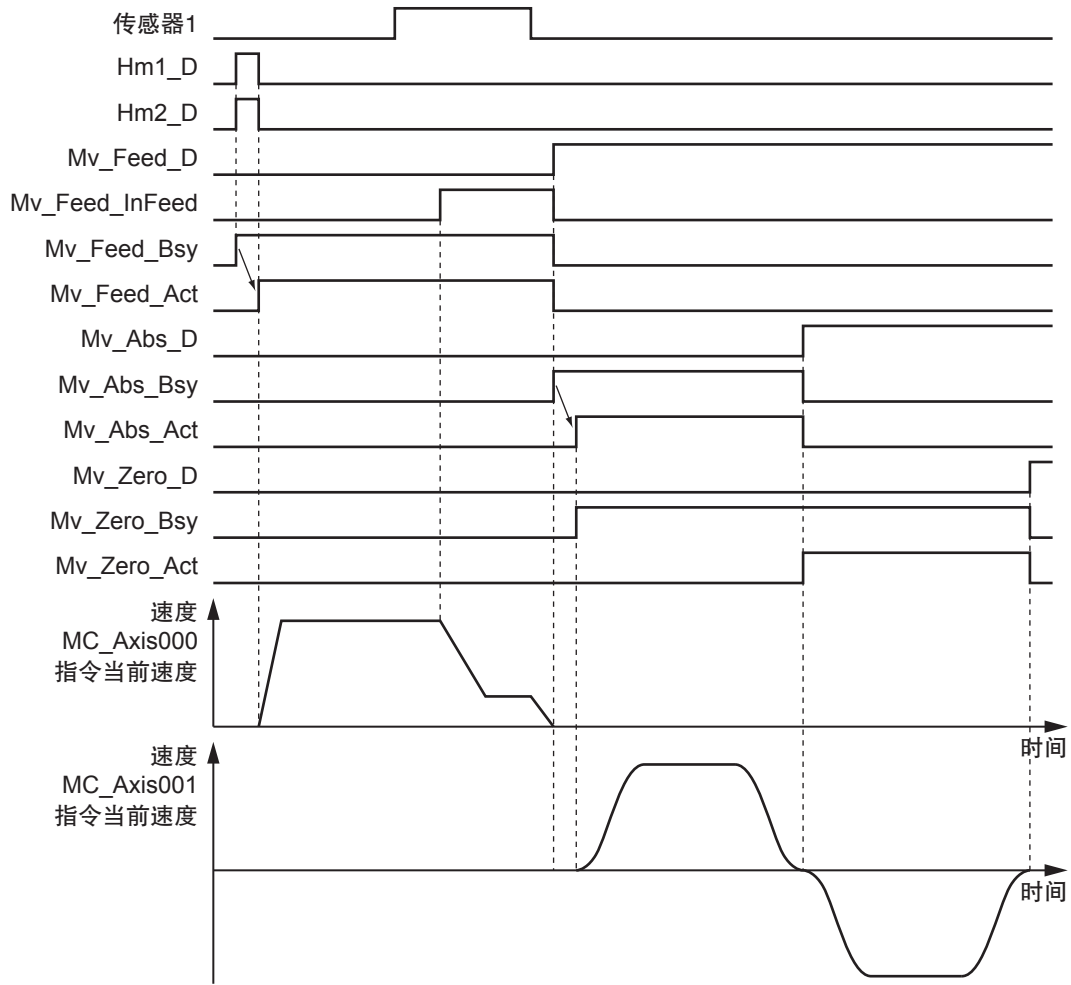
上一指令的输出Active为TRUE时，多重启动指令。

梯形图

● 主要变量

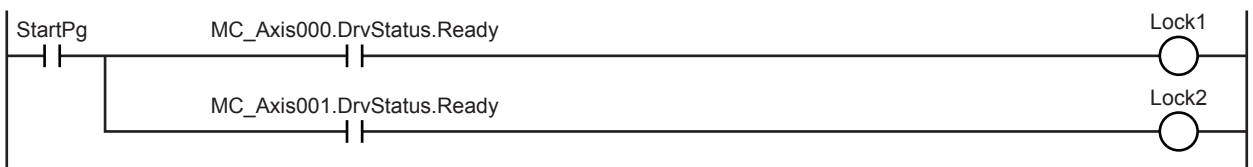
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	轴1的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴1发生轻度故障等级的异常时变为TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴1为原点确定状态时变为TRUE。
MC_Axis001	_sAXIS_REF	-	轴2的轴变量。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴2发生轻度故障等级的异常时变为TRUE。
MC_Axis001.Details.Homed	BOOL	FALSE	轴2为原点确定状态时变为TRUE。
TrigRef	_sTRIGGER_REF	-	是中断输入的指定变量。 本程序中，使用伺服驱动器的锁定功能1。检测出外部输入(传感器1)的上升沿，执行标准移动。
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则进入伺服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

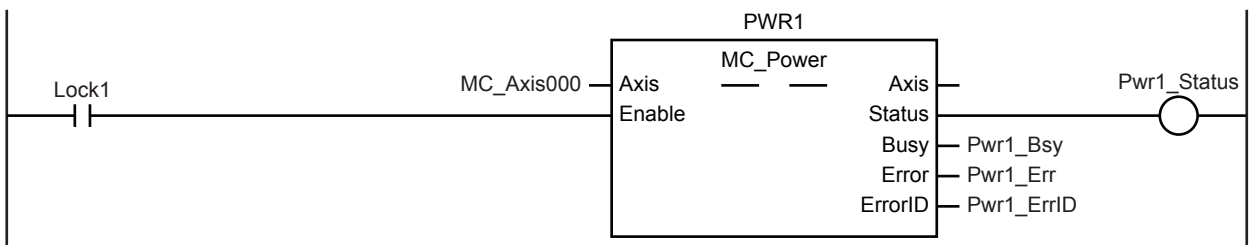


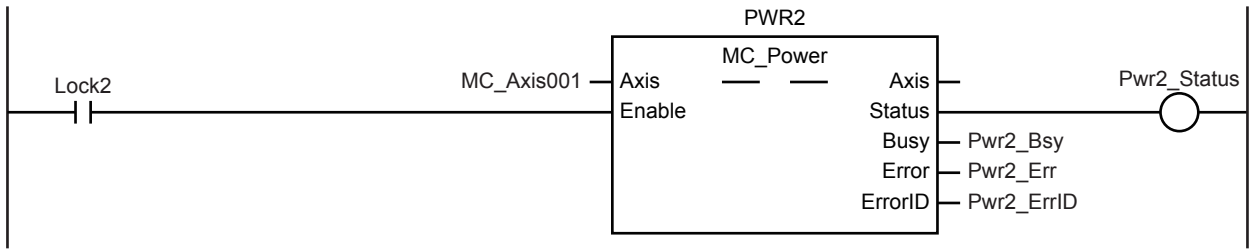
● 示例程序

触点StartPg变为TRUE时，确认各轴的伺服驱动器处于伺服准备就绪状态

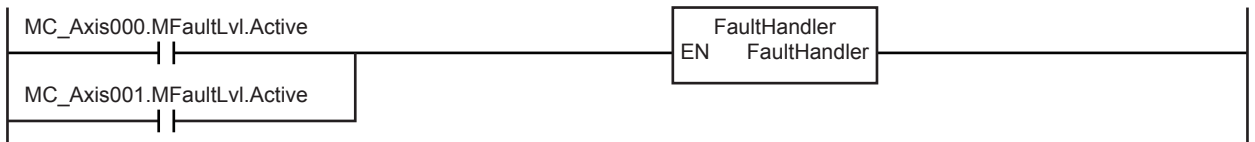


如果处于伺服准备就绪状态，则将各轴设为伺服ON状态

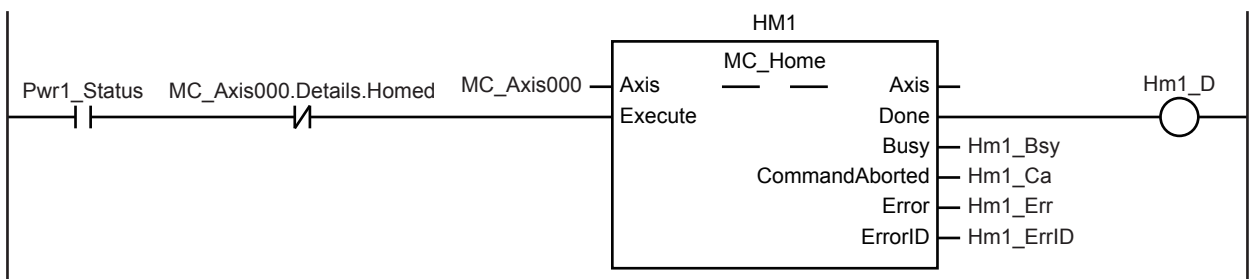




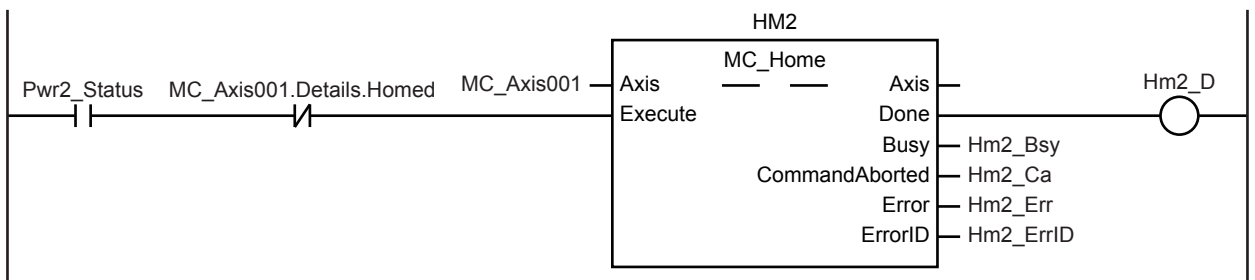
轴构成发生轻度故障程度的异常时，执行设备的异常处理(FaultHandler)。
发生异常时的处理(FaultHandler)由客户根据装置进行编程



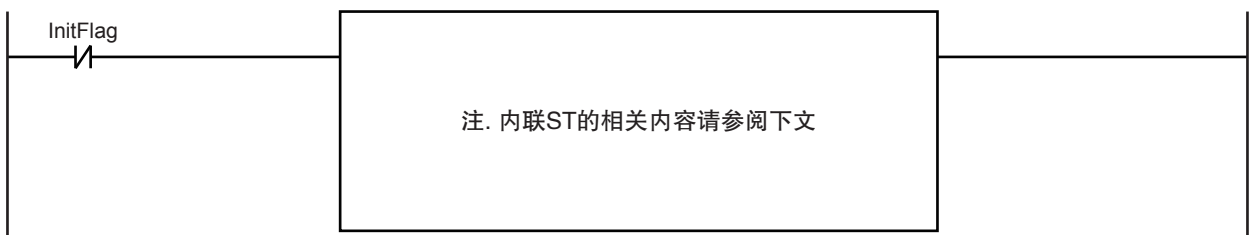
轴1变为伺服ON状态，若原点未确定，则执行原点复位



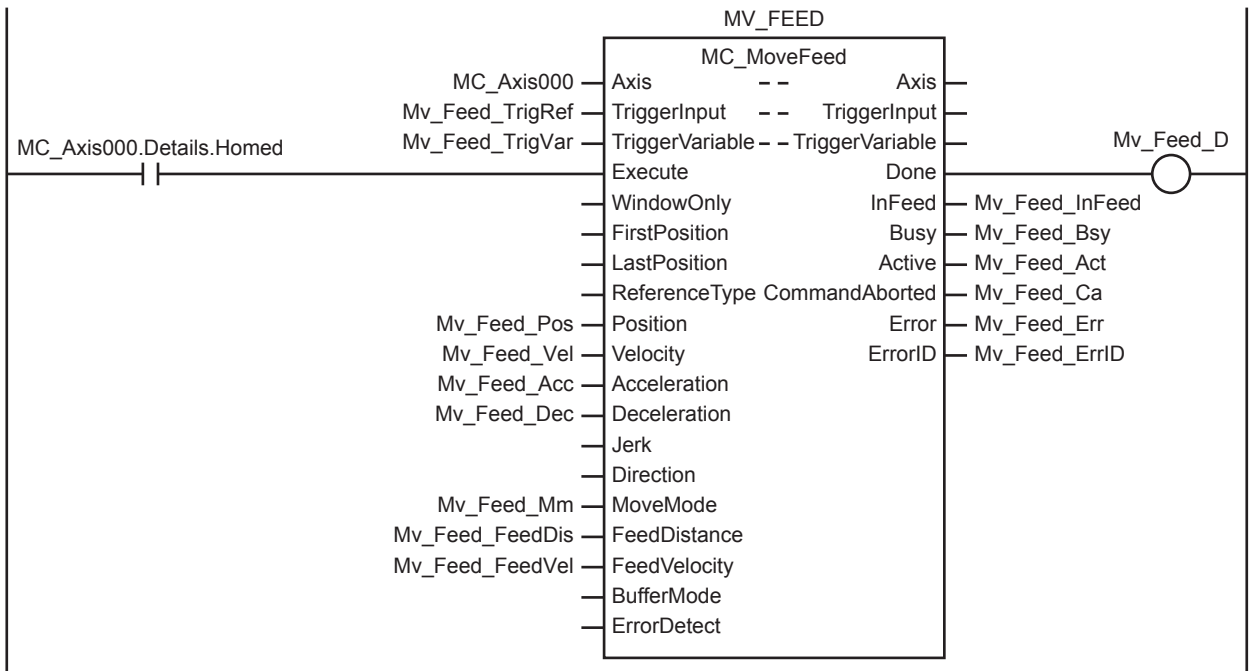
轴2变为伺服ON状态，若原点未确定，则执行原点复位



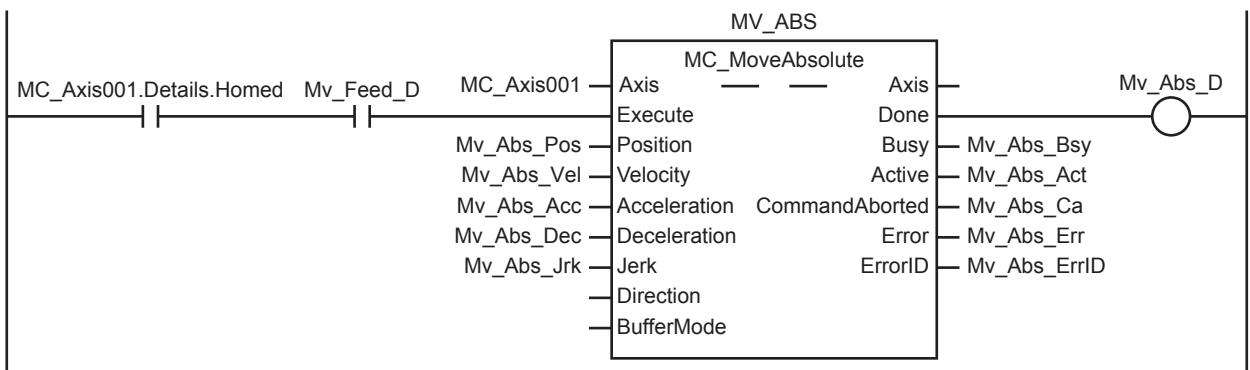
设定中断标准定位、绝对值定位、高速原点复位的参数



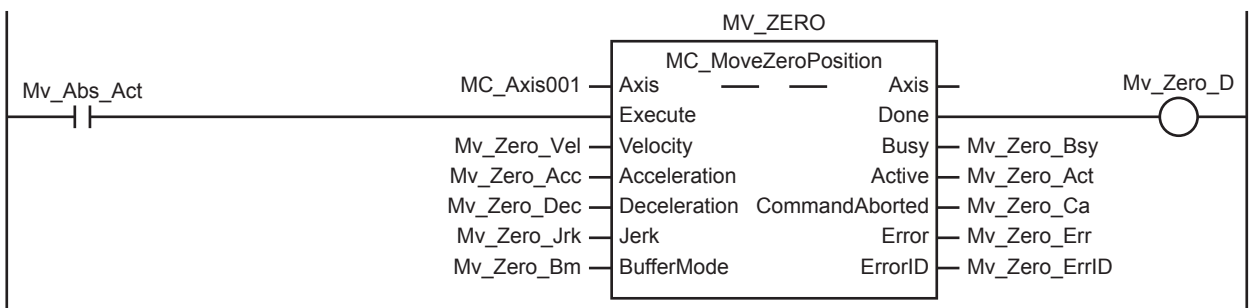
确认轴1已确定原点后，执行中断标准定位



轴2原点已确定，且轴1的中断标准定位完成后，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，移动至原点



内联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;

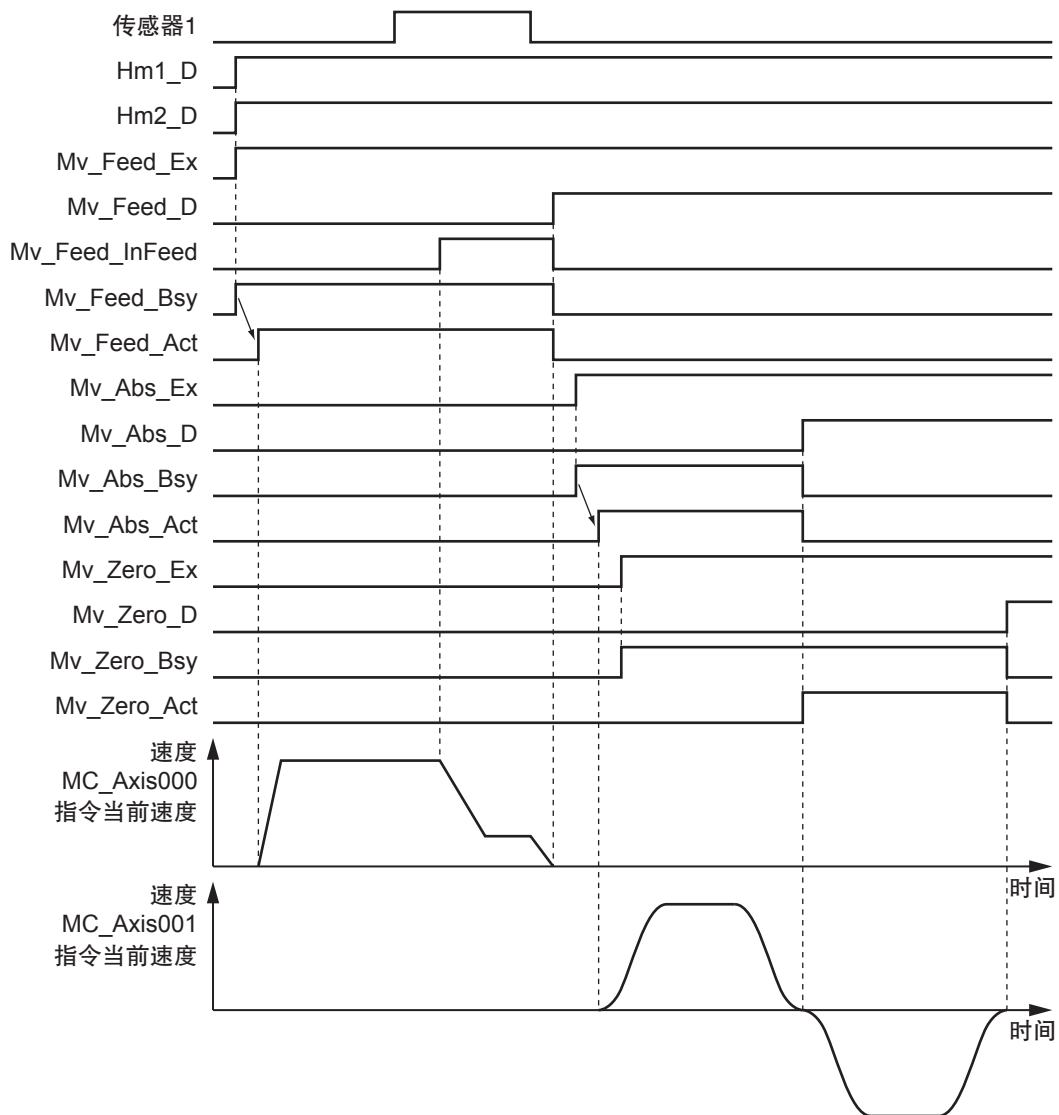
```

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为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为原点确定状态时变为TRUE。
TrigRef	_sTRIGGER_REF	-	是中断输入的指定变量。 本程序中，使用伺服驱动器的锁定功能1。检测出外部输入(传感器1)的上升沿，执行标准移动。
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则进入伺服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	该变量由FALSE→TRUE时，启动MC_Home的实例HM1。
Hm2_Ex	BOOL	FALSE	该变量由FALSE→TRUE时，启动MC_Home的实例HM2。
Mv_Feed_Ex	BOOL	FALSE	该变量由FALSE→TRUE时，启动MC_MoveFeed的实例MV_FEED。
Mv_Abs_Ex	BOOL	FALSE	该变量由FALSE→TRUE时，启动MC_MoveAbsolute的实例MV_ABS。
Mv_Zero_Ex	BOOL	FALSE	该变量由FALSE→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_TriggerRef.Mode      := _eMC_TRIGGER_MODE#_mcDrive;
Mv_Feed_TriggerRef.LatchID   := _eMC_TRIGGER_LATCH_ID#_mcLatch1;
Mv_Feed_TriggerRef.InputDrive := _eMC_TRIGGER_INPUT_DRIVE#_mcEXT;
Mv_Feed_TriggerVar           := 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, 确认伺服驱动器处于伺服准备就绪状态,
//将轴1设为伺服ON状态。
//如果未处于伺服准备就绪状态, 则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) 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, 确认伺服驱动器处于伺服准备就绪状态,
//将轴2设为伺服ON状态。
//如果未处于伺服准备就绪状态, 则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis001.DrvStatus.Ready=TRUE) 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)由客户根据装置进行编程。
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
  FaultHandler();
END_IF;

//轴1处于伺服ON状态, 且原点未确定时, 进行轴1的原点复位
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
  Hm1_Ex:=TRUE;
END_IF;

//轴2处于伺服ON状态, 且原点未确定时, 进行轴2的原点复位
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
  Hm2_Ex:=TRUE;
END_IF;

//轴1确定原点后, 执行MC_MoveFeed
IF MC_Axis000.Details.Homed=TRUE THEN
  Mv_Feed_Ex:=TRUE;
END_IF;

```

```

//轴2原点已确定，且轴1的中断标准定位完成后，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;

//MC_MoveAbsolute开始时，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

使轴减速停止。

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	在上升沿开始指令。
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	缓存模式选择	_eMC_BUFFER_MODE	0: _mcAborting	0*2	指定多重启动运动指令时的动作。 0: 中断

*1. 关于指令单位，请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*2. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

*1. 请参见“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	减速停止完成，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	开始指令后	<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"> 利用其它的MC_Stop(强制停止)指令多重启动指令(中断),中断本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 	<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含有异常因素时	异常已解除时

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. 请在使用在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

- 使轴由当前速度变为速度“0”，进行减速控制。
- 在Execute(启动)的上升沿，开始减速停止的动作。
- 因启动MC_Stop(强制停止)指令而处于运行中的指令执行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。
错误减速停止中使轴动作停止时，请使用MC_ImmediateStop(立即停止)指令。

指令详情

下面对指令详细说明。

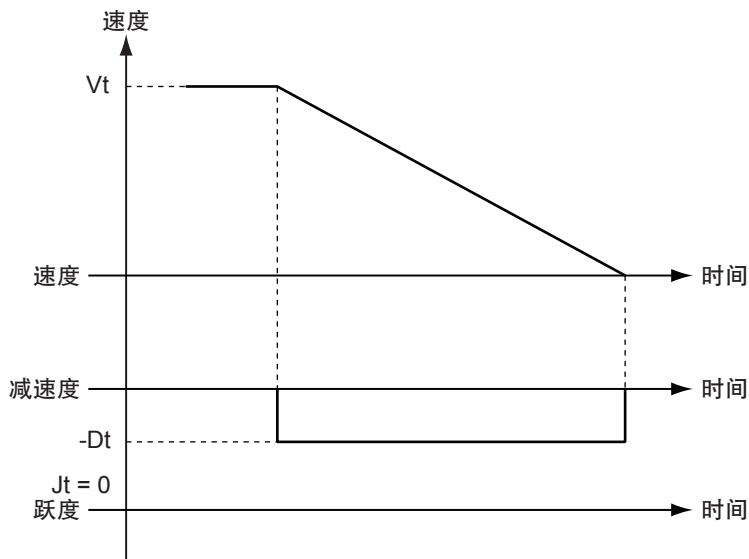
● Deceleration(减速度)、Jerk(跃度)的指定

减速停止时的减速度、跃度由输入变量的Deceleration(减速度)、Jerk(跃度)指定。

跃度指定为“0”和非“0”时的减速度、速度的关系如下所示。

跃度为“0”时

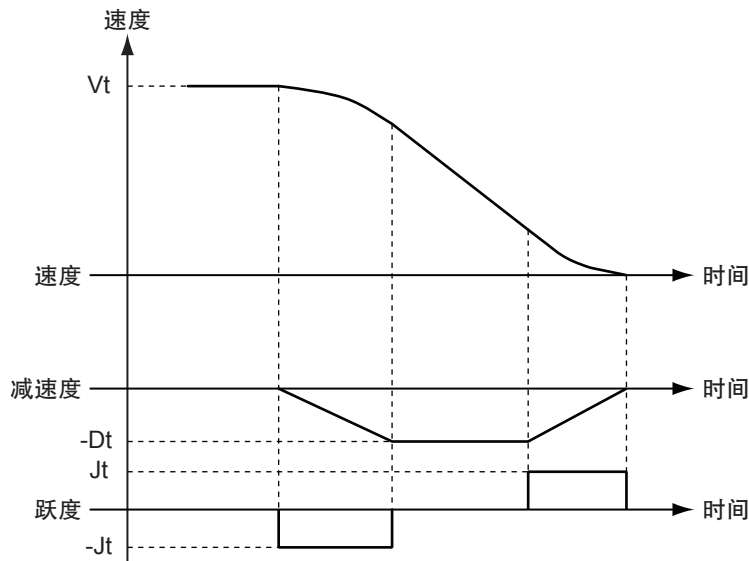
以减速度Dt生成速度的指令值。



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

跃度为非“0”时

以减速度的上限为Dt，以当前速度为基点，生成速度的指令值。



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

- 将减速度指定为“0”时，即为即停指令。不减速而立即停止。仅将减速度指定为“0”时，执行立即停止。与[加减速超限]的设置无关。

● BufferMode(缓存模式选择)的指定

指定前一个轴动作和本次动作的连接方式。

本指令的BufferMode(缓存模式选择)为将来扩展用的保留参数。

有以下1种。

缓存模式选择	说明
中断	立即中止当前执行中的指令，执行本指令。

关于BufferMode(缓存模式选择)，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

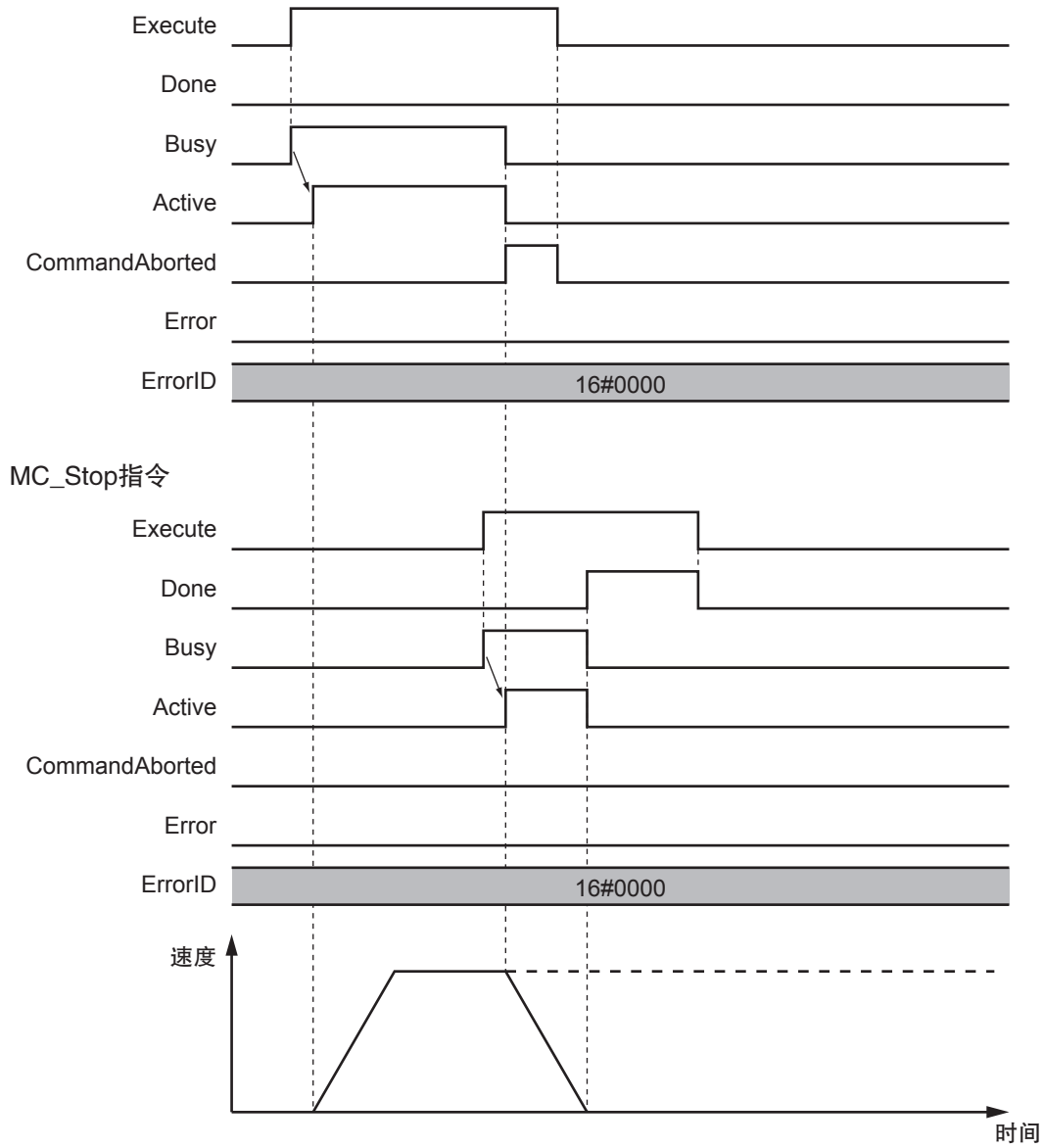
● 到位检查

利用本指令停止时，不进行到位检查。

时序图

- 在启动Execute(启动)的同时, Busy(执行中)变为TRUE。在下一个周期Active(控制中)变为TRUE。
- 速度达到“0”时, Done(完成)变为TRUE。
- 利用其它指令中断本指令时, 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

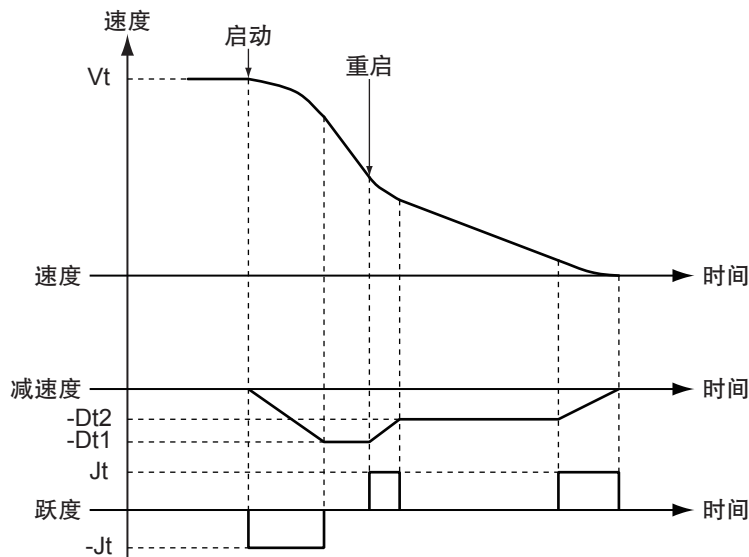


重启运动指令

在本指令执行中再次启动Execute(启动)时，进行Deceleration(减速度)的变更。
重启运动指令时，不进行Jerk(跃度)的变更。

● 跃度为非“0”时

以当前速度、当前减速度为基点，以变更后减速度的上限为Dt2，生成速度的指令值。



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

多重启动运动指令

多重启动运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动

轴组动作中

指定轴在轴组动作中，执行MC_Stop(强制停止)指令时，轴会发生异常。
轴组也发生异常，轴组动作也停止。

轴变量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。
错误减速停止中启动MC_Stop(强制停止)指令后，无法正常启动，CommandAborted(执行中断)变为TRUE。
要使发生异常的轴停止，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

MC_ResetFollowingError(偏差计数器复位)指令执行中启动MC_Stop(强制停止)指令时，MC_ResetFollowingError(偏差计数器复位)指令CommandAborted(执行中断)变为TRUE，执行MC_Stop(强制停止)指令。但是，本指令的Deceleration(减速度)不适用，表现为立即停止。

● 本指令执行中的其它指令启动

- 除了利用MC_Stop(强制停止)指令使轴减速停止外,启动如下指令时,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(启用外部锁定)指令的立即停止中
 - 其它MC_Stop(强制停止)指令的Execute(启动)至少一个变为TRUE时
- Status.Stopping(减速停止中)对TRUE的轴表现为如下情形。
 - 进行单轴定位、连续动作、同步动作、手动动作时,这些指令的 CommandAborted(执行中断)变为TRUE。
 - MC_Stop(强制停止)指令执行中启动MC_ResetFollowingError(偏差计数器复位)指令时,MC_Stop(强制停止)指令的Done(完成)变为TRUE,执行MC_ResetFollowingError(偏差计数器复位)指令。
 - 可多重启动多个MC_Stop(强制停止)指令。执行中的MC_Stop(强制停止)指令的Done(完成)变为TRUE。
- 启动MC_Stop(强制停止)指令后呈如下状态时,MC_Stop(强制停止)指令的Done(完成)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)
 - MC_TouchProbe(启用外部锁定)指令的输入变量“StopMode(停止方法选择)”选择[1: _mcImmediate Stop]的状态下,触发条件成立,欧姆龙伺服驱动器1S系列或G5系列立即停止时

异常

本指令执行中，轴等发生异常时，停止动作。

通过轴参数，从立即停止、减速停止、伺服OFF中指定一种停止方法。

停止方法指定减速停止时，继续减速停止动作。

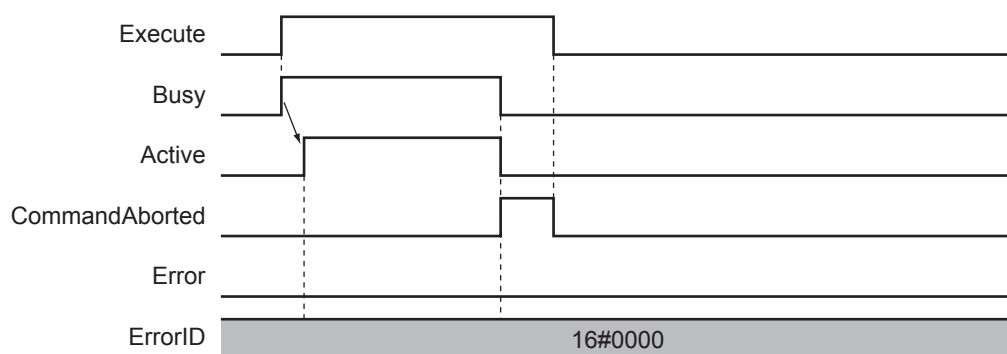
关于轴参数停止方法的设定，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 发生异常时的时序图

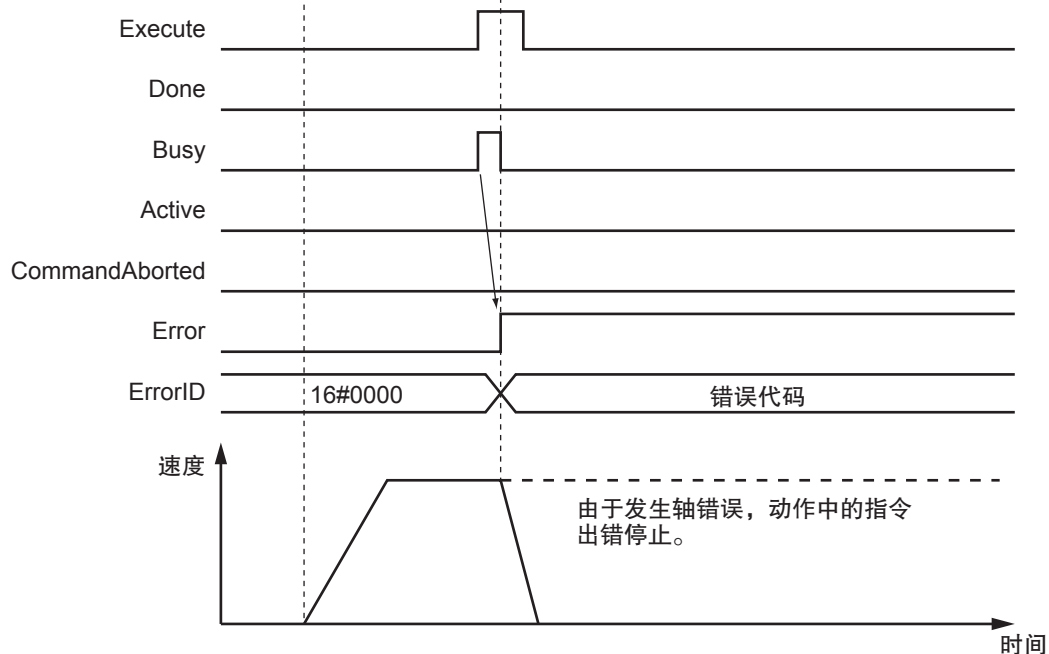
在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

MC_MoveVelocity



MC_Stop

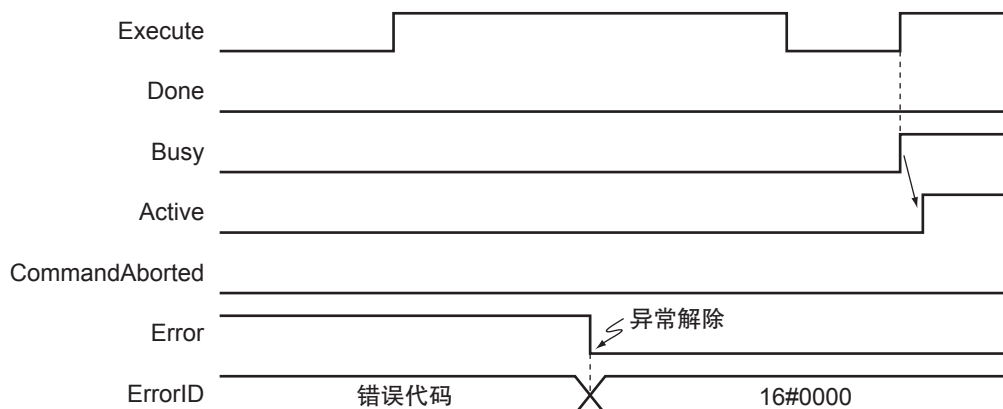


版本相关信息

根据CPU单元的版本，异常解除时的动作如下所示。
请在轴停止后解除异常。轴动作中无法解除异常。

- Ver.1.10以上版本的CPU单元时
解除本指令的异常后，在下一个Execute(启动)的上升沿才可启动本指令。

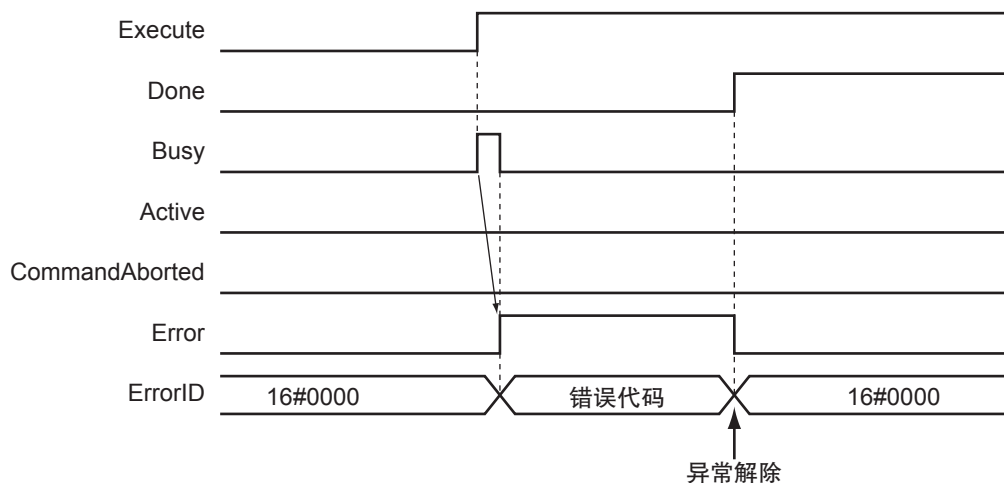
Ver.1.10以上



- Ver.1.09以下版本的CPU单元时
本指令发生异常时，如果在Execute(启动)为TRUE的状态下解除异常，会发生以下动作。
 - 如解除了异常原因，则Error(错误)变为FALSE、Done(完成)变为TRUE。轴变量 Status.Stopping(减速停止中)与正常执行减速停止时一样，变为TRUE。
 - 未解除异常原因时，则本指令将再次变为Error(错误)，且轴发生异常。

解除异常原因时的时序图如下所示。

Ver.1.09以下



● 错误代码

关于指令发生的异常，请参阅 “A-1 错误代码一览(P.A-2)”。

MC_ImmediateStop

与轴的状态无关，根据输入变量“StopMode(停止方法选择)”的停止方法执行停止。

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	在上升沿开始指令。
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，并进行偏差计数器复位 3: 立即停止，并变为伺服OFF

*1. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

* 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	减速停止完成后	<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"> • 利用其它的MC_Stop(强制停止)指令多重启动运动指令(中断),中止本指令时 • 因发生异常,中止本指令时 	<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含有异常因素时	异常已解除时

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. 请使用在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

- 本指令是可在任何状态下执行的指令。
例如, 即使由于异常而处于减速停止中, 也可以利用本指令立即停止。
MC_Stop(强制停止)指令在Status.ErrorStop(错误减速停止中)为TRUE时无法使用, 但可使用MC_ImmediateStop(立即停止)指令。
- 执行指令后, 则按照StopMode(停止方法选择)的指定立即停止动作, 动作中的指令变为CommandAborted(执行中断)状态。
- 执行本指令后, 轴状态的Status.ErrorStop(错误减速停止中)变为TRUE, 发生“启动立即停止指令(错误代码: 5485 Hex)”错误。



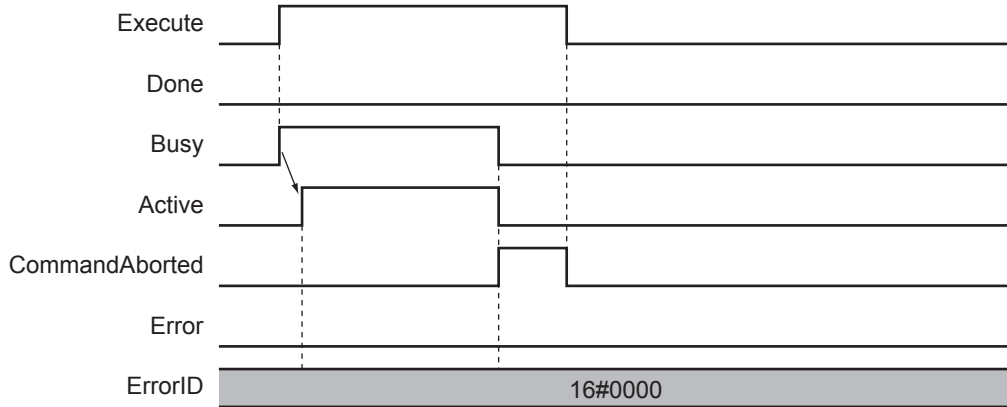
使用注意事项

有关对于主轴的注意事项, 请参阅 □ “同步控制的主轴及辅轴的注意事项(P.1-6)”。

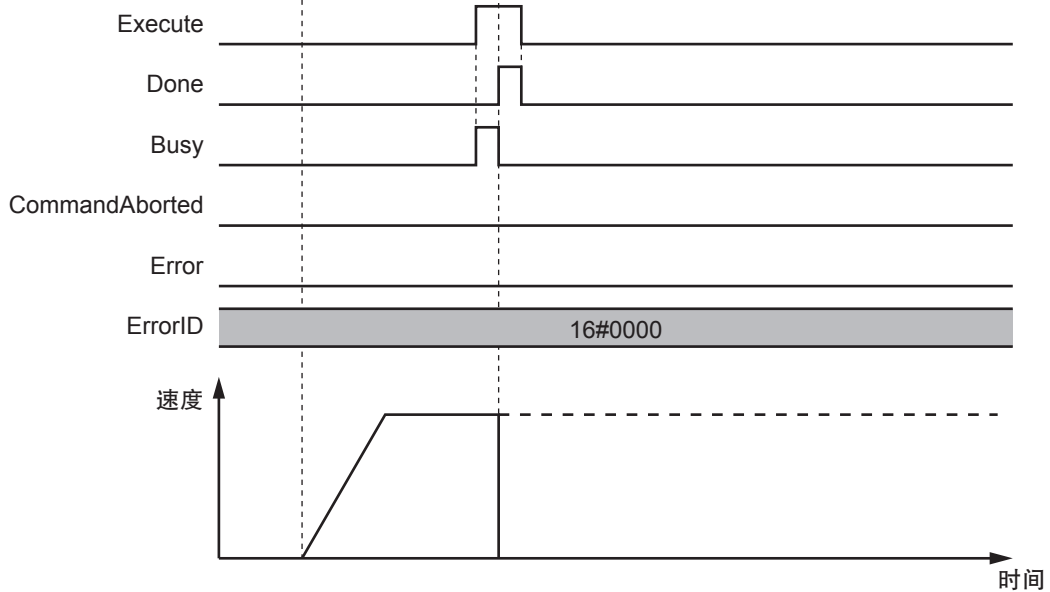
时序图

- 在启动Execute(启动)的同时，Busy(执行中)变为TRUE。
- 立即停止指令处理完成时，Done(完成)变为TRUE。

MC_Move指令



MC_ImmediateStop指令



重启运动指令

- 无法重启本指令。
- 重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动

轴组动作中

在轴组动作中对指定轴执行指令时，轴发生异常，立即停止。
轴组也发生异常，轴组动作也停止。

轴变量Status.Stopping(减速停止中)为TRUE时

下述状态下，轴变量Status.Stopping(减速停止中)变为TRUE。

- 通过MC_Stop(强制停止)指令使轴减速中
- MC_ResetFollowingError(偏差计数器复位)指令执行中
- MC_TouchProbe(启用外部锁定)指令的立即停止中
- 多重启动MC_Stop(强制停止)指令的Execute，至少1个变为TRUE时

对于Status.Stopping(减速停止中)为TRUE的轴，也可执行本指令。

启动本指令后，下列执行中指令的CommandAborted(执行中断)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(启用外部锁定)指令

轴变量Status.ErrorStop(错误减速停止中)为TRUE时

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

因异常而处于减速停止中时，也可执行本指令。

异常

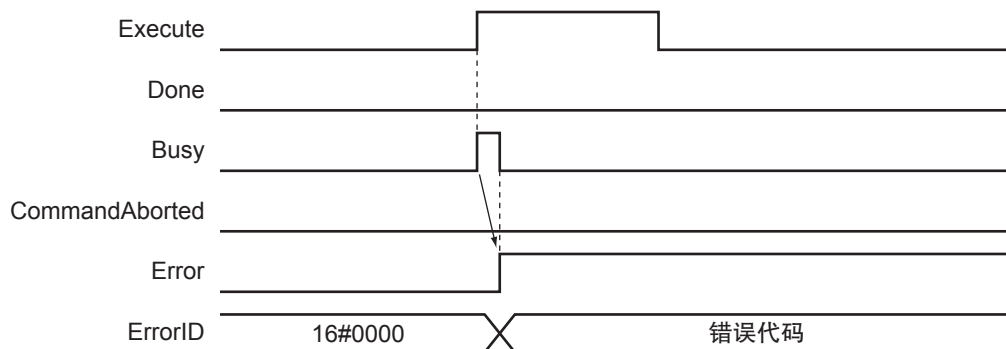
本指令执行中轴等发生异常时，也直接继续立即停止动作。

关于轴参数停止方法的设定，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 发生异常时的时序图

在执行本指令中发生异常时，Error(错误)变为TRUE，轴立即停止。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。



● 错误代码

关于指令发生的异常，请参阅 □ “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 two inputs on the left: Axis and Execute. It has four outputs on the right: Done, Busy, Error, and ErrorID. The main block also has four inputs on the left: Position, ReferenceType, Relative, and ExecutionMode. The main block has four outputs on the right: Done, Busy, Error, and ErrorID.</p>	<pre>MC_SetPosition_instance (Axis := 《参数》, Execute := 《参数》, Position := 《参数》, ReferenceType := 《参数》, Relative := 《参数》, ExecutionMode := 《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	在上升沿开始指令。
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	位置类型选择	_eMC_ REFERENCE_ TYPE	0: _mcCommand 1: _mcFeedback	0 ^{*2}	指定位置类型。 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)
ExecutionMode (Reserved)	执行模式 选择	_eMC_ EXECUTION_ MODE	0: _mcImmediately	0 ^{*2}	(Reserved)

*1. 关于指令单位，请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*2. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

*1.  请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	指令当前位置和反馈当前位置的变更已完成时	<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，中止本指令时 发生异常过程中，启动本指令时 执行MC_Stop指令中，启动本指令时 	<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含有异常因素时	异常已解除时

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. 请使用在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，对编码器轴执行指令后，将反馈当前位置变更为指定的目标位置。
- 以绝对坐标的位置指定目标位置。
- 在指令当前位置变更的同时，反馈当前位置也会变更，保持该时的位置偏差进行变更。对指令伺服轴执行指令后，保持当前位置和反馈当前位置的差，进行变更。
因此，执行本指令后，轴的反馈当前位置变为下列计算公式所示的值。

变更后的反馈当前位置 = 目标位置 - 变更前的位置偏差

- 对伺服轴指定反馈位置时，或者对编码器轴指定指令位置时，发生位置类型选择异常。
- 计数模式为[旋转模式]时，在环计数器下限值以上、低于环计数器上限值的范围内设定目标位置。如果目标位置超出该范围，则环计数器发生异常。
- 计数模式为[线性模式]时，即使超出软件限制的范围，也可设定目标位置。
- 无论轴停止还是动作中，均可使用本指令。



使用注意事项

本指令完成后，指定轴的原点未确定。
因此，本指令完成后，无法使用如下功能和指令，敬请注意。

- 软件限制功能
- 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(位置类型选择)

- 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)”。

● 轴种类与位置类型的关系

可监控的轴种类和要监控的位置类型的关系如下所示。

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	○

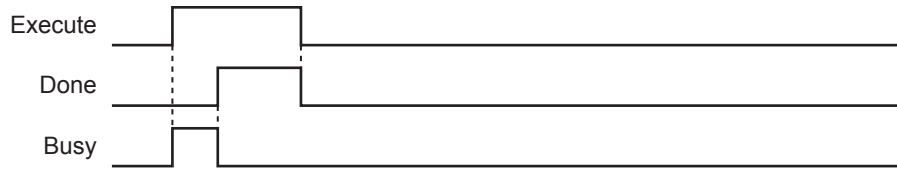
*1. 启动指令时，发生“超过位置类型选择范围(错误代码：5430 Hex)”错误。

时序图

● 轴处于停止状态下使用本指令时

在Execute(启动)的上升沿, 开始当前位置的变更。

在启动Execute(启动)的同时, Busy(执行中)变为TRUE。当前位置的变更完成时, Done(完成)变为TRUE。



● 轴动作中使用本指令时

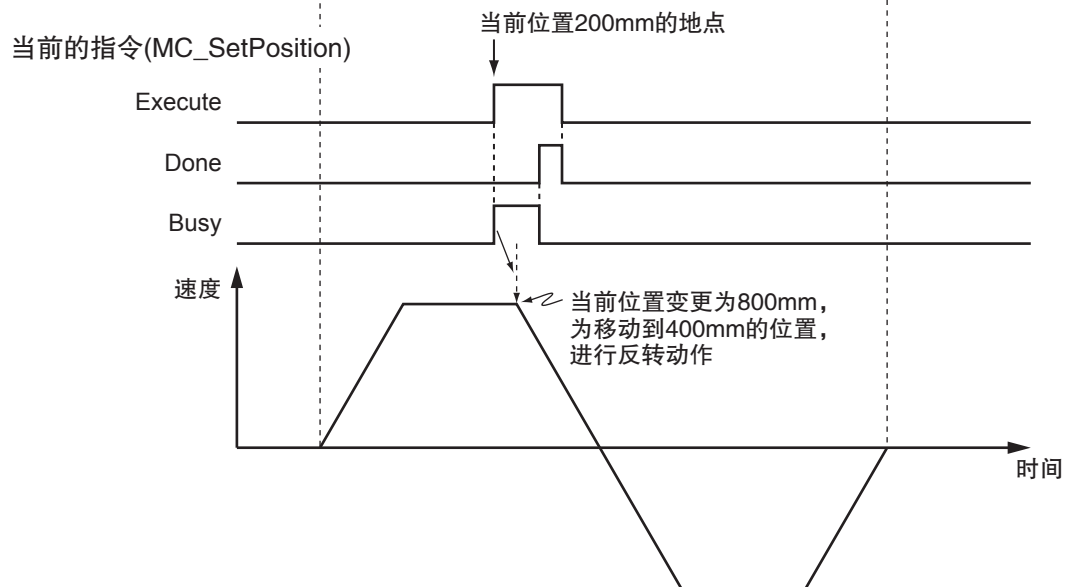
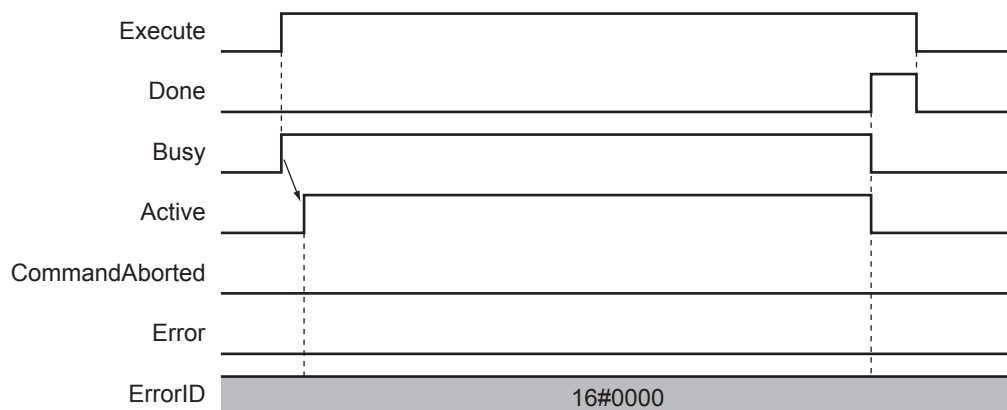
在绝对位置定位中执行本指令后, 随着位置的变更, 动作的目标值也会变更。

例如, 利用MC_MoveAbsolute(绝对值定位)指令, 在移动到400(mm)的过程中, 将当前位置由200(mm)变更为800(mm)后的轴的动作和时序图如下所示。

此时, 由于当前值为800(mm)、目标值为400(mm), 因此轴反转。

如下图所示, 即使利用MC_MoveAbsolute(绝对值定位)指令变更了当前位置, 在从新的当前位置向MC_MoveAbsolute(绝对值定位)指令指定的目标位置定位完成后, Done(完成)也变为TRUE。

之前的指令(Ex. MC_MoveAbsolute)





参考

- 在MC_MoveRelative(相对值定位)指令动作中或MC_MoveVelocity(速度控制)指令动作中执行本指令后，当前位置发生变更。但是，对MC_MoveRelative(相对值定位)指令和MC_MoveVelocity(速度控制)指令的动作没有影响。
- 存在已缓存指令时，切换缓存时相对变更后的位置进行定位。

重启运动指令

无法重启本指令。

重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动

不能对下一个指令执行中的轴使用MC_SetPosition(当前位置变更)指令。

否则，会发生多重启动异常。

MC_MoveJog(微动移动)指令	MC_MoveLink(梯形模式凸轮)指令
MC_Home(原点复位)指令	MC_CombineAxes(加减运算定位)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(转矩控制)指令
MC_GearIn(齿轮动作开始)指令	MC_SyncMoveVelocity(周期同步速度控制)指令
MC_GearInPos(位置指定齿轮动作)指令	



使用注意事项

Ver.1.09以下版本的CPU单元时

请勿对MC_GearIn(齿轮动作开始)指令等的同步用指令的Master(主轴)使用MC_SetPosition(当前位置变更)指令。

如果对Master(主轴)使用了该指令，Master(主轴)的指令当前位置和反馈当前位置变化时，将判断为Master(主轴)发生了移动。此时，Slave(从轴)将根据Master(主轴)的移动量进行动作。因此，可能出现Slave(从轴)的动作急剧变化、或者凸轮结束动作等情况。

要对Master(主轴)使用MC_SetPosition(当前位置变更)指令时，请解除Master(主轴)和Slave(从轴)的关系后执行该指令。

要解除Master(主轴)与Slave(从轴)的关系时，请执行MC_GearOut(齿轮动作解除)指令等。

有关对于主轴的注意事项，请参阅 □ “同步控制的主轴及辅轴的注意事项(P.1-6)”。

● 本指令执行中的其它指令启动

在MC_SetPosition(当前位置变更)指令执行中，执行其它的MC_SetPosition(当前位置变更)指令时，优先执行后一个MC_SetPosition(当前位置变更)指令。

此时，已处于执行中的前一个MC_SetPosition(当前位置变更)指令的Done(完成)变为TRUE，但是无法变更为先前设定的位置。敬请注意。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。



参考

在轴动作中执行本指令时，将会重启当前动作中轴的指令，重新计算定位用数据。此时，如果发生异常，则当前动作中轴的指令也发生异常，本指令不会异常。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 “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	超调值设定	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	设为TRUE时，超调变为有效。 设为FALSE时，超调返回“100%”。
VelFactor	速度超调值	LREAL	0 ~ 500	100	指定速度的超调值。 超调值的有效范围为“0.01 ~ 500.00”。 “500.00以上”作为“500”处理，“0.01以下(包括负数)”作为“0.01”处理。 只有指定为“0”时才作为“0”动作。 单位为[%]。
AccFactor (Reserved)	加减速度超调值	LREAL	0 ~ 500	100	(Reserved)
JerkFactor (Reserved)	跃度超调值	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	控制中变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为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为正常。

* 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Enabled	开始执行本指令时	<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"> • 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含有异常因素时	异常已解除时

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. 请在使用在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

- 变更对轴的目标速度的超调。通过变更超调, 变更动作中的目标速度。
- 可变更超调的指令如下。

MC_Move(定位)指令	MC_MoveJog(微动移动)指令
MC_MoveAbsolute(绝对值定位)指令	MC_MoveFeed(中断标准定位)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。

变更后的目标速度 = 当前执行中指令的目标速度 × 超调

- 超调的单位为[%]。“100”表示“100%”。
- 变更后的目标速度超过轴参数的[最高速度]时, 即为最高速度。
- 相对变更后的目标速度, 使轴加速或减速。
- 将超调指定为“0”时, 目标速度变为“0”, 轴的动作表现为减速, 以速度“0”动作。希望保持动作状态、但又想暂时停止轴动作时, 将超调设为“0”。此时, 轴变量Status.Discrete、Status.Continuous不变。
- 在Enable(有效)为TRUE的状态下执行指令时, 始终反映超调值。
- 利用动作指令从停止中启动、重启运动指令、多重启动运动指令, 可使超调值对应设定的新的目标速度。
- Enable(有效)为FALSE时, 超调返回100(=100%)。
- MC_SetOverride(超调值设定)指令执行中发生轴异常时, MC_SetOverride(超调值设定)指令在Enabled(有效)保持TRUE的状态下持续执行。



使用注意事项

将本指令的Enable(有效)设为FALSE时, 本指令的Enabled(有效)和Busy(执行中)变为FALSE。此时, 以超调“100%”为目标进行加速或减速。



参考

对其它指令的影响

要临时变更各指令的目标速度时, 使用本指令。

因此, 对于输入不带目标速度的指令、如周期同步速度控制那样, 每个周期变更目标速度的指令, 本指令不起作用。

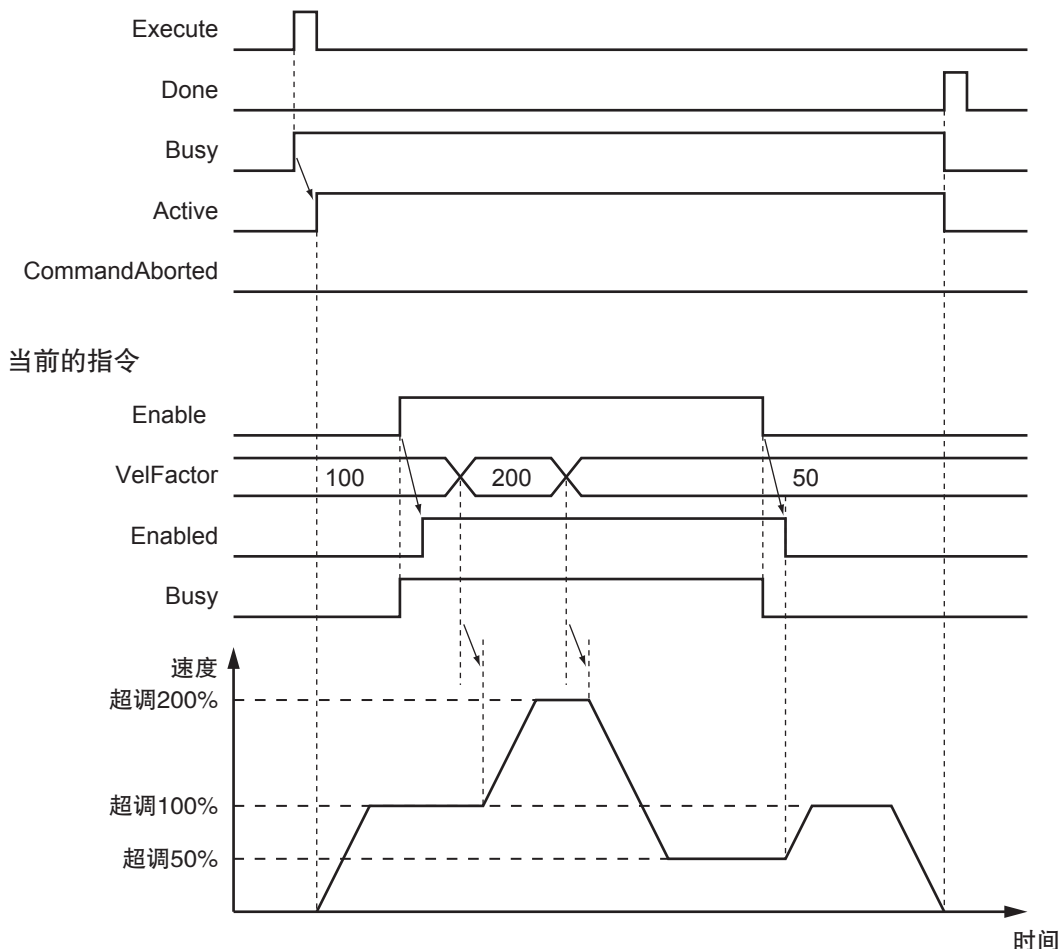
但是, 对于本指令无效的指令, 即使将MC_SetOverride(超调值设定)指令设为有效, Enabled(有效)也会继续保持TRUE。

时序图

● 对MC_MoveAbsolute(绝对值定位)指令的超调

在MC_MoveAbsolute(绝对值定位)指令中使用超调指令时的时序图的示例如下所示。

之前的指令(MC_MoveAbsolute)



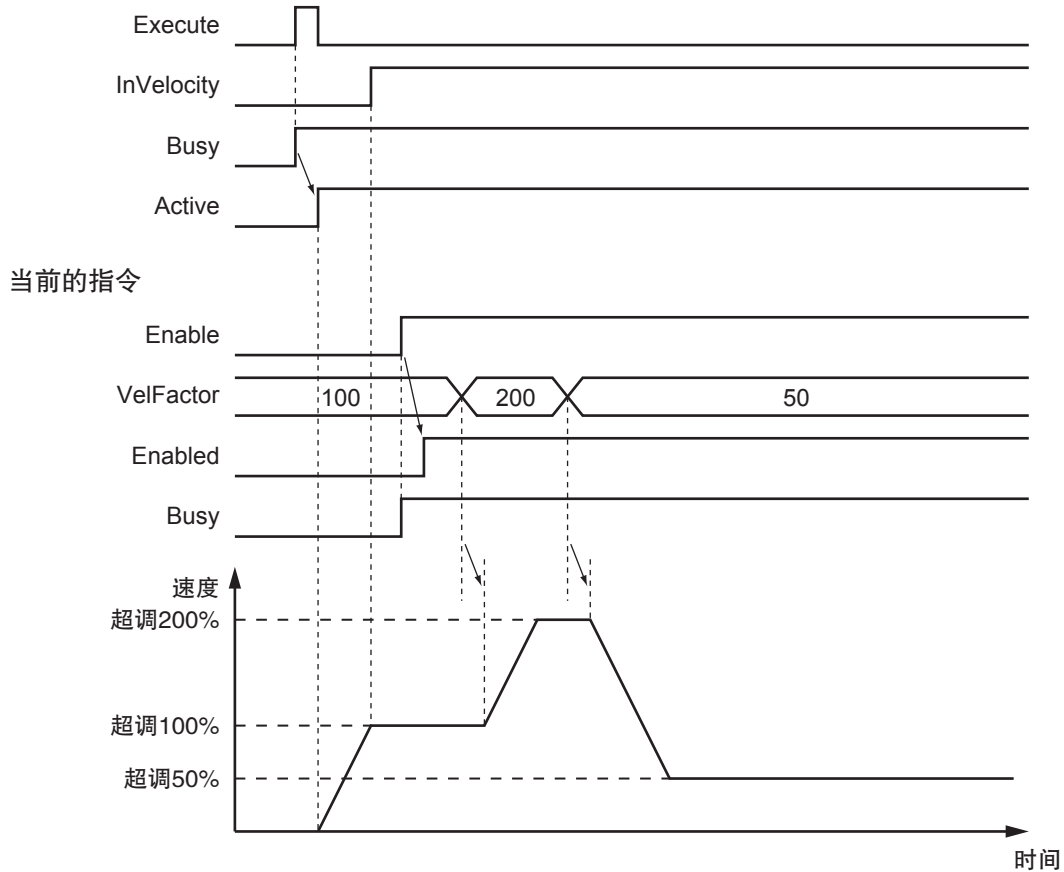
将MC_SetOverride(超调值设定)指令设为无效时, 返回超调100%的目标速度。

● 对MC_MoveVelocity(速度控制)指令的超调

在MC_MoveVelocity(速度控制)指令中使用超调指令时的时序图的示例如下所示。

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状态。

之前的指令(MC_MoveVelocity)



重启运动指令

输入为“Enable型”指令时，不会重启运动指令。

多重启动运动指令

多重启动运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 多重启动多个本指令

在正在执行MC_SetOverride(超调值设定)指令的轴中，启动其它实例的MC_SetOverride(超调值设定)指令时，后执行的实例被优先处理。

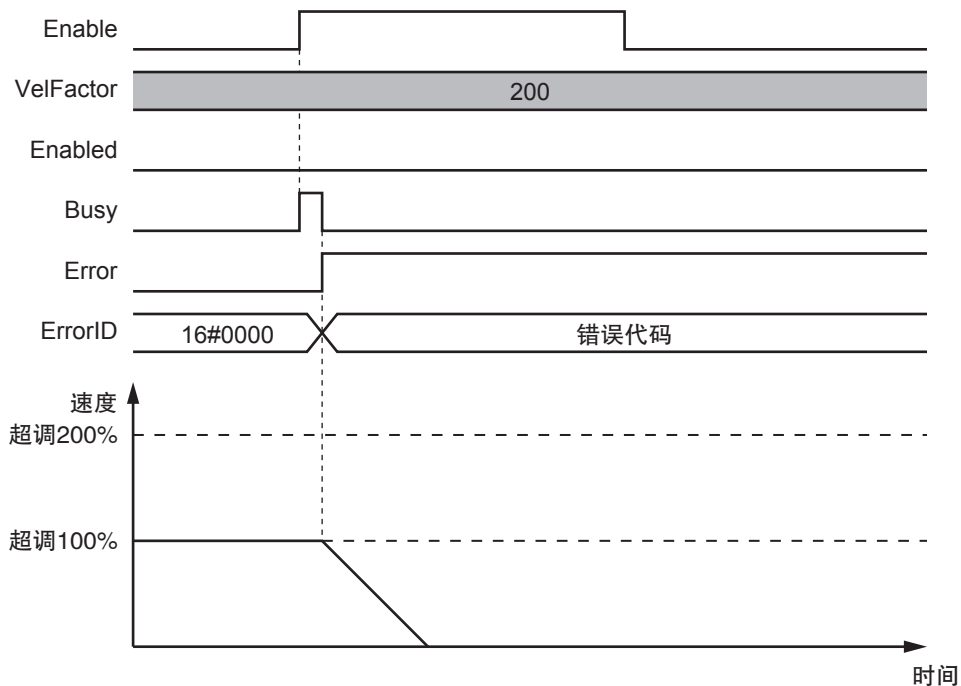
Enabled(有效)对两种指令均表现为TRUE。

作为具体动作，后执行的实例的超调值生效。将后执行的实例的Enable(有效)设为FALSE后，超调无效。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，发生轻度故障时，轴停止动作。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。
排除异常因素时，Error(错误)变为FALSE。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 □□ “A-1 错误代码一览(P.A-2)”。

MC_ResetFollowingError

对指令当前位置和反馈当前位置的偏差进行复位。

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	在上升沿开始指令。
BufferMode	缓存 模式选择	_eMC_BUFFER_ MODE	0: _mcAborting	0 *1	指定多重启动运动指令时的动作。 0: 中断

*1. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

*1. 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	偏差计数器复位完成后	<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	开始指令后	<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"> 利用其它的MC_ResetFollowingError指令多重启动运动指令(中断),中断本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 	<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含有异常因素时	异常已解除时

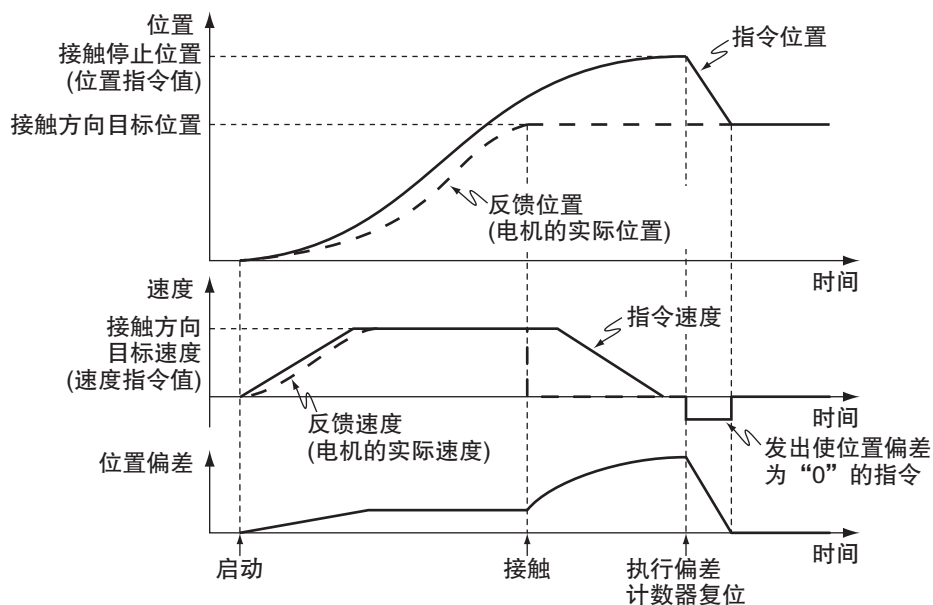
输入输出变量

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. 请使用在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

- 利用本指令，可在周期同步位置模式下，使MC功能模块的指令当前位置和反馈当前位置的偏差为“0”。
 - 检测出Execute(启动)的上升沿时，将该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，在发生位置偏差的接触动作中启动本指令时，朝相反方向发出位置指令，使位置偏差为“0”。
- 对于发生位置偏差的指令，CommandAborted(指令中断)变为TRUE，指令中止执行。



- 将位置偏差设为“0”时，利用轴参数设定的最高速度，发出位置指令。最大加速度、最大减速度不适用。
- 到达新的目标位置而完成指令时，输出变量“Done(完成)”变为TRUE。
- 本指令朝与发生位置偏差的动作相反的方向发出位置指令，但是不适用于轴参数的“反转时动作”。



使用注意事项

- 请在轴速度较低的状态下启动本指令。
本指令朝上一指令(接触方向)相反方向赋予指令值。因此，如果在轴速度较高的状态下启动本指令，可能对机器造成冲击。
- 对于垂直轴等需要持续施加扭矩的轴使用本指令时，请确认在启动本指令后不会发生扭矩不足现象后再使用。
- 使用NX系列脉冲输出单元时，脉冲输出单元连接的伺服驱动器内的偏差无法复位。详情请参阅 □□ “NX系列 位置接口单元 用户手册(SBCE-374)”。
- 针对同步控制的主轴使用本指令时
对于同步控制的主轴，并且将指令位置作为同步数据使用的情况下，针对主轴启动本指令时，从轴根据齿轮比和凸轮数据变量进行反转。敬请注意。
有关对于主轴的注意事项，请参阅 □□ “同步控制的主轴及辅轴的注意事项(P.1-6)”。

指令详情

下面对指令详细说明。

● 可使用轴和启动条件

- 伺服轴或虚拟伺服轴可用于如下场合。
 - 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、虚拟编码器轴则会发生启动时异常。

● 轴变量的状态

轴变量的状态Status.Stopping(减速停止中)变为TRUE。

● 在控制模式切换中启动本指令

执行MC_TorqueControl(转矩控制)指令或MC_SyncMoveVelocity(周期性同步速度控制)指令时，如果在执行控制模式变为位置控制模式的指令，实际切换至位置控制模式之前的时间段内启动本指令，则根据CPU单元的版本，偏差为“0”的时间如下所示。

Ver.1.10以上版本的CPU单元时

切换至位置控制模式后，将指令当前位置和反馈当前位置的偏差设为“0”。

Ver.1.09以下版本的CPU单元时

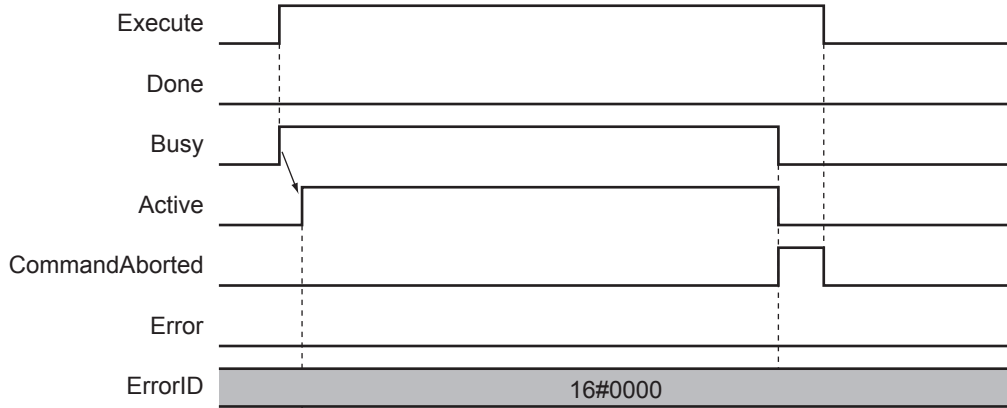
启动本指令时，将指令当前位置和反馈当前位置的偏差设为“0”。

控制模式切换的详情请参阅 □ “MC_TorqueControl(P.3-305)”及 □ “MC_SyncMoveVelocity(P.3-360)”。

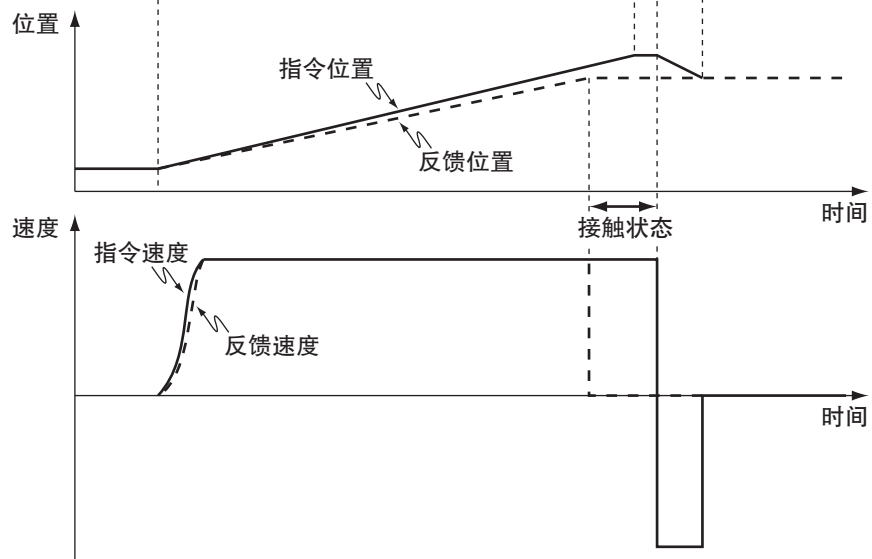
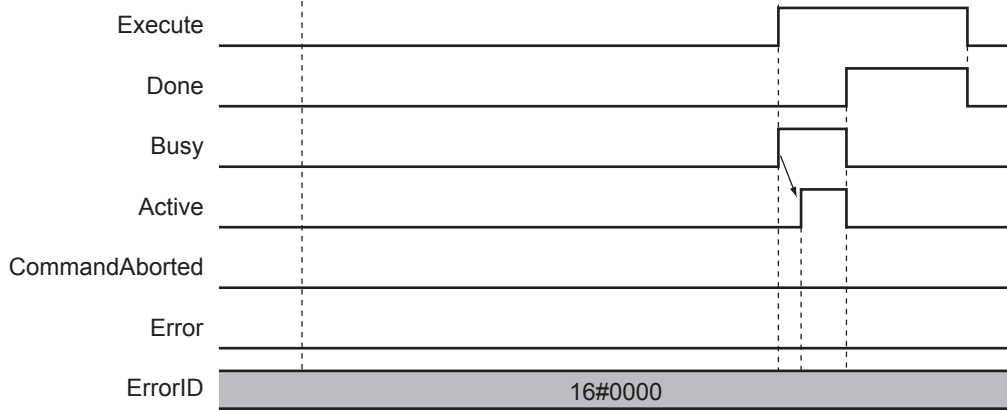
时序图

启动MC_MoveAbsolute(绝对值定位)指令后，在接触状态下启动本指令时的时序图如下所示。

MC_MoveAbsolute指令



MC_ResetFollowingError指令



重启运动指令

无法重启本指令。

重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动

对当前正在执行的指令启动本指令，可切换或缓存到本指令。

各轴可缓存到1个。

多重启动时的本指令的动作由BufferMode(缓存模式选择)指定。

缓存模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。

关于BufferMode(缓存模式选择)，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

与MC_Stop(强制停止)指令的关系

MC_Stop(强制停止)指令执行中启动MC_ResetFollowingError(偏差计数器复位)指令时，MC_Stop(强制停止)指令的Done变为TRUE，执行MC_ResetFollowingError(偏差计数器复位)指令。

● 本指令执行中的其它指令启动

本指令执行中处于减速停止中。

因此，请启动可在减速停止中多重启动的指令。

对于无法启动的指令，本指令的CommandAborted(执行中断)变为TRUE。

与MC_Stop(强制停止)指令的关系

MC_ResetFollowingError(偏差计数器复位)指令执行中启动MC_Stop(强制停止)指令时，MC_ResetFollowingError(偏差计数器复位)指令的CommandAborted(执行中断)变为TRUE，执行MC_Stop(强制停止)指令。但是，MC_Stop(强制停止)指令的减速度不适用，表现为立即停止。

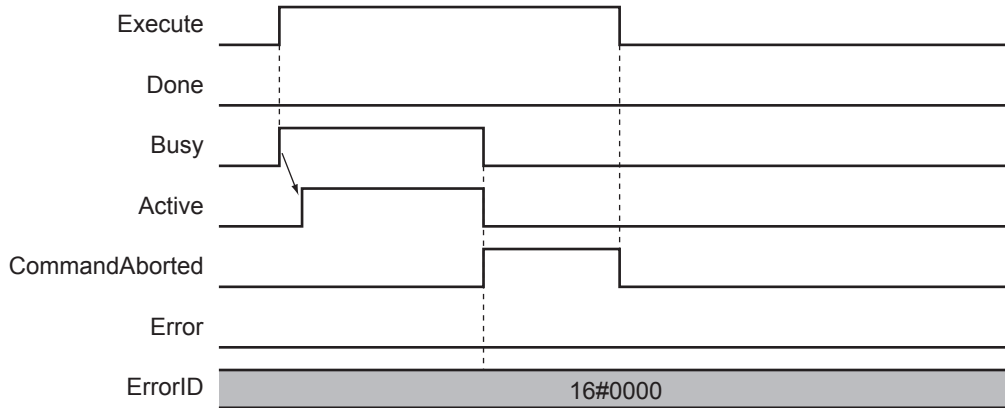
异常

本指令执行中发生异常时，立即停止，CommandAborted(执行中断)变为TRUE。此外，启动本指令时检测到异常时，Error(错误)变为TRUE。

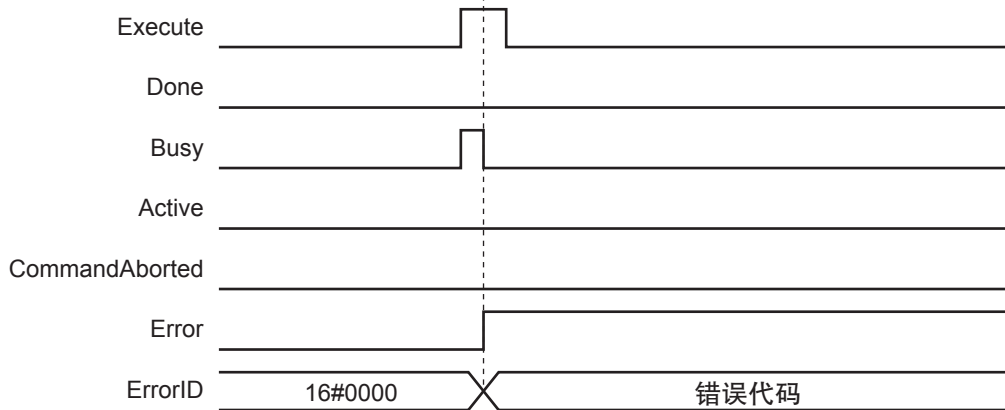
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

启动MC_MoveAbsolute(绝对值定位)指令后，在接触状态下启动本指令发生异常时的时序图如下所示。

MC_MoveAbsolute指令



MC_ResetFollowingError指令



● 错误代码

关于指令发生的异常，请参阅 □ “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	在上升沿开始指令。
Periodic	重复模式	BOOL	TRUE, FALSE	FALSE	指定是反复执行指定的凸轮表，还是只执行1次。 TRUE：重复 FALSE：不重复
StartMode	起始位置方式指定	_eMC_ START_MODE	0: _mcAbsolutePosition 1: _mcRelativePosition	0 *1	指定MasterStartDistance(主轴跟踪距离)使用的坐标系。 0：绝对位置 1：相对位置
StartPosition	凸轮表起始位置	LREAL	负数、正数、“0”	0	利用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	主轴跟踪距离	LREAL	负数、正数、“0”	0	指定从轴开始凸轮动作时主轴的位置。 StartMode(起始位置方式指定)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,以StartPosition(凸轮表起始位置)的相对量指定。 单位为[指令单位]。*2
Master Scaling	主轴系数	LREAL	正值(>0.0)	1.0	以指定的比例放大/缩小主轴的相位。
SlaveScaling	从轴系数	LREAL	正值(>0.0)	1.0	以指定的比例放大/缩小从轴的位移。
MasterOffset	主轴偏置	LREAL	负数、正数、“0”	0	以指定的偏移值移动主轴的相位。
SlaveOffset	从轴偏置	LREAL	负数、正数、“0”	0	以指定的偏移值移动从轴的位移。
ReferenceType	位置类型选择	_eMC_REFERENCE_TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 *1	指定主轴的位置类型。 0: 指令位置(最近任务周期*3下的计算值) 1: 反馈位置(同一任务周期*3下的取值) 2: 指令位置(同一任务周期*3下的计算值)
Direction	方向选择	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection 4: _mcNoDirection	4 *1	仅主轴沿指定方向动作时,从轴进行凸轮动作。主轴沿指定方向的反方向动作时,从轴停止凸轮动作。 0: 指定为正方向 2: 指定为负方向 4: 无方向指定
CamTransition (Reserved)	凸轮过渡选择	_eMC_CAM_TRANSITION	0: _mcCTNone	0 *1	(Reserved)
BufferMode	缓存模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow *4 3: _mcBlendingPrevious *4 4: _mcBlendingNext *4 5: _mcBlendingHigh *4	0 *1	指定多重启动运动指令时的动作。 0: 中断 1: 等待 2: 以低速合并 3: 以前一个速度合并 4: 以后一个速度合并 5: 以高速合并

*1. 有效范围为枚举体的变量,其实际初始值不是数值,而是枚举元素。

*2. 关于指令单位,请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”的“单位转换设定”。

*3. 任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先级5)的周期。

*4. Ver.1.06以上的CPU单元和Ver.1.07以上的Sysmac Studio组合时可使用。无论指定4种合并中的哪一种动作均相同。

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	-	执行凸轮表起点后变为TRUE。
InSync	同步中	BOOL	TRUE, FALSE	-	与主轴同步时变为TRUE。
EndOfProfile	凸轮周期完成	BOOL	TRUE, FALSE	-	执行凸轮表终点后变为TRUE。
Index	索引	UINT	0或 正值	-	输出凸轮数据的索引编号。 ^{*1}
Busy	执行中	BOOL	TRUE, FALSE	-	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	-	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	-	指令中止时, 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为正常。

*1. InCam(凸轮动作中)为FALSE期间, 输出变为“0”。

*2. □ 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
InSync	从轴开始凸轮动作时	<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	作为指令位置, 输出凸轮表终点的相位和位移的周期	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(凸轮表起始位置)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	开始指令后	<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"> • 利用其它指令多重启动运动指令(中断), 中止本指令时 • 因发生异常, 中止本指令时 • 发生异常过程中, 启动本指令时 • 执行MC_Stop指令中, 启动本指令时 • 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含有异常因素时	异常已解除时

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型的数组变量作为凸轮表指定。 ^{*2}

*1. 请使用在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. 数组元素[N]通过Sysmac Studio自动设定。请指定通过Sysmac Studio的凸轮编辑器创建的凸轮数据变量。



使用注意事项

主轴和从轴指定同一轴时，会发生轻度故障等级的“主轴从轴相同(错误代码：5436 Hex)”的异常。

功能说明

- 本指令可使相位(主轴)和位移(从轴)按照凸轮表进行同步凸轮动作。
- 由本指令指定的凸轮表，必须事先使用凸轮编辑器编制，并下载到控制器主体中。
- 在Execute(启动)的上升沿，本指令启动。



使用注意事项

凸轮表必须使用由Sysmac Studio的凸轮编辑器编制的凸轮数据变量。



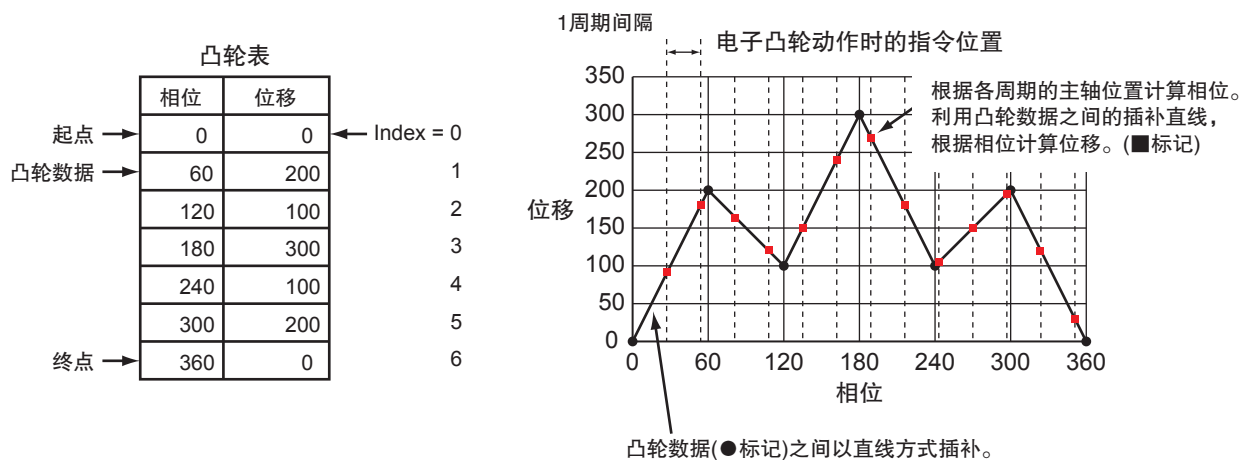
参考

“下载”使用Sysmac Studio的“同步”功能。

关于凸轮表，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

凸轮表的相位和位移均以起点“0.0”起的相对量指定。关于主轴、从轴的指令位置，则以直线方式对2个凸轮数据之间进行插补，求得相当于相位(主轴)的位移(从轴)。

凸轮数据数较少时，相位间隔增大，表现为精度较低的凸轮动作。相反，凸轮数据数较多时，相位间隔减小，表现为精度较高的凸轮动作。



使用注意事项

- Ver.1.09以下的CPU单元使用本指令时，请勿对Master(主轴)执行MC_SetPosition(当前位置变更)指令。对Master(主轴)执行MC_SetPosition(当前位置变更)指令时，可能出现Slave(从轴)急速跟踪的情况，非常危险。
要对Master(主轴)使用MC_SetPosition(当前位置变更)指令时，请解除Master(主轴)和Slave(从轴)的关系后执行该指令。
有关对于主轴的注意事项，请参阅 □ “同步控制的主轴及辅轴的注意事项(P.1-6)”。
- 通过本指令，可以确认启动时相位是否表现为升序排列。相位表现为升序排列时，检测到异常。
启动本指令后，如果确认在变更凸轮数据后相位表现为升序排列，请执行MC_SetCamTable Property(更新凸轮表属性)指令。
要在凸轮动作中变更相位，请确认处于升序状态后再进行变更。如果并非升序，则凸轮动作可能会中途停止。
- 凸轮数据变量为全局变量，可以从多个任务中查看或变更凸轮数据变量值。从多个任务中变更凸轮数据变量值时，请编写通过多个任务进行改写处理时相互不会发生冲突的程序。
- 使用“全局变量的任务间排他性控制”功能进行凸轮数据变量的排他性控制时，请不要使用排他对象的凸轮数据变量启动运动指令。否则会发生轻度故障等级的异常“凸轮表指定错误(错误代码：5439 Hex)”。

指令详情

下面对指令详细说明。

● 指令启动条件

在主轴停止中、或位置控制中、速度控制中、同步控制中的任一状态下都可以启动本指令。
从轴请参阅 □ “重启运动指令(P.3-184)”、□ “多重启动运动指令(P.3-185)”。

● 软件限位

在凸轮动作中从轴超过软件限制时，会发生异常。

● 凸轮数据变量

凸轮数据变量对凸轮数据结构体进行数组宣言并使用。凸轮数据结构体事先进行的类型宣言如下所示。

```

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上生成凸轮数据变量。

凸轮数据变量名称的凸轮表名称可任意取名。

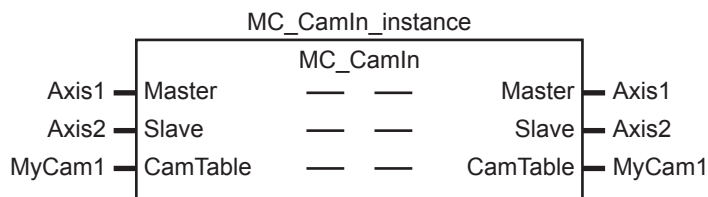
例如，凸轮表名称 “MyCam1”、凸轮数据数1000个时，在Sysmac Studio中自动声明变量。如下所示。

```

VAR
  (* 凸轮表 *)
  MyCam1      :   ARRAY [0..999] OF _sMC_CAM_REF;
END_VAR

```

通过本指令指定 “MyCam1” 时，记述如下。其中，将主轴设为Axis1、从轴设为Axis2。



控制器中不存在指定凸轮表时，检测到异常。此外，多根轴可指定同一个凸轮表。

可通过用户程序改写凸轮数据变量的值。但是，改写值在控制器电源OFF、通过Sysmac Studio执行下载时消失。

接通CPU单元电源后，以及下载完成后，返回通过Sysmac Studio下载的值。要保存改写值，请使用MC_Save CamTable(保存凸轮表)指令。

CPU单元的动作模式变更时，保存改写凸轮数据变量的值。



参考

- 凸轮数据变量不在网络上公开。
例如，“MyCam1[10].Phase”和“MyCam1[10].Distance”，虽然可在Sysmac Studio的Watch View中浏览，但无法通过EIP从其他控制器访问。
- “下载”使用Sysmac Studio的“同步”功能。

● 凸轮动作的开始

启动指令后，等待主轴到达StartPosition(凸轮表起始位置)。

主轴通过StartPosition(凸轮表起始位置)时，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以零起点的相对量指定。因此，各相位各轴的绝对位置，是以凸轮表起点各轴的绝对位置为起点的相对值。例如，主轴的计数模式为0~360°的旋转模式时，StartPosition(凸轮表起始位置)=60。如下所示，主轴的绝对位置是凸轮表的相位加上StartPosition的值，从轴的绝对位置是凸轮表的位移加上凸轮表起点的从轴绝对位置的值。

凸轮表		StartPosition=60	各轴的绝对位置	
相位	位移		主轴	从轴
起点 → 0	0	→	60	0+凸轮表起点的从轴绝对位置
凸轮数据 → 60	200		120	200+凸轮表起点的从轴绝对位置
120	100		180	100+凸轮表起点的从轴绝对位置
180	300		240	300+凸轮表起点的从轴绝对位置
240	100		300	100+凸轮表起点的从轴绝对位置
300	200		360	200+凸轮表起点的从轴绝对位置
360	0		60	0+凸轮表起点的从轴绝对位置
终点 →				

并且，通过MasterStartDistance(主轴跟踪距离)时，开始从轴的凸轮动作，输出变量InSync(同步中)变为TRUE。

MasterStartDistance(主轴跟踪距离)以绝对位置或StartPosition(凸轮表起始位置)起的相对距离来指定。绝对位置或相对位置通过StartMode(起始位置指定方式)来设定。

例1)从轴动作因MasterStartDistance的不同而异

本例中，使用相同凸轮表、相同主轴。

凸轮表的设定如下。

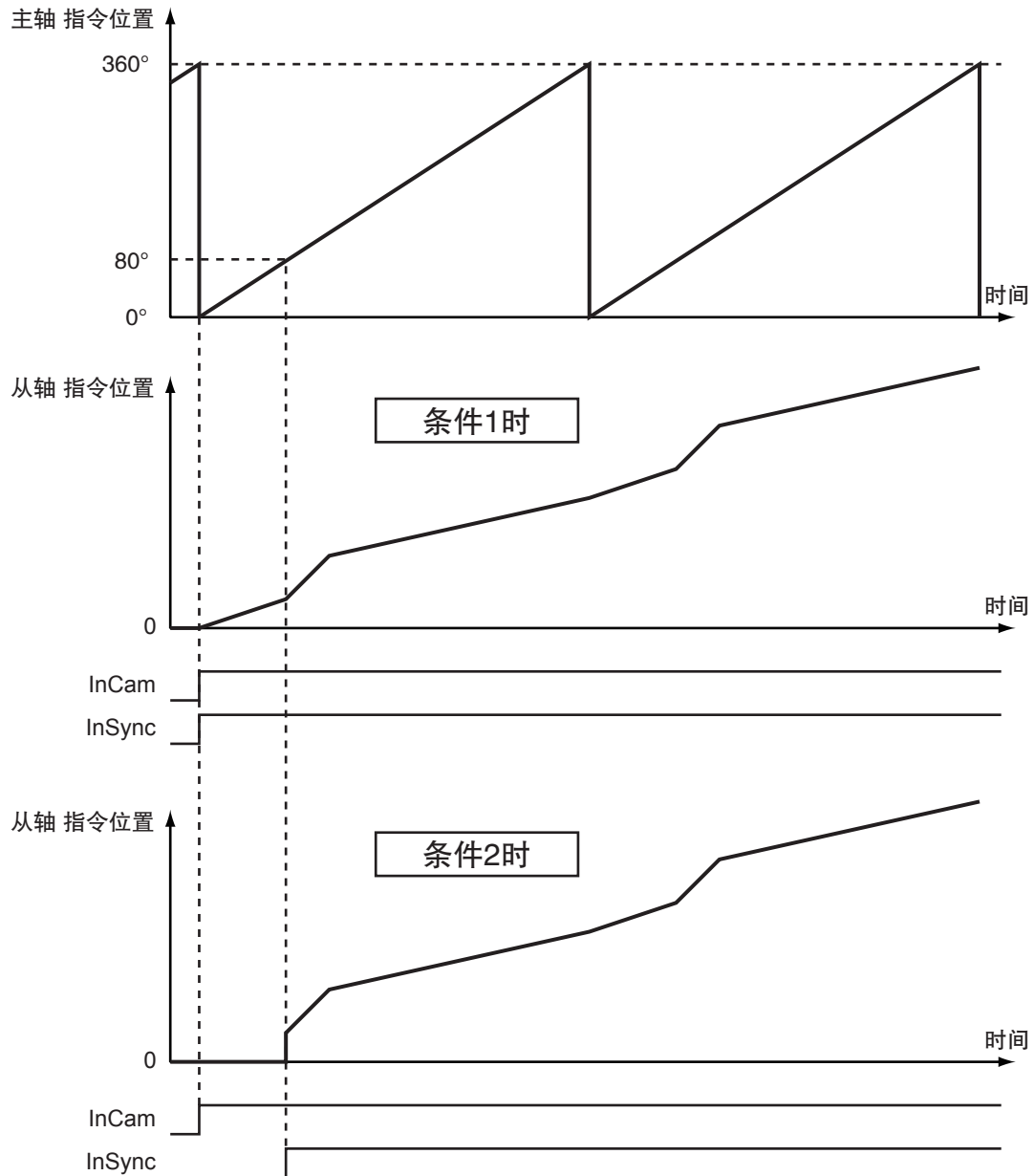
主轴	从轴	凸轮曲线	连接速度	连接加速度	相位间隔宽度
0.000	0.000	-	-	-	-
80.000	80.000	直线	360.000	0.000	0.010
120.000	200.000	直线	1080.000	0.000	0.010
360.000	360.000	直线	240.000	0.000	0.010

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(起始位置方式指定)	_mcRelativePosition(相对位置)	_mcRelativePosition(相对位置)
StartPosition(凸轮表起始位置)	0	0
MasterStartDistance(主轴跟踪距离)	0	80

条件1下，主轴通过0°时，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下，主轴通过0°时，输出变量InCam(凸轮动作中)变为TRUE，并且，通过80°*¹时，输出变量InSync(同步中)变为TRUE，从轴开始凸轮动作。条件2时，从凸轮表的中途开始凸轮动作，从轴急剧加速。

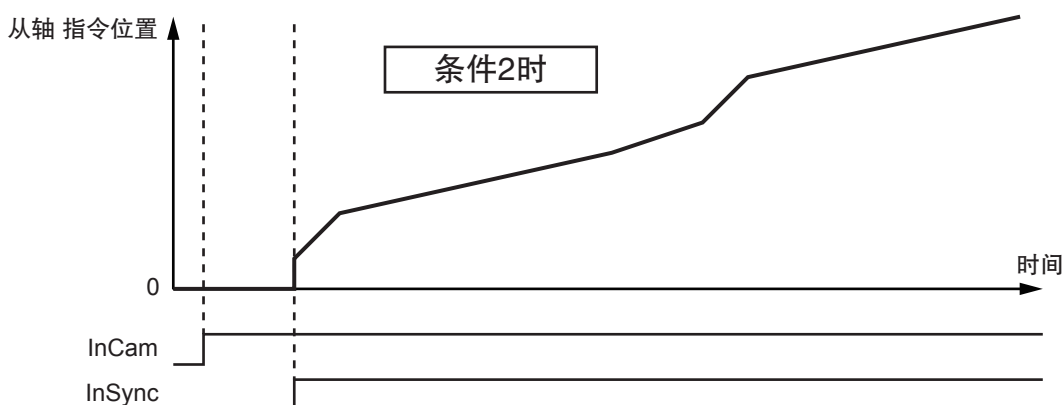


*1. StartMode为_mcRelativePosition，因此StartPosition + MasterStartDistance=80°为凸轮动作起始位置。

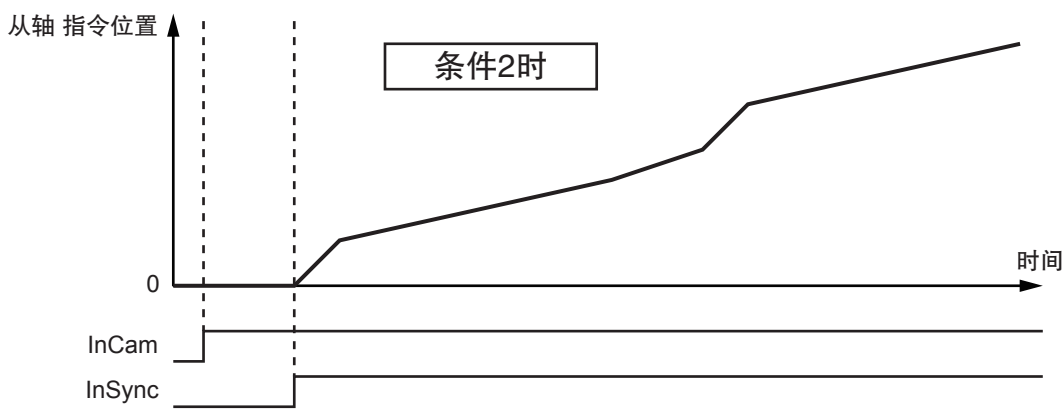


参考

条件2时，若设定SlaveOffset = 0，则从轴将如下图所示急剧加速。



若设定SlaveOffset = -80，则从轴将根据MasterStartDistance(主轴跟踪距离)的位移开始同步，而不会急剧加速。



例2)从轴动作因StartPosition和MasterStartDistance的不同而异

凸轮表的设定与前述示例相同。凸轮动作的开始条件如下。

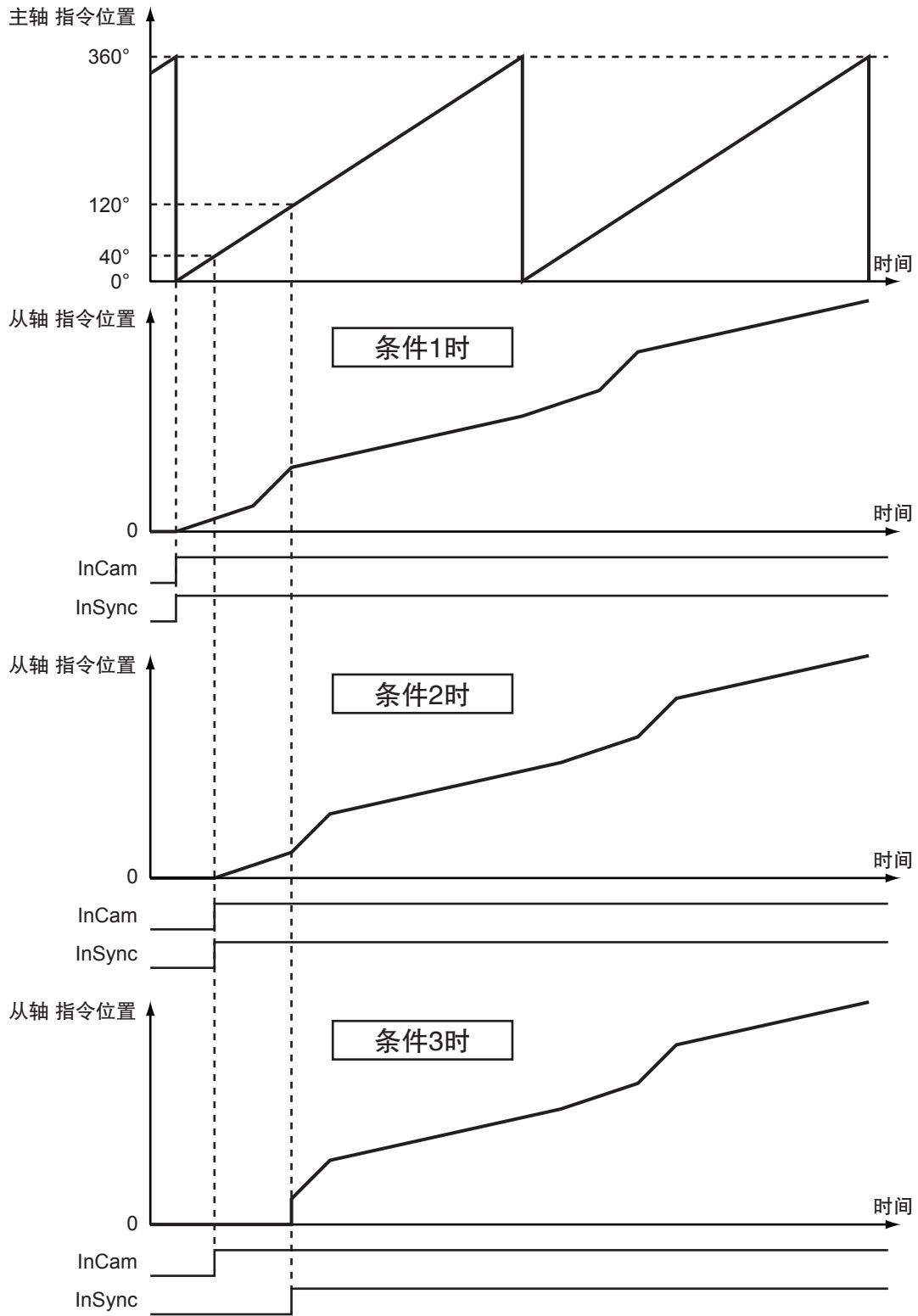
输入变量	条件1	条件2	条件3
Periodic(重复模式)	TRUE(重复)	TRUE(重复)	TRUE(重复)
StartMode(起始位置方式指定)	_mcRelativePosition (相对位置)	_mcRelativePosition (相对位置)	_mcRelativePosition (相对位置)
StartPosition(凸轮表起始位置)	0	40	40
MasterStartDistance(主轴跟踪距离)	0	0	80

条件1下，主轴通过0°时，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下，主轴通过StartPosition(凸轮表起始位置)指定的40°时，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下，主轴通过40°时，输出变量InCam(凸轮动作中)变为TRUE，并且，通过120°*1时，输出变量InSync(同步中)变为TRUE，从轴开始凸轮动作。

*1. StartMode为_mcRelativePosition，因此StartPosition + MasterStartDistance=120°为凸轮动作起始位置。



例3)从轴凸轮动作开始因StartMode(起始位置方式指定)的不同而异

通过StartMode(起始位置方式指定)，可以指定将MasterStartDistance(主轴跟踪距离)的指定值作为绝对位置处理，还是作为相对位置处理。

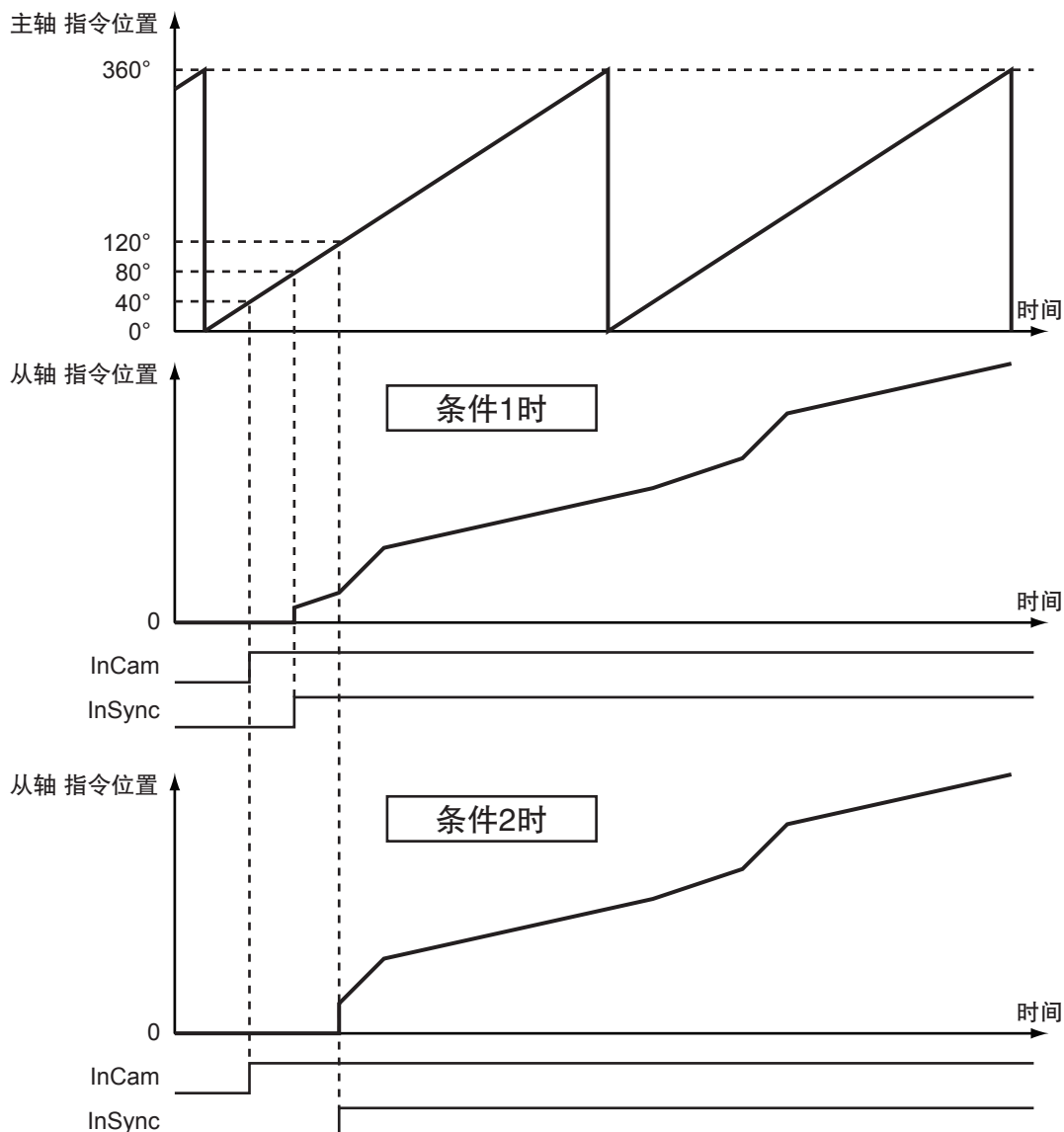
下面对因StartMode(起始位置方式指定)的不同导致的从轴凸轮动作开始的不同进行说明。凸轮表的设定与前述示例相同。

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(起始位置方式指定)	_mcAbsolutePosition(绝对位置)	_mcRelativePosition(相对位置)
StartPosition(凸轮表起始位置)	40	40
MasterStartDistance(主轴跟踪距离)	80	80

条件1和条件2下，主轴通过40°时，输出变量InCam(凸轮动作中)均变为TRUE。条件1下，由于StartMode(起始位置方式指定)为_mcAbsolutePosition(绝对位置)，因此，主轴通过80°时，输出变量InSync(同步中)变为TRUE，从轴开始凸轮动作。

条件2下，由于StartMode(起始位置方式指定)为_mcRelativePosition(相对位置)，因此，主轴通过120°(=40°+ 80°)时，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

为执行凸轮动作，必须事先使用Sysmac Studio的凸轮编辑器绘制凸轮曲线，并下载到CPU单元中。“下载”使用Sysmac Studio的“同步”功能。

● Periodic(重复模式)

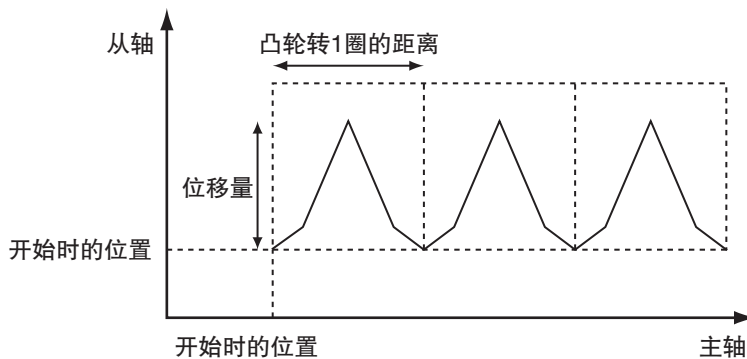
在Periodic(重复模式)中指定TRUE[重复]时，从凸轮表的起点到终点，重复执行凸轮动作。

指定FALSE[不重复]时，如果持续执行到凸轮表的终点，则结束凸轮动作。

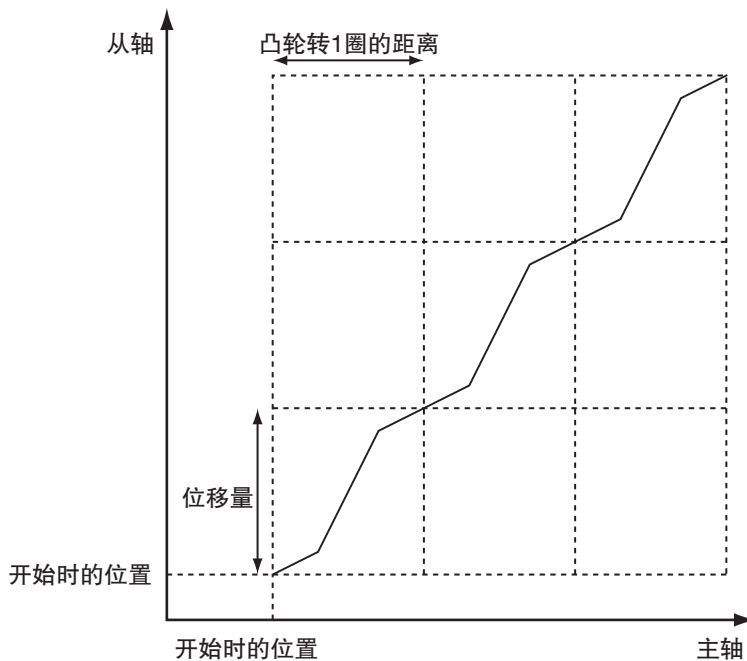
指定TRUE[重复]时，如果在凸轮表的起点和终点从轴的行程位置相同，则作为往复凸轮进行动作(参阅往复凸轮动作)。如果在起点和终点从轴的行程位置不同，则作为传送凸轮进行动作(参阅传送凸轮动作)。

在下图中，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)

在输出凸轮动作的指令值的一个周期内，将EndOfProfile(凸轮周期完成)设为TRUE。该凸轮动作使用了以凸轮表终点定义的相位、位移。

对StartPosition(凸轮表起始位置)设定主轴的绝对位置，凸轮表即为该位置起的相对表。

EndOfProfile(凸轮周期完成)即为表现凸轮表终端的输出。

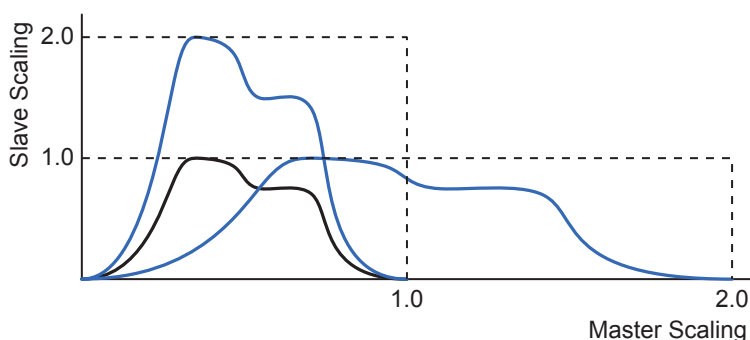
● 凸轮动作的结束

要在中途结束凸轮动作，可使用MC_CamOut(解除凸轮动作)指令、或MC_Stop(强制停止)指令。

● 系数(缩放)

针对指定凸轮表的主轴相位、从轴位移，可按照指定比例进行缩放。

主轴、从轴都可以设定不同的倍率。

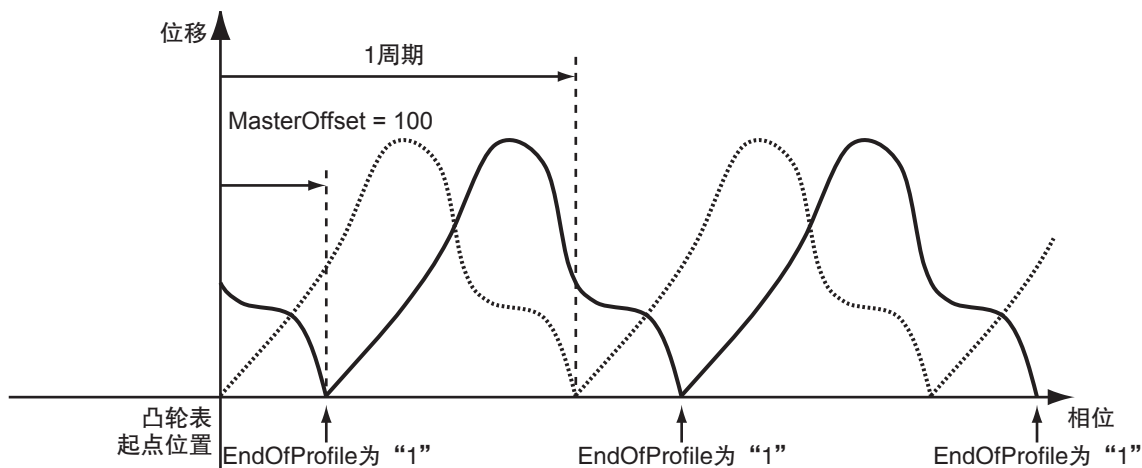


● 偏置

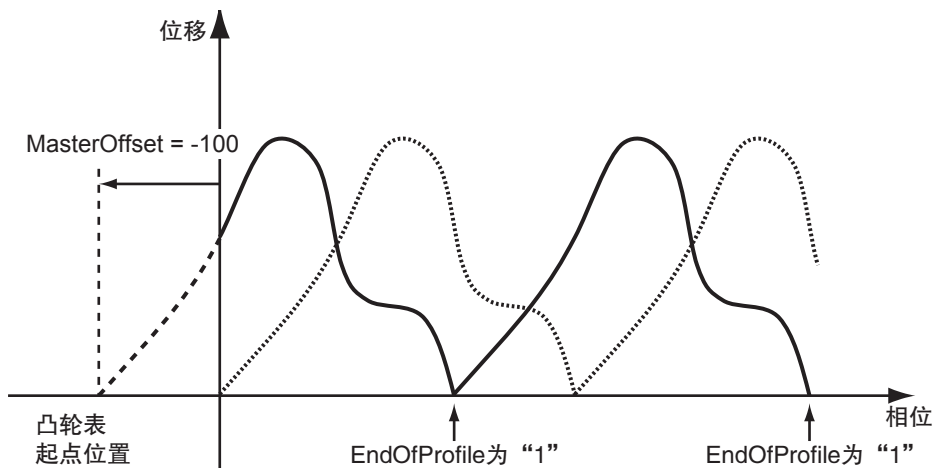
针对指定凸轮表，可按照偏置量移动相位和位移的功能。

对主轴相位、从轴位移可指定不同的偏置量。

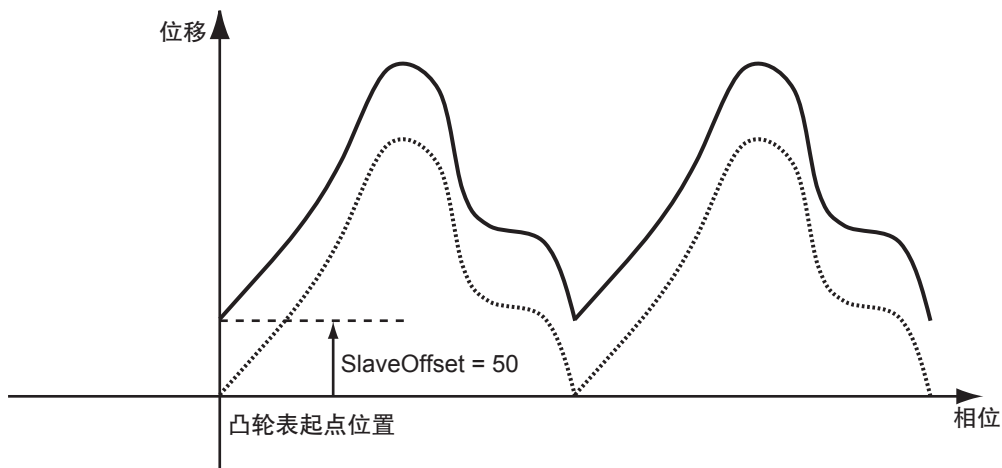
MasterOffset(主轴偏置)> 0 时



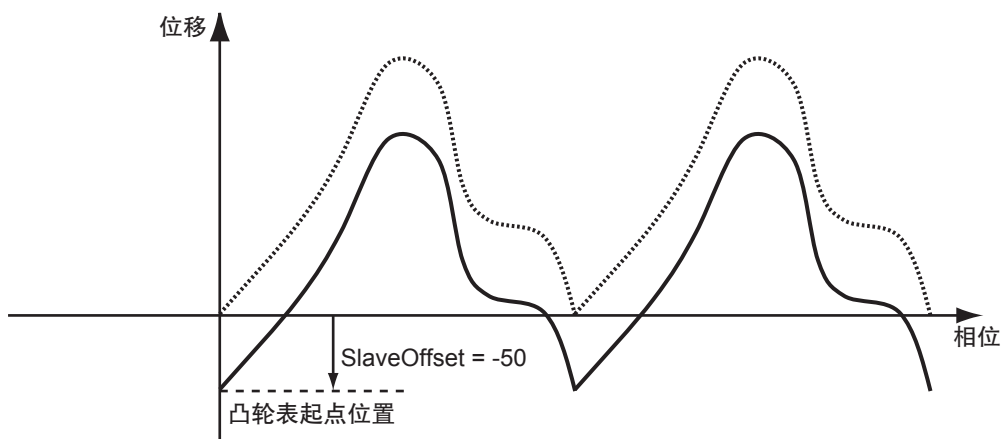
MasterOffset(主轴偏置)< 0 时



SlaveOffset(从轴偏置)> 0 时



SlaveOffset(从轴偏置)< 0 时



● ReferenceType(位置类型选择)

可从下列选项中选择从轴同步的主轴位置类型。

- `_mcCommand`: 指令位置(最近任务周期下的计算值)
对当前周期, 使用之前任务周期时计算的主轴指令位置。
在计算从轴指令位置之前的固定周期任务中, 使用计算的主轴指令位置。
- `_mcFeedback`: 同一任务周期下的取值
使用同一任务周期时获取的主轴反馈位置。
- `_mcLatestCommand`: 指令位置(同一任务周期下的计算值)
使用同一任务周期时计算的主轴指令位置。
也可通过`_mcCommand`使用最新信息。但是, 设定时必须保证主轴编号比从轴编号小。
否则, 本指令的Error(错误)将变为TRUE。ErrorID(错误代码)输出“主轴/从轴 轴号非升序(错误代码: 5438 Hex)”。



使用注意事项

这里的任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先度5)的周期。同样, 固定周期任务表示原始恒定周期任务或固定周期任务(执行优先度5)。



参考

相比选择之前任务周期时的计算值, 选择同一任务周期时计算的指令位置时, 同步精度更高。但是必须使用主轴、从轴, 确保“Master(主轴)设定的运动控制系统变量的轴号 < Slave(从轴)设定的运动控制系统变量的轴号”。

● 轴种类与位置类型的关系

可监控的轴种类和要监控的位置类型的关系如下所示。

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	○

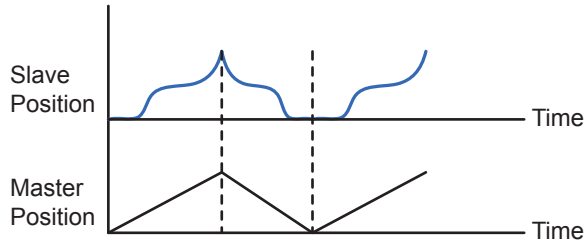
*1. 启动指令时, 发生“超过位置类型选择范围(错误代码: 5430 Hex)”错误。

● Direction(方向选择)

只有主轴移动方向与Direction(方向选择)一致时，才可开始从轴的凸轮动作。
Direction(方向选择)有效即为InSync(同步中)中。

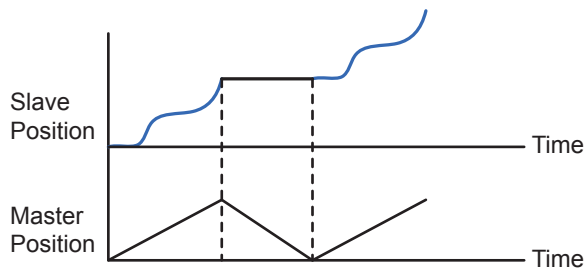
_mcNoDirection(无方向指定)

无论主轴正负方向移动，均开始凸轮动作。



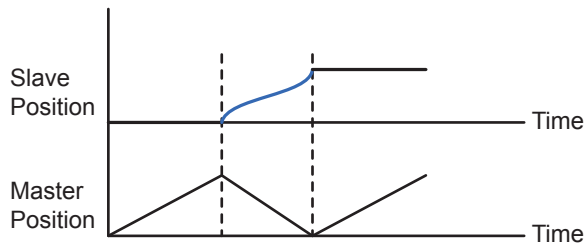
_mcPositiveDirection(指定为正方向)

主轴正方向移动时开始凸轮动作



_mcNegativeDirection(指定为负方向)

主轴负方向移动时开始凸轮动作



版本相关信息

主轴沿与Direction(方向旋转)相反方向移动期间, 超过MasterStartDistance(主轴跟踪距离)、InSync(同步中)变为TRUE时, 不同CPU单元版本的动作如下所示。

- Ver.1.10以上版本的CPU单元时

InSync(同步中)变为TRUE后, 从轴根据凸轮表相位对应的位移进行移动。之后, 主轴沿Direction(方向选择)同向移动时, 从轴开始凸轮动作。

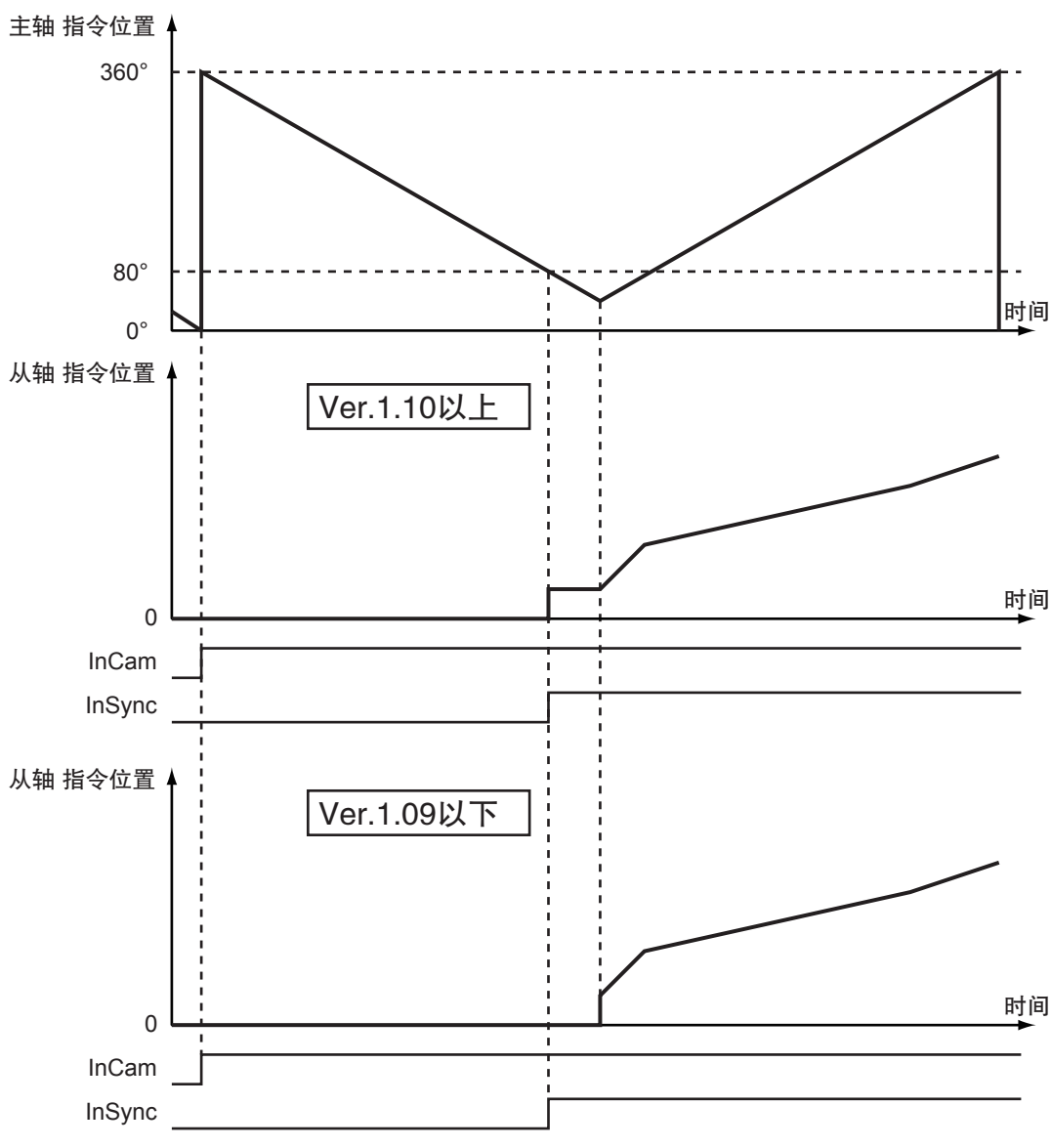
- Ver.1.09以下版本的CPU单元时

InSync(同步中)变为TRUE后, 主轴沿Direction(方向选择)同向移动时, 从轴开始凸轮动作。

示例如下所示。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(起始位置方式指定)	_mcAbsolutePosition(绝对位置)
Direction(方向选择)	_mcPositiveDirection(指定为正方向)
StartPosition(凸轮表起始位置)	0
MasterStartDistance(主轴跟踪距离)	80



● BufferMode(缓存模式选择)

指定前一个轴动作和本次动作的连接方式。

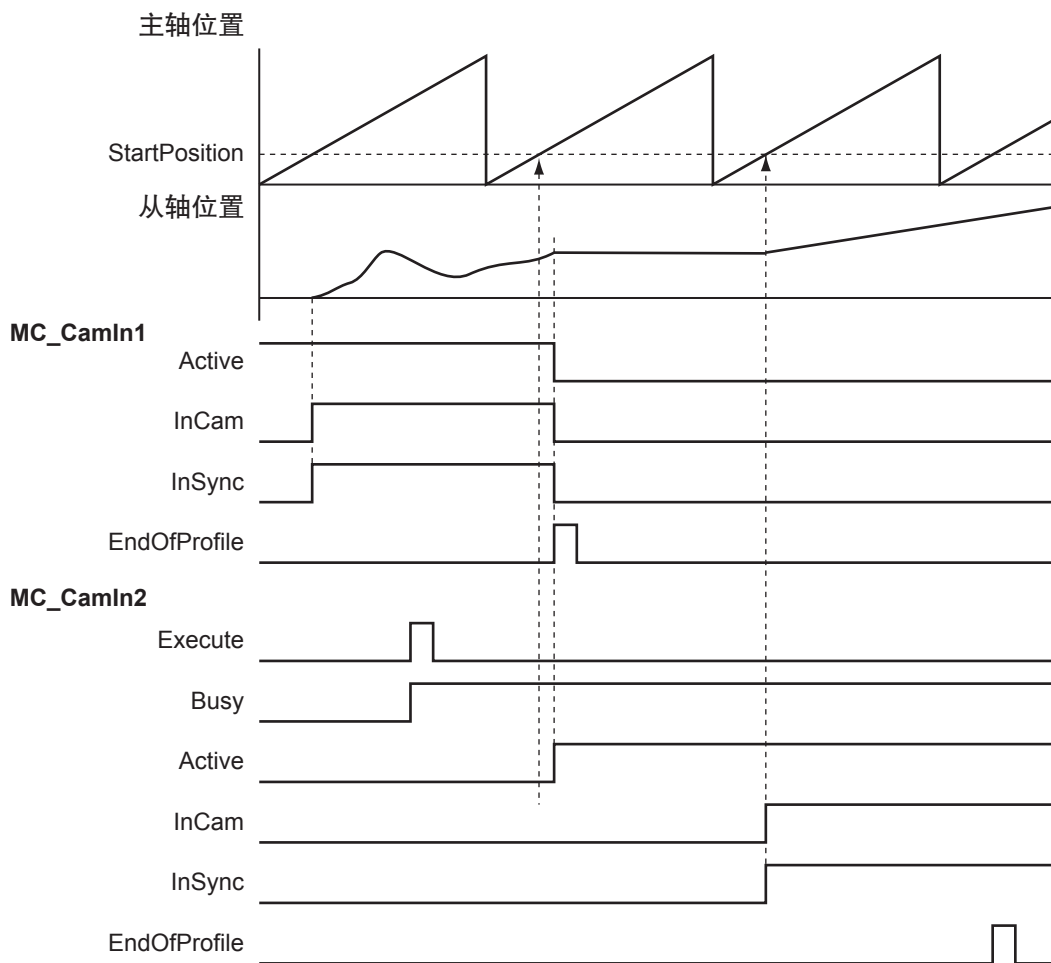
有以下6种。

缓存模式选择	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(凸轮表起始位置)后，通过MasterStartDistance(主轴跟踪距离)时，开始从轴的凸轮动作，输出变量InSync(同步中)变为TRUE。 从轴停止，直至已缓存的本指令的输出变量InSync(同步中)变为TRUE。 凸轮表起点为StartPosition(凸轮表起始位置)。
等待	从当前执行中的指令正常结束的周期起，已缓存的本指令自动启动。主轴通过StartPosition(凸轮表起始位置)后，通过MasterStartDistance(主轴跟踪距离)时，开始从轴的凸轮动作，输出变量InSync(同步中)变为TRUE。 从轴停止，直至已缓存的本指令的输出变量InSync(同步中)变为TRUE。 凸轮表起点为StartPosition(凸轮表起始位置)。
合并*1	从当前执行中的指令正常结束的周期起，已缓存的本指令的输出变量InSync(同步中)变为TRUE，从轴不停止而直接开始凸轮动作。
以低速合并	即使指定了已缓存的本指令的StartPosition(凸轮表起始位置)、
以前一个速度合并	MasterStartDistance(主轴跟踪距离)，无论指定值如何，从轴都将在指令启动的同时开始凸轮动作。
以后一个速度合并	凸轮表起点为当前执行中的指令的结束位置。
以高速合并	

*1. Ver.1.06以上的CPU单元和Ver.1.07以上的Sysmac Studio组合时可使用。无论指定4种合并中的哪一种动作均相同。

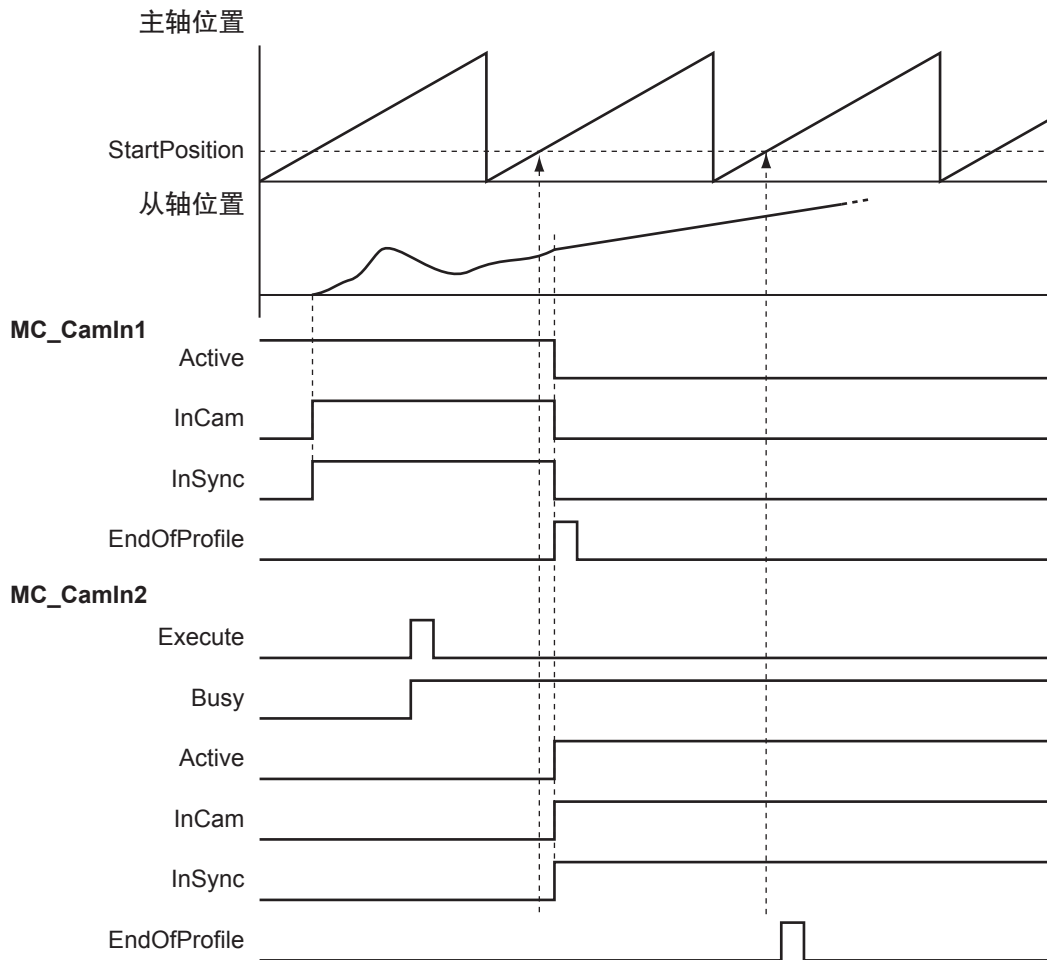
● 指定等待多重启动时

执行MC_CamIn1期间，指定等待后多重启动MC_CamIn2。在MC_CamIn1结束、MC_CamIn2的Active(控制中)变为TRUE的下一StartPosition(凸轮表起始位置)，InSync(同步中)变为TRUE，开始凸轮动作。



● 指定合并多重启动时

执行MC_CamIn1期间，指定合并后多重启动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。



关于BufferMode(缓存模式选择)，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 索引

从为了求得主轴、从轴的指令位置而使用的两个凸轮数据中，通过输出变量“Index(索引)”输出凸轮数据索引编号较小者。

通过凸轮编辑器或用户程序对凸轮数据进行微调时使用该数值。

● 到位检查

本指令不执行到位检查。

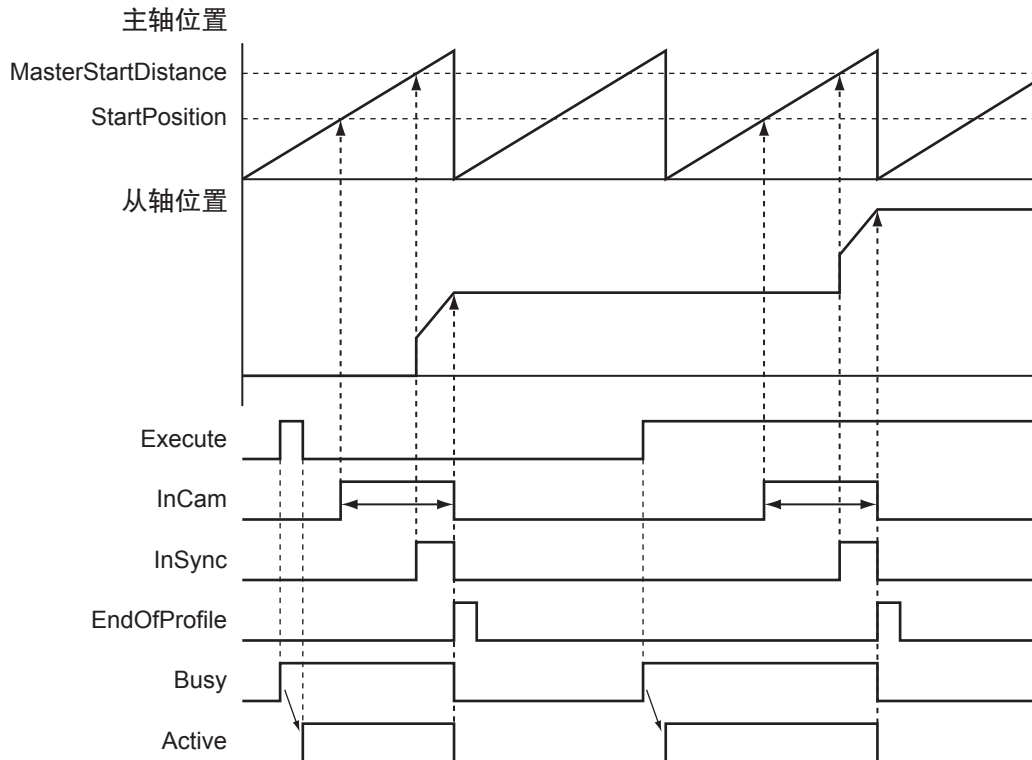
● 超调

使用本指令时，无法通过MC_SetOverride(超调值设定)指令进行超调。

时序图

● 无重复时

MC_CamIn(凸轮动作开始)指令的Periodic(重复模式)为FALSE[无重复]时,如下图所示。



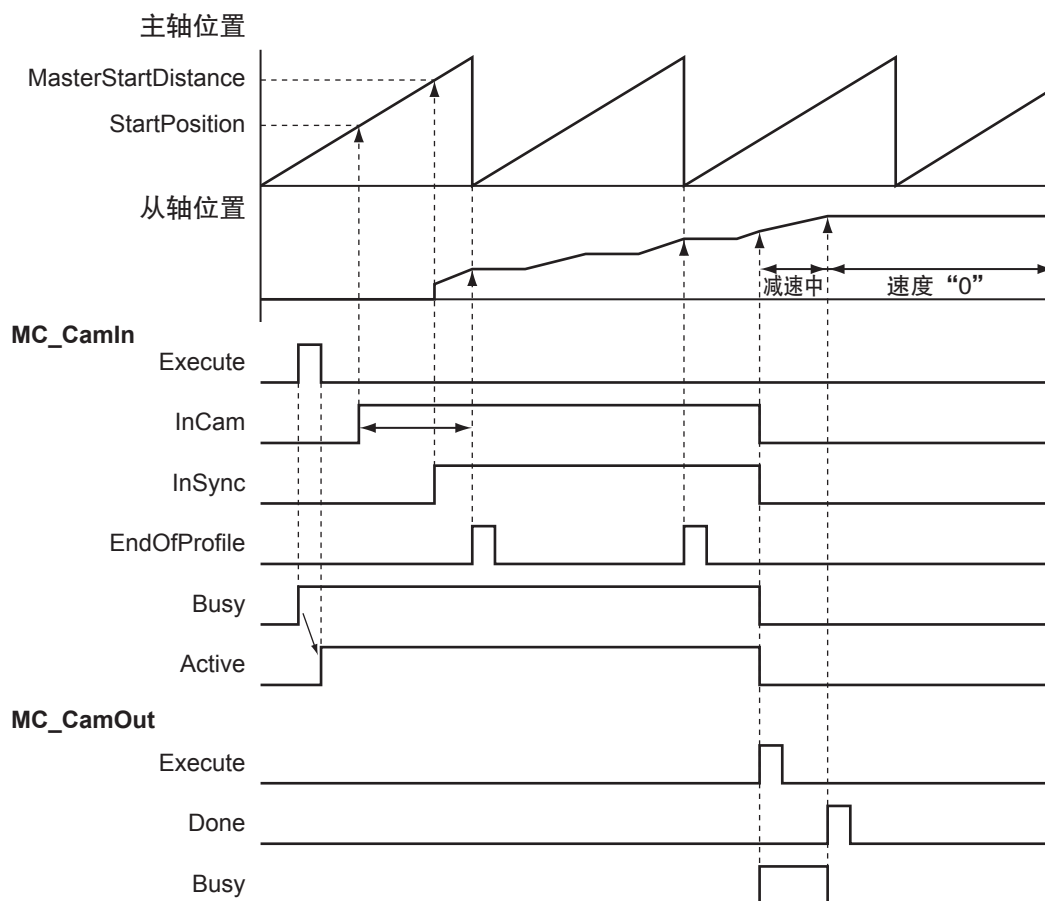
● 有重复时

凸轮动作重复执行。

使用MC_CamOut(解除凸轮动作)指令解除凸轮时，从轴减速至速度“0”。

不影响主轴动作。

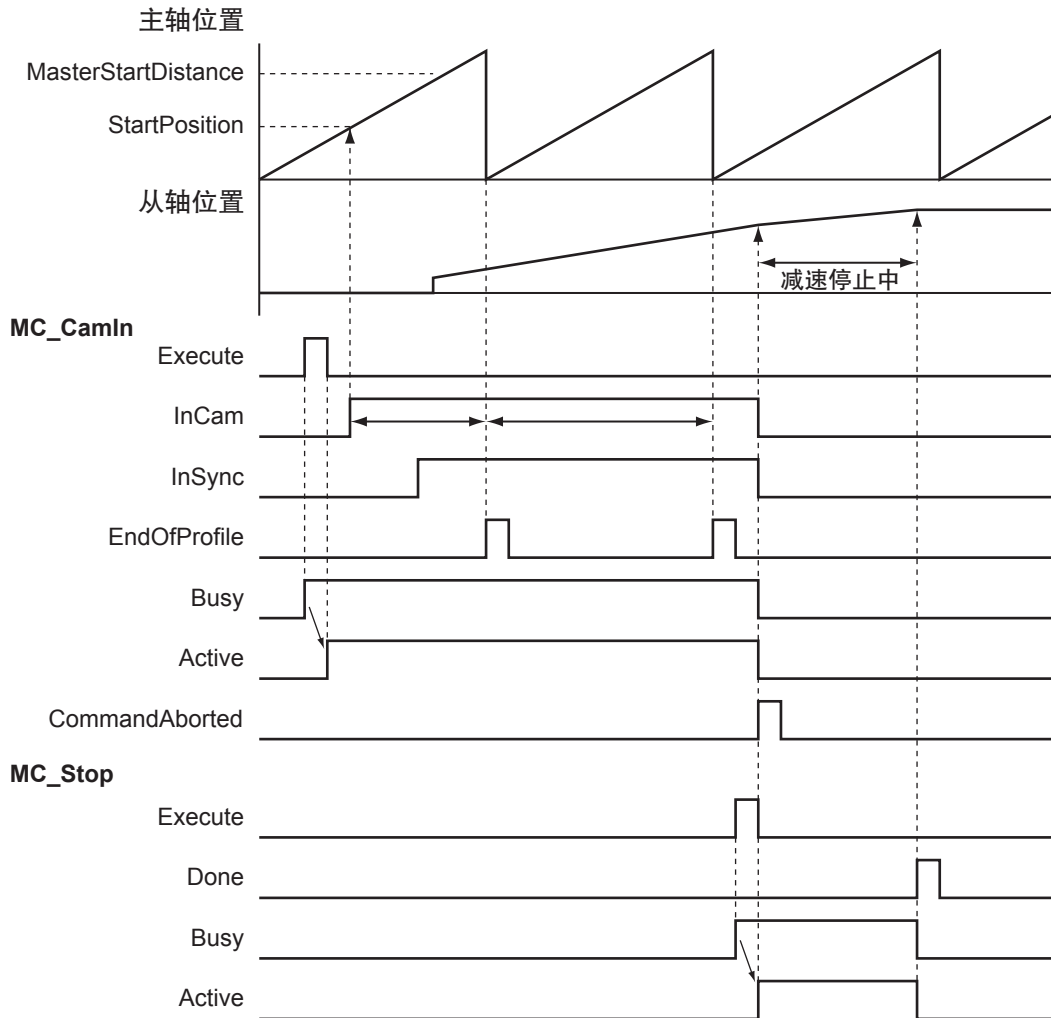
MC_CamIn(凸轮动作开始)指令的Periodic(重复模式)为TRUE[重复]、执行MC_CamOut(解除凸轮动作)指令时，如下图所示。



● MC_Stop(强制停止)指令时

对凸轮动作中的主轴启动MC_Stop(强制停止)指令时，保持主轴和从轴的同步关系。
对凸轮动作中的从轴启动MC_Stop(强制停止)指令时，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]、对从轴执行MC_Stop(强制停止)指令时，如下所示。

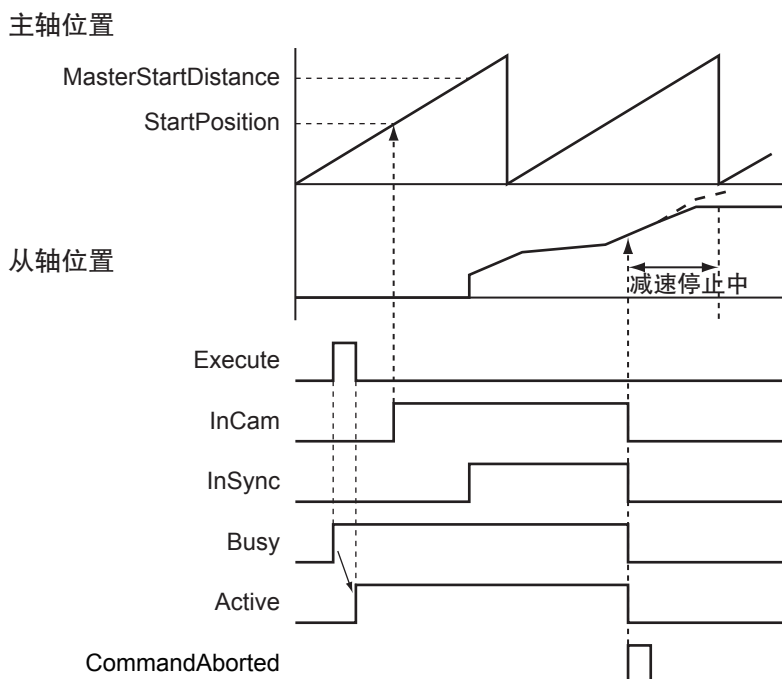


指令的中止

本指令执行中从轴发生轴异常时，从轴以轴的最大减速度停止。对从轴启动 MC_Stop(强制停止)指令时，CommandAborted(执行中断)变为TRUE，从轴以MC_Stop(强制停止)指令指定的减速度停止。

关于轴异常发生原因的特定方法，请参阅 □□ “A-1 错误代码一览(P.A-2)”。

主轴发生轴异常时，继续凸轮动作。



重启运动指令

切换动作中的凸轮表时，可重启本指令。要变更凸轮表切换以外的条件，请对本指令执行指令多重启动。重启指令时，InCam(凸轮动作中)、InSync(同步中)保持重启前的状态。InSync(同步中)中重启时，从比根据当前位置求得的相位大的相位起执行凸轮动作。

根据当前位置求得的相位是指，对凸轮数据之间直线插补时求得的相位。

重启对象仅限凸轮表。



使用注意事项

凸轮动作中重启指令切换凸轮表时，在重启前后，从轴的速度和加速度可能会急剧变化。可能对机械机构产生影响，因此重启时请注意。

重启运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 重启方法

关于重启方法，请参阅 □□ “示例程序1(P.3-187)” 及 □□ “示例程序2(P.3-198)”。

多重启动运动指令

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 本指令执行中的其它指令启动

本指令执行中，使用同一CamTable(凸轮表)的MC_GenerateCamTable(生成凸轮表)指令无法多重启动。对本指令执行运动指令多重启动时，请指定从轴。

本指令动作中多重启动其它指令时，根据缓存模式，存在如下限制。

- 以中断方式多重启动其它指令时，在该时中断凸轮动作，开始下一动作。
- 以等待方式多重启动其它指令时，在EndOfProfile(凸轮周期完成)变为TRUE时，输出下一个动作的指令值。

● 合并多个本指令进行的多重启动

- 只有使用多个本指令进行多重启动时方可指定合并。其它指令执行中，无法以合并方式多重启动本指令。
- Master(主轴)和ReferenceType(位置类型选择)请分别指定与当前执行中的指令相同的值。指定不同的值时，将发生“无法多重启动运动指令”异常。
- 从当前执行中的指令正常结束的周期起，已缓存的本指令的输出变量InSync(同步中)变为TRUE，从轴不停止而直接开始凸轮动作。
- 即使指定了StartPosition(凸轮表起始位置)、MasterStartDistance(主轴跟踪距离)，无论指定值如何，从轴都将在指令启动的同时开始凸轮动作。凸轮表起点为当前执行中的指令的结束位置。无论指定4种合并中的哪一种动作均相同。

● 其它指令执行中的本指令启动

MC_GenerateCamTable(生成凸轮表)执行中，无法多重启动使用同一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执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)

对于 MC_CamIn(凸轮动作开始) 指令以外的同步控制指令，因多重启动导致动作中的同步控制指令变为CommandAborted(中断执行)时，执行相位补偿的指令也同时变为CommandAborted(中断执行)。

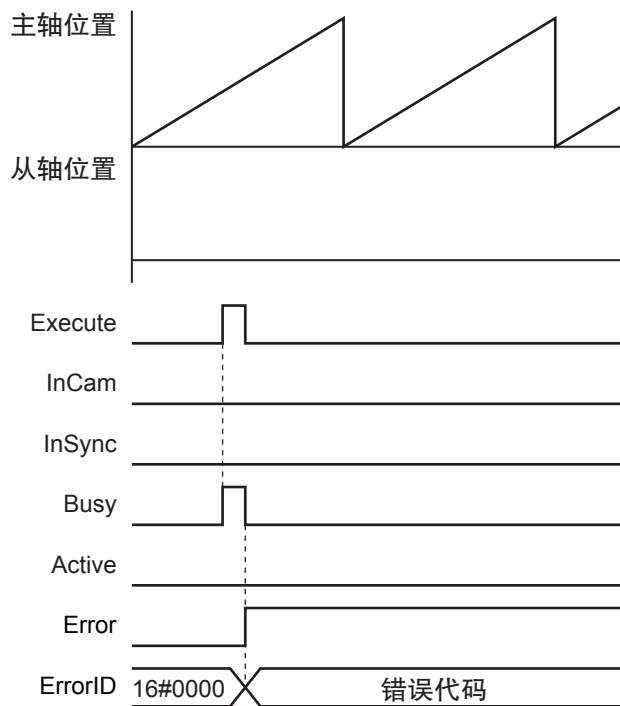
以合并方式多重启动多个MC_CamIn(凸轮动作开始)指令时，执行相位补偿的指令不会变为CommandAborted(中断执行)，而是继续处理。

异常

启动本指令检测到异常时，Error(错误)变为TRUE。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 □ “A-1 错误代码一览(P.A-2)”。

示例程序1

下面，对有重复的凸轮动作的示例程序进行说明。
本例中，以主轴为轴1、从轴为轴2。



参考

作为保留(Reserved)处理的输入变量只能指定为初始值。
本例中，对具备各指令定义初始值的变量进行了分配，但在编程时，无需分配变量和参数。

参数设定

该示例程序中使用的最少的必要设定如下所示。

● 轴参数的设定

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	线性模式

环计数器

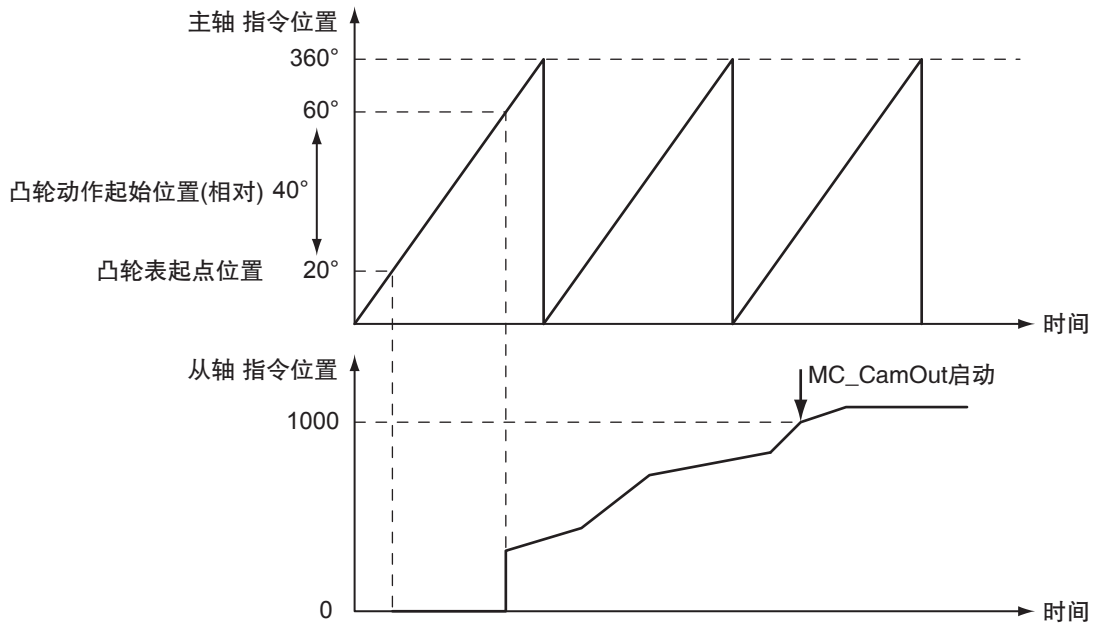
轴	上限值	下限值
轴1	360	0

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

以凸轮表起点位置(零相位位置)为20°, 主轴从该位置到达与相对角度成40°的角度后, 从轴开始凸轮动作。凸轮动作以重复动作方式动作。

将Periodic(重复模式)设为TRUE, 执行重复动作。

2 解除凸轮动作

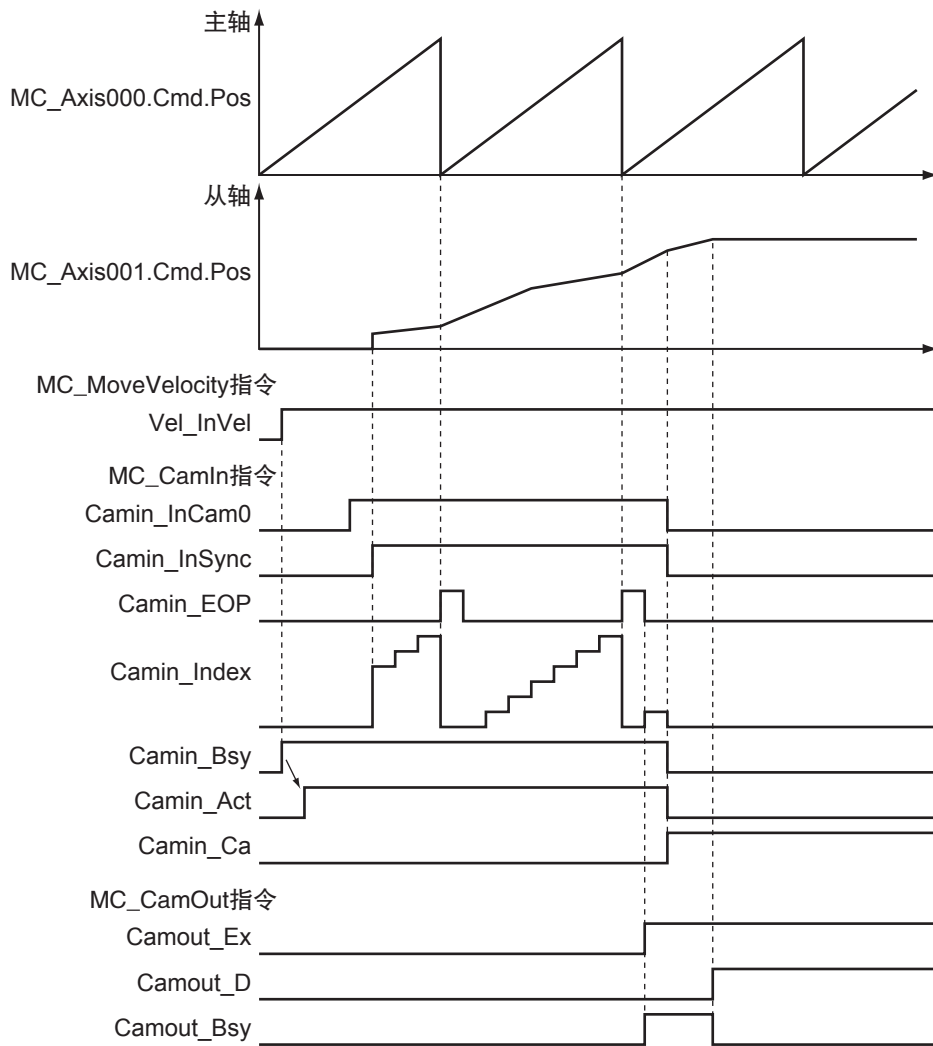
从轴的反馈位置(MC_Axis001.Act.Pos)达到1000.0以上后解除凸轮动作, 以减速度(DecRate2)停止从轴动作。

梯形图

● 主要变量

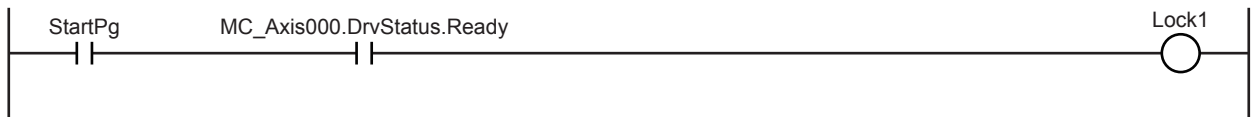
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	是主轴轴1的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴1发生轻度故障等级的异常时变为TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴1为原点确定状态时变为TRUE。
MC_Axis001	_sAXIS_REF	-	是从轴轴2的轴变量。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴2发生轻度故障等级的异常时变为TRUE。
MC_Axis001.Details.Homed	BOOL	FALSE	轴2为原点确定状态时变为TRUE。
Pwr1_S	BOOL	FALSE	分配给MC_Power的实例PWR1的输出Status的变量。进入伺服ON状态时，该变量变为TRUE。
Pwr2_S	BOOL	FALSE	分配给MC_Power的实例PWR2的输出Status的变量。进入伺服ON状态时，该变量变为TRUE。
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	-	是凸轮数据变量。 数组的元素数ARRAY[0..N]由凸轮编辑器设定。本例中设为0~360，但是数组的元素数因凸轮编辑器的设定而异。
DecRate2	LREAL	10000.0	设定执行MC_CamOut时的减速度。
Vel_InVel	BOOL	FALSE	是对MC_MoveVelocity的实例VEL的输出InVelocity进行分配的变量。达到目标速度时，该变量变为TRUE。
Camin_InCam0	BOOL	FALSE	分配给MC_CamIn的实例CAMIN的输出InCam的变量。凸轮动作中时，该变量变为TRUE。
Camout_Ex	BOOL	FALSE	该变量为TRUE时，启动MC_CamOut的实例CAMOUT。
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则进入伺服ON状态。

● 时序图

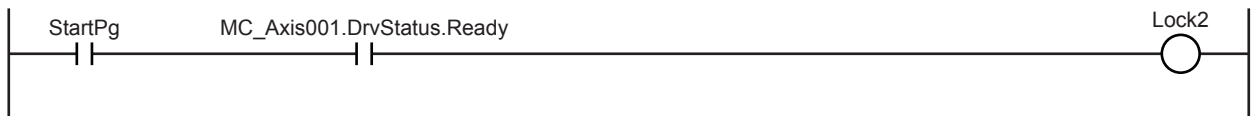


● 示例程序

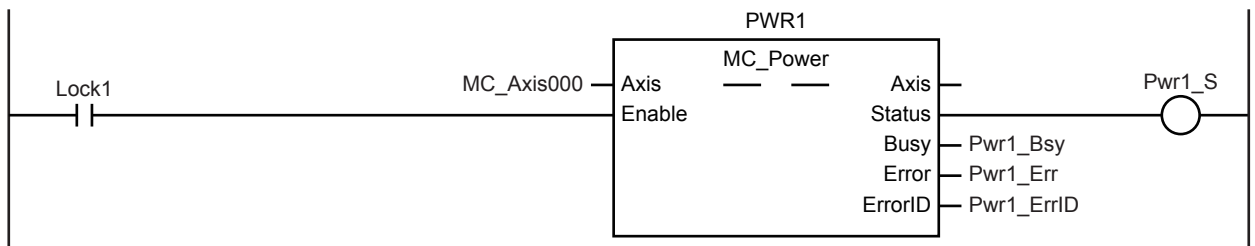
触点StartPg变为TRUE时，确认轴1的伺服驱动器处于伺服准备就绪状态



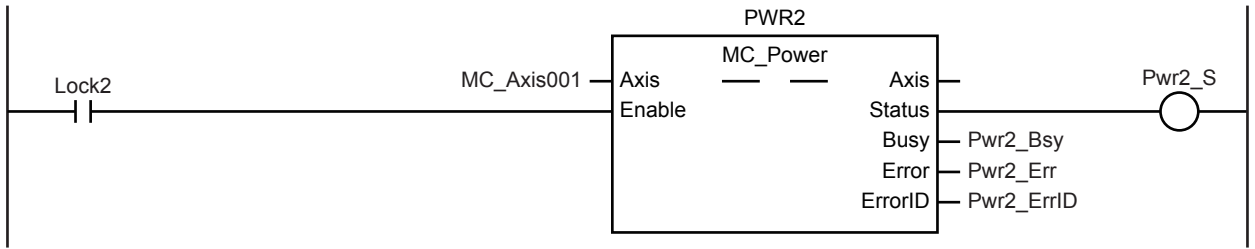
触点StartPg变为TRUE时，确认轴2的伺服驱动器处于伺服准备就绪状态



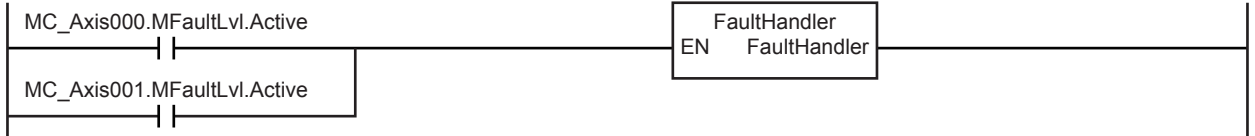
如果轴1处于伺服准备就绪状态，则设为伺服ON状态



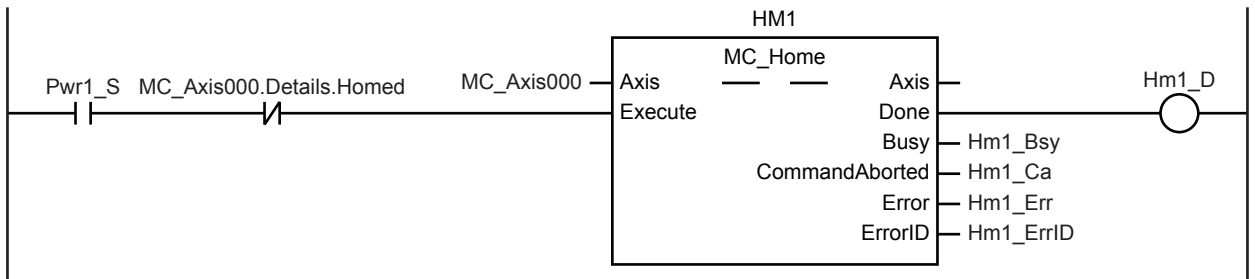
如果轴2处于伺服准备就绪状态，则设为伺服ON状态



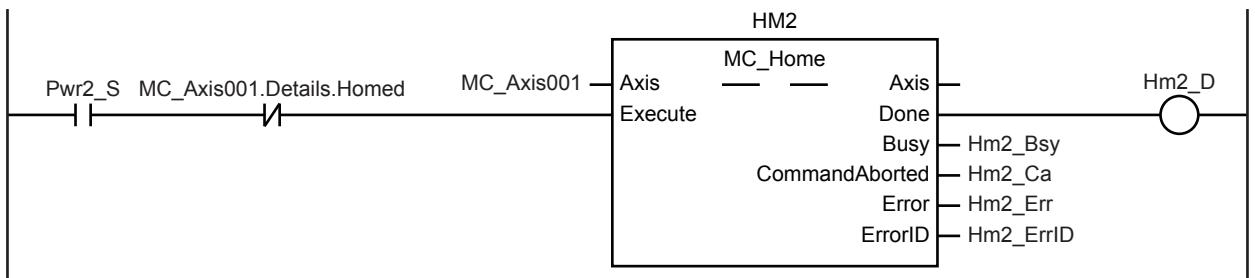
轴构成发生轻度故障程度的异常时，执行设备的异常处理(FaultHandler)。
发生异常时的处理(FaultHandler)由客户根据装置进行编程



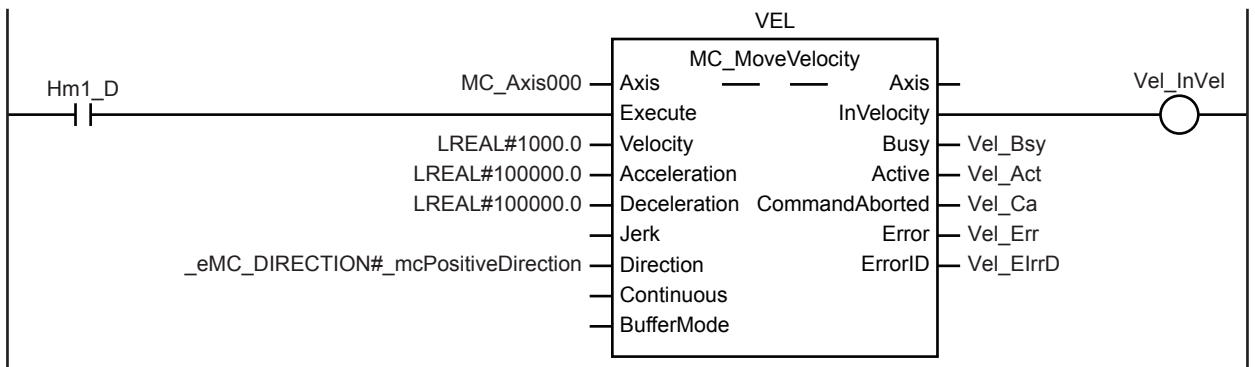
轴1变为伺服ON 状态，若原点未确定，则执行原点复位，确定原点



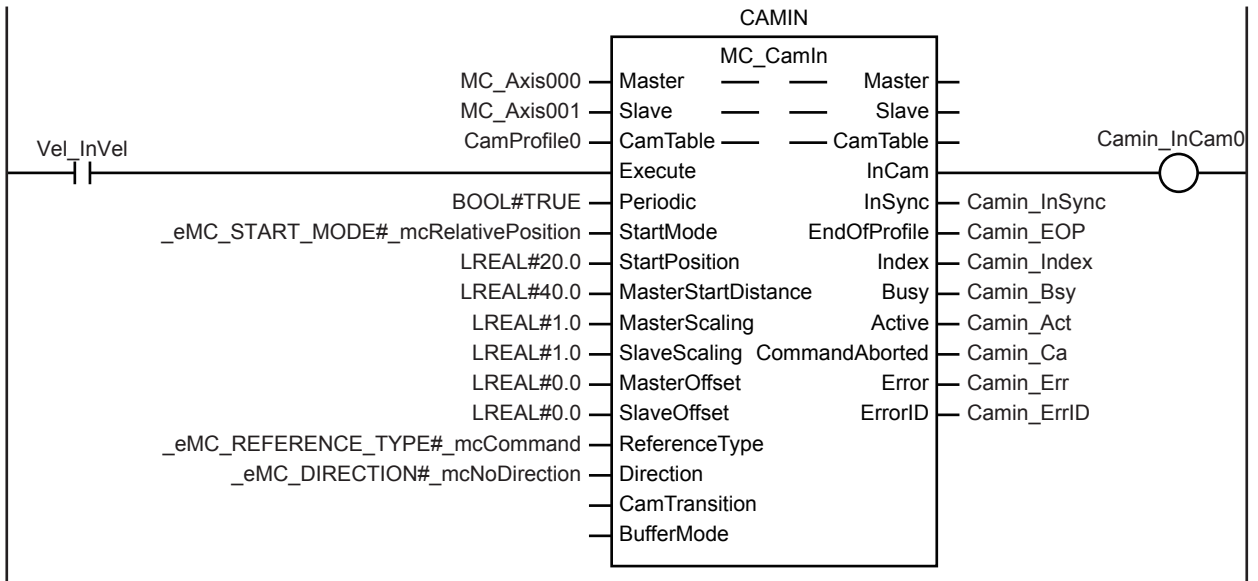
轴2变为伺服ON 状态，若原点未确定，则执行原点复位，确定原点



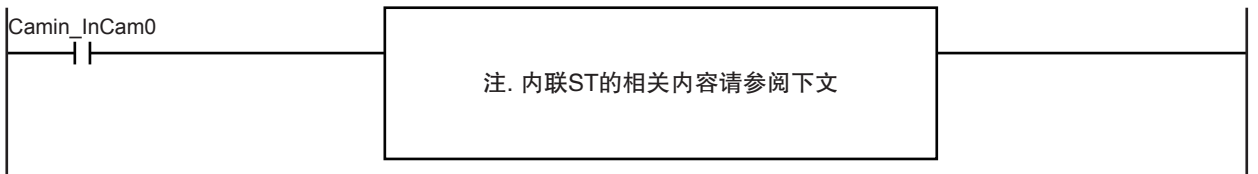
轴1进入原点复位完成状态后，执行MC_MoveVelocity(速度控制)指令



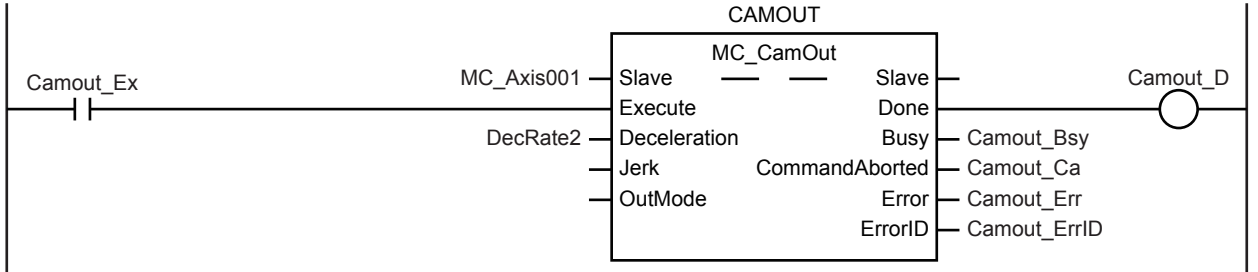
MC_MoveVelocity(速度控制)指令的Vel_InVel变为TRUE后，启动MC_CamIn(凸轮动作开始)指令



Camin_InCam0为TRUE、MC_Axis001.Act.Pos超过1000后，将CamOut_Ex设为TRUE



将Camout_Ex设为TRUE，启动MC_CamOut(解除凸轮动作)指令。
作为减速度，指定DecRate2



内联ST的内容

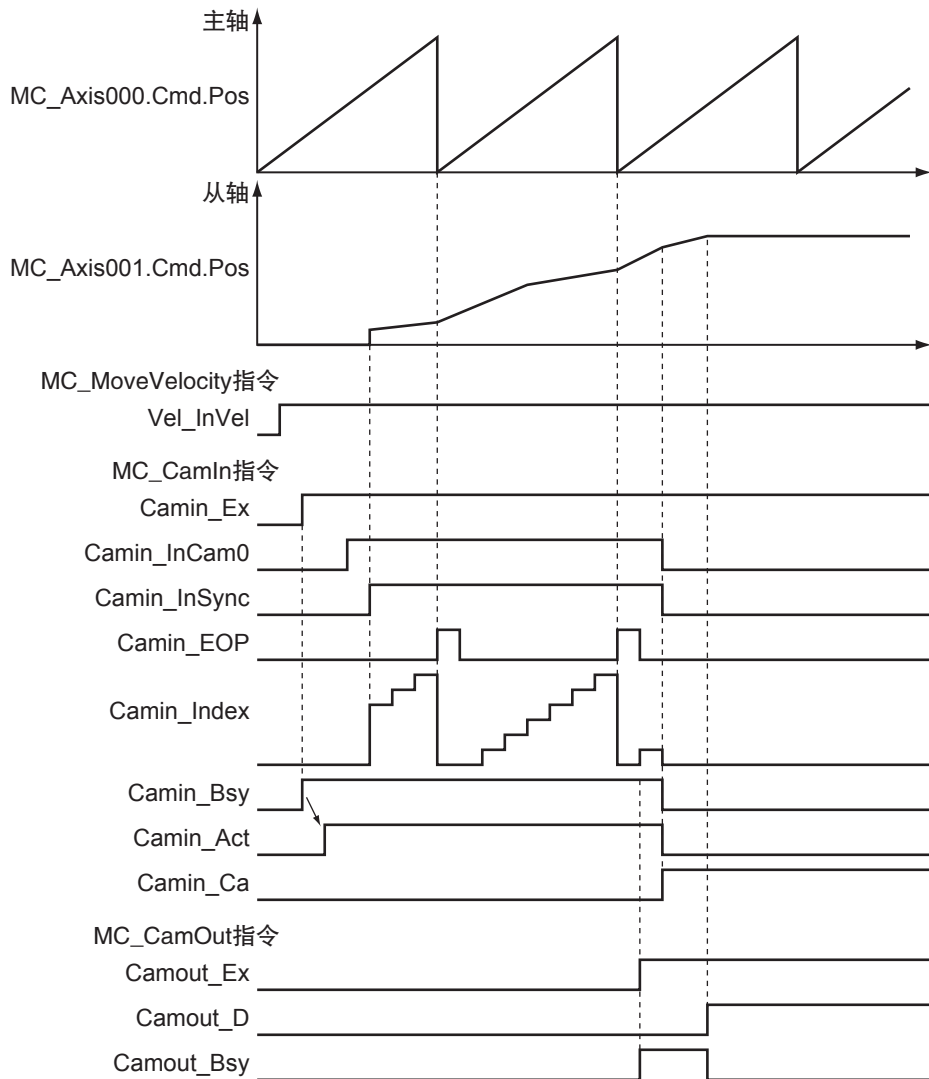
```
IF MC_Axis001.Act.Pos>LREAL#1000.0 THEN
    Camout_Ex := TRUE;
END_IF;
```

结构文本(ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	是主轴轴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为原点确定状态时变为TRUE。
MC_Axis001	_sAXIS_REF	-	是从轴轴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为原点确定状态时变为TRUE。
Pwr1_S	BOOL	FALSE	分配给MC_Power的实例PWR1的输出Status的变量。进入伺服ON状态时, 该变量变为TRUE。
Pwr2_S	BOOL	FALSE	分配给MC_Power的实例PWR2的输出Status的变量。进入伺服ON状态时, 该变量变为TRUE。
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	-	是凸轮数据变量。 数组的元素数ARRAY[0..N]由凸轮编辑器设定。本例中设为0~360, 但是数组的元素数因凸轮编辑器的设定而异。
DecRate2	LREAL	10000.0	设定执行MC_CamOut时的减速度。
Vel_InVel	BOOL	FALSE	是对MC_MoveVelocity的实例VEL的输出InVelocity进行分配的变量。达到目标速度时, 该变量变为TRUE。
Camin_InCam0	BOOL	FALSE	分配给MC_CamIn的实例CAMIN的输出InCam的变量。凸轮动作中时, 该变量变为TRUE。
Camout_Ex	BOOL	FALSE	该变量为TRUE时, 启动MC_CamOut的实例CAMOUT。
StartPg	BOOL	FALSE	如果该变量为TRUE, EtherCAT的过程数据通信已建立, 则进入伺服ON状态。
Camin_Ex	BOOL	FALSE	启动Vel_InVel时设为TRUE, 将MC_CamIn的实例CAMIN的Execute设为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;    // 起点的主轴绝对位置
Camin_MStDis  := LREAL#40.0;    // 凸轮动作开始的主轴位置
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; //位置类型选择
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;

//确认StartPg为TRUE, 且伺服驱动器处于伺服准备就绪状态,
//则将轴1设为伺服ON状态。
//如果未处于伺服准备就绪状态, 则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
  Pwr1_En:=TRUE;
ELSE
  Pwr1_En:=FALSE;
END_IF;

//确认StartPg为TRUE, 且伺服驱动器处于伺服准备就绪状态,
//则将轴2设为伺服ON状态。
//如果未处于伺服准备就绪状态, 则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
  Pwr2_En:=TRUE;
ELSE
  Pwr2_En:=FALSE;
END_IF;

//轴1 ~ 轴2发生轻度故障后, 执行异常时处理。
//发生异常时的处理(FaultHandler)由客户根据装置进行编程。
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
  FaultHandler();
END_IF;

//轴1处于伺服ON状态, 且原点未确定时, 进行轴1的原点复位
IF (Pwr1_S=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
  Hm1_Ex:=TRUE;
END_IF;

//轴2处于伺服ON状态, 且原点未确定时, 进行轴2的原点复位
IF (Pwr2_S=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
  Hm2_Ex:=TRUE;
END_IF;

// 轴1 进入原点复位完成状态后, 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，对充填液体的凸轮动作的示例程序进行说明。



参考

作为保留(Reserved)处理的输入变量只能指定为初始值。
本例中，不指定参数。

参数设定

该示例程序中使用的最少的必要设定如下所示。

● 轴参数的设定

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

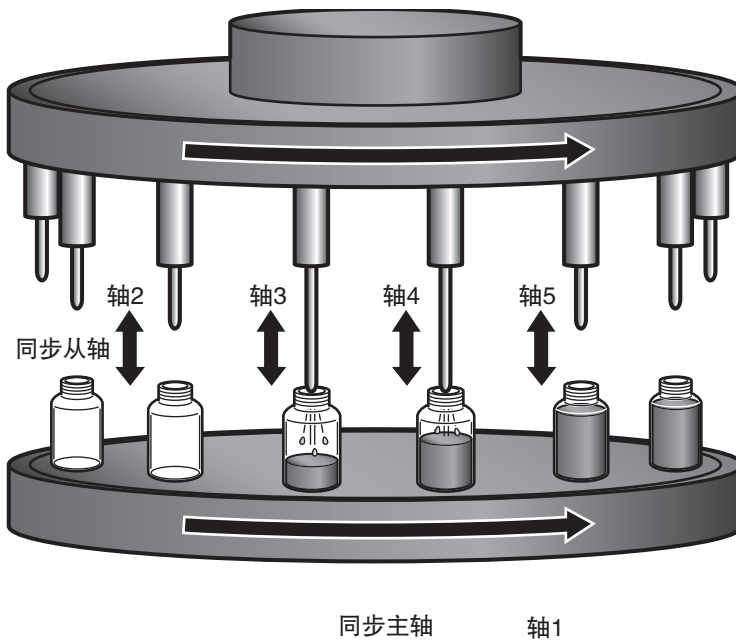
环计数器

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

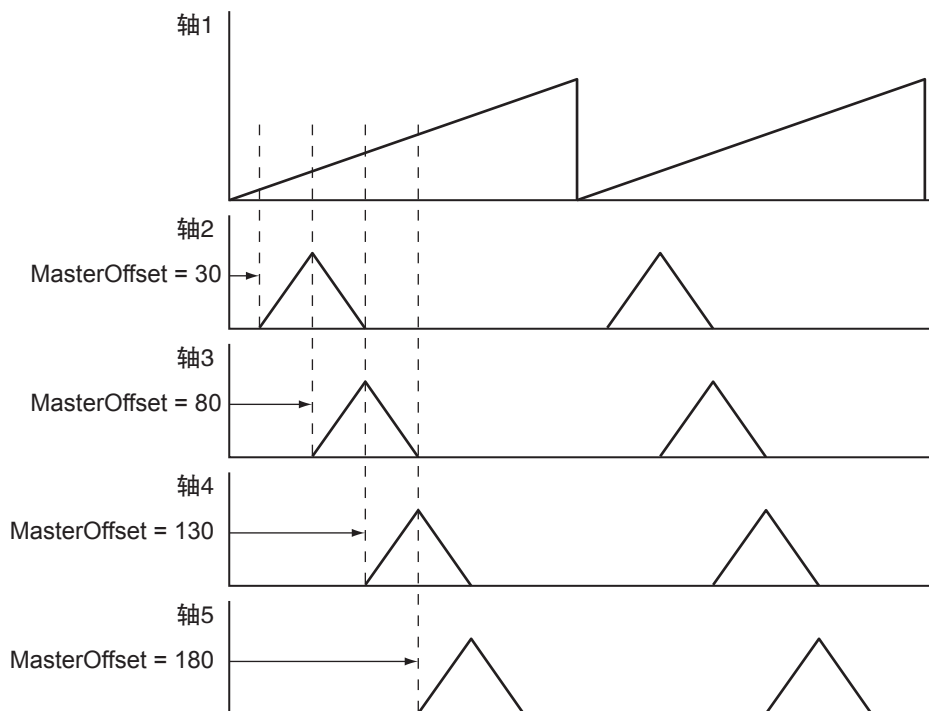
坐标单位选择

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



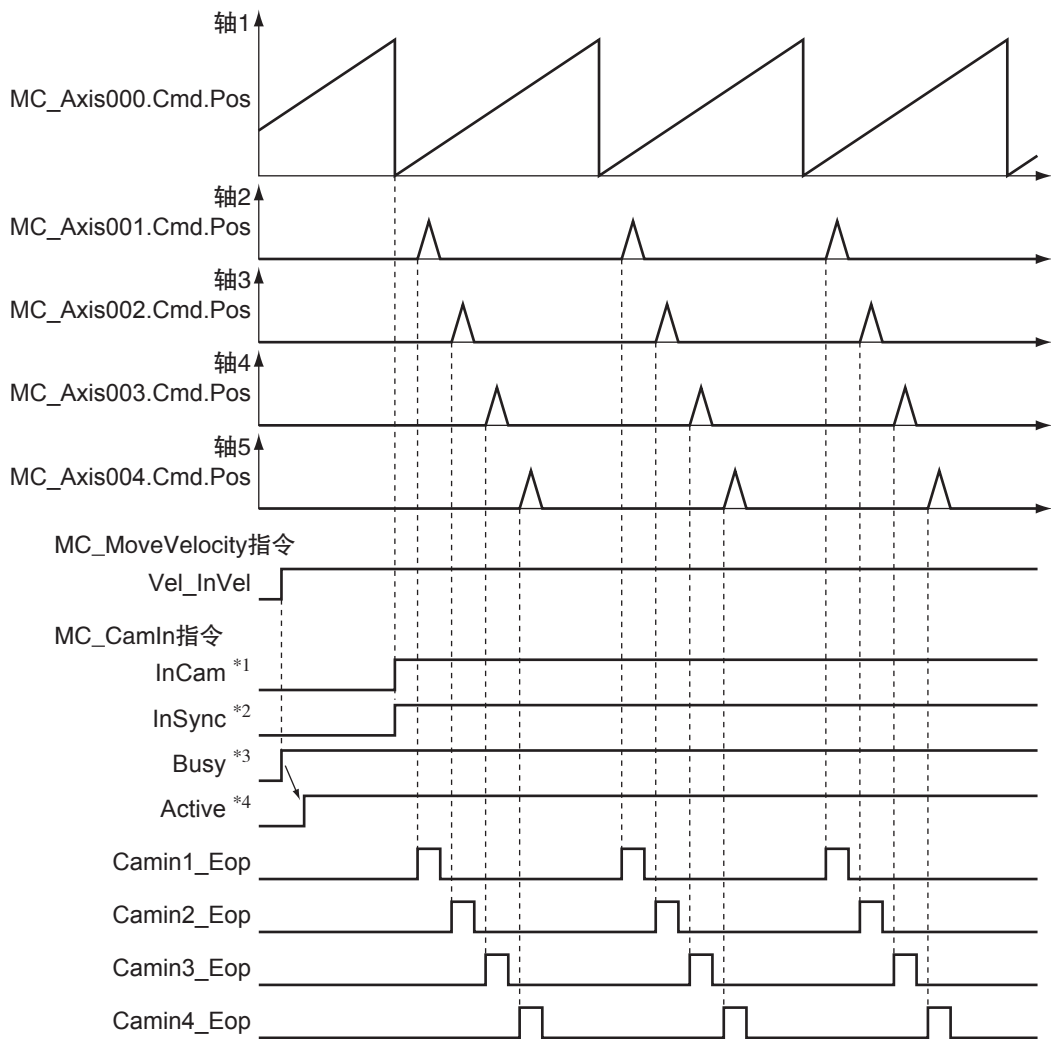
- 1** 凸轮动作开始
与主轴(轴1)同步, 从轴(轴2 ~ 轴5)进行凸轮动作。各轴相位分别移动50degree, 开始凸轮动作。
- 2** 重复
各轴重复执行指定的凸轮动作。

梯形图

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	是主轴轴1的轴变量。
MC_Axis001	_sAXIS_REF	-	是从轴轴2的轴变量。
MC_Axis002	_sAXIS_REF	-	是从轴轴3的轴变量。
MC_Axis003	_sAXIS_REF	-	是从轴轴4的轴变量。
MC_Axis004	_sAXIS_REF	-	是从轴轴5的轴变量。
StartPg	BOOL	FALSE	如果该变量为TRUE, EtherCAT的过程数据通信已建立, 则进入伺服ON状态。
Vel_InVel	BOOL	FALSE	是对MC_MoveVelocity的实例VEL的输出InVelocity进行分配的变量。达到目标速度时, 该变量变为TRUE。
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	-	是凸轮数据变量。 在MC_CamIn的实例CAMIN1 ~ CAMIN4的输入CamTable中指定该变量。 数组的元素数ARRAY[0..N]由凸轮编辑器设定。

● 时序图



*1 Camin1_InCam, Camin2_InCam, Camin3_InCam, Camin4_InCam为同一时间。

*2 Camin1_InSync, Camin2_InSync, Camin3_InSync, Camin4_InSync为同一时间。

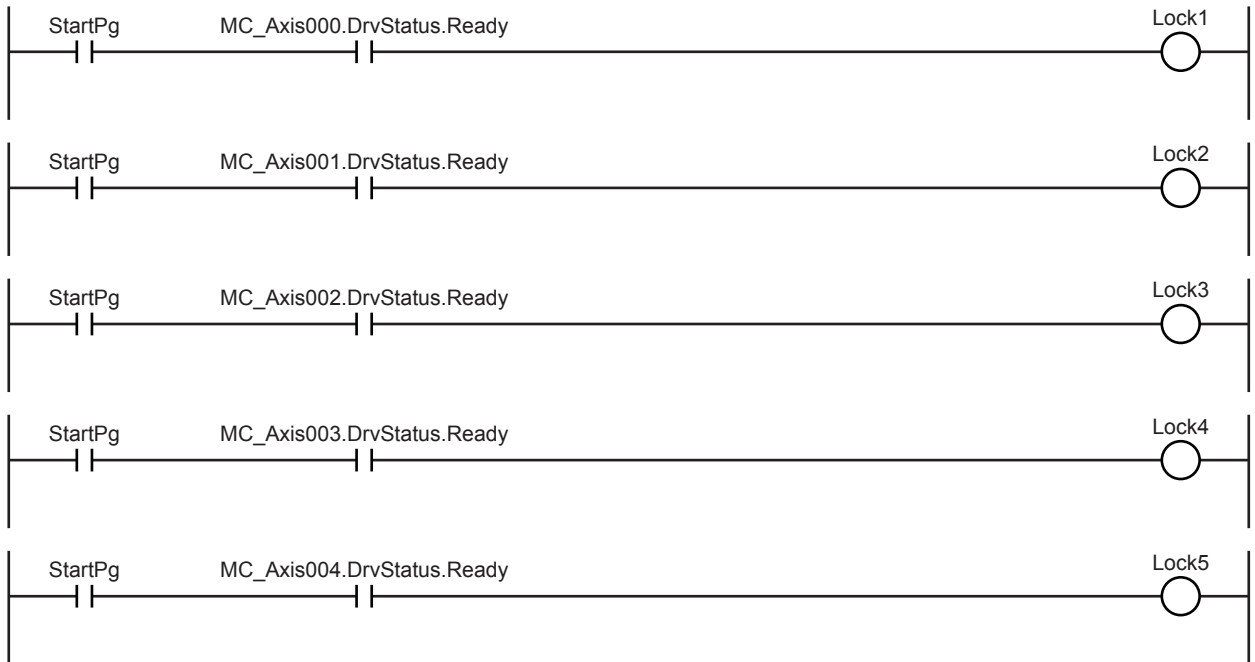
本例中，MasterStartDistance为“0”，因此，主轴为“0°”起，InSync变为TRUE。

*3 Camin1_Bsy, Camin2_Bsy, Camin3_Bsy, Camin4_Bsy为同一时间。

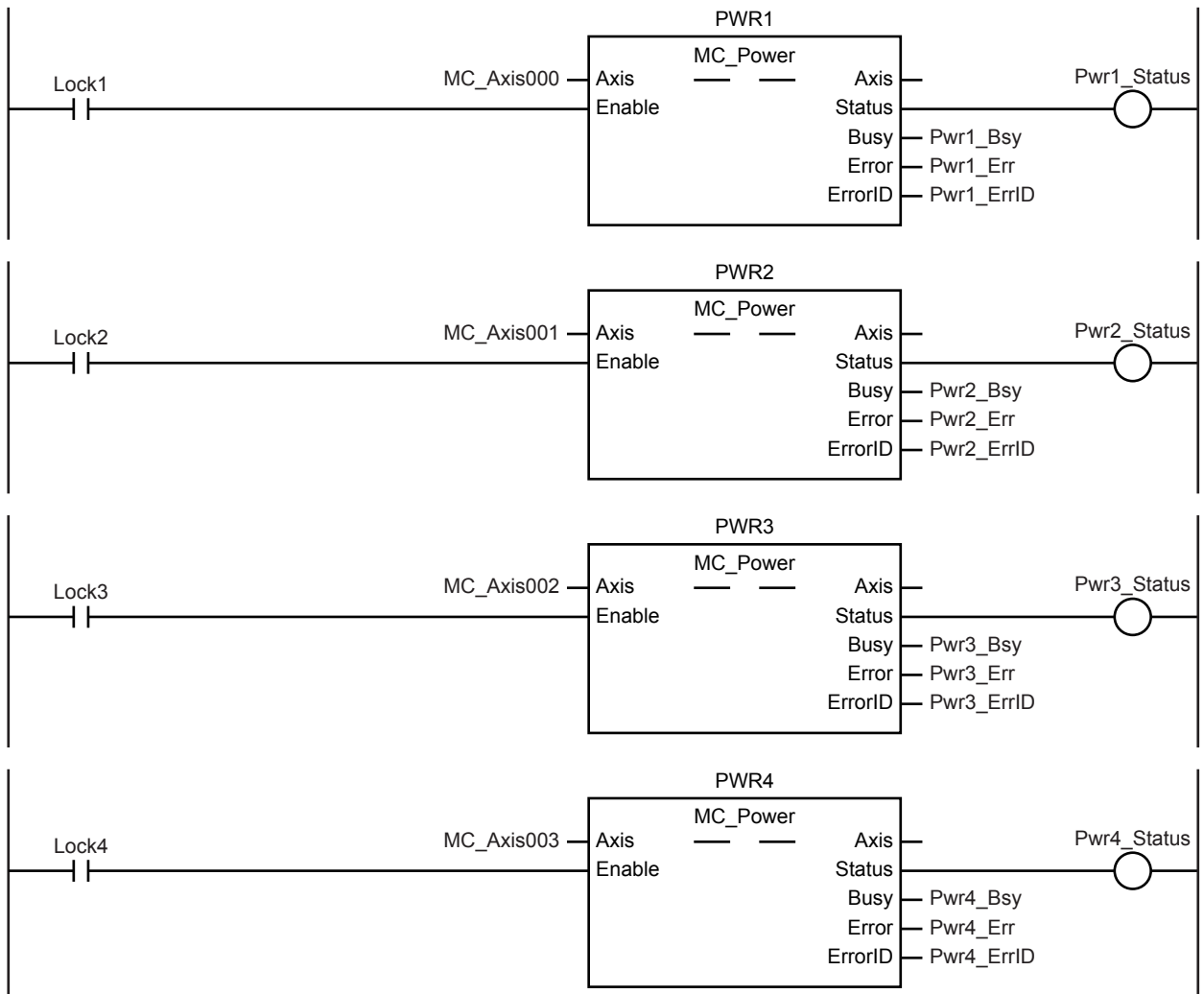
*4 Camin1_Act, Camin2_Act, Camin3_Act, Camin4_Act为同一时间。

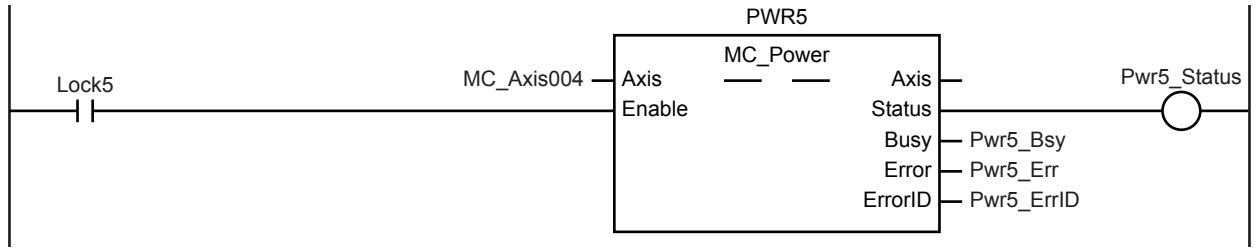
● 示例程序

触点StartPg变为TRUE时，确认各轴的伺服驱动器处于伺服准备就绪状态

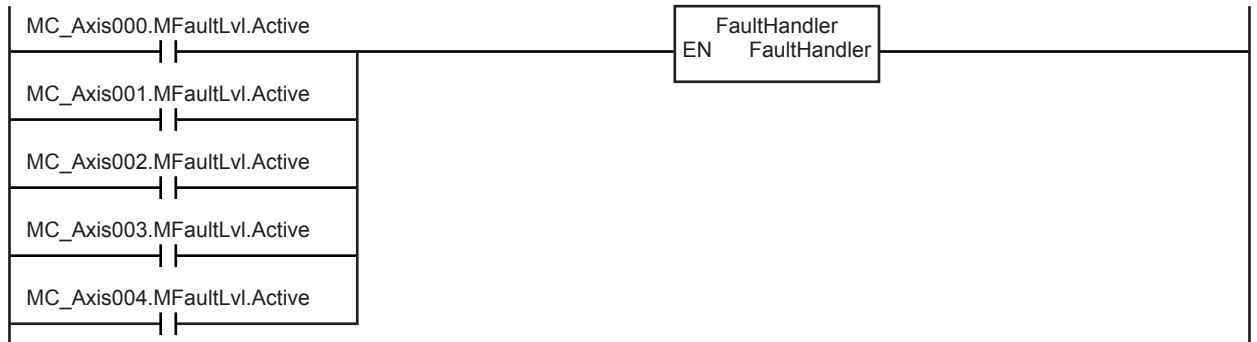


如果处于伺服准备就绪状态，则将各轴设为伺服ON状态

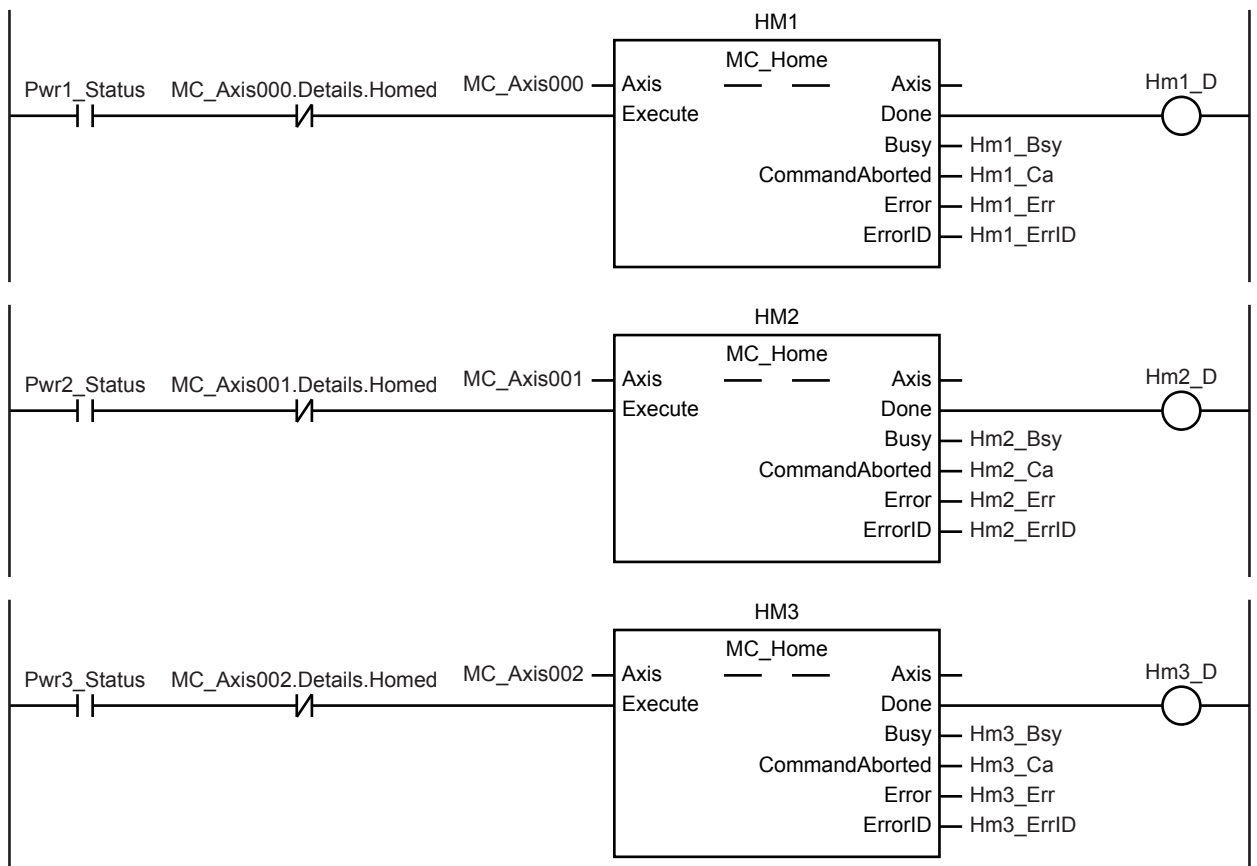


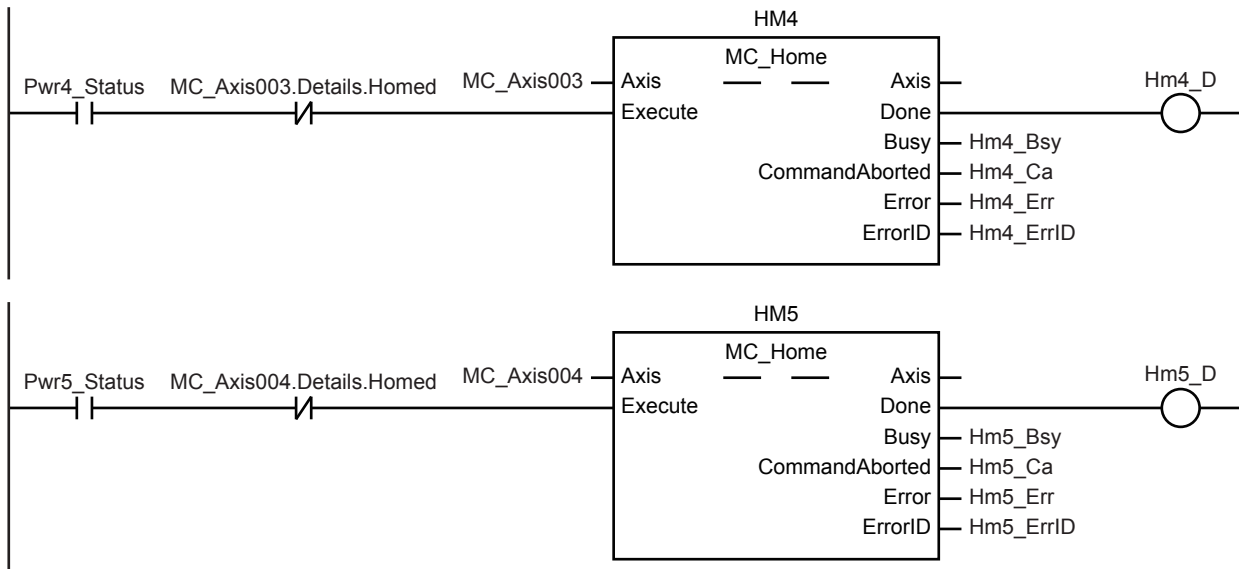


轴构成发生轻度故障程度的异常时，执行设备的异常处理(FaultHandler)。
发生异常时的处理(FaultHandler)由客户根据装置进行编程

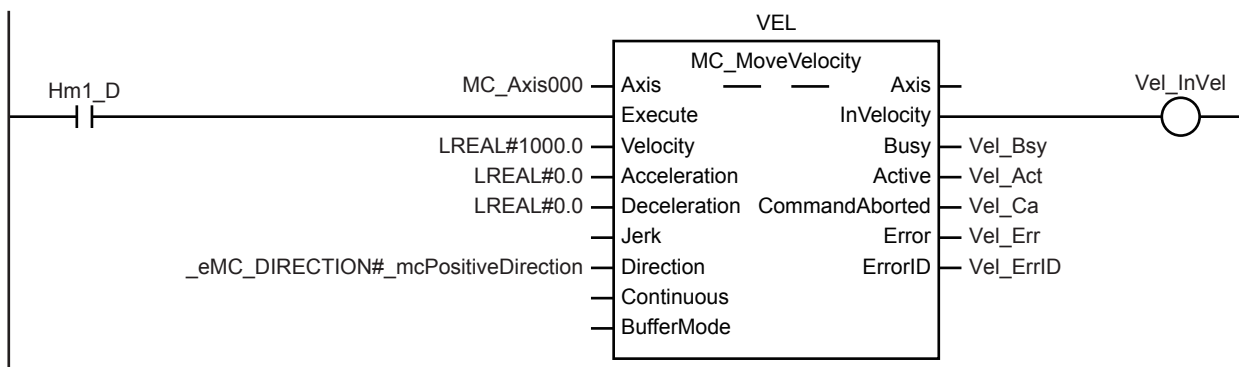


轴1 ~ 轴5变为伺服ON状态，若原点未确定，则对各轴执行原点复位

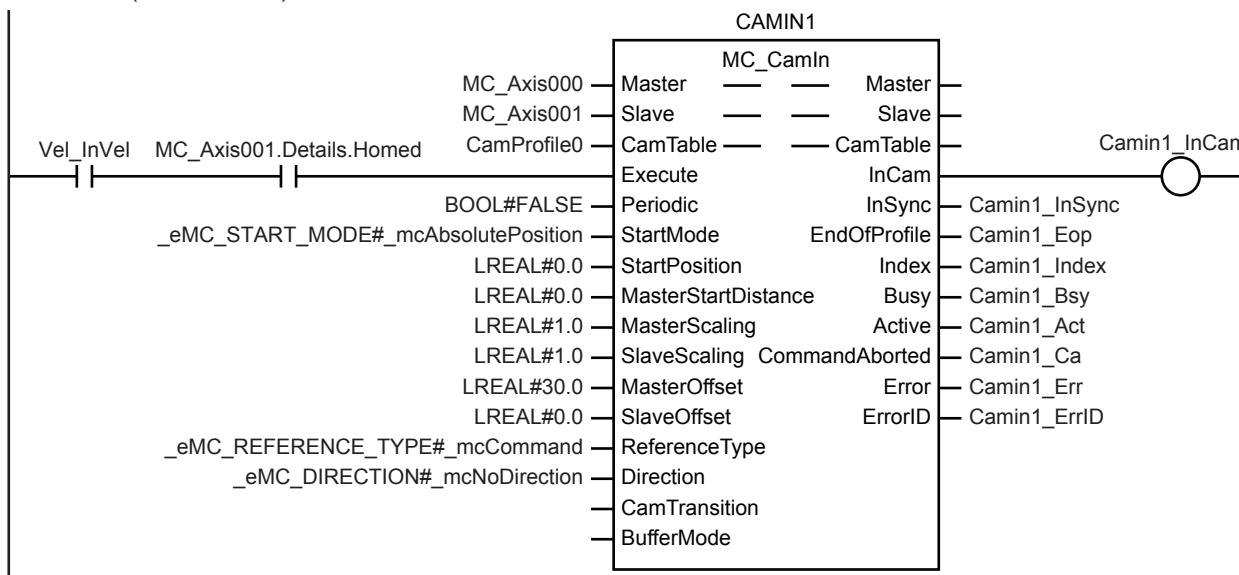




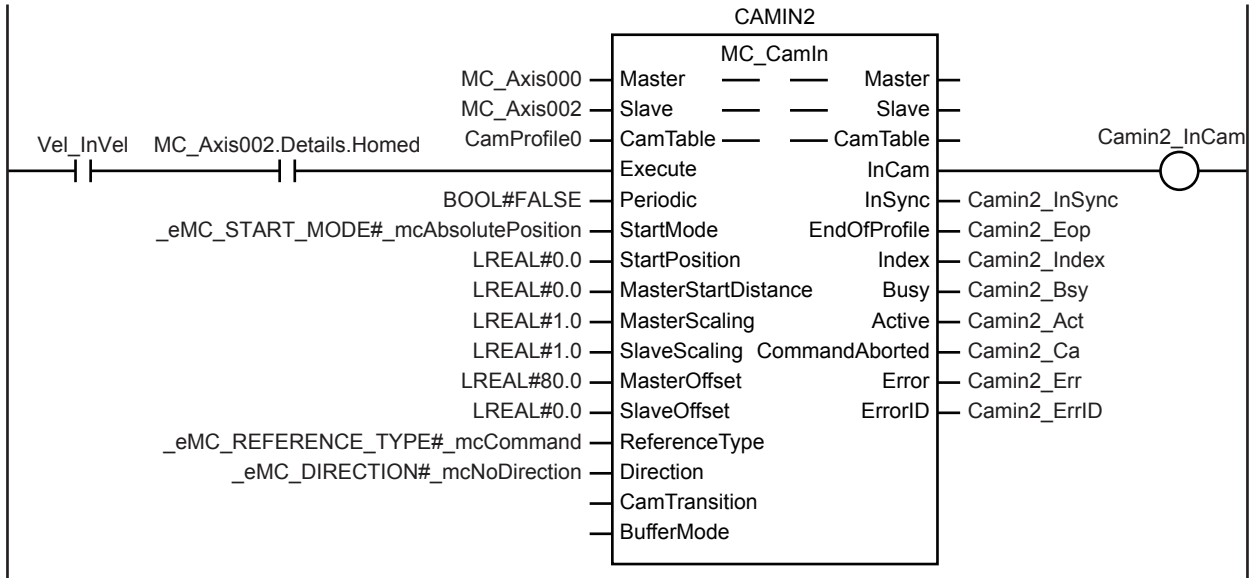
轴1的原点复位完成后，启动MC_MoveVelocity(速度控制)指令



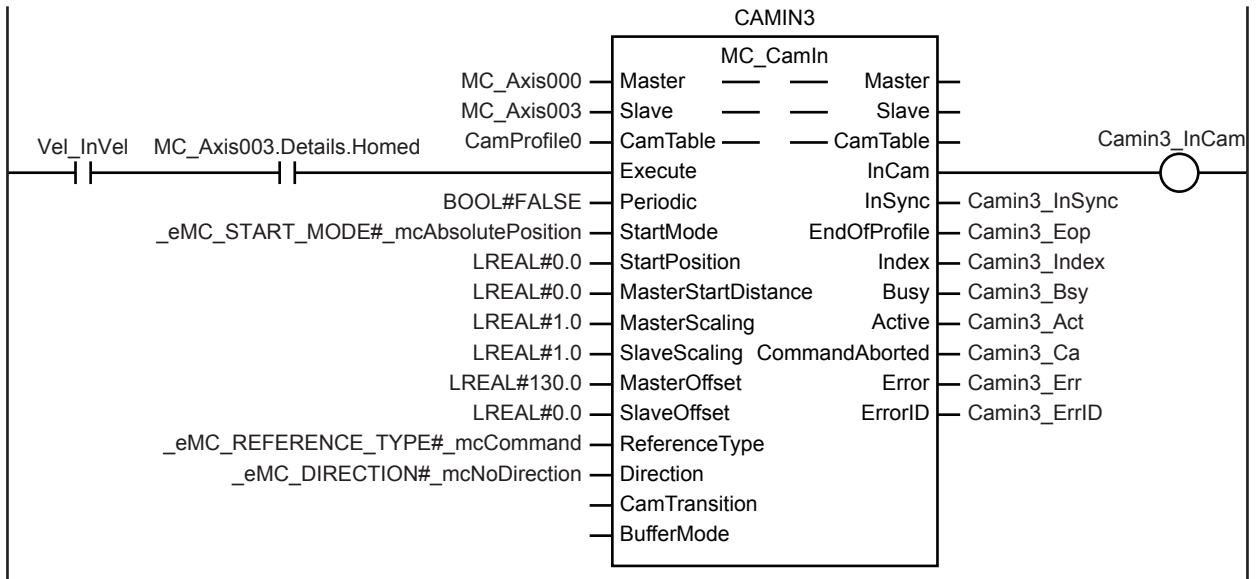
MC_MoveVelocity(速度控制)指令的Vel_InVel为TRUE、轴2的原点复位完成后，执行从轴(轴2)设定的MC_CamIn(凸轮动作开始)指令



MC_MoveVelocity(速度控制)指令的Vel_InVel为TRUE、轴3的原点复位完成后，执行从轴(轴3)设定的MC_CamIn(凸轮动作开始)指令

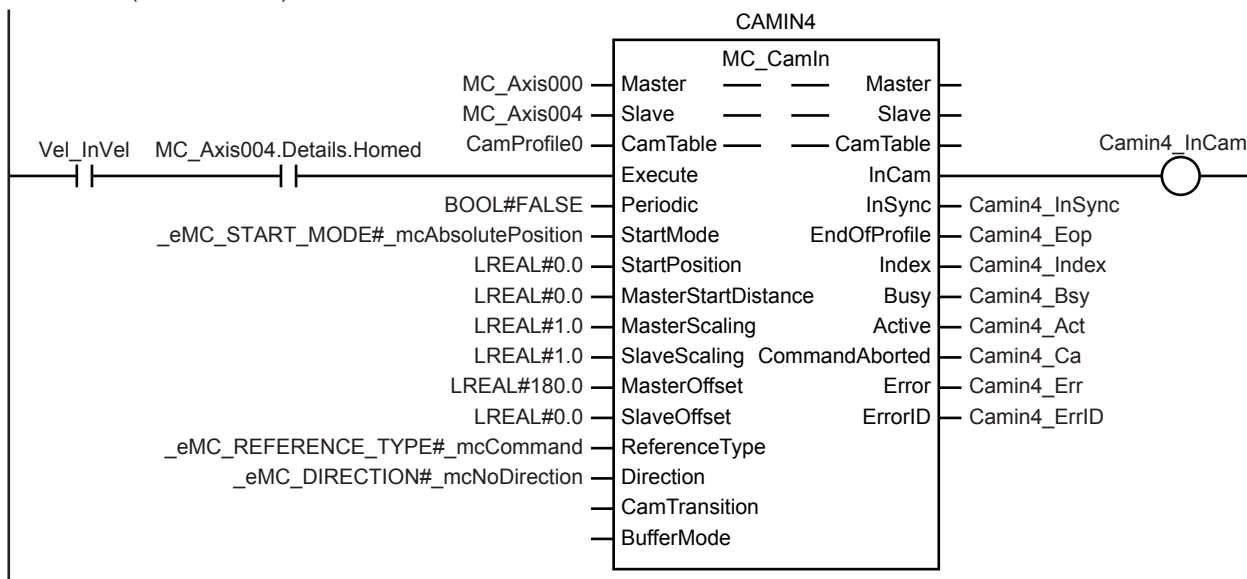


MC_MoveVelocity(速度控制)指令的Vel_InVel为TRUE、轴4的原点复位完成后，执行从轴(轴4)设定的MC_CamIn(凸轮动作开始)指令



MC_CamIn
3
示例程序2

MC_MoveVelocity(速度控制)指令的Vel_InVel为TRUE、轴5的原点复位完成后, 执行从轴(轴5)设定的MC_CamIn(凸轮动作开始)指令

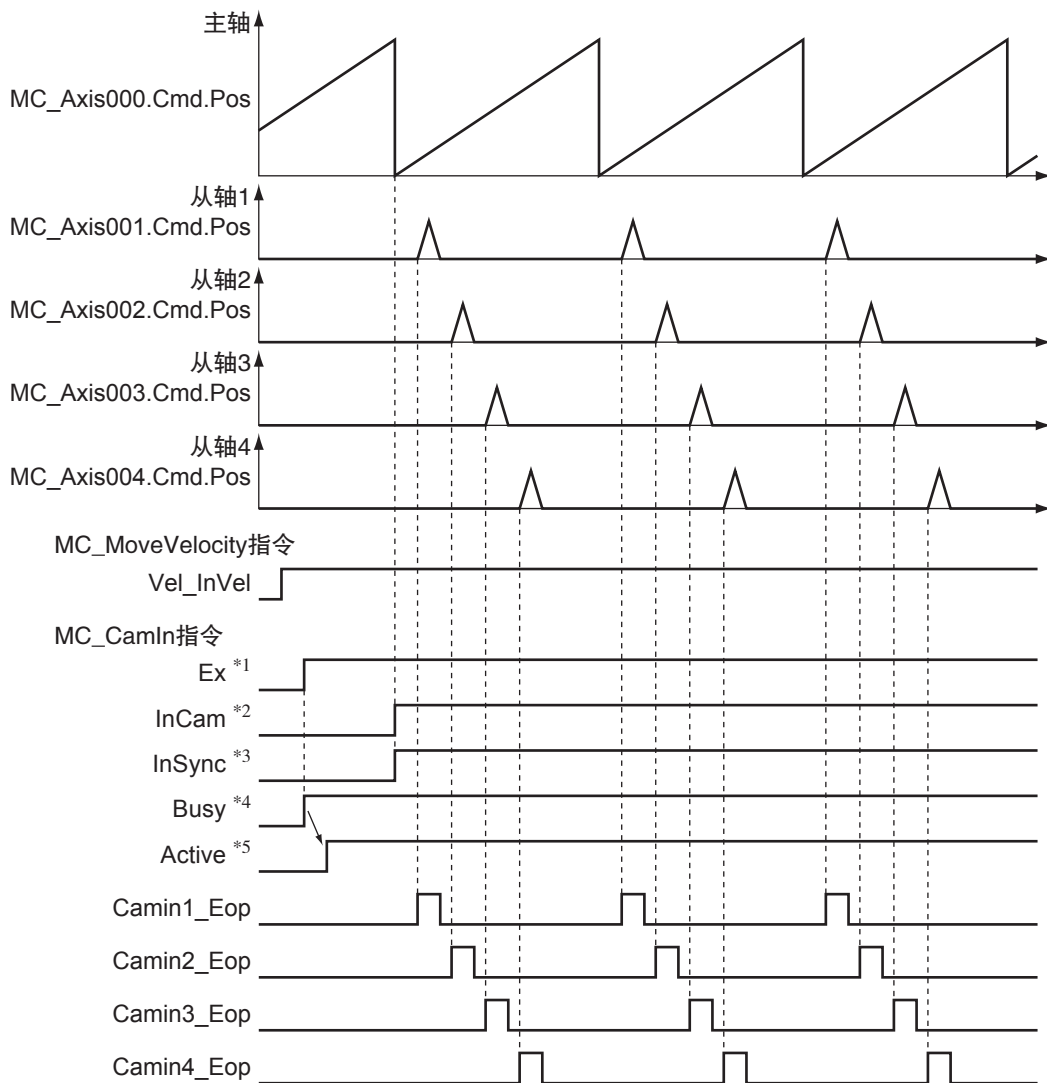


结构文本(ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	是主轴轴1的轴变量。
MC_Axis001	_sAXIS_REF	-	是从轴轴2的轴变量。
MC_Axis002	_sAXIS_REF	-	是从轴轴3的轴变量。
MC_Axis003	_sAXIS_REF	-	是从轴轴4的轴变量。
MC_Axis004	_sAXIS_REF	-	是从轴轴5的轴变量。
StartPg	BOOL	FALSE	如果该变量为TRUE, EtherCAT的过程数据通信已建立, 则进入伺服ON状态。
Vel_InVel	BOOL	FALSE	是对MC_MoveVelocity的实例VEL的输出InVelocity进行分配的变量。达到目标速度时, 该变量变为TRUE。
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	-	是凸轮数据变量。在MC_CamIn的实例CAMIN1 ~ CAMIN4的输入CamTable中指定该变量。 数组的元素数ARRAY[0..N]由凸轮编辑器设定。
Camin1_Ex	BOOL	FALSE	该变量由FALSE→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为同一时间。
- *2 Camin1_InSync, Camin2_InSync, Camin3_InSync, Camin4_InSync为同一时间。
本例中, MasterStartDistance为“0”, 因此, 主轴为“0°”起, InSync变为TRUE。
- *3 Camin1_Bsy, Camin2_Bsy, Camin3_Bsy, Camin4_Bsy为同一时间。
- *4 Camin1_Act, Camin2_Act, Camin3_Act, Camin4_Act为同一时间。

● 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;

//确认StartPg为TRUE，且伺服驱动器处于伺服准备就绪状态，
//则将轴1设为伺服ON状态。
//如果未处于伺服准备就绪状态，则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE;
  ELSE
    Pwr1_En:=FALSE;
  END_IF;

//确认StartPg为TRUE，且伺服驱动器处于伺服准备就绪状态，
//则将轴2设为伺服ON状态。
//如果未处于伺服准备就绪状态，则设为伺服OFF。
IF (StartPg=TRUE)

```

```

AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
  Pwr2_En:=TRUE;
ELSE
  Pwr2_En:=FALSE;
END_IF;

//确认StartPg为TRUE, 且伺服驱动器处于伺服准备就绪状态,
//则将轴3设为伺服ON状态。
//如果未处于伺服准备就绪状态, 则设为伺服OFF。
IF (StartPg=TRUE)
AND (MC_Axis002.DrvStatus.Ready=TRUE) THEN
  Pwr3_En:=TRUE;
ELSE
  Pwr3_En:=FALSE;
END_IF;

//确认StartPg为TRUE, 且伺服驱动器处于伺服准备就绪状态,
//则将轴4设为伺服ON状态。
//如果未处于伺服准备就绪状态, 则设为伺服OFF。
IF (StartPg=TRUE)
AND (MC_Axis003.DrvStatus.Ready=TRUE) THEN
  Pwr4_En:=TRUE;
ELSE
  Pwr4_En:=FALSE;
END_IF;

//确认StartPg为TRUE, 且伺服驱动器处于伺服准备就绪状态,
//则将轴5设为伺服ON状态。
//如果未处于伺服准备就绪状态, 则设为伺服OFF。
IF (StartPg=TRUE)
AND (MC_Axis004.DrvStatus.Ready=TRUE) THEN
  Pwr5_En:=TRUE;
ELSE
  Pwr5_En:=FALSE;
END_IF;

//轴1 ~ 轴5发生轻度故障后, 执行异常时处理FaultHandler。
//发生异常时的处理(FaultHandler)由客户根据装置进行编程。
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处于伺服ON状态, 且原点未确定时, 进行轴1的原点复位
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
  Hm1_Ex:=TRUE;
END_IF;

//轴2处于伺服ON状态, 且原点未确定时, 进行轴2的原点复位
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
  Hm2_Ex:=TRUE;
END_IF;

//轴3处于伺服ON状态, 且原点未确定时, 进行轴3的原点复位
IF (Pwr3_Status=TRUE) AND (MC_Axis002.Details.Homed=FALSE) THEN
  Hm3_Ex:=TRUE;

```

```

END_IF;

//轴4处于伺服ON状态，且原点未确定时，进行轴4的原点复位
IF (Pwr4_Status=TRUE) AND (MC_Axis003.Details.Homed=FALSE) THEN
  Hm4_Ex:=TRUE;
END_IF;

//轴5处于伺服ON状态，且原点未确定时，进行轴5的原点复位
IF (Pwr5_Status=TRUE) AND (MC_Axis004.Details.Homed=FALSE) THEN
  Hm5_Ex:=TRUE;
END_IF;

// 轴1的原点复位完成后，执行MC_MoveVelocity
IF Hm1_D=TRUE THEN
  Vel_Ex:=TRUE;
END_IF;

// 轴2的原点已确定，且MC_MoveVelocity的Vel_InVel为TRUE时
// 从轴启动轴2设定的MC_CamIn
IF (Vel_InVel=TRUE) AND (MC_Axis001.Details.Homed=TRUE) THEN
  Camin1_Ex := TRUE;
END_IF;

// 轴3的原点已确定，且MC_MoveVelocity的Vel_InVel为TRUE时
// 从轴启动轴3设定的MC_CamIn
IF (Vel_InVel=TRUE) AND (MC_Axis002.Details.Homed=TRUE) THEN
  Camin2_Ex := TRUE;
END_IF;

// 轴4的原点已确定，且MC_MoveVelocity的Vel_InVel为TRUE时
// 从轴启动轴4设定的MC_CamIn
IF (Vel_InVel=TRUE) AND (MC_Axis003.Details.Homed=TRUE) THEN
  Camin3_Ex := TRUE;
END_IF;

// 轴5的原点已确定，且MC_MoveVelocity的Vel_InVel为TRUE时
// 从轴启动轴5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

结束通过输入参数指定的轴的凸轮动作

指令	名称	FB/ FUN	图形表现	ST表现
MC_CamOut	解除凸轮动作	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	在上升沿开始指令。
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. 关于指令单位，请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*2. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

* 请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	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
CommandAborted	<ul style="list-style-type: none"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 执行MC_Stop指令中，启动本指令时 	<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含有异常因素时	异常已解除时

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. 请在使用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

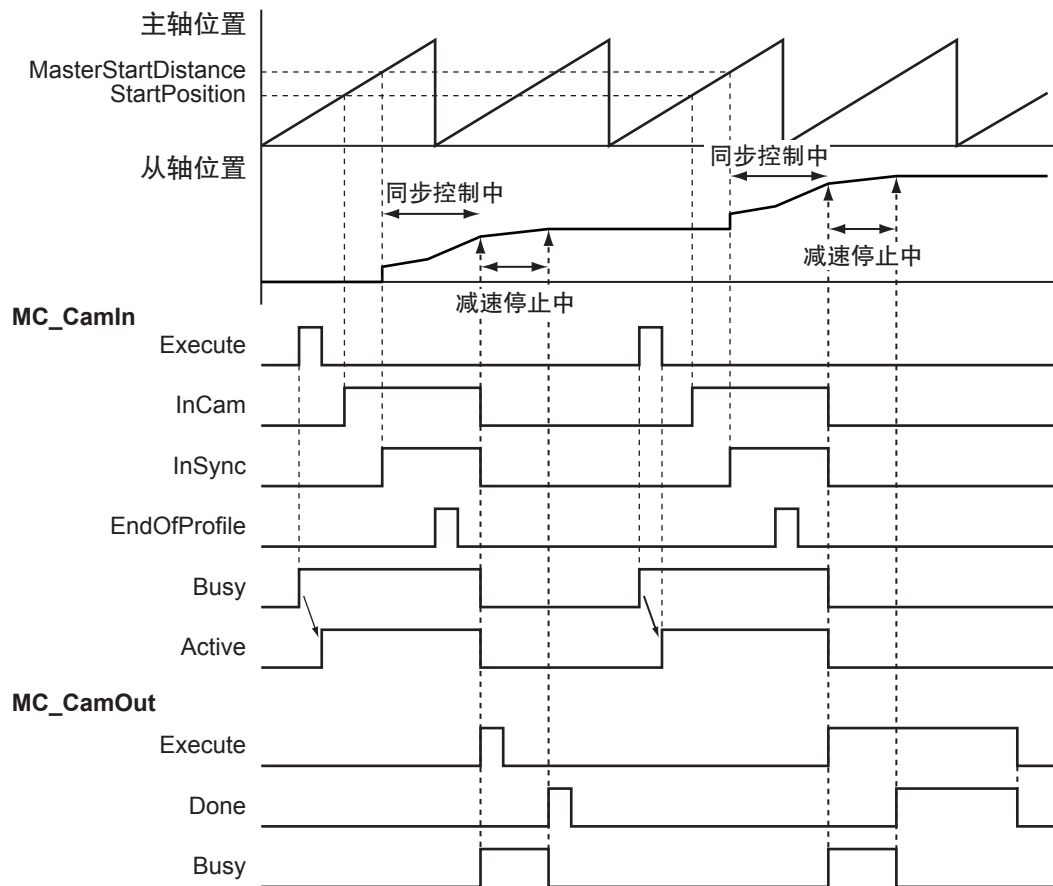
- 通过本指令解除从轴的凸轮动作。
- 将Execute(启动)设为TRUE时，根据Deceleration(减速度)，以速度“0”为目标减速。
- 指令速度变为“0”后即完成。
- 对未进行凸轮动作的轴启动本指令时，会发生异常。



使用注意事项

- 凸轮数据变量为全局变量，可以从多个任务中查看或变更凸轮数据变量值。从多个任务中变更凸轮数据变量值时，请编写通过多个任务进行改写处理时相互不会发生冲突的程序。
- 使用“全局变量的任务间排他性控制”功能进行凸轮数据变量的排他性控制时，请不要使用排他对象的凸轮数据变量启动运动指令。否则会发生轻度故障等级的异常“凸轮表指定错误(错误代码：5439 Hex)”。

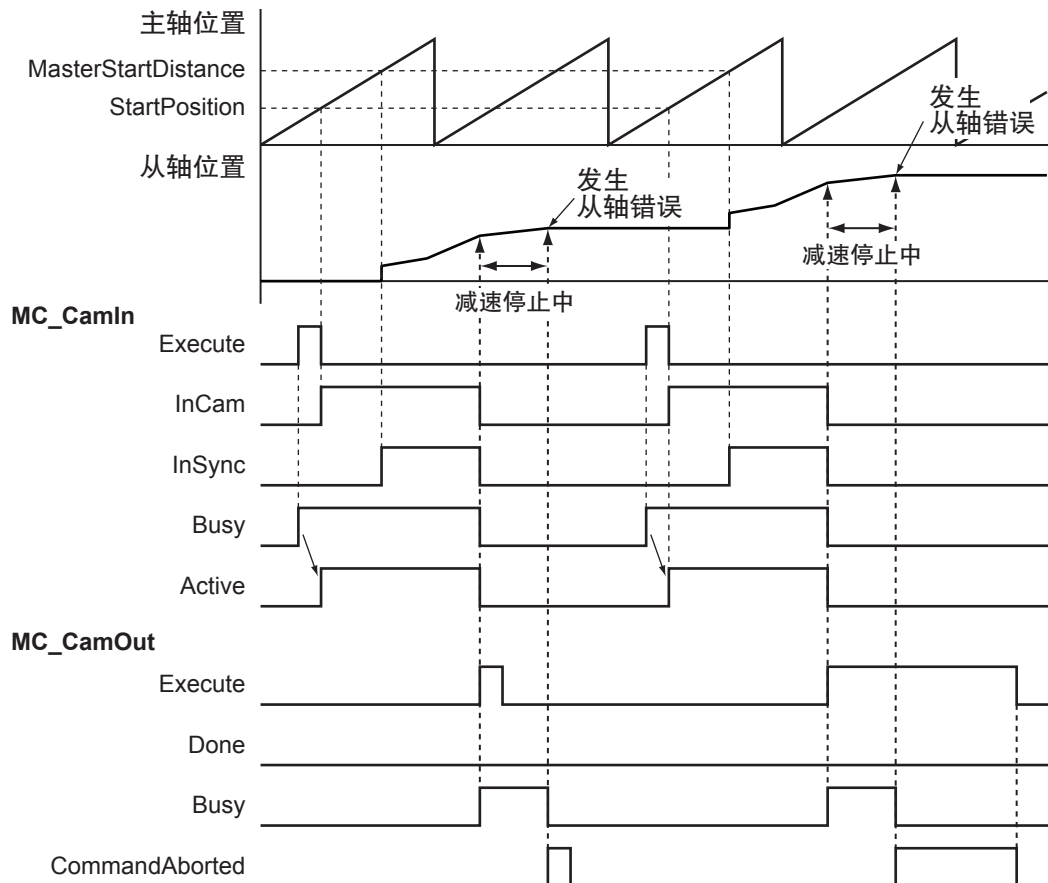
时序图



指令的中止

本指令执行中从轴发生轴异常时，CommandAborted(执行中断)变为TRUE，Busy(动作中)变为FALSE。并且，轴根据本指令的Deceleration(减速度)进行减速。

关于轴异常，请参阅 □ “A-1 错误代码一览(P.A-2)”。



重启运动指令

无法重启本指令。

重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

● 其它指令执行中的本指令启动

MC_CamIn(凸轮动作开始)指令执行中启动本指令时，中断执行(CommandAborted)MC_CamIn(凸轮动作开始)指令，本指令变为Busy(执行中)。

MC_CamIn(凸轮动作开始)指令非执行中时，如果启动本指令，会发生Error(错误)。

● 本指令执行中的其它指令启动

对本指令执行运动指令多重启动时，请指定从轴。

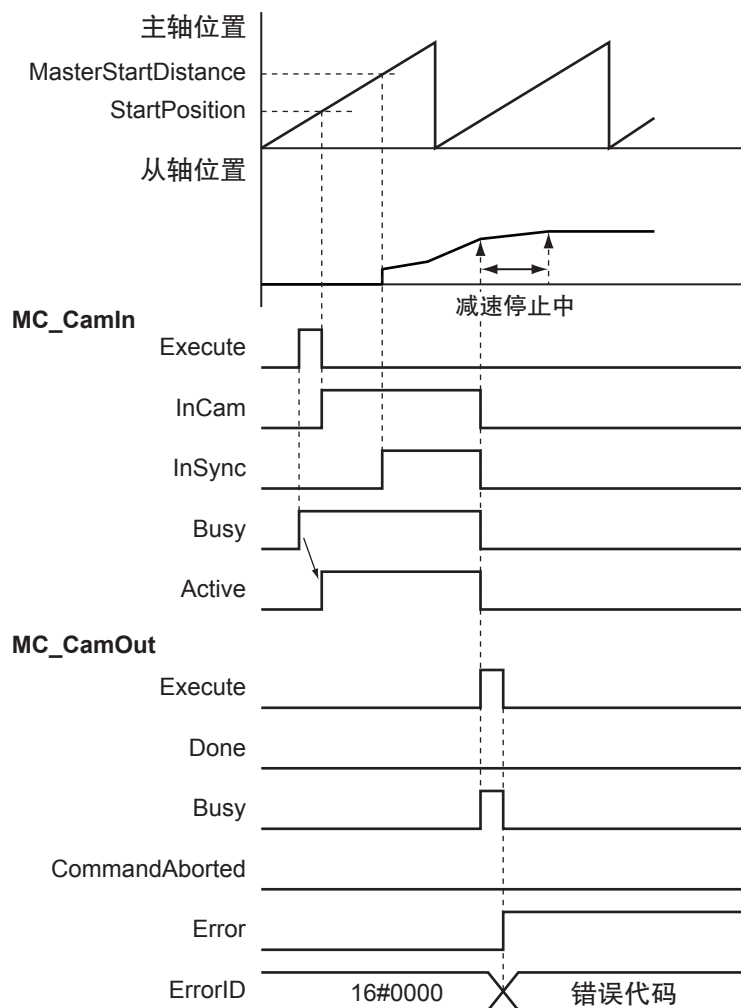
对本指令进行多重启动运动指令时，可选择中断、等待。

异常

启动本指令发生异常时，Error(错误)变为TRUE。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 □ “A-1 错误代码一览(P.A-2)”。

MC_GearIn

设定主轴和从轴间的齿轮比，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	在上升沿开始指令。
Ratio Numerator	齿轮比分子	DINT*1	正数或负数 *1	10000	指定主轴和从轴间的电子齿轮的分子。
Ratio Denominator	齿轮比分母	UDINT*2	正数	10000	指定主轴和从轴间的电子齿轮的分母。
Reference Type*3	位置类型选择	_eMC_REFERENCE_TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 *4	指定位置类型。 0: 指令位置(最近任务周期*5下的计算值) 1: 反馈位置(同一任务周期*5下的取值) 2: 指令位置(同一任务周期*5下的计算值)
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	缓存模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0 *2	指定多重启动运动指令时的动作。 0: 中断 1: 等待

*1. Ver.1.02以上的CPU单元和Ver.1.03以上的Sysmac Studio组合时可使用。上述以下版本进行组合时，数据类型为“UINT”，有效范围为“正数”。

*2. Ver.1.02以上的CPU单元和Ver.1.03以上的Sysmac Studio组合时可使用。上述以下版本进行组合时，数据类型为“UINT”。

*3. 使用_mcLatestCommand时，必须在满足以下关系的条件下使用主轴、从轴。

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

- *4. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。
- *5. 任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先级5)的周期。
- *6. 关于指令单位，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 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	齿轮比到达	BOOL	TRUE, FALSE	从轴达到目标速度时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

* 请参阅 □ “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
InGear	以下关系成立时 <ul style="list-style-type: none"> 加速中：从轴速度\geq主轴速度\times齿轮比 减速中：从轴速度\leq主轴速度\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	开始指令后	<ul style="list-style-type: none"> Error变为TRUE时 CommandAborted变为TRUE时
CommandAborted	<ul style="list-style-type: none"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 执行MC_Stop指令中，启动本指令时 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含有异常因素时	异常已解除时

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. 请在使用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[*])。

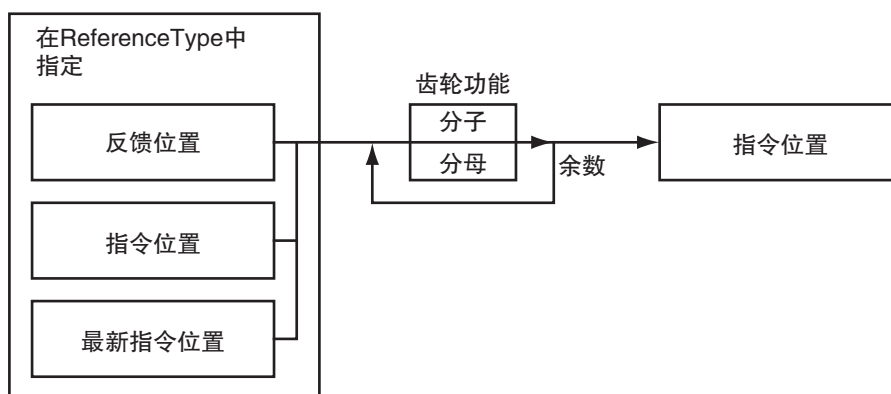


使用注意事项

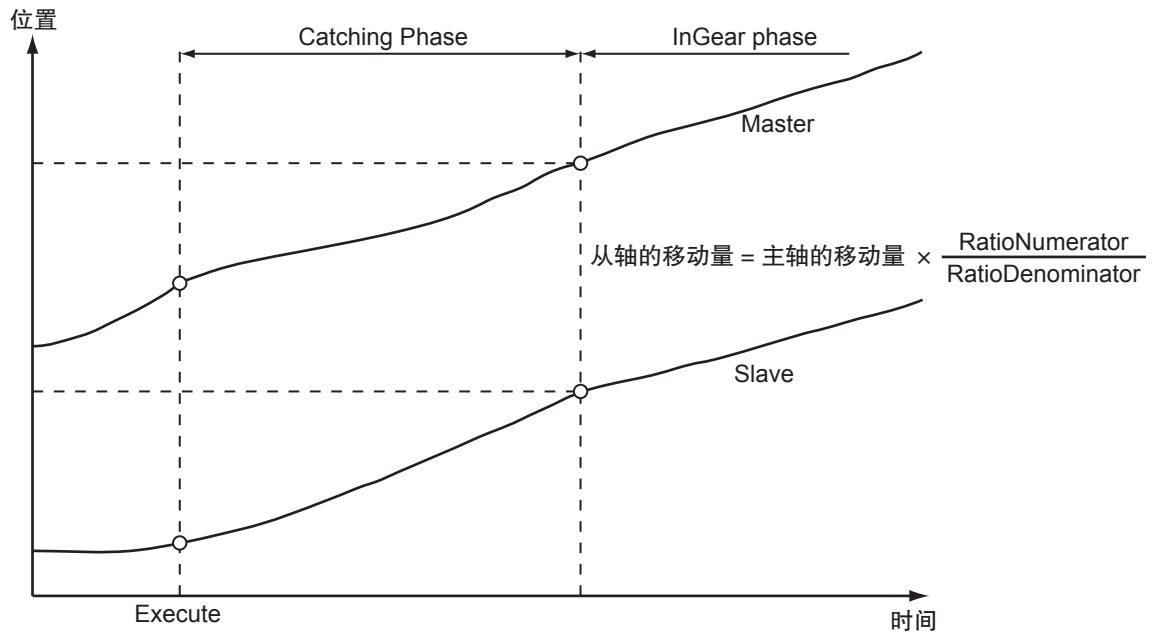
主轴和从轴指定同一轴时，会发生轻度故障等级的“主轴从轴相同(错误代码：5436 Hex)”的异常。

功能说明

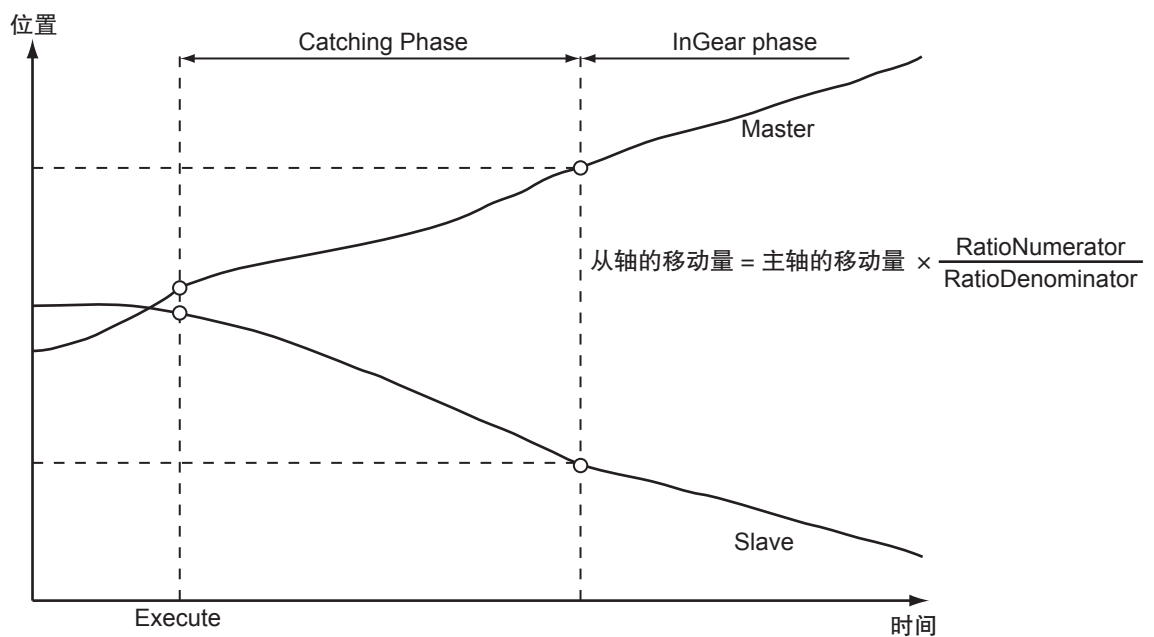
- 通过Slave(从轴)指定动作对象轴，指定 RatioNumerator(齿轮比分子)、RatioDenominator(齿轮比分母)、ReferenceType(位置类型选择)、Acceleration(加速度)、Deceleration(减速度)，进行齿轮动作。
- 可对Master(主轴)指定指令位置和反馈位置、最新的指令位置。



- 开始动作后，Slave(从轴)以Master(主轴)速度乘以齿轮比得到的速度为目标速度，进行加减速动作。
- 达到目标速度之前称为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)沿Master(主轴)的同方向移动。



- 齿轮比为负数时，Slave(从轴)沿Master(主轴)的反方向移动。



- 在Execute(启动)的上升沿，开始电子齿轮动作。

**使用注意事项**

Ver.1.09以下的CPU单元使用本指令时，请勿对Master(主轴)执行MC_SetPosition(当前位置变更)指令。对Master(主轴)执行MC_SetPosition(当前位置变更)指令时，可能出现Slave(从轴)急速跟踪的情况，非常危险。

要对Master(主轴)使用MC_SetPosition(当前位置变更)指令时，请解除Master(主轴)和Slave(从轴)的关系后执行该指令。

有关对于主轴的注意事项，请参阅 “同步控制的主轴及辅轴的注意事项(P.1-6)”。

● ReferenceType(位置类型选择)

可从下列选项中选择位置的类型。

- **_mcCommand**: 指令位置(最近任务周期下的计算值)
对当前周期，使用之前任务周期时计算的主轴指令位置。
在计算从轴指令位置之前的固定周期任务中，使用计算的主轴指令位置。
- **_mcFeedback**: 同一任务周期下的取值
使用同一任务周期时获取的主轴反馈位置。
- **_mcLatestCommand**: 指令位置(同一任务周期下的计算值)
使用同一任务周期时计算的主轴指令位置。
也可通过**_mcCommand**使用最新信息。但是，设定时必须保证主轴编号比从轴编号小。
否则，本指令的Error(错误)将变为TRUE。ErrorID(错误代码)输出“主轴/从轴 轴号非升序(错误代码: 5438 Hex)”。

**使用注意事项**

这里的任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先度5)的周期。同样，固定周期任务表示原始恒定周期任务或固定周期任务(执行优先度5)。

**参考**

相比选择之前任务周期时的计算值，选择同一任务周期时计算的指令位置时，同步精度更高。但是必须使用主轴、从轴，确保“Master(主轴)设定的运动控制系统变量的轴号 < Slave(从轴)设定的运动控制系统变量的轴号”。

● 轴种类与位置类型的关系

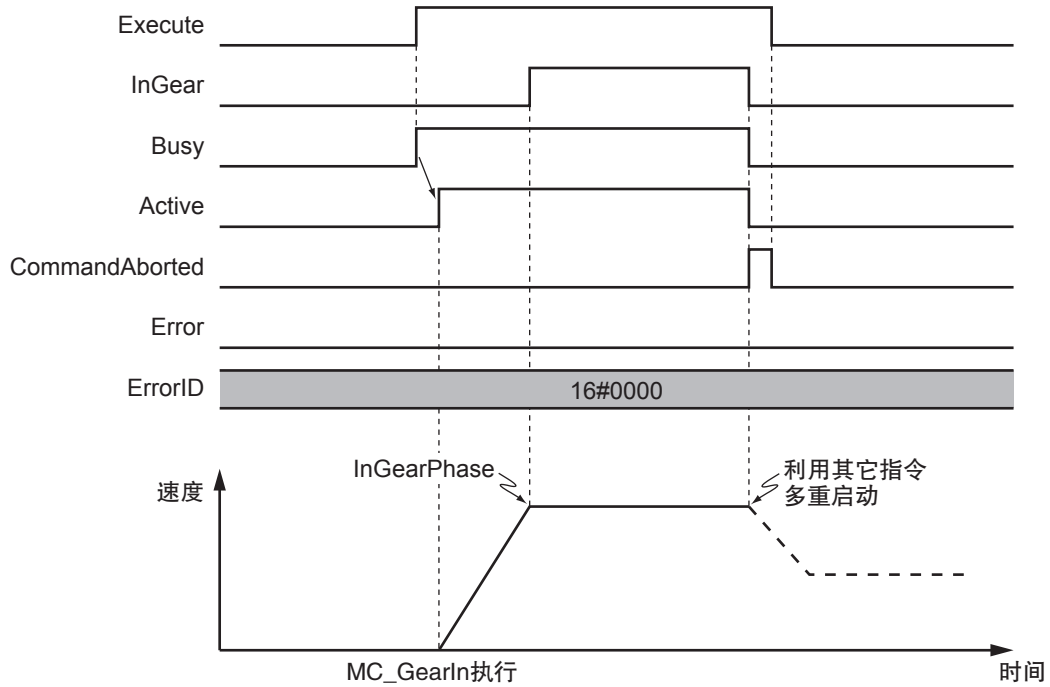
可监控的轴种类和要监控的位置类型的关系如下所示。

轴种类	ReferenceType	
	_mcCommand、或 _mcLatestCommand	_mcFeedback
伺服轴	○	○
编码器轴	× *1	○
虚拟伺服轴	○	○
虚拟编码器轴	× *1	○

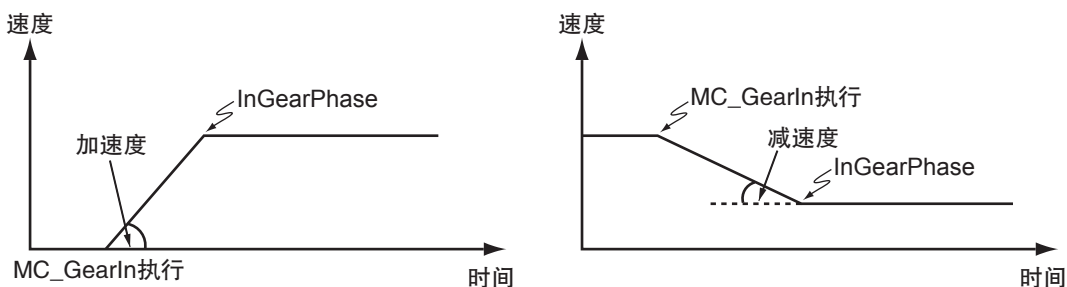
*1. 启动指令时，发生“超过位置类型选择范围(错误代码: 5430 Hex)”错误。

时序图

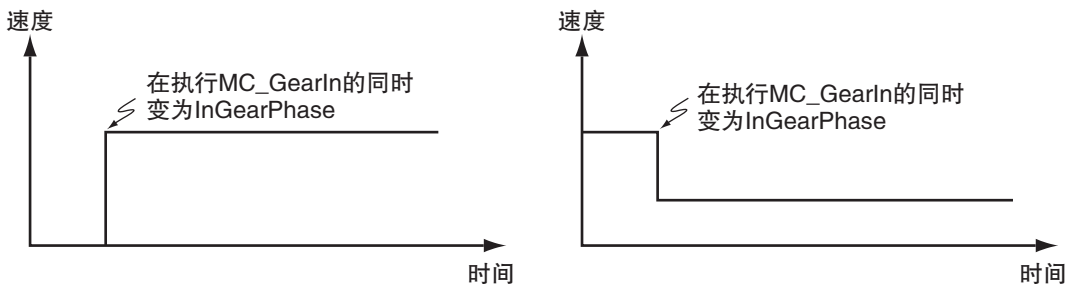
- 在启动Execute(启动)的同时，Busy(执行中)变为TRUE。在下一个周期Active(控制中)变为TRUE。
- 达到目标速度时，InGear(达到齿轮比)变为TRUE。
- 利用其它指令中断本指令时，CommandAborted(执行中断)变为TRUE，Busy(执行中)、Active(控制中)、InGear(达到齿轮比)变为FALSE。
- 要中途结束电子齿轮动作，可使用MC_GearOut(齿轮动作解除)指令、或MC_Stop(强制停止)指令。



可在输入变量中指定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，不作加减速而达到目标速度。

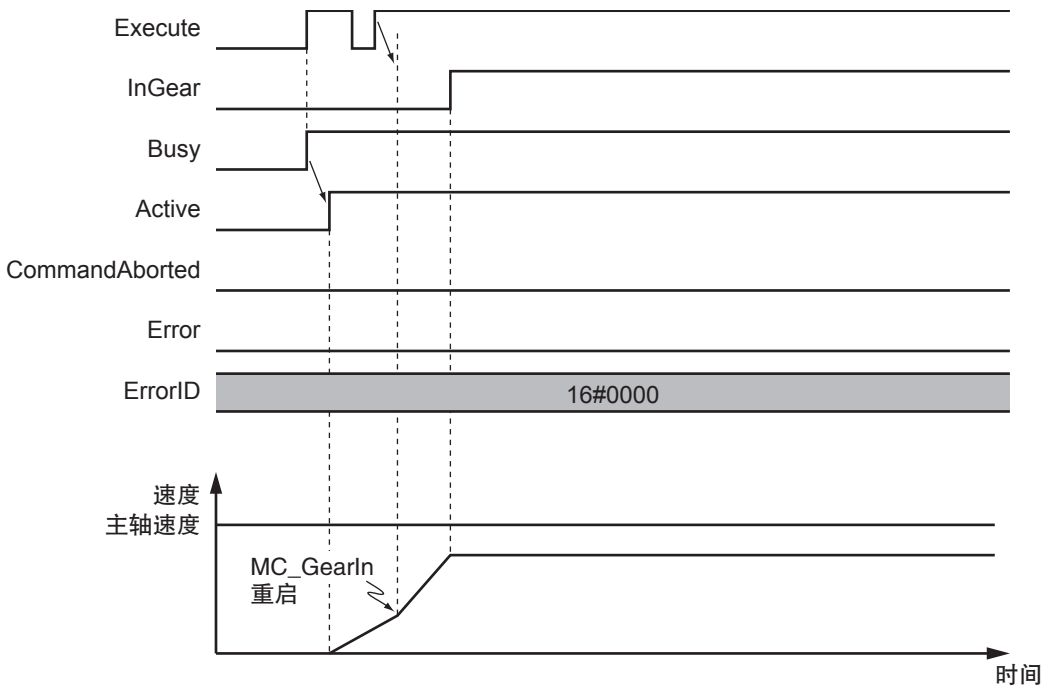


重启运动指令

在定位动作中变更输入变量，再次将Execute(启动)设为TRUE，可变更本指令的动作。
 重启运动指令可变更的输入变量为，RatioNumerator(齿轮比分子)、RatioDenominator(齿轮比分母)、Acceleration(加速度)、Deceleration(减速度)。

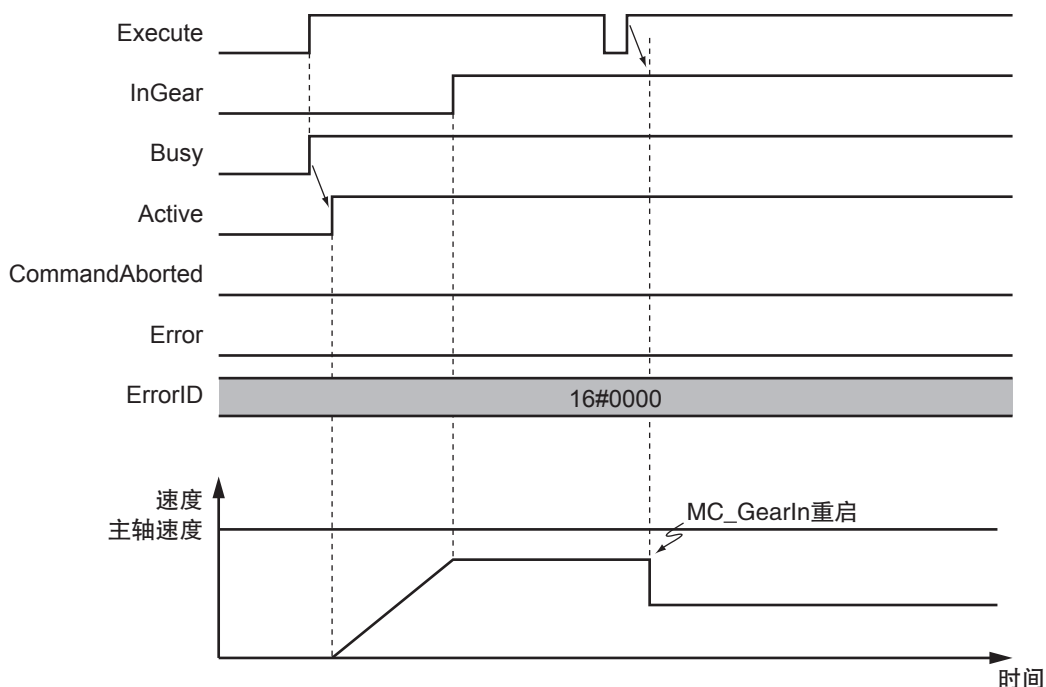
重启运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

作为参考，通过Catching phase(追赶中)重启，变更Acceleration(加速度)时的时序图如下所示。



通过InGear phase(齿轮同步中)重启, 变更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)均设为“0”时相同。



多重启动运动指令

多重启动运动指令的详情, 请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动

对当前正在执行的指令启动本指令, 可切换或缓存到本指令。

各轴可缓存到1个。

多重启动指令时的本指令的动作由BufferMode(缓存模式选择)指定。

缓存模式选择	说明
中断	立即中止当前正在执行的指令, 切换为本指令。 轴的动作方向因指令切换而反转时, 根据轴参数中的“反转时动作”进行反转。
等待	当前正在执行的指令正常完成后, 已缓存的本指令自动启动。

关于BufferMode(缓存模式选择), 请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 本指令执行中的其它指令启动

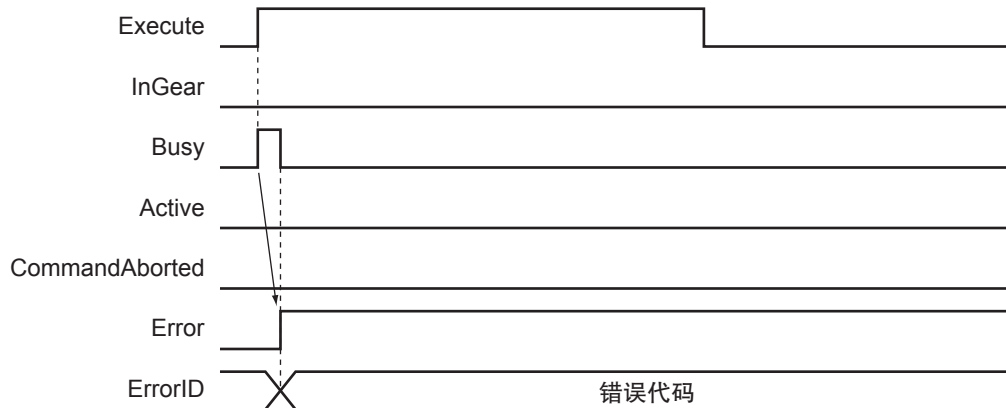
可对从轴以中断方式执行指令的多重启动。

此时, 中止齿轮动作, 开始多重启动的动作。

无法以非中断方式多重启动指令。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。



参考

- 同步控制中，不受主轴的异常状态的影响。主轴的异常状态解除，正常动作时，从轴继续进行电子齿轮动作。
- 启动本指令时以及执行本指令中，从轴发生异常时，不会对主轴产生影响。

● 错误代码

关于指令发生的异常，请参阅  “A-1 错误代码一览(P.A-2)”。

示例程序

下面，对与齿轮比成正比动作的示例程序进行说明。



参考

作为保留(Reserved)处理的输入变量只能指定为初始值。
本例中，不指定参数。

参数设定

该示例程序中使用的最少的必要设定如下所示。

● 轴参数的设定

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

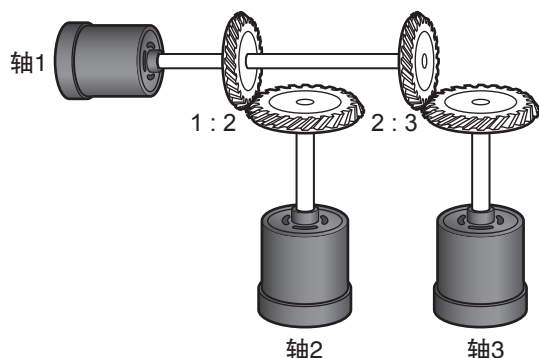
环计数器

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



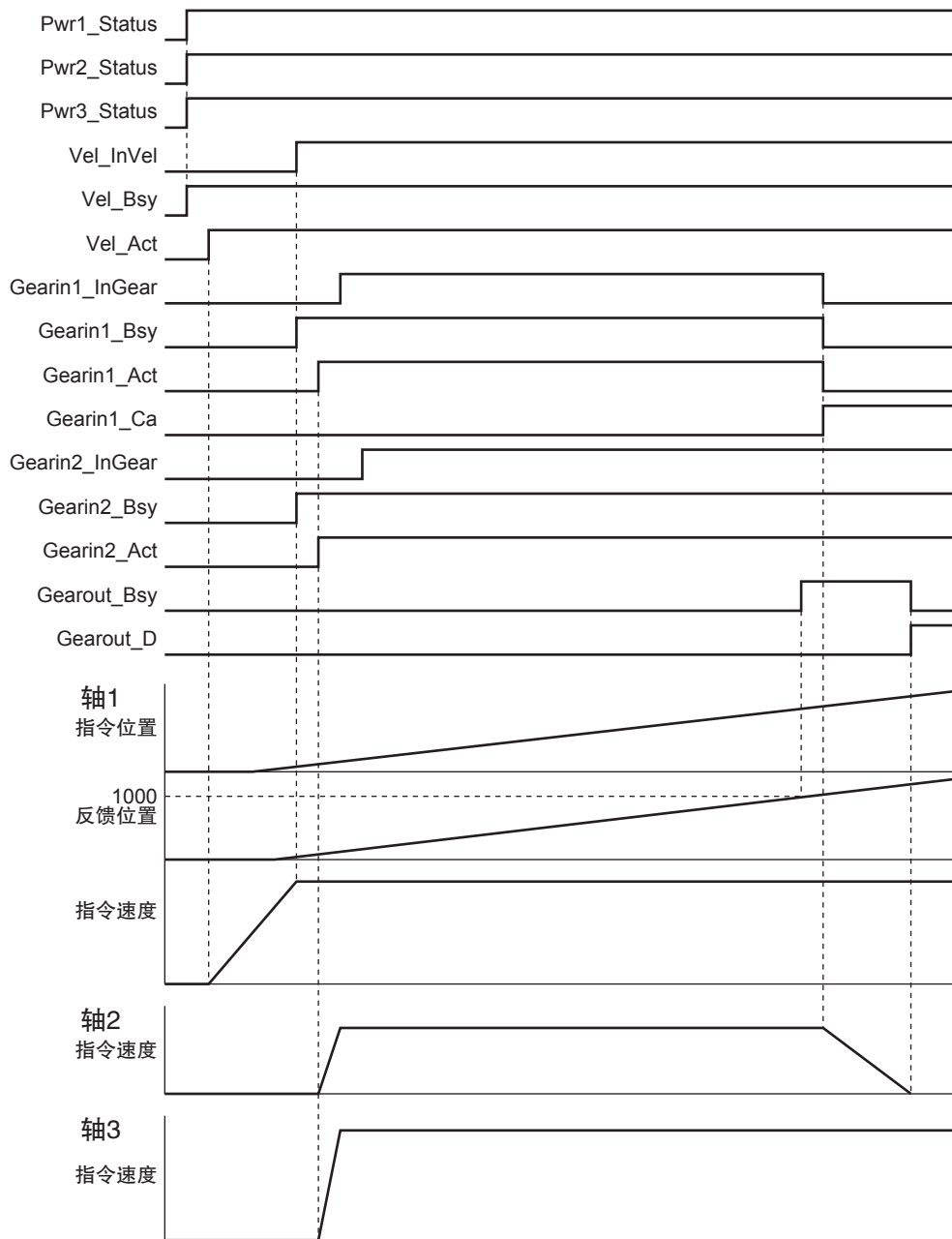
- 1 主轴的开始**
主轴(轴1)是实际的伺服轴，控制速度进行动作。
- 2 从轴的启动**
主轴的反馈速度达到目标速度后，相对主轴的反馈位置，从轴(轴2)以齿轮比1:2进行齿轮动作，从轴(轴3)以齿轮比2:3进行齿轮动作。
- 3 从轴的停止**
主轴的反馈位置(MC_Axis000.Act.Pos)达到1000.0以上时，解除轴2的齿轮动作，以减速度DecRate使轴2减速停止。此时，轴3继续进行齿轮动作。

梯形图

● 主要变量

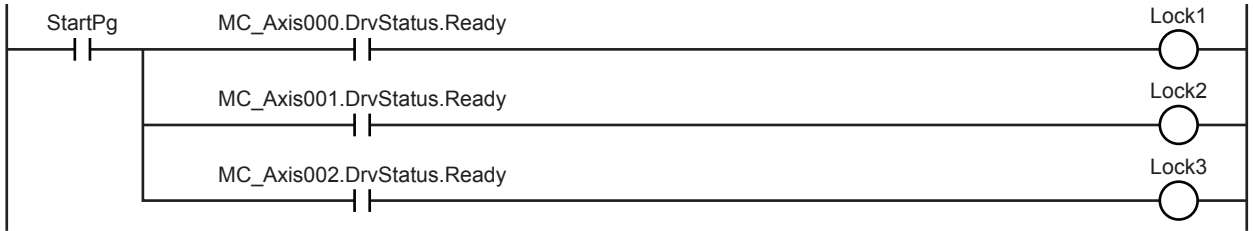
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	是主轴轴1的轴变量。
MC_Axis000.Act.Pos	LREAL	-	表示轴1的反馈当前位置。
MC_Axis001	_sAXIS_REF	-	是从轴轴2的轴变量。
MC_Axis002	_sAXIS_REF	-	是从轴轴3的轴变量。
Pwr1_Status	BOOL	FALSE	分配给MC_Power的实例PWR1的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
Pwr2_Status	BOOL	FALSE	分配给MC_Power的实例PWR2的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
Pwr3_Status	BOOL	FALSE	分配给MC_Power的实例PWR3的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则进入伺服ON状态。
Vel_InVel	BOOL	FALSE	分配给MC_MoveVelocity的实例VEL的输出InVelocity的变量。达到目标速度时，该变量变为TRUE。
Gearin1_Act	BOOL	FALSE	分配给MC_GearIn的实例GEARIN1的输出Active的变量。GEARIN1变为控制中时，该变量变为TRUE。
Gearout_Ex	BOOL	FALSE	该变量由FALSE→TRUE时，启动MC_GearOut的实例GEAROUT。

● 时序图

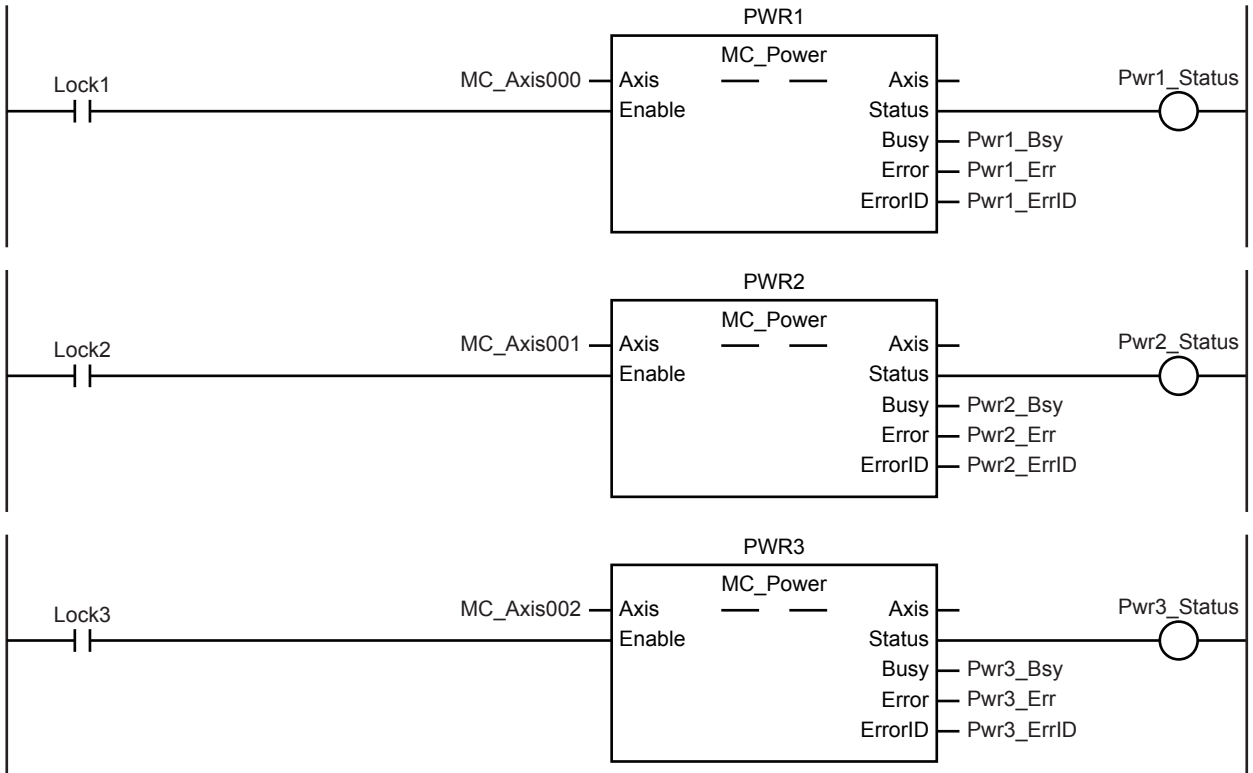


● 示例程序

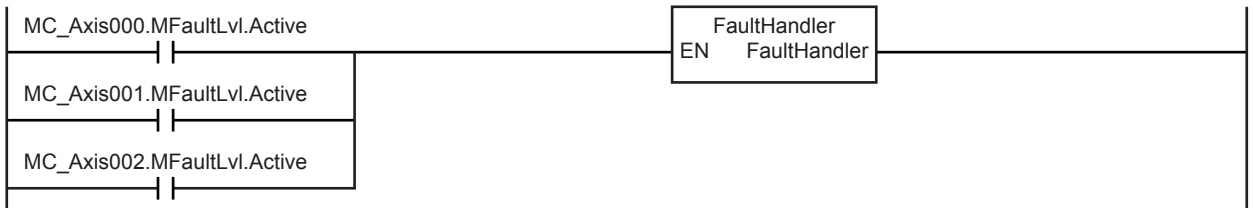
触点StartPg变为TRUE时，确认各轴的伺服驱动器处于伺服准备就绪状态



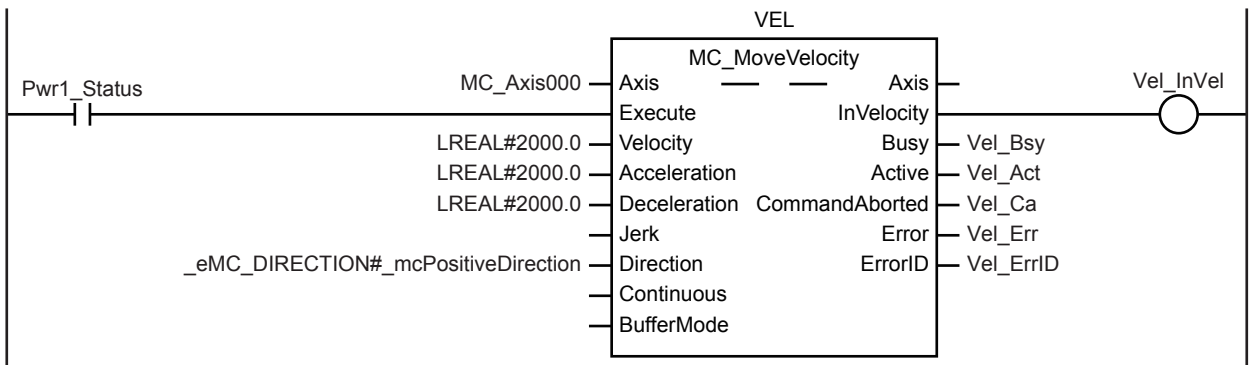
如果处于伺服准备就绪状态，则将各轴设为伺服ON状态



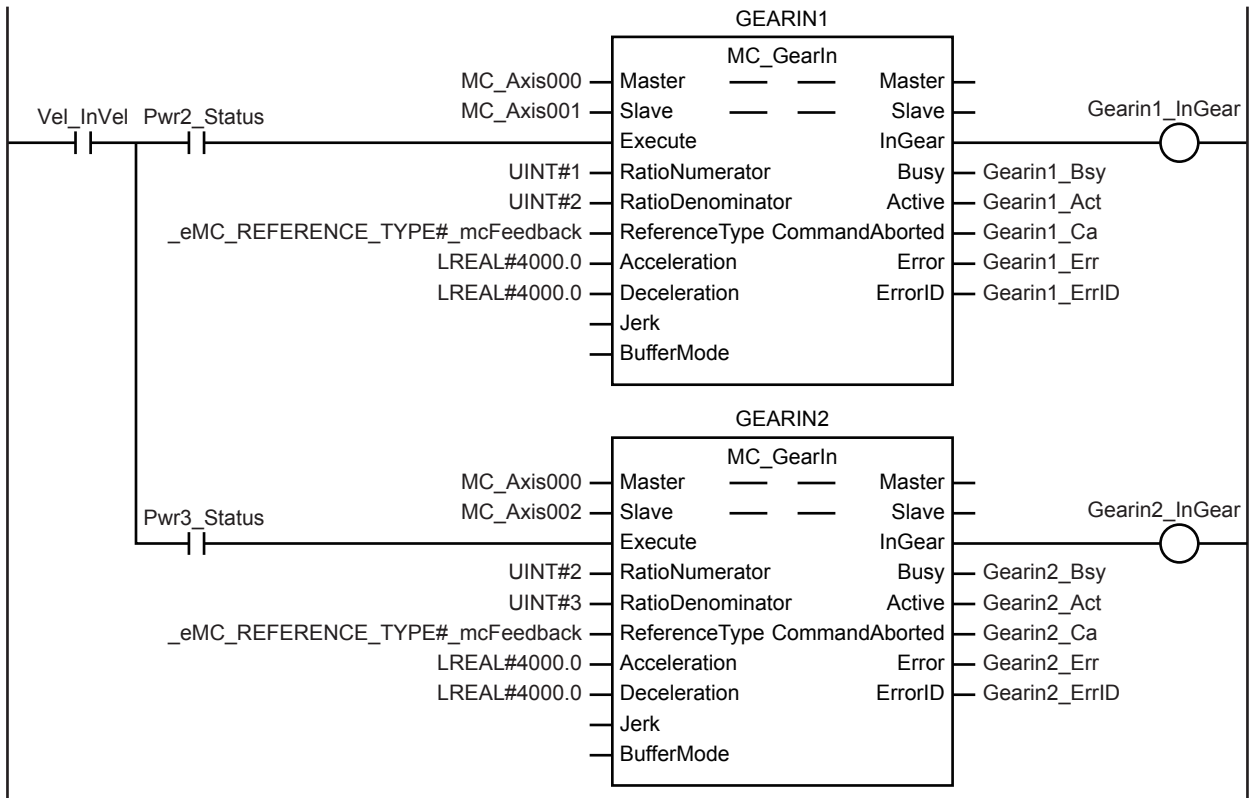
轴构成发生轻度故障程度的异常时，执行设备的异常处理(FaultHandler)。发生异常时的处理(FaultHandler)由客户根据装置进行编程



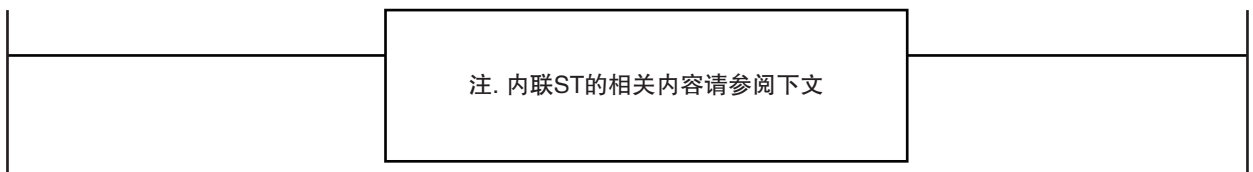
主轴(轴1)进入可运行状态后，启动MC_MoveVelocity(速度控制)指令



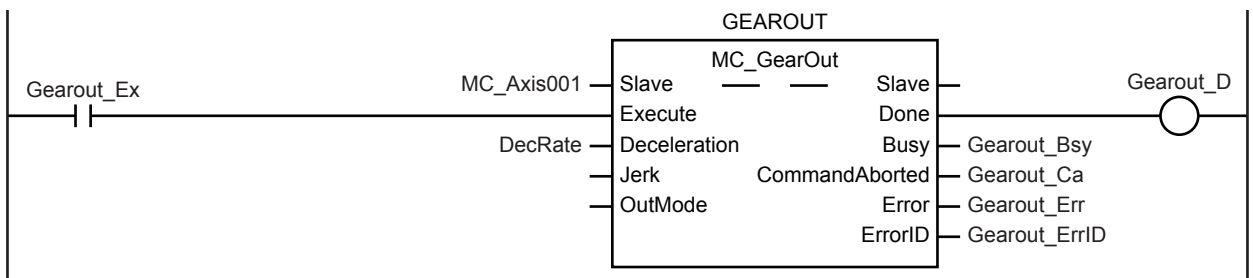
主轴(轴1)的反馈速度达到目标速度后, 从轴(轴2)以齿轮比1:2、从轴(轴3)以齿轮比2:3开始齿轮动作



从轴(轴2)在齿轮动作中, 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, 进行减速停止



内联ST的内容

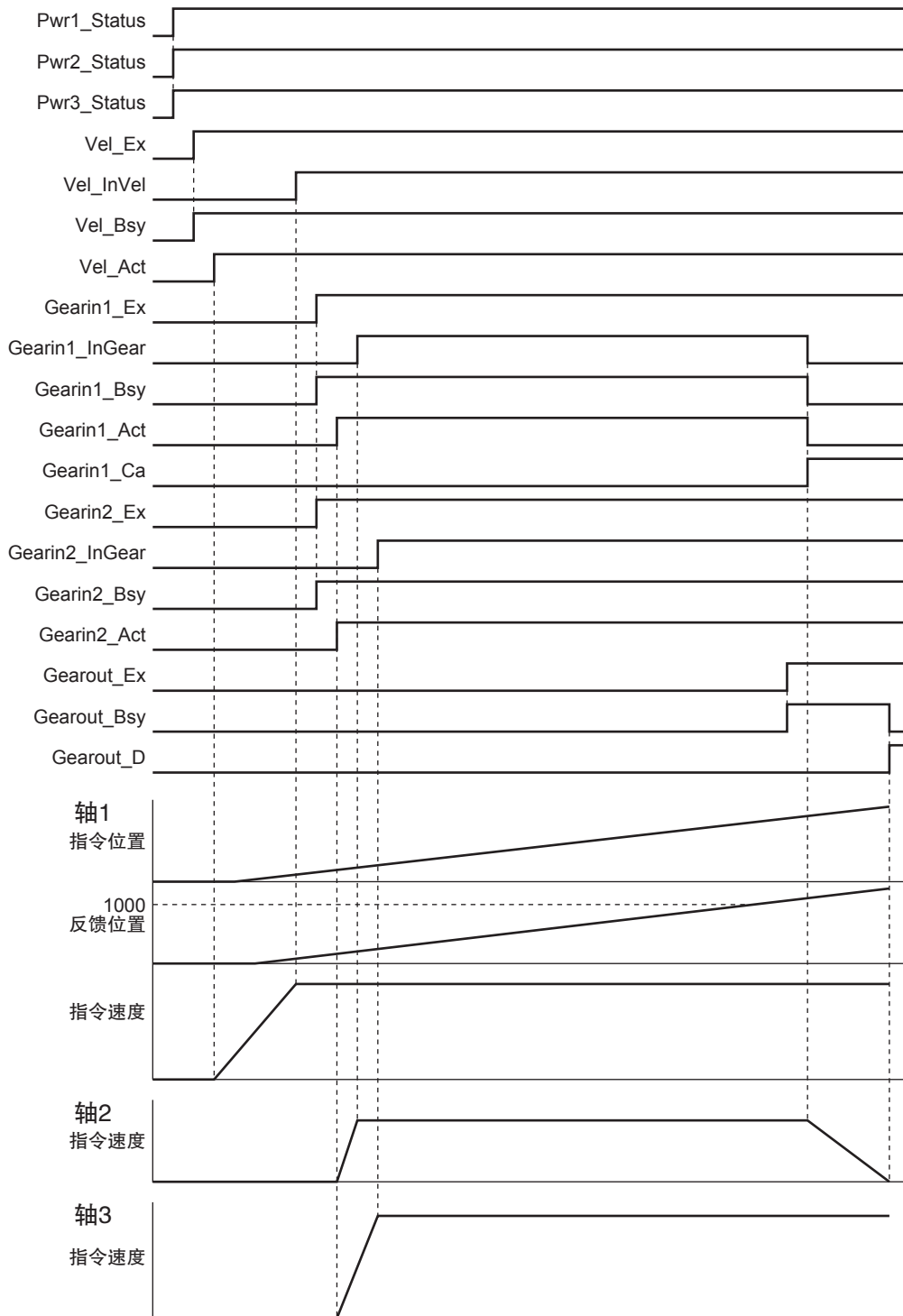
```
IF (Gearin1_Act=TRUE) AND (MC_Axis000.Act.Pos>=LREAL#1000.0) THEN
    Gearout_Ex := TRUE;
END_IF;
```

结构文本(ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	是主轴轴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	-	是从轴轴2的轴变量。
MC_Axis002	_sAXIS_REF	-	是从轴轴3的轴变量。
Pwr1_Status	BOOL	FALSE	分配给MC_Power的实例PWR1的输出变量Status的变量。进入伺服ON状态时, 该变量变为TRUE。
Pwr2_Status	BOOL	FALSE	分配给MC_Power的实例PWR2的输出变量Status的变量。进入伺服ON状态时, 该变量变为TRUE。
Pwr3_Status	BOOL	FALSE	分配给MC_Power的实例PWR3的输出变量Status的变量。进入伺服ON状态时, 该变量变为TRUE。
StartPg	BOOL	FALSE	如果该变量为TRUE, EtherCAT的过程数据通信已建立, 则进入伺服ON状态。
Vel_InVel	BOOL	FALSE	分配给MC_MoveVelocity的实例VEL的输出InVelocity的变量。达到目标速度时, 该变量变为TRUE。
Gearin1_Act	BOOL	FALSE	分配给MC_GearIn的实例GEARIN1的输出Active的变量。GEARIN1变为控制中时, 该变量变为TRUE。
Gearout_Ex	BOOL	FALSE	该变量由FALSE→TRUE时, 启动MC_GearOut的实例GEAROUT。
Vel_Ex	BOOL	FALSE	该变量由FALSE→TRUE时, 启动MC_MoveVelocity的实例VEL。
Gearin1_Ex	BOOL	FALSE	该变量由FALSE→TRUE时, 启动MC_GearIn的实例GEARIN1。
Gearin2_Ex	BOOL	FALSE	该变量由FALSE→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;

//确认StartPg为TRUE，且伺服驱动器处于伺服准备就绪状态，
//则将轴1设为伺服ON状态。
//如果未处于伺服准备就绪状态，则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
  Pwr1_En:=TRUE;
ELSE
  Pwr1_En:=FALSE;
END_IF;

//确认StartPg为TRUE，且伺服驱动器处于伺服准备就绪状态，
//则将轴2设为伺服ON状态。
//如果未处于伺服准备就绪状态，则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
  Pwr2_En:=TRUE;
ELSE
  Pwr2_En:=FALSE;
END_IF;

//确认StartPg为TRUE，且伺服驱动器处于伺服准备就绪状态，
//则将轴3设为伺服ON状态。
//如果未处于伺服准备就绪状态，则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis002.DrvStatus.Ready=TRUE) THEN
  Pwr3_En:=TRUE;
ELSE

```



```

    Pwr3_En:=FALSE;
END_IF;

//轴1~轴3发生轻度故障后,执行异常时处理FaultHandler。
//发生异常时的处理(FaultHandler)由客户根据装置进行编程。
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) OR
(MC_Axis002.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;

//轴1处于伺服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处于伺服ON状态时,启动主轴设为1、从轴设为轴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处于伺服ON状态时,启动主轴设为1、从轴设为轴3的
MC_GearIn
IF (Vel_InVel=TRUE) AND (Pwr3_Status=TRUE) THEN
    Gearin2_Ex := TRUE;
END_IF;

// 轴2齿轮动作中,且轴1的反馈位置达到1000.0以上时,
// 将从轴设为轴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
);

//将轴1设为主轴、轴2设为从轴的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
);

//将轴1设为主轴、轴3设为从轴的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

设定主轴和从轴间的齿轮比，进行电子齿轮动作。
指定开始同步的主轴及从轴的位置。

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	在上升沿开始指令。
Ratio Numerator	齿轮比分子	DINT*1	正数或负数 *1	10000	指定主轴和从轴间的电子齿轮的分子。
Ratio Denominator	齿轮比分母	UDINT*2	正数	10000	指定主轴和从轴间的电子齿轮的分母。
Reference Type*3	位置类型选择	_eMC_ REFERENCE_ TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0*4	指定位置类型。 0: 指令位置(最近任务周期*5下的计算值) 1: 反馈位置(同一任务周期*5下的取值) 2: 指令位置(同一任务周期*5下的计算值)
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。单位为[指令单位]。*6
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。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。不作设定就执行动作，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)	缓存 模式选择	_eMC_ BUFFER_ MODE	0: _mcAborting	0 ^{*2}	(Reserved)

*1. Ver.1.02以上的CPU单元和Ver.1.03以上的Sysmac Studio组合时可使用。上述以下版本进行组合时，数据类型为“UINT”，有效范围为“正数”。



*2. Ver.1.02以上的CPU单元和Ver.1.03以上的Sysmac Studio组合时可使用。上述以下版本进行组合时，数据类型为“UINT”。

*3. 使用_mcLatestCommand时，必须在满足以下关系的条件下使用主轴、从轴。

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


*4. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

*5. 任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先度5)的周期。

*6. 关于指令单位，请参阅  “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或  “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 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	跟踪中	BOOL	TRUE, FALSE	开始为同步进行加减速动作时变为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	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

*  请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
StartSync	轴开始移动后	<ul style="list-style-type: none"> • Error变为TRUE时 • CommandAborted变为TRUE时
InSync	从轴位置达到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	接收指令时	<ul style="list-style-type: none"> • Error变为TRUE时 • CommandAborted变为TRUE时
CommandAborted	<ul style="list-style-type: none"> • 利用其它指令多重启动运动指令(中断), 中止本指令时 • 因发生异常, 中止本指令时 • 发生异常过程中, 启动本指令时 • 执行MC_Stop指令中, 启动本指令时 • 启动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含有异常因素时	异常已解除时

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. 请在使用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[*])。

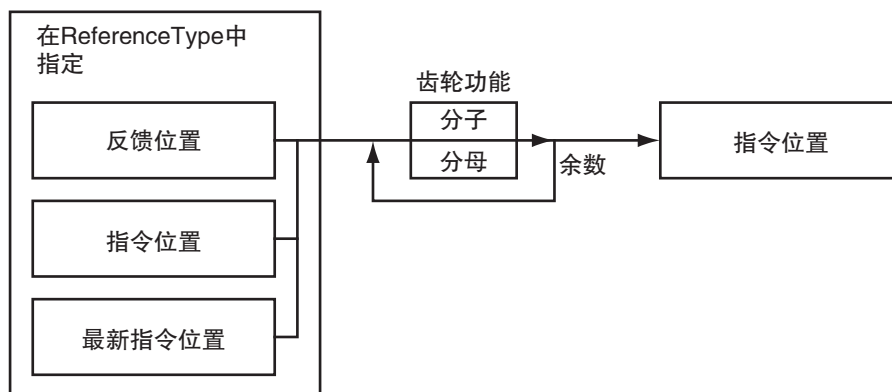


使用注意事项

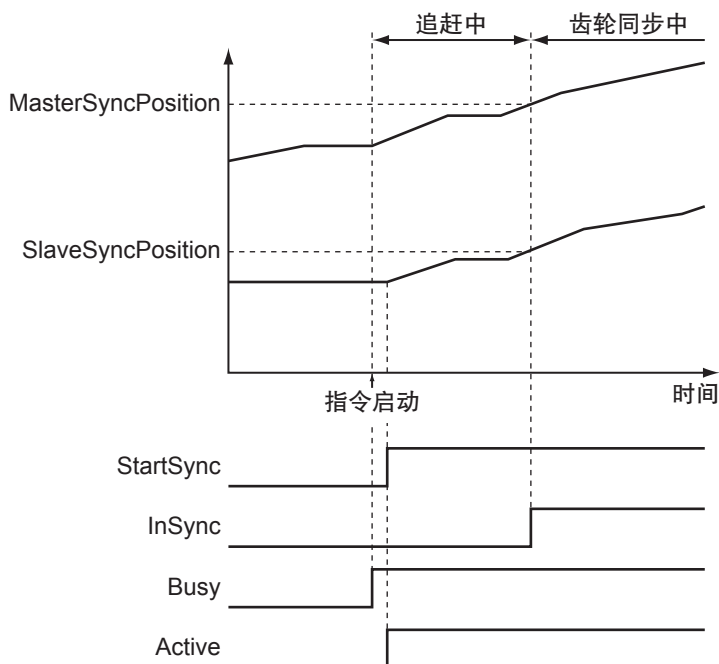
主轴和从轴指定同一轴时, 会发生轻度故障等级的“主轴从轴相同(错误代码: 5436 Hex)”的异常。

功能说明

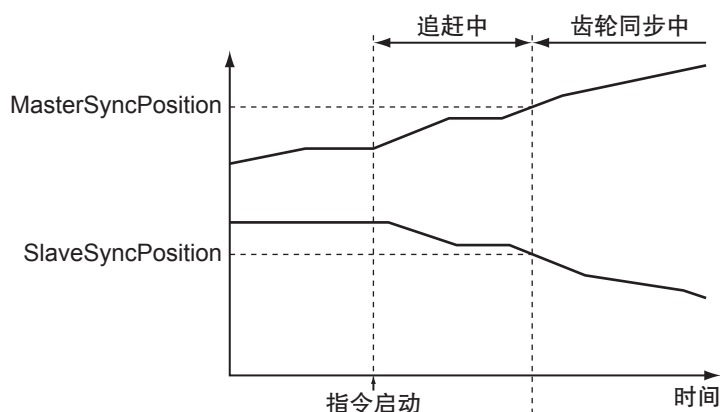
- 通过 Slave(从轴)指定动作对象轴, 指定 RatioNumerator(齿轮比分子)、RatioDenominator(齿轮比分母)、ReferenceType(位置类型选择)、Acceleration(加速度)、Deceleration(减速度), 进行齿轮动作。
- 可对 Master(主轴)指定指令位置和反馈位置、最新的指令位置。



- 开始动作后, Slave(从轴)通过追赶动作, 与Master(主轴)同步进行加减速动作。
- 从轴到达从轴同步位置之前称为Catching phase(追赶中)、到达后称为InGear phase(齿轮同步中)。任何区间均与主轴的位置同步。
- 输入变量的Velocity(目标速度)即为Catching phase(追赶中)时的目标速度。
- 从轴沿开始动作时主轴的相同方向进行动作。
开始动作时, 如果主轴速度为“0”, 会发生异常。
此外, 主轴正方向动作, 且SlaveSyncPosition(从轴同步位置)比启动时的从轴位置小时, 从轴进行反转动作。
同样, 主轴负方向动作, 且SlaveSyncPosition(从轴同步位置)比启动时的从轴位置大时, 从轴也进行反转动作。
- 主轴速度不同周期变化较大时, 从轴的速度也不固定。
- 齿轮比为正数时, Slave(从轴)沿Master(主轴)的同方向移动。



- 齿轮比为负数时，Slave(从轴)沿Master(主轴)的反方向移动。

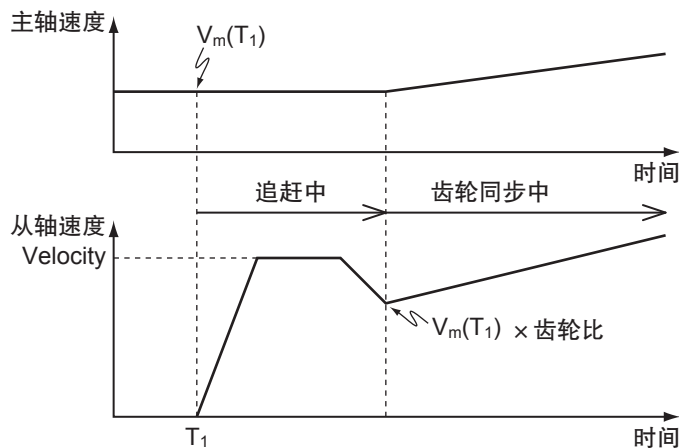


使用注意事项

Ver.1.09以下的CPU单元使用本指令时，请勿对Master(主轴)执行MC_SetPosition(当前位置变更)指令。对Master(主轴)执行MC_SetPosition(当前位置变更)指令时，可能出现Slave(从轴)急速跟踪的情况，非常危险。
要对Master(主轴)使用MC_SetPosition(当前位置变更)指令时，请解除Master(主轴)和Slave(从轴)的关系后执行该指令。

- MC功能模块通过以下3种速度，对利用Acceleration(加速度)和Deceleration(减速度)直线加减速的速度曲线进行计算。
 - 以启动指令时的Slave(从轴)的速度为初速度
 - 以启动指令时的Master(主轴)的速度乘以齿轮比得到的速度为终速度
 - 以Velocity(目标速度)为目标速度

追赶中的移动量较短时，可能达不到目标速度。



在MasterSyncPosition(主轴同步位置)、SlaveSyncPosition(从轴同步位置)，为使Slave(从轴)追上Master(主轴)，Velocity(目标速度)必须满足下列条件。

$$\text{Velocity} > \frac{\text{MC_GearInPos启动时的主轴速度} \times \text{齿轮比分子}}{\text{齿轮比分母}}$$

作为主轴速度使用的信息，因位置类型ReferenceType的设定而异。

_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)追不上Master(主轴)时，会发生“位置指定齿轮动作目标速度不足(错误代码：5447 Hex)”的错误。

- 计数模式为旋转模式时，根据CPU单元的版本，存在如下差异。
 - Ver.1.10以上版本的CPU单元时
主轴计数模式为旋转模式时，可指定超出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通过MC_MoveAbsolute(绝对值定位)指令在Direction(方向选择)中指定“无方向指定”时相同。MC_MoveAbsolute(绝对值定位)指令请参阅 □ “MC_MoveAbsolute(P.3-47)”。
 - 同样，从轴计数模式为旋转模式时，可指定超出环计数器上下限设定值范围的SlaveSyncPosition(从轴同步位置)。
 - Ver.1.09以下版本的CPU单元时
计数模式为旋转模式时，请确保在环计数器1周以内开始同步动作。



使用注意事项

有关对于主轴的注意事项，请参阅 □ “同步控制的主轴及辅轴的注意事项(P.1-6)”。

● ReferenceType(位置类型选择)

可从下列选项中选择位置的类型。

- _mcCommand：指令位置(最近任务周期下的计算值)
对当前周期，使用之前任务周期时计算的主轴指令位置。
在计算从轴指令位置之前的固定周期任务中，使用计算的主轴指令位置。
- _mcFeedback：同一任务周期下的取值
使用同一任务周期时获取的主轴反馈位置。
- _mcLatestCommand：指令位置(同一任务周期下的计算值)
使用同一任务周期时计算的主轴指令位置。
也可通过_mcCommand使用最新信息。但是，设定时必须保证主轴编号比从轴编号小。
否则，本指令的Error(错误)将变为TRUE。ErrorID(错误代码)输出“主轴/从轴 轴号非升序(错误代码：5438 Hex)”。



使用注意事项

这里的任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先级5)的周期。同样，固定周期任务表示原始恒定周期任务或固定周期任务(执行优先级5)。



参考

相比选择之前任务周期时的计算值，选择同一任务周期时计算的指令位置时，同步精度更高。但是必须使用主轴、从轴，确保“Master(主轴)设定的运动控制系统变量的轴号 < Slave(从轴)设定的运动控制系统变量的轴号”。

● 轴种类与位置类型的关系

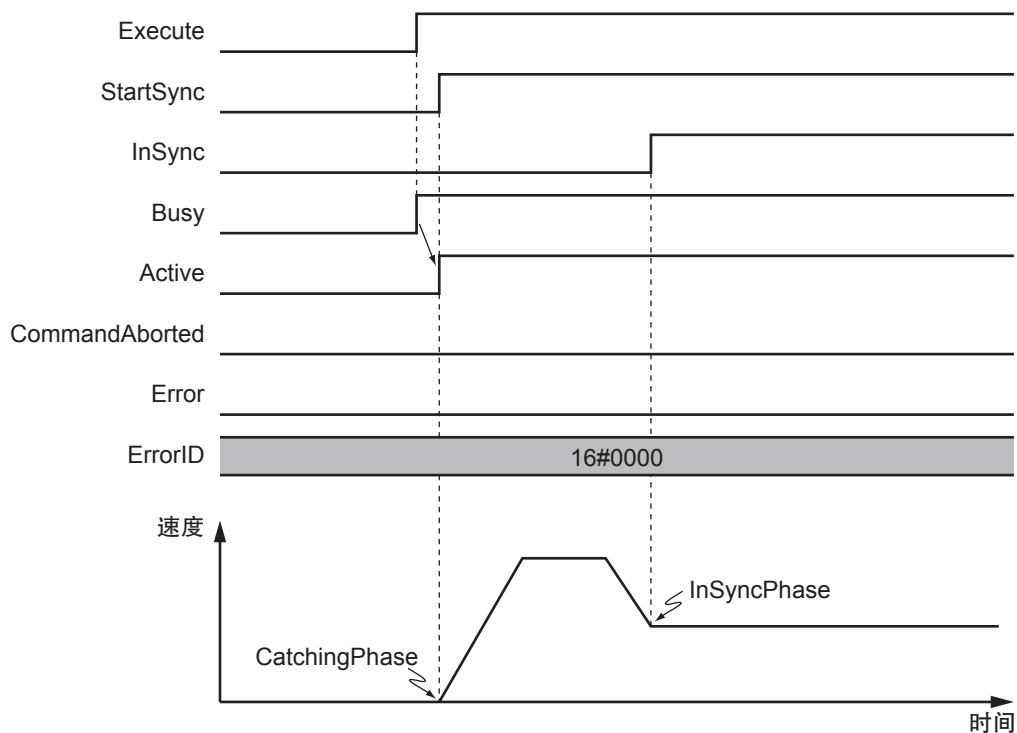
可监控的轴种类和要监控的位置类型的关系如下所示。

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	○

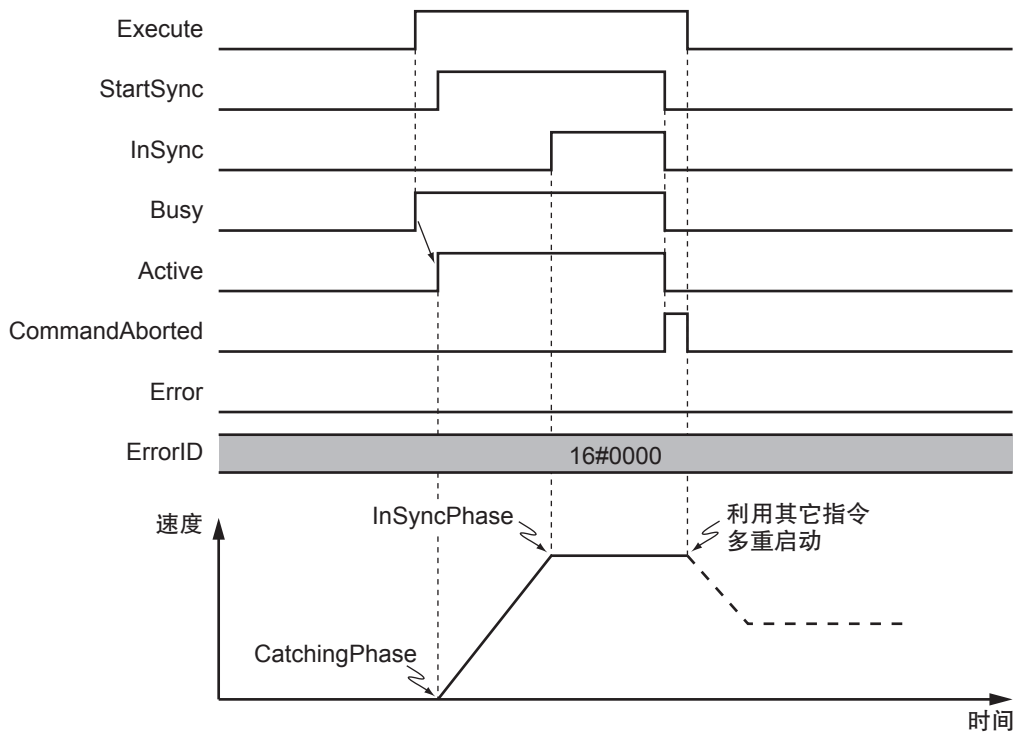
*1. 启动指令时，发生“超过位置类型选择范围(错误代码：5430 Hex)”错误。

时序图

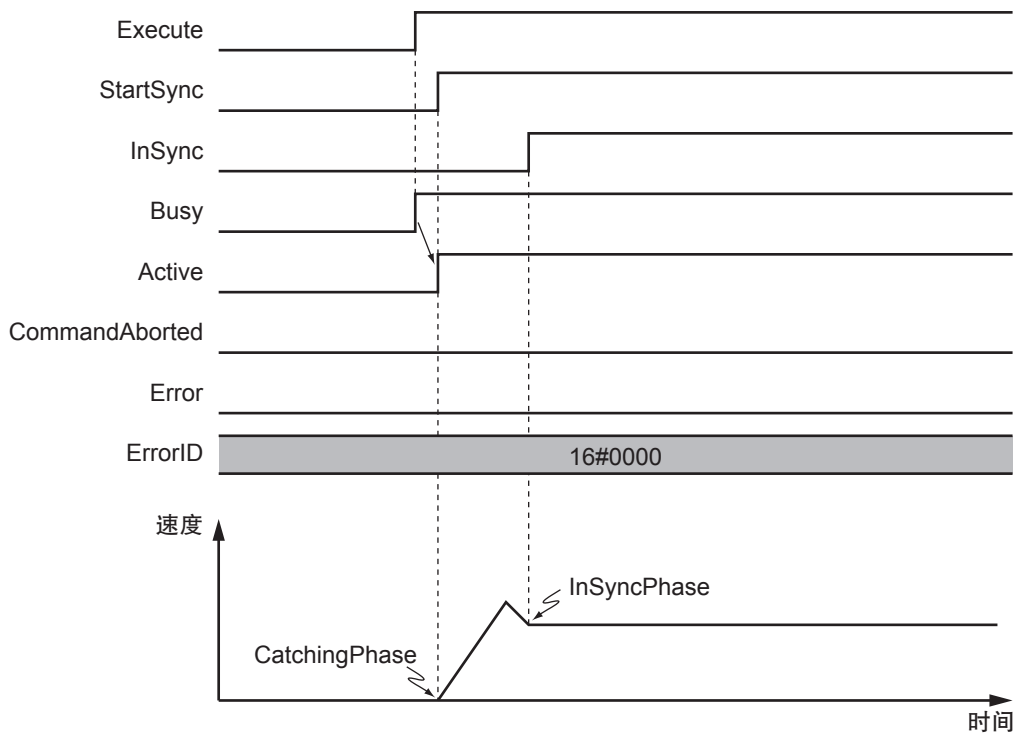
- 在Execute(启动)的上升沿，开始电子齿轮动作。
- 在启动Execute(启动)的同时，Busy(执行中)变为TRUE。开始动作后，Active(控制中)和StartSync(跟踪中)变为TRUE，Slave(从轴)开始齿轮动作。
- 到达MasterSyncPosition(主轴同步位置)、SlaveSyncPosition(从轴同步位置)时，InSync(同步中)变为TRUE。
- 利用其它指令中断本指令时，CommandAborted(执行中断)变为TRUE，Busy(执行中)、Active(控制中)、StartSync(跟踪中)、InSync(同步中)变为FALSE。



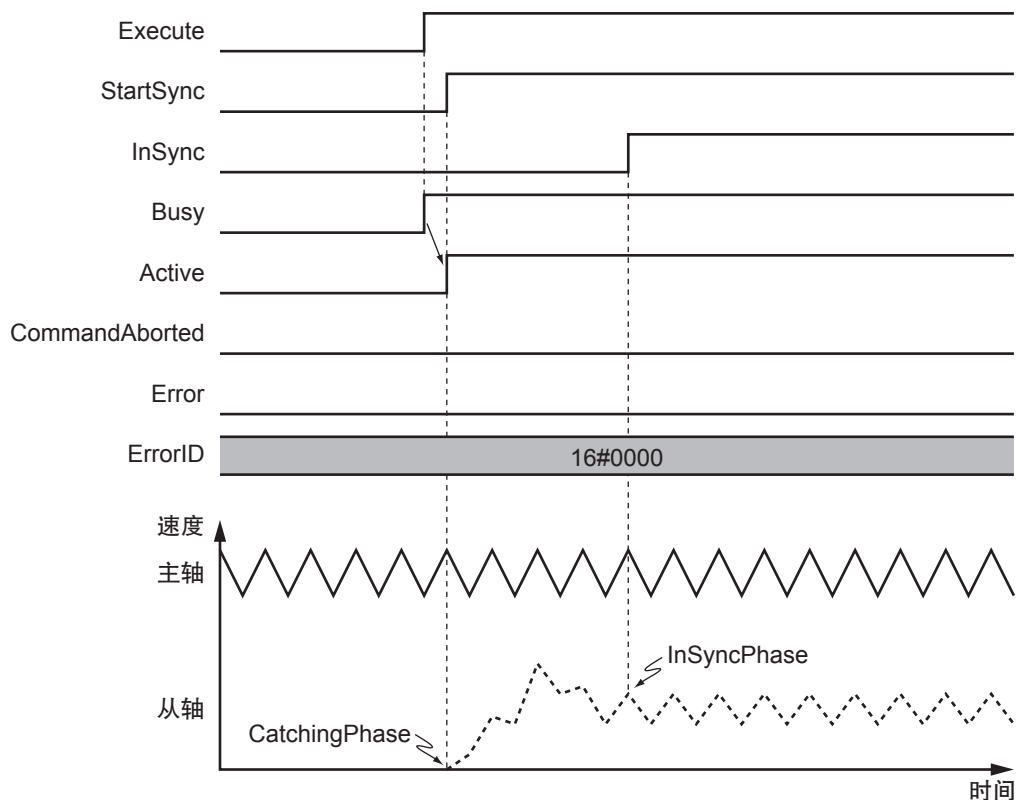
利用其它指令中断本指令的情形如下所示。



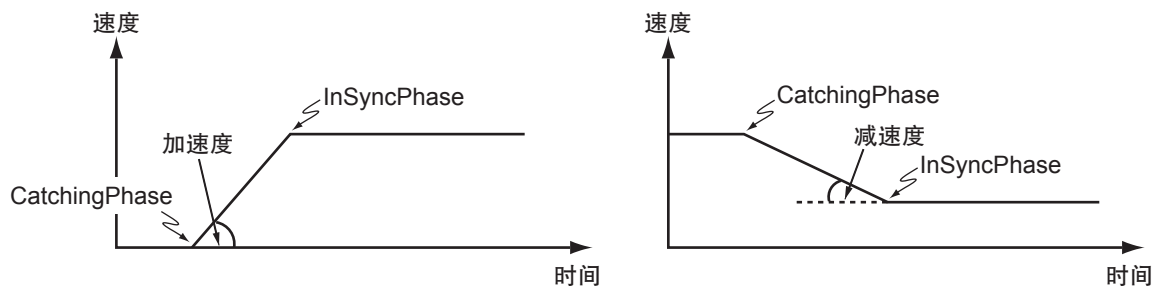
根据SlaveSyncPosition的设定不同，可能达不到目标速度。该时的示例如下所示。



此外，在达到InSync phase之前，也跟踪主轴位置。该时的示例如下所示。



可在输入变量中指定Acceleration(加速度)、Deceleration(减速度)。电子齿轮的动作示例如下所示。



重启运动指令

无法重启本指令。

重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

包含异常停止中、减速停止中的主轴轴指令状态下，不影响本指令的执行。

多重启动运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 本指令执行中的其它指令启动

多重启动时的本指令的动作由BufferMode(缓存模式选择)指定。

缓存模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 轴的动作方向因指令切换而反转时，根据轴参数中的“反转时动作”进行反转。

关于BufferMode(缓存模式选择)，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。



使用注意事项

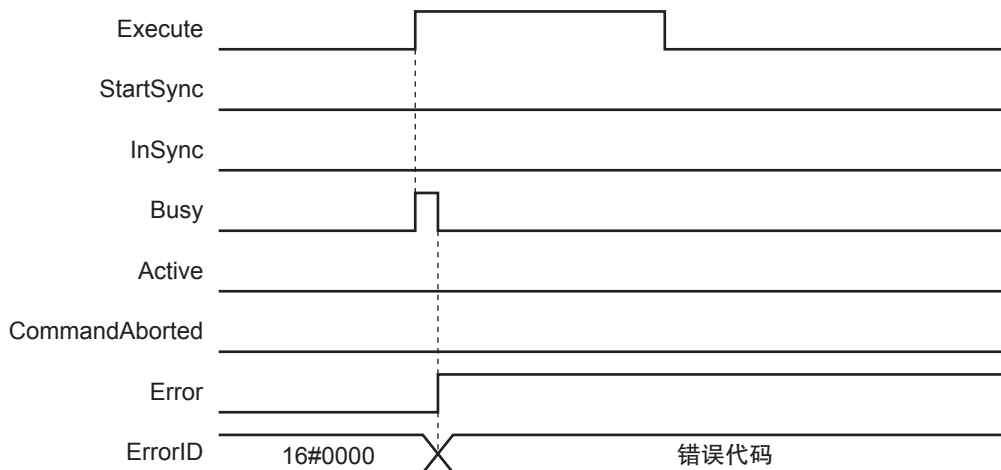
以BufferMode(缓存模式选择)的“0: 中断”以外的方式启动其它指令时，其它指令会发生异常。

异常

启动本指令发生异常时，Error(错误)变为TRUE。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



同步控制中，不受主轴的异常状态的影响。主轴的异常状态解除，正常动作时，从轴继续进行电子齿轮动作。

启动本指令时以及执行本指令中，从轴发生异常时，不会对主轴产生影响。

● 错误代码

关于指令发生的异常，请参阅 □□ “A-1 错误代码一览(P.A-2)”。

示例程序

下面，对以主轴同步位置为Pos1、从轴同步位置为Pos2进行动作的示例程序进行说明。

参数设定

该示例程序中使用的最少的必要设定如下所示。

● 轴参数的设定

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

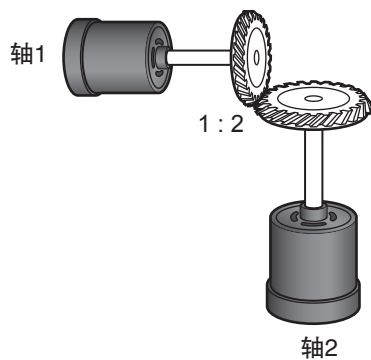
环计数器

轴	上限值	下限值
轴1	360	0
轴2	360	0

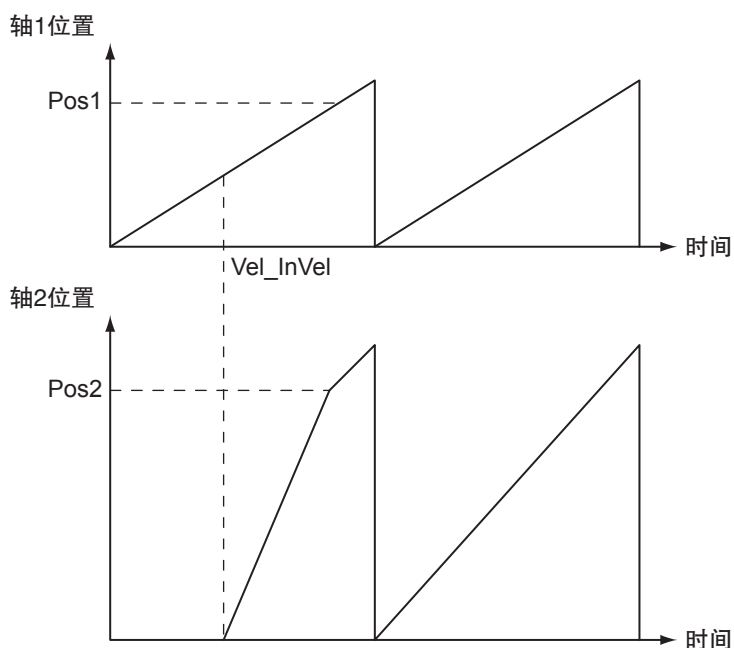
坐标单位选择

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)是实际的伺服轴，控制速度进行动作。

2 主轴达到目标速度

主轴的指令速度达到目标速度时，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，相对主轴的反馈位置，从轴(轴2)以齿轮比1:2进行齿轮动作。

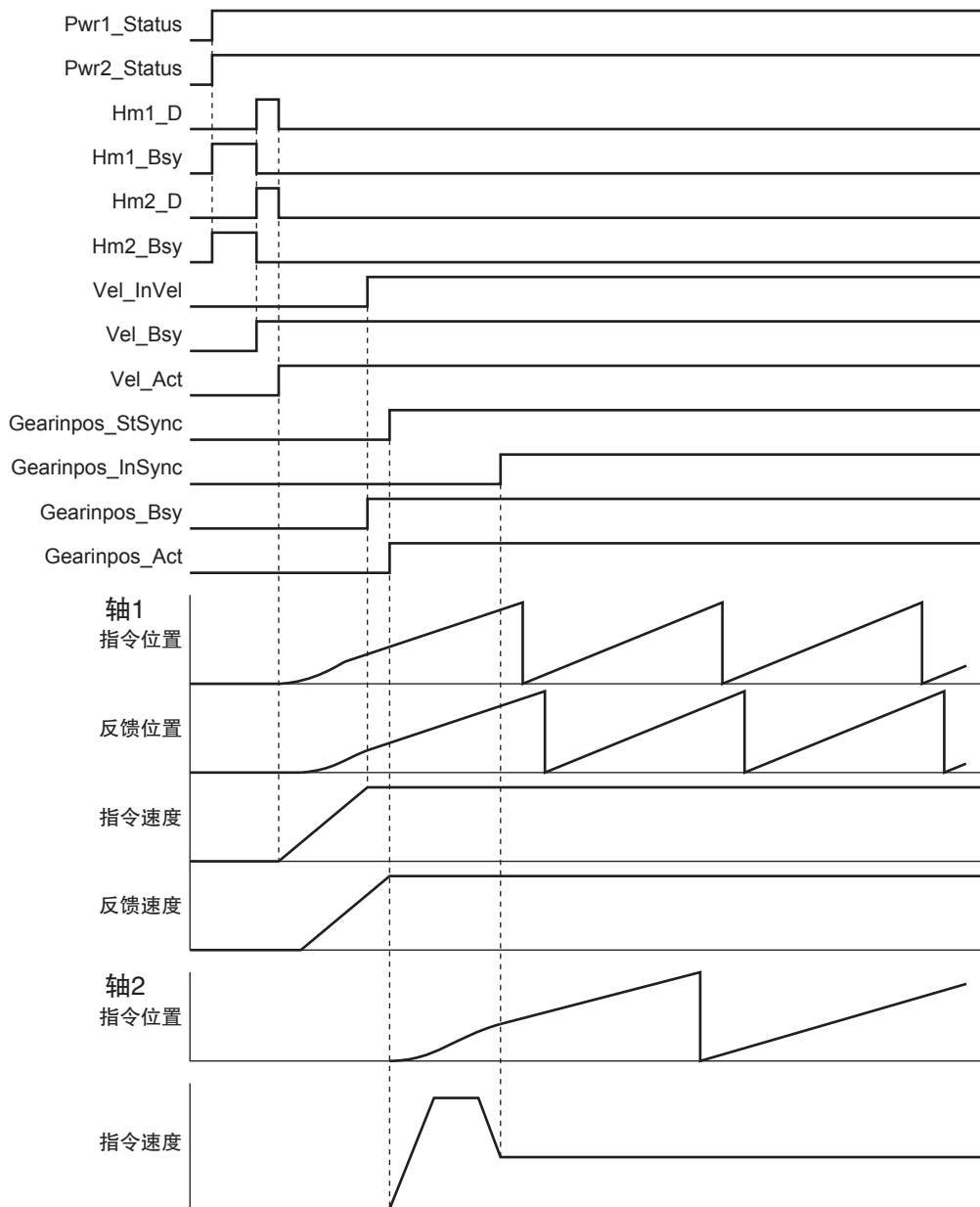
同步位置主轴为Pos1、从轴为Pos2。

梯形图

● 主要变量

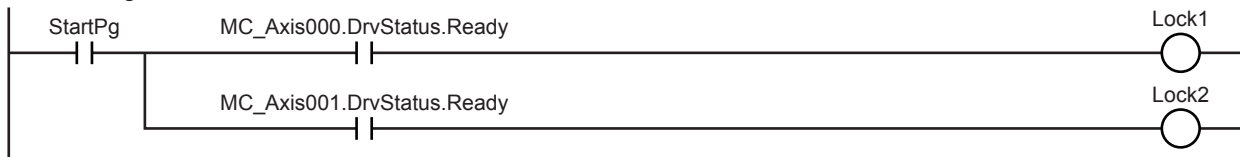
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	是主轴轴1的轴变量。
MC_Axis000.Details.Homed	BOOL	FALSE	轴1为原点确定状态时变为TRUE。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴1发生轻度故障等级的异常时变为TRUE。
MC_Axis001	_sAXIS_REF	-	是从轴轴2的轴变量。
MC_Axis001.Details.Homed	BOOL	FALSE	轴2为原点确定状态时变为TRUE。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴2发生轻度故障等级的异常时变为TRUE。
Pwr1_Status	BOOL	FALSE	分配给MC_Power的实例PWR1的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
Pwr2_Status	BOOL	FALSE	分配给MC_Power的实例PWR2的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
Hm1_D	BOOL	FALSE	分配给MC_Home的实例HM1的输出Done的变量。
Hm2_D	BOOL	FALSE	分配给MC_Home的实例HM2的输出Done的变量。
Vel_InVel	BOOL	FALSE	分配给MC_MoveVelocity的实例VEL的输出InVelocity的变量。达到目标速度时，该变量变为TRUE。
Pos1	LREAL	-	是表示主轴同步位置的变量。
Pos2	LREAL	-	是表示从轴同步位置的变量。
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则进入伺服ON状态。

● 时序图

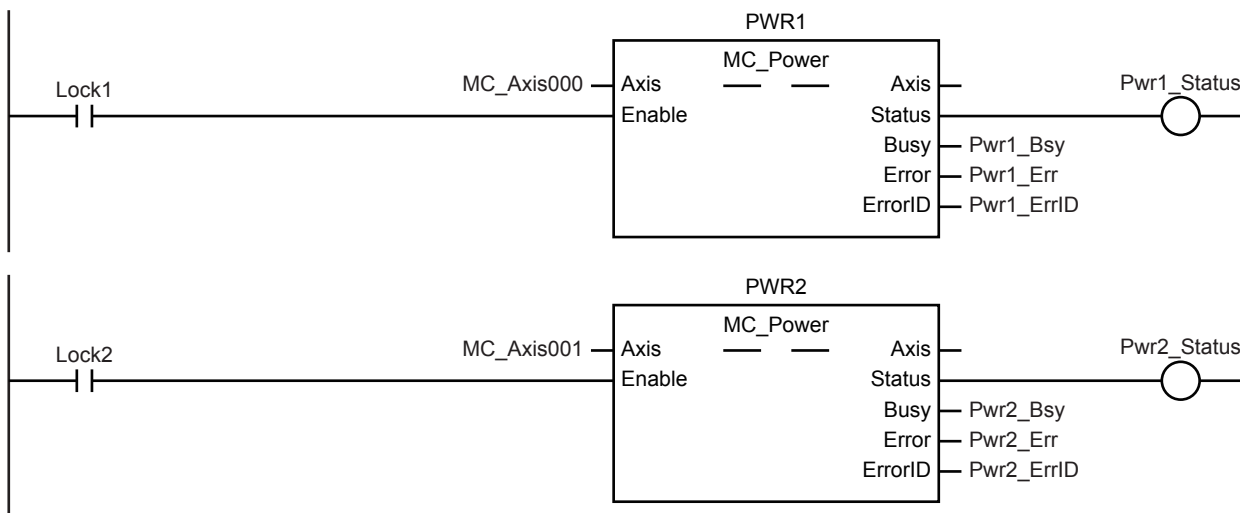


● 示例程序

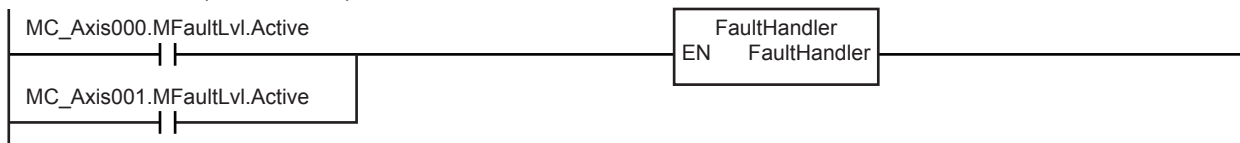
触点StartPg变为TRUE时，确认各轴的伺服驱动器处于伺服准备就绪状态



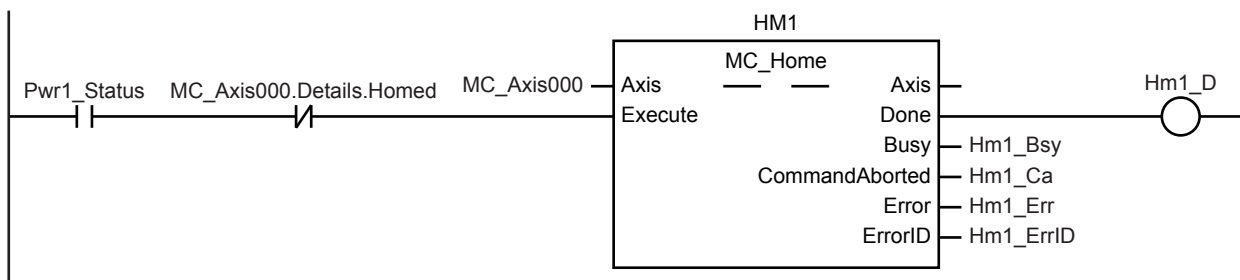
如果处于伺服准备就绪状态，则将各轴设为伺服ON状态



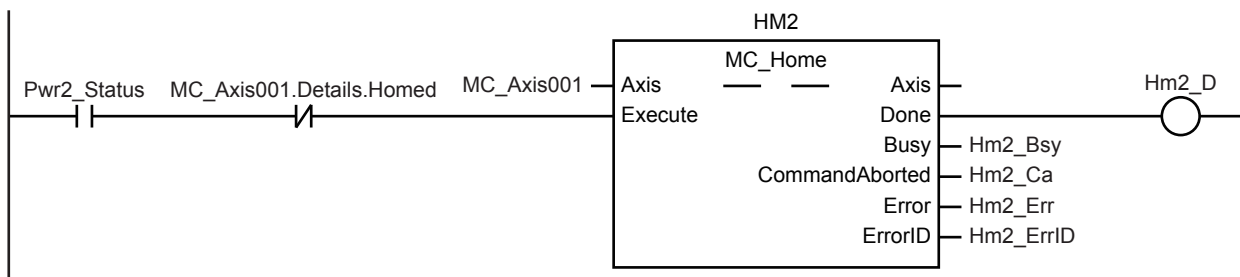
轴构成发生轻度故障程度的异常时，执行设备的异常处理(FaultHandler)。发生异常时的处理(FaultHandler)由客户根据装置进行编程



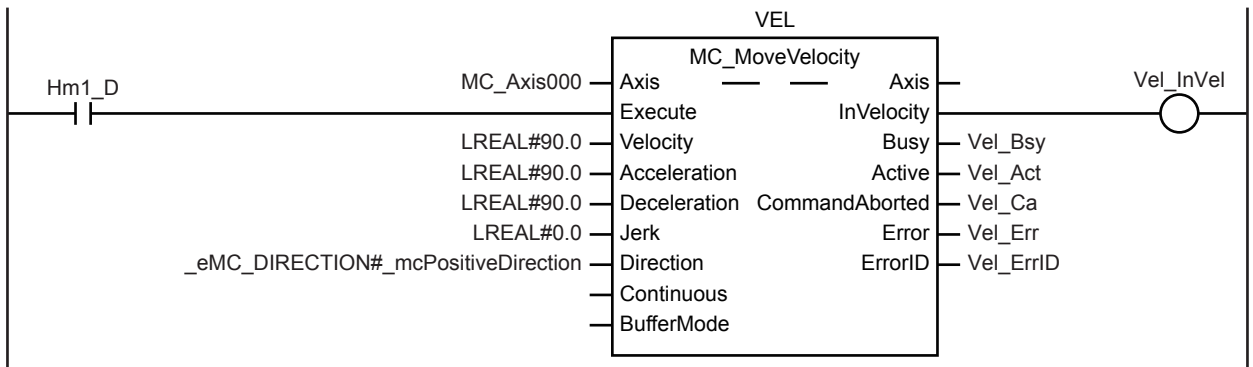
主轴(轴1)变为伺服ON状态，若原点未确定，则执行原点复位，确定原点



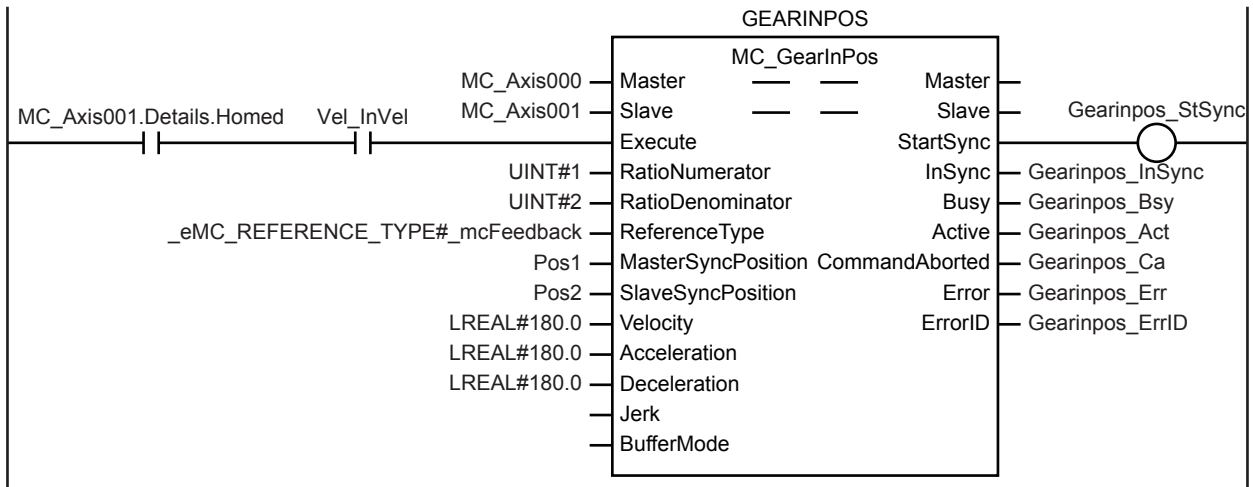
从轴(轴2)变为伺服ON状态，若原点未确定，则执行原点复位，确定原点



主轴(轴1)原点复位完成后, 启动MC_MoveVelocity(速度控制)指令



从轴(轴2)原点复位完成后, MC_MoveVelocity的Vel_InVel为TRUE时, 启动MC_GearInPos(位置指定齿轮动作), 开始齿轮动作



MC_GearInPos

3

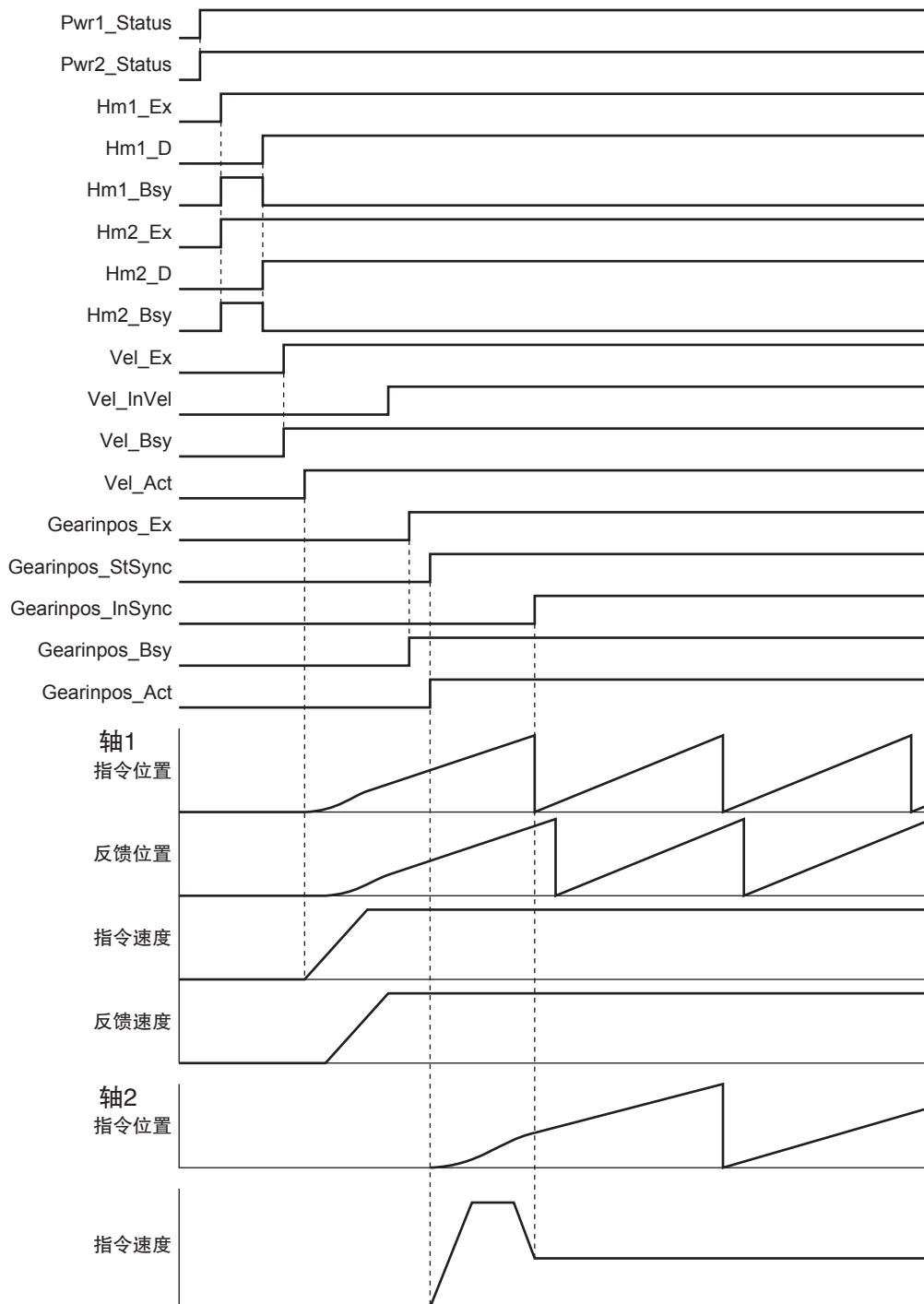
示例程序

结构文本(ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	是主轴轴1的轴变量。
MC_Axis000.Details.Homed	BOOL	FALSE	轴1为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	-	是从轴轴2的轴变量。
MC_Axis001.Details.Homed	BOOL	FALSE	轴2为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	分配给MC_Power的实例PWR1的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
Pwr2_Status	BOOL	FALSE	分配给MC_Power的实例PWR2的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
Hm1_D	BOOL	FALSE	分配给MC_Home的实例HM1的输出Done的变量。
Hm2_D	BOOL	FALSE	分配给MC_Home的实例HM2的输出Done的变量。
Vel_InVel	BOOL	FALSE	分配给MC_MoveVelocity的实例VEL的输出InVelocity的变量。达到目标速度时，该变量变为TRUE。
Pos1	LREAL	-	是表示主轴同步位置的变量。
Pos2	LREAL	-	是表示从轴同步位置的变量。
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则进入伺服ON状态。
Gearinpos_Ex	BOOL	FALSE	该变量由FALSE→TRUE时，启动MC_GearInPos的实例GEARIN1。
Vel_Ex	BOOL	FALSE	该变量由FALSE→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;

//确认StartPg为TRUE，且伺服驱动器处于伺服准备就绪状态，
//则将轴1设为伺服ON状态。
//如果未处于伺服准备就绪状态，则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
  Pwr1_En:=TRUE;
ELSE
  Pwr1_En:=FALSE;
END_IF;

//确认StartPg为TRUE，且伺服驱动器处于伺服准备就绪状态，
//则将轴2设为伺服ON状态。
//如果未处于伺服准备就绪状态，则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
  Pwr2_En:=TRUE;
ELSE
  Pwr2_En:=FALSE;
END_IF;

//轴1~轴2发生轻度故障后，执行异常时处理FaultHandler。
//发生异常时的处理(FaultHandler)由客户根据装置进行编程。
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
  FaultHandler();
END_IF;

// 轴1处于可运行状态，且原点未确定时，执行原点复位
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
  Hm1_Ex:=TRUE;
END_IF;

// 轴2处于可运行状态，且原点未确定时，执行原点复位

```

```

IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
  Hm2_Ex:=TRUE;
END_IF;
// 轴1原点复位完成后, 启动MC_MoveVelocity
IF Hm1_D=TRUE THEN
  Vel_Ex := TRUE;
END_IF;

// 轴2原点复位完成后, MC_MoveVelocity的Vel_InVel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

中止执行中的MC_GearIn(齿轮动作开始)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	在上升沿开始指令。
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. 关于指令单位，请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*2. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

* 请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 执行MC_Stop指令中，启动本指令时 	<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含有异常因素时	异常已解除时

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}

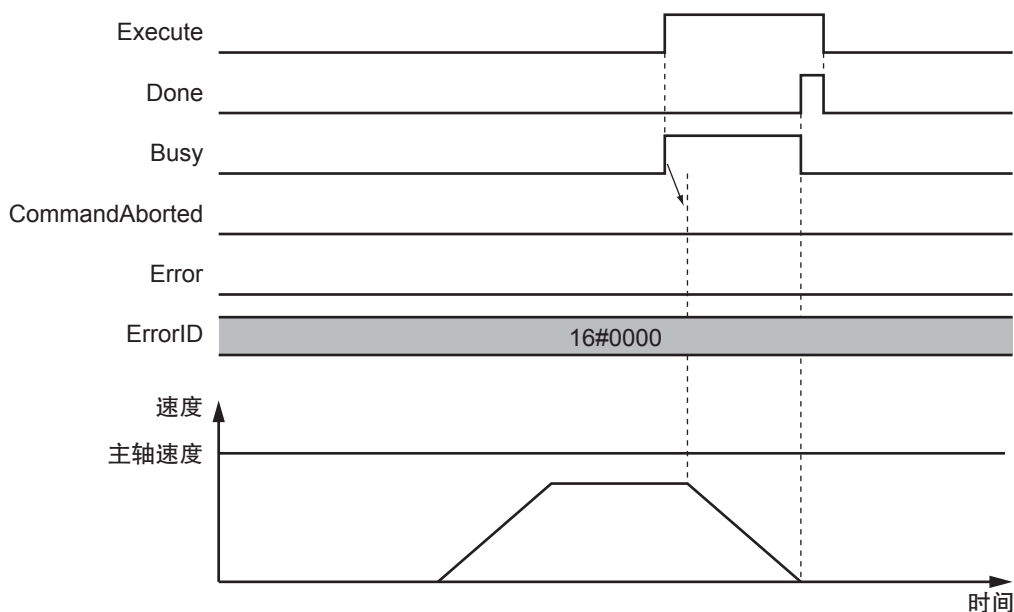
*1. 请在使用在Sysmac Studio的轴基本设定画面中创建的用户定义变量的轴变量名称(默认“MC_Axis***”)或系统定义变量的轴变量名称(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])。

功能说明

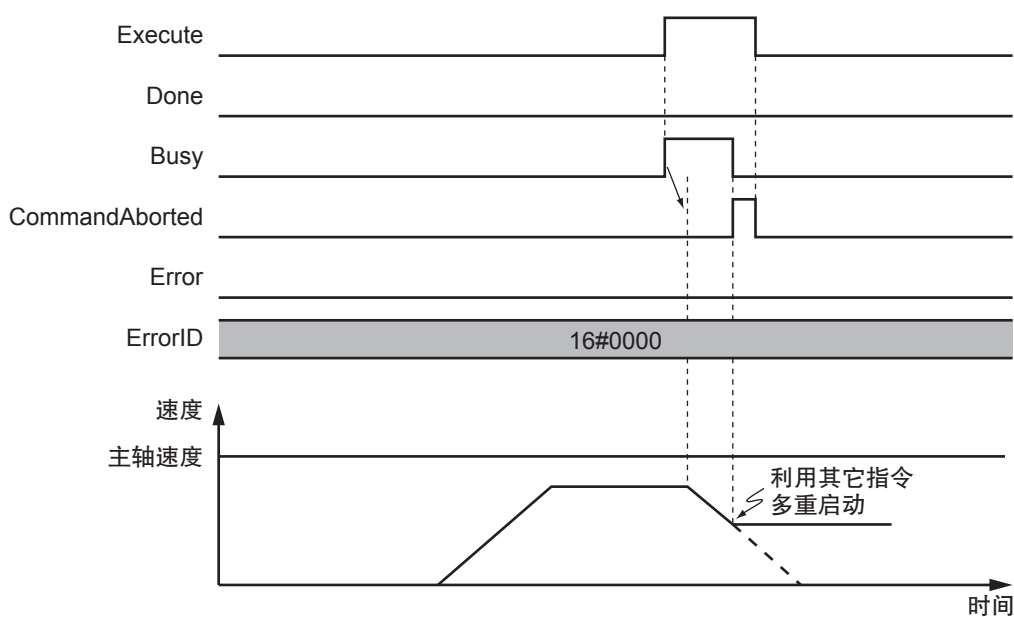
- 通过Slave(从轴)指定动作对象轴, 指定Deceleration(减速度), 中止执行中的MC_GearIn(齿轮动作开始)指令、MC_GearInPos(位置指定齿轮动作)指令。
- 本指令对MC_GearIn(齿轮动作开始)指令、MC_GearInPos(位置指定齿轮动作)指令的主轴动作没有影响。

时序图

- 在启动Execute(启动)的同时, Busy(执行中)变为TRUE。
- 达到目标速度时, Done(完成)变为TRUE。
- 利用其它指令中断本指令时, CommandAborted(执行中断)变为TRUE, Busy(执行中)变为FALSE。

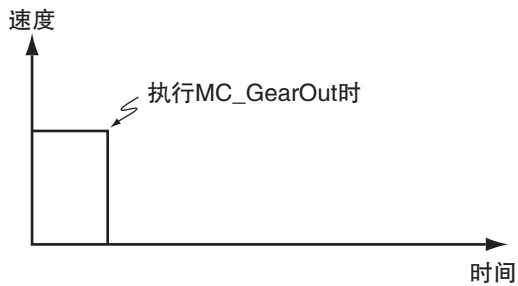


● 利用其它指令中断本指令时



- 将减速度指定为“0”启动时

将减速度指定为“0”启动时，不减速而停止。
减速度为“0”时的动作示例如下所示。



重启运动指令

无法重启本指令。
重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

多重启动运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

- 其它指令执行中的本指令启动

在执行MC_GearIn(齿轮动作开始)指令或MC_GearInPos(位置指定齿轮动作)指令中启动本指令时，中断执行(CommandAborted)MC_GearIn(齿轮动作开始)指令或MC_GearInPos(位置指定齿轮动作)指令，本指令变为Busy(执行中)。

在执行MC_GearIn(齿轮动作开始)指令或MC_GearInPos(位置指定齿轮动作)指令以外的指令中启动本指令时，本指令变为Error(错误)。

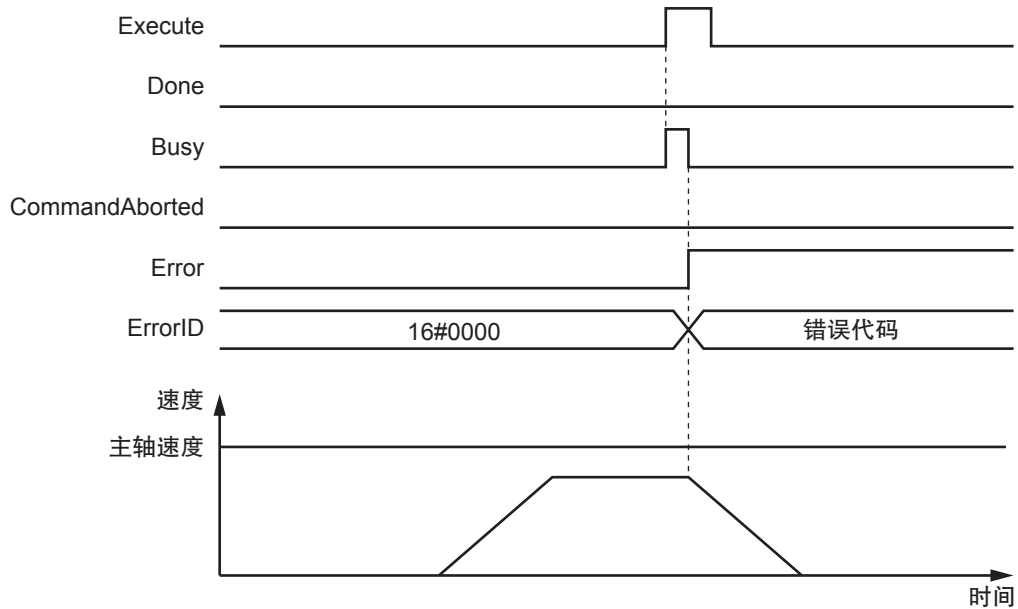
- 本指令执行中的其它指令启动

对本指令执行运动指令多重启动时，请指定从轴。
对本指令进行多重启动运动指令时，可选择中断、等待。

异常

启动本指令发生异常时，Error(错误)变为TRUE。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 [□](#) “A-1 错误代码一览(P.A-2)”。

MC_MoveLink

与指定的主轴同步，执行定位。

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	在上升沿开始指令。
Reference Type *1	位置类型选择	_eMC_REFERENCE_TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 *2	指定位置类型。 0: 指令位置(最近任务周期*3下的计算值) 1: 反馈位置(同一任务周期*3下的取值) 2: 指令位置(同一任务周期*3下的计算值)
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	以无符号的绝对值形式指定主轴的移动距离。主轴的移动方向正负均有效。 单位为[指令单位]。*4
Master DistanceInACC	主轴加速移动距离	LREAL	正数或“0”	0	以无符号的绝对值形式指定从轴加速时主轴的移动距离。主轴的移动方向正负均有效。 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
MasterDistanceInDEC	主轴减速移动距离	LREAL	正数或“0”	0	以无符号的绝对值形式指定从轴减速时主轴的移动距离。主轴的移动方向正负均有效。 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: 主轴到达主轴跟踪距离时
MasterStartDistance	主轴跟踪距离	LREAL	负数、正数、“0”	0	指定从轴开始跟踪主轴的绝对坐标的主轴位置。 单位为[指令单位]。 ^{*3}
BufferMode	缓存模式选择	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0 ^{*2}	指定多重启动运动指令时的动作。 0: 中断 1: 等待

*1. 使用 _mcLatestCommand 时，必须在满足以下关系的条件下使用主轴、从轴。

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


*2. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

*3. 任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先度5)的周期。

*4. 关于指令单位，请参阅  “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或  “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 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	指令执行完毕时变为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	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常

*  请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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	接收指令时	<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"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 执行MC_Stop指令中，启动本指令时 	<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含有异常因素时	异常已解除时

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	在触发条件下指定控制器模式时，指定触发的输入变量。

*1. 请使用在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型的用户定义变量。

● _sTRIGGER_REF

结构要素变量	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	在驱动器模式下，指定使用2个锁定功能中的哪一个。 0: 锁定功能1 1: 锁定功能2
InputDrive	驱动触发输入信号	_eMC_TRIGGER_INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	在驱动器模式下指定伺服驱动器的触发信号。 0: Z相 1: 外部输入

功能说明

- 从轴与指定主轴同步动作。
- 属于电子凸轮的一种，从轴相对于主轴以梯形曲线方式进行动作。
- 要停止通过本指令处于动作中的轴，请使用MC_Stop(强制停止)指令。



使用注意事项

Ver.1.09以下的CPU单元使用本指令时，请勿对Master(主轴)执行MC_SetPosition(当前位置变更)指令。

对Master(主轴)执行MC_SetPosition(当前位置变更)指令时，可能出现Slave(从轴)急速跟踪的情况，非常危险。

要对Master(主轴)使用MC_SetPosition(当前位置变更)指令时，请解除Master(主轴)和Slave(从轴)的关系后执行该指令。

有关对于主轴的注意事项，请参阅 □ “同步控制的主轴及辅轴的注意事项(P.1-6)”。

● 数据对象的映射

LinkOption(同步开始条件选择)选择[_mcTriggerDetection]，Mode选择[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。

请在Sysmac Studio的轴基本设置的[高级设置]中进行映射。

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

未进行这些设定时，将发生过程数据对象设定不足(错误代码：3461Hex)错误。

关于数据对象的映射，请参阅 □ “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-363)”或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。使用NX系列位置接口单元时，请参阅 □ “NX系列 位置接口单元 用户手册(SBCE-374)”中的“I/O入口映射”。

指令详情

下面对指令详细说明。

● Master(主轴)指定

通过Master(主轴)指定主轴。

● Slave(从轴)指定

通过Slave(从轴)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。

关于输入信号的选择、触发发生时间的详情，请参阅 □ “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。指定控制器模式时，将TriggerVariable(触发变量)用作触发信号。

● ReferenceType(位置类型选择)

可从下列选项中选择位置的类型。

- **_mcCommand**: 指令位置(最近任务周期下的计算值)
对当前周期, 使用之前任务周期时计算的主轴指令位置。
在计算从轴指令位置之前的固定周期任务中, 使用计算的主轴指令位置。
- **_mcFeedback**: 同一任务周期下的取值
使用同一任务周期时获取的主轴反馈位置。
- **_mcLatestCommand**: 指令位置(同一任务周期下的计算值)
使用同一任务周期时计算的主轴指令位置。
也可通过**_mcCommand**使用最新信息。但是, 设定时必须保证主轴编号比从轴编号小。
否则, 本指令的Error(错误)将变为TRUE。ErrorID(错误代码)输出“主轴/从轴 轴号非升序(错误代码: 5438 Hex)”。



使用注意事项

这里的任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先度5)的周期。同样, 固定周期任务表示原始恒定周期任务或固定周期任务(执行优先度5)。



参考

相比选择之前任务周期时的计算值, 选择同一任务周期时计算的指令位置时, 同步精度更高。但是必须使用主轴、从轴, 确保“Master(主轴)设定的运动控制系统变量的轴号 < Slave(从轴)设定的运动控制系统变量的轴号”。

● 轴种类与位置类型的关系

可监控的轴种类和要监控的位置类型的关系如下所示。

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	○

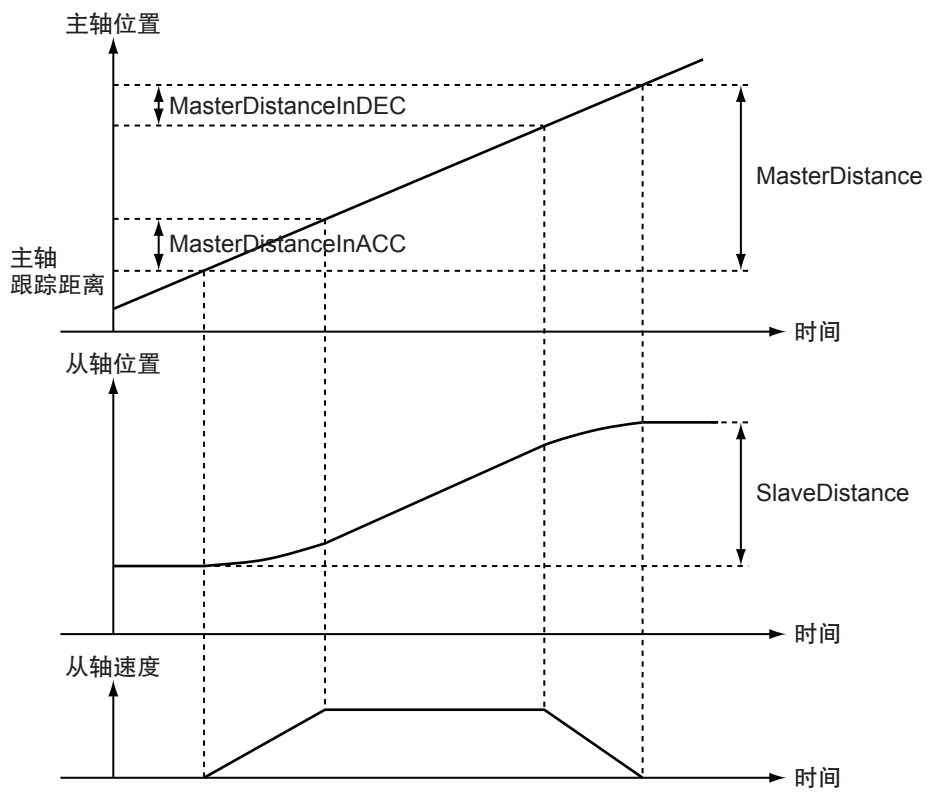
*1. 启动指令时, 发生“超过位置类型选择范围(错误代码: 5430 Hex)”错误。

● 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)

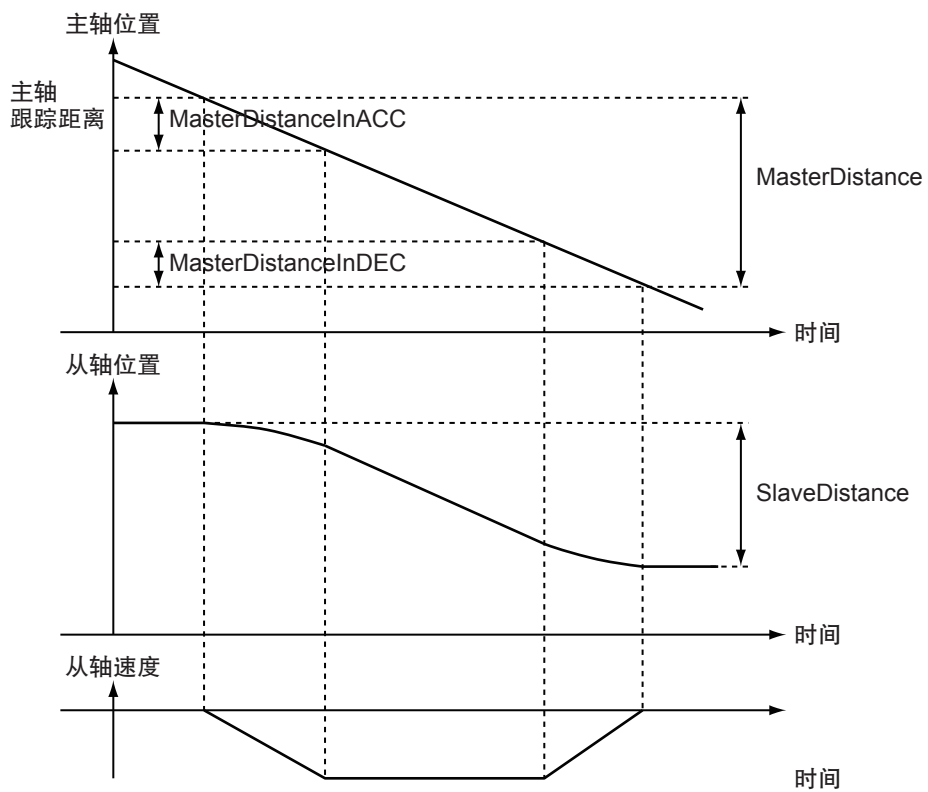
如下图所示, 从轴的速度和位置由主轴与从轴的移动量之比决定。

下图的主轴跟踪距离表示同步开始条件有效时的位置。

主轴正方向移动时



主轴负方向移动时



主轴与从轴移动量之间的关系如下表所示。

区间		主轴与从轴移动量之间的关系	
加速	主轴	主轴加速移动距离	
	从轴	从轴移动距离 × $\frac{\text{主轴加速移动距离}}{2} + (\text{主轴移动距离} - \text{主轴加速移动距离} - \text{主轴减速移动距离}) + \frac{\text{主轴减速移动距离}}{2}$	
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	从轴移动距离 × $\frac{\text{主轴减速移动距离}}{2} + (\text{主轴移动距离} - \text{主轴加速移动距离} - \text{主轴减速移动距离}) + \frac{\text{主轴加速移动距离}}{2}$	

主轴的等速区间为负数时，发生等速移动量异常而停止动作。

此外，从轴以主轴相同速度传送时，请将从轴的移动量指定为下列值。

$$\text{从轴移动距离} = \frac{\text{主轴加速移动距离}}{2} + (\text{主轴移动距离} - \text{主轴加速移动距离} - \text{主轴减速移动距离}) + \frac{\text{主轴减速移动距离}}{2}$$



使用注意事项

主轴的计数器模式为旋转模式时，请将MasterDistance(主轴移动距离)指定为主轴的环计数器1周以内的数值。

● 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
启动本指令时，从下一个周期起，从轴与主轴同步定位。
- 检测到触发时
从输入触发指定中指定的输入信号有效的下一个周期起，从轴与主轴同步定位。
- 主轴到达主轴跟踪距离时
在本指令启动过程中，主轴到达主轴跟踪距离时，从下一个周期起，从轴同步定位。
此外，主轴在主轴跟踪距离处停止的状态下，启动本指令时，从轴也是从下一个周期起同步定位。



使用注意事项

LinkOption(同步开始条件选择)选择[_mcTriggerDetection]、Mode选择[驱动器模式]时，必须映射对象数据。

请设定下列对象。

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

未进行这些设定时，将发生过程数据对象设定不足(错误代码：3461Hex)错误。关于对象数据的映射，请参阅 □□ “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-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇 (SBCE-379)”。

- **MasterStartDistance(主轴跟踪距离)**

以绝对位置指定从轴开始同步的主轴的位置。

- Ver.1.10以上版本的CPU单元时**

主轴计数模式为旋转模式时，可指定超出环计数器上下限设定值范围的MasterStartDistance(主轴跟踪距离)。此时，主轴当前位置和主轴跟踪距离的关系与通过MC_MoveAbsolute(绝对值定位)指令在Direction(方向选择)中指定“无方向指定”时相同。

MC_MoveAbsolute(绝对值定位)指令请参阅 □□ “MC_MoveAbsolute(P.3-47)”。

- Ver.1.09以下版本的CPU单元时**

主轴计数模式为旋转模式时，请在环计数器上下限设定值范围内指定MasterStartDistance(主轴跟踪距离)。如果指定超出范围，将在执行指令时输出“超过凸轮动作(主轴跟踪)起始位置设定范围(错误代码：547B Hex)”。

- **BufferMode(缓存模式选择)**

指定前一个轴动作和本次动作的连接方式。

有以下2种。

缓存模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。
等待	当前正在执行的指令正常完成后，已缓存的本指令自动启动。

关于BufferMode(缓存模式选择)，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

- **到位检查**

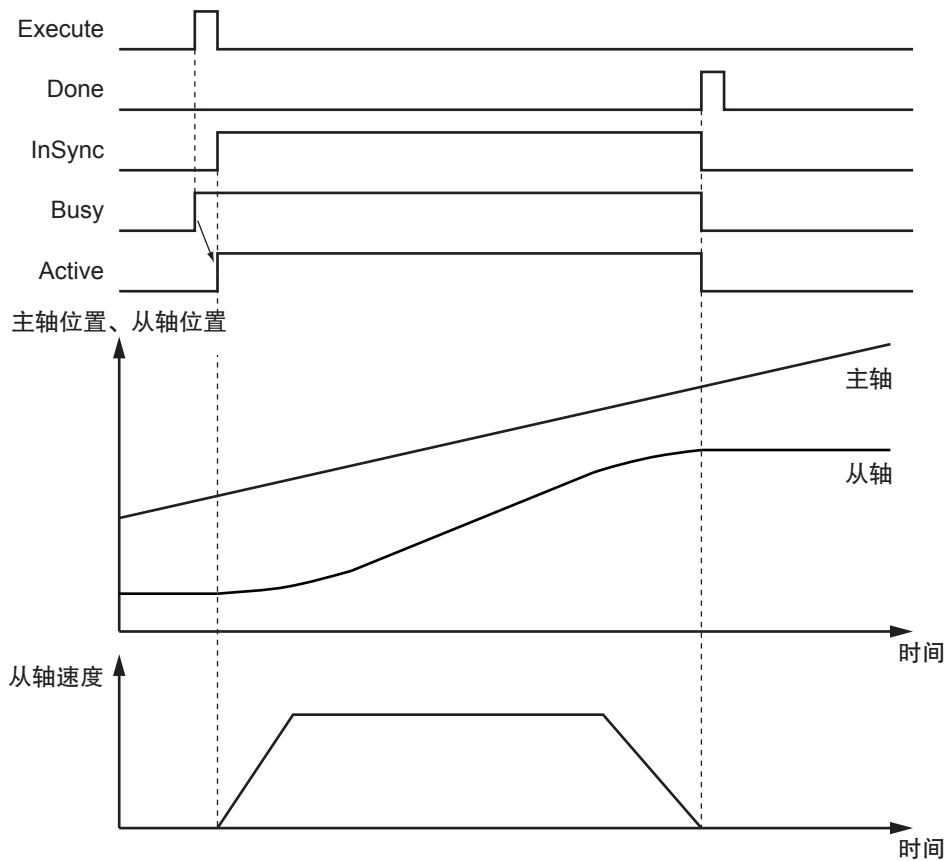
从轴动作完成时，执行到位检查。

根据轴参数的[到位宽度]、[到位检查时间]的设定，进行本指令的到位检查。

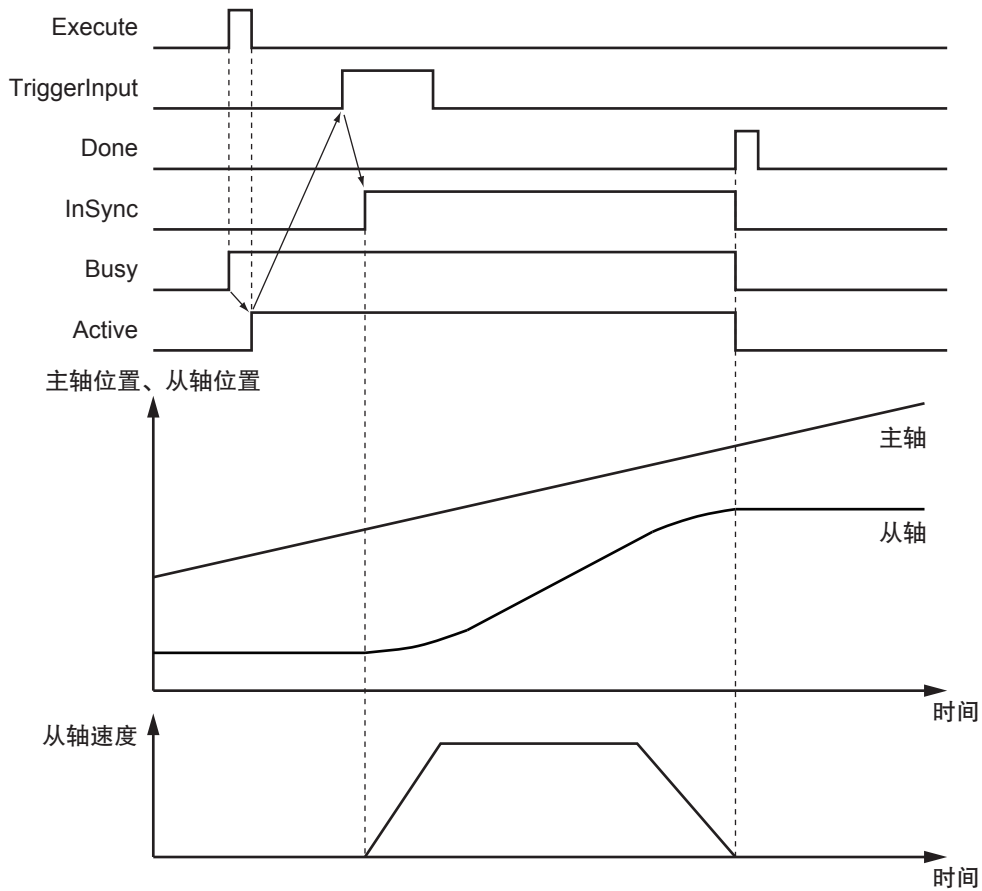
时序图

● 启动本指令时

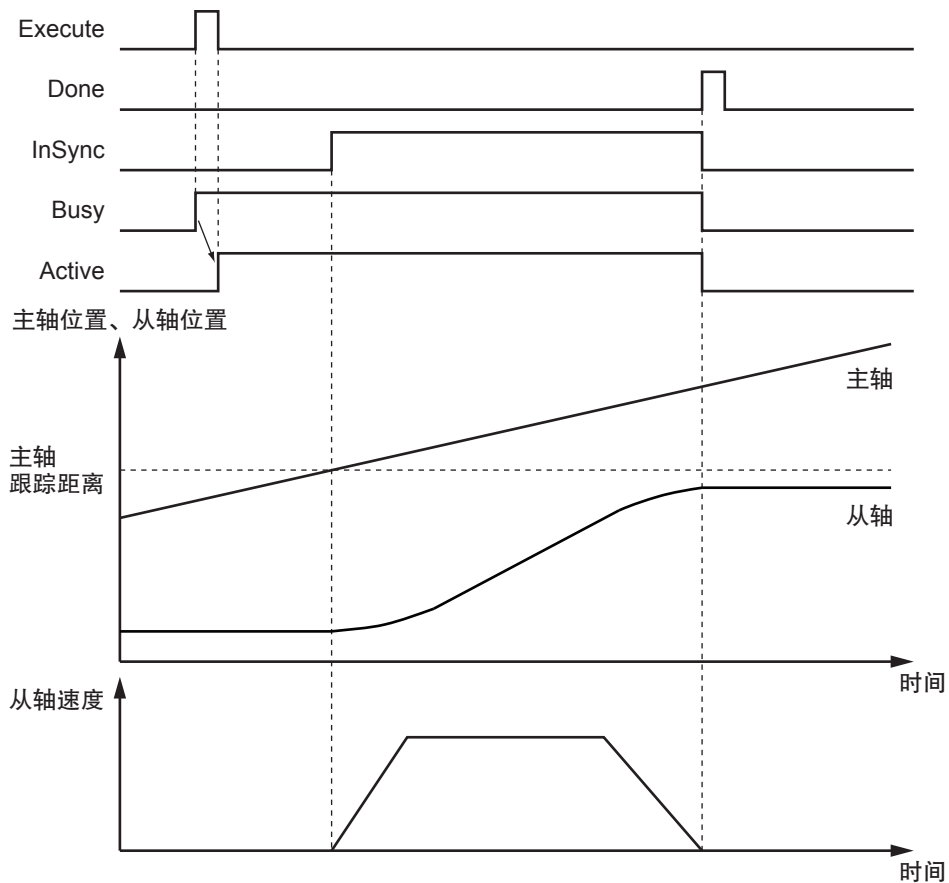
同步开始条件指定 = 指令开始时



同步开始条件指定 = 检测到触发时



同步开始条件指定 = 主轴到达主轴跟踪距离时



- 前一个动作中以中断方式启动BufferMode(缓存模式选择)时

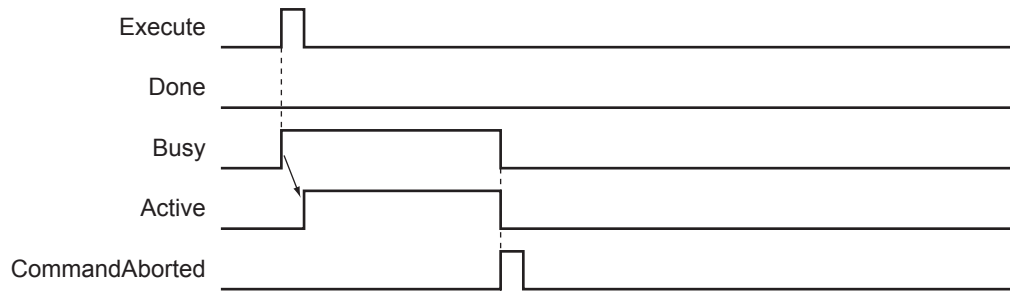
前一个动作执行MC_MoveAbsolute(绝对值定位)指令时，通过同步开始条件指定指定指令开始时，执行本指令时的情形如下所述。



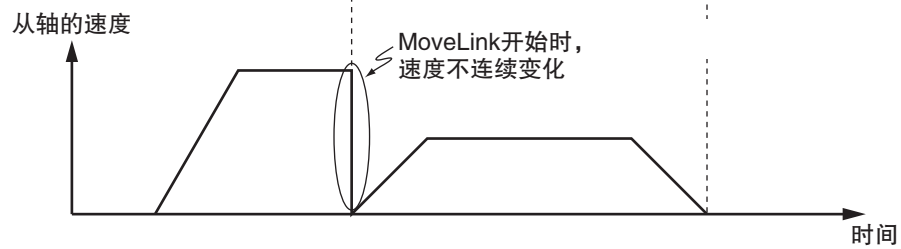
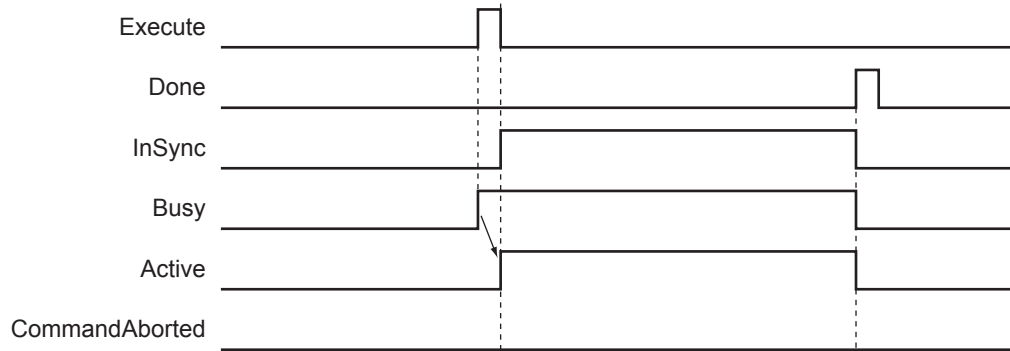
使用注意事项

如下图所示，本指令开始时从轴的速度是不连续的。敬请注意。

之前的指令为MC_MoveAbsolute(绝对值定位)时



当前的指令



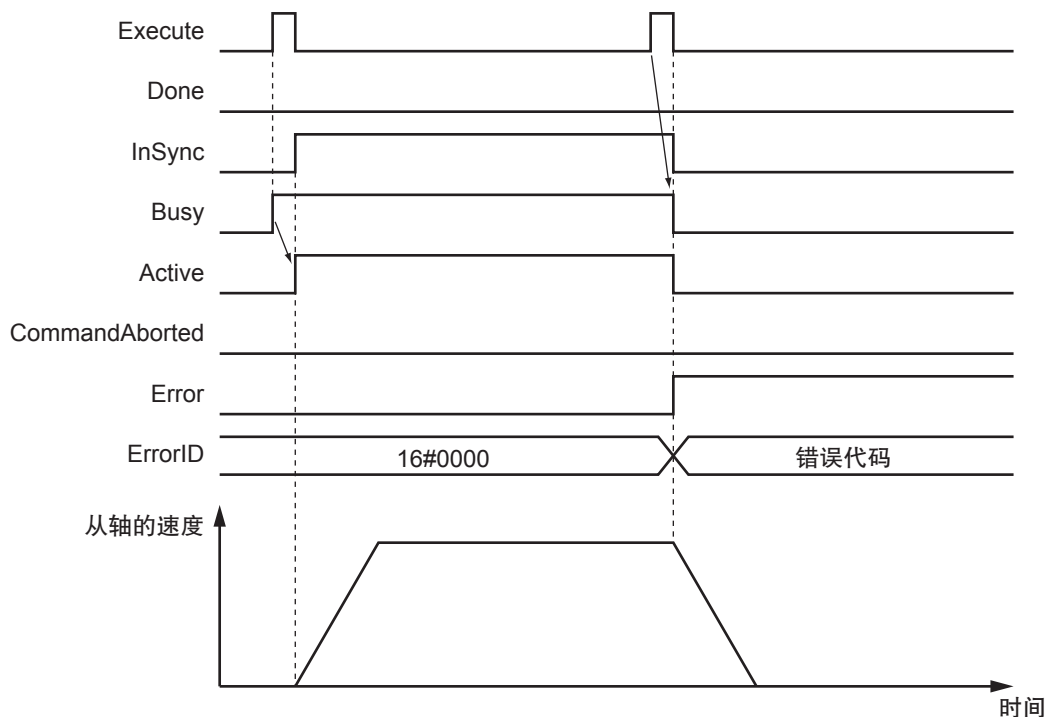
- 前一个动作中以等待方式启动BufferMode(缓存模式选择)时

前一个动作完成后执行本指令。

重启运动指令

无法重启本指令。

若执行了重启，将发生“无法重启运动指令(错误代码：543B Hex)”的错误，该轴停止动作。



多重启动运动指令

指令多重启动的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 本指令执行中的其它指令启动

对本指令进行多重启动运动指令时，可选择中断、等待。

无法实现基于合并的指令多重启动。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

关于发生异常后的时序图，请参阅 □□ “重启运动指令(P.3-277)”。

● 错误代码

关于指令发生的异常，请参阅 □□ “A-1 错误代码一览(P.A-2)”。

示例程序

下面，对切断控制的示例程序进行说明。

参数设定

该示例程序中使用的最少的必要设定如下所示。

● 轴参数的设定

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	线性模式

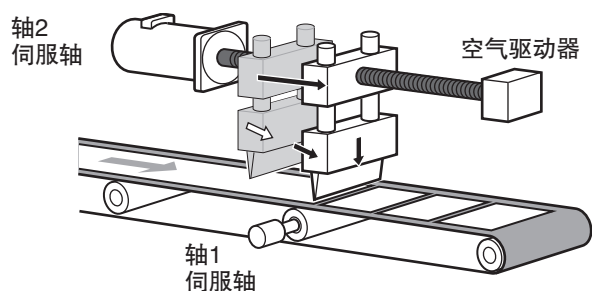
环计数器

轴	上限值	下限值
轴1	360	0

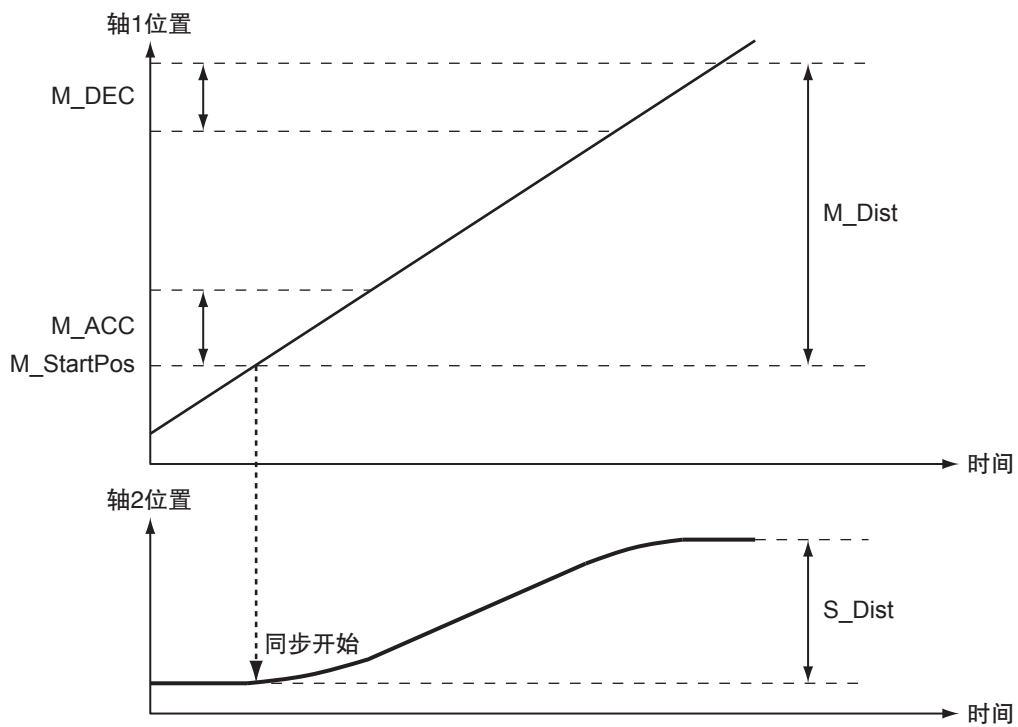
坐标单位选择

轴	坐标单位
轴1	degree
轴2	mm

动作示例



● 动作模式



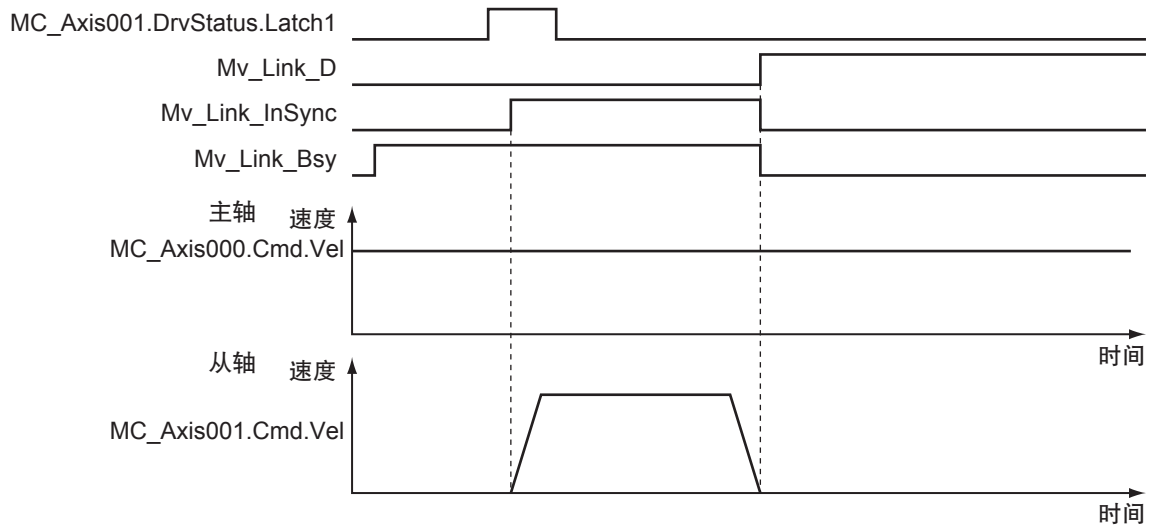
- 1 主轴的开始**
以皮带输送机上安装的轴1为主轴，反馈位置。
- 2 从轴的启动**
沿水平方向动作的滚柱丝杠上安装的轴2与轴1同步动作。
- 3 切刀切断**
轴2同步中空气驱动器变为“ON”，空气驱动器上连接的切刀垂直下降，切断工件。

梯形图

● 主要变量

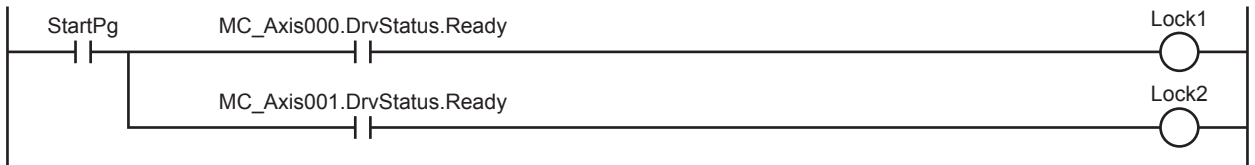
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	是主轴轴1的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴1发生轻度故障等级的异常时变为TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴1为原点确定状态时变为TRUE。
MC_Axis001	_sAXIS_REF	-	是从轴轴2的轴变量。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴2发生轻度故障等级的异常时变为TRUE。
MC_Axis001.Details.Homed	BOOL	FALSE	轴2为原点确定状态时变为TRUE。
Pwr1_Status	BOOL	FALSE	分配给MC_Power的实例PWR1的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
Pwr2_Status	BOOL	FALSE	分配给MC_Power的实例PWR2的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则进入伺服ON状态。
Actuator	BOOL	FALSE	轴1与轴2同步时，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

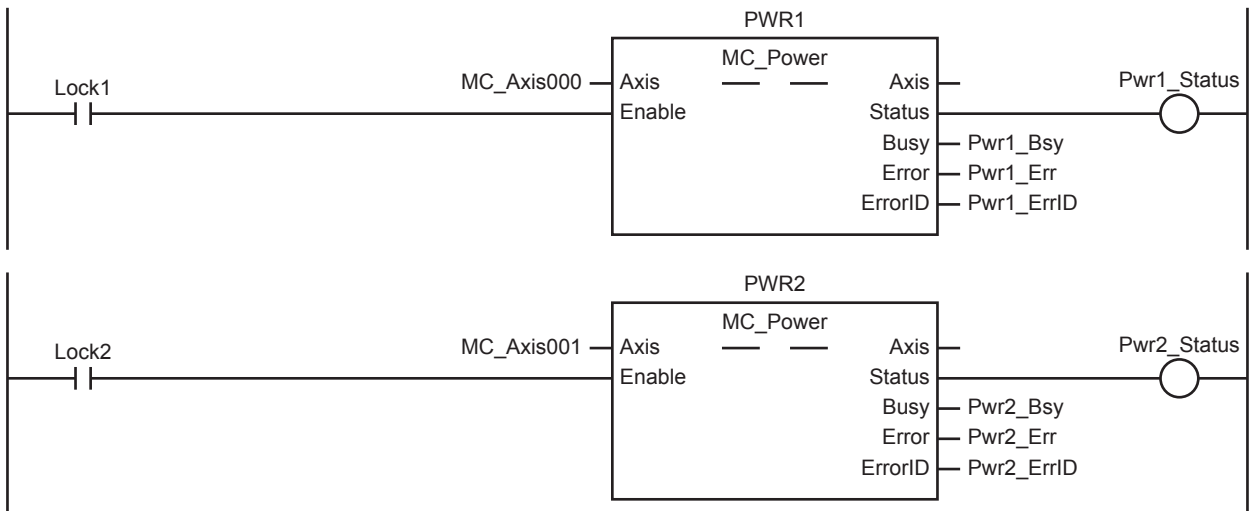


● 示例程序

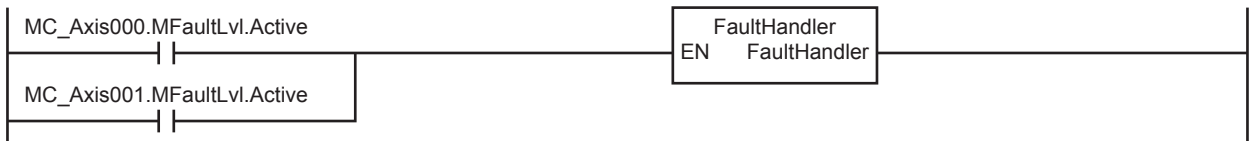
触点StartPg变为TRUE时，确认各轴的伺服驱动器处于伺服准备就绪状态



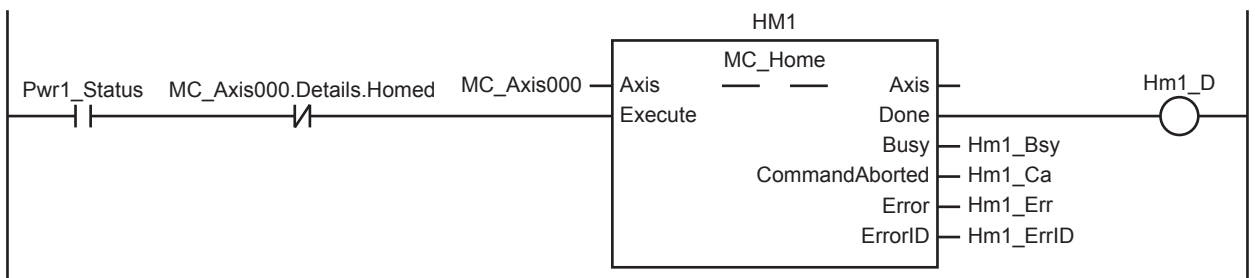
如果处于伺服准备就绪状态，则将各轴设为伺服ON状态



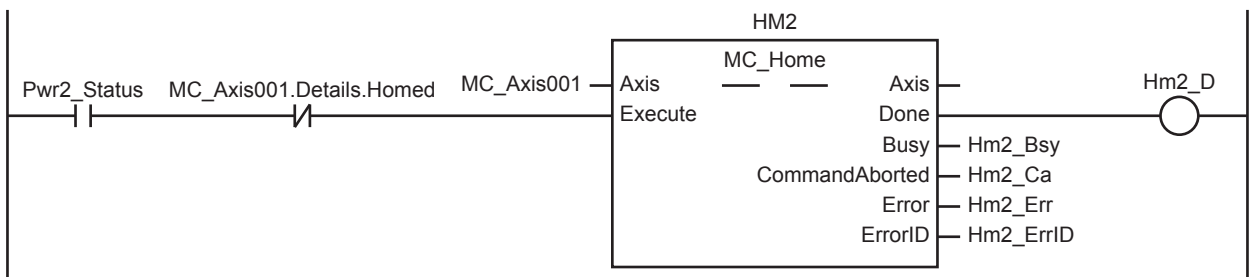
轴构成发生轻度故障程度的异常时，执行设备的异常处理(FaultHandler)。发生异常时的处理(FaultHandler)由客户根据装置进行编程



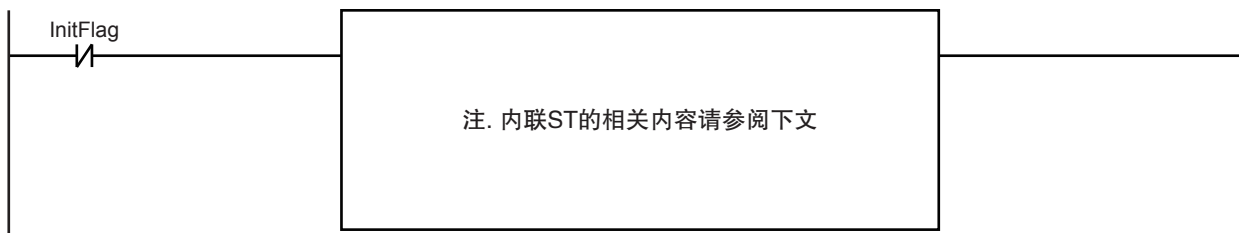
轴1变为伺服ON状态，若原点未确定，则执行原点复位，确定原点



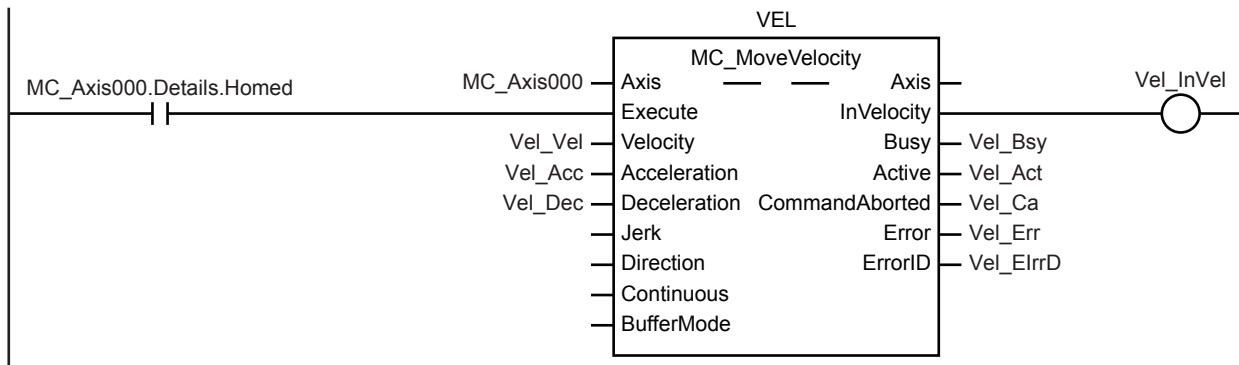
轴2变为伺服ON状态，若原点未确定，则执行原点复位，确定原点



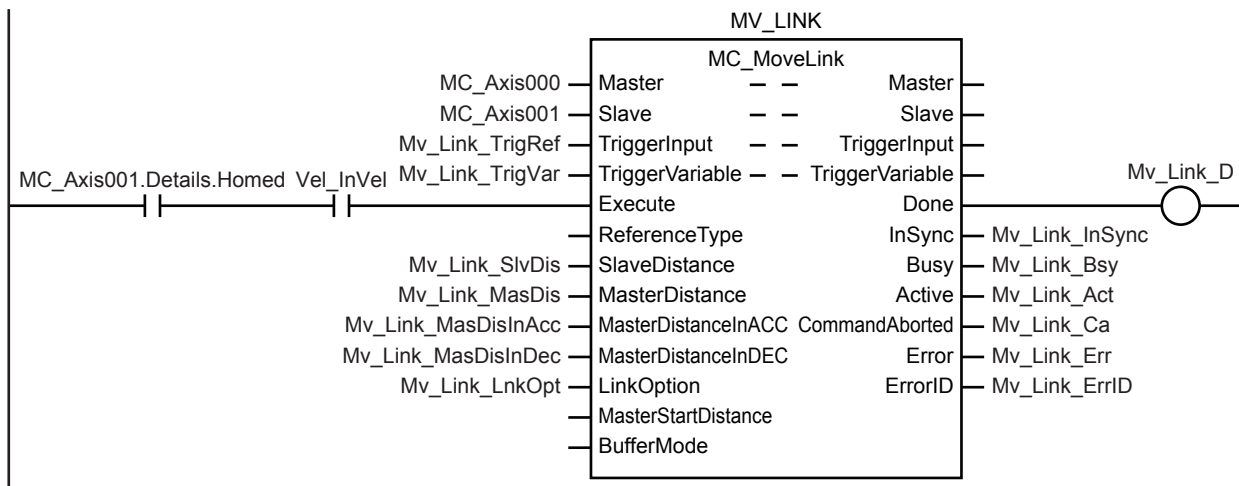
设定MC_MoveVelocity(速度控制)指令和MC_MoveLink(梯形模式凸轮)指令的参数



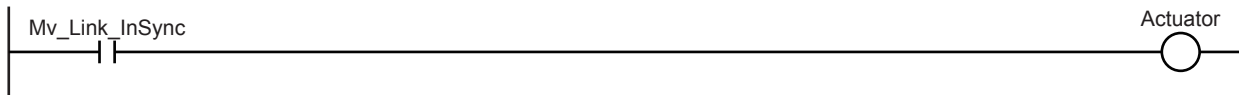
轴1已确定原点后，执行MC_MoveVelocity(速度控制)指令



从轴(轴2)已确定原点后，执行MC_MoveLink(梯形模式凸轮)指令



同步中将Actuator设为ON



内联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;

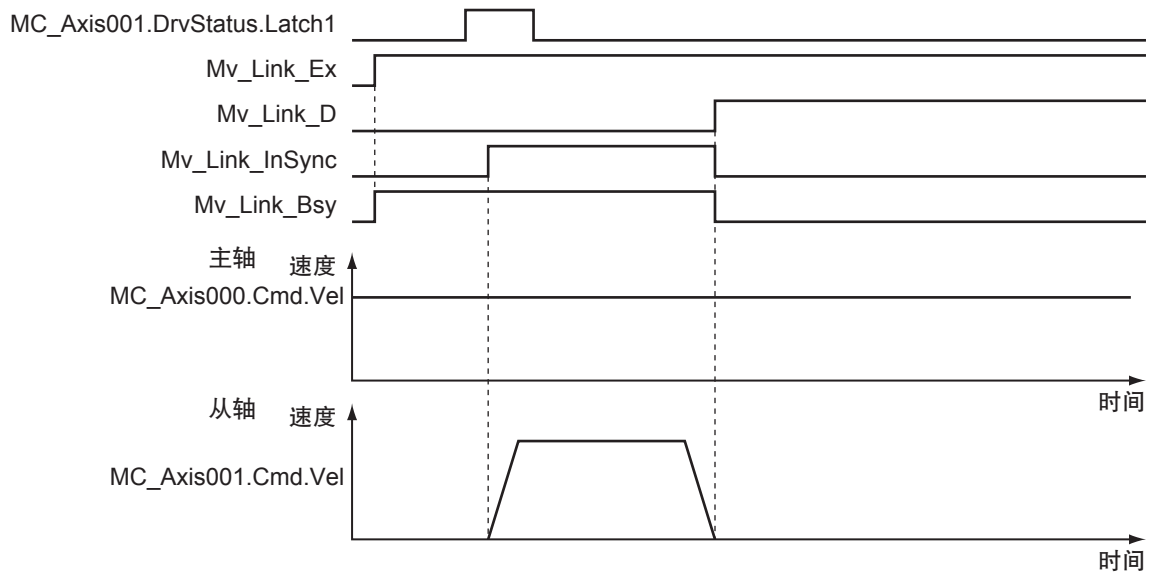
//设定输入参数后，将InitFlag设为TRUE
InitFlag := TRUE;
```

结构文本(ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	是主轴轴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为原点确定状态时变为TRUE。
MC_Axis001	_sAXIS_REF	-	是从轴轴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为原点确定状态时变为TRUE。
Pwr1_Status	BOOL	FALSE	分配给MC_Power的实例PWR1的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
Pwr2_Status	BOOL	FALSE	分配给MC_Power的实例PWR2的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则进入伺服ON状态。
Actuator	BOOL	FALSE	轴1与轴2同步时，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，确认伺服驱动器处于伺服准备就绪状态，
//将轴1设为伺服ON状态。
//如果未处于伺服准备就绪状态，则设为伺服OFF。
IF (StartPg=TRUE)
    AND (MC_Axis000.DrvStatus.Ready=TRUE) 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，确认伺服驱动器处于伺服准备就绪状态，
//将轴2设为伺服ON状态。
//如果未处于伺服准备就绪状态，则设为伺服OFF。
IF (StartPg=TRUE)
    AND (MC_Axis001.DrvStatus.Ready=TRUE) 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)由客户根据装置进行编程。
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;

//轴1处于伺服ON状态，且原点未确定时，执行原点复位
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

//轴2处于伺服ON状态，且原点未确定时，执行原点复位
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN

```

```

    Hm2_Ex:=TRUE;
END_IF;

//轴1已确定原点后，执行MC_MoveVelocity
IF MC_Axis000.Details.Homed=TRUE THEN
    Vel_Ex:=TRUE;
END_IF;

//确认轴2原点已确定，且轴1已达到目标速度，轴2执行MC_MoveLink
IF (MC_Axis001.Details.Homed=TRUE) AND (Vel_InVel=TRUE) THEN
    Mv_Link_Ex:=TRUE;
END_IF;

//轴1与轴2同步时，驱动器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	在上升沿开始指令。
CombineMode	加减法运算 方法选择	_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)	辅轴 齿轮比分子	DINT*2	正数或负数 *2	10000	(Reserved)
Ratio Denominator Auxiliary (Reserved)	辅轴 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
ReferenceTypeMaster	主轴位置类型选择	_eMC_REFERENCE_TYPE	1: _mcFeedback 2: _mcLatestCommand	2*1	指定主轴的位置类型。 1: 反馈位置(同一任务周期*4下的取值) 2: 指令位置(同一任务周期*4下的计算值)
ReferenceTypeAuxiliary	辅轴位置类型选择	_eMC_REFERENCE_TYPE	1: _mcFeedback 2: _mcLatestCommand	2*1	指定辅轴的位置类型。 1: 反馈位置(同一任务周期*4下的取值) 2: 指令位置(同一任务周期*4下的计算值)
BufferMode	缓存模式选择	_eMC_BUFFER_MODE	0: _mcAborting	0*1	指定多重启动运动指令时的动作。 0: 中断

*1. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

*2. Ver.1.02以上的CPU单元和Ver.1.03以上的Sysmac Studio组合时可使用。上述以下版本进行组合时，数据类型为“UINT”，有效范围为“正数”。

*3. Ver.1.02以上的CPU单元和Ver.1.03以上的Sysmac Studio组合时可使用。上述以下版本进行组合时，数据类型为“UINT”。

*4. 任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先级5)的周期。

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	加减运算时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

* 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
InCombination	开始加减运算定位时	<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	开始指令后	<ul style="list-style-type: none"> Error变为TRUE时 CommandAborted变为TRUE时
CommandAborted	<ul style="list-style-type: none"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 执行MC_Stop指令中，启动本指令时 	<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含有异常因素时	异常已解除时

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	辅轴	_sAXIS_REF	-	指定辅轴。 ^{*1}
Slave	从轴	_sAXIS_REF	-	指定从轴。 ^{*1}

*1. 请在使用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

对主轴、从轴、辅轴中的两个指定相同轴时，发生如下轻度故障等级的异常。

- “主轴从轴相同(错误代码：5436 Hex)”异常
- “主轴辅轴相同(错误代码：5437 Hex)”异常
- “辅轴从轴相同(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

Ver.1.09以下的CPU单元使用本指令时，请勿对Master(主轴)执行MC_SetPosition(当前位置变更)指令。对Master(主轴)执行MC_SetPosition(当前位置变更)指令时，可能出现Slave(从轴)急速跟踪的情况，非常危险。

要对Master(主轴)使用MC_SetPosition(当前位置变更)指令时，请解除Master(主轴)和Slave(从轴)的关系后执行该指令。

有关对于主轴的注意事项，请参阅 “同步控制的主轴及辅轴的注意事项(P.1-6)”。

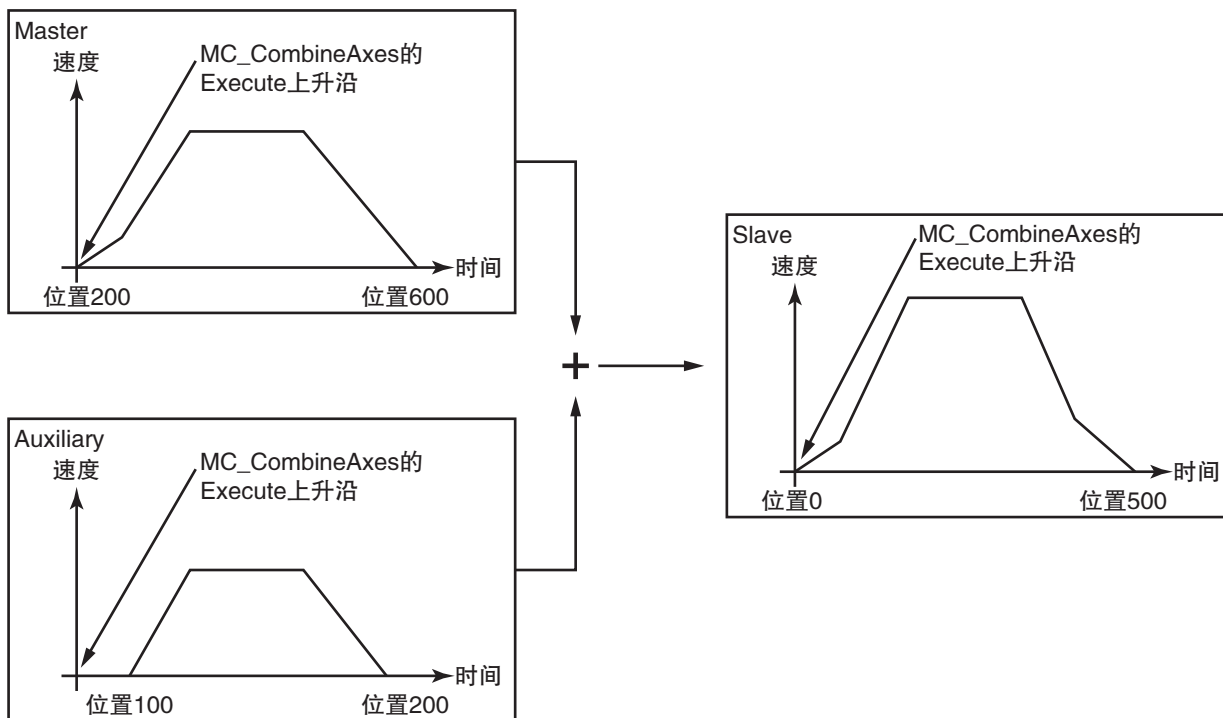
Auxiliary(辅轴)也与Master(主轴)相同。

指令详情

对起始点起的Master(主轴)位置的相对值和Auxiliary(辅轴)位置的相对值相加或相减，将算出的相对值作为Slave(从轴)的指令位置输出。

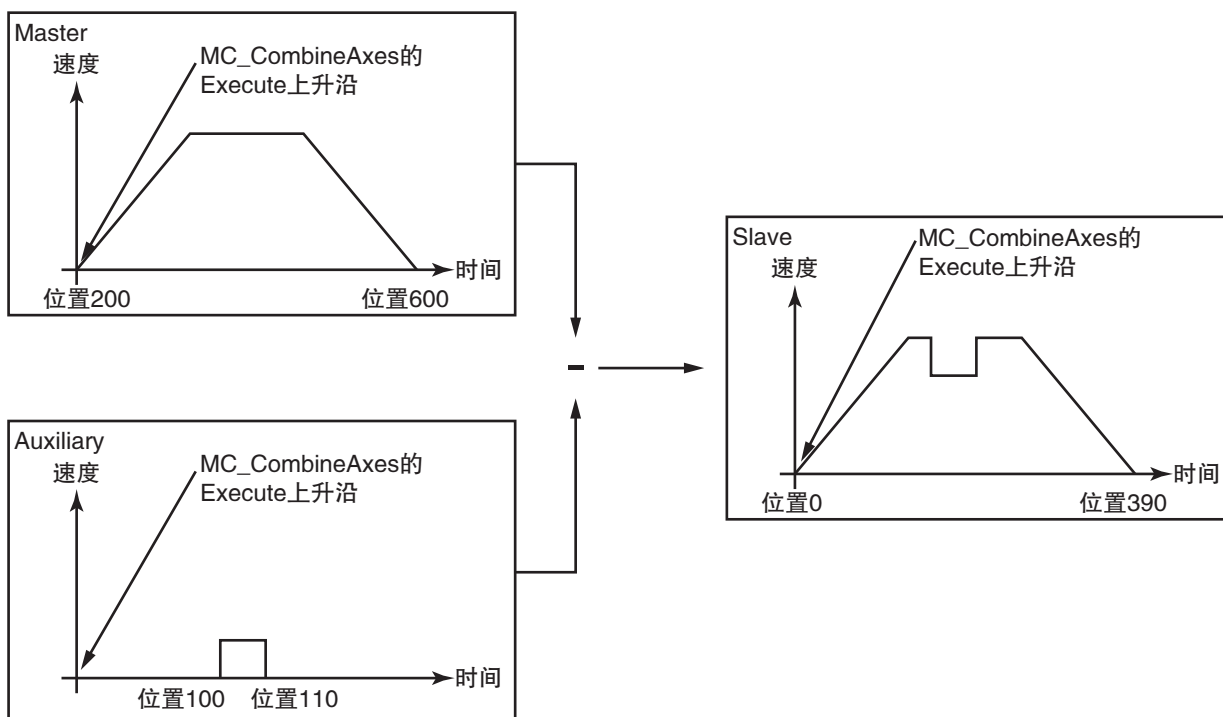
● CombineMode为_mcAddAxes时

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(辅轴)位置(相对位置)



- 位置的加法或减法运算，是对数值进行运算，与轴参数的轴的坐标单位选择无关。

- 要结束本指令，请启动MC_Stop(强制停止)指令。



使用注意事项

根据Master(主轴)和Auxiliary(辅轴)数值的不同，Slave(从轴)的移动量、速度、加速度可能发生急剧变化。
请谨慎使用。

● 到位检查

本指令不执行到位检查。

● 超调

使用本指令时，无法通过MC_SetOverride(超调值设定)指令进行超调。

● ReferenceType(位置类型选择)

可从下列选项中选择位置的类型。

- `_mcFeedback`：同一任务周期下的取值
使用同一任务周期时获取的主轴反馈位置。
 - `_mcLatestCommand`：指令位置(同一任务周期下的计算值)
使用同一任务周期时计算的主轴指令位置。
选择`_mcLatestCommand`时，主轴编号和辅轴编号的设定值必须小于从轴编号。
否则，本指令的Error(错误)将变为TRUE。ErrorID(错误代码)输出“主轴/从轴 轴号非升序(错误代码：5438 Hex)”。
- 主轴和辅轴之间不存在轴编号的限制。



使用注意事项

这里的任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先度5)的周期。同样，固定周期任务表示原始恒定周期任务或固定周期任务(执行优先度5)。

● 轴种类与位置类型的关系

可监控的轴种类和要监控的位置类型的关系如下所示。

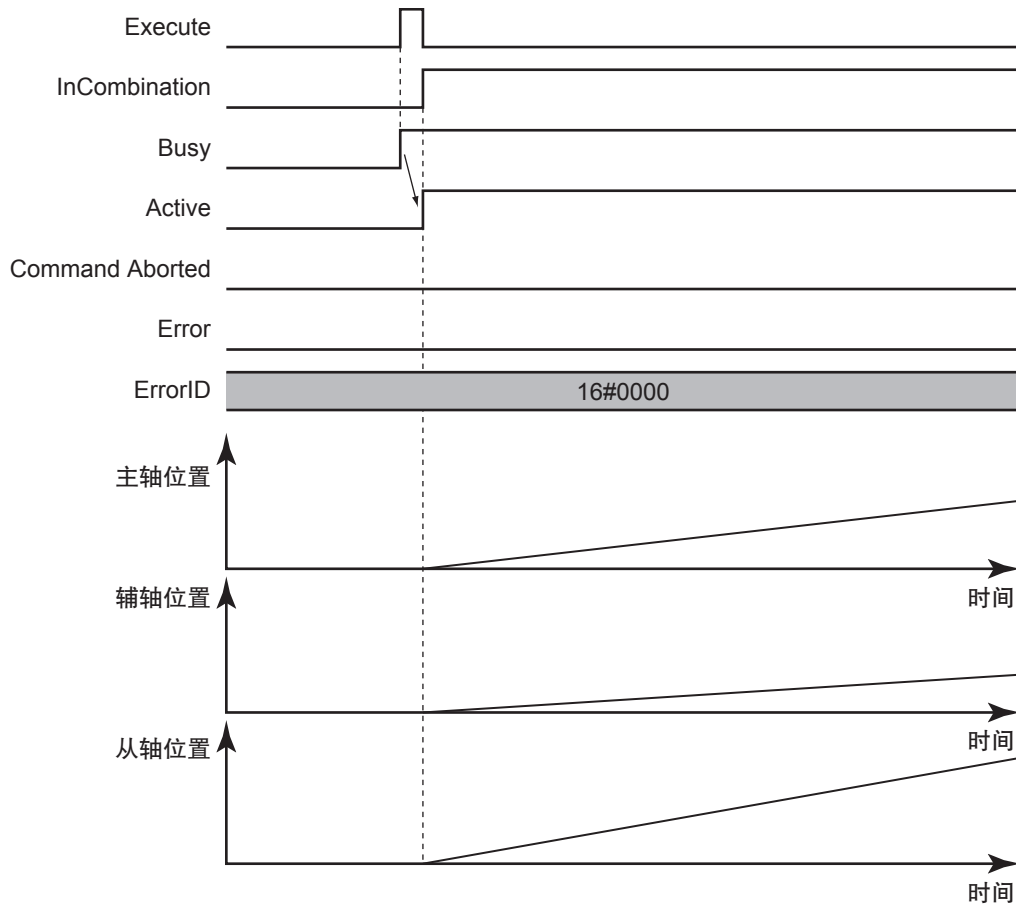
轴种类	ReferenceType	
	<code>_mcFeedback</code>	<code>_mcLatestCommand</code>
伺服轴	○	○
编码器轴	○	x *1
虚拟伺服轴	○	○
虚拟编码器轴	○	x *1

*1. 启动指令时，发生“超过位置类型选择范围(错误代码：5430 Hex)”错误。

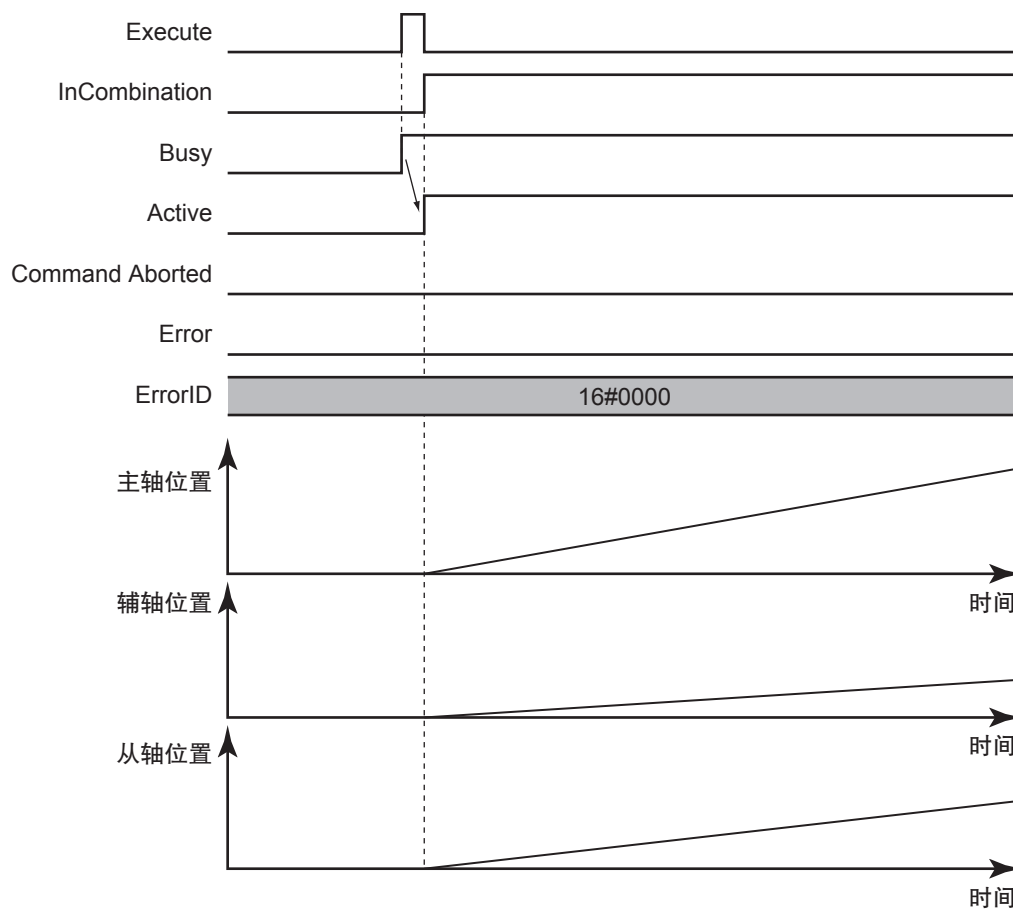
时序图

- 在启动Execute(启动)的同时，Busy(执行中)变为TRUE。在下一个周期Active(控制中)变为TRUE。
- 在开始加减运算输出的周期，InCombination(加减运算中)变为TRUE。
- 利用其它指令中断本指令时，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。

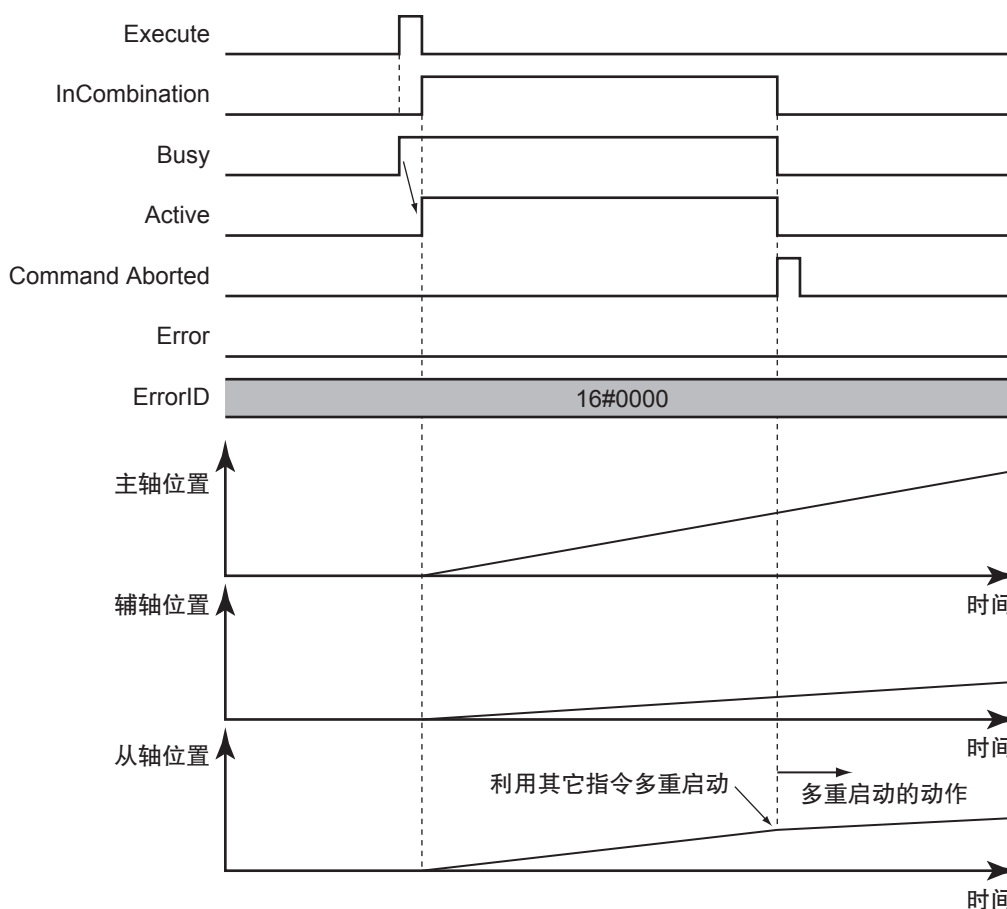
● CombineMode为_mcAddAxes时



● CombineMode为_mcSubAxes时



● 中断时



重启运动指令

无法重启本指令。

重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 本指令执行中的其它指令启动

对本指令执行指令多重启动时，请指定从轴。

本指令动作中多重启动其它指令时，根据BufferMode(缓存模式选择)，存在如下限制。

- 本指令动作中其它指令的多重启动，可以选择中断方式。

无法以等待、合并方式多重启动指令。

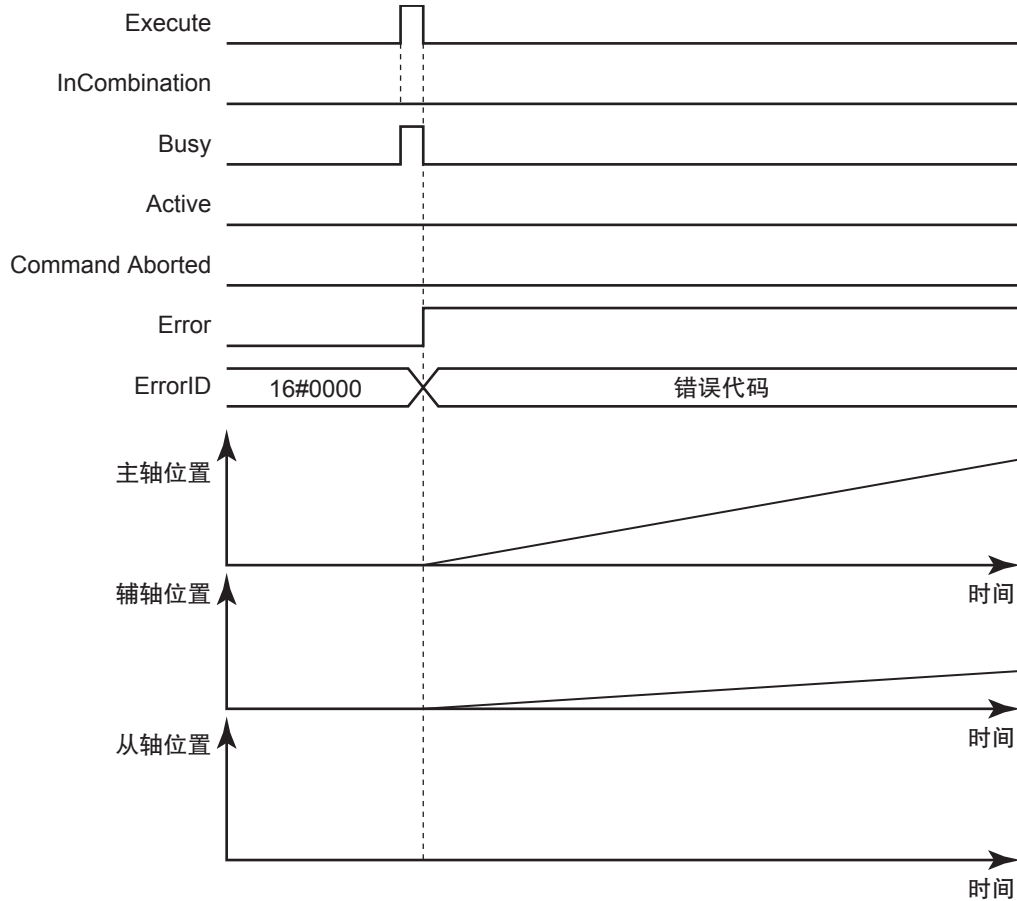
异常

● 发生异常时的时序图

启动本指令发生异常时，Error(错误)变为TRUE，轴停止动作。

减速停止时的减速度为轴参数的最大减速度。

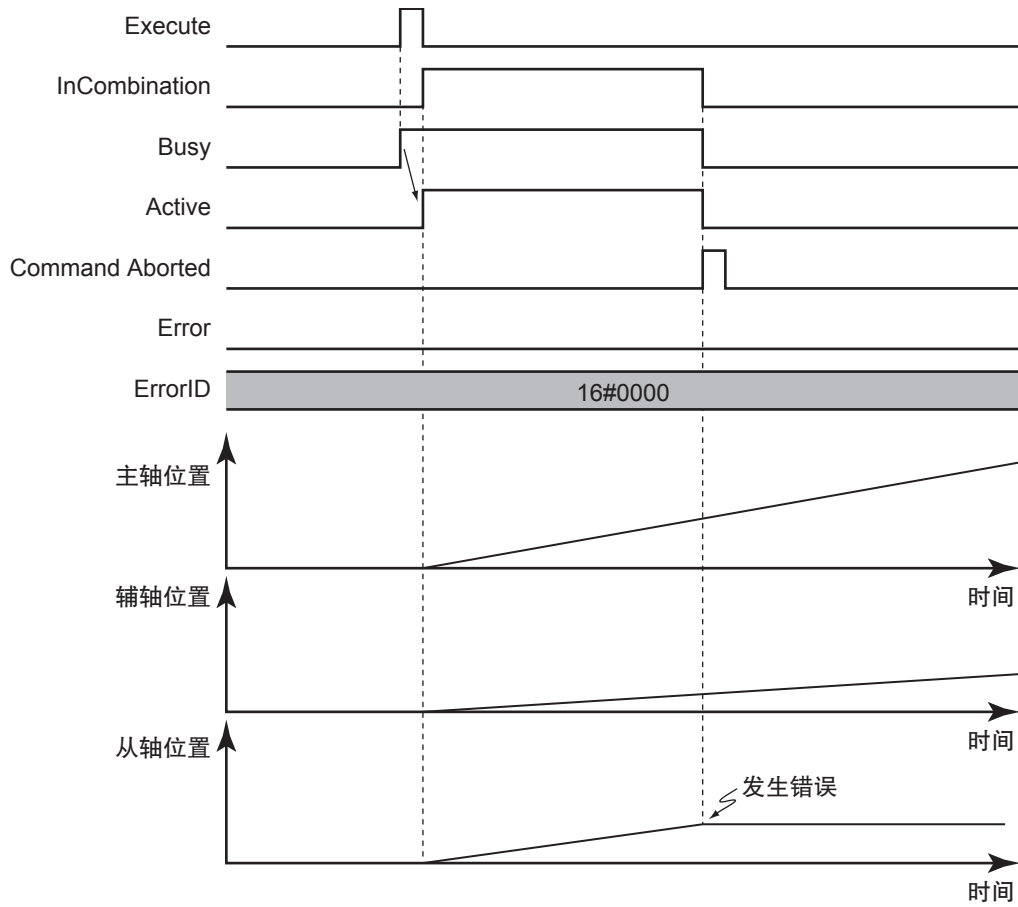
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。



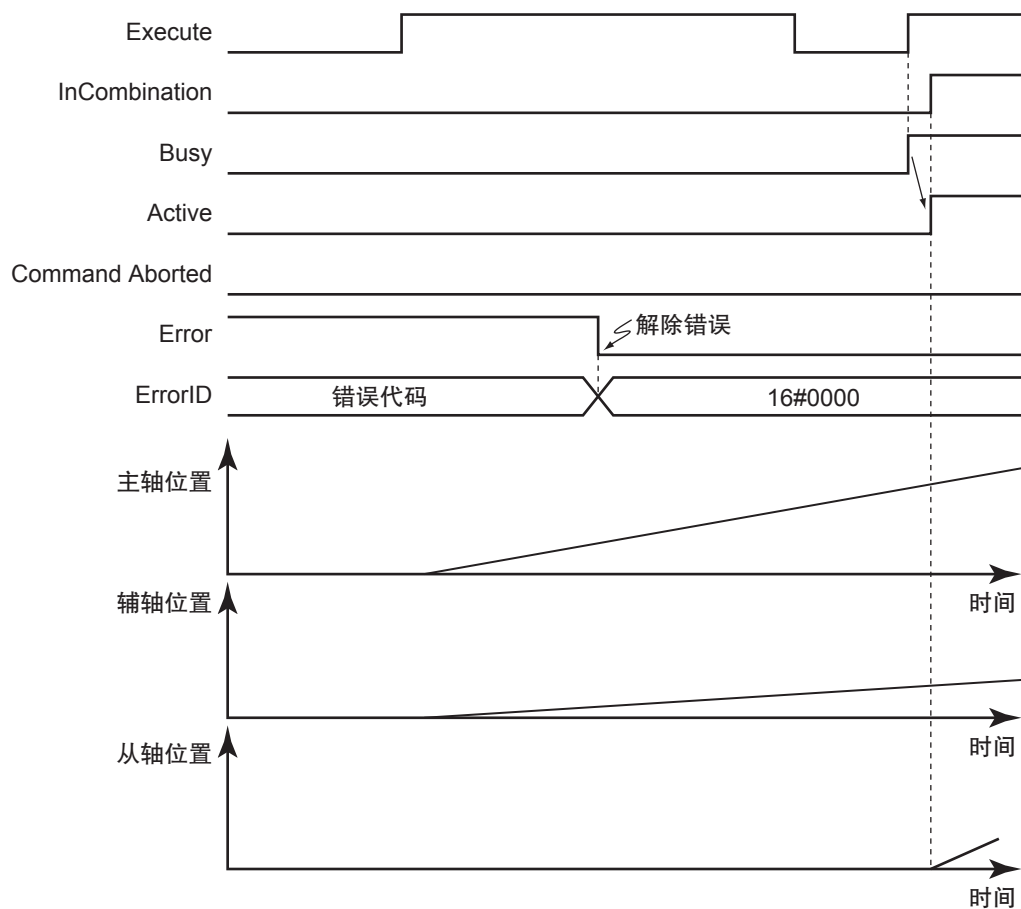
参考

- 本指令不受Master(主轴)、Auxiliary(辅轴)的异常状态的影响。
- 异常状态解除，Master(主轴)、Auxiliary(辅轴)动作时，Slave(从轴)继续加减运算定位动作。此外，启动本指令时以及执行本指令中，从轴发生异常时，中断本指令。但对Master(主轴)、Auxiliary(辅轴)没有影响。

本指令执行中发生轻度故障等级的异常时，CommandAborted(执行中断)变为TRUE，轴停止动作。减速停止时的减速度为轴参数的最大减速度。
查看Slave(从轴)的轴变量MFalutLvl.Code的输出值，可了解发生异常的原因。



解除本指令的异常后，在下一个Execute(启动)的上升沿才可启动本指令。



● 错误代码

关于指令发生的异常，请参阅 “A-1 错误代码一览(P.A-2)”。

MC_Phasing

执行同步控制中主轴的相位补偿。

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	在上升沿开始指令。
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	缓存 模式选择	_eMC_ BUFFER_ MODE	0: _mcAborting	0 ^{*3}	指定多重启动运动指令时的动作。 0: 中断


*1. 关于指令单位，请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*2. 请务必设定目标速度。不作设定就执行动作，将超出目标速度设定范围(错误代码：5422Hex)。

*3. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

*  请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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	开始相位补偿后	<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"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 执行MC_Stop指令中，启动本指令时 执行中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含有异常因素时	异常已解除时

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. 请在使用在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[*])。

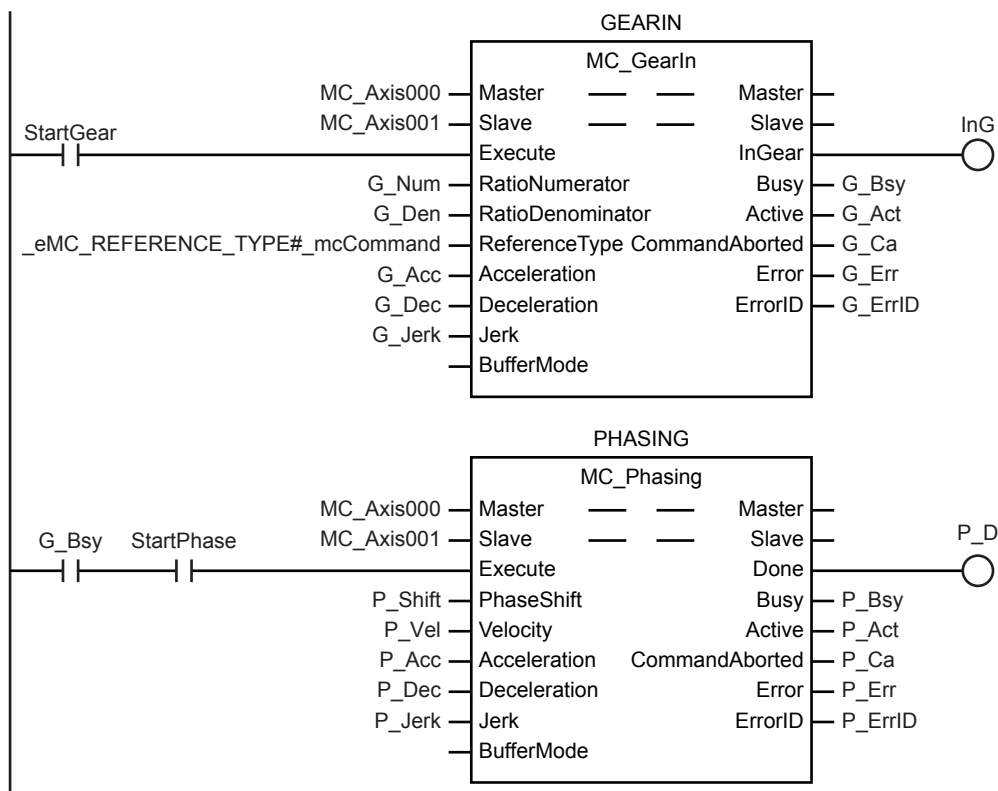


使用注意事项

主轴和从轴指定同一轴时，会发生轻度故障等级的“主轴从轴相同(错误代码：5436 Hex)”的异常。

功能说明

- 在MC_CombineAxes(加减运算定位)指令执行中以外的情况下,如果在单轴同步控制中启动本指令,则根据设定的PhaseShift(相位补偿量)、Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度),对主轴相位进行补偿。
- 主轴的指令当前位置和反馈当前位置保持不变,以相对量对这两个位置进行补偿的值即为“主轴的相位”。
从轴与已补偿的“主轴的相位”同步。
- 达到PhaseShift(相位补偿量)时,Done(完成)变为TRUE。
- 执行中的同步控制指令完成时,补偿结束。再次执行同步控制指令时,之前的补偿量无影响。
- 主轴相位补偿有效的同步控制指令分为:MC_CamIn(凸轮动作开始)指令、MC_GearIn(齿轮动作开始)指令、MC_GearInPos(位置指定齿轮动作)指令、MC_MoveLink(梯形模式凸轮)指令4种。
- 在用户程序中记载本指令时,如下图所示,请在同步控制指令之后描述MC_Phasing(主轴相对值相位补偿)指令。



使用注意事项

有关对于主轴的注意事项,请参阅 □ “同步控制的主轴及辅轴的注意事项(P.1-6)”。

指令详情

下面对指令详细说明。

● Master(主轴)、Slave(从轴)的指定

Master(主轴)和Slave(从轴)均以轴变量名指定轴。

指定同步控制指令非执行中的Master(主轴)、Slave(从轴)的组合时,将发生轴指定异常。

● PhaseShift(相位补偿量)

对PhaseShift(相位补偿量)指定从Slave(从轴)观察的Master(主轴)的相位补偿量。

相位补偿量是指定相对量。

● Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度)、Jerk(跃度)

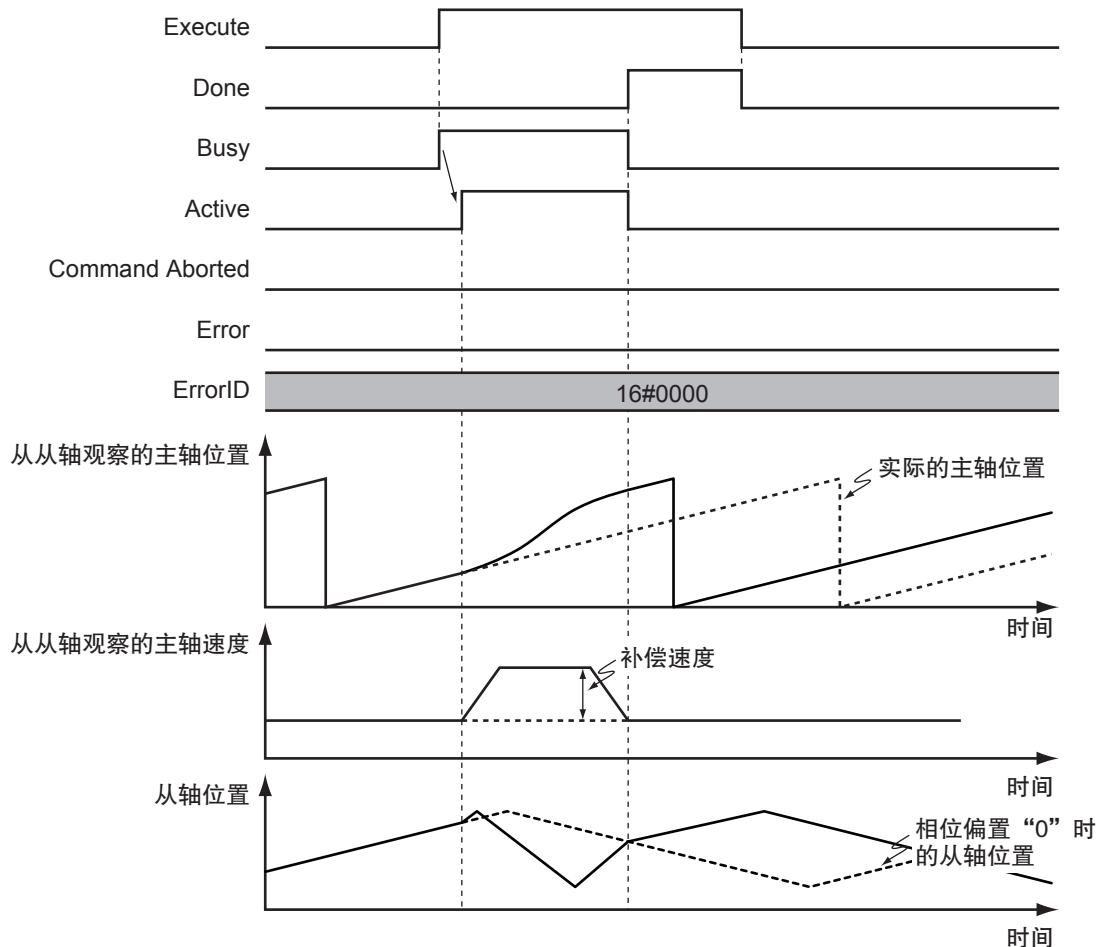
相位补偿量的目标速度(补偿速度)、加速度、减速度、跃度通过Velocity、Acceleration、Deceleration、Jerk指定。

从Slave(从轴)观察的Master(主轴)的目标速度(补偿速度)为相对主轴速度的速度。

根据相位补偿量、加速度、减速度、跃度的不同,从Slave(从轴)观察的Master(主轴)的目标速度(补偿速度)如下所示。

例)进行电子凸轮的主轴相对值相位补偿时

从Slave(从轴)观察的Master(主轴)的补偿速度为相对主轴速度的速度。



- 相位补偿量指定为“0”时，Master(主轴)的相位补偿量为“0”，正常结束补偿。
- 目标速度(补偿速度)指定为“0”时，指定超出范围，发生Slave(从轴)的轴异常。
- 指定的目标速度(补偿速度)和主轴速度的合计速度，可指定为超过Master(主轴)的最高速度。



参考

执行MC_Phasing(主轴相对值相位补偿)指令后，作为同步的Slave(从轴)的动作，对异常进行检测。

因此，启动指令时，不对本指令的Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度)的设定进行异常检测。

● BufferMode(缓存模式选择)

指定前一个轴动作和本次动作的连接方式。

有以下1种。

缓存模式选择	说明
中断	重启本指令时，立即开始利用本指令进行补偿。

多重启动的反转时动作，遵照主轴设定的[反转时动作选择]。

关于BufferMode(缓存模式选择)，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 到位检查

相位补偿动作的完成即为补偿值输出的结束，不执行到位检查。

重启运动指令

本指令执行中，启动再次指令开始时，进行PhaseShift(相位补偿量)、Velocity(目标速度)、Acceleration(加速度)、Deceleration(减速度)的变更。

变更的动作与相对定位相同。

多重启动运动指令

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 本指令执行中的其它指令启动

Slave(从轴)中，本指令执行中其它指令的多重启动，可以选择中断方式。

无法以等待、合并方式多重启动指令。

● 多重启动多个本指令

指定的从轴在MC_Phasing(主轴相对值相位补偿)指令动作中时，可以启动MC_Phasing(主轴相对值相位补偿)指令。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，从轴停止动作。

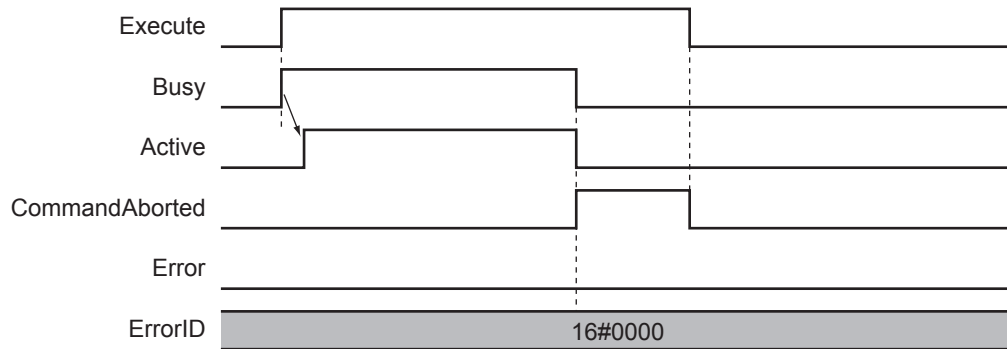
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

在因Slave(从轴)发生异常而解除同步状态时，MC_Phasing(主轴相对值相位补偿)指令动作变为Command Aborted(执行中断)。

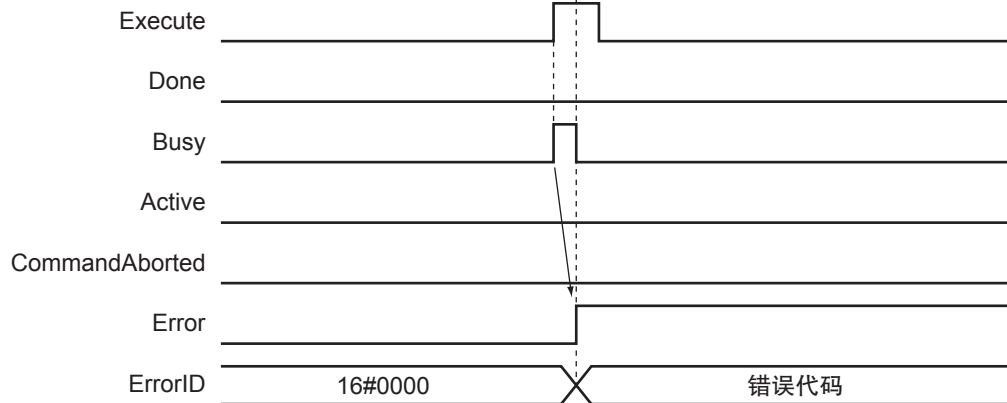
主轴的异常状态对本指令的动作没有影响。

● 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

关于指令发生的异常，请参阅 □ “A-1 错误代码一览(P.A-2)”。

MC_TorqueControl

利用伺服驱动器的转矩控制模式进行转矩控制。



使用注意事项

- 无法对NX系列脉冲输出单元使用本指令。

指令	名称	FB/ FUN	图形表现	ST表现
MC_TorqueControl	转矩控制	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	在上升沿开始指令。
Torque	目标转矩	LREAL	0 ~ 1000.0	300.0	以[0.1%]为单位指定向伺服驱动器输出的目标转矩。 以额定转矩为“100.0%”时的比率进行指定。 ^{*1} 单位为[%]。
TorqueRamp	转矩指示灯	LREAL	正数或“0”	0	指定从当前转矩到输出目标转矩为止的转矩变化率。 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}	指定目标转矩的方向。 0: 指定为正方向 2: 指定为负方向
BufferMode	缓存 模式选择	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered	0 ^{*3}	指定多重启动运动指令时的动作。 0: 中断 1: 等待

*1. 指定值超过“1000.0%”时，作为“1000.0%”处理。指定负值时，作为“0.0%”处理。

*2. 关于指令单位，请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”的“单位转换设定”。

*3. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

输出变量

输出变量	名称	数据类型	有效范围	内容
InTorque	目标转矩到达	BOOL	TRUE, FALSE	达到目标转矩时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

* □ 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
InTorque	目标转矩输出中	<ul style="list-style-type: none"> • Error变为TRUE时 • CommandAborted变为TRUE时 • 重启指令，变更目标转矩时
Busy	Execute的上升沿	<ul style="list-style-type: none"> • Error变为TRUE时 • CommandAborted变为TRUE时
Active	开始转矩指令值的输出时	<ul style="list-style-type: none"> • Error变为TRUE时 • CommandAborted变为TRUE时
CommandAborted	<ul style="list-style-type: none"> • 利用其它指令多重启动运动指令(中断)，中止本指令时 • 因发生异常，中止本指令时 • 发生异常过程中，启动本指令时 • 执行MC_Stop指令中，启动本指令时 	<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含有异常因素时	异常已解除时

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. 请使用在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

- 直接指定转矩指令值，控制伺服电机的输出转矩。
- 以[0.1%]为单位指定目标转矩。指定值小数点后超过第2位时，第2位四舍五入。
- 要停止执行本指令，请使用MC_Stop(强制停止)指令。
- 使用伺服驱动器的转矩控制模式功能实现了转矩控制。
- 保持之前的控制模式，直至切换控制模式。

例

由位置控制切换为转矩控制时：保持位置控制，直至伺服驱动器变为转矩控制

由转矩控制切换为位置控制时：保持转矩控制，直至伺服驱动器变为位置控制

● 数据对象的映射

使用MC_TorqueControl(转矩控制)指令时，请在Sysmac Studio的轴基本设置的[高级设置]中映射如下对象数据。

- 目标转矩(6071Hex)
- 操作模式(6060Hex)
- 当前转矩(6077Hex)
- 操作模式显示(6061Hex)

未进行这些设定时，将发生过程数据对象设定不足(错误代码：3461Hex)错误。

关于数据对象的映射，请参阅 □□ “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-363)”或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

指令详情

下面对本指令进行详细说明。

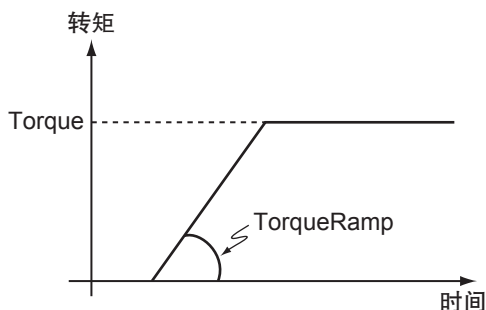
● Axis(轴)指定

通过Axis(轴)指定转矩控制对象轴。

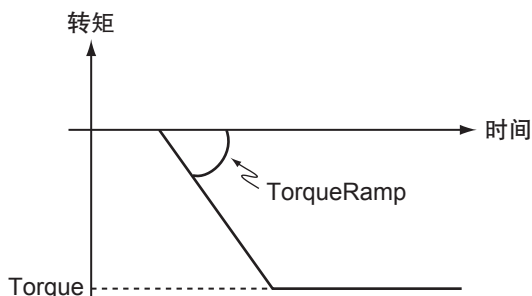
● TorqueRamp(转矩斜度)

指定从当前指定的指令转矩到输出目标转矩的倾斜度。

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

设定目标转矩时，必须确保小于电机最大转矩。
超过电机最大转矩设定时的动作取决于伺服驱动器。

● Velocity(限制速度)

对转矩控制时轴的最大速度进行限制。

轴的速度达到该限制值时，通过伺服驱动器减小转矩，降低轴的速度。

使用伺服驱动器的功能，可实现限制速度值的功能。

详情请参阅 □□ “各伺服驱动器手册的转矩控制功能”。



使用注意事项

- 进行转矩控制时，轴的速度变快。为安全起见，请务必设定Velocity(限制速度)。
- 使用欧姆龙制伺服驱动器 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”。未设定时，速度限制值不起作用。并且，即使极限输入信号变为ON，轴也不会停止动作。
- 速度限制值使用过程数据607F Hex。
使用欧姆龙制伺服驱动器1S系列或G5系列时，请在Sysmac Studio的轴参数设定画面的“详细设定”中，设置为使用速度限制值(607F Hex)。
此外，通过其他公司伺服驱动器使用速度限制功能时，请参阅各公司伺服驱动器的手册。

● Direction(方向选择)

指定目标转矩的输出方向。

要沿轴的正方向输出时，指定正方向；要沿轴的负方向输出时，指定负方向。

● BufferMode(缓存模式选择)

指定前一个轴动作和本次动作的连接方式。

有以下2种。

缓存模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。
等待	当前正在执行的指令正常完成后，已缓存的本指令自动启动。

关于BufferMode(缓存模式选择)，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 转矩控制中的轴停止

使用欧姆龙制伺服驱动器1S系列或G5系列时，如果在本指令执行中启动MC_Stop(强制停止)指令，则不适用MC_Stop(强制停止)指令的减速度，表现为立即停止。

在发生减速停止处理的异常时，也表现为立即停止。

● 转矩控制中的指令当前位置、反馈当前位置

基于本指令的转矩控制中，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：以齿轮比对从伺服驱动器获取的值转换后的值。

指令当前位置：与1个周期前的反馈当前位置的值相同。

● 可使用轴和启动条件

- 在伺服轴中，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。
- 虚拟伺服轴可常时接收。但是，不执行伺服驱动器的控制模式切换处理。
- 编码器轴、虚拟编码器轴则会发生启动时异常。

● 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功能模块切换为CSP模式。但是，在伺服OFF状态下，欧姆龙制伺服驱动器G5系列无法受理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。

并且，轴变量的DrvStatus(伺服驱动器状态)下，CST(周期同步转矩(CST)模式中)变为TRUE。

● 原点状态

原点确定状态保持不变。

● 软件限位

可以采用软件限制。

通过轴参数选择[对指令位置有效，执行减速停止]、或[对指令位置有效，执行累积脉冲停止]时，也可采用软件限制。

● 计数模式为[线性模式]时

选择溢出或下溢时，作为不带目标位置的动作处理。

● 反转时动作

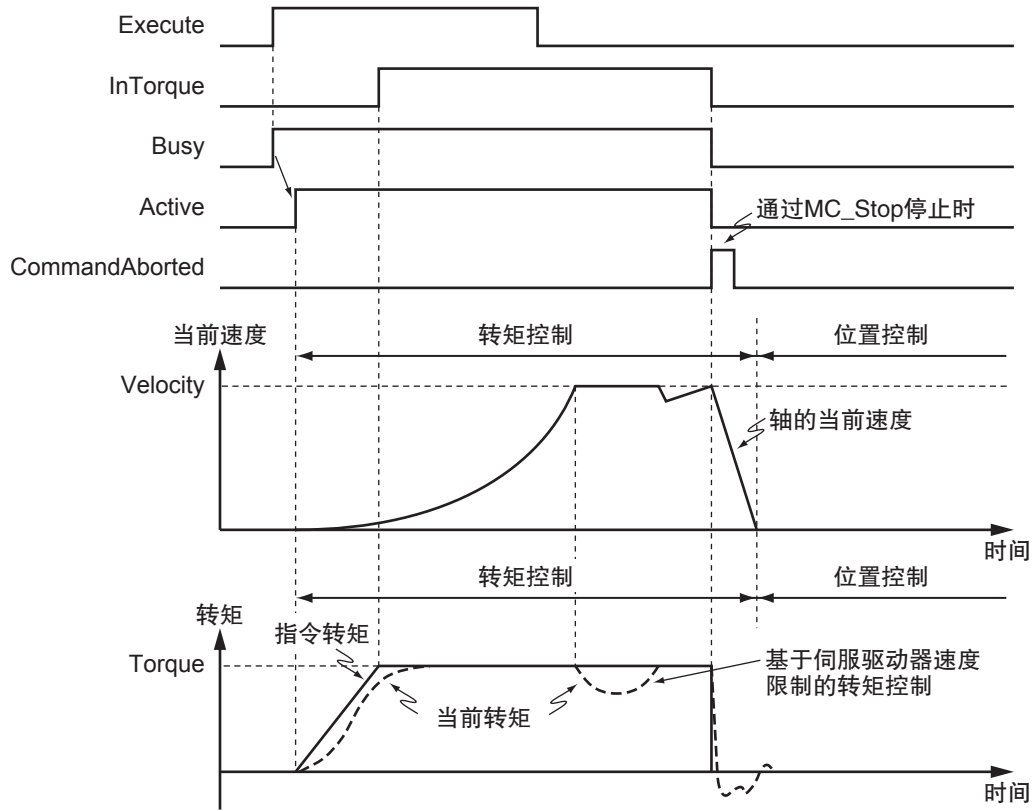
执行多重启动时，如果转矩指令值反转，则通过本指令的TorqueRamp(转矩斜度)进行动作。与轴参数的[反转时动作选择]的设定内容无关。

利用多重启动反转时的动作如下所示。

- 在本指令动作中，多重启动使用CSP的指令，指令位置反转时，根据轴参数的[反转时动作选择]，进行反转动作。
- 在使用CSP或CSV的指令的动作中，多重启动本指令，转矩指令值反转时，根据TorqueRamp(转矩斜度)，使转矩指令反转。
- 在本指令动作中，多重启动本指令，转矩指令值反转时，根据TorqueRamp(转矩斜度)，使转矩指令反转。

时序图

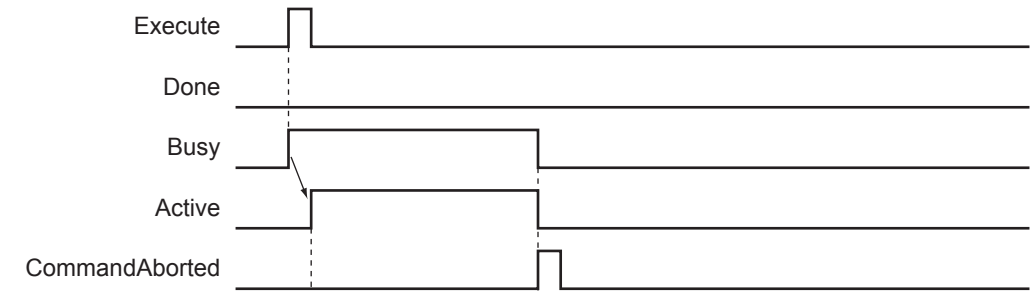
启动本指令，使之停止时



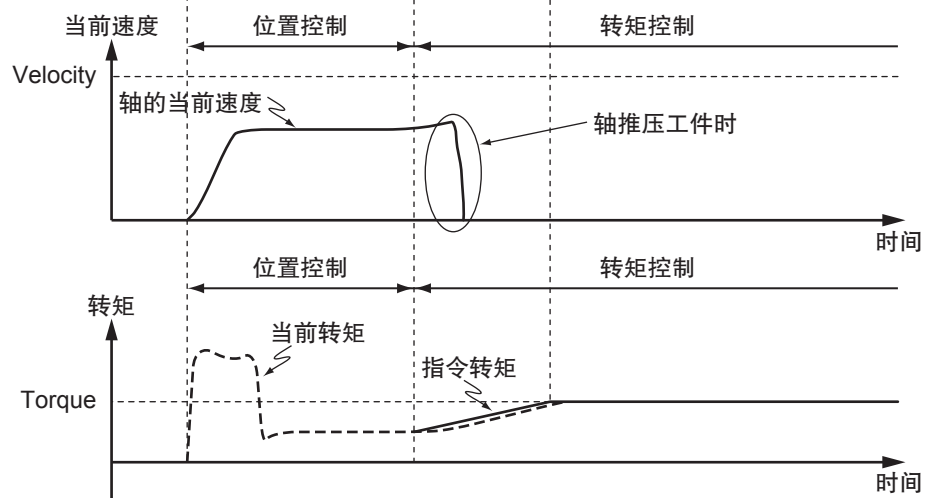
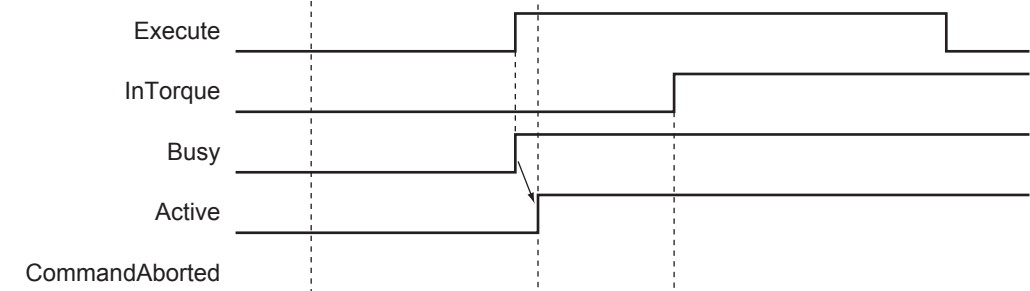
前一个动作中以中止方式启动时

在本指令执行中，通过轴停止的接触的应用进行说明。

之前的位置控制指令



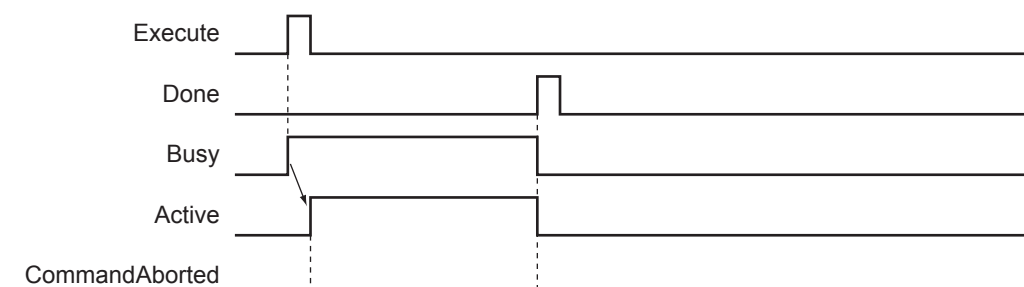
当前的指令



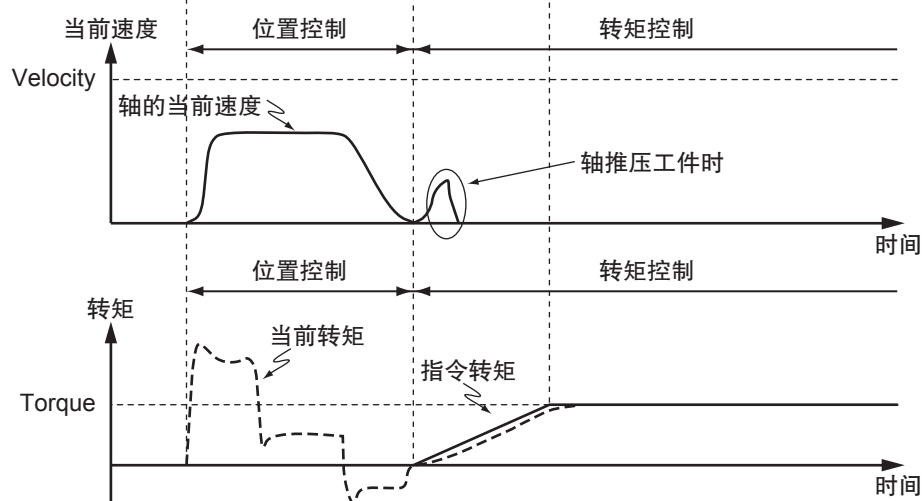
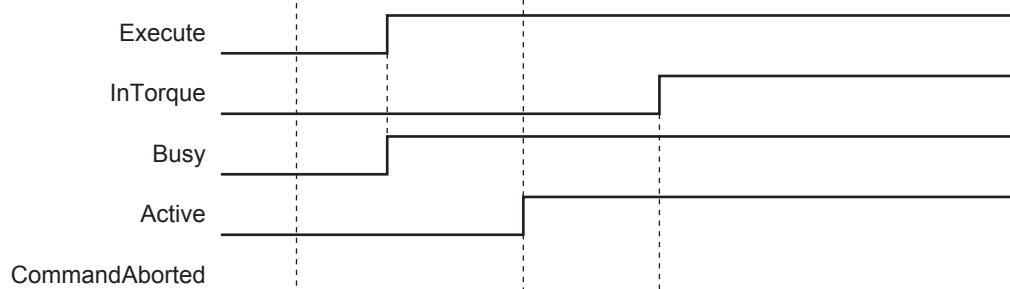
前一个动作中以等待方式启动时

在本指令执行中，通过轴停止的接触的应用进行说明。

之前的位置控制指令



当前的指令



● 控制模式的切换

- 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的过程中，执行MC_TorqueControl(转矩控制)指令时，根据BufferMode(缓存模式选择)设定的不同，动作发生变化。

BufferMode(缓存模式选择)选择中断时，在执行指令时切换为转矩控制。选择缓存时，在前一个动作完成时切换为转矩控制。

- 在MC_TorqueControl(转矩控制)指令执行中，通过MC_MoveAbsolute(绝对值定位)等其他指令中止时、或者轴发生异常时，在该时间点切换为位置控制。
- 在指令执行时，Active(控制中)会发生切换，但是在伺服驱动器侧的控制模式切换之前，需要几个周期。控制模式切换之前的时间取决于伺服驱动器。

控制模式切换判断

欧姆龙制伺服驱动器1S系列或G5系列停止轴动作时，MC功能模块将速度限制值(607F Hex)设为“0”。此后，在固定周期任务中，反馈当前速度连续3次满足下列条件式时，切换为CSP。

$$\text{反馈当前速度} \leq (\text{最高速度}) \times 0.1$$

使用其他公司伺服驱动器时，在将伺服驱动器的控制模式从CST切换为CSP的同时，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原始恒定周期任务或固定周期任务(执行优先度5)。

控制模式切换失败

MC功能模块对伺服驱动器执行控制模式切换指令并经过1秒钟后，伺服驱动器的切换还未完成时，发生“伺服驱动器的控制模式切换异常(错误代码：7439 Hex)”的错误，执行伺服OFF(自由运行停止)。

关于“伺服驱动器的控制模式切换异常(错误代码：7439 Hex)”，请参阅 [□□](#) “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 [□□](#) “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

将速度限制值设为“0”经过10秒钟后，切换条件式仍然无法成立时，同上执行伺服OFF。

控制模式切换动作示例

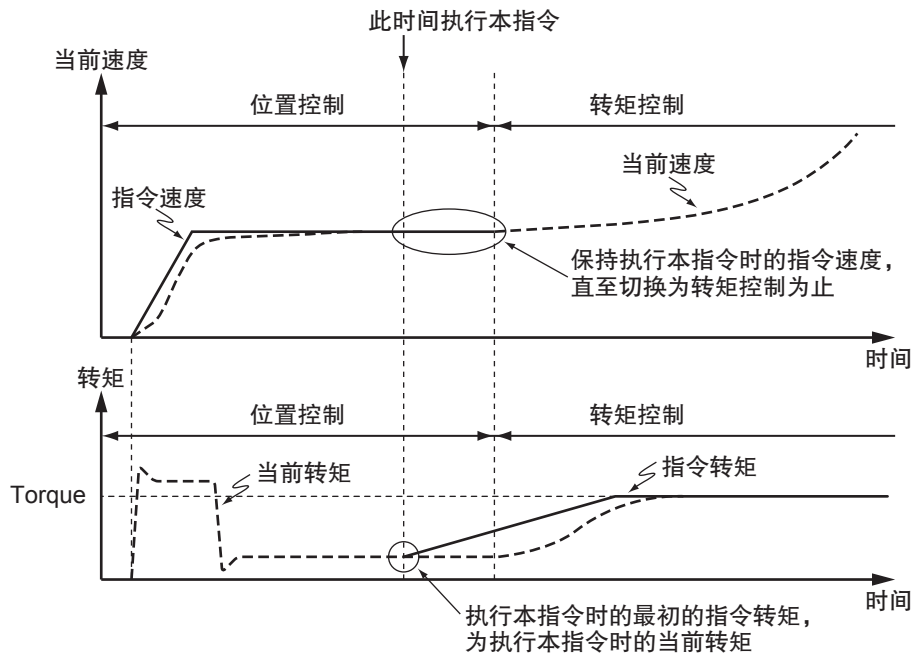
下面，以轴动作中切换时为例，对切换控制模式之前的指令转矩和指令速度的关系进行说明。



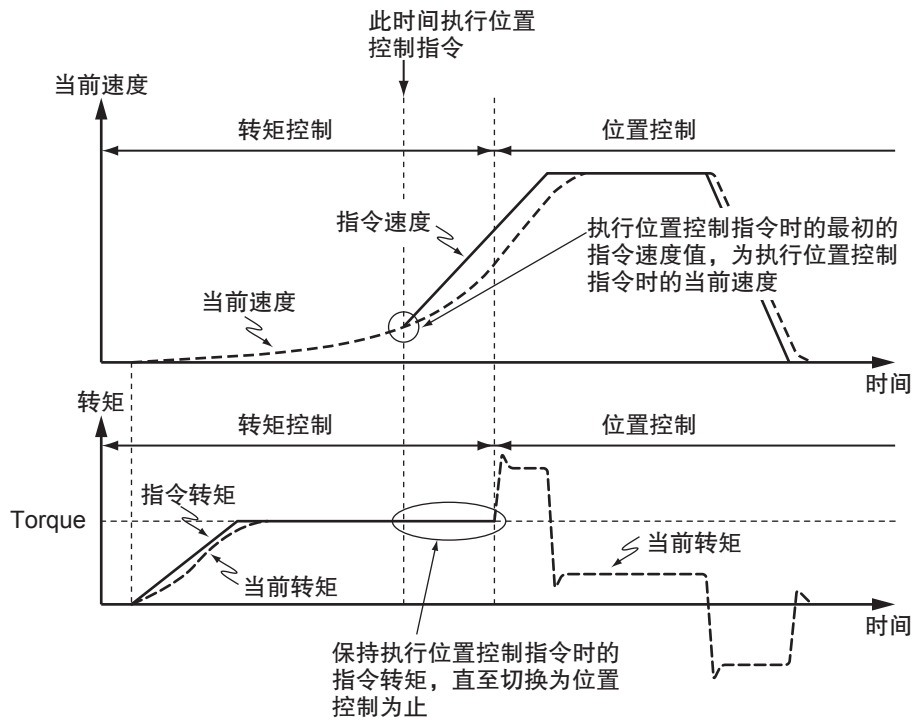
使用注意事项

轴动作时，如果在伺服驱动器侧切换控制模式，根据伺服驱动器的不同，可能发生异常。

从位置控制切换为转矩控制时



从转矩控制切换为位置控制时



重启运动指令

在转矩控制动作中变更输入参数，再次将Execute(启动)设为TRUE，可变更本指令的动作。

重启运动指令可变更的输入变量有Torque(目标转矩)、TorqueRamp(转矩斜度)、Velocity(限制速度)。

重启运动指令变更Torque(目标转矩)时，InTorque(达到目标转矩)针对重启新设定的目标转矩进行动作。

重启运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

多重启动运动指令

多重启动运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或

□□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动

对当前正在执行的指令启动本指令，可切换或缓存到本指令。

各轴可缓存到1个。

多重启动时的本指令的动作由BufferMode(缓存模式选择)指定。

缓存模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。
等待	当前正在执行的指令正常完成后，已缓存的本指令自动启动。

关于BufferMode(缓存模式选择)，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或

□□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

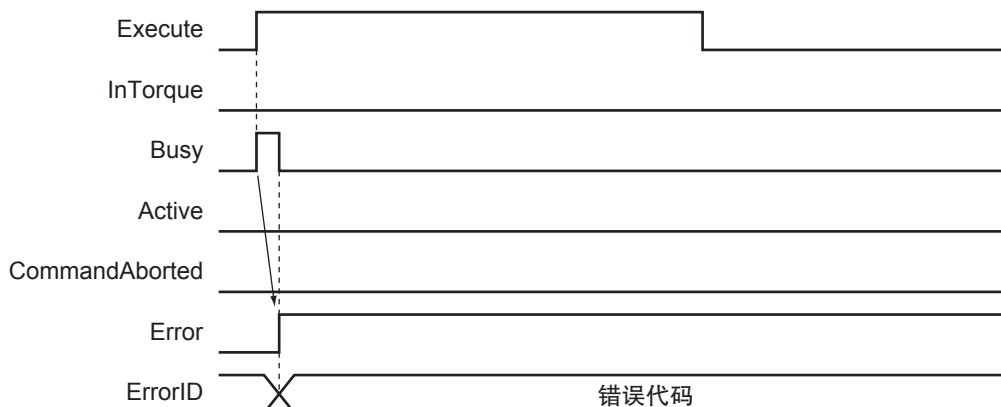
● 本指令执行中的其它指令启动

只有对其他指令的BufferMode(缓存模式选择)指定中断或等待时，才能在执行本指令时，利用其它指令多重启动。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。



● 错误代码

关于指令发生的异常，请参阅 □□ “A-1 错误代码一览(P.A-2)”。

MC_SetTorqueLimit

使用伺服驱动器的转矩限制功能，对伺服驱动器的输出转矩进行限制。



使用注意事项

- 无法对NX系列脉冲输出单元使用本指令。

指令	名称	FB/ FUN	图形表现	ST表现
MC_SetTorqueLimit	转矩限制	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期间执行指令。
PositiveEnable	正方向有效	BOOL	TRUE, FALSE	FALSE	TRUE: 将正方向的转矩限制设为有效。 FALSE: 将正方向的转矩限制设为无效。
PositiveValue	正方向转矩限制值	LREAL	0.1 ~ 1000.0、 或“0.0”	300.0	以0.1%为单位指定正方向的转矩限制值。 输入值超过轴参数的[正方向转矩限制上限值]时，使用[正方向转矩限制上限值]。 指定为“0”及“负数”时作为“0”动作。
NegativeEnable	负方向有效	BOOL	TRUE, FALSE	FALSE	TRUE: 将负方向的转矩限制设为有效。 FALSE: 将负方向的转矩限制设为无效。
NegativeValue	负方向转矩限制值	LREAL	0.1 ~ 1000.0、 或“0.0”	300.0	以0.1%为单位指定负方向的转矩限制值。 输入值超过轴参数的[负方向转矩限制上限值]时，使用[负方向转矩限制上限值]。 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	控制中变为TRUE。 ^{*1}
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为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为正常。

*1. Enabled(有效)表示本指令的状态，并不是表示伺服驱动器的转矩限制状态。

*2. □ 请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Enabled	<ul style="list-style-type: none"> • Enable变为TRUE时 • MC_Power指令执行中时 	<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含有异常因素时	异常已解除时

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. 请用在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

- 本指令对伺服驱动器设定转矩限制值。
- Enable(有效)为TRUE时，
如果将PositiveEnable(正方向有效)设为TRUE，则以PositiveValue(正方向转矩限制值)进行限制。
如果将NegativeEnable(负方向有效)设为TRUE，则以NegativeValue(负方向转矩限制值)进行限制。
- 将PositiveEnable(正方向有效)设为FALSE时，在伺服驱动器中设定轴参数的[正方向转矩限制上限值]。
同样，将NegativeEnable(负方向有效)设为FALSE时，在伺服驱动器中设定轴参数的[负方向转矩限制上限值]。
- 将本指令的Enable(有效)设为FALSE时，在伺服驱动器中设定[正方向转矩限制上限值][负方向转矩限制上限值]。同时将Busy(执行中)、Enabled(有效)设为FALSE。
- 可相对电机转矩，以0.1%为单位设定转矩限制值。指定值小数点后超过第2位时，第2位四舍五入。



使用注意事项

轴参数的[正方向转矩限制上限值][负方向转矩限制上限值]请设定为所用伺服驱动器中设定的转矩限制的上限值。

● 数据对象的映射

使用MC_SetTorqueLimit(转矩限制)指令时，请在Sysmac Studio的轴基本设置的[高级设置]中映射如下对象数据。

- 正方向转矩限制值(60E0 Hex)
- 负方向转矩限制值(60E1 Hex)

未进行这些设定时，将发生过程数据对象设定不足(错误代码：3461Hex)错误。



使用注意事项

- 使用其他公司生产的伺服驱动器时，请对伺服驱动器进行设定，确保正方向转矩限制值(60E0 Hex)、负方向转矩限制值(60E1 Hex)为转矩限制值。
关于设定方法，请参阅 “各伺服驱动器的手册”。
- 通过其他公司生产的伺服驱动器无法将正方向转矩限制值、负方向转矩限制值映射到PDO中时，不能使用本指令。
此时，请使用各公司的支持工具或SDO通信设定转矩限制值。

关于数据对象的映射，请参阅 “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-363)”或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 输入参数的改写

Enable(有效)为TRUE时，可以经常变更下列输入变量的值。

- PositiveEnable(正方向有效)
- NegativeEnable(负方向有效)
- PositiveValue(正方向转矩限制值)
- NegativeValue(负方向转矩限制值)

● 与MC_Home(原点复位)指令、MC_HomeWithParameter(参数指定原点复位)指令的接触动作的关系 (使用欧姆龙制伺服驱动器1S系列或G5系列时)

选择原点复位的[12: 指定为附近避让、接触时间]、[13: 指定为无近原点输入、接触原点输入]时，使用伺服驱动器事先设定的转矩限制值，自动向接触方向开始转矩限制。



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)指令的自动转矩限制功能不起作用。

请使用MC_SetTorqueLimit(转矩限制)指令、SDO通信、伺服驱动器的支持工具等，设定合适值。



参考

- 原点复位正常完成后也将继续转矩限制。
- 启动向反方向移动的指令后，转矩限制自动解除。
原点复位的详情请参阅 [□](#) “MC_Home(P.3-15)” 或 [□](#) “MC_HomeWithParameter(P.3-37)”。

● 欧姆龙制伺服驱动器 1S系列的设定

使用本指令时，必须利用Sysmac Studio设定1S系列伺服驱动器的[转矩限制]的[切换选择](3330Hex-01Hex)。

- 在零点复位的接触动作时，在零点检测方向上进行转矩限制。对于其他动作，在使用MC_SetTorqueLimit(转矩限制)指令的转矩限制方向和转矩限制值时，选择“2”。
这种情况下，在零点复位的接触动作中，MC_SetTorqueLimit(转矩限制)指令输入变量的内容将被忽略。
- 设定为“0”时，始终采用MC_SetTorqueLimit(转矩限制)指令输入变量的内容。零点复位的接触动作时和其他动作时都必须设定合适的转矩限制值。

		[转矩限制]的[切换选择](3330Hex-01Hex)	
		2	0
正方向转矩限制值	原点复位中	采用伺服驱动器的[转矩限制]的[正方向转矩限制值2](3330Hex-05Hex)。	采用MC_SetTorqueLimit(转矩限制)指令的PositiveValue(正方向转矩限制值)。
	非原点复位	采用MC_SetTorqueLimit(转矩限制)指令的PositiveValue(正方向转矩限制值)。	
负方向转矩限制值	原点复位中	采用伺服驱动器的[转矩限制]的[负方向转矩限制值2](3330Hex-06Hex)。	采用MC_SetTorqueLimit(转矩限制)指令的NegativeValue(负方向转矩限制值)。
	非原点复位	采用MC_SetTorqueLimit(转矩限制)指令的NegativeValue(负方向转矩限制值)。	

详情请参阅 □ “AC伺服电机/驱动器1S系列EtherCAT通信内置型用户手册(SBCE-377)”。

● 欧姆龙制伺服驱动器 G5系列的设定

使用本指令时，必须利用伺服驱动器的支持软件，设定伺服驱动器的转矩限制选择(3521 Hex)。

- 在零点复位的接触动作时，在零点检测方向上进行转矩限制。对于其他动作，在使用本指令的转矩限制方向或转矩限制值时，选择“6”。
此时，在零点复位的接触动作中，本指令输入变量的内容忽略。
- 设定为“4”时，通常使用本指令输入变量的内容。零点复位的接触动作时和其他动作时都必须设定合适的转矩限制值。

		转矩限制选择(3521 Hex)	
		6(推荐)	4
正方向转矩限制值	原点复位中 *1	采用第3转矩限制(3525 Hex)。	采用本指令的PositiveValue(正方向转矩限制值)与第1转矩限制(3013 Hex)中的较小值。
	非原点复位	采用本指令的PositiveValue(正方向转矩限制值)与第1转矩限制(3013 Hex)中的较小值。	
负方向转矩限制值	原点复位中 *1	采用第4转矩限制(3526 Hex)。	采用本指令的NegativeValue(负方向转矩限制值)与第2转矩限制(3522 Hex)中的较小值。
	非原点复位	采用本指令的NegativeValue(负方向转矩限制值)与第2转矩限制(3522 Hex)中的较小值。	

*1. 自动解除转矩限制前

详情请参阅 □ “AC伺服电机/驱动器 G5系列 EtherCAT通信内置型 用户手册(SBCE-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-366)”。

● 与MC_TorqueControl(转矩控制)指令的关系

MC_SetTorqueLimit(转矩限制)指令和MC_TorqueControl(转矩控制)指令可组合使用。

● 对轴组中轴的使用

对于有效轴组中的轴，可使用本指令。

● 与CPU单元动作模式的关系

在运行模式下通过本指令设定的内容，即使切换为程序模式也会保持。

● 可使用轴和启动条件

- 伺服轴可以使用本指令。

但是，本指令的输出变量“Enabled(有效)”的状态因伺服状态而异。

	伺服ON状态	伺服OFF状态
Enabled(有效)的状态	TRUE	FALSE *1

*1. 本指令的Enabled(有效)为FALSE时，伺服驱动器的转矩限制不起作用。

- 虚拟伺服轴也可接收指令，但不执行转矩限制。
- 编码器轴、虚拟编码器轴则会发生启动时异常。

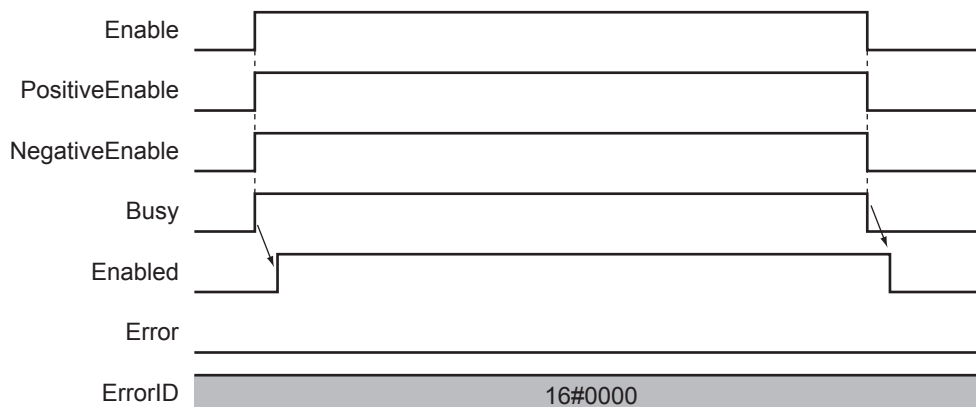
● 轴变量的状态(伺服驱动器状态)

伺服驱动器的内部限制功能启动时，轴变量的DrvStatus(伺服驱动器状态)的ILA(驱动器内部功能限制中)变为TRUE。

该变量有伺服驱动器的转矩限制、速度限制、驱动禁止输入、软件限制等4大OR输出。

时序图

执行转矩限制时的时序图如下所示。



重启运动指令

输入为“Enable型”指令时，不会重启运动指令。

多重启动运动指令

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 多重启动多个本指令

对本指令执行中的轴启动本指令的其它实例时，后执行的实例被优先处理。

Enabled(有效)对两种指令均表现为TRUE。

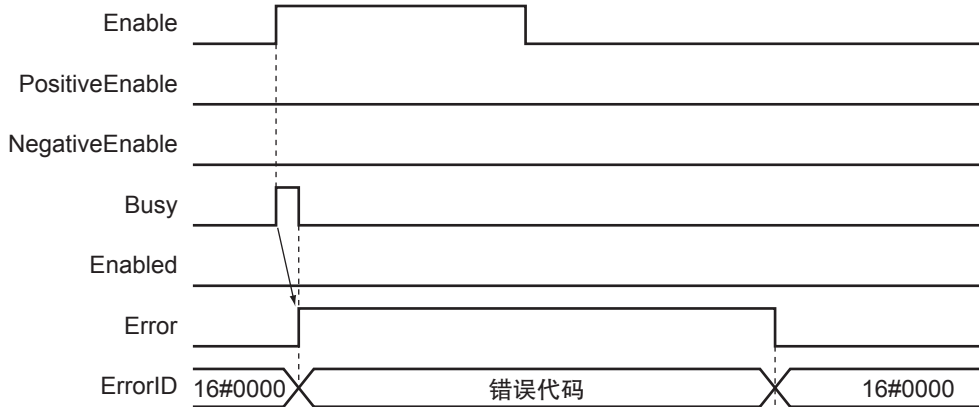
作为具体动作，后执行的实例的转矩限制值生效。将后执行的实例的Enable(有效)设为FALSE后，转矩限制无效。

异常

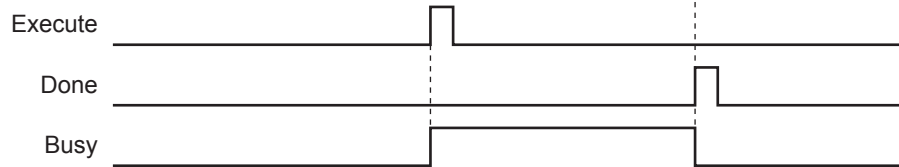
在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图

MC_SetTorqueLimit指令



MC_Reset指令



● 错误代码

关于指令发生的异常，请参阅 “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在指定范围内。

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期间执行指令。
FirstPosition	起始位置	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	终止位置	LREAL	负数、正数、 “0”	0	指定区域范围的终止位置。 ^{*3} 单位为[指令单位]。 ^{*1}
ReferenceType	位置类型选择	_eMC_ REFERENCE_ TYPE	0: _mcCommand 1: _mcFeedback	0 ^{*4}	指定要监视的轴信息。 0: 指令位置(最近任务周期 ^{*5} 下的计算值) 1: 反馈位置(同一任务周期 ^{*5} 下的取值)

*1. 请指定比终止位置小的值。

*2. 关于指令单位，请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*3. 请指定比开始位置大的值。

*4. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

*5. 任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先度5)的周期。

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	接收指令后变为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为正常。

* □ 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Enabled	Enable变为TRUE时	Enable变为FALSE时
InZone	在区域内时	<ul style="list-style-type: none"> 在区域外时 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含有异常因素时	异常已解除时

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. 请在使用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

- 本指令处于Enable(有效)的状态下，轴的指令位置、或反馈当前位置在FirstPosition(起始位置)以上、LastPosition(终止位置)以下的区域内时，输出变量的InZone(区域内)变为TRUE。
通过RerenceType(位置类型选择)，可将监视对象的轴信息设定为指令位置或反馈位置。
- 可对所有轴种类进行区域监视。
- Enable(有效)为TRUE的状态下，改写FirstPosition(起始位置)或LastPosition(终止位置)时，从改写后的周期开始采用新的数值。
- 可以对1根轴设定多个区域，即使区域重复也可设定。并且，在软件限制范围外也可设定区域。



使用注意事项

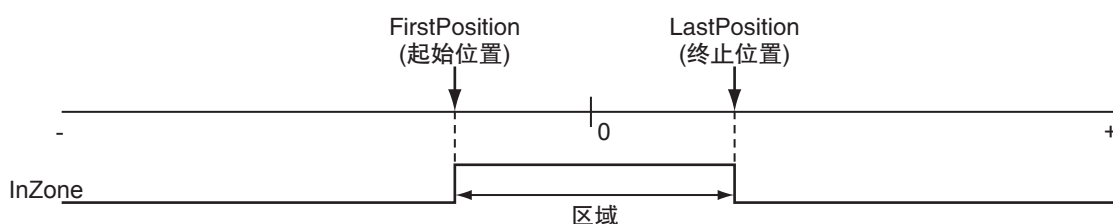
FirstPosition(起始位置)、LastPosition(终止位置)含有除不尽的除法结果等, 由于误差的原因, 处理结果可能会出现意外。

指令详情

设定FirstPosition(起始位置)和LastPosition(终止位置)时, 根据计数模式的不同, 请确保如下条件式成立。条件式不成立时, 会发生异常。

● 线性模式

指定时请确保FirstPosition(起始位置) ≤ LastPosition(终止位置)。

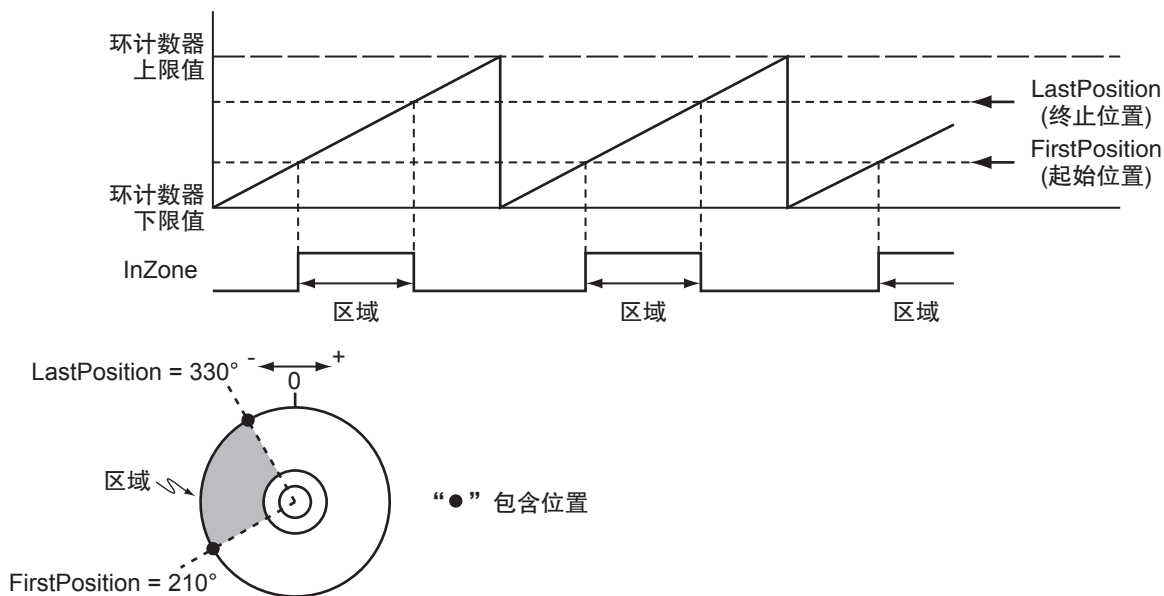


● 旋转模式

[旋转模式]下, 根据是否跨越环计数器的上下限位置, 存在如下不同。

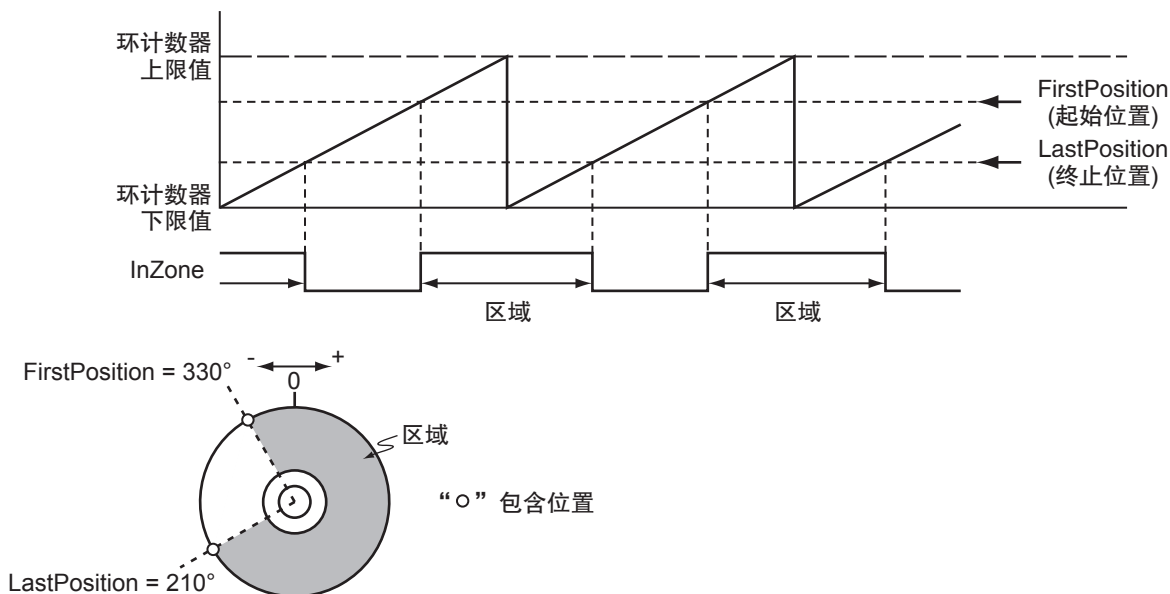
跨越上下限位置时

指定时请确保FirstPosition(起始位置) ≤ LastPosition(终止位置)。



跨越上下限位置时

指定时请确保FirstPosition(起始位置)>LastPosition(终止位置)。



● ReferenceType(位置类型选择)

可从下列选项中选择位置的类型。

- `_mcCommand`: 指令位置(最近任务周期下的计算值)
对当前周期, 使用之前任务周期时计算的主轴指令位置。
在计算从轴指令位置之前的固定周期任务中, 使用计算的主轴指令位置。
- `_mcFeedback`: 同一任务周期下的取值
使用同一任务周期时获取的主轴反馈位置。



使用注意事项

这里的任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先度5)的周期。同样, 固定周期任务表示原始恒定周期任务或固定周期任务(执行优先度5)。

● 轴种类与位置类型的关系

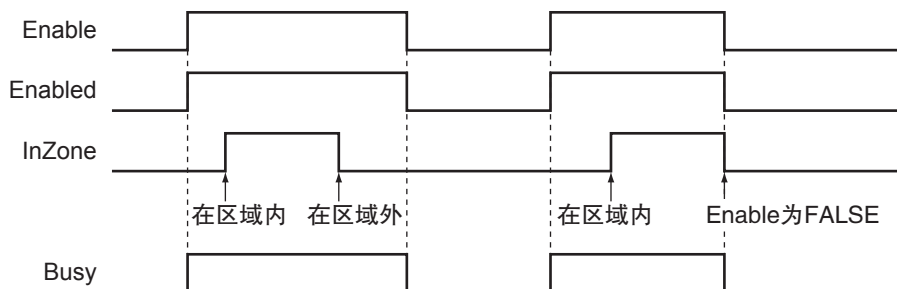
可监控的轴种类和要监控的位置类型的关系如下所示。

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. 启动指令时, 发生“超过位置类型选择范围(错误代码: 5430 Hex)”错误。

时序图

- 动作中进入区域内时、以及在区域内Enable(有效)变为FALSE时



- 启动指令前位于区域内，中途转到区域外时



重启运动指令

输入为“Enable型”指令时，不会重启运动指令。

多重启动运动指令

本指令的各个指令独立动作，没有运动指令多重启动的限制。

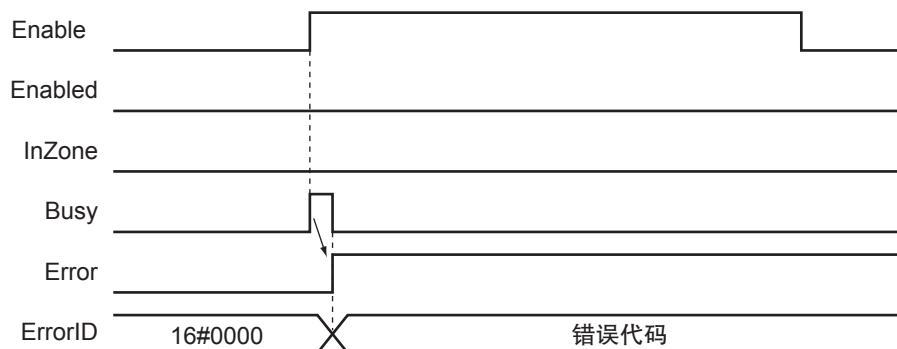
多重启动运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或

□□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

异常

未能启动本指令时，本指令会发生异常，Error(错误)变为TRUE。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 □ “A-1 错误代码一览(P.A-2)”。

MC_TouchProbe

根据触发信号的发生记录轴的位置。

指令	名称	FB/ FUN	图形表现	ST表现
MC_TouchProbe	启用外部锁定	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	在上升沿开始指令。
WindowOnly	窗口有效	BOOL	TRUE, FALSE	FALSE	指定窗口掩码有效/无效。
FirstPosition	起始位置	LREAL	负数、正数、“0”	0	指定接收触发的开始位置。使用通过ReferenceType(位置类型选择)指定的种类。 ^{*1} 单位为[指令单位]。 ^{*2}
LastPosition	终止位置	LREAL	负数、正数、“0”	0	指定接收触发的结束位置。使用通过ReferenceType(位置类型选择)指定的种类。 ^{*1} 单位为[指令单位]。 ^{*2}
ReferenceType (Reserved)	位置类型选择	_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. 详情请参阅 “WindowOnly(窗口有效)(P.3-335)”。

*2. 关于指令单位，请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*3. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为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	指令中止时，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为正常。

*1. 关于指令单位，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的 “单位转换设定”。

*2. □ 请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	<ul style="list-style-type: none"> 发生触发信号后，记录锁定位置，完成指令时 停止指定有效时，完成指令并且轴停止在锁定位置时 	<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"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 将StopMode设为[_mcImmediateStop]并执行时，变更为非CSP模式时 *1 从站脱离时 发生从站通信异常时(过程数据通信中除外) MC_AbortTrigger指令启动时 	<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含有异常因素时	异常已解除时

*1. 使用NX系列CPU单元时，不受控制模式的限制，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	在触发条件下指定控制器模式时，指定触发的输入变量。

*1. 请使用在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型的用户定义变量。

● _sTRIGGER_REF

结构要素变量	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	在驱动器模式下，指定使用2个锁定功能中的哪一个。 0: 锁定功能1 1: 锁定功能2
InputDrive	驱动触发 输入信号	_eMC_TRIGGER_INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	在驱动器模式下指定驱动器的触发信号。 0: Z相 1: 外部输入

功能说明

- 为了实现以传感器输入等发生触发为基点进行位置控制，发生触发时记录(锁定)轴位置。
- 根据触发设定，将通过Axis(轴)指定的轴的位置输出到RecordedPosition(锁定位置)。
作为触发设定，可指定TriggerInput(触发输入条件)、WindowOnly(窗口有效)、FirstPosition(起始位置)、LastPosition(终止位置)、StopMode(停止方法选择)。
- RecordedPosition(锁定位置)的输出值保持不变，直至在MC_TouchProbe(启用外部锁定)指令的实例中再次记录轴位置。

● 数据对象的映射

Mode选择[驱动器模式]，使用MC_TouchProbe(启用外部锁定)指令时，请映射如下对象数据。
请在Sysmac Studio的轴基本设置的[高级设置]中进行映射。

轴类型为伺服轴时

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

轴类别为编码器轴时

- 锁定功能(4020Hex)
- 编码器输入从站软开关(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)

未进行这些设定时，将发生过程数据对象设定不足(错误代码：3461Hex)错误。

关于数据对象的映射，请参阅 □ “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-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。使用NX系列位置接口单元时，请参阅 □ “NX系列 位置接口单元 用户手册(SBCE-374)” 中的 “I/O入口映射”。使用编码器输入从站时，请参阅 □ “GX系列EtherCAT从站 用户手册 (SBCE-350)” 的 “固定PDO映射”。

指令详情

下面对本指令进行详细说明。

● Axis(轴)指定

- 通过Axis(轴)指定锁定对象轴。
- 指定Axis(轴)在执行MC_GroupEnable(多轴协调控制)指令的过程中，MC_TouchProbe(启用外部锁定)指令将发生异常而无法执行。
- 可以针对每根轴指定LatchID(锁定ID选择)，同时执行2个以下的MC_TouchProbe(启用外部锁定)指令。
- 通过MC_AbortTrigger(不启用外部锁定)指令指定要中止的锁定时，也可使用LatchID(锁定ID选择)。



参考

- 使用欧姆龙制 GX系列 EtherCAT 编码器输入从站 GX-EC02□□上连接的编码器轴，也可通过本指令进行锁定。
- 使用NX系列脉冲输出单元，也可通过本指令进行锁定。详情请参阅 □ “NX系列 位置接口单元 用户手册(SBCE-374)”。

● 触发输入条件

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

Mode

- Mode的设定有两种可选，在触发中指定伺服驱动器等发出信号的“驱动器模式”、在触发中指定Trigger Variable(触发变量)的“控制器模式”。
- 将触发信号的上升沿视为发生触发，对执行锁定指令后最初发生触发(FALSE→TRUE)时轴的位置进行记录。
- 本指令为Busy(动作中)中时，即使Execute(启动)变为FALSE，也会将TriggerVariable(触发变量)的变化作为触发进行动作。

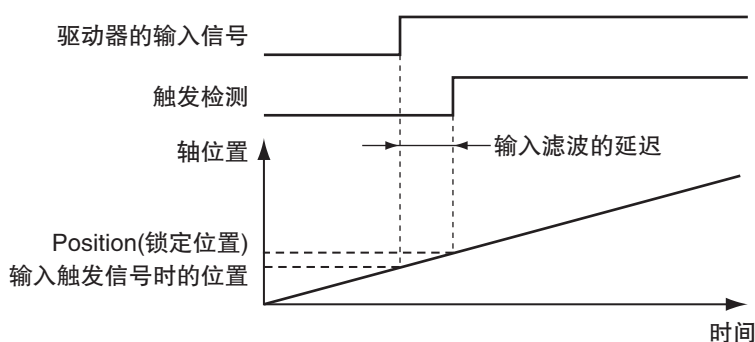


参考

使用欧姆龙制 GX系列 EtherCAT 编码器输入从站 GX-EC02□□时，请将Mode设定为[_mcDrive (驱动模式)]。

驱动器模式

“驱动器模式”是利用伺服驱动器等的功能检测触发、获取当前位置，因此，相比“控制器模式”，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等的性能等。



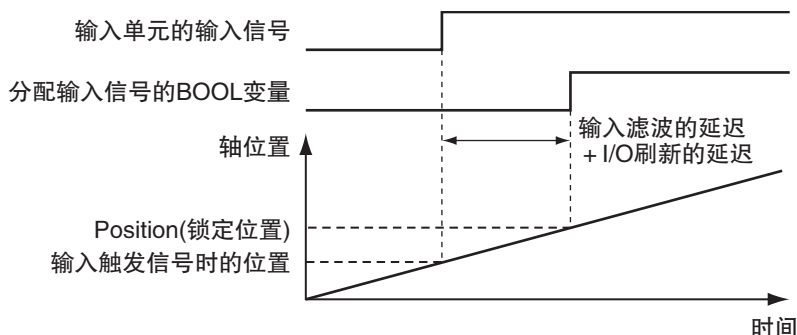
参考

使用欧姆龙制 GX系列 EtherCAT 编码器输入从站 GX-EC02□□时，请在InputDrive中指定 _mcEXT(外部输入)。

欧姆龙制 GX系列 EtherCAT 编码器输入从站 GX-EC02□□的Z相不具备锁定功能。因此，指定 _mcEncoderMark(Z相)时，启动本指令将发生异常。启动时，Error(错误)变为TRUE，ErrorID(错误代码)输出“过程数据对象设定不足(错误代码：3461 Hex)”。

控制器模式

- 在“控制器模式”下，可对触发指定BOOL变量。
- 在TriggerVariable(触发变量)中指定要触发的BOOL变量。
- 与“驱动器模式”相比，在“控制器模式”下，由于I/O刷新的延迟，触发输入信号的BOOL变量的反映会产生较大延迟。



使用注意事项

使用控制器模式时，按照任务周期的间隔进行锁定处理。因此，触发变量必须在任务周期的一个周期内为TRUE。

并且，在触发变量变为TRUE后，MC功能模块执行处理之前，需要任务周期的一个周期的时间。这里的任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先度5)的周期。

LatchID

- 可以对1根轴同时执行2个以下的MC_TouchProbe(启用外部锁定)指令。通过LatchID(锁定ID选择)指定使用哪一个。
- 对于同一根轴如果指定的锁定ID已在执行中，则后执行的指令有效。之前的指令中断执行(Command Aborted)。
- 伺服驱动器等中，LatchID(锁定ID选择)表示伺服驱动器的锁定电路1(第1ch)、锁定电路2(第2ch)。

关于LatchID，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。



参考

使用欧姆龙制 GX系列 EtherCAT 编码器输入从站 GX-EC02□□时，采用如下对应措施。

枚举元素	编码器输入终端的信号名
_mcLatch1	外部锁定输入A
_mcLatch2	外部锁定输入B

InputDrive

- 作为触发，可选择_mcEncoderMark(Z相)或_mcEXT(外部输入)。
- _mcEncoderMark(Z相)中，将伺服驱动器等的Z相用作触发。
_mcEXT(外部输入)中，将伺服驱动器中输入的外部触发信号用作触发。
- 欧姆龙制伺服驱动器 1S 系列中，_mcEXT(外部输入)为设定的Ext1 或 Ext2。G5 系列中，为设定的Ext1、Ext2、Ext3三者之一。通过Sysmac Studio进行设定。
驱动器中设定的两个触发可以相同。

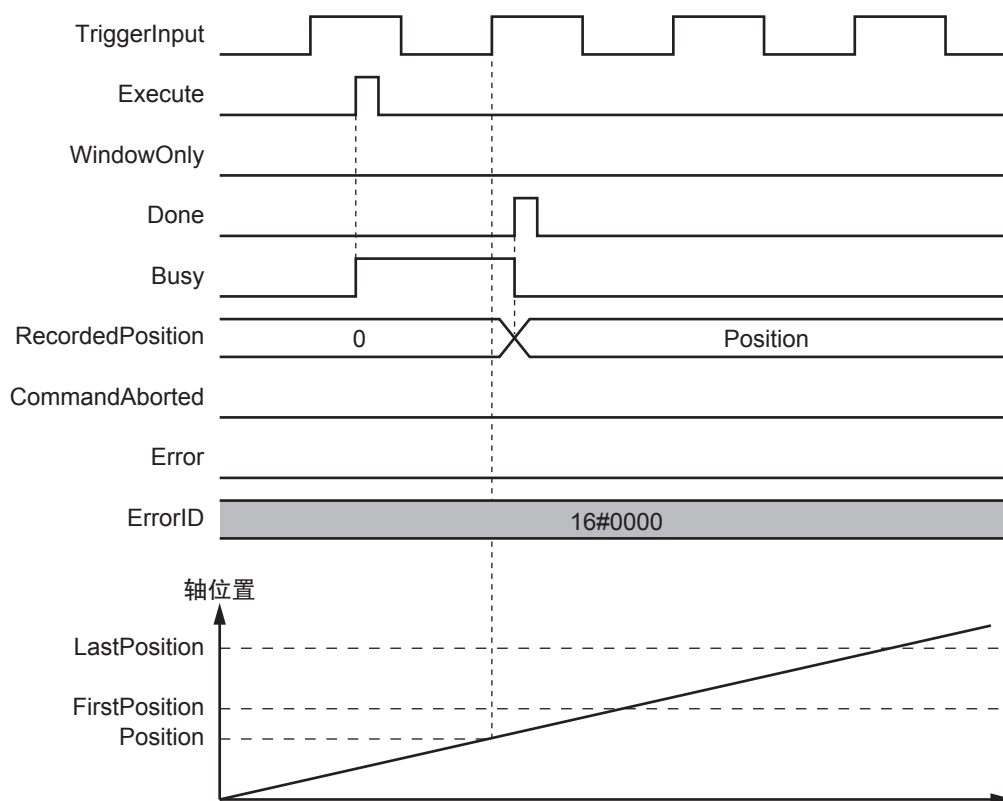
● WindowOnly(窗口有效)

- 在WindowOnly(窗口有效)中，指定窗口的有效/无效。
- 指定Disable(无效)时，在所有轴位置检出触发。
- 指定Enable(有效)时，仅轴位置在FirstPosition(起始位置)和LastPosition(终止位置)的范围内时检出触发。

WindowOnly(窗口有效)的指定不同，动作也不同。如时序图所示。

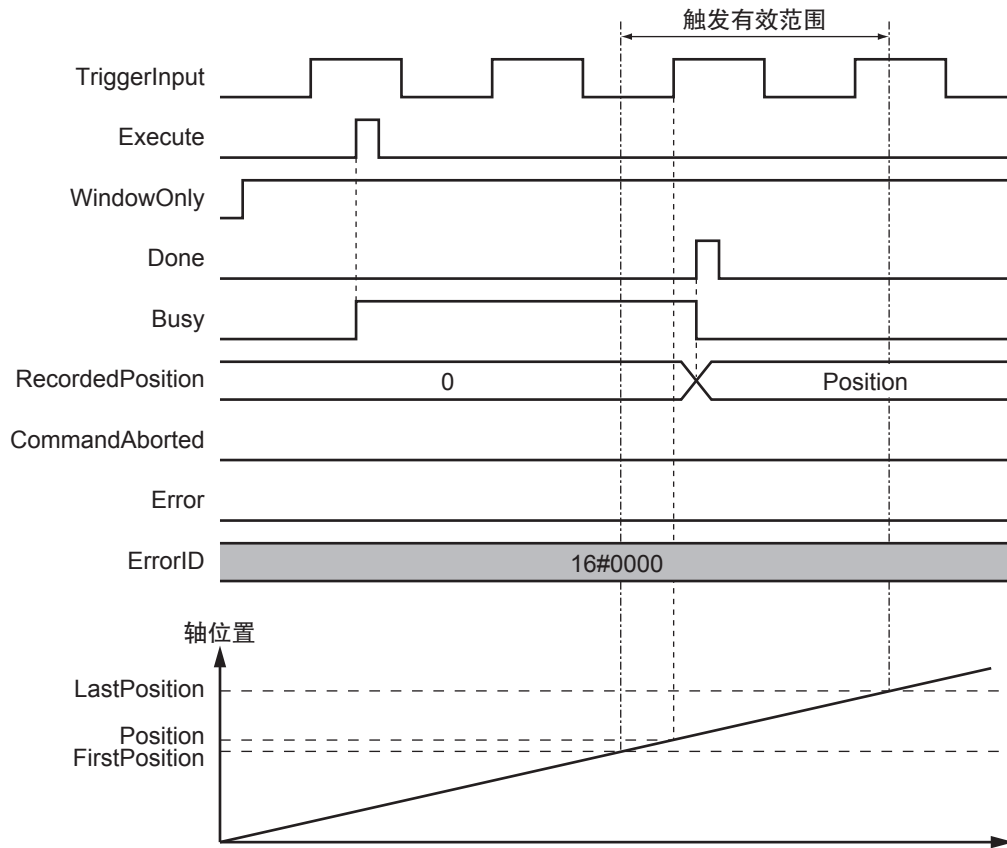
WindowOnly为Disable时

将Execute(启动变为TRUE后最初发生触发时的轴位置输出到RecordedPosition(锁定位置)。



WindowOnly为Enable时

仅在窗口的范围内检出触发输入，获取轴位置。



使用注意事项

- WindowOnly(窗口有效)由FALSE变化为TRUE的瞬间以及锁定功能启动之间的时间无法锁定。
- 锁定功能启动需要时间，因此，WindowOnly(窗口有效)的有效范围极短时无法锁定。可锁定有效范围的临界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和EtherCAT的通信。

不同计数模式的FirstPosition(起始位置)和LastPosition(终止位置)的范围如下所示。

线性模式

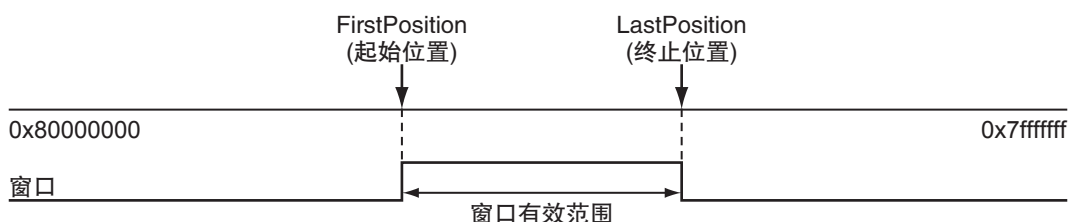
- 窗口的有效范围如下所示。

FirstPosition(起始位置) ≤ 窗口范围 ≤ LastPosition(终止位置)

- 指定FirstPosition(起始位置) > LastPosition(终止位置)时, 会发生异常。
- 超过[线性模式]的位置范围指定时, 也会发生异常。
- 此外, 由于FirstPosition(起始位置)和LastPosition(终止位置)为LREAL型变量, 因此, 请勿指定FirstPosition(起始位置) = LastPosition(终止位置)。

关于LREAL型, 请参阅 □ “实数(REAL型、LREAL型)的使用(P.1-14)”。

[线性模式]时的窗口有效范围如下图所示。

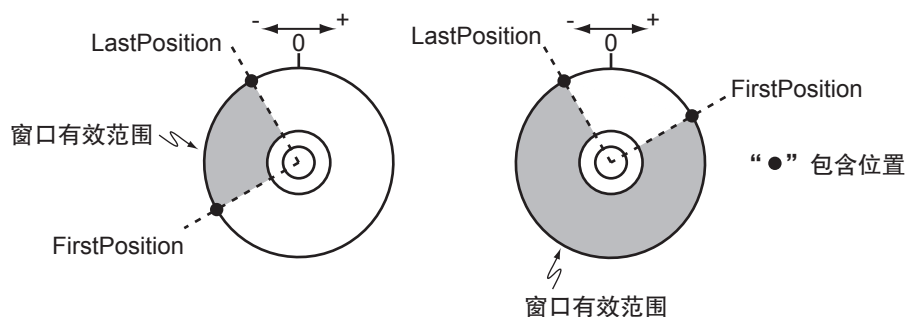


(注) FirstPosition(起始位置)和LastPosition(终止位置)包含在窗口有效范围中。

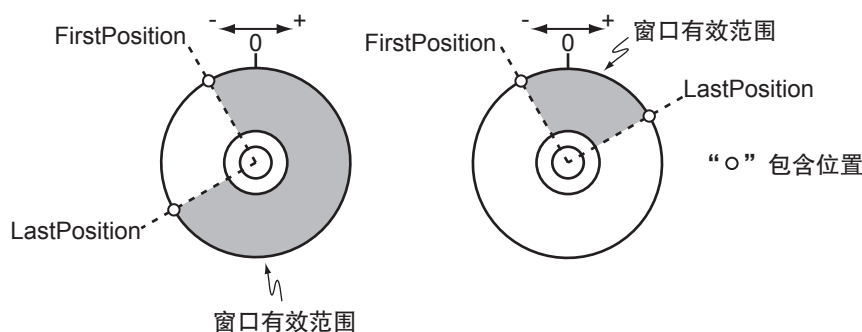
旋转模式

- FirstPosition(起始位置) ≤ LastPosition(终止位置)和FirstPosition(起始位置) > LastPosition(终止位置)两者均可指定。
- FirstPosition(起始位置) > LastPosition(终止位置)时, 设定值跨越环计数器的上下限位置。
- 超过环计数器上下限范围指定时, 会发生异常。

FirstPosition ≤ LastPosition



FirstPosition > 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)。
- 即使发生触发，_mcNonStop(非停止)也无法停止轴动作。
- _mcImmediateStop(立即停止)可在发生触发时使轴立即停止在锁定位置。通过本停止使该轴动作的指令为CommandAborted(执行中断)。
- 在停止轴指令时，_mcImmediateStop(立即停止)变为Done(完成)。Busy(执行中)变为TRUE，直至停止在锁定位置。
- 通过_mcImmediateStop(立即停止)停止时，不执行到位检查。

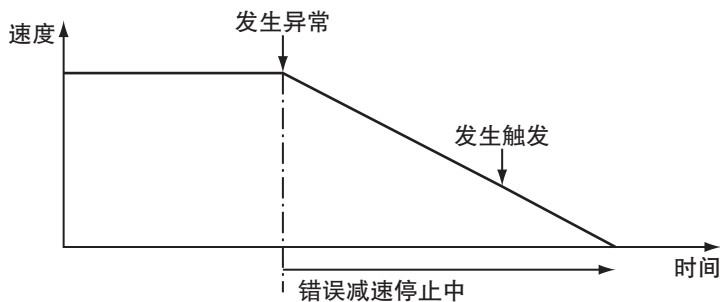


参考

- 使用欧姆龙伺服驱动器1S系列时，与CPU单元版本无关，动作与如下所述Ver.1.10以上版本相同。
- 使用欧姆龙伺服驱动器G5系列时，根据CPU单元版本，动作如下所述有所不同。

Ver.1.10以上版本的CPU单元时

- 指定_mcImmediateStop(立即停止)的轴发生轴异常，轴停止完成之前发生触发时，轴将继续减速停止。而且，发生触发会导致CommandAborted(执行中断)。

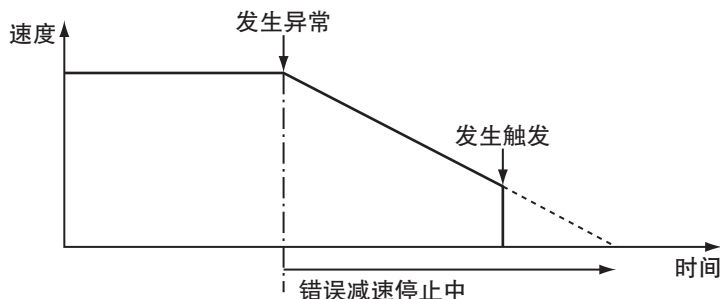


使用注意事项

- _mcImmediateStop(立即停止)由MC功能模块进行停止处理。实际表现为：超过锁定位置停止动作，根据控制器的指示返回锁定位置。敬请注意。
- 使用_mcImmediateStop(立即停止)时，指令速度越快，返回锁定位置的量越大。设定时请注意避免指令速度过快。
- 请对编码器轴指定_mcNonStop(非停止)。指定_mcImmediateStop(立即停止)后，启动本指令时Error(错误)变为TRUE。同时，ErrorID(错误代码)输出“无法执行启用外部锁定指令(错误代码：5492 Hex)”。

Ver.1.09以下版本的CPU单元时

- 指定_mcImmediateStop(立即停止)的轴发生轴异常，轴停止完成之前发生触发时，通过欧姆龙制伺服驱动器G5系列的功能使轴立即停止。而且，发生触发会导致CommandAborted(执行中断)。
- 运动控制功能模块在检测到欧姆龙制伺服驱动器G5系列的停止后，使停止指令立即停止。下图中的虚线为未发生触发的目标轨道。



使用注意事项

- _mcImmediateStop(立即停止)在CSP模式时启动。
- 在CSV/CST模式状态下，指定_mcImmediateStop(立即停止)并启动时，将发生“无法执行启用外部锁定指令(错误代码：5492Hex)”。在通过_mcImmediateStop(立即停止)动作中变更为CSV/CST模式时，变为CommandAborted(执行中断)。
- _mcImmediateStop(立即停止)在使用欧姆龙制伺服驱动器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进行停止。
使用欧姆龙制伺服驱动器1S系列及其他公司生产的伺服驱动器时，MC功能模块进行停止处理。
任一种情况下，实际表现为：超过锁定位置停止动作，根据控制器的指示返回锁定位置。敬请注意。
- 使用_mcImmediateStop(立即停止)时，指令速度越快，返回锁定位置的量越大。
设定时请注意避免指令速度过快。
- 请对编码器轴指定_mcNonStop(非停止)。指定_mcImmediateStop(立即停止)后，启动本指令时Error(错误)变为TRUE。同时，ErrorID(错误代码)输出“无法执行启用外部锁定指令(错误代码：5492 Hex)”。
- 使用_mcImmediateStop(立即停止)时，如果在同一控制周期执行如下两种处理，则欧姆龙制伺服驱动器G5系列变为Target Ignore状态。
 - 锁定信号ON时，伺服驱动器的计数器锁定处理
 - 执行MC_TouchProbe(启用外部锁定)指令后，CommandAborted(执行中断)变为TRUE时的解锁处理
 锁定信号变为ON时，请注意避免使本指令的CommandAborted(执行中断)变为TRUE。

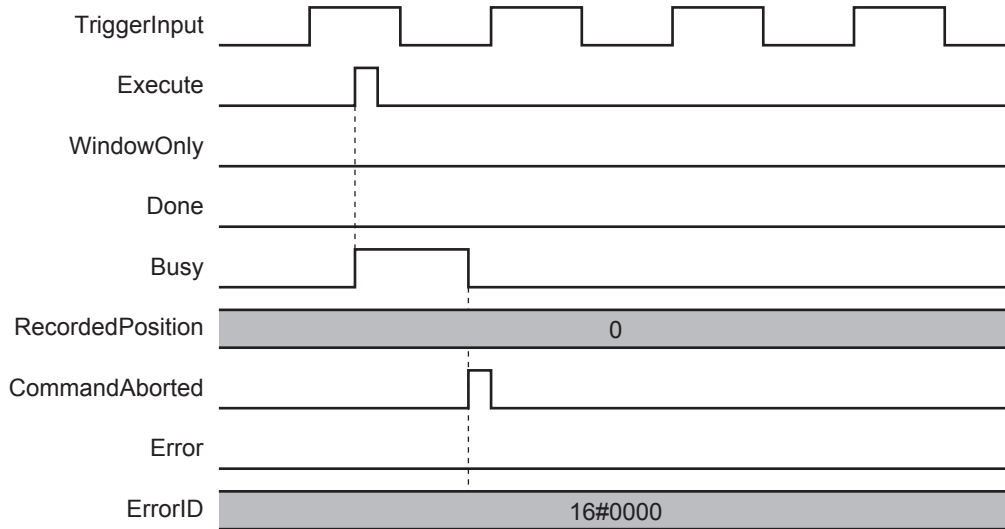
● 轴变量的状态

StopMode(停止方法选择)指定_mcImmediateStop(立即停止)时，发生触发而处于停止处理中时，轴变量Status.Stopping(减速停止中)变为TRUE。

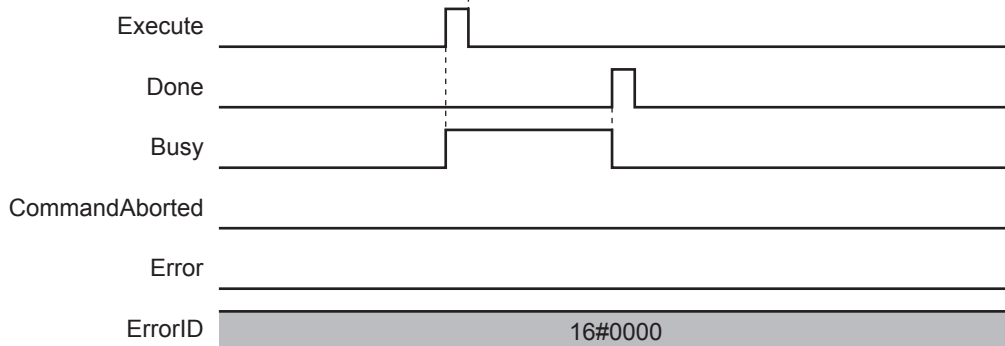
指令的中止

可通过MC_AbortTrigger(不启用外部锁定)指令结束MC_TouchProbe(启用外部锁定)指令。
对要中止的Axis(轴)和LatchID(锁定ID选择)指定为MC_AbortTrigger(不启用外部锁定), 启动指令, 结束动作。

MC_TouchProbe指令



MC_AbortTrigger指令

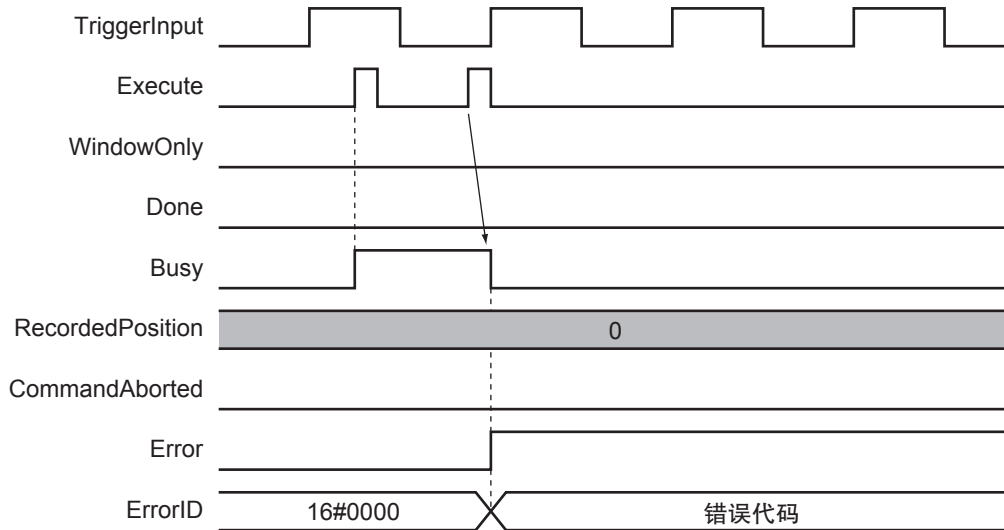


重启运动指令

无法重启本指令。

重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

在对相同MC_TouchProbe(启用外部锁定)指令的实例，读取RecordedPosition(锁定位置)完成之前，再次将Execute(启动)由FALSE → TRUE时，会发生异常。



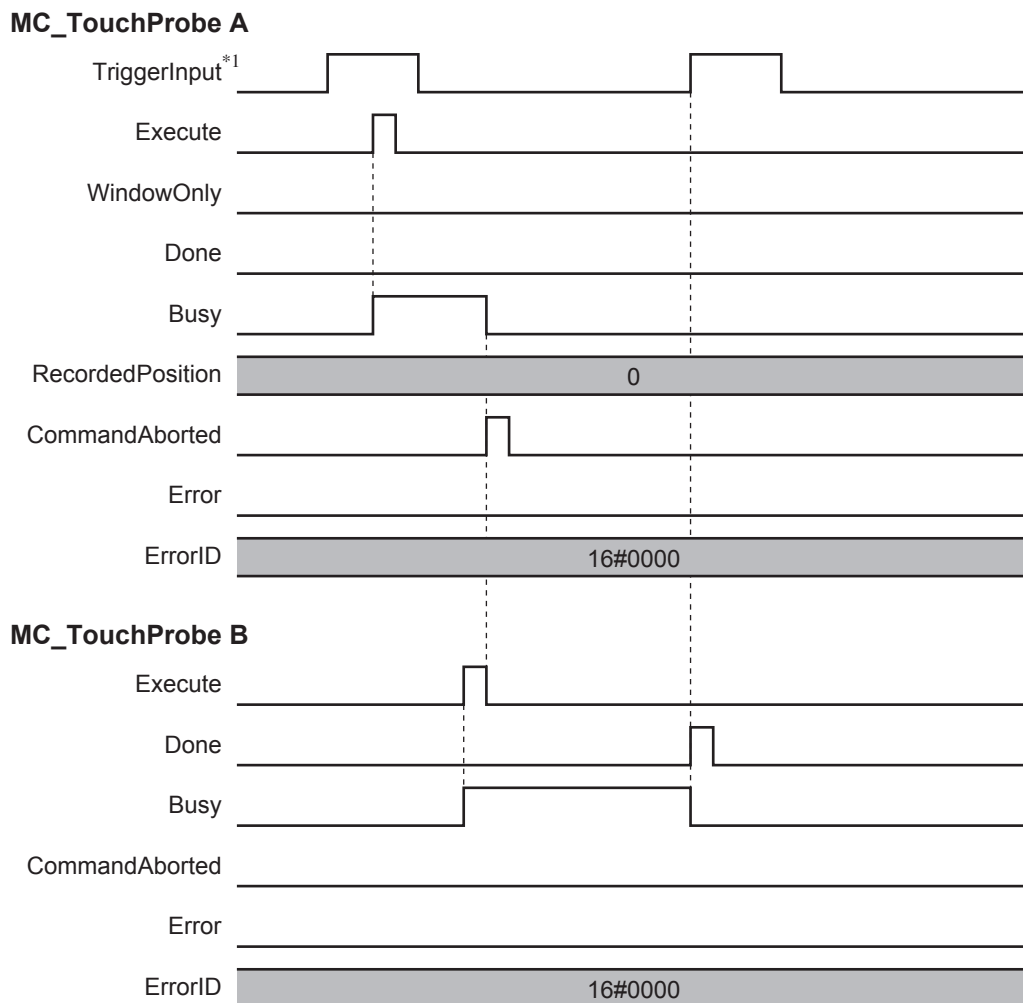
多重启动运动指令

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动

只能对相同Axis(轴)的1个LatchID(锁定ID选择)同时设定1个触发并执行。

在MC_TouchProbe(启用外部锁定)指令执行中，对相同LatchID(锁定ID选择)执行别的MC_TouchProbe(启用外部锁定)指令的实例时，第一个中断执行(CommandAborted)，第二个全新执行。



*1 伺服驱动器等触发输入信号时的情形。



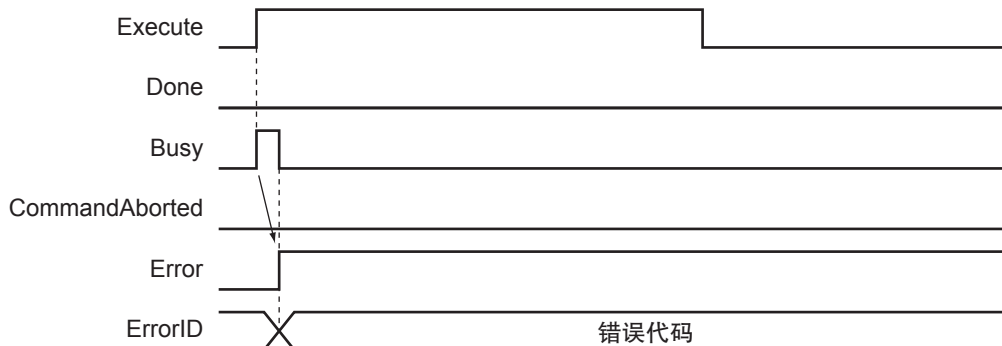
参考

StopMode(停止方法选择)指定_mcImmediateStop(立即停止)时，发生触发而停止时，多重启动的指令的CommandAborted(执行中断)变为TRUE。

异常

执行MC_TouchProbe(启用外部锁定)指令时若发生异常，Error(错误)变为TRUE。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 □□ “A-1 错误代码一览(P.A-2)”。

示例程序

下面，对基于外部传感器的位置取出控制的示例程序进行说明。

参数设定

该示例程序中使用的最少的必要设定如下所示。

● 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

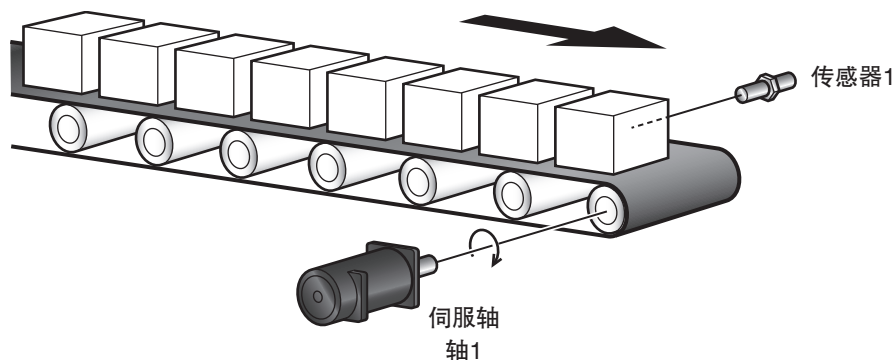
环计数器

轴	上限值	下限值
轴1	360	0

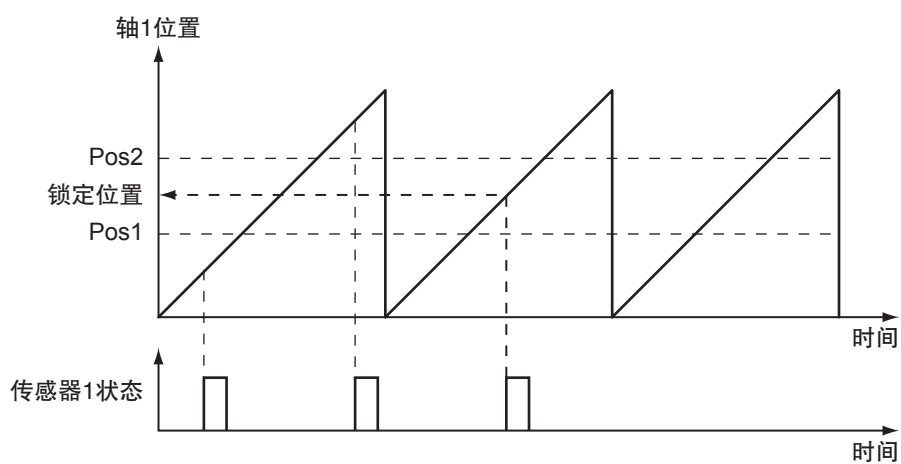
坐标单位选择

轴	坐标单位
轴1	degree

动作示例



● 动作模式



- 1** 主轴的开始
轴1进行速度控制。
- 2** 工件的检测
通过传感器1检测工件的有无。
- 3** 位置的获取
在窗口的TRUE的范围内(Pos1 ~ Pos2)检出工件后，获取该时轴1的锁定位置。

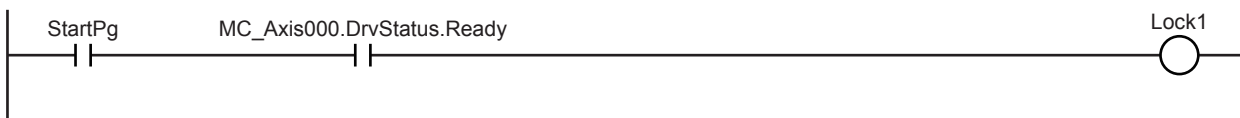
梯形图

● 主要变量

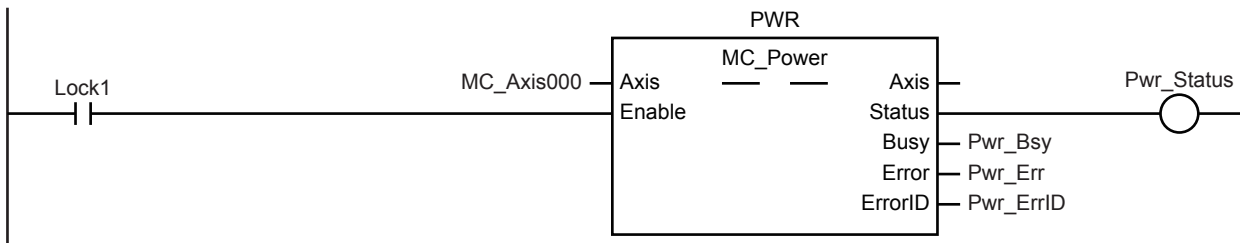
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	轴1的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴1发生轻度故障等级的异常时变为TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴1为原点确定状态时变为TRUE。
Pwr_Status	BOOL	FALSE	分配给MC_Power的实例PWR的输出变量Status的变量。进入伺服ON状态时，该变量变为TRUE。
Pos1	LREAL	-	表示窗口有效的起始位置。
Pos2	LREAL	-	表示窗口有效的终止位置。
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则进入伺服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

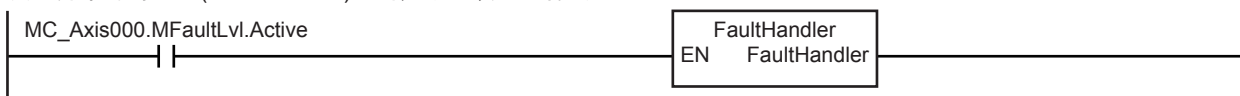
触点StartPg变为TRUE时，确认伺服驱动器处于伺服准备就绪状态



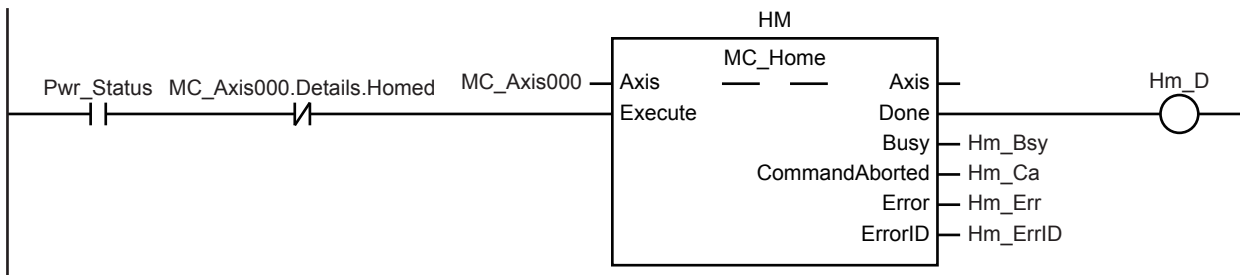
如果轴1处于伺服准备就绪状态，则设为伺服ON状态



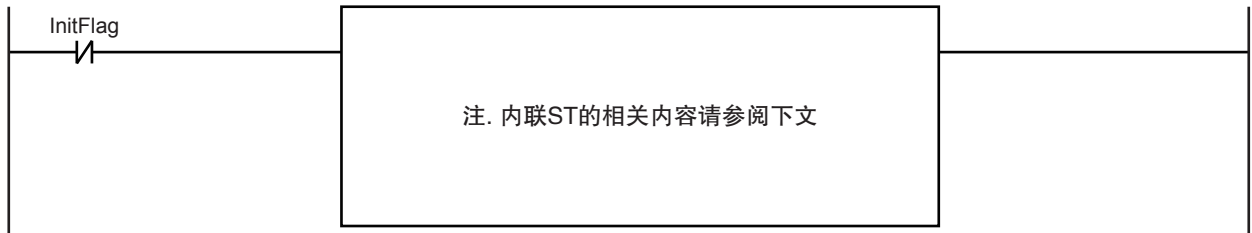
轴1发生轻度故障程度的异常时，执行设备的异常处理(FaultHandler)。
发生异常时的处理(FaultHandler)由客户根据装置进行编程



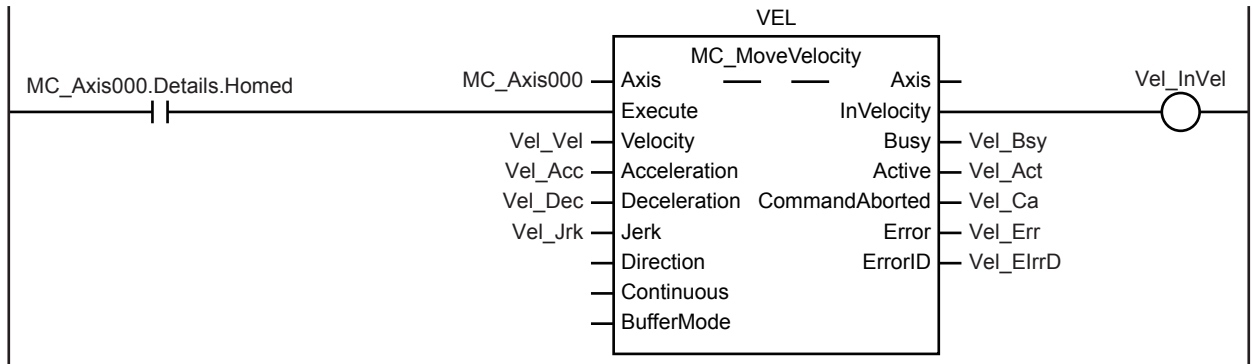
处于伺服ON状态、且原点未确定时，执行原点复位



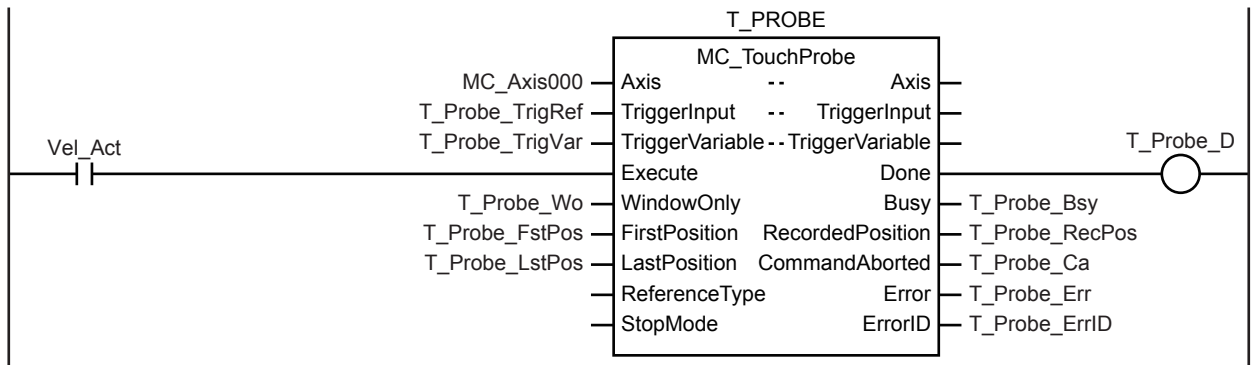
设定MC_MoveVelocity(速度控制)指令和MC_TouchProbe(启用外部锁定)指令的参数



轴1已确定原点, 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



内联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为原点确定状态时变为TRUE。
Pwr_Status	BOOL	FALSE	分配给MC_Power的实例PWR的输出变量Status的变量。进入伺服ON状态时,该变量变为TRUE。
Pos1	LREAL	-	表示窗口有效的起始位置。
Pos2	LREAL	-	表示窗口有效的终止位置。
StartPg	BOOL	FALSE	如果该变量为TRUE, EtherCAT的过程数据通信已建立, 则进入伺服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;

//确认StartPg为TRUE, 且伺服驱动器处于伺服准备就绪状态,
//则设为伺服ON状态。
//如果未处于伺服准备就绪状态, 则设为伺服OFF。
IF (StartPg=TRUE)
AND (MC_Axis000.DrvStatus.Ready=TRUE) 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)由客户根据装置进行编程。
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;

//在伺服ON状态下，且原点未确定时，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;

//MC_MoveVelocity启动后，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

使执行中的锁定无效。

指令	名称	FB/ FUN	图形表现	ST表现
MC_AbortTrigger	不启用外部锁定	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	在上升沿开始指令。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

* 请参见“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	<ul style="list-style-type: none"> 执行锁定的停止时 对非锁定执行中的锁定功能，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"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 	<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	本指令的启动或输入参数中含有异常因素时	异常已解除时

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	-	指定锁定对象轴。 ^{*1}
TriggerInput	触发选择	_sTRIGGER_REF	-	选择触发条件。 ^{*2} _sTRIGGER_REF请参阅下表。

*1. 请使用在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型的用户定义变量。

● _sTRIGGER_REF

结构要素变量	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	在驱动器模式下，指定使用2个锁定功能中的哪一个。 0: 锁定功能1 1: 锁定功能2
InputDrive	驱动触发输入信号	_eMC_TRIGGER_INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	在驱动器模式下指定驱动器的触发信号。 0: Z相 1: 外部输入

功能说明

- 中止执行中的锁定。
- 通过MC_AbortTrigger(不启用外部锁定)指令指定Axis(轴)和LatchID(锁定ID选择),从而指定要中止的锁定。
- 对未执行锁定要求的触发执行MC_AbortTrigger(不启用外部锁定)指令时,MC_AbortTrigger(不启用外部锁定)指令直接正常结束。

MC_TouchProbe(启用外部锁定)指令的执行变为Done(完成)后,执行MC_AbortTrigger(不启用外部锁定)时也是同样情况。

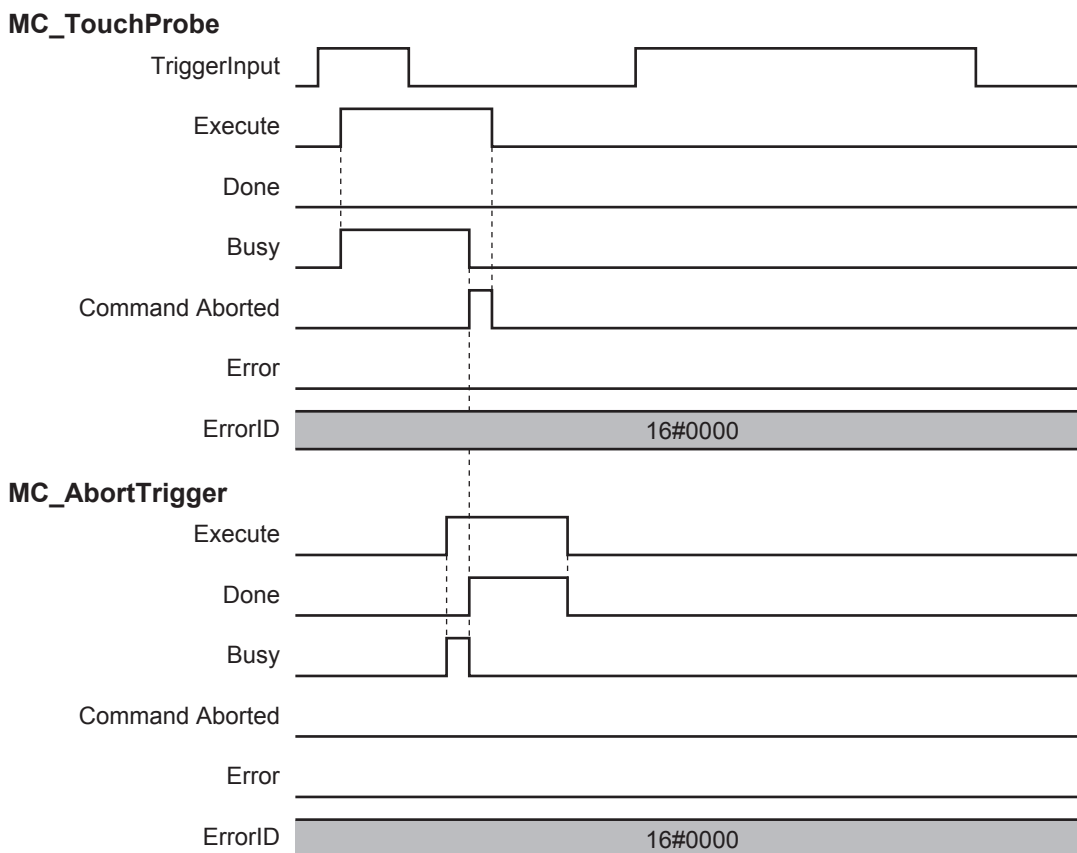


使用注意事项

- 指定MC_AbortTrigger(不启用外部锁定)指令的Axis(轴)在MC_GroupEnable(启用轴组)指令有效时,MC_AbortTrigger(不启用外部锁定)指令发生异常,无法执行。
- 在MC_Home(原点复位)指令、MC_HomeWithParameter(参数指定原点复位)指令、MC_MoveFeed(中断标准定位)指令、MC_MoveLink(梯形模式凸轮)指令执行中,执行MC_AbortTrigger(不启用外部锁定)指令时,本指令发生异常。

时序图

- MC_AbortTrigger(不启用外部锁定)指令的Done(完成)在Execute(启动)的1个周期后变为TRUE。



重启运动指令

无法重启本指令。

重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

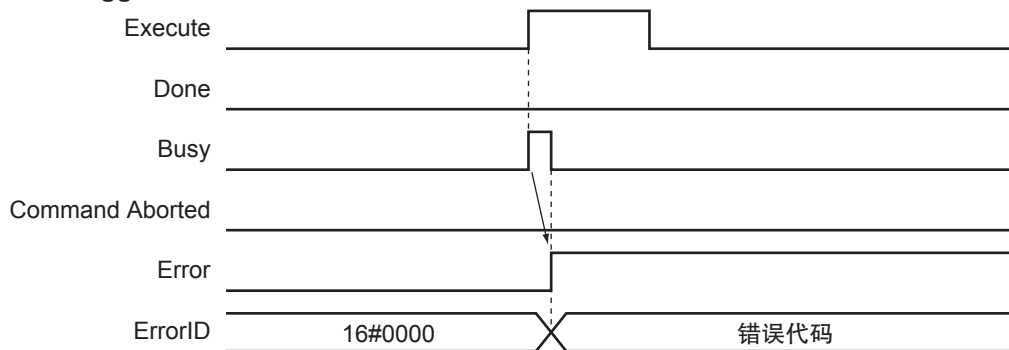
多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

异常

执行MC_AbortTrigger(不启用外部锁定)指令时若发生异常，Error(错误)变为TRUE。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

MC_AbortTrigger



● 错误代码

关于指令发生的异常，请参阅 □ “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过了容许值。

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期间执行指令。
ReferenceType	位置类型选择	_eMC_ REFERENCE_ TYPE	0: _mcCommand 1: _mcFeedback	0 *1	指定位置类型。 0: 指令位置(最近任务周期*2下的 计算值) 1: 反馈位置(同一任务周期 *2下的 取值)
PermittedDeviation	轴间偏差容许值	LREAL	正数、 或“0”	0	指定可容许的主轴及从轴的位置偏 差最大值。 单位为[指令单位]。*3

*1. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

*2. 任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先级5)的周期。

*3. 关于指令单位，请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 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	控制中变为TRUE。
Invalid	轴间偏差超调	BOOL	TRUE, FALSE	超过轴间偏差容许值时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
DeviatedValue	轴间偏差值	LREAL	负数、正数、 “0”	输出指定的主轴和从轴的差异量。 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为正常。

*1. 关于指令单位，请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的 “单位转换设定”。

*2. 请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Enabled	Enable为TRUE时	<ul style="list-style-type: none"> • Error变为TRUE时 • Enable变为FALSE时
Invalid	超过轴间偏差容许量时	<ul style="list-style-type: none"> • Error变为TRUE时 • Enable变为FALSE时 • 未超过轴间偏差容许量时
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含有异常因素时	异常已解除时

*1. Deviated Value在Enable变为FALSE时不返回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. 请使用在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

主轴和从轴请分配至同一任务。将任务各异的轴指定为主轴时，将发生 “主轴指定错误(错误代码：5462 Hex)” 异常。



参考

也可设定属于组的轴。

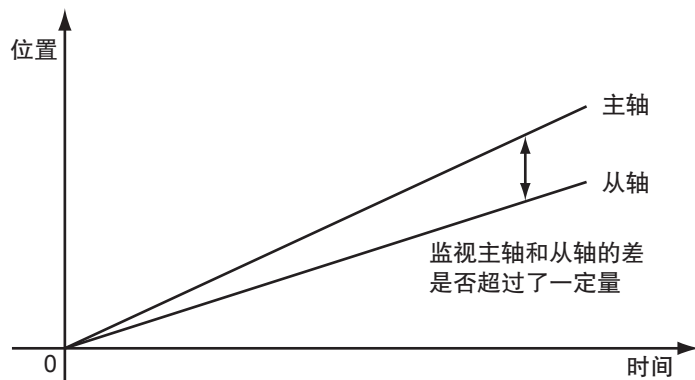
功能说明

- 指定的Master(主轴)和Slave(从轴)的指令位置、或反馈位置的差超过轴间偏差容许值时, Invalid(轴间偏差超限)变为TRUE。

即, 下列条件成立时, Invalid(轴间偏差超限)变为TRUE。

$| \text{DeviatedValue}(\text{轴间偏差值}) | > \text{PermittedDeviation}(\text{轴间偏差容许值})$ 成立时

- 本指令对轴动作没有影响。
- 请对根据输出变量的Invalid(轴间偏差超限)的状态, 进行轴停止处理等处理进行编程。



使用注意事项

- Master(主轴)和Slave(从轴)的计数模式请设定为同一模式。
指定不同模式的轴时, 比较结果为[线性模式]。
此外, 即使同为[旋转模式], 如果环计数器的设定范围不同, 则比较结果也是[线性模式]。
- PermittedDeviation(轴间偏差容许值)含有除不尽的除法结果等, 由于误差的原因, 处理结果可能会出现意外。
- NX系列CPU单元的Master(主轴)和Slave(从轴)请分配至同一任务。分配至不同任务时, Slave(从轴)侧将发生“主轴指定错误(错误代码: 5462 Hex)”。
- NX系列脉冲输出单元的指令位置=反馈当前位置, 因此通过本指令进行监视没有实效性。

指令详情

下面对指令详细说明。

● ReferenceType(位置类型选择)

可从下列选项中选择从轴同步的主轴位置类型。

- `_mcCommand`: 指令位置(最近任务周期下的计算值)
对当前周期, 使用之前任务周期时计算的主轴指令位置。
在计算从轴指令位置之前的固定周期任务中, 使用计算的主轴指令位置。
- `_mcFeedback`: 同一任务周期下的取值
使用同一任务周期时获取的主轴反馈位置。



使用注意事项

这里的任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先级5)的周期。同样, 固定周期任务表示原始恒定周期任务或固定周期任务(执行优先级5)。

● 轴种类与位置类型的关系

可监控的轴种类和要监控的位置类型的关系如下所示。

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. 启动指令时，发生“超过位置类型选择范围(错误代码：5430 Hex)”错误。

● DeviatedValue(轴间偏差值)的计算方法

按照如下方法计算DeviatedValue(轴间偏差值)。

线性模式时

ReferenceType(位置类型选择)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(位置类型选择)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(位置类型选择)为_mcCommand时

DeviatedValue(轴间偏差值)在环计数器的范围上，选择Master(主轴)的指令当前位置和Slave(从轴)的指令当前位置的较短的距离。

此时，从较短的距离来看，DeviatedValue(轴间偏差值)的符号分别为：

- Master(主轴)的指令当前位置 \geq Slave(从轴)的指令当前位置时，“+”。
- Master(主轴)的指令当前位置 $<$ Slave(从轴)的指令当前位置时，“-”。

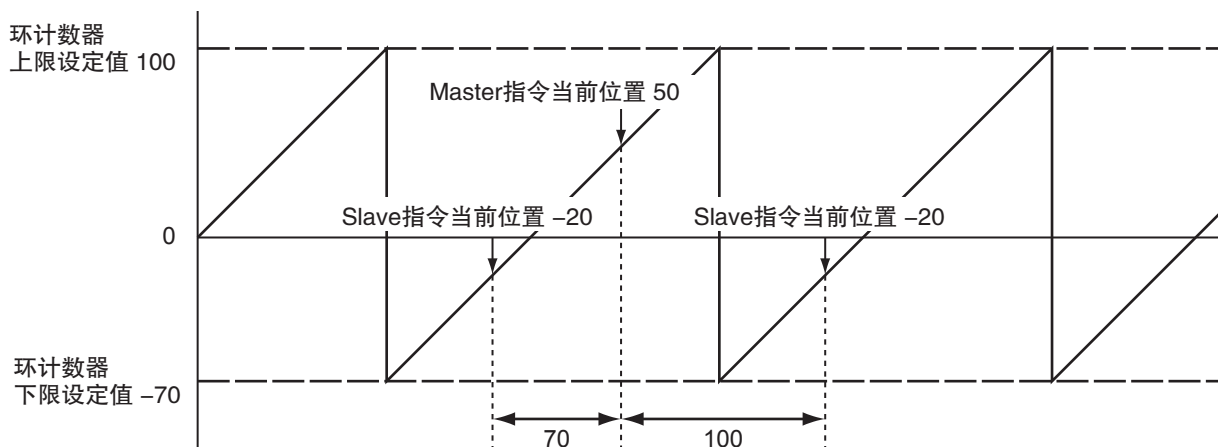
ReferenceType(位置类型选择)为_mcFeedback时

DeviatedValue(轴间偏差值)在环计数器的范围上，选择Master(主轴)的反馈当前位置和Slave(从轴)的反馈当前位置的较短的距离。

此时，从较短的距离来看，DeviatedValue(轴间偏差值)的符号分别为：

- Master(主轴)的反馈当前位置 \geq Slave(从轴)的反馈当前位置时，“+”。
- Master(主轴)的反馈当前位置 $<$ Slave(从轴)的反馈当前位置时，“-”。

[旋转模式]下，ReferenceType(位置类型选择)为_mcCommand时的DeviatedValue(轴间偏差值)的计算示例



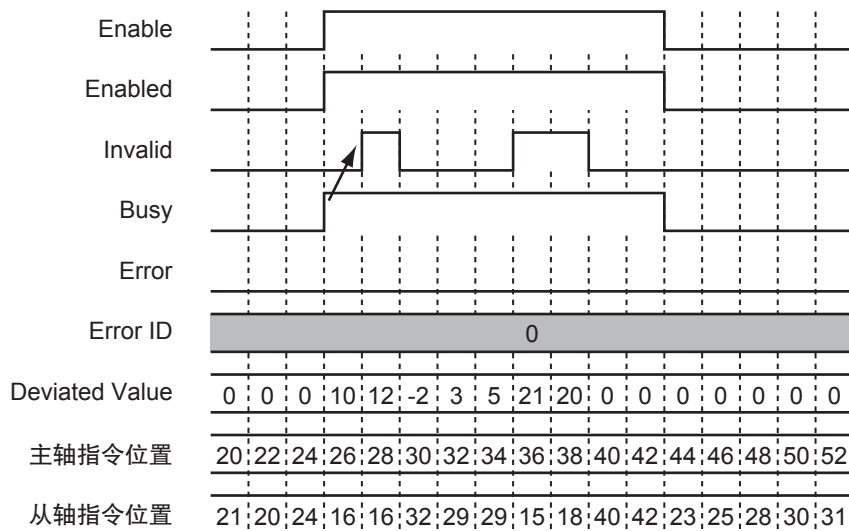
由于 $|DeviatedValue|$ 为 $100 > 70$ ，因此为 70
 由于 Master 指令当前位置 \geq Slave 指令当前位置，
 因此符号为 “+” 所以，DeviatedValue 为 “+70”

[旋转模式]下，ReferenceType(位置类型选择)为_mcFeedback时，上图的“指令当前位置”变为“反馈当前位置”。

时序图

本指令的异常对轴及轴组的动作没有影响。

PermittedDeviation(轴间偏差容许值)为“10.0”时的时序图如下所示。



重启运动指令

输入为“Enable型”指令时，不会重启运动指令。

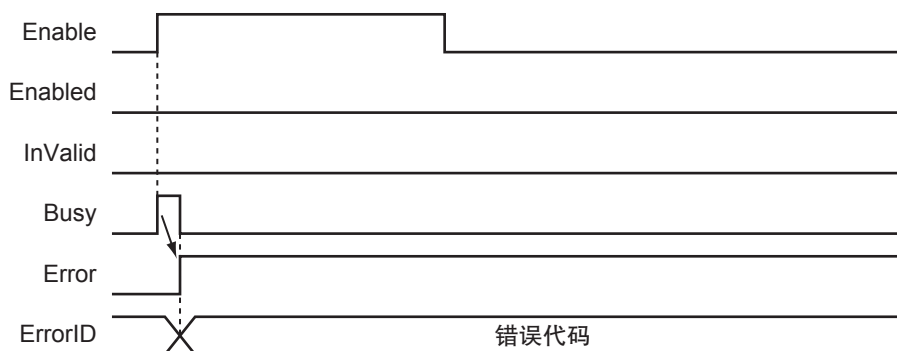
多重启动运动指令

- 无多重启动的限制。

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

异常

- 本指令执行中发生异常时，Error(错误)变为TRUE。
- 可查看ErrorID(错误代码)的输出值，了解发生异常的原因。
- 本指令的异常信息输出到MC通用运动变量的轻度故障。



● 错误代码

关于指令发生的异常，请参阅 □ “A-1 错误代码一览(P.A-2)”。

MC_SyncMoveVelocity

按照任务周期，以周期同步速度模式(CSV)将目标速度设定值输出到伺服驱动器。



使用注意事项

- 无法对NX系列脉冲输出单元使用本指令。
- 这里的任务周期分为原始恒定周期任务的周期和固定周期任务(执行优先度5)的周期。

指令	名称	FB/ FUN	图形表现	ST表现
MC_SyncMoveVelocity	周期同步速度控制	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	在上升沿开始指令。
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: 正方向移动 负数: 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_M ODE	0: _mcCount	0 *2	0: 使用反馈当前位置更新指令当前 位置。 原点确定状态保持不变。
BufferMode	缓存模式 选择	_eMC_BUFFER_MO DE	0: _mcAborting 1: _mcBuffered	0 *2	指定多重启动运动指令时的动作。 0: 中断 1: 等待

*1. 关于指令单位，请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的 “单位转换设定”。

*2. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时, 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为正常。

*  请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
InVelocity	达到目标速度后	<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	开始指令后	<ul style="list-style-type: none"> • Error变为TRUE时 • CommandAborted变为TRUE时
CommandAborted	<ul style="list-style-type: none"> • 利用其他指令中止本指令时 • 其他指令发生异常、使本指令中止时 • 轴异常发生过程中启动本指令时 • 执行MC_Stop指令中, 启动本指令时 	<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含有异常因素时	异常已解除时

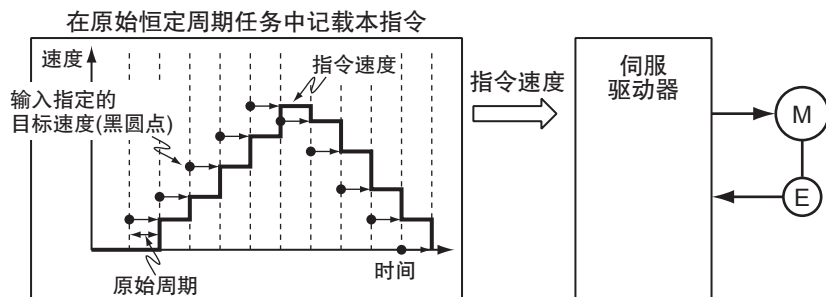
输入输出变量

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

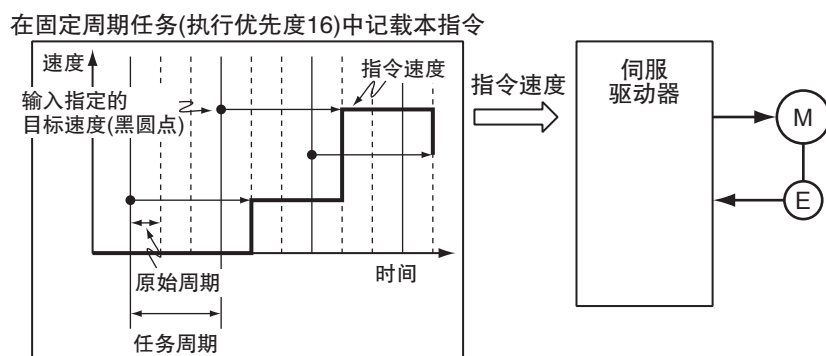
*1. 请使用在Sysmac Studio的轴基本设定画面中创建的用户定义变量的轴变量名称(默认 “MC_Axis***”)或系统定义变量的轴变量名称(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])。

功能说明

- 本指令按照任务周期、以周期同步速度模式(CSV)将用户程序给定的目标速度输出到伺服驱动器。
- 在Execute(启动)的上升沿，切换伺服驱动器控制模式的同时，输出指令速度。
- 原始恒定周期任务或固定周期任务(执行优先级5)中记载有本指令时，在下1个任务周期达到目标速度。原始恒定周期任务中记载有本指令的示例如下所示。固定周期任务(执行优先级5)时也一样。



- 固定周期任务(执行优先级16)中记载有本指令时，在下1个固定周期任务中达到目标速度。



使用注意事项

关于针对同步控制的主轴使用本指令时的注意事项，请参阅 □ “同步控制的主轴及辅轴的注意事项(P.1-6)”。



参考

使用MC_SyncMoveVelocity(周期同步速度控制)指令时，MC_SetOverride(超调值设定)指令不起作用。

● 数据对象的映射

使用MC_SyncMoveVelocity(周期同步速度控制)指令时，请在Sysmac Studio的轴基本设置的[高级设置]中映射如下对象数据。

- 目标速度(60FFHex)
- 操作模式(6060Hex)
- 操作模式显示(6061Hex)

未进行这些设定时，将发生过程数据对象设定不足(错误代码：3461Hex)错误。

关于数据对象的映射，请参阅 □□ “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-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

指令详情

下面对指令详细说明。

● Velocity(目标速度)

可以以LREAL型(以“0”为基准)设定输入变量Velocity(目标速度)。

设为正数时，沿正方向移动；设为负数时，沿负方向移动。

设定为“0”时，指令速度变为“0”，但是Status.Continuous(连续动作中)保持TRUE。

每个周期都可以通过用户程序设定Velocity(目标速度)。

指定与上一周期不同的目标速度时，采用新的目标速度。指定与上一周期相同值时，以上一周期的目标速度进行动作。



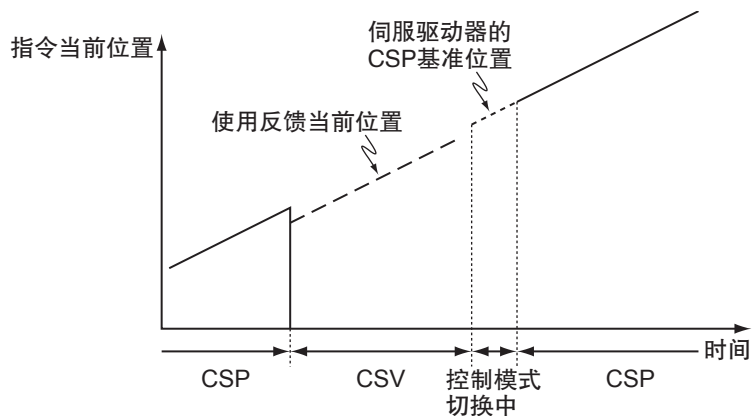
使用注意事项

- 设定目标速度时，请指定为合适的值，避免因速度变化而对装置的机械机构造成过大负担。
- 将使用本指令的轴作为同步控制的主轴使用时，根据对主轴目标速度设定的不同，从轴可能急剧动作。敬请注意。
- 切换控制模式时，指令当前位置可能突然发生变化。请注意。

● 指令当前位置

如果输入变量CmdPosMode(指令当前位置计数选择)选择_mcCount, 则启动本指令时, 指令当前位置即为1个周期前的反馈当前位置, 使用反馈当前位置直至本指令结束。

欧姆龙制伺服驱动器1S系列及G5系列在切换到CSP的处理中, 事先映射的“CSP基准位置”被PDO发送。将该基准位置用作指令当前位置, 直至从CSV切换到CSP的处理完毕。切换到CSP的处理完毕后, 指令当前位置 = 指令位置。



● 使用欧姆龙制伺服驱动器1S系列时

使用控制模式切换途中的伺服驱动器的“CSP基准位置”时, 请将CSP用基准位置(3010Hex-87Hex)映射到过程数据中。

在Sysmac Studio的PDO编辑画面中, 将CSP用基准位置(3010Hex-87Hex)映射到过程数据中。然后, 请通过轴基本设置的[高级设置]映射MC功能模块的“CSP切换用基准位置”和CSP用基准位置(3010 Hex-87Hex)。

● 使用欧姆龙制伺服驱动器 G5系列时

使用控制模式切换途中的伺服驱动器的“CSP基准位置”时, 请将CSP用基准位置(4020 Hex)映射到过程数据中。

在Sysmac Studio的PDO编辑画面中, 将CSP用基准位置(4020 Hex)映射到过程数据中。然后, 请通过轴基本设置的[高级设置]映射MC功能模块的“CSP切换用基准位置”和CSP用基准位置(4020 Hex)。



使用注意事项

将“CSP用基准位置(4020 Hex)”映射到PDO中时, 请将原始周期或固定周期任务(执行优先度5)的任务周期设定为1ms以上。

否则, 伺服驱动器G5系列侧会发生异常。

详情请参阅 □□ “AC伺服电机/驱动器 G5系列 EtherCAT通信内置型 用户手册(SBCE-365)”。



参考

欧姆龙制伺服驱动器 G5系列中, 直线电机型不适用于CSP基准位置(4020 Hex)。

● 使用欧姆龙制伺服驱动器1S系列或G5系列以外的驱动器时

启动本指令时, 或在本指令执行中多重启动使用CSP的指令时, 根据伺服驱动器的不同, 在进行控制模式切换处理时可能发生异常。

其时, 请在轴停止的状态(速度“0”)下, 启动本指令或多次启动使用CSP的指令。

● 使用NX系列位置接口单元时

无法对NX系列位置接口单元使用本指令。

● 停止处理

下面对轴动作停止时的控制模式和指令速度进行说明。

要减速停止时，请以本指令的目标速度为初速度，采用各停止方法具有的减速度进行减速停止。

基于MC_ImmediateStop(立即停止)指令的停止

将指令速度设为“0”。下列切换条件式成立时，切换为CSP。

基于MC_Stop(强制停止)指令的停止

以指令的减速度将指令速度设为“0”。下列切换条件式成立时，切换为CSP。

因发生轻度故障等级的异常而停止

各异常以规定的减速度将指令速度设为“0”。下列切换条件式成立时，切换为CSP。

因发生全部停止故障等级以及部分停止故障等级的异常而停止

将指令速度设为“0”。下列切换条件式成立时，切换为CSP。

但是，根据异常等级的不同，可能无法正常执行控制模式的切换而以CSV停止。

基于伺服OFF的停止

利用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。

将CPU单元的动作模式切换为程序模式时的停止

利用规定的方法将指令速度设为“0”。下列切换条件式成立时，切换为CSP。

● 切换条件式

控制模式可否切换取决于伺服驱动器的规格。

为了在启动停止指令或发生异常进行停止处理时切实切换控制模式，请在确认伺服电机已充分减速后切换到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次满足下列条件式时，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原始恒定周期任务或固定周期任务(执行优先级5)。

● 恢复周期同步位置模式

可能因停止而无法切换到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就属于这种情况。

因此，MC_Power(可运行)指令的输出变量“Status(可运行)”变为FALSE时，切换到CSP。

● 控制模式切换失败时的动作

对伺服驱动器执行控制模式切换指令并经过1秒钟后，伺服驱动器的切换还未完成时，发生“伺服驱动器的控制模式切换异常(错误代码：7439 Hex)”的错误。

并且，对伺服驱动器执行控制模式切换指令，指令速度变为“0”并经过10秒钟后，切换条件式仍然不成立时，也和上述同样，发生“伺服驱动器的控制模式切换异常(错误代码：7439 Hex)”的错误。

发生“伺服驱动器的控制模式切换异常(错误代码：7439 Hex)”的错误时，将指令速度设为“0”，执行伺服OFF(自由运行停止)。

关于“伺服驱动器的控制模式切换异常(错误代码：7439 Hex)”，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-363)” 或 □□ “NY 系列 工业用平板电脑 / 工业用台式电脑 用户手册 运动控制篇 (SBCE-379)”。

● 可使用轴和启动条件

- 伺服轴可以使用本指令。
使用时，请将MC_Power(可运行)指令的Enable(有效)设为TRUE(伺服ON状态)。
- 虚拟伺服轴可常时接收。
但是，不执行伺服驱动器的控制模式切换处理。
- 编码器轴、虚拟编码器轴则会发生启动时异常。

● 轴变量的状态

轴变量的状态Status.Continuous(连续动作中)变为TRUE。

需要了解伺服驱动器设定的控制模式时，请确认轴变量DrvStatus(伺服驱动器状态)。伺服驱动器状态如下所示。

变量名称	数据类型	名称	说明
CSP	BOOL	周期同步位置(CSP)模式中	伺服ON、CSP模式时变为TRUE。
CSV	BOOL	周期同步速度(CSV)模式中	伺服ON、CSV模式时变为TRUE。
CST	BOOL	周期同步转矩(CST)模式中	伺服ON、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，原点确定状态保持不变。

● 超调

超调对本指令无效。

● 软件限位

软件限制对本指令有效。

通过轴参数选择如下设定时也有效。

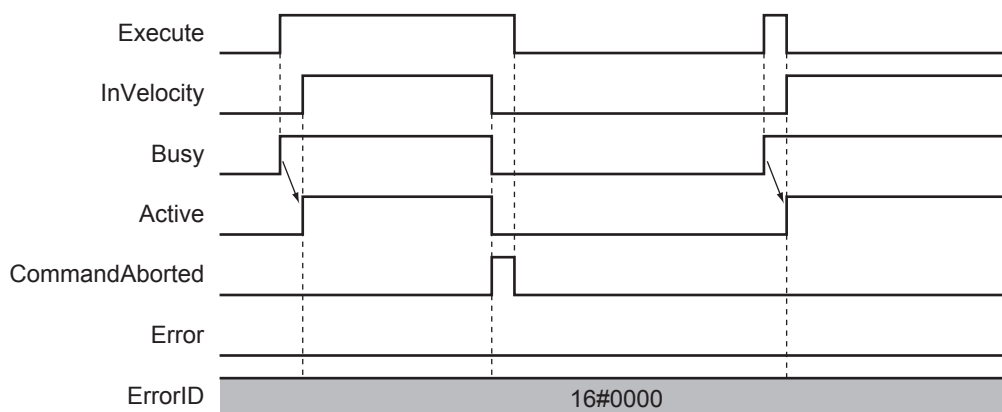
- 对指令位置启用。执行减速停止。
- 对指令位置启用。执行累积脉冲停止。

时序图

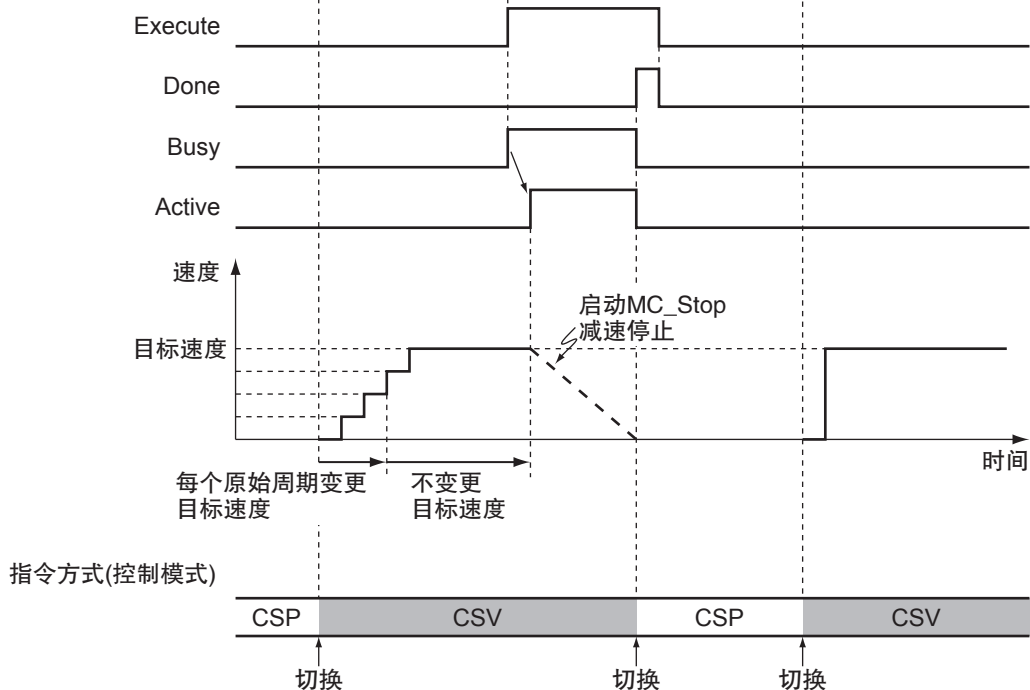
- 在启动Execute(启动)的同时, Busy(执行中)变为TRUE。在下一个周期Active(控制中)变为TRUE。
- 指令速度达到Velocity(目标速度)时, InVelocity(达到目标速度)变为TRUE。
- 利用其它指令中止本指令时, CommandAborted(执行中断)变为TRUE, Busy(执行中)、Active(控制中)、InVelocity(达到目标速度)变为FALSE。
- 要结束本指令, 请启动MC_Stop(强制停止)指令。

原始恒定周期任务中记载有本指令时的情形如下所示。

MC_SyncMoveVelocity指令



MC_Stop指令



参考

MC功能模块使用上述时序图对伺服驱动器发出控制模式的指令。但是, 反映到伺服驱动器的时序, 则因各伺服驱动器的规格而异。

重启运动指令

无法重启本指令。

重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

多重启动运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动

与MC_MoveVelocity(速度控制)指令相同，支持BufferMode(缓存模式选择)的中断和等待。

此外，在开始多重启动指令的处理时切换控制模式。

关于BufferMode(缓存模式选择)，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 本指令执行中的其它指令启动

与MC_MoveVelocity(速度控制)指令相同，支持BufferMode(缓存模式选择)的中断和等待。

等待方式下，InVelocity(达到目标速度)变为TRUE时，启动其他指令。

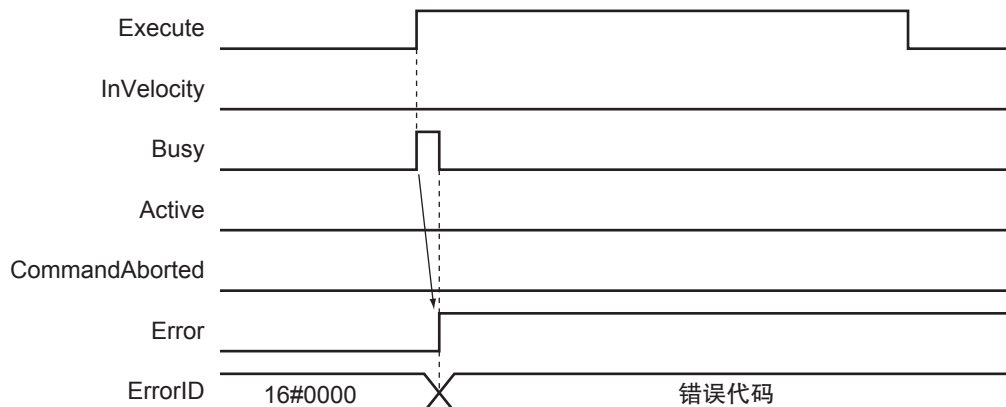
此外，在开始多重启动指令的处理时切换控制模式。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 □□ “A-1 错误代码一览(P.A-2)”。

MC_SyncMoveAbsolute

按周期输出轴的指定目标位置。

指令	名称	FB/ FUN	图形表现	ST表现
MC_SyncMoveAbsolute	周期同步绝对定位	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

Ver.1.03以上的CPU单元和Ver.1.04以上的Sysmac Studio组合时可使用本指令。

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	在上升沿开始指令。
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: 指定为正方向 1: 指定为附近 2: 指定为负方向 3: 指定为当前方向 4: 无方向指定
BufferMode	缓存模式 选择 ^{*2}	_eMC_BUFFER_ MODE	0: _mcAborting	0 ^{*3}	指定多重启动运动指令时的动作。 0: 中断

*1. 关于指令单位，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*2. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

*3. 关于旋转方向的指定内容，请参阅 □□ “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	到位	BOOL	TRUE, FALSE	反馈当前位置进入目标位置的到位宽度内时，到位变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

*1. 请参见“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
InPosition	反馈当前位置进入目标位置的到位宽度内时	<ul style="list-style-type: none"> 反馈当前位置在到位宽度以外时 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	开始指令后	<ul style="list-style-type: none"> Error变为TRUE时 CommandAborted变为TRUE时
CommandAborted	<ul style="list-style-type: none"> 利用其它指令多重启动运动指令,中止本指令时 其他指令发生异常、使本指令中止时 轴异常发生过程中启动本指令时 执行MC_Stop指令中，启动本指令时 	<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含有异常因素时	异常已解除时

输入输出变量

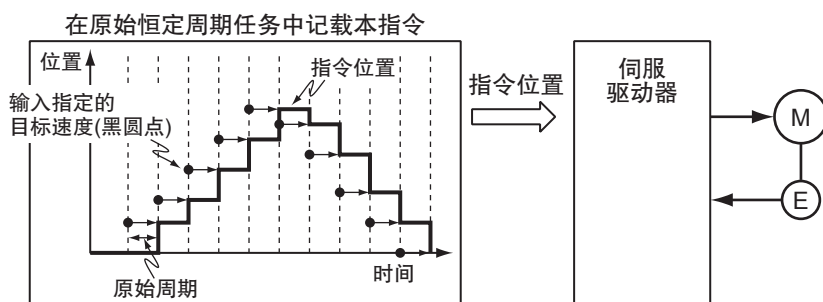
输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	-	指定轴。*1

*1. 请在使用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[*])。

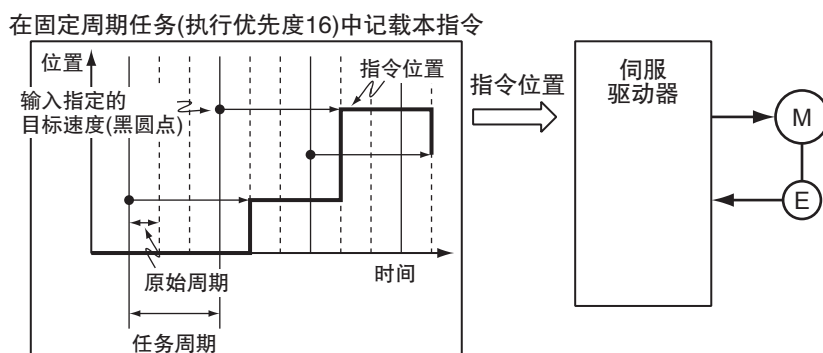
功能说明

- 本指令按照任务周期、以周期同步位置模式(CSP)将用户程序给定的目标位置输出到伺服驱动器等。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。[最大加速度]、[最大减速度]不适用。
- 原始恒定周期任务或固定周期任务(执行优先度5)中记述有本指令时，通过输入指定的目标位置在下一个任务周期输出到伺服驱动器。

原始恒定周期任务中记载有本指令的示例如下所示。固定周期任务(执行优先度5)时也一样。



- 固定周期任务(执行优先度16)中记述有本指令时，通过输入指定的目标位置在下一个固定周期任务时输出到伺服驱动器。



使用注意事项

请指定恰当的目标位置，使指定目标位置的移动量不超过轴参数的[最高速度]。若指定的目标位置超过了[最高速度]，会发生指令速度饱和，输出以[最高速度]限制的移动量。距离指定目标位置不足的部分的移动量在下一个周期以后输出。
此时，轴控制状态的Details.VelLimit(指令速度饱和)变为TRUE。

指令详情

下面对指令详细说明。

● 到位检查

不更新Position(目标位置)时，如果目标位置和反馈位置的幅度在轴参数[到位宽度]的范围内，则InPosition(到位)变为TRUE。

InPosition(到位)为TRUE时，即使变更目标位置，在该周期内仍保持TRUE不变，到下一个周期时才变为FALSE。

轴参数[到位检查时间]的设定无效。

● 停止处理

下面对停止轴动作时的动作进行说明。

要停止动作，请使用MC_Stop(强制停止)指令或MC_ImmediateStop(立即停止)指令。启动这些指令后，本指令的CommandAborted(执行中断)变为TRUE。

基于MC_Stop(强制停止)指令的停止

立即停止。

基于MC_ImmediateStop(立即停止)指令的停止

根据轴参数设定[立即停止输入停止方法]的指定进行立即停止。

● 异常发生时的停止

发生使轴停止动作的异常时，无论如何设定，均“立即停止”。

● 可以使用的轴

- 伺服轴可以使用本指令。
使用时，请将MC_Power(可运行)指令的Enable(有效)设为TRUE(伺服ON状态)。
- 虚拟伺服轴可常时接收。
- 编码器轴、虚拟编码器轴则会发生启动时异常。

● 轴变量的状态

轴变量的轴状态Status.Discrete(定位动作中)变为TRUE。
对轴控制状态没有影响。

● 超调

超调对本指令无效。

● 当前位置变更

通过MC_SetPosition(当前位置变更)指令变更当前位置时，输出从变更的当前位置到Position(目标位置)指定的目标位置的移动量。

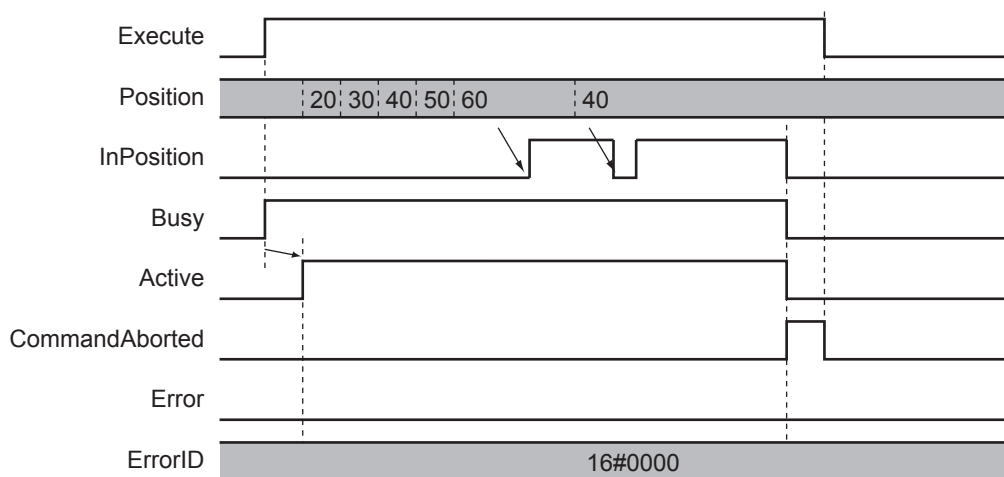
当前位置变更前 InPosition(到位) 为 TRUE 时，在当前位置变更后的下一个周期，InPosition(到位) 变为 FALSE。

时序图

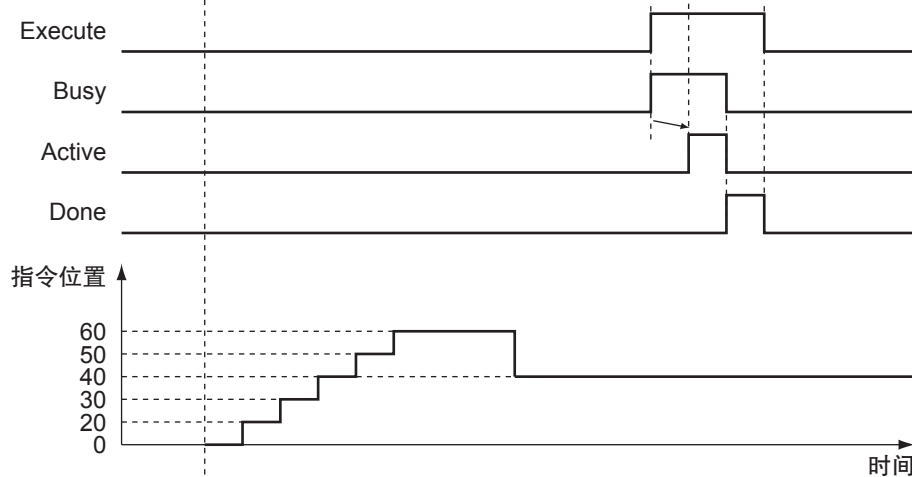
- 在启动Execute(启动)的同时，Busy(执行中)变为TRUE。在下一个周期Active(控制中)变为TRUE。
- 反馈当前位置进入Position(目标位置)的到位宽度内时，InPosition(到位)变为TRUE。
- 利用其它指令中止本指令时，CommandAborted(执行中断)变为TRUE，Busy(执行中)、Active(控制中)、InPosition(到位)变为FALSE。
- 要结束本指令，请启动MC_Stop(强制停止)指令。

原始恒定周期任务中记载有本指令时的情形如下所示。

MC_SyncMoveAbsolute指令



MC_Stop指令



参考

MC功能模块使用上述时序图对伺服驱动器发出控制模式的指令。但是，反映到伺服驱动器的时序，则因各伺服驱动器的规格而异。

重启运动指令

无法重启本指令。

重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

多重启动运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动

对当前正在执行的指令启动本指令，可切换或缓存到本指令。

各轴可缓存到1个。

多重启动时的本指令的动作由BufferMode(缓存模式选择)指定。

缓存模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 轴的动作方向因指令切换而反转时，根据轴参数中的“反转时动作”进行反转。

关于BufferMode(缓存模式选择)，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 本指令执行中的其它指令启动

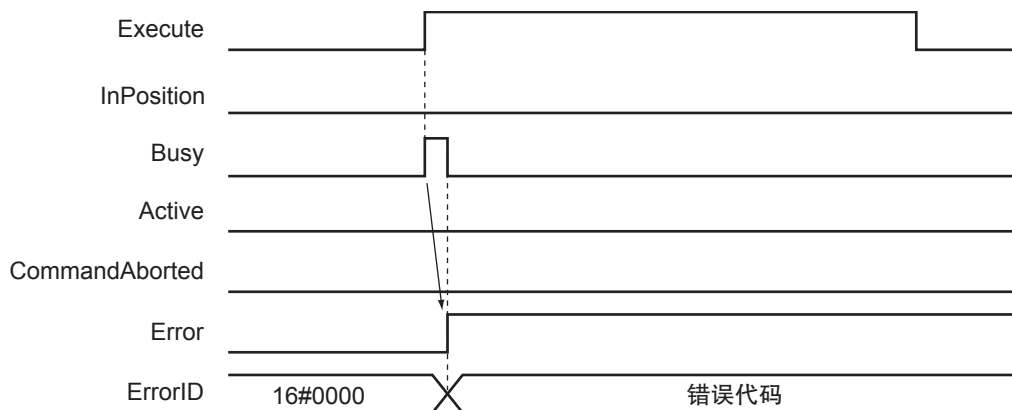
对本指令进行多重启动运动指令时，可使用中断模式。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 □□ “A-1 错误代码一览(P.A-2)”。

MC_Reset

解除轴的异常。

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>

MC_Reset

3

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	在上升沿开始指令。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Failure	非法结束	BOOL	TRUE, FALSE	指令未正常执行时变为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为正常。

* 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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"> 轴因异常而减速停止的过程中执行异常解除时 因发生轴公共异常而导致轴发生异常的过程中执行异常解除时 	<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含有异常因素时	异常已解除时

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. 请使用在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，对Axis(轴)指定轴的异常开始解除处理。
异常解除处理是指，轴的异常解除、以及在驱动器侧发生异常时进行驱动器错误复位。
无论轴的运行状态如何，均执行异常解除处理。
- 可对所有轴种类使用本指令。
- 仅对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，应先执行驱动器错误复位处理，然后再执行异常解除处理。
驱动器错误复位处理可选择清除驱动器错误，还是在轴参数[驱动器错误复位监视时间]内保持不变。
- 在错误减速停止中的轴减速停止中执行本指令时，无法执行，Failure(非法结束)变为TRUE。这是因为在轴停止之前无法进行异常解除。
此外，发生MC公共异常的轴也无法通过本指令解除异常，Failure(非法结束)会变为TRUE。
MC公共异常分为“MC公共部分停止故障”以及“MC公共轻度故障”。
- 可解除的异常对象为Execute(启动)上升沿时发生的异常。不能对在异常解除过程中发生的异常执行异常解除。



使用注意事项

- 通过本指令进行的异常解除处理有时会跨越多个控制周期。
- 发生“MC公共部分停止故障”以及“MC公共轻度故障”等公共异常时，或轴动作中时，本指令的Failure(非法结束)变为TRUE。在排除异常原因之后，请执行重试处理，直到变为Done(完成)。
- 请确认轴已完全停止后排除异常发生原因。
需要了解轴是否已完全停止，必须确认轴变量Act.Vel(反馈当前速度)是否已变为“0”。
- 对欧姆龙制伺服驱动器G5系列使用本指令时，请进行指令的排他性处理，以免同时执行ResetECEError(EtherCAT异常解除)指令。
- 对NX系列脉冲输出单元使用本指令时，脉冲输出单元连接的驱动器的异常状态无法解除。详情请参阅 □ “NX系列 位置接口单元 用户手册(SBCE-374)”。



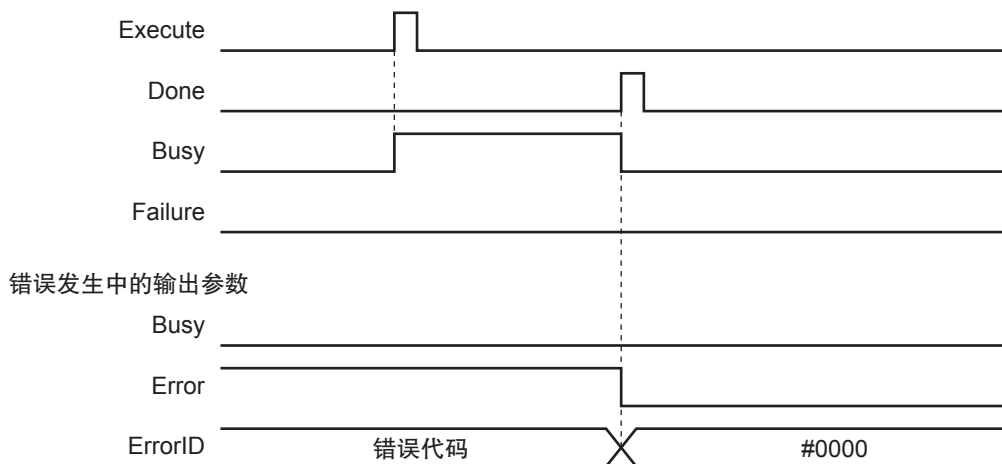
参考

本指令不能解除以下异常。

- 所有轴公共异常：请执行ResetMcError(所有错误复位)指令。
- 所有轴组异常：请执行MC_GroupReset(轴组错误复位)指令。

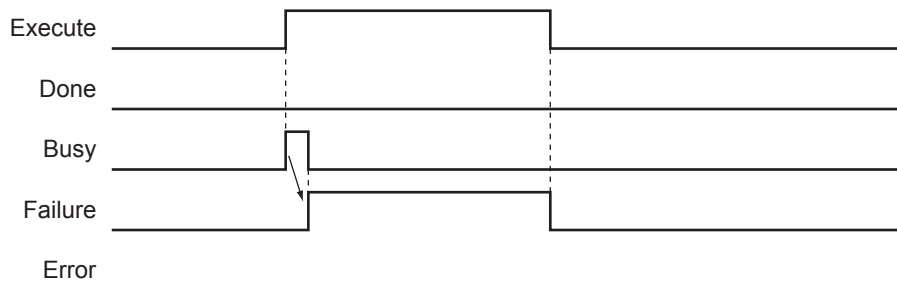
此外，执行本指令无法排除从站通信异常等网络异常的原因。请执行ResetECATError指令。

时序图



指令的中止

轴因异常而减速停止时、或者因发生轴公共异常而导致轴发生异常的过程中无法解除异常时，请中断指令。



错误发生中的输出参数



错误代码

关于指令发生的异常，请参阅 “A-1 错误代码一览(P.A-2)”。

MC_ChangeAxisUse

暂时切换轴参数的[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

通过本指令改写的值未保存至CPU单元的非易失性存储器中。因此，关闭控制器电源、以及开始下载、开始执行MC功能模块的重启处理时，改写值消失，恢复为通过Sysmac Studio设定的值。

保存到非易失性存储器时，请使用Sysmac Studio进行参数传送。



参考

- “下载”使用Sysmac Studio的“同步”功能。
- 使用本指令的应用示例，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 软件篇(SBCA-359)” 或 □□ “NY系列工业用平板电脑/工业用台式电脑用户手册软件篇(SBCA-436)”。



版本相关信息

Ver.1.04以上的CPU单元和Ver.1.05以上的Sysmac Studio组合时可使用本指令。

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	在上升沿开始指令。
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. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

* 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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	其他指令发生异常、使本指令中止时	<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含有异常因素时	异常已解除时

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. 请在使用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)指定的轴使用。
- 在指令执行完成的同时, 切换为轴变量_MC_AX[*].Cfg.AxEnable(轴使用)指定的轴使用。
- 可切换轴是指, 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]。
- 无论原点确定还是未确定, 均可执行指令。如果原点已确定时切换为未使用轴, 则变为原点未确定状态。
- 将使用绝对值编码器的轴从使用轴切换为未使用轴时, 在切断CPU单元电源时, CPU单元内置的电池备份内存中保存的[绝对值编码器原点位置偏置]为切换为未使用轴之前的值。



使用注意事项

- 各种CPU单元型号中, 轴编号超过控制轴最大数量的轴, 无法切换为使用轴。可切换为使用轴的最大实轴数为最大使用实轴数。

规格项目	NX701-17□□	NX701-16□□
可设定的轴号	0 ~ 255	0 ~ 127
使用实轴最大数量	256轴	128轴

规格项目	NX1P2-11□□	NX1P2-10□□	NX1P2-90□□
可设定的轴号	0 ~ 11	0 ~ 9	0 ~ 3
使用实轴最大数量	8轴	6轴	4轴
使用运动控制伺服轴	4轴	2轴	-
使用单轴位置控制伺服轴	4轴	4轴	4轴

规格项目	NJ501- /NY5□2-		
	□5□□	□4□□	□3□□
可设定的轴号	0 ~ 63	0 ~ 31	0 ~ 15
使用实轴最大数量	64轴	32轴	16轴

规格项目	NJ301-12□□	NJ301-11□□	NJ101-10□□
可设定的轴号	0 ~ 14 ^{*1}	0 ~ 14 ^{*2}	0 ~ 5
使用实轴最大数量	8轴	4轴	2轴

*1 Ver.1.05 以下版本的 CPU 单元的轴号为“0 ~ 7”。

*2 Ver.1.05 以下版本的 CPU 单元的轴号为“0 ~ 3”。

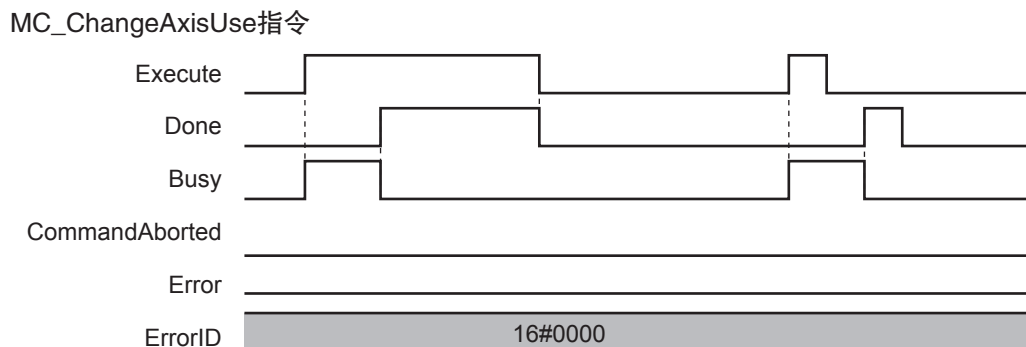
- 仅轴变量_MC_AX[*].Status.Disabled(轴无效)或_MC_AX[*].Status.Standstill(停止中)为TRUE时可执行指令。FALSE时执行会发生异常。
- 轴变量_MC_AX[*].Details.VelLimit(指令速度饱和)为TRUE时执行会发生异常。
- 对轴使用设为_mcUnusedAxis(未使用轴)的轴, 通过MC_write(MC设定写入)指令改写轴参数无效。请将轴使用切换为_mcUsedAxis(使用轴)后执行MC_write(MC设定写入)指令。
- 对于包含通过本指令切换为未使用轴的轴的轴组, 执行MC_GroupEnable(启用轴组)指令会发生异常。
- 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通用异常。
- 通过本指令切换轴使用后执行其它运动控制指令时, 请确认本指令的输出变量“Done(完成)”为TRUE后再执行。



参考

对于NX系列CPU单元, 以_MC_AX[*]开头的变量名可能为_MC1_AX[*]、_MC2_AX[*]。

时序图



重启运动指令

无法重启本指令。

重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

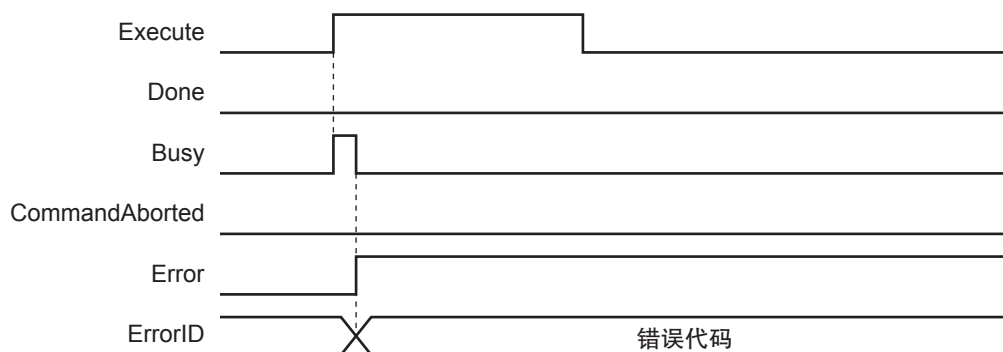
多重启动运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

异常

启动本指令发生异常时，Error(错误)变为TRUE，轴使用无法切换。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 □□ “A-1 错误代码一览(P.A-2)”。



使用注意事项

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通用异常。

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

- 请务必在轴速恒定的情况下使用MC_DigitalCamSwitch(数字凸轮开关有效)指令。
- 请确认MC_DigitalCamSwitch(数字凸轮开关有效)指令的InOperation(工作中)为TRUE后，再使用NX_AryDOutTimeStamp(时间戳数字输出数组写入)指令。



使用注意事项

- 本指令仅对分配了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□□□。
- 本指令必须与支持NX_AryDOutTimeStamp指令及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Ver.1.06以上的CPU单元和Ver.1.07以上的Sysmac Studio组合时使用。

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期间执行指令。FALSE时不更新“Outputs”的值。 ^{*1}
EnableMask	轨段有效	WORD	16#0000 ~ FFFF	16#0000	按轨段指定有效或无效。最多16个轨段中，第0位的值指定轨段编号0的有效或无效，第15位的值指定轨段编号15的有效或无效。 0: 无效 ^{*2} 1:有效
ValueSource (Reserved)	输入信息	_sMC_SO URCE	-	-	(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为OFF。

*2. EnableMask的位值为“0”时，相应轨段编号的Outputs(输出信号)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	有效输出信号的输出中变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为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为正常。

* 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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含有异常因素时	异常已解除时

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	-	指定参考位置的轴。 ^{*1}
Switches	开关	ARRAY[0..255] OF _sCAMSWITCH_REF* ²	-	将开关结构体_sCAMSWITCH_REF型的数组变量指定为开关的ON/OFF模式数据。 数组元素编号表示开关编号。
Outputs	输出信号	ARRAY[0..15] OF _sOUTPUT_REF* ² * ³	-	将输出信号_sOUTPUT_REF型的数组变量指定为基于开关的ON/OFF模式数据计算出的数字输出的ON/OFF时刻的输出对象。数组元素编号表示轨段编号。 通过将该数组变量指定为NX_AryDOutTimeStamp指令的输入输出变量，将实际的数字输出设为ON/OFF。
TrackOptions	轨段选项	ARRAY[0..15] OF _sTRACK_REF* ² * ³	-	将轨段选项结构体_sTRACK_REF型的数组变量指定为开关的动作条件。数组元素编号表示轨段编号。

*1. 请使用在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. 数组元素的开始编号为“0”以外时发生异常。此外，2维以上数组时也会发生异常。

*3. Outputs的元素数和TrackOptions的元素数不一致时发生异常。

功能说明

- 本指令将轴反馈位置到达“Switches”设定位置的时刻作为数字输出ON/OFF的时刻。
- 本指令必须与支持NX_AryDOutTimeStamp指令及时间戳方式的数字输出单元组合使用。
- NX_AryDOutTimeStamp 指令会按指定的时间戳，将指定的数字输出设为 ON/OFF。通过使用时间戳方式，可以不受控制处理的时间影响，在任意时间将数字输出信号设为ON/OFF。



使用注意事项

- 仅通过本指令不能将数字输出设为ON/OFF。
- 本指令使用NX系列编码器输入单元，变化时刻获取功能需处于动作状态。
以下场合，变化时刻获取功能不动作。
 - 使用不支持变化时刻获取功能的编码器输入或伺服驱动器时
 - NX系列编码器输入单元的对象6010Hex(时间戳)未分配至PDO时
 - 将EtherCAT耦合器单元的[DC有效]设为[无效]时
- 即使未更新时间戳，本指令也不会发生异常。此时也计算ON/OFF时刻，但结果不是预期值。
请通过Sysmac Studio的MC监控表或可视窗口确认轴变量的要素“TimeStamp(时间戳)”更新后再使用本指令。
- 与支持NX_AryDOutTimeStamp指令及时间戳方式的数字输出单元组合使用时，最小ON/OFF幅度与任务周期值及转速值成正比。
例如，转台旋转1圈360°的转速为800r/min、任务周期为500μs时，最小ON/OFF幅度为5°。仅将任务周期变更为1000μs时，最小ON/OFF幅度为10°。
- 请指定开关结构体型变量的FirstOnPosition、LastOnPosition及Duration的值，使数字输出的ON/OFF幅度大于最小ON/OFF幅度。小于最小ON/OFF幅度时，实际的数字输出可能不会执行ON/OFF。

- 本指令根据轴的当前位置和当前速度计算到达指定位置的时间戳。算出的时间戳精度受编码器分辨率和轴转速的影响。编码器分辨率低或轴转速低时，误差增大。最大误差标准可通过下式计算。

$$\text{时间戳的最大误差(s)} = 180 / [\text{编码器分辨率(脉冲/圈)} \times \text{转速(r/min)}]$$

编码器分辨率和转速对应的时间戳的最大误差示例如下。

编码器分辨率 (脉冲/圈)	转速 (r/min)	算出的时间戳最大误差 (μ s)
3600	400	± 125.0
	800	± 62.5
131072	400	± 3.4
	800	± 1.7

此外，如果轴进行急剧的加减速，计算误差可能进一步增大，因此请在轴处于恒速状态下使用。

请进行充分的动作验证，并确认安全。

- 指定未使用轴时，或MC试运行中时，Enable为TRUE时Busy变为TRUE，InOperation和Error变为FALSE。
- 请勿配置两个以上名称相同的实例。否则可能导致输出异常。

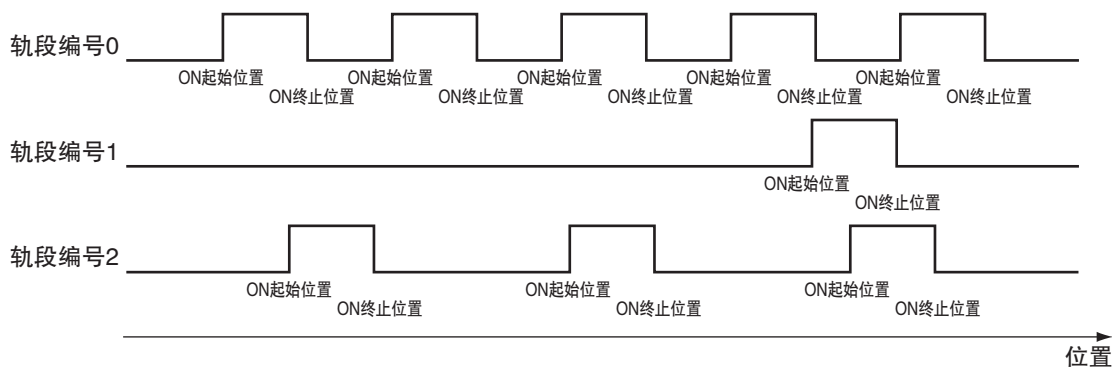
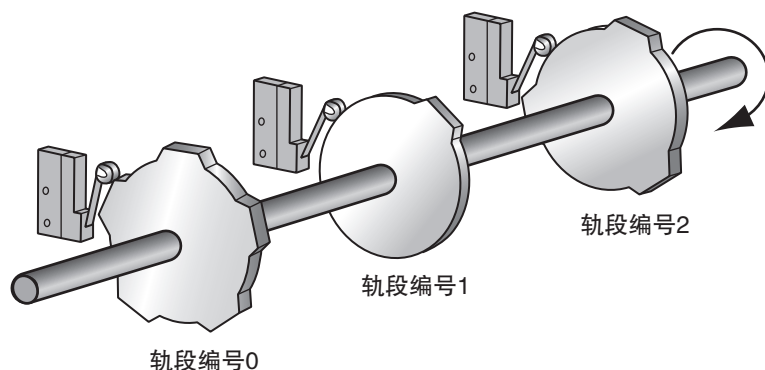


参考

关于NX_AryDOutTimeStamp指令，请参阅 □ “NJ/NX系列 指令基准手册 基本篇(SBCA-360)” 或 □ “NY系列指令基准手册基本篇(SBCA-437)”。

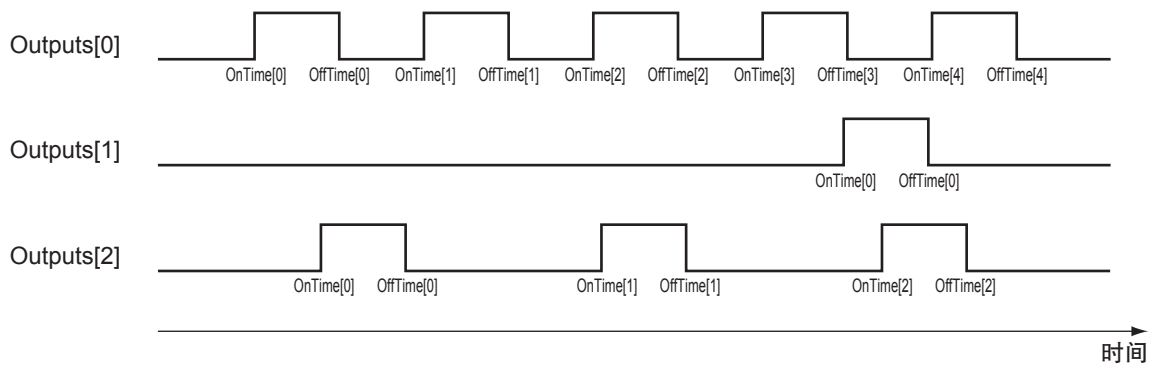
指令详情

本指令通过控制器程序，实现根据凸轮旋转角度控制传感器输出信号的机械式凸轮开关的功能。1个轨段相当于1个凸轮。



TrackNumber(轨段编号)相当于凸轮编号。FirstOnPosition(ON起始位置)、LastOnPosition(ON终止位置)相当于凸轮形状。

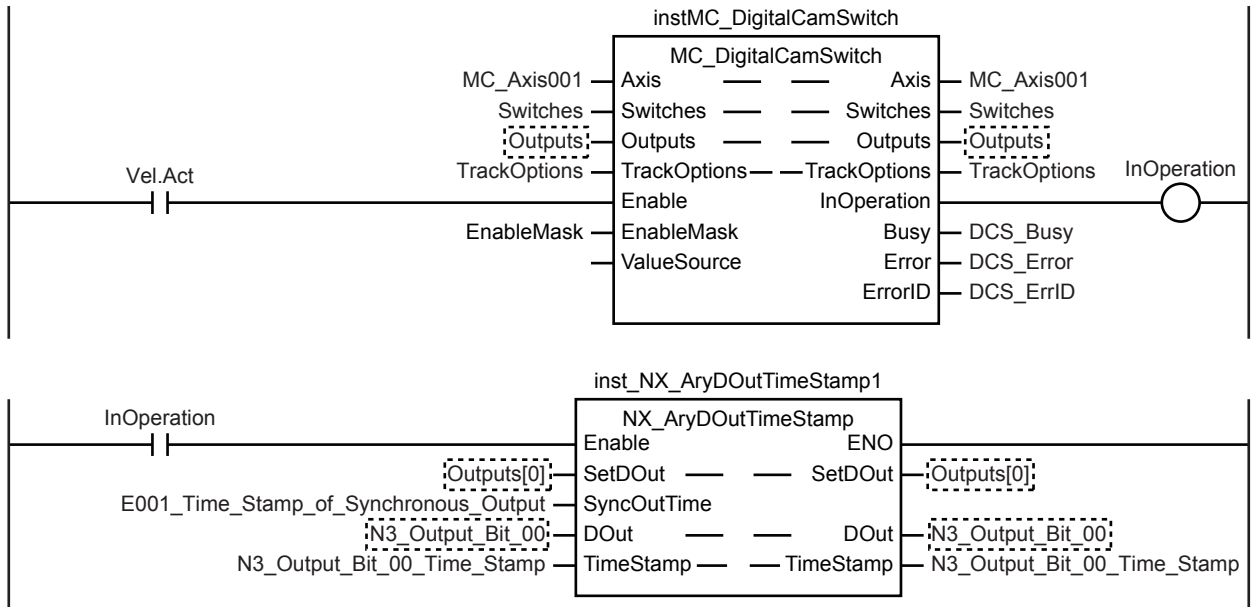
MC_DigitalCamSwitch指令对到达设定ON起始位置/ON终止位置的OnTime(ON时刻)/OffTime(OFF时刻)进行计算，并将结果保存在Outputs(输出信号)指定的参数中。



NX_AryDOutTimeStamp指令根据SetDOut(输出脉冲)指定的参数的OnTime(ON时刻)/OffTime(OFF时刻), 将实际的数字输出设为ON/OFF。

在SetDOut(输出脉冲)中, 指定由MC_DigitalCamSwitch指令的Outputs(输出信号)参数指定的数组变量的元素。

在DOut(DOut单元输出触点)中, 指定分配至时间戳方式对应的数字输出单元的输出触点的设备变量, 作为实际的数字输出。



下面对指令变量进行说明。

● Enable(有效)

- Enable(有效)TRUE期间执行指令。FALSE时不更新“Outputs”的值。

● EnableMask(轨段有效)

- Enable(有效)为TRUE时, 可通过EnableMask(轨段有效)指令, 指定各轨段有效或无效。要关闭数字输出单元的输时, 请将对象轨段设为无效。
- 第0位表示轨段编号0, 第15位表示轨段编号15。将各位的值设为“1”时对象轨段有效, 设为“0”则无效。将位值由“1”变为“0”后, 对象轨段的数字输出OFF。
- EnableMask中指定的值将被反映在对应轨段编号的EnableOut中。

● 开关结构体(_sCAMSWITCH_REF型)

开关结构体(_sCAMSWITCH_REF型)为结构体型数据类型,用于指定输出信号的ON/OFF模式。本指令最多可将256种ON/OFF模式指定为数组变量。1个轨段中,最多可指定16种ON/OFF模式。

下面对开关结构体的要素进行说明。

变量	名称	数据类型	有效范围	初始值	内容
TrackNumber	轨段编号	UINT	0 ~ 15 ^{*1}	0	指定对象轨段编号。
FirstOnPosition	ON起始位置	LREAL	负数、正数、“0”	0	指定输出ON的位置。 ^{*2}
LastOnPosition	ON终止位置	LREAL	负数、正数、“0”	0	指定输出OFF的位置。 ^{*2} 将开关模式选择指定为位置基准时有效。
AxisDirection	轴方向选择	_eMC_DIRECTION	0:_mcPositiveDirection 2:_mcNegativeDirection 4:_mcNoDirection	0 ^{*3}	指定旋转方向。 0: 指定为正方向 2: 指定为负方向 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ON的时间。将开关模式选择指定为时间基准时有效。

*1. 可指定到Outputs(输出信号)中指定的变量元素编号为止。

*2. 以指令单位表示。指令单位有[mm]、[μm]、[nm]、[degree]、[inch]或[pulse]。

计数模式为[线性模式]时,设定范围为脉冲单位转换后的带符号整数型40位(0x8000000000 ~ 7FFFFFFF)。计数模式为[旋转模式]时,设定范围在环计数器下限值以上、环计数器上限值以下。

*3. 有效范围为枚举体的变量,其实际初始值不是数值,而是枚举元素。

*4. 指定1:_mcPositionBased(位置基准)时,根据FirstOnPosition(ON起始位置)和LastOnPosition(ON终止位置)的值进行动作。与Duration(ON时间)的值无关。

*5. 指定2:_mcTimeBased(时间基准)时,根据FirstOnPosition(ON起始位置)和Duration(ON时间)的值进行动作。与LastOnPosition(ON终止位置)的值无关。

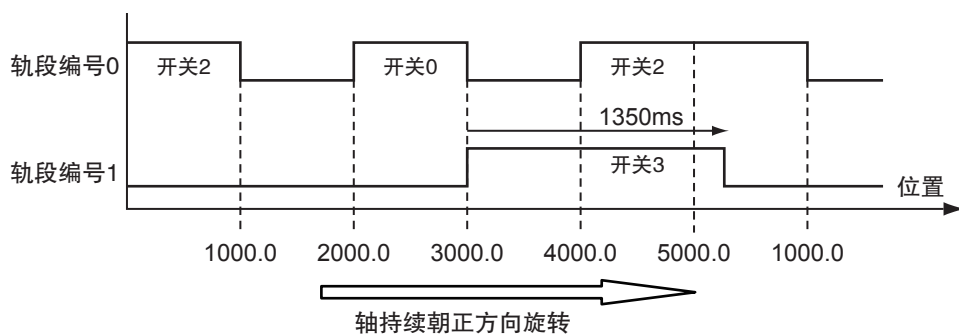
设定示例

变量	名称	开关0	开关1	开关2	开关3	...	开关255
TrackNumber	轨段编号	0	0	0	1		
FirstOnPosition	ON起始位置	2000.0	2500.0	4000.0	3000.0		
LastOnPosition	ON终止位置	3000.0	3000.0	1000.0	_ *1		
AxisDirection	轴方向选择	指定为正方向	指定为负方向	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		

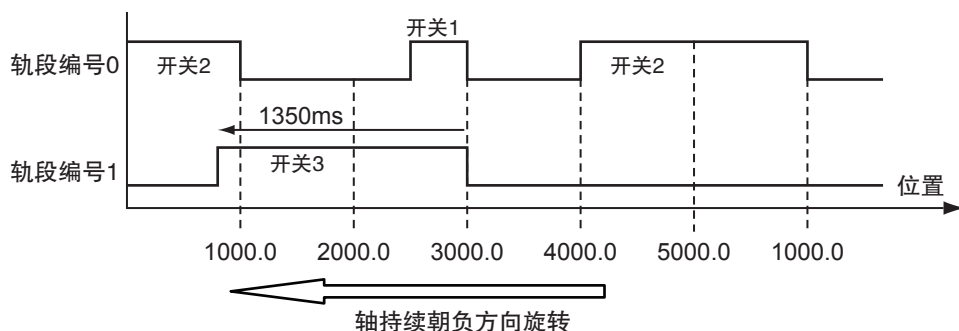
*1. 开关模式选择为时间基准时,在ON起始位置和ON时间进行动作。与ON终止位置无关。

*2. 开关模式选择为位置基准时,在ON起始位置和ON终止位置进行动作。与ON时间无关。

轴持续朝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，环计数器的范围为0~5000。



轴持续朝负方向旋转时的动作如下所示。轴参数的设定与上述相同。



● 输出信号结构体(_sOUTPUT_REF型)

输出信号结构体(_sOUTPUT_REF型)为结构体型数据类型，用于表示基于开关的ON/OFF模式数据计算出的数字信号的ON/OFF时刻。本指令最多将16个元素视作Outputs(输出信号)的数组变量。Outputs(输出信号)的数组元素编号表示轨段编号。

下面对输出信号结构体的要素进行说明。

变量	名称	数据类型	有效范围	内容
EnableOut	输出有效	BOOL	TRUE, FALSE	表示对应轨段编号输出的有效/无效。反映相当于EnableMask的相同轨段编号的位值。 ^{*1} TRUE:对应轨段编号的输出有效 FALSE:对应轨段编号的输出无效
OnTime	ON时刻	ARRAY[0..15] OF ULINT	正数、“0”	表示将数字输出设为ON的时间戳。表示以NX系列编码器输入单元内的时刻为基准的时间戳。按任务周期更新值。单位为[ns]。
OffTime	OFF时刻	ARRAY[0..15] OF ULINT	正数、“0”	表示将数字输出设为OFF的时间戳。表示以NX系列编码器输入单元内的时刻为基准的时间戳。按任务周期更新值。单位为[ns]。

*1. EnableMask的i位值反映在Outputs[i].EnableOut中。

● 轨段选项结构体(_sTRACK_REF型)

轨段选项结构体(_sTRACK_REF型)为结构体型数据类型，用于指定开关动作条件。本指令最多可将16种输出条件指定为数组变量。

请将TrackOptions和Outputs中指定变量的元素数设为同一元素数。

下面对轨段选项结构体的要素进行说明。

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	补偿输出ON的时间。正数时将时间延迟，负数时将时间提前。
OffCompensation	OFF时间补偿	TIME	T#-1s ~ T#1s *1	T#0s	补偿输出OFF的时间。正数时将时间延迟，负数时将时间提前。

*1 计数模式使用旋转模式时，根据CPU单元的版本，存在以下限制。

- 对于Ver.1.09以上版本的CPU单元，设定超过轴的 \pm 半圈范围的值时InOperation(动作中)为FALSE。
- Ver.1.08以下版本的CPU单元的有效范围为“T#-1s ~ T#0s”。

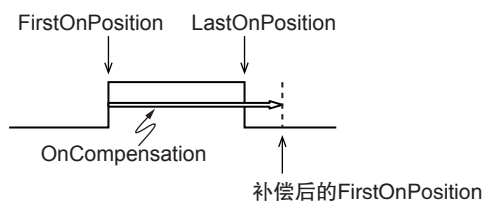
OnCompensation(ON时间补偿)及OffCompensation(OFF时间补偿)用于补偿机械的细微动作延迟或偏移等。



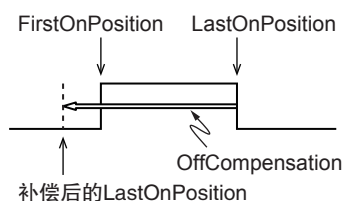
安全要点

请务必在轴速恒定的情况下使用MC_DigitalCamSwitch(数字凸轮开关有效)指令。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)，当轴速骤变时，会发生以下动作。

- OnCompensation(ON时间补偿)或OffCompensation(OFF时间补偿)的值与轴旋转半圈以上的时间相当时，InOperation(工作中)为FALSE。
- OnCompensation(ON时间补偿)的值超过LastOnPosition(ON终止位置)时，输出时间不定。



- OffCompensation(OFF时间补偿)的值超过FirstOnPosition(ON起始位置)时，输出时间不定。





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，请勿将FirstOnPosition(ON起始位置)和LastOnPosition(ON终止位置)的位置关系反转。
输出时间不定。
动作请参阅前述的“安全要点”。



版本相关信息

计数模式使用旋转模式时，根据CPU单元的版本，存在以下限制。

Ver.1.09 以上版本和Ver.1.08以下版本CPU单元的有效范围不同。CPU单元升级为Ver.1.09以上版本时，请注意有效范围。

Ver.1.09以上版本的CPU单元

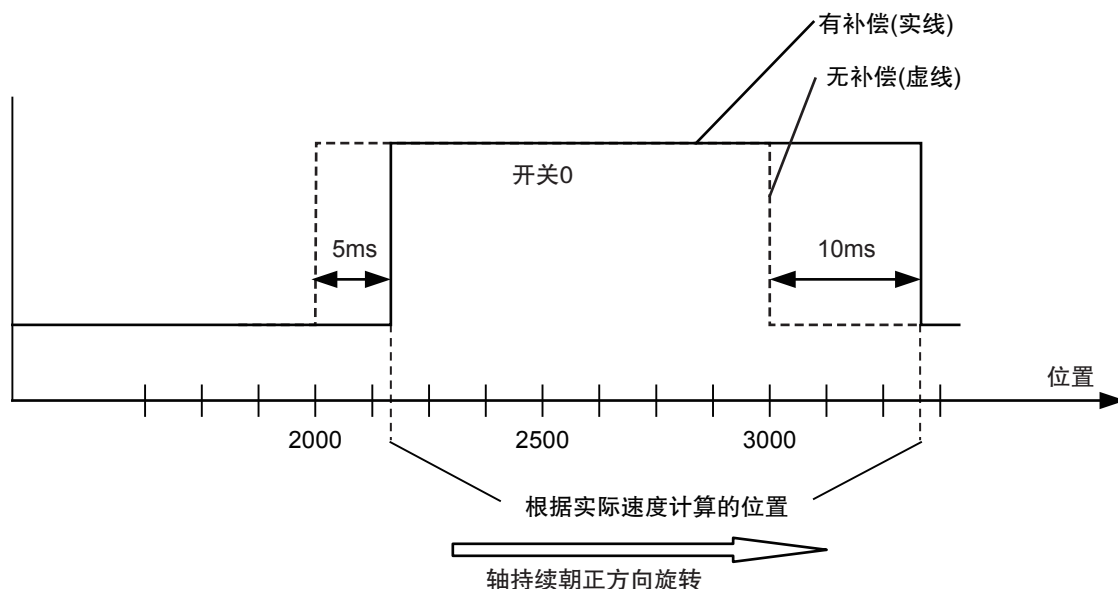
有效范围为“ $T\#-1s \sim T\#1s$ ”，但设定存在以下限制。

- 设定范围请勿超过轴旋转半圈的时间。
例如，以 500rpm 的速度旋转时，转 1 圈的时间为 120ms。由于是旋转半圈的时间，请将 OnCompensation(ON时间补偿)和OffCompensation(OFF时间补偿)设为-60ms到60ms之间的值。
- 设定时间超过轴旋转半圈的时间时，InOperation(动作中)变为FALSE的同时，EnableOut(输出有效)也变为FALSE。请务必在使用的同时确认InOperation(动作中)的状态。
- 如上所述InOperation(动作中)变为FALSE后，若值恢复到正常范围内，则InOperation(动作中)变为TRUE。

Ver.1.08以下版本的CPU单元

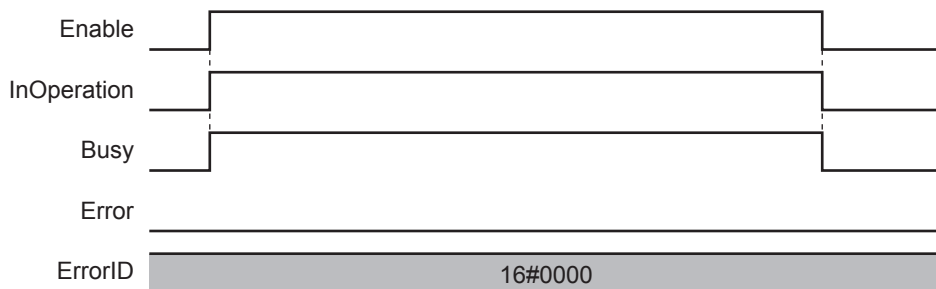
- 有效范围为“ $T\#-1s \sim T\#0s$ ”。

前述的“设定示例(P.3-382)”中，进一步将OnCompensation(ON时间补偿)和OffCompensation(OFF时间补偿)分别设为 $T\#5ms$ 和 $T\#10ms$ 时的动作如下所示。



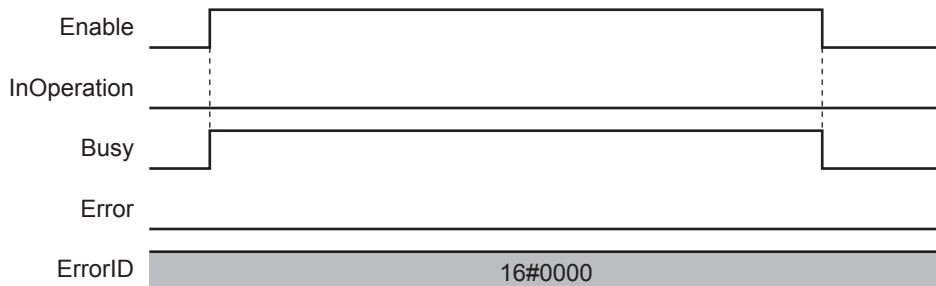
时序图

执行本指令后的时序图如下所示。



表示指定了未使用轴或MC试运行中的时序图。

Ver.1.09 以上版本的 CPU 单元中，计数模式为旋转模式时，若 OnCompensation(ON 时间补偿) 或 OffCompensation(OFF时间补偿)设定的时间超过轴的半圈时间，则时序图与此相同。



此时，Outputs的EnableOut变为FALSE。

重启运动指令

输入为“Enable型”指令时，不会重启运动指令。

多重启动运动指令

本指令的各个指令实例独立动作，没有运动指令多重启动的限制。

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或

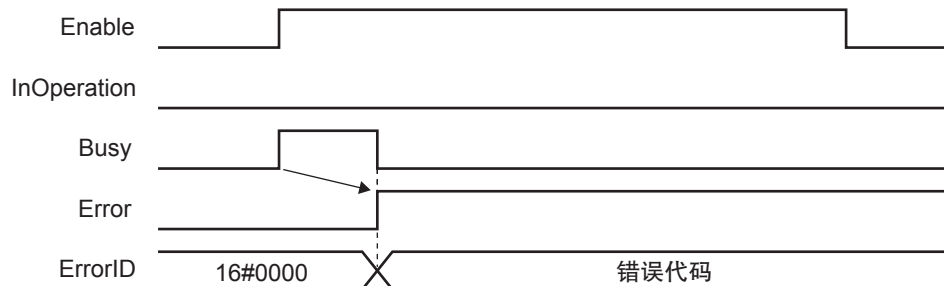
□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

异常

未能启动本指令时，本指令会发生异常，Error(错误)变为TRUE。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



此时，Outputs的EnableOut变为FALSE。

● 错误代码

关于指令发生的异常，请参阅 □ “A-1 错误代码一览(P.A-2)”。

示例程序

下面，对前述“设定示例(P.3-389)”中所示的示例程序进行说明。

使用设备

该示例程序中使用的设备如下所示。

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. 将节点地址设为“1”，设备名称设为“E001”。
- *2. 将NX单元编号设为“1”，设备名称设为“N1”。分配至轴1。
- *3. 将NX单元编号设为“2”，设备名称设为“N2”。分配至轴2。
- *4. 将NX单元编号设为“3”，设备名称设为“N3”。

参数设定

该示例程序中使用的最少的必要设定如下所示。

● 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

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

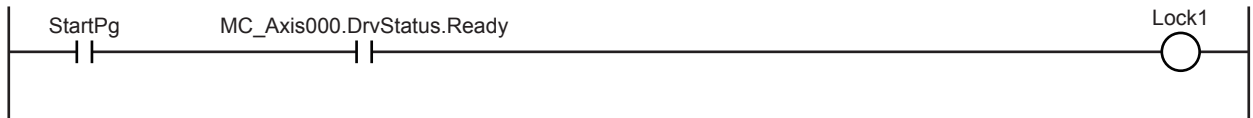
● 主要变量

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。 元素编号对应开关编号。
Outputs	ARRAY[0..1] OF _sOUTPUT_REF	-	输入输出变量“Outputs”的输入参数。 元素编号对应轨段编号。
TrackOptions	ARRAY[0..1] OF _sTRACK_REF	-	输入输出变量“TrackOptions”的输入参数。 元素编号对应轨段编号。
EnableMask	WORD	16#0003	输入变量“EnableMask”的输入参数。 轨段0和轨段1有效。

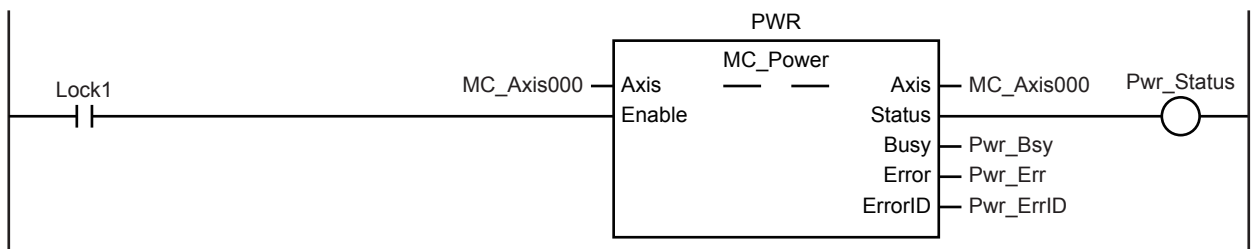
*1. EtherCAT耦合器单元的I/O入口，必须事先追加0x200A:02(Time Stamp of Synchronous Output)。

● 示例程序

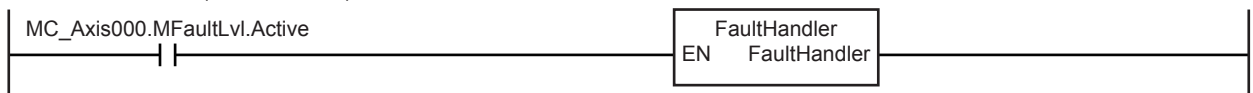
触点StartPg变为TRUE时，确认轴1的伺服驱动器处于伺服准备就绪状态



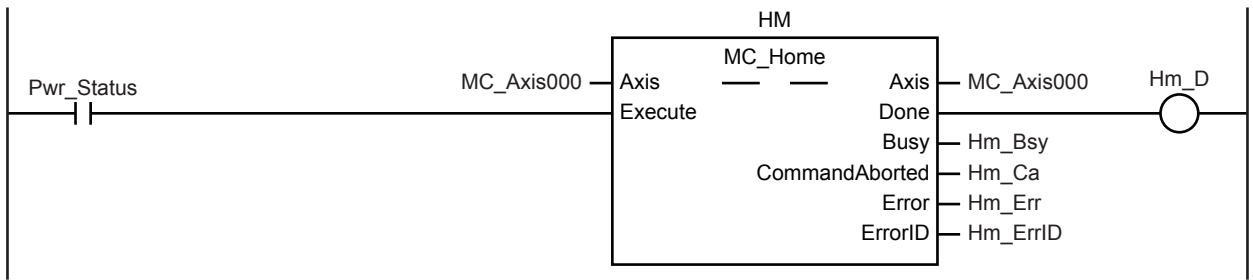
如果轴1处于伺服准备就绪状态，则设为伺服ON状态



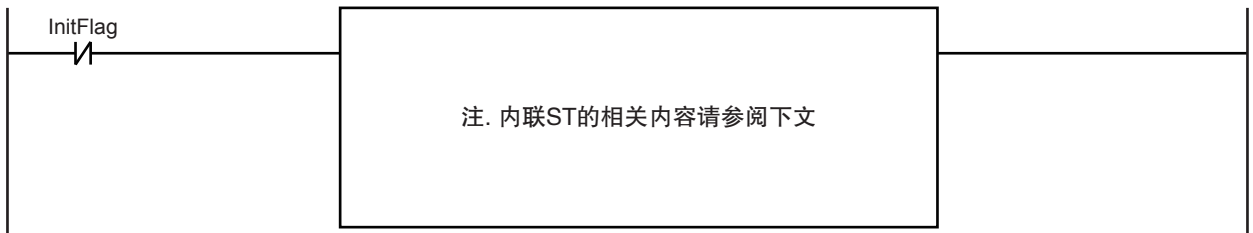
轴构成发生轻度故障程度的异常时，执行设备的异常处理(FaultHandler)。
发生异常时的处理(FaultHandler)由客户根据装置进行编程



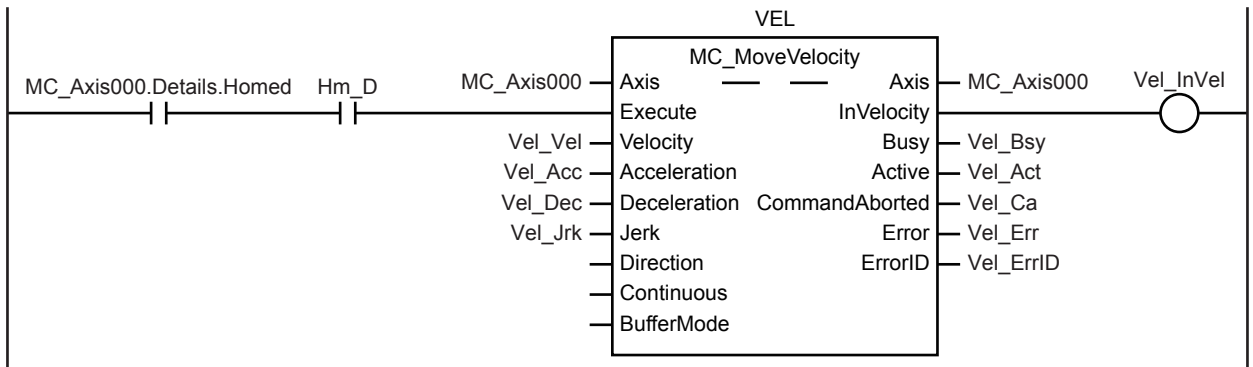
轴1变为伺服ON状态，若原点未确定，则执行原点复位，确定原点



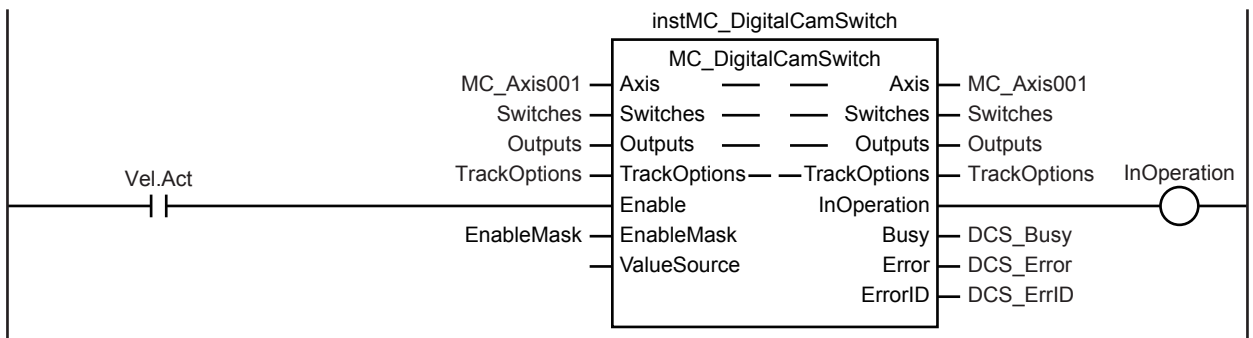
设定MC_MoveVelocity(速度控制)指令的参数和MC_DigitalCamSwitch(数字凸轮开关有效)指令的Switches的参数



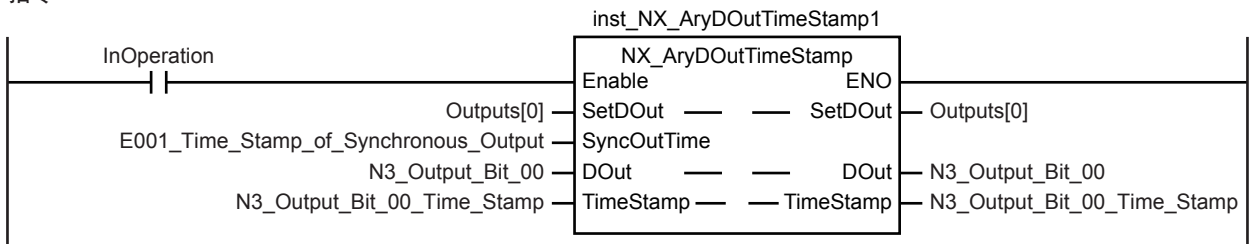
轴1已确定原点，执行MC_MoveVelocity(速度控制)指令

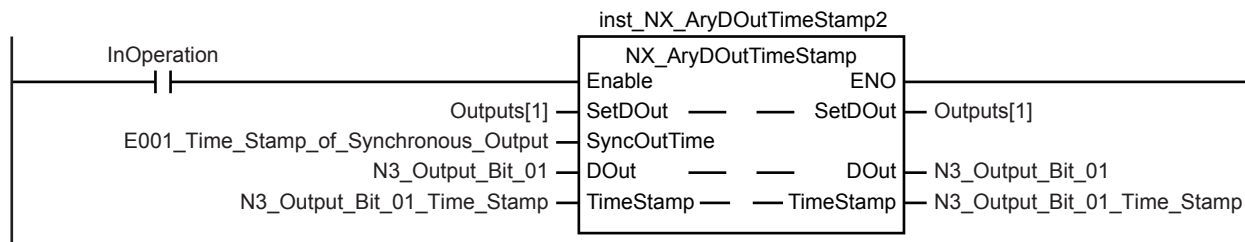


执行轴1的MC_MoveVelocity(速度控制)指令后，执行轴2的MC_DigitalCamSwitch(数字凸轮开关有效)指令



执行轴2的MC_DigitalCamSwitch(数字凸轮开关有效)指令后，执行NX_AryDOutTimeStamp(时间戳数字输出数组写入)指令





内联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	-	分配给MC_Power的实例PWR的输入变量Enable的变量。
Switches	ARRAY[0..3] OF _sCAMSWITCH_REF	-	输入输出变量“Switches”的输入参数。 元素编号对应开关编号。
Outputs	ARRAY[0..1] OF _sOUTPUT_REF	-	输入输出变量“Outputs”的输入参数。 元素编号对应轨段编号。
TrackOptions	ARRAY[0..1] OF _sTRACK_REF	-	输入输出变量“TrackOptions”的输入参数。 元素编号对应轨段编号。
EnableMask	WORD	16#0003	输入变量“EnableMask”的输入参数。 轨段0和轨段1有效。
Hm_Ex	BOOL	FALSE	该变量由FALSE→TRUE时,启动MC_Home的实例HM。
Vel_Ex	BOOL	FALSE	该变量由FALSE→TRUE时,启动MC_MoveVelocity的实例VEL。

*1. EtherCAT耦合器单元的I/O入口, 必须事先追加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;

//StartPg为TRUE时，确认伺服驱动器处于伺服准备就绪状态，
//将轴1设为伺服ON状态。
//如果未处于伺服准备就绪状态，则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En:=TRUE;
  ELSE
    Pwr_En:=FALSE;
  END_IF;

//轴1发生轻度故障后，执行异常时处理FaultHandler。
//发生异常时的处理(FaultHandler)由客户根据装置进行编程。
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
  FaultHandler();
END_IF;

//在伺服ON状态下，且原点未确定时，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_AryDOOutTimeStamp1(
    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_AryDOOutTimeStamp2(
    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

计算指定的时间戳的轴位置。

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

本指令仅对分配了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Ver.1.06以上的CPU单元和Ver.1.07以上的Sysmac Studio组合时使用。

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期间执行指令。
TimeStamp	时间戳	ULINT	正数或“0”	0	指定需计算位置的时间戳。 指定以时间戳方式对应的数字输入单元和编码器输入单元内的时刻为基准的时间戳。单位为[ns]。
ValueSource (Reserved)	输入信息	_sMC_SO URCE	-	-	(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	控制中变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CalcPosition	计算位置	LREAL	负数、正数、“0”	输出指定的时间戳的位置。 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为正常。

*1. 关于指令单位，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*2. □ 请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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含有异常因素时	异常已解除时

输入输出变量

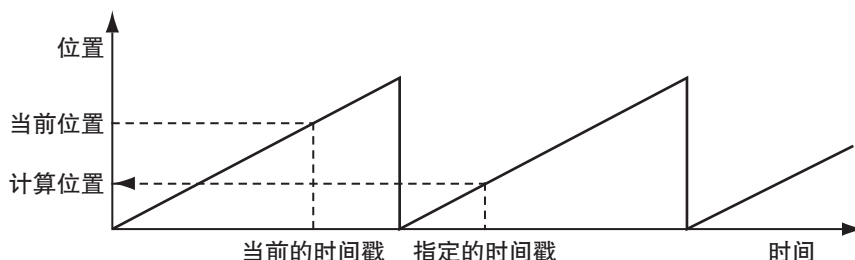
输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	-	指定轴。*1 *2

*1. 请使用在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

- 本指令以轴的反馈当前位置和时间戳为基准，计算输入变量指定的时间戳的反馈当前位置。
- 请在轴中指定NX系列编码器输入单元。
- 轴参数的计数模式为线性模式时，若算出的位置为溢出或下溢，则算出值为溢出或下溢的值。此时，不会发生异常。





使用注意事项

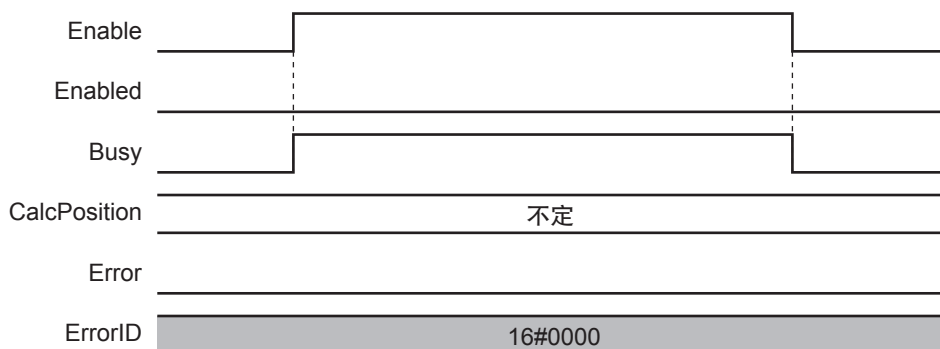
- 本指令使用NX系列编码器输入单元，变化时刻获取功能需处于动作状态。
以下场合，变化时刻获取功能不动作。
 - 使用不支持变化时刻获取功能的编码器输入或伺服驱动器时
 - NX系列编码器输入单元的对象6010Hex(时间戳)未分配至PDO时
 - 将EtherCAT耦合器单元的[DC有效]设为[无效]时
 - 即使未更新时间戳，本指令也不会发生异常。此时也将算出位置，但不会为指定的时间戳中的位置。
请通过Sysmac Studio的MC监控表或可视窗口确认轴变量的要素 “TimeStamp(时间戳)” 更新后再使用本指令。
 - 本指令以轴的当前位置和当前速度为基准，计算其在指定的时间戳中的位置。如果轴进行急剧的加减速，计算误差可能增大，因此请在轴处于恒速状态下使用。
请进行充分的动作验证，并确认安全。
 - 指定未使用轴时，或MC试运行中时，Enable为TRUE时Busy变为TRUE，Enabled和Error变为FALSE。
 - 请勿配置两个以上名称相同的实例。否则可能导致输出异常。
-

时序图

执行本指令后的时序图如下所示。



表示指定了未使用轴或MC试运行中的时序图。



重启运动指令

输入为“Enable型”指令时，不会重启运动指令。

多重启动运动指令

本指令的各个指令独立动作，没有运动指令多重启动的限制。

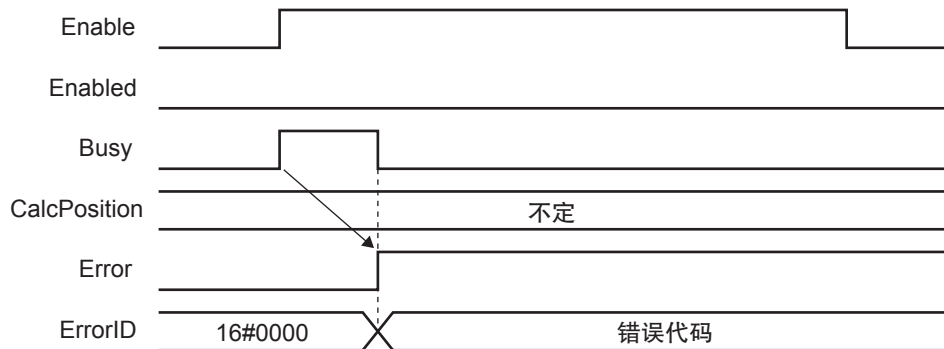
多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

异常

未能启动本指令时，本指令会发生异常，Error(错误)变为TRUE。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 “A-1 错误代码一览(P.A-2)”。

示例程序

下面，对测量在皮带输送机上移动的工件间距离的示例程序进行说明。

使用设备

该示例程序中使用的设备如下所示。

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. 将节点地址设为“1”，设备名称设为“E001”。

*2. 将NX单元编号设为“1”，设备名称设为“N1”。分配至轴1。

*3. 将NX单元编号设为“2”，设备名称设为“N2”。分配至轴2。

*4. 将NX单元编号设为“3”，设备名称设为“N3”。

参数设定

该示例程序中使用的最少的必要设定如下所示。

● 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

环计数器

轴	上限值	下限值
轴1	360	0
轴2	360	0

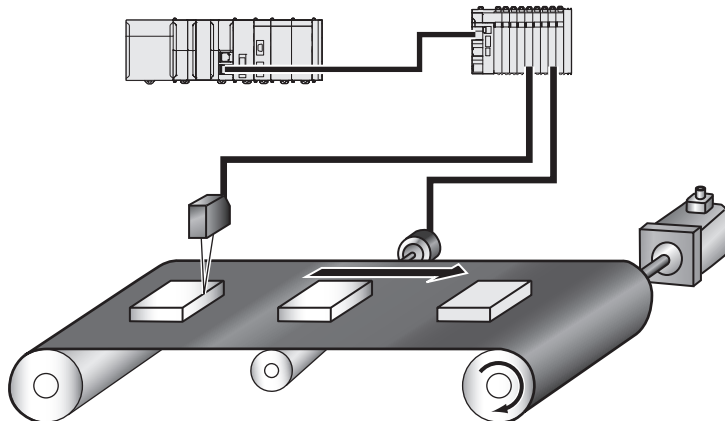
坐标单位选择

轴	坐标单位
轴1	degree
轴2	degree

动作示例

传感器检出工件时传感器输出变为 ON，工件通过时传感器输出变为 OFF。传感器检出下一个工件时，传感器输出再次变为 ON。

根据传感器输出为 ON 时的时间戳，计算编码器输入的位置。两个位置之差为工件间的距离。



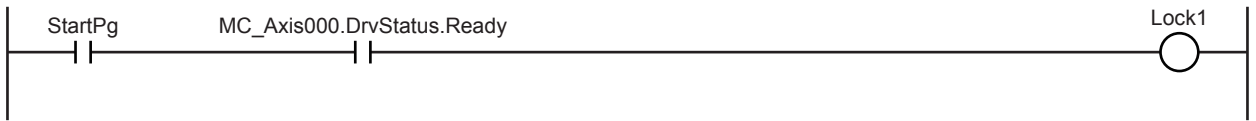
梯形图

● 主要变量

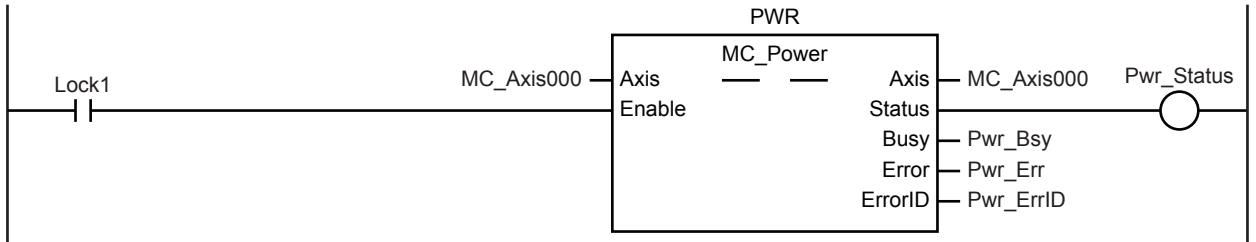
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	轴1的轴变量。
MC_Axis001	_sAXIS_REF	-	轴2的轴变量。
N3_Input_Bit_00	BOOL	-	设备变量。
N3_Input_Bit_00_Time_Stamp	ULINT	-	设备变量。
Position	ARRAY[0..1] OF LREAL	-	保存计算位置。
Count	ARRAY[0..1] OF ULINT	-	保存多圈数。
FirstPoint	UINT	-	处理用的变量。
LastPoint	UINT	-	处理用的变量。
Distance	LREAL	-	工件间的距离。

● 示例程序

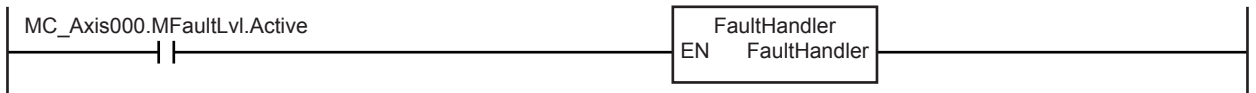
触点StartPg变为TRUE时，确认轴1的伺服驱动器处于伺服准备就绪状态



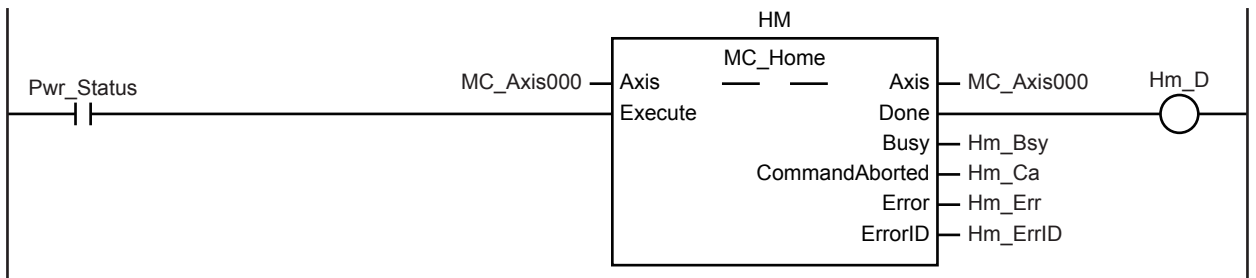
如果轴1处于伺服准备就绪状态，则设为伺服ON状态



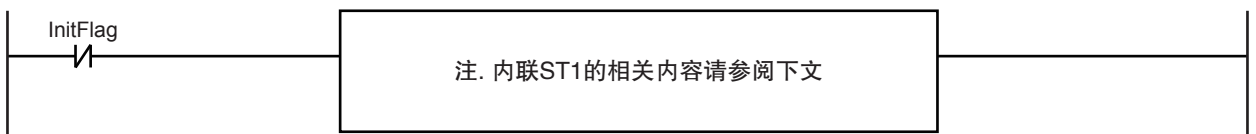
轴构成发生轻度故障程度的异常时，执行设备的异常处理(FaultHandler)。发生异常时的处理(FaultHandler)由客户根据装置进行编程



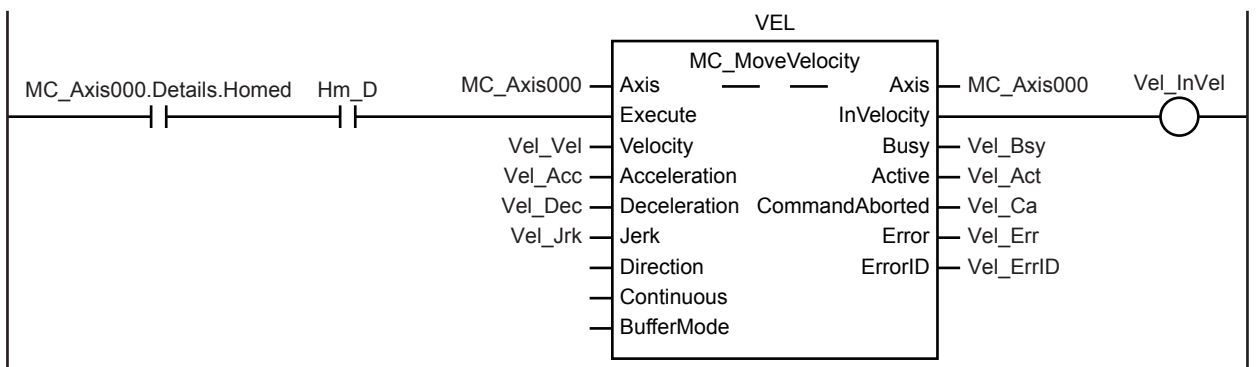
轴1变为伺服ON 状态，若原点未确定，则执行原点复位，确定原点



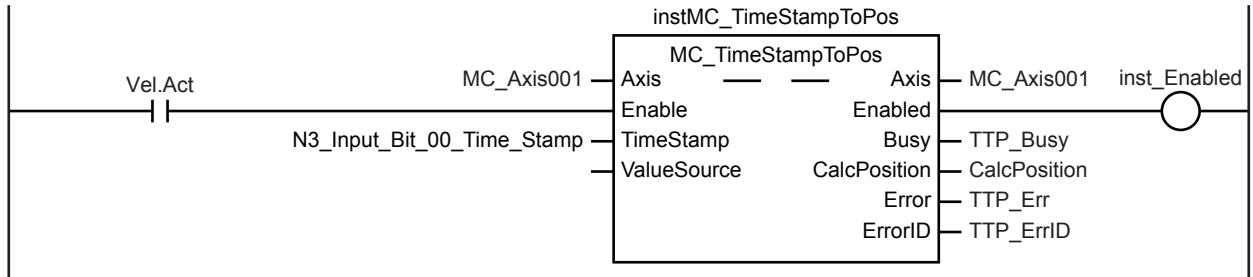
设定MC_MoveVelocity(速度控制)指令的参数



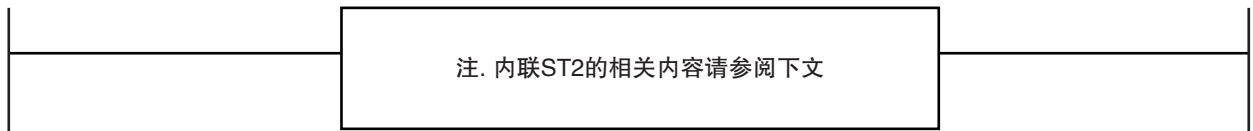
轴1已确定原点后，执行MC_MoveVelocity(速度控制)指令



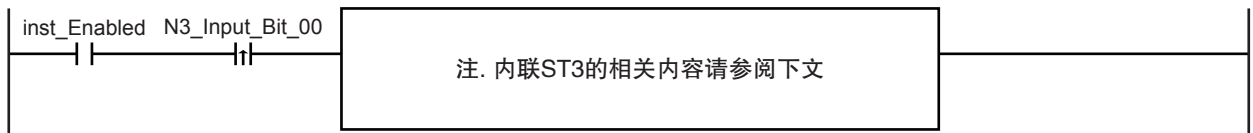
执行轴1的MC_MoveVelocity(速度控制)指令后, 执行轴2的MC_TimeStampToPos(时间戳→轴位置计算)指令



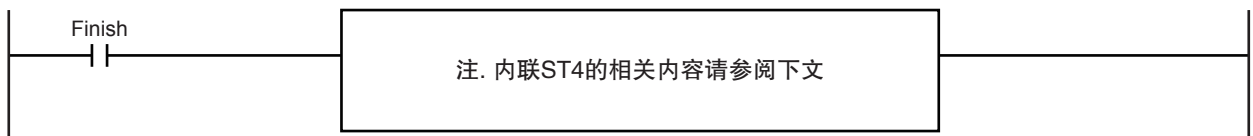
对编码器轴的多圈数进行计数(本次值<上次值时, 判断为超过环计数器的上限值)



传感器输出的ON上升沿时, 获取CalcPosition(计算位置)(交替保存至Position[0]和Position[1])



检测到第二个工件后, 计算工件间的距离



内联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;
```

内联ST2的内容

```
IF MC_Axis001.Act.Pos < PreAxis001ActPos THEN
  Inc(RotaryCount);
END_IF;
PreAxis001ActPos := MC_Axis001.Act.Pos;
```

内联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;
```

内联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	该变量由FALSE→TRUE时, 启动MC_Home的实例HM。
Vel_Ex	BOOL	FALSE	该变量由FALSE→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	-	保存计算位置。
Count	ARRAY[0..1] OF ULINT	-	保存多圈数。
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;
```

//StartPg为TRUE时, 确认伺服驱动器处于伺服准备就绪状态,

//将轴1设为伺服ON状态。

//如果未处于伺服准备就绪状态, 则设为伺服OFF。

```
IF (StartPg=TRUE)
```

```
    AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
```

```
        Pwr_En:=TRUE;
```

```
    ELSE
```

```
        Pwr_En:=FALSE;
```

```
    END_IF;
```

//轴1发生轻度故障后, 执行异常时处理FaultHandler。

//发生异常时的处理(FaultHandler)由客户根据装置进行编程。

```
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
```

```
    FaultHandler();
```

```
END_IF;
```

//在伺服ON状态下, 且原点未确定时, 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;

//对编码器轴的多圈数进行计数。
//(本次值<上次值时, 判断为超过环计数器的上限值)
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)。
//(交替保存至Position[0]和Position[1])
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;

//检测到第二个工件后, 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

在固定周期内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_Periodic SyncVariables	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

NX1P2 CPU单元及NJ系列 CPU单元无法使用本指令。

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	在上升沿开始指令。
ExecID	执行ID	UINT	2	2	指定变量同步指定对象的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	控制中变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为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为正常。

*1. 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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	Enabled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含有异常因素时	异常已解除时

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. 请在使用在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, 在原始恒定周期任务和固定周期任务(执行优先度5)的任务间, 将轴变量按周期进行同步。
- 同步后的变量输出至指定对象任务的轴变量区域(系统变量)。



使用注意事项

- 本指令仅支持从原始恒定周期任务到固定周期任务(执行优先度5)的变量同步。
- ExecID(执行ID)中指定的值与对应轴分配的任务相同时, 本指令无异常, 保持Busy(执行中)不变。



参考

不同任务控制的轴的轴变量通过用户程序宣布外部变量并进行查看时, 根据用户程序的执行开始位置, 无法得知要查看的轴变量的更新时间点。
电子凸轮等控制的轴通过不同任务控制具有主从关系的应用时, 请使用MC_PeriodicSyncVariables(轴变量任务间固定周期同步)指令。

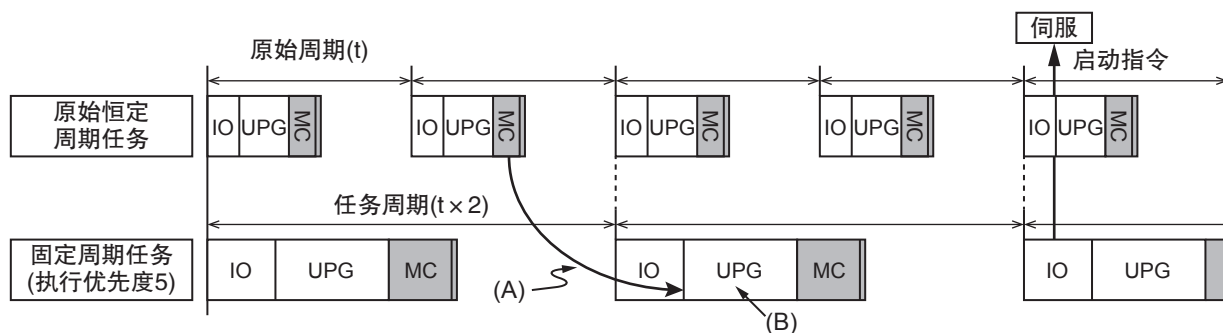
指令详情

下面对指令详细说明。

● 同步时间

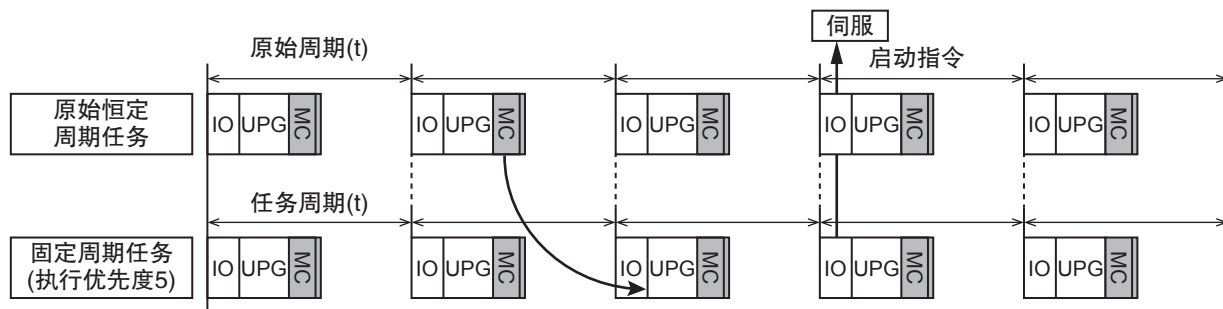
以原始恒定周期任务控制的轴的轴变量值可通过固定周期任务(执行优先度5)的用户程序查看。固定周期任务(执行优先度5)的用户程序执行过程中,原始恒定周期任务的轴变量值不会改写。固定周期任务(执行优先度5)的用户程序,将反映原始恒定周期任务与固定周期任务(执行优先度5)的任务周期起点重合的周期即将开始之前的原始恒定周期任务的执行结果。

将原始恒定周期任务的轴变量值反映至固定周期任务(执行优先度5)的时间如下所示。



记号	说明
(A)	轴变量的更新。 不论固定周期任务(执行优先度5)的用户程序执行开始位置如何,都会反映任务周期起点重合的周期即将开始之前的原始恒定周期任务的执行结果。
(B)	执行用户程序的过程中,轴变量值不会改写。

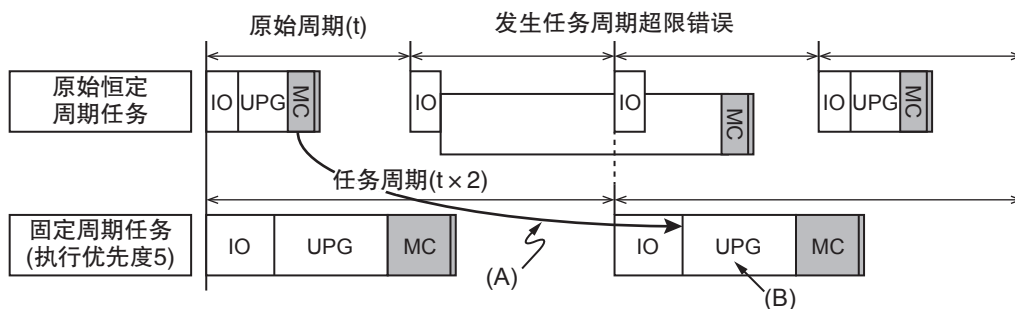
任务周期相同时也一样。





参考

发生任务周期超限时，不是反映任务周期起点重合的周期即将开始之前的任务周期的执行结果，而是反映再前一个周期的执行结果。



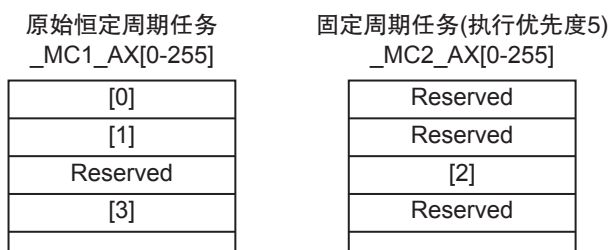
记号	说明
(A)	轴变量的更新。 发生任务周期超限时，反映再前一个周期的值。
(B)	执行用户程序的过程中，轴变量值不会改写。

● 变量的输出区域

同步后的变量输出至指定对象任务的轴变量区域(系统变量)。

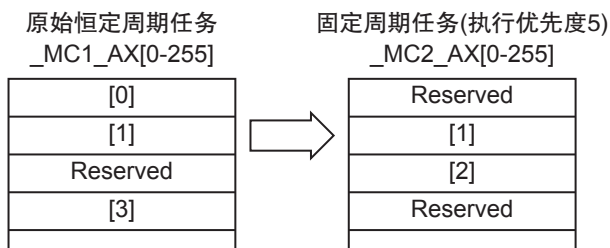
例如，轴参数的“运动控制”设定如下时，轴变量(系统变量)的下图区域变为更新对象。

轴号	运动控制(分配任务)
轴0	原始恒定周期任务
轴1	原始恒定周期任务
轴2	固定周期任务(执行优先级5)
轴3	原始恒定周期任务



(注) Reserved除基本设定外为初始值。

在MC_PeriodicSyncVariables(轴变量任务间固定周期同步)指令的“Axis(轴)”中指定“轴1”、“ExecID(执行ID)”中指定“10#2”并执行时，_MC2_Ax[1]的区域将被更新。



重启运动指令

输入为“Enable型”指令时，不会重启运动指令。

多重启动运动指令

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”。

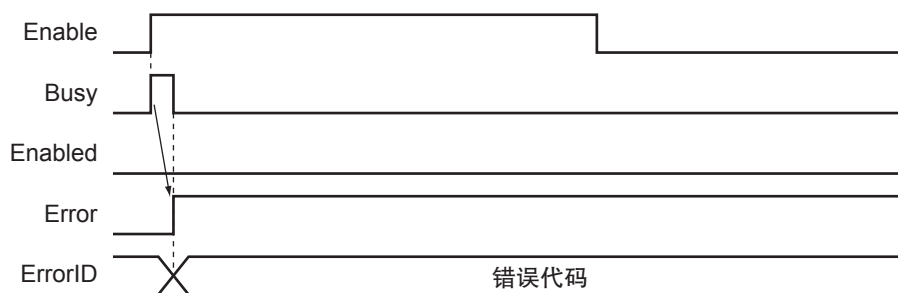
● 多重启动多个本指令

在正在执行MC_PeriodicSyncVariables(轴变量任务间固定周期同步)指令的轴中，启动其它实例的MC_PeriodicSyncVariables(轴变量任务间固定周期同步)指令后，两指令均执行。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，不执行参数的写入。保持指令执行前的值。可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



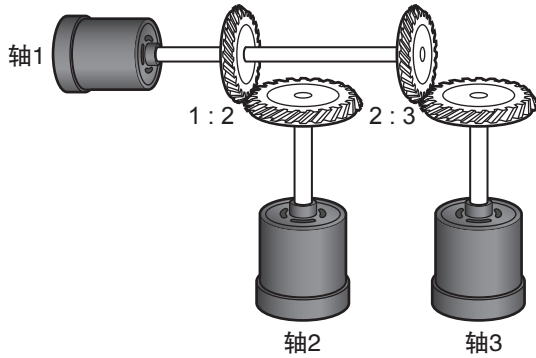
● 错误代码

关于指令发生的异常，请参阅 □ “A-1 错误代码一览(P.A-2)”。

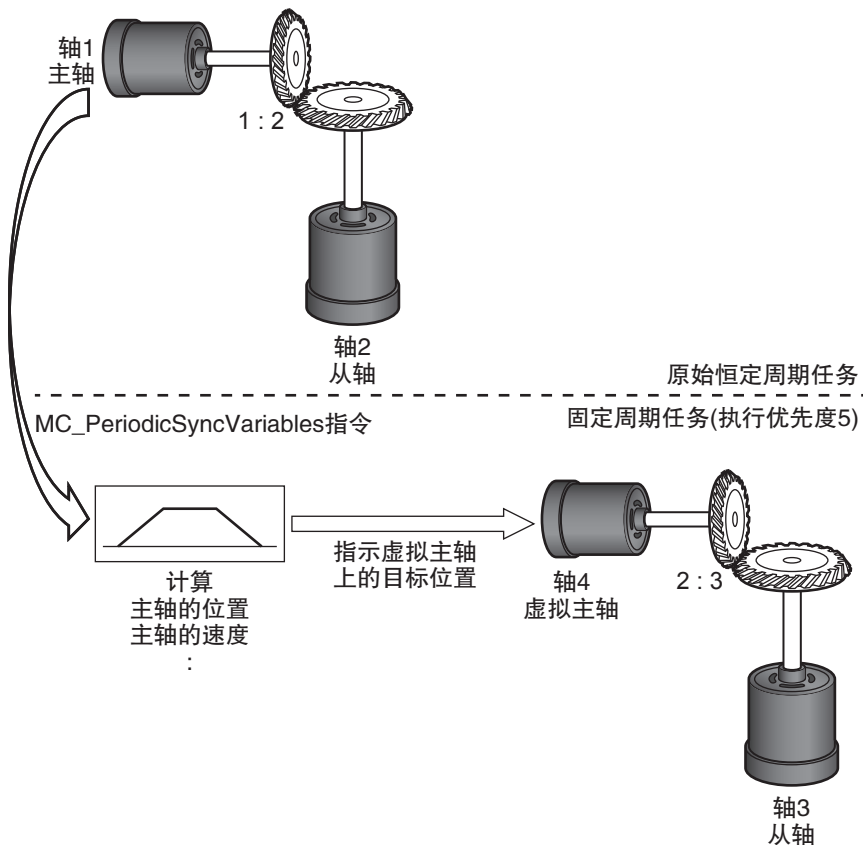
示例程序

在构成上，相对于主轴(轴1)，将高速且需要高精度的从轴(轴2)分配至原始恒定周期任务，将不需要精度的从轴(轴3)分配至固定周期任务(执行优先级5)进行控制。并将主轴(轴1)分配至原始恒定周期任务。

● 物理轴构成



● 逻辑轴构成



通过为原始恒定周期任务和固定周期任务(执行优先级5)分别编写程序，实现上述应用的动作。

原始恒定周期任务的程序

- 通过原始恒定周期任务，对分配至原始恒定周期任务的主轴(Primary_MasterAxis)执行MC_PeriodicSyncVariables(轴变量任务间固定周期同步)指令。
- 通过执行MC_PeriodicSyncVariables(轴变量任务间固定周期同步)指令，按周期更新固定周期任务(执行优先度5)对应的主轴的轴变量。
- 程序中使用的主要变量如下所示。

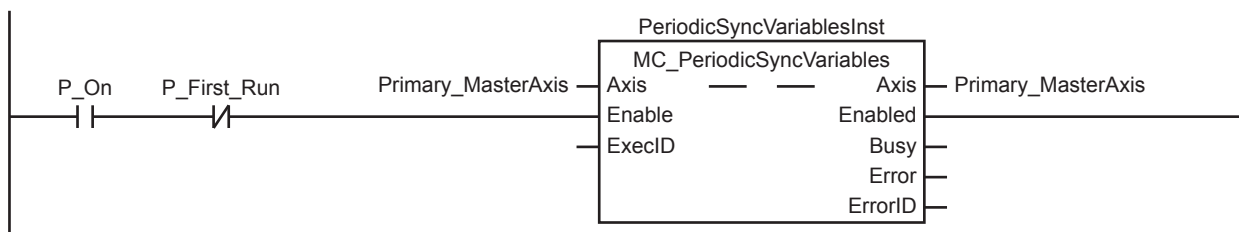
名称	数据类型	注释
Primary_MasterAxis	_sAXIS_REF	主轴(轴1)。轴变量设为_MC1_AX[0]。 分配至原始恒定周期任务 ^{*1}

*1. 主轴请务必分配至原始恒定周期任务。

● 梯形图

原始恒定周期任务的梯形图程序如下所示。

通过原始恒定周期任务，执行MC_PeriodicSyncVariables(轴变量固定周期同步)指令。



固定周期任务(执行优先度5)的程序

- 通过MC_PeriodicSyncVariables(轴变量任务间固定周期同步)指令，在固定周期任务(执行优先度5)的程序中，可查看按周期更新的主轴(Primary_MasterAxis)的轴变量。
- 根据主轴信息，对分配至固定周期任务(执行优先度5)的虚拟主轴(Secondary_VirtualMasterAxis)执行MC_SyncMoveAbsolute(周期同步绝对定位)指令跟踪主轴。
- 相对于从轴(Secondary_SlaveAxis)，将虚拟主轴(Secondary_VirtualMasterAxis)指定为主轴。
- 通过执行MC_GearIn(齿轮动作开始)指令等的同步控制指令，可在分配至不同任务周期的轴间实现同步控制功能。
- 程序中使用的主要变量如下所示。

名称	数据类型	注释
Secondary_VirtualMasterAxis	_sAXIS_REF	虚拟主轴。分配至固定周期任务(执行优先度5)
Secondary_SlaveAxis	_sAXIS_REF	从轴(轴3)。分配至固定周期任务(执行优先度5)
Primary_MasterAxis ^{*1}	_sAXIS_REF	主轴。轴变量设为_MC2_AX[4]。 通过MC_PeriodicSyncVariables(轴变量任务间固定周期同步)指令，按周期进行同步。
PhaseShift	LREAL	相位补偿量
VirtualMasterAxisTargetPosition	LREAL	虚拟主轴各任务周期的目标位置
SyncExecute	BOOL	从轴的同步控制开始信号

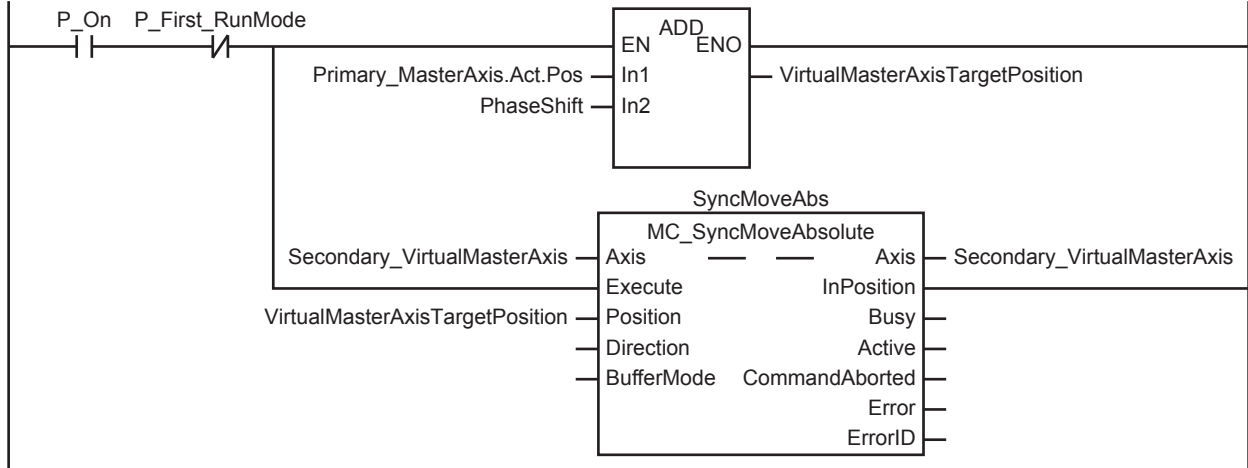
*1. 在该示例中，对与原始恒定周期任务的主轴同名的内部变量“Primary_MasterAxis”进行定义并使用。

● 梯形图

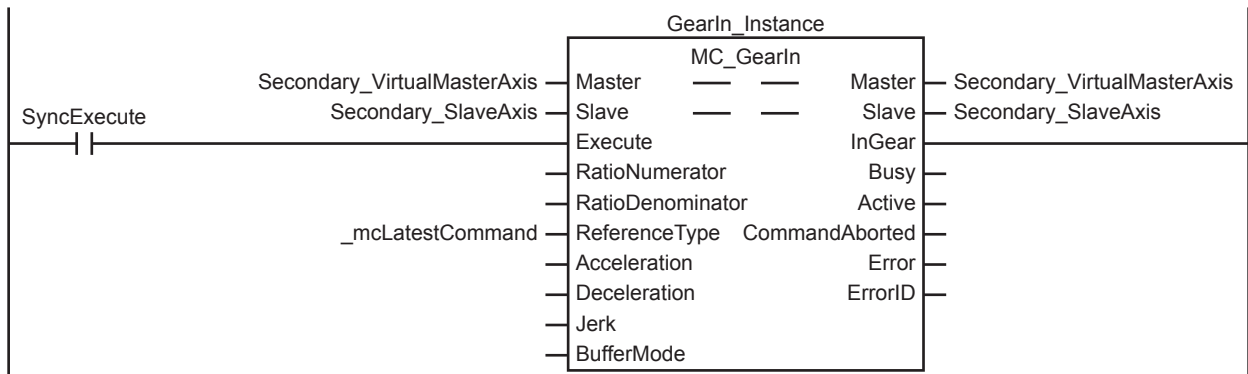
固定周期任务(执行优先级5)的梯形图程序如下所示。

将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(周期同步绝对定位)指令，使Secondary_VirtualMasterAxis始终跟踪Primary_MasterAxis。



将Secondary_VirtualMasterAxis作为虚拟主轴，对Secondary_SlaveAxis执行同步控制指令。



MC_SyncOffsetPosition

执行同步控制中的从轴的位置补偿。

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 Ver.1.10 以上的 CPU 单元和 Ver.1.12 以上的 Sysmac Studio 组合时使用。

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	在上升沿开始指令。
OffsetPosition	位置偏置	LREAL	负数、正数、“0”	0	指定在指令当前位置叠加的位置偏置。单位为[指令单位]。 ^{*1}
BufferMode	缓存模式选择	_eMC_BUFFER_MODE	0: _mcAborting	0 ^{*2}	指定多重启动运动指令时的动作。 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. 关于指令单位，请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*2. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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”	输出在指令当前位置叠加的位置偏置。 Active为TRUE时更新值。 CommandAborted或Error为TRUE时，停止更新并保持 值。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

*1. 请参阅 □□ “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Busy	Execute的上升沿	<ul style="list-style-type: none"> • Error变为TRUE时 • CommandAborted变为TRUE时
Active	开始指令后	<ul style="list-style-type: none"> • Error变为TRUE时 • CommandAborted变为TRUE时
CommandAborted	<ul style="list-style-type: none"> • 执行中的同步控制指令停止时 • 利用其它指令多重启动运动指令,中止本指令时 • 其他指令发生异常、使本指令中止时 • 轴异常发生过程中启动本指令时 • 执行MC_Stop指令中，启动本指令时 	<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含有异常因素时	异常已解除时

输入输出变量

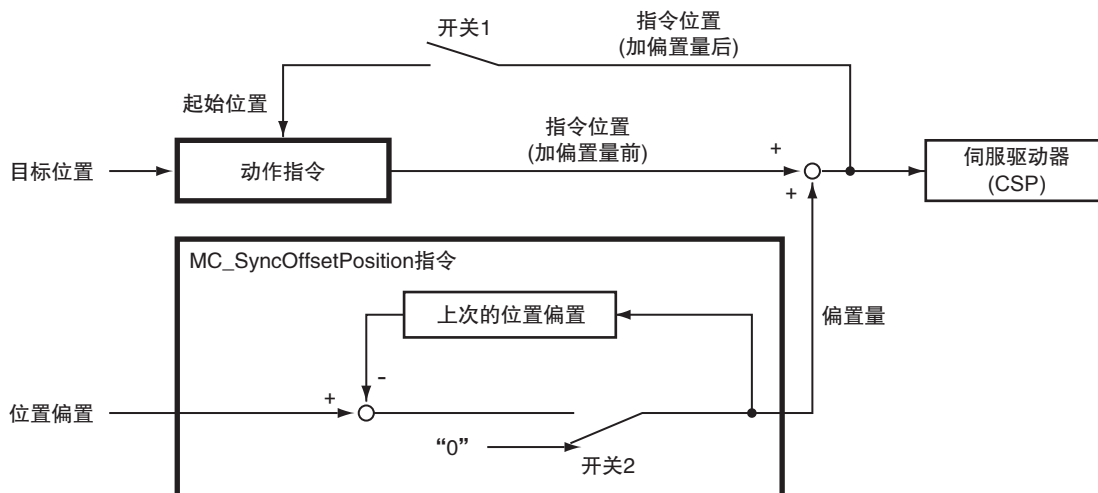
输入输出变量	名称	数据类型	有效范围	内容
Axis	轴	_sAXIS_REF	-	指定轴。*1 *2

*1. 请使用在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。

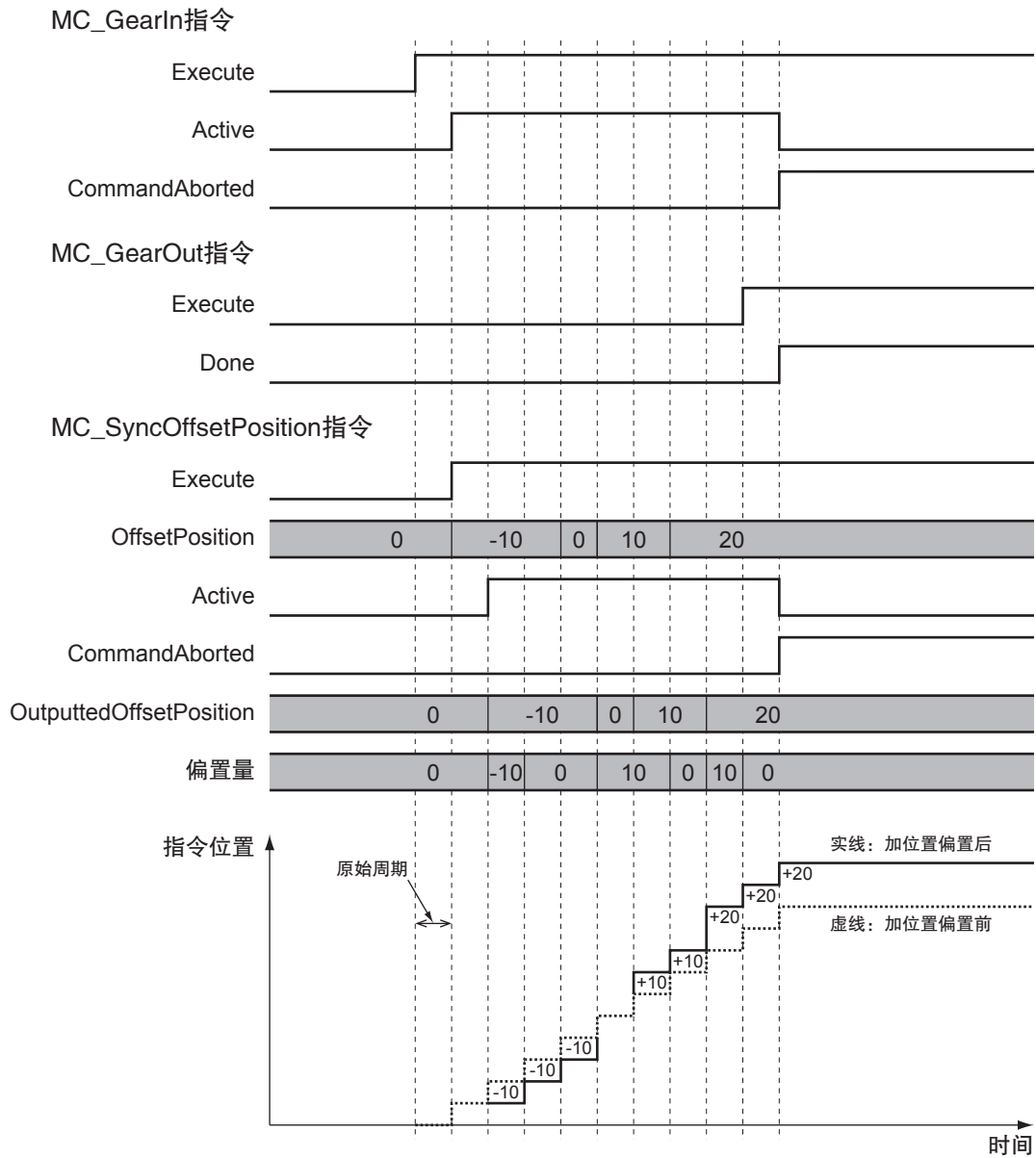
功能说明

- 本指令对于同步控制中的从轴，在指令当前位置上加上根据输入变量的OffsetPosition(位置偏置)值计算出的偏置量，输出至伺服驱动器。
- 输出变量的Active(控制中)为TRUE期间，可变更输入变量的OffsetPosition(位置偏置)的值。
- 本指令的Active(控制中)为TRUE期间，OffsetPosition(位置偏置)将本指令启动时作为起点处理。详情请参阅“重复执行本指令的启动和停止时(P.3-427)”。



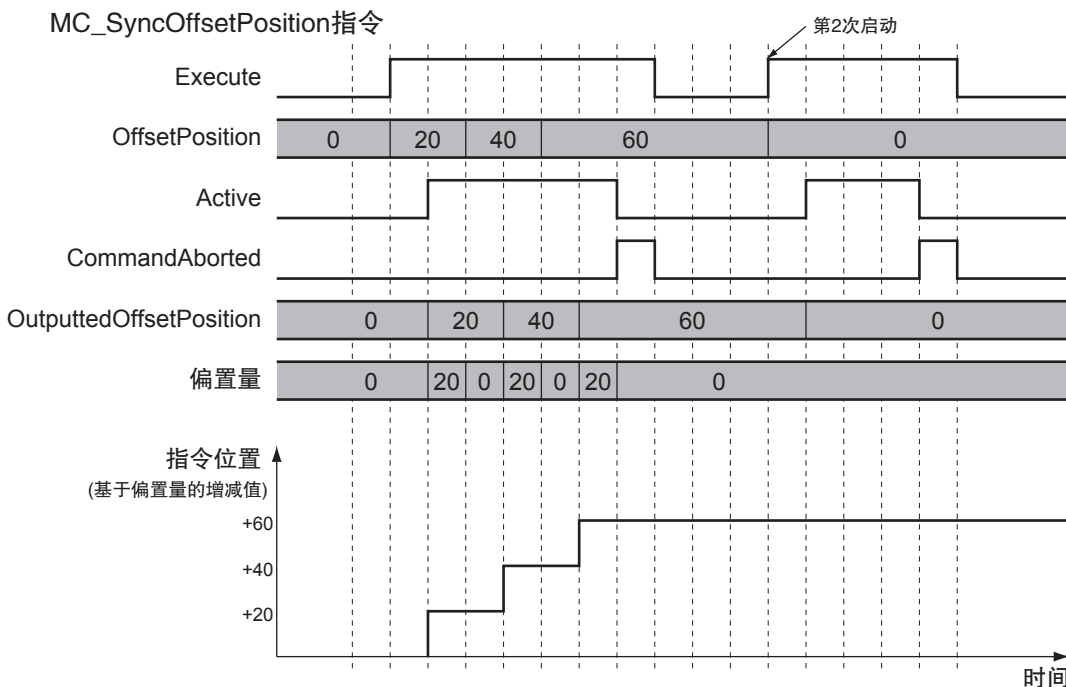
- 开关1仅在动作指令的Execute(启动)的上升沿时ON一次。
 - 开关2在MC_SyncOffsetPosition指令的Active(控制中)为TRUE时ON。
 - 开关2为OFF时，指令位置中叠加的偏置量为“0”。
- 本指令仅可对同步控制中的从轴执行。
 - 启动本指令后，对象同步控制指令的Done(完成)、CommandAborted(执行中断)或Error(错误)为TRUE时，本指令的CommandAborted(执行中断)变为TRUE。
 - 本指令的CommandAborted(执行中断)或Error(错误)变为TRUE时，指令位置保持之前叠加的偏置量。

- 在原始恒定周期任务中的MC_GearIn(齿轮动作开始)指令动作中执行本指令后, MC_GearIn(齿轮动作开始)指令的CommandAborted(执行中断)变为TRUE后, 时序图如下所示。



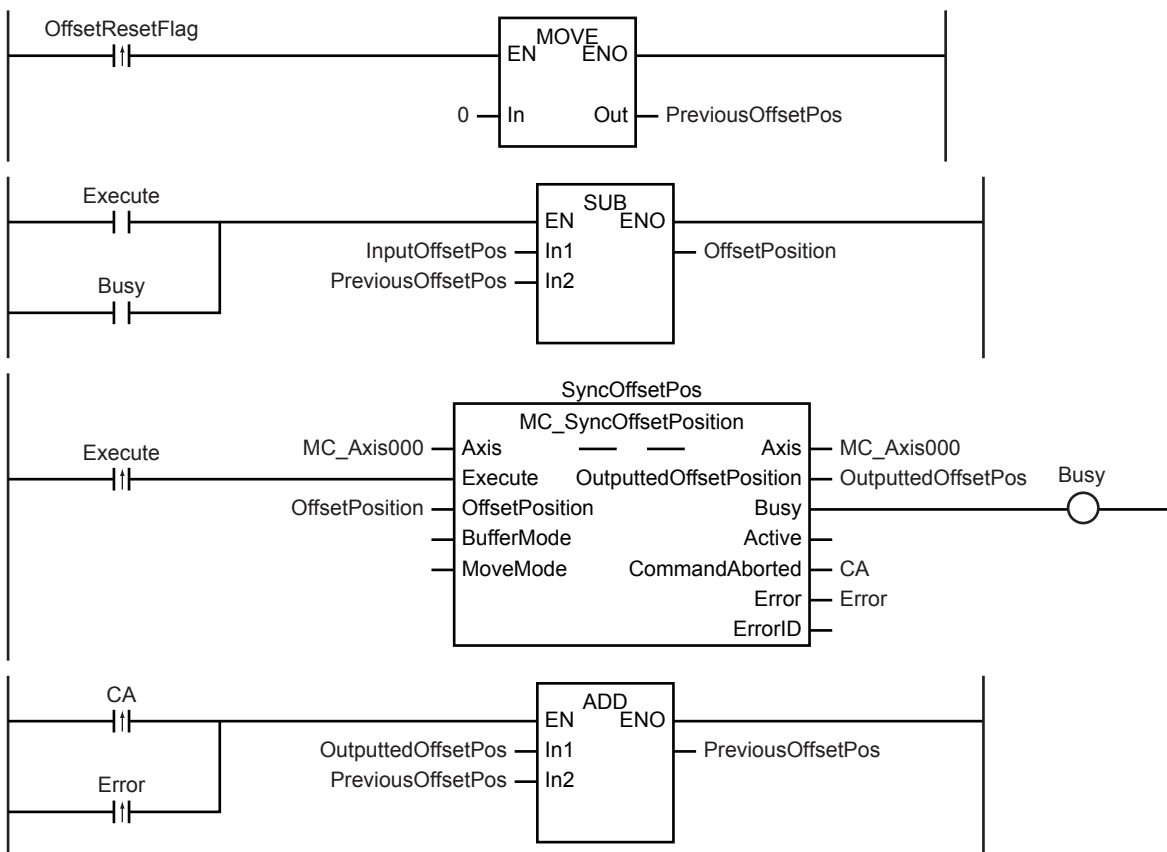
● 重复执行本指令的启动和停止时

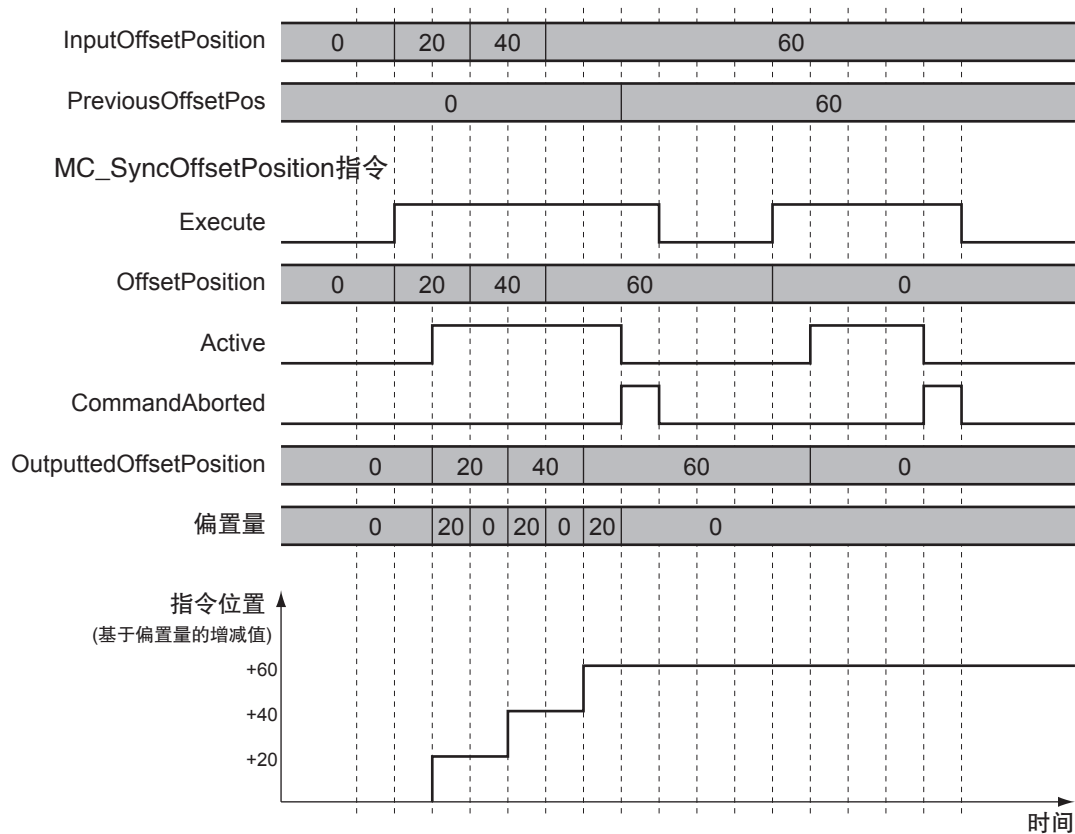
本指令执行中断后再次执行时，输入变量的OffsetPosition(位置偏置)将以“0”为基点。第2次启动时若将OffsetPosition(位置偏置)指定为“0”，则指令位置中叠加的偏置量为“0”。



本指令执行中断后再次执行时，欲以与前次指令相同的基点发出指令时，利用OutputtedOffsetPosition(位置偏置输出值)。

如以下示例程序所示，参照OutputtedOffsetPosition(位置偏置输出值)中执行中断时的位置偏置，从本指令中指定的输入参数变量中减去该值。





● 可执行的轴的状态

仅Axis(轴)指定的轴状态Status.Synchronized(同步动作中)为TRUE时可执行本指令。其他状态下执行会发生异常。

● 指令位置的处理

轴的指令位置以叠加位置偏置后的值进行管理。因此，根据位置偏置中设定的值，可能会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
 - 到位检查时间
- 轴参数的限制设定
 - 软件限制功能
 - 位置偏差超过值
 - 位置偏差警告值
- 指令位置溢出
- 指令位置下溢

时序图

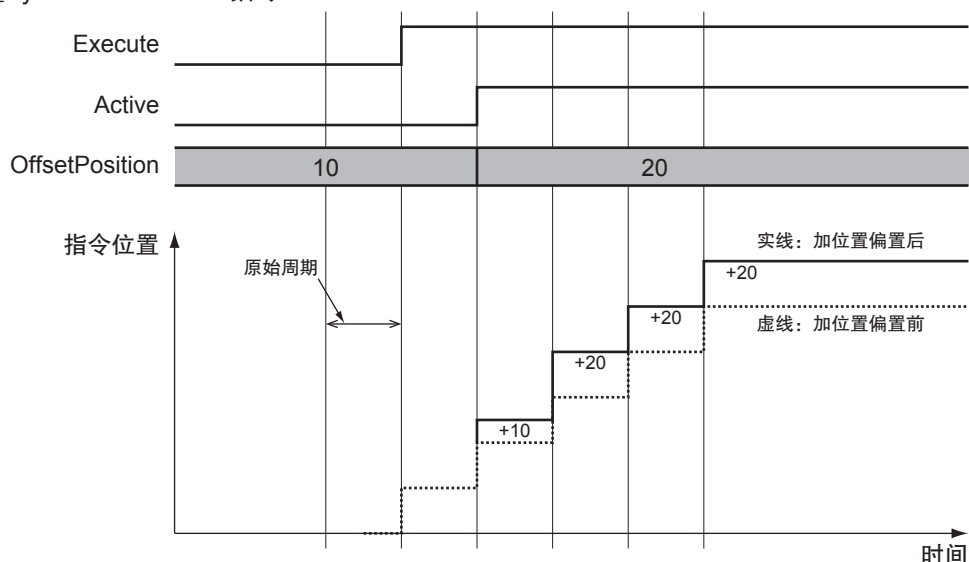
执行本指令时的位置偏置的反映时间如下所示。

● 原始恒定周期任务或固定周期任务(执行优先级5)中记述有本指令时

通过输入指定的位置偏置在下一个任务周期时输出到伺服驱动器。

原始恒定周期任务中记载有本指令的示例如下所示。固定周期任务(执行优先级5)时也一样。

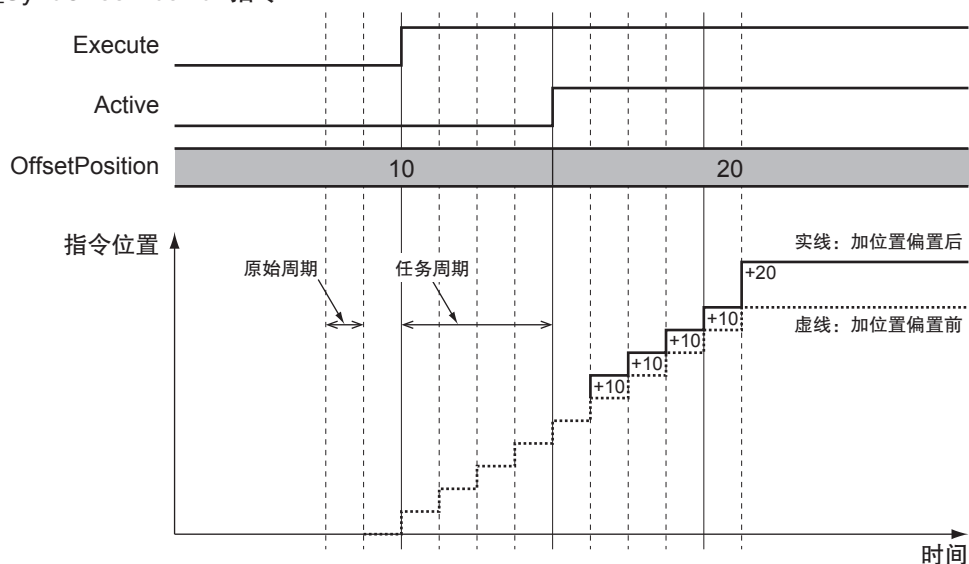
MC_SyncOffsetPosition指令



● 固定周期任务(执行优先级16)中记载有本指令时

通过输入指定的位置偏置在下一个固定周期任务(执行优先级16)+原始周期1周期后的时间输出到伺服驱动器。

MC_SyncOffsetPosition指令



重启运动指令

无法重启本指令。

重启后, 将发生“无法重启运动指令(错误代码: 543B Hex)”错误。

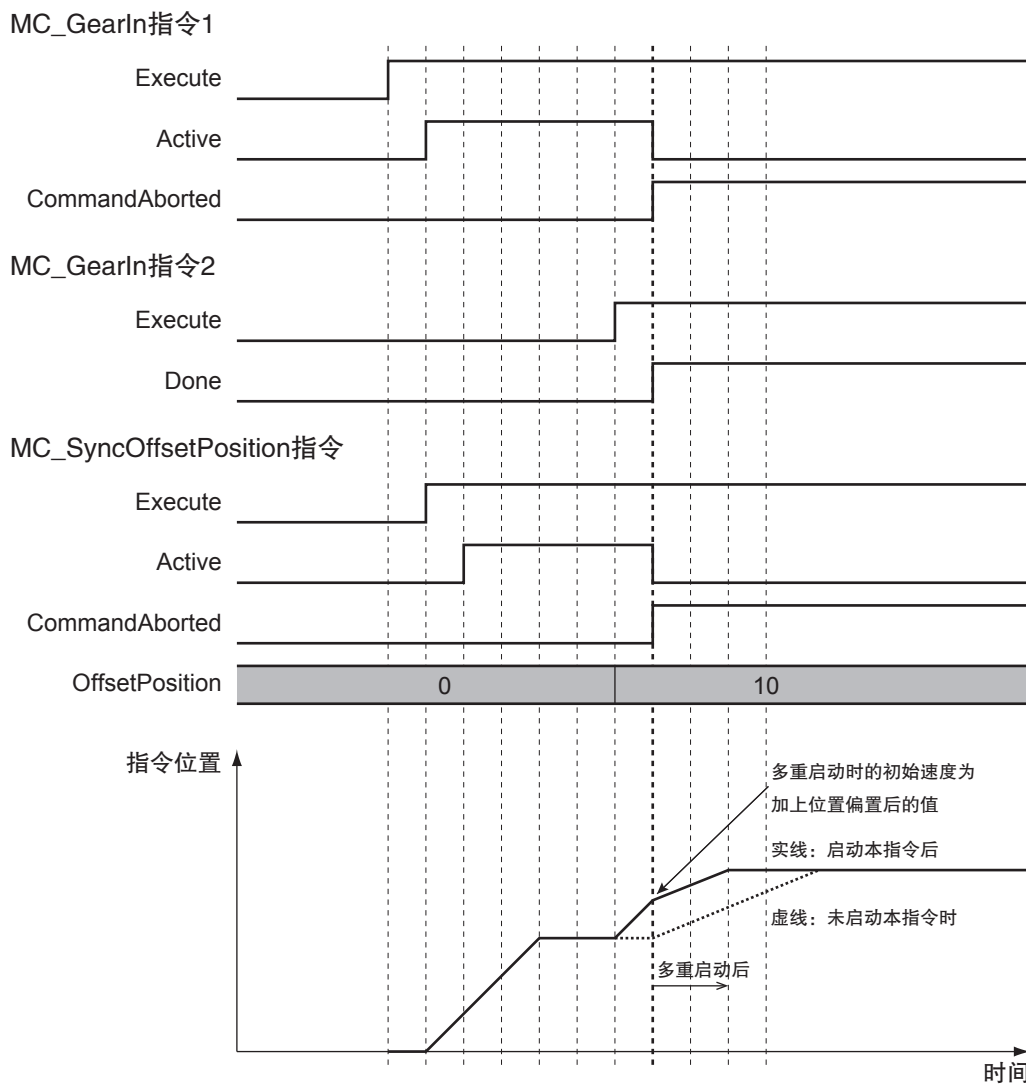
多重启动运动指令

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 对象同步控制指令对应的多重启动

对本指令的对象同步控制指令进行多重启动时，本指令的CommandAborted(执行中断)变为TRUE。但是，合并多个MC_CamIn(凸轮动作开始)指令进行多重启动时，本指令的CommandAborted(执行中断)将保持FALSE状态继续处理。此时的初始速度为叠加了OffsetPosition(位置偏置)的值。

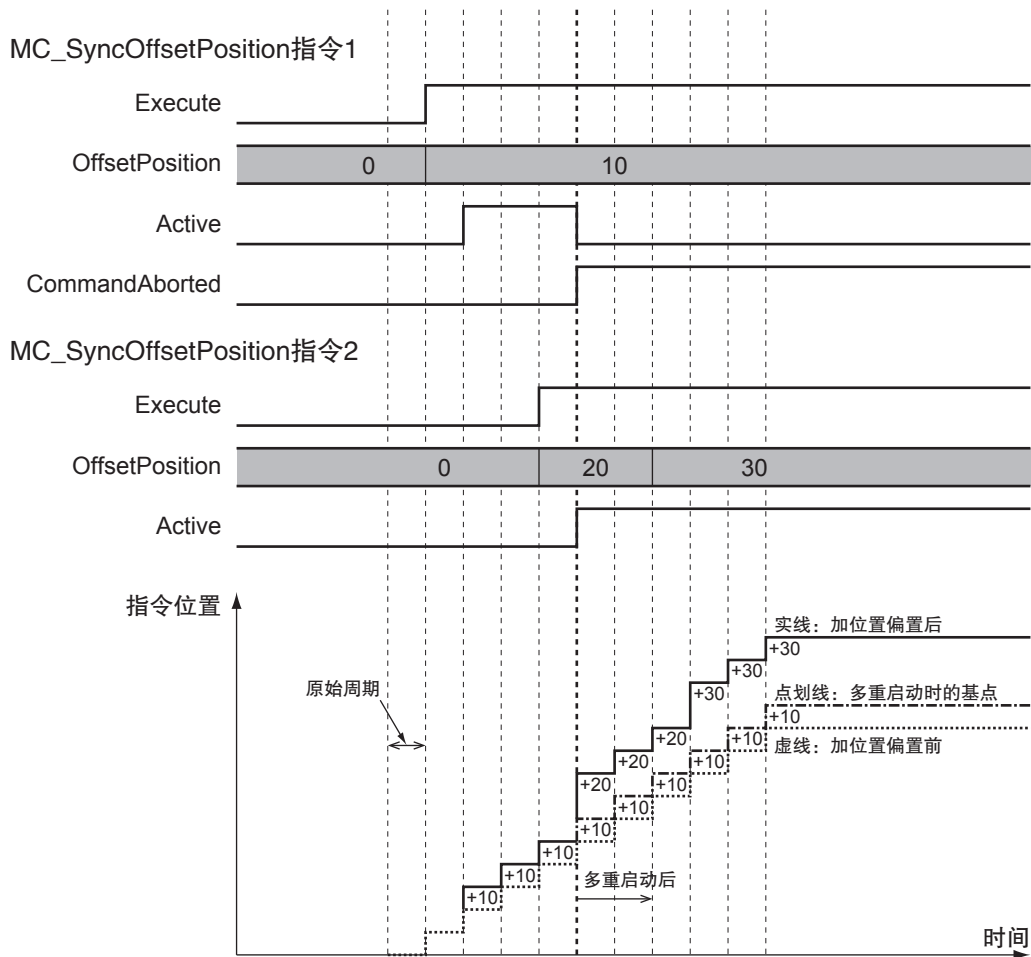
对象同步控制指令为 MC_GearIn(齿轮动作开始) 指令，在执行本指令过程中，多重启动其它实例的MC_GearIn(齿轮动作开始)指令时，时序图如下所示。



● 本指令对应的多重启动

使用多个本指令进行多重启动时，执行中实例的CommandAborted(执行中断)变为TRUE，后执行的实例被执行。后执行的实例基点位置变为，叠加先执行的实例被中断1个周期前输出的位置偏置后的位置。没有其他因本指令的多重启动而执行中断的指令。

进行多个本指令的多重启动时的时序图如下所示。

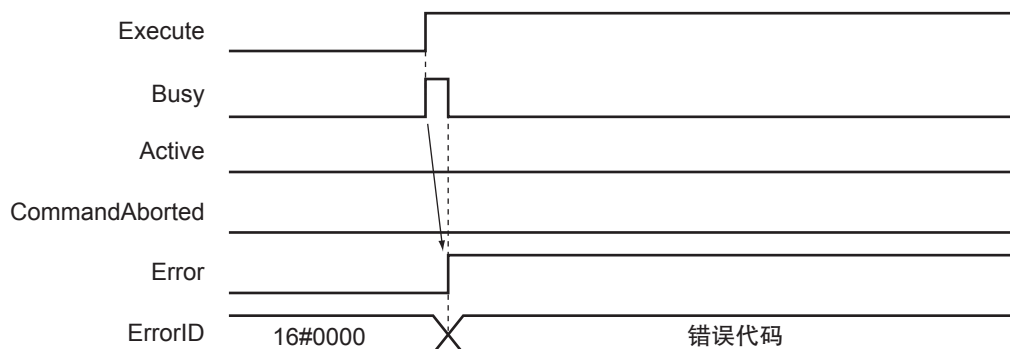


异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 [□□](#) “A-1 错误代码一览(P.A-2)”。

4

轴组指令

下面对MC功能模块执行多轴协调控制的指令进行说明。

MC_GroupEnable	4-2
MC_GroupDisable	4-6
MC_MoveLinear	4-11
MC_MoveLinearAbsolute	4-38
MC_MoveLinearRelative	4-41
MC_MoveCircular2D	4-44
MC_GroupStop	4-68
MC_GroupImmediateStop	4-76
MC_GroupSetOverride	4-80
MC_GroupReadPosition	4-84
MC_ChangeAxesInGroup	4-88
MC_GroupSyncMoveAbsolute	4-92
MC_GroupReset	4-98

MC_GroupEnable

使轴组有效。

指令	名称	FB/ FUN	图形表现	ST表现
MC_GroupEnable	启用轴组	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	在上升沿开始指令。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

* 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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	中止本指令时	<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含有异常因素时	异常已解除时

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. 请使用在Sysmac Studio的轴组基本设定画面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

下面，在说明本指令的功能之前，先对轴组的状态进行说明。

- 轴组的状态有GroupEnable(启用轴组)和GroupDisable(不启用轴组)两种。

状态	说明
GroupEnable (启用轴组)	多轴协调控制有效的状态。 可以执行除MC_ChangeAxesInGroup(写入轴组构成轴)指令以外的所有多轴协调指令。
GroupDisable (不启用轴组)	多轴协调控制无效的状态。 多轴协调控制指令只能执行以下指令。 <ul style="list-style-type: none"> MC_GroupEnable(启用轴组)指令 MC_GroupDisable(不启用轴组)指令 MC_GroupReset(轴组错误复位)指令 MC_GroupSetOverride(轴组超调值设定)指令 MC_GroupReadPosition(轴组位置获取)指令 MC_ChangeAxesInGroup(写入轴组构成轴)指令

- 要进行多轴协调控制，轴组必须为GroupEnable(启用轴组)状态。

通过显示运动控制系统变量的轴组变量，可以确认轴组的启用、不启用状态。

下面以_MC_GRP[*]为例对轴组变量进行说明。

_MC1_GRP[*]、_MC2_GRP[*]也是如此。

变量名称	名称	说明
_MC_GRP*.Status.Ready	启用轴组	轴组停止、且已通过轴组指令完成启动准备时变为TRUE。
_MC_GRP*.Status.Disabled	不启用轴组	因轴组无效而停止时变为TRUE。

(注) “_MC_GRP*”的“*”中，NX701 CPU单元填入数字“0 ~ 63”、NX1P2 CPU单元填入数字“0 ~ 7”、NJ系列CPU单元及NY系列控制器填入数字“0 ~ 31”。

基本功能

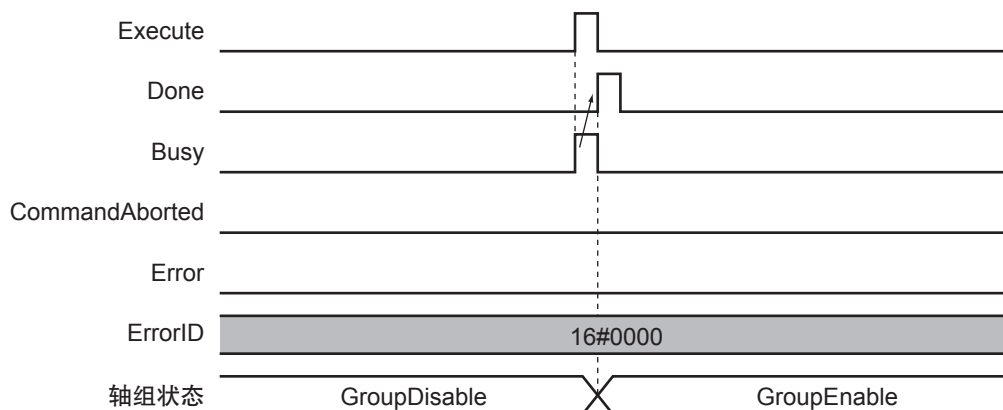
- MC_GroupEnable(启用轴组)指令将通过AxesGroup(轴组)指定的轴组变为GroupEnable(启用轴组)状态。
- 变为GroupEnable(启用轴组)状态后，轴组便可以执行所有多轴协调指令。
- 可指定到轴组中的轴的种类只能是“伺服轴”和“虚拟伺服轴”，若指定了其他轴种类，将出现异常。
- 属于轴组的所有轴必须处于停止状态。
轴变量Status.Disabled(轴无效)或Status.Standstill(停止中)为TRUE时，即表明轴处于停止状态。
- 如果存在已经属于其他轴组、且该轴组已启用的轴，则不能执行MC_GroupEnable(启用轴组)指令，会发生异常。
- 轴组启用后，属于轴组的轴的状态为“多轴协调动作中”。
轴变量Status.Coordinated(多轴协调动作中)变为TRUE。
- 使轴组无效的条件有执行MC_GroupDisable(不启用轴组)指令、切换到程序模式使运行停止、以及开始MC试运行。



使用注意事项

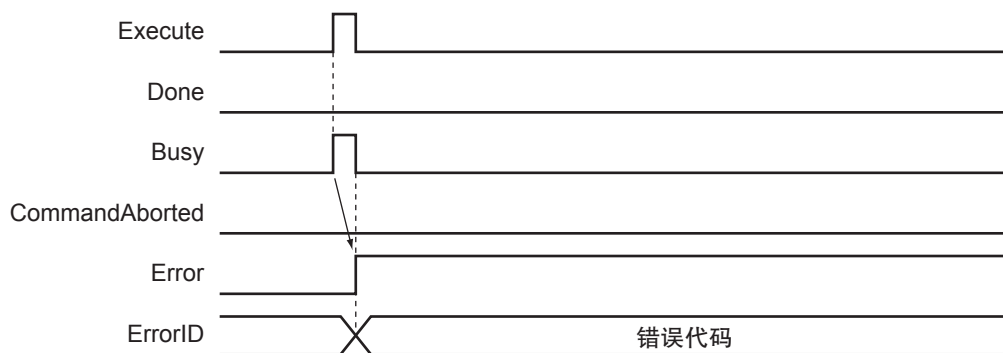
- 要使用轴组，必须先通过Sysmac Studio创建轴组，然后将设定下载到CPU单元。在程序中不能变更轴组中的轴。
Ver.1.01以上的CPU单元和Ver.1.02以上的Sysmac Studio组合时，可使用MC_ChangeAxesInGroup(轴组构成轴写入)指令暂时变更。
- “下载”使用Sysmac Studio的“同步”功能。

时序图



异常

执行MC_GroupEnable(启用轴组)指令时若发生异常，Error(错误)变为TRUE。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。



参考

轴组即使发生异常，轴组中包含的各轴并不会有异常。

● 错误代码

关于指令发生的异常，请参阅 [“A-1 错误代码一览\(P.A-2\)”](#)。

MC_GroupDisable

使轴组无效。

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	在上升沿开始指令。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

* 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	轴组状态变更为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含有异常因素时	异常已解除时

输入输出变量

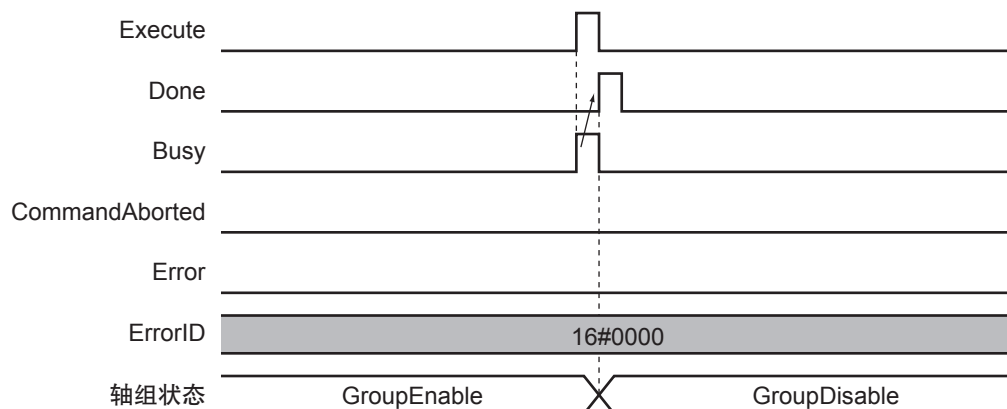
输入输出变量	名称	数据类型	有效范围	内容
AxesGroup	轴组	_sGROUP_REF	-	指定轴组。*1

*1. 请使用在Sysmac Studio的轴组基本设定画面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

- 本指令使轴组状态变为无效。
关于轴组状态的说明，请参阅 □ “功能说明(P.4-3)”。
- MC_GroupDisable(不启用轴组)指令将通过AxesGroup(轴组)指定的轴组变为GroupDisable(不启用轴组)状态。
变为GroupDisable(不启用轴组)状态后，轴组便不能接受多轴协调指令。
- 轴组状态变为GroupDisable(不启用轴组)时，指定的AxesGroup(轴组)的缓存指令会被清除。
- 轴组在运行停止时也会变为无效。
轴组无效时，属于轴组的轴的状态从Status.Coordinated(多轴协调动作中)为TRUE的状态变为各轴的状态。
要了解各轴的状态，请确认轴变量中的Status(轴状态)。

时序图

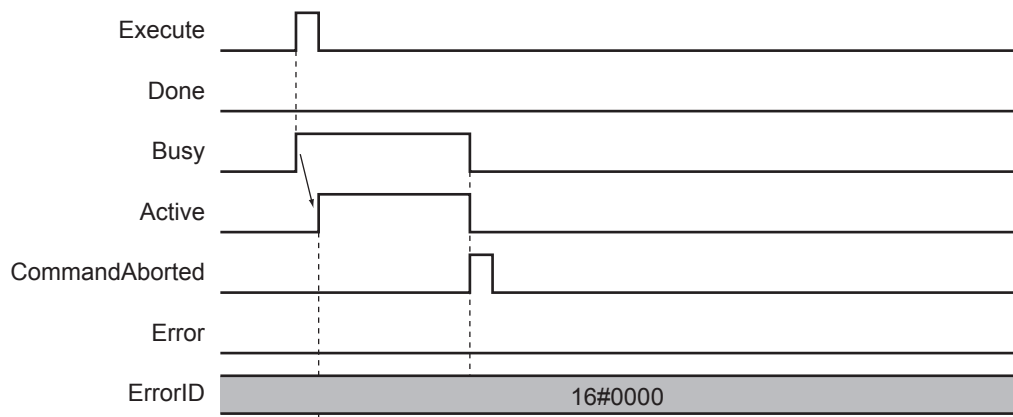


对多轴协调控制中的AxesGroup(轴组)执行MC_GroupDisable(不启用轴组)指令时，多轴协调控制会变为CommandAborted(执行中断)。

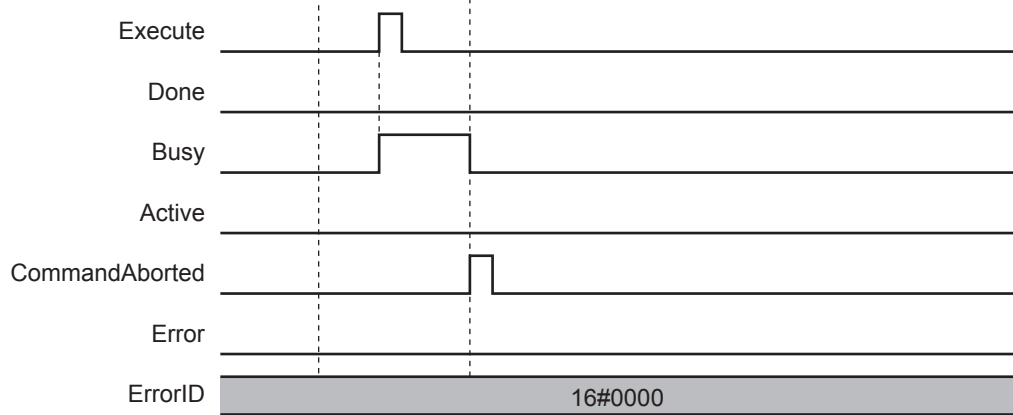
此时，缓存的多轴协调控制指令也会变为CommandAborted(执行中断)。

动作中的轴以各轴的最大减速速度减速并停止。

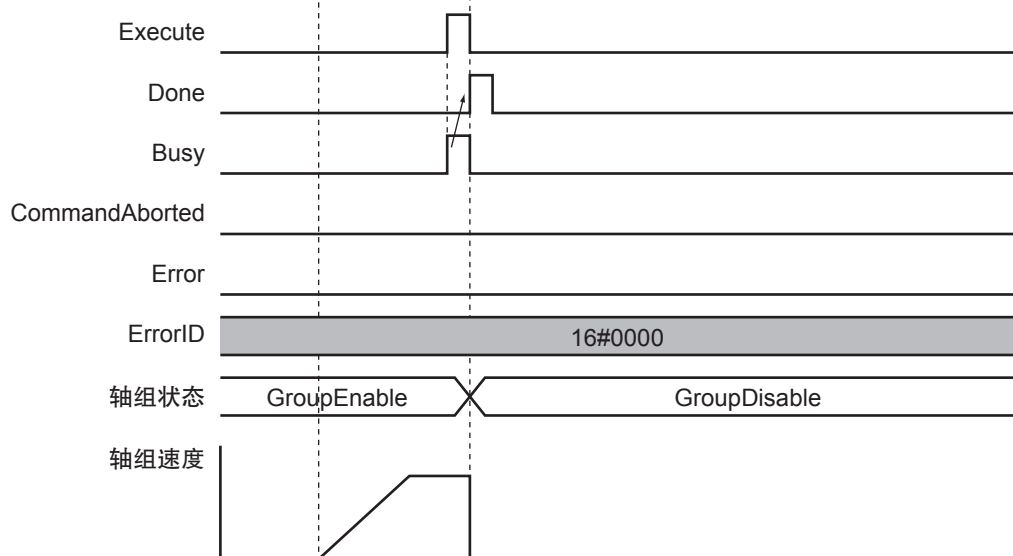
多轴协调控制1(Ex. MC_MoveLinearAbsolute)



多轴协调控制2(Ex. MC_MoveLinearAbsolute.Buffered)

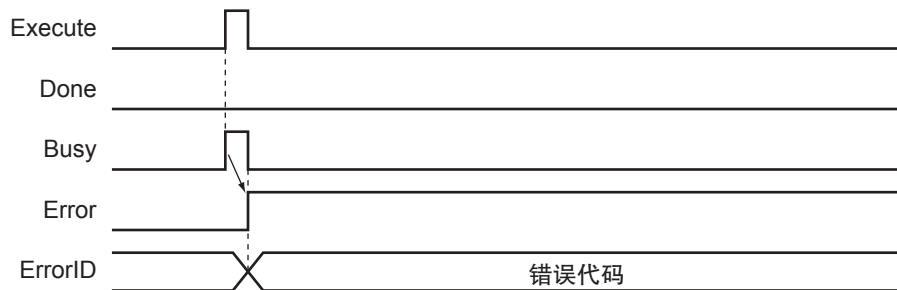


MC_GroupDisable



异常

执行MC_GroupDisable(不启用轴组)指令时若发生异常，Error(错误)变为TRUE。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。



● 错误代码

关于指令发生的异常，请参阅 □ “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>

MC_MoveLinear

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	在上升沿开始指令。
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)
BufferMode	缓存模式选择	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 ^{*3}	指定多重启动运动指令时的 动作。 0: 中断 1: 缓存 2: 以低速合并 3: 以前一个速度合并 4: 以后一个速度合并 5: 以高速合并

输入变量	名称	数据类型	有效范围	初始值	内容
TransitionMode	过渡模式 (切换模式)	_eMC_ TRANSITION_ MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 *3	指定动作的路径。 0: 过渡无效 10: 附加角
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. 关于指令单位, 请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的 “单位转换设定”。

*2. 请务必设定目标速度。不作设定就执行动作, 将发生异常。

*3. 有效范围为枚举体的变量, 其实际初始值不是数值, 而是枚举元素。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时, 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为正常。

* □ 请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	定位完成后	<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"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 执行MC_GroupStop指令中，启动本指令时 	<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含有异常因素时	异常已解除时

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. 请使用在Sysmac Studio的轴组基本设定画面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

- 轴组中的任意一个构成轴原点未确定时，将发生“原点未确定状态下的指令启动异常(异常代码：5466 Hex)”的错误。
- 进行直线插补时，属于该轴组的任意一个逻辑轴的极限输入“ON”时，将无法启动指令。

指令详情

下面对本指令详细说明。

● 直线插补的步骤

直线插补按以下步骤执行。

1

登录进行插补的轴组

- 确定进行插补的轴组。
轴组即_MC_GRP[*]或_MC1_GRP[*]、_MC2_GRP[*]。
- 通过该轴组变量的轴构成指定轴的构成。可指定的轴数为2 ~ 4轴。
- 通过该轴组变量的轴选择指定进行插补的轴的组合。
- 轴不使用轴号，而是使用逻辑轴(轴A0 ~ 轴A3)。
- 通过轴选择以前移方式向逻辑轴A0 ~ A3分别指定轴号。
- 根据型号、系列不同，轴号如下所示。

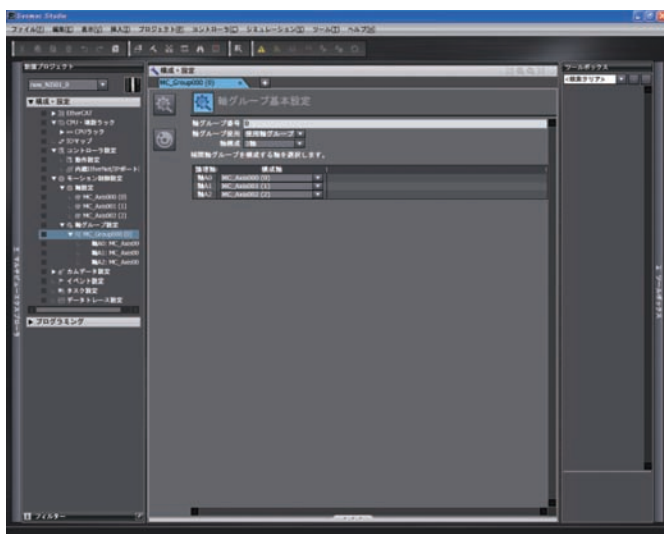
型号或系列名称	轴号
NX701 CPU单元	轴0 ~ 轴255
NX1P2 CPU单元	轴0 ~ 轴11
NJ系列CPU单元	轴0 ~ 轴63
NY系列控制器	轴0 ~ 轴63



参考

轴号的详情，请参阅 “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

- 通过Sysmac Studio的轴组设定选择要使用的“轴构成”，将轴号分配给逻辑轴。以下是向拥有3个轴的轴组“MC_Group000”登录MC_Axis000、MC_Axis001、MC_Axis002的示例。



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“不符合轴类型(错误代码：543D Hex)”的错误，结束动作。
请务必选择伺服轴或虚拟伺服轴。

2 启用轴组

- 将轴组的各构成轴置为伺服ON状态，并使其处于原点确定状态。
- 执行MC_GroupEnable(启用轴组)指令，使登录的轴组有效。

这样，就可以使用直线插补指令了。

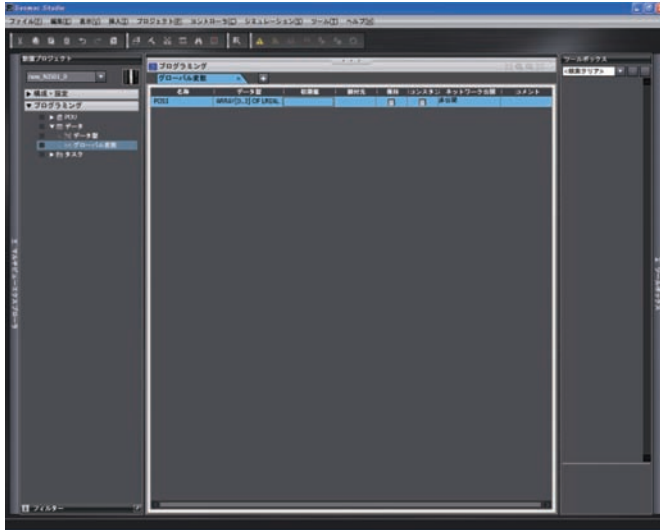
● Position(目标位置)

- 向Position(目标位置)指定通过轴组变量的轴选择指定的轴的目标位置。
- Position(A0, A1, A2, A3)中代入的变量需要在Sysmac Studio上创建1×4的数组变量。变量名称可任意设定。

向该数组中代入各轴的目标位置。

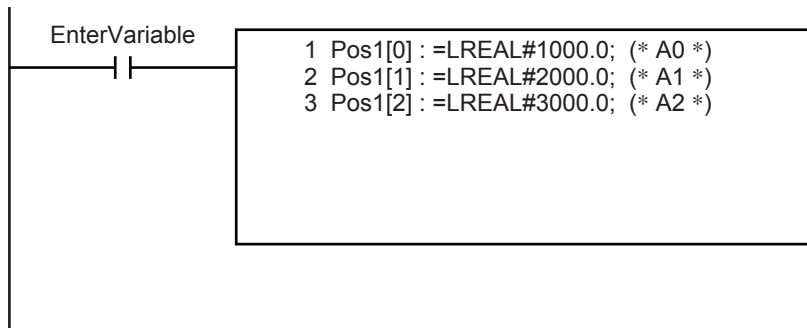
所用轴组中的轴数少于4个轴时，也请务必宣布为1×4的数组。

以下是通过Sysmac Studio宣布数组变量“Pos1”时的示例。



以下是使用内联ST将目标位置代入“Pos1”的示例。

图中表示向A0～A2的各轴中代入目标位置(1000.0, 2000.0, 3000.0)。

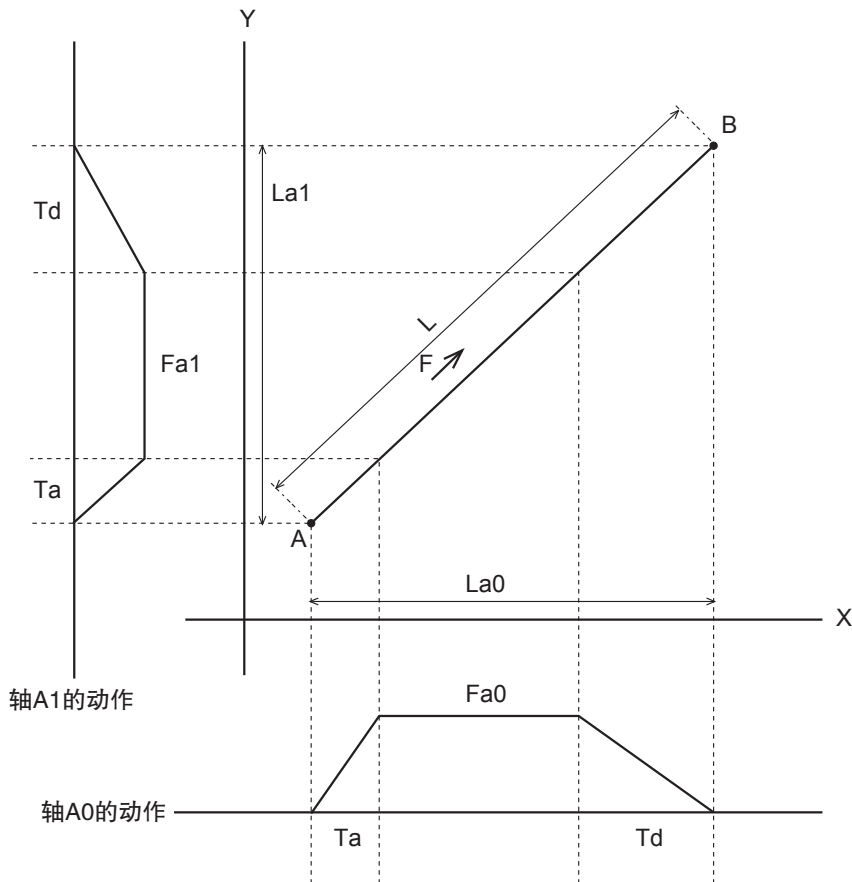


- 将计数模式选择设定为[旋转模式]的轴指定为插补轴时，若指定绝对位置，目标值与Direction(方向选择)选择[无方向指定]时相同。
详情请参阅 “Direction(方向选择)(P.3-51)”。

● 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、跃度通过Velocity、Acceleration、Deceleration、Jerk指定。
- 直线插补时，将直线分解到各轴进行运行。

例如，使用2轴，表示从A点到B点的直线插补请见下图。



4轴直线插补时，各轴的目标速度根据插补速度和各轴的移动距离如下所示。

- F : 指定的插补进给速度
- Fa_0 : F 用于轴A0时的插补进给速度
- Fa_1 : F 用于轴A1时的插补进给速度
- Fa_2 : F 用于轴A2时的插补进给速度
- Fa_3 : F 用于轴A3时的插补进给速度
- Ta : 插补加速时间
- Td : 插补减速时间
- L : 指定轨迹的移动量
- La_0 、 La_1 、 La_2 、 La_3 : 轴A0、轴A1、轴A2、轴A3 的移动量

因此，L, Fa0, Fa1, Fa2, Fa3 可以下式表示。

$$Fa0 = F \times \frac{La0}{L}$$

$$Fa1 = F \times \frac{La1}{L}$$

$$Fa2 = F \times \frac{La2}{L}$$

$$Fa3 = F \times \frac{La3}{L}$$

$$L = \sqrt{La0^2 + La1^2 + La2^2 + La3^2}$$

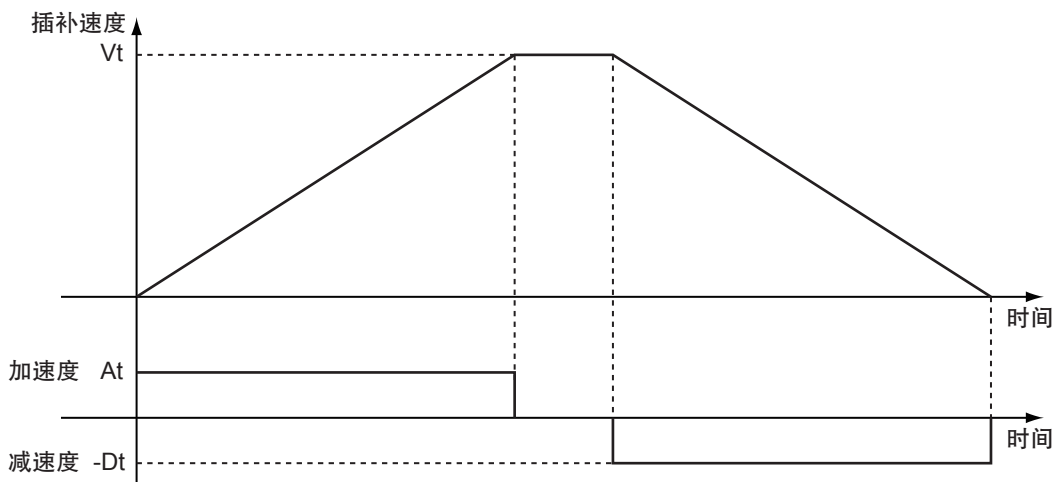
Velocity(目标速度)

- Velocity(目标速度)指定为“0”时，会发生插补速度指定异常。若指定轴组中的轴正在动作，则全轴停止。
- 算出的Velocity(目标速度)Fa0 ~ Fa3中的任意一个超过各轴的最高速度时，将自动调整Velocity(目标速度)，使任意轴以最高速度动作。

Jerk(跃度)

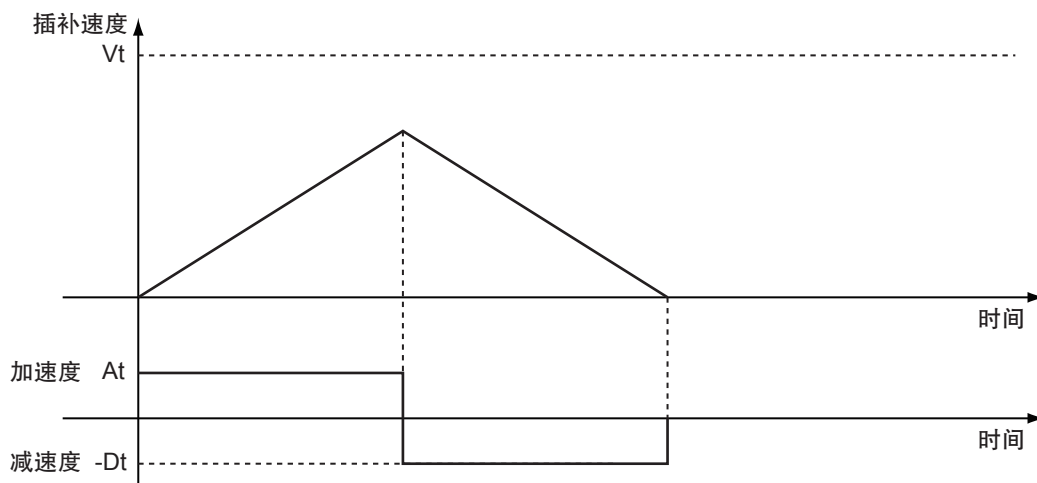
Jerk(跃度)指定为“0”时和指定为非“0”时的Acceleration(加速度)、Deceleration(减速度)、Velocity(目标速度)的关系如下所示。

- Jerk(跃度)为“0”时
以加速度At、减速度Dt生成速度指令值。



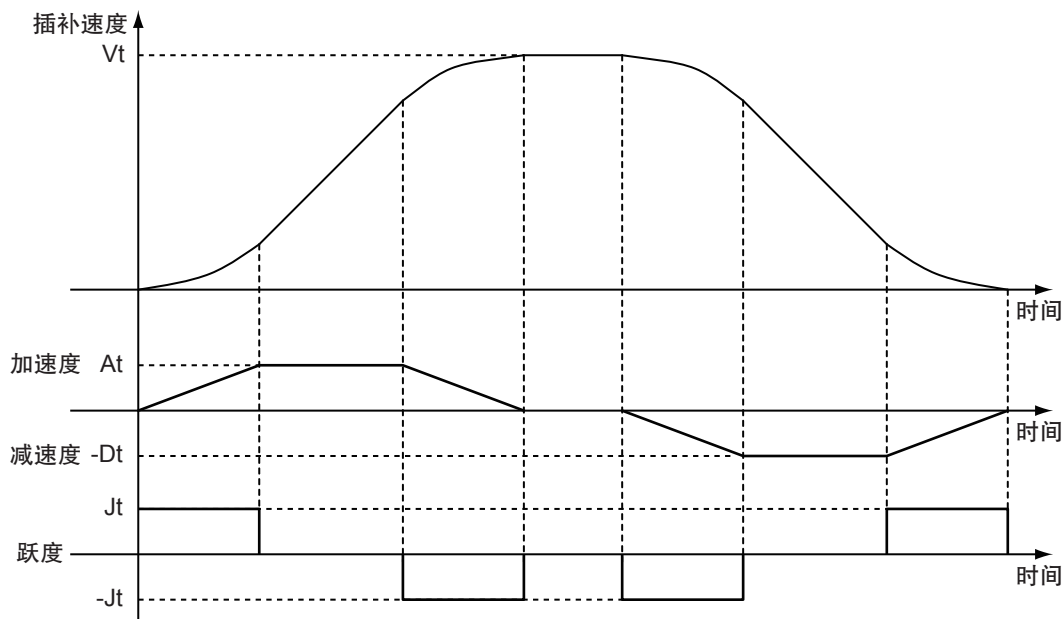
Vt: 插补速度的指定值、At: 加速度的指定值、Dt: 减速度的指定值

- Jerk(跃度)为“0”、移动距离短时
插补速度达不到指定值 V_t (目标速度)。



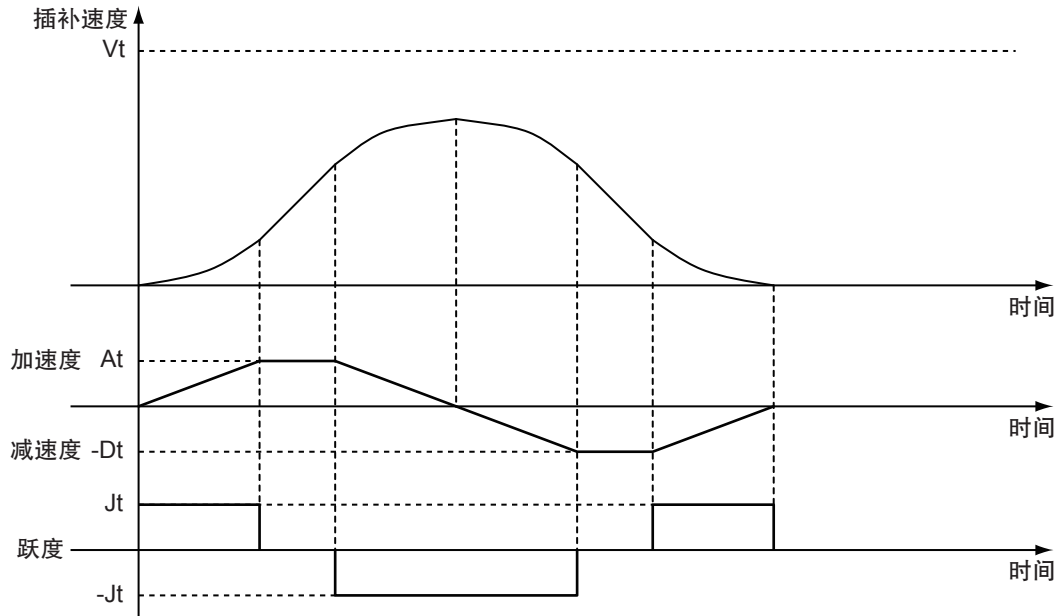
V_t : 插补速度的指定值、 A_t : 加速度的指定值、 D_t : 减速度的指定值

- Jerk(跃度)为非“0”时
以 A_t 为加速度上限、 D_t 为减速度上限生成速度指令值。



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 : 跃度的指定值

- Jerk(跃度)为非“0”、移动距离短时插补速度达不到指定值 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 : 跃度的指定值



参考

- Acceleration(加速度)指定为“0”时，立即以指定的Velocity(目标速度)动作。
- Deceleration(减速度)指定为“0”时，立即停止。缓存模式指定为合并时，不停止动作，而在下一个动作时切换到指定的目标速度。
详情请参阅 □ “BufferMode(缓存模式选择)(P.4-21)”。
- Acceleration(加速度)或Deceleration(减速度)指定为“0”时，跃度设定无效。

● CoordSystem(坐标系)

- 指定进行直线插补的坐标系。
- 只使用由2个以上的轴构成的轴坐标系(ACS)。

● BufferMode(缓存模式选择)

- 指定前一个插补动作和本次插补动作的连接方式。
- 有以下6种。

缓存模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 轴的动作方向因指令切换而反转时，根据轴参数中的“反转时动作”进行反转。
等待	当前正在执行的指令正常完成后，已缓存的本指令自动启动。
合并	以当前正在执行的指令到达目标位置时的速度(中继速度)为启动速度，连续使已缓存的本指令动作。变更当前正在执行的指令的动作，确保以中继速度到达目标位置。中继速度的指定方法分为如下4种。 此外，还有过渡指定(后述)作为合并的选项。
以低速合并	当前正在执行的目标速度与已缓存的目标速度中，以速度较低者为中继速度。
以前一个速度合并	以当前正在执行的目标速度为中继速度。
以后一个速度合并	以已缓存的本指令的目标速度为中继速度。
以高速合并	当前正在执行的目标速度与已缓存的目标速度中，以速度较高者为中继速度。

关于BufferMode(缓存模式选择)，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑用户手册运动控制篇(SBCE-379)”。

● TransitionMode(过渡模式)

- 指定前一个插补动作和本次插补动作的轨迹如何连接。
- 指定TransitionMode(过渡模式)下_mcTMNone(过渡无效)或_mcTMCornerSuperimposed(附加角)的任意一个。
- BufferMode(缓存模式选择)指定为合并时，TransitionMode(过渡模式)即变为有效。
- 若在未指定合并的情况下将过渡模式指定为_mcTMNone(过渡无效)以外的选项，则会发生异常。

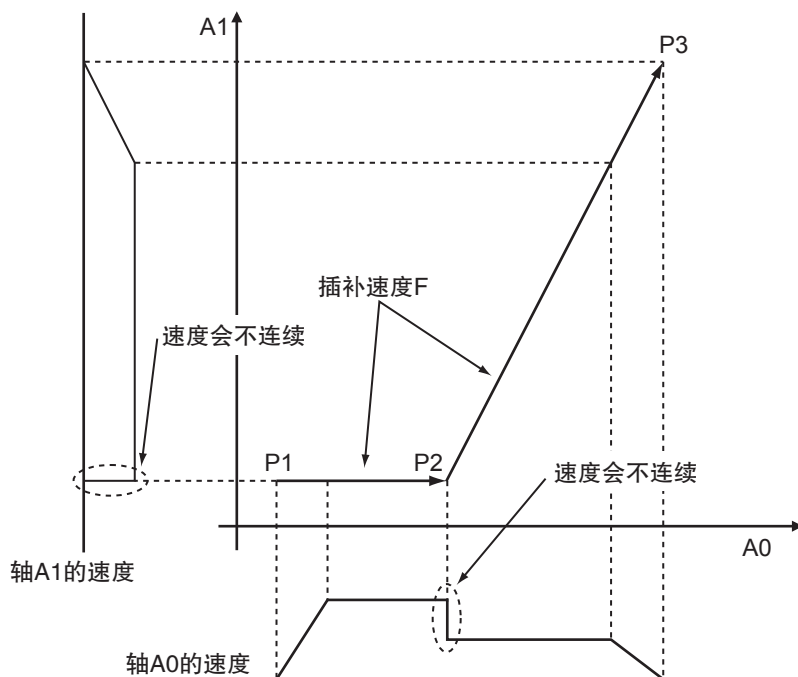
禁用过渡

由于是优先轨迹生成速度指令值，因此各轴的速度指令值在每个动作的连接处会发生速度急剧变化。

动作示例

P1到P2、P2到P3的Velocity(目标速度)、BufferMode(缓存模式选择)、TransitionMode(过渡模式)如下所示。

- P1到P2的动作：Velocity = F、BufferMode = 中断、TransitionMode = `_mcTMNone`(过渡无效)
- P2到P3的动作：Velocity = F、BufferMode = 以后一个速度合并、TransitionMode = `_mcTMNone`(过渡无效)
- 从位置P1开始动作，通过位置P2，向位置P3做直线插补动作。
- 通过位置P2时，保持直线插补的速度F。因此，在位置P2，如图所示，速度不连续。



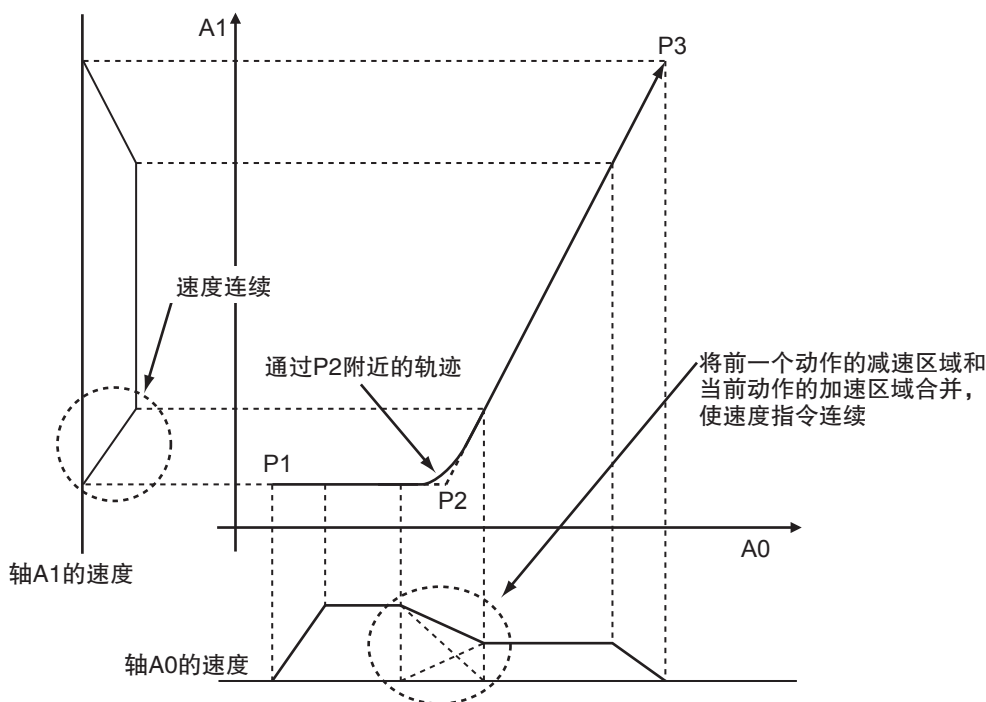
附加角

在希望以各轴的指令速度连续动作时使用。

动作示例

P1到P2、P2到P3的Velocity(目标速度)、BufferMode(缓存模式选择)、TransitionMode(过渡模式)如下所示。

- P1到P2的动作：Velocity = F、BufferMode = 中断、TransitionMode = `_mcTMNone`(过渡无效)
- P2到P3的动作：Velocity = F、BufferMode = 以后一个速度合并、TransitionMode = `_mcTMCornerSuperimposed`(附加角)
- 从位置P1开始动作，通过位置P2的附近，向位置P3做直线插补动作。
- 要以各轴的指令速度连续动作，应将前一个动作的减速区域和当前动作的加速区域合并，生成指令速度。因此，当前动作的加速时间应与前一个动作的减速时间相同。



合并后的轨迹就是通过P2附近的轨迹。

从P2起该轨迹的距离

- 插补速度加快、或前一个指令的减速度减慢时，距离变长
- 插补速度减慢、或前一个指令的减速度加快时，距离变短

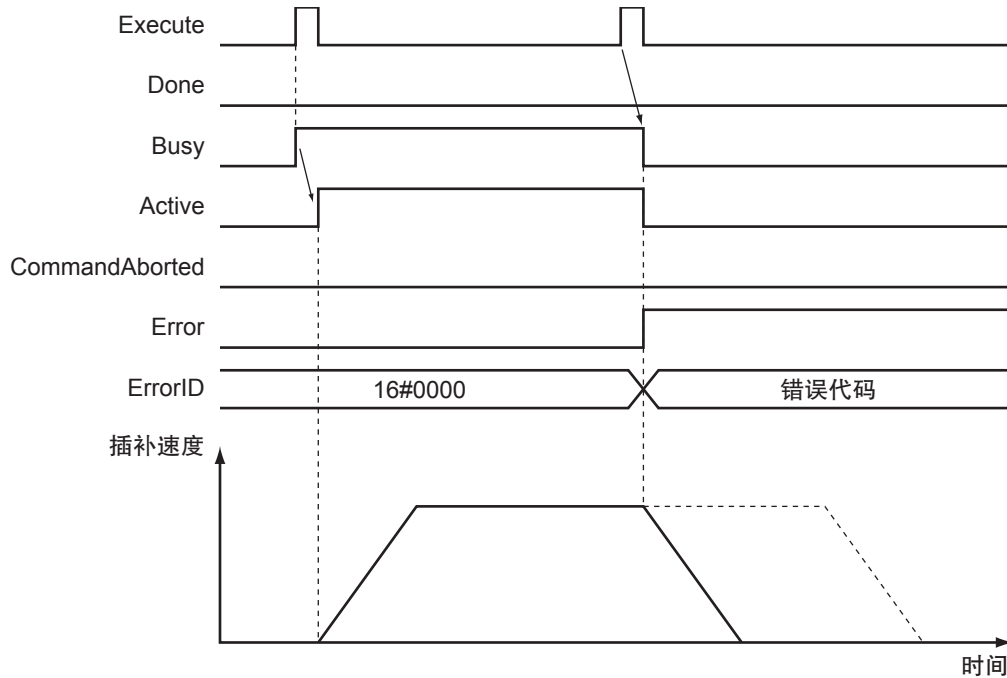


参考

在附加角区域，Jerk(跃度)的指定无效。

重启运动指令

无法重启本指令。若执行了重启，将发生“无法重启运动指令(错误代码: 543B Hex)”的错误，直线插补动作中的所有轴均停止动作。



多重启动运动指令

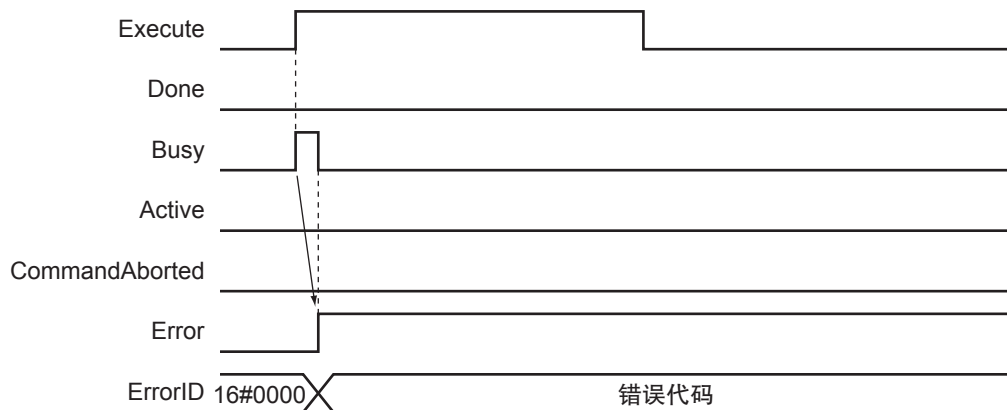
执行本指令时，可使用的指令有限制。

多重启动运动指令的详情，请参阅 [□ “NJ/NX 系列 CPU 单元 用户手册 运动控制篇 \(SBCE-363\)”](#) 或 [□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇\(SBCE-379\)”](#)。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。



● 错误代码

关于指令发生的异常，请参阅 [□ “A-1 错误代码一览\(P.A-2\)”](#)。

示例程序

下面，对基于反复多重启动的直线插补的示例程序进行说明。

参数设定

该示例程序中使用的最少的必要设定如下所示。

- 轴参数的设定

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	线性模式
轴2	线性模式

坐标单位选择

轴	坐标单位
轴1	mm
轴2	mm

- 轴组参数的设定

轴构成

设为[2轴]。

轴选择

设为[轴1]、[轴2]。

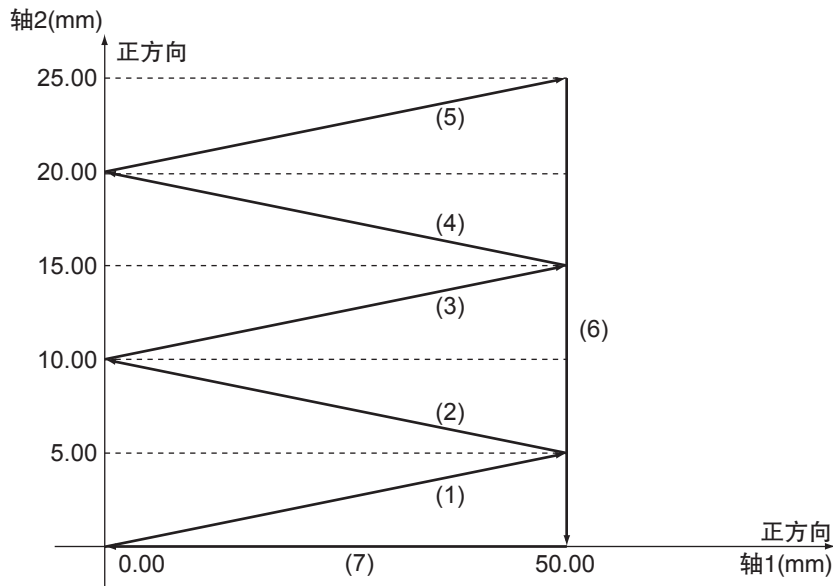
动作示例

以下是自动执行直线插补动作，在返回原点时停止的动作示例。

在直线插补(1)动作过程中多重启动直线插补(2)~(7)。此时的缓存模式选择设定为等待。

在本例中，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。多轴协调动作最多可多重启动7个指令。

● 动作模式



使用直线插补，按照 (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) 的顺序执行定位，然后停止。

梯形图

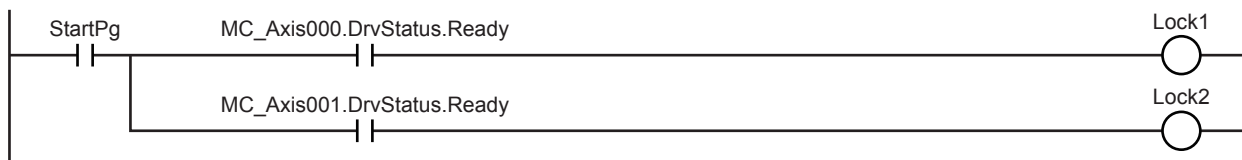
● 主要变量

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无效时变为TRUE。
MC_Axis000	_sAXIS_REF	-	轴1的轴变量。
MC_Axis000.Details.Homed	BOOL	FALSE	轴1为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为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	分配给MC_Power的实例PWR1的输出Status的变量。伺服ON状态时变为TRUE。
Pwr2_Status	BOOL	FALSE	分配给MC_Power的实例PWR2的输出Status的变量。伺服ON状态时变为TRUE。

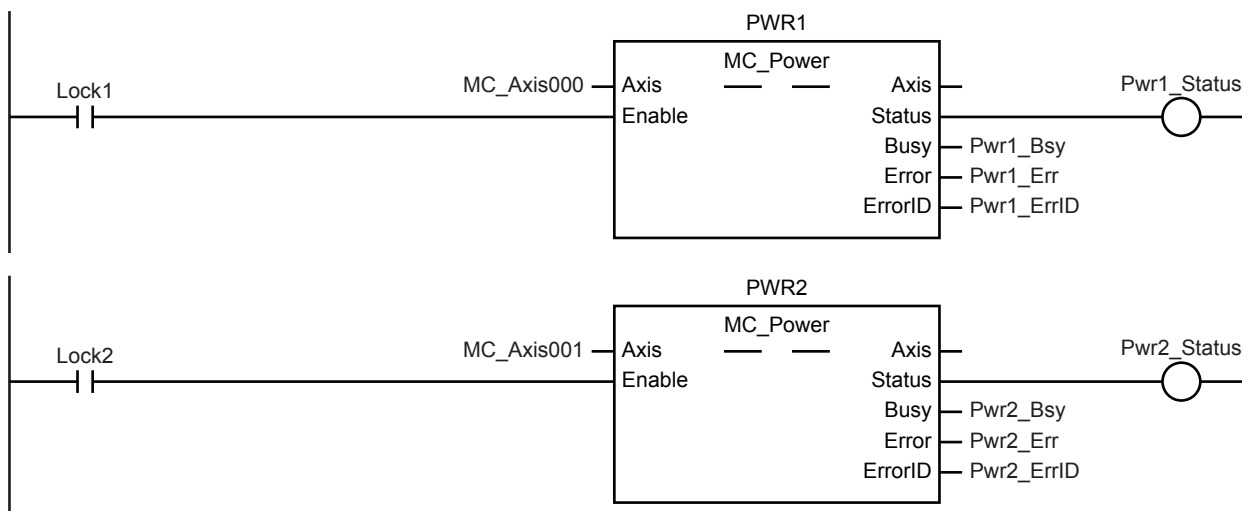
名称	数据类型	初始值	注释
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则将轴组中的各轴变为伺服ON状态。
InitFlag	BOOL	FALSE	表示输入参数设定完成。 FALSE时，设定输入参数。输入参数设定完成后，变为TRUE。

● 示例程序

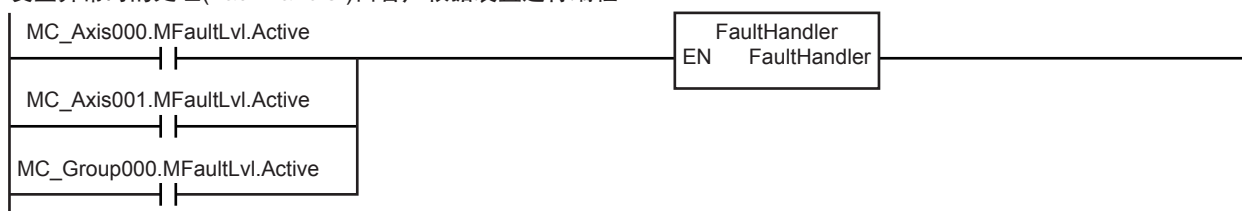
触点StartPg变为TRUE时，确认各轴的伺服驱动器处于伺服准备就绪状态



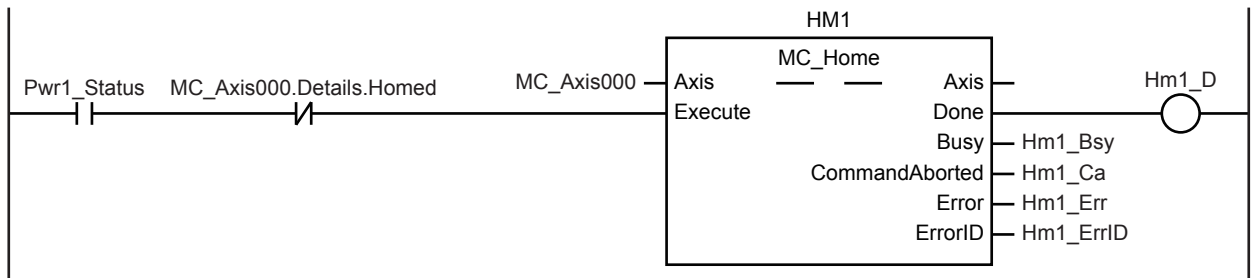
如果处于伺服准备就绪状态，则将各轴设为伺服ON状态



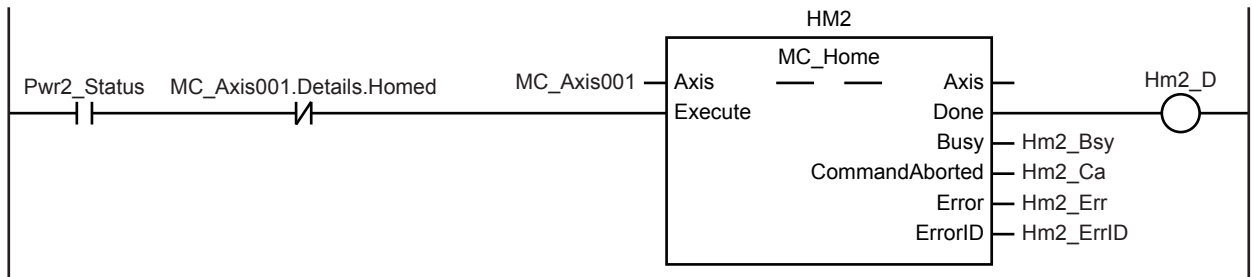
轴构成发生轻度故障程度的异常时，执行设备的异常处理(FaultHandler)。
发生异常时的处理(FaultHandler)由客户根据装置进行编程



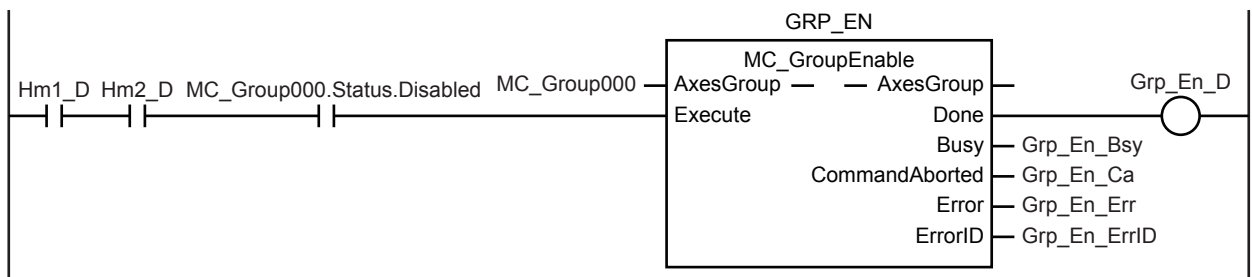
轴1变为伺服ON状态，若原点未确定，则执行原点复位，确定原点



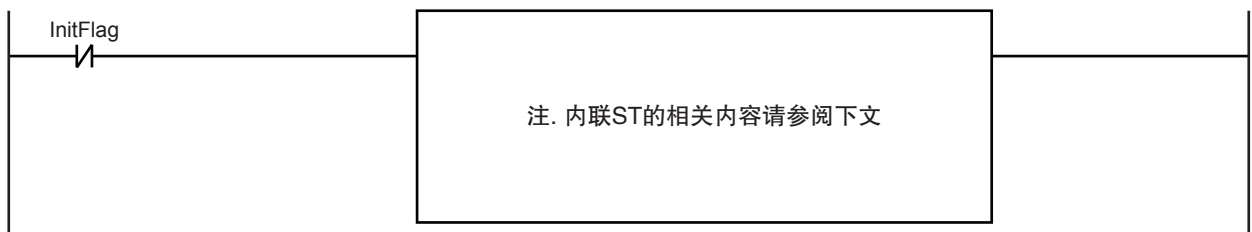
轴2变为伺服ON状态，若原点未确定，则执行原点复位，确定原点



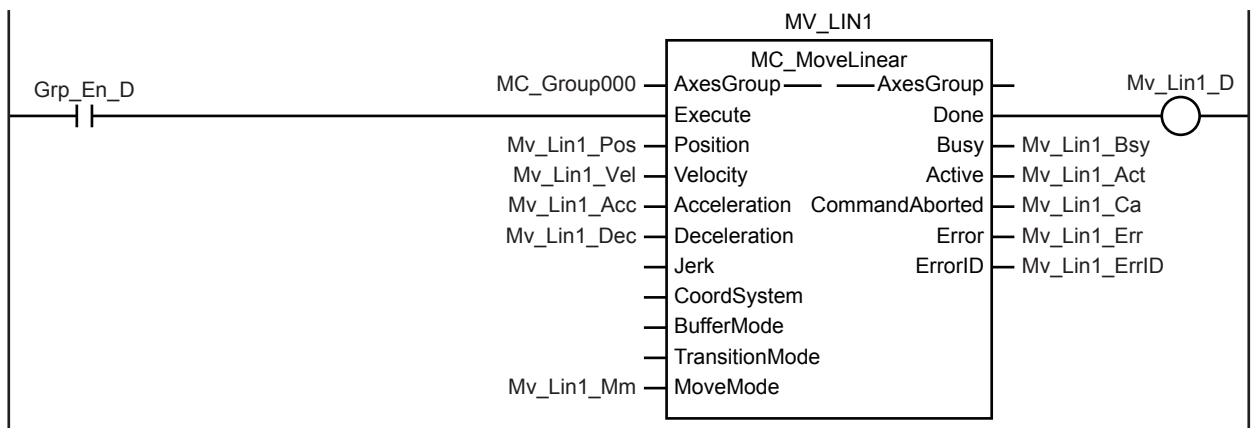
轴1、轴2原点确定后，启用轴组



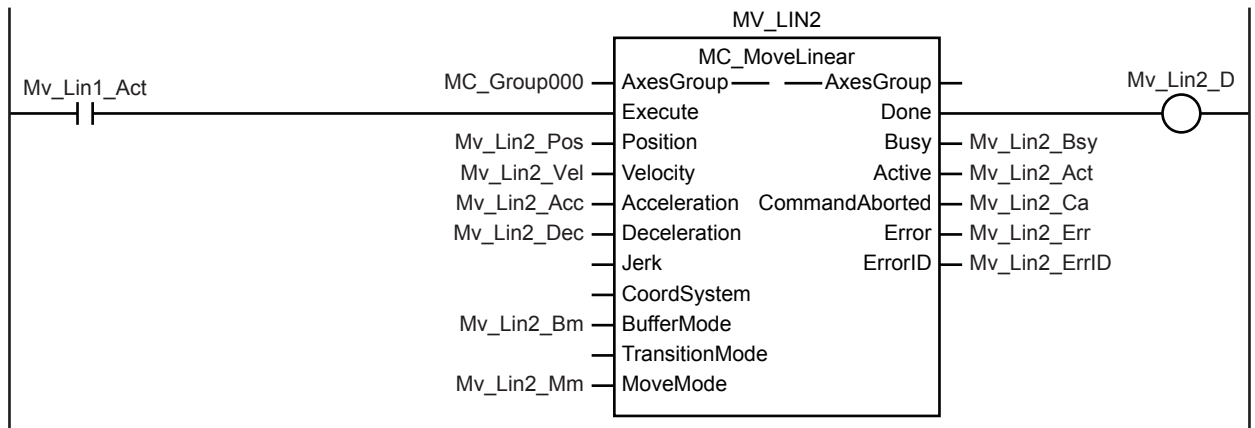
设定直线插补的参数



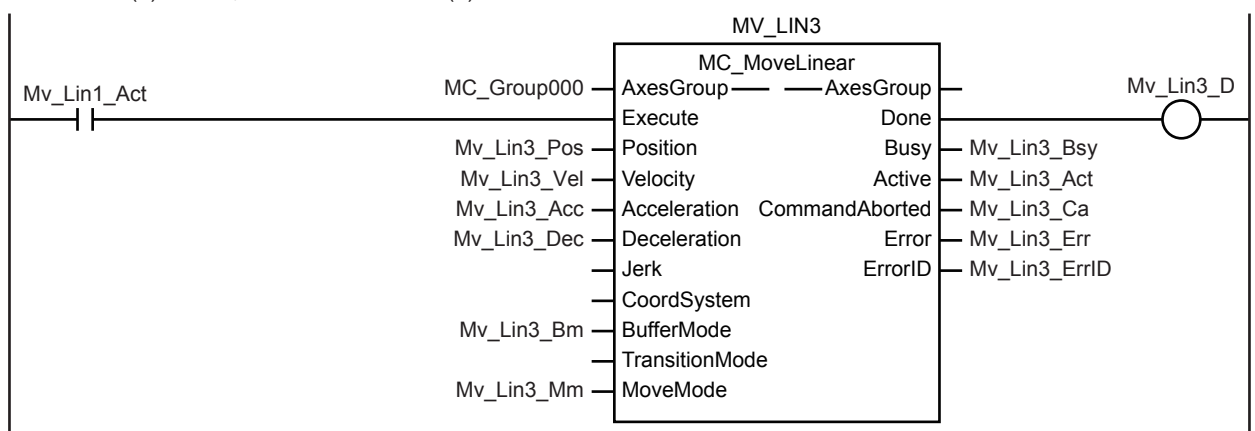
确认轴组有效后，执行直线插补(1)



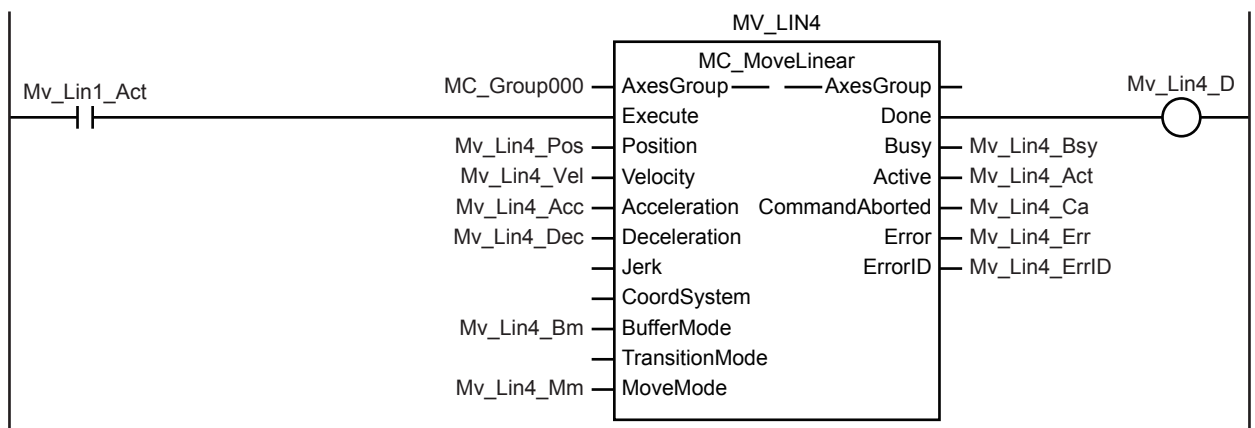
在直线插补(1)开始后，多重启动直线插补(2)



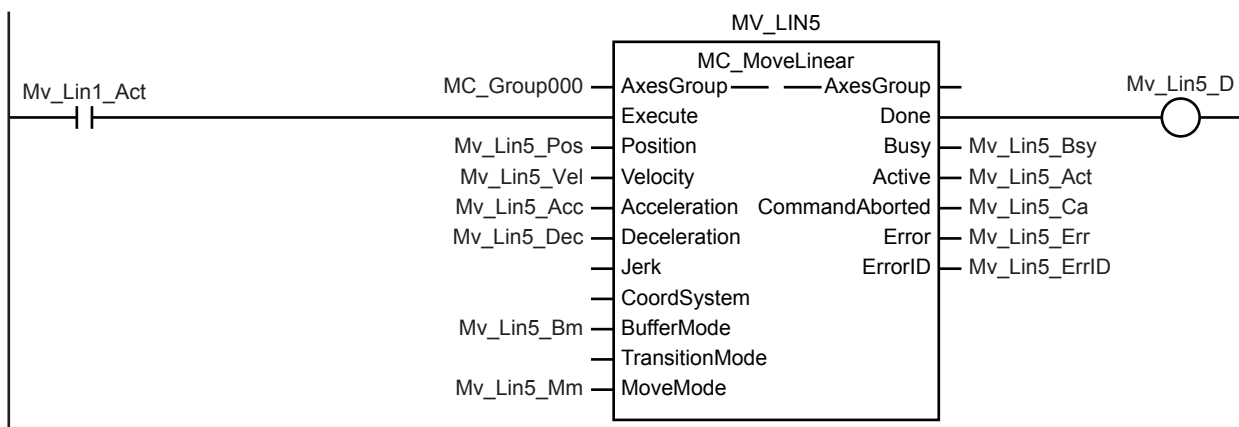
在直线插补(1)开始后，多重启动直线插补(3)



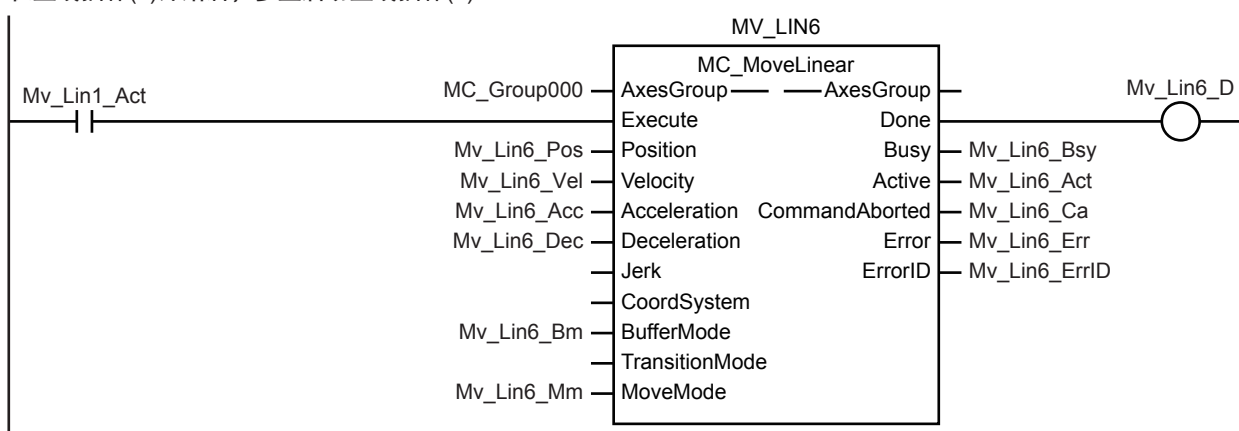
在直线插补(1)开始后，多重启动直线插补(4)



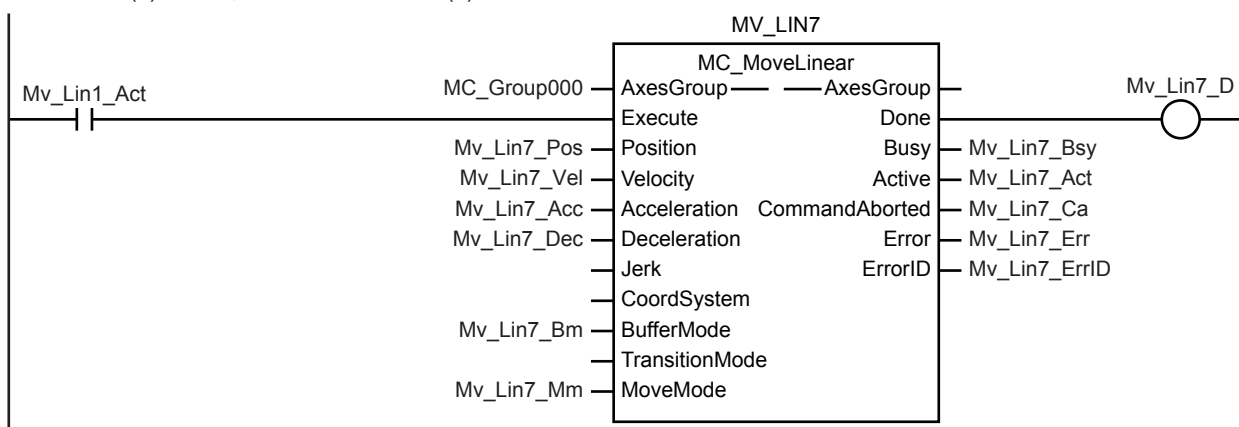
在直线插补(1)开始后, 多重启动直线插补(5)



在直线插补(1)开始后, 多重启动直线插补(6)



在直线插补(1)开始后, 多重启动直线插补(7)



内联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无效时变为TRUE。
MC_Axis000	_sAXIS_REF	-	轴1的轴变量。
MC_Axis000.Details.Homed	BOOL	FALSE	轴1为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为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	分配给MC_Power的实例PWR1的输出Status的变量。伺服ON状态时变为TRUE。
Pwr2_Status	BOOL	FALSE	分配给MC_Power的实例PWR2的输出Status的变量。伺服ON状态时变为TRUE。
StartPg	BOOL	FALSE	如果该变量为TRUE, EtherCAT的过程数据通信已建立, 则将轴组中的各轴变为伺服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，确认伺服驱动器处于伺服准备就绪状态，
//将轴1设为伺服ON状态。
//如果未处于伺服准备就绪状态，则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) 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，确认伺服驱动器处于伺服准备就绪状态，
//将轴2设为伺服 ON状态。
//如果未处于伺服准备就绪状态，则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis001.DrvStatus.Ready=TRUE) 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)内容由客户根据装置进行编程。
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、且原点未确定时，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、且原点未确定时，执行原点确定
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;

//轴1和轴2为原点确定状态、且轴组0无效时，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;

//启用轴组 MC_GroupEnable 结束后，执行直线插补(1)
IF Grp_En_D=TRUE THEN
    Mv_Lin1_Ex:=TRUE;
END_IF;

//在直线插补(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	在上升沿开始指令。
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)
BufferMode	缓存模式选择	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 ^{*3}	指定多重启动运动指令时的动作。 0: 中断 1: 缓存 2: 以低速合并 3: 以前一个速度合并 4: 以后一个速度合并 5: 以高速合并

输入变量	名称	数据类型	有效范围	初始值	内容
TransitionMode	过渡模式 (切换模式)	_eMC_ TRANSITION_ MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 *3	指定动作的路径。 0: 过渡无效 10: 附加角

- *1. 关于指令单位，请参阅  “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或  “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的 “单位转换设定”。
- *2. 请务必设定目标速度。不作设定就执行动作，将发生异常。
- *3. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

- *  请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	定位完成后	<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"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 执行MC_GroupStop指令中，启动本指令时 	<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含有异常因素时	异常已解除时

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. 请使用在Sysmac Studio的轴组基本设定画面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)指令规格相同。
详情请参阅 □ “功能说明(P.4-14)”。



使用注意事项

- 轴组中的任意一个构成轴原点未确定时，将发生“原点未确定状态下的指令启动异常(异常代码：5466 Hex)”的错误。
- 进行直线插补时，属于该轴组的任意一个逻辑轴的极限输入“ON”时，将无法启动指令。

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	在上升沿开始指令。
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)
BufferMode	缓存模式选择	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 ^{*3}	指定多重启动运动指令时的 动作。 0: 中断 1: 缓存 2: 以低速合并 3: 以前一个速度合并 4: 以后一个速度合并 5: 以高速合并

输入变量	名称	数据类型	有效范围	初始值	内容
Transition Mode	过渡模式 (切换模式)	_eMC_ TRANSITION_ MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 *3	指定动作的路径。 0: 过渡无效 10: 附加角

- *1. 关于指令单位，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的 “单位转换设定”。
- *2. 请务必设定目标速度。不作设定就执行动作，将发生异常。
- *3. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

- * □ 请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	定位完成后	<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"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 执行MC_GroupStop指令中，启动本指令时 	<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含有异常因素时	异常已解除时


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. 请使用在Sysmac Studio的轴组基本设定画面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)指令规格相同。
详情请参阅  “功能说明(P.4-14)”。



使用注意事项

- 轴组中的任意一个构成轴原点未确定时，将发生“原点未确定状态下的指令启动异常(异常代码：5466 Hex)”的错误。
- 进行直线插补时，属于该轴组的任意一个逻辑轴的极限输入“ON”时，将无法启动指令。

MC_MoveCircular2D

进行两轴的圆弧插补。

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	在上升沿开始指令。
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
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

输入变量	名称	数据类型	有效范围	初始值	内容
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	缓存 模式选择	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *1	指定多重启动运动指令时的 动作。 0: 中断 1: 缓存 2: 以低速合并 3: 以前一个速度合并 4: 以后一个速度合并 5: 以高速合并
Transition Mode	过渡模式 (切换模式)	_eMC_ TRANSITION_ MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 *1	指定动作的路径。 0: 过渡无效 10: 附加角
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. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

*2. 关于指令单位，请参阅  “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或  “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*3. 请务必设定目标速度。不作设定就执行动作，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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

* □ 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	定位完成后	<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"> 利用其它指令多重启动运动指令(中断)，中止本指令时 因发生异常，中止本指令时 发生异常过程中，启动本指令时 执行MC_GroupStop指令中，启动本指令时 	<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含有异常因素时	异常已解除时

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. 请在使用在Sysmac Studio的轴组基本设定画面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

- 以2维平面执行2轴的圆弧插补。



使用注意事项

- 轴组中的任意一个构成轴原点未确定时，将发生“原点未确定状态下的指令启动异常(异常代码：5466 Hex)”的错误。
- 进行2轴圆弧插补时，属于该轴组的任意一个逻辑轴的极限输入“ON”时，将无法启动指令。

指令详情

下面对本指令详细说明。

● 圆弧插补的步骤

圆弧插补按以下步骤执行。

1 登录进行插补的轴组

- 确定进行插补的轴组。
轴组即 `_MC_GRP[*]`或`_MC1_GRP[*]`、`_MC2_GRP[*]`。
- 通过该轴组变量的轴构成指定轴的构成。
- 通过该轴组变量的轴选择指定进行插补的轴的组合。
- 轴不使用轴号，而是使用逻辑轴(轴A0 ~ 轴A3)。
- 通过轴选择以前移方式向逻辑轴A0 ~ A3分别指定轴号。
- 根据型号、系列不同，轴号如下所示。

型号或系列名称	轴号
NX701 CPU单元	轴0 ~ 轴255
NX1P2 CPU单元	轴0 ~ 轴11
NJ系列CPU单元	轴0 ~ 轴63
NY系列控制器	轴0 ~ 轴63

例：轴构成为2轴，将轴号0 ~ 1指定为轴A0 ~ A1时，按以下方法指定。

逻辑轴	轴号	说明
轴A0	轴0	从轴A0开始，按照轴号从小到大的顺序进行指定。
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“不符合轴类型(错误代码：543D Hex)”的错误，结束动作。
请务必选择伺服轴或虚拟伺服轴。



参考

轴号的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

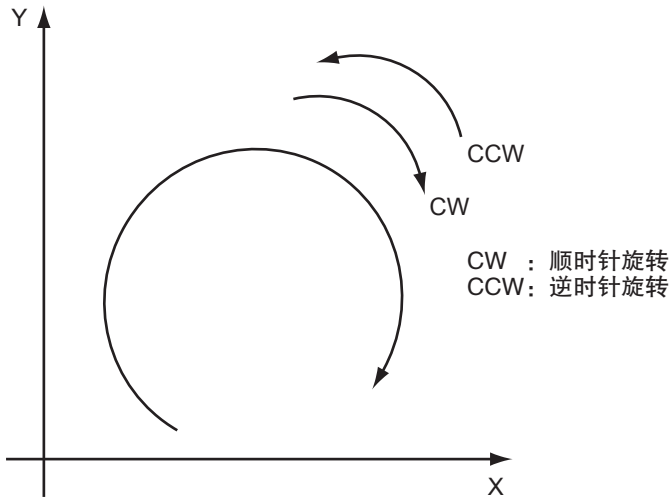
2 启用轴组

- 将轴组的各构成轴置为伺服ON状态，并使其处于原点确定状态。
- 执行MC_GroupEnable(启用轴组)指令，使登录的轴组有效。

这样，就可以使用圆弧插补指令了。

● CircAxes(圆弧轴指定)

圆弧插补使用的轴为X轴、Y轴。



通过CircAxes(圆弧轴指定)来指定向X轴、Y轴分配的轴。
轴的指定不使用轴号，而是使用逻辑轴(轴A0 ~ 轴A3)。



使用注意事项

指定为X轴或Y轴的轴的计数模式请设定为[线性模式]。
作为[旋转模式]启动时，将发生“计数模式设定导致的指令启动异常(错误代码：544A Hex)”。

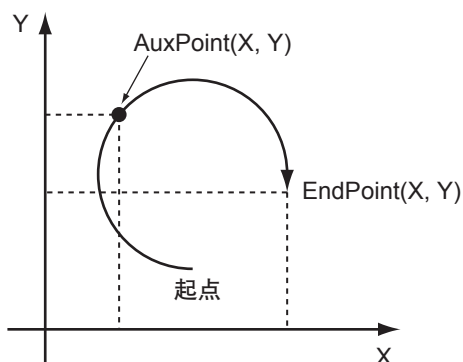
● CircMode(圆弧插补模式)

圆弧插补方式有“通过点指定/中心点指定/半径指定”三种，通过CircMode(圆弧插补模式)指定。以这些方法指定的位置有“绝对值定位”、“相对值定位”，通过MoveMode(移动方法选择)指定。

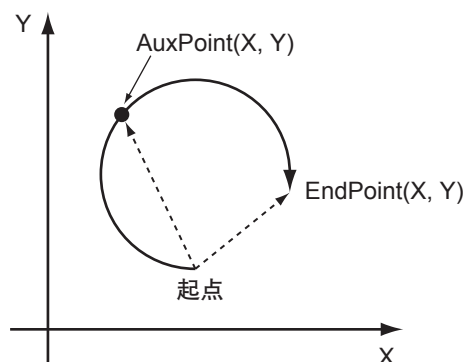
MoveMode	说明
绝对值定位	由通过点指定指定的通过点位置、以及由中心点指定指定的中心点位置和终点以自坐标系原点起的绝对位置来指定。
相对值定位	由通过点指定指定的通过点位置、以及由中心点指定指定的中心点位置和终点以自起点起的相对位置来指定。

下面以通过点指定为例，说明“绝对值定位”与“相对值定位”的差别。

圆弧方式：通过点指定 < 绝对值定位 >



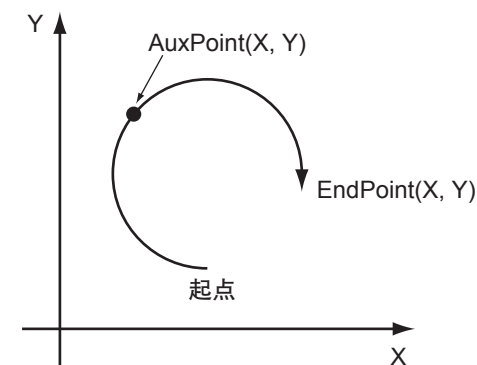
圆弧方式：通过点指定 < 相对值定位 >



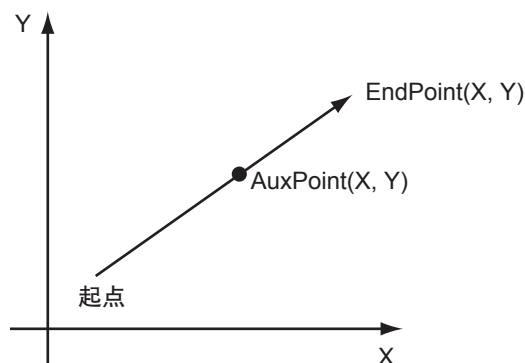
此后的说明均以MoveMode(移动方式选择)指定为“绝对值定位”为前提进行说明。

通过点指定

以当前位置为起点，执行连接通过点位置AuxPoint(X, Y)和终点EndPoint(X, Y)的圆弧插补。



起点、通过点位置与终点在同一条直线上、或者通过点位置与终点为同一点、以及起点和通过点位置为同一点时，执行从起点到终点的直线插补。

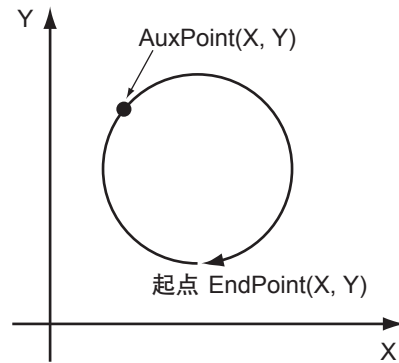




使用注意事项

- 所谓在同一条直线上，是指连接起点和终点的直线与通过点位置的距离的X坐标和Y坐标均不满1个脉冲。
- 起点、通过点位置、终点为同一点时，会发生异常。
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、脉冲转换位置相同时不视作同一点，不会发生异常。

起点和终点为同一点时，以起点和通过点为直径绘制正圆。这种情况下，通过PathChoice(路径选择)指定圆弧的旋转方向。



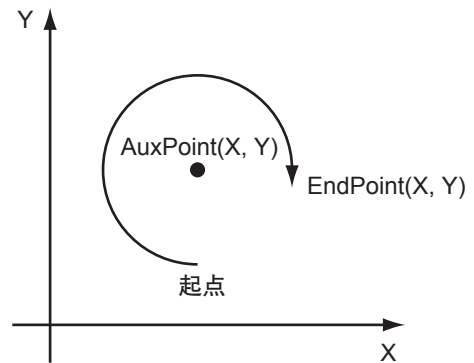
中心点指定

以当前位置为起点，执行以通过中心位置AuxPoint(X, Y)指定的圆弧连接终点 EndPoint(X, Y)的圆弧插补。

圆弧的旋转方向通过PathChoice(路径选择)指定。

起点、终点为同一点时，绘制正圆。

从指定的中心位置到起点的半径与到终点的半径不同时，使用两个半径的平均值进行圆弧插补。这种情况下，按照与半径指定同样的方法计算中心位置，使用该半径和中心位置。



使用注意事项

轴组参数[中心点补偿容许率]设定为非“0”时，若指定的中心点位置超出下式中的圆，会发生“超过圆弧插补中心点指定位置范围(错误代码：5449 Hex)”的错误。

将根据补偿的中心点算出的半径作为“100%”，
超过“半径×中心点指定式检查方法[%]/100”的圆时

半径指定

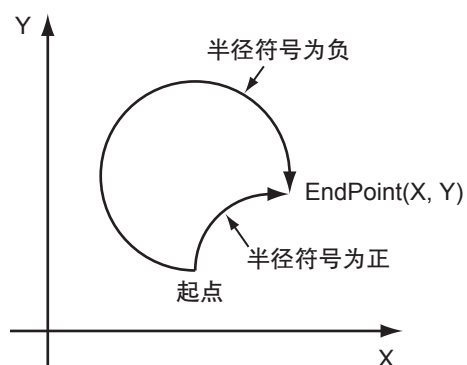
以当前位置为起点，执行以通过半径AuxPoint(X, Y)指定的圆弧连接终点EndPoint(X, Y)的圆弧插补。

半径通过AuxPoint(X, Y)的最初的元素来指定。不使用第二个元素。

例如，半径为100时，AuxPoint(X, Y)指定为AuxPoint(100, 0)。

半径符号为负时，绘制出较长的圆弧；半径符号为正时，绘制出较短的圆弧。

圆弧的旋转方向通过PathChoice(路径选择)指定。



使用注意事项

- 起点和终点为同一点时，会发生圆弧插补起点终点相同的异常，该组中的所有轴停止动作。
- 指定的半径小于起点和终点间距离的1/2时，无法绘制圆弧，将发生异常。

● 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、跃度通过Velocity、Acceleration、Deceleration、Jerk指定。
- 圆弧插补的插补速度指定为“0”时，会发生Velocity指定异常，该组中的所有轴停止动作。
- 指定的插补速度高于任意轴的最高速度时，将发生以下动作。

仅高于1个轴的最高速度时：自动调整插补速度，以该轴的最高速度动作。

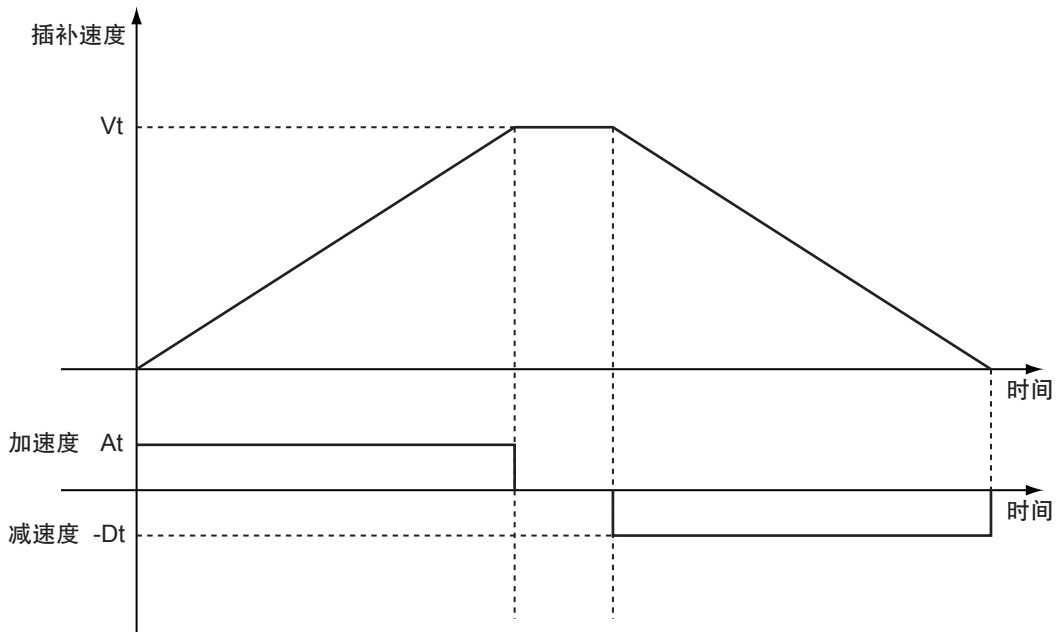
高于2个轴的最高速度时：自动调整插补速度，以两轴的最高速度中较慢的轴的最高速度动作。

Jerk(跃度)

Jerk(跃度)指定为“0”时和指定为非“0”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

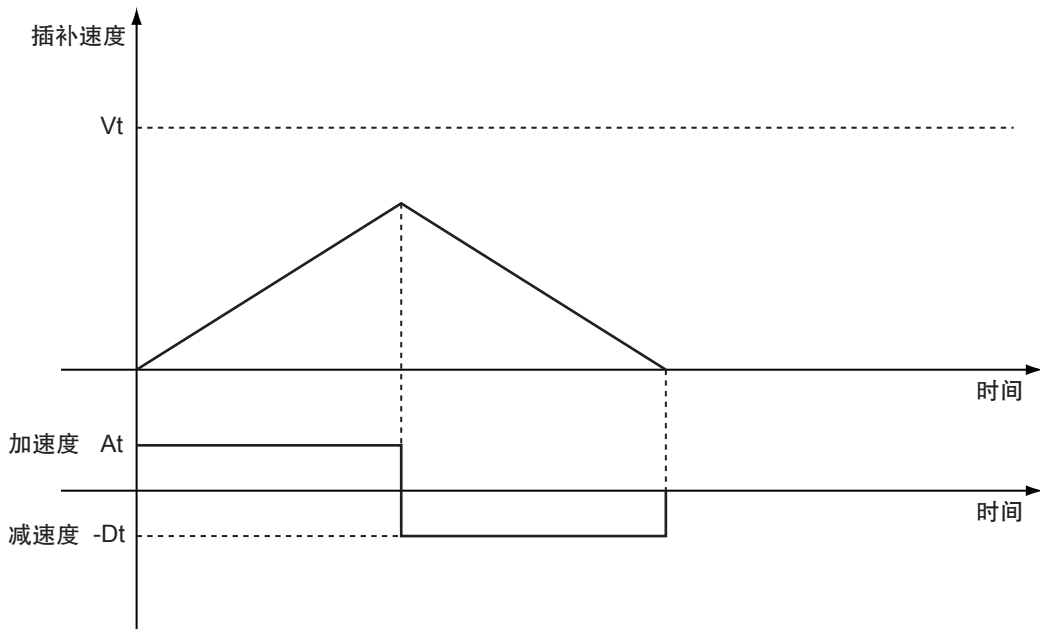
以加速度 A_t 、减速度 D_t 生成速度指令值。



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

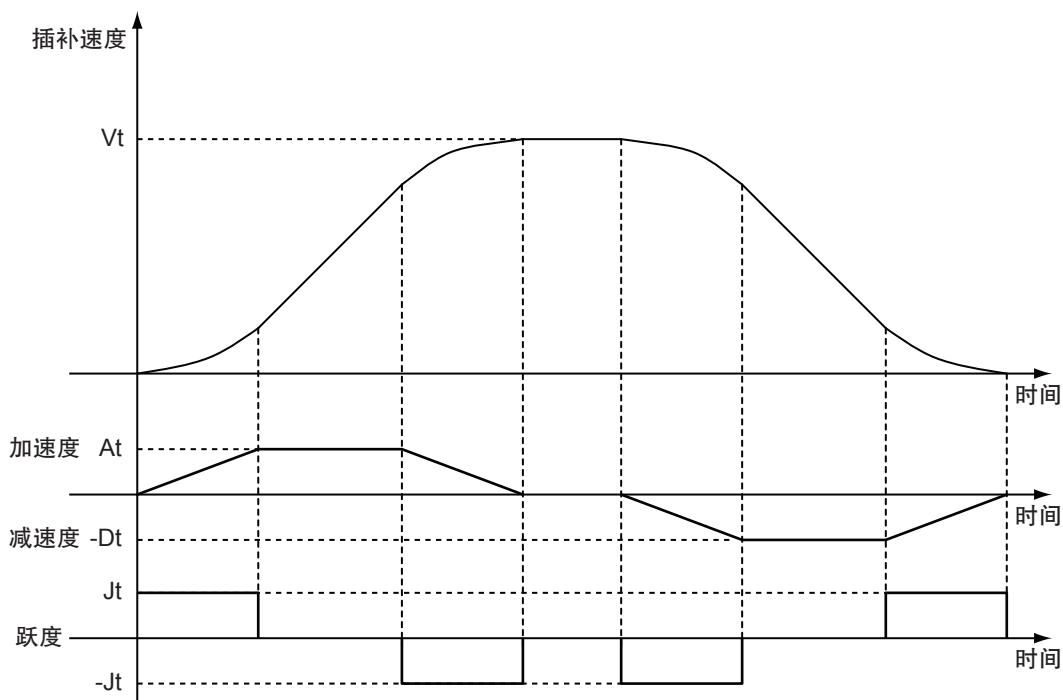
- Jerk(跃度)为“0”、移动距离短时

插补速度达不到指定值 V_t (目标速度)。



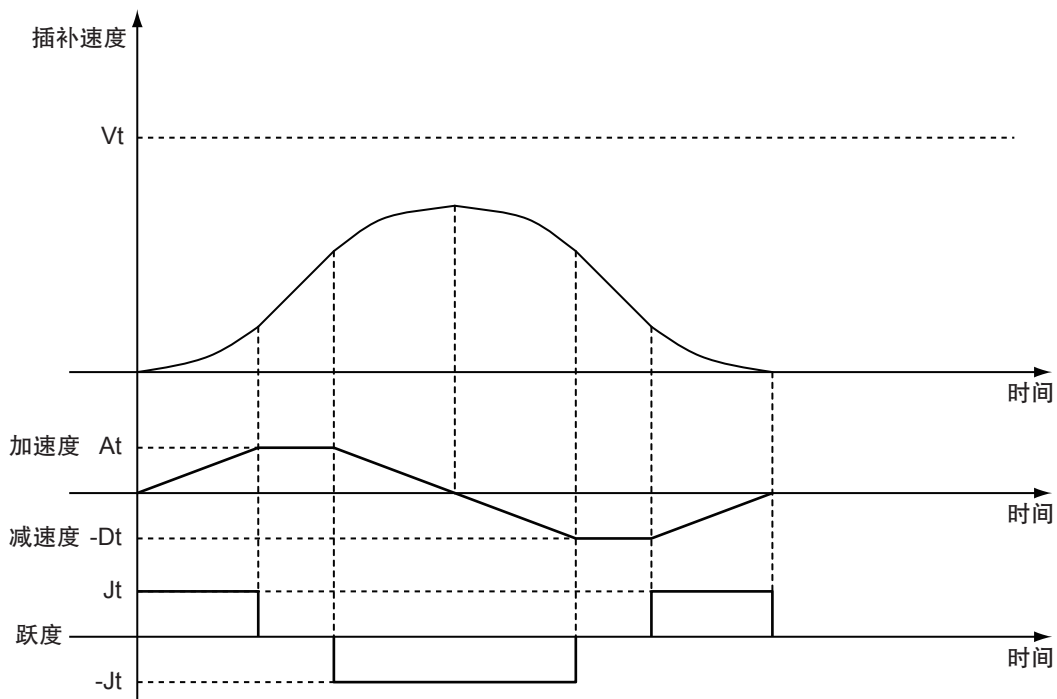
V_t : 插补速度的指定值、 A_t : 加速度的指定值、 D_t : 减速度的指定值

- Jerk(跃度)为非“0”时
以 A_t 为加速度上限、 D_t 为减速度上限生成速度指令值。



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 : 跃度的指定值

- Jerk(跃度)为非“0”、移动距离短时
插补速度达不到指定值 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 : 跃度的指定值



参考

- Acceleration(加速度)指定为“0”时，立即以指定的插补速度动作。
- Deceleration(减速度)指定为“0”时，立即停止。缓存模式指定为合并时，不停止动作，而在下一个动作时立即切换到指定的插补速度。
详情请参阅 □□ “BufferMode(缓存模式选择)(P.4-54)”。
- Acceleration(加速度)或Deceleration(减速度)指定为“0”时，Jerk(跃度)设定无效。

● CoordSystem(坐标系)

- 指定进行圆弧插补的坐标系。
- 只使用由2个以上的轴构成的轴坐标系(ACS)。

● BufferMode(缓存模式选择)

- 指定前一个插补动作和本次插补动作的连接方式。
- 有以下6种。

缓存模式选择	说明
中断	立即中止当前正在执行的指令，切换为本指令。 轴的动作方向因指令切换而反转时，根据轴参数中的“反转时动作”进行反转。
等待	当前正在执行的指令正常完成后，已缓存的本指令自动启动。
合并	以当前正在执行的指令到达目标位置时的速度(中继速度)为启动速度，连续使已缓存的本指令动作。变更当前正在执行的指令的动作，确保以中继速度到达目标位置。中继速度的指定方法分为如下4种。 此外，还有过渡指定(后述)作为合并的选项。
以低速合并	当前正在执行的目标速度与已缓存的目标速度中，以速度较低者为中继速度。
以前一个速度合并	以当前正在执行的目标速度为中继速度。
以后一个速度合并	以已缓存的本指令的目标速度为中继速度。
以高速合并	当前正在执行的目标速度与已缓存的目标速度中，以速度较高者为中继速度。

关于BufferMode(缓存模式选择)，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● TransitionMode(过渡模式)

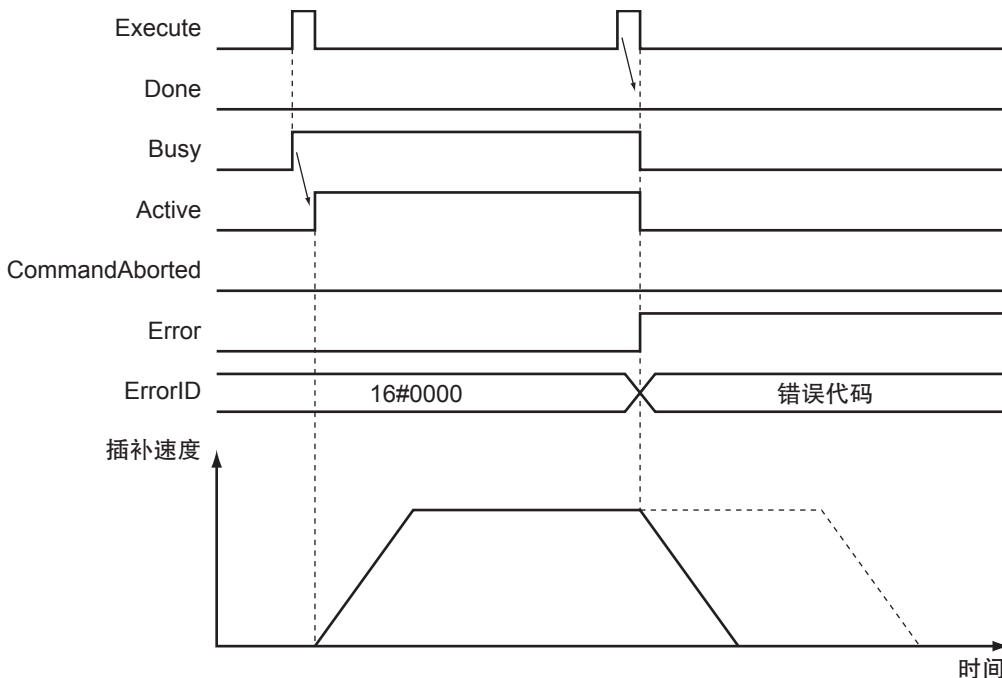
- 指定前一个插补动作和本次插补动作的轨迹如何连接。
- BufferMode(缓存模式选择)指定为合并时，TransitionMode(过渡模式)即变为有效。
- 若在未指定合并的情况下将TransitionMode(过渡模式)指定为_mcTMNone(过渡无效)以外的选项，则会发生异常。

详情请参阅 □□ “TransitionMode(过渡模式)(P.4-21)”。

重启运动指令

无法重启本指令。

若执行了重启，将发生“无法重启运动指令(错误代码: 543B Hex)”的错误，圆弧插补动作中的所有轴均停止动作。



多重启动运动指令

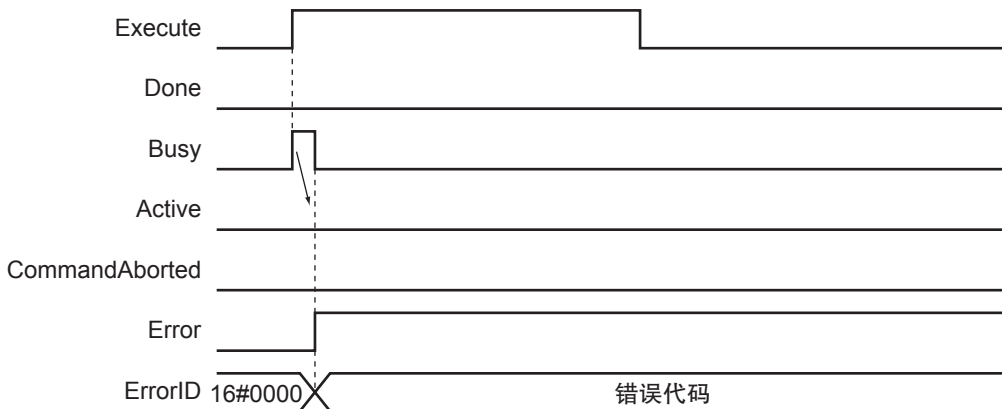
执行本指令时，可使用的指令有限制。

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。



● 错误代码

关于指令发生的异常，请参阅 □ “A-1 错误代码一览(P.A-2)”。

示例程序

下面，对通过多重启动进行圆弧插补的示例程序进行说明。

参数设定

该示例程序中使用的最少的必要设定如下所示。

● 轴参数的设定

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	线性模式
轴2	线性模式

坐标单位选择

轴	坐标单位
轴1	mm
轴2	mm

● 轴组参数的设定

轴构成

设为[2轴]。

轴选择

设为[轴1]、[轴2]。

动作示例

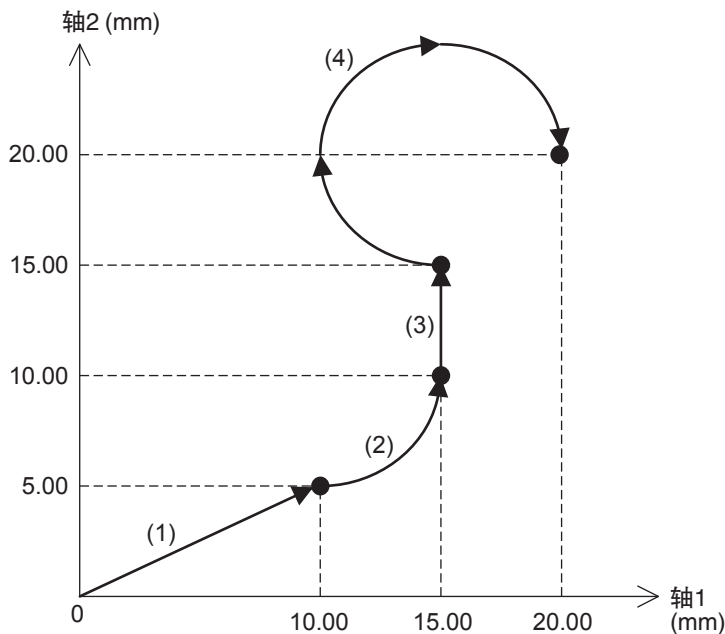
以下是从直线插补动作到自动进行圆弧插补定位的动作示例。

通过直线插补和圆弧插补移动到最终目标位置(20.00mm、20.00mm)。

将缓存模式选择指定为等待，通过多重启动执行指令。

在本例中，直线插补(1)的输出Active变为TRUE时，执行(2)~(4)的指令多重启动。多轴协调动作最多可多重启动7个指令。

● 动作模式



1 启动

在原点使动作开始开关“ON”，执行直线插补动作，移动到(10.00mm,5.00mm)处。

2 连续移动

执行圆弧插补动作，移动到(15.00mm,10.00mm)处，再执行直线插补动作，移动到(15.00mm,15.00mm)处，然后再次执行圆弧插补动作，定位至(20.00mm,20.00mm)处停止。

此时的速度为10.00mm/s。

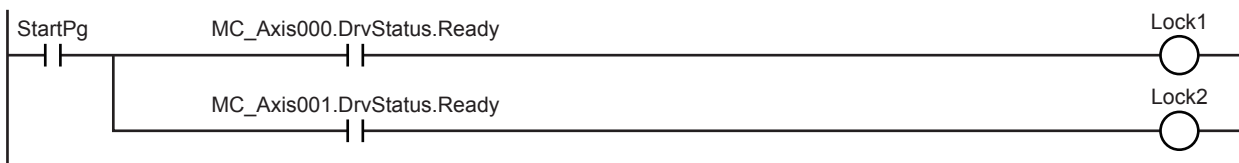
梯形图

● 主要变量

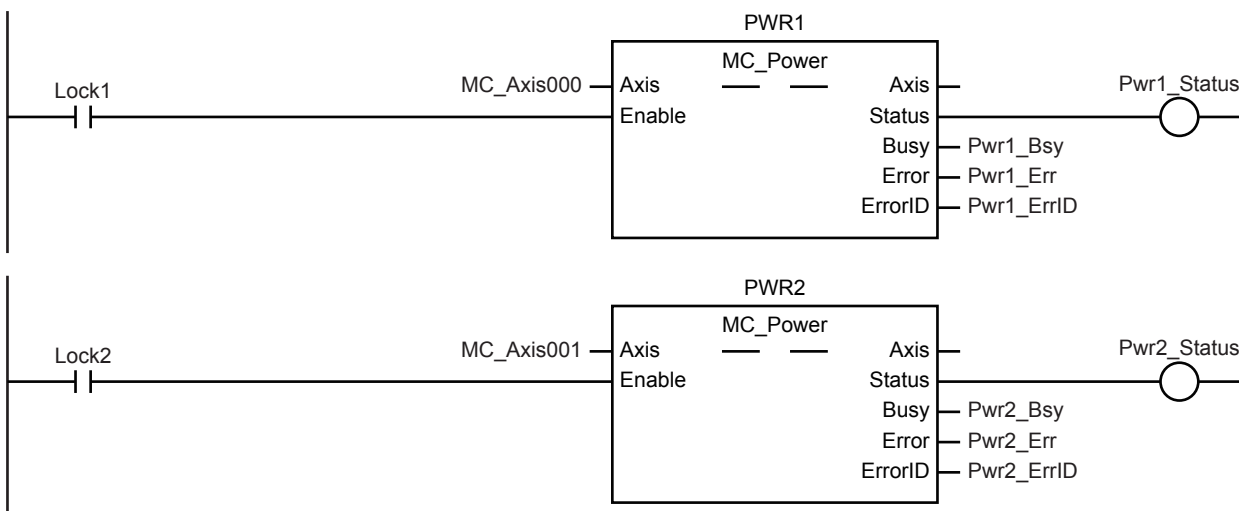
名称	数据类型	初始值	注释
MC_Group000	_sGROUP_REF	-	轴组0的轴组变量。
MC_Group000.MFaultLvl.Active	BOOL	FALSE	轴组0发生轻度故障等级的异常时变为TRUE。
MC_Group000.Status.Disabled	BOOL	FALSE	轴组0无效时变为TRUE。
MC_Axis000	_sAXIS_REF	-	轴1的轴变量。
MC_Axis000.Details.Homed	BOOL	FALSE	轴1为原点确定状态时变为TRUE。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴1发生轻度故障等级的异常时变为TRUE。
MC_Axis001	_sAXIS_REF	-	轴2的轴变量。
MC_Axis001.Details.Homed	BOOL	FALSE	轴2为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	分配给MC_Power的实例PWR1的输出Status的变量。伺服ON状态时变为TRUE。
Pwr2_Status	BOOL	FALSE	分配给MC_Power的实例PWR2的输出Status的变量。伺服ON状态时变为TRUE。
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则将轴组中的各轴变为伺服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

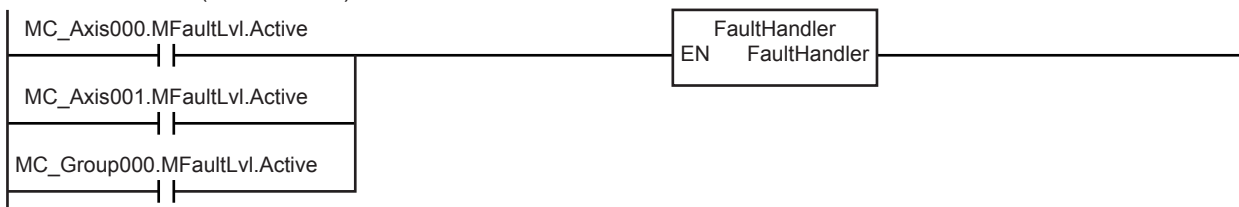
触点StartPg变为TRUE时，确认各轴的伺服驱动器处于伺服准备就绪状态



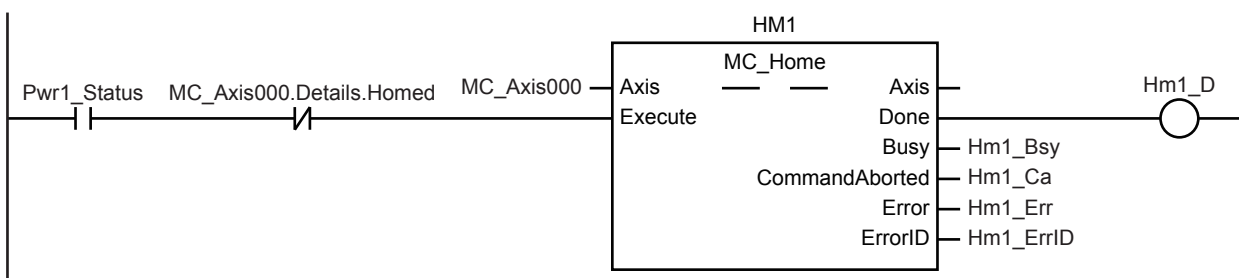
如果处于伺服准备就绪状态，则设为伺服ON状态



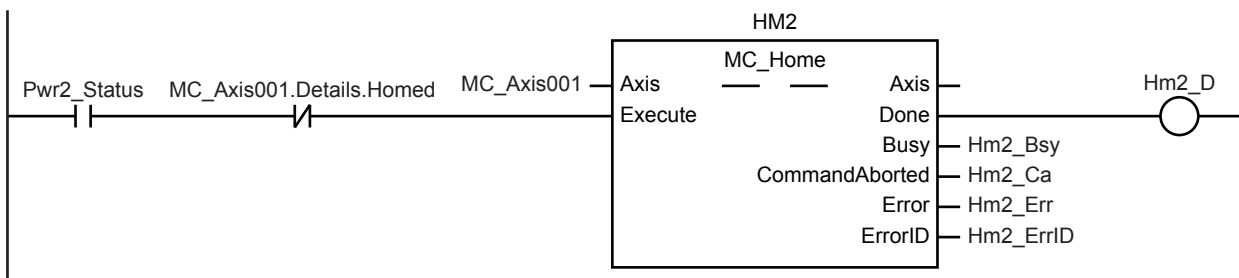
轴构成发生轻度故障程度的异常时，执行设备的异常处理(FaultHandler)。发生异常时的处理(FaultHandler)由客户根据装置进行编程



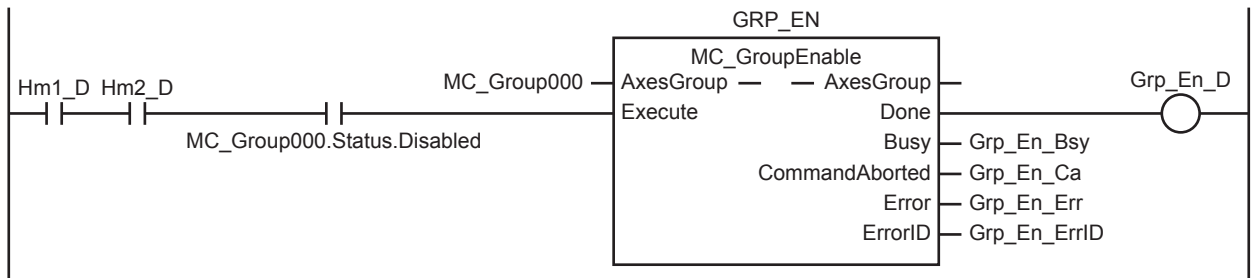
轴1变为伺服ON状态，若原点未确定，则执行原点复位



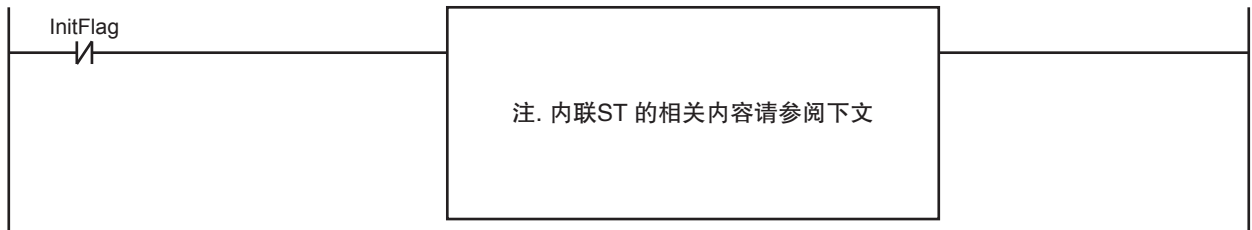
轴2变为伺服ON状态，若原点未确定，则执行原点复位



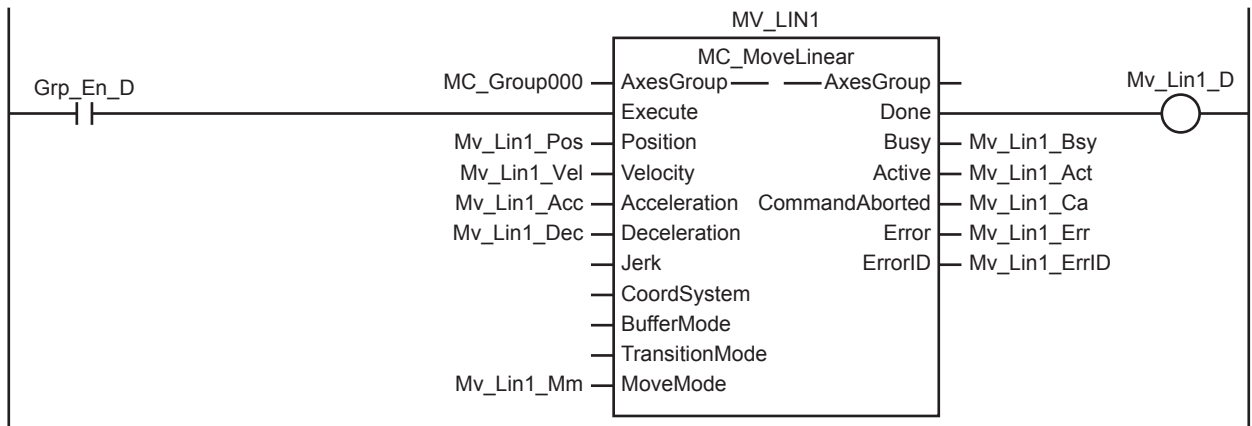
轴1、轴2原点确定后，启用轴组



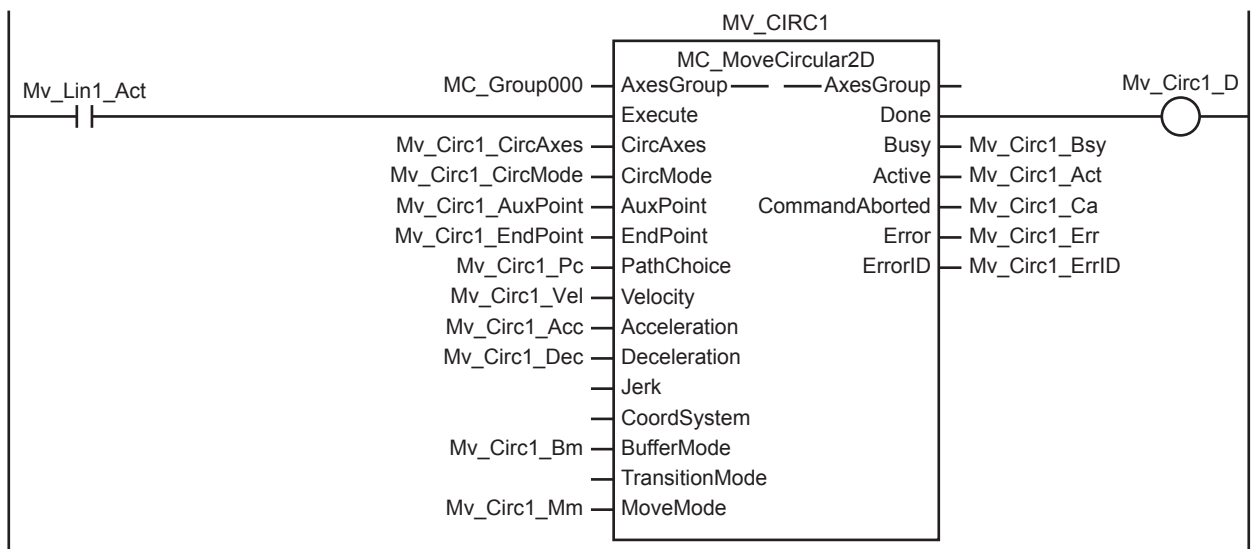
设定直线插补和圆弧插补的参数



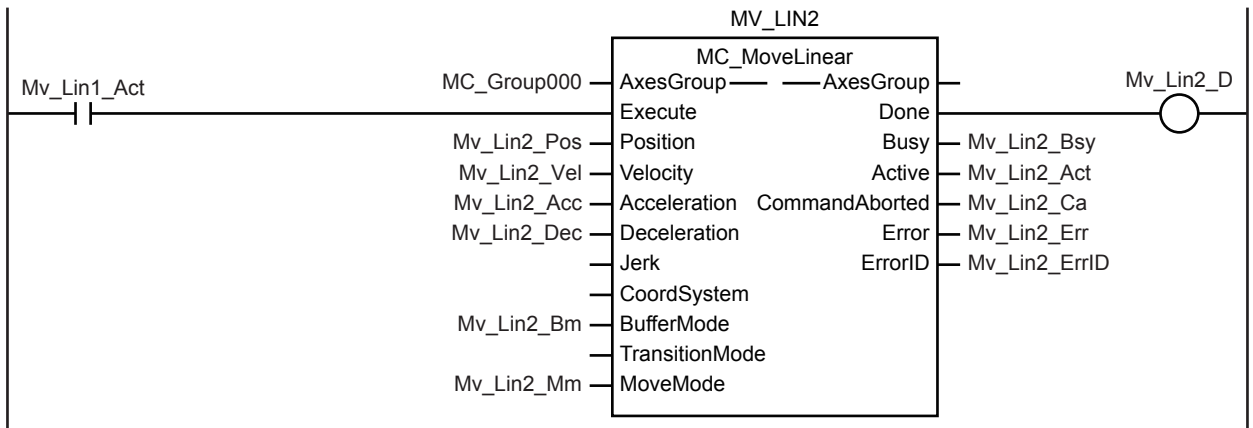
确认轴组有效后，执行直线插补(1)



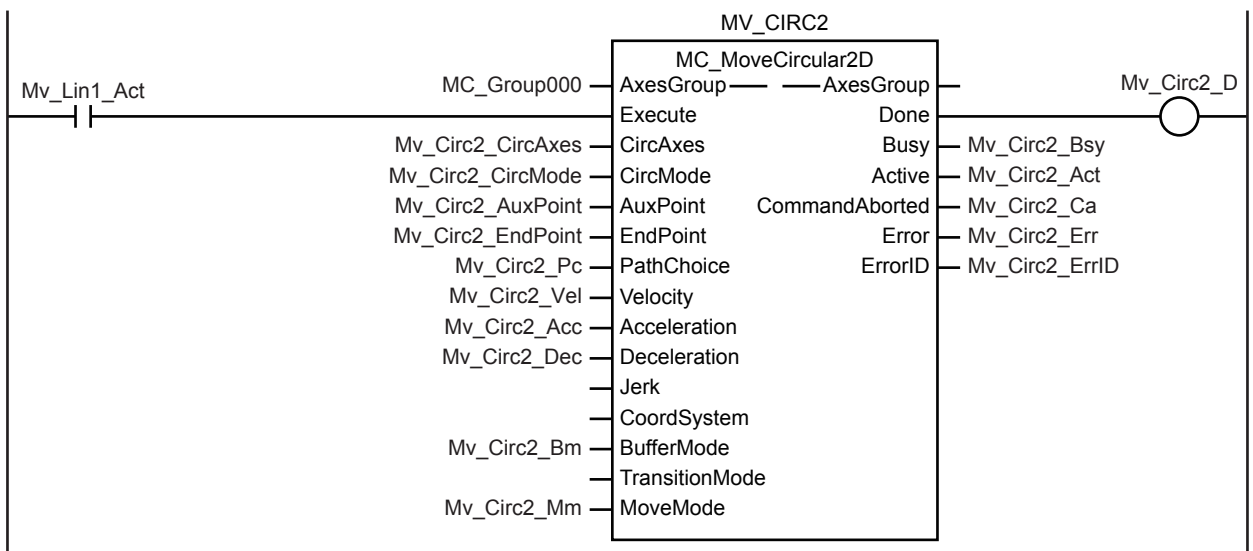
在直线插补(1)启动后，多重启动圆弧插补(2)



在直线插补(1)启动后，多重启动直线插补(3)



在直线插补(1)启动后，多重启动圆弧插补(4)



内联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无效时变为TRUE。
MC_Axis000	_sAXIS_REF	-	轴1的轴变量。
MC_Axis000.Details.Homed	BOOL	FALSE	轴1为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为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	分配给MC_Power的实例PWR1的输出Status的变量。伺服ON状态时变为TRUE。
Pwr2_Status	BOOL	FALSE	分配给MC_Power的实例PWR2的输出Status的变量。伺服ON状态时变为TRUE。
StartPg	BOOL	FALSE	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则将轴组中的各轴变为伺服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	该变量由FALSE→TRUE时，启动MC_Home的实例HM1。
Hm2_Ex	BOOL	FALSE	该变量由FALSE→TRUE时，启动MC_Home的实例HM2。
Grp_En_Ex	BOOL	FALSE	该变量由FALSE→TRUE时，启动MC_GroupEnable的实例GRP_EN。
Mv_Lin1_Ex	BOOL	FALSE	该变量由FALSE→TRUE时，启动MC_MoveLinear的实例MV_LIN1。
Mv_Lin2_Ex	BOOL	FALSE	该变量由FALSE→TRUE时，启动MC_MoveLinear的实例MV_LIN2。
Mv_Circ1_Ex	BOOL	FALSE	该变量由FALSE→TRUE时，启动MC_MoveCircular的实例MV_CIRC1。
Mv_Circ2_Ex	BOOL	FALSE	该变量由FALSE→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, 确认伺服驱动器处于伺服准备就绪状态,
//将轴1设为伺服ON状态。
//如果未处于伺服准备就绪状态, 则设为伺服OFF。
IF (StartPg=TRUE)
AND (MC_Axis000.DrvStatus.Ready=TRUE) 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，确认伺服驱动器处于伺服准备就绪状态，
//将轴2设为伺服ON状态。
//如果未处于伺服准备就绪状态，则设为伺服OFF。
IF (StartPg=TRUE)
  AND (MC_Axis001.DrvStatus.Ready=TRUE) 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)内容由客户根据装置进行编程。
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、且原点未确定时执行原点复位。
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、且原点未确定时执行原点复位。
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
  Hm2_Ex:=TRUE;
END_IF;

//轴1和轴2为原点复位完成状态、且轴组0无效时，将轴组0变为有效
IF (Hm1_D=TRUE) AND (Hm2_D=TRUE) AND (MC_Group000.Status.Disabled=TRUE) THEN
  Grp_En_Ex:= TRUE;
END_IF;

//轴组0变为有效时，启动直线插补(1)
IF Grp_En_D=TRUE THEN
  Mv_Lin1_Ex:=TRUE;
END_IF;

//在直线插补(1)的输出Active变为TRUE时，多重启动此后的指令
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
);

//启用轴组0
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	在上升沿开始指令。
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	缓存模式选择	_eMC_ BUFFER_ MODE	0: _mcAborting	0 ^{*2}	指定多重启动运动指令时的动作。 0: 中断

*1. 关于指令单位，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的“单位转换设定”。

*2. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时, 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为正常。

* 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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	指令开始时	<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"> 利用其它指令多重启动运动指令(中断), 中止本指令时 因发生异常, 中止本指令时 发生异常过程中, 启动本指令时 	<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含有异常因素时	异常已解除时

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. 请使用在Sysmac Studio的轴组基本设定画面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

- 本指令使通过轴组指令正在动作的AxesGroup(轴组)指定的轴组中的所有轴停止动作，并使轴组指令变为无效状态。
停止轴动作时，以Deceleration(减速度)减速停止的指令包括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。
停止轴动作时，与Deceleration(减速度)的设定值无关、立即停止的指令是MC_GroupSyncMoveAbsolute(轴组周期同步绝对位置控制)指令。
- 本指令的启动会导致插补动作中的指令CommandAborted(执行中断)。
- 若在执行插补指令的时候执行本指令，会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。
要在错误减速停止中停止轴组动作，请使用MC_GroupImmediateStop(立即停止)。

指令详情

下面对本指令详细说明。

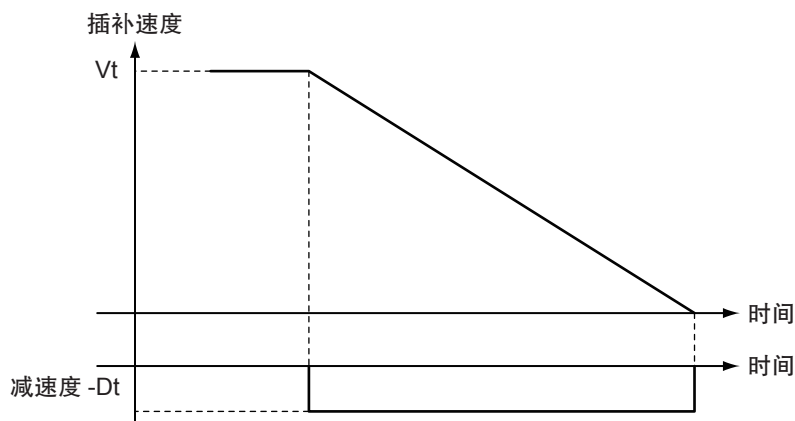
● Deceleration(减速度)、Jerk(跃度)

减速停止时的减速度、跃度由输入变量的Deceleration(减速度)、Jerk(跃度)指定。
执行本指令后，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”和非“0”时的减速度、插补速度的关系如下所示。

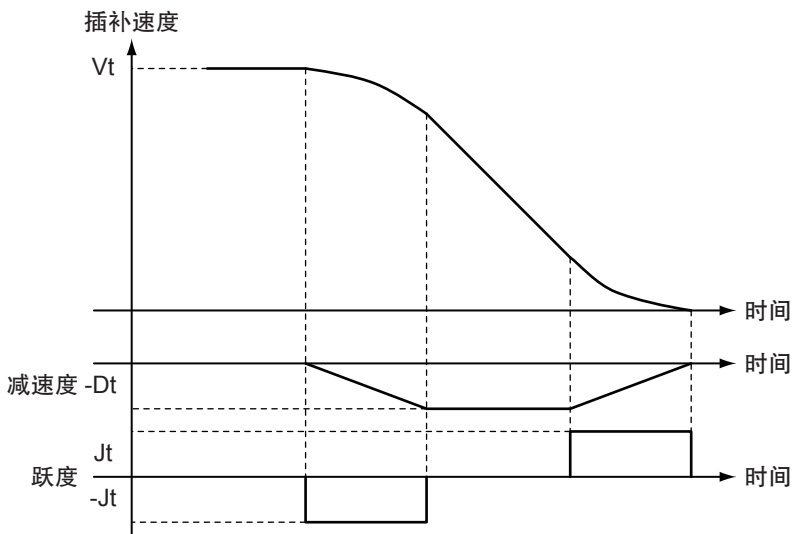
- Jerk(跃度)为“0”时
以减速度Dt生成速度的指令值。



V_t : 减速启动时的速度、 Dt : 减速度的指定值

- Jerk(跃度)为非“0”时

以减速度的上限为Dt,以当前速度为基点,生成速度的指令值。



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

- 将减速度指定为“0”时,执行立即停止指令,立即停止动作。
- 仅将减速度指定为“0”时,执行立即停止。与轴参数的“加减速超限”设定无关。

● BufferMode(缓存模式选择)

指定前一个插补动作和本次插补动作的连接方式。

有以下1种。

缓存模式选择	说明
中断	立即中止当前执行中的指令,执行本指令。

关于BufferMode(缓存模式选择),请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 到位检查

利用本指令停止时,不进行到位检查。

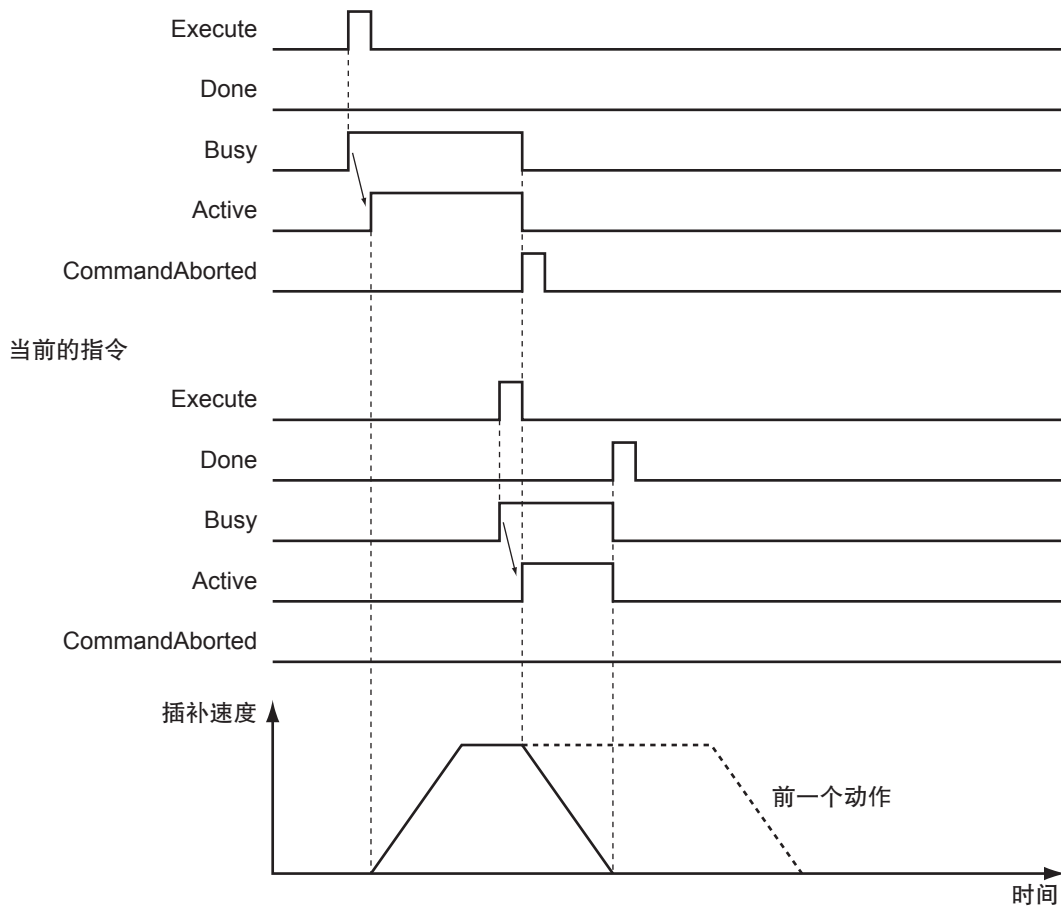
时序图

- 在启动Execute(启动)的同时，Busy(执行中)变为TRUE。在下一个周期Active(控制中)变为TRUE。
- 速度达到“0”时，Done(完成)变为TRUE。
- 利用其它指令中断本指令时，CommandAborted(执行中断)变为TRUE，Busy(执行中)、Active(控制中)变为FALSE。

以下是直线插补中减速停止的时序图。

本指令的启动会导致插补动作中的指令CommandAborted(执行中断)。

之前的指令(MC_MoveLinear)

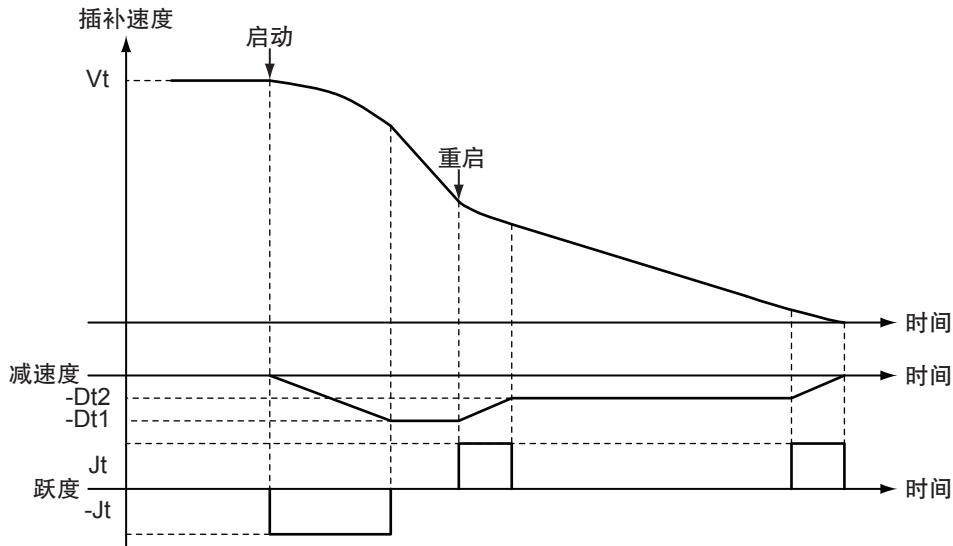


重启运动指令

在本指令执行中再次启动Execute(启动)时, 进行减速度的变更。
重启运动指令时, 不进行Jerk(跃度)的变更。

● Jerk(跃度)为非“0”时

以当前速度、当前减速度为基点, 以变更后减速度的上限为Dt2, 生成速度的指令值。



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

多重启动运动指令

执行本指令或本指令执行过程中, 可使用的状态及指令有限制。

多重启动运动指令的详情, 请参阅 □ “NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-363)” 或 □ “NY 系列工业用平板电脑/工业用台式电脑用户手册运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动 不启用轴组

若对无效的轴组执行MC_GroupStop(轴组强制停止)指令, 轴组会发生异常。对轴则没有影响。

轴组变量Status.Stopping(轴组减速停止中)为TRUE时

下述状态下, 轴变量Status.Stopping(减速停止中)变为TRUE。

- 通过MC_GroupStop(轴组强制停止)指令使轴组减速中
- 多重启动MC_GroupStop(轴组强制停止)指令的Execute, 至少1个变为TRUE时

对减速停止中的轴组启动以下指令时, 所启动指令的CommandAborted(执行中断)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

执行MC_GroupStop(轴组强制停止)指令时, 可以进行MC_GroupStop(轴组强制停止)指令的多重启动。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Status.ErrorStop(轴组错误减速停止中)变为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)会变为TRUE。

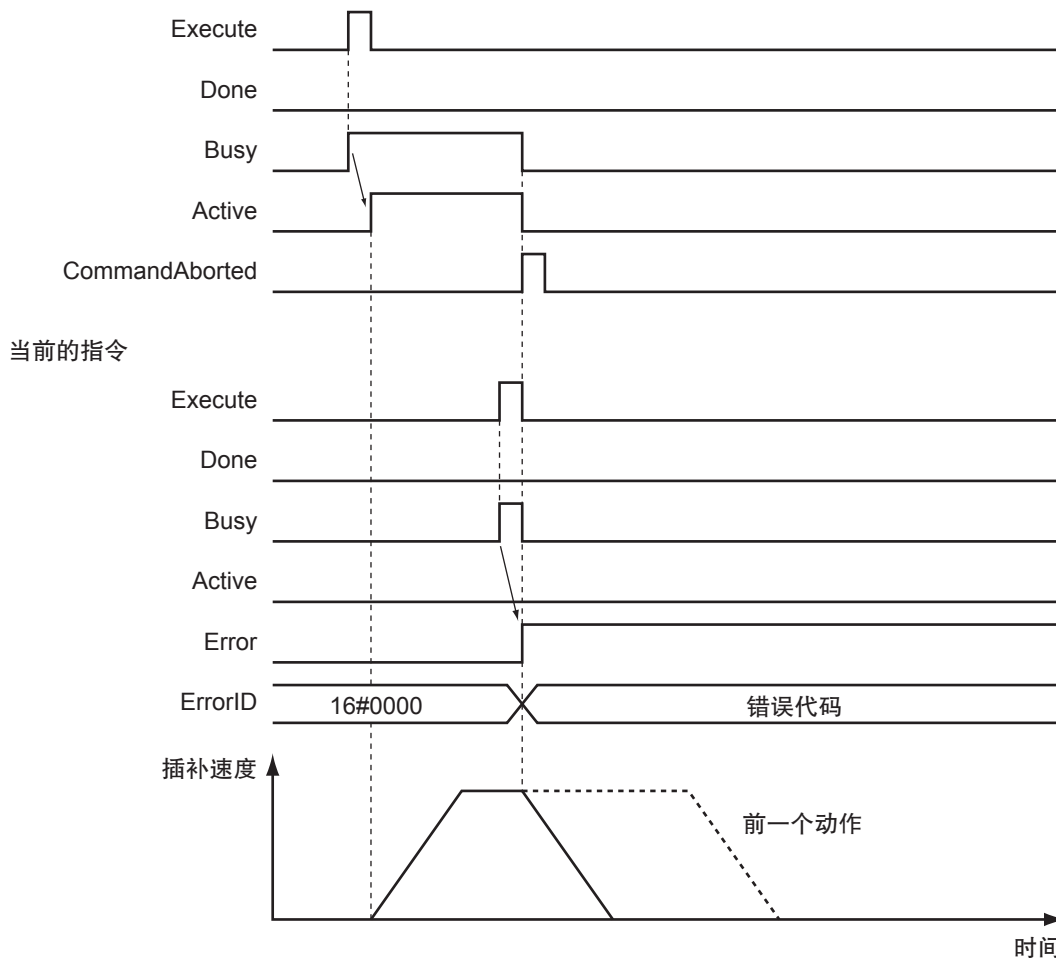
这种情况下，请使用MC_GroupImmediateStop(轴组立即停止)指令。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

之前的指令(MC_MoveLinear)



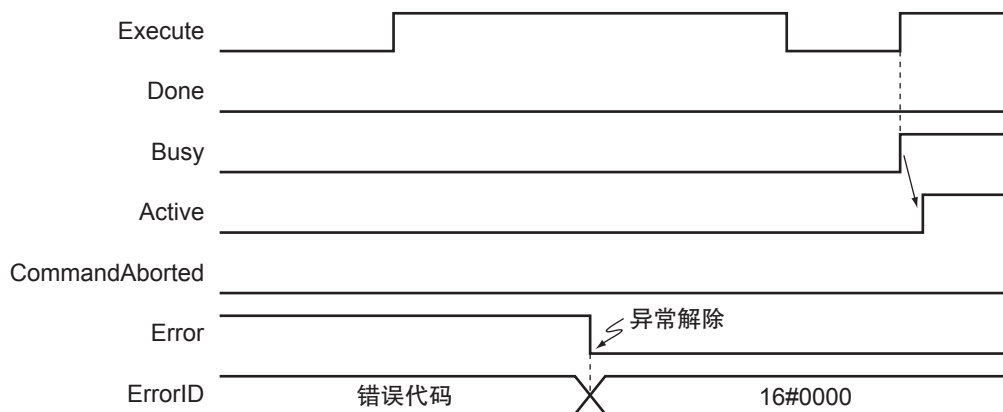
版本相关信息

根据CPU 单元的版本，异常解除时的动作如下所示。
请在轴组停止后解除异常。轴组动作中不能进行异常解除。

- Ver.1.10 以上版本的CPU 单元时

解除本指令的异常后，在下一个Execute(启动)的上升沿才可启动本指令。

Ver.1.10以上



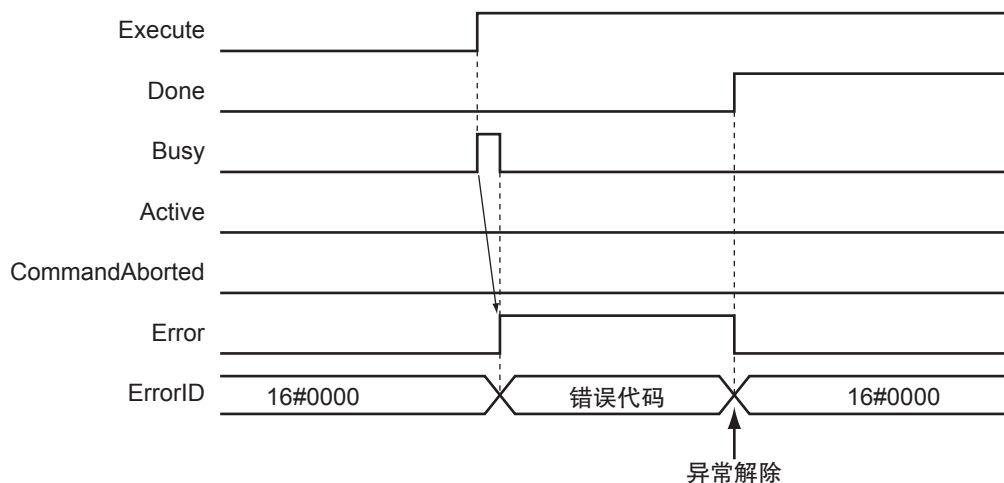
- Ver.1.09以下版本的CPU单元时

本指令发生异常时，如果在Execute(启动)为TRUE的状态下解除异常，会发生以下动作。

- 如解除了异常原因，则Error(错误)变为FALSE、Done(完成)变为TRUE。轴变量 Status.Stopping(减速停止中)与正常执行减速停止时一样，变为TRUE。
- 未解除异常原因时，则本指令将再次变为Error(错误)，且轴组发生异常。

解除异常原因时的时序图如下所示。

Ver.1.09以下



● 错误代码

关于本指令发生的异常，请参阅 “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。

若对象轴组为有效状态，则无论轴的状态如何，构成轴的所有轴均按照轴参数的[立即停止输入停止方法]停止。

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	在上升沿开始指令。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

* 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	立即停止完成时	<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"> 利用其它的MC_GroupStop(轴组强制停止)指令多重启动运动指令(中断),中止本指令时 因发生异常，中止本指令时 	<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含有异常因素时	异常已解除时

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. 请使用在Sysmac Studio的轴组基本设定画面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[*])。

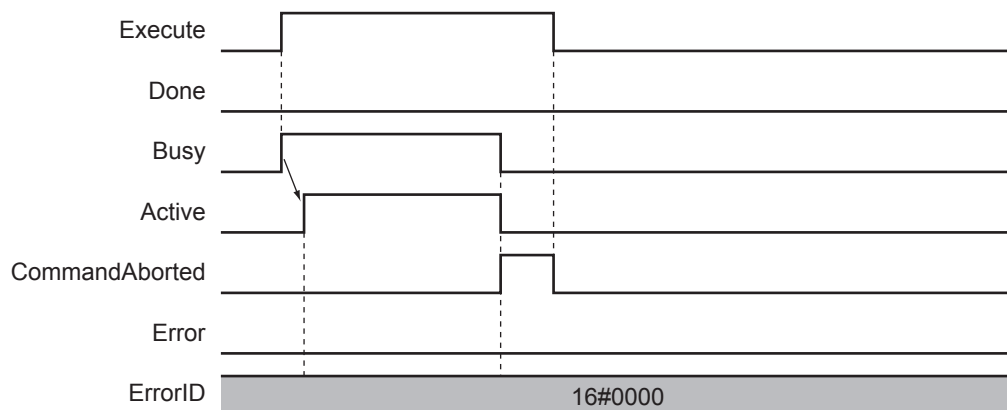
功能说明

- 可以对通过轴组指令正在动作的AxesGroup(轴组)指定的轴组中的所有轴、以及MC_GroupStop(轴组强制停止)指令、或因异常发生而停止的轴组执行本指令。
- 执行指令后，则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，动作中的指令变为CommandAborted(执行中断)状态。
- 执行本指令后，轴组状态Status.ErrorStop(错误减速停止中)变为TRUE，发生“轴组立即停止指令启动(错误代码：5486 Hex)”错误。
关于“轴组立即停止指令启动(错误代码：5486 Hex)”，请参阅 □□ “NJ/NX系列 CPU单元用户手册 运动控制篇(SBCE-363)”或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

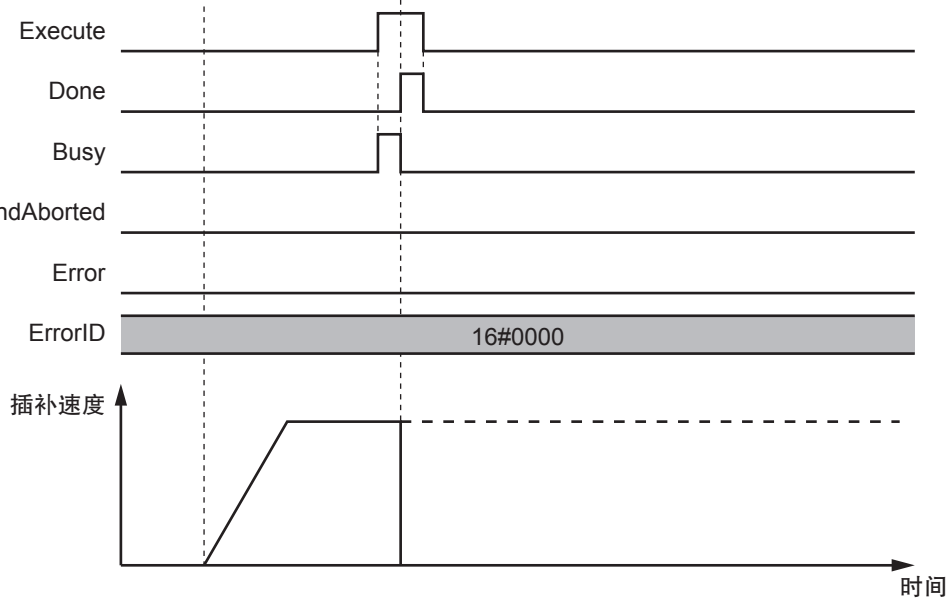
时序图

- 在启动Execute(启动)的同时，Busy(执行中)变为TRUE。
- 本指令处理完成时，Done(完成)变为TRUE。

MC_MoveLiner指令



MC_GroupImmediateStop指令



重启运动指令

无法重启本指令。

若执行了重启，将发生“无法重启运动指令(错误代码：543B Hex)”的错误，该轴停止动作。

多重启动运动指令

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动 不启用轴组

若对无效的轴组执行本指令，轴组会发生异常。对轴则没有影响。

轴组变量Status.Stopping(减速停止中)为TRUE时

以下状态下，轴组变量Status.Stopping(减速停止中)变为TRUE。

- 通过MC_GroupStop(轴组强制停止)指令使轴组减速中
- 多重启动MC_GroupStop(轴组强制停止)指令的Execute，至少1个变为TRUE时

对减速停止中状态的轴组也能通过本指令执行多重启动。

通过本指令的启动，执行中的MC_GroupStop(轴组强制停止)指令的CommandAborted(执行中断)变为TRUE。

轴组变量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。

轴组错误减速停止中也能执行本指令。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 错误代码

关于本指令发生的异常，请参阅 □ “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 超调值设定	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	设为TRUE时，超调变为有效。 设为FALSE时，超调返回“100%”。
VelFactor	速度超调值	LREAL	0 ~ 500	100	指定速度的超调值。 超调值的有效范围为“0.01 ~ 500.00”。 “500.00以上”作为“500”处理，“0.01以下(包括负数)”作为“0.01”处理。 只有指定为“0”时才作为“0”动作。 单位为[%]。
AccFactor (Reserved)	加减速度超调值	LREAL	0 ~ 500	100	(Reserved)
JerkFactor (Reserved)	跃度超调值	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	控制中变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为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为正常。

*  请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Enabled	开始执行本指令时	<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含有异常因素时	异常已解除时

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. 请使用在Sysmac Studio的轴组基本设定画面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

- 变更对于以下一个指令动作的组的插补目标速度的超调。
通过变更超调，变更动作中的目标速度。
对象指令如下所示。

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。
变更后的目标速度 = 当前执行中指令的插补速度 × 超调[%]
- 超调单位为[%]。以“100”表示“100%”。
- 变更后的插补速度超过轴组变量的最高插补速度时，使用最高插补速度。
- 相对变更后的插补速度，使轴加速或减速。
- 将超调指定为“0”时，目标速度变为“0”，组的动作表现为减速停止，以速度“0”动作。动作中轴组变量Status.Moving(动作中)变为TRUE。
希望保持动作状态、但又想暂时停止轴动作时，将超调设为“0”。
- Enable(有效)为FALSE时，超调返回100(=100%)。
- 执行MC_GroupSetOverride(轴组超调值设定)指令时若发生轴组异常，MC_GroupSetOverride(轴组超调值设定)指令保持Enabled(有效)状态。



使用注意事项

将本指令的Enable(有效)设为FALSE时，本指令的Enabled(有效)和Busy(执行中)变为FALSE。此时，以超调“100%”为目标进行加速或减速。



参考

对其它指令的影响

要临时变更各指令的目标速度时，使用本指令。

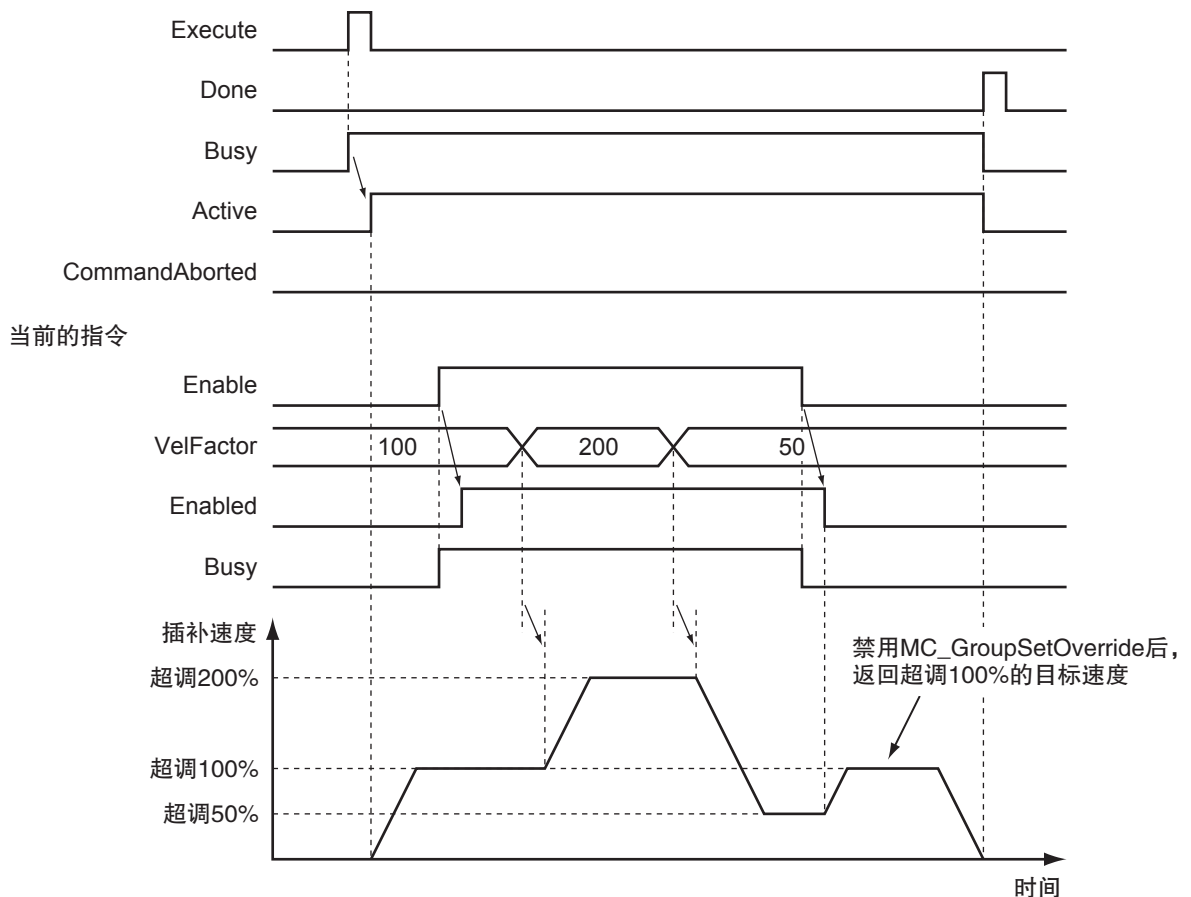
因此，对于输入不带目标速度的指令，本指令不起作用。

但是，对于本指令无效的指令，即使将MC_GroupSetOverride(轴组超调值设定)指令设为有效，Enabled(有效)也会继续保持TRUE。

时序图

● MC_MoveLinear(直线插补)指令使用本指令时

之前的指令(MC_MoveLinear)



重启运动指令

输入为“Enable型”指令时，不会重启运动指令。

多重启动运动指令

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 本指令执行中的其它指令启动

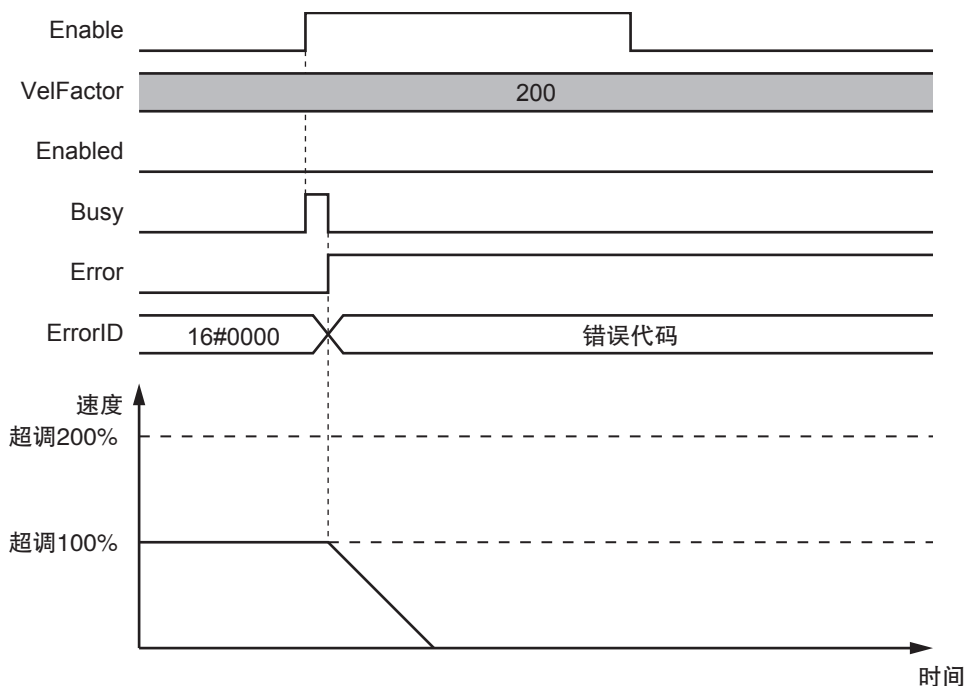
执行MC_GroupSetOverride(轴组超调值设定)指令时，若对同一个轴组启动了其它实例的MC_GroupSetOverride(轴组超调值设定)指令，则优先处理后执行的实例。

Enabled(有效)对两种指令均表现为TRUE。

作为具体动作，后执行的实例的超调值生效。将后执行的实例的Enable(有效)设为FALSE后，超调无效。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，发生轴组轻度故障时，轴组停止动作。可查看ErrorID(错误代码)的输出值，了解发生异常的原因。



● 错误代码

关于指令发生的异常，请参阅 □ “A-1 错误代码一览(P.A-2)”。

MC_GroupReadPosition

获取轴组的指令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_GroupRead Position	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

Ver.1.01以上的CPU单元和Ver.1.02以上的Sysmac Studio组合时可使用本指令。

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	在上升沿开始指令。
CoordSystem	坐标系	_eMC_COORD_ SYSTEM	0: _mcACS	0*1	指定坐标系。 0: 轴坐标系(ACS)

*1. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

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	控制中变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为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为正常。
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. 请参阅 □ “A-1 错误代码一览(P.A-2)”。

*2. 关于指令单位, 请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的 “单位转换设定”。

(注) 1 轴组由2轴或3轴构成时, 空的构成轴的CommandPosition(指令当前位置)及ActualPosition(反馈当前位置)的值不稳定。

2 Enable(有效)变为FALSE时, CommandPosition(指令当前位置)及ActualPosition(反馈当前位置)的值不稳定。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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含有异常因素时	异常已解除时

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. 请使用在Sysmac Studio的轴组基本设定画面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，按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。
- 不同轴种类可使用的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	可以使用	可以使用
虚拟伺服轴	可以使用	可以使用*1
编码器轴	不可使用	可以使用
虚拟编码器轴	不可使用	可以使用

*1. 虚拟伺服轴时，“反馈位置 = 指令位置”。

(由于在MC功能模块内部是以双精度实数型进行处理，因此可能包括运算误差。)

- 即使原点未确定也可执行本指令。
- 可以使用本指令，与轴组有效、无效无关。



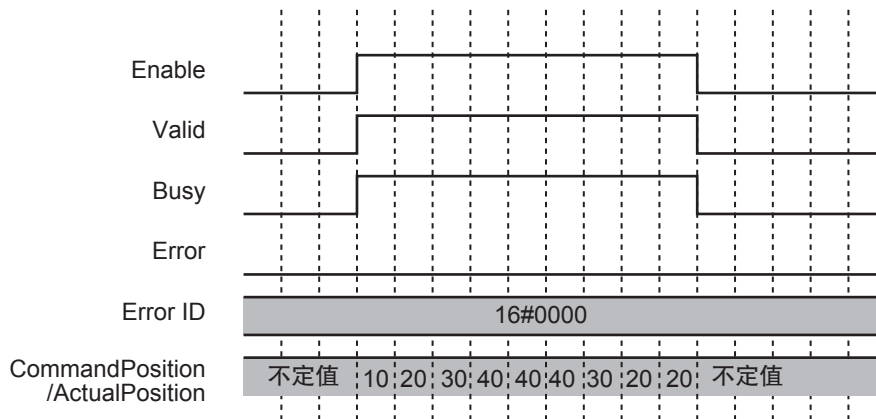
使用注意事项

在未分配轴组的任务中使用本指令时，请在使用的任务中将Axes(轴组构成轴)指定的构成轴对应的轴变量宣布为外部变量后再使用。

时序图

- 将Enable(有效)设为TRUE时，Busy(执行中)和Valid(有效)变为“TRUE”。
- 将Enable(有效)设为FALSE时，Busy(执行中)和Valid(有效)变为FALSE。

MC_GroupReadPosition



重启运动指令

输入为“Enable型”指令时，不会重启运动指令。

多重启动运动指令

本指令的各个指令独立动作，没有多重启动的限制。

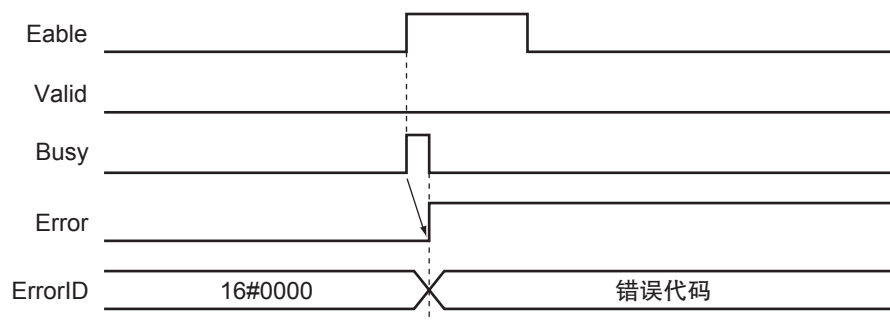
多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

异常

本指令执行中发生异常时，Error(错误)变为TRUE。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

MC_GroupReadPosition



● 错误代码

关于指令发生的异常，请参阅 □ “A-1 错误代码一览(P.A-2)”。

MC_ChangeAxesInGroup

暂时改写轴组参数的[构成轴]。

指令	名称	FB/ FUN	图形表现	ST表现
MC_ChangeAxesInGroup	写入轴组构成轴	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

通过本指令改写的值未保存至CPU单元的非易失性存储器中。因此，关闭控制器电源、以及开始下载、开始执行MC功能模块的重启处理时，改写值消失，恢复为通过Sysmac Studio设定的值。

保存到非易失性存储器时，请使用Sysmac Studio进行参数传送。



参考

“下载”使用Sysmac Studio的“同步”功能。



版本相关信息

Ver.1.01以上的CPU单元和Ver.1.02以上的Sysmac Studio组合时可使用本指令。


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	在上升沿开始指令。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

*1. 请参阅  “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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	其他指令发生异常、使本指令中止时	<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含有异常因素时	异常已解除时

输入输出变量

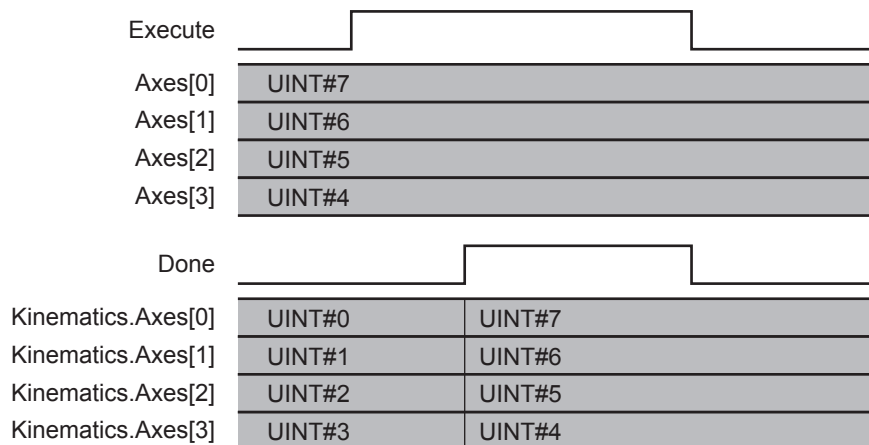
输入输出变量	名称	数据类型	有效范围	内容
AxesGroup	轴组	_sGROUP_REF	-	指定改写对象的轴组。*1
Axes	轴组构成轴	ARRAY[0.0.3] OF UINT	-	指定改写后的构成轴的轴号。*2

*1. 请使用在Sysmac Studio的轴组基本设定画面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. 构成轴请设定伺服轴或虚拟伺服轴。请使改写后的构成轴数与写入前的构成轴数一致。

功能说明

- 在Execute(启动)的上升沿, 将通过Axes(轴组构成轴)指定的构成轴写入通过AxesGroup(轴组)指定的轴组的轴组参数。
- 指令执行完成后, 轴组变量Kinematics(运动学转换设定)的Axis[]被改写为指定的构成轴。
- 该指令不改写轴构成(构成轴数)。例如, 对轴构成为3轴的轴组执行本指令后, 轴构成仍为3轴。
- 构成轴的轴号从“0, 1, 2, 3”变更为“7, 6, 5, 4”时的动作如下。



- 无论原点确定还是未确定, 均可执行本指令。
- 仅轴组无效时可执行本指令。



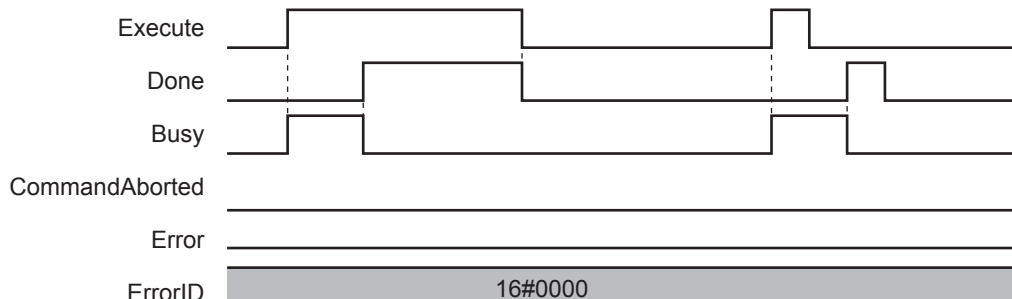
使用注意事项

- 通过本指令改写的值未保存至CPU单元的非易失性存储器中。因此, 关闭控制器电源、以及开始下载、开始执行MC功能模块的重启处理时, 改写值消失, 恢复为通过Sysmac Studio设定的值。
保存到非易失性存储器时, 请使用Sysmac Studio进行参数传送。
- 使用NX系列CPU单元时, 通过Axes(轴组构成轴)指定的所有构成轴需要分配至与执行指令的对象轴组相同的任务。指定分配至不同任务的轴时, 将发生“超过轴组构成轴设定范围(错误代码: 549D Hex)”。

时序图

- 将Execute(启动)设为TRUE时, Busy(执行中)变为TRUE。
- 构成轴改写完成时, Done(完成)变为TRUE。

MC_ChangeAxesInGroup指令



重启运动指令

- 无法重启本指令。
- 重启后, 将发生“无法重启运动指令(错误代码: 543B Hex)”错误。

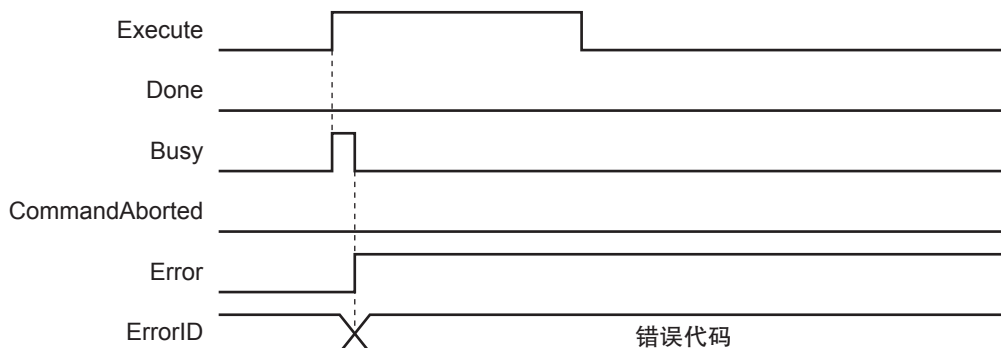
多重启动运动指令

- 多重启动运动指令的详情, 请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

异常

- 本指令执行中发生异常时, Error(错误)变为TRUE。
- 此时, 不进行参数写入, 保持执行指令前的值。
- 可查看ErrorID(错误代码)的输出值, 了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

- 关于指令发生的异常, 请参阅 □ “A-1 错误代码一览(P.A-2)”。

MC_GroupSyncMoveAbsolute

按周期输出各轴的指定目标位置。

指令	名称	FB/ FUN	图形表现	ST表现
MC_GroupSyncMove Absolute	轴组周期同步 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

Ver.1.01以上的CPU单元和Ver.1.02以上的Sysmac Studio组合时可使用本指令。

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	在上升沿开始指令。
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	缓存模式 选择	_eMC_BUFFER_ MODE	0: _mcAborting	0	指定多重启动运动指令时的动作。 0: 中断

*1. 关于指令单位，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 的 “单位转换设定”。

*2. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

输出变量

输出变量	名称	数据类型	有效范围	内容
InPosition	到位	BOOL	TRUE, FALSE	所有构成轴的反馈当前位置进入目标位置的到位宽度内时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Active	控制中	BOOL	TRUE, FALSE	控制中变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时, 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为正常。

* 请参阅 “A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
InPosition	所有构成轴的反馈当前位置进入目标位置的到位宽度内时	<ul style="list-style-type: none"> 反馈当前位置在到位宽度以外时 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	指令开始时	<ul style="list-style-type: none"> Error变为TRUE时 CommandAborted变为TRUE时
CommandAborted	<ul style="list-style-type: none"> 利用其它指令多重启动运动指令,中止本指令时 其他指令发生异常、使本指令中止时 轴异常发生过程中启动本指令时 执行MC_GroupStop指令中, 启动本指令时 	<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含有异常因素时	异常已解除时

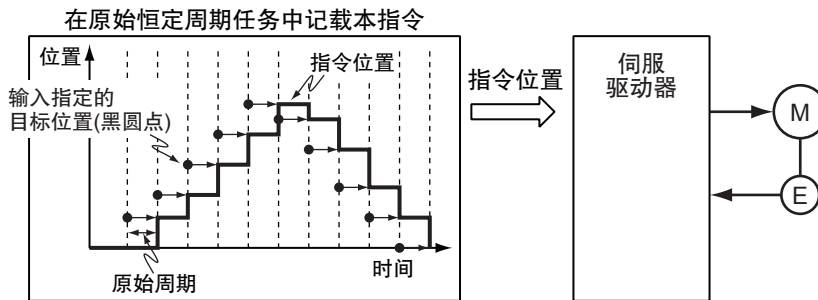
输入输出变量

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

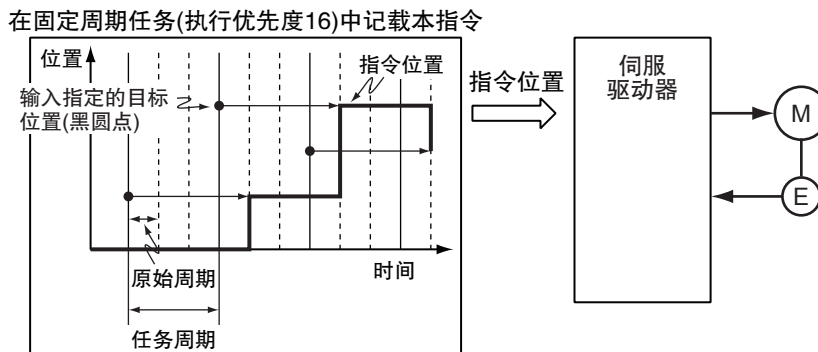
*1. 请使用在Sysmac Studio的轴组基本设定画面创建的用户定义变量的轴组变量名称(默认 “MC_Group***”)或系统定义变量的轴组变量名称(_MC_GRP[*]、_MC1_GRP[*]、_MC2_GRP[*])。

功能说明

- 本指令按照任务周期、以周期同步位置模式(CSP)将用户程序给定的目标位置输出到伺服驱动器等。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。最大加速度、最大减速度不适用。
- 原始恒定周期任务或固定周期任务(执行优先度5)中记载有本指令时，通过输入指定的目标位置在下一个任务周期输出到伺服驱动器。
原始恒定周期任务中记载有本指令的示例如下所示。固定周期任务(执行优先度5)时也一样。



- 固定周期任务(执行优先度16)中记载有本指令时，通过输入指定的目标位置在下一个固定周期任务时输出到伺服驱动器。



使用注意事项

- 轴组中的任意一个构成轴原点未确定时，将发生“原点未确定状态下的指令启动异常(异常代码：5466 Hex)”的错误。
- 请指定恰当的目标位置，使指定目标位置的移动量不超过轴参数的[最高速度]。若指定的目标位置超过了[最高速度]，会发生指令速度饱和，输出以[最高速度]限制的移动量。距离指定目标位置不足的部分的移动量在下一个周期以后输出。
此时，轴控制状态的Details.VelLimit(指令速度饱和)变为TRUE。

指令详情

下面对指令详细说明。

● 到位检查

不更新Position(目标位置)时, 如果目标位置和反馈位置的幅度在轴参数[到位宽度]的范围内, 则InPosition(到位)变为TRUE。

InPosition(到位)为TRUE时, 即使变更目标位置, 在该周期内仍保持TRUE不变, 到下一个周期时才变为FALSE。

轴参数[到位检查时间]的设定无效。

● 停止处理

下面对使轴组动作停止时的动作进行说明。

停止轴组动作使用MC_GroupStop(轴组强制停止)指令或MC_GroupImmediateStop(轴组立即停止)指令。启动这些指令后, 本指令的CommandAborted(执行中断)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

根据各轴的轴参数设定[立即停止输入停止方法]的指定进行立即停止。

● 异常发生时的停止

发生使轴停止动作的异常时, 无论如何设定, 均“立即停止”。

● 可以使用的轴

- 伺服轴可以使用本指令。
- 使用时, 请将MC_Power(可运行)指令的Enable(有效)设为TRUE(伺服ON状态)。
- 虚拟伺服轴可常时接收。

● 启动条件

- 请将轴参数[计数模式]设定为线性模式。
- 请将所有构成轴设为原点确定状态。
- 执行MC_GroupEnable(启用轴组)指令, 使轴组有效。

● 轴变量的状态

轴组变量的轴组状态Status.Moving(动作中)变为TRUE。

对轴组控制状态无影响。

● 超调

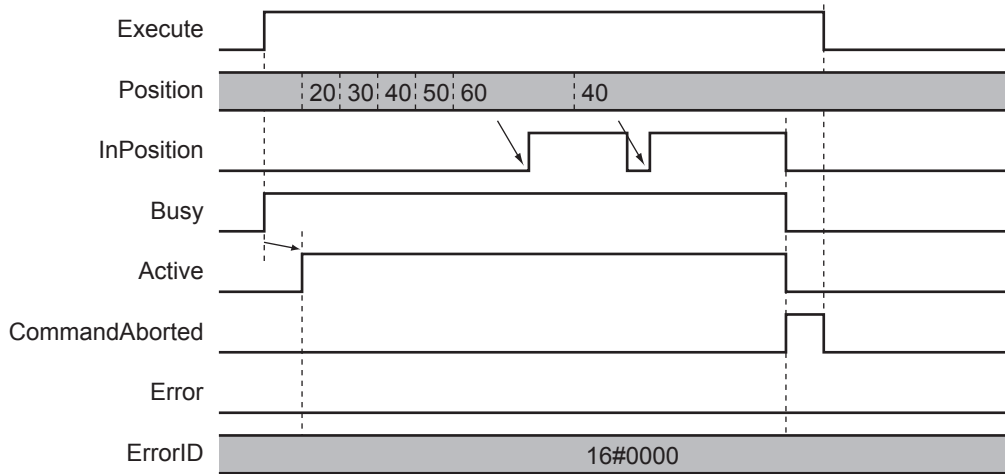
超调对本指令无效。

时序图

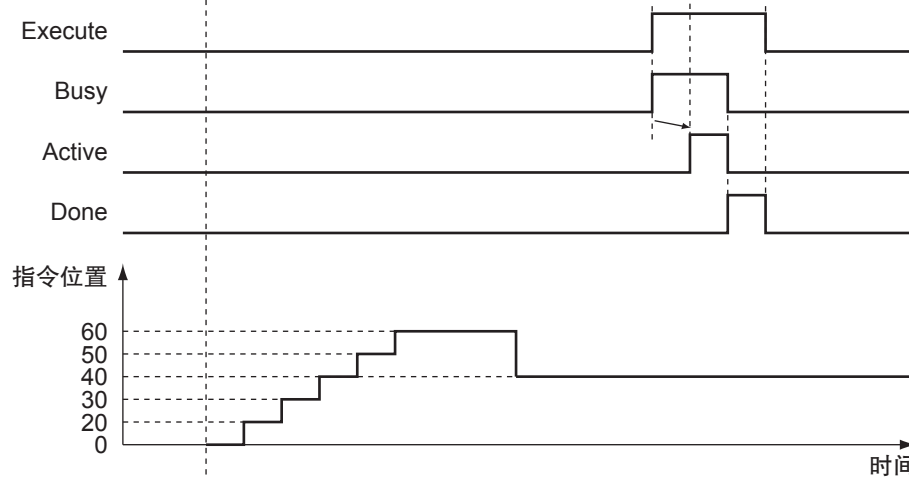
- 在启动Execute(启动)的同时，Busy(执行中)变为TRUE。在下一个周期Active(控制中)变为TRUE。
- 所有构成轴的反馈当前位置进入Position(目标位置)的到位宽度内时，InPosition(到位)变为TRUE。
- 利用其它指令中止本指令时，CommandAborted(执行中断)变为TRUE，Busy(执行中)、Active(控制中)、InPosition(到位)变为FALSE。
- 要终止本指令，请启动MC_GroupStop(轴组强制停止)指令。

原始恒定周期任务中记载有本指令时的情形如下所示。

MC_GroupSyncMoveAbsolute指令



MC_GroupStop指令



参考

MC功能模块使用上述时序图对伺服驱动器发出控制模式的指令。但是，反映到伺服驱动器的时序，则因各伺服驱动器的规格而异。

重启运动指令

无法重启本指令。

重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

- 其它指令执行中的本指令启动

本指令的BufferMode(缓存模式选择)只支持中断模式。

- 本指令执行中的其它指令启动

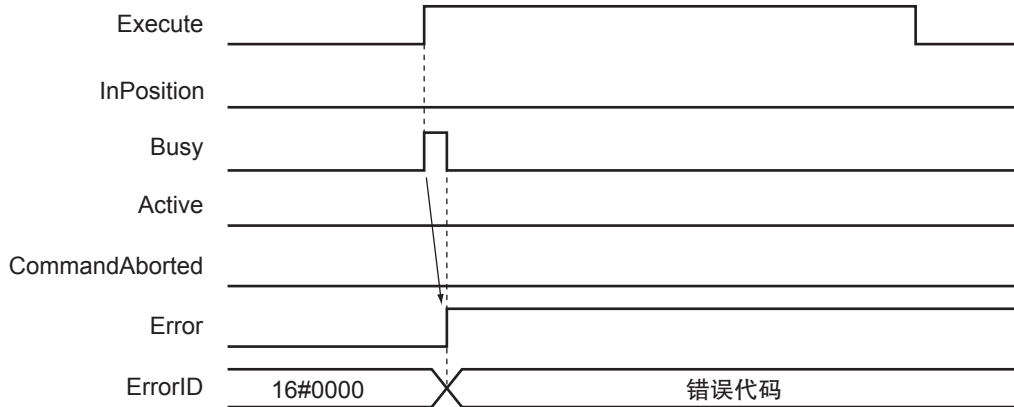
对本指令进行多重启动运动指令时，只能使用中断模式。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，轴停止动作。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

- 发生异常时的时序图



- 错误代码

关于指令发生的异常，请参阅 □ “A-1 错误代码一览(P.A-2)”。

MC_GroupReset

解除轴组及轴的异常。

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>

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	在上升沿开始指令。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
Failure	非法结束	BOOL	TRUE, FALSE	指令未正常执行时变为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为正常。

* □ 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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、轴组因异常减速停止中执行指令时 因发生轴公共异常而导致轴组发生异常的过程中执行指令时 	<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含有异常因素时	异常已解除时

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. 请使用在Sysmac Studio的轴组基本设定画面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，对GroupEnable(启用轴组)状态的AxesGroup(轴组)指定的轴组的异常及轴组所属轴的异常进行解除处理。
可进行解除的异常有：轴及轴组发生的“轻度故障”、“监视信息”的异常、以及驱动器错误复位。
- 无论轴是伺服ON状态还是伺服OFF状态，均可执行异常解除处理。
- 对于发生驱动器错误的轴，应先执行驱动器错误复位处理，然后再执行异常解除处理。
- 驱动器错误复位处理可选择清除驱动器错误，还是在轴参数[驱动器错误复位监视时间]内保持不变。驱动器错误复位对属于轴组的所有轴同时进行。
- 可解除的异常对象为Execute(启动)上升沿时发生的异常。不能对在异常解除过程中发生的异常执行异常解除。
- 若在轴组的错误减速停止中执行指令，则该指令无法执行，Failure(非法结束)变为TRUE。这是因为在轴停止之前无法进行异常解除。
此外，发生轴公共异常的轴也无法通过本指令解除异常，Failure(非法结束)会变为TRUE。



使用注意事项

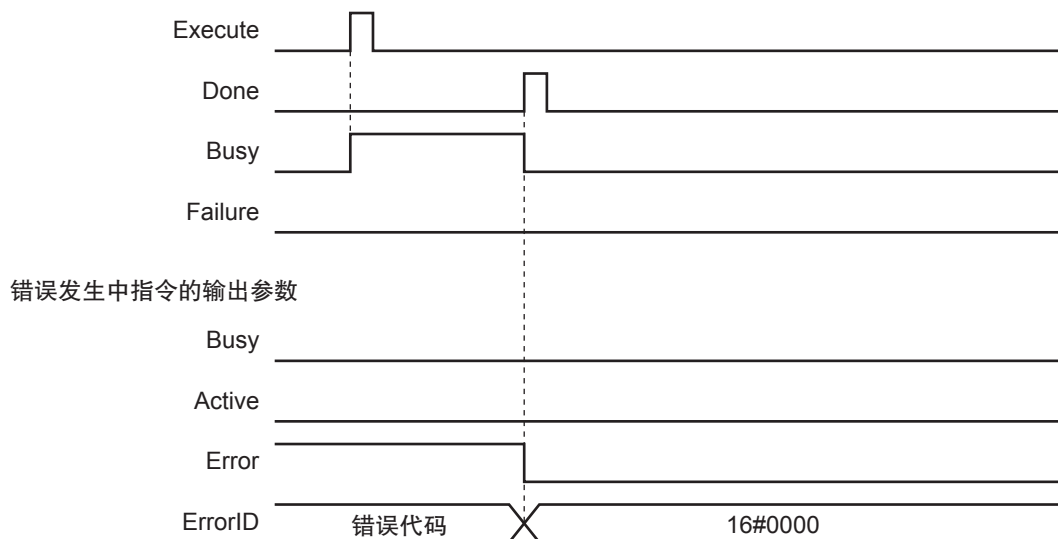
- 通过本指令进行的异常解除处理有时会跨越多个控制周期。
- 启动本指令时如果轴仍在动作中，本指令的输出变量“Failure(非法结束)”会变为TRUE。在排除异常原因之后，请执行重试处理，直到Done(完成)变为TRUE。
- 排除异常发生原因时，请预先确认各轴已完全停止。
- 对欧姆龙制伺服驱动器G5系列使用本指令时，请进行指令的排他性处理，以免同时执行ResetEError(EtherCAT异常解除)指令。
- 对NX系列脉冲输出单元使用本指令时，脉冲输出单元连接的驱动器的异常状态无法解除。详情请参阅 “NX系列 位置接口单元 用户手册(SBCE-374)”。



参考

- 轴异常只有在轴组有效时可以解除。
- 本指令不能解除以下异常。
所有轴公共异常
要解除异常，请另外执行ResetMError(所有错误复位)指令。
- 若对未发生异常的轴组执行本指令，指令将正常结束，保持原来状态不变。

时序图



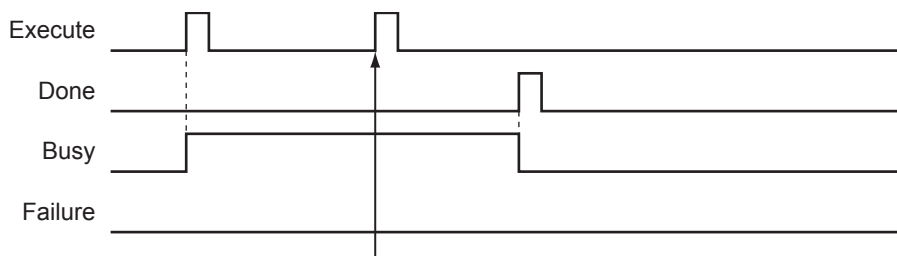
指令的中止

在轴、轴组因异常而减速停止时、或者因发生轴公共异常而导致无法解除正在发生的轴组异常时，请中断指令。



重启运动指令

若在本指令执行中再次执行Execute(启动)，将忽视重启的指令，继续进行异常解除处理。



不执行通过重启发出的指令，
继续进行执行中的处理

多重启动运动指令

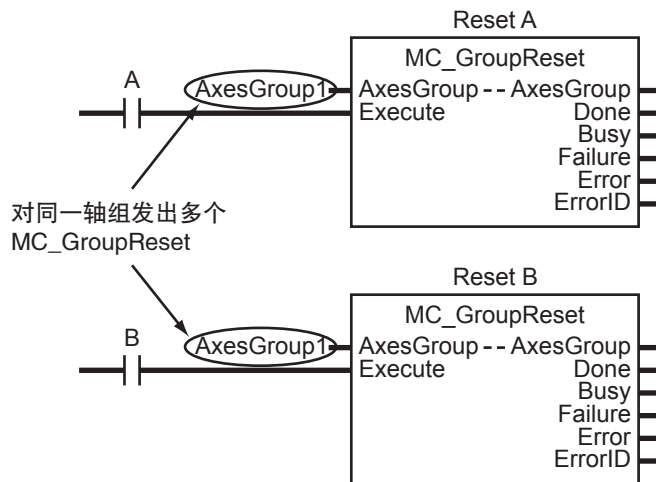
多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 本指令执行中的其它指令启动

对同一轴组同时执行其它实例的MC_GroupReset(轴组错误复位)指令时，分别执行两个指令。

发生从站系统异常时，会等待到轴设定的[驱动器错误复位监视时间]。经过时间也按每个指令实例进行计数。

若在执行本指令的过程中对指定轴组下属的轴执行了MC_Reset(轴错误复位)指令，也同样分别执行两个指令。



错误代码

关于指令发生的异常，请参阅 □ “A-1 错误代码一览(P.A-2)”。

5

通用指令

本章对轴及轴组的通用指令进行说明。

MC_SetCamTableProperty	5-2
MC_SaveCamTable	5-8
MC_Write	5-13
MC_GenerateCamTable	5-18
MC_WriteAxisParameter	5-45
MC_ReadAxisParameter	5-57

MC_SetCamTableProperty

更新通过输入输出参数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	在上升沿开始指令。

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	指令执行完毕时变为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	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

* 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	凸轮表改写完成时	<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含有异常因素时	异常已解除时

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型的数组变量作为凸轮表指定。 ^{*1}

*1. 数组元素[N]通过Sysmac Studio自动设定。请指定通过Sysmac Studio的凸轮编辑器创建的凸轮数据变量。

功能说明

- 本指令用于更新通过输入输出参数指定的凸轮表的终点索引。
- 所谓终点，是指从起点开始按顺序搜索凸轮表时，排列在相位最初为“0”的凸轮数据前一位的数据。
- 终点的数组编号输出到EndPointIndex(终点索引)。
- 检测到相位“0”时，此后的数据均作为无效凸轮数据处理，其相位/位移值均被忽略。
- 所谓最大凸轮数据数，是指凸轮数据结构体数组的最大元素数。
最大凸轮数据数是通过Sysmac Studio在公布结构体数组时指定的值。
- 若从用户程序变更了凸轮数据的终点索引，需要进行终点更新。这种情况下，请使用本指令进行有效凸轮数据数的更新。

关于凸轮表，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。



使用注意事项

- 搜索凸轮表时，若在发现相位“0”之前检测到相位不是以升序排列，则会发生异常。
- 最大凸轮数据数不能通过用户程序变更。
- 请在变更凸轮数据的终点、或者改写凸轮数据值之后执行本指令。
变更终点索引、或者相位并非升序时，凸轮动作及MC_CamIn(凸轮动作开始)指令的EndOfProfile(凸轮周期完成)将无法正确动作。
- 若在本指令处理过程中变更了凸轮表数据，将无法正确进行更新处理。通过程序变更凸轮数据时，请在本指令完成后再进行变更。
- 凸轮数据变量为全局变量，可以从多个任务中查看或变更凸轮数据变量值。从多个任务中变更凸轮数据变量值时，请编写通过多个任务进行改写处理时相互不会发生冲突的程序。
- 使用“全局变量的任务间排他性控制”功能进行凸轮数据变量的排他性控制时，请不要使用排他对象的凸轮数据变量启动运动指令。否则会发生轻度故障等级的异常“凸轮表指定错误(错误代码：5439 Hex)”。

指令详情

例如，下图所示的凸轮表，EndPointIndex(终点索引)为“999”，MaxDataNumber(最大凸轮数据数)为“5000”。

凸轮数据结构体数组	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

以下是凸轮数据改写和EndPointIndex(终点索引)之间的关系。

若通过MyCam1[1000]以后的相位为“0”的凸轮表启动本指令，EndPointIndex(终点索引)变为“999”。
若将MyCam1[997]的相位改写为“0”后启动本指令，EndPointIndex(终点索引)变为“996”。

凸轮数据结构体数组	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
MyCam1 [997]	359.8	2	0	0
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

图中，MyCam1 [999] 的相位 360.0 和 MyCam1 [996] 的相位 360.0 均被圈出并标注为“终点”。MyCam1 [1000] 至 MyCam1 [4999] 的相位均为 0，位移均为 0，被标注为“无效数据”。

若将从MyCam1[1000]到MyCam1[4997]的相位改写为非“0”后启动本指令，EndPointIndex(终点索引)变为“4997”。

凸轮数据结构体数组	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
MyCam1 [1000]	0	0	100.5	20.5
.
.
.
MyCam1 [4996]	0	0	359.99	0.01
MyCam1 [4997]	0	0	360.00	0.0
MyCam1 [4998]	0	0	0	0
MyCam1 [4999]	0	0	0	0

图中，MyCam1 [999] 的相位 360.0 和 MyCam1 [4997] 的相位 360.00 均被圈出并标注为“终点”。MyCam1 [1000] 至 MyCam1 [4999] 的相位均为 0，位移均为 0，被标注为“无效数据”。

即使将从 MyCam1[1000] 到 MyCam1[4997] 的相位改写为非“0”数值，不启动本指令时，凸轮动作将在 MyCam1[0]到MyCam1[999]间执行。

凸轮表虽然被改写，但EndPointIndex(终点索引)保持不变。

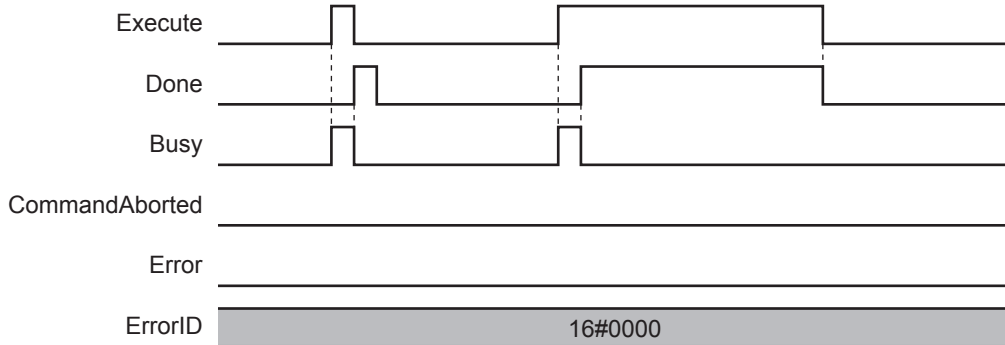
关于凸轮表数据结构的详情，请参阅 □□ “MC_CamIn(P.3-161)”。

时序图

本指令的启动方法有两种。作为动作，无论哪种启动方法均可正常结束。

第一次启动，将Execute(启动)设为TRUE，结束前将Execute(启动)设为False。

第二次启动，保持Execute(启动)不变。



重启运动指令

无法重启本指令。

重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

● 其它指令执行中的本指令启动

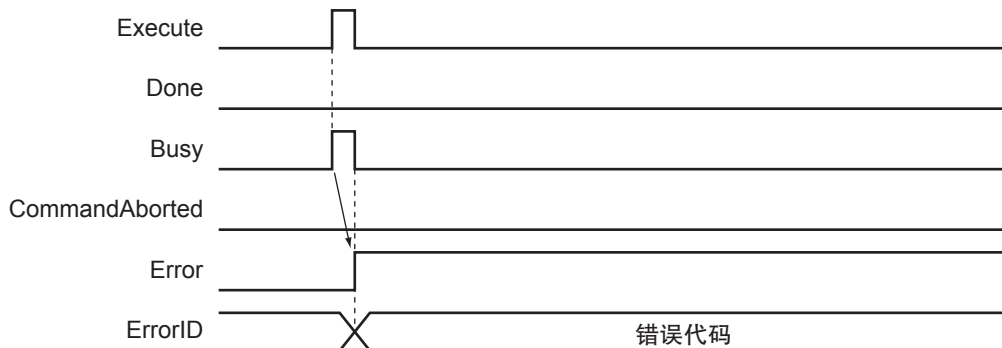
与本指令使用相同的CamTable(凸轮表)的MC_SaveCamTable(保存凸轮表)指令、MC_SetCamTableProperty(更新凸轮表属性)指令、以及MC_GenerateCamTable(生成凸轮表)指令已启动时，本指令不能多重启动。

● 本指令执行中的其它指令启动

本指令执行中，与本指令使用相同的CamTable(凸轮表)的MC_SaveCamTable(保存凸轮表)指令、MC_SetCamTableProperty(更新凸轮表属性)指令、以及MC_GenerateCamTable(生成凸轮表)指令无法多重启动。

异常

本指令执行中发生异常时，Error(错误)变为TRUE。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。



● 错误代码

关于本指令发生的异常，请参阅 “A-1 错误代码一览(P.A-2)”。

MC_SaveCamTable

将通过输入参数指定的凸轮表保存到非易失性存储器。

指令	名称	FB/ FUN	图形表现	ST表现
MC_SaveCamTable	保存凸轮表	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	在上升沿开始指令。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

* 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	凸轮表保存完成时	<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含有异常因素时	异常已解除时

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型的数组变量作为凸轮表指定。 ^{*1}

*1. 数组元素[N]通过Sysmac Studio自动设定。请指定通过Sysmac Studio的凸轮编辑器创建的凸轮数据变量。

功能说明

- 本指令将通过输入输出变量指定的凸轮表保存到非易失性存储器。
- 进行凸轮表保存处理时，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)变为TRUE。

关于凸轮表的数据规格，请参阅  “MC_CamIn(P.3-161)”。



使用注意事项

- 执行凸轮数据改写后，请在控制器电源“OFF”前通过本指令进行保存。若不保存就关闭控制器电源，改写的数据会丢失。
- 若在本指令处理过程中变更了凸轮表数据，将无法正确进行更新处理。通过程序变更凸轮表时，请不要在本指令处理过程中向凸轮表写入。
- 本指令的处理时间比其他指令明显要长。此外，处理时间还因CPU单元的运算负载而大为不同。因此，将本指令完成作为触发启动下一个指令时，应注意下一个指令的启动时间。
- 在进行本指令的处理时，请勿切断控制器的电源。否则，数据将无法正常工作保存。并且，还可能损坏保存在非易失性存储器中的凸轮表。
- 执行本指令的处理时，不能进行凸轮数据的上传、下载以及在线开始操作、在线编辑、数据跟踪开始等操作。

此外，如果在进行凸轮数据的上传、下载以及在线编辑等操作时执行本指令，将发生无法执行凸轮表保存指令(错误代码: 743CHex)的错误。此时请进行重试，直到能正常工作保存凸轮数据。

“下载”及“上传”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缩短，因此请尽量少使用本指令。
- 凸轮数据变量为全局变量，可以从多个任务中查看或变更凸轮数据变量值。从多个任务中变更凸轮数据变量值时，请编写通过多个任务进行改写处理时相互不会发生冲突的程序。
- 使用“全局变量的任务间排他性控制”功能进行凸轮数据变量的排他性控制时，请不要使用排他对象的凸轮数据变量启动运动指令。否则会发生轻度故障等级的异常“凸轮表指定错误(错误代码: 5439 Hex)”。
- 在线编辑反映中(*)请勿执行本指令。否则，在线编辑所进行的变更可能无法正确反映。

* 从在反映确认窗口中选择[是]、到反映确认窗口关闭的期间。如果是Ver.1.04以上的CPU单元和Ver.1.05以上的Sysmac Studio组合，则是指到内置非易失性存储器备份中窗口关闭的期间。

● 与CPU单元动作模式的关系

通过本指令保存凸轮数据时，在将CPU单元的动作模式切换为程序模式后，仍然继续执行凸轮数据的保存处理。

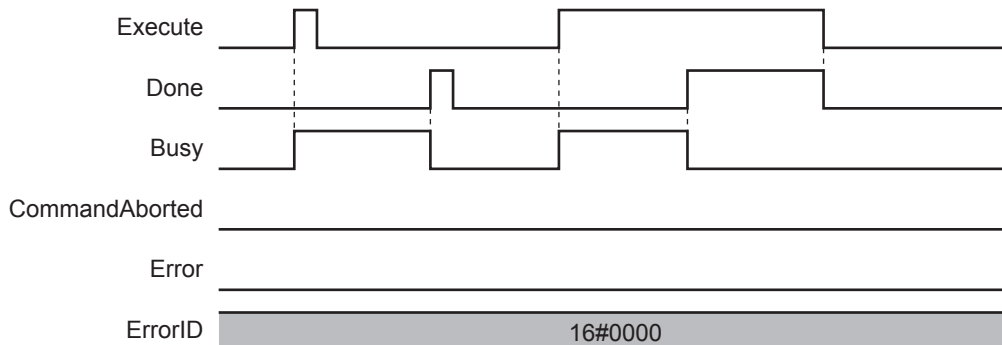
要确认程序模式中是否正在执行凸轮数据保存处理，请将Sysmac Studio设为在线，并监视运动控制系统变量 `_MC_COM.Status.CamTableBusy`。

● 利用在线编辑删除指令时

即使在通过本指令保存凸轮数据时通过在线编辑删除了本指令，仍会继续执行凸轮数据的保存处理。

时序图

本指令的启动方法有两种。作为动作，无论哪种启动方法均可正常结束。
第一次启动，将Execute(启动)设为TRUE，结束前将Execute(启动)设为False。
第二次启动，保持Execute(启动)不变。



重启运动指令

无法重启本指令。
重启后，将发生“无法重启运动指令(错误代码：543B Hex)”错误。

多重启动运动指令

多重启动运动指令的详情，请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● 其它指令执行中的本指令启动

与本指令使用相同的CamTable(凸轮表)的MC_SetCamTableProperty(更新凸轮表属性)指令已启动时，本指令不能多重启动。
此外，正在执行MC_SaveCamTable(保存凸轮表)时，本指令不能多重启动。

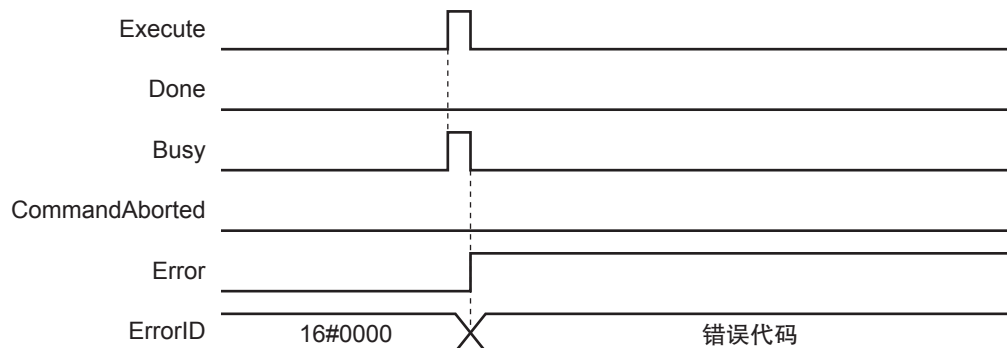
● 本指令执行中的其它指令启动

本指令执行中，与本指令使用相同CamTable(凸轮表)的其他指令的MC_SetCamTableProperty(更新凸轮表属性)指令无法多重启动。
此外，本指令执行中也无法进行MC_SaveCamTable(保存凸轮表)指令的多重启动。

异常

本指令执行中发生异常时，Error(错误)变为TRUE。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。



● 错误代码

关于本指令发生的异常，请参阅 □ “A-1 错误代码一览(P.A-2)”。

MC_Write

改写MC功能模块的一部分运动控制参数设定。

指令	名称	FB/ FUN	图形表现	ST表现
MC_Write	写入MC设定	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

通过本指令改写的值未保存至CPU单元的非易失性存储器中。因此，关闭控制器电源、以及开始下载、开始执行MC功能模块的重启处理时，改写值消失，恢复为通过Sysmac Studio设定的值。

保存到非易失性存储器时，请使用Sysmac Studio进行参数传送。



参考

“下载”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	在上升沿开始指令。
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*1 12:_mcStartVel*2	0*3	指定写入目标位置的参数。 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: 正方向转矩警告值 4: 负方向转矩警告值 5: 位置偏差超过值 6: 位置偏差警告值 7: 软件限制功能 8: 正方向软件限制 9: 负方向软件限制 10: 到位检查时间 11: 到位宽度 12: 启动速度

*1. Ver.1.01以上的CPU单元和Ver.1.02以上的Sysmac Studio组合时可指定。

*2. Ver.1.10以上的CPU单元和Ver.1.12以上的Sysmac Studio组合时可指定。

*3. 有效范围为枚举体的变量，其实际初始值不是数值，而是枚举元素。

● 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 : 正方向转矩警告值	UINT	0 ~ 1000	单位为[%]。
4 : 负方向转矩警告值	UINT	0 ~ 1000	单位为[%]。
5 : 位置偏差超过值	LREAL	正数	单位为[指令单位]。*1
6 : 位置偏差警告值	LREAL	正数	应小于位置偏差超过值。 单位为[指令单位]。*1
7 : 软件限制功能	 _eMC_ SWLMT_ MODE	0:_mcNonSwLmt 1:_mcCmdDecelerationStop 2:_mcCmdImmediateStop 3:_mcActDecelerationStop 4:_mcActImmediateStop	0: 软件限制功能无效 1: 对指令位置有效, 执行减速停止 2: 对指令位置执行有效, 执行立即停止(累积脉冲停止) 3: 对反馈位置有效, 执行减速停止 4: 对反馈位置有效, 执行立即停止(累积脉冲停止)
8 : 正方向软件限制	LREAL	负数、正数、“0”	单位为[指令单位]。*1
9 : 负方向软件限制	LREAL	负数、正数、“0”	单位为[指令单位]。*1
10 : 到位检查时间	UINT	0 ~ 10000	单位为[ms]。
11 : 到位宽度	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. 关于指令单位, 请参阅 □□ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)” 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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时, 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为正常。

* □□ 请参阅 “A-1 错误代码一览(P.A-2)”。


● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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	其他指令发生异常、使本指令中止时	<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含有异常因素时	异常已解除时

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者轴组。 ^{*1}
SettingValue	设定值	视指定的变量 类型而定 ^{*2}	-	指定写入的值。 有效范围根据通过ParameterNumber(参数编号)指定的运动控制参数而定。初始值为“0”。

*1. 请使用在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***”)以及在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***”)。均可使用系统定义变量的变量名称(_MC_AX[*], _MC1_AX[*], _MC2_AX[*], _MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])。

*2. 关于变量的数据类型，请参阅  “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间

输入输出变量	写入时间
SettingValue	Done的上升沿时

功能说明

- 在Execute(启动)的上升沿, 向通过Target(写入对象)以及ParameterNumber(参数编号)指定的运动控制系统变量写入SettingValue(设定值)。
- 通过输入变量指定的参数适用于轴变量Status.Standstill(停止中)变为TRUE时、或轴组变量Status.Standby(停止中)为TRUE时的动作开始时、以及通过中断指定进行多重启动时。
因此, 通过重启以及中断指定以外的多重启动继续动作时, 不适用上述参数。



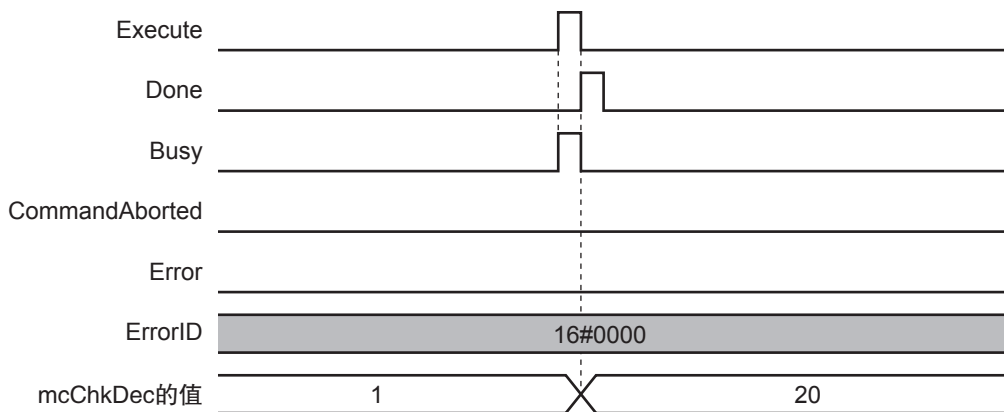
使用注意事项

通过本指令改写的值未保存至CPU单元的非易失性存储器中。因此, 关闭控制器电源、以及开始下载、开始执行MC功能模块的重启处理时, 改写值消失, 恢复为通过Sysmac Studio设定的值。

保存到非易失性存储器时, 请使用Sysmac Studio进行参数传送。

时序图

以下是向轴参数设定的_mcChkDec(减速度警告值)中写入数据“20”时的时序图。



重启运动指令

本指令的Busy(执行中)为TRUE时, 若有相同实例的Execute(启动)的上升沿, 则重启指令。此时, 改写为此后的Execute(启动)的上升沿时的Target(写入对象)及ParameterNumber(参数编号)、SettingValue(设定值)。

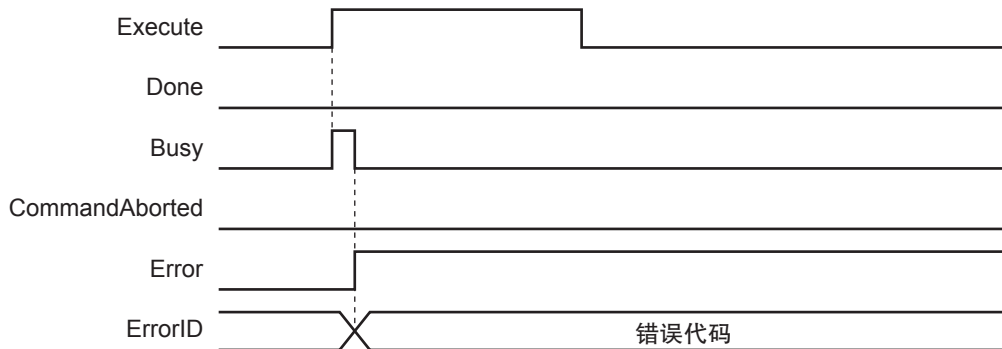
多重启动运动指令

多重启动运动指令的详情, 请参阅 [□□ “NJ/NX系列 CPU单元 用户手册 运动控制篇\(SBCE-363\)”](#) 或 [□□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇\(SBCE-379\)”](#)。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，不执行参数的写入。保持指令执行前的值。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

● 发生异常时的时序图



● 错误代码

关于指令发生的异常，请参阅 [□□](#) “A-1 错误代码一览(P.A-2)”。

MC_GenerateCamTable

基于通过输入输出参数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

由本指令指定的凸轮表，必须事先使用Sysmac Studio的凸轮编辑器编制，并下载到CPU单元中。“下载”使用Sysmac Studio的“同步”功能。



版本相关信息

Ver.1.08以上的CPU单元和Ver.1.09以上的Sysmac Studio组合时可使用本指令。

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	在上升沿开始指令。

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	指令执行完毕时变为TRUE。
EndPointIndex	终点索引	UINT	正数或“0”	在指令执行完成时，输出凸轮表的终点索引。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

输出变量	名称	数据类型	有效范围	内容
ErrorParameterCode	参数详情代码	WORD	*	有些错误代码存在附属信息。如果存在该信息，则输出异常参数的详情代码。
ErrorNodePointIndex	节点元素编号	UINT	*	有些错误代码存在附属信息。如果存在该信息，则输出异常节点的元素编号。

* 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
Done	凸轮表生成完成时	<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含有异常因素时	异常已解除时

输入输出变量

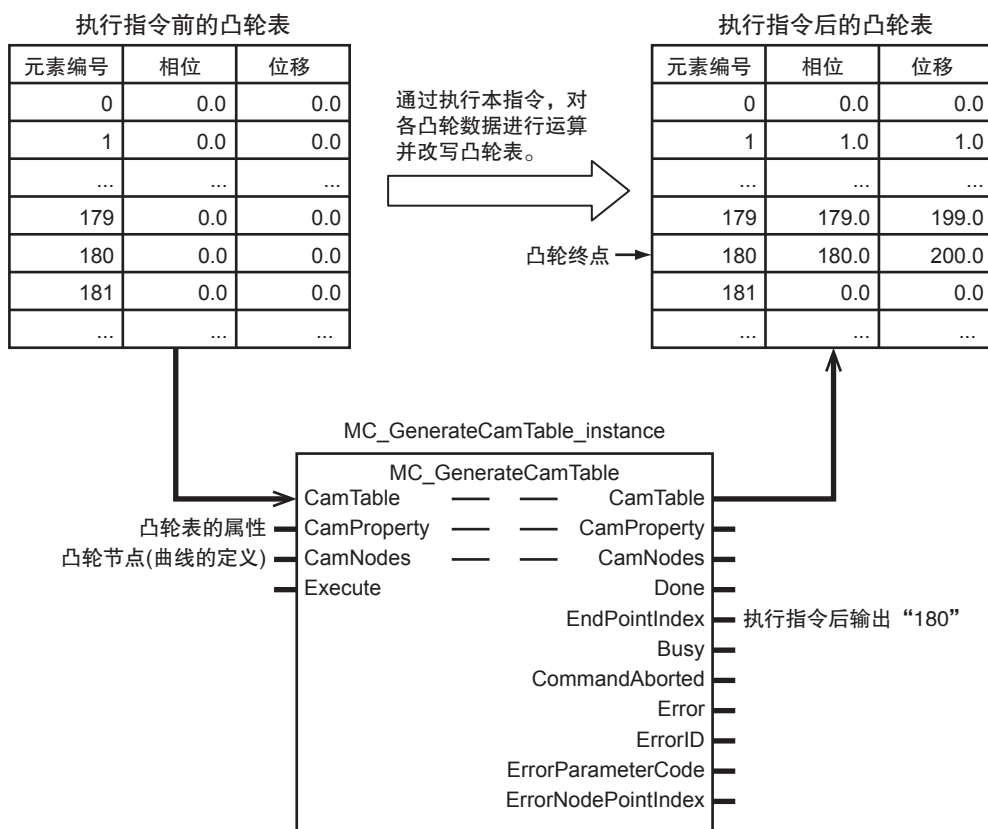
输入输出变量	名称	数据类型	有效范围	内容
CamTable	凸轮表	ARRAY[0..N] OF _sMC_CAM_REF	-	将凸轮数据结构体_sMC_CAM_REF型的数组变量作为凸轮表指定。 ^{*1} 请指定通过Sysmac studio的凸轮编辑器创建的凸轮数据变量。
CamProperty	凸轮属性	_sMC_CAM_ PROPERTY	-	指定凸轮属性结构体_sMC_CAM_PROPERTY型的变量。 请指定_sMC_CAM_PROPERTY型的用户定义变量，或通过Sysmac studio创建的凸轮属性变量。
CamNodes	凸轮节点	ARRAY[0..N] OF _sMC_CAM_NODE	-	指定凸轮节点结构体_sMC_CAM_NODE型的数组变量。 请指定sMC_CAM_NODE型的用户定义变量 ^{*2} ，或通过Sysmac studio创建的凸轮节点变量。

*1. 数组元素[N]通过Sysmac Studio自动设定。

*2. 创建用户定义变量时，请确保数组变量的元素开始编号为0，数组元素[N]为358以下。

功能说明

- 本指令在Execute(启动)的上升沿，基于输入输出变量中指定的CamProperty(凸轮属性)和CamNodes(凸轮节点)的值运算凸轮数据。
- 将输入输出变量中指定的CamTable(凸轮表)改写为运算出的凸轮数据的值。
- 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。
- 完成凸轮表的改写后更新凸轮表的终点索引，将凸轮终点的元素编号输出到EndPointIndex(终点索引)。本指令结束后，无需执行MC_SetCamTableProperty(凸轮表属性更新)指令。
- 进行凸轮表生成处理时，运动控制系统变量的MC通用变量中的“凸轮表生成执行中(_MC_COM.Status.GenerateCamBusy)”变为TRUE。





使用注意事项

- 指定的凸轮表的最大凸轮数据数请确保在本指令生成的凸轮表的凸轮数据数以上。
- Sysmac Studio的凸轮编辑器中可以指定“自由曲线”作为曲线形状，但本指令中则无法指定“自由曲线”。
- 本指令的处理时间比其他指令明显要长。此外，处理时间还因CPU单元的运算负载而大为不同。因此，将本指令完成作为触发启动下一个指令时，应注意下一个指令的启动时间。
- 即使Sysmac Studio的凸轮编辑器的设定值与本指令的设定值相同，由于内部的运算处理不同，也可能会导致生成的凸轮数据值不同。
- 凸轮数据变量为全局变量，可以从多个任务中查看或变更凸轮数据变量值。从多个任务中变更凸轮数据变量值时，请编写通过多个任务进行改写处理时相互不会发生冲突的程序。
- 使用“全局变量的任务间排他性控制”功能进行凸轮数据变量的排他性控制时，请不要使用排他对象的凸轮数据变量启动运动指令。否则会发生轻度故障等级的异常“凸轮表指定错误(错误代码：5439 Hex)”。
- 将本指令的输入输出变量CamProperty(凸轮属性)和CamNodes(凸轮节点)中指定的变量作为用户定义变量创建时，变量保存属性的初始值为“不保存”。若要在变更变量值，切换至程序模式或重新接通电源后再次使用，请将变量的保存属性设定为“保存”。通过Sysmac studio创建的凸轮属性变量和凸轮节点变量的保存属性固定为“保存”。
- 通过本指令生成的凸轮数据变量未保存至CPU单元的非易失性存储器中。要保存至非易失性存储器中时，请执行MC_SaveCamTable(保存凸轮表)指令。
- 执行本指令的过程中，请勿变更CamNodes(凸轮节点)中指定的数组变量的值。否则本指令可能引起意外动作。
- 若CamNodes(凸轮节点)的数组大小过大，则指令执行时间可能会变长，超过任务周期。届时，会发生任务周期超限错误(错误代码：6001 Hex)，请调整数组大小或变更任务周期。
- 执行本指令的过程中，在从运行模式切换为程序模式的情况下，仍然继续执行凸轮表的生成处理。如果在该状态下再次切换至运行模式，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“无法多重启动运动指令(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来确认，或通过用户程序执行SizeOfAry(获取数组元素数)指令来确认。
- 本指令生成的凸轮表的凸轮数据数可通过下式计算。

$$T_{cd} = \sum_{k=0}^{n-1} \frac{M_k - M_{k-1}}{P_k} + 1$$

T_{cd} : 凸轮表的凸轮数据数

k : 凸轮节点变量的数组元素编号

M_k : 凸轮节点变量的数组元素编号k的Phase(主轴相位)
(但是, M_{-1} 的Phase(主轴相位)为“0”)

P_k : 凸轮节点变量的数组元素编号k的PhasePitch(相位间隔宽度)

n : 节点数

注) Phase(主轴相位)/PhasePitch(相位间隔宽度)除不尽时, 请将小数点后第1位进位。

- 本指令生成的凸轮表的凸轮数据数请控制在65535以下。
- 可通过EndPointIndex(终点索引)的值确认生成的凸轮表的凸轮数据数。可以在显示器等中显示EndPointIndex(终点索引)的值, 用于改善凸轮表的分辨率和新增节点等的调整。
- 关于凸轮表的凸轮数据的详情, 请参阅MC_CamIn(凸轮动作开始)指令。
- 关于终点索引的详情, 请参阅MC_SetCamTableProperty(更新凸轮表属性)指令。
- 关于CamProperty(凸轮属性)和CamNodes(凸轮节点)的各设定项目, 请参见“Sysmac Studio Version 1 操作手册(SBCA-362K以上)”。
- 关于曲线的形状和连接速度、连接加速度的关系, 请参阅“Sysmac Studio Version 1 操作手册(SBCA-362K以上)”。

● 与CPU单元动作模式的关系

通过本指令进行凸轮表的生成处理时, 在将CPU单元的动作模式切换为程序模式后, 仍然继续执行凸轮表的生成处理。

● 利用在线编辑删除指令时

即使在执行本指令时通过在线编辑删除了本指令, 仍会继续执行凸轮表的生成处理。

● 执行MC试运行时

通过本指令进行凸轮表的生成处理时执行了MC试运行的情况下, 本指令的输出变量“CommandAborted(执行中断)”变为TRUE, 但仍然继续执行凸轮表的生成处理。

指令详情

下面对指令详细说明。

● 凸轮属性结构体(_sMC_CAM_PROPERTY型)

凸轮属性结构体(_sMC_CAM_PROPERTY型)是用于指定属性的结构体数据类型。各要素相当于通过 Sysmac Studio的凸轮编辑器设定的“凸轮表属性”的项目的一部分。

在本指令中,作为输入输出变量CamProperty(凸轮属性)中指定的变量使用。

下面对各凸轮属性结构体的要素进行说明。

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”	指定起始节点(相位=“0”、位移=“0”)的动作开始时的速度。 起始节点的下一个凸轮节点指定的曲线形状为3次曲线或5次曲线时,初始速度生效。 单位为[指令单位/s]。
InitAcc	初始加速度	REAL	负数*1、正数*1、“0”	指定起始节点(相位=“0”、位移=“0”)的动作开始时的加速度。 起始节点的下一个凸轮节点指定的曲线形状为5次曲线时,初始加速度生效。 单位为[指令单位/s ²]。
CycleTime	周期时间	REAL	正数*1	指定凸轮动作的1周期的时间。 单位为[s]。

*1. 指定时请确保绝对值在0.001以上。小数点后第4位四舍五入。

● 凸轮节点结构体(_sMC_CAM_NODE型)

凸轮节点结构体(_sMC_CAM_NODE型)是用于指定曲线定义的结构体数据类型。各要素相当于通过 Sysmac Studio的凸轮编辑器设定的“凸轮节点”的项目。

	主軸	從軸	力△曲線	接統速度	接統加速度	位相間隔幅
起始节点	0.000	0.000				
Cam-	1.000	1.000	5次曲線	<input checked="" type="checkbox"/>	100.000	<input checked="" type="checkbox"/> 10.000
Cam-	2.000	2.000	3次曲線	<input checked="" type="checkbox"/>	100.000	<input type="checkbox"/> 1176.000

} 凸轮节点

在本指令中,作为输入输出变量CamNodes(凸轮节点)中指定的数组变量使用。

凸轮节点的数组变量元素称为“节点”。数组变量的元素数请确保为所设定的节点数以上。

凸轮曲线的起始点(相位=“0”、位移=“0”)的节点称为“起始节点”。在数组变量中,按元素编号顺序指定除起始节点外的节点。

如果在Phase(主轴相位)中指定0.001以上的正数,则为有效节点。指定为“0”时,之后的节点无效。但是,如果对元素编号为“0”的Phase(主轴相位)指定“0”,则会发生异常。

有效节点数为5、数组变量的元素数为10时的示例如下所示。

元素编号	Phase(主轴相位)
0	正数
...	...
4	正数
5	0
6	任意
...	...
9	任意

有效节点 { 0, ..., 4 }
无效节点 { 5, 6, ..., 9 }

指定时请确保Phase(主轴相位)的值按数组编号的升序升序排列。(上次值 < 本次值)

下面对各凸轮节点结构体的要素进行说明。

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”	指定节点的主轴相位的值。 单位为[指令单位]。
Distance	从轴位移	REAL	负数 ^{*1} *2、正数 ^{*1} *2、“0”	指定节点的从轴位移的值。 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: _mcReverseDouble Harmonic 12: _mcNC2Curve 13: _mcPolynomic3 14: _mcPolynomic5	指定到节点为止的凸轮曲线的形状。 0: 直线(位移保持) ^{*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: 逆Trapeclloid 9: 单弦 10: 复弦 11: 逆复弦 12: NC2曲线 13: 3次曲线 14: 5次曲线
ConnectingVel Enable	连接速度有效	BOOL	TRUE, FALSE	设为TRUE, 则在指定的曲线形状为3次曲线或5次曲线时, 指定的连接速度生效。 设为FALSE, 则连接速度无效。
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, 可以指定与后续曲线的连接部的速度。 需要平稳连接曲线与曲线时指定。 单位为[指令单位/s]。
ConnectingAcc Enable	连接加速度有效	BOOL	TRUE, FALSE	设为TRUE, 则在指定的曲线形状为5次曲线时, 指定的连接加速度生效。 设为FALSE, 则连接加速度无效。
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, 可以指定与后续曲线的连接部的加速度。 需要平稳连接曲线与曲线时指定。 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. 指定时请确保绝对值在0.001以上。小数点后第4位四舍五入。

*2. Phase(主轴相位)和Distance(从轴位移)的有效位为7位。输入8位以上时, 有效位第8位后将舍去。舍去后与其他Phase(主轴相位)的值相同时, 会发生“凸轮节点主轴相位非升序(错误代码: 5740Hex)”错误。请在7位的有效位以内输入升序的值。

*3. 指定为直线(位移保持)时, Distance(从轴位移)无效, 按与前一节点指定值相同的值处理。如果在数组的元素编号为0时指定为直线(位移保持), Distance(从轴位移)按0处理。

*4. 指定时请确保各节点生成的凸轮数据的合计在65535以下。

● 凸轮表的生成示例

下面对通过本指令生成凸轮表的示例进行说明。

设CamNodes(凸轮节点)中指定的数组变量的元素数为4, Curve(曲线形状)为“_mcStraightLine(直线)”。ConnectingVel(连接速度)和ConnectingAcc(连接加速度), 以及CamProperty(凸轮属性)的InitVel(初始速度)和InitAcc(初始加速度)无效, 因此省略记载。Phase(主轴相位)和Distance(从轴位移)的值如下表所示。

CamNodes(凸轮节点)的 数组变量的元素编号	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元素数为4000, 相位和位移的值均为“不定值”。

本指令执行前的凸轮表的凸轮数据变量如下表所示。

元素编号	Phase(相位)	Distance(位移)
0	(不定值)	(不定值)
1	(不定值)	(不定值)
...		
1799	(不定值)	(不定值)
1800	(不定值)	(不定值)
1801	(不定值)	(不定值)
...		
3599	(不定值)	(不定值)
3600	(不定值)	(不定值)
3601	(不定值)	(不定值)
...		
3999	(不定值)	(不定值)

下面按顺序对因本指令的执行而发生的凸轮数据变量的变更过程进行说明。通过背景填充来表示变更位置。

执行本指令后, 凸轮数据变量的元素编号“0”的相位和位移改写为“0”。

元素编号	Phase(相位)	Distance(位移)
0	0.0	0.0
1	(不定值)	(不定值)
...		
1799	(不定值)	(不定值)
1800	(不定值)	(不定值)
1801	(不定值)	(不定值)
...		
3599	(不定值)	(不定值)
3600	(不定值)	(不定值)
3601	(不定值)	(不定值)
...		
3999	(不定值)	(不定值)

接着，根据CamNodes(凸轮节点)的元素编号“0”的指定值，计算从起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”，因此将改写凸轮数据变量的元素编号“1”至“1800”的相位和位移。

元素编号	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	(不定值)	(不定值)
...		
3599	(不定值)	(不定值)
3600	(不定值)	(不定值)
3601	(不定值)	(不定值)
...		
3999	(不定值)	(不定值)

同样，根据CamNodes(凸轮节点)的元素编号“1”的指定值，计算节点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”，因此将改写凸轮数据变量的元素编号“1801”至“3600”的相位和位移。

元素编号	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	(不定值)	(不定值)
...		
3999	(不定值)	(不定值)

接着，CamNodes(凸轮节点)的元素编号“2”为无效节点，因此结束计算。凸轮数据变量的元素编号“3601”开始的凸轮数据无效，因此将相位的值改写为“0”。

元素编号	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	(不定值)
...		
3999	(不定值)	(不定值)

但是，通过Sysmac studio的凸轮编辑器创建的凸轮数据变量的元素数为3601时，不存在无效凸轮数据，因此不会将相位的值改写为“0”。

元素编号	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

最后将改写后的凸轮数据变量的元素编号作为凸轮表的终点索引的值，输出到本指令的输出变量EndPointIndex(终点索引)。在该示例中，输出“3600”。

至此，本指令执行完毕。

● 发生凸轮表生成位移溢出时

通过本指令计算出的凸轮数据的Distance(位移)的值超出了REAL型的有效范围时，发生“凸轮表生成位移溢出(错误代码：5742 Hex)”错误。此时，不更新对应的Distance(位移)的值，并中止凸轮表的生成处理。凸轮表生成位移溢出仅可能在Curve(曲线形状)指定为3次曲线或5次曲线时发生。发生了该异常时，请参阅之后记载的3次曲线的位移计算方法或5次曲线的位移计算方法，修正输入输出变量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)中指定的数组变量的元素编号设为“m”。

元素编号为“m”的Curve(曲线形状)指定为3次曲线时，以元素编号“m-1”的Phase为主轴相位的初始值 P_{init} 。同样，以Distance为从轴位移的初始值 d_{init} 。m=0时， P_{init} 和 d_{init} 按“0”计算。此外，以元素编号“m”的Phase为主轴相位的终端值 P_{final} 。同样，以元素编号“m”的Distance为从轴位移的终端值

d_{final} 。

以从 d_{init} 起第n个凸轮数据的Distance(位移)为 $d(n)$ 时， $d(n)$ 如下式所示。

$0 \leq n < N$ 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

$$N = \frac{P_{final} - P_{phase}}{pitch}$$

$(P_{final} - P_{phase}) / pitch$ 不为整数时

$$N = \text{floor}\left(\frac{P_{final} - P_{phase}}{pitch}\right) + 1$$

$$a_1 = \frac{V_{init} \cdot T}{(d_{final} - d_{init})}$$

$$a_2 = 3 - (2 \cdot V_{init} + V_{fin}) \cdot \frac{T}{(d_{final} - d_{init})}$$

$$a_3 = (V_{init} + V_{fin}) \cdot \frac{T}{(d_{final} - d_{init})} - 2$$

$$T = \text{CycleTime} \cdot \frac{(P_{final} - P_{init})}{P_{max}}$$

pitch: CamNodes(凸轮节点)的元素编号 “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)的有效节点的Phase(主轴相位)的最大值

V_{init} : 凸轮曲线的初始速度[指令单位/s]

但是,

$m=0$ 时

$$V_{init} = \text{CamProperty(凸轮属性)的InitVel(初始速度)}$$

$m \geq 1$ 的情况下, CamNodes(凸轮节点)的元素编号 “m-1” 的Curve(曲线形状)为直线(_mcStraightLine)时

$$V_{init} = \frac{(\text{元素编号}m-1\text{的 } d_{final} - \text{元素编号}m-1\text{的 } d_{init})}{(\text{元素编号}m-1\text{的 } P_{final} - \text{元素编号}m-1\text{的 } P_{init})} \cdot \frac{P_{max}}{\text{CycleTime}}$$

$m \geq 1$ 的情况下, CamNodes(凸轮节点)的元素编号 “m-1” 的Curve(曲线形状)为3次曲线或5次曲线时

- CamNodes(凸轮节点)的元素编号 “m-1” 的ConnectingVelEnable(连接速度有效)为TRUE

$$V_{init} = \text{CamNodes(凸轮节点)的元素编号 “m-1” 的ConnectingVel(连接速度)}$$

- CamNodes(凸轮节点)的元素编号 “m-1” 的ConnectingVelEnable(连接速度有效)为FALSE

$$V_{init} = 0$$

$m \geq 1$ 的情况下, 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)的元素编号 “m” 的ConnectingVelEnable(连接速度有效)为TRUE时

$$V_{fin} = \text{CamNodes(凸轮节点)的元素编号 “m” 的ConnectingVel(连接速度)}$$

CamNodes(凸轮节点)的元素编号“m”的ConnectingVelEnable(连接速度有效)为FALSE时

- CamNodes(凸轮节点)的元素编号“m+1”的Curve(曲线形状)为直线(_mcStraightLine)

$$V_{fin} = \frac{(\text{元素编号}m+1\text{的 } d_{final} - \text{元素编号}m+1\text{的 } d_{init})}{(\text{元素编号}m+1\text{的 } P_{final} - \text{元素编号}m+1\text{的 } P_{init})} \cdot \frac{P_{max}}{CycleTime}$$

- CamNodes(凸轮节点)的元素编号“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)中指定的数组变量的元素编号设为“m”。

元素编号为“m”的Curve(曲线形状)指定为5次曲线时,以元素编号“m-1”的Phase为主轴相位的初始值 P_{init} 。同样,以Distance为从轴位移的初始值 d_{init} 。m=0时, P_{init} 和 d_{init} 按“0”计算。此外,以元素编号“m”的Phase为主轴相位的终端值 P_{final} 。同样,以元素编号“m”的Distance为从轴位移的终端值

d_{final} 。

以从 d_{init} 起第n个凸轮数据的Distance(位移)为 $d(n)$ 时, $d(n)$ 如下式所示。

$0 \leq n < N$ 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

$$N = \frac{P_{final} - P_{phase}}{pitch}$$

$(P_{final} - P_{phase}) / pitch$ 不为整数时

$$N = \text{floor}\left(\frac{P_{final} - P_{phase}}{pitch}\right) + 1$$

$$a_1 = \frac{V_{init} \cdot T}{(d_{final} - d_{init})}$$

$$a_2 = \frac{A_{init} \cdot T^2}{2 \cdot (d_{final} - d_{init})}$$

$$a_3 = \frac{-\left((3A_{init} - A_{fin}) \cdot \frac{T^2}{(d_{final} - d_{init})} + (8V_{fin} + 12V_{init}) \cdot \frac{T}{(d_{final} - d_{init})} - 20 \right)}{2}$$

$$a_4 = \left((1.5A_{init} - A_{fin}) \cdot \frac{T^2}{(d_{final} - d_{init})} + (7V_{fin} + 8V_{init}) \cdot \frac{T}{(d_{final} - d_{init})} - 15 \right)$$

$$a_5 = \frac{-\left((A_{init} - A_{fin}) \cdot \frac{T^2}{(d_{final} - d_{init})} + (6V_{fin} + 6V_{init}) \cdot \frac{T}{(d_{final} - d_{init})} - 12 \right)}{2}$$

$$T = CycleTime \cdot \frac{(P_{final} - P_{init})}{P_{max}}$$

pitch: CamNodes(凸轮节点)的元素编号 “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)的有效节点的Phase(主轴相位)的最大值

A_{init} : 凸轮曲线的初始加速度[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的情况下, CamNodes(凸轮节点)的元素编号 “m-1” 的Curve(曲线形状)为5次曲线时

• CamNodes(凸轮节点)的元素编号 “m-1” 的ConnectingVelEnable(连接速度有效)为TRUE

$A_{init} = CamNodes(凸轮节点)的元素编号 “m-1” 的ConnectingAcc(连接加速度)$

• CamNodes(凸轮节点)的元素编号 “m-1” 的ConnectingVelEnable(连接速度有效)为FALSE

$A_{init} = 0$

m ≥ 1的情况下, CamNodes(凸轮节点)的元素编号 “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)的元素编号 “m” 的ConnectingAccEnable(连接加速度有效)为TRUE时

$A_{fin} = CamNodes(凸轮节点)的元素编号 “m” 的ConnectingAcc(连接加速度)$

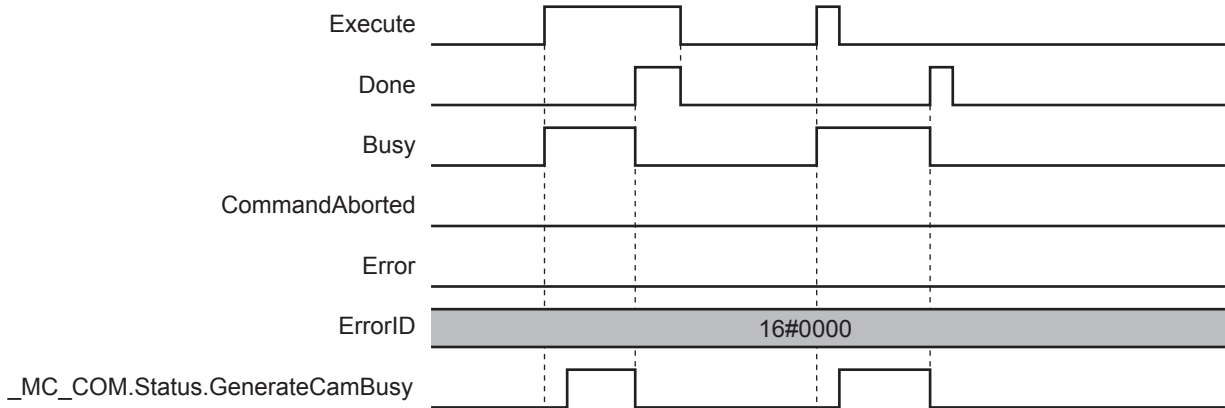
CamNodes(凸轮节点)的元素编号 “m” 的ConnectingAccEnable(连接加速度有效)为FALSE时

$A_{fin} = 0$

V_{init} : 凸轮曲线的初始速度[指令单位/s]及 V_{fin} : 关于凸轮曲线的终端速度[指令单位/s], 请参阅 “3次曲线的位移计算方法(P.5-27)”。

时序图

Busy(执行中)在Execute(执行)的上升沿变为TRUE,“凸轮表生成执行中(_MC_COM.Status.GenerateCam Busy)”在下一个周期变为TRUE。



重启运动指令

无法重启本指令。

重启后,将发生“无法重启运动指令(错误代码: 543B Hex)”错误。但是,执行中的凸轮表生成处理会继续执行。

多重启动运动指令

● 其它指令执行中的本指令启动

与本指令使用相同的CamTable(凸轮表)的MC_CamIn(凸轮动作开始)指令、MC_SaveCamTable(保存凸轮表)指令、以及MC_SetCamTableProperty(更新凸轮表属性)指令已启动时,本指令不能多重启动。

● 本指令执行中的其它指令启动

本指令执行中,与本指令使用相同的CamTable(凸轮表)的MC_CamIn(凸轮动作开始)指令、MC_SaveCamTable(保存凸轮表)指令、以及MC_SetCamTableProperty(更新凸轮表属性)指令无法多重启动。

● 本指令执行中的本指令启动

本指令执行中,无法启动本指令的其它实例。此外,“凸轮表生成执行中(_MC_COM.Status.GenerateCam Busy)”为TRUE时,无法启动本指令。请使用“凸轮表生成执行中(_MC_COM.Status.GenerateCam Busy)”或本指令的输出变量进行排他处理。

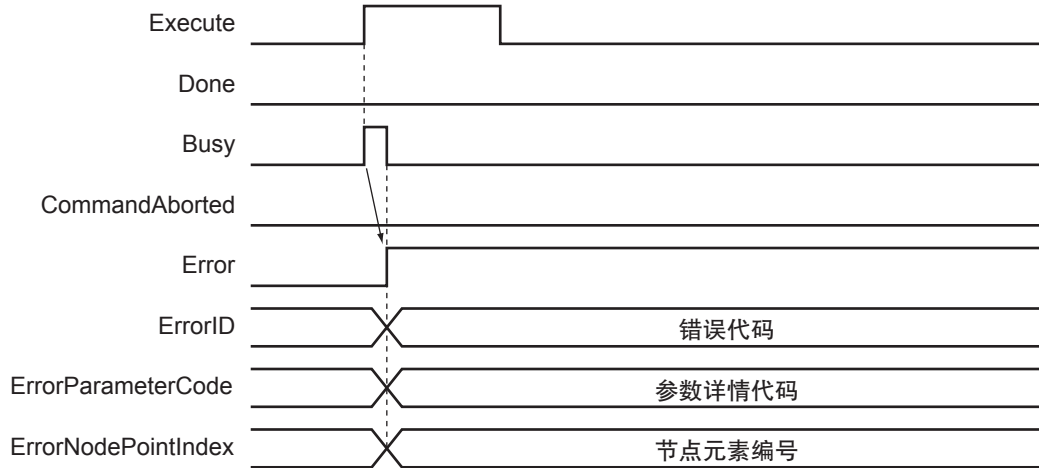
以相同周期启动本指令的多个实例的情况下,该周期中“凸轮表生成执行中(_MC_COM.Status.GenerateCam Busy)”变为FALSE。请使用其他实例的Busy(执行中)进行排他处理。

异常

本指令执行中发生异常时，Error(错误)变为TRUE。

可查看ErrorID(错误代码)的输出值，了解发生异常的原因。

此外，根据ErrorID(错误代码)，附属信息会被输出到ErrorParameterCode(参数详细代码)和ErrorNodePointIndex(节点元素编号)。



即使在本指令执行中发生异常，CamTable(凸轮表)中指定的凸轮数据变量也会保存本指令执行前的值。但是，下列情况下不保存凸轮数据变量的值。

- 启动Execute(执行)后，变更了CamNodes(凸轮节点)中指定的参数时
- 通过本指令计算出的凸轮数据的Distance(位移)的值超出了REAL型的有效范围时

未保存凸轮数据变量的值时，凸轮表会变为错误状态。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(保存凸轮表)指令、以及MC_SetCamTableProperty(更新凸轮表属性)指令时会发生异常。

要解除凸轮表的错误状态，请在通过指令解除发生中的异常后，执行以下任意一项操作。

- 对错误状态的凸轮表重新执行本指令，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

在程序模式或MC试运行中，执行中的本指令变为异常时，ErrorID(错误代码)中不会输出错误代码。此时，请通过事件日志功能或运动控制系统变量的MC通用 轻度故障代码(_MC_COM.MFaultLvl.Code)来确认异常的原因。

● 错误代码

关于本指令发生的异常，请参阅 □□ “A-1 错误代码一览(P.A-2)”。

示例程序

下面，对改写通过Sysmac Studio的凸轮编辑器创建的凸轮表的凸轮数据变量的示例程序进行说明。

参数设定

该示例程序中使用的最少的必要设定如下所示。

● 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

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

改写内容

下面，对通过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，以及通过该示例程序改写的值进行说明。

● 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

该示例程序中，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。

元素编号	Phase (主轴相位)	Distance (从轴位移)	Curve (曲线形状)	ConnectingVel Enable (连接速度有效)	ConnectingV el (连接速度)	Connecting AccEnable (连接加速度 有效)	ConnectingAcc (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

该示例程序中，将变更元素编号“0”的曲线形状，在元素编号“1”中增加节点。

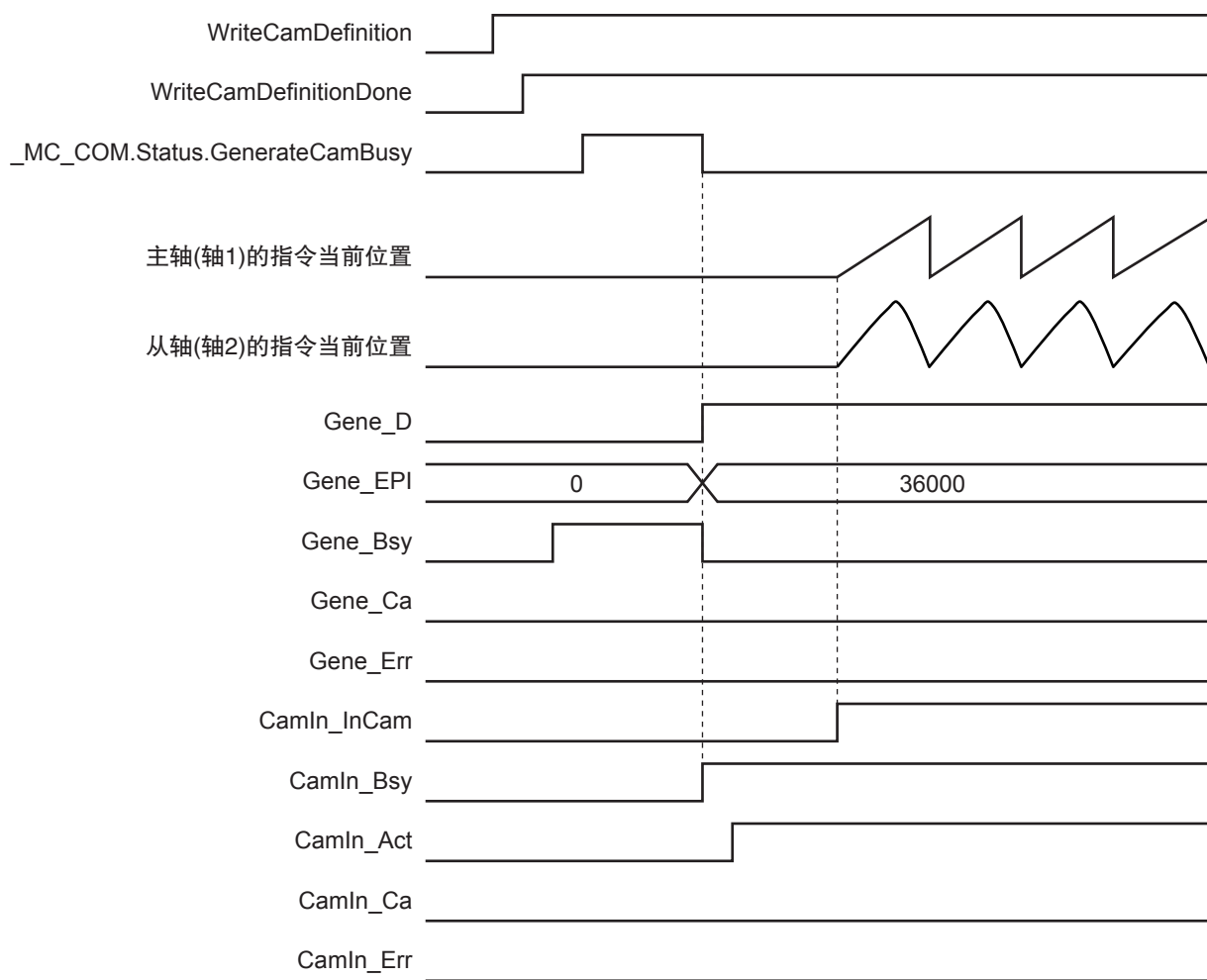
元素编号	Phase (主轴相位)	Distance (从轴位移)	Curve (曲线形状)	ConnectingVel Enable (连接速度有效)	ConnectingV el (连接速度)	Connecting AccEnable (连接加速度 有效)	ConnectingAcc (连接加速度)	PhasePitch (相位间隔 宽度)
0	180.000	-100.000	_mcPolynomial5	FALSE	0.000	FALSE	0.000	0.010
1	200.000	-102.000	_mcStraightLine	FALSE	0.000	FALSE	0.000	0.010
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

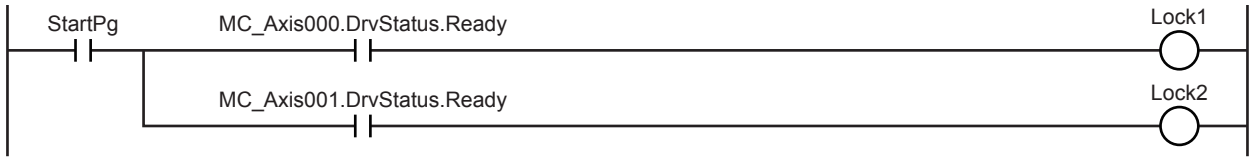
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	是主轴轴1的轴变量。
MC_Axis001	_sAXIS_REF	-	是从轴轴2的轴变量。
CamProfile0	ARRAY[0..36000] OF _sMC_CAM_REF	-	是凸轮数据变量。 事先通过Sysmac Studio的凸轮编辑器创建。
Pwr1_Status	BOOL	-	分配给MC_Power的实例PWR1的输出Status的变量。进入伺服ON状态时，该变量变为TRUE。
Pwr2_Status	BOOL	-	分配给MC_Power的实例PWR2的输出Status的变量。进入伺服ON状态时，该变量变为TRUE。
StartPg	BOOL	-	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则进入伺服ON状态。
WriteCamDefinition	BOOL	-	该变量为TRUE时改写凸轮属性变量和凸轮节点变量的值。
CamProperty0	_sMC_CAM_PROPERTY	-	是凸轮属性变量。
CamNode0	ARRAY[0..3] OF _sMC_CAM_NODE	-	是凸轮节点变量。
_MC_COM.Status.GenerateCamBusy	BOOL	-	是运动控制系统变量。凸轮表生成处理中变为TRUE。

● 时序图

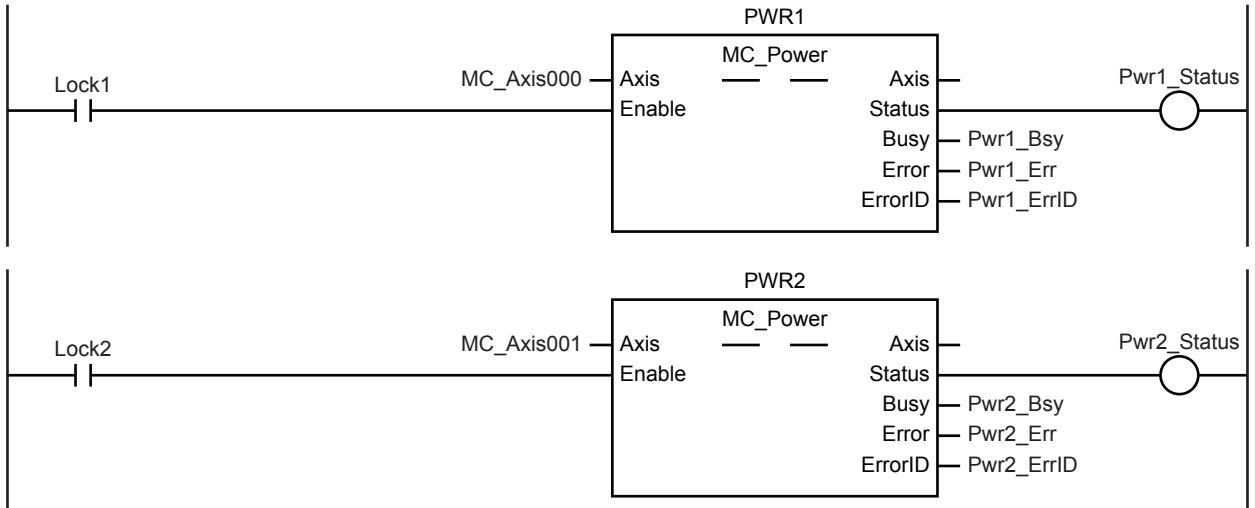


● 示例程序

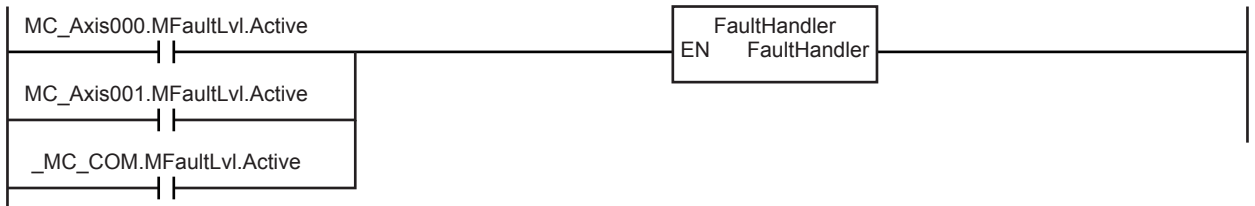
触点StartPg变为TRUE时，确认各轴的伺服驱动器处于伺服准备就绪状态



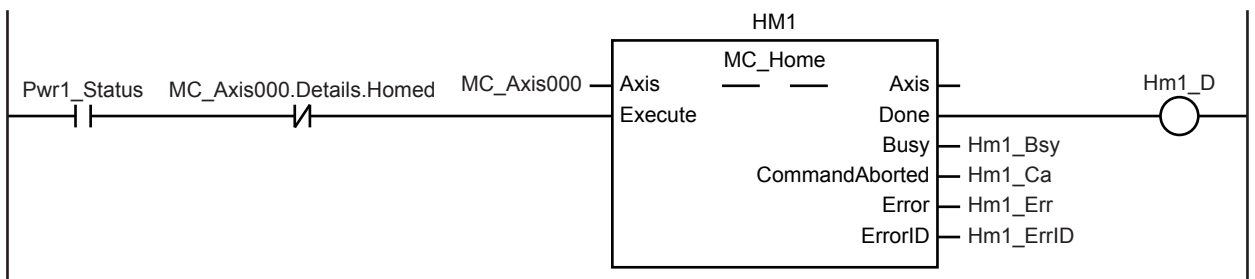
如果处于伺服准备就绪状态，则将各轴设为伺服ON状态



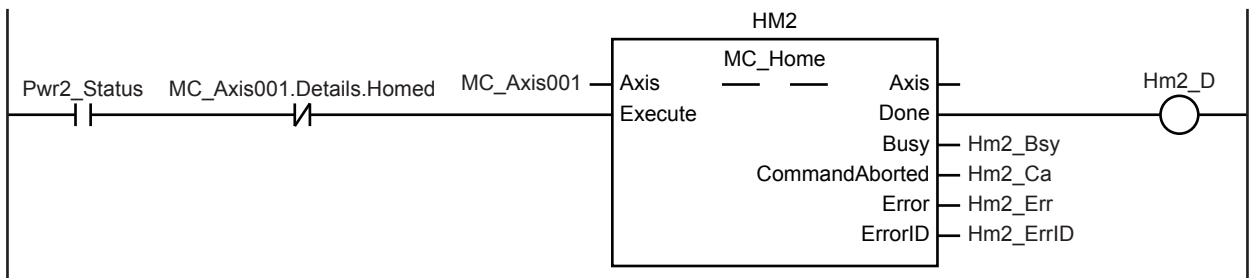
轴或MC通用发生轻度故障程度的异常时，执行设备的异常处理(FaultHandler)。发生异常时的处理(FaultHandler)由客户根据装置进行编程



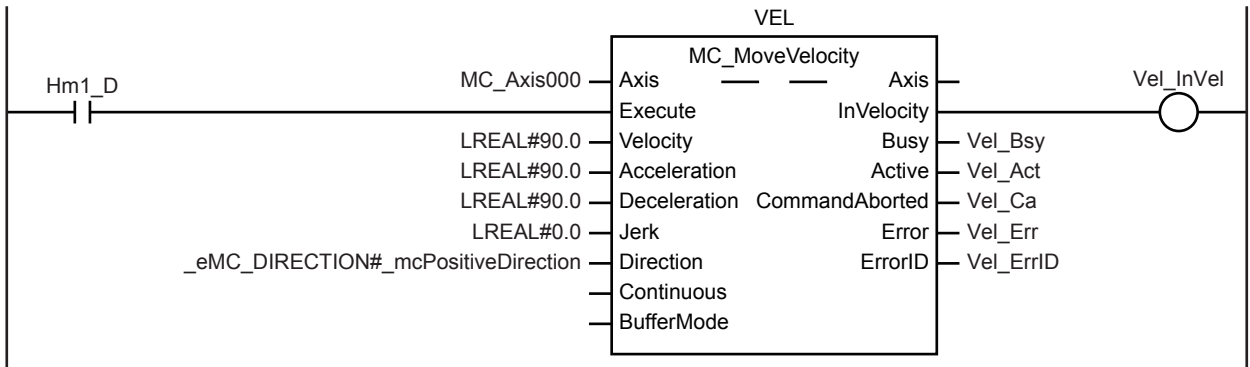
轴(轴1)变为伺服ON状态，若原点未确定，则执行原点复位，确定原点



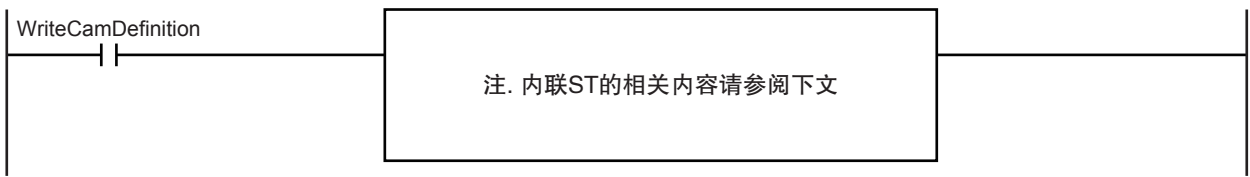
从轴(轴2)变为伺服ON状态，若原点未确定，则执行原点复位，确定原点



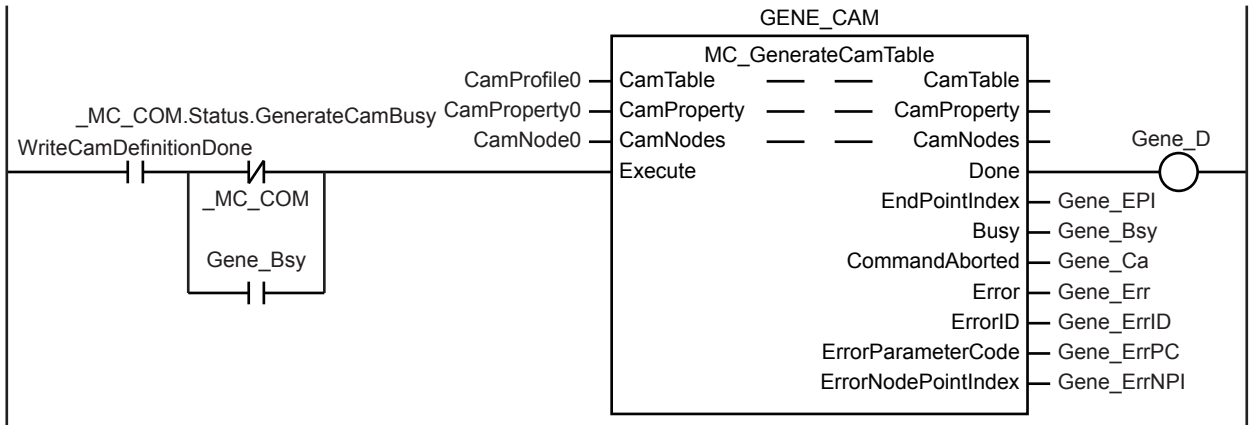
主轴(轴1)原点复位完成后, 启动MC_MoveVelocity(速度控制)指令



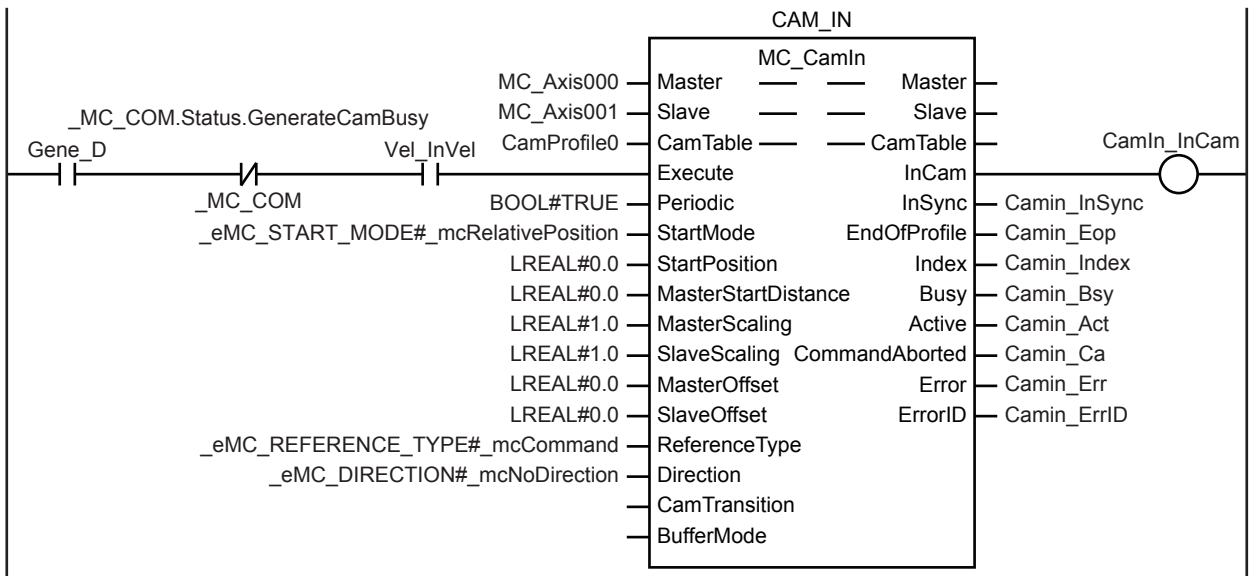
WriteCamDefinition为TRUE, MC_GenerateCamTable(凸轮表生成)指令为未执行状态时, 改写凸轮属性和凸轮节点。改写后将WriteCamDefinitionDone设为TRUE



WriteCamDefinitionDone为TRUE, 凸轮表生成处理中以外时, 启动MC_GenerateCamTable(生成凸轮表)指令



如果凸轮表生成处理完成, 且主轴(轴1)到达目标速度, 则启动MC_CamIn(凸轮动作开始)指令



内联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;

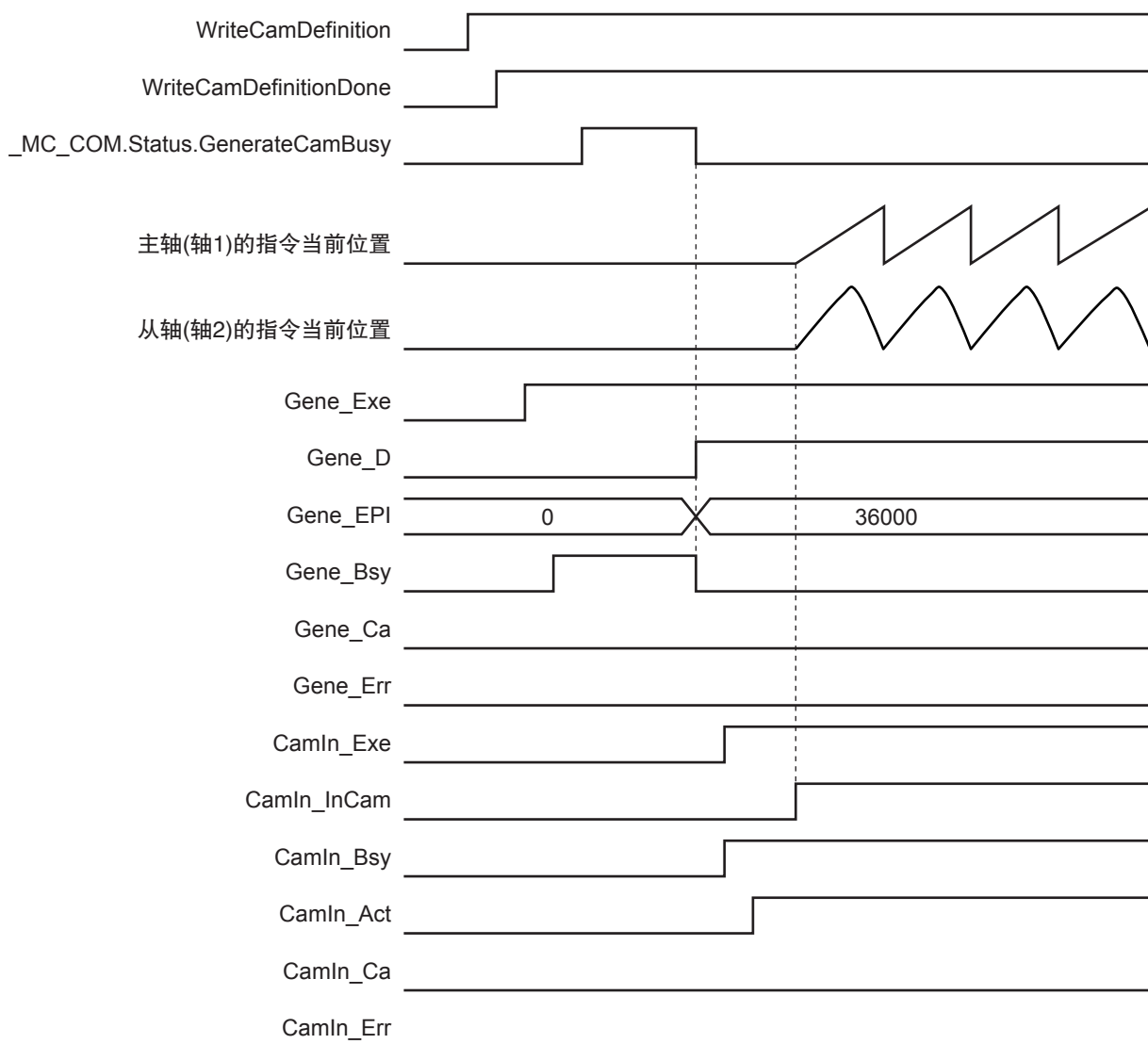
```

结构文本(ST)

● 主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	-	是主轴轴1的轴变量。
MC_Axis001	_sAXIS_REF	-	是从轴轴2的轴变量。
CamProfile0	ARRAY[0..36000] OF _sMC_CAM_REF	-	是凸轮数据变量。 事先通过Sysmac Studio的凸轮编辑器创建。
Pwr1_Status	BOOL	-	分配给MC_Power的实例PWR1的输出Status的变量。进入伺服ON状态时，该变量变为TRUE。
Pwr2_Status	BOOL	-	分配给MC_Power的实例PWR2的输出Status的变量。进入伺服ON状态时，该变量变为TRUE。
StartPg	BOOL	-	如果该变量为TRUE，EtherCAT的过程数据通信已建立，则进入伺服ON状态。
WriteCamDefinition	BOOL	-	该变量为TRUE时改写凸轮属性变量和凸轮节点变量的值。
CamProperty0	_sMC_CAM_PROPERTY	-	是凸轮属性变量。
CamNode0	ARRAY[0..3] OF _sMC_CAM_NODE	-	是凸轮节点变量。
_MC_COM.Status.GenerateCamBusy	BOOL	-	是运动控制系统变量。凸轮表生成处理中变为TRUE。

● 时序图



● 示例程序

```

// StartPg为TRUE时，确认伺服驱动器处于伺服准备就绪状态，
// 将轴1 设为伺服ON状态
// 如果未处于伺服准备就绪状态，则设为伺服OFF
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE;
ELSE
    Pwr1_En:=FALSE;
END_IF;

// StartPg为TRUE时，确认伺服驱动器处于伺服准备就绪状态，
// 将轴2 设为伺服ON状态
// 如果未处于伺服准备就绪状态，则设为伺服OFF
IF (StartPg=TRUE)
  AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En:=TRUE;
ELSE
    Pwr2_En:=FALSE;
END_IF;

// 轴或MC通用发生轻度故障后，执行异常时处理FaultHandler。
// 发生异常时的处理(FaultHandler)由客户根据装置进行编程
IF (MC_Axis000.MFaultLvl.Active=TRUE)
OR (MC_Axis001.MFaultLvl.Active=TRUE)
OR (_MC_COM.MFaultLvl.Active=TRUE) THEN
  FaultHandler();
END_IF;

// 轴1处于可运行状态，且原点未确定时，执行原点复位
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
  Hm1_Ex:=TRUE;
END_IF;

// 轴2处于可运行状态，且原点未确定时，执行原点复位
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
  Hm2_Ex:=TRUE;
END_IF;

// 轴1原点复位完成后，启动MC_MoveVelocity
IF Hm1_D=TRUE THEN
  Vel_Ex := TRUE;
END_IF;

// WriteCamDefinition为TRUE，MC_GenerateCamTable(凸轮表生成)指令为未执行状态
// 时，改写凸轮属性和凸轮节点
// 改写后将WriteCamDefinitionDone设为TRUE
IF (WriteCamDefinition = TRUE) THEN
  //凸轮属性的改写
  CamProperty0.CycleTime                := REAL#0.800;
  //凸轮节点的改写
  CamNode0[0].Curve                      := _eMC_CAM_CURVE#_mcPolynomial5;
  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, 凸轮表生成处理中以外时, 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;

// 如果凸轮表生成处理完成, 且主轴(轴1)到达目标速度, 则启动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

改写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

Ver.1.08以上的CPU单元和Ver.1.09以上的Sysmac Studio组合时可使用本指令。

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	在上升沿开始指令。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时, 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为正常。
ErrorParameterCode	参数详情代码	WORD	*	有些错误代码存在附属信息。如果存在该信息, 则输出异常参数的详情代码。

* 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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，中止本指令时	<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含有异常因素时	异常已解除时

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	-	指定改写对象轴。 ^{*1}
AxisParameter	轴参数	_sAXIS_PARAM	-	指定改写的值。 ^{*2}

*1. 请在使用在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型的用户定义变量。

功能说明

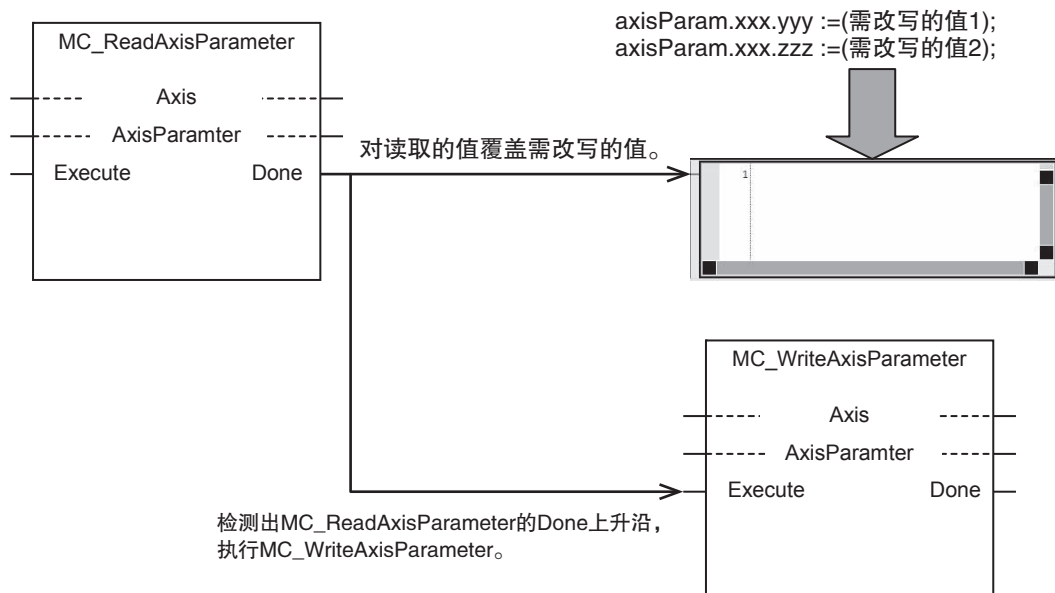
- 本指令在Execute(启动)的上升沿，将AxisParameter(轴参数)指定的值写入指定Axis(轴)的轴参数中。
- 仅在指定轴的轴使用为未使用轴时可以写入。在除此以外的情况下执行时会发生错误，且不执行轴参数的写入。保持指令执行前的值。
- 通过MC_ChangeAxisUse(轴使用变更)指令切换轴使用后执行本指令时，请确认MC_ChangeAxisUse(轴使用变更)指令的输出变量“Done(完成)”为TRUE后再执行。
- AxisParameter(轴参数)为错误值的情况下，或轴参数间不匹配的情况下，执行时会发生错误，且不执行轴参数的写入。保持指令执行前的值。

关于轴参数可设定范围检查的内容及轴参数间的匹配检查，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)”或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

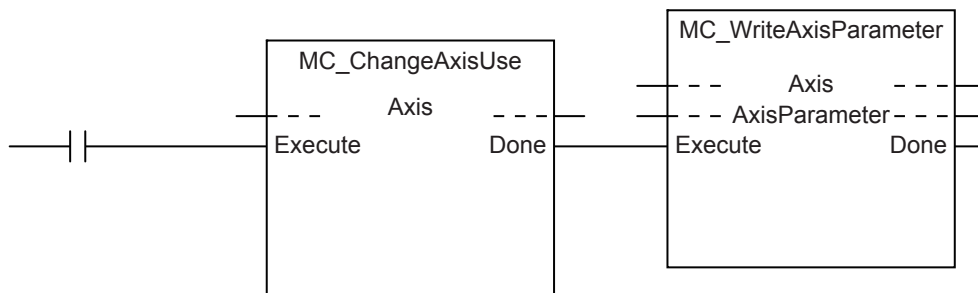


使用注意事项

- 通过本指令改写的值未保存至CPU单元的非易失性存储器中。因此，关闭CPU单元的电源、开始下载、开始执行MC功能模块的重启处理时，改写值消失，恢复为通过Sysmac Studio设定的值。
- 本指令改写的值无法通过Sysmac Studio上传。
- 除了写入轴参数外，本指令还需要对写入对象轴的全部轴参数进行参数设定。关于对象参数，请参阅 □ “作为写入及读取对象的轴参数(P.5-54)”。无需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。
- 同时使用本指令与MC_ReadAxisParameter(轴参数读取)指令时的执行示意图如下所示。



- 同时使用本指令与MC_ChangeAxisUse(轴使用变更)指令时的示例如下所示。请在确认MC_ChangeAxisUse(轴使用变更)指令的Done(完成)后再执行本指令。



- 对编码器种类设定为“绝对值编码器(ABS)”的轴，使用本指令变更“单位转换设定”或“位置计数设定”后，机械的物理位置与MC功能模块上的轴的位置关系会发生变化。此时，在通过MC_ChangeAxisUse(轴使用变更)指令变更为使用轴后，请执行原点复位指令，再次确定原点。
变更为使用轴时发生了“无法计算绝对值编码器当前位置(错误代码：6458 Hex)”错误时，在解除轴的异常之后，请执行原点复位指令，再次确定原点。
- 轴参数的Reserved变量请勿指定为“0”以外的值。

指令详情

下面对指令详细说明。

● sAXIS_PARAM

sAXIS_PARAM型是用于指定轴参数的值的结构体型数据类型。由各轴参数分类的结构体型数据类型构成。

在本指令中，作为输入输出变量AxisParameter(轴参数)中指定的变量使用。

下面对各结构要素进行说明。

结构要素变量	参数名称	数据类型	内容
UnitConversion	单位转换设定	<u>s</u> AXIS_UNIT_CONVERSION_SETTINGS	对单位转换设定的各要素变量指定改写的值。
Operation	动作设定	<u>s</u> AXIS_OPERATION_SETTINGS	对动作设定的各要素变量指定改写的值。
OtherOperation	扩展动作设定	<u>s</u> AXIS_OTHER_OPERATION_SETTINGS	对扩展动作设定的各要素变量指定改写的值。
Limit	限制设定	<u>s</u> AXIS_LIMIT_SETTINGS	对限制设定的各要素变量指定改写的值。
PosCount	位置计数设定	<u>s</u> AXIS_POSITION_COUNT_SETTINGS	对位置计数设定的各要素变量指定改写的值。
Homing	原点复位设定	<u>s</u> AXIS_HOMING_SETTINGS	对原点复位设定的各要素变量指定改写的值。
Reserved	(备用区域)	ARRAY[0..255] OF BYTE	-

● sAXIS_UNIT_CONVERSION_SETTINGS(单位转换设定)

sAXIS_UNIT_CONVERSION_SETTINGS型是用于指定轴参数的单位转换设定的值的结构体数据类型。

下面对各结构要素进行说明。

结构要素变量	参数名称	数据类型	有效范围	内容
UnitDisplay	显示单位	<u>e</u> MC_ UNITS	0: <u>mc</u> Pls 1: <u>mc</u> Mm 2: <u>mc</u> Um 3: <u>mc</u> Nm 4: <u>mc</u> Deg 5: <u>mc</u> Inch	设定指令位置的单位。 0: pulse 1: mm 2: μ m 3: nm 4: degree 5: inch
CmdPlsCount PerMotor Rotation	电机转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转1圈的脉冲数。 根据电子齿轮比将指令值转换为脉冲量。
WorkTravel DistancePer MotorRotation	电机转1圈的移动量	LREAL	0.000000001 ~ 4294967295	设定指令位置电机转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
WorkTravel DistancePer WorkSide Rotation *1 *2 *3	工件侧转1圈的移动量	LREAL	双精度实数型正数	设定工件侧转1圈时的移动量。
WorkSideGear Ratio *1 *2	工件侧齿轮比	UDINT	1 ~ 4294967295	设定工件侧的齿轮比。
MotorSideGearRatio	电机侧齿轮比	UDINT	1 ~ 4294967295	设定电机侧的齿轮比。
Reserved	(备用区域)	ARRAY [0..7] OF BYTE *4	-	-

*1. Ver.1.11以上的CPU单元和Ver.1.15以上的Sysmac Studio组合时可使用。

*2. 将减速机设定设定为“使用”时有效。

*3. 仅计数模式为线性模式时可设定。为旋转模式时，根据环计数器上限设定值及环计数器下限设定值计算得出。

*4. 对于Ver.1.10以下版本的CPU单元，为ARRAY [0..31] OF BYTE。

● _sAXIS_OPERATION_SETTINGS(动作设定)

_sAXIS_OPERATION_SETTINGS型是用于指定轴参数的动作设定的值的结构体型数据类型。

下面对各结构要素进行说明。

结构要素变量	参数名称	数据类型	有效范围	内容
MaxVel	最高速度	LREAL	双精度实数型正数	设定各轴的最高速度。 设定时请勿超出使用电机的最大转速。
StartVel	启动速度	LREAL	0.0 ~ 双精度实数型正的上限值	设定各轴的启动速度。 请勿超过最高速度。
MaxJogVel	微动最高速度	LREAL	双精度实数型正数	设定各轴的微动最高速度。 请勿超过最高速度。
MaxAcc	最大加速度	LREAL	双精度实数型正数、0	设定轴动作指令时各轴的最大加速度。 设定为“0”时，无加速度限制。
MaxDec	最大减速度	LREAL	双精度实数型正数、0	设定轴动作指令时各轴的最大减速度。 设定为“0”时，无减速度限制。
AccDecOver	加减速超限	<u>_e</u> MC_ ACCDECOV ER	0: <u>_mc</u> AccDecOverBuffer 1: <u>_mc</u> AccDecOverRapid 2: <u>_mc</u> AccDecOverError Stop	在轴的加减速控制中，优先向目标位置 停止后，产生减速超限。指定超过最大 加减速度的动作。 0: 提高加减速速度(将合并切换为等待)*1 1: 提高加减速速度 2: 异常停止*2
ReverseMode	反转时动作	<u>_e</u> MC_ REVERSE_ MODE	0: <u>_mc</u> ReverseMode DecelerationStop 1: <u>_mc</u> ReverseMode ImmediateStop	指定多重启动指令时、重启指令时、中 断标准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	设定对最高速度(用于输出各轴的速度 警告)的比例。 设定为“0”时，不输出速度警告。
AccWarningVal	加速度警告值	UINT	0、1 ~ 100	设定对最高加速度(用于输出各轴的加速 度警告)的比例。 设定为“0”时，不输出加速度警告。
DecWarningVal	减速度警告值	UINT	0、1 ~ 100	设定对最高减速度(用于输出各轴的减速 度警告)的比例。 设定为“0”时，不输出减速度警告。
PosiTrqWarningVal	正方向转矩警告 值	UINT	0、1 ~ 1000	设定转矩指令值(用于输出各轴的正方向 转矩警告)。 设为“0”时，不输出正方向转矩警告。
NegaTrq WarningVal	负方向转矩警告 值	UINT	0、1 ~ 1000	设定转矩指令值(用于输出各轴的负方向 转矩警告)。 设为“0”时，不输出负方向转矩警告。
InPosRange	到位宽度	LREAL	双精度实数型正数、0	设定定位完成宽度。
InPosCheckTime	到位检查时间	UINT	0、1 ~ 10000	以毫秒为单位，设定定位完成的检查 时间。 设定为“0”时，仅原点复位时的原点 确定时，持续定位完成的检查。非原点 复位时，不作定位完成检查。
ActVelFilter TimeConstant	反馈速度滤波时 间常数	UINT	0、1 ~ 100	以毫秒为单位，设定反馈速度移动平均 值的计算时间。 设定为“0”时，不计算移动平均值。
ZeroPosRange	原点位置范围	LREAL	双精度实数型正数、0	设定原点位置检测宽度。
Reserved	(备用区域)	ARRAY [0..31] OF BYTE	-	-

*1. Ver.1.10以上的CPU单元，无法将合并切换为等待。详情请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

*2. Ver.1.10以上的CPU单元，合并动作时不会异常停止。详情请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇 (SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

● _sAXIS_OTHER_OPERATION_SETTINGS(扩展动作设定)

_sAXIS_OTHER_OPERATION_SETTINGS型是用于指定轴参数的扩展动作设定的值的结构体型数据类型。

下面对各结构要素进行说明。

结构要素变量	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	<u>_e</u> MC_STOP _MODE	1: <u>_mc</u> ImmediateStop 2: <u>_mc</u> ImmediateStopFEReset 3: <u>_mc</u> FreeRunStop	设定立即停止输入有效时的MC功能模块侧的停止方法。 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
LimitInputStop Mode	极限输入停止方法	<u>_e</u> MC_STOP _MODE	0: <u>_mc</u> DecelerationStop 1: <u>_mc</u> ImmediateStop 2: <u>_mc</u> ImmediateStopFEReset 3: <u>_mc</u> FreeRunStop	设定正方向极限输入或负方向极限输入有效时的MC功能模块侧的停止方法。 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	驱动器错误复位监视时间	UINT	1 ~ 1000	设定执行驱动器错误复位时的监视时间。(单位: ms) 超过监视时间时, 即使驱动器异常未复位, 也会结束复位处理。
MaxPosiTrqLimit	正方向转矩限制上限值	LREAL	0.0 ~ 1000.0	设定正方向转矩限制值的上限。
MaxNegaTrqLimit	负方向转矩限制上限值	LREAL	0.0 ~ 1000.0	设定负方向转矩限制值的上限。
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
NegaLimitInput LogicInversion	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	设定原点附近输入信号的逻辑反转。 TRUE: 反转 FALSE: 不反转
Reserved	(备用区域)	ARRAY [0..31] OF BYTE	-	-

● _sAXIS_LIMIT_SETTINGS(限制设定)

_sAXIS_LIMIT_SETTINGS型是用于指定轴参数的限制设定的值的结构体型数据类型。

下面对各结构要素进行说明。

结构要素变量	参数名称	数据类型	有效范围	内容
SwLimitMode	软件限制功能	<u>_eMC_</u> SWLMT_ MODE	0: <u>_mcNonSwLmt</u> 1: <u>_mcCmdDecelerationStop</u> 2: <u>_mcCmdImmediateStop</u> 3: <u>_mcActDecelerationStop</u> 4: <u>_mcActImmediateStop</u>	选择软件限制的功能。 0: 无效 1: 对指令位置启用, 减速停止 2: 对指令位置启用, 立即停止 3: 对反馈位置启用, 减速停止 4: 对反馈位置启用, 立即停止
PosiSwLimit	正方向软件限制	LREAL	双精度实数型负数、正数、0	设定正方向侧的软件限制。
NegaSwLimit	负方向软件限制	LREAL	双精度实数型负数、正数、0	设定负方向侧的软件限制。
FollowingError OverVal	位置偏差超过值	LREAL	双精度实数型正数、0	设定超过位置偏差的检查值。 设定为“0”时, 超过位置偏差的检查无效。
FollowingError WarningVal	位置偏差警告值	LREAL	双精度实数型正数、0	设定位置偏差警告的检查值。 设定为“0”时, 位置偏差警告检查无效。
Reserved	(备用区域)	ARRAY [0..31] OF BYTE	-	-

● _sAXIS_POSITION_COUNT_SETTINGS(位置计数设定)

_sAXIS_POSITION_COUNT_SETTINGS型是用于指定轴参数的位置计数设定的值的结构体型数据类型。

下面对各结构要素进行说明。

结构要素变量	参数名称	数据类型	有效范围	内容
CountMode	计数模式	<u>_eMC_</u> COUNT_ MODE	0: <u>_mcCountModeLinear</u> 1: <u>_mcCountModeRotary</u>	设定位置的计数模式。 0: 线性模式(有限长) 1: 旋转模式(无限长)
ModuloMaxPosVal	环计数器上限设定值	LREAL	双精度实数型负数、正数、0	设定将计数模式设为旋转模式时的环计数器上限值。
ModuloMinPosVal	环计数器下限设定值	LREAL	双精度实数型负数、正数、0	设定将计数模式设为旋转模式时的环计数器下限值。
Reserved	(备用区域)	ARRAY [0..31] OF BYTE	-	-

● _sAXIS_HOMING_SETTINGS(原点复位设定)

_sAXIS_HOMING_SETTINGS型是用于指定轴参数的原点复位设定的值的结构体型数据类型。

下面对各结构要素进行说明。

结构要素变量	参数名称	数据类型	有效范围	内容
Mode	原点复位动作	<u>_eMC_</u> HOMING_ MODE	0: <u>_mcHomeSwTurnHomeSwOff</u> 1: <u>_mcHomeSwTurnHomeSwOn</u> 4: <u>_mcHomeSwOff</u> 5: <u>_mcHomeSwOn</u> 8: <u>_mcLimitInputOff</u> 9: <u>_mcHomeSwTurnHomeMask</u> 11: <u>_mcLimitInputOnly</u> 12: <u>_mcHomeSwTurn</u> HoldingTime 13: <u>_mcNoHomeSw</u> HoldingHomeInput 14: <u>_mcHomePreset</u>	设定原点复位的动作方法。 0: 指定为附近避让、近原点输入OFF 1: 指定为附近避让、近原点输入ON 4: 指定为近原点输入OFF 5: 指定为近原点输入ON 8: 指定为极限输入OFF 9: 指定为附近避让、原点输入屏蔽距离 11: 仅极限输入 12: 指定为附近避让、接触时间 13: 指定为无近原点输入、接触原点输入 14: 原点预设

结构要素变量	参数名称	数据类型	有效范围	内容
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_DIRE CTION	0: _mcPositiveDirection 2: _mcNegativeDirection	设定启动原点复位时的开始方向。 0: 正方向 2: 负方向
HomeInput Detection Direction	原点检测方向	_eMC_DIRE CTION	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
PosiLimitInput Mode	正方向极限输入 时动作	_eMC_ LIMIT_ REVERSE_ MODE	0: _mcErrorStop 1: _mcRevImmediateStop 2: _mcRevDecelerationStop	设定原点复位中正方向极限输入“ON” 时的停止方法。 0: 不反转、轴异常而停止 (遵照极限输入停止方法) 1: 反转、立即停止 2: 反转、减速停止
NegaLimitInput Mode	负方向极限输入 时动作	_eMC_ LIMIT_ REVERSE_ MODE	0: _mcErrorStop 1: _mcRevImmediateStop 2: _mcRevDecelerationStop	设定原点复位中负方向极限输入“ON” 时的停止方法。 0: 不反转、轴异常而停止 (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	双精度实数型正数	设定原点复位中的速度。
ApproachVel	原点复位接近速度	LREAL	双精度实数型正数	设定近原点输入ON后的速度。
Acc	原点复位加速度	LREAL	双精度实数型正数、0	指定原点复位时的加速度。 设为“0”时，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	双精度实数型正数、0	指定原点复位时的减速度。 设为“0”时，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	双精度实数型正数、0	指定原点复位时的跃度。 设定为“0”时无跃度。
HomeInput MaskDistance	原点输入屏蔽距离	LREAL	双精度实数型正数、0	对原点复位动作模式指定为附近避让、 原点输入屏蔽距离时原点输入的屏蔽距离进行设定。
HomeOffset	原点位置偏置	LREAL	双精度实数型负数、正数、0	原点复位完成时，按照设定值预设当前位置。
HoldingTime	原点复位接触时间	UINT	0 ~ 10000	以毫秒为单位，对原点复位动作模式指定为附近避让、接触时间时的接触时间进行设定。
CompensationVal	原点复位补偿值	LREAL	双精度实数型负数、正数、0	设定原点复位的原点确定后的原点复位补偿量。
CompensationVel	原点复位补偿速度	LREAL	双精度实数型正数	设定原点复位补偿时的速度。
Reserved	(备用区域)	ARRAY [0..31] OF BYTE	-	-

作为写入及读取对象的轴参数

以下记载了作为本指令的写入对象的轴参数。

作为MC_ReadAxisParameter(轴参数读取)指令的读取对象的轴参数也和以下相同。

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	轴号	×
	运动控制*1	
	轴使用	×
	轴种类	×
	控制功能*2	
	输入/输出设备	×
单位转换设定	显示单位	○
	电机转1圈的脉冲数	○
	电机转1圈的移动量	○
	减速机使用*3	○
	工件侧转1圈的移动量 *3	○ *4
	工件侧齿轮比 *3	○
	电机侧齿轮比 *3	○
动作设定	最高速度	○ *5
	启动速度	○ *5
	微动最高速度	○ *5
	最大加速度	○ *5
	最大减速度	○ *5
	加减速超限	○ *5
	反转时动作	○ *5
	速度警告值	○ *5
	加速度警告值	○ *5
	减速度警告值	○ *5
	正方向转矩警告值	○ *5
	负方向转矩警告值	○ *5
	到位宽度	○ *5
	到位检查时间	○ *5
	反馈速度滤波时间常数	○
	原点位置范围	○ *5
扩展动作设定	立即停止输入停止方法	○ *5
	极限输入停止方法	○ *5
	驱动器错误复位监视时间	○ *5
	正方向转矩限制上限值	○ *5
	负方向转矩限制下限值	○ *5
	立即停止输入逻辑反转	○ *5
	正方向极限输入逻辑反转	○ *5
	负方向极限输入逻辑反转	○ *5
	近原点输入逻辑反转	○ *5

轴参数分类	轴参数名称	○：写入/读取对象 ×：非写入/读取对象
限制设定	软件限制功能	○ *5
	正方向软件限制	○ *5
	负方向软件限制	○ *5
	位置偏差超过值	○ *5
	位置偏差警告值	○ *5
位置计数设定	计数模式	○
	环计数器上限设定值	○
	环计数器下限设定值	○
	编码器种类	×
伺服驱动器设定	环计数器上限设定值	×
	环计数器下限设定值	×
	PDS状态控制方式 *6	×
原点复位设定	原点复位动作	○ *5
	原点输入信号	○ *5
	原点复位开始方向	○ *5
	原点检测方向	○ *5
	正方向极限输入时动作	○ *5
	负方向极限输入时动作	○ *5
	原点复位速度	○ *5
	原点复位接近速度	○ *5
	原点复位加速度	○ *5
	原点复位减速度	○ *5
	原点复位跃度	○ *5
	原点输入屏蔽距离	○ *5
	原点位置偏置	○ *5
	原点复位接触时间	○ *5
	原点复位补偿值	○ *5
	原点复位补偿速度	○ *5

*1 NX701 CPU单元独有的参数。

*2 NX1P2 CPU单元独有的参数。

*3 Ver.1.11以上的CPU单元和Ver.1.15以上的Sysmac Studio组合时可使用。

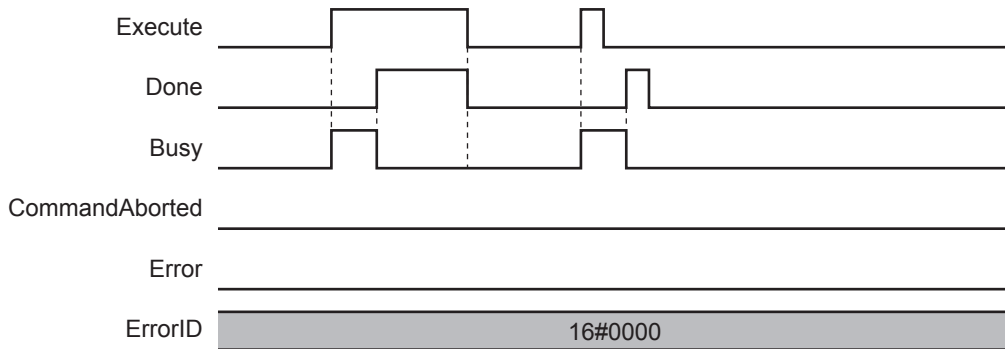
*4 作为写入对象的轴的“减速机使用”为“使用”且“计数模式”为“旋转模式”时，不写入“工件侧转1圈的移动量”的值。|环计数器上限值-环计数器下限值|的值作为“工件侧转1圈的移动量”处理。此外，作为读取对象的轴的“减速机使用”为“使用”且“计数模式”为“旋转模式”时，“工件侧转1圈的移动量”的值将读取|环计数器上限值-环计数器下限值|的值。

*5 作为写入对象的轴的轴种类为编码器轴或虚拟编码器轴时，不执行写入轴参数。作为读取对象的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，请参阅 □□ “NJ/NX 系列 CPU 单元 用户手册 运动控制篇(SBCE-363)” 或 □□ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

*6 Ver.1.10以上的CPU单元和Ver.1.12以上的Sysmac Studio组合时可指定。

时序图

执行MC_WriteAxisParameter(轴参数写入)指令后的时序图如下所示。



重启运动指令

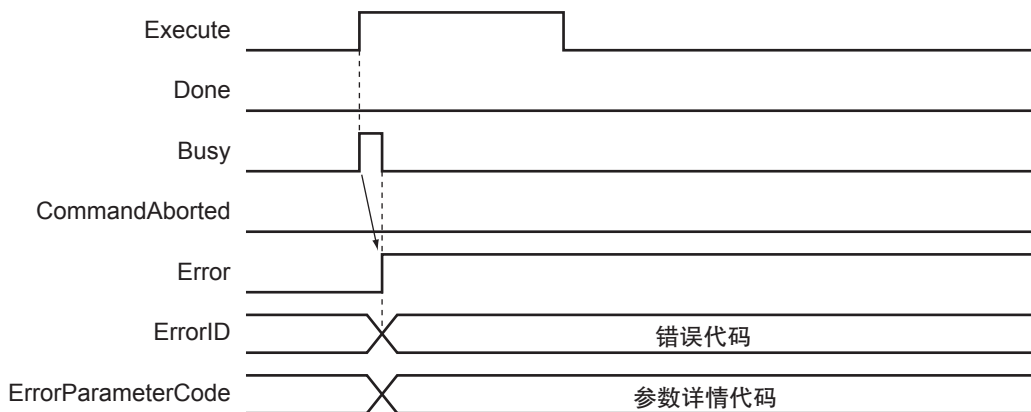
重启运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

多重启动运动指令

多重启动运动指令的详情，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

异常

在执行本指令中发生异常时，Error(错误)变为TRUE，不执行参数的写入。保持指令执行前的值。可查看ErrorID(错误代码)的输出值，了解发生异常的原因。此外，根据ErrorID(错误代码)，附属信息会被输出到ErrorParameterCode(参数详细代码)。



● 错误代码

关于本指令发生的异常，请参阅 □ “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

Ver.1.08以上的CPU单元和Ver.1.09以上的Sysmac Studio组合时可使用本指令。

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	在上升沿开始指令。

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	指令执行完毕时变为TRUE。
Busy	执行中	BOOL	TRUE, FALSE	接收指令后变为TRUE。
CommandAborted	执行中断	BOOL	TRUE, FALSE	指令中止时，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为正常。

* 请参阅“A-1 错误代码一览(P.A-2)”。

● 输出变量的反映时间

变量	变为TRUE的时间	变为FALSE的时间
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，中止本指令时	<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含有异常因素时	异常已解除时

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	-	指定读取对象轴。 ^{*1}
AxisParameter	轴参数	_sAXIS_PARAM	-	保存读取的值。 ^{*2}

*1. 请在使用在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型的用户定义变量。

● _sAXIS_PARAM

结构要素变量	参数名称	数据类型 ^{*1}	功能
UnitConversion	单位转换设定	_sAXIS_UNIT_CONVERSION_SETTINGS	将读取的单位转换设定的值保存到各要素变量中。
Operation	动作设定	_sAXIS_OPERATION_SETTINGS	将读取的动作设定的值保存到各要素变量中。
OtherOperation	扩展动作设定	_sAXIS_OTHER_OPERATION_SETTINGS	将读取的扩展动作设定的值保存到各要素变量中。
Limit	限制设定	_sAXIS_LIMIT_SETTINGS	将读取的限制设定的值保存到各要素变量中。
PosCount	位置计数设定	_sAXIS_POSITION_COUNT_SETTINGS	将读取的位置计数设定的值保存到各要素变量中。
Homing	原点复位设定	_sAXIS_HOMING_SETTINGS	将读取的原点复位设定的值保存到各要素变量中。
Reserved	(备用区域)	ARRAY[0..255] OF BYTE	-

*1. 数据类型的详情，请参阅  “指令详情(P.5-48)”。

功能说明

- 在Execute(启动)的上升沿, 将指定的Axis(轴)的轴参数读取至AxisParameter(轴参数)。
- 与轴变量的Cfg.AxEnable(轴使用的状态)无关, 本指令可以读取轴参数。



使用注意事项

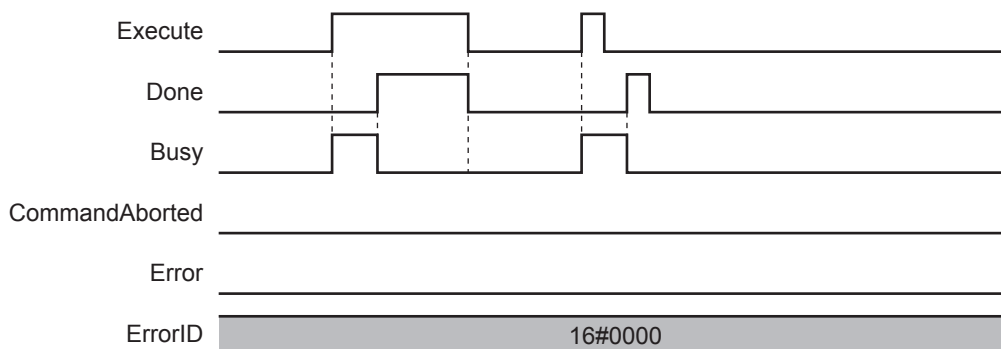
- 本指令可读取的值并非CPU单元的非易失性存储器中保存的值, 而是读取在执行了本指令时生效的轴参数。
例如, 使用MC_Write(写入MC设定)指令写入了轴参数时, 会读取写入的轴参数。
- 同时使用本指令和MC_WriteAxisParameter(轴参数写入)指令时, 请在确认MC_WriteAxisParameter(轴参数写入)指令的Done(完成)后再启动。
- 仅在本指令完成时读取至AxisParameter(轴参数)。之后, 不读取至AxisParameter(轴参数)。因此, 对AxisParameter(轴参数)执行了写入时, 会被覆盖。

作为读取对象的轴参数

本指令的读取对象轴参数, 请参阅 □ “作为写入及读取对象的轴参数(P.5-54)”。

时序图

执行MC_ReadAxisParameter(轴参数读取)指令后的时序图如下所示。



重启运动指令

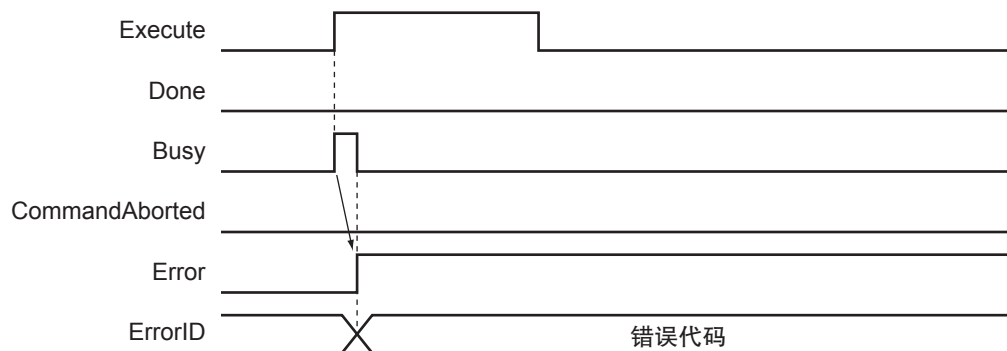
重启运动指令的详情, 请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

多重启动运动指令

多重启动运动指令的详情, 请参阅 □ “NJ/NX 系列 CPU 单元 用户手册 运动控制篇 (SBCE-363)” 或 □ “NY系列 工业用平板电脑/工业用台式电脑 用户手册 运动控制篇(SBCE-379)”。

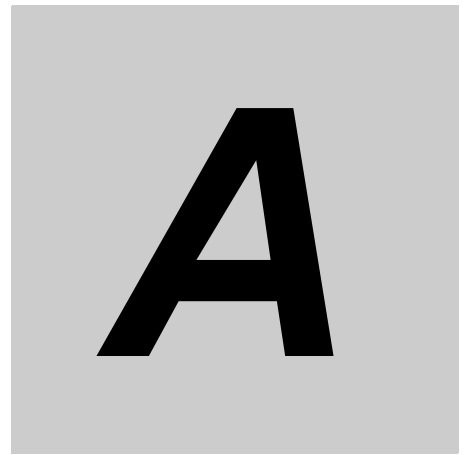
异常

在执行本指令中发生异常时，Error(错误)变为TRUE，不执行参数的读取。
可查看ErrorID(错误代码)的输出值，了解发生异常的原因。



● 错误代码

关于本指令发生的异常，请参阅 [□](#) “A-1 错误代码一览(P.A-2)”。



附录

下面对指令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-17
A-2-1 控制器异常说明的解释	A-17
A-2-2 异常说明	A-18
A-3 多重启动可否	A-83
A-3-1 轴的状态、轴组的状态	A-84
A-3-2 状态切换和多重启动的可否	A-86
A-4 版本相关信息	A-92

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)。

错误代码的详情，请参阅 □□ “A-2 错误代码详情(P.A-17)”。

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	监控信息
总	一般信息

事件代码栏()内表示限定发生该事件的CPU单元时的CPU单元的单元版本。

关于NJ/NX系列的所有事件代码，请参阅 □□ “NJ/NX系列故障诊断手册(SBCA-361)”。

事件代码	事件名称	内容	发生原因(推测原因)	重要程度					参考页
				全	部	轻	监	总	
34610000Hex	过程数据对象设定不足	PDO映射不正确	<ul style="list-style-type: none"> 未通过运动指令进行所需的PDO映射 对不带支持相应指令对象的对象设备执行了相应指令 针对映射了欧姆龙生产的EtherCAT编码器从站GX-EC02 □□的轴，作为触发条件指定Z相(_mcEncoderMark)，启动了运动指令 			○			P.A-18
54200000Hex	超过电子齿轮分子设定范围	运动控制指令的输入变量“RatioNumerator”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-18
54210000Hex	超过电子齿轮分母设定范围	运动控制指令的输入变量“RatioDenominator”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-19
54220000Hex	超过目标速度设定范围	运动控制指令的输入变量“Velocity”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-19
54230000Hex	超过加速度设定范围	运动控制指令的输入变量“Acceleration”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-19
54240000Hex	超过减速度设定范围	运动控制指令的输入变量“Deceleration”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-20
54250000Hex	超过跃度设定范围	运动控制指令的输入变量“Jerk”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-20
54270000Hex	超过扭矩倾斜设定范围	运动控制指令的输入变量“TorqueRamp”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-20

事件代码	事件名称	内容	发生原因(推测原因)	重要程度					参考页
				全	部	轻	监	总	
54280000Hex	超过主轴系数设定范围	运动控制指令的输入变量“Master Scaling”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-21
54290000Hex	超过从轴系数设定范围	运动控制指令的输入变量“SlaveScaling”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-21
542A0000Hex	超过标准速度设定范围	运动控制指令的输入变量“FeedVelocity”指定的参数超过范围	• 标准速度(输入变量“Feed Velocity”)保持为初始值(0)			○			P.A-21
542B0000Hex	超过缓存模式选择范围	运动控制指令的输入变量“BufferMode”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-22
542C0000Hex	超过坐标系选择范围	运动控制指令的输入变量“CoordSystem”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-22
542D0000Hex	超过圆弧插补模式选择范围	运动控制指令的输入变量“CircMode”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-22
542E0000Hex	超过方向选择范围	运动控制指令的输入变量“Direction”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-23
542F0000Hex	超过路径选择范围	运动控制指令的输入变量“PathChoice”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-23
54300000Hex	超过位置类型选择范围	运动控制指令的输入变量“Reference Type”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-23
54310000Hex	超过移动方法选择范围	运动控制指令的输入变量“MoveMode”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-24
54320000Hex	超过过渡模式选择范围	运动控制指令的输入变量“Transition Mode”指定的参数超过范围	• 指令的输入参数超过输入变量的范围 • “BufferMode”下指定“_mcAborting”、“_mcBuffered”且将“TransitionMode”指定为“_mcTMCornerSuperimposed”			○			P.A-24
54330000Hex	超过持续方法选择范围	变更了运动控制指令的输入变量“Continuous (Reserved)”的值	• 变更了输入变量“Continuous (Reserved)”的值			○			P.A-25
54340000Hex	超过加减法运算方法选择范围	运动控制指令的输入变量“Combine Mode”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-25
54350000Hex	超过开始同步条件指定范围	运动控制指令的输入变量“LinkOption”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-25
54360000Hex	主轴从轴相同	运动控制指令的输入变量“Master”和“Slave”指定的轴相同	• 指令的输入变量“Master”和“Slave”的参数相同			○			P.A-26

事件代码	事件名称	内容	发生原因(推测原因)	重要程度					参考页
				全	部	轻	监	总	
54370000Hex	主轴辅轴相同	运动控制指令的输入变量“Master”和“Auxiliary”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-26
54380000Hex	主轴/从轴轴号非升序	运动控制指令的输入变量“Master”和“Slave”指定的轴号非升序	<ul style="list-style-type: none"> 指令的输入变量“ReferenceType”指定“_mcLatestCommand”时, 指令的输入变量“Master”和“Slave”的参数非升序 			○			P.A-26
54390000Hex	凸轮表指定错误	运动控制指令的输入变量“CamTable”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入变量“CamTable”指定了非凸轮数据变量 			○			P.A-27
543A0000Hex	同步停止中	执行了运动控制的同步控制指令, 但并非可执行条件。	<ul style="list-style-type: none"> 执行了MC_CamOut(凸轮动作解除)指令, 但MC_CamIn(凸轮动作开始)指令未在运行 执行了MC_GearOut(齿轮动作解除)指令, 但MC_GearIn(齿轮动作开始)指令、MC_GearInPos(位置指定齿轮动作)指令未在运行 执行了MC_Phasing(主轴相对值相位补偿)指令, 但MC_CamIn(凸轮动作开始)指令、MC_GearIn(齿轮动作开始)指令、MC_GearInPos(位置指定齿轮动作)指令、MC_MoveLink(梯形模式凸轮)指令未在运行 			○			P.A-27
543B0000Hex	无法重启运动指令	重启了无法重启的运动控制指令	<ul style="list-style-type: none"> 重启了无法重启的运动控制指令 			○			P.A-28
543C0000Hex	无法多重启动运动指令	对相同对象(MC通用/轴/轴组)执行了多个无法同时执行的功能	<ul style="list-style-type: none"> 对相同对象(MC通用/轴)执行了多个无法同时执行的功能 			○			P.A-28
543D0000Hex	不符合轴类型	对编码器轴执行了动作指令	<ul style="list-style-type: none"> 对编码器轴执行了动作指令 			○			P.A-29
543E0000Hex	无法启动多轴协调动作中的指令	<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 执行了启用轴组时无法使用的机器人指令 	<ul style="list-style-type: none"> 对多轴协调动作中的轴或轴组执行了动作指令 对轴组有效状态下的轴组执行了MC_SetKinTransform指令 			○			P.A-29
543F0000Hex	启动轴组无效状态下的多轴协调指令	对轴组无效状态下的轴组启动了多轴协调指令	<ul style="list-style-type: none"> 对轴组无效状态下的轴组启动了多轴协调指令 对轴组无效状态下的轴组启动了以下指令 MC_MoveTimeAbsolute MC_SyncLinearConveyor MC_SyncOut MC_RobotJog 			○			P.A-30
54400000Hex	无法启用轴组	MC_GroupEnable(启用轴组)指令执行失败	<ul style="list-style-type: none"> 执行MC_GroupEnable(启用轴组)指令时, 构成轴中存在未处于停止状态的轴 执行MC_GroupEnable(启用轴组)指令时, 构成轴中存在正在执行MC_TouchProbe(启用外部锁定)指令的轴 			○			P.A-30

事件代码	事件名称	内容	发生原因(推测原因)	重要程度					参考页
				全	部	轻	监	总	
54410000Hex	无法运行(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中的轴执行了动作指令 对未建立EtherCAT的过程数据通信的轴执行了通过MC_Home(原点复位)指令或MC_HomeWithParameter(参数指定原点复位)指令进行的原点预设。 			○			P.A-31
54420000Hex	构成轴强制停止中错误	针对构成轴, 向已执行MC_Stop(强制停止)指令的轴组执行了动作指令	<ul style="list-style-type: none"> 针对构成轴, 向已执行MC_Stop(强制停止)指令的轴组执行了动作指令 			○			P.A-31
54430000Hex	多重启动运动指令数超限	通过缓存模式Buffered、Blending缓存的运动控制指令的缓存数超限	<ul style="list-style-type: none"> 正在执行的轴指令和缓存中的轴指令的总数超过了“2” 正在执行的轴组指令和缓存中的轴组指令的总数超过了“8” 			○			P.A-32
54440000Hex	移动量不足	多重启动/重启定位指令时, 无法执行以指定减速度或加速度指定的动作	<ul style="list-style-type: none"> 如果已将“加减速超限”设定为“作为异常停止”, 则无法在多重启动/重启定位指令时, 在指定减速度/加速度的条件下在目标位置停止 			○			P.A-32
54450000Hex	用于达到混合中继速度的移动量不足	用于中继速度加减速的移动量不足	<ul style="list-style-type: none"> 将“加减速超限”设定为“作为异常停止”时, 用于将当前指令加减速至中继速度的移动量不足 			○			P.A-33
54460000Hex	梯形模式凸轮等速移动量不足	主轴的等速移动量小于“0”	<ul style="list-style-type: none"> MC_MoveLink(梯形模式凸轮)指令下, 主轴的等速移动量小于“0” 			○			P.A-33
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)指令下, 从轴的“目标速度”较小, 因此达不到所需速度	<ul style="list-style-type: none"> MC_GearInPos(位置指定齿轮动作)指令下, 输入变量“Velocity(目标速度)”的值小于(启动指令时的主轴速度×齿轮比) 			○			P.A-33
54480000Hex	圆弧插补起点终点相同	MC_MoveCircular2D(2轴圆弧插补)指令下, 指定半径指定方式时起点和终点的位置相同。或指定通过点指定方式时起点、终点及通过点的位置相同	<ul style="list-style-type: none"> MC_MoveCircular2D(2轴圆弧插补)指令下, 指定了半径指定方式, 起点和终点的位置相同 MC_MoveCircular2D(2轴圆弧插补)指令下, 指定了通过点指定方式, 起点、终点及通过点的位置相同 			○			P.A-34
54490000Hex	超过圆弧插补中心点指定位置范围	MC_MoveCircular2D(2轴圆弧插补)指令下, 指定中心点指定方式时, 中心点的位置指定超过容许范围	<ul style="list-style-type: none"> MC_MoveCircular2D(2轴圆弧插补)指令下, 指定了中心点指定方式, 起点与中心点的距离、终点与中心点的距离之差超过了轴组设定的“中心点补偿容许率”指定的容许范围 			○			P.A-34
544A0000Hex	计数模式设定导致的指令启动异常	对计数模式已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"> 以旋转模式下无法使用的指令使用了计数模式已设定为旋转模式的轴 			○			P.A-35
544C0000Hex	超过参数选择范围	运动控制指令的输入变量“Parameter Number”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			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					参考页
				全	部	轻	监	总	
544D0000Hex	超过停止方法选择范围	运动控制指令的输入变量“StopMode”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-35
544E0000Hex	超过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”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-36
544F0000Hex	超过MC设定写入的设定范围	运动控制指令的输入变量“SettingValue”指定的参数超过范围	• 指令的输入参数超过输入变量的范围 • 参数指定和设定值的数据类型不一致			○			P.A-36
54500000Hex	超过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”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-36
54510000Hex	超过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”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-37
54530000Hex	无法重启运动指令(轴指定)	变更了重启运动控制指令时无法变更的输入变量“Axis”的参数	• 变更了重启时无法变更的输入变量的参数			○			P.A-37
54540000Hex	无法重启运动指令(缓存模式选择)	变更了重启运动控制指令时无法变更的输入变量“Buffer Mode”的参数	• 变更了重启时无法变更的输入变量的参数			○			P.A-37
54550000Hex	无法重启运动指令(方向选择)	变更了重启运动控制指令时无法变更的输入变量“Direction”的参数	• 变更了重启时无法变更的输入变量的参数			○			P.A-38
54560000Hex	无法重启运动指令(重复模式)	变更了重启运动控制指令时无法变更的输入变量“Periodic”的参数	• 变更了重启时无法变更的输入变量的参数			○			P.A-38
54570000Hex	无法重启运动指令(轴组指定)	变更了重启运动控制指令时无法变更的输入变量“Axes Group”的参数	• 变更了重启时无法变更的输入变量的参数			○			P.A-38
54580000Hex	无法重启运动指令(跳动设定)	变更了重启运动控制指令时无法变更的输入变量“Jerk”的参数	• 变更了重启时无法变更的输入变量的参数			○			P.A-39
54590000Hex	无法重启运动指令(主轴)	变更了重启运动控制指令时无法变更的输入变量“Master”的参数	• 变更了重启时无法变更的输入变量的参数			○			P.A-39
545A0000Hex	无法重启运动指令(MasterOffset)	变更了重启运动控制指令时无法变更的输入变量“Master Offset”的参数	• 变更了重启时无法变更的输入变量的参数			○			P.A-40
545B0000Hex	无法重启运动指令(Master Scaling)	变更了重启运动控制指令时无法变更的输入变量“Master Scaling”的参数	• 变更了重启时无法变更的输入变量的参数			○			P.A-40

事件代码	事件名称	内容	发生原因(推测原因)	重要程度					参考页
				全	部	轻	监	总	
545C0000Hex	无法重启运动指令 (MasterStart Distance)	变更了重启运动控制指令时无法变更的输入变量“MasterStart Distance”的参数	<ul style="list-style-type: none"> 变更了重启时无法变更的输入变量的参数 			○			P.A-40
545D0000Hex	无法重启运动指令 (Continuous)	变更了重启运动控制指令时无法变更的输入变量“Continuous”的参数	<ul style="list-style-type: none"> 变更了重启时无法变更的输入变量的参数 			○			P.A-41
545E0000Hex	无法重启运动指令 (MoveMode)	变更了重启运动控制指令时无法变更的输入变量“MoveMode”的参数	<ul style="list-style-type: none"> 变更了重启时无法变更的输入变量的参数 			○			P.A-41
545F0000Hex	辅轴指定错误	运动控制指令的输入变量“Auxiliary”指定的轴不存在	<ul style="list-style-type: none"> 指令的输入变量“Auxiliary”指定的轴不存在的变量 			○			P.A-41
54600000Hex	轴指定错误	运动控制指令的输入变量“Axis”指定的轴不存在	<ul style="list-style-type: none"> 指令的输入变量“Axis”指定的轴不存在的变量 			○			P.A-42
54610000Hex	轴组指定错误	运动控制指令的输入变量“AxesGroup”指定的轴组不存在或不是使用轴组	<ul style="list-style-type: none"> 指令的“AxesGroup”指定的轴组不存在的变量 指令的“AxesGroup”指定的轴组未设定为使用轴组 			○			P.A-42
54620000Hex	主轴指定错误	运动控制指令的输入变量“Master”指定的轴错误	<ul style="list-style-type: none"> 指令的输入变量“Master”指定的轴不存在的变量 MC_Phasing(主轴相对值相位补偿)指令时, 输入变量“Master”指定的轴不是同步主轴 分配主轴和从轴的任务不同 			○			P.A-43
54630000Hex	无法重启运动指令 (SlaveOffset)	变更了重启运动控制指令时无法变更的输入变量“Slave Offset”的参数	<ul style="list-style-type: none"> 变更了重启时无法变更的输入变量的参数 			○			P.A-43
54640000Hex	无法重启运动指令 (SlaveScaling)	变更了重启运动控制指令时无法变更的输入变量“Slave Scaling”的参数	<ul style="list-style-type: none"> 变更了重启时无法变更的输入变量的参数 			○			P.A-44
54650000Hex	无法重启运动指令 (StartPosition)	变更了重启运动控制指令时无法变更的输入变量“Start Position”的参数	<ul style="list-style-type: none"> 变更了重启时无法变更的输入变量的参数 			○			P.A-44
54660000Hex	原点未确定状态下的指令启动异常	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"> 原点未确定状态下执行了高速原点复位 向包含原点未确定状态的构成轴执行了插补指令 向包含原点未确定状态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-45

事件代码	事件名称	内容	发生原因(推测原因)	重要程度					参考页
				全	部	轻	监	总	
54670000Hex	无法重启运动指令 (位置类型)	变更了重启运动控制指令时无法变更的输入变量“Reference Type”的参数	• 变更了重启时无法变更的输入变量的参数			○			P.A-45
54680000Hex	未使用轴指定(主轴)	运动控制指令指定的主轴为未使用轴	• 运动控制指令指定的主轴为未使用轴			○			P.A-46
54690000Hex	超过起始位置设定范围	运动控制指令的输入变量“FirstPosition”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-46
546A0000Hex	超过终止位置设定范围	运动控制指令的输入变量“LastPosition”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-46
546B0000Hex	起始位置/终止位置、大小关系错误(线性模式)	运动控制指令的输入变量“LastPosition”指定的参数值小于输入变量“FirstPosition”指定的参数值	• 计数模式为线性模式时,指令的输入参数“LastPosition”的值小于“FirstPosition”的值			○			P.A-47
546C0000Hex	超过主轴同步位置设定范围	运动控制指令的输入变量“MasterSync Position”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-47
546D0000Hex	超过从轴同步位置设定范围	运动控制指令的输入变量“SlaveSync Position”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-47
546E0000Hex	触发输入条件的锁定ID重复	多个指令已重复运动控制指令指定的锁定ID	<ul style="list-style-type: none"> MC_TouchProbe(启用外部锁定)指令、MC_MoveLink(梯形模式凸轮)指令、MC_MoveFeed(中断标准定位)指令已同时使用相同的锁定ID 试图通过MC_AbortTrigger(不启用外部锁定)指令中止正在以非MC_TouchProbe(启用外部锁定)指令使用的锁定 			○			P.A-48
546F0000Hex	超过跃度超调值范围	运动控制指令的输入变量“JerkFactor”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-48
54700000Hex	超过加减速超调值范围	运动控制指令的输入变量“AccFactor”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-49
54710000Hex	超过起始位置方式指定范围	运动控制指令的输入变量“StartMode”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-49
54720000Hex	无法重启运动指令(起始位置方式)	变更了重启运动控制指令时无法变更的输入变量“StartMode”的参数	• 变更了重启时无法变更的输入变量的参数			○			P.A-49
54740000Hex	未使用轴指定(辅轴)	运动控制指令的输入变量“Auxiliary”指定的轴为未使用轴	• 指令的“Auxiliary”指定的轴为未使用轴			○			P.A-50
54750000Hex	位置指定齿轮指定值异常	无法以运动控制指令输入的速度/加速度/减速度进行同步动作	• 无法以指令输入的速度/加速度/减速度进行指定的同步动作			○			P.A-50

事件代码	事件名称	内容	发生原因(推测原因)	重要程度					参考页
				全	部	轻	监	总	
54760000Hex	位置指定齿轮主轴零速	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-50
54780000Hex	超过目标位置设定范围	运动控制指令的输入变量“Position”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 对于旋转模式的轴，目标位置超过环设定的范围 			○			P.A-51
54790000Hex	超过移动距离范围	运动控制指令的输入变量“Distance”指定的参数超过范围或加上“Distance”值后的目标位置超过范围	<ul style="list-style-type: none"> 将指令输入参数的绝对值转换为脉冲单位时超过40位的范围 对于线性模式的轴，将加上移动距离后的目标位置转换为脉冲单位时超过带符号40位的范围 			○			P.A-51
547A0000Hex	超过凸轮表起点位置设定范围	运动控制指令的输入变量“StartPosition”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-52
547B0000Hex	超过凸轮动作(主轴跟踪)开始位置设定范围	运动控制指令的输入变量“MasterStartDistance”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-52
547C0000Hex	圆弧插补半径指定异常	MC_MoveCircular2D(2轴圆弧插补)指令下，指定半径指定方式时无法以指定的半径创建圆弧的轨迹	<ul style="list-style-type: none"> MC_MoveCircular2D(2轴圆弧插补)指令下，指定了半径指定方式，无法以指定的半径创建圆弧的轨迹 			○			P.A-52
547D0000Hex	圆弧插补半径溢出	MC_MoveCircular2D(2轴圆弧插补)指令下，通过点指定方式/中心点指定方式时，圆弧的半径超过最大值	<ul style="list-style-type: none"> MC_MoveCircular2D(2轴圆弧插补)指令下，通过点指定方式/中心点指定方式时，将圆弧的半径转换为脉冲单位时超过40位的范围 			○			P.A-53
547E0000Hex	超过圆弧轴指定范围	运动控制指令的输入变量“CircAxes”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 “CircAxes”指定的轴未包含在轴组设定的构成轴中 “CircAxes”的2轴已指定相同轴 			○			P.A-53
547F0000Hex	辅轴/从轴、轴号非升序	运动控制指令指定的输入变量“Auxiliary”和“Slave”的参数值未以升序排列	<ul style="list-style-type: none"> 指令的输入变量“Auxiliary”和“Slave”的参数非升序 			○			P.A-54
54800000Hex	凸轮表属性更新中数据升序异常	判定有效数据数计算中相位非升序。或判定计算后有效数据数为“0”	<ul style="list-style-type: none"> 判定有效数据数计算中相位非升序 判定计算后有效数据数为“0” 			○			P.A-54
54810000Hex	超过MC设定写入的对象范围	运动控制指令的输入变量“Target”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-54
54820000Hex	超过主轴移动距离指定范围	运动控制指令的输入变量“MasterDistance”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-55

事件代码	事件名称	内容	发生原因(推测原因)	重要程度					参考页
				全	部	轻	监	总	
54830000Hex	超过主轴加速移动距离指定范围	运动控制指令的输入变量“Master DistanceInACC”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-55
54840000Hex	超过主轴减速移动距离指定范围	运动控制指令的输入变量“Master DistanceInDEC”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-55
54870000Hex	超过执行模式选择范围	运动控制指令的输入变量“Execution Mode”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-56
54880000Hex	超过轴间偏差容许值范围	运动控制指令的输入变量“Permitted Deviation”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-56
54890000Hex	通过点位置/中心位置/超过半径指定范围	运动控制指令的输入变量“AuxPoint”指定的参数超过范围	• 通过点指定或中心指定时，将“AuxPoint”的值转换为脉冲单位时超过带符号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过40位的范围			○			P.A-56
548A0000Hex	超过终点指定范围	运动控制指令的输入变量“EndPoint”指定的参数超过范围	• 将指令的输入参数转换为脉冲单位时，已超过带符号40位的范围			○			P.A-57
548B0000Hex	超过从轴移动距离指定范围	运动控制指令的输入变量“Slave Distance”指定的参数超过范围	• 将指令的输入参数转换为脉冲单位时，已超过40位的范围			○			P.A-57
548C0000Hex	超过相位补偿量范围	运动控制指令的输入变量“PhaseShift”指定的参数超过范围	• 将指令输入参数的绝对值转换为脉冲单位时超过40位的范围			○			P.A-57
548D0000Hex	超过标准距离范围	运动控制指令的输入变量“Feed Distance”指定的参数超过范围	• 将指令输入参数的绝对值转换为脉冲单位时超过40位的范围			○			P.A-58
548E0000Hex	辅轴/从轴相同	运动控制指令的输入变量“Auxiliary”和“Slave”指定的轴相同	• 指令的输入变量“Auxiliary”和“Slave”的参数相同			○			P.A-58
548F0000Hex	超过相对位置选择范围	运动控制指令的输入变量“Relative”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-58
54900000Hex	超过凸轮过渡指定选择范围	运动控制指令的输入变量“Cam Transition”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			P.A-59
54910000Hex	超过同步控制解除模式选择范围	运动控制指令的输入变量“OutMode”指定的参数超过范围	• 指令的输入参数超过输入变量的范围			○			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					参考页
				全	部	轻	监	总	
54920000Hex	无法执行启用外部锁定指令	对于编码器轴，输入变量“StopMode”指定为“_mcImmediateStop(立即停止)”，执行了驱动模式的MC_TouchProbe(启用外部锁定)指令	<ul style="list-style-type: none"> 对于编码器轴，输入变量“StopMode”指定为“_mcImmediateStop(立即停止)”，执行了驱动模式的MC_TouchProbe(启用外部锁定)指令 			○			P.A-59
54930000Hex	超过主轴偏移设定范围	运动控制指令的输入变量“MasterOffset”指定的参数超过范围	<ul style="list-style-type: none"> 将指令的输入参数转换为脉冲单位时，已超过带符号40位的范围 			○			P.A-60
54940000Hex	超过从轴偏移设定范围	运动控制指令的输入变量“SlaveOffset”指定的参数超过范围	<ul style="list-style-type: none"> 将指令的输入参数转换为脉冲单位时，已超过带符号40位的范围 			○			P.A-60
54950000Hex	超过指令当前位置计数选择范围	运动控制指令的输入变量“CmdPosMode”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-60
54960000Hex	超过主轴齿轮比分子范围	运动控制指令的输入变量“RatioNumeratorMaster”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-61
54970000Hex	超过主轴齿轮比分母范围	运动控制指令的输入变量“RatioDenominatorMaster”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-61
54980000Hex	超过辅轴齿轮比分子范围	运动控制指令的输入变量“RatioNumeratorAuxiliary”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-61
54990000Hex	超过辅轴齿轮比分母范围	运动控制指令的输入变量“RatioDenominatorAuxiliary”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-62
549A0000Hex	超过主轴位置类型选择范围	运动控制指令的输入变量“ReferenceTypeMaster”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-62
549B0000Hex	超过辅轴位置类型选择范围	运动控制指令的输入变量“ReferenceTypeAuxiliary”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-62
549C0000Hex	超过目标位置环计数器范围	已执行指令的目标位置超过环计数器的范围，因此无法动作	<ul style="list-style-type: none"> 设定为环计数器范围内不含0时，执行了高速原点复位 			○			P.A-63
549D0000Hex (Ver.1.01以上)	超过轴组构成轴设定范围	运动控制指令的输入变量“Axes”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 分配轴组构成轴的任务不同 			○			P.A-63
549E0000Hex (Ver.1.04以上)	超过轴使用设定范围	运动控制指令的输入变量“AxisUse”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-64
57000000Hex (Ver.1.03以上)	超过原点复位参数设定范围	运动控制指令的输入变量“HomingParameter”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-64

事件代码	事件名称	内容	发生原因(推测原因)	重要程度					参考页
				全	部	轻	监	总	
57020000Hex (Ver.1.04以上)	轴未使用切换异常	向非处于停止状态或处于指令速度饱和状态的轴执行了MC_ChangeAxisUse(轴使用变更)指令	<ul style="list-style-type: none"> 向非处于停止状态或处于指令速度饱和状态的轴执行了MC_ChangeAxisUse(轴使用变更)指令 			○			P.A-65
57030000Hex (Ver.1.06以上)	无法进行轴使用切换	试图超出使用实轴最大数量或使用运动控制伺服轴最大数量MC_ChangeAxisUse(轴使用变更)指令	<ul style="list-style-type: none"> 试图超出使用实轴最大数量执行MC_ChangeAxisUse(轴使用变更)指令 试图超出使用运动控制伺服轴最大数量执行MC_ChangeAxisUse(轴使用变更)指令 			○			P.A-65
57200000Hex (Ver.1.04以上)	轴使用切换时运动控制参数设定异常	切换为使用轴的轴的运动控制参数设定错误	<ul style="list-style-type: none"> MC_ChangeAxisUse(轴使用变更)指令下,从未使用轴切换到使用轴的轴的运动控制参数设定错误 下载运动控制参数设定过程中电源断开 非易失性存储器的故障或非易失性存储器的寿命 			○			P.A-66
57210000Hex (Ver.1.04以上)	未设定轴使用切换时所需的过程数据对象	未设定切换为使用轴的轴类型的所需对象	<ul style="list-style-type: none"> 切换为使用轴的轴类型的所需对象未设定PDO映射 下载运动控制参数设定过程中电源断开 非易失性存储器的故障或非易失性存储器的寿命 对[轴使用]设定为[未使用轴(无法切换为使用轴)]的轴执行了MC_ChangeAxisUse(轴使用变更)指令 			○			P.A-67
572F0000Hex (Ver.1.06以上)	无法多重启动运动指令(主轴)	变更了多重启动指令时无法变更的输入输出变量“Master”	<ul style="list-style-type: none"> 变更了多重启动指令时无法变更的输入输出变量“Master” 			○			P.A-67
57300000Hex (Ver.1.06以上)	无法多重启动运动指令(位置类型选择)	变更了多重启动指令时无法变更的输入输出变量“ReferenceType”	<ul style="list-style-type: none"> 变更了多重启动指令时无法变更的输入输出变量“ReferenceType” 			○			P.A-68
573A0000Hex (Ver.1.08以上)	轴参数无法写入	对未使用轴以外的轴启动了相应指令	<ul style="list-style-type: none"> 对使用轴或未创建轴启动了相应指令 			○			P.A-68
573B0000Hex (Ver.1.08以上)	轴参数超过设定范围	运动控制指令的输入变量“AxisParameter”指定的参数超过有效范围	<ul style="list-style-type: none"> 指令的输入变量“AxisParameter”指定的参数超过输入变量的范围 			○			P.A-69
573C0000Hex (Ver.1.08以上)	凸轮属性超过设定范围	运动控制指令的输入变量“CamProperty”指定的参数超过有效范围	<ul style="list-style-type: none"> 指令的输入变量“CamProperty”指定的参数超过输入变量的范围 			○			P.A-70
573D0000Hex (Ver.1.08以上)	凸轮节点超过设定范围	运动控制指令的输入变量“CamNodes”指定的参数超过有效范围	<ul style="list-style-type: none"> 指令的输入变量“CamNodes”指定的参数超过输入变量的范围 			○			P.A-70

事件代码	事件名称	内容	发生原因(推测原因)	重要程度					参考页
				全	部	轻	监	总	
573E0000Hex (Ver.1.08以上)	凸轮节点类型指定错误	运动控制指令的输入变量“CamNodes”指定的参数非_sMC_CAM_NODE型数组变量	<ul style="list-style-type: none"> 指令的输入变量“CamNodes”指定的参数非_sMC_CAM_NODE型数组变量 			○			P.A-71
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”指定的参数数组变量中元素编号0的Phase(主轴相位)值为“0”	<ul style="list-style-type: none"> 指令的输入变量“CamNodes”指定的参数数组变量中元素编号0的Phase(主轴相位)值为“0” 			○			P.A-71
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”指定的参数数组变量中Phase的值未按元素编号顺序升序排列	<ul style="list-style-type: none"> 指令的输入变量“CamNodes”指定的参数数组变量中Phase(主轴相位)的值未按元素编号顺序升序排列。或者,舍去有效位第8位后的结果为非升序状态 			○			P.A-71
57410000Hex (Ver.1.08以上)	凸轮表生成数据点数过多	生成的凸轮数据数超过了运动控制指令的输入变量“CamTable”指定的凸轮数据变量的数组元素数	<ul style="list-style-type: none"> 生成的凸轮表的凸轮数据数超过了指令的输入变量“CamTable”指定的凸轮数据变量的数组元素数 			○			P.A-72
57420000Hex (Ver.1.08以上)	凸轮表生成位移溢出	生成的凸轮表的Distance超过了REAL型可表示的范围	<ul style="list-style-type: none"> 生成的凸轮表的Distance超过了REAL型可表示的范围 			○			P.A-72
57430000Hex (Ver.1.08以上)	生成中止凸轮表的使用	相应指令的输入变量“CamTable”指定了被中止生成的凸轮数据变量	<ul style="list-style-type: none"> 相应指令的输入变量“CamTable”指定了因MC_GenerateCamTable(凸轮表生成)指令异常而被中止生成的凸轮数据变量 			○			P.A-73
57490000Hex (Ver.1.10以上)	执行ID超过设定范围	运动控制指令的输入变量“ExecID”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入变量“ExecID”指定的参数超过输入变量的范围 			○			P.A-73
574A0000Hex (Ver.1.10以上)	位置偏置超过范围	运动控制指令的输入变量“Offset Position”指定的参数超过范围	<ul style="list-style-type: none"> 将位置偏置转换为脉冲单位时,已超过带符号40位的范围 			○			P.A-74
574B0000Hex (Ver.1.10以上)	PDS状态变化指令选择超过范围	运动控制指令的输入变量“Transition Cmd”指定的参数超过范围	<ul style="list-style-type: none"> 指令的输入参数超过输入变量的范围 			○			P.A-74
574C0000Hex (Ver.1.13以上)	无法启动单轴位置控制轴运动指令	执行了无法对单轴位置控制轴执行的轴运动指令	<ul style="list-style-type: none"> 执行了无法对单轴位置控制轴执行的轴运动指令 			○			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					参考页
				全	部	轻	监	总	
6440000Hex	目标位置 正方向软件 超限	指定的位置超过正方向软件限制的范围	<ul style="list-style-type: none"> 指令的输入变量“Position”指定的参数超过正方向软件限制的范围 起始位置超过正方向软件限制的范围，执行了指定与软件限制的范围相反方向的动作的指令 已指定通过点的MC_MoveCircular2D(2轴圆弧插补)指令的输入变量“AuxPoint”指定的参数超过正方向软件限制的范围 			○			P.A-75
6441000Hex	目标位置 负方向软件 超限	指定的位置超过负方向软件限制的范围	<ul style="list-style-type: none"> 指令的输入变量“Position”指定的参数超过负方向软件限制的范围 起始位置超过负方向软件限制的范围，执行了指定与软件限制的范围相反方向的动作的指令 已指定通过点的MC_MoveCircular2D(2轴圆弧插补)指令的输入变量“AuxPoint”指定的参数超过负方向软件限制的范围 			○			P.A-76
6442000Hex	指令位置溢出/正在发生下溢	指令位置溢出/正在发生下溢时，执行了定位、溢出/下溢方向的指令或无法确定方向的指令	<ul style="list-style-type: none"> 指令位置溢出/正在发生下溢时进行了以下操作 执行了定位的指令 执行了溢出/下溢方向的连续控制指令 执行了无法确定方向的指令(同步功能、扭矩控制) 			○			P.A-76
6443000Hex	正方向限制输入中	正方向限制输入为“ON”的状态下执行了正方向动作的指令	<ul style="list-style-type: none"> 正方向限制输入为“ON”的状态下，执行了正方向动作的指令；或正方向限制输入为“ON”的状态下，执行了不指定动作方向的指令。正方向限制输入为“ON”的状态下，执行了轴组动作指令 			○			P.A-77
6444000Hex	负方向限制输入中	负方向限制输入为“ON”的状态下，执行了负方向动作的指令	<ul style="list-style-type: none"> 负方向限制输入为“ON”的状态下，执行了负方向动作的指令；或负方向限制输入为“ON”的状态下，执行了不指定动作方向的指令。负方向限制输入为“ON”的状态下，执行了轴组动作指令 			○			P.A-77
7422000Hex	伺服主电路电源OFF状态	伺服驱动器的主电路电源为“OFF”的状态下，执行了伺服ON	<ul style="list-style-type: none"> 伺服驱动器的主电路电源为“OFF”状态下，执行了伺服ON 			○			P.A-78
5722000Hex (Ver.1.06以上)	反馈位置正在发生溢出/下溢	反馈位置正在发生溢出/下溢时启动了无法执行的指令	<ul style="list-style-type: none"> 反馈位置正在发生溢出或下溢时启动了无法执行的指令 				○		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					参考页
				全	部	轻	监	总	
57230000Hex (Ver.1.06以上)	开关结构体的轨段编号超过设定范围	运动控制指令的输入输出变量 “Switches”指定的 “TrackNumber”值超过范围	• 指令的输入输出变量指定的结构体型变量的结构要素值超过范围				○		P.A-78
57240000Hex (Ver.1.06以上)	开关结构体的ON开始位置超过设定范围	运动控制指令的输入输出变量 “Switches”指定的 “FirstOnPosition”值超过范围	• 指令的输入输出变量指定的结构体型变量的结构要素值超过范围				○		P.A-79
57250000Hex (Ver.1.06以上)	开关结构体的ON结束位置超过设定范围	运动控制指令的输入输出变量 “Switches”指定的 “LastOnPosition”值超过范围	• 指令的输入输出变量指定的结构体型变量的结构要素值超过范围				○		P.A-79
57260000Hex (Ver.1.06以上)	开关结构体的方向选择超过范围	运动控制指令的输入输出变量 “Switches”指定的 “AxisDirection”值超过范围	• 指令的输入输出变量指定的结构体型变量的结构要素值超过范围				○		P.A-79
57270000Hex (Ver.1.06以上)	开关结构体的开关模式选择超过范围	运动控制指令的输入输出变量 “Switches”指定的 “CamSwitchMode”值超过范围	• 指令的输入输出变量指定的结构体型变量的结构要素值超过范围				○		P.A-80
57280000Hex (Ver.1.06以上)	开关结构体的ON时间超过设定范围	运动控制指令的输入输出变量 “Switches”指定的 “Duration”值超过范围	• 指令的输入输出变量指定的结构体型变量的结构要素值超过范围				○		P.A-80
57290000Hex (Ver.1.06以上)	轨段选项结构体的ON时刻补偿超过设定范围	运动控制指令的输入输出变量 “TrackOptions”指定的 “OnCompensation”值超过范围	• 指令的输入输出变量指定的结构体型变量的结构要素值超过范围				○		P.A-80
572A0000Hex (Ver.1.06以上)	轨段选项结构体的OFF时刻补偿超过设定范围	运动控制指令的输入输出变量 “TrackOptions”指定的 “OffCompensation”值超过范围	• 指令的输入输出变量指定的结构体型变量的结构要素值超过范围				○		P.A-81
572B0000Hex (Ver.1.06以上)	开关结构体型变量的数组元素数超过范围	运动控制指令的输入输出变量 “Switches”指定的结构体型变量的数组元素数超过范围	• 指令的输入输出变量指定的结构体型变量的数组元素数超过范围				○		P.A-81
572C0000Hex (Ver.1.06以上)	输出信号结构体型变量的数组元素数超过范围	运动控制指令的输入输出变量 “Outputs”指定的结构体型变量的数组元素数超过范围	• 指令的输入输出变量指定的结构体型变量的数组元素数超过范围				○		P.A-81
572D0000Hex (Ver.1.06以上)	轨段选项结构体型变量的数组元素数超过范围	运动控制指令的输入输出变量 “TrackOptions”指定的结构体型变量的数组元素数超过范围	• 指令的输入输出变量指定的结构体型变量的数组元素数超过范围				○		P.A-82

事件代码	事件名称	内容	发生原因(推测原因)	重要程度					参考页
				全	部	轻	监	总	
572E0000Hex (Ver.1.06以上)	输出信号与 轨段选项的 数组元素数 不一致	运动控制指令的输入 输出变量 “Outputs” 与 “TrackOptions” 指定的结构体型变量 的数组元素数不一致	<ul style="list-style-type: none"> 指令的输入输出变量指定的输出信号结构体型变量与机架选项结构体型变量的数组元素数不一致 				○		P.A-82
57310000Hex (Ver.1.06以上)	开关结构体 的同一轨段 指定数超过 范围	运动控制指令的输入 输出变量 “Switches” 指定的 “TrackNumber” 的同 一轨段编号数量超过 范围	<ul style="list-style-type: none"> 指令的输入输出变量指定的开关结构体型变量的 “TrackNumber” 指定的同一轨 段编号数量超过了1个轨段可 指定的个数范围 				○		P.A-82

A-2 错误代码详情

A-2-1 控制器异常说明的解释

各异常的说明使用的表的各项目的含义在[]内表示。

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]					
发生源	[异常(事件)发生的部位]		发生源详情	[发生源的详情]	检测时间	[异常检测的时间]
异常的属性	重要程度	[对控制产生影响的程度] ^{*1}	恢复方法	[恢复方法] ^{*2}	日志类别	[待保存的日志种类] ^{*3}
发生后的影响	用户程序	[用户程序的执行状态] ^{*4}	动作	[异常(事件)发生时的动作相关特别说明]		
LED	内置EtherNet/IP端口用LED、内置EtherCAT端口用LED的显示状态。仅EtherCAT主站功能模块、EtherNet/IP功能模块时记载发生源]					
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、导致异常的系统定义变量的变量名称和数据类型、名称]					
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	[异常(事件)的发生原因、处理措施及防止再次发生的方法]					
附加信息	[Sysmac Studio/NS系列显示器中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. 为以下任意一个。

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
 监控信息
 一般信息

*2. 为以下任意一个。

自动恢复：排除原因后自动恢复正常。
 异常解除：排除原因后通过执行异常解除恢复正常。
 重新接通电源：排除原因后通过重新接通电源恢复正常。
 控制器复位：排除原因后通过控制器复位恢复正常。
 基于发生原因：取决于发生原因。

*3. 为以下任意一个。

系统日志：系统事件日志
 访问日志：访问事件日志

*4. 为以下任意一个。

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. 关于显示器故障诊断器的适用范围，请参阅 □ “NJ/NX系列 故障诊断手册(SBCA-361)”的附录。



使用注意事项

对于Ver.1.10以上的CPU单元，以_MC_AX[*]开头的变量名可能为_MC1_AX[*]、_MC2_AX[*]。同样，以_MC_GRP[*]开头的变量名可能为_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)的详细内容。

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	3461000HHex	
内容	PDO映射不正确					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	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		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	未通过运动指令进行所需的PDO映射		请以相应指令进行必要的PDO映射。 必要的PDO映射请参阅相应指令的“功能说明”		请以需使用的指令进行必要的PDO映射。 以各指令进行的必要的PDO映射(伺服驱动器的设定)请参阅《NJ/NX系列用户手册 运动控制篇》	
	对不带支持相应指令对象的对象设备执行了相应指令		存在不带支持相应指令功能的对象设备。 请参阅对象设备的手册，确认是否支持相应指令，修正程序以避免执行不支持的指令		请参阅对象设备的手册，编写程序以避免执行不支持的指令	
	针对映射了欧姆龙生产的EtherCAT编码器从站GX-EC02□□的轴，作为触发条件指定Z相(_mcEncoderMark)，启动了运动指令		请针对映射了欧姆龙生产的EtherCAT编码器从站GX-EC02□□的轴，将外部输入(_mcEXT)用于触发条件		请针对映射了欧姆龙生产的EtherCAT编码器从站GX-EC02□□的轴，将外部输入(_mcEXT)用于触发条件	
附加信息	无					
注意事项/备注	无					

事件名称	超过电子齿轮分子设定范围			事件代码	5420000HHex	
内容	运动控制指令的输入变量“RatioNumerator”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过目标速度设定范围			事件代码	54220000Hex	
内容	运动控制指令的输入变量“Velocity”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时: 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时: 相应轴组无法动作。在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过加速度设定范围			事件代码	54230000Hex	
内容	运动控制指令的输入变量“Acceleration”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时: 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时: 相应轴组无法动作。在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过减速度设定范围			事件代码	54240000Hex	
内容	运动控制指令的输入变量“Deceleration”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时： 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时： 相应轴组无法动作。在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过跃度设定范围			事件代码	54250000Hex	
内容	运动控制指令的输入变量“Jerk”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时： 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时： 相应轴组无法动作。在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过扭矩倾斜设定范围			事件代码	54270000Hex	
内容	运动控制指令的输入变量“TorqueRamp”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过主轴系数设定范围		事件代码	54280000Hex		
内容	运动控制指令的输入变量“MasterScaling”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过从轴系数设定范围		事件代码	54290000Hex		
内容	运动控制指令的输入变量“SlaveScaling”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过标准速度设定范围		事件代码	542A0000Hex		
内容	运动控制指令的输入变量“FeedVelocity”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	标准速度(输入变量“Feed Velocity”)保持为初始值(0)		请为标准速度(输入变量“Feed Velocity”)指定正值		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过缓存模式选择范围			事件代码	542B0000Hex	
内容	运动控制指令的输入变量“BufferMode”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时： 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时： 相应轴组无法动作。在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过坐标系选择范围			事件代码	542C0000Hex	
内容	运动控制指令的输入变量“CoordSystem”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴组无法动作 在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴组无法动作 在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过方向选择范围		事件代码	542E0000Hex		
内容	运动控制指令的输入变量“Direction”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过路径选择范围		事件代码	542F0000Hex		
内容	运动控制指令的输入变量“PathChoice”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴组无法动作 在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过位置类型选择范围		事件代码	54300000Hex		
内容	运动控制指令的输入变量“ReferenceType”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	MC通用/轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		正在发生MC通用轻度故障	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过移动方法选择范围			事件代码	54310000Hex	
内容	运动控制指令的输入变量“MoveMode”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时： 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时： 相应轴组无法动作。在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
	_MC_GRP[*].MFAultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过过渡模式选择范围			事件代码	54320000Hex	
内容	运动控制指令的输入变量“TransitionMode”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴组无法动作 在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFAultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
	“BufferMode”下指定 “_mcAborting”、“_mcBuffered”， 且将“TransitionMode”指定为 “_mcTMCornerSuperimposed”		“BufferMode”指定为 “_mcAborting”、“_mcBuffered” 时，请将“TransitionMode”指定 为“_mcTMNone”， “TransitionMode”指定为 “_mcTMCornerSuperimposed”时， 请将“BufferMode”指定为 “_mcBlendingLow”、 “_mcBlendingPrevious”、 “_mcBlendingNext”、 “_mcBlendingHigh”		“BufferMode”指定为 “_mcAborting”、“_mcBuffered” 时，请将“TransitionMode”指定 为“_mcTMNone”， “TransitionMode”指定为 “_mcTMCornerSuperimposed”时， 请将“BufferMode”指定为 “_mcBlendingLow”、 “_mcBlendingPrevious”、 “_mcBlendingNext”、 “_mcBlendingHigh”	
附加信息	无					
注意事项/备注	无					

事件名称	超过持续方法选择范围			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					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	变更了输入变量“Continuous (Reserved)”的值		请修正程序，以不变更输入变量“Continuous (Reserved)”的值		请创建程序，以不变更输入变量“Continuous (Reserved)”的值	
附加信息	无					
注意事项/备注	无					

事件名称	超过加减法运算方法选择范围			事件代码	54340000Hex	
内容	运动控制指令的输入变量“CombineMode”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过开始同步条件指定范围			事件代码	54350000Hex	
内容	运动控制指令的输入变量“LinkOption”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	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”指定的轴相同					
发生源	运动控制功能模块		发生源详情	MC通用/轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		正在发生MC通用轻度故障	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入变量“Master”和“Slave”的参数相同		请修正参数,以避免相应指令的输入变量“Master”和“Slave”指定的轴相同		请为指令的输入变量“Master”和“Slave”指定不同的轴	
附加信息	无					
注意事项/备注	无					

事件名称	主轴辅轴相同			事件代码	54370000Hex	
内容	运动控制指令的输入变量“Master”和“Auxiliary”指定的轴相同					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入变量“Master”和“Auxiliary”的参数相同		请修正参数,以避免相应指令的输入变量“Master”和“Auxiliary”指定的轴相同		请为指令的输入变量“Master”和“Auxiliary”指定不同的轴	
附加信息	无					
注意事项/备注	无					

事件名称	主轴/从轴 轴号非升序			事件代码	54380000Hex	
内容	运动控制指令的输入变量“Master”和“Slave”指定的轴号非升序					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入变量“ReferenceType”指定“_mcLatestCommand”时,指令的输入变量“Master”和“Slave”的参数非升序		相应指令的输入变量“ReferenceType”指定“_mcLatestCommand”时,请修正参数,以使相应指令的输入变量“Master”和“Slave”指定的轴号为升序。或以主轴位置类型选择指定“_mcCommand”		变量“ReferenceType”指定“_mcLatestCommand”时,请将输入变量的主轴和从轴指定为升序	
附加信息	无					
注意事项/备注	无					

事件名称	凸轮表指定错误		事件代码	54390000Hex		
内容	运动控制指令的输入变量“CamTable”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	MC通用/轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“MC通用”时： 无影响 具体发生源为“轴”时： 相应从轴不可动作		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		正在发生MC通用轻度故障	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入变量“CamTable”指定了非凸轮数据变量		请修正参数，以使相应指令的输入变量“CamTable”指定的参数为凸轮数据变量		请为指令的输入变量“CamTable”指定凸轮数据变量	
附加信息	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，但并非可执行条件。					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	<ul style="list-style-type: none"> 执行了MC_CamOut(凸轮动作解除)指令，但MC_CamIn(凸轮动作开始)指令未在运行 执行了MC_GearOut(齿轮动作解除)指令，但MC_GearIn(齿轮动作开始)指令、MC_GearInPos(位置指定齿轮动作)指令未在运行 执行了MC_Phasing(主轴相对值相位补偿)指令，但MC_CamIn(凸轮动作开始)指令、MC_GearIn(齿轮动作开始)指令、MC_GearInPos(位置指定齿轮动作)指令、MC_MoveLink(梯形模式凸轮)指令未在运行 		请修正程序，以确保执行相应指令时可执行		请确保执行同步控制指令时可执行	
附加信息	无					
注意事项/备注	无					

事件名称	无法重启运动指令		事件代码	543B0000Hex		
内容	重启了无法重启的运动控制指令					
发生源	运动控制功能模块		发生源详情	MC通用/轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“MC通用”时： 无影响 具体发生源为“轴”时： 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时： 相应轴组无法动作。在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		正在发生MC通用轻度故障	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障		
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	重启了无法重启的运动控制指令		请修正程序，以避免相应指令的输出变量“Busy”变为FALSE前，输入变量“Execute”处于上升沿		使用无法重启的指令时，请确保输入变量“Execute”的启动条件中包含该指令的输出变量“Busy”为FALSE。或确保再次启动指令前暂时停止指令	
附加信息	无					
注意事项/备注	无					

事件名称	无法多重启动运动指令		事件代码	543C0000Hex		
内容	对相同对象(MC通用/轴/轴组)执行了多个无法同时执行的功能					
发生源	运动控制功能模块		发生源详情	MC通用/轴/轴组	检测时间	开始多重启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“MC通用”时： 无影响 具体发生源为“轴”时： 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时： 相应轴组无法动作。在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		正在发生MC通用轻度故障	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障		
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	对相同对象(MC通用/轴/轴组)执行了多个无法同时执行的功能		请确认相应指令的多重启动的规格，修正程序以确保不同时执行的指令		请确认待使用指令的多重启动的规格，确保不同时执行无法同时执行的指令	
附加信息	无					
注意事项/备注	无					

事件名称	不符合轴类型		事件代码	543D0000Hex		
内容	对编码器轴执行了动作指令					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	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		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	对编码器轴执行了动作指令		请为相应指令的轴类型设定指定伺服轴或虚拟伺服轴，或修正程序以确保不对编码器轴执行相应动作指令		请确保对伺服轴或虚拟伺服轴执行动作指令	
附加信息	无					
注意事项/备注	无					

事件名称	无法启动多轴协调动作中的指令		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 执行了启用轴组时无法使用的机器人指令 					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	开始多重启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	轴组减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	对多轴协调动作中的轴或轴组执行了动作指令		请修正程序，以确保在相应轴或相应轴组未进行多轴协调动作过程中执行相应指令		请向未进行多轴协调动作过程中的轴或轴组执行轴动作指令	
	对轴组有效状态下的轴组执行了MC_SetKinTransform(运动学转换设定)指令		请修正程序，以确保在禁用轴组时执行相应指令。		请确保禁用轴组时执行相应指令。	
附加信息	无					
注意事项/备注	无					

事件名称	启动轴组无效状态下的多轴协调指令			事件代码	543F0000Hex	
内容	对轴组无效状态下的轴组启动了多轴协调指令					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴组无法动作 在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	对轴组无效状态下的轴组启动了多轴协调指令		请修正程序，以将轴组设为有效状态后执行相应指令。将轴组设为有效状态时，请执行		请将轴组设为有效状态后执行多轴协调动作指令。将轴组设为有效状态时，请执行MC_GroupEnable(启用轴组)指令	
	对轴组无效状态下的轴组启动了以下指令		MC_GroupEnable(启用轴组)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(输送机同步动作开始)指令 • MC_SyncOut(同步动作解除)指令 • MC_RobotJog(轴组微动移动)指令 					
附加信息	无					
注意事项/备注	无					

事件名称	无法启用轴组			事件代码	54400000Hex	
内容	MC_GroupEnable(启用轴组)指令执行失败					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴组无法动作。继续执行构成轴的动作		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	执行MC_GroupEnable(启用轴组)指令时，构成轴中存在未处于停止状态的轴		请修正程序，以确保在构成轴均处于停止状态下执行MC_GroupEnable(启用轴组)指令。轴变量的“Status.Disabled”或“Status.Standstill”为TRUE时，该轴处于停止状态		请编写程序，以确保在构成轴均处于停止状态下执行MC_GroupEnable(启用轴组)指令。轴变量的“Status.Disabled”或“Status.Standstill”为TRUE时，该轴处于停止状态	
	执行MC_GroupEnable(启用轴组)指令时，构成轴中存在正在执行MC_TouchProbe(启用外部锁定)指令的轴		请修正程序，以确保在所有构成轴未执行MC_TouchProbe(启用外部锁定)指令时执行MC_GroupEnable(启用轴组)指令		请编写程序，以确保在所有构成轴未执行MC_TouchProbe(启用外部锁定)指令时执行MC_GroupEnable(启用轴组)指令	
附加信息	无					
注意事项/备注	无					

事件名称	无法运行(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					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	不启动动作指令		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
	_MC_GRP[*].MFAultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	对伺服OFF中的轴执行了动作指令		请修正程序, 以确保在伺服ON后执行相应指令		请确保在伺服ON后执行轴动作指令	
对未建立EtherCAT的过程数据通信的轴执行了通过MC_Home(原点复位)指令或MC_HomeWithParameter(参数指定原点复位)指令进行的原点预设。		EtherCAT主站的系统定义变量“_EC_PDslavTbl”(过程数据通信中从站表)变为FALSE时, 请排除其原因, “_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)指令执行原点预设		请编写程序, 以确保在控制器的电源ON或下载后, 抑或从站的通信异常解除、取消、再次添加、不启用、启用后, 通过MC_Home(原点复位)指令或MC_HomeWithParameter(参数指定原点复位)指令执行原点预设时, 确认EtherCAT主站的系统定义变量“_EC_PDslavTbl”(过程数据通信中从站表)变为TRUE后再执行		
附加信息	附属信息1: 因发生源详情而异 轴时: “0” 轴组时: 发生异常的逻辑轴号					
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	针对构成轴, 向已执行MC_Stop(强制停止)指令的轴组执行了动作指令					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴组无法动作		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFAultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	针对构成轴, 向已执行MC_Stop(强制停止)指令的轴组执行了动作指令		请将构成轴的强制停止指令的输入变量“Execute”设为FALSE并停止执行, 执行异常解除后赋予动作指令		请将构成轴强制停止指令的输入变量“Execute”设为FALSE并退出后, 执行轴组的动作指令	
附加信息	附属信息1: 强制停止的逻辑轴号					
注意事项/备注	无					

事件名称	多重启动运动指令数超限			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限					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	开始多重启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时： 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时： 相应轴组无法动作。在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	正在执行的轴指令和缓存中的轴指令的总数超过了“2”		请修正程序，以确保执行相应指令时缓存数不超过上限		请确保正在执行的轴组指令和缓存中的轴组指令的总数不超过“2”	
	正在执行的轴组指令和缓存中的轴组指令的总数超过了“8”				请确保正在执行的轴组指令和缓存中的轴组指令的总数不超过“8”	
附加信息	无					
注意事项/备注	无					

事件名称	移动量不足			事件代码	54440000Hex	
内容	多重启动/重启定位指令时，无法执行以指定减速度或加速度指定的动作					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时： 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时： 相应轴组无法动作。在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	如果已将“加减速超限”设定为“作为异常停止”，则无法在多重启动/重启定位指令时，在指定减速度/加速度的条件下在目标位置停止		请根据相应指令的动作规格修正程序，以确保定位动作不会在多重启动/重启时指定的减速度或加速度下超过目标位置。 或将“加减速超限”变更设定为非“作为异常停止”		请先确认待使用指令的动作规格，再编写程序以避免发生本异常。或将“加减速超限”设定为非“作为异常停止”	
附加信息	无					
注意事项/备注	无					

事件名称	用于达到混合中继速度的移动量不足			事件代码	54450000Hex	
内容	用于中继速度加减速的移动量不足					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	开始多重启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时： 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时： 相应轴组无法动作。在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	将“加减速超限”设定为“作为异常停止”时，用于将当前指令加减速至中继速度的移动量不足		请根据相应指令的动作规格修正程序，以避免移动量不足。 或将“加减速超限”变更设定为非“作为异常停止”		请先确认待使用指令的动作规格，再编写程序以避免发生本异常。 或将“加减速超限”设定为非“作为异常停止”	
附加信息	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	54460000Hex	
内容	主轴的等速移动量小于“0”					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	MC_MoveLink(梯形模式凸轮)指令下，主轴的等速移动量小于“0”		请修正程序，以满足(主轴移动距离 \geq 主轴加速移动距离+主轴减速移动距离)		请先确认待使用指令的动作规格，再编写程序以避免发生本异常	
附加信息	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	54470000Hex	
内容	MC_GearInPos(位置指定齿轮动作)指令下，从轴的“目标速度”较小，因此达不到所需速度					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	MC_GearInPos(位置指定齿轮动作)指令下，输入变量“Velocity(目标速度)”的值小于(启动指令时的主轴速度 \times 齿轮比)		请根据相应指令的动作规格设定输入变量“Velocity(目标速度)”的值，使其大于(指令启动时的主轴速度 \times 齿轮比)		请先确认待使用指令的动作规格，再编写程序以避免发生本异常	
附加信息	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)指令下,指定半径指定方式时起点和终点的位置相同。或指定通过点指定方式时起点、终点及通过点的位置相同					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴组无法动作 在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	MC_MoveCircular2D(2轴圆弧插补)指令下,指定了半径指定方式,起点和终点的位置相同		请修正程序,以避免在相应指令的起点和终点位置相同的状态下指定半径指定方式		执行半径指定方式的圆弧插补时,请确保起点和终点不同	
	MC_MoveCircular2D(2轴圆弧插补)指令下,指定了通过点指定方式,起点、终点及通过点的位置相同		请修正程序,以避免在相应指令的起点和终点与通过点位置相同的状态下指定通过点指定方式		执行通过点指定方式的圆弧插补时,请确保起点和终点与通过点不同	
附加信息	无					
注意事项/备注	无					

事件名称	超过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)指令下,指定中心点指定方式时,中心点的位置指定超过容许范围					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴组无法动作 在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	MC_MoveCircular2D(2轴圆弧插补)指令下,指定了中心点指定方式,起点与中心点的距离、终点与中心点的距离之差超过了轴组设定的“中心点补偿容许率”指定的容许范围		请进行修正以设定中心点,使其满足起点与中心点的距离、终点与中心点的距离之差小于轴组设定的“中心点补偿容许率”指定的容许范围		请满足起点与中心点的距离、终点与中心点的距离之差小于轴组设定的“中心点补偿容许率”指定的容许范围	
附加信息	无					
注意事项/备注	无					

事件名称	计数模式设定导致的指令启动异常		事件代码	544A0000Hex		
内容	对计数模式已设定为旋转模式的轴，执行了旋转模式下无法使用的指令					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴组无法动作 在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	以旋转模式下无法使用的指令使用了计数模式已设定为旋转模式的轴		请将相应指令已指定的相应轴的计数模式的设定变更为线性模式		请确认可执行待使用指令的计数模式，设定轴的计数模式。	
附加信息	无					
注意事项/备注	无					

事件名称	超过参数选择范围		事件代码	544C0000Hex		
内容	运动控制指令的输入变量“ParameterNumber”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	MC通用	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	不执行相应指令		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		正在发生MC通用轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过停止方法选择范围		事件代码	544D0000Hex		
内容	运动控制指令的输入变量“StopMode”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	MC通用	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	不执行相应指令		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		正在发生MC通用轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
	参数指定和设定值的数据类型不一致		请进行修正,以使参数指定和设定值的数据类型一致		请使参数指定和设定值的数据类型一致	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围				
发生源	运动控制功能模块		发生源详情	轴	检测时间 启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别 系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围
附加信息	无				
注意事项/备注	无				

事件名称	无法重启运动指令(轴指定)		事件代码	54530000Hex	
内容	变更了重启运动控制指令时无法变更的输入变量“Axis”的参数				
发生源	运动控制功能模块		发生源详情	轴	检测时间 重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别 系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	变更了重启时无法变更的输入变量的参数		请修正程序，以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更，并编写程序，以避免无法变更的输入变量在重启时输入参数发生变化。
附加信息	无				
注意事项/备注	无				

事件名称	无法重启运动指令(缓存模式选择)		事件代码	54540000Hex	
内容	变更了重启运动控制指令时无法变更的输入变量“BufferMode”的参数				
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间 重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别 系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时： 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时： 相应轴组无法动作。在相应轴组的动作中减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	变更了重启时无法变更的输入变量的参数		请修正程序，以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更，并编写程序，以避免无法变更的输入变量在重启时输入参数发生变化。
附加信息	无				
注意事项/备注	无				

事件名称	无法重启运动指令(方向选择)			事件代码	54550000Hex	
内容	变更了重启运动控制指令时无法变更的输入变量“Direction”的参数					
发生源	运动控制功能模块		发生源详情	轴	检测时间	重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	变更了重启时无法变更的输入变量的参数		请修正程序,以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更,并编写程序,以避免无法变更的输入变量在重启时输入参数发生变化。	
附加信息	无					
注意事项/备注	无					

事件名称	无法重启运动指令(重复模式)			事件代码	54560000Hex	
内容	变更了重启运动控制指令时无法变更的输入变量“Periodic”的参数					
发生源	运动控制功能模块		发生源详情	轴	检测时间	重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	变更了重启时无法变更的输入变量的参数		请修正程序,以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更,并编写程序,以避免无法变更的输入变量在重启时输入参数发生变化。	
附加信息	无					
注意事项/备注	无					

事件名称	无法重启运动指令(轴组指定)			事件代码	54570000Hex	
内容	变更了重启运动控制指令时无法变更的输入变量“AxesGroup”的参数					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴组无法动作 在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	变更了重启时无法变更的输入变量的参数		请修正程序,以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更,并编写程序,以避免无法变更的输入变量在重启时输入参数发生变化。	
附加信息	无					
注意事项/备注	无					

事件名称	无法重启运动指令(跳动设定)			事件代码	54580000Hex	
内容	变更了重启运动控制指令时无法变更的输入变量“Jerk”的参数					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时： 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时： 相应轴组无法动作。在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	变更了重启时无法变更的输入变量的参数		请修正程序，以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更，并编写程序，以避免无法变更的输入变量在重启时输入参数发生变化。	
附加信息	无					
注意事项/备注	无					

事件名称	无法重启运动指令(主轴)			事件代码	54590000Hex	
内容	变更了重启运动控制指令时无法变更的输入变量“Master”的参数					
发生源	运动控制功能模块		发生源详情	轴	检测时间	重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	变更了重启时无法变更的输入变量的参数		请修正程序，以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更，并编写程序，以避免无法变更的输入变量在重启时输入参数发生变化。	
附加信息	无					
注意事项/备注	无					

事件名称	无法重启运动指令(MasterOffset)			事件代码	545A0000Hex	
内容	变更了重启运动控制指令时无法变更的输入变量“MasterOffset”的参数					
发生源	运动控制功能模块		发生源详情	轴	检测时间	重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	变更了重启时无法变更的输入变量的参数		请修正程序，以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更，并编写程序，以避免无法变更的输入变量在重启时输入参数发生变化。	
附加信息	无					
注意事项/备注	无					

事件名称	无法重启运动指令(MasterScaling)			事件代码	545B0000Hex	
内容	变更了重启运动控制指令时无法变更的输入变量“MasterScaling”的参数					
发生源	运动控制功能模块		发生源详情	轴	检测时间	重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	变更了重启时无法变更的输入变量的参数		请修正程序，以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更，并编写程序，以避免无法变更的输入变量在重启时输入参数发生变化。	
附加信息	无					
注意事项/备注	无					

事件名称	无法重启运动指令(MasterStartDistance)			事件代码	545C0000Hex	
内容	变更了重启运动控制指令时无法变更的输入变量“MasterStartDistance”的参数					
发生源	运动控制功能模块		发生源详情	轴	检测时间	重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	变更了重启时无法变更的输入变量的参数		请修正程序，以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更，并编写程序，以避免无法变更的输入变量在重启时输入参数发生变化。	
附加信息	无					
注意事项/备注	无					

事件名称	无法重启运动指令(Continuous)		事件代码	545D0000Hex	
内容	变更了重启运动控制指令时无法变更的输入变量“Continuous”的参数				
发生源	运动控制功能模块		发生源详情	轴	检测时间 重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别 系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	变更了重启时无法变更的输入变量的参数		请修正程序，以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更，并编写程序，以避免无法变更的输入变量在重启时输入参数发生变化。
附加信息	无				
注意事项/备注	无				

事件名称	无法重启运动指令(MoveMode)		事件代码	545E0000Hex	
内容	变更了重启运动控制指令时无法变更的输入变量“MoveMode”的参数				
发生源	运动控制功能模块		发生源详情	轴	检测时间 重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别 系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	变更了重启时无法变更的输入变量的参数		请修正程序，以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更，并编写程序，以避免无法变更的输入变量在重启时输入参数发生变化。
附加信息	无				
注意事项/备注	无				

事件名称	辅轴指定错误		事件代码	545F0000Hex	
内容	运动控制指令的输入变量“Auxiliary”指定的轴不存在				
发生源	运动控制功能模块		发生源详情	轴	检测时间 启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别 系统
发生后的影响	用户程序	继续	动作	相应从轴无法动作。 相应从轴在动作中减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	指令的输入变量“Auxiliary”指定的轴不存在的变量		请将相应指令已指定的轴修正为存在的变量		为指令的输入参数指定变量时，请指定存在的变量
附加信息	无				
注意事项/备注	无				

事件名称	轴指定错误			事件代码	5460000Hex	
内容	运动控制指令的输入变量“Axis”指定的轴不存在					
发生源	运动控制功能模块		发生源详情	MC通用	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	不执行相应指令		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		正在发生MC通用轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入变量“Axis”指定的轴不存在的变量		请将相应指令已指定的轴修正为存在的变量		为指令的输入参数指定变量时, 请指定存在的变量	
附加信息	无					
注意事项/备注	无					

事件名称	轴组指定错误			事件代码	54610000Hex	
内容	运动控制指令的输入变量“AxesGroup”指定的轴组不存在或不是使用轴组					
发生源	运动控制功能模块		发生源详情	MC通用	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	不执行相应指令		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		正在发生MC通用轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入变量“AxesGroup”指定的轴组不存在的变量		请将相应指令已指定的轴组修正为存在的轴组变量		为指令的输入参数指定变量时, 请指定存在的变量	
	指令的输入变量“AxesGroup”指定的轴组未设定为使用轴组		请将相应指令已指定的轴组设定为使用轴组		请将指定为输入变量“AxesGroup”的轴组设定为使用轴组	
附加信息	无					
注意事项/备注	无					

事件名称	主轴指定错误		事件代码	54620000Hex		
内容	运动控制指令的输入变量“Master”指定的轴错误					
发生源	运动控制功能模块		发生源详情	MC通用/轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 相应从轴在轴动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		正在发生MC通用轻度故障	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入变量“Master”指定的轴不存在的变量		请将相应指令已指定的轴修正为存在的变量		为指令的输入参数指定变量时, 请指定存在的变量	
	MC_Phasing(主轴相对值相位补偿)指令时, 输入变量“Master”指定的轴不是同步主轴		请将MC_Phasing(主轴相对值相位补偿)指令的输入变量“Master”输入的变量修正为同步控制指令的主轴已指定的轴		请将MC_Phasing(主轴相对值相位补偿)指令的输入变量“Master”输入的变量设定为同步控制指令的主轴已指定的轴	
	分配主轴和从轴的任务不同		请将相应指令的输入变量“Master”和“Slave”中输入的轴, 分配至同一任务		主轴、从轴请指定为分配在同一任务中的轴	
附加信息	无					
注意事项/备注	无					

事件名称	无法重启运动指令(SlaveOffset)		事件代码	54630000Hex		
内容	变更了重启运动控制指令时无法变更的输入变量“SlaveOffset”					
发生源	运动控制功能模块		发生源详情	轴	检测时间	重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	变更了重启时无法变更的输入变量的参数		请修正程序, 以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更, 并编写程序, 以避免无法变更的输入变量在重启时输入参数发生变化。	
附加信息	无					
注意事项/备注	无					

事件名称	无法重启运动指令(SlaveScaling)			事件代码	54640000Hex	
内容	变更了重启运动控制指令时无法变更的输入变量 “SlaveScaling”					
发生源	运动控制功能模块		发生源详情	轴	检测时间	重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	变更了重启时无法变更的输入变量的参数		请修正程序，以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更，并编写程序，以避免无法变更的输入变量在重启时输入参数发生变化。	
附加信息	无					
注意事项/备注	无					

事件名称	无法重启运动指令(StartPosition)			事件代码	54650000Hex	
内容	变更了重启运动控制指令时无法变更的输入变量 “StartPosition”					
发生源	运动控制功能模块		发生源详情	轴	检测时间	重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	变更了重启时无法变更的输入变量的参数		请修正程序，以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更，并编写程序，以避免无法变更的输入变量在重启时输入参数发生变化。	
附加信息	无					
注意事项/备注	无					

事件名称	原点未确定状态下的指令启动异常		事件代码	54660000Hex		
内容	原点未确定状态下执行了高速原点复位或插补指令					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时： 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时： 相应轴组无法动作。在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	原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后再执行相应的高速原点复位		请在通过原点复位确定原点的状态下执行高速原点复位指令	
	向包含原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后执行相应的插补指令		请通过原点复位确定所有构成轴的原点后执行插补指令	
	向包含原点未确定状态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(输送机同步动作开始)指令 • MC_SyncOut(同步动作解除)指令 • MC_GroupMon(轴组监控)指令 • MC_RobotJog(轴组微动移动)指令					
附加信息	附属信息1: 因发生源详情而异 轴时: “0” 轴组时: 逻辑轴号					
注意事项/备注	重新通过执行原点复位设为原点确定状态, 以实现执行原点复位, 执行“当前位置变更”指令后进入原点未确定状态。					

事件名称	无法重启运动指令(位置类型)		事件代码	54670000Hex		
内容	变更了重启运动控制指令时无法变更的输入变量“ReferenceType”					
发生源	运动控制功能模块		发生源详情	轴	检测时间	重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	变更了重启时无法变更的输入变量的参数		请修正程序, 以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更, 并编写程序, 以避免无法变更的输入变量在重启时输入参数发生变化。	
附加信息	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	运动控制指令指定的主轴为未使用轴					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 相应从轴在动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	运动控制指令指定的主轴为未使用轴		请将相应指令已指定的主轴修正为使用轴		请为运动控制指令指定的主轴指定使用轴	
附加信息	无					
注意事项/备注	无					

事件名称	超过起始位置设定范围			事件代码	54690000Hex	
内容	运动控制指令的输入变量“FirstPosition”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过终止位置设定范围			事件代码	546A0000Hex	
内容	运动控制指令的输入变量“LastPosition”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	起始位置/终止位置、大小关系错误(线性模式)		事件代码	546B0000Hex	
内容	运动控制指令的输入变量“LastPosition”指定的参数值小于输入变量“FirstPosition”指定的参数值				
发生源	运动控制功能模块		发生源详情	轴	检测时间
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	计数模式为线性模式时,指令的输入参数“LastPosition”的值小于“FirstPosition”的值		请进行修正,以满足相应指令指定的“LastPosition”的值大于“FirstPosition”的值。或将计数模式变更为旋转模式		请编写程序,以满足相应指令指定的“LastPosition”的值大于“FirstPosition”的值。或确认相应轴的计数模式已变为旋转模式
附加信息	无				
注意事项/备注	无				

事件名称	超过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”指定的参数超过范围				
发生源	运动控制功能模块		发生源详情	轴	检测时间
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别
发生后的影响	用户程序	继续	动作	相应从轴不可动作 相应从轴在动作中减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围
附加信息	无				
注意事项/备注	无				

事件名称	超过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”指定的参数超过范围				
发生源	运动控制功能模块		发生源详情	轴	检测时间
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围
附加信息	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	多个指令已重复运动控制指令指定的锁定ID					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	MC_TouchProbe(启用外部锁定)指令、MC_MoveLink(梯形模式凸轮)指令、MC_MoveFeed(中断标准定位)指令已同时使用相同的锁定ID		请修正程序，以避免相应指令和其他指令同时使用相同的锁定ID。请确保使用不同的锁定ID或避免同时执行使用相同锁定ID的指令。 正在执行MC_Home(原点复位)指令或MC_HomeWithParameter(参数指定原点复位)指令即正在使用锁定功能1/2		请避免MC_TouchProbe(启用外部锁定)指令、MC_MoveLink(梯形模式凸轮)指令、MC_MoveFeed(中断标准定位)指令同时使用相同的锁定ID	
	试图通过MC_AbortTrigger(不启用外部锁定)指令中止正在以非MC_TouchProbe(启用外部锁定)指令使用的锁定		请避免通过外部锁定无效指令中止非外部锁定有效指令使用过程中的锁定		请避免非外部锁定有效指令使用过程中的锁定执行外部锁定无效指令	
附加信息	无					
注意事项/备注	变更锁定ID时，请确认其他指令未同时使用该锁定ID					

事件名称	超过跃度超调值范围			事件代码	546F0000Hex	
内容	运动控制指令的输入变量“JerkFactor”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过加减速度超调值范围			事件代码	5470000Hex	
内容	运动控制指令的输入变量“AccFactor”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过起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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	无法重启运动指令(起始位置方式)			事件代码	54720000Hex	
内容	变更了重启运动控制指令时无法变更的输入变量“StartMode”的参数					
发生源	运动控制功能模块		发生源详情	轴	检测时间	重启指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	变更了重启时无法变更的输入变量的参数		请修正程序,以避免重启相应指令时相应输入变量的参数发生变化		请手动确认待使用的运动控制指令的各输入变量在重启时可否变更,并编写程序,以避免无法变更的输入变量在重启时输入参数发生变化。	
附加信息	无					
注意事项/备注	无					

事件名称	未使用轴指定(辅轴)			事件代码	54740000Hex	
内容	运动控制指令的输入变量“Auxiliary”指定的轴为未使用轴					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 相应从轴在动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入变量“Auxiliary”指定的轴为未使用轴		请将相应指令已指定的轴设定为使用轴。或修正参数,以指定使用轴		请将指令指定的轴设为使用轴	
附加信息	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	54750000Hex	
内容	无法以运动控制指令输入的速度/加速度/减速度进行同步动作					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	无法以指令输入的速度/加速度/减速度进行指定的同步动作		请根据MC_GearInPos(位置指定齿轮动作)指令的动作规格修正程序,以设定为可同步动作		请确认待使用指令的处理,以设定为可同步动作	
附加信息	无					
注意事项/备注	无					

事件名称	位置指定齿轮主轴零速			事件代码	54760000Hex	
内容	启动运动控制指令时主轴的速度为“0”					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	启动指令时主轴的速度为“0”		请修正程序,以避免启动相应指令时主轴的速度变为“0”		请编写程序,以避免执行指令时主轴的速度变为“0”	
附加信息	无					
注意事项/备注	无					

事件名称	超过目标位置设定范围		事件代码	54780000Hex		
内容	运动控制指令的输入变量“Position”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时： 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时： 相应轴组无法动作。在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
	对于旋转模式的轴，目标位置超过环设定的范围		对于旋转模式的轴，请修正目标位置，以将其控制在环设定的范围内		对于旋转模式的轴，请将目标位置控制在环设定的范围内	
附加信息	附属信息1：因发生源详情而异 轴时：“0” 轴组时：相应指令的输入变量“Position”中超出范围的元素编号					
注意事项/备注	无					

事件名称	超过移动距离范围		事件代码	54790000Hex		
内容	运动控制指令的输入变量“Distance”指定的参数超过范围或加上“Distance”值后的目标位置超过范围					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时： 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时： 相应轴组无法动作。在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	将指令输入参数的绝对值转换为脉冲单位时超过40位的范围		请修正相应指令的输入变量“Distance”指定的输入参数，以将移动距离和目标位置控制在范围内		请编写程序，已将指令执行的移动距离和目标位置控制在范围内	
	对于线性模式的轴，将加上移动距离后的目标位置转换为脉冲单位时超过带符号40位的范围					
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过凸轮动作(主轴跟踪)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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	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)指令下，指定半径指定方式时无法以指定的半径创建圆弧的轨迹					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴组无法动作 在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	MC_MoveCircular2D(2轴圆弧插补)指令下，指定了半径指定方式，无法以指定的半径创建圆弧的轨迹		请修正半径，以确保可生成圆弧的轨迹		请确认待使用指令的处理，并设定半径，以确保可生成圆弧的轨迹	
附加信息	无					
注意事项/备注	无					

事件名称	辅轴/从轴、轴号非升序			事件代码	547F0000Hex	
内容	运动控制指令指定的输入变量“Auxiliary”和“Slave”的参数值未以升序排列					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入变量“Auxiliary”和“Slave”的参数非升序		请进行修正,以使相应指令的“Auxiliary”和“Slave”的输入参数指定的轴号为升序		请编写程序,以使“Auxiliary”和“Slave”指定的轴号为升序	
附加信息	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	判定有效数据数计算中相位非升序。或判定计算后有效数据数为“0”					
发生源	运动控制功能模块		发生源详情	MC通用	检测时间	执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	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		正在发生MC通用轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	判定有效数据数计算中相位非升序		请修正凸轮表,以使相位为升序		请确保凸轮表数据的相位为升序	
	判定计算后有效数据数为“0”		请修正凸轮表的数据,以包含非“0”的相位		请设定凸轮表的数据,以包含非“0”的相位	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	MC通用	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	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		正在发生MC通用轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过执行模式选择范围			事件代码	54870000Hex	
内容	运动控制指令的输入变量“ExecutionMode”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	MC通用	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	不执行相应指令		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		正在发生MC通用轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	通过点位置/中心位置/超过半径指定范围			事件代码	54890000Hex	
内容	运动控制指令的输入变量“AuxPoint”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴组无法动作 在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	通过点指定或中心指定时，将“AuxPoint”的值转换为脉冲单位时超过带符号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过40位的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过终点指定范围			事件代码	548A0000Hex	
内容	运动控制指令的输入变量“EndPoint”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴组无法动作 在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFAultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	将指令的输入参数转换为脉冲单位时,已超过带符号40位的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	将指令的输入参数转换为脉冲单位时,已超过40位的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过相位补偿量范围			事件代码	548C0000Hex	
内容	运动控制指令的输入变量“PhaseShift”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	将指令输入参数的绝对值转换为脉冲单位时超过40位的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过标准距离范围			事件代码	548D0000Hex	
内容	运动控制指令的输入变量“FeedDistance”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	将指令输入参数的绝对值转换为脉冲单位时超过40位的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	辅轴/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”指定的轴相同					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入变量“Auxiliary”和“Slave”的参数相同		请修正参数，以避免相应指令的输入变量“Auxiliary”和“Slave”指定的轴相同		请为指令输入变量的辅轴和从轴指定不同的轴	
附加信息	无					
注意事项/备注	无					

事件名称	超过相对位置选择范围			事件代码	548F0000Hex	
内容	运动控制指令的输入变量“Relative”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过凸轮过渡指定选择范围			事件代码	5490000Hex	
内容	运动控制指令的输入变量“CamTransition”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	无法执行启用外部锁定指令			事件代码	5492000Hex	
内容	对于编码器轴,输入变量“StopMode”指定为“_mcImmediateStop(立即停止)”,执行了驱动模式的MC_TouchProbe(启用外部锁定)指令					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	对于编码器轴,输入变量“StopMode”指定为“_mcImmediateStop(立即停止)”,执行了驱动模式的MC_TouchProbe(启用外部锁定)指令		请对编码器轴修正程序,以免在“StopMode”中指定“_mcImmediateStop(立即停止)”		指定“_mcImmediateStop(立即停止)”,执行驱动模式的启用外部锁定指令时,请确保对伺服轴执行	
附加信息	无					
注意事项/备注	无					

事件名称	超过主轴偏移设定范围			事件代码	54930000Hex	
内容	运动控制指令的输入变量“MasterOffset”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	将指令的输入参数转换为脉冲单位时, 已超过带符号40位的范围		请修正参数, 以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过从轴偏移设定范围			事件代码	54940000Hex	
内容	运动控制指令的输入变量“SlaveOffset”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴不可动作 在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	将指令的输入参数转换为脉冲单位时, 已超过带符号40位的范围		请修正参数, 以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数, 以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴无法动作。在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴无法动作。在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过辅轴齿轮比分子范围			事件代码	54980000Hex	
内容	运动控制指令的输入变量“RatioNumeratorAuxiliary”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴无法动作。在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFAultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数,以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过辅轴齿轮比分母范围			事件代码	54990000Hex	
内容	运动控制指令的输入变量“RatioDenominatorAuxiliary”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴无法动作。在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴无法动作。在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过辅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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应从轴无法动作。在相应从轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无					
注意事项/备注	无					

事件名称	超过目标位置环计数器范围		事件代码	549C0000Hex	
内容	已执行指令的目标位置超过环计数器的范围，因此无法动作				
发生源	运动控制功能模块		发生源详情	轴	检测时间 启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别 系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	设定为环计数器范围内不含0时，执行了高速原点复位		设定为环计数器范围内不含0时，无法执行高速原点复位。请修正程序，以避免执行高速原点复位。或进行变更，以设定为相应轴的环计数器范围内不含0		设定为环计数器范围内不含0时，无法执行高速原点复位。请编写程序，以避免执行高速原点复位。或设定为相应轴的环计数器范围内不含0
附加信息	无				
注意事项/备注	无				

事件名称	超过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”指定的参数超过范围				
发生源	运动控制功能模块		发生源详情	轴组	检测时间 启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别 系统
发生后的影响	用户程序	继续	动作	相应轴组无法动作。在相应轴组的轴动作中减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	指令的输入参数超过输入变量的范围		请修正参数，以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围
	分配轴组构成轴的任务不同		请将相应指令的输入变量“Axes”指定的轴全部分配至同一任务		轴组构成轴请指定为分配在同一任务中的轴
附加信息	附属信息1：异常详情 01Hex：类型指定异常 02Hex：数组的元素数小于构成轴数 03Hex：指定的轴号重复、指定轴号的轴种类在对象范围外、指定的轴号在范围之外 04Hex：指定轴号的轴为单轴位置控制轴，无法设定为构成轴				
注意事项/备注	无				

*1. 单元版本Ver.1.01以上的CPU单元发生的事件。

事件名称	超过轴使用设定范围		事件代码	549E0000Hex *1		
内容	运动控制指令的输入变量“AxisUse”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	MC通用/轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	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		正在发生MC通用轻度故障	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数, 以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	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通用异常。					

*1. 单元版本Ver.1.04以上的CPU单元发生的事件。

事件名称	超过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”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	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		正在发生MC通用轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围		请修正参数, 以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	附属信息1: 异常详情 1: 原点复位动作范围外、2: 原点输入信号范围外、3: 原点复位开始方向范围外、4: 原点检测方向范围外、5: 正方向极限输入时动作范围外、6: 负方向极限输入时动作范围外、7: 原点复位速度范围外、8: 原点复位接近速度范围外、9: 原点复位加速度范围外、10: 原点复位减速度范围外、11: 原点复位跃度范围外、12: 原点输入屏蔽距离范围外、13: 原点位置偏置范围外、14: 原点复位接触时间范围外、15: 原点复位补偿值范围外、16: 原点补偿速度范围外、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: 未满足原点复位补偿速度≤最高速度、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					
注意事项/备注	无					

*1. 单元版本Ver.1.03以上的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	向非处于停止状态或处于指令速度饱和状态的轴执行了MC_ChangeAxisUse(轴使用变更)指令				
发生源	运动控制功能模块	发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别
发生后的影响	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		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)	处理措施		防止再次发生	
	向非处于停止状态或处于指令速度饱和状态的轴执行了MC_ChangeAxisUse(轴使用变更)指令	执行异常解除, 轴已停止时, 请在指令速度处于非饱和的状态下执行MC_ChangeAxisUse(轴使用变更)指令。 轴变量的“Status.Disabled”或“Status.Standstill”为TRUE时, 该轴处于停止状态。轴变量的“Details.VelLimit”为TRUE时, 该轴处于指令速度饱和状态		轴已停止时, 请在指令速度处于非饱和的状态下执行MC_ChangeAxisUse(轴使用变更)指令	
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.04以上的CPU单元发生的事件。

事件名称	无法进行轴使用切换		事件代码	57030000Hex ^{*1}	
内容	试图超出使用实轴最大数量或使用运动控制伺服轴最大数量执行MC_ChangeAxisUse(轴使用变更)指令				
发生源	运动控制功能模块	发生源详情	MC通用	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别
发生后的影响	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		正在发生MC通用轻度故障	
发生原因及其处理	发生原因(推测原因)	处理措施		防止再次发生	
	试图超出使用实轴最大数量执行MC_ChangeAxisUse(轴使用变更)指令	请修正程序, 以免使用的CPU单元的使用实轴超过最大数量		请编写程序, 以免使用的CPU单元的使用实轴超过最大数量	
	试图超出使用运动控制伺服轴最大数量执行MC_ChangeAxisUse(轴使用变更)指令	请修正程序, 以免使用的CPU单元的使用运动控制伺服轴超过最大数量		请编写程序, 以免使用的CPU单元的使用运动控制伺服轴超过最大数量	
附加信息	附属信息1: 因发生源详情而异 1: 超出使用实轴最大数量 2: 超出使用运动控制伺服轴最大数量				
注意事项/备注	无				

*1. 单元版本Ver.1.06以上的CPU单元发生的事件。

事件名称	轴使用切换时运动控制参数设定异常		事件代码	5720000Hex ^{*1}	
内容	切换为使用轴的轴的运动控制参数设定错误				
发生源	运动控制功能模块	发生源详情	MC通用	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别
发生后的影响	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		正在发生MC通用轻度故障	
发生原因及其处理	发生原因(推测原因)	处理措施		防止再次发生	
	MC_ChangeAxisUse(轴使用变更)指令下,从未使用轴切换到使用轴的轴的运动控制参数设定错误	请使用Sysmac Studio将发生异常的轴的[轴使用]变更为[使用轴],确认异常部位并修正。如果未发生异常,请再次变更为[未使用轴]并下载		请确认“使用轴”的设定正常动作后作为[未使用轴]下载	
	下载运动控制参数设定过程中电源断开	请从Sysmac Studio下载运动控制参数设定		保存参数设定过程中请勿使电源断开	
	非易失性存储器的故障或非易失性存储器的寿命	进行上述处理后仍未解除本异常时,请更换CPU单元		无	
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.04以上的CPU单元发生的事件。

事件名称	未设定轴使用切换时所需的过程数据对象		事件代码	57210000Hex *1	
内容	未设定切换为使用轴的轴类型的所需对象				
发生源	运动控制功能模块		发生源详情	MC通用	检测时间 启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别 系统
发生后的影响	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		正在发生MC通用轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	切换为使用轴的轴类型的所需对象未设定PDO映射		请在Sysmac Studio的[编辑PDO映射设定]中设定发生异常的轴的所需对象。 关于必需对象,请参阅“2-3 PDO映射(P.2-30)”		请确认“使用轴”的设定正常动作后作为[未使用轴]下载
	下载运动控制参数设定过程中电源断开		请从Sysmac Studio下载运动控制参数设定		保存参数设定过程中请勿使电源断开
	非易失性存储器的故障或非易失性存储器的寿命		进行上述处理后仍未解除本异常时,请更换CPU单元		无
	对[轴使用]设定为[未使用轴(无法切换为使用轴)]的轴执行了MC_ChangeAxisUse(轴使用变更)指令		请修正程序,以免对[轴使用]设定为[未使用轴(无法切换为使用轴)]的轴执行MC_ChangeAxisUse(轴使用变更)指令		请编写程序,以免对[轴使用]设定为[未使用轴(无法切换为使用轴)]的轴执行MC_ChangeAxisUse(轴使用变更)指令
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.04以上的CPU单元发生的事件。

事件名称	无法多重启动运动指令(主轴)		事件代码	572F0000Hex *1	
内容	变更了多重启动指令时无法变更的输入输出变量“Master”				
发生源	运动控制功能模块		发生源详情	轴	检测时间 多重启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别 系统
发生后的影响	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		正在发生轴轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	变更了多重启动指令时无法变更的输入输出变量“Master”		请修正程序,以避免在相应指令多重启动时变更输入输出变量“Master”的值		请编写程序,以避免在相应指令多重启动时变更输入输出变量“Master”的值
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.06以上的CPU单元发生的事件。

事件名称	无法多重启动运动指令(位置类型选择)			事件代码	5730000Hex *1	
内容	变更了多重启动指令时无法变更的输入输出变量 “ReferenceType”					
发生源	运动控制功能模块		发生源详情	轴	检测时间	多重启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	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		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	变更了多重启动指令时无法变更的输入输出变量 “ReferenceType”		请修正程序, 以避免在相应指令多重启动时变更输入输出变量 “ReferenceType” 的值		请编写程序, 以避免在相应指令多重启动时变更输入输出变量 “ReferenceType” 的值	
附加信息	无					
注意事项/备注	无					

*1. 单元版本Ver.1.06以上的CPU单元发生的事件。

事件名称	轴参数无法写入			事件代码	573A0000Hex *1	
内容	对未使用轴以外的轴启动了相应指令					
发生源	运动控制功能模块		发生源详情	MC通用	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	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		正在发生MC通用轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	对使用轴或未创建轴启动了相应指令		请修正程序, 以确保在使用 MC_ChangeAxisUse(轴使用变更)指令将对象轴变更成未使用轴后再启动相应指令		请编写程序, 以确保相应指令启动时对象轴为未使用轴	
附加信息	无					
注意事项/备注	无					

*1. 单元版本Ver.1.08以上的CPU单元发生的事件。

事件名称	轴参数超过设定范围		事件代码	573B0000Hex *1	
内容	运动控制指令的输入变量“AxisParameter”指定的参数超过有效范围				
发生源	运动控制功能模块		发生源详情	MC通用	检测时间
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别
发生后的影响	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		正在发生MC通用轻度故障	
发生原因及其处理	发生原因(推测原因)	处理措施		防止再次发生	
	指令的输入变量“AxisParameter”指定的参数超过输入变量的范围	请修正参数，以避免超过相应指令输入变量的范围。 超过范围的参数或参数间的不匹配事项可通过附属信息进行确认		请避免指令的输入参数超过输入变量的范围。 输入变量的有效范围请参阅MC_WriteAxisParameter(轴参数写入)指令	
附加信息	<p>附属信息1: 异常详情</p> <ul style="list-style-type: none"> 范围检查的详细代码 <p>0000Hex: 表示单位范围外、0001Hex: 电机转1圈的脉冲数范围外、0002Hex: 电机转1圈的移动量范围外、0003Hex: 工件侧转1圈的移动量范围外、0004Hex: 工件侧齿轮比范围外、0005Hex: 电机侧齿轮比范围外、0100Hex: 最高速度范围外、0101Hex: 启动速度范围外、0102Hex: 微动最高速度范围外、0103Hex: 最大加速度范围外、0104Hex: 最大减速度范围外、0105Hex: 加减速超限范围外、0106Hex: 取反时动作范围外、0107Hex: 速度警告值范围外、0108Hex: 加速度警告值范围外、0109Hex: 减速度警告值范围外、010AHex: 正方向转矩警告值范围外、010BHex: 负方向转矩警告值范围外、010CHex: 到位宽度范围外、010DHex: 到位检查时间范围外、010EHex: 反馈速度滤波时间常数范围外、010FHex: 原点位置范围外、0200Hex: 立即停止输入停止方法范围外、0201Hex: 极限输入停止方法范围外、0202Hex: 驱动器错误复位监视时间范围外、0203Hex: 正方向转矩限制上限值范围外、0204Hex: 负方向转矩限制上限值范围外、0300Hex: 软件限制功能范围外、0301Hex: 正方向软件限制范围外、0302Hex: 负方向软件限制范围外、0303Hex: 位置偏差超过值范围外、0304Hex: 位置偏差警告值范围外、0400Hex: 计数模式范围外、0401Hex: 环计数器上限设定值范围外、0402Hex: 环计数器下限设定值范围外、0500Hex: 原点复位动作范围外、0501Hex: 原点输入信号范围外、0502Hex: 原点复位开始方向范围外、0503Hex: 原点检测方向范围外、0504Hex: 正方向极限输入时动作范围外、0505Hex: 负方向极限输入时动作范围外、0506Hex: 原点复位速度范围外、0507Hex: 原点复位接近速度范围外、0508Hex: 原点复位加速度范围外、0509Hex: 原点复位减速度范围外、050AHex: 原点复位跃度范围外、050BHex: 原点输入屏蔽距离范围外、050CHex: 原点位置偏置范围外、050DHex: 原点复位接触时间范围外、050EHex: 原点复位补偿值范围外、050FHex: 原点复位补偿速度范围外</p> <ul style="list-style-type: none"> 匹配检查的详细代码 <p>1000Hex: 工件侧转1圈的移动量×工件侧齿轮比/电机侧齿轮比的值为0.000000001~231的范围外、1001Hex: 电机转1圈的脉冲数×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: 脉冲单位转换后的最高速度超过500MHz、1101Hex: 启动速度超过最高速度、1102Hex: 微动最高速度超过最高速度、1103Hex: 到位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: 正方向软件限制在脉冲单位转换后超过40位、1301Hex: 负方向软件限制在脉冲单位转换后超过40位、1302Hex: 未满足正方向软件限制>负方向软件限制、1303Hex: 位置偏差超过值在脉冲单位转换后超过40位、1304Hex: 未满足位置偏差超过值≥位置偏差警告值、1400Hex: 环计数器上限设定值在脉冲单位转换后超过40位、1401Hex: 环计数器下限设定值在脉冲单位转换后超过40位、1402Hex: 未满足环计数器上限设定值>环计数器下限设定值、1403Hex: 未满足 环计数器上限设定值-环计数器下限设定值 在脉冲单位转换后为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: 原点位置偏置环在范围外、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) 即使发生了多个异常, 显示的代码也仅有一个。显示的优先顺序为“范围检查的详细代码”→“匹配检查的详细代码”。</p>				
注意事项/备注	无				

*1. 单元版本Ver.1.08以上的CPU单元发生的事件。

*2. 单元版本在Ver.1.10以下时最高速度的上限值为500MHz, Ver.1.11以上时则为2,147,483,647Hz。

事件名称	凸轮属性超过设定范围			事件代码	573C0000Hex ^{*1}
内容	运动控制指令的输入变量“CamProperty”指定的参数超过有效范围				
发生源	运动控制功能模块	发生源详情	MC通用	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别
发生后的影响	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		正在发生MC通用轻度故障	
发生原因及其处理	发生原因(推测原因)	处理措施		防止再次发生	
	指令的输入变量“CamProperty”指定的参数超过输入变量的范围	请修正参数，以避免超过相应指令输入变量的范围。 超过范围的参数可通过附属信息进行确认		请避免指令的输入参数超过输入变量的范围。	
附加信息	附属信息1：异常详情 0000Hex：初始速度范围外 0001Hex：初始加速度范围外 0002Hex：周期时间范围外				
注意事项/备注	无				

*1. 单元版本Ver.1.08以上的CPU单元发生的事件。

事件名称	凸轮节点超过设定范围			事件代码	573D0000Hex ^{*1}
内容	运动控制指令的输入变量“CamNodes”指定的参数超过有效范围				
发生源	运动控制功能模块	发生源详情	MC通用	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别
发生后的影响	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		正在发生MC通用轻度故障	
发生原因及其处理	发生原因(推测原因)	处理措施		防止再次发生	
	指令的输入变量“CamNodes”指定的参数超过输入变量的范围	请修正参数，以避免超过相应指令输入变量的范围。 超过范围的参数可通过附属信息进行确认		请避免指令的输入参数超过输入变量的范围。	
附加信息	附属信息1：异常详情 0000Hex：主轴相位范围外 0001Hex：从轴位移范围外 0002Hex：曲线形状范围外 0003Hex：连接速度范围外 0004Hex：连接加速度范围外 0005Hex：相位间隔宽度范围外 附属信息2：异常的节点元素编号				
注意事项/备注	无				

*1. 单元版本Ver.1.08以上的CPU单元发生的事件。

事件名称	凸轮节点类型指定错误		事件代码	573E0000Hex *1	
内容	运动控制指令的输入变量“CamNodes”指定的参数非_sMC_CAM_NODE型数组变量				
发生源	运动控制功能模块		发生源详情	MC通用	检测时间 启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别 系统
发生后的影响	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		正在发生MC通用轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	指令的输入变量“CamNodes”指定的参数非_sMC_CAM_NODE型数组变量		请修正程序,以确保在相应指令的输入变量中指定sMC_CAM_NODE型数组变量		请编写程序,以确保在相应指令的输入变量中指定sMC_CAM_NODE型数组变量
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.08以上的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”指定的参数数组变量中元素编号0的Phase值为“0”				
发生源	运动控制功能模块		发生源详情	MC通用	检测时间 启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别 系统
发生后的影响	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		正在发生MC通用轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	指令的输入变量“CamNodes”指定的参数数组变量中元素编号0的Phase(主轴相位)值为“0”		请修正程序,以确保输入变量“CamNodes”指定的参数数组变量中元素编号0的Phase(主轴相位)值不为“0”		请编写程序,以确保输入变量“CamNodes”指定的参数数组变量中元素编号0的Phase(主轴相位)值不为“0”
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.08以上的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”指定的参数数组变量中Phase的值未按元素编号顺序升序排列				
发生源	运动控制功能模块		发生源详情	MC通用	检测时间 启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别 系统
发生后的影响	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		正在发生MC通用轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	指令的输入变量“CamNodes”指定的参数数组变量中Phase(主轴相位)的值未按元素编号顺序升序排列。或者,舍去有效位第8位后的结果为非升序状态		请修正程序,以确保输入变量“CamNodes”指定的参数数组变量中Phase(主轴相位)的值按元素编号顺序升序排列		请编写程序,以确保输入变量“CamNodes”指定的参数数组变量中Phase(主轴相位)的值按元素编号顺序升序排列
附加信息	附属信息1: 异常的节点元素编号				
注意事项/备注	无				

*1. 单元版本Ver.1.08以上的CPU单元发生的事件。

事件名称	凸轮表生成数据点数过多			事件代码	57410000Hex *1	
内容	生成的凸轮数据数超过了运动控制指令的输入变量“CamTable”指定的凸轮数据变量的数组元素数					
发生源	运动控制功能模块		发生源详情	MC通用	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	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		正在发生MC通用轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	生成的凸轮表的凸轮数据数超过了指令的输入变量“CamTable”指定的凸轮数据变量的数组元素数		请修正程序，以避免生成的凸轮表的凸轮数据数超过指令的输入变量“CamTable”指定的凸轮数据变量的数组元素数。 生成的凸轮表的凸轮数据数请参阅MC_GenerateCamTable(凸轮表生成)指令		请编写程序，以避免生成的凸轮表的凸轮数据数超过指令的输入变量“CamTable”指定的凸轮数据变量的数组元素数 生成的凸轮表的凸轮数据数请参阅MC_GenerateCamTable(凸轮表生成)指令	
附加信息	附属信息1：异常的节点元素编号					
注意事项/备注	无					

*1. 单元版本Ver.1.08以上的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	生成的凸轮表的Distance超过了REAL型可表示的范围					
发生源	运动控制功能模块		发生源详情	MC通用	检测时间	启动指令时/ 执行指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	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		正在发生MC通用轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	生成的凸轮表的Distance超过了REAL型可表示的范围		输入变量“CamNodes”的Curve(曲线形状)指定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，以避免Distance溢出。 Distance的计算方法请参阅MC_GenerateCamTable(凸轮表生成)指令		输入变量“CamNodes”的Curve(曲线形状)指定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，以避免Distance溢出。 Distance的计算方法请参阅MC_GenerateCamTable(凸轮表生成)指令	
附加信息	附属信息1：异常的节点元素编号					
注意事项/备注	无					

*1. 单元版本Ver.1.08以上的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	相应指令的输入变量“CamTable”指定了被中止生成的凸轮数据变量					
发生源	运动控制功能模块		发生源详情	MC通用/轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	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		正在发生MC通用轻度故障	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	相应指令的输入变量“CamTable”指定了因MC_GenerateCamTable(凸轮表生成)指令异常而被中止生成的凸轮数据变量		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(节点元素编号)后修正程序,以生成正常的凸轮数据变量		请编写程序,以确保MC_GenerateCamTable(凸轮表生成)指令生成正常的凸轮数据变量。或编写程序,仅在MC_GenerateCamTable(凸轮表生成)指令正常完成时启动相应指令	
附加信息	无					
注意事项/备注	无					

*1. 单元版本Ver.1.08以上的CPU单元发生的事件。

事件名称	执行ID超过设定范围			事件代码	57490000Hex *1	
内容	运动控制指令的输入变量“ExecID”指定的参数超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入变量“ExecID”指定的参数超过输入变量的范围		请修正程序,以确保指令的输入变量“ExecID”指定的输入参数在范围内		请编写程序,以确保指令的输入变量“ExecID”指定的输入参数在范围内	
附加信息	无					
注意事项/备注	无					

*1. 单元版本Ver.1.10以上的CPU单元发生的事件。

事件名称	位置偏置超过范围		事件代码	574A0000Hex *1	
内容	运动控制指令的输入变量“OffsetPosition”指定的参数超过范围				
发生源	运动控制功能模块	发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)	处理措施		防止再次发生	
	将指令的输入参数转换为脉冲单位时, 已超过带符号40位的范围	请修正参数, 以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.10以上的CPU单元发生的事件。

事件名称	PDS状态变化指令选择超过范围		事件代码	574B0000Hex *1	
内容	运动控制指令的输入变量“TransitionCmd”指定的参数超过范围				
发生源	运动控制功能模块	发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)	处理措施		防止再次发生	
	指令的输入参数超过输入变量的范围	请修正参数, 以避免超过相应指令输入变量的范围		请避免指令的输入参数超过输入变量的范围	
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.10以上的CPU单元发生的事件。

事件名称	无法启动单轴位置控制轴运动指令		事件代码	574C0000Hex *1	
内容	执行了无法对单轴位置控制轴执行的运动指令				
发生源	运动控制功能模块	发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止	
系统定义变量	变量名称	数据类型		名称	
	_MC_AX[*].MFaultLvl.Active	BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)	处理措施		防止再次发生	
	执行了无法对单轴位置控制轴执行的运动指令	请在指令指定轴的轴基本设定的控制功能中设定“0:全部”。或者, 请确保指定轴基本设定的控制功能设定成“0:全部”的轴		同左	
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.13以上的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	指定的位置超过正方向软件限制的范围				
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时： 相应轴不可动作。在相应轴的动作中减速停止 具体发生源为“轴组”时： 相应轴组无法动作。在相应轴组的动作中减速停止	
系统定义变量	变量名称		数据类型		名称
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	指令的输入变量“Position”指定的参数超过正方向软件限制的范围		请修正相应指令的输入变量“Position”指定的参数，以将其控制在正方向软件限制的范围		请将指令的输入变量“Position”指定的输入参数控制在正方向软件限制的范围
	起始位置超过正方向软件限制的范围，执行了指定与软件限制的范围相反方向的动作的指令		请修正程序，以确保相应指令下朝正方向软件限制的范围移动		请编写程序，以确保起始位置超过正方向软件限制的范围时，朝正方向软件限制的范围移动
	已指定通过点的MC_MoveCircular2D(2轴圆弧插补)指令的输入变量“AuxPoint”指定的参数超过正方向软件限制的范围		请修正相应指令的输入变量“AuxPoint”指定的参数，以将其控制在正方向软件限制的范围		请将已指定通过点的MC_MoveCircular2D(2轴圆弧插补)指令的输入变量“AuxPoint”指定的参数控制在正方向软件限制的范围
附加信息	附属信息1: 因发生源详情而异 轴时: “0” 轴组时: 逻辑轴号				
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	指定的位置超过负方向软件限制的范围					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时： 相应轴不可动作的轴。动作中减速停止 具体发生源为“轴组”时： 相应轴组无法动作。在相应轴组的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入变量“Position”指定的参数超过负方向软件限制的范围		请修正相应指令的输入变量“Position”指定的参数，以将其控制在负方向软件限制的范围		请将指令的输入变量“Position”指定的输入参数控制在负方向软件限制的范围	
	起始位置超过负方向软件限制的范围，执行了指定与软件限制的范围相反方向的运动的指令		请修正程序，以确保相应指令下朝负方向软件限制的范围移动		请编写程序，以确保起始位置超过负方向软件限制的范围时，朝负方向软件限制的范围移动	
	已指定通过点的MC_MoveCircular2D(2轴圆弧插补)指令的输入变量“AuxPoint”指定的参数超过负方向软件限制的范围		请修正相应指令的输入变量“AuxPoint”指定的参数，以将其控制在负方向软件限制的范围		请将已指定通过点的MC_MoveCircular2D(2轴圆弧插补)指令的输入变量“AuxPoint”指定的参数控制在负方向软件限制的范围	
附加信息	附属信息1：因发生源详情而异 轴时：“0” 轴组时：逻辑轴号					
注意事项/备注	无					

事件名称	指令位置溢出/正在发生下溢		事件代码	64420000Hex		
内容	指令位置溢出/正在发生下溢时，执行了定位、溢出/下溢方向的指令或无法确定方向的指令					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴不可动作。在相应轴的动作中减速停止		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令位置溢出/正在发生下溢时进行了以下操作 • 执行了定位的指令 • 执行了溢出/下溢方向的连续控制指令 • 执行了无法确定方向的指令(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预设解除溢出/下溢状态		请避免发生溢出/下溢	
附加信息	无					
注意事项/备注	无					

事件名称	正方向限制输入中		事件代码	64430000Hex		
内容	正方向限制输入为“ON”的状态下执行了正方向动作的指令					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时： 相应轴不可动作 具体发生源为“轴组”时： 相应轴组无法动作		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	正方向限制输入为“ON”的状态下，执行了正方向动作的指令；或正方向限制输入为“ON”的状态下，执行了不指定动作方向的指令。正方向限制输入为“ON”的状态下，执行了轴组动作指令		请执行异常解除，朝负方向进行复位动作。执行轴组动作指令过程中发生异常时，请将禁用发生异常的轴组，再执行上述复位动作。再次发生本异常时，请确认正方向限制信号的连接、正方向限制输入的逻辑设定、启动指令的执行条件，再修正错误。请同时确认轴参数和从站侧的逻辑设定。		请确认正方向限制信号的连接、正方向限制输入的逻辑设定、启动指令的执行条件无误。请同时确认轴参数和从站侧的逻辑设定。	
附加信息	附属信息1：因发生源详情而异 轴时：“0” 轴组时：逻辑轴号					
注意事项/备注	无					

事件名称	负方向限制输入中		事件代码	64440000Hex		
内容	负方向限制输入为“ON”的状态下，执行了负方向动作的指令					
发生源	运动控制功能模块		发生源详情	轴/轴组	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	具体发生源为“轴”时： 相应轴不可动作 具体发生源为“轴组”时： 相应轴组无法动作		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
	_MC_GRP[*].MFaultLvl.Active		BOOL		正在发生轴组轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	负方向限制输入为“ON”的状态下，执行了负方向动作的指令；或负方向限制输入为“ON”的状态下，执行了不指定动作方向的指令。负方向限制输入为“ON”的状态下，执行了轴组动作指令		请执行异常解除，朝正方向进行复位动作。执行轴组动作指令过程中发生异常时，请将禁用发生异常的轴组，再执行上述复位动作。再次发生本异常时，请确认负方向限制信号的连接、负方向限制输入的逻辑设定、启动指令的执行条件，再修正错误。请同时确认轴参数和从站侧的逻辑设定。		请确认负方向限制信号的连接、负方向限制输入的逻辑设定、启动指令的执行条件无误。请同时确认轴参数和从站侧的逻辑设定。	
附加信息	附属信息1：因发生源详情而异 轴时：“0” 轴组时：逻辑轴号					
注意事项/备注	无					

事件名称	伺服主电路电源OFF状态			事件代码	74220000Hex	
内容	伺服驱动器的主电路电源为“OFF”的状态下，执行了伺服ON					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	轻度故障	恢复方法	异常解除	日志类别	系统
发生后的影响	用户程序	继续	动作	相应轴伺服“OFF”		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		正在发生轴轻度故障	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	伺服驱动器的主电路电源为“OFF”状态下，执行了伺服ON		请接通发生本异常的轴的伺服驱动器主电路电源，再执行伺服ON		请接通伺服驱动器的主电路电源，再执行伺服ON	
附加信息	无					
注意事项/备注	无					

事件名称	反馈位置正在发生溢出/下溢			事件代码	57220000Hex *1	
内容	反馈位置正在发生溢出/下溢时启动了无法执行的指令					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别	系统
发生后的影响	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	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	反馈位置正在发生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，以避免发生溢出或下溢	
附加信息	无					
注意事项/备注	无					

*1. 单元版本Ver.1.06以上的CPU单元发生的事件。

事件名称	开关结构体的轨段编号超过设定范围			事件代码	57230000Hex *1	
内容	运动控制指令的输入输出变量“Switches”指定的“TrackNumber”值超过范围					
发生源	运动控制功能模块		发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别	系统
发生后的影响	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	
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生	
	指令的输入输出变量指定的结构体型变量的结构要素值超过范围		请修正数值，以避免相应指令的输入输出变量中指定的结构体型变量的结构要素超过有效范围		请避免指令的输入输出变量中指定的结构体型变量的结构要素值超过有效范围	
附加信息	无					
注意事项/备注	无					

*1. 单元版本Ver.1.06以上的CPU单元发生的事件。

事件名称	开关结构体的ON开始位置超过设定范围			事件代码	57240000Hex *1	
内容	运动控制指令的输入输出变量“Switches”指定的“FirstOnPosition”值超过范围					
发生源	运动控制功能模块	发生源详情	轴	检测时间	启动指令时	
异常的属性	重要程度	监控信息	恢复方法	-	日志类别	系统
发生后的影响	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		
发生原因及其处理	发生原因(推测原因)	处理措施		防止再次发生		
	指令的输入输出变量指定的结构体型变量的结构要素值超过范围	请修正数值，以避免相应指令的输入输出变量中指定的结构体型变量的结构要素超过有效范围		请避免指令的输入输出变量中指定的结构体型变量的结构要素值超过有效范围		
附加信息	无					
注意事项/备注	无					

*1. 单元版本Ver.1.06以上的CPU单元发生的事件。

事件名称	开关结构体的ON结束位置超过设定范围			事件代码	57250000Hex *1	
内容	运动控制指令的输入输出变量“Switches”指定的“LastOnPosition”值超过范围					
发生源	运动控制功能模块	发生源详情	轴	检测时间	启动指令时	
异常的属性	重要程度	监控信息	恢复方法	-	日志类别	系统
发生后的影响	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		
发生原因及其处理	发生原因(推测原因)	处理措施		防止再次发生		
	指令的输入输出变量指定的结构体型变量的结构要素值超过范围	请修正数值，以避免相应指令的输入输出变量中指定的结构体型变量的结构要素超过有效范围		请避免指令的输入输出变量中指定的结构体型变量的结构要素值超过有效范围		
附加信息	无					
注意事项/备注	无					

*1. 单元版本Ver.1.06以上的CPU单元发生的事件。

事件名称	开关结构体的方向选择超过范围			事件代码	57260000Hex *1	
内容	运动控制指令的输入输出变量“Switches”指定的“AxisDirection”值超过范围					
发生源	运动控制功能模块	发生源详情	轴	检测时间	启动指令时	
异常的属性	重要程度	监控信息	恢复方法	-	日志类别	系统
发生后的影响	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		
发生原因及其处理	发生原因(推测原因)	处理措施		防止再次发生		
	指令的输入输出变量指定的结构体型变量的结构要素值超过范围	请修正数值，以避免相应指令的输入输出变量中指定的结构体型变量的结构要素超过有效范围		请避免指令的输入输出变量中指定的结构体型变量的结构要素值超过有效范围		
附加信息	无					
注意事项/备注	无					

*1. 单元版本Ver.1.06以上的CPU单元发生的事件。

事件名称	开关结构体的开关模式选择超过范围			事件代码	57270000Hex *1
内容	运动控制指令的输入输出变量 “Switches” 指定的 “CamSwitchMode” 值超过范围				
发生源	运动控制功能模块	发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别
发生后的影响	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	
发生原因及其处理	发生原因(推测原因)	处理措施		防止再次发生	
	指令的输入输出变量指定的结构体型变量的结构要素值超过范围	请修正数值，以避免相应指令的输入输出变量中指定的结构体型变量的结构要素超过有效范围		请避免指令的输入输出变量中指定的结构体型变量的结构要素值超过有效范围	
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.06以上的CPU单元发生的事件。

事件名称	开关结构体的ON时间超过设定范围			事件代码	57280000Hex *1
内容	运动控制指令的输入输出变量 “Switches” 指定的 “Duration” 值超过范围				
发生源	运动控制功能模块	发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别
发生后的影响	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	
发生原因及其处理	发生原因(推测原因)	处理措施		防止再次发生	
	指令的输入输出变量指定的结构体型变量的结构要素值超过范围	请修正数值，以避免相应指令的输入输出变量中指定的结构体型变量的结构要素超过有效范围		请避免指令的输入输出变量中指定的结构体型变量的结构要素值超过有效范围	
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.06以上的CPU单元发生的事件。

事件名称	轨段选项结构体的ON时刻补偿超过设定范围			事件代码	57290000Hex *1
内容	运动控制指令的输入输出变量 “TrackOptions” 指定的 “OnCompensation” 值超过范围				
发生源	运动控制功能模块	发生源详情	轴	检测时间	启动指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别
发生后的影响	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	
发生原因及其处理	发生原因(推测原因)	处理措施		防止再次发生	
	指令的输入输出变量指定的结构体型变量的结构要素值超过范围	请修正数值，以避免相应指令的输入输出变量中指定的结构体型变量的结构要素超过有效范围		请避免指令的输入输出变量中指定的结构体型变量的结构要素值超过有效范围	
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.06以上的CPU单元发生的事件。

事件名称	轨段选项结构体的OFF时刻补偿超过设定范围		事件代码	572A0000Hex *1	
内容	运动控制指令的输入输出变量“TrackOptions”指定的“OffCompensation”值超过范围				
发生源	运动控制功能模块		发生源详情	轴	检测时间 启动指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别 系统
发生后的影响	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
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	指令的输入输出变量指定的结构体型变量的结构要素值超过范围		请修正数值,以避免相应指令的输入输出变量中指定的结构体型变量的结构要素超过有效范围		请避免指令的输入输出变量中指定的结构体型变量的结构要素值超过有效范围
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.06以上的CPU单元发生的事件。

事件名称	开关结构体型变量的数组元素数超过范围		事件代码	572B0000Hex *1	
内容	运动控制指令的输入输出变量“Switches”指定的结构体型变量的数组元素数超过范围				
发生源	运动控制功能模块		发生源详情	轴	检测时间 启动指令时
异常的属性	重要程度	监控信息	恢复方法	异常解除	日志类别 系统
发生后的影响	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
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	指令的输入输出变量指定的结构体型变量的数组元素数超过范围		请进行修正,以避免相应指令的输入输出变量中指定的结构体型变量的数组元素数超过范围		请避免指令的输入输出变量中指定的结构体型变量的数组元素数超过范围
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.06以上的CPU单元发生的事件。

事件名称	输出信号结构体型变量的数组元素数超过范围		事件代码	572C0000Hex *1	
内容	运动控制指令的输入输出变量“Outputs”指定的结构体型变量的数组元素数超过范围				
发生源	运动控制功能模块		发生源详情	轴	检测时间 启动指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别 系统
发生后的影响	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
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	指令的输入输出变量指定的结构体型变量的数组元素数超过范围		请进行修正,以避免相应指令的输入输出变量中指定的结构体型变量的数组元素数超过范围		请避免指令的输入输出变量中指定的结构体型变量的数组元素数超过范围
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.06以上的CPU单元发生的事件。

事件名称	轨段选项结构体型变量的数组元素数超过范围			事件代码	572D0000Hex ^{*1}
内容	运动控制指令的输入输出变量“TrackOptions”指定的结构体型变量的数组元素数超过范围				
发生源	运动控制功能模块		发生源详情	轴	检测时间 启动指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别 系统
发生后的影响	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
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	指令的输入输出变量指定的结构体型变量的数组元素数超过范围		请进行修正, 以避免相应指令的输入输出变量中指定的结构体型变量的数组元素数超过范围		请避免指令的输入输出变量中指定的结构体型变量的数组元素数超过范围
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.06以上的CPU单元发生的事件。

事件名称	输出信号与轨段选项的数组元素数不一致			事件代码	572E0000Hex ^{*1}
内容	运动控制指令的输入输出变量“Outputs”与“TrackOptions”指定的结构体型变量的数组元素数不一致				
发生源	运动控制功能模块		发生源详情	轴	检测时间 启动指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别 系统
发生后的影响	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
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	指令的输入输出变量指定的输出信号结构体型变量与机架选项结构体型变量的数组元素数不一致		请进行修正, 使相应指令的输入输出变量中指定的输出信号结构体型变量与轨段选项结构体型变量的数组元素数一致		请进行设定, 使指令的输入输出变量中指定的输出信号结构体型变量与轨段选项结构体型变量的数组元素数一致
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.06以上的CPU单元发生的事件。

事件名称	开关结构体的同一轨段指定数超过范围			事件代码	57310000Hex ^{*1}
内容	运动控制指令的输入输出变量“Switches”指定的“TrackNumber”的同一轨段编号数量超过范围				
发生源	运动控制功能模块		发生源详情	轴	检测时间 启动指令时
异常的属性	重要程度	监控信息	恢复方法	-	日志类别 系统
发生后的影响	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
发生原因及其处理	发生原因(推测原因)		处理措施		防止再次发生
	指令的输入输出变量指定的开关结构体型变量的“TrackNumber”指定的同一轨段编号数量超过了1个轨段可指定的个数范围		请修正数值, 以避免“TrackNumber”指定的同一轨段编号数量超过1个轨段可指定的个数		请避免“TrackNumber”指定的同一轨段编号数量超过1个轨段可指定的个数
附加信息	无				
注意事项/备注	无				

*1. 单元版本Ver.1.06以上的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，请参阅  “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-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，请参阅 □ “NJ/NX系列 CPU单元 用户手册 运动控制篇(SBCE-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]也是如此。

变量名称	数据类型	名称	功能
_MC_AX[0-255]	_sAXIS_REF	轴变量	
Status	_sAXIS_REF_STA	轴状态	
Disabled	BOOL	轴无效	轴因伺服OFF而停止时变为TRUE。
Standstill	BOOL	停止中	轴因伺服ON而停止时变为TRUE。
Discrete	BOOL	定位动作中	在向目标位置执行定位控制时变为TRUE。 也包括定位动作中将超调值设定为“0”而使速度为“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。 速度控制或转矩控制时变为此状态。 也包括将目标速度设为“0”而使速度为“0”的状态、以及连续动作中将超调值设定为“0”而使速度为“0”的状态。
Synchronized	BOOL	同步动作中	执行同步控制时变为TRUE。 也包括同步控制指令切换后的同步等待状态。
Homing	BOOL	原点复位中	通过MC_Home指令、MC_HomeWithParameter指令执行原点复位时变为TRUE。
Stopping	BOOL	减速停止中	在通过MC_Stop指令、MC_TouchProbe指令停止轴动作之前的状态下变为TRUE。 也包括通过MC_Stop指令停止轴动作后Execute变为TRUE时。 在该状态下不能启动轴动作指令(CommandAborted = TRUE)。
ErrorStop	BOOL	错误减速停止中	通过MC_ImmediateStop指令、或者在发生轴轻度故障时(_MC_AX[*].MFAuliLvl.Active=TRUE)，轴停止之前、以及停止后变为TRUE。 在该状态下不能启动轴动作指令(CommandAborted = TRUE)。
Coordinated	BOOL	多轴协调动作中	在通过多轴协调指令使轴组有效的状态下变为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]也是如此。

变量名称	数据类型	名称	功能
_MC_GRP[0-63]	_sGROUP_REF	轴组变量	
Status	_sGROUP_REF_STA	轴组状态	
Disabled	BOOL	不启用轴组	因禁用轴组而停止时变为TRUE。
Standby	BOOL	停止中	轴组动作指令停止时变为TRUE。 与轴组构成轴的伺服ON/OFF状态无关。
Moving	BOOL	动作中	在向目标位置执行轴组动作指令时变为TRUE。 也包括到位等待的状态、以及因超调而使速度为“0”的状态。
Stopping	BOOL	减速停止中	在通过MC_GroupStop指令停止轴组动作之前的状态下变为TRUE。 也包括通过MC_GroupStop指令停止轴组动作后Execute变为TRUE时。 在该状态下不能启动轴组动作指令(CommandAborted = TRUE)。
ErrorStop	BOOL	错误减速停止中	通过MC_GroupImmediateStop指令、或者在发生轴组轻度故障时(_MC_GRP[*].MFAultLvl.Active=TRUE)，轴组停止之前、以及停止后变为TRUE。 在此状态下不能启动轴组动作指令(CommandAborted = TRUE)。

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，还将对各轴的状态及各轴组的状态的切换目标进行说明。

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，请参阅 □ “NJ/NX 系列 CPU 单元 用户手册 运动控制篇(SBCE-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 : 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)

轴组状态

- a : 不启用轴组(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	不启用轴组	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
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_CamIn*6	H	H	H	H	-	H	G	H	-	-	-	-	-	-
MC_CamOut	H	H	H	H	C*7	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_GearIn*6	H	H	H	H	-	H	G	H	-	-	-	-	-	-
MC_GearInPos	H	E	H	H	H	H	G	H	I	a	e	e	e	e
MC_GearInPos*6	H	H	H	H	-	H	G	H	-	-	-	-	-	-
MC_GearOut	H	H	H	H	C*8	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_MoveLink*6	H	H	H	H	-	H	G	H	-	-	-	-	-	-
MC_CombineAxes	H	E	E	E	E	H	G	H	I	a	e	e	e	e
MC_CombineAxes*6	H	H	H	H	-	H	G	H	-	-	-	-	-	-
MC_Phasing	H	H	H	H	E*9	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_TorqueControl*6	H	H	H	H	-	H	G	H	-	-	-	-	-	-
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*10	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_SyncMoveVelocity*6	H	H	H	H	-	H	G	H	-	-	-	-	-	-
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	*11	I	a	b	c	d	e
MC_ChangeAxisUse (未使用轴→使用轴)	A	-	-	-	-	-	-	-	-	-	-	-	-	-

A-3 多重启动可否

A

A-3-2 状态切换和多重启动的可否

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	不启用轴组	停止中	动作中	减速停止中	错误减速停止中
MC_ChangeAxisUse (使用轴→未使用轴)	-	*12	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	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 轴功能限制设定为“1：可执行单轴控制/指定动作曲线的控制(仅指令位置输出)”时。

*7 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。

*8 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。

*9 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。

*10 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。

*11 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。

12 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

对于NX系列CPU单元，以_MC_AX[*]开头的变量名可能为_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	不启用轴组	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。

单轴位置控制轴无法设定为轴组构成轴, 因此发生异常。

*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 : 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
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

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_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

下面将介绍CPU单元和Sysmac Studio的各版本更新相关的信息。

因版本更新而变更规格的指令及新指令一览

指令的适用情况和规格可能因CPU单元和Sysmac Studio的版本而异。该内容如下所示。


○：该版本新增

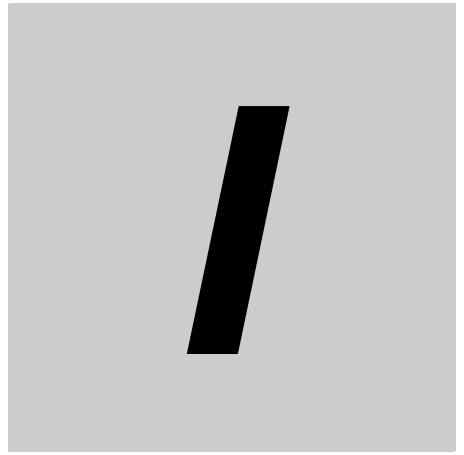
△：该版本规格变更

种类	指令	名称	版本		页码
			CPU单元	Sysmac Studio	
通用指令	MC_Write	写入MC设定	△ Ver.1.01 △ Ver.1.10	△ Ver.1.02 △ Ver.1.12	5-13
	MC_GenerateCamTable	凸轮表生成	○ Ver.1.08	○ Ver.1.09	5-18
	MC_WriteAxisParameter	轴参数写入	○ Ver.1.08	○ Ver.1.09	5-45
	MC_ReadAxisParameter	轴参数读取	△ Ver.1.10	△ Ver.1.12	5-57
轴组指令	MC_GroupReadPosition	轴组位置获取	○ Ver.1.01	○ Ver.1.02	4-84
	MC_ChangeAxesInGroup	写入轴组构成轴			4-88
	MC_GroupSyncMove Absolute	轴组周期同步绝对位置控制			4-92
轴指令	MC_Power	可运行	△ Ver.1.10	△ Ver.1.12	3-3
	MC_SetPosition	当前位置变更			3-143
	MC_ResetFollowingError	偏差计数器复位			3-154
	MC_CamIn	凸轮动作开始			3-161
	MC_CombineAxes	加减运算定位			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	启用外部锁定	△ Ver.1.10	△ Ver.1.12	3-329
	MC_HomeWithParameter	参数指定原点复位	○ Ver.1.03	○ Ver.1.04	3-37
	MC_SyncMoveAbsolute	周期同步绝对定位			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.1.07	3-383
			△ Ver.1.09	△ Ver.1.10	
	MC_TimeStampToPos	时间戳→轴位置计算	○ Ver.1.06	○ Ver.1.07	3-402
MC_PeriodicSyncVariables	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组合使用CPU单元Ver.1.05以上版本与Sysmac Studio Ver.1.06以上版本。

对NX系列位置接口单元使用运动控制指令时，部分指令的功能受限或无法使用。详情请参阅  “NX系列 位置接口单元 用户手册(SBCE-374)”。



索引



索引

Numerics

2轴圆弧插补 4-44

B

保存凸轮表 5-8
 不启用外部锁定 3-350
 不启用轴组 4-6

C

参数指定原点复位 3-37
 超调值设定 3-149, 4-80
 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-18
 5420Hex A-2, A-18
 5421Hex A-2, A-19
 5422Hex A-2, A-19
 5423Hex A-2, A-19
 5424Hex A-2, A-20
 5425Hex A-2, A-20
 5427Hex A-2, A-20
 5428Hex A-3, A-21
 5429Hex A-3, A-21
 542AHex A-3, A-21
 542BHex A-3, A-22
 542CHex A-3, A-22
 542DHex A-3, A-22
 542EHex A-3, A-23
 542FHex A-3, A-23
 5430Hex A-3, A-23
 5431Hex A-3, A-24
 5432Hex A-3, A-24
 5433Hex A-3, A-25
 5434Hex A-3, A-25
 5435Hex A-3, A-25
 5436Hex A-3, A-26
 5437Hex A-4, A-26
 5438Hex A-4, A-26
 5439Hex A-4, A-27
 543AHex A-4, A-27
 543BHex A-4, A-28
 543CHex A-4, A-28
 543DHex A-4, A-29
 543EHex A-4, A-29
 543FHex A-4, A-30
 5440Hex A-4, A-30
 5441Hex A-5, A-31
 5442Hex A-5, A-31
 5443Hex A-5, A-32

5444Hex A-5, A-32
 5445Hex A-5, A-33
 5446Hex A-5, A-33
 5447Hex A-5, A-33
 5448Hex A-5, A-34
 5449Hex A-5, A-34
 544AHex A-5, A-35
 544CHex A-5, A-35
 544DHex A-6, A-35
 544EHex A-6, A-36
 544FHex A-6, A-36
 5450Hex A-6, A-36
 5451Hex A-6, A-37
 5453Hex A-6, A-37
 5454Hex A-6, A-37
 5455Hex A-6, A-38
 5456Hex A-6, A-38
 5457Hex A-6, A-38
 5458Hex A-6, A-39
 5459Hex A-6, A-39
 545AHex A-6, A-40
 545BHex A-6, A-40
 545CHex A-7, A-40
 545DHex A-7, A-41
 545EHex A-7, A-41
 545FHex A-7, A-41
 5460Hex A-7, A-42
 5461Hex A-7, A-42
 5462Hex A-7, A-43
 5463Hex A-7, A-43
 5464Hex A-7, A-44
 5465Hex A-7, A-44
 5466Hex A-7, A-45
 5467Hex A-8, A-45
 5468Hex A-8, A-46
 5469Hex A-8, A-46
 546AHex A-8, A-46
 546BHex A-8, A-47
 546CHex A-8, A-47
 546DHex A-8, A-47
 546EHex A-8, A-48
 546FHex A-8, A-48
 5470Hex A-8, A-49
 5471Hex A-8, A-49
 5472Hex A-8, A-49
 5474Hex A-8, A-50
 5475Hex A-8, A-50
 5476Hex A-9, A-50
 5478Hex A-9, A-51
 5479Hex A-9, A-51
 547AHex A-9, A-52
 547BHex A-9, A-52
 547CHex A-9, A-52
 547DHex A-9, A-53
 547EHex A-9, A-53

547FHex A-9, A-54
 5480Hex A-9, A-54
 5481Hex A-9, A-54
 5482Hex A-9, A-55
 5483Hex A-10, A-55
 5484Hex A-10, A-55
 5487Hex A-10, A-56
 5488Hex A-10, A-56
 5489Hex A-10, A-56
 548AHex A-10, A-57
 548BHex A-10, A-57
 548CHex A-10, A-57
 548DHex A-10, A-58
 548EHex A-10, A-58
 548FHex A-10, A-58
 5490Hex A-10, A-59
 5491Hex A-10, A-59
 5492Hex A-11, A-59
 5493Hex A-11, A-60
 5494Hex A-11, A-60
 5495Hex A-11, A-60
 5496Hex A-11, A-61
 5497Hex A-11, A-61
 5498Hex A-11, A-61
 5499Hex A-11, A-62
 549AHex A-11, A-62
 549BHex A-11, A-62
 549CHex A-11, A-63
 549DHex A-11, A-63
 549EHex A-11, A-64
 5700Hex A-11, A-64
 5702Hex A-12, A-65
 5703Hex A-12, A-65
 5720Hex A-12, A-66
 5721Hex A-12, A-67
 5722Hex A-14, A-78
 5723Hex A-15, A-78
 5724Hex A-15, A-79
 5725Hex A-15, A-79
 5726Hex A-15, A-79
 5727Hex A-15, A-80
 5728Hex A-15, A-80
 5729Hex A-15, A-80
 572AHex A-15, A-81
 572BHex A-15, A-81
 572CHex A-15, A-81
 572DHex A-15, A-82
 572EHex A-16, A-82
 572FHex A-12, A-67
 5730Hex A-12, A-68
 5731Hex A-16, A-82
 573AHex A-12, A-68
 573BHex A-12, A-69
 573CHex A-12, A-70
 573DHex A-12, A-70
 573EHex A-13, A-71
 573FHex A-13, A-71
 5740Hex A-13, A-71
 5741Hex A-13, A-72

5742Hex A-13, A-72
 5743Hex A-13, A-73
 5749Hex A-13, A-73
 574AHex A-13, A-74
 574BHex A-13, A-74
 574CHex A-13, A-74
 6440Hex A-14, A-75
 6441Hex A-14, A-76
 6442Hex A-14, A-76
 6443Hex A-14, A-77
 6444Hex A-14, A-77
 7422Hex A-14, A-78
 错误复位 3-375, 4-98

D

当前位置变更 3-143
 单轴位置控制轴 2-26, 2-28
 电子齿轮 3-240
 定位 3-42, 3-47, 3-73, 3-102

G

高速原点复位 3-95

J

加减运算定位 3-288
 解除凸轮动作 3-215
 绝对值定位 3-47
 绝对值直线插补 4-38

K

可运行 3-3

L

立即停止 3-138

M

MC_AbortTrigger(不启用外部锁定) 3-350
 MC_AxesObserve(轴间偏差监视) 3-354
 MC_CamIn(凸轮动作开始) 3-161
 MC_CamOut(解除凸轮动作) 3-215
 MC_ChangeAxesInGroup(轴组构成轴写入) 4-88
 MC_ChangeAxisUse(轴使用变更) 3-379
 MC_CombineAxes(加减运算定位) 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(不启用轴组) 4-6
 MC_GroupEnable(启用轴组) 4-2
 MC_GroupImmediateStop(轴组立即停止) 4-76
 MC_GroupReadPosition(轴组位置获取) 4-84
 MC_GroupReset(轴组错误复位) 4-98

MC_GroupSetOverride(轴组超调值设定)	4-80
MC_GroupStop(轴组强制停止)	4-68
MC_GroupSyncMoveAbsolute (轴组周期同步绝对位置控制)	4-92
_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)	3-37
MC_ImmediateStop(立即停止)	3-138
MC_Move(定位)	3-42
MC_MoveAbsolute(绝对值定位)	3-47
MC_MoveCircular2D(2轴圆弧插补)	4-44
MC_MoveFeed(中断标准定位)	3-102
MC_MoveJog(微动移动)	3-8
MC_MoveLinear(直线插补)	4-11
MC_MoveLinearAbsolute(绝对值直线插补)	4-38
MC_MoveLinearRelative(相对值直线插补)	4-41
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)	5-57
MC_Reset(轴错误复位)	3-375
MC_ResetFollowingError(偏差计数器复位)	3-154
MC_SaveCamTable(保存凸轮表)	5-8
MC_SetCamTableProperty(更新凸轮表属性)	5-2
MC_SetOverride(超调值设定)	3-149
MC_SetPosition(当前位置变更)	3-143
MC_SetTorqueLimit(转矩限制)	3-316
MC_Stop(强制停止)	3-129
MC_SyncMoveAbsolute(周期同步绝对定位)	3-369
MC_SyncMoveVelocity(周期同步速度控制)	3-360
MC_SyncOffsetPosition(周期性同步位置偏置补偿)	3-423
MC_TimeStampToPos(时间戳→轴位置计算)	3-402
MC_TorqueControl(转矩控制)	3-305
MC_TouchProbe(启用外部锁定)	3-329
MC_Write(写入MC设定)	5-13
MC_WriteAxisParameter(轴参数写入)	5-45
MC_ZoneSwitch(区域监视)	3-323

P

偏差计数器复位	3-154
---------	-------

Q

强制停止	3-129, 4-68
启用外部锁定	3-329
启用轴组	4-2
区域监视	3-323

S

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
凸轮表生成	5-18
凸轮表属性更新	5-2
凸轮动作开始	3-161

W

微动移动	3-8
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	4-41
相位补偿	3-299
写入MC设定	5-13
写入设定	5-13
写入轴组构成轴	4-88

Y

原点复位	3-15
圆弧插补	4-44

Z

直线插补	4-11, 4-38, 4-41
执行ID	2-10
中断标准定位	3-102
轴变量任务间固定周期同步	3-415
轴参数读取	5-57
轴参数写入	5-45
轴错误复位	3-375
轴间偏差监视	3-354
周期同步绝对定位	3-369
周期同步速度控制	3-360
周期性同步位置偏置补偿	3-423
轴使用变更	3-379
轴组超调值设定	4-80
轴组错误复位	4-98
轴组立即停止	4-76
轴组强制停止	4-68
轴组位置获取	4-84
轴组周期同步绝对位置控制	4-92
转矩控制	3-305
转矩限制	3-316
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、结构部件不提供维修服务。)
 - (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。

欧姆龙自动化(中国)有限公司

欧姆龙自动化(中国)有限公司北京分公司
欧姆龙自动化(中国)有限公司天津分公司
欧姆龙自动化(中国)有限公司广州分公司



服务



资讯

技术咨询

网 址: <http://www.fa.omron.com.cn>
400咨询热线: 400-820-4535

上海总公司	021-60230333	太原事务所	0351-5229870
南京事务所	025-83240556	天津分公司	022-83191580
徐州事务所	0516-83736516	沈阳事务所	024-22815131
武汉事务所	027-82282145	西安事务所	029-88851505
苏州事务所	0512-68669277	银川联络处	0951-5670076
昆山事务所	0512-50110866	成都事务所	028-86765345
杭州事务所	0571-87652855	重庆事务所	023-68796406
宁波事务所	0574-27888220	大连事务所	0411-39948181
温州事务所	0577-88919195	昆明事务所	0871-63527224
合肥事务所	0551-63454209	兰州事务所	0931-8720101
长沙事务所	0731-84585551	长春事务所	0431-81928301
无锡事务所	0510-85169303	乌鲁木齐事务所	0991-5198587
张家港事务所	0512-56313157	贵阳事务所	0851-84812320
南昌事务所	0791-86304711	广州分公司	020-87557798
郑州事务所	0371-65585192	深圳事务所	0755-26948238
北京分公司	010-57395399	厦门事务所	0592-2686709
唐山事务所	0315-4795118	东莞事务所	0769-22423200
石家庄事务所	0311-86906790	佛山事务所	0757-83305298
济南事务所	0531-82929795	中山事务所	0760-88224545
青岛事务所	0532-66775819	福州事务所	0591-88088551
烟台事务所	0535-6865018	南宁事务所	0771-5531371

汕头事务所 0754-88706001 香港事务所 00852-23753827

特约店

注:规格如有变更,恕不另行通知。请以最新产品说明书为准。