OMRON

机器自动化控制器

NJ/NX系列

CPU单元

用户手册
运动控制篇

NX701-17□□
NX701-16□□
NJ501-□5□□
NJ501-□4□□
NJ501-□3□□
NJ301-12□□
NJ301-11□□
NJ101-10□□

CPU单元



SYSMAC
always in control

SBCE-CN5-363M

声明

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

商标

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

此外，记载的公司名称和产品名称为各公司的注册商标或商标。

著作权

Screen shot已取得Microsoft Corporation的使用许可。

前言

感谢您购买 NJ/NX 系列 CPU 单元。
本手册记载了使用 NJ/NX 系列 CPU 单元的运动控制功能模块所必需的信息。请在使用前仔细阅读本手册，在充分理解功能和性能的基础上，灵活用于系统构建。
此外，请妥善保管本手册，阅读之后放于随时可取阅之处。

阅读对象

本手册提供给下列阅读对象。
对象为具有电工专业知识的人员 （合格的电气工程师或具有同等知识的人员）：

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

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

对象产品

本手册的对象为以下产品：
NX 系列 CPU 单元

- NX701-17□□
- NX701-16□□

NJ 系列 CPU 单元

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

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

分册构成

本产品的手册如下表所示分册。请根据不同目的选读，灵活运用本产品。
本产品的操作主要使用自动化软件 Sysmac Studio。关于 Sysmac Studio，请参考 🕮《Sysmac Studio Version 1 操作手册 （SBCA-CN5-362）》。

使用目的	手册											
	基本信息											
	NX 系列 CPU 单元 用户手册 硬件篇	NJ 系列 CPU 单元 用户手册 硬件篇	NJ/NX 系列 CPU 单元 用户手册 软件篇	NJ/NX 系列 指令基准手册 基本篇	NJ/NX 系列 CPU 单元 用户手册 运动控制篇	NJ/NX 系列 指令基准手册 运动篇	NJ/NX 系列 CPU 单元 内置 EtherCAT 端口 用户手册	NJ/NX 系列 CPU 单元 内置 EtherNet/IP 端口用户手册	NJ 系列 数据库连接 CPU 单元 用户手册	NJ 系列 配备 SECS/GEM 的 CPU 单元 用户手册	NJ 系列 NJ Robotics CPU 单元 用户手册	NJ/NX 系列 故障排除手册
了解 NX 系列的概要	○											
了解 NJ 系列的概要		○										
进行安装、设置和硬件设定	○	○										
进行运动控制时					○							
使用 EtherCAT 时							○					
使用 EtherNet/IP 时								○				
使用数据库连接服务时									○			
使用 GEM 服务时										○		
进行机器人控制时											○	
进行软件设定			○									
进行运动控制时					○							
使用 EtherCAT 时							○					
使用 EtherNet/IP 时								○				
使用数据库连接服务时									○			
使用 GEM 服务时										○		
进行机器人控制时											○	
创建用户程序			○	○								
进行运动控制时					○	○						
使用 EtherCAT 时							○					
使用 EtherNet/IP 时								○				
使用数据库连接服务时									○			
使用 GEM 服务时										○		
进行机器人控制时											○	
组合异常处理时												○
进行动作确认、调试时			○									
进行运动控制时					○							
使用 EtherCAT 时							○					
使用 EtherNet/IP 时								○				
使用数据库连接服务时									○			
使用 GEM 服务时										○		
进行机器人控制时											○	
了解异常管理功能或故障发生时的处理方法 *1	△	△	△		△		△	△	△	△		○

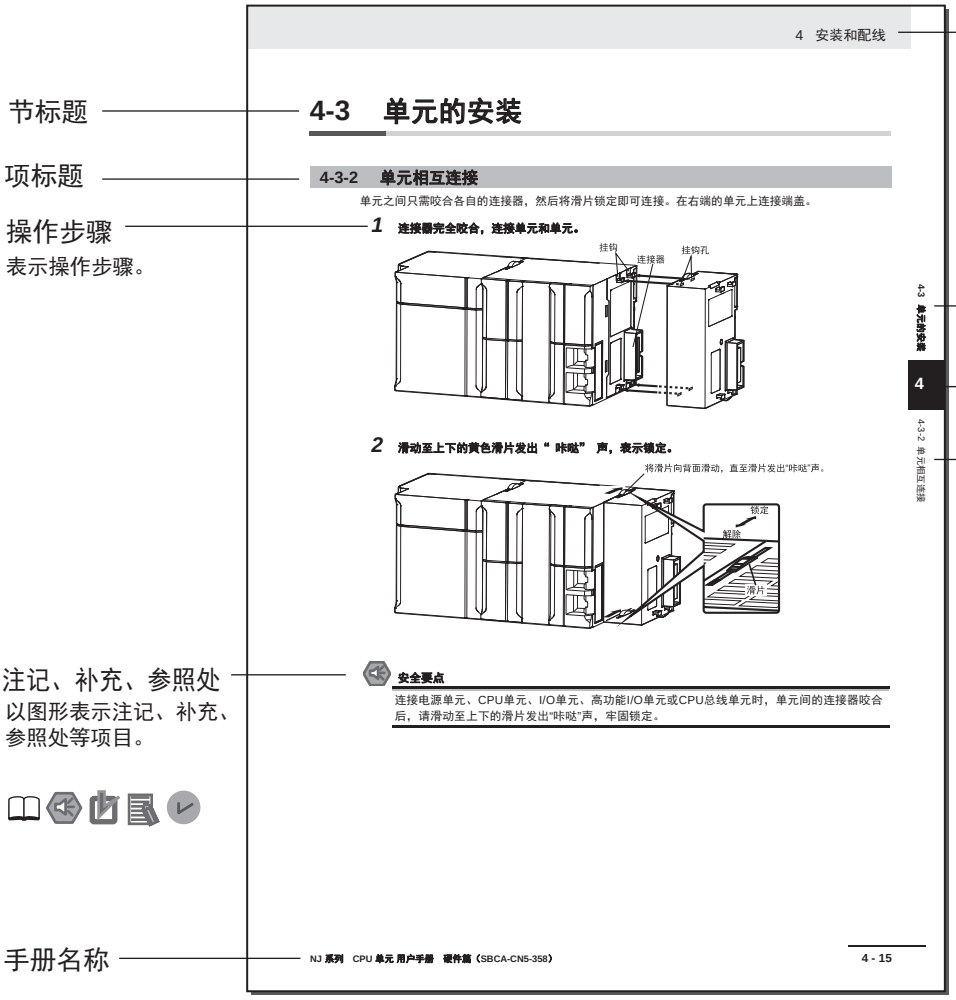
使用目的	手册											
	基本信息											
	NX 系列 CPU 单元 用户手册 硬件篇	NJ 系列 CPU 单元 用户手册 硬件篇	NJ/NX 系列 CPU 单元 用户手册 软件篇	NJ/NX 系列 指令基准手册 基本篇	NJ/NX 系列 CPU 单元 用户手册 运动控制篇	NJ/NX 系列 指令基准手册 运动篇	NJ/NX 系列 CPU 单元 内置 EtherCAT 端口 用户手册	NJ/NX 系列 CPU 单元 内置 EtherNet/IP 端口用户手册	NJ 系列 数据库连接 CPU 单元 用户手册	NJ 系列 配备 SECS/GEM 的 CPU 单元 用户手册	NJ 系列 NJ Robotics CPU 单元 用户手册	NJ/NX 系列 故障排除手册
了解维护作业	○	○										
进行运动控制时					○							
使用 EtherCAT 时							○					
使用 EtherNet/IP 时								○				

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

手册说明

页面构成

本手册各页的构成如下。



本页为说明用样本。与实际内容不同。

图标

本手册中使用的图标含义如下。



安全要点

表示为了安全使用，应该实施或避免的行为。



使用注意事项

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



参考

需要时阅读的项目。
了解后有助于使用的信息以及使用时可参考的内容。



版本相关信息

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



表示记载有详细信息、相关信息的位置。

表述注意事项

本手册中，从 Sysmac Studio 向控制器实体传送数据记载为 “ 下载 ”，从控制器实体向 Sysmac Studio 传送数据记载为 “ 上传 ”。

在 Sysmac Studio 中，无论是 “ 下载 ” 还是 “ 上传 ”，都使用 “ 同步 ” 功能。Sysmac Studio 的 “ 同步 ” 功能可以自动核对电脑上的数据和控制器实体上的数据，由用户选择传送方向。

目录结构

1 运动控制功能模块的概要

2 运动控制系统的构成和原理

3 轴和轴组的设定

4 通过Sysmac Studio确认配线

5 运动控制参数

6 运动控制程序

7 手动运行

8 定原点

9 运动控制功能

10 示例程序

11 异常的确认和处理

A 附录

I 索引

目录

前言 1
阅读对象 1
对象产品 1

分册构成 2

手册说明 4
页面构成 4
图标 5
表述注意事项 5

目录结构 7

目录 8

承诺事项 14

安全注意事项 16

安全要点 17

使用注意事项 18

法规和标准 19
日本国外的使用 19
符合 EU 指令 19
KC 标准的适用 20
船舶标准的适用 20
软件许可和著作权 20

版本 21
版本的确认方法 21
CPU 单元的单元版本和 Sysmac Studio 的版本 23

相关手册 24

手册修订记录 26

第 1 章　运动控制功能模块的概要

1-1　特点 1-2

1-2　系统构成 1-3

1-3　基本使用流程 1-4

1-4　规格 1-6
1-4-1　一般规格 1-6
1-4-2　性能规格 1-6
1-4-3　功能规格 1-8

第 2 章　运动控制系统的构成和原理

2-1　CPU 单元内部构成 2-2

2-2　运动控制系统的构成 2-3

2-3 运动控制系统的原理 2-4
2-3-1 CPU 单元的任务 2-4
2-3-2 运动控制的任务动作示例 2-10

2-4 与 EtherCAT 通信的关系 2-17
2-4-1 CAN application protocol over EtherCAT (CoE) 2-17
2-4-2 EtherCAT 主机功能和运动控制功能模块的关系 2-18
2-4-3 过程数据通信周期和运动控制周期的关系 2-20

第 3 章 轴和轴组的设定

3-1 轴 3-2
3-1-1 轴的概要 3-2
3-1-2 轴参数的概要 3-3
3-1-3 轴变量的概要 3-6
3-1-4 轴变量的同步 3-8
3-1-5 在用户程序中指定轴的方法 3-8

3-2 轴的设定步骤 3-9
3-2-1 轴的设定流程 3-9
3-2-2 设定步骤 3-9

3-3 轴组 3-19
3-3-1 轴组的概要 3-19
3-3-2 轴组参数的概要 3-20
3-3-3 轴组变量的概要 3-21
3-3-4 在用户程序中指定轴组的方法 3-22

3-4 轴组的设定步骤 3-24
3-4-1 轴组的设定流程 3-24
3-4-2 设定步骤 3-24

第 4 章 通过 Sysmac Studio 确认配线

4-1 Sysmac Studio 的功能 4-2
4-1-1 MC 调试功能一览 4-2
4-1-2 使用步骤 4-3
4-1-3 轴参数的设定示例 4-4
4-1-4 MC 调试功能的启动步骤 4-5

4-2 传感器信号的监视 4-7

4-3 电机动作的确认 4-8
4-3-1 伺服 ON 4-8
4-3-2 点动进给 4-9
4-3-3 原点复位 4-9
4-3-4 绝对值定位 4-10
4-3-5 相对值定位 4-11

第 5 章 运动控制参数

5-1 概要 5-2

5-2 轴参数 5-4
5-2-1 轴参数一览 5-4
5-2-2 轴基本设定 5-6
5-2-3 单位转换设定 5-10
5-2-4 动作设定 5-17
5-2-5 扩展动作设定 5-20
5-2-6 限位设定 5-21
5-2-7 位置计数设定 5-22
5-2-8 伺服驱动器设定 5-24
5-2-9 原点复位设定 5-25

5-2-10 轴参数的设定示例 5-27

5-3 轴组参数 5-29
5-3-1 轴组参数一览 5-29
5-3-2 轴组基本设定 5-30
5-3-3 轴组动作设定 5-32
5-3-4 轴组的有效 5-33

第 6 章 运动控制程序

6-1 概要 6-2

6-2 运动控制指令 6-4
6-2-1 PLCopen® 运动控制用功能块 6-4
6-2-2 运动控制功能模块的运动控制指令 6-4

6-3 状态变化 6-5
6-3-1 运动控制功能模块的状态 6-5
6-3-2 轴的状态 6-5
6-3-3 轴组的状态 6-7

6-4 运动控制指令的启动和状态 6-8
6-4-1 启动相关的共通规则 6-8
6-4-2 启动时序图 6-10
6-4-3 运动控制指令重新执行时的时序 6-12
6-4-4 运动控制指令多重执行时的时序 6-13

6-5 位置处理 6-14
6-5-1 位置的种类 6-14
6-5-2 各种轴类别可使用的位置类型 6-15

6-6 运动控制系统变量 6-16
6-6-1 运动控制系统变量的概要 6-16
6-6-2 运动控制系统变量的原理 6-18
6-6-3 运动控制系统变量的一览 6-19

6-7 凸轮表和凸轮数据变量 6-30

6-8 运动控制程序的创建方法 6-33

6-9 凸轮表的创建方法 6-35

第 7 章 手动运行

7-1 概要 7-2

7-2 伺服 ON 7-3
7-2-1 伺服 ON 的步骤 7-4
7-2-2 设定轴参数 7-4
7-2-3 程序示例 7-5

7-3 点动进给 7-6
7-3-1 点动进给的步骤 7-6
7-3-2 设定轴参数 7-7
7-3-3 输入变量的设定示例 7-7
7-3-4 程序示例 7-8

第 8 章 定原点

8-1 概要 8-2

8-2 原点复位的步骤 8-5
8-2-1 原点复位参数的设定 8-6
8-2-2 原点复位的监视 8-12

8-3 原点复位的动作 8-13
8-4 绝对编码器原点设定 8-14
8-4-1 功能概要 8-15
8-4-2 设定步骤 8-16
8-5 高速原点复位 8-18

第 9 章 运动控制功能

9-1 单轴位置控制 9-3
9-1-1 动作概要 9-3
9-1-2 绝对值定位 9-4
9-1-3 相对值定位 9-4
9-1-4 中断固定尺寸定位 9-5
9-1-5 周期性同步定位 9-6
9-1-6 停止 9-6
9-1-7 超驰 9-11
9-2 单轴同步控制 9-12
9-2-1 同步控制的概要 9-12
9-2-2 齿轮动作 9-12
9-2-3 位置指定齿轮动作 9-13
9-2-4 凸轮动作 9-14
9-2-5 凸轮表 9-15
9-2-6 梯形模式凸轮动作 9-23
9-2-7 加减法定位 9-24
9-2-8 主轴相位补偿 9-25
9-2-9 从轴位置补偿 9-26
9-2-10 在多运动中实现同步控制的方法 9-26
9-3 单轴速度控制 9-28
9-3-1 速度控制 9-28
9-3-2 周期性同步速度控制 9-29
9-4 单轴扭矩控制 9-30
9-5 单轴控制的共通功能 9-31
9-5-1 位置 9-31
9-5-2 速度 9-33
9-5-3 加速和减速 9-34
9-5-4 跃度 9-36
9-5-5 动作方向指定 9-37
9-5-6 运动控制指令的重启 9-40
9-5-7 运动控制指令的多重启动 （缓冲模式） 9-45
9-6 多轴协调控制 9-51
9-6-1 动作概要 9-51
9-6-2 直线插补 9-54
9-6-3 圆弧插补 9-55
9-6-4 轴组周期性同步位置控制 9-55
9-6-5 多轴协调控制的停止 9-56
9-6-6 多轴协调控制的超驰 9-58
9-7 多轴协调控制的共通功能 9-59
9-7-1 多轴协调控制的速度 9-59
9-7-2 多轴协调控制的加速和减速 9-60
9-7-3 多轴协调控制的跃度 9-61
9-7-4 多轴协调控制时运动控制指令的重启 9-62
9-7-5 多轴协调控制时运动控制指令的多重启动 （缓冲模式） 9-63
9-8 其他功能 9-70
9-8-1 当前位置变更功能 9-70
9-8-2 扭矩限制功能 9-71
9-8-3 锁定功能 9-71
9-8-4 区域选择 9-72
9-8-5 软件限位功能 9-73

9-8-6 位置偏差监视功能 9-75
9-8-7 偏差计数器复位 9-75
9-8-8 轴间偏差监视功能 9-76
9-8-9 位置检查 9-77
9-8-10 轴使用变更功能 9-79
9-8-11 数字凸轮开关有效功能 9-79
9-8-12 任意坐标系的 3D 运动监视显示 9-80

第 10 章 示例程序

10-1 示例程序的概要 10-2
10-1-1 使用设备 10-2
10-1-2 安装和配线 10-2
10-1-3 设定 10-2

10-2 基本程序示例 10-3
10-2-1 EtherCAT 通信的监视和伺服 ON 10-3
10-2-2 通过主控指令进行轴动作的联锁 10-5
10-2-3 异常的监视和异常解除 （单轴动作、同步动作） 10-7
10-2-4 异常的监视和异常解除 （多轴协调动作） 10-9
10-2-5 指令异常的监视方法 10-14
10-2-6 异常解除的监视方法 10-16
10-2-7 单轴动作时的轴停止 10-18
10-2-8 多轴协调动作中的轴组停止 10-22
10-2-9 原点复位和绝对值定位 10-28
10-2-10 通过重启指令来变更目标位置 10-32
10-2-11 中断固定尺寸定位 10-37
10-2-12 通过重启指令，切换凸轮表 10-42
10-2-13 使用凸轮轮廓曲线进行同步位置修正 10-51
10-2-14 凸轮动作时进行主轴的相位补偿 10-60
10-2-15 在速度控制过程中变更当前位置 10-68
10-2-16 凸轮数据变量的变更和凸轮表的保存 10-74
10-2-17 轴参数的临时替换 10-83
10-2-18 凸轮表的终点索引更新 10-86

第 11 章 异常的确认和处理

11-1 异常的概要 11-2
11-1-1 异常的确认方法 11-4
11-1-2 MC 功能模块相关的异常 11-7

11-2 故障排除 11-12
11-2-1 异常一览 11-12
11-2-2 异常内容 11-20
11-2-3 异常现象的推测原因和处理措施 11-52

附录

A-1 与伺服驱动器连接 A-2
A-1-1 与伺服驱动器配线 A-2
A-1-2 伺服驱动器的设定 A-2

A-2 与编码器输入终端连接 A-12
A-2-1 与编码器输入终端配线 A-12
A-2-2 编码器输入终端的设定 A-12

A-3 与 NX 单元连接 A-18

A-4 PDS 状态变化 A-19
A-4-1 PDS 状态控制方式 A-20
A-4-2 主回路电源 OFF 检测 A-21

A-5 术语说明 A-22
A-5-1 NJ/NX 系列通用 A-22
A-5-2 运动控制功能 A-23
A-5-3 EtherCAT 通信 A-24
A-6 版本相关信息 A-25

索引

承诺事项

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

● 定义

本承诺事项中术语的定义如下所示。

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

● 记载内容的注意事项

关于 “ 产品样本等 ” 中的内容，请注意以下几点。

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

● 使用注意事项

使用时，请注意以下几点。

- 使用时请符合额定值、性能以及 “ 使用条件等 ”。
- 请用户自行确认 “ 适用性等 ”，判断是否可使用 “ 本公司产品 ”。“ 本公司 ” 对 “ 适用性等 ” 不作任何保证。
- 用户将 “ 本公司产品 ” 用于整个系统时，请务必事先自行确认配电、设置是否恰当。
- 使用 “ 本公司产品 ” 时，请注意以下各事项。（i）使用 “ 本公司产品 ” 时，应在额定值和性能方面留有余量，采用冗余设计等安全设计，（ii）采用安全设计，即使 “ 本公司产品 ” 发生故障，也可将 “ 用户用途 ” 造成的危险降至最低程度，（iii）对整个系统采取安全措施，以便向使用者告知危险，（iv）定期维护 “ 本公司产品 ” 及 “ 用户用途 ”。
- 本公司设计并制造面向一般工业产品的通用产品。但是，不可用于以下用途。如果用户将 “ 本公司产品 ” 用于以下用途，则 “ 本公司 ” 不对 “ 本公司产品 ” 作任何保证。但如果属于本公司许可的特别产品用途或与本公司签订特殊协议的场合除外。
  - (a) 需高安全性的用途 （例：原子能控制设备、燃烧设备、航空航天设备、铁路设备、起重设备、娱乐设备、医疗设备、安全装置以及其他危及生命、健康的用途）
  - (b) 需高可靠性的用途 （例：煤气、自来水、电力等供应系统、 24 小时连续运行的系统、支付系统等涉及权利、财产的用途等）
  - (c) 用于严格条件或环境下 （例：需设置在室外的设备、会受化学污染的设备、会受电磁波干扰的设备、会受振动、冲击影响的设备等）
  - (d) 在 “ 产品样本等 ” 中未记载的条件或环境下使用

- 上述 （a）～ （d）以及 “ 本产品样本等中记载的产品 ” 不可用于汽车 （含两轮车。下同）。请勿装入汽车进行使用。关于可装入汽车的产品，请咨询本公司销售负责人。

● 保修条件

“ 本产品 ” 的保修条件如下所述。

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

● 责任免除

本承诺事项中的保修即与 “ 本公司产品 ” 相关的保修的所有内容。
对因 “ 本公司产品 ” 造成的损害，“ 本公司 ” 及 “ 本公司产品 ” 的销售店概不负责。

● 出口管理

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

安全注意事项

关于安全注意事项的内容，请参考以下手册。

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

安全要点

关于安全要点的内容，请参考以下手册。

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

使用注意事项

关于使用注意事项的内容，请参考以下手册。

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

法规和标准

日本国外的使用

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

符合 EU 指令

适用指令

- EMC 指令
- 低电压指令

概念

● EMC 指令

由于 OMRON 公司的产品为安装在各种机械、制造装置中使用的电气设备，为了使安装 PLC 的机械、装置更容易符合 EMC 标准，必须先使产品本身达到相关 EMC 标准 (*1)。
但是，由于客户使用的机械、装置各不相同， EMC 性能因符合 EU 指令的产品安装的设备和控制柜的构成、接线及配置等条件而异，无法在客户使用状态下确认其适合性。因此，机械、装置整体最终的 EMC 适用性确认请客户自行实施。

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

● 低电压指令

对于以电源电压 AC50V ～ 1000V 以及 DC75V ～ 1500V 工作的设备，要求必须确保必要的安全性。适用标准为 EN61010-2-201。

● 符合 EU 指令

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

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

KC 标准的适用

在韩国使用本产品时，请遵守以下注意事项。

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

ClassA 设备 （商用广播通信设备）

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

船舶标准的适用

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

各船舶标准的使用条件 （NK、 LR）

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

● 噪声滤波器

厂家	型号
COSEL	TAH-06-683

软件许可和著作权

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

版本

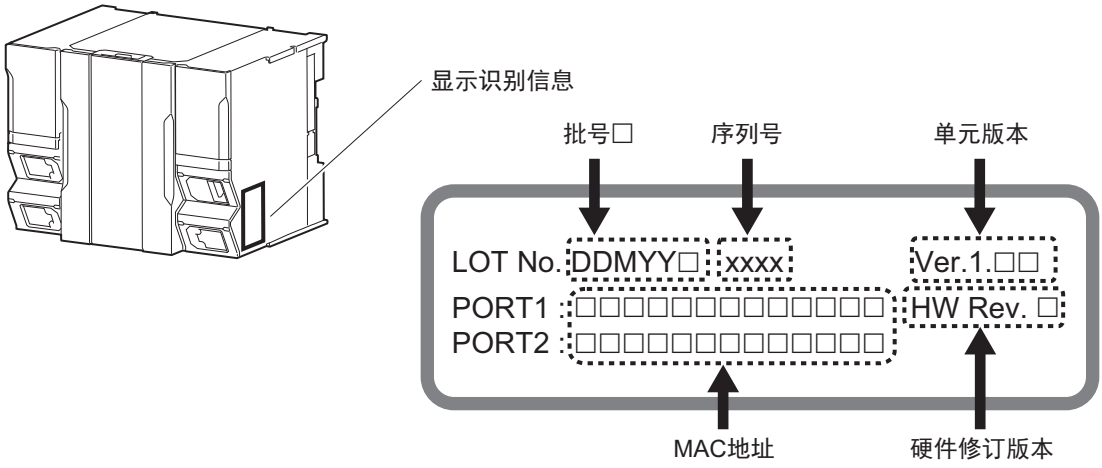
NJ/NX 系列的各单元及各 EtherCAT 从站的硬件或软件按照硬件修订版本或单元版本等编号管理。每次硬件或软件发生规格变更后，硬件修订版本或单元版本都会更新。因此，即使是同一型号的单元或 EtherCAT 从站，若硬件修订版本或单元版本不同，配备的功能或性能可能不同。

版本的确认方法

版本可通过标示的识别信息或 Sysmac Studio 确认。

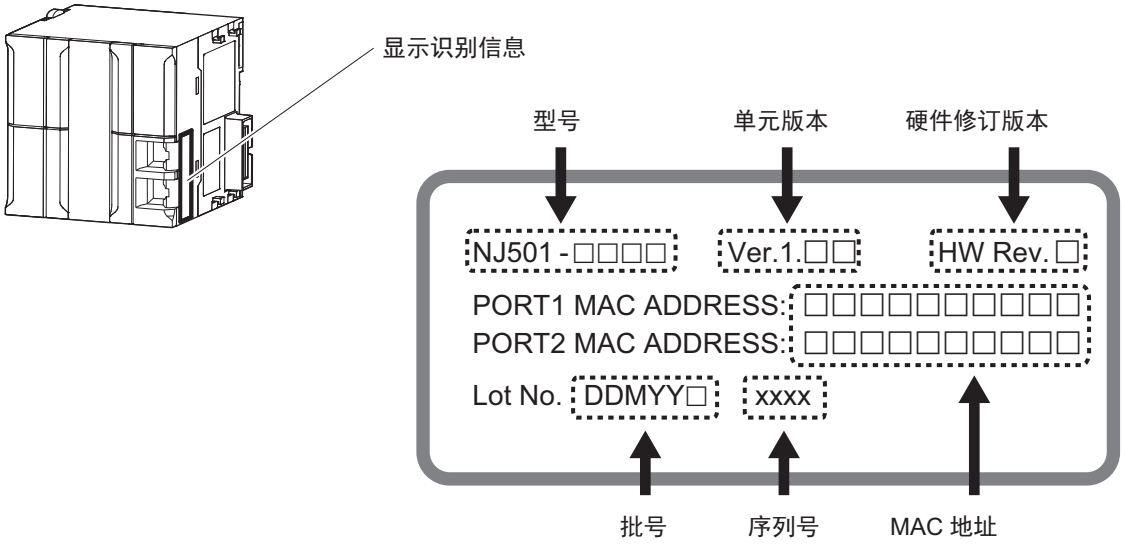
通过标示的识别信息确认

可通过产品侧面标示的识别信息确认版本。
NX 系列 CPU 单元 NX701-□□□□ 的识别信息标示如下图所示。



（注）硬件修订版本为 “ 无 ” 的单元中，不标示硬件修订版本。

NJ 系列 CPU 单元 NJ501-□□□□ 的识别信息标示如下图所示。



（注）硬件修订版本为 “ 无 ” 的单元中，不标示硬件修订版本。

通过 Sysmac Studio 确认的方法

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

● NX 系列 CPU 单元的版本确认方法

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

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

● NJ 系列 CPU 单元的版本确认方法

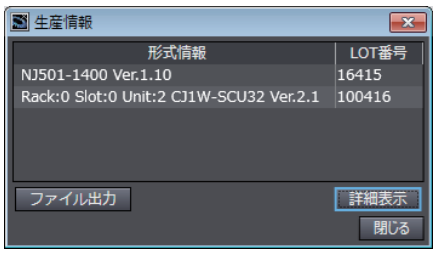
单元版本可通过在线状态的［生产信息］确认。但是，只能确认 CPU 单元、 CJ 系列高功能 I/O 单元及 CJ 系列 CPU 总线单元的版本。 CJ 单元的基本 I/O 单元的版本无法通过 Sysmac Studio 确认。版本的确认方法如下所示。

1 **在多视图浏览器中双击［配置和设定］–［CPU• 扩展装置］。或者右击［配置和设定］–［CPU• 扩展装置］，然后选择［编辑］。**
显示单元编辑器。

2 **右击单元编辑器的空白栏，选择［显示生产信息］。**
显示［生产信息］对话框。

● 切换生产信息的显示内容

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



简易显示



详细显示

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

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

● EtherCAT 从站的版本确认方法

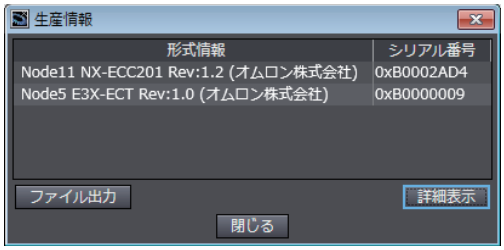
EtherCAT 从站的版本可通过在线状态的［生产信息］确认。确认方法如下所示。

1 **在多视图浏览器中双击［配置和设定］–［EtherCAT］。或者右击［配置和设定］–［EtherCAT］，然后选择［编辑］。**
显示 EtherCAT 配置编辑画面。

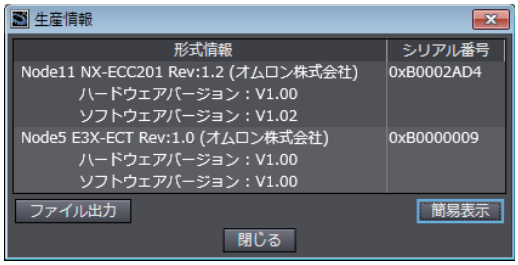
2 **右击 EtherCAT 配置编辑画面中的主机，选择［显示生产信息］。**
显示生产信息对话框。
单元版本以带 “Rev” 的形式显示。

● 切换生产信息的显示内容

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



简易显示



详细显示

CPU 单元的单元版本和 Sysmac Studio 的版本

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

CPU 单元的单元版本种类和 Sysmac Studio 版本的关系请参考 🕮《NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-CN5-359)》。

单元版本对应的支持功能一览请参考 🕮「*A-6 版本相关信息* (P.A-25)」。

相关手册

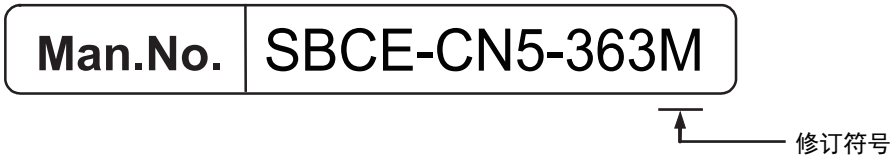
本手册相关的手册如下表所示。请同时参考。

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

手册名称	Man.No.	型号	用途	内容
NJ/NX 系列 CPU 单元 内置 EtherNet/IP™ 端口 用户手册	SBCD-CN5-359	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	使用 NJ/NX 系列 CPU 单元的内置 EtherNet/IP 端口。	对内置 EtherNet/IP 端口进行说明。 对基本设定、标签数据链接、其它功能进行说明。 请同时使用用户手册 硬件篇（SBCA-CN5-418 或 SBCA-CN5-358）/ 软件篇 （SBCA-CN5-359）。
NJ 系列 数据库连接 CPU 单元 用户手册	SBCA-CN5-411	NJ501-1□20	在 NJ 系列中使用数据库连接服务功能时。	对数据库连接服务功能进行说明。
NJ 系列 搭载 SECS/GEM 的 CPU 单元 用户手册	SBCA-CN5-412	NJ501-1340	在 NJ 系列中使用 GEM 服务功能时。	对 GEM 服务的功能进行说明。
NJ 系列 NJ Robotics CPU 单元 用户手册	SBCA-CN5-421	NJ501-4□□□	用 NJ 系列进行机器人控制时。	对机器人控制功能进行说明。
NJ/NX 系列 故障排除 手册	SBCA-CN5-361	NX701-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	希望了解 NJ/NX 系列检出的异常详情时。	对 NJ/NX 系列系统检出的异常的管理思路和各异常项目进行说明。 请同时使用用户手册 硬件篇（SBCA-CN5-418 或 SBCA-CN5-358）/ 软件篇 （SBCA-CN5-359）。
Sysmac Studio Version 1 操作手册	SBCA-CN5-362	SYSMAC-SE2□□□	了解 Sysmac Studio 的操作方法、功能。	对 Sysmac Studio 的操作方法进行说明。
NX 系列 EtherCAT® 耦合器单元 用户手册	SBCD-CN5-361	NX-ECC□□□	希望了解 NX 系列 EtherCAT 耦合器单元及 EtherCAT 从站终端的使用方法时。	对 NX 系列 EtherCAT 耦合器单元和 NX 单元构成的 EtherCAT 从站终端的系统概要、配置方法，以及通过 EtherCAT 对 NX 单元进行设定、控制、监视所需的 EtherCAT 耦合器单元的硬件、设定方法和功能进行说明。
NX 系列 NX 单元 用户手册	SBCA-CN5-407	NX-ID□□□□ NX-IA□□□□ NX-OC□□□□ NX-OD□□□□	希望了解 NX 单元的使用方法时。	对 NX 单元的硬件、设定方法和功能进行说明。 包括以下单元的手册。 数字 I/O 单元、模拟量 I/O 单元、系统单元、位置接口单元
	SBCA-CN5-408	NX-AD□□□□ NX-DA□□□□ NX-TS□□□□		
	SBCA-CN5-409	NX-PD1□□□ NX-PF0□□□ NX-PC0□□□ NX-TBX□□		
	SBCE-CN5-374	NX-EC0□□□ NX-ECS□□□ NX-PG0□□□		
NX 系列 数据基准 手册	SBCA-CN5-410	NX-□□□□□□	希望列表查看 NX 系列各单元的系统构建所需的数据时。	汇总记载 NX 系列各单元的消耗电力、质量等系统构建所需的数据。
EtherCAT 远程 I/O 终端 GX 系列 EtherCAT 从站 用户手册	SBCD-CN5-350	GX-ID□□□□ GX-OD□□□□ GX-OC□□□□ GX-MD□□□□ GX-AD□□□□ GX-DA□□□□ GX-EC□□□□ XWT-ID□□ XWT-OD□□	希望了解 EtherCAT 远程 I/O 终端的使用方法时。	对 EtherCAT 远程 I/O 终端的硬件、设定方法和功能进行说明。
AC 伺服电机 / 驱动器 G5 系列 EtherCAT® 通信内置型 用户手册	SBCE-CN5-360	R88M-K□ R88D-KN□-ECT-R	希望了解 EtherCAT 通信内置型伺服电机 / 驱动器的使用方法时。	对 EtherCAT 通信内置型伺服电机 / 驱动器的硬件、设定方法和功能进行说明。 G5 系列包括直线电机型和位置控制限定机型。
	SBCE-CN5-365	R88M-K□ R88D-KN□-ECT		
	SBCE-CN5-366	R88L-EC-□ R88D-KN□-ECT-L		

手册修订记录

手册修订符号在封面和封底中记载的 Man.No. 末尾标注。



修订符号	修订年月	修订理由
A	2011 年 7 月	第一版
B	2012 年 1 月	追加 NJ301-□□□□ 因 CPU 单元版本升级到 1.01 发生的修订
C	2012 年 5 月	因 CPU 单元版本升级到 1.02 发生的修订 误记修正
D	2012 年 8 月	因 CPU 单元版本升级到 1.03 发生的修订
E	2013 年 2 月	因 CPU 单元版本升级到 1.04 发生的修订
F	2013 年 4 月	因 CPU 单元版本升级到 1.05 发生的修订 误记修正
G	2013 年 6 月	因 CPU 单元版本升级到 1.06 发生的修订 误记修正
H	2013 年 12 月	因 CPU 单元版本升级到 1.08 发生的修订 误记修正
J	2014 年 7 月	因 CPU 单元版本升级到 1.09 发生的修订 误记修正
K	2015 年 1 月	因 CPU 单元版本升级到 1.10 发生的修订 误记修正
L	2015 年 4 月	因追加 NX 系列 CPU 单元 NX701-□□□□ 及 NJ 系列 CPU 单元 NJ101-10□□ 发生的修订 误记修正
M	2016 年 4 月	因 CPU 单元版本升级到 1.11 发生的修订 误记修正

1

运动控制功能模块的概要

下面介绍运动控制功能模块的特点和系统构成、使用流程。

1-1 特点 1-2

1-2 系统构成 1-3

1-3 基本使用流程 1-4

1-4 规格 1-6

1-4-1 一般规格 1-6

1-4-2 性能规格 1-6

1-4-3 功能规格 1-8

1-1 特点

运动控制功能模块 （以下简称 “MC 功能模块 ”）是指 CPU 单元中内置软件的功能模块。
MC 功能模块可通过 CPU 单元的内置 EtherCAT 端口，对最多 256 个轴进行运动控制。
与 EtherCAT 端口上连接的伺服驱动器等进行周期性通信，可实现高速、高精度的机器控制。

符合 PLCopen® 标准的运动控制指令

MC 功能模块的运动控制指令以 PLCopen® 的标准化运动控制功能块为基础。
包括单轴 PTP 定位、插补控制、电子齿轮等的同步控制以及速度控制、扭矩控制等，可编写各种符合标准的指令。
此外，每次启动运动控制指令时，均可设定速度、加速度、减速度及跃度，因此可根据具体用途进行灵活的动作控制。



参考

什么是 PLCopen®

PLCopen® 为总部位于欧洲的 IEC 61131-3 普及团体，是一个全球性会员组织。
PLCopen® 通过对运动控制用功能块进行标准化，定义了 IEC 61131-3 （JIS B 3503）标准语言的程序界面。

什么是跃度

跃度为加速度、减速度的变化比率，也称为 “ 跃动 ”、 “ 加加速度 ”。指定跃度后，加减速时的速度波形变为 S 字型。

利用 EtherCAT 通信传送数据

与欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型组合，所有控制信息均可通过高速数据通信进行交换。
利用数据通信传达各种控制指令，不受编码器反馈脉冲的响应频率等接口规格的制约，可最大限度地发挥伺服电机的性能。
此外，伺服驱动器的各种控制参数或监视信息可在上位控制器上处理，因此可实现系统信息管理的统一化。



参考

什么是 EtherCAT

EtherCAT 为 Ethernet(IEEE802.3）标准的开放架构高速商用网络系统。各节点高速传输 Ethernet 帧，可实现较短的通信周期时间。此外，利用时钟信息共享机制，可实现通信抖动低、精度高的同步控制。

1-2　系统构成

MC 功能模块根据装置或控制柜发出的传感器信号的状态和用户程序的运动控制指令发出的指令，进行伺服驱动器、编码器输入终端、 NX 系列位置接口单元的运动控制。

运动控制的构成

MC 功能模块中，使用 EtherCAT 网络配置、 EtherCAT 耦合器单元的从站终端构成及 Sysmac Studio。

● EtherCAT 网络配置

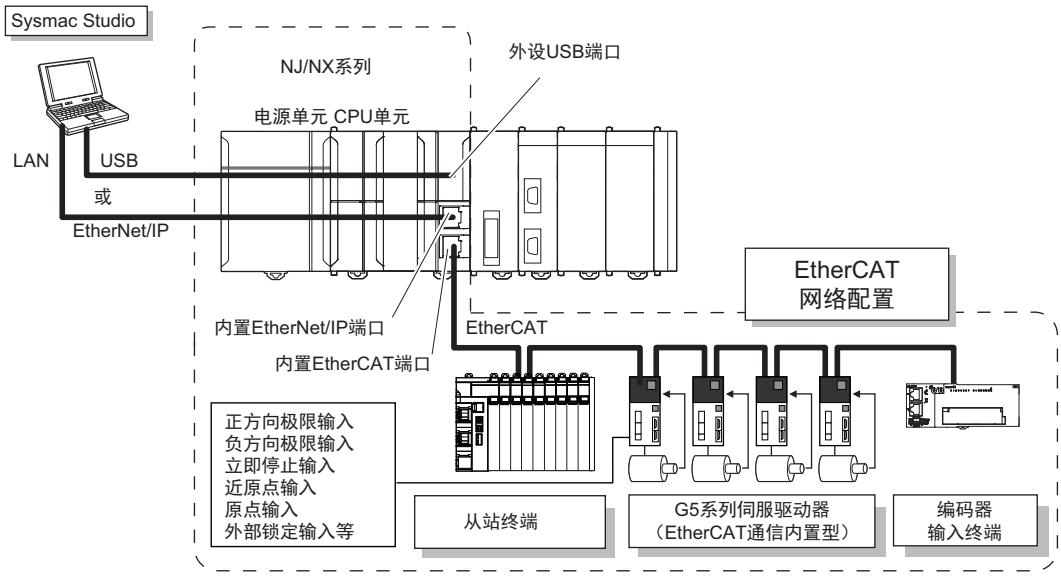
MC 功能模块使用 CPU 单元中内置的 EtherCAT 通信主机功能端口，对伺服驱动器、编码器输入终端进行控制。
使用这样的构成，可实现周期恒定、波动小、精度高的运动控制。

● EtherCAT 耦合器单元的从站终端构成

MC 功能模块使用 EtherCAT 耦合器单元中安装的位置接口单元，导入电机控制用的脉冲输出机编码器输入。
使用这样的构成，也可实现周期恒定、波动小、精度高的运动控制。

● Sysmac Studio

可将市售的 USB 电缆连接到 CPU 单元的外围 USB 端口上。或者将 Ethernet 电缆连接到 CPU 单元内置的 EtherNet/IP 端口上。

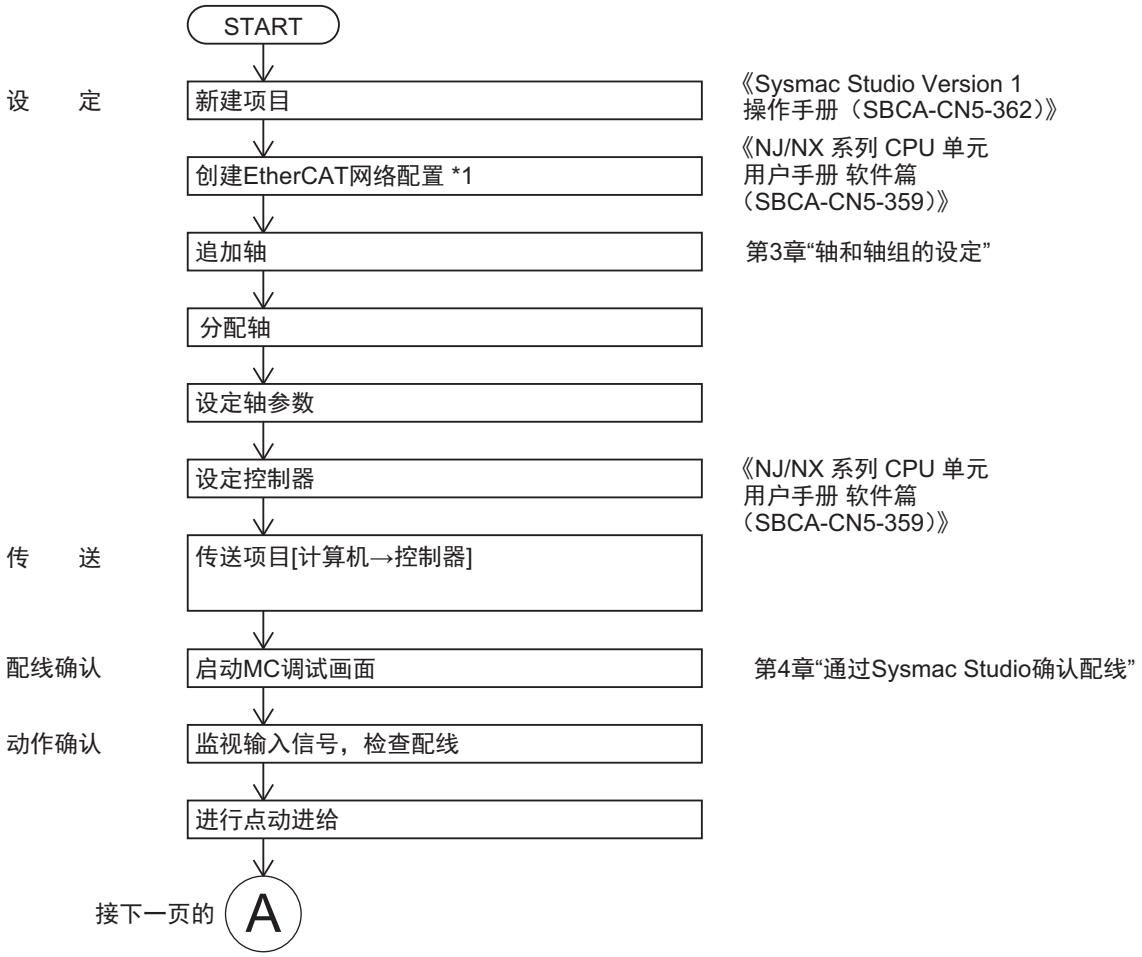


版本相关信息

NX 系列位置接口单元可在 CPU 单元版本 1.05 以上和 Sysmac Studio Ver.1.06 以上的组合中使用。
使用 NX 系列位置接口单元时，运动控制功能的部分功能可能不同。详情请参考 🕮《NX 系列位置接口单元 用户手册 (SBCE-CN5-374)》。

1-3 基本使用流程

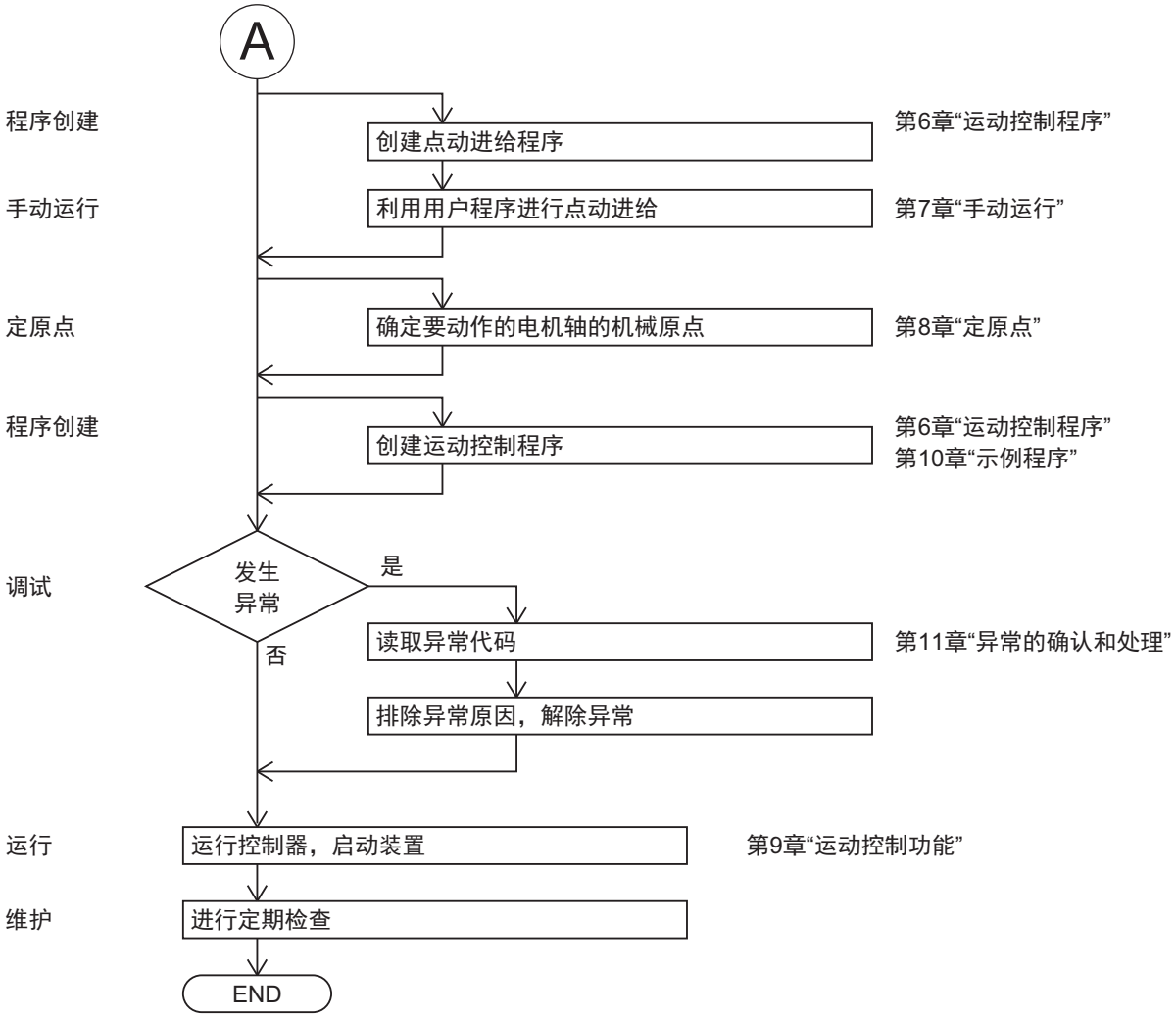
下面介绍用 MC 功能模块进行运动控制时的基本使用步骤。



*1. 连接了实体机时，可在线设定实体机构成。
没有实体机时，可选择离线使用的 EtherCAT 从站并设定。

参考

使用 NX 系列位置接口单元时的使用步骤请参考 《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。


A
程序创建
创建点动进给程序
第6章“运动控制程序”
手动运行
利用用户程序进行点动进给
第7章“手动运行”
定原点
确定要动作的电机轴的机械原点
第8章“定原点”
程序创建
创建运动控制程序
第6章“运动控制程序”
第10章“示例程序”
调试
发生
异常
是
否
读取异常代码
第11章“异常的确认和处理”
排除异常原因，解除异常
运行
运行控制器，启动装置
第9章“运动控制功能”
维护
进行定期检查
END

1-4 规格

下面介绍 MC 功能模块的规格。

使用注意事项

NJ101-90□□ 中无运动控制功能模块。

1-4-1 一般规格

一般规格遵照 CPU 单元的一般规格。

详情请参考 《NX 系列 CPU 单元 用户手册 硬件篇 （SBCA-CN5-418）》或 《NJ 系列 CPU 单元 用户手册 硬件篇 （SBCA-CN5-358）》。

1-4-2 性能规格

介绍各 CPU 单元的性能规格。

规格项目			规格 （NX701-）	
			17□□	16□□
控制轴数	最大控制轴数 *1		256 轴	128 轴
	最大使用实轴数 *2		256 轴	128 轴
	最大单轴控制数		256 轴	128 轴
	直线插补控制最大轴数		每 1 轴组 4 轴	
	圆弧插补控制轴数		每 1 轴组 2 轴	
轴组最大数			64 组	
超驰			0.00, 0.01 ～ 500.00%	
运动控制周期			与 EtherCAT 通信的过程数据通信周期相同	
多运动			可	
凸轮	凸轮数据点数	每 1 凸轮表的最大点数	65,535 点	
		所有凸轮表的最大点数	1,048,560 点	
	凸轮表最大表数		640 表	

*1. 所有轴种类的合计。
*2. 轴种类为 ［伺服轴］或 ［编码器轴］，且轴使用设定为 ［使用轴］的轴合计。

规格项目			规格 （NJ501-）		
			□5□□	□4□□	□3□□
控制轴数	最大控制轴数 *1		64 轴	32 轴	16 轴
	最大使用实轴数 *2 *3		64 轴	32 轴	16 轴
	最大单轴控制数		64 轴	32 轴	16 轴
	直线插补控制最大轴数		每 1 轴组 4 轴		
	圆弧插补控制轴数		每 1 轴组 2 轴		
轴组最大数			32 组		
超驰			0.00, 0.01 ～ 500.00%		
运动控制周期			与 EtherCAT 通信的过程数据通信周期相同		
多运动			不可		
凸轮	凸轮数据点数	每 1 凸轮表的最大点数	65,535 点		
		所有凸轮表的最大点数	1,048,560 点		
	凸轮表最大表数		640 表		

*1. 所有轴种类的合计。
*2. 轴种类为［伺服轴］或［编码器轴］，且轴使用设定为［使用轴］的轴合计。
*3. CPU 单元版本升级后追加的功能。升级相关的信息请参考 🕮 请参考 *「A-6 版本相关信息* (P.A-25)」。

规格项目			规格 （NJ301-）		规格 （NJ101-）
			12□□	11□□	10□□
控制轴数	最大控制轴数 *1		15 轴 *2 *3	15 轴 *3 *4	6 轴
	最大使用实轴数 *3 *5		8 轴	4 轴	2 轴
	最大单轴控制数		15 轴 *2 *3	15 轴 *3 *4	6 轴
	直线插补控制最大轴数		每 1 轴组 4 轴		
	圆弧插补控制轴数		每 1 轴组 2 轴		
轴组最大数			32 组		
超驰			0.00, 0.01 ～ 500.00%		
运动控制周期			与 EtherCAT 通信的过程数据通信周期相同		
多运动			不可		
凸轮	凸轮数据点数	每 1 凸轮表的最大点数	65,535 点		
		所有凸轮表的最大点数	262,140 点		
	凸轮表最大表数		160 表		

*1. 所有轴种类的合计。
*2. CPU 单元版本 1.06 以上和 Sysmac Studio Ver.1.07 以上组合中的轴数。其他情况下最大控制轴数为 8 轴。
*3. CPU 单元版本升级后追加的功能。升级相关的信息请参考 🕮 *「A-6 版本相关信息* (P.A-25)」。
*4. CPU 单元版本 1.06 以上和 Sysmac Studio Ver.1.07 以上组合中的轴数。其他情况下最大控制轴数为 4 轴。
*5. 轴种类为［伺服轴］或［编码器轴］，且轴使用设定为［使用轴］的轴合计。

1-4-3 功能规格

下面介绍与欧姆龙生产的控制对象设备连接时的功能。

功能项目			内容
控制对象伺服驱动器			欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型 *1 *2
控制对象编码器输入终端			欧姆龙产 EtherCAT 远程 I/O 终端 GX 系列 GX-EC0211/EC0241 *3
控制对象位置接口单元 *2 *4			欧姆龙产增量编码器输入单元 NX-EC0□□□ 欧姆龙产 SSI 输入单元 NX-ECS□□□ 欧姆龙产脉冲输出单元 NX-PG0□□□
控制方式			通过 EtherCAT 通信发行控制指令
控制模式			位置控制 (cyclic synchronous position) 速度控制 (cyclic synchronous velocity) 扭矩控制 (cyclic synchronous torque)
单位转换		显示单位	脉冲、 mm、 μm、 nm、 degree、 inch
		电子齿轮比	电机每转 1 圈的脉冲数 / 电机每转 1 圈的移动量或 （电机每转 1 圈的脉冲数 × 电机侧齿轮比） /（工件侧每转 1 圈的移动量 × 工件侧齿轮比）
可管理位置			指令位置、反馈位置
轴种类			伺服轴、虚拟伺服轴、编码器轴、虚拟编码器轴
位置指令			双精度实数型 （LREAL 型）的负数、正数、 0 （指令单位 *5）
速度指令			双精度实数型 （LREAL 型）的负数、正数、 0 （指令单位 /s）
加速度指令、减速度指令			双精度实数型 （LREAL 型）的正数、 0 （指令单位 /s^2）
跃度指令			双精度实数型 （LREAL 型）的正数、 0 （指令单位 /s^3）
单轴	单轴位置控制	绝对值定位	指定绝对坐标的目标位置，进行定位的功能
		相对值定位	指定指令当前位置起的移动距离，进行定位的功能
		中断固定尺寸定位	指定基于外部输入的中断输入发生位置起的移动距离，进行定位的功能
		周期性同步绝对位置控制	在定位控制模式下，按控制周期输出指令位置的功能
	单轴速度控制	速度控制	在定位控制模式下进行速度控制的功能
		周期性同步速度控制	在速度控制模式下，按控制周期输出速度指令的功能
	单轴转矩控制	转矩控制	执行电机转矩控制的功能
	单轴同步控制	凸轮动作开始	使用指定的凸轮表开始凸轮动作的功能
		凸轮动作解除	完成输入参数指定轴的凸轮动作的功能
		齿轮动作开始	设定主轴与从轴间的齿轮比，进行齿轮动作的功能
		位置指定齿轮动作	设定主轴与从轴间的齿轮比和要同步的位置，进行齿轮动作的功能
		齿轮动作解除	中止执行中的齿轮动作、位置指定齿轮动作的功能
		梯形模式凸轮	与指定的主轴同步进行定位的功能
		主轴相对值相位补偿	执行同步控制中的主轴相位补偿的功能
		加减法定位	将 2 个轴的指令位置相加或相减得到的数值，作为指令位置输出的功能
	单轴手动操作	可运转	将伺服驱动器的状态切换为伺服 ON 状态，实现轴动作的功能
		点动进给	依据指定的目标速度执行点动进给的功能

功能项目			内容
单轴	单轴控制辅助	轴错误复位	解除轴异常的功能
		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的功能
		参数指定原点复位 *2	指定参数驱动电机，使用极限信号、近原点信号、原点信号确定机械原点的功能
		高速原点复位	将绝对坐标 “0” 作为目标位置，进行定位并返回原点的功能
		强制停止	使轴减速停止的功能
		立即停止	使轴立即停止的功能
		超驰值设定	变更轴目标速度的功能
		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数值的功能
		外部锁定有效	发生触发，记录轴位置的功能
		外部锁定无效	使执行中的锁定无效的功能
		区域监视	判断轴的指令位置或反馈当前位置是否存在于指定范围 （区域）内的功能
		数字凸轮开关有效 *2	根据轴的位置将数字输出设为 ON 或 OFF 的功能
		轴间偏差监视	监视指定的 2 轴指令位置或反馈位置的差异量是否超出了容许值的功能
		偏差计数器复位	将指令当前位置和反馈当前位置间的偏差归零的功能
		转矩限制	通过切换伺服驱动器转矩限制功能的有效 / 无效和设定转矩限制值，限制输出转矩的功能
		轴使用变更 *2	临时切换轴参数 ［轴使用］的功能
		启动速度 *2	设定轴动作开始时初速度的功能
轴组	多轴协调控制	绝对值直线插补	指定绝对位置进行直线插补的功能
		相对值直线插补	指定相对位置进行直线插补的功能
		2 轴圆弧插补	进行 2 轴圆弧插补的功能
		轴组周期性同步绝对位置控制 *2	在定位控制模式下，按控制周期输出指令位置的功能
	多轴协调控制辅助	轴组错误复位	解除轴组及轴异常的功能
		轴组有效	将轴组设为有效的功能
		轴组无效	将轴组设为无效的功能
		轴组构成轴写入 *2	暂时改写轴组参数的 ［构成轴］的功能
		轴组强制停止	使插补动作中的所有轴减速停止的功能
		轴组即停	使插补动作中的所有轴即停的功能
		轴组超驰值设定	变更插补动作中合成目标速度的功能
		轴组位置获取 *2	获取轴组指令当前位置和反馈当前位置的功能
通用	凸轮 *6	凸轮表属性更新	对输入参数指定的凸轮表的终点索引进行更新的功能
		凸轮表保存	将输入参数指定的凸轮表保存至 CPU 单元内置非易失性存储器中的功能
		凸轮表生成 *2	根据输入参数指定的凸轮属性和凸轮节点生成凸轮表的功能
	参数	MC 设定写入	暂时改写部分轴参数及轴组参数的功能
		轴参数的变更 *2	通过用户程序查看、变更轴参数的功能

功能项目			内容
辅助功能	计数模式		可选择线性模式 （有限长）或旋转模式 （无限长）
	单位转换		可根据机械设定各轴的显示单位
	加减速控制	自动加减速控制	以跃度设定轴及轴组动作时加减速曲线的功能 （梯形曲线、 S 字型曲线）
		变更加减速度	即使在加减速动作中仍可变更加减速度的功能
	位置检查		旨在检查定位完成，设定位置宽度和位置检查时间的功能
	停止方法选择		设定即停输入信号及极限输入信号有效时的停止方法的功能
	运动控制指令的重启		变更执行中运动控制指令的输入变量并重启，在动作中变更目标值的功能
	运动控制指令的多重启动（缓冲模式）		对动作中启动其它运动控制指令时的执行开始时间和动作间速度的连接方法进行指定的功能
	监视功能	软件限位	监视轴动作范围的功能
		位置偏差	监视各轴的指令当前值和反馈当前值之间的位置偏差的功能
		速度 / 加减速度 / 扭矩 / 插补速度 / 插补加减速度	按各轴及轴组分别设定警告值并监视的功能
	适用于绝对编码器		使用欧姆龙产伺服电机 G5 系列的带绝对编码器伺服电机，接通电源时无需原点复位的功能 *7
	输入信号的逻辑反转 *2		使立即停止输入信号、正方向极限输入信号、负方向极限输入信号、近原点输入信号的逻辑反转的功能
外部 I/F 信号			可使用伺服驱动器侧的下列输入信号 • 原点信号、近原点信号、正方向极限信号、负方向极限信号、即停信号、中断输入信号

*1. 推荐的伺服驱动器 G5 系列版本：电机为气缸型时 2.1 以上；直线电机型时 1.1 以上。
*2. CPU 单元版本升级后追加的功能。升级相关的信息请参考 🕮 「*A-6 版本相关信息* (P.A-25)」。
*3. 推荐单元版本为 1.1 以上。
*4. 使用 NX 系列位置接口单元时，运动控制功能的部分功能可能不同。详情请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。
*5. 转换为脉冲单位的值可在带符号整数型 40 位的范围内设定。
*6. 凸轮表可通过 Sysmac Studio 的凸轮编辑器或用户程序中的凸轮表生成指令创建。指定主轴的相位和从轴的变位。可变更各区间的相位间隔宽度。可在用户程序中替换凸轮数据。
*7. 在欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置直线电机型中，可使用绝对型外部光栅尺。

2

运动控制系统的构成和原理

本章介绍 CPU 单元的内部构成概要和运动控制功能模块的构成和原理。

2-1 CPU 单元内部构成 2-2

2-2 运动控制系统的构成 2-3

2-3 运动控制系统的原理 2-4

2-3-1 CPU 单元的任务 2-4

2-3-2 运动控制的任务动作示例 2-10

2-4 与 EtherCAT 通信的关系 2-17

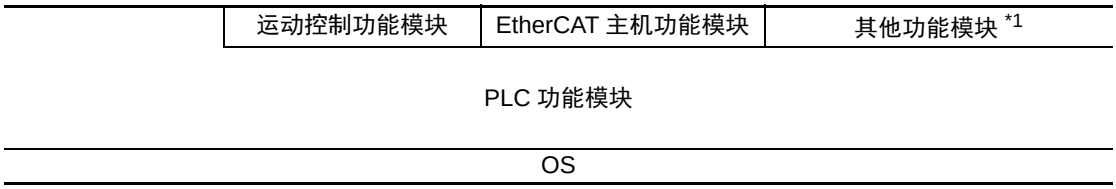
2-4-1 CAN application protocol over EtherCAT (CoE) 2-17

2-4-2 EtherCAT 主机功能和运动控制功能模块的关系 2-18

2-4-3 过程数据通信周期和运动控制周期的关系 2-20

2-1 CPU 单元内部构成

下面介绍 NJ/NX 系列 CPU 单元的内部原理。
CPU 单元的软件构成如下。
运动控制功能模块为执行运动控制的软件。



*1. 其他功能模块请参考 《NJ/NX 系列 CPU 单元 用户手册 软件篇 （SBCA-CN5-359）》。

OS 上有 PLC 功能模块，且各功能模块动作。
各功能模块的内容如下所示。

功能模块名称	简称	内容
PLC 功能模块	PLC	进行整体的日程表管理、用户程序的执行、向运动控制功能模块发出指令，以及与 USB•SD 存储卡的接口。
运动控制功能模块	MC	按照用户程序的运动控制指令执行运动控制，向 EtherCAT 主机功能模块发送数据。
EtherCAT 主机功能模块	ECAT	作为 EtherCAT 主机与 EtherCAT 从站进行通信。

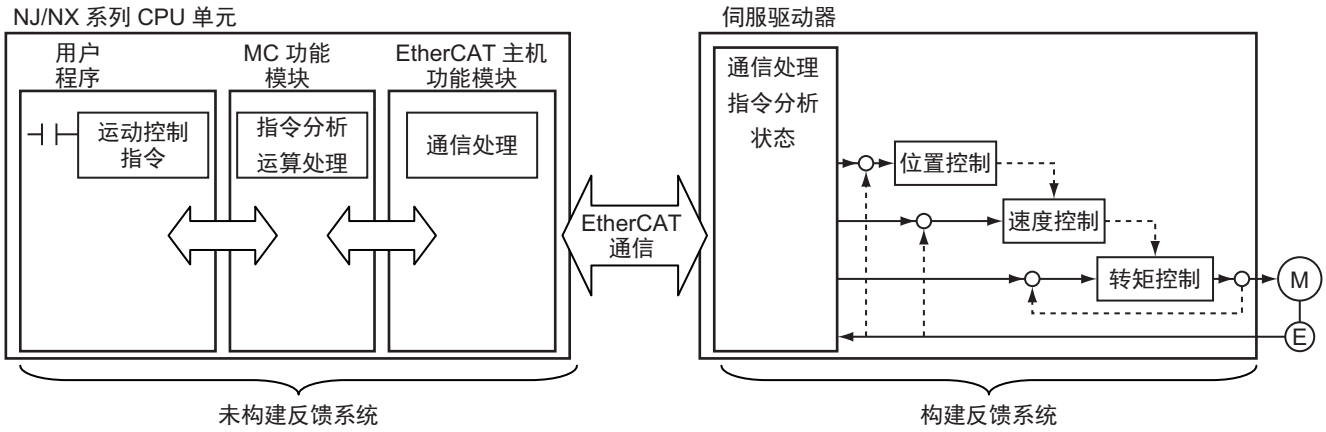
（注）其他功能模块请参考 《NJ/NX 系列 CPU 单元 用户手册 软件篇 （SBCA-CN5-359）》。

本手册中将介绍运动控制功能模块 （以下可能简称 “MC 功能模块 ”）的规格和使用。
使用 MC 功能模块时，请同时参考 NJ/NX 系列相关的用户手册。

2-2 运动控制系统的构成

由伺服构建的控制系统一般通过半闭环方式控制电机动作。半闭环方式中，通过电机上安装的编码器检测与指令值对应的电机旋转量。然后作为机械移动量反馈。计算指令值和电机实际旋转量的偏差，使其变为 “0” 的控制方法。

在使用 MC 功能模块的机器构成中，对于 CPU 单元的用户程序发出的指令，不使用反馈信息。在伺服驱动器内构件反馈系统。



- 若通过用户程序启动运动控制指令，由 MC 功能模块分析指令。
- MC 功能模块将根据指令的分析结果，按恒定周期执行运动计算，生成发送至伺服驱动器的指令值。生成的指令值包括目标位置、目标速度、目标扭矩。
- 生成的指令值将按照 EtherCAT 通信的过程数据通信周期通过 PDO 通信发送。
- 伺服驱动器根据以 EtherCAT 通信的过程数据通信周期发送的指令值，执行位置回路控制、速度回路控制、扭矩回路控制。
- 编码器的当前值或伺服驱动器的状态按照 EtherCAT 通信的过程数据通信周期发送至 CPU 单元。

参考

- 运动计算与 EtherCAT 通信的过程数据通信为相同周期。
- MC 功能模块的对象为内置有位置控制回路、速度控制回路、扭矩控制回路的伺服驱动器。
- 使用 NX 系列位置接口单元时的构成请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

2-3 运动控制系统的原理

下面介绍 CPU 单元的任务和运动控制之间的关系。

2-3-1 CPU 单元的任务

任务是指用户程序的处理执行条件或顺序属性。
NJ/NX 系列 CPU 单元中包含以下任务。

任务类型	任务名称
按恒定周期执行的任务	主固定周期任务
	固定周期任务 （执行优先度 5 *1、 16、 17、 18）
在事先指定的执行条件成立时，仅执行 1 次的任务	事件任务 （执行优先度 8、 48）*2

*1. 固定周期任务 （执行优先度 5）仅在 NX 系列 CPU 单元时可以使用。不可在 NJ 系列 CPU 单元中使用。
*2. 事件任务可在 CPU 单元版本 1.03 以上和 Sysmac Studio Ver.1.04 以上的组合中使用。

用户程序、任务详情及设定方法请参考 🕮《NJ/NX 系列 CPU 单元 用户手册 软件篇（SBCA-CN5-359）》。

任务类型和优先度

NX 系列 CPU 单元配备多核处理器，可并列执行主固定周期任务和固定周期任务 （执行优先度 5）。
NJ 系列 CPU 单元不可同时执行多个任务。
任务有执行优先度，会优先执行优先度较高的任务。正在执行某一任务时，若有执行优先度更高的任务满足执行条件，将优先执行优先度更高的任务。
下表为 NJ/NX 系列 CPU 单元中可使用运动控制指令的任务之种类和优先度。事件任务不可使用运动控制指令。

任务类型	任务数	优先度	动作
主固定周期任务	1 个	4	按设定的任务周期执行 I/O 刷新、用户程序执行、运动控制。 任务的执行优先度最高，可高速、高精度执行，因此适用于需要同步控制或高速响应的控制。 以单任务执行所有控制时，使用主固定周期任务。
固定周期任务	0 ～ 1 个	5 *1	按设定的任务周期执行 I/O 刷新、用户程序执行、运动控制。 任务的执行优先度仅次于主固定周期任务，可高速、高精度执行。 在构成装置的功能中，用主固定周期任务控制需要以最快速度控制的功能，用固定周期任务 （执行优先度 5）控制其他功能时使用。 使用多任务进行运动控制时，使用主固定周期任务和固定周期任务 （执行优先度 5）。*2

任务类型	任务数	优先度	动作
固定周期任务	0 ～ 1 个	16 *3	按设定的任务周期执行 I/O 刷新和用户程序。 以长于主固定周期任务或固定周期任务 （执行优先度 5）的周期执行用户程序时，使用固定周期任务 （执行优先度 16）。 固定周期任务 （执行优先度 16）中可对在主固定周期任务中进行 I/O 刷新的部分从站或单元记述用户程序。 例如，同步控制或需要高速响应的控制用主固定周期任务或固定周期任务 （执行优先度 5）控制，装置整体的控制则分配到固定周期任务 （执行优先度 16）进行控制。

*1. 固定周期任务 （执行优先度 5）仅在 NX 系列 CPU 单元时可以使用。不可在 NJ 系列 CPU 单元中使用。
*2. 使用主固定周期任务和固定周期任务 （执行优先度 5）两个任务，并列执行运动控制，这被称为 “ 多运动 ”。
*3. CPU 单元中包含执行优先度为 17 及 18 的固定周期任务。但是，这些任务中不可使用运动控制指令。此外，不会执行 I/O 刷新。

使用注意事项

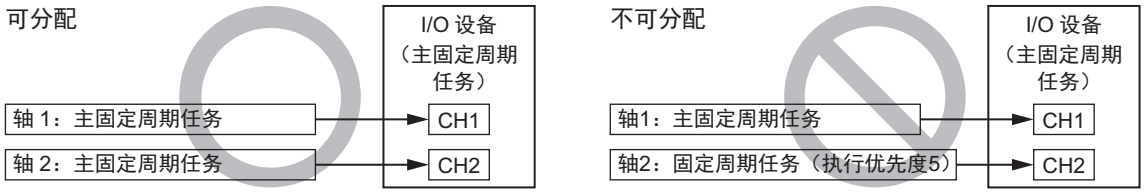
运动控制指令可在主固定周期任务和固定周期任务 （执行优先度 5 和 16）中使用。
若在上述以外的任务中使用运动控制指令，用 Sysmac Studio 链接时将发生异常。

参考

在 NX 系列 CPU 单元上，可使用主固定周期任务和固定周期任务 （执行优先度 5）执行运动控制。需要区分两种运动控制时，主固定周期任务的运动控制称为 “ 运动控制 ”，固定周期任务 （执行优先度 5）的运动控制称为 “ 运动控制 2”。

任务的分配

- 轴及轴组可分配到主固定周期任务或固定周期任务 （执行优先度 5）中的任意一个任务中。分配到轴的 I/O 设备的任务必须与所分配轴的任务为同一个。



- 对分配到主固定周期任务的轴及轴组，可通过固定周期任务 （执行优先度 16）中运行的用户程序执行运动控制指令。

使用注意事项

- 对分配到主固定周期任务的轴及轴组，不可通过固定周期任务 （执行优先度 5）中运行的用户程序执行运动控制指令。若执行，将发生 “ 执行 ID 超出设定范围 （事件代码：57490000 Hex） ”。

  同样，对分配到固定周期任务 （执行优先度 5）的轴及轴组，不可通过主固定周期任务中运行的用户程序执行运动控制指令。
- 对分配到固定周期任务 （执行优先度 5）的轴及轴组，不可通过固定周期任务 （执行优先度 16）中运行的用户程序执行运动控制指令。若执行，将发生 “ 执行 ID 超出设定范围（事件代码：57490000 Hex） ”。

参考

关于轴及轴组，请参考 *「第 3 章 轴和轴组的设定」*。

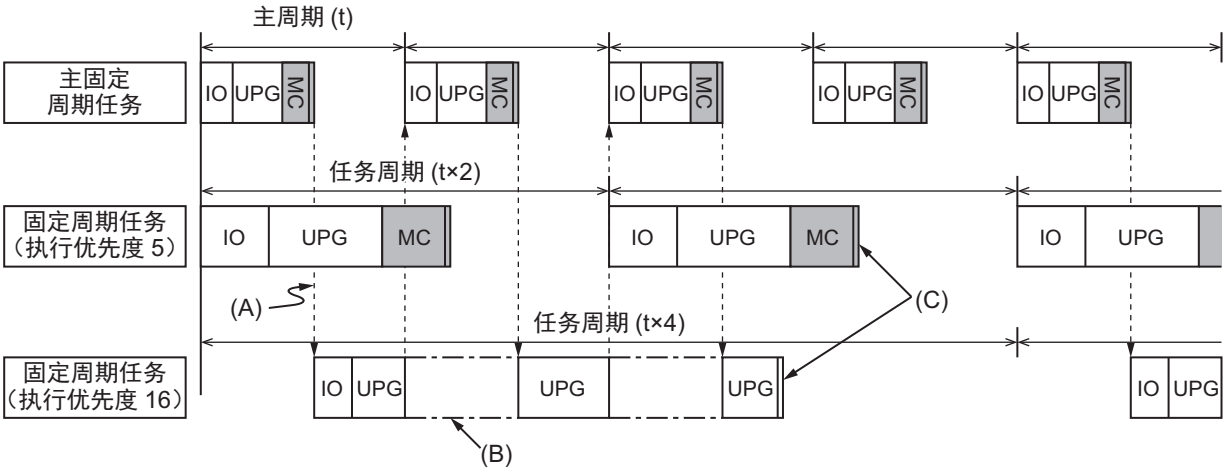
任务的基本动作

● 任务整体的动作

主固定周期任务和固定周期任务以主固定周期任务的任务周期 （以下称 “ 主周期 ”）为基准进行动作。

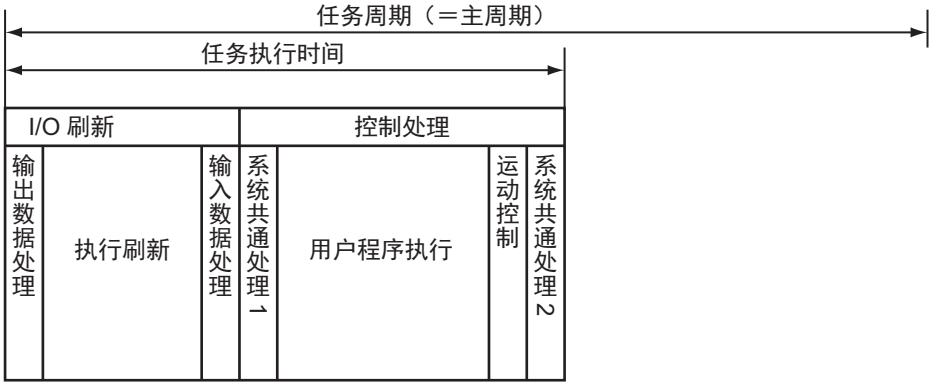
主固定周期任务和固定周期任务 （执行优先度 5）中除了 I/O 刷新、用户程序执行外，还包括系统共通处理及运动控制。

用户程序中记述的运动控制指令会在执行该任务的 END 指令后，执行下次运动控制 (MC) 时进行运算。



符号	内容
IO	I/O 刷新
UPG	用户程序执行
MC	运动控制
(A)	虚线表示任务变化。
(B)	短划线表示任务处理中断状态。
(C)	双线表示任务内处理完成。

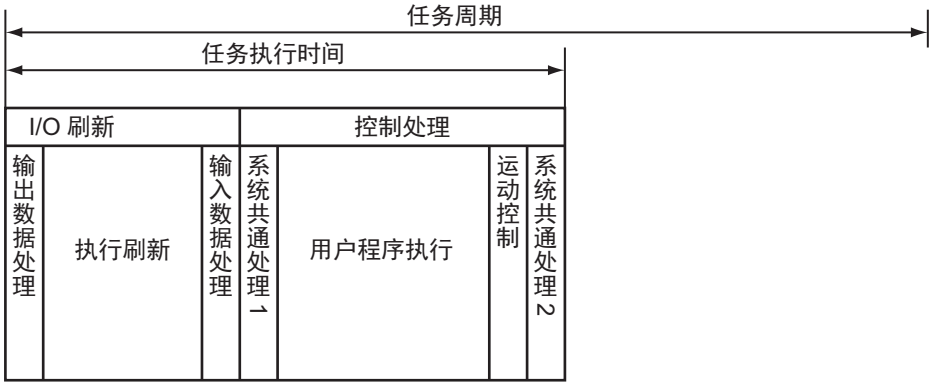
● 主固定周期任务的动作



处理	处理内容
输出数据处理	• 生成向执行 I/O 刷新的 I/O 输出的 OUT 刷新数据。 • 设定为强制值刷新时，将在 OUT 刷新数据中反映强制值 （输出）。
执行刷新	与 I/O 执行数据交换。
输入数据处理	• 判断事件任务的条件式是否一致。 • 导入执行了 I/O 刷新的 I/O （输入）单元之 IN 刷新数据。 • 设定为强制值刷新时，将在导入的 IN 刷新数据中反映强制值 （输入）。
系统共通处理 1	• 变量的任务间排他控制处理 （参照任务时） • 运动输入处理 *1 • 数据追踪处理 （采样、触发判定）
用户程序执行	按分配顺序执行任务中分配的用户程序。
运动控制 1 *2	通过主固定周期任务和固定周期任务 （执行优先度 16）的用户程序中的运动控制指令，执行运动控制指令。 运动输出处理 *3
系统共通处理 2	• 变量的任务间排他控制处理 （更新任务时） • 为确保与任务执行的同时性，从控制器外部进行的变量访问处理 （［任务设定］中设定的 “ 变量访问时间 ” 分） • EtherNet/IP 标签数据链接处理时，若对标签 （网络公开属性的变量）设定更新任务，将作为上述变量访问处理执行。

*1. 根据从伺服驱动器接收的数据，更新运动控制系统变量的伺服驱动器状态或轴当前值。
*2. 运动控制 1 中分配的轴的系统定义变量使用 “_MC_AX[0-255]” 或 “_MC1_AX[0-255]”。同样，轴组的系统定义变量使用 “_MC_GRP[0-63]” 或 “_MC1_GRP[0-63]”。
轴的系统定义变量请参考 🕮 *「3-1-3 轴变量的概要* (P.3-6)」，轴组的系统定义变量请参考 🕮 *「3-3-3 轴组变量的概要* (P.3-21)」。
*3. 在下个主固定周期任务的 “I/O 刷新 ” 时，向伺服驱动器等发送。

● 执行优先度 5 的固定周期任务动作

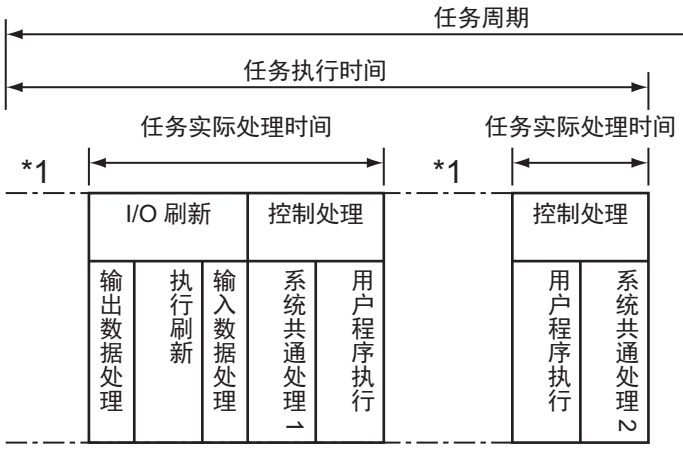


处理	处理内容
输出数据处理	• 生成向执行 I/O 刷新的 I/O 输出的 OUT 刷新数据。 • 设定为强制值刷新时，将在 OUT 刷新数据中反映强制值 （输出）。
执行刷新	与 I/O 执行数据交换。
输入数据处理	• 导入执行了 I/O 刷新的 I/O （输入）单元之 IN 刷新数据。 • 设定为强制值刷新时，将在导入的 IN 刷新数据中反映强制值 （输入）。
系统共通处理 1	• 变量的任务间排他控制处理 （参照任务时） • 运动输入处理 *1 • 数据追踪处理 （采样、触发判定）
用户程序执行	按分配顺序执行任务中分配的用户程序。
运动控制 2 *2	• 通过固定周期任务 （执行优先度 5）的用户程序中的运动控制指令，执行运动控制指令。 • 运动输出处理 *3
系统共通处理 2	• 变量的任务间排他控制处理 （更新任务时） • 为确保与任务执行的同时性，从控制器外部进行的变量访问处理 （［任务设定］中设定的 “ 变量访问时间 ” 分）

*1. 根据从伺服驱动器接收的数据，更新运动控制系统变量的伺服驱动器状态或轴当前值。
*2. 运动控制 2 中分配的轴的系统定义变量使用 “_MC2_AX[0-255]”。同样，轴组的系统定义变量使用 “_MC2_GRP[0-63]”。
轴的系统定义变量请参考 🕮「*3-1-3 轴变量的概要* (P.3-6)」，轴组的系统定义变量请参考 🕮「*3-3-3 轴组变量的概要* (P.3-21)」。
*3. 在下个固定周期任务 （执行优先度 5）的 “I/O 刷新 ” 时，向伺服驱动器等发送。

● 执行优先度 16 的固定周期任务动作

固定周期任务 （执行优先度 16）可执行 I/O 刷新。



*1. 若执行优先度更高的任务，将临时中断任务执行。

任务的周期

单任务时，以主固定周期任务的任务周期，即主周期为基准进行动作。
单任务时，主周期自动变为运动控制周期 （= EtherCAT 通信的过程数据通信周期）。
NJ 系列 CPU 单元中，仅为单任务的控制。

多运动时，以主周期和固定周期任务 （执行优先度 5）的任务周期动作。
多运动时，运动控制周期也变为 2 个， EtherCAT 通信的过程数据通信周期也自动变为不同的任务周期。

固定周期任务与主周期同步执行。固定周期任务的任务周期请设定为主周期的整数倍。
例如，设定为主周期 1ms、固定周期任务 （执行优先度 5）的任务周期 2ms、固定周期任务 （执行优先度 16）的任务周期 4ms 时，则以每两个主周期一次为间隔，主固定周期任务和固定周期任务 （执行优先度 5）的周期起点重叠。同样，固定周期任务 （执行优先度 16）和主固定周期任务以四个周期一次为间隔，周期起点重叠。
任务周期的详情请参考 🕮《NJ/NX 系列 CPU 单元 用户手册 软件篇 （SBCA-CN5-359）》。

参考

需要区分两个过程数据通信周期时，主固定周期任务的通信周期称为 “ 过程数据通信周期 1”，固定周期任务 （执行优先度 5）的通信周期称为 “ 过程数据通信周期 2”。

● NX 系列 CPU 单元可设定的任务周期

NX 系列 CPU 单元时，因任务类型不同，可设定的任务周期如下。

任务	可设定的任务周期
主固定周期任务	125μs, 250μs ～ 8ms （可以 250μs 为单位指定）
固定周期任务 （执行优先度 5）	125μs, 250μs ～ 100ms （可以 250μs 为单位指定）
固定周期任务 （执行优先度 16）	1ms ～ 100ms （可以 250μs 为单位指定）

（注）主固定周期任务和固定周期任务的任务周期设定条件如下。
固定周期任务的任务周期请设定为主固定周期任务的整数倍。
各任务的任务周期最小公倍数请控制在 600ms 以下。

● NJ 系列 CPU 单元可设定的任务周期

在 NJ 系列 CPU 单元上，可组合的主固定周期任务和固定周期任务 （执行优先度 16）的任务周期如下表所示。

主周期	固定周期任务的可设定任务周期
500μs *1	1ms, 2ms, 3ms, 4ms, 5ms, 8ms, 10ms, 15ms, 20ms, 25ms, 30ms, 40ms, 50ms, 60ms, 75ms, 100ms
1ms	1ms, 2ms, 3ms, 4ms, 5ms, 8ms, 10ms, 15ms, 20ms, 25ms, 30ms, 40ms, 50ms, 60ms, 75ms, 100ms
2ms	2ms, 4ms, 8ms, 10ms, 20ms, 30ms, 40ms, 50ms, 60ms, 100ms
4ms	4ms, 8ms, 20ms, 40ms, 60ms, 100ms

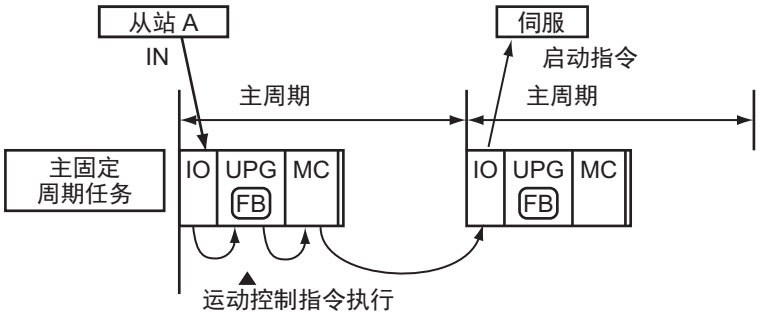
*1. NJ301-□□□□ 可在单元版本 1.03 以上的 CPU 单元中使用。 NJ101-10□□ 不可使用。

2-3-2 运动控制的任务动作示例

运动控制指令可在主固定周期任务和固定周期任务 （执行优先度 5 和 16）中记述。
下面介绍不同情况下的任务动作示例。

在主固定周期任务中使用运动控制指令时

要求高速的运动控制时，在主固定周期任务中记述运动控制指令 (FB)。



1 **导入数据**

通过 I/O 刷新 (IO) 导入 EtherCAT 从站 （从站 A）的输入数据。

2 启动指令

在用户程序执行 (UPG) 中，按照导入的输入数据启动运动控制指令 (FB)。
运动控制指令的输出变量在此时更新。

3 生成指令

按照启动的运动控制指令 (FB)，在该主固定周期任务后的运动控制 (MC) 中，执行运动计算，生成伺服驱动器的启动指令等。

4 发送指令

生成的启动指令通过下一周期的 I/O 刷新 (IO) 向伺服驱动器等发送。

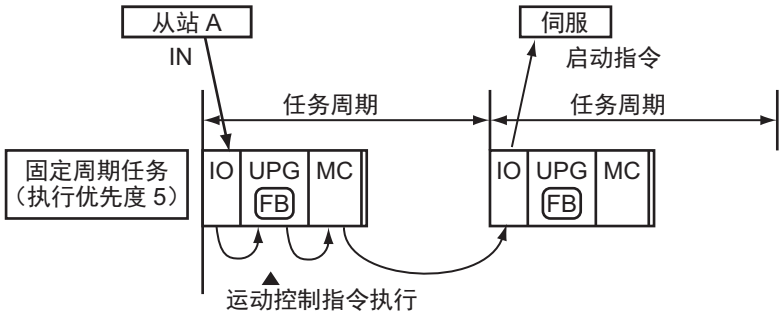
参考

从输入到向伺服驱动器输出启动指令之间的处理非常快速，因此建议在主固定周期任务中记述运动控制指令。

在执行优先度 5 的固定周期任务中使用运动控制指令时

要求仅次于主固定周期任务的高速运动控制时，在固定周期任务 （执行优先度 5）中记述运动控制指令 (FB)。
基本动作与主固定周期任务相同。

● 处理时序



1 导入数据

通过 I/O 刷新 (IO) 导入 EtherCAT 从站 （从站 A）的输入数据。

2 启动指令

在用户程序执行 (UPG) 中，按照导入的输入数据启动运动控制指令 (FB)。
运动控制指令的输出变量在此时更新。

3 生成指令

按照启动的运动控制指令 (FB)，在该主固定周期任务后的运动控制 (MC) 中，执行运动计算，生成伺服驱动器的启动指令等。

4 **发送指令**

生成的启动指令通过下一周期的 I/O 刷新 (IO) 向伺服驱动器等发送。

参考

- 固定周期任务 （执行优先度 5）仅在 NX 系列 CPU 单元时可以使用。不可在 NJ 系列 CPU 单元中使用。
- 在构成装置的功能中，将需要以最快速度控制的功能分配到主固定周期任务，将其他功能分配到固定周期任务 （执行优先度 5），进行并列控制时，使用固定周期任务 （执行优先度 5）。

● 多运动中轴变量的反映时序

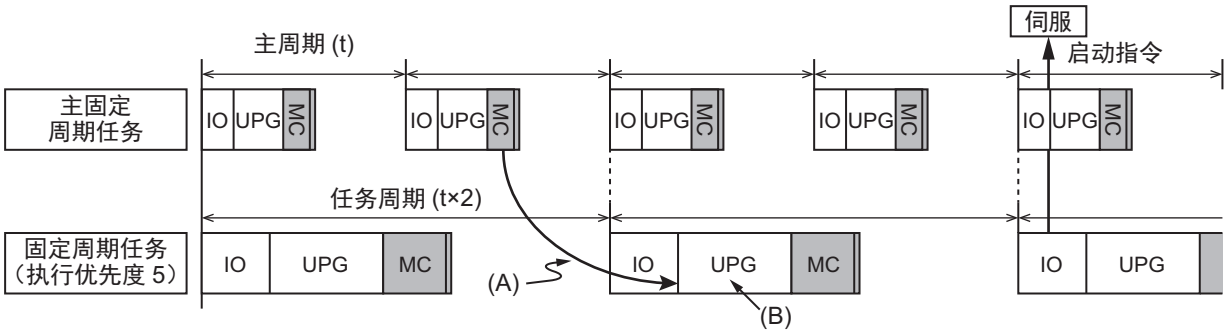
多运动是指使用主固定周期任务和固定周期任务 （执行优先度 5）两个任务执行并列控制。
在多运动中，主固定周期任务控制的轴的变量值可通过固定周期任务 （执行优先度 5）的用户程序参照。相反，固定周期任务 （执行优先度 5）控制的轴的变量值可通过主固定周期任务的用户程序参照。

参考

- 关于轴变量的同步，请参考 *「3-1-4 轴变量的同步* (P.3-8)」。
- 主固定周期任务控制的轴的变量值可通过固定周期任务 （执行优先度 16）的用户程序参照。详情请参考 *「在执行优先度 16 的固定周期任务中使用运动控制指令时* (P.2-14)」及 *「3-1-4 轴变量的同步* (P.3-8)」。

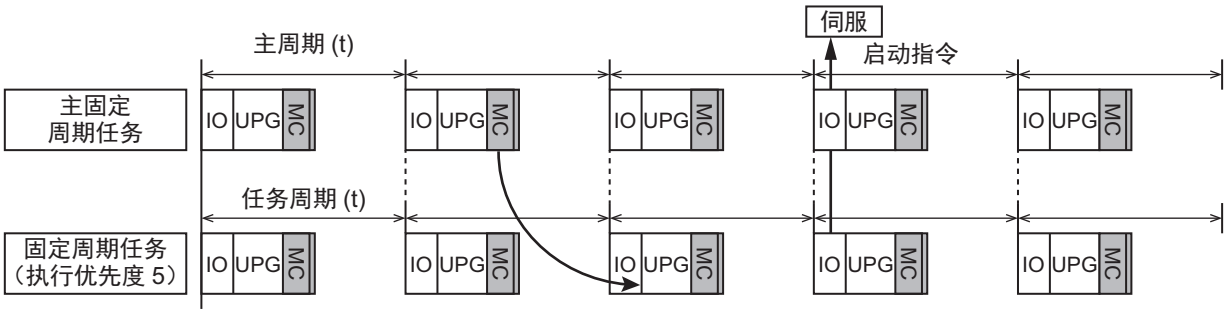
轴变量的值与参照侧的任务周期同步后执行。
参照轴变量值的任务所在的用户程序正在执行时，参照的轴变量值不会发生替换。

主固定周期任务的轴变量值反映到固定周期任务 （执行优先度 5）的时间如下所示。



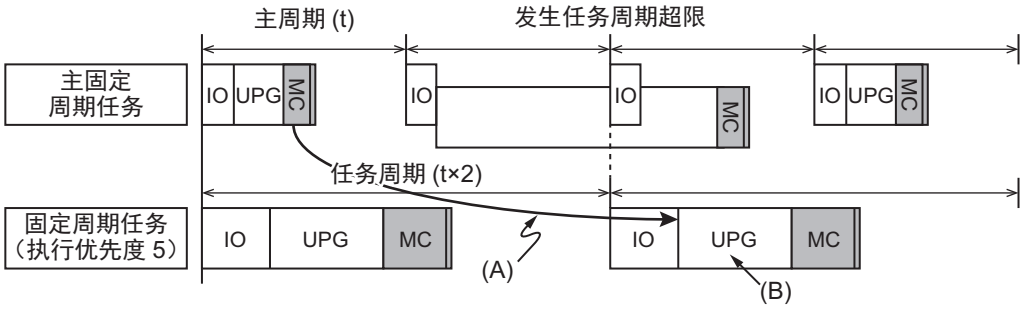
符号	说明
(A)	轴变量的更新。 无论固定周期任务 （执行优先度 5）的用户程序执行开始位置如何，都将反映任务周期起点重叠前的主固定周期任务的执行结果。
(B)	用户程序正在执行时，轴变量的值不会发生替换。

任务周期相同时也一样。



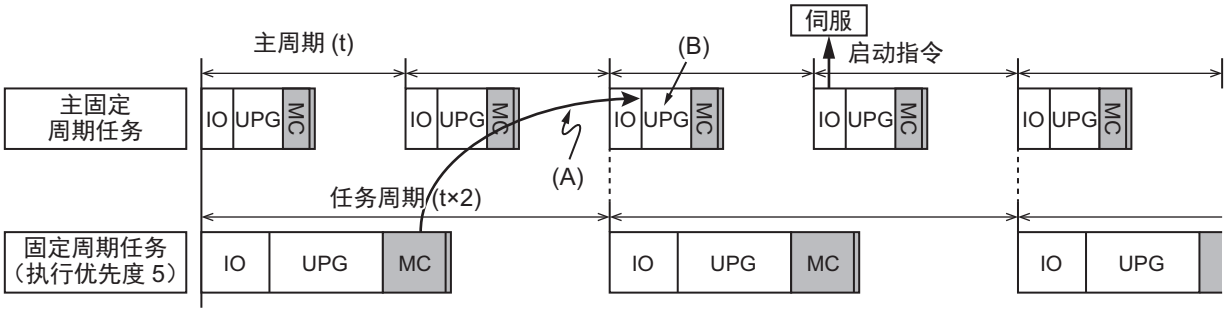
参考

发生任务周期超限时，不会反应任务周期起点重叠前的主固定周期任务的执行结果，而是再前一个周期的执行结果。



符号	说明
(A)	轴变量的更新。 发生任务周期超限时，反映前一个周期的值。
(B)	用户程序正在执行时，轴变量的值不会发生替换。

固定周期任务 （执行优先度 5）的轴变量值反映到主固定周期任务的时间如下所示。



符号	说明
(A)	轴变量的更新。 反映任务周期起点重叠前的固定周期任务 （执行优先度 5）的执行结果。
(B)	用户程序正在执行时，轴变量的值不会发生替换。

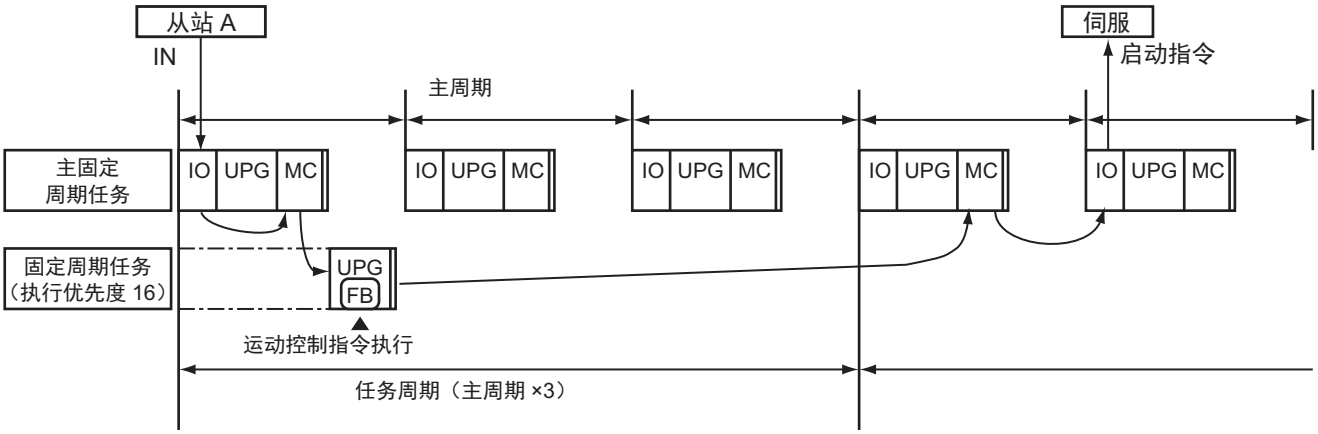
在执行优先度 16 的固定周期任务中使用运动控制指令时

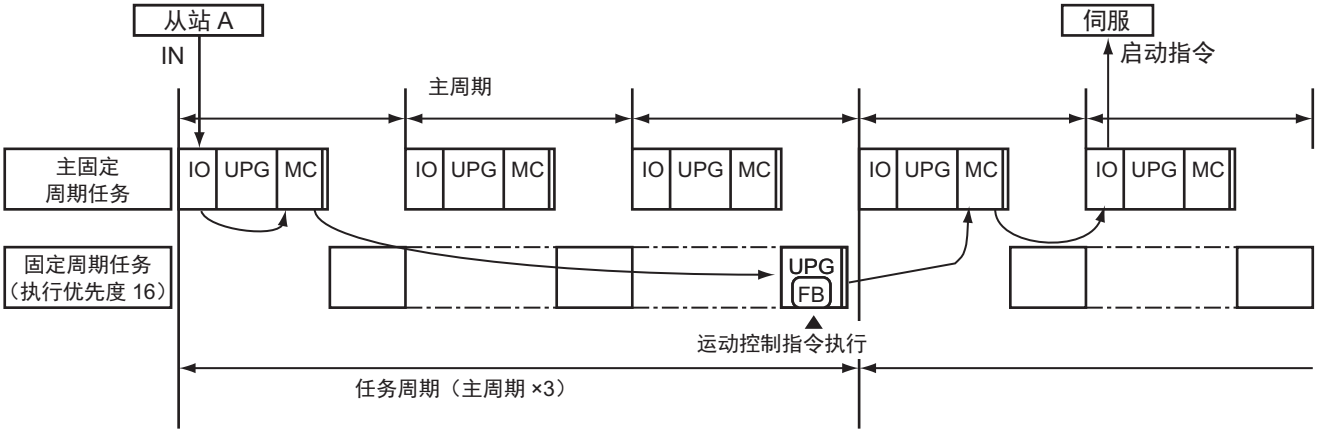
不要求高速运动控制、用户程序较大时，在固定周期任务 （执行优先度 16）中记述运动控制指令。

● 处理时序

以与固定周期任务 （执行优先度 16）相同的任务周期执行的运动控制指令 (FB)，将在相同的时间进行运动控制 (MC)。

因此，可同时启动或中断多个轴。





1 **导入数据**

通过 I/O 刷新 (IO) 导入 EtherCAT 从站 （从站 A）的输入数据。

2 **启动指令**

在固定周期任务 （执行优先度 16）的用户程序执行 (UPG) 中，按照导入的输入数据启动运动控制指令 (FB)。

运动控制指令的输出变量在此时更新。

3 **生成指令**

按照启动的运动控制指令 (FB)，以固定周期任务 （执行优先度 16）的任务周期执行运动控制指令 (FB)。

在该固定周期任务后的主固定周期任务的运动控制 (MC) 中，执行运动计算，生成伺服驱动器的启动指令等。

4 发送指令

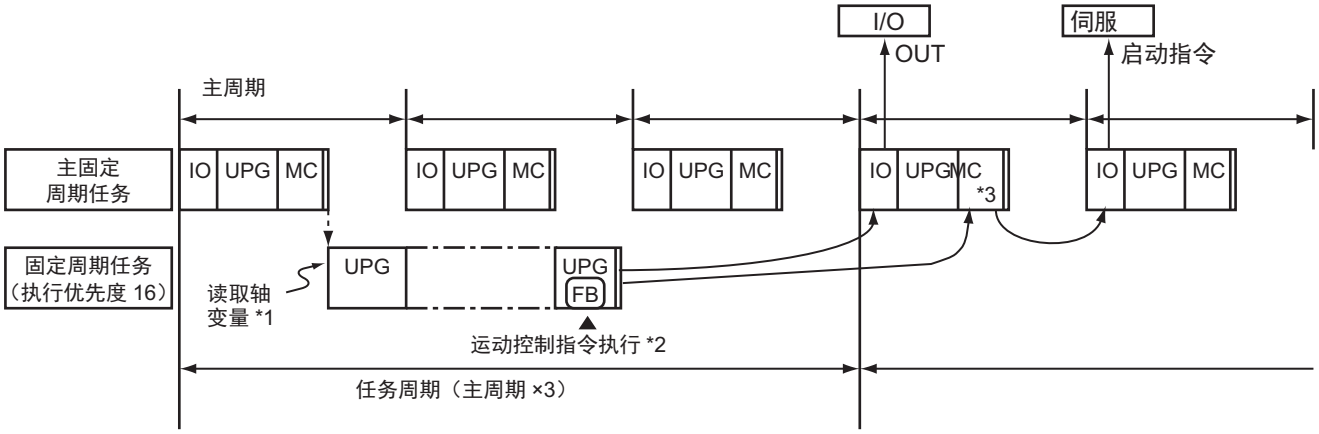
生成的启动指令通过下一周期的 I/O 刷新 (IO) 向伺服驱动器等发送。

● 轴变量的反映时序

轴变量是指由 MC 功能模块的轴参数的一部分和当前位置、异常信息等监视信息组成的系统定义变量。

在固定周期任务 （执行优先度 16）中参照主固定周期任务的轴变量时，将读取开始执行固定周期任务 （执行优先度 16）时的值。

此外，执行运动控制指令 (FB) 时，轴变量的值不会发生替换。在下一固定周期任务 （执行优先度 16）中第一个处理的运动控制 (MC) 中替换。



*1. 读取开始执行固定周期任务 （执行优先度 16）的用户程序时的主固定周期任务的轴变量值。

*2. 执行固定周期任务 （执行优先度 16）的运动控制指令 (FB) 时，轴变量的值不会发生替换。

*3. 在本运动控制 (MC) 中替换。

使用注意事项

- 在固定周期任务 （执行优先度 16）中记述运动控制指令时，若固定周期任务 （执行优先度 16）的任务周期较长，伺服驱动器的响应时间也会变长。
- 关于异常停止等外部的紧急停止，请事先确保能安全停止。
- 编写程序时，请注意固定周期任务 （执行优先度 16）中记述的运动控制指令执行时间和 I/O 控制的执行时间是不同的。

参考

关于轴变量，请参考 *「3-1-3 轴变量的概要* (P.3-6)」。

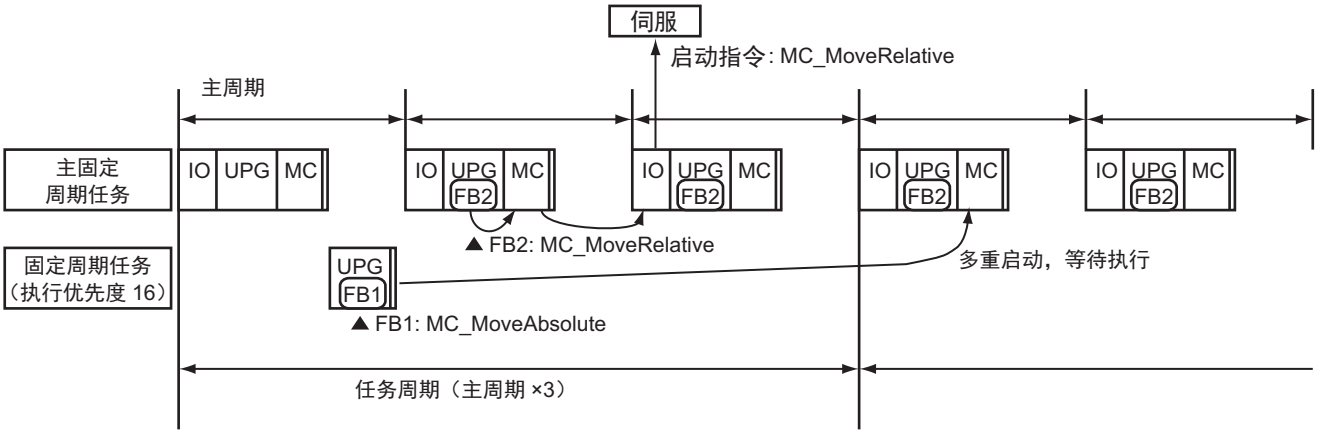
在 2 种任务中使用运动控制指令时

同一个轴中有要求高速运动控制的处理和不要求的处理时，在主固定周期任务和固定周期任务 （执行优先度 16）中均记述运动控制指令 (FB)。

在固定周期任务 （执行优先度 16）的任务周期内，若启动两种任务的运动控制指令 (FB)， MC 功能模块将先执行主固定周期任务指令的运动计算。

例如，在固定周期任务 （执行优先度 16）中启动 MC_MoveAbsolute。然后，在主固定周期任务中，对同一个轴启动 MC_MoveRelative，则动作如下。

- MC 功能模块将先执行 MC_MoveRelative，将 MC_MoveAbsolute 作为指令的多重启动来执行。



运动控制指令的输出变量值或运动控制系统变量的值，因启动该指令的任务的 I/O 刷新时间不同而变化。因此，若在多个任务中对同一个轴使用运动控制指令，状态可能因任务的不同而异。
请在充分理解各任务的处理后编写程序。

使用注意事项

- 在主固定周期任务和固定周期任务 （执行优先度 16）上均记述同一轴的运动控制指令时，编写程序时请注意运动控制指令的执行顺序、运动控制系统变量的更新时间及指令值的输出时间。
- 在多个任务中使用同一轴的运动控制系统变量时，编写程序时请注意运动控制系统变量的更新时间的不同。

参考

关于多重启动，请参考 *「9-5-7 运动控制指令的多重启动 （缓冲模式）*(P.9-45)」。

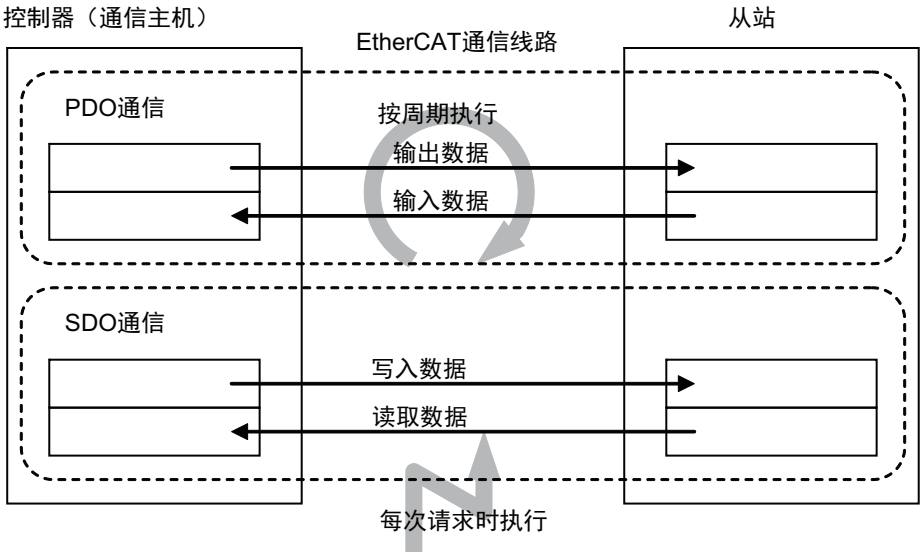
2-4 与 EtherCAT 通信的关系

MC 功能模块通过 CPU 单元内置的 EtherCAT 主机功能模块的 PDO 通信来控制伺服驱动器、计数器及 NX 系列位置接口单元。
下面介绍与 EtherCAT 通信相关的 MC 功能模块项目。

2-4-1 CAN application protocol over EtherCAT (CoE)

MC 功能模块使用 CAN application protocol over EtherCAT (CoE) 协议与 EtherCAT 上的从站进行信息交换。
CoE 中，各种从站拥有的参数、控制信息由名为对象字典 (OD) 的数据规格规定。
为了在控制器 / 通信主机和从站之间传达这些数据，可使用以固定周期进行实时信息交换的过程数据对象 (Process Data Objects: PDO) 和在任意时间传输信息的服务数据对象 (Service Data Objects: SDO)。

MC 功能模块中，对于伺服电机的位置控制等以固定控制周期更新输入输出数据的指令，使用 PDO 通信。
与之不同的是，对于参数传送等以指定时间读取或写入数据的指令，则使用 SDO 通信。



2-4-2 EtherCAT 主机功能和运动控制功能模块的关系

NJ/NX 系列 CPU 单元中可连接 EtherCAT 从站，执行顺序控制或运动控制。

顺序控制

- 若在 Sysmac Studio 的 EtherCAT 构成编辑画面中创建 EtherCAT 构成，将自动生成构成从站的 “I/O 端口 ”。
- 在 Sysmac Studio 的 I/O 映射中分配设备变量。
- 用运动控制指令以外的指令执行顺序控制。

运动控制

- 若在 Sysmac Studio 的 EtherCAT 构成编辑画面中创建 EtherCAT 构成，将自动生成构成从站的 “I/O 端口 ”。
- 在运动控制设定视图中创建轴变量，分配作为运动控制对象的 EtherCAT 从站。
- 用运动控制指令执行运动控制。

轴变量中可分配的设备有 EtherCAT 从站的伺服驱动器和编码器输入终端，以及 NX 系列位置接口单元。

参考

- 在运动控制指令以外的指令中，对于轴变量中分配的 EtherCAT 从站或 NX 系列位置接口单元，不可直接通过 PDO 通信发送指令。但是，可通过轴变量间接地参照状态。
- 使用 SDO 通信，还可对轴变量中分配的 EtherCAT 从站或 NX 系列位置接口单元的对象进行读写。
  但是， PDO 通信中映射的对象请勿通过 SDO 通信写入。若写入，则动作将遵照从站的规格。使用欧姆龙生产的从站时，将发生 SDO 通信异常。
- 若轴变量中未分配 EtherCAT 从站的伺服驱动器、编码器输入终端以及 NX 系列位置接口单元，请与通用 EtherCAT 从站一样，执行顺序控制。

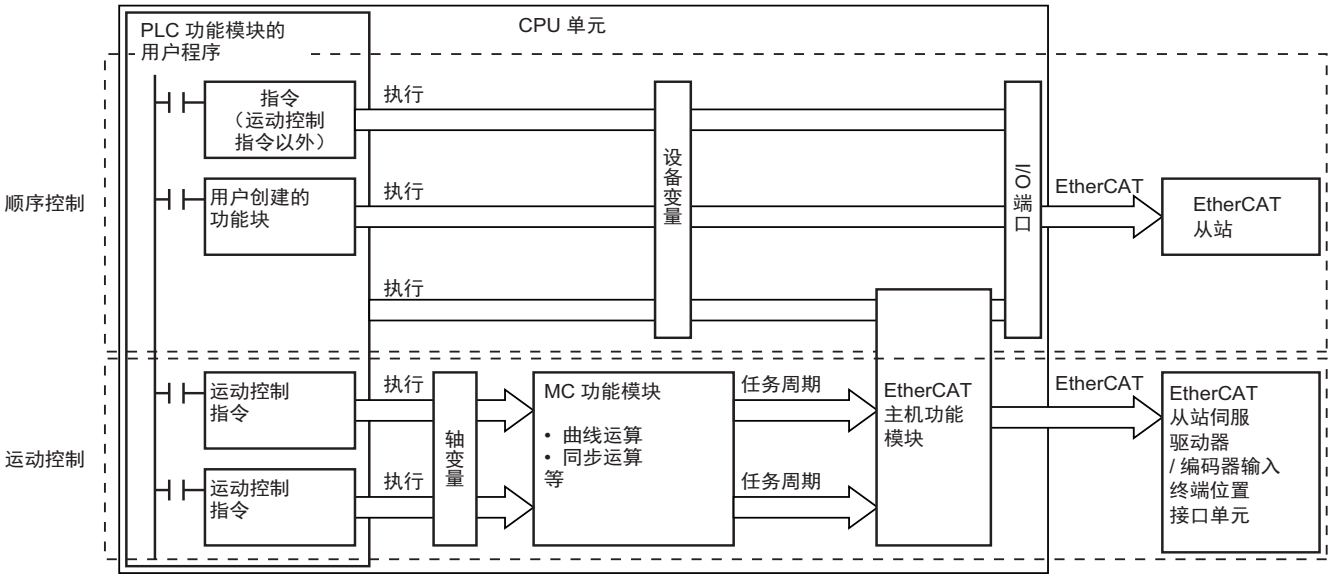
版本相关信息

使用 Sysmac Studio Ver.1.09 以上，可向轴变量中分配的从站和单元的 I/O 端口分配设备变量。可分配的 I/O 端口的条件如下。

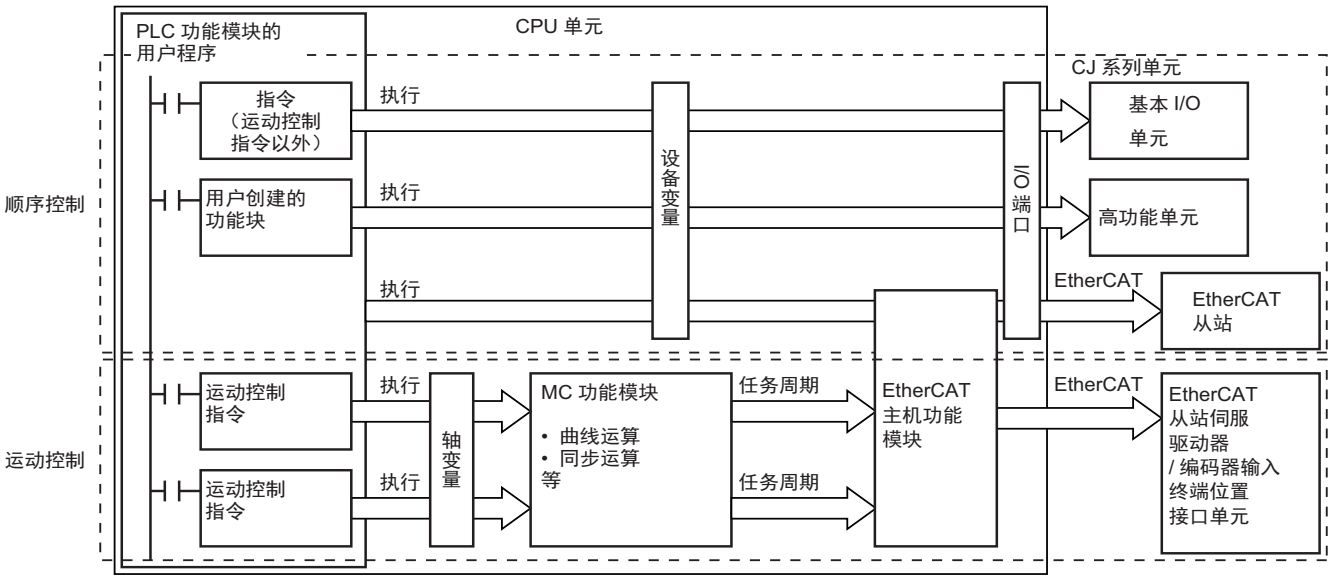
- I/O 端口的 Read/Write 属性为 Read 属性 （R: 只读）的端口
- I/O 端口的 Read/Write 属性为 Write 属性 （W: 只写），在 Sysmac Studio 的 ［轴基本设定］画面的 ［详细设定］中将过程数据设定为 ［＜不分配＞］的端口

用 Sysmac Studio Ver.1.08 或更旧的版本打开了使用 Sysmac Studio Ver.1.09 以上在轴变量中分配设备变量的项目时，设备变量的分配将解除。

● NX 系列 CPU 单元时



● NJ 系列 CPU 单元时



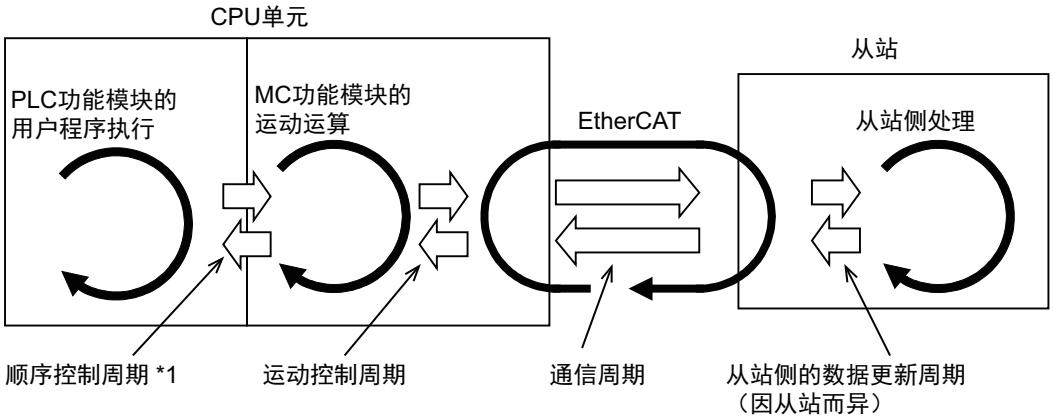
2-4-3 过程数据通信周期和运动控制周期的关系

PLC 基本功能模块通过启动用户程序中的运动控制指令，向运动控制功能模块发行运动控制指令。运动控制功能模块将根据该指令执行运动计算，并将计算结果发送到 EtherCAT 上的伺服驱动器。

以上数据交换按以下周期更新。

● NX 系列 CPU 单元时

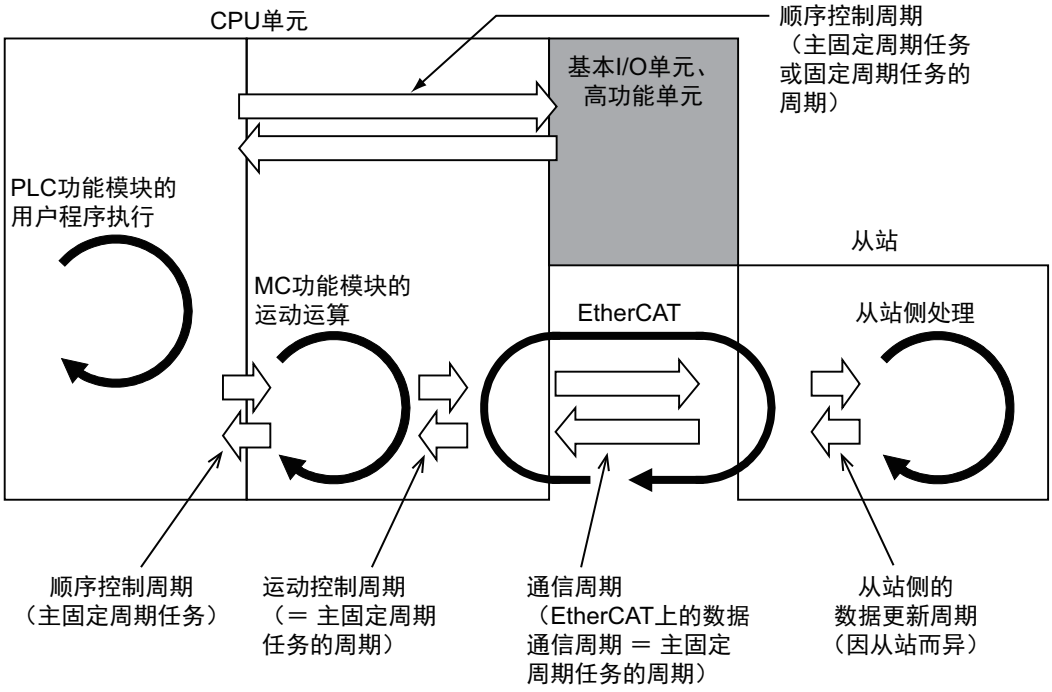
- “ 主周期 ” ＝ “ 运动控制 1 的周期 ” ＝ “EtherCAT 通信的过程数据通信周期 1”
- “ 固定周期任务 （执行优先度 5）的任务周期 ” ＝ “ 运动控制 2 的周期 ” ＝ “EtherCAT 通信的过程数据通信周期 2”



*1. 顺序控制周期为主周期时，运动控制周期及通信周期变为主周期。
顺序控制周期为固定周期任务 （执行优先度 5）的任务周期时，运动控制周期及通信周期变为固定周期任务（执行优先度 5）的任务周期。

● NJ 系列 CPU 单元时

- “ 主周期 ” = “ 运动控制的周期 ” = “EtherCAT 通信的过程数据通信周期 ”



3

轴和轴组的设定

本章介绍轴和轴组的概要、用 Sysmac Studio 的 MC 调试功能动作所需的轴设定项目以及用 Sysmac Studio 创建和设定的步骤。

3-1 轴 3-2
3-1-1 轴的概要 3-2
3-1-2 轴参数的概要 3-3
3-1-3 轴变量的概要 3-6
3-1-4 轴变量的同步 3-8
3-1-5 在用户程序中指定轴的方法 3-8

3-2 轴的设定步骤 3-9
3-2-1 轴的设定流程 3-9
3-2-2 设定步骤 3-9

3-3 轴组 3-19
3-3-1 轴组的概要 3-19
3-3-2 轴组参数的概要 3-20
3-3-3 轴组变量的概要 3-21
3-3-4 在用户程序中指定轴组的方法 3-22

3-4 轴组的设定步骤 3-24
3-4-1 轴组的设定流程 3-24
3-4-2 设定步骤 3-24

3-1 轴

下面介绍 MC 功能模块的轴。

3-1-1 轴的概要

在运动控制系统中，运动控制的对象称为 “ 轴 ”。
轴包括通过 EtherCAT 连接的实际的伺服驱动器、编码器，以及 MC 功能模块内部的虚拟伺服驱动器和编码器。

MC 功能模块的轴种类如下表所示。

轴种类	内容
伺服轴	使用 EtherCAT 从站伺服驱动器或 NX 系列位置接口单元的*1 轴。与实际的伺服驱动器等进行分配后使用。 将 1 台伺服电机作为 1 个轴使用。 使用 NX 系列位置接口单元时，可将脉冲输出单元和数字输入单元等多个设备分配到 1 个轴上。
虚拟伺服轴	MC 功能模块中的虚拟轴。不使用实际的伺服驱动器。作为同步控制的主轴等使用。
编码器轴	使用 EtherCAT 从站编码器输入终端或 NX 系列位置接口单元 *1 的轴。与实际的编码器输入终端等进行分配后使用。 1 台编码器输入终端中有 2 个计数器时，每个计数器作为 1 个轴使用。
虚拟编码器轴	虚拟执行编码器动作的轴。无编码器实物时，临时代替编码器轴使用。*2

*1. 控制对象设备请参考 「*1-4-3 功能规格* (P.1-8)」。
*2. 与更新虚拟编码器轴当前位置的运动控制指令组合使用。 MC 功能模块的版本不支持该指令时，无法使用计数器功能。

MC 功能模块的轴包含以下相关要素。
它们的数量与各型号的最大控制轴数对应。例如， NJ501-1300 为 16 轴、 NJ501-1500 为 64 轴、NX701-1600 为 128 轴、 NX701-1700 为 256 轴。

构成要素	内容	参照页
轴参数	对 MC 功能模块控制的各轴，设定最高速度、点动进给、原点复位等轴动作。 设定轴参数时需要使用 Sysmac Studio。	P. 3-3
轴变量	轴变量是指由 MC 功能模块控制的各轴的当前位置、异常信息等监视信息组成的系统定义变量。 若通过 Sysmac Studio 的多视图浏览器追加轴，将创建轴变量。轴变量的名称 （以下称轴变量名称）可在该视图中设定。	P. 3-6
在用户程序中指定轴的方法	在用户程序中，使用运动控制指令进行运动控制。进行单轴控制的运动控制指令称为 “ 轴指令 ”。 轴指令对哪个轴执行控制，需要在指令的输入输出变量 “Axis （轴） ” 中指定系统定义变量的轴变量名称或 Sysmac Studio 中设定的轴变量名称。	P. 3-8

3-1-2 轴参数的概要

● 轴参数一览

区分	参数名称
轴基本设定	轴编号
	运动控制 *1
	轴使用
	轴种类
	输入设备 / 输出设备
单位转换设定	显示单位
	电机每转 1 圈的脉冲数
	电机每转 1 圈的移动量
	减速机使用 *2
	工件侧每转 1 圈的移动量 *2
	工件侧齿轮比 *2
	电机侧齿轮比 *2
动作设定	最高速度
	启动速度 *3
	点动最高速度
	最大加速度
	最大减速度
	加减速超限
	反转时动作
	速度警告值
	加速度警告值
	减速度警告值
	正方向扭矩警告值
	负方向扭矩警告值
	反馈速度过滤器时间常数
	位置宽度
	位置检查时间
	原点位置范围
扩展动作设定	立即停止输入停止方法
	极限输入停止方法
	驱动器错误重置监视时间
	正方向扭矩限制上限值
	负方向扭矩限制上限值
	立即停止输入逻辑反转 *3
	正方向极限输入逻辑反转 *3
	负方向极限输入逻辑反转 *3
	近原点输入逻辑反转 *3
限位设定	软件限位功能
	正方向软件限位
	负方向软件限位
	位置偏差超限值
	位置偏差警告值
位置计数设定	计数模式
	循环计数器上限设定值
	循环计数器下限设定值
	编码器种类

区分	参数名称
伺服驱动器设定	循环计数器上限设定值
	循环计数器下限设定值
	PDS 状态控制方式
原点复位设定	原点复位动作
	原点输入信号
	原点复位开始方向
	原点检测方向
	正方向极限输入时动作
	负方向极限输入时动作
	原点复位速度
	原点复位接近速度
	原点复位加速度
	原点复位减速度
	原点复位跃度
	原点输入掩码距离
	原点位置偏置
	原点复位推压时间
	原点复位修正值
	原点复位修正速度

*1. NX 系列 CPU 单元时设定。
*2. 可在 CPU 单元版本 1.11 以上和 Sysmac Studio Ver.1.15 以上的组合中使用。
*3. 可在 CPU 单元版本 1.05 以上和 Sysmac Studio Ver.1.06 以上的组合中使用。

关于轴参数的详情，请参考 🕮 「*5-2 轴参数* (P.5-4)」。

● 使用轴时必需的设定项目

对 Sysmac Studio 创建的轴设定以下项目后，即可使用轴。

区分	参数名称	设定内容	参照页
轴基本设定	轴编号	按轴的创建顺序自动设定。	P. 5-6
	运动控制 *1	选择［主固定周期任务］。	
	轴使用	选择［使用轴］。	
	轴种类	选择要控制的轴的种类。	
	输入设备 / 输出设备	选择要分配到轴的 EtherCAT 从站设备的节点地址。 轴种类为［虚拟轴］时无法选择。	

*1. NX 系列 CPU 单元时设定。

● 在 Sysmac Studio 上进行伺服驱动器调试的必需设定项目

若要用 Sysmac Studio 的 MC 调试功能运行通过 EtherCAT 连接的伺服驱动器，需要设定以下项目。

区分	参数名称	设定内容	参照页
轴基本设定	轴编号	按轴的创建顺序自动设定。	P. 5-6
	运动控制 *1	选择［主固定周期任务］。	
	轴使用	选择［使用轴］。	
	轴种类	选择［伺服轴］。	
	输入设备 / 输出设备	选择要分配到轴的 EtherCAT 从站设备的节点地址。 轴种类为［虚拟轴］时无法选择。	
单位转换设定	显示单位	选择 [mm]、[degree] 等显示的单位。	P. 5-10
	电机每转 1 圈的脉冲数	根据编码器分辨率，设定电机每转 1 圈的脉冲数。*2	
	电机每转 1 圈的移动量	根据机械规格，设定电机每转 1 圈的工件移动量。	
	减速机使用 *3	指定是否使用减速机设定。	
	工件侧每转 1 圈的移动量 *3	设定每转 1 圈后工件的移动量。	
	工件侧齿轮比 *3	设定工件侧的齿轮比。	
	电机侧齿轮比 *3	设定电机侧的齿轮比。	
位置计数设定	计数模式	根据机械规格选择。	P. 5-22
限位设定	软件限位功能	根据装置的规格选择。	P. 5-21

*1. NX 系列 CPU 单元时设定。
*2. 例如，编码器分辨率为 10,000 脉冲 / 圈时，请设定为 10,000。
*3. 可在 CPU 单元版本 1.11 以上和 Sysmac Studio Ver.1.15 以上的组合中使用。



使用注意事项

- 实际运行电机时，最高速度、最大加减速度、停止设定等请根据机械的运行条件设定合适的值。
- 欧姆龙生产的伺服驱动器 G5 系列可通过驱动器正面的旋转开关将节点地址设定为固定值。该旋转开关设定为 “00” 时，节点地址以 Sysmac Studio 的 EtherCAT 编辑器中设定的值为准。

  将连接的所有驱动器的旋转开关设定为 “00” 时，即使变更驱动器的连接位置，也不会发生异常。要对装置的各控制对象分配特定的伺服驱动器时，请用旋转开关设定驱动器的节点地址。

3-1-3 轴变量的概要

轴变量是指由 MC 功能模块的轴参数的一部分和当前位置、异常信息等监视信息组成的系统定义变量。
若用 Sysmac Studio 创建轴，将按照创建顺序将轴变量登录到变量表中。
轴变量的数据类型为结构体型，数据类型名称为 “_sAXIS_REF”。

轴变量名称

MC 功能模块的轴变量名称中，包括系统定义变量的轴变量名称和通过 Sysmac Studio 追加轴时的轴变量名称两种。
系统定义变量的轴变量名称为 “_MC_AX[0] ～ _MC_AX[255]”、 “_MC1_AX[0] ～ _MC1_AX[255]”、 “_MC2_AX[0] ～ _MC2_AX[255]”。
若通过 Sysmac Studio 追加轴， “_MC_AX［0] ～ _MC_AX［255］ ” 将设定为默认名称 “MC_Axis000 ～ MC_Axis255”。编号按追加的顺序从前往后排。这些轴变量名称可在 Sysmac Studio 中按不同的轴变更为任意变量名称。
在用户程序中，无论是系统定义变量的轴变量名称，还是通过 Sysmac Studio 追加的轴变量名称，均可使用。

● 使用 _MC_AX[0-255] 时的示例

系统定义变量的轴变量名称（全局变量视图的分配目标 *1）	通过 Sysmac Studio 追加的默认轴变量名称	轴编号示例
_MC_AX[0]	MC_Axis000	轴 0
_MC_AX[1]	MC_Axis001	轴 1
:	:	:
_MC_AX[255]	MC_Axis255	轴 255

*1. 若变更 Sysmac Studio 的全局变量视图中的分配目标栏的名称，将发生异常。



参考

- NX 系列 CPU 单元时，可使用 “_MC_AX[0-255]”、 “_MC1_AX[0-255]” 和 “_MC2_AX[0-255]”。
  NJ 系列 CPU 单元时，只能使用 “_MC_AX[0-63]”。
- NX 系列 CPU 单元时，若 “_MC_AX[0-255]” 和 “_MC1_AX[0-255]” 的轴编号相同，则参照相同的值。任一轴变量均可使用，也可混合使用。
- 主固定周期任务中分配的轴变量为 “_MC_AX[0-255]” 或 “_MC1_AX[0-255]”。
  固定周期任务 （执行优先度 5）中分配的轴变量为 “_MC2_AX[0-255]”。

轴变量的层级和轴变量名称变更示例

以 _MC_AX[0] 为例说明。其他轴变量亦同。

_MC_AX[0]	: 轴变量
_MC_AX[0].Status	: 表示 “ 轴状态 ” 的层级
_MC_AX[0].Status.Ready	: 表示 “ 轴启动准备完成 ” 的变量
_MC_AX[0].Status.Disabled	: 表示 “ 轴无效 ” 的变量
:	
_MC_AX[0].Details	: 表示 “ 轴控制状态 ” 的层级
_MC_AX[0].Details.Idle	: 表示 “ 停止中 ” 的变量
_MC_AX[0].Details.InPosWaiting	: 表示 “ 位置等待 ” 的变量
:	
_MC_AX[0].Cmd	: 表示 “ 轴指令值 ” 的层级
_MC_AX[0].Cmd.Pos	: 表示 “ 指令当前位置 ” 的变量
_MC_AX[0].Cmd.Vel	: 表示 “ 指令当前速度 ” 的变量
_MC_AX[0].Cmd.AccDec	: 表示轴监视器的 “ 指令当前加减速度 ” 的变量
:	
_MC_AX[0].Act	: 表示 “ 轴当前值 ” 的层级
_MC_AX[0].Act.Pos	: 表示 “ 反馈当前位置 ” 的变量
_MC_AX[0].Act.Vel	: 表示 “ 反馈当前速度 ” 的变量
:	
_MC_AX[0].Cfg	: 表示 “ 轴基本设定 ” 的层级
_MC_AX[0].Cfg.AxNo	: 表示 “ 轴编号 ” 的变量
_MC_AX[0].Cfg.AxEnable	: 表示 “ 轴使用 ” 的变量
_MC_AX[0].Cfg.AxType	: 表示 “ 轴种类 ” 的变量
:	
_MC_AX[0].Scale.Units	: 表示 “ 显示单位 ” 的变量
_MC_AX[1]	: 轴变量
:	

● 例

将 “MC_Axis000” 变更为 “MyAxis1” 后，表示 “ 反馈当前位置 ” 的变量可使用 “MyAxis1.Act.Pos” 和 “_MC_AX[0].Act.Pos” 中的任意一个。

关于轴变量的详情，请参考  *「轴变量* (P.6-20)」。

3-1-4 轴变量的同步

主固定周期任务控制的轴的变量值可通过固定周期任务 （执行优先度 5）的用户程序参照。相反，固定周期任务 （执行优先度 5）控制的轴的变量值可通过主固定周期任务的用户程序参照。

此外，主固定周期任务控制的轴变量值可通过固定周期任务 （执行优先度 16）的用户程序参照。

参照任务的类型及轴变量的关系如下。

○: 可参照 ×: 不可参照

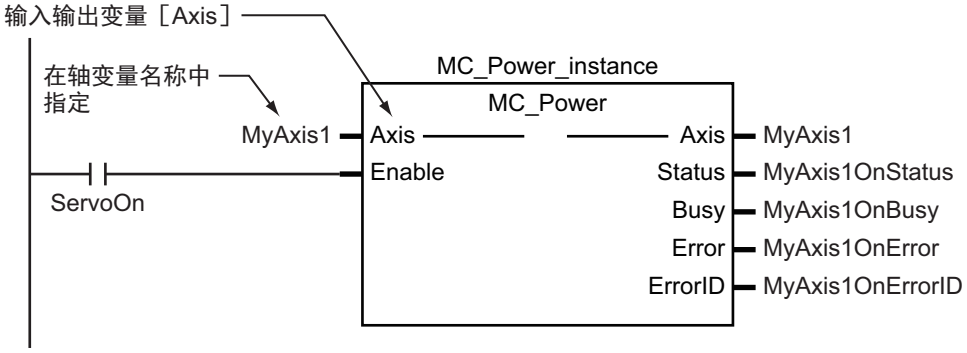
任务类型	运动控制 1 的轴变量		运动控制 2 的轴变量	
	用户定义 *1	系统定义 *2	用户定义 *1	系统定义 *3
主固定周期任务	○	○	○ *4	×
固定周期任务 （执行优先度 5）	○ *5	×	○	○
固定周期任务 （执行优先度 16）	○ *5	○ *6	○ *4	×
固定周期任务 （执行优先度 17、 18）	○ *5	○ *7	○ *4	○ *8
事件任务 （8、 48）	○ *5	○ *7	○ *4	○ *8

*1. 用 Sysmac Studio 创建 “ 轴 ” 时自动生成的用户定义的轴变量 “MC_Axis000” 或 “Loader1” 等。建议使用用户定义的轴变量。
*2. “_MC_AX[*]” 或 “_MC1_AX[*]”。
*3. “_MC2_AX[*]”。
*4. 每个固定周期任务 （执行优先度 5）的任务周期，处理 1 个轴的轴变量。
*5. 每个主周期，处理 1 个轴的轴变量。
*6. 每个主周期，处理最大轴数的轴变量。主固定周期任务的任务执行时间会增加，因此不推荐。
*7. 每个主周期，处理最大轴数的轴变量。
*8. 每个固定周期任务 （执行优先度 5）的任务周期，处理最大轴数的轴变量。

3-1-5 在用户程序中指定轴的方法

在用户程序中，对运动控制指令的输入输出变量 “Axis” 指定轴变量名称。

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



还可使用系统定义变量 “_MC_AX[0]” 代替 “MyAxis1”。

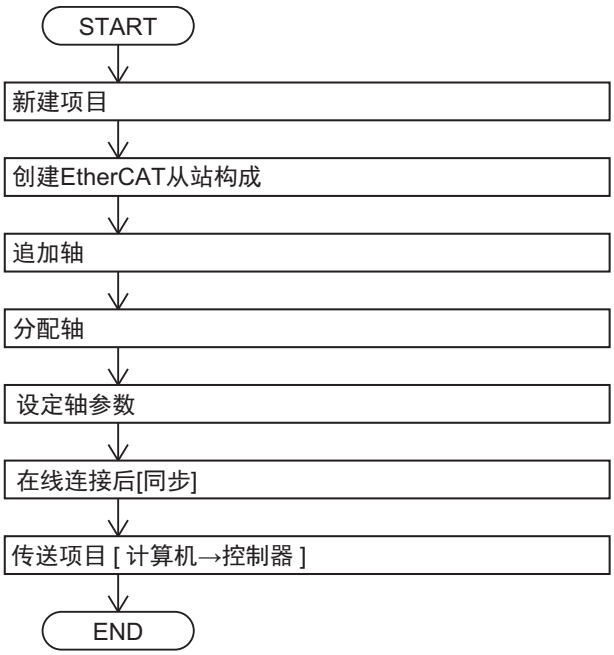
关于运动控制指令的详情，请参考 🕮「*6-2 运动控制指令* (P.6-4)」。

各运动控制指令的详细功能请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的各指令说明。

3-2 轴的设定步骤

下面介绍使用 Sysmac Studio 设定新伺服轴的步骤。

3-2-1 轴的设定流程



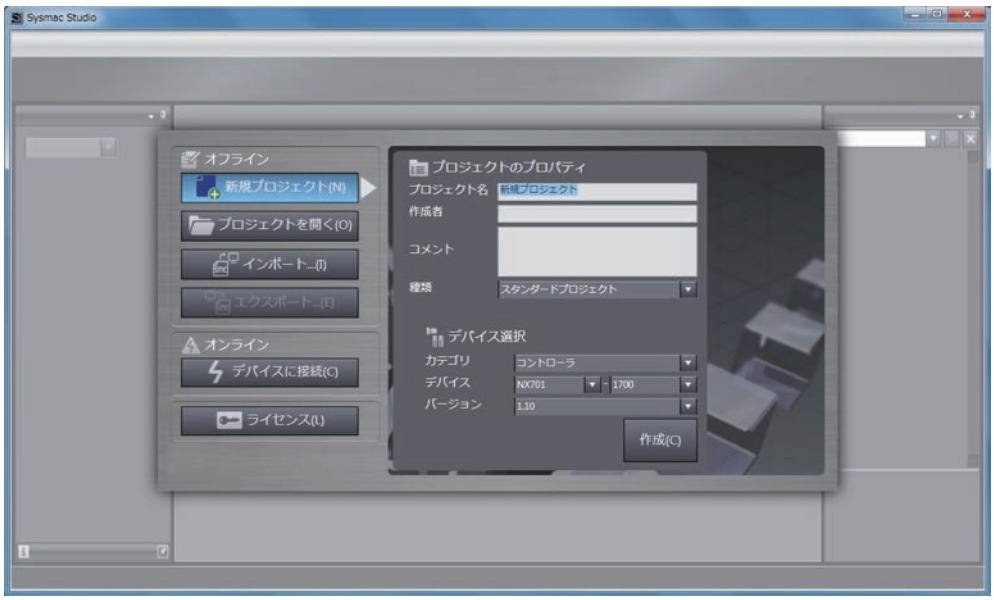
3-2-2 设定步骤

下面介绍轴的设定方法。

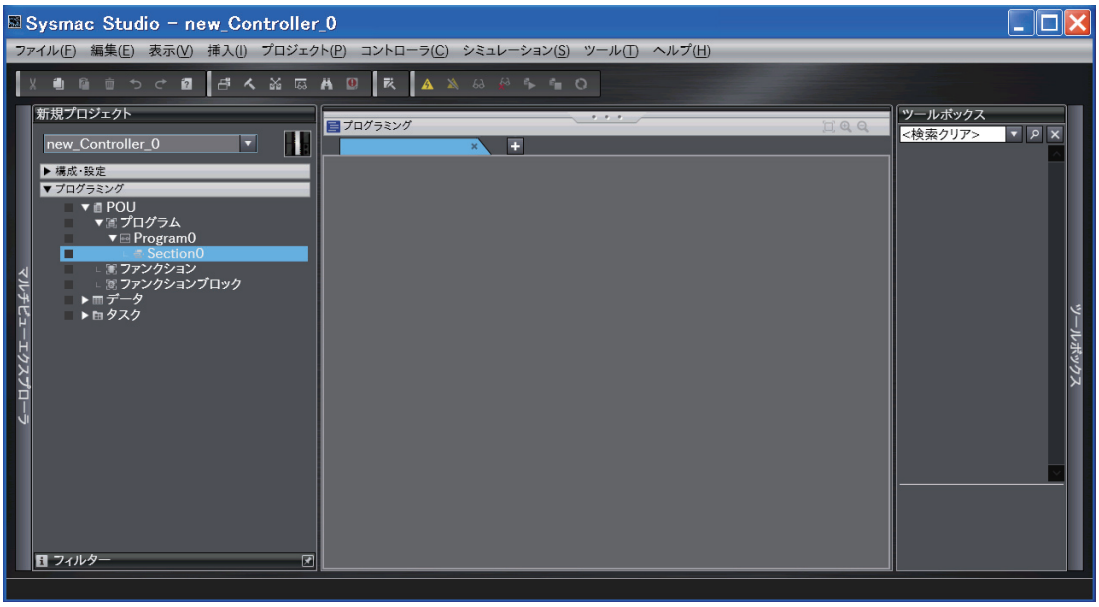
启动 Sysmac Studio

1 **启动 Sysmac Studio，单击［新建项目］按钮。**

2 **设定项目属性和设备选择，单击［创建］按钮。**



显示新项目画面。



创建 EtherCAT 网络配置

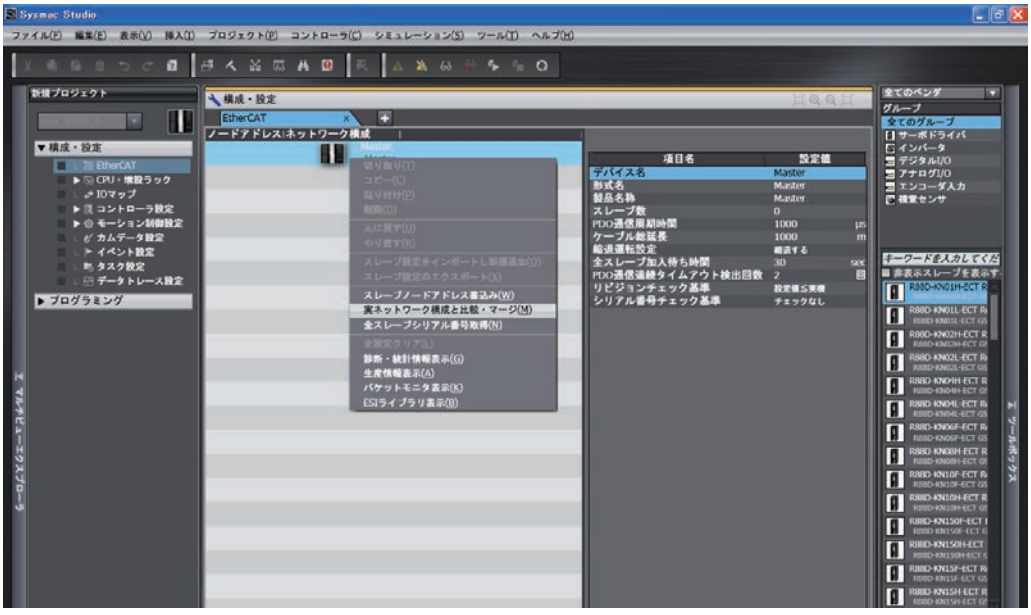
创建 EtherCAT 网络配置的方法有在线创建和离线创建两种。

● 在线创建方法

1 **在多视图浏览器中双击［EtherCAT］。**
显示 EtherCAT 配置编辑画面。

2 **单击［控制器］菜单的［在线］，变为在线状态。**

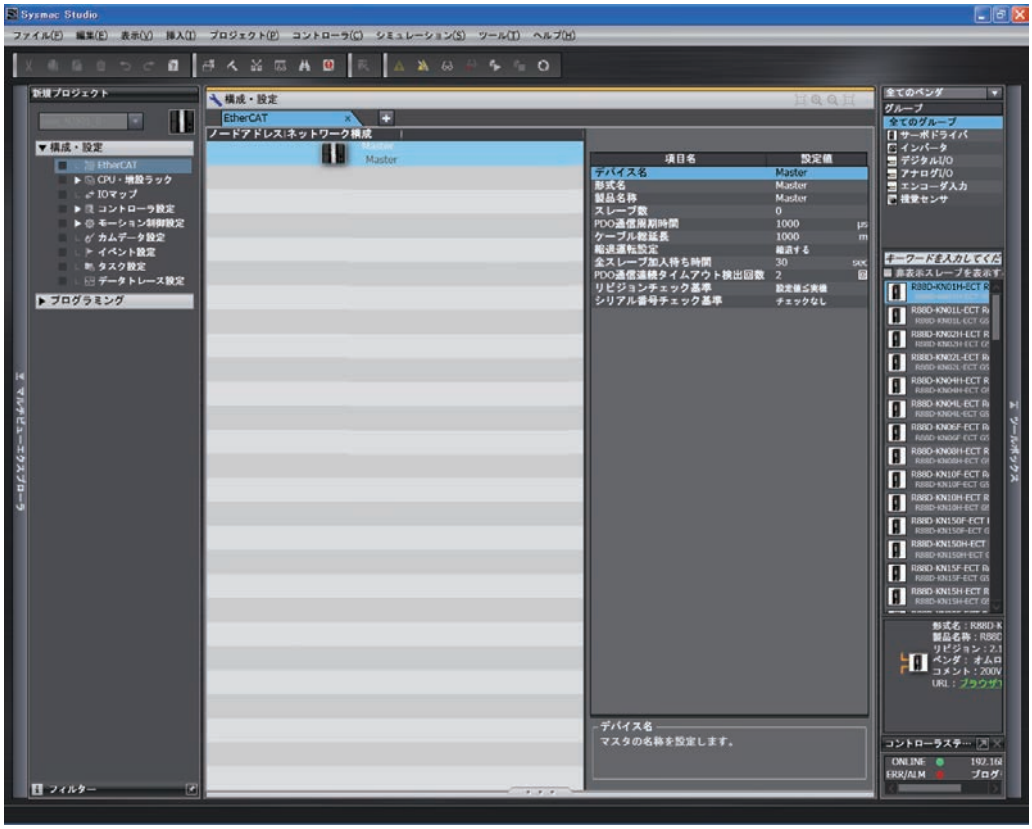
3 **右击画面中的［Master］，选择［与实际网络配置比较和合并］。**



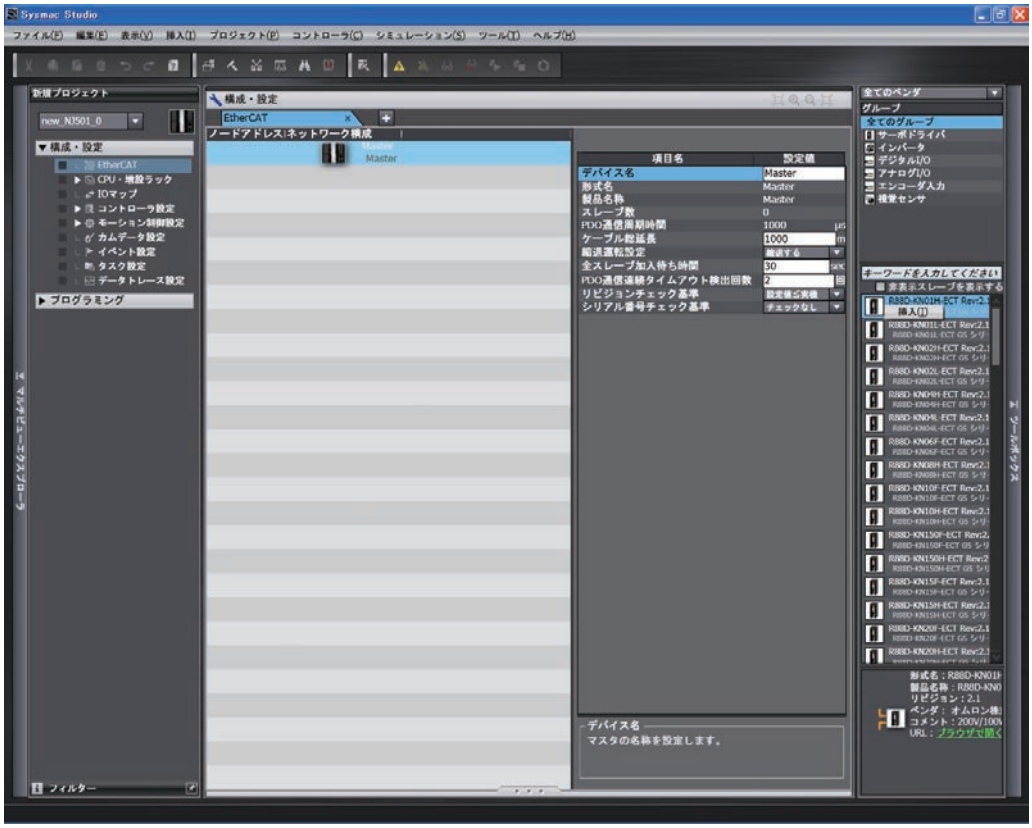
获取信息后，显示 EtherCAT 从站的实体机构成。
右击显示的实体机构成，单击［使用实际网络配置］。

● 离线创建时

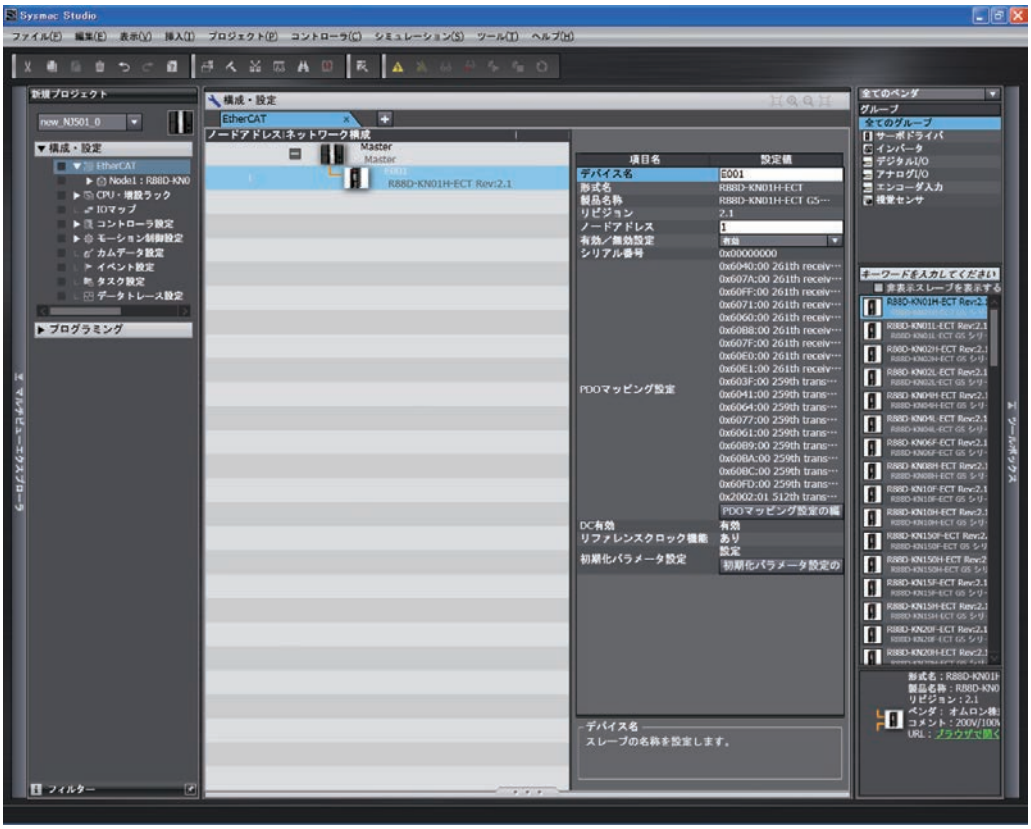
1 **在多视图浏览器中双击［EtherCAT］。**
显示 EtherCAT 配置编辑画面。



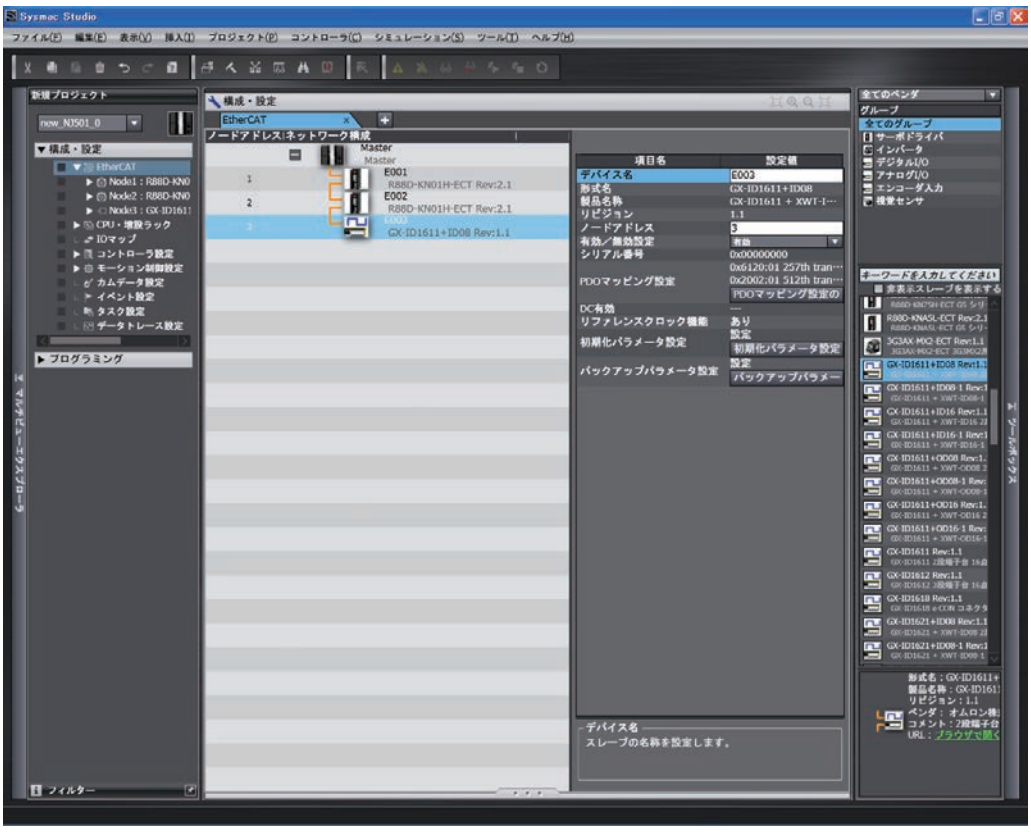
2 **右击要连接的从站，单击［插入］。**



显示插入的从站。

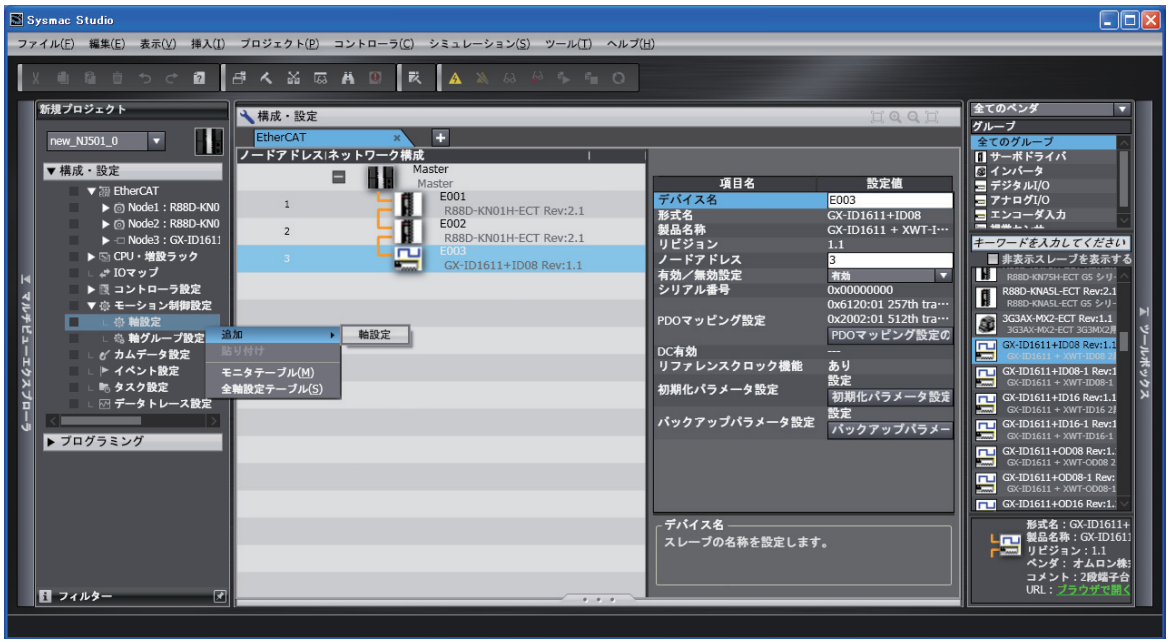


3 **插入所有要使用的从站。**



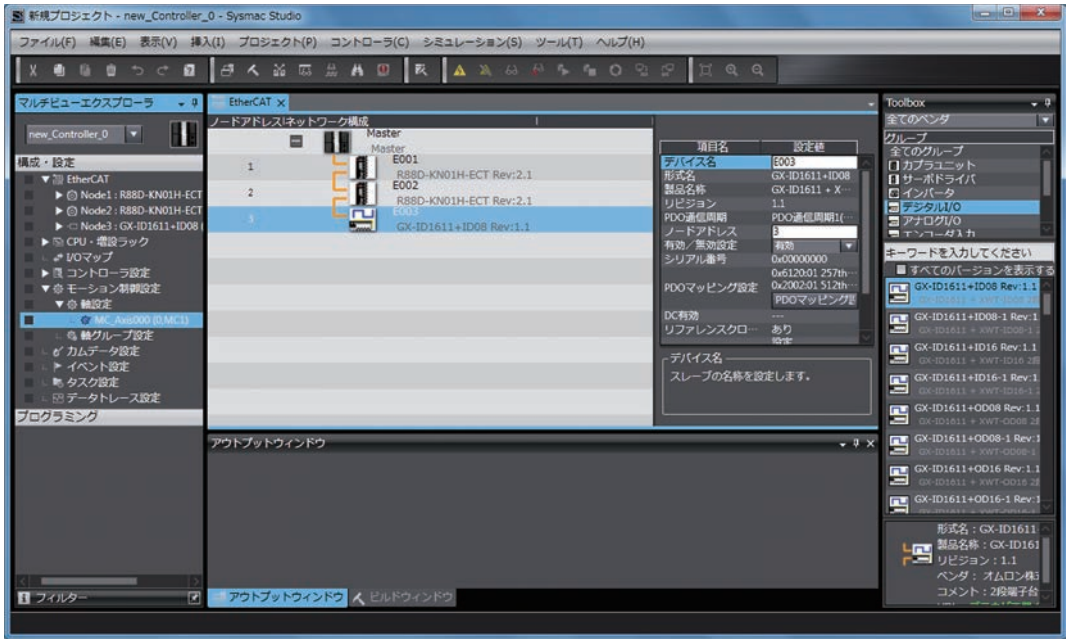
追加轴

1 **在多视图浏览器中右击［轴设定］，单击［追加］中的［轴设定］。**



轴将追加到多视图浏览器中。

所追加轴的默认变量名称为 “MC_Axis000”。

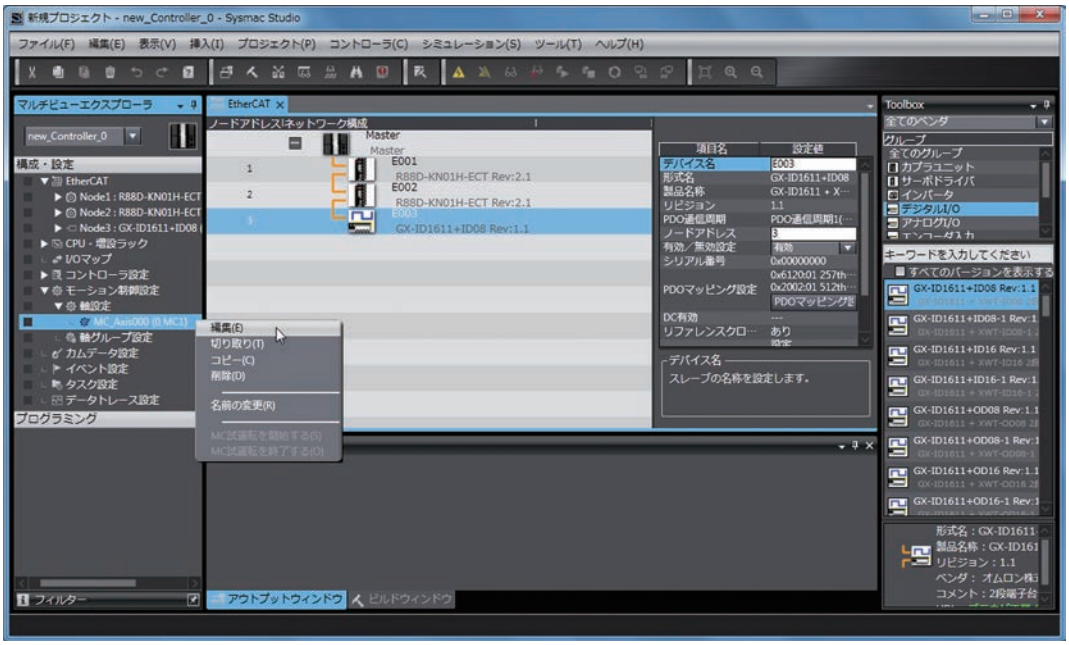


● 通过复制追加轴

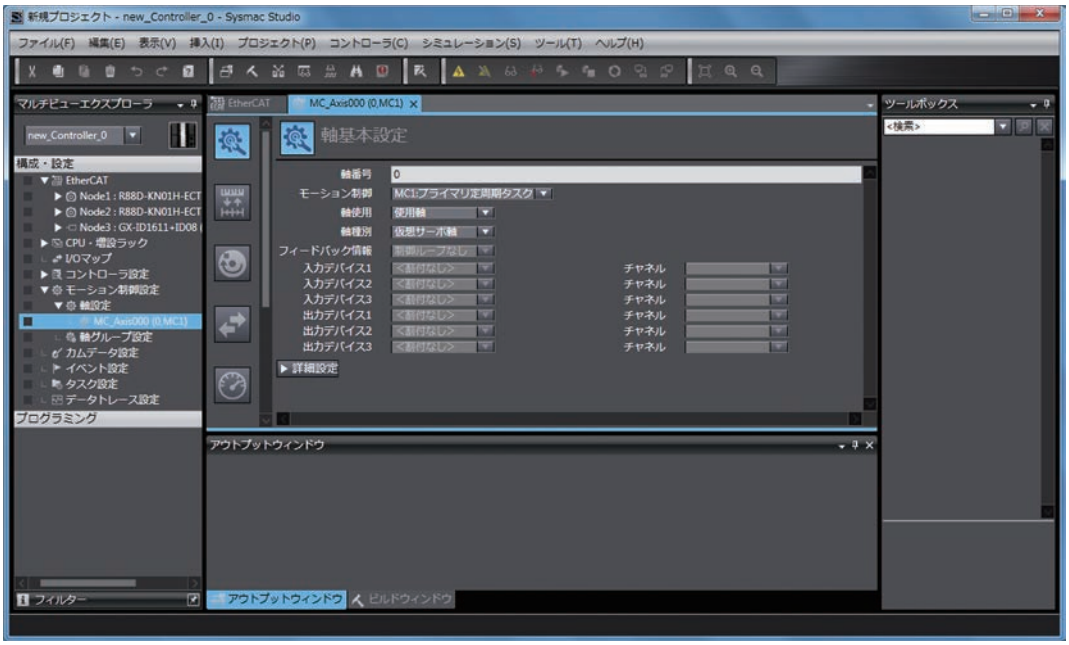
复制已有的轴设定并粘贴，也可追加轴。

分配轴

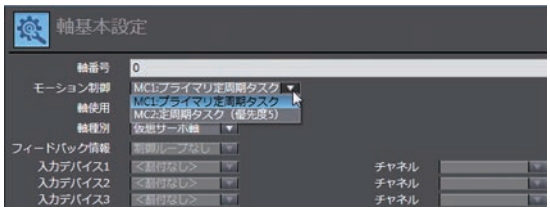
1 **在多视图浏览器中右击追加的轴，单击［编辑］。**



显示轴参数设定画面的［轴基本设定］画面。



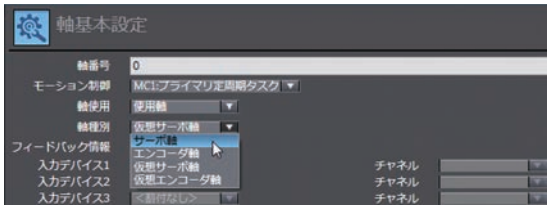
2 **从［运动控制］中选择［主固定周期任务］或［固定周期任务（优先度 5）］。**



参考

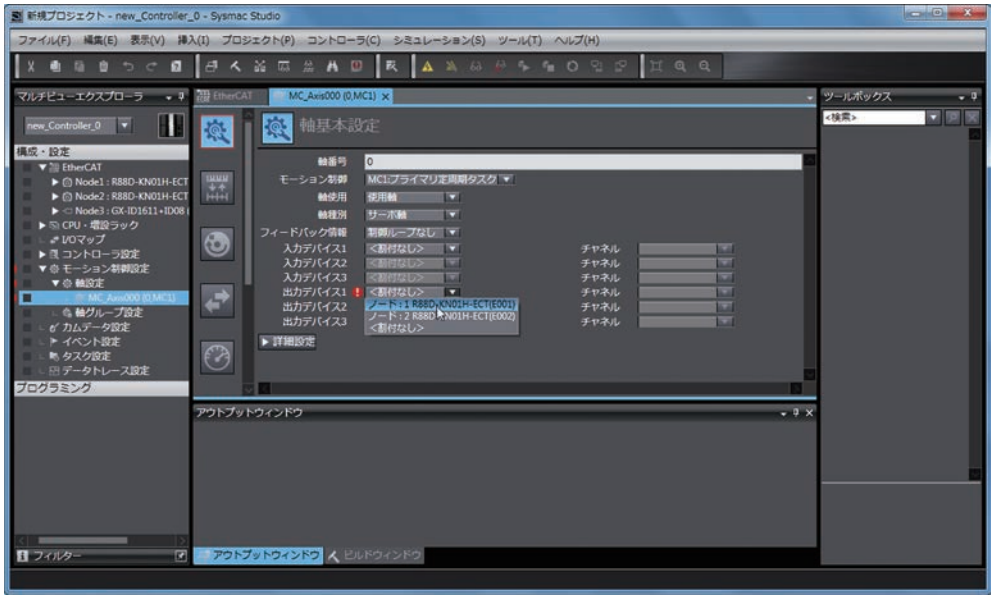
NX 系列 CPU 单元时设定。 NJ 系列 CPU 单元中无需设定。

3 **选择［轴种类］中的［伺服轴］。**



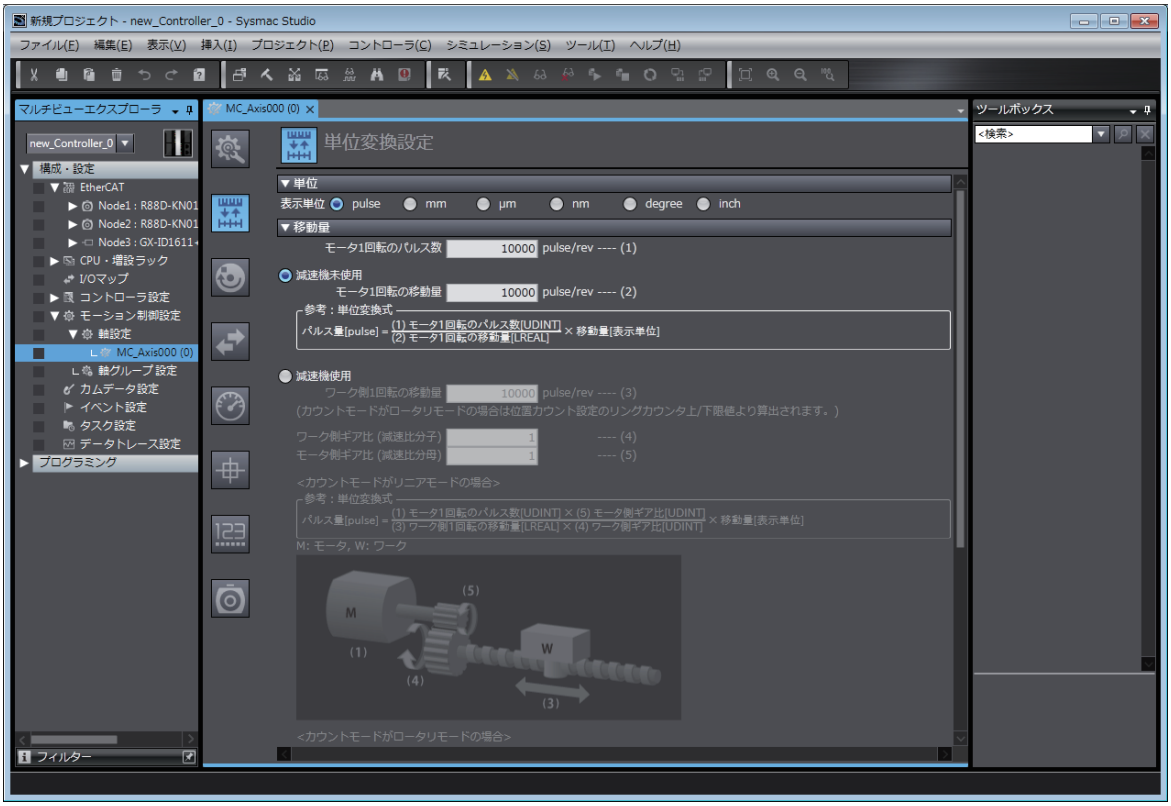
4 **选择要使用的伺服驱动器。**

通过该操作， EtherCAT 从站的伺服驱动器可作为轴使用。

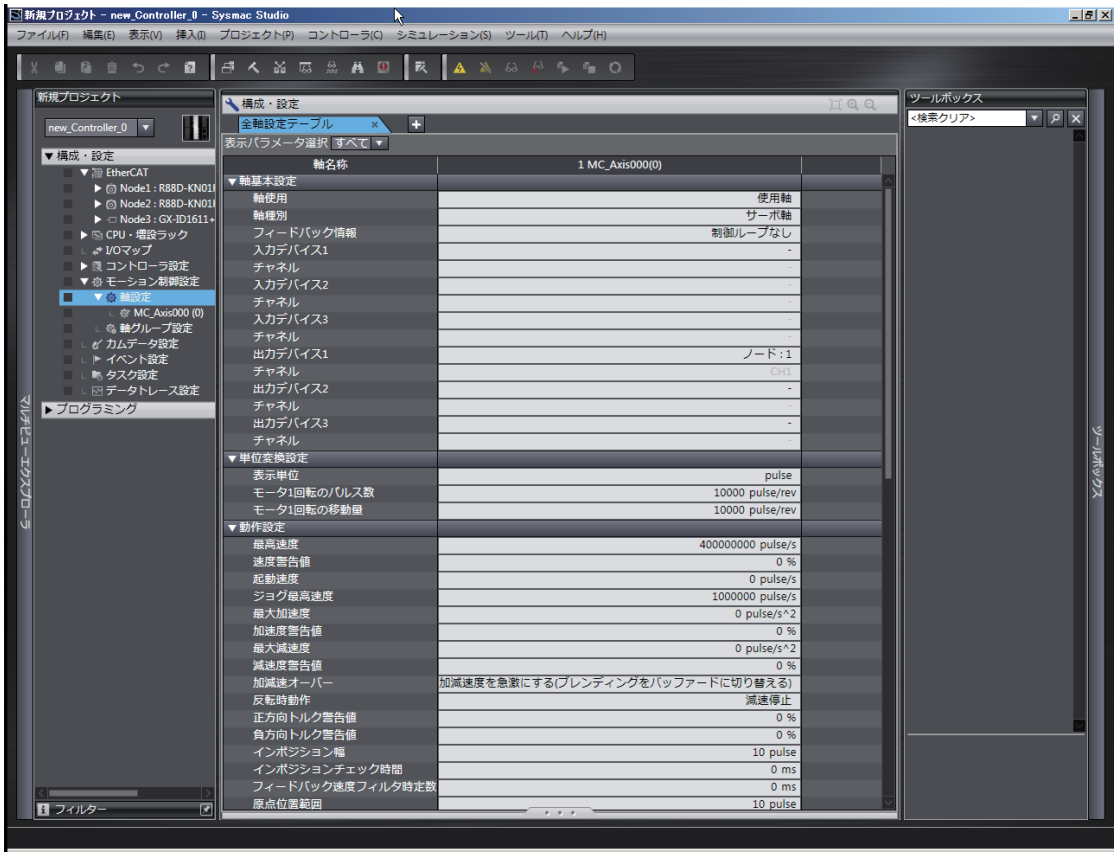


设定轴参数

单击轴参数设定画面中的图标。
显示各图标下分类的轴参数设定画面。



右击多视图浏览器中的［轴设定］，单击［轴设置表］，可在一览画面中设定所有轴的轴参数。



使用注意事项

显示单位、电子齿轮 （单位转换公式）及最高速度、最大加减速度的动作设定请根据装置的运行条件设定合适的值。

参考

变更用户程序中使用的轴变量名称

要变更已使用的轴变量名称时，请进行以下两项操作。

- 变更变量声明的变量 . 表中的轴变量名称。
- 变更用户程序的轴变量名称。

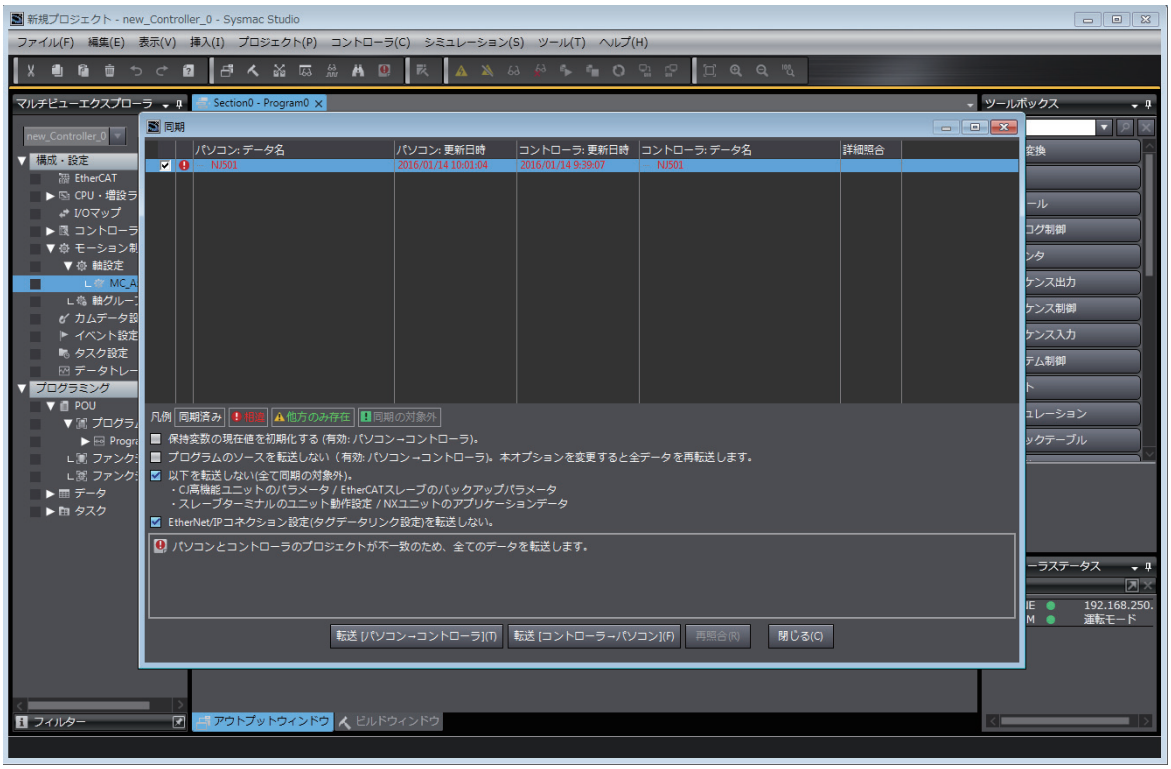
即使变更了变量表的轴变量名称，也不会反映到用户程序的轴变量名称中。
若将变量表中未声明的变量名称用于用户程序，将发生异常。请务必两者都变更。

下载到 CPU 单元

下载到 CPU 单元时，使用 Sysmac Studio 的 “ 同步 ” 功能。

1 **单击［控制器］菜单的［在线］，变为在线状态。**

2 **在［控制器］菜单中选择［同步］，单击［传送［计算机→控制器］］。**



参考

MC 功能模块的连接对象为欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型及 NX 系列脉冲输出单元。

伺服驱动器的连接对象机型

- 伺服驱动器的对象机型型号为 R88D-KN□□□-ECT 及 R88D-KN□□□-ECT-L。
- R88D-KN□□□-ECT-R 为仅支持位置控制 (cyclic synchronous position) 模式的伺服驱动器。因此，使用速度控制 (cyclic synchronous velocity) 模式或扭矩控制 (cyclic synchronous torque) 模式的功能无法使用。

伺服驱动器侧的设定

- MC 功能模块使用伺服驱动器侧的输入信号和部分功能。要正确使用 MC 功能模块的功能，需要在伺服驱动器侧进行信号配线并设定对象。

  关于设定内容，请参考 🕮「*A-1 与伺服驱动器连接* (P.A-2)」。

使用 NX 系列脉冲输出单元时的设定请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

3-3 轴组

下面介绍 MC 功能模块的轴组。

3-3-1 轴组的概要

直线插补、圆弧插补等，同时运行多个轴时，使用轴组。
因此，轴组由多个轴构成。
使用 Sysmac Studio 设定轴组变量后，可启动轴组的运动控制指令或获取轴组的状态。
在 MC 功能模块中，最多可处理 64 个组。
轴组的规格概要如下所示。

项目	规格	
	NJ 系列 CPU 单元	NX 系列 CPU 单元
轴组数	最多 32 组	最多 64 组
构成轴数	最多 4 轴 / 轴组	

MC 功能模块的轴组包含以下相关要素。

构成要素	内容	参照页
轴组参数	对 MC 功能模块控制的各轴组，设定最高插补速度、最大插补加减速度等轴组动作。 设定轴组参数时需要使用 Sysmac Studio。	P. 3-20
轴组变量	轴组变量是指由 MC 功能模块控制的各轴组的部分轴组参数、指令插补速度、异常信息等监视信息组成的系统定义变量。 若通过 Sysmac Studio 的多视图浏览器追加轴组，将创建轴组变量。轴组变量的名称 （以下称轴组变量名称）可在该视图中设定。	P. 3-21
在用户程序中指定轴组的方法	在用户程序中，使用运动控制指令进行运动控制。进行多轴协调控制的运动控制指令称为 “ 轴组指令 ”。 轴组指令对哪个轴组执行控制，需要在指令的输入输出变量 “AxesGroup（轴组） ” 中指定系统定义变量的轴组变量名称或 Sysmac Studio 中设定的轴组变量名称。	P. 3-22

3-3-2 轴组参数的概要

● 轴组参数一览

区分	参数名称
轴组基本设定	轴组编号
	运动控制 *1
	轴组使用
	轴构成
	构成轴
轴组动作设定	最高插补速度
	最大插补加速度
	最大插补减速度
	插补加减速超限
	插补速度警告值
	插补加速度警告值
	插补减速度警告值
	轴组停止方法
	中心点修正容许率

*1. NX 系列 CPU 单元时设定。

关于轴组参数的详情，请参考 *「5-3 轴组参数* (P.5-29)」。

● 使用轴组时必需的设定项目

对 Sysmac Studio 创建的轴组设定以下项目后，即可使用轴组。

区分	参数名称	设定内容	参照页
轴组基本设定	轴组编号	按轴组的创建顺序自动设定。	P. 5-30
	运动控制 *1	选择［主固定周期任务］。	
	轴组使用	选择［使用］。	
	轴构成	选择要控制的轴构成。	
	构成轴	选择分配到轴组的轴。	

*1. NX 系列 CPU 单元时设定。

使用注意事项

最高插补速度、停止方法等请根据运行条件设定合适的值。

3-3-3 轴组变量的概要

轴组变量是指由 MC 功能模块的各轴组设定信息、当前位置、异常信息等监视信息组成的系统定义变量。

若用 Sysmac Studio 创建轴组，将按照创建顺序将轴组变量登录到变量表中。轴组变量的数据类型为结构体型，数据类型名称为 “_sGROUP_REF”。

轴组变量名称

MC 功能模块的轴组变量名称中，包括系统定义变量的轴组变量名称和通过 Sysmac Studio 追加轴组时的轴组变量名称两种。

系统定义变量的轴组变量名称为 “_MC_GRP[0] ～ _MC_GRP[63]”、 “_MC1_GRP[0] ～ _MC1_GRP[63]”、 “_MC2_GRP[0] ～ _MC2_GRP[63]”。

若通过 Sysmac Studio 追加轴组， “_MC_GRP[0] ～ _MC_GRP[63]” 将设定为默认名称 “MC_Group000 ～ MC_Group063”。编号按追加的顺序从前往后排。这些轴组变量名称可在 Sysmac Studio 中按不同的轴组变更为任意变量名称。

在用户程序中，无论是系统定义变量的轴组变量名称，还是通过 Sysmac Studio 追加的轴组变量名称，均可使用。

● 使用 _MC_GRP[0-63] 时的示例

系统定义变量的轴组变量名称（全局变量视图的分配目标 *1）	通过 Sysmac Studio 追加的默认轴组变量名称	轴组编号示例
_MC_GRP[0]	MC_Group000	轴组 0
_MC_GRP[1]	MC_Group001	轴组 1
:	:	:
_MC_GRP[63]	MC_Group063	轴组 63

*1. 若变更 Sysmac Studio 的全局变量视图中的分配目标栏的名称，将发生异常。

参考

- NX 系列 CPU 单元时，可使用 “_MC_GRP[0-63]”、 “_MC1_GRP[0-63]” 和 “_MC2_GRP[0-63]”。

  NJ 系列 CPU 单元时，只能使用 “_MC_GRP[0-31]”。
- NX 系列 CPU 单元时，若 “_MC_GRP[0-63]” 和 “_MC1_GRP[0-63]” 的轴组编号相同，则参照相同的值。任一轴组变量均可使用，也可混合使用。
- 主固定周期任务中分配的轴组变量为 “_MC_GRP[0-63]” 或 “_MC1_GRP[0-63]”。

  固定周期任务 （执行优先度 5）中分配的轴组变量为 “_MC2_GRP[0-63]”。

轴组变量的层级和轴组变量名称变更示例

以 _MC_GRP[0] 为例说明。其他轴组变量亦同。

_MC_GRP[0]	: 轴组变量
_MC_GRP[0].Status	: 表示 “ 轴组状态 ” 的层级
:	
_MC_GRP[0].Cmd	: 表示 “ 轴组指令值 ” 的层级
_MC_GRP[0].Cmd.Vel	: 表示 “ 指令插补速度 ” 的变量
_MC_GRP[0].Cmd.AccDec	: 表示 “ 指令插补加减速度 ” 的变量
:	
_MC_GRP[0].Cfg	: 表示 “ 轴组基本设定 ” 的层级
_MC_GRP[0].Cfg.GrNo	: 表示 “ 轴组编号 ” 的变量
_MC_GRP[0].Cfg.GrEnable	: 表示 “ 轴组使用 ” 的变量
_MC_GRP[0].Kinematics	: 表示 “ 运动学转换设定 ” 的层级
_MC_GRP[0].Kinematics.GrType	: 表示 “ 轴构成 ” 的变量
_MC_GRP[0].Kinematics.Axis[0]	: 表示 “ 构成轴 （轴 A0） ” 轴编号的变量
:	
_MC_GRP[0].Kinematics.Axis[3]	: 表示 “ 构成轴 （轴 A3） ” 轴编号的变量
_MC_GRP[1]	: 轴组变量
:	

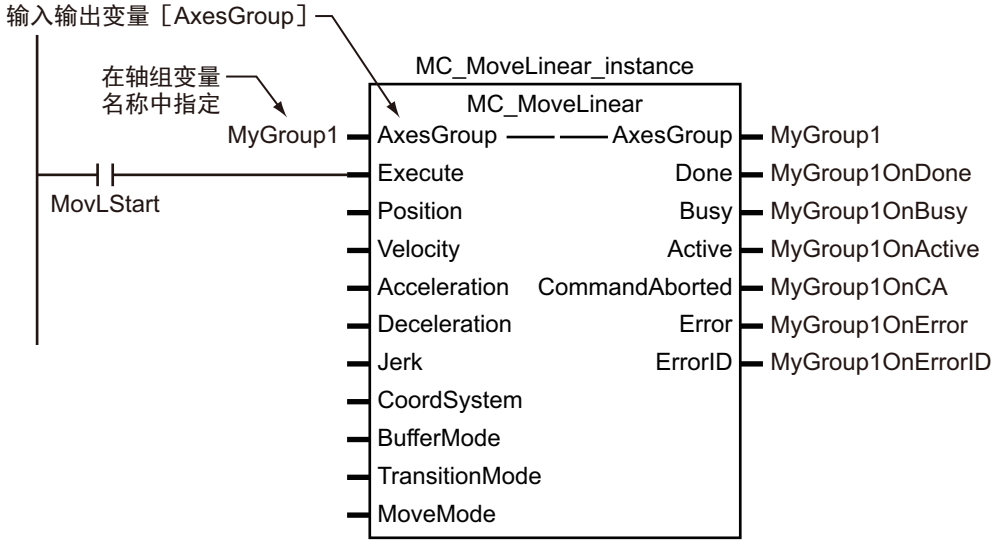
● **例**

将 “MC_Group000” 变更为 “MyGroup1” 后，表示 “ 指令插补速度 ” 的变量可使用 “MyGroup1.Cmd.Vel” 和 “_MC_GRP[0].Cmd.Vel” 中的任意一个。

关于轴组变量的详情，请参考 *「轴组变量* (P.6-27)」。

3-3-4 在用户程序中指定轴组的方法

在用户程序中，对运动控制指令的输入输出变量 “AxesGroup” 指定轴组变量名称。
下图示例表示针对系统定义变量的轴变量名称为 “_MC_GRP[0]” 的轴组，将通过 Sysmac Studio 追加时的轴组变量名称变更为 “MyGroup1”。



还可使用系统定义变量 “_MC_GRP[0]” 代替 “MyGroup1”。

关于运动控制指令的详情，请参考 🕮「*6-2 运动控制指令* (P.6-4)」。

各运动控制指令的详细功能请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的各指令说明。

3-4 轴组的设定步骤

下面介绍使用 Sysmac Studio 设定轴组的步骤。
不使用直线插补、圆弧插补等轴组指令时，不需要设定。

3-4-1 轴组的设定流程

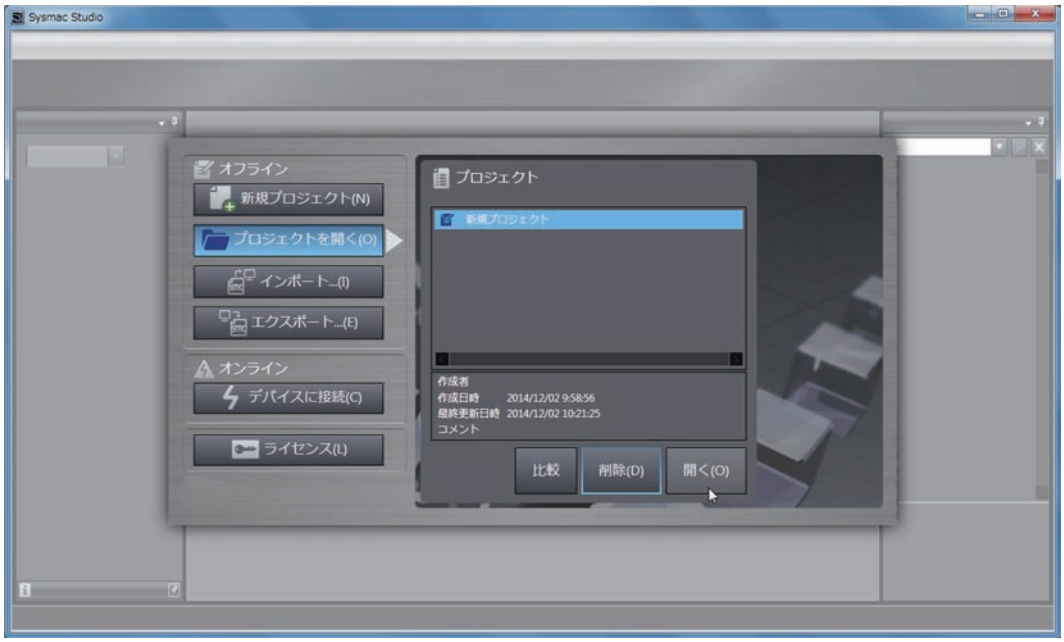


3-4-2 设定步骤

下面介绍在已追加轴的现有项目中设定轴组的方法。

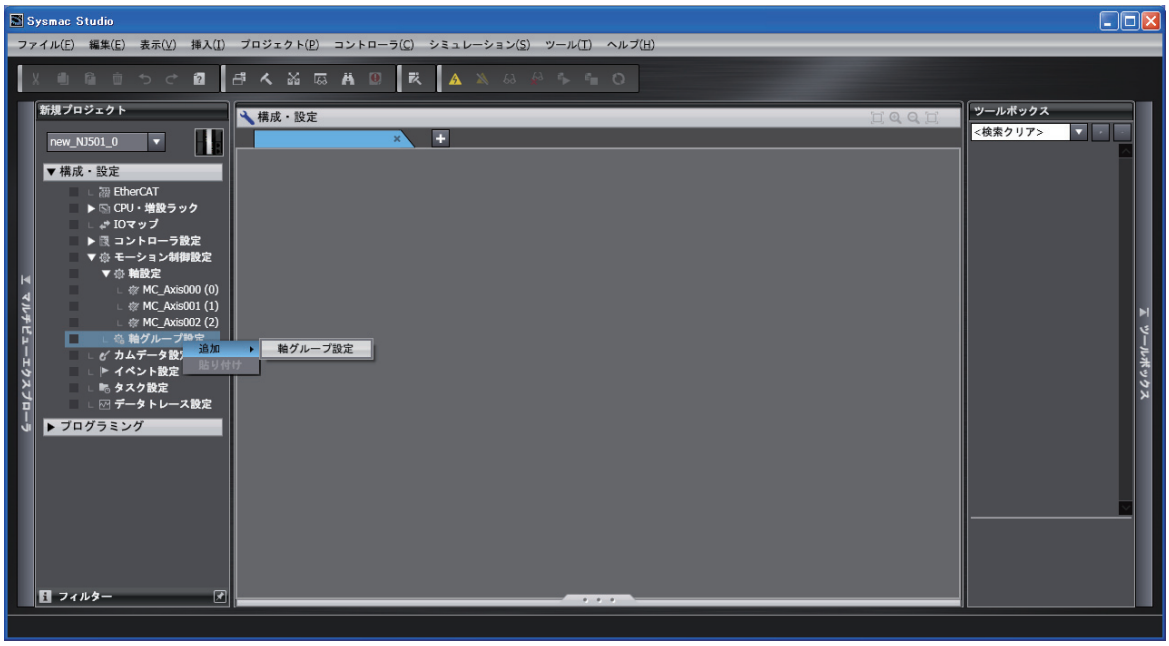
启动 Sysmac Studio

启动 Sysmac Studio，打开已有项目。



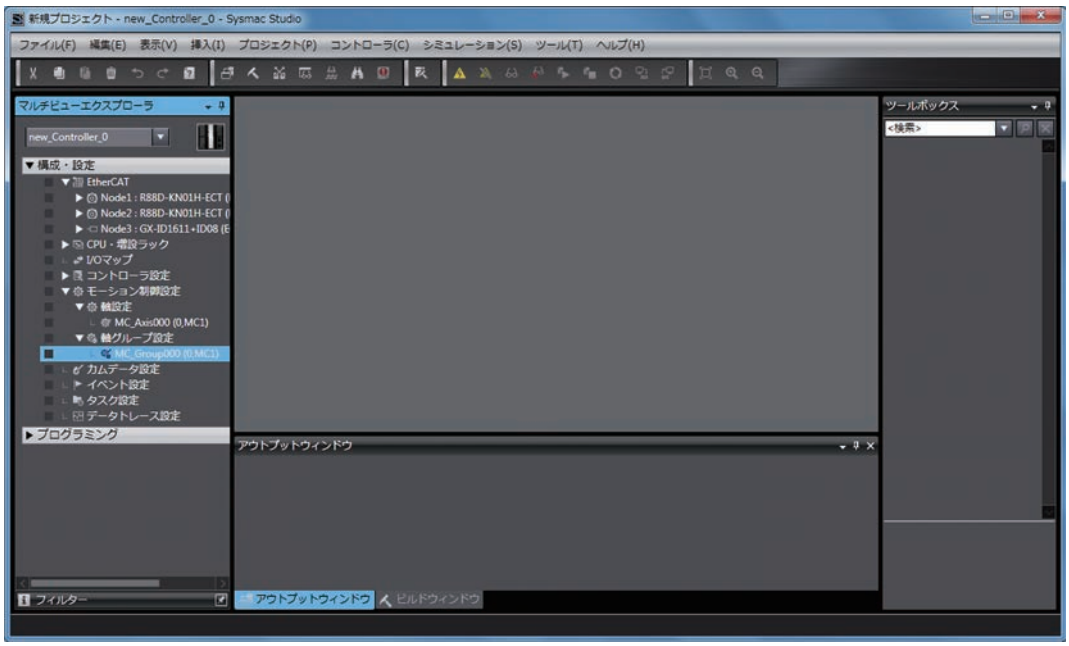
追加轴组

在多视图浏览器中右击［轴组设定］，单击［追加］中的［轴组设定］。



轴组将追加到多视图浏览器中。

所追加轴组的默认变量名称为 “MC_Group000”。

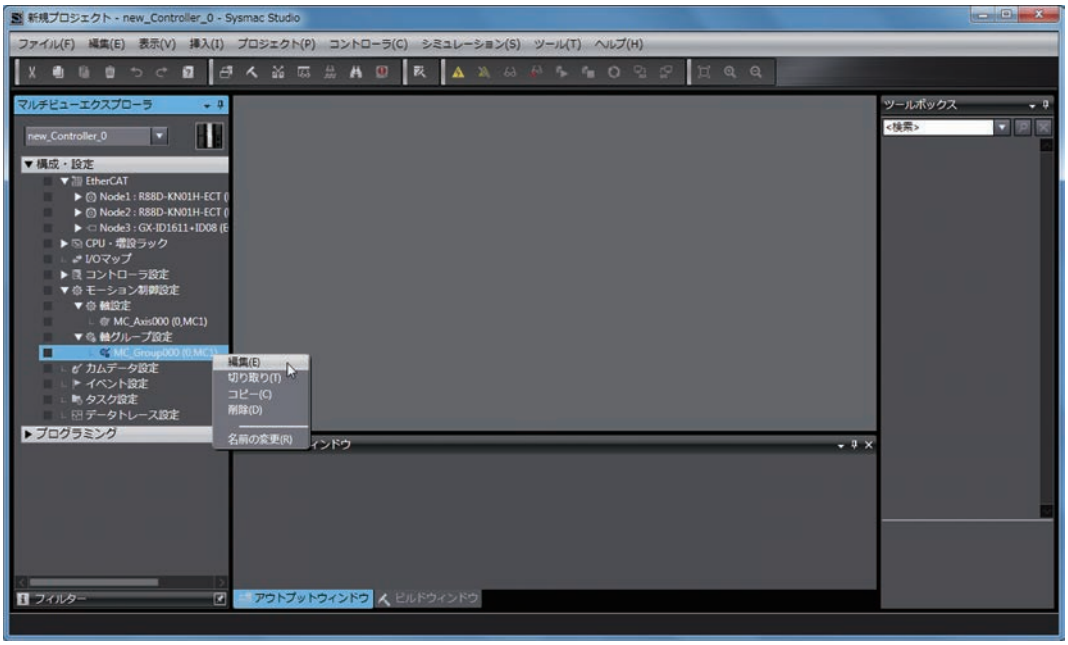


● 通过复制创建轴

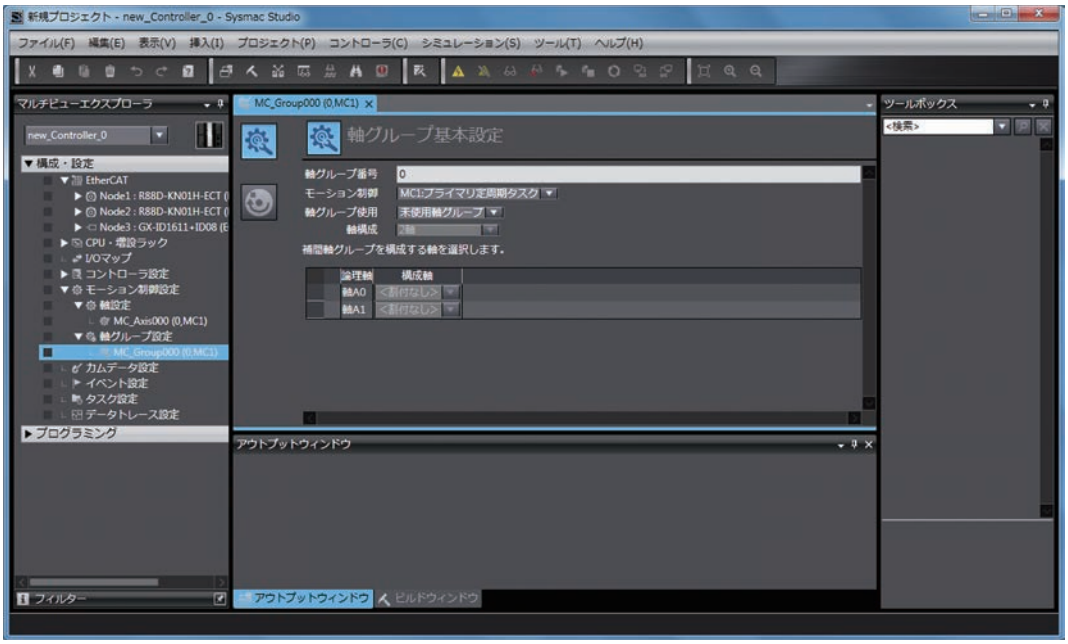
还可复制已有项目的轴组，创建新轴组。

设定轴组参数

1 **在多视图浏览器中右击追加的轴组，单击［编辑］。**



显示轴组参数设定画面的［轴组基本设定］画面。



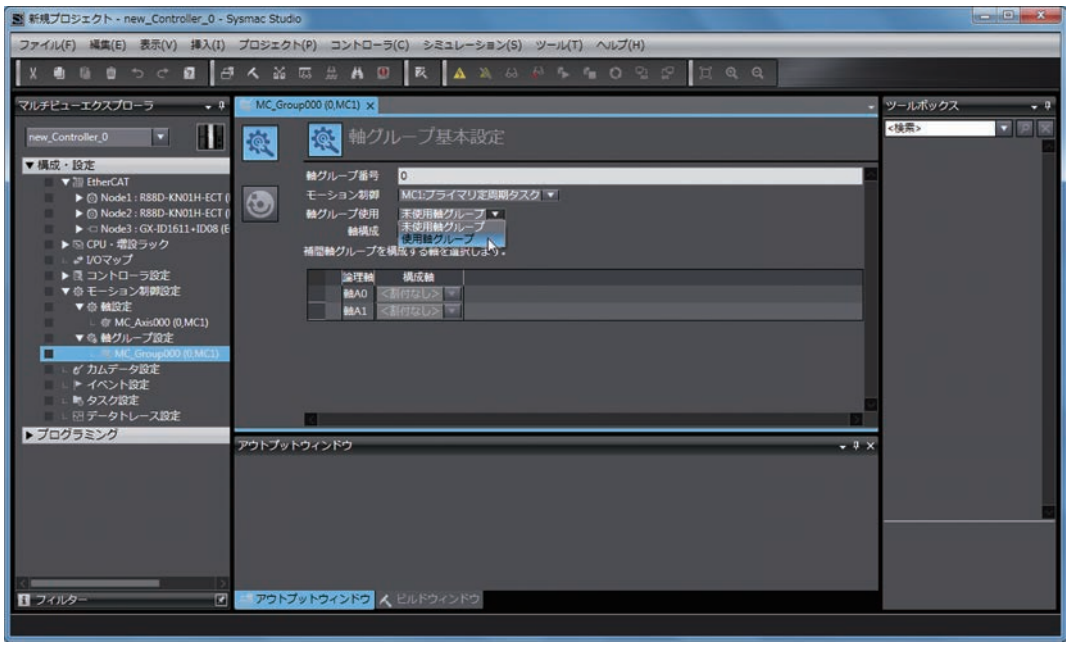
2 从［运动控制］中选择［主固定周期任务］或［固定周期任务（优先度 5）］。



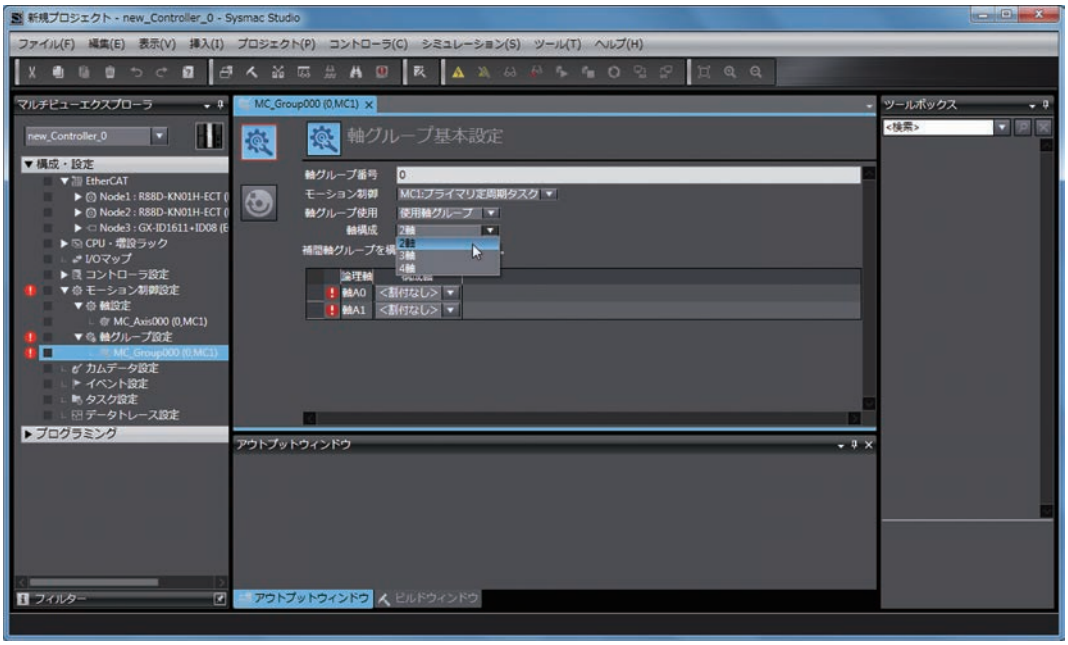
参考

NX 系列 CPU 单元时设定。 NJ 系列 CPU 单元中无需设定。

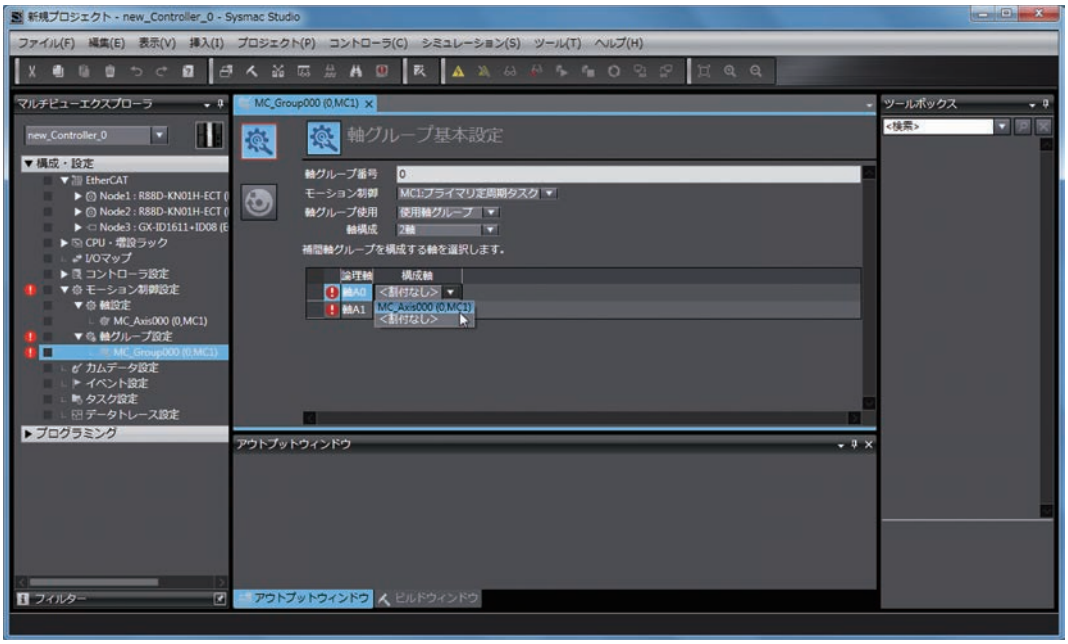
3 在［轴组使用］中选择［使用轴组］。



4 在［轴构成］中选择轴组的构成。以下为选择了 2 轴构成的情况。

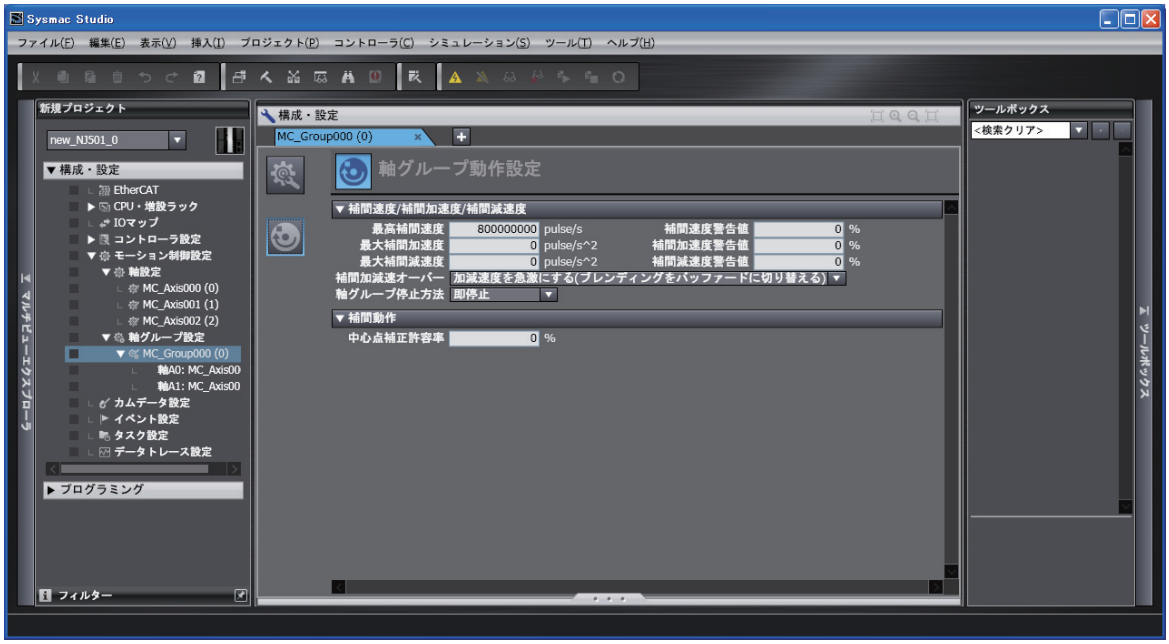


5 将要使用的轴分配到［逻辑轴］中。



6 **单击下侧的图标，显示［轴组动作设定］画面。**

各设定请根据装置的运行条件设定合适的值。



参考

变更用户程序中使用的轴组变量名称

要变更已使用的轴组变量名称时，请进行以下两项操作。

- 变更变量声明的变量表中的轴组变量名称。
- 变更用户程序的轴组变量名称。

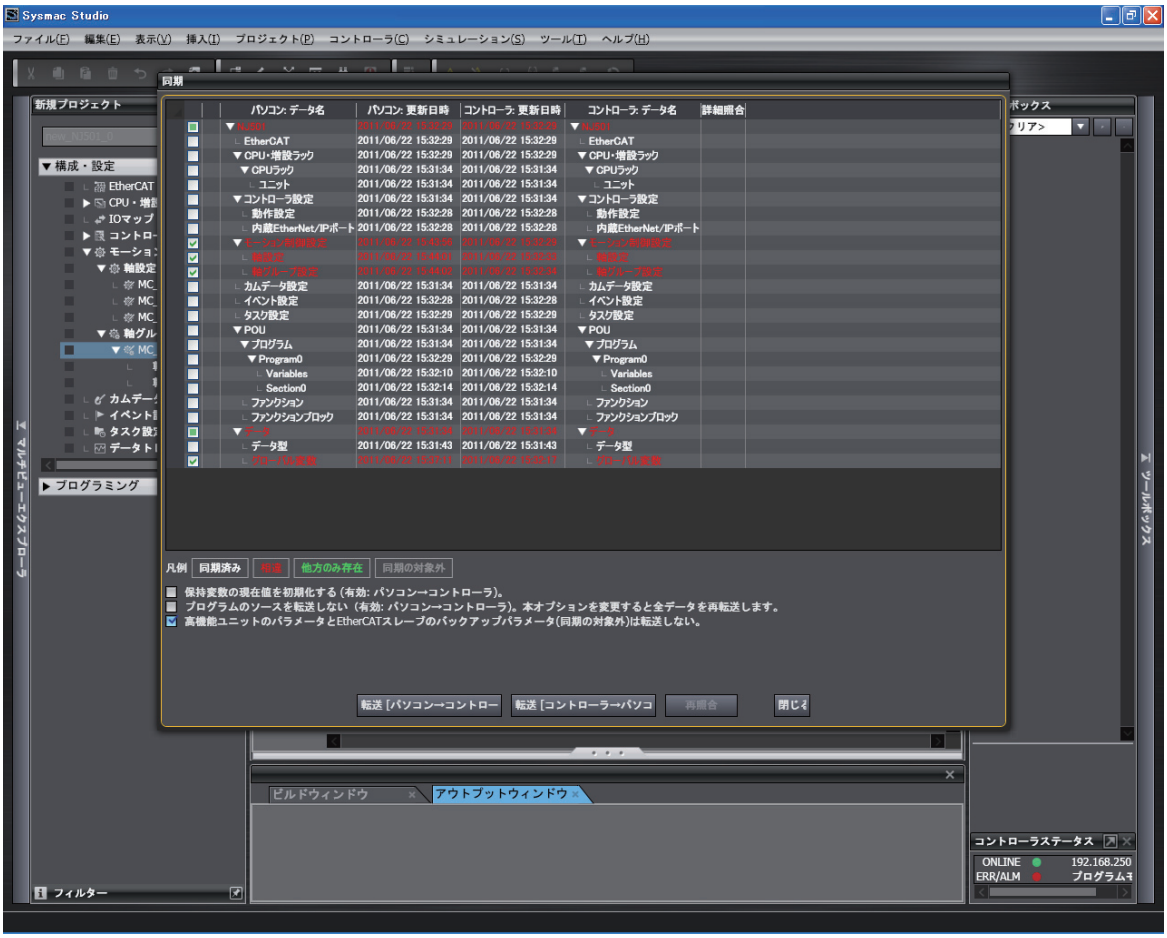
即使变更了变量表的轴组变量名称，也不会反映到用户程序的轴组变量名称中。
若将变量表中未声明的变量名称用于用户程序，将发生异常。请务必两者都变更。

下载到 CPU 单元

下载到 CPU 单元时，使用 Sysmac Studio 的 “ 同步 ” 功能。

1 **单击［控制器］菜单的［在线］，变为在线状态。**

2 **在［控制器］菜单中选择［同步］，单击［传送［计算机→控制器］］。**



4

通过 Sysmac Studio 确认配线

对 Sysmac Studio 的 MC 调试功能进行说明。利用 MC 调试功能，无需程序即可进行传感器信号的监视或电机配线的确认。

4-1 Sysmac Studio 的功能 4-2
4-1-1 MC 调试功能一览 4-2
4-1-2 使用步骤 4-3
4-1-3 轴参数的设定示例 4-4
4-1-4 MC 调试功能的启动步骤 4-5
4-2 传感器信号的监视 4-7
4-3 电机动作的确认 4-8
4-3-1 伺服 ON 4-8
4-3-2 点动进给 4-9
4-3-3 原点复位 4-9
4-3-4 绝对值定位 4-10
4-3-5 相对值定位 4-11

4-1 Sysmac Studio 的功能

下面介绍如何使用 MC 调试功能，进行配线确认和基本设定的确认。
利用 Sysmac Studio 的 MC 调试功能，无需程序即可确认配线。

4-1-1 MC 调试功能一览

MC 调试功能中包含以下功能。

分类	功能	说明	设定项目 / 监视项目
轴动作	减速停止	在 MC 调试过程中减速停止。	—
	伺服 ON/OFF	伺服的 ON/OFF。	—
	解除异常	解除 MC 功能模块中发生的异常。	—
	点动进给	向正方向或负方向进行点动进给。	目标速度 加速度 / 减速度
	绝对值定位	进行绝对值定位。*1	目标位置 目标速度 加速度 / 减速度 跃度
	相对值定位	进行相对值定位。	移动距离 目标速度 加速度 / 减速度 跃度
	原点复位	根据原点复位参数的设定进行原点复位。	原点复位参数
监视器	异常一览	监视 MC 功能模块中发生的异常。	MC 共通异常 轴异常 轴组异常
	轴状态	监视轴的状态。	轴启动准备完成 停止中 定位动作中 连续动作中 原点复位中 减速停止中 原点确定 原点停止
	当前位置监视器	监视当前位置。	当前位置 （指令 / 反馈）
	当前速度监视器	监视当前速度。	当前速度 （指令 / 反馈）
	伺服驱动器状态	监视伺服驱动器的状态。	伺服 ON/OFF 伺服就绪 主回路电源
	输入信号	监视输入信号的状态。	正方向极限输入 / 负方向极限输入 立即停止输入 近原点输入 原点输入 外部锁定输入 1 ～ 2

*1. 轴的计数模式为旋转模式时，向正方向的目标位置进行定位。
详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_MoveAbsolute （绝对值定位）指令。

（注）MC 调试功能的连接对象为欧姆龙生产的伺服驱动器 G5 系列及 NX 系列脉冲输出单元。请勿使用其他公司生产的驱动器。

4-1-2 使用步骤

要进行 MC 调试，请确认以下两点。

- Sysmac Studio 与控制器本体是否在线连接。
- 是否在其他 Sysmac Studio 上启动了 MC 调试。

确认后，请按照以下流程操作。



使用注意事项

- 若利用 Sysmac Studio 的指令执行伺服 ON、点动进给、相对值定位、绝对值定位、原点复位，电机将以设定的速度运行。
  请在充分确认电机运行不会造成危险后再执行。
- 通过 Sysmac Studio 操作时，请在外部设置紧急停止回路，确保电机能安全停止。计算机发生异常时，可能 Sysmac Studio 不会发出指令。
- 通过 Sysmac Studio 操作时，请进行 EtherCAT 通信的设定，待建立通信后再执行。
- 使用 NX 系列位置接口单元时的步骤请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

参考

通过以下操作，可随时结束 MC 调试功能。

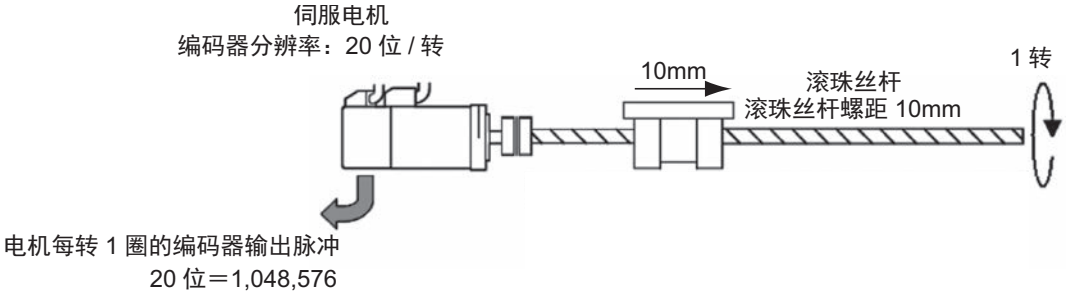
- 在 Sysmac Studio 的［控制器］菜单的［MC 调试］中选择［结束］。
- 右击 Sysmac Studio 的多视图浏览器的对称轴，选择［结束 MC 调试］。
- 关闭 Sysmac Studio 的 MC 调试画面。
- 结束 Sysmac Studio。

关于操作方法的详情，请参考 🕮《Sysmac Studio Version 1 操作手册 （SBCA-CN5-362）》。

4-1-3 轴参数的设定示例

使用 Sysmac Studio 的 MC 调试功能时，应设定以下轴参数。

下面以单轴装置为例介绍设定示例。



设定项目	设定内容
轴变量名称	Axis1 *1
轴编号	1 *2
轴使用	使用轴
轴种类	伺服轴
输入设备 / 输出设备	1 *3
显示单位	μm
电机每转 1 圈的脉冲数	1,048,576 *4
电机每转 1 圈的移动量	10,000 *4
最高速度	500,000 *5
JOG 最高速度	50,000 *6
最大加速度	5,000,000 *7
最大减速度	5,000,000 *7
软件限位功能	对指令位置有效，使其立即停止。
正方向软件限位	500,000 *8
负方向软件限位	0 *8
计数模式	直线模式

*1. 有多个轴时，对不同的轴设定不同的变量名称。
*2. 有多个轴时，对不同的轴设定不同的值。
*3. 设定为与伺服驱动器的节点地址相同的值。有多个轴时，对不同的轴设定不同的值。
*4. 位置的指令单位为 1(μm)。
*5. 最高速度为 3,000r/min = 30m/min = 0.5m/s = 500,000μm/s。
*6. JOG 最高速度为最高速度的 10%， 0.05m/s = 50,000μm/s。
*7. 最大加速度及最大减速度为 5m/s^2。加速至最高速度 3,000r/min 的时间为 0.1s。
*8. 设定装置可移动范围的内侧位置。正方向软件限位的设定值为 50cm ＝ 500,000μm。

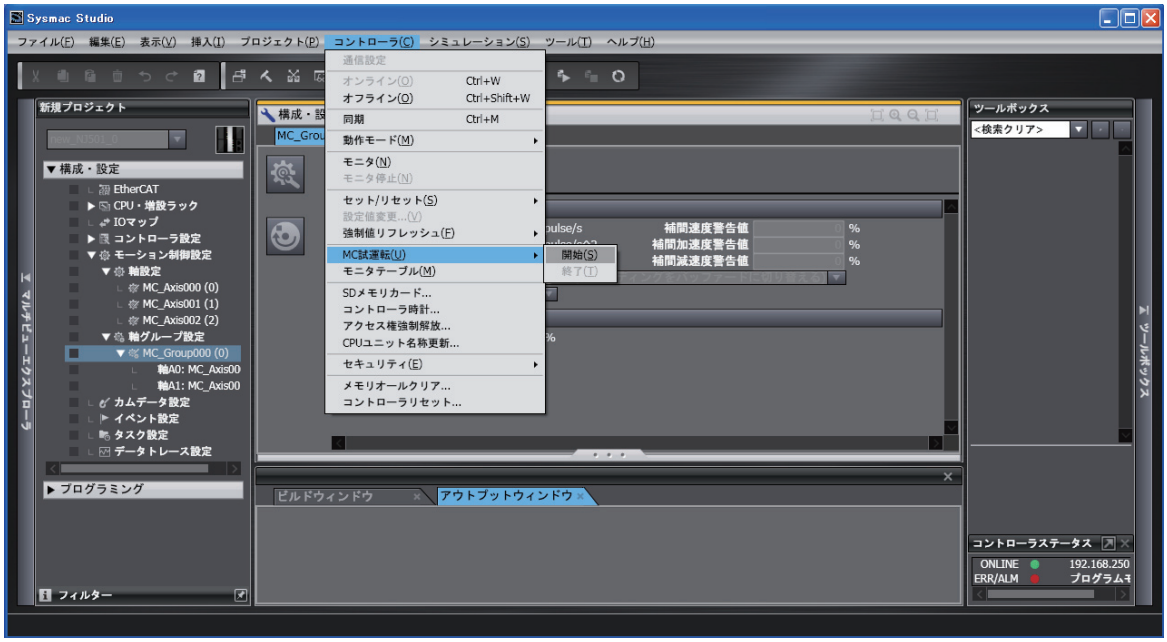
4-1-4　MC 调试功能的启动步骤

MC 调试功能通过 Sysmac Studio 启动。

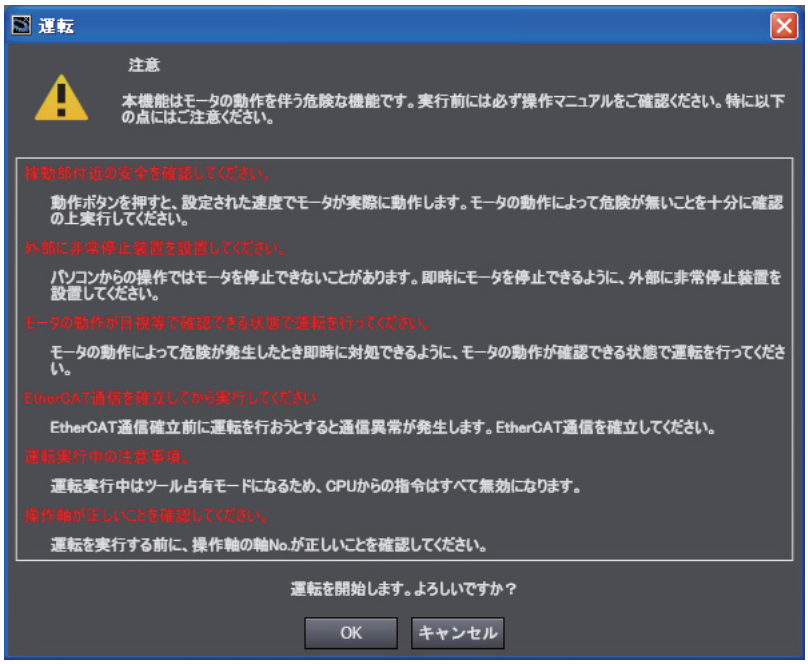
1　**启动 Sysmac Studio，打开已完成伺服轴设定的已有项目。**

2　**单击［控制器］菜单的［在线］，变为在线状态。**

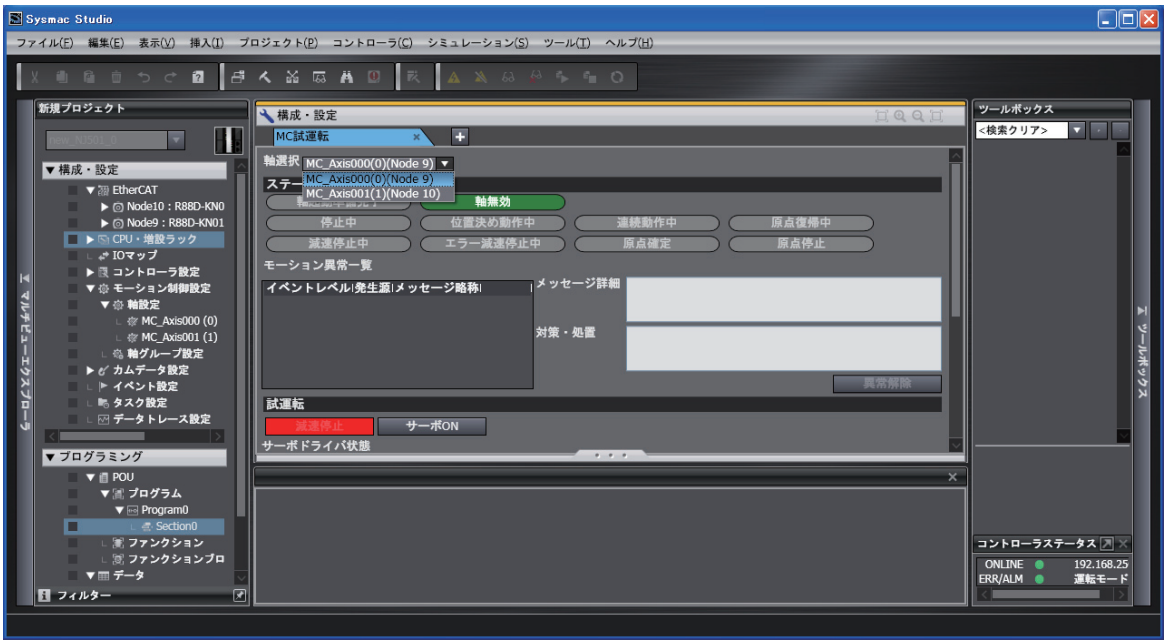
3　**在［控制器］菜单的［MC 调试］中单击［开始］。**



显示以下注意对话框，请仔细阅读内容，确认安全后，单击 [OK]。



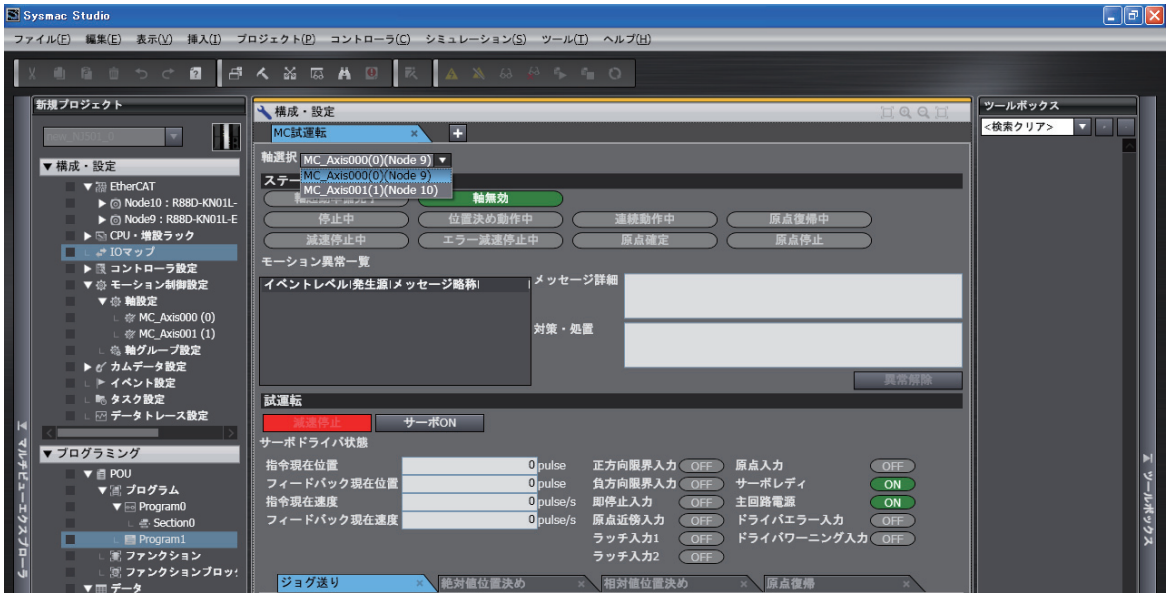
显示 MC 调试画面。


Sysmac Studio
ファイル(F)　編集(E)　表示(V)　挿入(I)　プロジェクト(P)　コントローラ(C)　シミュレーション(S)　ツール(T)　ヘルプ(H)
新規プロジェクト
構成・設定
EtherCAT
Node10：R88D-KN0
Node9：R88D-KN01
CPU・増設ラック
IOマップ
コントローラ設定
モーション制御設定
軸設定
MC_Axis000 (0)
MC_Axis001 (1)
軸グループ設定
カムデータ設定
イベント設定
タスク設定
データトレース設定
プログラミング
POU
プログラム
Program0
Section0
ファンクション
ファンクションブロ
データ
フィルター
構成・設定
MC試運転
軸選択
MC_Axis000(0)(Node 9)
MC_Axis000(0)(Node 9)
MC_Axis001(1)(Node 10)
軸無効
停止中
位置決め動作中
連続動作中
原点復帰中
減速停止中
エラー減速停止中
原点確定
原点停止
モーション異常一覧
イベントレベル
発生源
メッセージ略称
メッセージ詳細
対策・処置
異常解除
試運転
減速停止
サーボON
サーボドライバ状態
ツールボックス
<検索クリア>
コントローラステータス
ONLINE
192.168.25
ERR/ALM
運転モード

4-2　传感器信号的监视

确认传感器信号的配线时，使用各输入信号的显示。

1　**选择要在 MC 调试画面中确认的轴。**



2　**打开 / 关闭各输入信号连接的传感器，在监视画面中确认相应的信号是否正确 ON/OFF。**

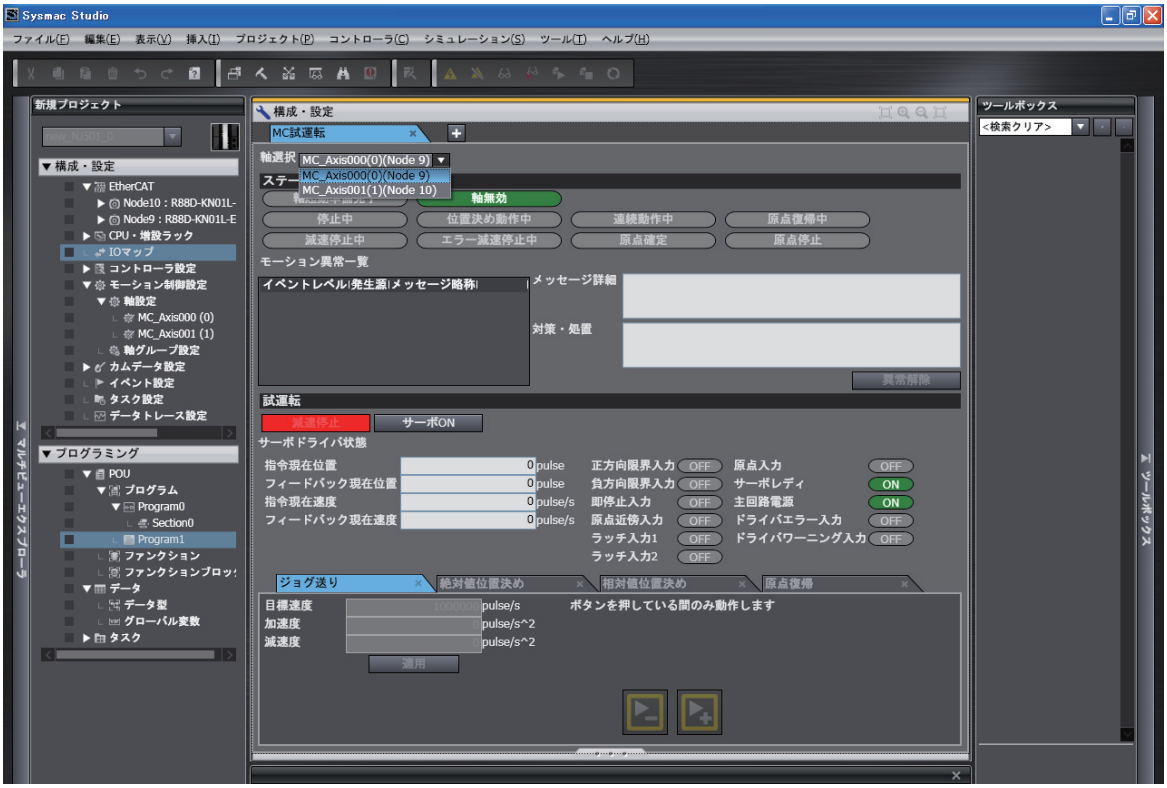
4-3 电机动作的确认

确认电机动作时，使用 MC 调试的各功能。

4-3-1 伺服 ON

按下［伺服 ON］按钮，可切换伺服 ON 和伺服 OFF。

1 **选择要打开伺服的轴。**



2 **按下［伺服 ON］按钮，变为伺服 ON 状态。**

3 **若按下［伺服 OFF］按钮，将恢复为伺服 OFF 状态。**

使用注意事项

- 若利用 Sysmac Studio 的指令执行伺服 ON、点动进给、相对值定位、绝对值定位、原点复位，电机将以设定的速度运行。
  请在充分确认电机运行不会造成危险后再执行。
- 通过 Sysmac Studio 操作时，请在外部设置紧急停止回路，确保电机能安全停止。计算机发生异常时，可能 Sysmac Studio 不会发出指令。
- 通过 Sysmac Studio 操作时，请进行 EtherCAT 通信的设定，待建立通信后再执行。
- 使用 NX 系列脉冲输出单元时，需要单独执行电机驱动器的通电 ON/OFF。详情请参考 《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

4-3-2　点动进给

1 **在 MC 调试画面的［点动进给］中选择要点动进给的轴。**

2 **按下［伺服 ON］按钮，变为伺服 ON 状态。**

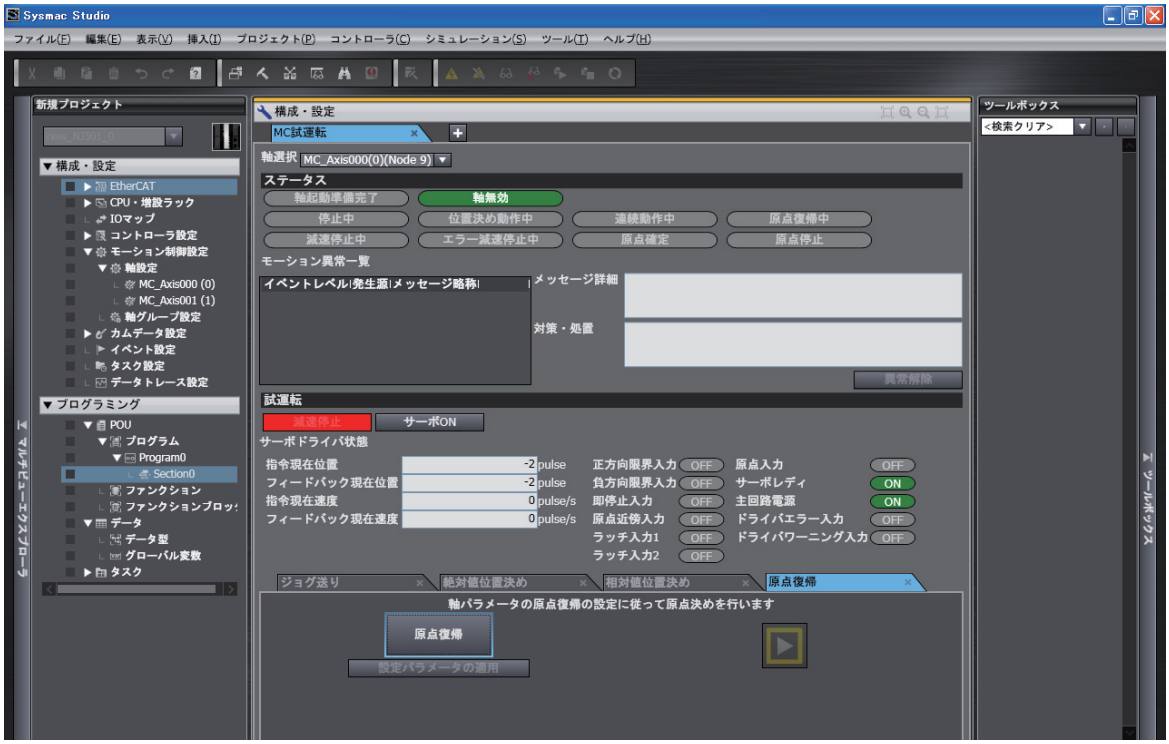
3 **输入目标速度、加速度、减速度的值，按［应用］按钮。**

4 **按 按钮或 按钮。**
按住按钮期间，向正方向或负方向动作。
请确认是否向设定的方向动作。

4-3-3　原点复位

1 **事先在轴参数设定画面的［原点复位设定］中设定原点复位参数。**

2 **选择调试画面的［原点复位］选项卡。**
显示以下画面。



3 **选择要进行原点复位的轴。**

4 **按下［伺服 ON］按钮，变为伺服 ON 状态。**

5 **按下［应用设定参数］按钮。**

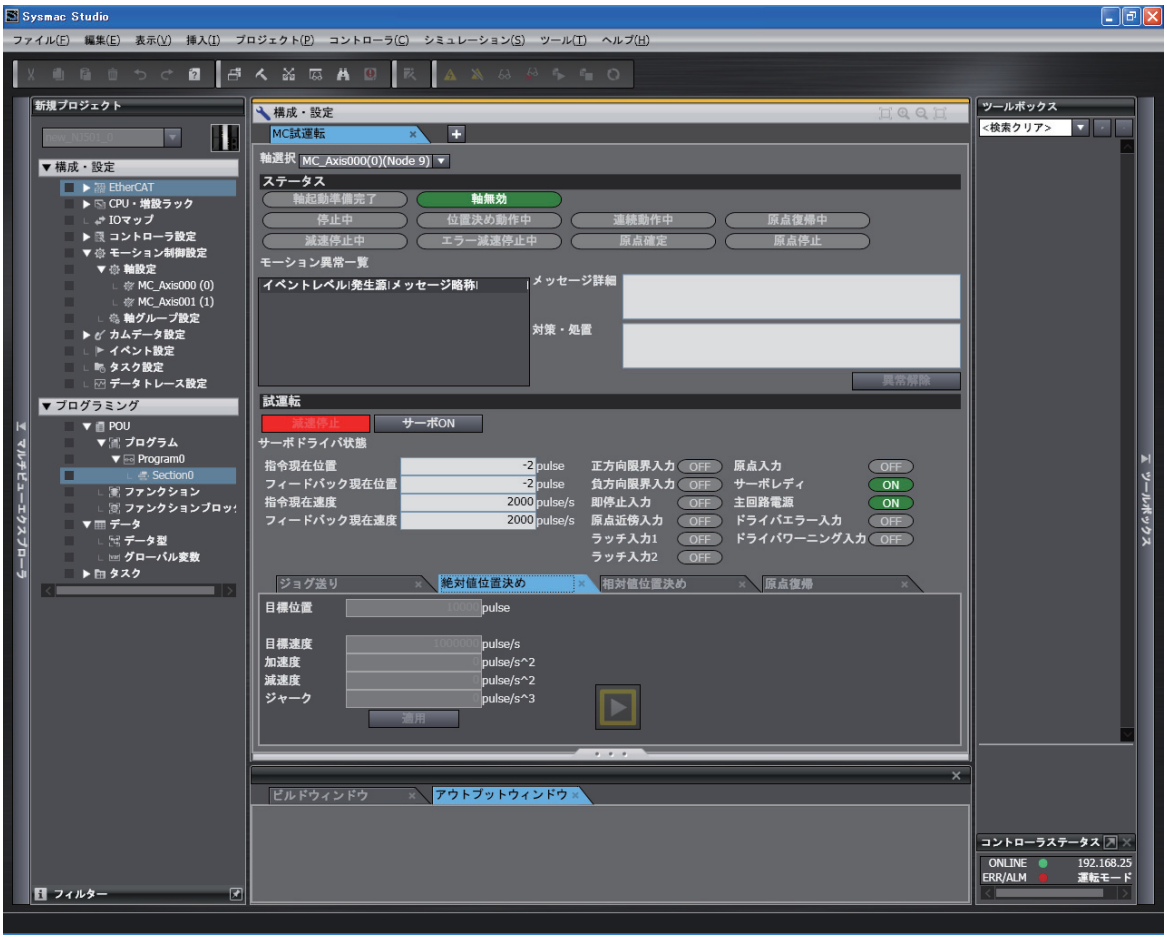
6 **按下按钮。**
请确认原点复位动作是否按设定执行。

参考

- 若按下［原点复位设定］按钮，将显示［轴参数设定］的［原点复位设定］画面，因此请设定［原点复位参数］。
- 事先设定［原点复位参数］后，若按下［应用设定参数］，将应用该设定

4-3-4 绝对值定位

1 **在 MC 调试画面中选择［绝对值定位］选项卡。**
显示以下画面。



2 **选择要进行绝对值定位的轴。**

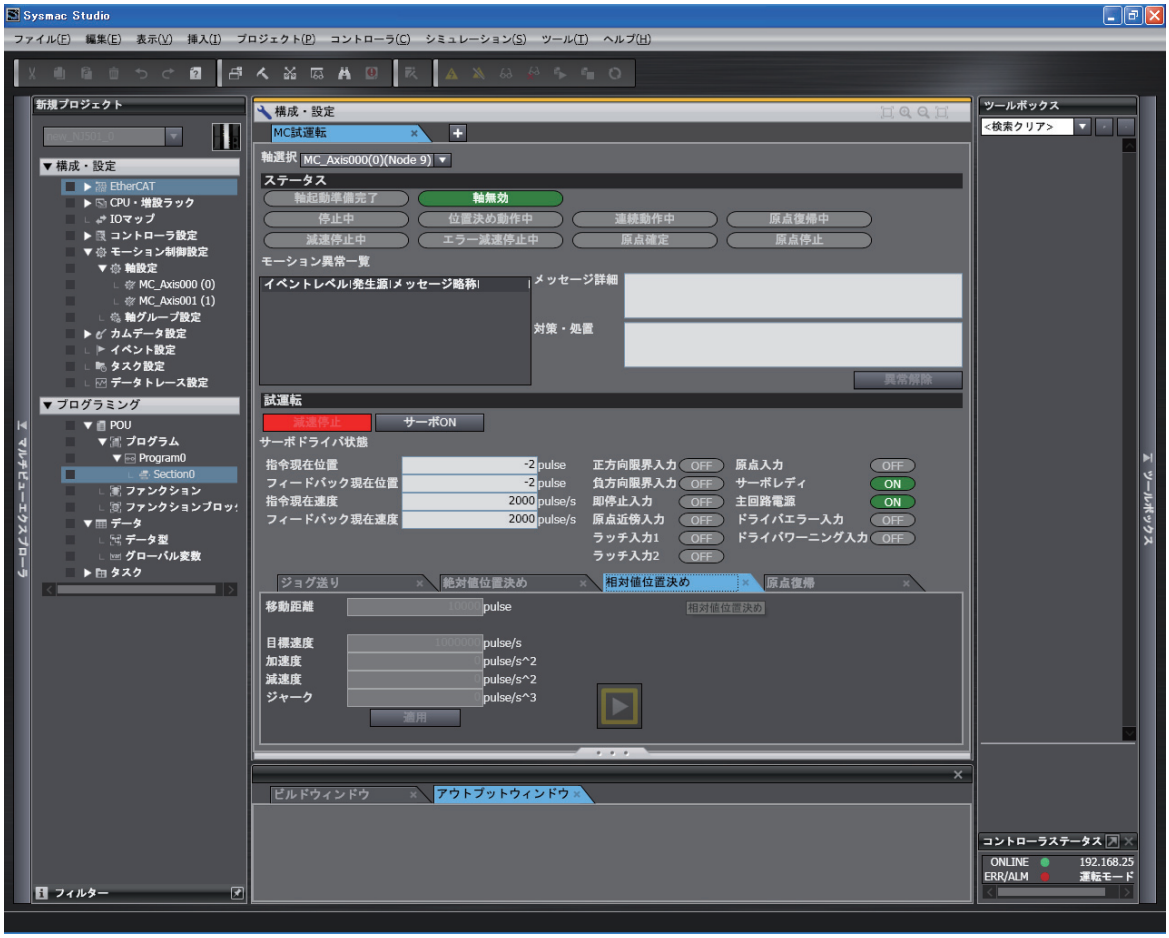
3 **按下［伺服 ON］按钮，变为伺服 ON 状态。**

4 **输入目标位置、目标速度、加速度、减速度、跃度的值，按［应用］按钮。**

5 **按下按钮后，开始绝对值定位。**
请确认完成定位的位置是否与设定一致。

4-3-5　相对值定位

1　**在 MC 调试画面中选择［相对值定位］选项卡。**
显示以下画面。



2　**选择要进行相对值定位的轴。**

3　**按下［伺服 ON］按钮，变为伺服 ON 状态。**

4　**输入目标移动距离、目标速度、加速度、减速度、跃度的值，按［应用］按钮。**

5　**按下按钮后，开始相对值定位。**
请确认移动距离是否与设定一致。

5

运动控制参数

本章介绍运动控制中使用的轴参数和轴组参数。

5-1 概要 5-2

5-2 轴参数 5-4

5-2-1 轴参数一览 5-4
5-2-2 轴基本设定 5-6
5-2-3 单位转换设定 5-10
5-2-4 动作设定 5-17
5-2-5 扩展动作设定 5-20
5-2-6 限位设定 5-21
5-2-7 位置计数设定 5-22
5-2-8 伺服驱动器设定 5-24
5-2-9 原点复位设定 5-25
5-2-10 轴参数的设定示例 5-27

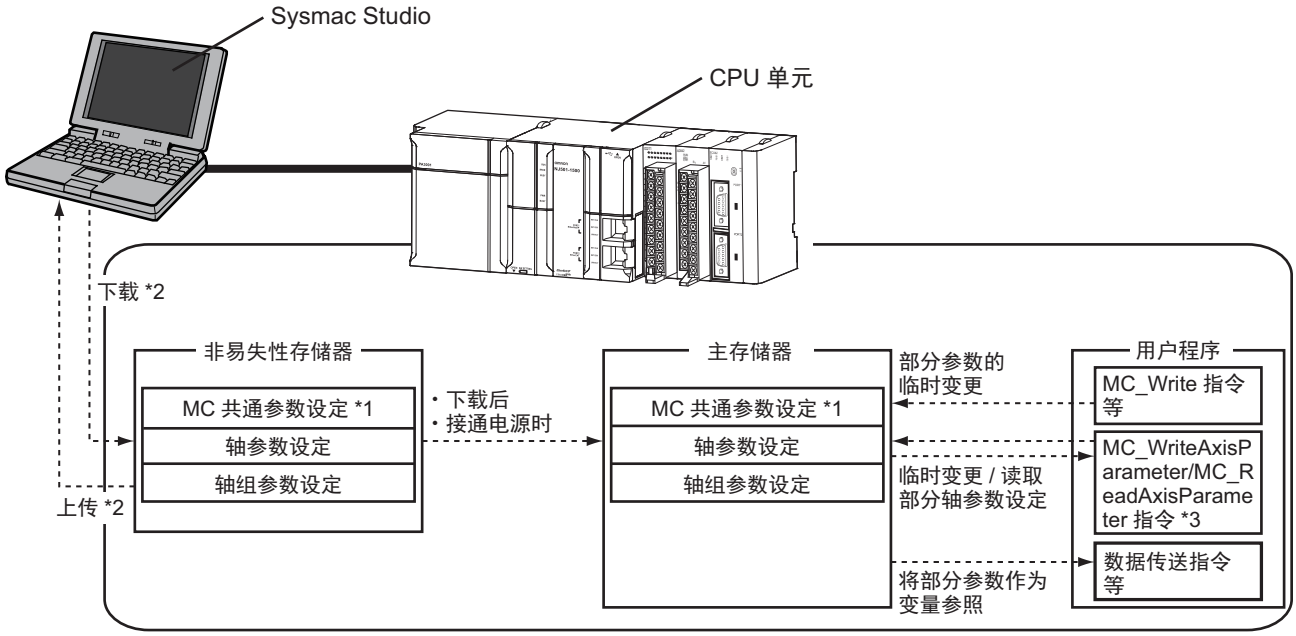
5-3 轴组参数 5-29

5-3-1 轴组参数一览 5-29
5-3-2 轴组基本设定 5-30
5-3-3 轴组动作设定 5-32
5-3-4 轴组的有效 5-33

5-1 概要

NJ/NX 系列 CPU 单元的 MC 功能模块中，可利用运动控制指令执行单轴动作和轴组的多轴动作。
轴参数和轴组参数是用于设定这些动作的参数。
轴参数始终需要设定，但不使用轴组多轴动作时，不需要设定轴组参数。

这些参数称为运动控制参数设定 （简称：MC 参数设定）。



*1 在当前版本的 MC 功能模块中，没有 “MC 共通参数设定 ” 这个设定项目。

*2 “ 下载 ” 和 “ 上传 ” 时使用 Sysmac Studio 的 “ 同步 ” 功能。

*3 MC_WriteAxisParameter （轴参数写入）指令和 MC_ReadAxisParameter （轴参数读取）指令可在 CPU 单元版本 1.08 以上和 Sysmac Studio Ver.1.09 以上的组合中使用。

MC 参数设定的数据流程

- 若将 Sysmac Studio 中设定的 MC 参数设定下载到 CPU 单元，将保存到 CPU 单元内置的非易失性存储器中。此外，若向 Sysmac Studio 上传 MC 参数设定，将上传非易失性存储器中保存的设定。
- 非易失性存储器中保存的设定会在下载后或接通电源时反映到主存储器中。
- 若设定内容正确， MC 功能模块将按照主存储器的设定内容执行控制。
- 部分参数可作为运动控制系统变量，参照设定内容。
- MC 参数设定的上传及下载与 CPU 单元的模式或 MC 功能模块的状态无关，均可执行。
- 开始下载后，动作中的轴将立即停止，伺服变为 OFF。

通过指令替换 MC 参数设定

- 运动控制指令的 MC_Write （MC 设定写入）指令、 MC_ChangeAxesInGroup （轴组构成轴写入）指令及 MC_WriteAxisParameter （轴参数写入）指令，可在用户程序的执行过程中变更主存储器中部分 MC 参数设定的内容。
- 若指定的设定值超出范围，指令的输出变量 “Error （错误）” 将变为 TRUE， MC 参数设定不会变更。
- 变更后的 MC 参数设定内容生效的时间为以下任一情况。
  a) 在轴或轴组停止的状态下，启动了轴指令或轴组指令。
  b) 将指令的缓冲模式选择设定为中断，并执行了多重启动时。
- MC_Write （MC 设定写入）指令、 MC_ChangeAxesInGroup （轴组构成轴写入）指令及 MC_WriteAxisParameter （轴参数写入）指令的详情请参考 《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。

使用注意事项

- 通过 MC_Write （MC 设定写入）等指令变更的 MC 参数设定将保存到 CPU 单元的主存储器中。不会保存到 CPU 单元内置的非易失性存储器中。因此，重新接通电源或从 Sysmac Studio 下载后，将恢复为非易失性存储器的参数设定内容。此外，不可通过 Sysmac Studio 上传这些数据。
  需要保存到非易失性存储器时，请通过 Sysmac Studio 变更参数设定，再下载到 CPU 单元。
- 可变更 MC 参数设定的指令如下所示。
  MC_Write （MC 设定写入）指令
  MC_ChangeAxesInGroup （轴组构成轴写入）指令
  MC_ChangeAxisUse （轴使用变更）指令
  MC_WriteAxisParameter （轴参数写入）指令
- 使用 NX 系列位置接口单元时请参考 《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

下面介绍通过 Sysmac Studio 设定的参数设定的内容。

5-2 轴参数

在轴参数中，对 MC 功能模块控制的轴，设定最高速度、跃度、原点复位等轴动作。
轴参数的数量与各型号的最大控制轴数对应。例如， NJ501-13□□ 为 16 轴、 NJ501-15□□ 为 64 轴、NX701-16□□ 为 128 轴、 NX701-17□□ 为 256 轴。
此外，各轴的参数设定相同，所以这里只介绍 1 个轴的设定。

5-2-1 轴参数一览

轴参数使用 Sysmac Studio 对各轴分别进行设定。

区分	参数名称	临时变更 *1		变量读取 *2	参照页
		可否	使用指令		
轴基本设定	轴编号			可	P. 5-6
	运动控制 *3			可 *4	
	轴使用	可 *5	MC_ChangeAxisUse	可	
	轴种类			可	
	输入设备 / 输出设备			可	
单位转换设定	显示单位	可 *6	MC_WriteAxisParameter *7	可	P. 5-10
	电机每转 1 圈的脉冲数			可	
	电机每转 1 圈的移动量			可	
	减速机使用 *8				
	工件侧每转 1 圈的移动量 *8				
	工件侧齿轮比 *8				
	电机侧齿轮比 *8				
动作设定	最高速度	可 *6	MC_WriteAxisParameter *7		P. 5-17
	启动速度 *9				
	点动最高速度				
	最大加速度				
	最大减速度				
	加减速超限				
	反转时动作				
	速度警告值	可	MC_Write MC_WriteAxisParameter *7		
	加速度警告值				
	减速度警告值				
	正方向扭矩警告值				
	负方向扭矩警告值				
	位置宽度	可 *10			
	位置检查时间	可			
	反馈速度过滤器时间常数	可 *6	MC_WriteAxisParameter *7		
	原点位置范围				
扩展动作设定	立即停止输入停止方法	可 *6	MC_WriteAxisParameter *7		P. 5-20
	极限输入停止方法				
	驱动器错误重置监视时间				
	正方向扭矩限制上限值				
	负方向扭矩限制上限值				
	立即停止输入逻辑反转 *9				
	正方向极限输入逻辑反转 *9				

区分	参数名称	临时变更 *1		变量读取 *2	参照页
		可否	使用指令		
扩展动作设定	负方向极限输入逻辑反转 *9	可 *6	MC_WriteAxisParameter *7		P. 5-20
	近原点输入逻辑反转 *9				
限位设定	软件限位功能	可	MC_Write MC_WriteAxisParameter *7		P. 5-21
	正方向软件限位				
	负方向软件限位				
	位置偏差超限值				
	位置偏差警告值				
位置计数设定	计数模式	可 *6	MC_WriteAxisParameter *7		P. 5-22
	循环计数器上限设定值			可 *4	
	循环计数器下限设定值			可 *4	
	编码器种类				
伺服驱动器设定	循环计数器上限设定值				P. 5-24
	循环计数器下限设定值				
	PDS 状态控制方式 *4				
原点复位设定	原点复位动作	可 *6	MC_WriteAxisParameter *7		P. 5-25
	原点输入信号				
	原点复位开始方向				
	原点检测方向				
	正方向极限输入时动作				
	负方向极限输入时动作				
	原点复位速度				
	原点复位接近速度				
	原点复位加速度				
	原点复位减速度				
	原点复位跃度				
	原点输入掩码距离				
	原点位置偏置				
	原点复位推压时间				
	原点复位修正值				
	原点复位修正速度				

*1. 表示是否可通过指令临时变更。
*2. 表示是否可从用户程序中读取运动控制系统变量。
*3. NX 系列 CPU 单元时设定。
*4. 可在 CPU 单元版本 1.10 以上和 Sysmac Studio Ver.1.12 以上的组合中使用。
*5. 在 CPU 单元版本 1.04 以上和 Sysmac Studio Ver.1.05 以上组合中可临时变更。
*6. 在 CPU 单元版本 1.08 以上和 Sysmac Studio Ver.1.09 以上组合中可临时变更。
*7. MC_WriteAxisParameter 指令可在 CPU 单元版本 1.08 以上和 Sysmac Studio Ver.1.09 以上的组合中使用。
可通过 MC_WriteAxisParameter 指令临时变更的参数可通过 MC_ReadAxisParameter 指令读取。
*8. 可在 CPU 单元版本 1.11 以上和 Sysmac Studio Ver.1.15 以上的组合中使用。
*9. 可在 CPU 单元版本 1.05 以上和 Sysmac Studio Ver.1.06 以上的组合中使用。
*10.在 CPU 单元版本 1.01 以上和 Sysmac Studio Ver.1.02 以上组合中可通过 MC_Write 指令临时变更。
在 CPU 单元版本 1.08 以上和 Sysmac Studio Ver.1.09 以上组合中可通过 MC_WriteAxisParameter 指令临时变更。

关于轴参数的设定方法，请参考 *「3-2 轴的设定步骤* (P.3-9)」。
MC_Write （MC 设定写入）指令等各指令的详情请参考 《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。
关于运动控制系统变量，请参考 *「6-6 运动控制系统变量* (P.6-16)」。

5-2-2 轴基本设定

设定轴的使用 / 不使用、使用时的轴种类以及 EtherCAT 从站设备的节点地址。

参数名称	功能	设定范围	初始值
轴编号	设定轴的逻辑编号。系统定义变量的变量名称 “_MC_AX[0-255]”、“_MC1_AX[0-255]”、“_MC2_AX[0-255]” 的编号为这里设定的编号。	—	—
运动控制 *1	可分配到主固定周期任务或固定周期任务 （执行优先度 5）中的任意一个任务中。 1: 主固定周期任务 2: 固定周期任务 （执行优先度 5）	1 ～ 2	1
轴使用	设定是否使用轴。*2 0: 未创建轴 1: 未使用轴 *3 2: 使用轴	0 ～ 2	0
轴种类	设定轴种类。 虚拟轴不需要输入输出配线。 0: 伺服轴 1: 编码器轴 2: 虚拟伺服轴 3: 虚拟编码器轴	0 ～ 3	2
输入设备 / 输出设备	选择要分配到轴的 EtherCAT 从站设备的节点地址。*4 轴种类为 ［虚拟轴］时无法选择。	0 ～ 65535	—

*1. NX 系列 CPU 单元时设定。 NJ 系列 CPU 单元只有主固定周期任务，无需设定。
*2. 若在未使用轴及未创建轴中执行运动控制指令， Busy （执行中）将变为 TRUE。若将 Execute （启动）或 Enable （有效）设为 FALSE， Busy （执行中）也变为 FALSE。若设定为 “ 未使用轴 ”，在轴构成不同的装置中共用 1 个用户程序时，不需要删除不使用的轴的程序。
*3. CPU 单元版本 1.04 以上和 Sysmac Studio Ver.1.05 以上组合时，在 Sysmac Studio 的设定画面中，将显示 ［未使用轴 （可切换为使用轴）］、［未使用轴 （不可切换为使用轴）］。若设定为 ［未使用轴 （可切换为使用轴）］，可设定轴参数，可通过 MC_ChangeAxisUse （轴使用变更）指令临时变更 ［轴使用］。
关于轴使用变更功能，请参考 *「9-8-10 轴使用变更功能* (P.9-79)」。
*4. NX 系列位置接口单元时，选择 EtherCAT 耦合器单元的节点地址及位置接口单元的 NX 单元编号。

使用注意事项

使用绝对编码器时
［绝对编码器原点位置偏置］与轴编号关联，按不同轴保存在电池备份存储器中。若变更轴编号，保存的偏置量将丢失。变更了轴编号时，请重新执行原点复位。

轴编号

轴编号中可设定的最大编号为最大控制轴数。此外，可切换为使用轴的实轴数最多为使用实轴最大数。

规格项目	NX701-17□□	NX701-16□□	NJ501-□5□□	NJ501-□4□□	NJ501-□3□□
可设定的轴编号	0 ～ 255	0 ～ 127	0 ～ 63	0 ～ 31	0 ～ 15
最大使用实轴数	256 轴	128 轴	64 轴	32 轴	16 轴

规格项目	NJ301-12□□	NJ301-11□□	NJ101-10□□
可设定的轴编号	0 ～ 14 *1	0 ～ 14 *2	0 ～ 5
最大使用实轴数	8 轴	4 轴	2 轴

*1. 单元版本 1.05 以下的 CPU 单元中，为 “0 ～ 7”。
*2. 单元版本 1.05 以下的 CPU 单元中，为 “0 ～ 3”。

运动控制

NX 系列 CPU 单元时，使用的轴可分配到主固定周期任务或固定周期任务 （执行优先度 5）中的任意一个任务中。

参考

NJ 系列 CPU 单元只有主固定周期任务，无需设定 “ 运动控制 ”。

轴种类

轴种类中可选择的轴内容如下表所示。

轴种类	内容
伺服轴	使用 EtherCAT 从站伺服驱动器或 NX 系列位置接口单元的*1 轴。与实际的伺服驱动器等进行分配后使用。 将 1 台伺服电机作为 1 个轴使用。 使用 NX 系列位置接口单元时，可将脉冲输出单元和数字输入单元等多个设备分配到 1 个轴上。
虚拟伺服轴	MC 功能模块中的虚拟轴。不使用实际的伺服驱动器。 作为同步控制的主轴等使用。
编码器轴	使用 EtherCAT 从站编码器输入终端或 NX 系列位置接口单元 *1 的轴。与实际的编码器输入终端等进行分配后使用。 1 台编码器输入终端中有 2 个计数器时，每个计数器作为 1 个轴使用。
虚拟编码器轴	虚拟执行编码器动作的轴。无编码器实物时，临时代替编码器轴使用。*2

*1. 控制对象设备请参考 *「1-4-3 功能规格* (P.1-8)」。
*2. 与更新虚拟编码器轴当前位置的运动控制指令组合使用。在不支持本指令的 MC 功能模块版本中，不可替代编码器轴使用。

● 虚拟伺服轴

虚拟伺服轴没有物理的编码器或外部输入输出信号。
因此，虚拟伺服轴与伺服轴有以下不同。

- 始终视为伺服 ON 状态。
- 反馈当前位置 ＝ 指令当前位置。*1
- 反馈当前速度 ＝ 指令当前速度。*1
- 不可使用外部输入信号。
- 通过 MC_Home （原点复位）指令或 MC_HomeWithParameter （参数指定原点复位）指令进行原点复位时，无论轴参数中原点复位动作的设定如何，均视为［原点预设］。
- 在使用锁定功能的运动控制指令中，需要将触发输入条件的模式设定为控制器模式。设定为驱动器模式也不会发生异常，但不会发生锁定触发，因此无法完成指令的执行。
  在使用锁定功能的指令中，包括 MC_TouchProbe （外部锁定有效）指令、 MC_MoveFeed （中断固定尺寸定位）指令及 MC_MoveLink （梯形模式凸轮）指令等。
- 由于没有立即停止输入或极限输入信号，所以不会发生立即停止输入或正方向 / 负方向极限输入检测的异常。

*1. 在 MC 功能模块内部，以双精度实数型处理，因此可能包括运算误差。

● 编码器轴 / 虚拟编码器轴

编码器轴 / 虚拟伺服轴与伺服轴 / 虚拟伺服轴有以下不同。

- 没有指令位置。只有反馈位置。
- 不可使用动力系统的运动控制指令。

输入设备 / 输出设备

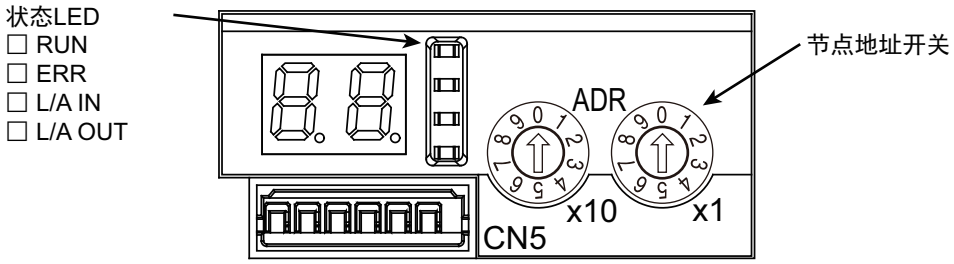
轴种类为伺服轴或编码器轴时，选择要分配到轴的 EtherCAT 设备的节点地址。 NX 系列位置接口单元时，选择 EtherCAT 耦合器单元的节点地址及位置接口单元的 NX 单元编号。
轴种类为 ［虚拟轴］时无法选择。

使用注意事项

- 欧姆龙生产的伺服驱动器 G5 系列可通过驱动器正面的节点地址开关将节点地址设定为固定值。该节点地址开关设定为 “00” 时，节点地址以 Sysmac Studio 的 EtherCAT 编辑器中设定的值为准。
  将连接的所有驱动器的节点地址开关设定为 “00” 时，即使变更驱动器的连接位置，也不会发生异常。要对装置的各控制对象分配特定的伺服驱动器时，请用节点地址开关设定驱动器的节点地址。
- 伺服驱动器节点地址开关的设定值只在接通伺服驱动器的控制电源时读取一次。
  接通电源后，即使发生变更，也不会反映到控制中，要在下次接通电源时变为有效。
  接通电源后，请勿变更节点地址开关的值。
- 若节点地址重复，将发生异常。

参考

- 下面以欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型为例，介绍 EtherCAT 设备的节点地址设定示例。



- 使用伺服驱动器显示部的旋转开关，设定 EtherCAT 节点地址。

旋转开关的设定值	节点地址的设定值	
	欧姆龙生产的从站	其他公司从站 *1
00	Sysmac Studio 的设定值（1 ～ 512）*2	Sysmac Studio 的设定值（1 ～ 512）*2
01 ～ 99	节点地址开关的设定值	

*1. 在其他公司生产的从站中，无论设定有无， Sysmac Studio 的设定值均为有效。
*2. NJ 系列 CPU 单元时，为 “1 ～ 192”。

5-2-3 单位转换设定

设定位置的单位。

参数名称	功能	设定范围	初始值
显示单位	设定指令位置的单位。 • pulse • mm • μm • nm • degree • inch	0 ～ 5	0
电机每转 1 圈的脉冲数 *1	根据编码器分辨率，设定指令位置的电机每转 1 圈的脉冲数。*2 根据电子齿轮比，从指令值转换为脉冲量。	1 ～ 4294967295	10000
电机每转 1 圈的移动量 *3	设定指令位置的电机每转 1 圈的工件移动量。	0.000000001 ～ 4294967295	10000
减速机使用 *4 *5	指定是否使用减速机设定。 TRUE: 使用 FALSE: 不使用	TRUE、 FALSE	FALSE
工件侧每转 1 圈的移动量 *4 *6 *7	设定每转 1 圈后工件的移动量。	双精度实数型的正数	10000
工件侧齿轮比 *4 *6	设定工件侧的齿轮比。	1 ～ 4294967295	1
电机侧齿轮比 *4 *6	设定电机侧的齿轮比。	1 ～ 4294967295	1

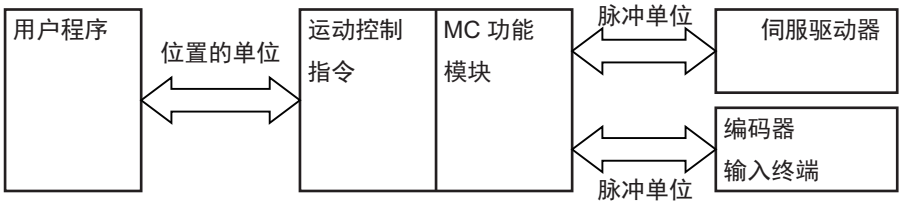
*1. 电子齿轮比 （单位转换公式）的分子。
*2. 例如，编码器分辨率为 10,000 脉冲 / 圈时，请设定为 10,000。
*3. 将减速机使用设定为 “ 不使用 ” 时电子齿轮比 （单位转换公式）的分母。
将减速机使用设定为 “ 使用 ” 时，为无效。
*4. 可在 CPU 单元版本 1.11 以上和 Sysmac Studio Ver.1.15 以上的组合中使用。
*5. 将减速机使用设定为 “ 使用 ” 时，单位转换设定是有条件的。关于条件，请参考 *「减速机使用的条件 (P.5-11)」*。
*6. 将减速机使用设定为 “ 使用 ” 时，为有效。
*7. 仅当计数器模式为直线模式时可设定。为旋转模式时，根据循环计数器上限设定值及循环计数器下限设定值计算。

使用注意事项

- 将减速机使用设定为 “ 不使用 ” 时，按照以下条件舍去位数。可能引起错位，因此请将减速机使用设定为 “ 使用 ”。

  计数器模式为旋转模式，循环计数器 1 圈的脉冲数不是整数时
- 计数器模式为旋转模式，循环计数器 1 圈的脉冲数不是整数时，可将减速机使用设定为 “ 使用 ”，消除错位。
- 除了显示单位的转换，变更单位转换设定后，物理的机械位置和 MC 功能模块的指令当前位置的关系可能发生偏差。因此，变更了单位转换设定时，请执行原点复位指令，重新确定原点。

伺服驱动器或编码器输入终端与 MC 功能模块之间，基本上使用脉冲单位的位置。
运动控制指令中使用的位置以 mm 或 degree 等表示，便于理解实际的动作。



通过显示单位和电子齿轮 （单位转换公式）的设定，脉冲单位的表示可变更为 mm 或 degree。

参考

虚拟伺服轴时，先将指令当前值转换为脉冲单位，再将该值转换为显示单位，将得到的值作为反馈当前值。



显示单位

显示单位为显示在 Sysmac Studio 中的设定单位。显示的是位置的显示单位。
下面介绍可设定的单位。

可设定的单位	说明
pulse	以脉冲单位表现时使用。
mm	以相对较长的距离直线运动时使用。
μm	进行高精度直线运动时使用。
nm	进行比 μm 更高精度的直线运动时使用。
degree	用于轮盘等旋转的轴。
inch	用于直线运动等。

减速机使用的条件

将减速机使用设定为 “ 使用 ” 时，单位转换设定需要满足以下条件。

- 工件侧每转 1 圈的移动量 × 工件侧齿轮比 ÷ 电机侧齿轮比的值为 0.000000001 以上、4294967295 以下。

计数模式为旋转模式时，还要满足以下条件。

- 电机每转 1 圈的脉冲数 × 电机侧齿轮比为 1099511627775 以下。

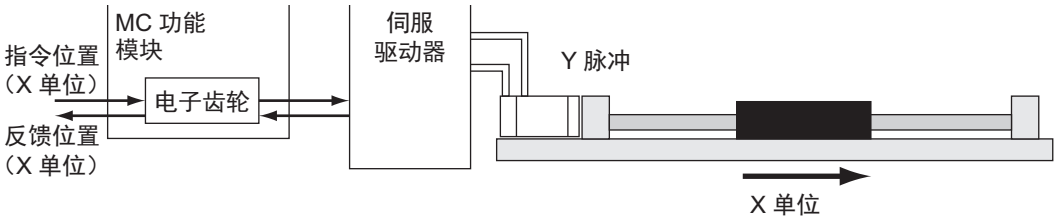
但是，以上条件中的工件侧齿轮比和电机侧齿轮比的值，是以两者的最大公约数为除数计算得到的值。因此，即使以设定值计算时不满足条件，但可能实际是满足条件的。

● **例**

工件侧齿轮比为 “4”、电机侧齿轮比为 “6” 时，将视为工件侧齿轮比为 “2”、电机侧齿轮比为 “3” 计算。

电子齿轮比 （单位转换公式）

在 MC 功能模块中，设定显示单位和脉冲单位的关系时，使用电子齿轮。
使用 Sysmac Studio，设定电子齿轮比。



指令位置 （脉冲）值 = 指令位置 （X 单位） × 电子齿轮比

● **减速机使用设定为 “ 不使用 ” 时**

将减速机使用设定为 “ 不使用 ” 时，电子齿轮比的公式如下。

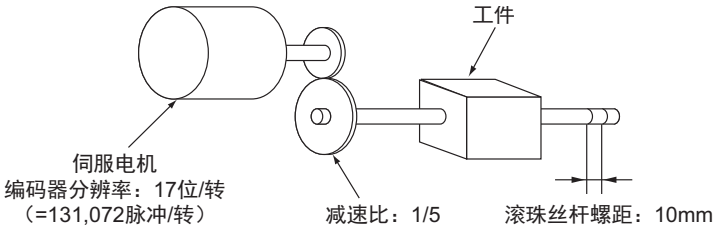
$$\text{电子齿轮比} = \frac{\text{电机每转 1 圈的脉冲数 *1（Y 脉冲）}}{\text{电机每转 1 圈的移动量 *2（X 单位）}}$$

*1. 编码器轴时，为编码器每转 1 圈的脉冲数。
*2. 编码器轴时，为编码器每转 1 圈的移动量。

使用注意事项

MC 功能模块或运动控制指令中使用的位置为通过电子齿轮进行单位转换后的值。
在运动控制指令中，目标位置以 LREAL 型指定，但通过电子齿轮转换为脉冲单位后，若指令位置超出 40 位，将在指令中发生异常。

下面以使用欧姆龙生产的伺服电机 G5 系列带 17 位绝对编码器时的情况为例进行说明。
假设机械系统的减速机减速比为 1/5，滚珠丝杆每转 1 圈，工件移动 10mm。



显示单位为 mm。在电机每转 1 圈的脉冲数中设定伺服电机的编码器分辨率。
由于使用减速比为 1/5 的减速机，因此伺服电机每转 5 圈，滚珠丝杆每转 1 圈。此时，工件移动 2mm （=10mm×1/5），因此电机每转 1 圈的移动量设定为 “2”。

设定项目	设定内容
显示单位	mm
电机每转 1 圈的脉冲数	131,072
电机每转 1 圈的移动量	2

根据以上设定，用户程序中位置的指令单位为 1(mm)。
例如，向 100.5mm 的位置进行绝对值定位时，将 MC_MoveAbsolute （绝对值定位）指令的输入变量 “Position （目标位置） ” 设定为 “100.5”。

参考

在与示例相同的机器上，使用减速比为 1/9 的减速机时
伺服电机每转 1 圈对应的工件移动量为 10mm×1/9 = 1.111•••mm，无法除尽。
出现这种无法除尽的数字时，将 ［电机每转 1 圈的脉冲数］和 ［电机每转 1 圈的移动量］设定为乘以相同系数后的值。
减速比为 1/9，因此将系数设定为 “9”。

- ［电机每转 1 圈的脉冲数］: 1,179,648 (=131,072×9)
- ［电机每转 1 圈的移动量］: 10 (=10×1/9×9)

● 减速机使用为 “ 使用 ”，计数模式为直线模式时

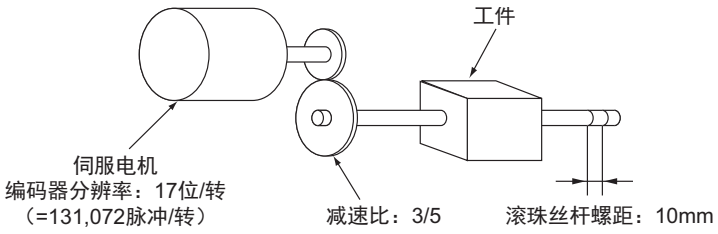
将减速机使用设定为 “ 使用 ” 且计数模式为直线模式时，电子齿轮比的公式如下。

$$\text{电子齿轮比} = \frac{\text{电机每转 1 圈的脉冲数} \times \text{电机侧齿轮比}}{\text{工件每转 1 圈的移动量} \times \text{工件侧齿轮比}}$$

使用注意事项

MC 功能模块或运动控制指令中使用的位置为通过电子齿轮进行单位转换后的值。
在运动控制指令中，目标位置以 LREAL 型指定，但通过电子齿轮转换为脉冲单位后，若指令位置超出 40 位，将在指令中发生异常。

下面以使用欧姆龙生产的伺服电机 G5 系列带 17 位绝对编码器时的情况为例进行说明。
假设机械系统的减速机减速比为 3/5，滚珠丝杆每转 1 圈，工件移动 10mm。



显示单位为 mm。在电机每转 1 圈的脉冲数中设定伺服电机的编码器分辨率。
工件每转 1 圈的移动量设定为滚珠丝杆间距 10mm。
由于使用减速比为 3/5 的减速机，因此伺服电机每转 5 圈，滚珠丝杆旋转 3 圈。此时的减速比设定，工件侧齿轮比 “3”，电机侧齿轮比 “5”。

设定项目	设定内容
显示单位	mm
电机每转 1 圈的脉冲数	131,072
工件每转 1 圈的移动量	10
工件侧齿轮比	3
电机侧齿轮比	5

根据以上设定，用户程序中位置的指令单位为 1(mm)。
例如，向 100.5mm 的位置进行绝对值定位时，将 MC_MoveAbsolute （绝对值定位）指令的输入变量 “Position （目标位置） ” 设定为 “100.5”。

● 减速机使用为 “ 使用 ”，计数模式为旋转模式时

将减速机使用设定为 “ 使用 ” 且计数模式为旋转模式时，电子齿轮比的公式如下。
旋转模式时，工件每转 1 圈的移动量为 “ | 循环计数器上限值 - 循环计数器下限值 | ”，自动设定。

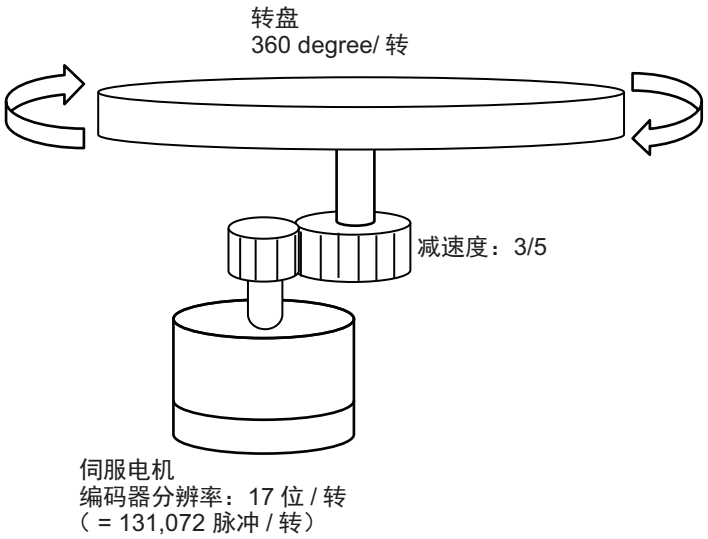
$$\text{电子齿轮比} = \frac{\text{电机每转 1 圈的脉冲数} \times \text{电机侧齿轮比}}{\text{工件每转 1 圈的移动量} \times \text{工件侧齿轮比}}$$

$$= \frac{\text{电机每转 1 圈的脉冲数} \times \text{电机侧齿轮比}}{|\text{循环计数器上限值 - 循环计数器下限值}| \times \text{工件侧齿轮比}}$$

使用注意事项

MC 功能模块或运动控制指令中使用的位置为通过电子齿轮进行单位转换后的值。
在运动控制指令中，目标位置以 LREAL 型指定，但通过电子齿轮转换为脉冲单位后，若指令位置超出 40 位，将在指令中发生异常。

例 1 以使用欧姆龙生产的伺服电机 G5 系列带 17 位绝对编码器时的情况为例进行说明。
假设机械系统的减速机减速比为 3/5，转盘每转 1 圈，工件移动 360degree。



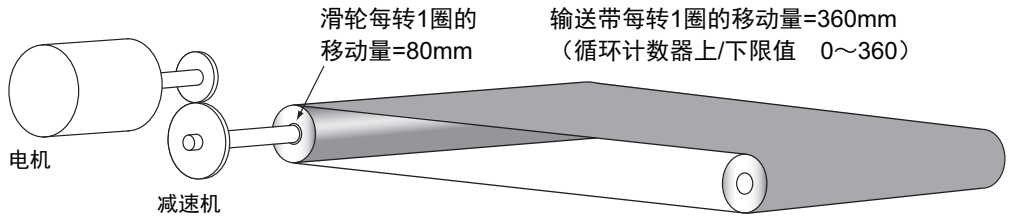
显示单位为 degree。在电机每转 1 圈的脉冲数中设定伺服电机的编码器分辨率。
工件每转 1 圈的移动量为 “ | 循环计数器上限值 - 循环计数器下限值 | ”，自动设定。
由于使用减速比为 3/5 的减速机，因此伺服电机每转 5 圈，转盘 （工件）旋转 3 圈。此时的减速比设定，工件侧齿轮比 “3”，电机侧齿轮比 “5”。

设定项目	设定内容
显示单位	degree
电机每转 1 圈的脉冲数	131,072
工件每转 1 圈的移动量	| 循环计数器上限值 - 循环计数器下限值 |
工件侧齿轮比	3
电机侧齿轮比	5

根据以上设定，用户程序中位置的指令单位为 1(degree)。
例如，向 100.5degree 的位置进行绝对值定位时，将 MC_MoveAbsolute （绝对值定位）指令的输入变量 “Position （目标位置） ” 设定为 “100.5”。

例 2 以使用欧姆龙生产的伺服电机 G5 系列带 20 位绝对编码器时的情况为例进行说明。
假设机械系统的减速机减速比为 3/5，滑轮每转 1 圈，输送带移动 80mm。输送带 1 圈为 360mm。

伺服电机
编码器分辨率： 20位/转
（ = 1,048,57 6脉冲/转）



减速机
比率：3/5

机械结构
比率：80/360

最终比率：

$$= \frac{3 \times 80}{5 \times 360} = \frac{\text{工件侧齿轮比}}{\text{电机侧齿轮比}}$$

显示单位为 mm。在电机每转 1 圈的脉冲数中设定伺服电机的编码器分辨率。
输送带每转 1 圈的移动量为 “ | 循环计数器上限值 - 循环计数器下限值 | ”，自动设定。

设定项目	设定内容
循环计数器上限值	360
循环计数器下限值	0

由于使用减速比为 3/5 的减速机，因此伺服电机每转 5 圈，滑轮旋转 3 圈。此外，滑轮每转 1 圈，输送带的移动量为 80mm，因此减速比设定为工件侧齿轮比 “3×80”=“240”，电机侧齿轮比 “5×360”=“1800”。

设定项目	设定内容
显示单位	mm
电机每转 1 圈的脉冲数	1,048,576
工件每转 1 圈的移动量	| 循环计数器上限值 - 循环计数器下限值 |
工件侧齿轮比	240
电机侧齿轮比	1800

参考

- 滑轮每转 1 圈的移动量不是整数时，如 80.1mm 等，请使用乘以 “10” 后变为整数后的值，然后计算工件侧齿轮比、电机侧齿轮比。若在上述示例中，工件侧齿轮比变为 “3×80.1×10”=“2403”，电机侧齿轮比变为 “5×360×10”=“18000”。
- 减速机为多档时，也与例 2 相同，请将多个比率相乘得到的最终比率设定为工件侧齿轮比、电机侧齿轮比。

5-2-4 动作设定

根据要控制的装置的规格，设定最高速度、最大加减速度等轴动作。

参数名称	功能	设定范围	初始值
最高速度	设定各轴的最高速度。*1 设定时请勿超出所用电机的最大转速。 （单位：指令单位 /s）	双精度实数型的正数 *2	400,000,000
启动速度 *3	设定各轴的启动速度。 设定时请勿超出最高速度。 （单位：指令单位 /s）	双精度实数型的正数	0
点动最高速度	设定各轴的点动最高速度。*4 设定时请勿超出最高速度。 （单位：指令单位 /s）	双精度实数型的正数	1,000,000
最大加速度	设定轴动作指令时各轴的最大加速度。 若设定为 “0”，将没有加速度的限制。 （单位：指令单位 /s^2）	双精度实数型的正数、 0	0
最大减速度	设定轴动作指令时各轴的最大减速度。 若设定为 “0”，将没有减速度的限制。 （单位：指令单位 /s^2）	双精度实数型的正数、 0	0
加减速超限	在轴的加减速控制中，为了优先停止在目标位置，会发生减速超限，导致超出最大加减速度，指定这种情况下的动作。 0: 剧烈加减速 （将共混切换为缓冲）。*5 1: 剧烈加减速。 2: 异常停止。*6	0 ～ 2	0
反转时动作	指定指令多重启动时、指令重新启动时、中断固定尺寸定位时的反转时动作。*7 0: 减速停止 1: 立即停止	0 ～ 1	0
速度警告值	设定相对于最高速度的比例，作为各轴输出速度警告的标准。 若设定为 “0”，将不输出速度警告。 （单位：%）	0 ～ 100	0
加速度警告值	设定相对于最高加速度的比例，作为各轴输出加速度警告的标准。 若设定为 “0”，将不输出加速度警告。 （单位：%）	0 ～ 100	0
减速度警告值	设定相对于最高减速度的比例，作为各轴输出减速度警告的标准。 若设定为 “0”，将不输出减速度警告。 （单位：%）	0 ～ 100	0
正方向扭矩警告值	设定扭矩指令值，作为各轴输出正方向扭矩警告的标准。 若设定为 “0”，将不输出正方向扭矩警告。 （单位：%）	0 ～ 1000	0
负方向扭矩警告值	设定扭矩指令值，作为各轴输出负方向扭矩警告的标准。 若设定为 “0”，将不输出负方向扭矩警告。 （单位：%）	0 ～ 1000	0

参数名称	功能	设定范围	初始值
反馈速度过滤器时间常数	以毫秒为单位设定计算反馈速度移动平均值的时间。 若设定为 “0”，将不计算移动平均值。 （单位：ms） 轴的速度较慢时，需要减少反馈当前速度变动的情况下使用。	0 ～ 100	0
位置宽度 *8	设定定位完成宽度 （单位：指令单位）	双精度实数型的正数、 0	10
位置检查时间 *8	以毫秒为单位设定完成定位的检查时间。 若设定为 “0”，仅在原点复位中确定原点时继续执行定位完成检查，不是原点复位时，不执行定位完成检查。（单位：ms）	0 ～ 10,000	0
原点位置范围	设定原点位置检测宽度。 （单位：指令单位）	双精度实数型的正数、 0	10

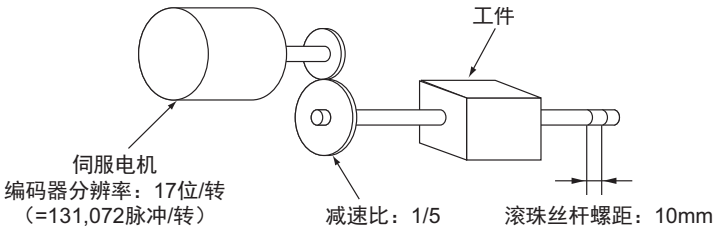
*1. 指定的速度指令值超出最高速度时，按最高速度发出指令。插补控制动作时也有效。
*2. 可设定的最大值为转换为脉冲单位的值，如下所示。
单元版本 1.02 以下的 CPU 单元中， 400,000,000 [pulse/s] ；
单元版本 1.03 以上的 CPU 单元中， 500,000,000 [pulse/s] ；
单元版本 1.11 以上的 CPU 单元中， 2,147,483,647 [pulse/s]。
*3. 可在 CPU 单元版本 1.05 以上和 Sysmac Studio Ver.1.06 以上的组合中使用。
*4. 指定的速度指令值超出点动最高速度时，按点动最高速度发出指令。
*5. 单元版本 1.10 以上的 CPU 单元中，无法将共混切换为缓冲。详情请参考 *「9-5-7 运动控制指令的多重启动 （缓冲模式）*(P.9-45)」。
*6. 单元版本 1.10 以上的 CPU 单元中，发生共混动作时，将视为发生异常而停止。详情请参考 *「9-5-7 运动控制指令的多重启动 （缓冲模式）*(P.9-45)」。
*7. 反转时动作的详情请参考 *「9-5-6 运动控制指令的重启* (P.9-40)」及 *「9-5-7 运动控制指令的多重启动 （缓冲模式）*(P.9-45)」。
*8. 位置检查在 MC 功能模块上执行。不使用伺服驱动器上的功能。

最高速度

下面介绍最高速度的设定示例。

● 最高速度的设定示例

下面介绍在与上述 *「减速机使用设定为 “ 不使用 ” 时* (P.5-12)」相同的机器上，伺服电机的最大转速为 6,000r/min 时的情况。



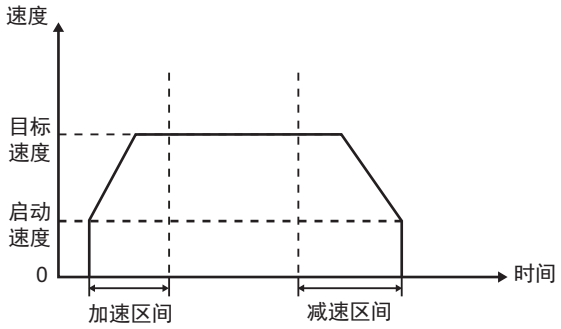
若按照最大转速：6,000r/min、减速比：1/5、滚珠丝杆间距：10mm 计算， 6,000r/min × 1/5 × 10mm = 12,000mm/min = 200mm/s，因此最高速度设定为 “200”。
初始值 400,000,000 超出电机的最大转速，因此请务必变更设定值。

设定项目	设定内容
显示单位	mm
电机每转 1 圈的脉冲数	131,072
电机每转 1 圈的移动量	2
最高速度	200

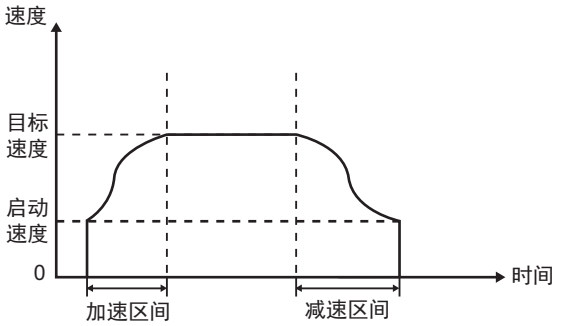
启动速度

使用伺服电机时，请设定为 “0”。
使用步进电机时，为了防止启动时失步，通常设定为最大自启动频率的 1/10 ～ 1/2。启动速度因负载情况的不同而变化，详情请参考步进电机的手册。

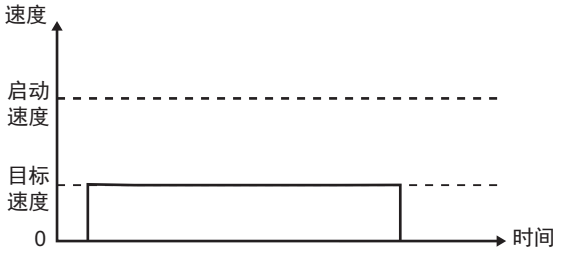
未指定跃度时



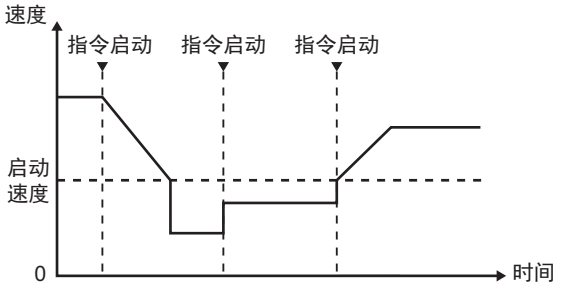
指定跃度时



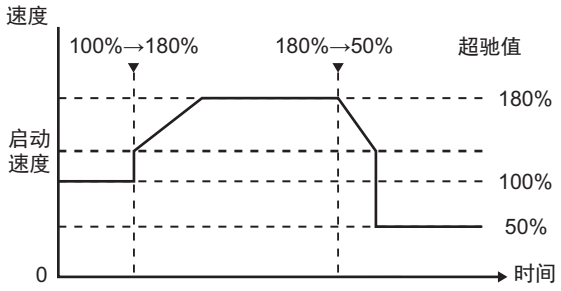
目标速度低于启动速度时，不进行加减速，直接以目标速度动作。



在动作过程中因运动控制指令的重新执行或多重启动，导致目标速度发生变更时，仍使用启动速度。目标速度高于启动速度时，按指定的加速度和减速度进行加减速动作。目标速度低于启动速度时，不进行加减速，直接动作。



利用 MC_SetOverride （超驰值设定）指令变更了速度时，仍使用启动速度。目标速度高于启动速度时，按指定的加速度和减速度进行加减速动作。目标速度低于启动速度时，不进行加减速，直接动作。



在以下功能中，不使用启动速度。

- 扭矩控制模式
- 周期性同步速度控制
- 周期性同步定位
- 同步控制
  但是，在 MC_GearOut （齿轮动作解除）、 MC_CamOut （凸轮动作解除）中，使用启动速度。
- 多轴协调控制
  但是，在 “ 减速停止各轴 ” 中减速时，使用各轴的启动速度。

5-2-5 扩展动作设定

设定各种输入信号有效时的停止方法和扭矩限制值等。

参数名称	功能	设定范围	初始值
立即停止输入停止方法	设定立即停止输入有效时 MC 功能模块的停止方法。 0: 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	0、 2、 3	0
极限输入停止方法	设定正方向极限输入或负方向极限输入有效时 MC 功能模块的停止方法。 0: 立即停止指令值 1: 减速停止指令值 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	0 ～ 3	0
驱动器错误重置监视时间	设定执行驱动器错误重置时的监视时间。（单位：ms） 若超出监视时间，即使驱动器的异常未重置，也将结束重置处理。	1 ～ 1000	200
正方向扭矩限制上限值	设定正方向扭矩限制值的上限。*1 （单位：%）	0.0 ～ 1000.0	300.0
负方向扭矩限制上限值	设定负方向扭矩限制值的上限。*1 （单位：%）	0.0 ～ 1000.0	300.0
立即停止输入逻辑反转 *2	设定立即停止输入信号的逻辑反转。 FALSE: 不反转 TRUE: 反转	FALSE、 TRUE	FALSE *3
正方向极限输入逻辑反转 *2	设定正方向极限输入信号的逻辑反转。 FALSE: 不反转 TRUE: 反转	FALSE、 TRUE	FALSE *3
负方向极限输入逻辑反转 *2	设定负方向极限输入信号的逻辑反转。 FALSE: 不反转 TRUE: 反转	FALSE、 TRUE	FALSE *3

参数名称	功能	设定范围	初始值
近原点输入逻辑反转 *2	设定近原点输入信号的逻辑反转。 FALSE: 不反转 TRUE: 反转	FALSE、TRUE	FALSE

*1. PDO 映射中映射了 “ 正转侧扭矩限制值 (60E0Hex)” 及 “ 反转侧扭矩限制值 (60E1Hex)” 时，这些设定值将通过 EtherCAT 的过程数据通信发送。通过 MC_SetTorqueLimit （扭矩限制）指令将扭矩限制设为有效时，将发送指令的输入变量中指定的值。

*2. 可在 CPU 单元版本 1.05 以上和 Sysmac Studio Ver.1.06 以上的组合中使用。单元版本 1.04 以下的 CPU 单元中，逻辑值不反转。
分配了 NX 系列数字输入单元等无法设定信号逻辑的设备时设定。在欧姆龙生产的伺服驱动器 G5 系列等可设定输入信号逻辑的设备中，通常设定为 “ 不反转 ”。

*3. 在分配了 NX 系列脉冲输出单元的轴中，初始值为 “TRUE”。

5-2-6 限位设定

选择位置偏差相关的限位功能和利用软件的限位功能。

参数名称	功能	设定范围	初始值
软件限位功能 *1	选择软件限位的功能。 0: 无效 1: 对指令位置有效，减速停止 2: 对指令位置有效，立即停止 3: 对反馈位置有效，减速停止 4: 对反馈位置有效，立即停止	0 ～ 4	0
正方向软件限位	设定正方向的软件限位。（单位：指令单位）	双精度实数型的负数、正数、 0	2147483647
负方向软件限位	设定负方向的软件限位。（单位：指令单位）	双精度实数型的负数、正数、 0	-2147483648
位置偏差超限值	设定位置偏差超限的阈值。 若设定为 “0”，位置偏差超限检查为无效。（单位：指令单位）	双精度实数型的正数、 0	0
位置偏差警告值	设定位置偏差警告的阈值。 若设定为 “0”，位置偏差警告检查为无效。（单位：指令单位）	双精度实数型的正数、 0 （小于位置偏差超限值）	0

*1. 本功能仅在计数模式为直线模式且已确定原点时有用。
关于软件限位功能的详情，请参考 「*9-8-5 软件限位功能* (P.9-73)」。

5-2-7 位置计数设定

设定位置的计数模式。

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

参数名称	功能	设定范围	初始值
计数模式	设定位置的计数模式。 0: 直线模式 （长度有限） 1: 旋转模式 （长度无限）	0 ～ 1	0
循环计数器上限设定值	设定计数模式为旋转模式时循环计数器的上限值。（单位：指令单位）	双精度实数型的负数、正数、 0	2147483647
循环计数器下限设定值	设定计数模式为旋转模式时循环计数器的下限值。（单位：指令单位）	双精度实数型的负数、正数、 0	-2147483648
编码器种类	选择编码器种类。*1 *2 0: 增量编码器 (INC) 1: 绝对编码器 (ABS)	0 ～ 1	0

*1. 使用以下产品时，编码器种类请设定为 ［1: 绝对编码器 (ABS)］。
使用欧姆龙生产的伺服电机 G5 系列带绝对编码器的伺服电机时
在欧姆龙生产的伺服电机 / 驱动器 G5 系列的全闭环控制中使用绝对型外部光栅尺时
在欧姆龙生产的伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型中，使用绝对型外部光栅尺

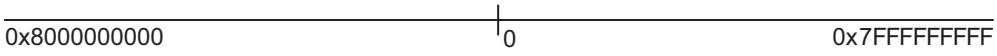
*2. 在欧姆龙生产的伺服电机 / 驱动器 G5 系列的全闭环控制中使用外部光栅尺时，或使用欧姆龙产伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型时，请分别替换为以下内容。
0: 增量型外部光栅尺
1: 绝对型外部光栅尺

计数模式

轴传送模式。按各个轴选择指令位置的计数模式。分为轴传送范围为有限长度的直线模式和无限长度的旋转模式。

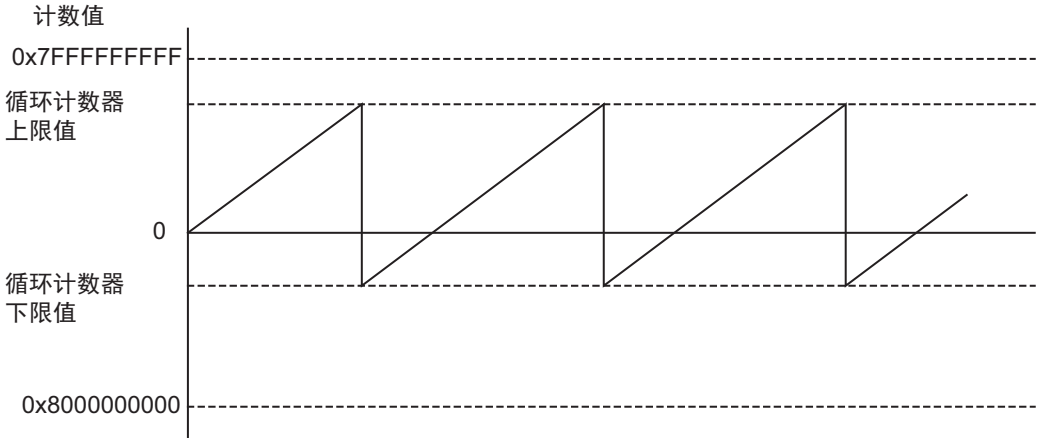
● 直线模式 （有限长度轴）

- 以 “0” 为中心的直线设定。
  在 X-Y 平台等机械动作范围有限制的装置上使用。
- 设定范围为转换为脉冲单位后的带符号整数型 40 位 （0x8000000000 ～ 7FFFFFFFFF）。
- 在相对值移动或绝对值移动等指定目标位置的定位中，指令中不可指定超出范围的目标位置。
- 在速度控制、原点复位、扭矩控制等不指定目标位置的动作中，若超出动作范围，将变为监视信息 （指令位置溢出 / 下溢）。将继续输出指令位置，但不会更新当前位置，固定为上限值或下限值。
- 在当前位置固定的状态下，在不指定目标位置的动作中，对于向范围内靠近的指令和停止，可以启动。对于远离范围的指令，启动指令时将发生异常。
- 在解除溢出 / 下溢状态前，将不执行当前位置的更新。



● 旋转模式 （无限长度轴）

- 在设定范围内重复进行无限计数的循环计数器的模式。
  在轮盘或卷绕轴中使用。
- 循环计数器的范围可通过 Sysmac Studio 设定循环计数器的上限值和下限值。
- 设定范围为转换为脉冲单位后的带符号整数型 40 位 （0x8000000000 ～ 7FFFFFFFFF）。



循环计数器上限值和循环计数器下限值

计数模式设定为旋转模式时有效的设定值。
设定循环计数器的上限值和下限值。

使用注意事项

- 计数模式为旋转模式时，请确保将循环计数器的 1 圈转换为脉冲单位得到的值为整数。
  若循环计数器的 1 圈不是整数脉冲，在不同单元版本中会出现以下情况。
  单元版本 1.11 以上的 CPU 单元中，可将减速机使用设定为 “ 使用 ”，可消除错位。减速机使用请参考 *「5-2-3 单位转换设定* (P.5-10)」。
  单元版本 1.10 以下的 CPU 单元中，脉冲数的尾数会被舍去，因此会导致错位。此外，出现尾数时，指令当前位置将无法正常显示。
- 若循环计数器的上限值和下限值中不含 “0”，执行 MC_MoveZeroPosition （高速原点复位）指令时将发生异常。
- 通过 MC_MoveAbsolute （绝对值定位）指令或 MC_Move （定位）指令执行绝对值定位时，请在循环计数器的范围内设定目标位置。超出范围时将发生异常。
  但是，若在指令的输入变量 Direction （方向选择）中指定了 “4: 无方向指定 ”，超出范围的值也可设定为目标位置。此时将进行相对值定位，以目标位置和指令当前位置的差为目标距离。
- 变更位置计数设定后，物理的机械位置和 MC 功能模块的指令当前位置的关系可能发生偏差。因此，变更了位置计数设定时，请执行原点复位指令，重新确定原点。

编码器种类

选择进行反馈输入的编码器种类。
使用以下产品时，编码器种类请设定为［1: 绝对编码器 (ABS)］。

- 使用欧姆龙生产的伺服电机 G5 系列带绝对编码器的伺服电机时
- 在欧姆龙生产的伺服电机 / 驱动器 G5 系列的全闭环控制中使用绝对型外部光栅尺时
- 在欧姆龙生产的伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型中，使用绝对型外部光栅尺

5-2-8 伺服驱动器设定

设定所连接伺服驱动器或编码器输入终端中设定的值。

参数名称	功能	设定范围	初始值 *1
循环计数器上限设定值	设定伺服驱动器或编码器输入终端中设定的循环计数器上限值。*2	$-2^{63} \sim 2^{63}-1$	2147483647
循环计数器下限设定值	设定伺服驱动器或编码器输入终端中设定的循环计数器下限值。*2	$-2^{63} \sim 2^{63}-1$	-2147483648
PDS 状态控制方式 *3	利用 MC_Power 指令，设定伺服 OFF 时 PDS 状态的变化目标。*4 0: 伺服 OFF 时 Switeched on 1: 伺服 OFF 时 Ready to switch on	0 ～ 1	0

*1. 初始值为整数型 (DINT) 的全部范围。欧姆龙生产的伺服驱动器 G5 系列为整数型 (DINT) 的全部范围，因此可按初始值使用。
使用 NX 系列位置接口单元时请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。
*2. 使用欧姆龙生产的 GX 系列 EtherCAT 从站编码器输入终端时，需要将编码器输入终端的循环计数器最大值（对象索引 0x4003）和循环计数器上限设定值统一。此外，循环计数器下限设定值需要设定为 “0”。
*3. 可在 CPU 单元版本 1.10 以上和 Sysmac Studio Ver.1.12 以上的组合中使用。
*4. 使用欧姆龙生产的伺服驱动器 G5 系列时，若设定为 “1”，将无法使用伺服驱动器的伺服就绪 (Switeched on)。使用伺服就绪 (Switeched on) 时，请设定为 “0”。关于 PDS 状态的变化，请参考 🕮「*A-4 PDS 状态变化* (P.A-19)」。

5-2-9 原点复位设定

设定用于确认机械原点的电机动作。

参数名称	功能	设定范围	初始值
原点复位动作 *1 *2	设定原点复位的动作方法。 0: 附近退避、近原点输入 OFF 指定 1: 附近退避、近原点输入 ON 指定 4: 近原点输入 OFF 指定 5: 近原点输入 ON 指定 8: 极限输入 OFF 指定 9: 附近退避、原点输入掩码距离指定 11: 仅极限输入 12: 附近退避、推压时间指定 13: 无近原点输入、推压原点输入指定 14: 原点预设	0, 1, 4, 5, 8, 9, 11 ～ 14	14
原点输入信号	选择原点输入信号。 0: 使用 Z 相输入。 1: 使用外部原点输入。*3	0 ～ 1	0
原点复位开始方向	设定原点复位启动时的开始方向。 0: 正方向 2: 负方向	0、 2	0
原点检测方向	设定原点复位的原点输入检测方向。 0: 正方向 2: 负方向	0、 2	0
正方向极限输入时动作	设定在原点复位过程中将正方向极限输入设为 “ON” 时的停止方法。 0: 不反转、轴异常停止 （停止方法遵照极限输入停止方法） 1: 反转、立即停止 2: 反转、减速停止	0 ～ 2	1
负方向极限输入时动作	设定在原点复位过程中将负方向极限输入设为 “ON” 时的停止方法。 0: 不反转、轴异常停止 （停止方法遵照极限输入停止方法） 1: 反转、立即停止 2: 反转、减速停止	0 ～ 2	1
原点复位速度	设定原点复位时的速度。 （单位：指令单位 /s）	双精度实数型的正数	10,000
原点复位接近速度	设定近原点输入变为 “ON” 后的速度。 （单位：指令单位 /s）	双精度实数型的正数	1,000
原点复位加速度	指定原点复位时的加速度。 若设定为 “0”，则不加速，达到原点复位速度等目标速度。 （单位：指令单位 /s^2）	双精度实数型的正数、 0	0
原点复位减速度	指定原点复位时的减速度。 若设定为 “0”，则不减速，达到原点复位接近速度等目标速度。（单位：指令单位 /s^2）	双精度实数型的正数、 0	0
原点复位跃度	指定原点复位时的跃度。 若设定为 “0”，将没有跃度。 （单位：指令单位 /s^3）	双精度实数型的正数、 0	0
原点输入掩码距离	设定原点复位动作模式为 “ 附近退避、原点输入掩码距离指定 ” 时的原点输入掩码距离。 （单位：指令单位）	双精度实数型的正数、 0	10,000

参数名称	功能	设定范围	初始值
原点位置偏置	原点复位完成时按设定的值预设当前位置。 （单位：指令单位）	双精度实数型的负数、正数、 0	0
原点复位推压时间	以毫秒为单位设定原点复位动作模式为 " 附近退避、推压时间指定 " 时的推压时间。（单位：ms）	0 ～ 10,000	100
原点复位修正值	设定原点复位的原点确定后的原点复位修正量。 （单位：指令单位）	双精度实数型的负数、正数、 0	0
原点复位修正速度	设定原点复位修正时的速度。 （单位：指令单位 /s）	双精度实数型的正数	1,000

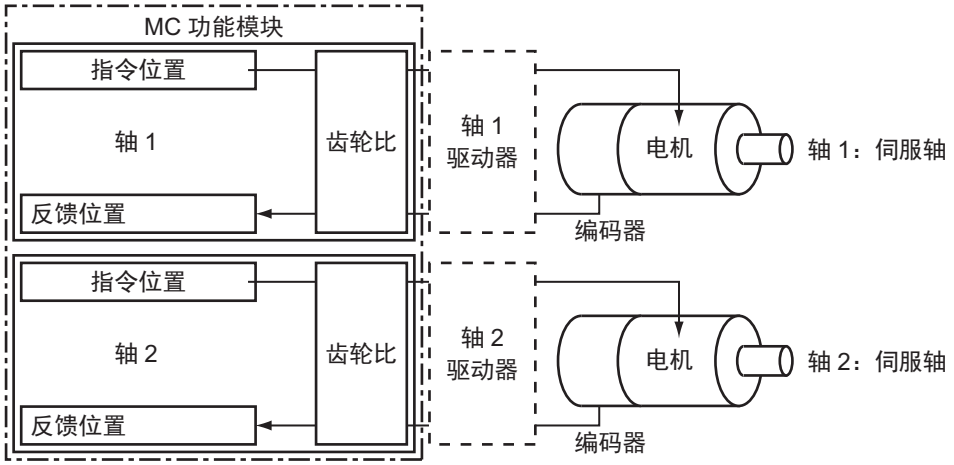
*1. 设定原点复位动作。详情请参考 🕮 *「第 8 章 定原点」*。
*2. 在欧姆龙产伺服驱动器 G5 系列的直线电机型中，无法将 Z 相输入映射到 PDO 中。因此，若使用原点复位动作中的 " 无近原点输入、推压原点输入指定 "，需要用到 PDO 中映射的 Z 相输入，因此请勿在原点输入信号中选择 Z 相输入。
*3. 使用欧姆龙生产的伺服驱动器 G5 系列时可以指定。将伺服驱动器 G5 系列的锁定功能 1 中分配的输入作为外部原点输入使用。在伺服驱动器 G5 系列的初始状态下，锁定功能 1 中分配了外部锁定输入 1。
详情请参考 🕮《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型 用户手册 (SBCE-CN5-365)》及 🕮《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型 用户手册 (SBCE-CN5-366)》。
使用 NX 系列位置接口单元时请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

5-2-10 轴参数的设定示例

下面介绍轴参数中与位置相关的项目设定示例。

单轴定位

下面以两个轴分别用单轴定位的装置为例进行说明。

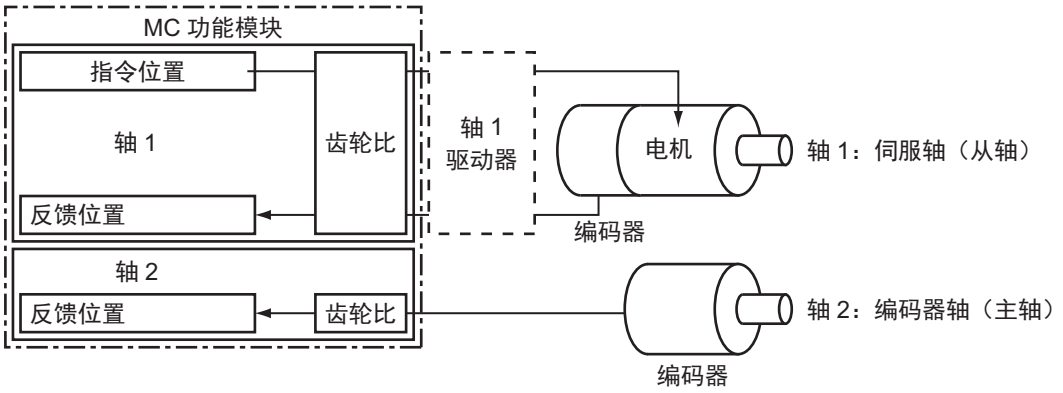


设定项目	设定内容	
	轴 1	轴 2
轴变量名称	Axis1	Axis2
轴编号	1	2
轴使用	使用轴	使用轴
轴种类	伺服轴	伺服轴
输入设备 / 输出设备	1	2
显示单位	μm *1	μm
电机每转 1 圈的脉冲数	1,048,576	1,048,576
电机每转 1 圈的移动量	10,000	10,000
最高速度	500,000 *2	500,000 *2
点动最高速度	50,000 *3	50,000 *3
最大加速度	5,000,000 *4	5,000,000 *4
最大减速度	5,000,000 *4	5,000,000 *4
软件限位功能	对指令位置有效，立即停止	对指令位置有效，立即停止
正方向软件限位	500,000 *5	500,000 *5
负方向软件限位	0 *5	0 *5
计数模式	直线模式	直线模式

*1. 位置的指令单位为 1(μm)。
*2. 最高速度为 3,000r/min = 30m/min = 0.5m/s = 500,000μm/s。
*3. JOG 最高速度为最高速度的 10%， 0.05m/s = 50,000μm/s。
*4. 最大加速度及最大减速度为 5m/s^2。加速至最高速度 3,000r/min 的时间为 0.1s。
*5. 设定装置可移动范围的内侧位置。
正方向软件限位的设定值为 50cm ＝ 500,000μm。

编码器为主轴的同步控制

下面以将输送带上安装的轴 2 （编码器）的反馈位置作为主轴，轴 1 的伺服驱动器在有限长度范围内同步的装置为例进行说明。



设定项目	设定内容	
	轴 1	轴 2
轴变量名称	Axis1	Axis2
轴编号	1	2
轴使用	使用轴	使用轴
轴种类	伺服轴	编码器轴
输入设备 / 输出设备	1	2
显示单位	μm *1	μm *1
电机每转 1 圈的脉冲数	1,048,576	1,048,576
电机每转 1 圈的移动量	10,000	10,000
最高速度	500,000 *2	－
点动最高速度	50,000 *3	－
最大加速度	5,000,000 *4	－
最大减速度	5,000,000 *4	－
软件限位功能	对指令位置有效，立即停止	无效
正方向软件限位	500,000 *5	－
负方向软件限位	0 *5	－
计数模式	直线模式	旋转模式
循环计数器上限值	－	1,000,000 *6
循环计数器下限值	－	0 *6

*1. 位置的指令单位为 1(μm)。
*2. 最高速度为 3,000r/min = 30m/min = 0.5m/s = 500,000μm/s。
*3. JOG 最高速度为最高速度的 10%， 0.05m/s = 50,000μm/s。
*4. 最大加速度及最大减速度为 5m/s^2。加速至最高速度 3,000r/min 的时间为 0.1s。
*5. 设定装置可移动范围的内侧位置。
正方向软件限位的设定值为 50cm ＝ 500,000μm。
*6. 位置的重复范围为 0 ～ 1m (1,000,000μm）。

参考

主轴可根据不同的装置构成选择轴种类。轴种类有伺服轴、虚拟伺服轴、编码器轴、虚拟编码器轴四种。
本示例中，主轴的轴种类为编码器轴，因此运动控制指令的输入参数 “ReferenceType （位置类型选择） ” 请指定为反馈位置。

5-3 轴组参数

在轴组参数中，对 MC 功能模块控制的轴组，设定轴构成、最高插补速度、轴组停止方法等轴组动作。
NJ301-11□□、NJ301-12□□、NJ501-□3□□、NJ501-□4□□、NJ501-□5□□ 的轴组参数为 32 组，NX701-16□□、 NX701-17□□ 为 64 组。
此外，各轴组的参数设定相同，所以这里只介绍 1 个组的设定。

5-3-1 轴组参数一览

轴组参数使用 Sysmac Studio 对各轴组分别进行设定。

区分	参数名称	临时变更 *1		变量读取 *2	参照页
		可否	使用指令		
轴组基本设定	轴组编号			可	P. 5-30
	运动控制 *3			可 *4	
	轴组使用			可	
	轴构成			可	
	构成轴	可 *5	MC_ChangeAxesInGroup	可	
轴组动作设定	最高插补速度				P. 5-32
	最大插补加速度				
	最大插补减速度				
	插补加减速超限				
	插补速度警告值	可	MC_Write		
	插补加速度警告值				
	插补减速度警告值				
	轴组停止方法				
	中心点修正容许率				

*1. 表示是否可通过指令临时变更。
*2. 表示是否可从用户程序中读取变量。
*3. NX 系列 CPU 单元时设定。
*4. 可在 CPU 单元版本 1.10 以上和 Sysmac Studio Ver.1.12 以上的组合中使用。
*5. 可在 CPU 单元版本 1.01 以上和 Sysmac Studio Ver.1.02 以上的组合中使用。

关于轴组参数的设定方法，请参考 🕮「*3-4 轴组的设定步骤* (P.3-24)」。
MC_Write （MC 设定写入）指令、 MC_ChangeAxesInGroup （轴组构成轴写入）指令的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。
关于运动控制系统变量，请参考 🕮「*6-6 运动控制系统变量* (P.6-16)」。

5-3-2 轴组基本设定

设定轴组的使用 / 不使用以及使用时的轴构成和使用的轴。

参数名称	功能	设定范围	初始值
轴组编号	设定轴组的逻辑编号。系统定义变量的变量名称 “_MC_GRP[0-63]”、 “_MC1_GRP[0-63]”、 “_MC2_GRP[0-63]” 的编号为这里设定的编号。	—	—
运动控制 *1	可分配到主固定周期任务或固定周期任务 （执行优先度 5）中的任意一个任务中。 1: 主固定周期任务 2: 固定周期任务 （执行优先度 5）	1 ～ 2	1
轴组使用	设定是否使用轴组。若对未创建轴组及未使用轴组执行运动控制指令，将发生异常。*2 0: 未创建轴组 1: 未使用轴组 2: 使用轴组	0 ～ 2	0
轴构成	设定轴组的轴构成。 0: 2 轴 1: 3 轴 2: 4 轴	0 ～ 2	0
构成轴	设定分配到轴组的轴编号。 在 Sysmac Studio 中，为轴 A0 ～ A3 选择轴变量名称。	2 ～ 4 轴	0

*1. NX 系列 CPU 单元时设定。 NJ 系列 CPU 单元只有主固定周期任务，无需设定。
*2. 若对包含未使用轴的轴组执行 MC_GroupEnable （轴组有效）指令，将发生异常。

轴构成

下表为 MC 功能模块中可使用的轴构成。

用 Sysmac Studio 根据实际装置设定轴构成。

轴构成	说明
2 轴	轴构成为 2 轴。例如， 2 轴直角坐标系的机械。
3 轴	轴构成为 3 轴。例如， 3 轴直角坐标系的机械。
4 轴	轴构成为 4 轴。例如， 3 轴直角坐标系和有前端工具旋转轴的机械。

构成轴

属于轴组的轴称为构成轴。此外，若为轴组指令中的插补指令，为了提高程序的重复利用性，使用逻辑轴 （轴 A0 ～轴 A3）代替轴编号 （轴 0 ～轴 255）。

在构成轴中设定轴组中轴的编号和逻辑轴的编号分配。

逻辑轴可从伺服轴或虚拟伺服轴中选择。

用 Sysmac Studio 从轴 A0 开始依次设定，轴的数量为轴构成中选择的轴数。

创建多个轴组时，各个轴组从轴 A0 开始依次设定轴编号。

此外，还可在多个轴组中设定同一个轴编号。

轴构成的设定	构成轴的设定
2 轴	设定轴 A0、轴 A1 的轴变量名称 （轴编号）。
3 轴	设定轴 A0、轴 A1、轴 A2 的轴变量名称 （轴编号）。
4 轴	设定轴 A0、轴 A1、轴 A2、轴 A3 的轴变量名称 （轴编号）。

使用注意事项

轴组的构成轴请全部分配到同一任务中。

版本相关信息

CPU 单元版本 1.04 以上和 Sysmac Studio Ver.1.05 以上组合时，如果是伺服轴或虚拟伺服轴，不仅是使用轴，连设定为未使用轴 （可切换为使用轴）的轴也可设定到轴组中。

● 构成轴的设定示例

- 将轴编号为 1、 2、 5、 8 的 4 个轴分配到轴组时的设定如下。

逻辑轴	轴编号
轴 A0	轴 1
轴 A1	轴 2
轴 A2	轴 5
轴 A3	轴 8

- 将轴编号为 1、 8、 2 的 3 个轴分配到轴组时的设定如下。

逻辑轴	轴编号
轴 A0	轴 1
轴 A1	轴 8
轴 A2	轴 2
轴 A3	（无）

5-3-3 轴组动作设定

根据要控制的装置的规格，设定最高插补速度、轴组停止方法等轴组动作。

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

参数名称	功能	设定范围	初始值
最高插补速度	设定轨迹的最高插补速度。 若设定为 “0”，将没有插补速度的限制。 轴组动作指令中指定了超出最高插补速度的插补目标速度时，以最高插补速度动作。（单位：指令单位 /s）	双精度实数型的正数、 0	800000000
最大插补加速度	设定轨迹的最大插补加速度。 若设定为 “0”，将没有插补加速度的限制。（单位：指令单位 /s^2）	双精度实数型的正数、 0	0
最大插补减速度	设定轨迹的最大插补减速度。 若设定为 “0”，将没有插补减速度的限制。（单位：指令单位 /s^2）	双精度实数型的正数、 0	0
插补加减速超限	在轴组的加减速控制中，为了优先停止在目标位置，会发生减速超限，导致超出最大插补加减速度，指定这种情况下的动作。 0: 剧烈加减速 （将共混切换为缓冲）。*1 1: 剧烈加减速。 2: 异常停止。*2	0 ～ 2	0
插补速度警告值	设定相对于最高插补速度的比例，作为输出插补速度警告的标准。 若设定为 “0”，将不输出插补速度警告。 （单位：%）	0 ～ 100	0
插补加速度警告值	设定相对于最高插补加速度的比例，作为输出插补加速度警告的标准。 若设定为 “0”，将不输出插补加速度警告。（单位：%）	0 ～ 100	0
插补减速度警告值	设定相对于最高插补减速度的比例，作为输出插补减速度警告的标准。 若设定为 “0”，将不输出插补减速度警告。（单位：%）	0 ～ 100	0
轴组停止方法	设定在多轴协调动作过程中，当某个轴发生立即停止异常时，未发生异常的构成轴的停止方法。 0: 立即停止 1: 各轴减速停止 （各轴的最大减速度） 3: 立即停止，同时执行伺服 OFF	0、 1、 3	0
中心点修正容许率	在圆弧插补指令的中心点指定方式中，起点和中心点、终点和中心点的距离不同时，将距离修正为相等。以相对于半径的比例设定该修正的容许范围。比例以 0.1% 以上设定。 若设定为 “0”，不检查误差。	单精度浮动小数点型， 0 ～ 100	0

*1. 单元版本 1.10 以上的 CPU 单元中，无法将共混切换为缓冲。详情请参考 🕮「*9-5-7 运动控制指令的多重启动 （缓冲模式）*(P.9-45)」。

*2. 单元版本 1.10 以上的 CPU 单元中，发生共混动作时，将视为发生异常而停止。详情请参考 🕮「*9-5-7 运动控制指令的多重启动 （缓冲模式）*(P.9-45)」。

5-3-4 轴组的有效

若要将用户程序中针对轴组的动作指令设为有效，需要利用 MC_GroupEnable （轴组有效）指令指定要设为有效的轴组编号。
若对未设定为有效的轴组启动运动控制指令，指令中将发生异常。
此外，可同时将多个轴组设为有效，但若将包含同一个轴的轴组设为有效，指令中将发生异常。

若要在不同作业工序的不同轴组中使用同一个轴，需要先创建多个包含该轴的轴组。然后依次切换 MC_GroupEnable （轴组有效）指令和 MC_GroupDisable （轴组无效）指令即可实现。

多轴动作过程中，若启动 MC_GroupDisable （轴组无效）指令，则减速停止。

6

运动控制程序

本章介绍运动控制程序的规格和创建程序为止的操作步骤。

6-1 概要 6-2

6-2 运动控制指令 6-4

6-2-1 PLCopen® 运动控制用功能块 6-4

6-2-2 运动控制功能模块的运动控制指令 6-4

6-3 状态变化 6-5

6-3-1 运动控制功能模块的状态 6-5

6-3-2 轴的状态 6-5

6-3-3 轴组的状态 6-7

6-4 运动控制指令的启动和状态 6-8

6-4-1 启动相关的共通规则 6-8

6-4-2 启动时序图 6-10

6-4-3 运动控制指令重新执行时的时序 6-12

6-4-4 运动控制指令多重执行时的时序 6-13

6-5 位置处理 6-14

6-5-1 位置的种类 6-14

6-5-2 各种轴类别可使用的位置类型 6-15

6-6 运动控制系统变量 6-16

6-6-1 运动控制系统变量的概要 6-16

6-6-2 运动控制系统变量的原理 6-18

6-6-3 运动控制系统变量的一览 6-19

6-7 凸轮表和凸轮数据变量 6-30

6-8 运动控制程序的创建方法 6-33

6-9 凸轮表的创建方法 6-35

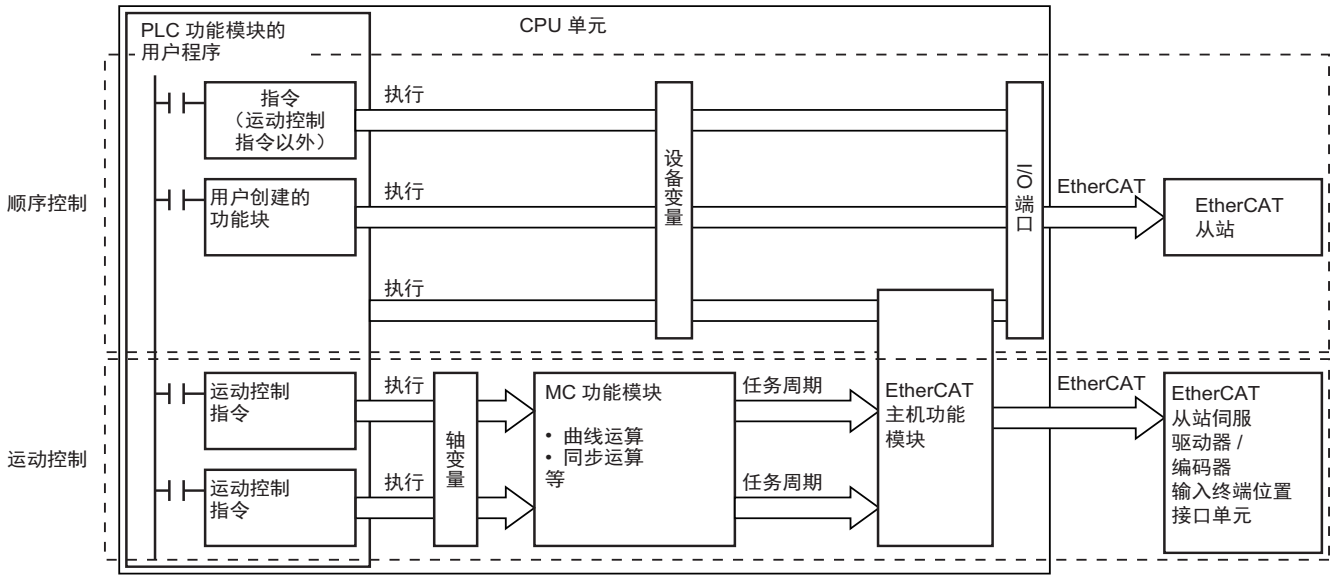
6-1 概要

NJ/NX 系列 CPU 单元中可执行顺序控制和运动控制。
通过在用户程序中记述运动控制指令，可以对 EtherCAT 从站伺服驱动器或 NX 系列位置接口单元等对象执行运动控制。
本手册中，记述了运动控制指令的程序称为运动控制程序。
EtherCAT 从站伺服驱动器或 NX 系列位置接口单元中需要分配轴变量。若不分配轴变量，与通用从站一样，分配 I/O 设备变量。
运动控制指令可在主固定周期任务或固定周期任务 （执行优先度 5 和 16）中使用。

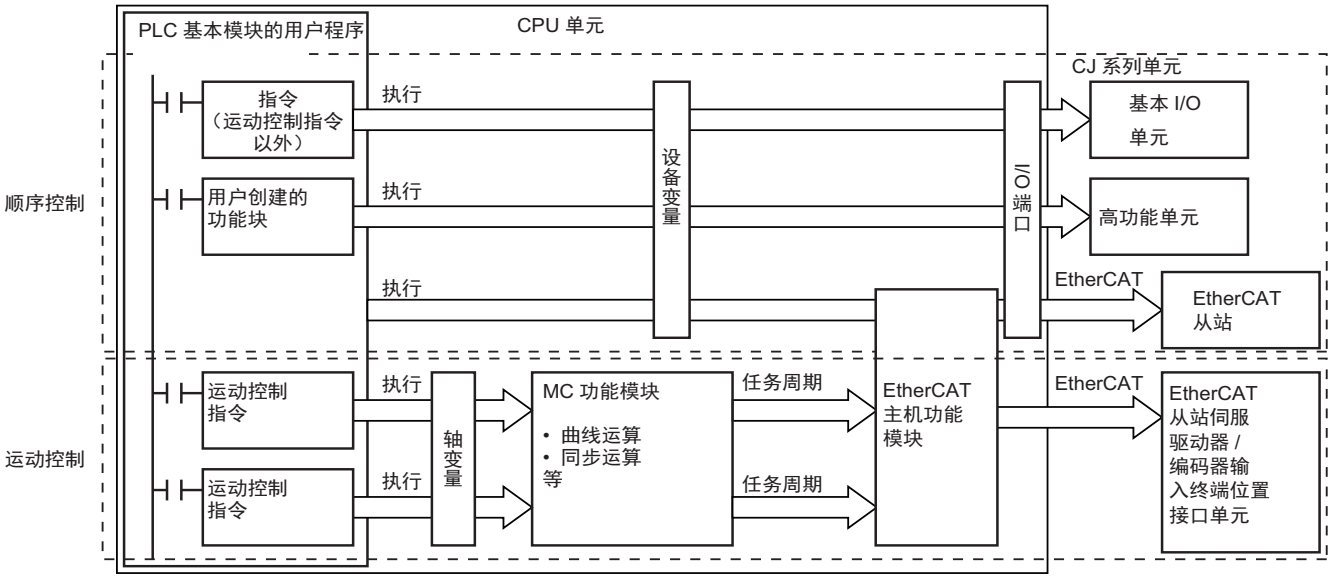
版本相关信息

使用 Sysmac Studio Ver.1.09 以上，可向轴变量中分配的从站和单元的 I/O 端口分配设备变量。
详情请参考 *「2-4-2 EtherCAT 主机功能和运动控制功能模块的关系* (P.2-18)」。

● NX 系列 CPU 单元时



● NJ 系列 CPU 单元时


PLC 基本模块的用户程序
CPU 单元
CJ 系列单元
顺序控制
运动控制
指令
（运动控制指令
以外）
用户创建的
功能块
执行
执行
执行
设备变量
I/O端口
基本 I/O
单元
高功能单元
EtherCAT
EtherCAT
从站
运动控制
指令
运动控制
指令
执行
执行
轴变量
MC 功能模块
• 曲线运算
• 同步运算
等
任务周期
任务周期
EtherCAT
主机功能
模块
EtherCAT
EtherCAT
从站伺服
驱动器 /
编码器输
入终端位置
接口单元

6-2 运动控制指令

若要通过 NJ/NX 系列的用户程序执行运动控制功能，需要使用定义为功能块 （以下可能简称 FB）的运动控制指令。
MC 功能模块的运动控制指令以 PLCopen® 的运动控制功能块技术规格为基础。运动控制指令分为 PLCopen® 中定义的指令和 MC 功能模块独立的指令两种。
下面介绍 PLCopen® 运动控制用 FB 的概要和 MC 功能模块的规格。

NJ/NX 系列功能块 (FB) 的基本事项请参考 🕮《NJ/NX 系列 CPU 单元 用户手册 软件篇（SBCA-CN5-359）》。

在 CPU 单元版本 1.05 以上和 Sysmac Studio Ver.1.06 以上的组合中，可使用 NX 系列位置接口单元。
使用 NX 系列位置接口单元时请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

6-2-1 PLCopen® 运动控制用功能块

PLCopen® 通过对运动控制用功能块进行标准化，定义了 IEC 61131-3 （JIS B 3503）标准语言的程序界面。
除了单轴定位、电子凸轮功能以及多轴插补控制等功能，还对启动指令所需的基本步骤做出定义。
利用 PLCopen® 运动控制功能块，可减少对硬件的依赖，提高用户程序的重复利用性。
此外，还可减少培训和支持的费用。



参考

什么是 PLCopen®
PLCopen® 为总部位于欧洲的 IEC 61131-3 普及团体，是一个全球性会员组织。
IEC 61131-3 为 PLC 的编程国际标准规格。
PLCopen® Japan 为该组织在日本市场的推广委员会，由关注日本市场的人员组成。

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

6-2-2 运动控制功能模块的运动控制指令

运动控制指令有以下三种。

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

MC 功能模块中可使用的指令一览和详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。

6-3 状态变化

MC 功能模块的轴、轴组状态以及启动指令后的状态变化以 PLCopen® 的运动控制功能块技术规格为基础。
下面介绍包括 MC 功能模块整体在内的各状态和状态变化。

6-3-1 运动控制功能模块的状态

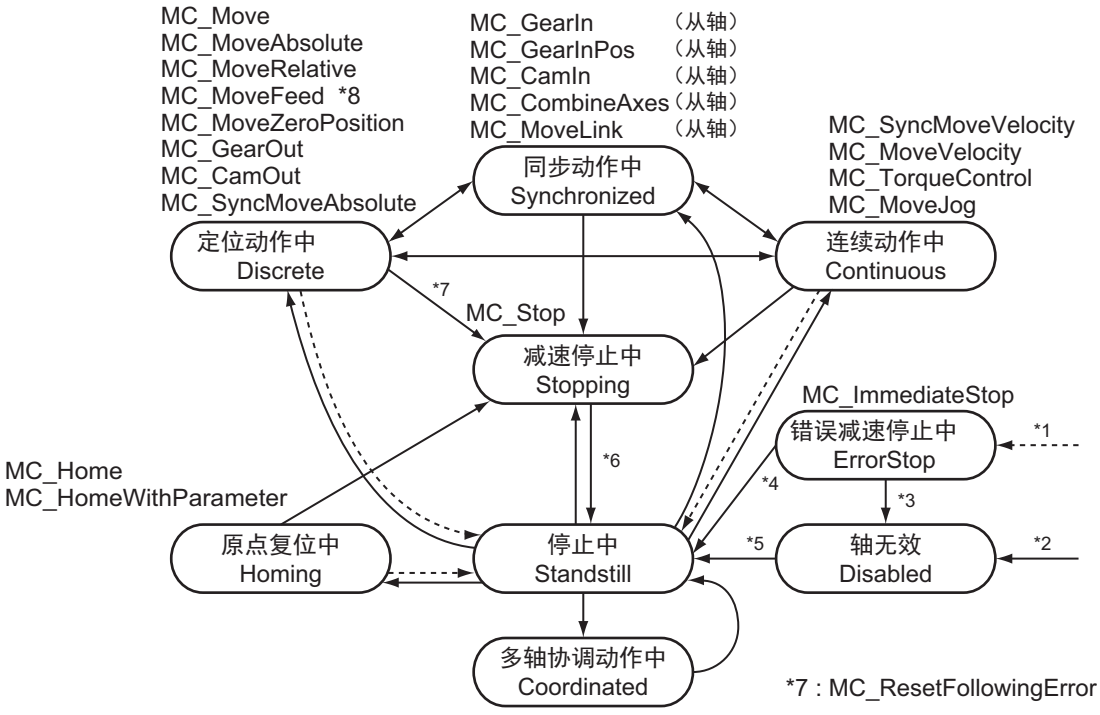
MC 功能模块整体的状态如下表所示。

状态名称	定义
MC 功能模块运行中 *1	运动控制指令为有效状态。 分析用户程序的运动控制指令，进行运动控制。 与 CPU 单元的动作模式无关，始终可设为 “MC 功能模块运行中 ”。
MC 功能模块试运行中 *2	正在通过 Sysmac Studio 执行试运行的状态。
凸轮表文件保存中 *3	凸轮表文件保存中或待机中的状态。
凸轮表生成执行中 *4	凸轮表生成执行中的状态。*5

*1. 可利用 MC 共通变量 “_MC_COM.Status.RunMode” 监视。
*2. 可利用 MC 共通变量 “_MC_COM.Status.TestMode” 监视。
*3. 可利用 MC 共通变量 “_MC_COM.Status.CamTabeleBusy” 监视。
*4. 可利用 MC 共通变量 “_MC_COM.Status.GenerateCamBusy” 监视。
*5. 若要切断 CPU 单元的电源，请确认不是正在执行凸轮表生成。若在执行凸轮表生成的过程中关闭 CPU 单元的电源，将无法正确生成凸轮表。

6-3-2 轴的状态

启动了针对轴的运动控制指令时，轴的动作如下表所示。
原则上，运动控制指令将依次执行，轴变为下表中的任一状态。



*1. 在 “ 多轴协调动作中 ” 以外的任一状态下，可在发生轴异常后变化。

*2. 在任意状态下，当 MC_Power 指令的 “Status （可运行） ” 输出为 “FALSE”、未发生轴异常时变化。（伺服 OFF）
*3. 在伺服 OFF 状态下，通过 MC_Reset 指令或 ResetMCError 指令解除异常后变化。
*4. 在伺服 ON 状态下，通过 MC_Reset 指令或 ResetMCError 指令解除异常后变化。
*5. 将 MC_Power 指令的 “Enable （有效） ” 输入设为 “TRUE”， MC_Power 指令的 “Status （可运行） ” 输出为 “TRUE” 时变化。（伺服 ON）
*6. MC_Stop 指令的 “Done （完成） ” 输出为 “TRUE”、 MC_Stop 指令的 “Execute （启动） ” 输入为 “FALSE” 时变化。
*7. 通过 MC_ResetFollowingError 指令变化为减速停止中。
*8. MC_MoveFeed 指令的输入变量 “MoveMode （移动方法选择） ” 设定为 “2: _mcVelocity （速度控制） ” 时，在检测到触发输入前，将变为 “ 连续动作中 ”。

状态名称		定义
伺服 OFF		轴为伺服 OFF 状态。 变为此状态后，将解除指令多重启动的待机状态。
	轴无效	轴为伺服 OFF、停止中、启动准备尚未完成的状态。
	错误减速停止中 *1	轴为伺服 OFF、停止中、轴发生异常的状态。
伺服 ON		轴为伺服 ON 状态。
	停止中	轴为伺服 ON、停止中状态。
	定位动作中	正在执行指定目标位置的定位。 包括等待定位完成的状态或在定位动作中将超驰值设定为 “0” 后速度变为 “0” 的状态。
	连续动作中	正在执行不指定目标位置的连续动作。 速度控制或扭矩控制时变为该状态。 包括将目标速度设为 “0” 后速度为 “0” 的状态或在连续动作中将超驰值设定为 “0” 后速度变为 “0” 的状态。
	同步动作中	利用同步控制指令对轴进行同步控制的状态。 包括切换同步控制指令后的同步等待状态。
	减速停止中	利用 MC_Stop （强制停止）指令或 MC_TouchProbe （外部锁定有效）指令停止轴时，轴完全停止前的状态。 包括利用 MC_Stop （强制停止）指令停止后， Execute （启动）为 TRUE 的状态。 在此状态下，无法利用轴指令执行动作。 若执行，指令的 CommandAborted （执行中断）将变为 TRUE。
	错误减速停止中 *1	轴为伺服 ON、轴发生异常的状态。 包括正在执行 MC_ImmediateStop （立即停止）指令的状态或因轴发生异常而正在减速的状态。 在此状态下，无法执行轴动作指令。若执行，指令将变为中断 （CommandAborted ＝ 1）状态。
	原点复位中	正在通过 MC_Home （原点复位）指令或 MC_HomeWithParameter （参数指定原点复位）指令寻找原点的状态。
	多轴协调动作中	轴收到轴组指令，处于轴组有效的状态。 轴组的状态为 “ 组动作中 ”、 “ 减速停止中 ” 或 “ 错误停止中 ”，处于正在动作的状态。

*1. “ 错误减速停止中 ” 包括轴为伺服 OFF 时和伺服 ON 时两种情况。

（注）轴状态可通过轴变量 “_MC_AX[0-255].Status” 的各成员变量监视。

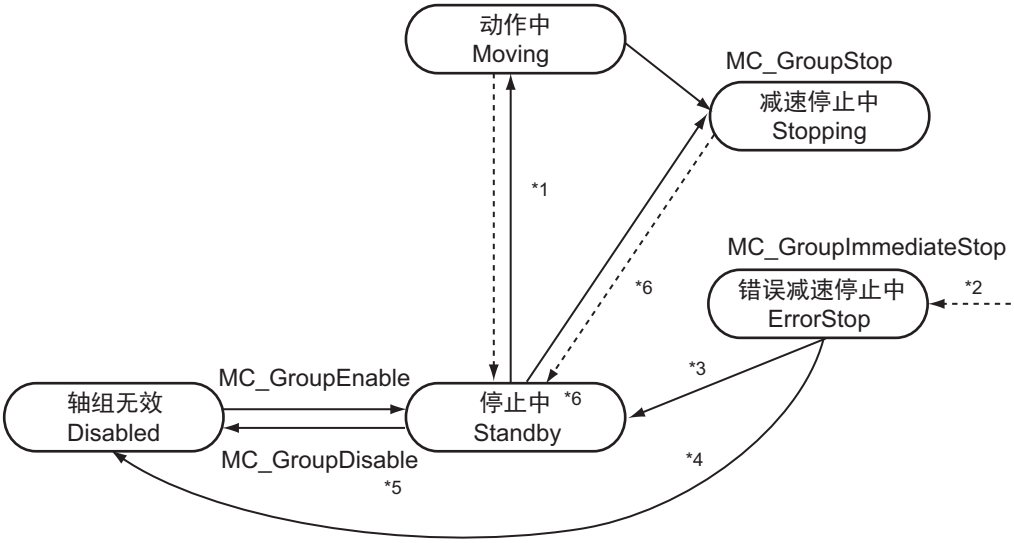
版本相关信息

NX 系列 CPU 单元时，以 _MC_AX[*] 开始的变量名称可能是 _MC1_AX[*]、 _MC2_AX[*]。

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

6-3-3 轴组的状态

启动了针对轴组的运动控制指令时，轴组的动作如下表所示。



*1. 在所有轴组动作指令下变化。
*2. 从所有状态变化。即使轴组为无效状态，若发生异常，也会变化。
*3. 轴组为有效时，在 MC_GroupReset 指令或 ResetMCError 指令下变化。
*4. 轴组为无效时，在 MC_GroupReset 指令或 ResetMCError 指令下变化。
*5. 错误减速停止中状态时，即使启动 MC_GroupDisable 指令，也保持相同的状态。
*6. MC_GroupStop 指令的输出变量 “Done” = TRUE、输入变量 “Execute” = FALSE 时变化。
*7. 在停止中 （Standby）状态下，满足以下所有条件时，则启动准备完成 (Ready)。
所有构成轴为伺服 ON。
所有构成轴未执行 MC_Stop 指令。
所有构成轴已确定原点。

状态名称		定义
轴组无效		轴组为无效状态。 变为本状态后，将解除指令多重启动的待机状态。
	错误减速停止中 *1	轴组为无效的状态下，轴组发生异常的状态。
轴组有效		轴组为有效状态。
	停止中	未执行轴组指令的动作指令的状态。 （与轴组构成轴的伺服 ON/OFF 状态无关。）
	动作中	正在利用轴组指令的动作指令，执行指定目标位置的定位。 包括位置等待的状态或在轴组动作中将超驰值设定为 “0” 后速度变为 “0” 的状态。
	减速停止中	正在执行 MC_GroupStop （轴组强制停止）指令的状态。 包括利用 MC_GroupStop （轴组强制停止）指令停止后， Execute （启动）为 TRUE 的状态。 在此状态下，无法利用轴组指令执行动作。 若执行，指令的 CommandAborted （执行中断）将变为 TRUE。
	错误减速停止中 *1	轴组发生异常的状态。 包括正在执行 MC_GroupImmediateStop （轴组立即停止）指令的状态或因轴组发生异常而正在减速的状态，即本状态。 在本状态下，禁止执行多轴协调指令。 若执行， CommandAborted （执行中断）将变为 TRUE。

*1. “ 错误减速停止中 ” 包括轴组无效时和轴组有效时两种情况。

（注）轴组的状态可通过轴组变量
“_MC_GRP[0-63].Status”、 “_MC1_GRP[0-63].Status”、 “_MC2_GRP[0-63].Status” 的各成员变量监视。

6-4 运动控制指令的启动和状态

对启动运动控制指令的变量和表示指令执行状态的变量作出定义，作为 MC 功能模块的共通规则。
启动运动控制指令的功能所需的输入变量有 “Execute” 和 “Enable” 两种。表示指令执行状态的输出变量有 “Busy”、 “Done”、 “CommandAborted”、 “Error” 等。

6-4-1 启动相关的共通规则

MC 功能模块的共通规则如下表所示。
启动示例记载于 *「6-4-2 启动时序图* (P.6-10)」。
请同时参考。

项目	原则
输出的排他性	输出变量 Busy （执行中）、 Done （完成）、 Error （错误）、 CommandAborted （执行中断）相互排斥，不会同时有多个变为 TRUE。 同样，输出变量 Active （控制中）、 Done （完成）、 Error （错误）、 CommandAborted （执行中断）也不会同时有多个变为 TRUE。 Busy （执行中）和 Active （控制中）可能会同时变为 TRUE。
输出状态	输出变量 Done （完成）、 InGear （达到齿轮比）、 InSync （同步中）、 InVelocity （达到目标速度）及 CommandAborted （执行中断）在输入变量 Execute （启动）的下降沿为 FALSE。 在 Execute （启动）的下降沿，不会停止执行实际的运动控制指令。 指令实例执行完成前，即使 Execute （启动）变为 FALSE，一旦指令实例的状态发生变化，至少在 1 个周期内，对应的输出变量将变为 TRUE。 输出变量 Error （错误）和 ErrorID （错误代码）不会清零，直至执行 MC_Reset （轴错误重置）指令、 MC_GroupReset （轴组错误重置）指令或 ResetMCError （所有错误重置）指令。*1 正在执行运动控制指令时，即使同一个指令实例的 Execute （启动）再次从 FALSE 变为 TRUE （指令重新启动）， CommandAborted （执行中断）也不会变为 TRUE。
输入参数	输入变量 Execute （启动）中启动的运动控制指令的输入参数使用 Execute （启动）上升沿时的值。 输入变量 Enable （有效）中启动的运动控制指令的输入参数使用 Enable （有效）为 TRUE 期间各周期的值。
输入参数的省略	省略了指令实例的输入参数时，应用初始值。*2
“Position(目标位置)” 和 “Distance(移动距离)”	输入变量的 Position （目标位置）为坐标系中定义的值。 输入变量 Distance （移动距离）为相对长度，是两个位置之间的差。
符号规则	输入变量 Acceleration （加速度）、 Deceleration （减速度）、 Jerk （跃度）为正值或 “0”。 Position （目标位置）、 Distance （移动距离）、 Velocity （目标速度）可以是正值、负值或 “0” 中的任意值。
异常处理动作	表示指令实例执行过程中发生的异常的输出变量有两个。 两者的输出定义如下。 • Error （错误）：输出变量 “Error” 的上升沿表示在指令实例执行过程中发生了异常。 • ErrorID （错误代码）：表示异常原因的代码。 为了表示正常完成或正在正常动作，输出变量 Done （完成）、 InVelocity （达到目标速度）、 InGear （达到齿轮比）、 InSync （同步中）与输出变量 Error （错误）有排他性。

项目	原则
异常处理动作 （接上页）	异常种类有以下两种。 • 指令实例的异常 （例：超出参数范围、违背状态变化的条件） • 轴异常 （例：位置偏差超限、伺服驱动器侧的异常） 发生指令实例异常时，可能不发生轴异常而停止轴。
输出变量 “Done （完成） ” 的动作	输出变量 Done （完成）、 InGear （达到齿轮比）、 InSync （同步中）在指令动作正常完成或达到指令的状态时变为 TRUE。若为指定目标位置的动作指令，变为 TRUE 的时间因轴参数 “ 位置检查时间 ” 的设定不同而异。 • 若为 “0 以外 ”，则在完成定位后的下一个周期变为 TRUE。 • 若为 “0”，则在指令结束 （输出完成）的下一个周期变为 TRUE。 同一个轴上动作的指令有多个时，若在轴动作到达最终目标位置前被其他动作指令中断，第一个指令的输出变量 “Done （完成） ” 不会变为 TRUE。
输出变量 “CommandAborted （执行中断） ” 的动作	输出变量 CommandAborted （执行中断）在指令的动作被其他动作指令中断时变为 TRUE。 本 MC 功能模块中，运动控制指令启动时，即使对象轴或轴组正在发生异常或正在减速停止，也会变为 TRUE。 CommandAborted （执行中断）变为 TRUE 后，其他输出变量将变为 FALSE。
超出有效范围的输入	若在输入变量的值超出有效范围的情况下启动，指令实例将发生异常。
输出变量 “Busy （执行中） ” 的动作	输出变量 Busy （执行中）表示正在执行指令实例。 Busy （执行中）在输入变量 Execute （启动）的上升沿变为 TRUE，输出变量 Done （完成）、 CommandAborted （执行中断）、 Error （错误）变为 TRUE 时变为 FALSE。 由于上述输出变量何时变化不确定，因此 Busy （执行中）为 TRUE 期间，为了监视输出变量的变化，请将程序编写为每个周期执行指令实例*3。 在一个轴或一个轴组中，可以有多个指令实例的 Busy （执行中）变为 TRUE，但是输出变量 Active （控制中）一次只能有一个指令实例变为 TRUE。 但是， MC_Phasing （主轴相对值相位补偿）指令例外。
输出变量 “Active （控制中） ”	输出变量 Active （控制中）在指令实例获得该轴的控制权后变为 TRUE。 输出变量 Active （控制中）的变化可能会晚于 Busy （执行中）。*4

*1. 这个规格与 PLCopen® 不同。 PLCopen® 中，在 Execute （启动）的下降沿， Error （错误）和 ErrorID （错误代码）将变为 “0”。
此外， Error （错误）为 TRUE 期间，不会启动运动控制指令。解除异常后，即使 Execute （启动）为 TRUE，由于不是上升沿，指令也不会启动。 Enable 型运动控制指令在解除异常后，一旦 Enable （有效）变为 TRUE，就会启动。
*2. 用梯形图语言创建程序时，请在输入变量 Execute （启动）或 Enable （有效）和左总线之间配置接点。若不配置接点而直接与左总线连接，链接时将发生异常。
请对预约 (Reserved) 处理的输入变量设定为省略输入参数或设定为初始值。
*3. ST 构造指令中条件时和选择值不一致时，该语句中记述的指令不会执行。详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。
*4. 输出变量的时序图请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。

使用注意事项

- 启动运动控制指令时，请确认 EtherCAT 通信的过程数据通信正在正常执行。
  详情请参考 🕮「*10-2-1 EtherCAT 通信的监视和伺服 ON* (P.10-3)」。
- 请创建用户程序，使 Execute （启动）在第一个用户程序执行周期内变为 “FALSE”。

6-4-2 启动时序图

MC 功能模块的运动控制指令为常时执行型 FB。

下面假设启动运动控制指令的功能所需的输入变量 “Execute （启动） ” 的指令为 Execute 型、 “Enable （有效） ” 的指令为 Enable 型进行说明。

启动条件	内容
Execute 型	运动控制指令的输入变量 “Execute （起動） ” 上升时启动的指令。 持续执行指令，直至变为以下状态。 • 指令中指定的动作完成 • 因启动其他运动控制指令而中断动作 • 因 Execute （启动）的再上升沿而重新启动 其他输入变量在 Execute （启动）上升时导入值。
Enable 型	运动控制指令的输入变量 “Enable （有效） ” 为 TRUE 期间，每个周期执行的指令。 “Enable （有效） ” 为 TRUE 期间，其他输入变量也会每个周期都导入。 但也有例外， MC_MoveJog （点动进给）的输入变量 “Velocity （目标速度） ”、 “Acceleration （加速度） ”、 “Deceleration （减速度） ” 将导入 “PositiveEnable （正方向有效） ” 或 “NegativeEnable （负方向有效） ” 上升时的值。

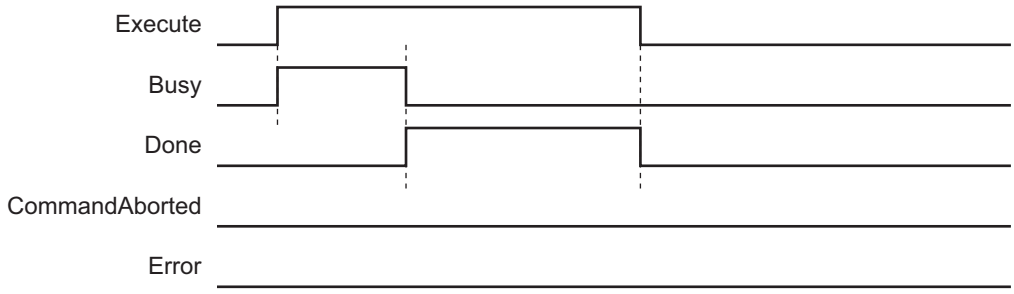
使用注意事项

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

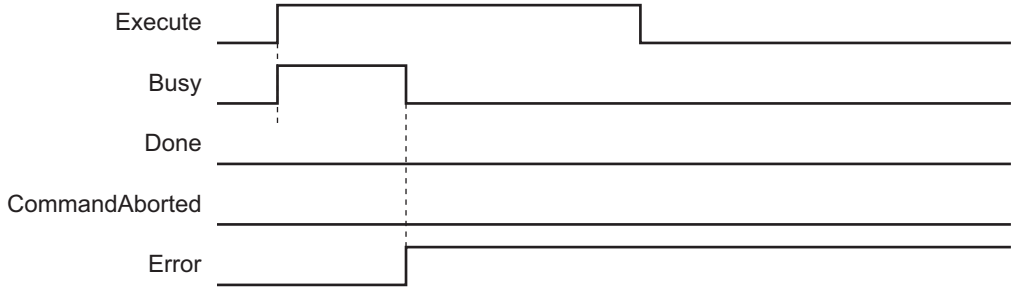
数据追踪的详情请参考 🕮《NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-CN5-359)》。

Execute 型的指令执行时序图

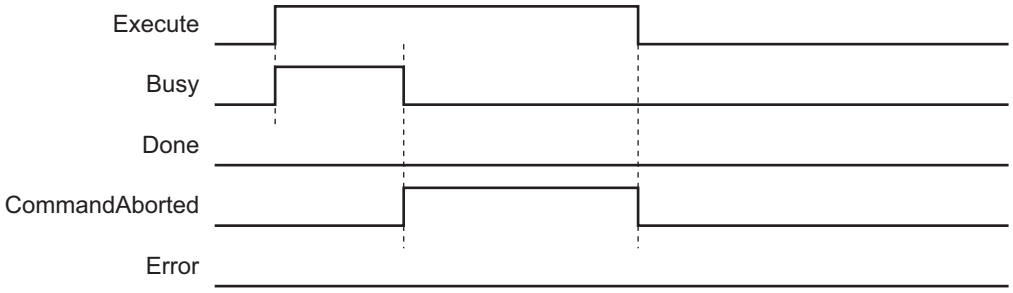
- 输入变量 Execute （启动）为 TRUE 状态，指令动作完成。
  之后，若在将 Execute （启动）设为 FALSE 前没有发生异常，情况如下图所示。



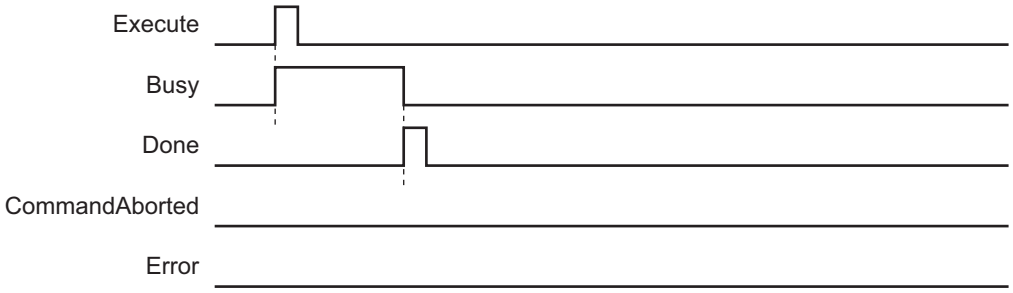
- 输入变量 Execute （启动）为 TRUE 状态，没有发生异常时，情况如下图所示。
  之后，即使将 Execute （启动）设为 FALSE，输出变量 Error （错误）仍保持 TRUE。



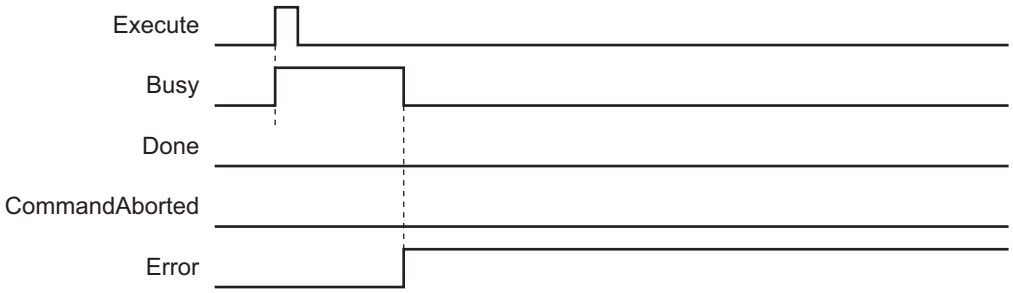
- 输入变量 Execute （启动）为 TRUE 状态，指令在中途中断执行时，情况如下图所示。



- 输入变量 Execute （启动）设为只在 1 个周期内变为 TRUE，指令没有发生异常时，情况如下图所示。
  指令动作完成后，输出变量 Done （完成）只在 1 个周期内变为 TRUE。

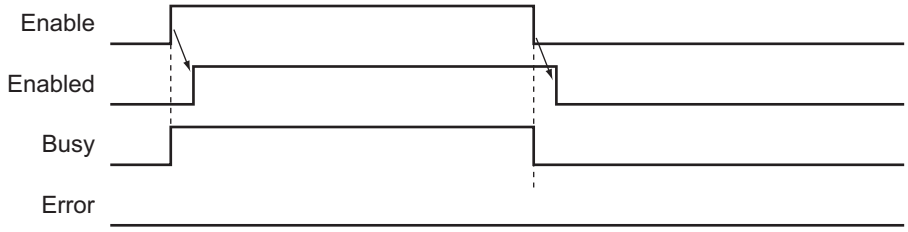


- 输入变量 Execute （启动）设为只在 1 个周期内变为 TRUE，指令发生异常时，情况如下图所示。
  输出变量 Error （错误）保持 TRUE。

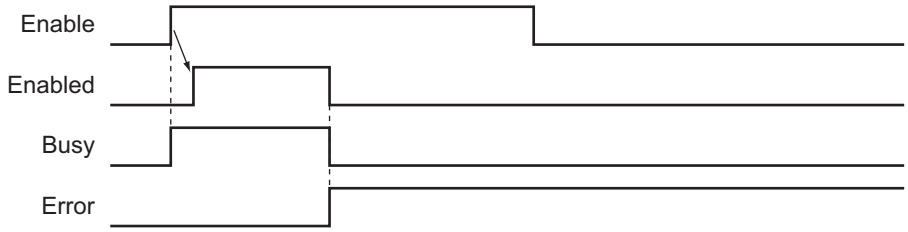


Enable 型的指令执行时序图

- 输入变量 “Enable （有效） ” 设为 TRUE，指令没有发生异常时，情况如下图所示。



- 输入变量 “Enable （有效） ” 设为 TRUE，指令发生异常时，情况如下图所示。



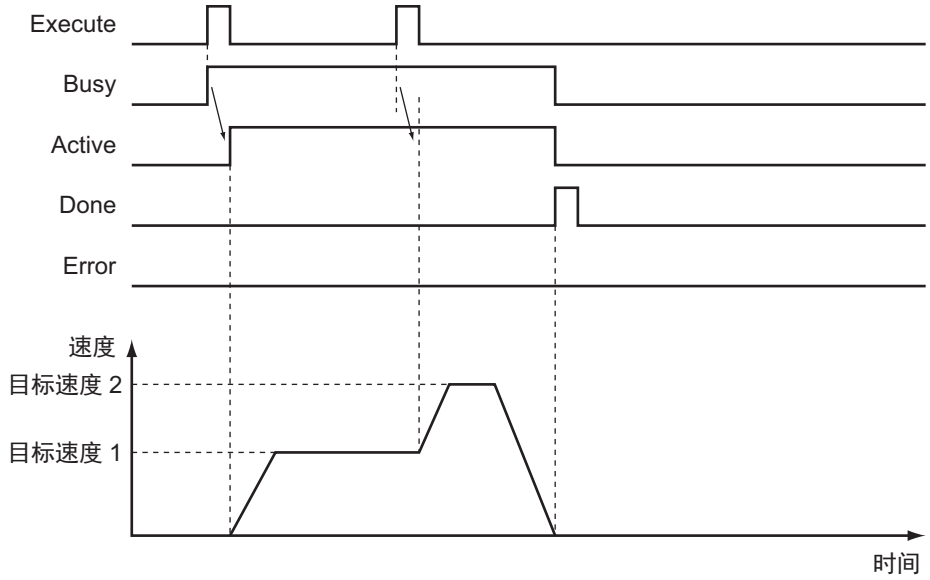
参考

还有 MC_ZoneSwitch （区域监视）指令或 MC_AxesObserve （轴间偏差监视）指令等，Enable 和 Enabled 时间相同的指令。各指令的时序详情请参考 《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。

6-4-3 运动控制指令重新执行时的时序

在执行运动控制指令的过程中，若变更同一个实例的输入变量值，输入变量 “Execute （启动） ” 依次按 TRUE ➞ FALSE ➞ TRUE 变化，将按变更后的输入变量值动作。

通过 MC_MoveAbsolute （绝对值定位）变更了速度后，情况如下图所示。

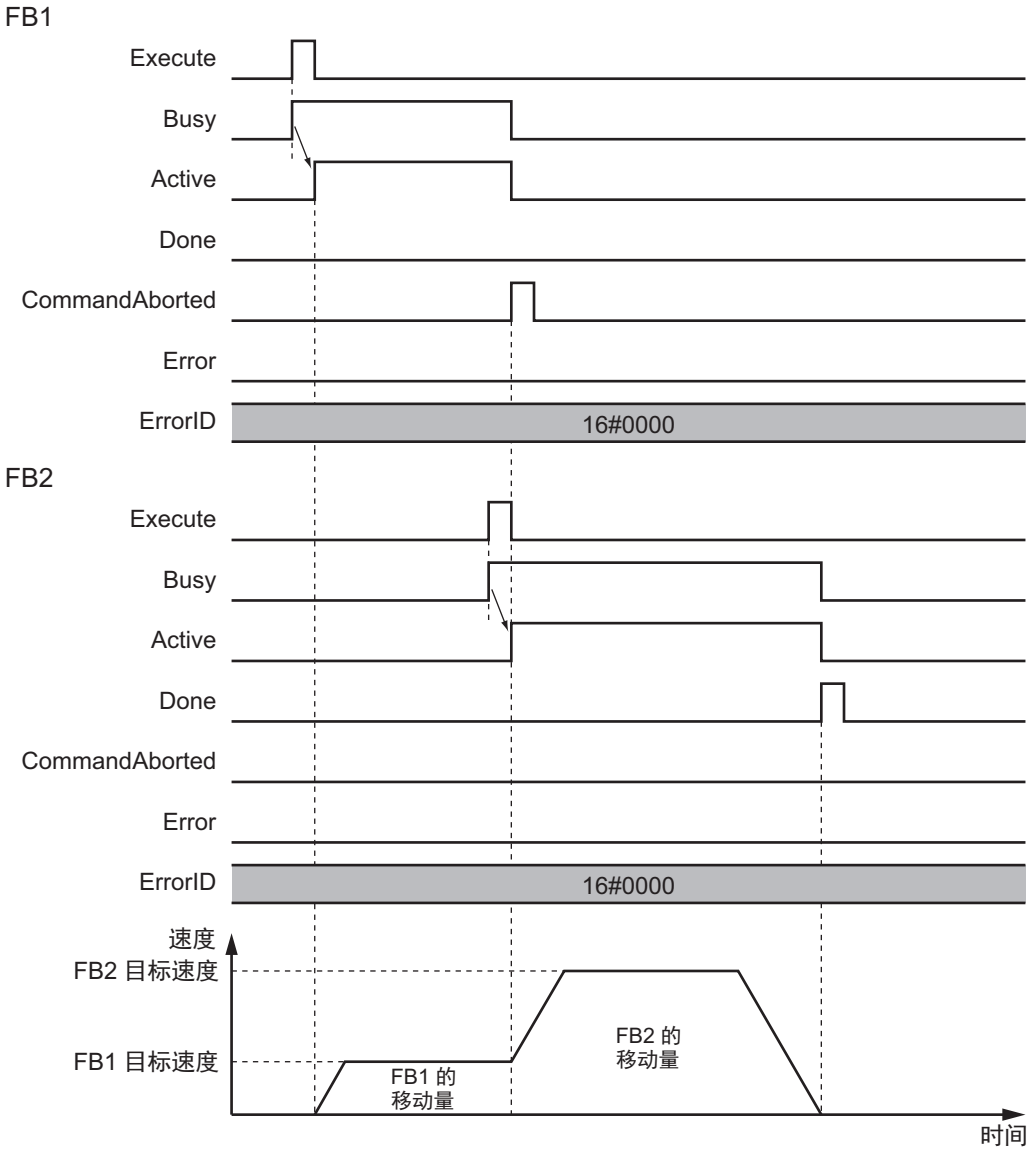


关于 MC 功能模块中的指令重新启动，请参考 「*9-5-6 运动控制指令的重启* (P.9-40)」、「*9-7-4 多轴协调控制时运动控制指令的重启* (P.9-62)」。

6-4-4 运动控制指令多重执行时的时序

可对正在动作的轴启动其他的实例。
通过设定输入变量 BufferMode （缓冲模式选择），可指定动作开始时间。

下图表示将 BufferMode （缓冲模式选择）设定为中断，多重启动 2 个 MC_MoveAbsolute （绝对值定位）指令的情况。
图中的 “FB1” 及 “FB2” 为指令的实例名。



关于 MC 功能模块中的指令多重启动，请参考 *「9-5-7 运动控制指令的多重启动 （缓冲模式）*(P.9-45)」、*「9-7-5 多轴协调控制时运动控制指令的多重启动 （缓冲模式）*(P.9-63)」。

6-5 位置处理

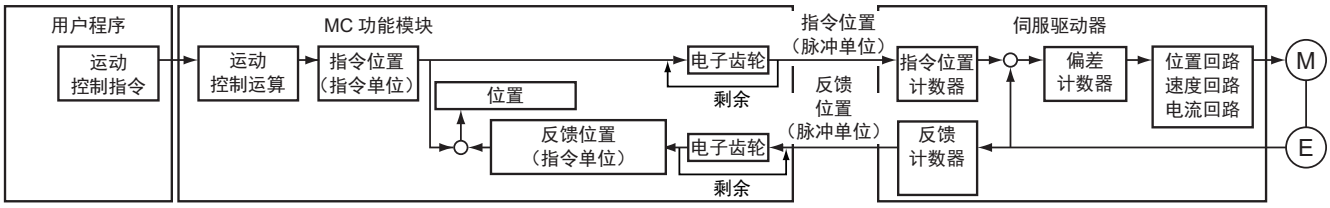
下面介绍运动控制程序中处理的位置。

6-5-1 位置的种类

MC 功能模块中有以下两类位置。

位置类别	定义
指令位置	MC 功能模块为了控制轴而输出的位置。
反馈位置	从伺服驱动器或计数器输入的实际位置。

EtherCAT 从站伺服驱动器时的指令位置和反馈位置的关系如下图所示。



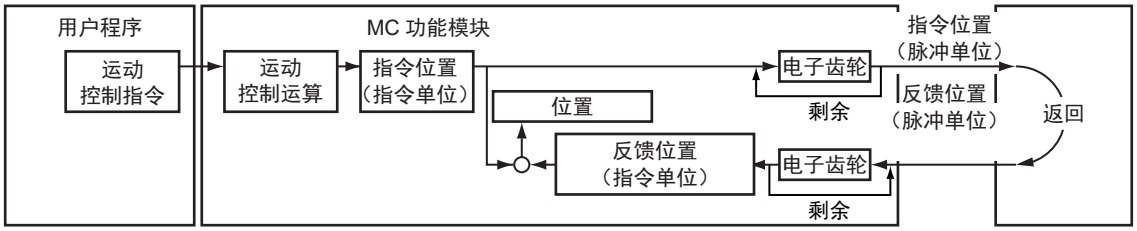
指令位置和反馈位置共享以下项目。

项目	指令位置	反馈位置
计数模式	设定直线模式或旋转模式。	变为与指令位置相同的计数模式。
位置的单位	设定为 mm、μm、nm、inch、degree、pulse 中的一个。	变为与指令位置相同的单位。
软件限位	设定软件的可动作范围。	变为与指令位置相同的范围。
当前位置变更	将当前位置变更为任意位置。	将同时设定与指令位置相同的值。*1
原点确定	保持原点确定或原点未确定状态。	变为与指令位置相同的原点状态。

*1. 若变更前有位置偏差，将在反馈位置中设定保存了位置变差的值。

参考

- 虚拟伺服轴时，先将指令当前值转换为脉冲单位，再将该值转换为显示单位，将得到的值作为反馈当前值。



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

6-5-2 各种轴类别可使用的位置类型

各种轴类别可使用的位置类型如下所示。

轴种类	位置的种类	
	指令位置	反馈位置
伺服轴	可使用	可使用
虚拟伺服轴	可使用	可使用 *1
编码器轴	不可使用	可使用
虚拟编码器轴	不可使用	可使用 *2

*1. 在虚拟伺服轴中，“反馈位置 ＝ 指令位置”。
在 MC 功能模块内部，以双精度实数型处理，因此可能包括运算误差。
*2. 不存在实际编码器时使用。

6-6 运动控制系统变量

下面介绍属于 MC 功能模块的变量。

6-6-1 运动控制系统变量的概要

NJ/NX 系列为符合 IEC 61131-3 标准的控制器。
在 NJ/NX 系列的用户程序中，参数设定、状态信息等数据作为变量处理。
在这些变量中，属于 MC 功能模块的系统定义变量称为 “ 运动控制系统变量 ”。

运动控制系统变量的种类

下表为运动控制系统变量的种类一览。

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

● MC 共通变量

监视 MC 功能模块共通状态的变量。变量名称为 “_MC_COM”。

● 轴变量

处理 EtherCAT 从站的伺服驱动器、编码器输入终端、 NX 系列位置接口单元及虚拟伺服驱动器或编码器输入的变量。
在用户程序中，无论是系统定义变量的变量名称，还是通过 Sysmac Studio 创建的变量名称，均可使用。
通过 Sysmac Studio 创建的变量名称可按不同的轴变更为任意变量名称。

a) 系统定义变量的变量名称如下所示。
- _MC_AX[0] ～ _MC_AX[255]
- _MC1_AX[0] ～ _MC1_AX[255]
- _MC2_AX[0] ～ _MC2_AX[255]

b) 通过 Sysmac Studio 创建的变量名称如下所示。
- MC_Axis000 ～ MC_Axis255 （默认）

● 轴组变量

处理整合了多个轴的轴组所需的变量。
在用户程序中，无论是系统定义变量的变量名称，还是通过 Sysmac Studio 创建的变量名称，均可使用。
通过 Sysmac Studio 创建的变量名称可按不同的轴组变更为任意变量名称。

a) 系统定义变量的变量名称如下所示。
- _MC_GRP[0] ～ _MC_GRP[63]
- _MC1_GRP[0] ～ _MC1_GRP[63]
- _MC2_GRP[0] ～ _MC2_GRP[63]

b) 通过 Sysmac Studio 创建的变量名称如下所示。
- MC_Group000 ～ MC_Group063 （默认）

NJ/NX 系列中处理的参数请参考 《NJ/NX 系列 CPU 单元 用户手册 软件篇 （SBCA-CN5-359）》。

运动控制系统变量中处理的数据类型

运动控制系统变量中包括基本数据型和派生数据型。

● 基本数据型

分类	数据类型	大小	值的范围	记述方法
布尔	BOOL	2 *1	FALSE、 TRUE	FALSE、 TRUE
整数	UINT	2	0 ～ +65535	2 进制表述时前面加 “2#”。*2 8 进制表述时前面加 “8#”。*3 10 进制表述时前面加 “10#”。*4 16 进制表述时前面加 “16#”。*5 若前面没有任何前缀，则解释为 10 进制数。
	UDINT	4	0 ～ +4294967295	
浮动小数点	LREAL	8	-1.79769313486231e+308 ～ -2.22507385850721e-308、 0、 2.22507385850721e-308 ～ 1.79769313486231e+308、 + ∞ /- ∞	按照 (符号)+ 整数部 +(小数点)+(小数部)+(指数部) 记述。*6 带 （ ）的可省略

*1. BOOL 型的数据大小为 1 位，但在存储器中会占用 2 字节。
*2. 2 进制表述示例：2#1111_1111、 2#1110_0000
*3. 8 进制表述示例：8#377、 8#340
*4. 10 进制表述示例：-12、 0、 123_456、 +986、 10#1234
*5. 16 进制表述示例：16#FF、 16#ff、 16#E0、 16#e0
*6. 例 : 2、 -12.0、 0.0、 0.4560、 3.14159_26、 -1.34E-12、 -1.34e-12、 1.0E+6、 1.0e+6、 1.234E6、 1.234e6

● 派生数据型

种类	内容
列举型	将事先准备的名称列表中的一个作为值的数据类型。 数据类型前面加 “_e”。
结构体型	将多个数据类型按层级整合为一个的数据类型。 数据类型前面加 “_s”。

NJ/NX 系列中处理的其他数据类型请参考 《NJ/NX 系列 CPU 单元 用户手册 软件篇（SBCA-CN5-359）》。

运动控制系统变量的属性

关于以下属性，运动控制系统变量全部为相同的属性。

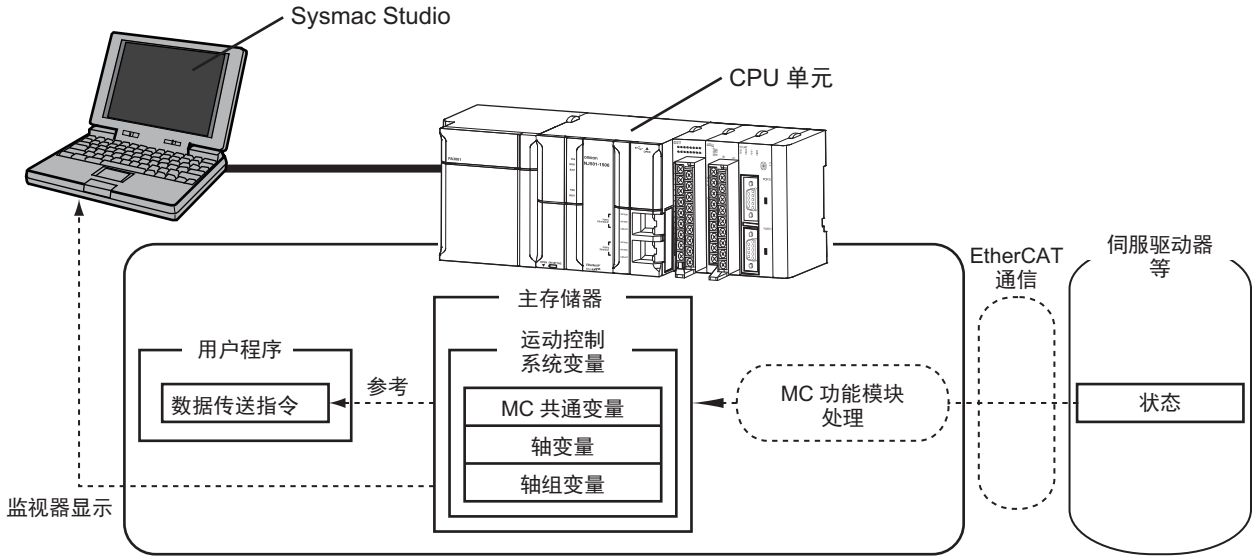
属性	运动控制系统变量的属性
全局 / 本地	全局变量
R/W 访问	只读
保持	非保持
网络公开	公开 *1

*1. 以系统定义变量的变量名称在网络上公开。通过 Sysmac Studio 创建轴或轴组时的变量名称不在网络上公开。

6-6-2 运动控制系统变量的原理

运动控制系统变量由表示 MC 功能模块状态的信息、通过 EtherCAT 通信连接的从站设备的状态信息以及执行运动控制所需的部分 MC 参数设定构成。

运动控制系统变量可通过用户程序作为变量参考，还可通过 Sysmac Studio 在显示器上显示。



运动控制系统变量的更新时间

在 NX 系列 CPU 单元和 NJ 系列 CPU 单元中，运动控制系统变量的更新时间有以下不同。

● NX 系列 CPU 单元时

在主固定周期任务和固定周期任务 （执行优先度 5）中分配了轴时，如下所示。

- 仅按主周期更新主固定周期任务中分配的任务区域和固定周期任务 （执行优先度 5）的基本设定区域。
- 固定周期任务 （执行优先度 5）中其他的区域保持初始值不更新，基本设定区域按固定周期任务（执行优先度 5）的任务周期更新。

更新对象的变量区域	主固定周期任务	固定周期任务 （执行优先度 5）
_MC_AX[0-255]	○	
_MC1_AX[0-255]	○	
_MC2_AX[0-255]		○
_MC_GRP[0-63]	○	
_MC1_GRP[0-63]	○	
_MC2_GRP[0-63]		○

● NJ 系列 CPU 单元时

运动控制系统变量按主周期更新。

6-6-3 运动控制系统变量的一览

下面以一览表的形式介绍不同种类的运动控制系统变量。

MC 共通变量

MC 共通变量是指变量名称为 _MC_COM、数据类型为 _sCOMMON_REF 型的结构体型变量。
下面介绍 MC 共通变量的构成及其成员的详情。

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

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

轴变量

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

下面以 _MC_AX[0-255] 为例，介绍轴变量的构成及其成员的详情。

_MC1_AX[0-255]、 _MC2_AX[0-255] 亦同。

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

变量名称		数据类型	名称	功能
	ErrorStop	BOOL	错误减速停止中	MC_ImmediateStop 指令或轴 发生轻度故障 (_MC_AX[*].MFaultLvl.Active) 为 “TRUE: 轴 发生轻度故障 ” 时，轴停止前和停止的状态。 在此状态下，无法启动轴动作指令。(CommandAborted=TRUE)
	Coordinated	BOOL	多轴协调动作中	因多轴协调指令，轴组处于有效状态时变为 TRUE。
Details		_sAXIS_REF_DET	轴控制状态	表示指令的控制状态。
	Idle	BOOL	停止中	除位置等待外，未计算指令值时变为 TRUE。*1 与 Idle 和 InPosWaiting 互斥，因此不会同时变为 TRUE。
	InPosWaiting	BOOL	位置等待	位置等待时变为 TRUE。 在带位置检查的定位中，正在进行位置检查的状态。
	Homed	BOOL	原点确定	原点确定状态时变为 TRUE。*2 FALSE: 原点未确定 TRUE: 原点确定
	InHome	BOOL	原点停止	在原点位置的范围内时为 TRUE。 以下的 AND 条件。 • 原点确定状态 • 反馈当前位置进入以原点为中心的 “ 原点位置范围 ” 内时 在指令状态下轴正在动作时，若通过原点位置，也会变为 TRUE。
	VelLimit *3	BOOL	指令速度饱和	同步控制中指令速度受最高速度限制时变为 TRUE。
Dir		_sAXIS_REF_DIR	指令方向状态	表示指令的移动方向。
	Posi	BOOL	正方向指令中	正在向正方向指令时变为 TRUE。
	Nega	BOOL	负方向指令中	正在向负方向指令时变为 TRUE。
DrvStatus		_sAXIS_REF_STA_DRV	伺服驱动器状态	表示驱动器等的状态。
	ServoOn	BOOL	伺服 ON	伺服电机通电时为 TRUE。
	Ready	BOOL	伺服就绪	伺服就绪状态*4 时变为 TRUE。
	MainPower	BOOL	主回路电源	伺服驱动器的主回路电源通电时为 TRUE。
	P_OT	BOOL	正方向极限输入	正方向极限输入为有效时变为 TRUE。
	N_OT	BOOL	负方向极限输入	负方向极限输入为有效时变为 TRUE。
	HomeSw	BOOL	近原点输入	近原点输入为有效时变为 TRUE。
	Home	BOOL	原点输入	原点输入为有效时变为 TRUE。*5 *6
	ImdStop	BOOL	立即停止输入	立即停止输入为有效时变为 TRUE。

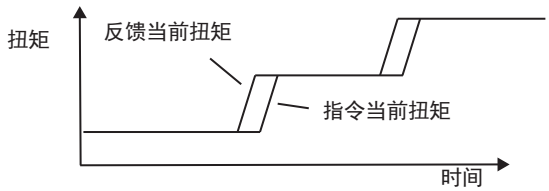
变量名称	数据类型	名称	功能
Latch1	BOOL	外部锁定输入 1	锁定输入 1 为有效时变为 TRUE。
Latch2	BOOL	外部锁定输入 2	锁定输入 2 为有效时变为 TRUE。
DrvAlarm	BOOL	驱动器错误输入	发生驱动器错误时变为 TRUE。
DrvWarning	BOOL	驱动器警告输入	发生驱动器警告时变为 TRUE。
ILA	BOOL	驱动器内部功能限制中	正在利用伺服驱动器的限制功能实际进行限制时变为 TRUE。*7
CSP	BOOL	周期性同步位置 (CSP) 模式中	伺服驱动器为伺服 ON、处于 CSP 模式时变为 TRUE。*8
CSV	BOOL	周期性同步速度 (CSV) 模式中	伺服驱动器为伺服 ON、处于 CSV 模式时变为 TRUE。*5
CST	BOOL	周期性同步扭矩 (CST) 模式中	伺服驱动器为伺服 ON、处于 CST 模式时变为 TRUE。*5
Cmd	_sAXIS_REF_CMD_DATA	轴指令值	
Pos	LREAL	指令当前位置	输出指令位置的当前值。（单位：指令单位） 伺服 OFF 时及位置控制模式以外时，输出反馈当前位置。*9
Vel	LREAL	指令当前速度	输出指令速度的当前值。（单位：指令单位 /s） 正方向移动时加 + 号，负方向移动时加 - 号。 根据指令当前位置的差计算得到的速度。 伺服 OFF 时及位置控制模式以外时，根据反馈当前位置计算得到的速度。
AccDec	LREAL	指令当前加减速度	输出指令加减速度的当前值。（单位：指令单位 /s^2） 根据指令当前速度的差计算得到的加减速度。加速时加 + 号，减速时加 - 号。 执行中的指令加减速度为 “0” 时，变为 “0”。
Jerk	LREAL	指令当前跃度	输出指令跃度的当前值。（单位：指令单位 /s^3） 加减速度的绝对值增加时加 + 号，减少时加 - 号。 执行中的指令加减速度、指令跃度为 “0” 时，变为 “0”。
Trq	LREAL	指令当前扭矩	输出指令扭矩的当前值。（单位：%） 正方向移动时加 + 号，负方向移动时加 - 号。 扭矩控制模式以外时，输出与反馈当前扭矩相同的值。*10

变量名称	数据类型	名称	功能
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）*9
Vel	LREAL	反馈当前速度	输出反馈速度的当前值。（单位：指令单位 /s） 正方向移动时加 + 号，负方向移动时加 - 号。
Trq	LREAL	反馈当前扭矩	输出反馈扭矩的当前值。（单位：%） 正方向移动时加 + 号，负方向移动时加 - 号。
TimeStamp *11	ULINT	时间戳	输出轴当前位置的更新时刻。对变化时刻获取功能正在运行的轴有效。（单位：ns）
MFaultLvl	_sMC_REF_EVENT	轴 轻度故障	
Active	BOOL	轴 发生轻度故障	发生轴的轻度故障时变为 TRUE。
Code	WORD	轴 轻度故障代码	输出轴的轻度故障代码。 与事件代码的前面 4 位相同的值。
Obsr	_sMC_REF_EVENT	轴 监视信息	
Active	BOOL	轴 监视信息发生中	发生轴的监视信息时变为 TRUE。
Code	WORD	轴 监视信息代码	输出轴的监视信息代码。 与事件代码的前面 4 位相同的值。
Cfg	_sAXIS_REF_CFG	轴基本设定	表示轴基本设定的设定内容。
AxNo	UINT	轴编号	表示轴的逻辑编号。
AxEnable	_eMC_AXIS_USE	轴使用	表示是否使用轴。 0: _mcNoneAxis （未创建轴） 1: _mcUnusedAxis （未使用轴） 2: _mcUsedAxis （使用轴）
AxType	_eMC_AXIS_TYPE	轴种类	表示轴种类。 虚拟轴不需要输入输出配线。 0: _mcServo （伺服轴） 1: _mcEncdr （编码器轴） 2: _mcVirServo （虚拟伺服轴） 3: _mcVirEncdr （虚拟编码器轴）
NodeAddress	UINT	节点地址	表示 EtherCAT 的从站地址。*12 16#FFFF 表示无。
ExecID *13	UINT	执行 ID	表示各任务的执行 ID。 0: 未分配任务 （未创建轴） 1: 分配到主固定周期任务 2: 分配到固定周期任务 （执行优先度 5）
Scale	_sAXIS_REF_SCALE	单位转换	表示电子齿轮比的设定值。
Num	UDINT	电机每转 1 圈的脉冲数	表示指令位置的电机每转 1 圈的脉冲数。 根据电子齿轮比，从指令值转换为脉冲量。
Den	LREAL	电机每转 1 圈的移动量	表示指令位置的电机每转 1 圈的工件移动量。*14

变量名称		数据类型	名称	功能
	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 *13	_eMC_COUNT_MODE	计数模式	表示计数模式。 0: _mcCountModeLinear （直线模式） 1: _mcCountModeRotary （旋转模式）
	MaxPos *13	LREAL	当前位置上限值	表示当前位置显示的上限值。*15
	MinPos *13	LREAL	当前位置下限值	表示当前位置显示的下限值。*16

*1. 以速度 “0” 动作、偏差计数器复位中、同步控制中及多轴协调动作中也视为正在运算的状态。
*2. 即使已变为 TRUE，在以下情况下，仍需重新确定原点。
变更了位置计数设定或单位转换设定时。
伺服驱动器上发生绝对位置信息丢失的异常或执行了此类操作时。例如，编码器电缆断线或清除了绝对编码器数据等。
*3. VelLimit 只能用于同步控制中的从轴。
*4. 伺服驱动器的 PDS 状态为 “Ready to switch on”、 “Switched on”、 “Operation enabled” 之一，且主回路电源 (voltage enabled) 为 ON 时变为 TRUE。但是，若［主回路电源 OFF 检测］设定为［不检测］，将忽略主回路电源的 ON/OFF。PDS 状态、主回路电源 OFF 检测的详情请参考 🕮 *「A-4 PDS 状态变化* (P.A-19)」。
*5. 在 Sysmac Studio 的［轴基本设定］画面的［详细设定］中，［数字输入］的［编码器 Z 相检测］中设定的信号的状态。在其他公司生产的驱动器中，此信号可能无法映射到 PDO 中。请确认所连接驱动器的手册。
*6. 在欧姆龙产伺服驱动器 G5 系列 EtherCAT 通信内置直线电机型中，无法将此信号映射到 PDO 中。
*7. 显示 PDO 中映射的 “ 状态字 (6041Hex)” 第 11 位 （内部限制功能有效）的状态。变为 TRUE 的条件取决于伺服驱动器的规格。请参考所连接伺服驱动器的手册。在欧姆龙生产的伺服驱动器 G5 系列中，为扭矩限制 / 速度限制 / 驱动禁止输入 / 软件限位之一。
*8. 根据 PDO 中映射的 “ 状态字 (6061Hex)” 的值显示。CSP、CSV、CST 变为 TRUE 的条件取决于伺服驱动器的规格。请参考所连接伺服驱动器的手册。
PDO 中未映射 “ 操作模式显示 (6061Hex)” 时，始终为 FALSE。
*9. 轴中分配的 EtherCAT 从站或 NX 单元与 CPU 单元尚未建立过程数据通信时，不同 CPU 单元版本中的情况如下。
CPU 单元版本 1.09 以下时，轴变量的反馈当前位置及指令当前位置为 “0” 或下限值。变为下限值的条件：计数模式为旋转模式，位置范围中不包含 “0” 时。
在单元版本 1.10 以上的 CPU 单元中，轴变量的反馈当前位置及指令当前位置为过程数据通信未建立前的反馈当前位置。
*10.在数据追踪中比较显示指令当前扭矩和反馈当前扭矩时，除了扭矩控制模式外，显示的指令当前扭矩将在 1 个任务周期后变化。这是因为受到数据追踪处理时间的影响。数据追踪的处理时间请参考 🕮《NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-CN5-359)》的数据追踪记述。

数据追踪画面示例



指令当前扭矩的变化会晚于反馈当前扭矩。

*11.可在 CPU 单元版本 1.06 以上和 Sysmac Studio Ver.1.07 以上的组合中使用。
*12.NX 系列位置接口单元时，表示安装了位置接口单元的 EtherCAT 耦合器单元的节点地址。
*13.可在 CPU 单元版本 1.10 以上和 Sysmac Studio Ver.1.12 以上的组合中使用。
*14.单元版本 1.11 以上的 CPU 单元中，将追加的单位转换设定的减速机使用设定为 “ 使用 ” 时，该参数无效。确认有效参数 “ 工件侧每转 1 圈的移动量 ”、 “ 工件侧齿轮比 ”、 “ 电机侧齿轮比 ” 的值时，请使用 MC_ReadAxisParameter （轴参数读取）指令。
*15.计数模式为直线模式时，显示即将溢出时的位置。为旋转模式时，显示循环计数器上限值。
*16.计数模式为直线模式时，显示即将下溢时的位置。为旋转模式时，显示循环计数器下限值。

● 轴变量和轴种类的关系

不同轴种类中轴变量的有效 / 无效成员不同。若参照无效的成员，将读取为 “FALSE” 或 “0”。

下面以 _MC_AX[0-255] 为例进行说明。 _MC1_AX[0-255]、 _MC2_AX[0-255] 亦同。此外，不同轴种类中各成员的有效 / 无效在一览中记载。

○：有效、 —：无效

变量名称	数据类型	名称	伺服轴	虚拟伺服轴	编码器轴	虚拟编码器轴
_MC_AX[0-255]	_sAXIS_REF	轴变量				
Status	_sAXIS_REF_STA	轴状态				
Ready	BOOL	轴启动准备完成	○	○	—	—
Disabled	BOOL	轴无效	○	○	○	○
Standstill	BOOL	停止中	○	○	—	—
Discrete	BOOL	定位动作中	○	○	—	—
Continuous	BOOL	连续动作中	○	○	—	—
Synchronized	BOOL	同步动作中	○	○	—	—
Homing	BOOL	原点复位中	○	○	—	—
Stopping	BOOL	减速停止中	○	○	—	—
ErrorStop	BOOL	错误减速停止中	○	○	—	—
Coordinated	BOOL	多轴协调动作中	○	○	—	—
Details	_sAXIS_REF_DET	轴控制状态				
Idle	BOOL	停止中	○	○	○	○
InPosWaiting	BOOL	位置等待	○	○	—	—
Homed	BOOL	原点确定	○	○	—	—
InHome	BOOL	原点停止	○	○	—	—
VelLimit	BOOL	指令速度饱和	○	○	—	—
Dir	_sAXIS_REF_DIR	指令方向状态				
Posi	BOOL	正方向指令中	○	○	—	—
Nega	BOOL	负方向指令中	○	○	—	—
DrvStatus	_sAXIS_REF_STA_DRV	伺服驱动器状态				
ServoOn	BOOL	伺服 ON	○	—	—	—
Ready	BOOL	伺服就绪	○	—	—	—
MainPower	BOOL	主回路电源	○	—	—	—
P_OT	BOOL	正方向极限输入	○	—	—	—
N_OT	BOOL	负方向极限输入	○	—	—	—
HomeSw	BOOL	近原点输入	○	—	—	—
Home	BOOL	原点输入	○	—	—	—
ImdStop	BOOL	立即停止输入	○	—	—	—
Latch1	BOOL	外部锁定输入 1	○	—	—	—
Latch2	BOOL	外部锁定输入 2	○	—	—	—
DrvAlarm	BOOL	驱动器错误输入	○	—	—	—

变量名称	数据类型	名称	伺服轴	虚拟伺服轴	编码器轴	虚拟编码器轴
DrvWarning	BOOL	驱动器警告输入	○	—	—	—
ILA	BOOL	驱动器内部功能限制中	○	—	—	—
CSP	BOOL	周期性同步位置 (CSP) 模式中	○	—	—	—
CSV	BOOL	周期性同步速度 (CSV) 模式中	○	—	—	—
CST	BOOL	周期性同步扭矩 (CST) 模式中	○	—	—	—
Cmd	_sAXIS_REF_CMD_DATA	轴指令值				
Pos	LREAL	指令当前位置	○	○	—	—
Vel	LREAL	指令当前速度	○	○	—	—
AccDec	LREAL	指令当前加减速度	○	○	—	—
Jerk	LREAL	指令当前跃度	○	○	—	—
Trq	LREAL	指令当前扭矩	○	○	—	—
Act	_sAXIS_REF_ACT_DATA	轴当前值				
Pos	LREAL	反馈当前位置	○	○	○	○
Vel	LREAL	反馈当前速度	○	○	○	○
Trq	LREAL	反馈当前扭矩	○	○	—	—
TimeStamp	ULINT	时间戳	○	—	○	—
MFaultLvl	_sMC_REF_EVENT	轴 轻度故障				
Active	BOOL	轴 发生轻度故障	○	○	○	○
Code	WORD	轴 轻度故障代码	○	○	○	○
Obsr	_sMC_REF_EVENT	轴 监视信息				
Active	BOOL	轴 监视信息发生中	○	○	○	○
Code	WORD	轴 监视信息代码	○	○	○	○
Cfg	_sAXIS_REF_CFG	轴基本设定				
AxNo	UINT	轴编号	○	○	○	○
AxEnable	_eMC_AXIS_USE	轴使用	○	○	○	○
AxType	_eMC_AXIS_TYPE	轴种类	○	○	○	○
NodeAddress	UINT	节点地址	○	—	○	—
ExecID *1	UINT	执行 ID	○	○	○	○
Scale	_sAXIS_REF_SCALE	单位转换				
Num	UDINT	电机每转 1 圈的脉冲数	○	○	○	○
Den *2	LREAL	电机每转 1 圈的移动量	○	○	○	○
Units	_eMC_UNITS	显示单位	○	○	○	○
CountMode *1	_eMC_COUNT_MODE	计数模式	○	○	○	○
MaxPos *1	LREAL	当前位置上限值	○	○	○	○
MinPos *1	LREAL	当前位置下限值	○	○	○	○

*1. 可在 CPU 单元版本 1.10 以上和 Sysmac Studio Ver.1.12 以上的组合中使用。
*2. 单元版本 1.11 以上的 CPU 单元中，将追加的单位转换设定的减速机使用设定为 “ 使用 ” 时，该参数无效。

轴组变量

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

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

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

变量名称	数据类型	名称	功能
_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	轴组处于停止状态、已经完成启动准备时变为 TRUE。 以下的 AND 条件为启动准备完成的条件。 • 构成轴未执行 MC_Stop 指令。 • _MC_GRP[*].Status.Standby 为 TRUE （停止中） • 构成轴为伺服 ON 状态 • 构成轴的 _MC_AX[*].Details.Homed 为 TRUE （原点确定）
Disabled	BOOL	轴组无效	轴组无效、处于停止状态时变为 TRUE。 与以下轴组状态互斥，只有一个会变为 TRUE。 Disabled/ Standby/ Moving/ Stopping/ ErrorStop
Standby	BOOL	停止中	轴组动作指令处于停止状态时变为 TRUE。 与轴组构成轴的伺服 ON/OFF 状态无关。
Moving	BOOL	动作中	正在向目标位置执行轴组动作指令时变为 TRUE。 包括位置等待的状态或因超驰而使速度变为 “0” 的状态。
Stopping	BOOL	减速停止中	利用 MC_GroupStop 指令停止轴组时，轴组完全停止前为 TRUE。 包括利用 MC_GroupStop 指令停止轴后，Execute 为 TRUE 时。 在本状态下，无法启动轴组动作指令。(CommandAborted=TRUE)
ErrorStop	BOOL	错误减速停止中	因 MC_GroupImmediateStop 指令或轴组发生轻度故障 (_MC_GRP[*].MFaultLvl.Active=TRUE)，轴组停止前及停止时变为 TRUE。在此状态下，无法启动轴组动作指令。(CommandAborted=TRUE)

变量名称	数据类型	名称	功能
Details	_sGROUP_REF_DET	轴组控制状态	表示指令的控制状态。
Idle	BOOL	停止中	除位置等待外，未计算指令值时变为 TRUE。*1 与 Idle 和 InPosWaiting 互斥，因此不会同时变为 TRUE。
InPosWaiting	BOOL	位置等待	任一构成轴处于位置等待时变为 TRUE。 在带位置检查的定位中，正在进行位置检查的状态。*2
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的差计算得到的插补速度。正方向移动时加 + 号，负方向移动时加 - 号。 轴组为无效时，变为 “0”。
AccDec	LREAL	指令插补加减速度	输出指令插补加减速度的当前值。 根据指令插补速度的差计算得到的插补加减速度。加速时加 + 号，减速时加 - 号。 轴组为无效时以及执行中的轴组动作指令加减速度为 “0” 时，变为 “0”。
MFaultLvl	_sMC_REF_EVENT	轴组 轻度故障	
Active	BOOL	轴组 发生轻度故障	发生轴组轻度故障时变为 TRUE。
Code	WORD	轴组 轻度故障代码	输出轴组轻度故障代码。 与事件代码的前面 4 位相同的值。
Obsr	_sMC_REF_EVENT	轴组 监视信息	
Active	BOOL	轴组 监视信息发生中	发生轴组的监视信息时变为 TRUE。
Code	WORD	轴组 监视信息代码	输出轴组的监视信息代码。 与事件代码的前面 4 位相同的值。
Cfg	_sGROUP_REF_CFG	轴组基本设定	表示轴组基本设定的设定内容。
GrpNo	UINT	轴组编号	表示轴组的逻辑编号。
GrpEnable	_eMC_GROUP_USE	轴组使用	表示是否使用轴组。 0: _mcNoneGroup （未创建轴组） 1: _mcUnusedGroup （未使用轴组） 2: _mcUsedGroup （使用轴组）
ExecID *3	UINT	执行 ID	表示已分配的各任务的执行 ID。 0: 未分配任务 （未创建组） 1: 分配到主固定周期任务 2: 分配到固定周期任务 （执行优先度 5）

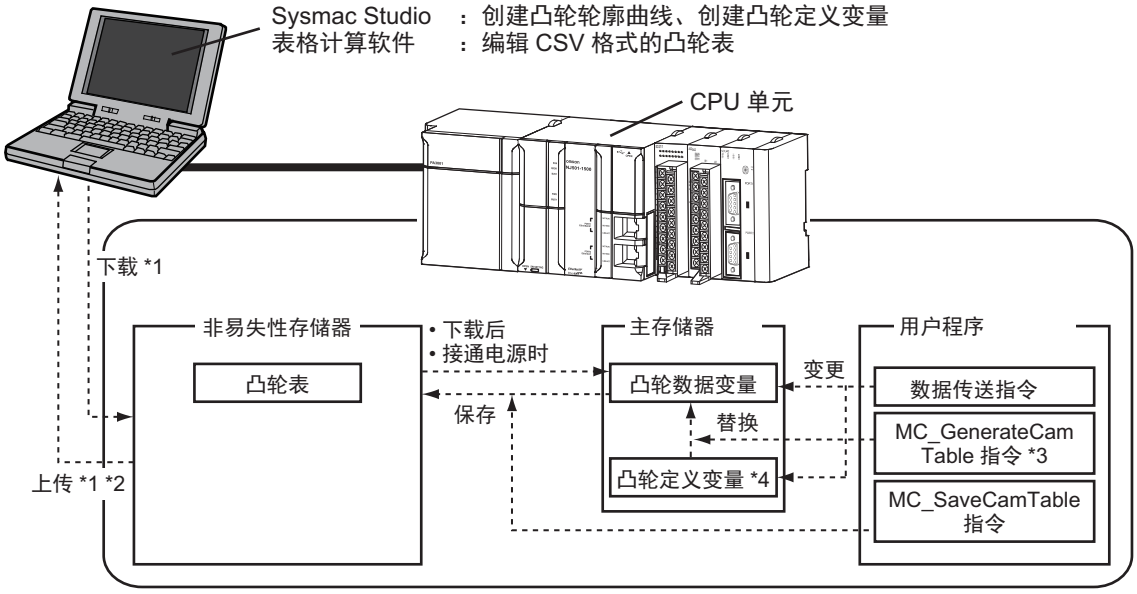
变量名称	数据类型	名称	功能
Kinematics	_sGROUP_REF_KIM	运动学转换设定	表示轴组的运动学转换所需的定义。
GrpType	_eMC_TYPE	轴构成	表示多轴协调控制中的轴构成。 0: _mcXY （2 轴） 1: _mcXYZ （3 轴） 2: _mcXYZU （4 轴）
Axis[0]	UINT	构成轴 （轴 A0）	表示轴 A0 中分配的轴编号。
Axis[1]	UINT	构成轴 （轴 A1）	表示轴 A1 中分配的轴编号。
Axis[2]	UINT	构成轴 （轴 A2）	表示轴 A2 中分配的轴编号。
Axis[3]	UINT	构成轴 （轴 A3）	表示轴 A3 中分配的轴编号。

*1. 以速度 “0” 动作时也视为正在运算的状态。
*2. 轴组中的所有构成轴进入各轴参数的位置宽度范围后，变为 FALSE。
*3. 可在 CPU 单元版本 1.10 以上和 Sysmac Studio Ver.1.12 以上的组合中使用。

6-7 凸轮表和凸轮数据变量

MC 功能模块中，可将通过 Sysmac Studio 的凸轮编辑器功能创建的凸轮轮廓曲线作为凸轮表使用。在 NJ/NX 系列的用户程序中，凸轮表的数据作为凸轮数据变量处理。

凸轮表的创建和保存原理



*1. “ 下载 ” 和 “ 上传 ” 时使用 Sysmac Studio 的 “ 同步 ” 功能。

*2. 已上传的凸轮数据变量无法通过凸轮编辑器功能编辑。如何在计算机上重新编辑用户程序中编辑过的凸轮数据变量，请参考下一页。
此外，若在 Sysmac Studio 中重建或编辑凸轮轮廓曲线，上传的凸轮数据变量会被凸轮轮廓曲线的数据覆盖。

*3. MC_GenerateCamTable （凸轮表生成）指令可在 CPU 单元版本 1.08 以上和 Sysmac Studio Ver.1.09 以上的组合中使用。

*4. 凸轮定义变量为凸轮属性变量和凸轮节点变量的总称。

● 凸轮表的数据流程

- 若将 Sysmac Studio 中创建的凸轮轮廓曲线下载到 CPU 单元，将作为凸轮表保存到 CPU 单元内置的非易失性存储器中。此外，若向 Sysmac Studio 上传凸轮表，将上传非易失性存储器中保存的凸轮表。
- 非易失性存储器中保存的凸轮表会在下载后或接通电源时作为凸轮数据变量反映到主存储器中。
- 通过用户程序，可编辑主存储器中的凸轮数据变量及凸轮定义变量。关于凸轮数据变量及凸轮定义变量，请参考 🕮 「*9-2-5 凸轮表* (P.9-15)」。
- 用户程序中的 MC_GenerateCamTable （凸轮表生成）指令可按照凸轮定义变量的值，替换主存储器的凸轮数据变量。
- 运动控制指令的 MC_SaveCamTable （凸轮表保存）指令将把主存储器的凸轮数据变量保存到非易失性存储器中。
- Sysmac Studio 的凸轮设定中创建的定义变量即使在用户程序中发生了变更，也可上传。作为用户定义变量创建的定义变量若在用户程序中发生了变更，将无法上传。

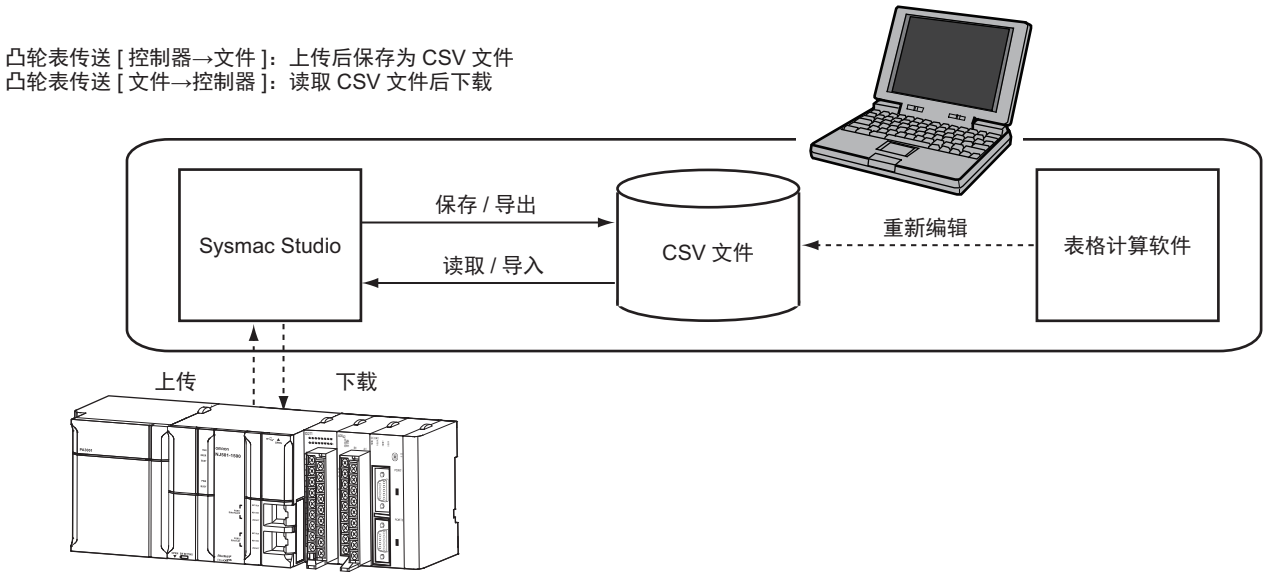
- 关于通过 Sysmac Studio 进行凸轮定义变量的创建和传送，请参考 📖《Sysmac Studio Version 1 操作手册 （SBCA-CN5-362K 以后）》。
- MC_GenerateCamTable （凸轮表生成）指令及 MC_SaveCamTable （凸轮表保存）指令的详情请参考 📖《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。
- 凸轮表的下载及上传与 CPU 单元的模式或 MC 功能模块的状态无关，均可执行。但是，在凸轮表保存过程中，除了凸轮数据的下载及上传外，在线开始操作、在线编辑、数据追踪开始的操作也将无法使用。此外，正在进行在线编辑等时，无法使用 MC_SaveCamTable （凸轮表保存）指令。
- 开始下载后，动作中的轴将以最大减速度减速，伺服变为 OFF。

使用注意事项

用户程序中变更过的凸轮数据在重新接通电源或从 Sysmac Studio 下载后，将恢复为非易失性存储器的凸轮表内容。此外，不可通过 Sysmac Studio 上传这些数据。

在计算机上重新编辑用户程序中编辑过的凸轮数据变量

通过用户程序编辑或替换凸轮数据变量，并通过 MC_SaveCamTable （凸轮表保存）指令保存到非易失性存储器中的凸轮表无法在 Sysmac Studio 的凸轮编辑器功能中重新编辑数据。下面介绍用表格计算软件重新编辑数据并作为凸轮表使用的方法。



*1. “ 下载 ” 和 “ 上传 ” 时使用 Sysmac Studio 的 “ 同步 ” 功能。

● 将非易失性存储器中保存的凸轮表保存为 CSV 文件

- 在 Sysmac Studio 的 ［凸轮数据设定］中右击要重新编辑的凸轮轮廓，从菜单中选择 ［凸轮表传送 ［控制器➡文件］］。
- 在 ［保存］对话框中指定文件的保存位置和文件名，然后单击 ［保存］按钮。

● 重新编辑 CSV 文件

- 使用表格计算软件编辑数据，并保存为 CSV 文件。

● 将 CSV 文件传送到 CPU 单元

- 右击要下载的凸轮轮廓，从菜单中选择［凸轮表传送［文件→控制器］］。
- 在［打开文件］对话框中指定要传送的文件名，然后单击［OK］按钮。
- 为了使传送的凸轮表有效，传送完成后，请重置控制器或重新接通控制器的电源。

使用注意事项

- 执行凸轮表传送［文件→控制器］前，请与控制器同步。
- 通过凸轮表传送［文件→控制器］传送凸轮表后，若通过同步功能将凸轮数据设定传送到控制器，将变更为凸轮数据设定的数据。
  此时，请再次通过凸轮表传送［文件→控制器］来传送，或将凸轮数据设定排除在同步对象之外。

通过凸轮编辑器创建的凸轮数据设定可导出为 CSV 文件。

关于凸轮数据设定和操作方法的详情，请参考 🕮《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

凸轮轮廓曲线的名称

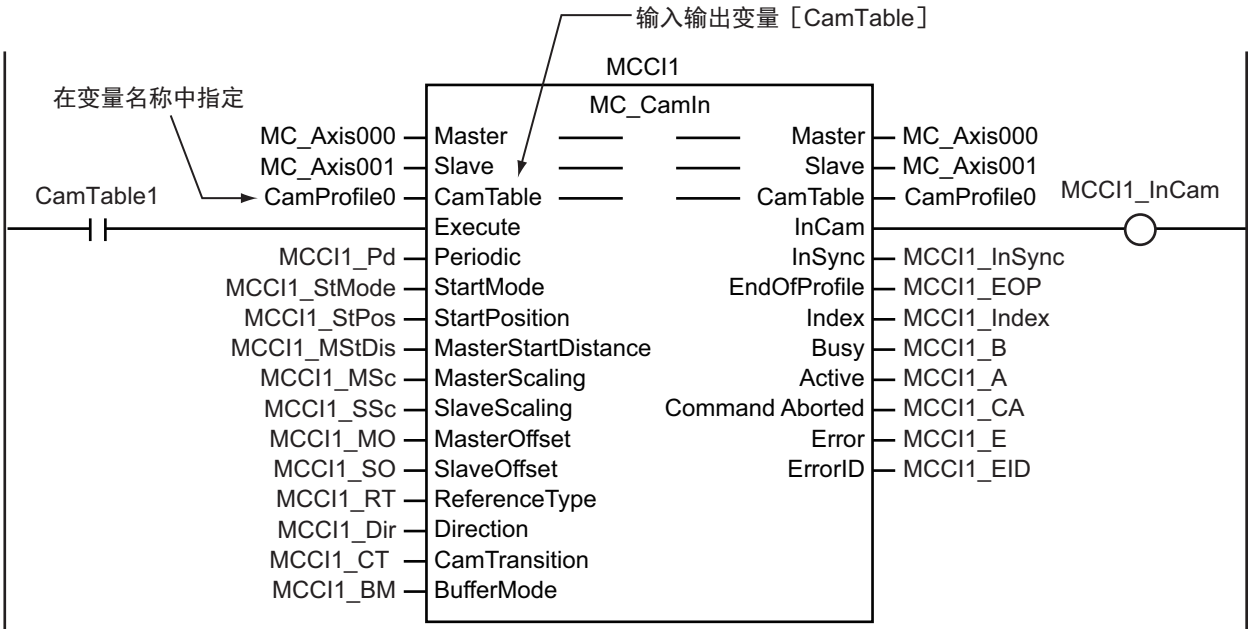
通过 Sysmac Studio 追加凸轮轮廓后，名称默认设定为 “CamProfile0”。末尾的数字在每次追加凸轮轮廓时不断增大。

这些名称可在 Sysmac Studio 中按不同的凸轮轮廓变更为任意名称。

这里设定的凸轮轮廓名称将视为凸轮表名称。

用户程序中的指定方法

在用户程序中，对运动控制指令的输入输出变量 “CamTable” 指定凸轮表名称。



6-8　运动控制程序的创建方法

通过在 NJ/NX 系列的用户程序中记述运动控制指令，可执行运动控制。记述了运动控制指令的程序称为运动控制程序。

使用注意事项

- 在 Sysmac Studio 中，无论哪个型号的 CPU 单元，最多可对 256 个轴进行设定和编程。但是，若项目超出 CPU 型号的最大控制轴数，将无法下载到 CPU 单元。
- 流用项目时，请确认 CPU 单元的型号，确保不会超出最大控制轴数。
- 轴使用为［未使用轴］的轴也在控制轴数对象内。

编程的详情请参考 🕮《NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-CN5-359)》。

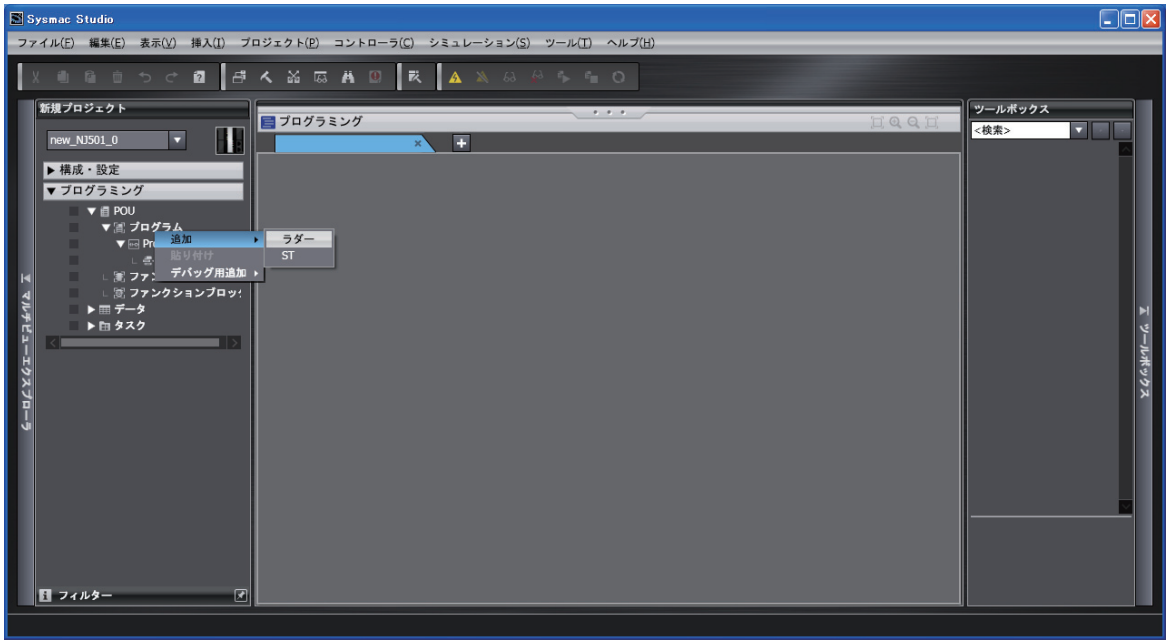
下面介绍使用 Sysmac Studio 在已有的项目中追加程序的步骤。

1 **启动 Sysmac Studio**

启动 Sysmac Studio，打开已有项目。

2 **追加程序**

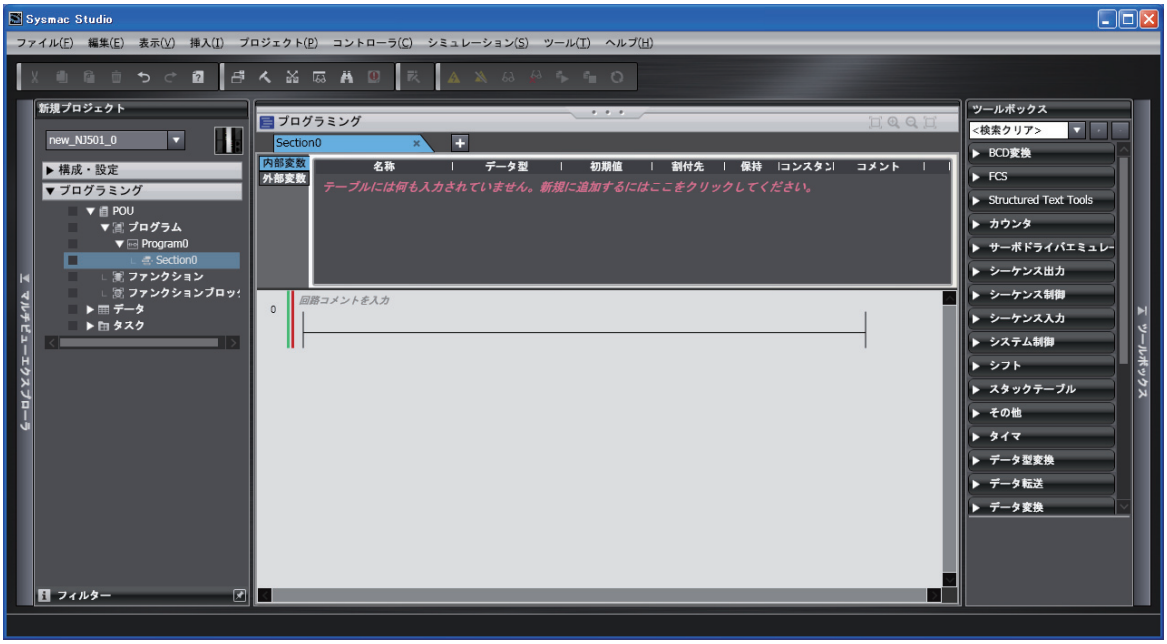
在多视图浏览器中右击［程序］，单击［追加］中的［梯形图］或［ST］。



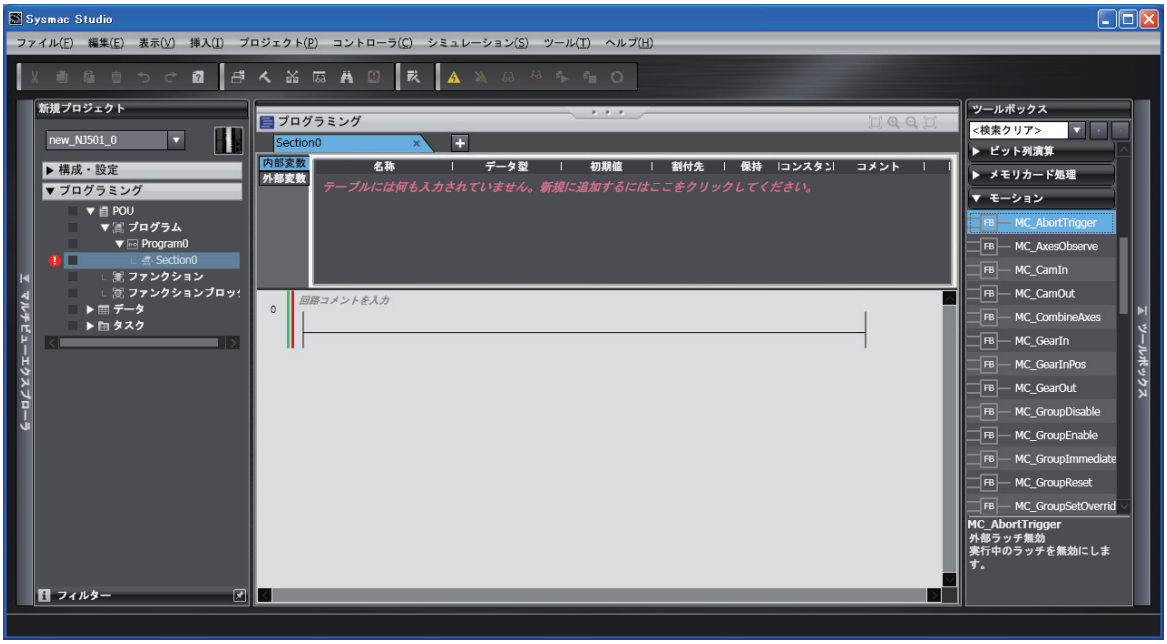
程序将追加到多视图浏览器中。

3 编辑程序

右击所追加程序下的段落，再单击［编辑］，将打开程序的编辑画面。



请从工具箱中选择指令，创建需要的程序。



编程的详情请参考 🕮《NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-CN5-359)》。

关于操作方法的详情，请参考 🕮《Sysmac Studio Version 1 操作手册 （SBCA-CN5-362）》。

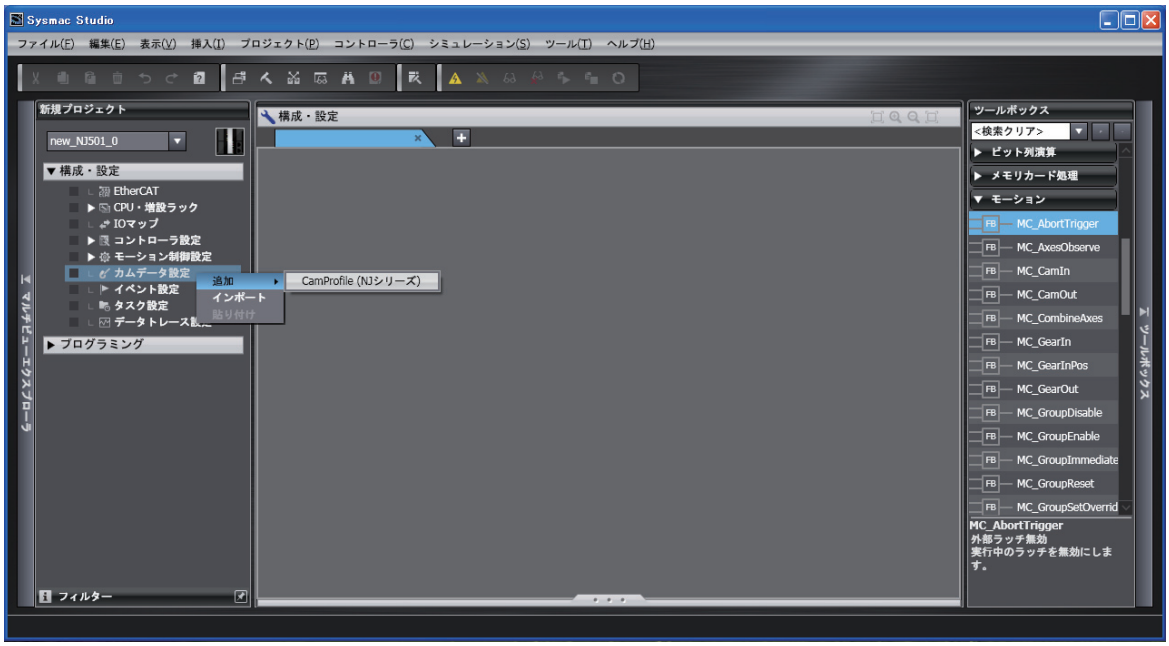
6-9　凸轮表的创建方法

下面介绍使用 Sysmac Studio 的凸轮编辑器功能创建凸轮表的步骤。

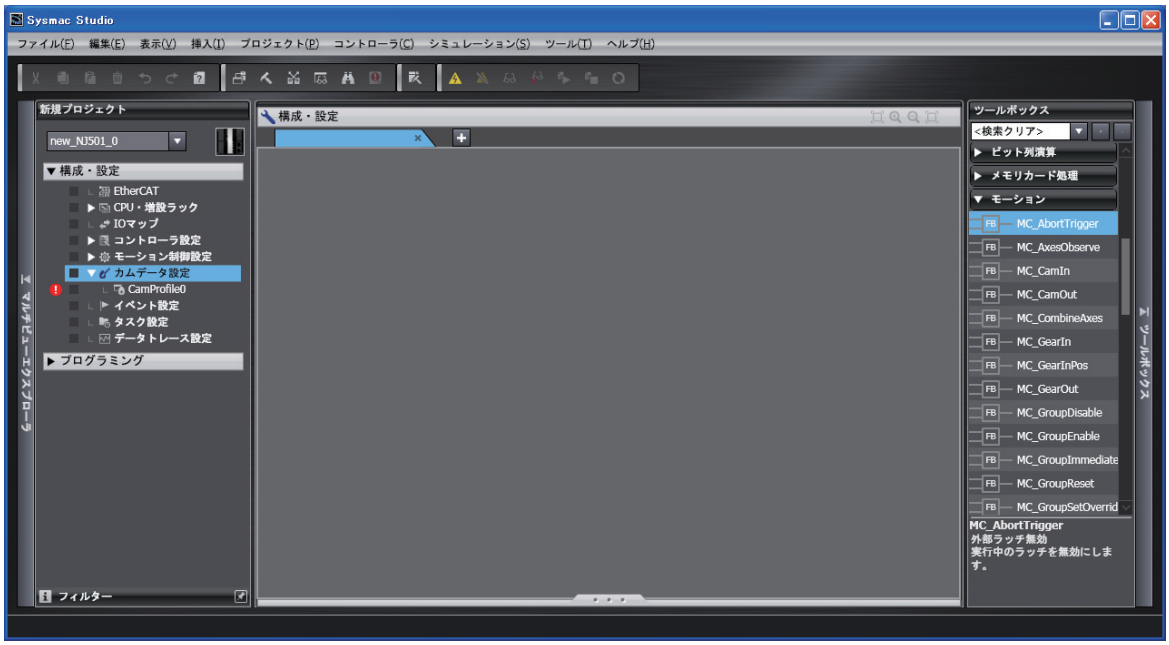
关于凸轮编辑器功能的详情，请参考 🕮《Sysmac Studio Version 1 操作手册 （SBCA-CN5-362）》。

1　追加凸轮轮廓

在多视图浏览器中右击［凸轮数据设定］，单击［追加］中的［CamProfile （NJ 系列）］。

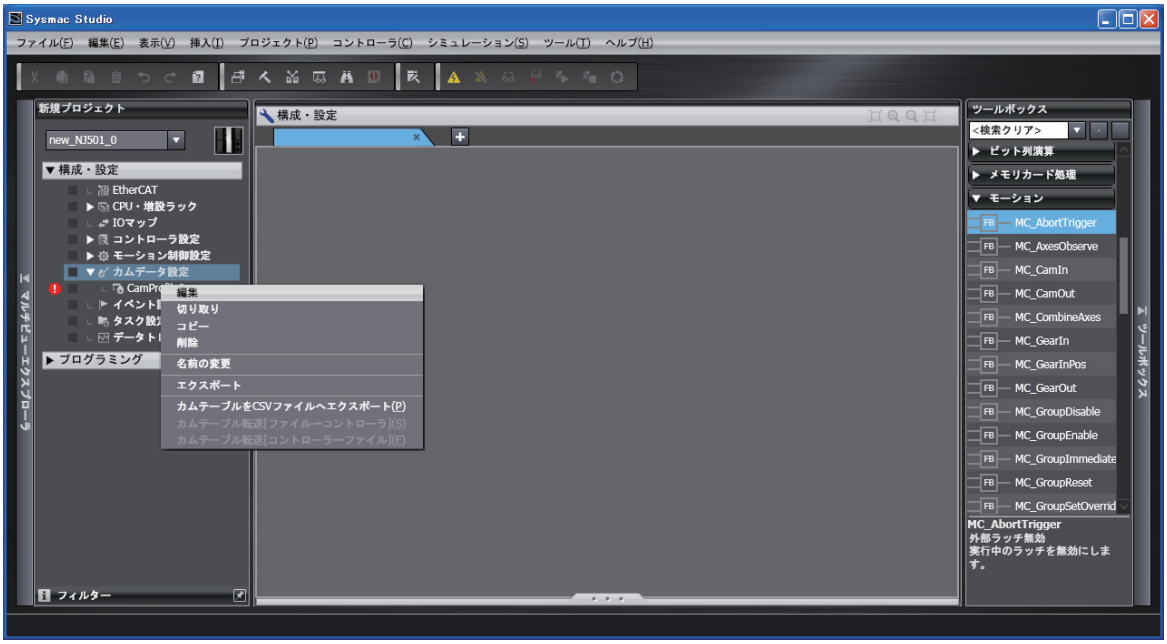


凸轮轮廓将追加到多视图浏览器中。所追加凸轮轮廓的名称默认为 “CamProfile0”，可变更为任意名称。

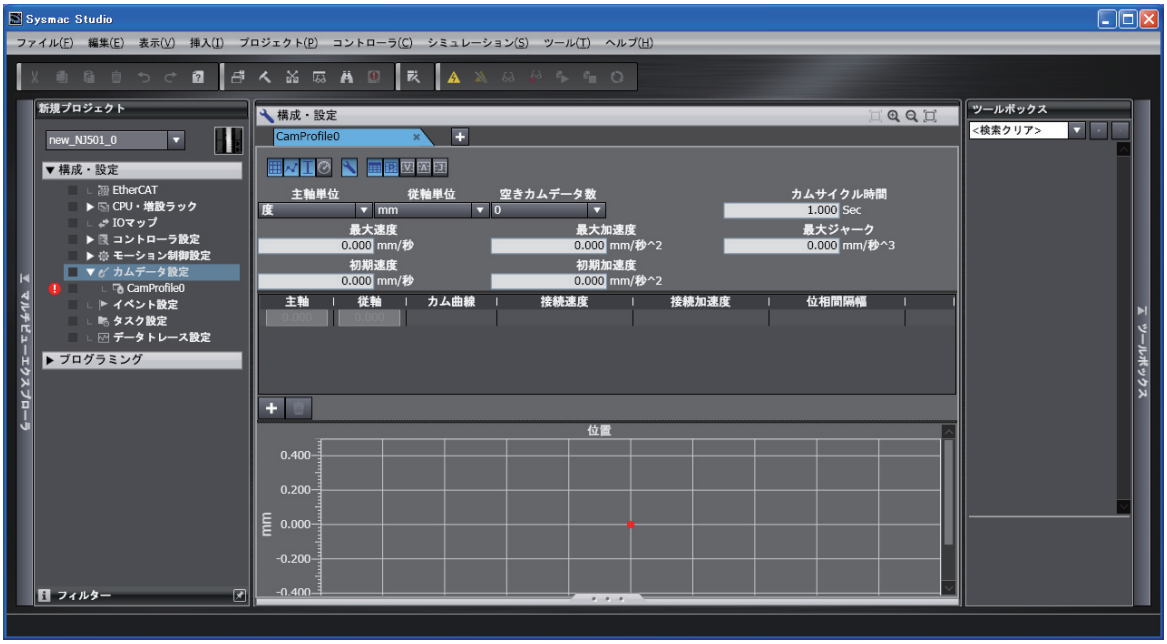


2 **编辑凸轮轮廓**

在多视图浏览器中右击追加的凸轮轮廓，单击［编辑］。



显示凸轮轮廓的编辑画面。



请设定各项目，编辑凸轮轮廓。

关于操作方法的详情，请参考 《Sysmac Studio Version 1 操作手册 （SBCA-CN5-362）》。

7

手动运行

介绍与欧姆龙生产的伺服驱动器 G5 系列组合时，手动操作所需的功能。

7-1 概要 7-2

7-2 伺服 ON 7-3

7-2-1 伺服 ON 的步骤 7-4
7-2-2 设定轴参数 7-4
7-2-3 程序示例 7-5

7-3 点动进给 7-6

7-3-1 点动进给的步骤 7-6
7-3-2 设定轴参数 7-7
7-3-3 输入变量的设定示例 7-7
7-3-4 程序示例 7-8

7-1 概要

介绍 MC 功能模块与欧姆龙生产的伺服驱动器 G5 系列组合时，如何通过用户程序使用运动控制指令，执行手动操作。

手动运行的运动控制指令包括将伺服驱动器设为伺服 ON 状态的 “MC_Power （可运行） ” 指令和进行点动进给的 “MC_MoveJog （点动进给） ” 指令。



使用注意事项

要进行手动运行，需要设定轴。关于轴的设定方法，请参考 *「第 3 章 轴和轴组的设定」*。

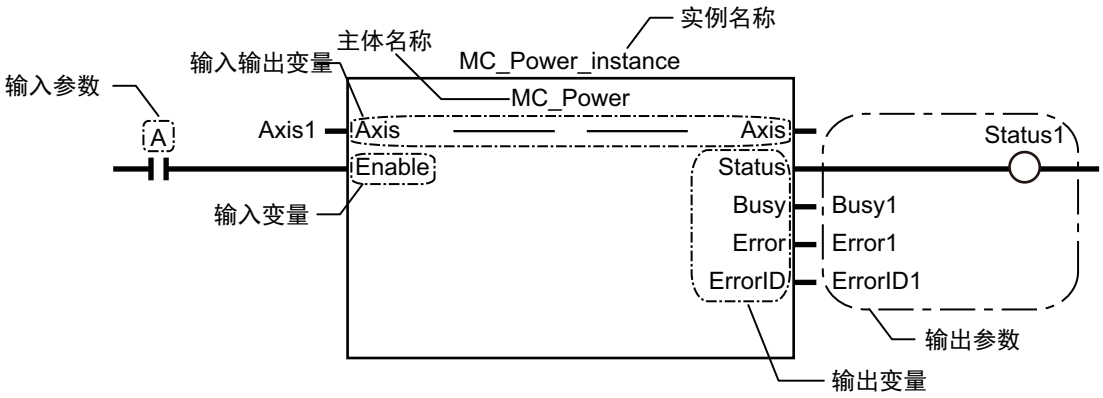


参考

- 想要不编写程序，直接手动运行时，请使用 Sysmac Studio。
  关于使用 Sysmac Studio 的手动运行，请参考 *「4-3 电机动作的确认* (P.4-8)」。
- 用户程序的创建方法请参考 *「第 6 章 运动控制程序」*。

7-2 伺服 ON

将针对伺服驱动器的操作指令设为 ON/OFF 的功能。
使用运动控制指令中的 MC_Power （可运行）指令。



要动作的轴在输入输出变量 Axis （轴）中指定。
将 MC_Power （可运行）指令的输入变量 Enable （有效）设为 TRUE 后，将变为伺服 ON 状态。将 Enable （有效）设为 FALSE 后，将变为伺服 OFF 状态。

使用注意事项

- 若在轴动作过程中将 Enable （有效）设为 FALSE，将立即停止指令，针对该轴的运动控制指令变为无效。
- 使用 NX 系列脉冲输出单元时，需要单独执行电机驱动器的通电 ON/OFF。详情请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

参考

在欧姆龙生产的伺服电机 G5 系列的带绝对编码器伺服电机或欧姆龙生产的电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型中使用绝对型外部光栅尺时，若将 MC_Power （可运行）指令的输入变量 Enable （有效）设为 TRUE，将确定原点。

版本相关信息

在单元版本 1.10 以上的 CPU 单元中，若在欧姆龙生产的伺服电机 G5 系列的带绝对编码器伺服电机或欧姆龙生产的电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型中使用绝对型外部光栅尺，除了上述 “ 参考 ” 外，当 EtherCAT 的过程数据通信从未建立状态变为建立状态后，也将确定原点。

7-2-1 伺服 ON 的步骤

1 **轴的追加和设定**

在 Sysmac Studio 中进行轴的追加和设定。

详情请参考 🕮「*3-2-2 设定步骤* (P.3-9)」。

2 **设定轴参数**

在 Sysmac Studio 中进行轴参数的设定。

详情请参考 🕮「*3-2-2 设定步骤* (P.3-9)」。

3 **创建用户程序**

在 Sysmac Studio 中创建用户程序。

详情请参考 🕮「*6-8 运动控制程序的创建方法* (P.6-33)」。

4 **下载轴参数和用户程序**

将设定的轴参数和用户程序下载到 CPU 单元。

下载到 CPU 单元时，使用 Sysmac Studio 的 “ 同步 ” 功能。

详情请参考 🕮「*3-2-2 设定步骤* (P.3-9)」。

5 **启动用户程序**

启动用户程序，将 “MC_Power” 指令的输入变量 “Enable” 设为 TRUE。

伺服驱动器变为伺服 ON 状态。

7-2-2 设定轴参数

仅设为伺服 ON 状态时，需要设定的轴参数为以下设定项目。

下表为设定示例。

设定项目	设定内容
轴变量名称	Axis1 *1
轴编号	1 *2
轴使用	使用轴
轴种类	伺服轴
输入设备 / 输出设备	1 *3

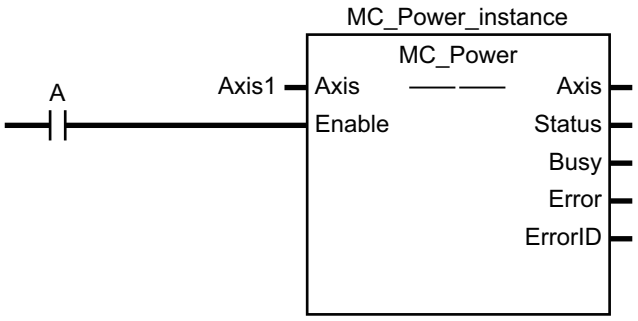
*1. 有多个轴时，请对不同的轴设定不同的变量名称。
*2. 有多个轴时，请对不同的轴设定不同的值。
*3. 请设定为与伺服驱动器的节点地址相同的值。有多个轴时，请对不同的轴设定不同的值。

参考

若正确连接欧姆龙生产的伺服驱动器 G5 系列，执行 Sysmac Studio 的网络扫描功能后，将自动设定上述轴变量名称以外的轴参数。

7-2-3 程序示例

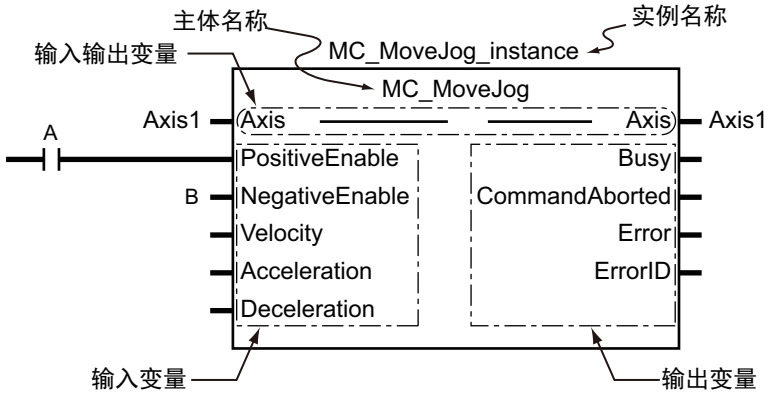
通过接点 A 切换 Axis1 的伺服 ON/OFF 的示例程序如下所示。



MC_Power （可运行）指令的详情请参考 📖《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。

7-3 点动进给

点动进给使用运动控制指令中的 MC_MoveJog （点动进给）指令。



要进行点动进给的轴在输入输出变量 Axis （轴）中指定。

若将输入变量 PositiveEnable 设为 TRUE，将按正方向上指定的 Velocity （目标速度）、 Acceleration （加速度）启动轴。若将 PositiveEnable 设为 FALSE，将按指定的 Deceleration （减速度）减速并停止。

同时，若将输入变量 NegativeEnable 设为 TRUE，将向负方向启动，变为 FALSE 后停止。

点动进给在原点未确定的状态下也可执行。

7-3-1 点动进给的步骤

1 **轴的追加和设定**

在 Sysmac Studio 中进行轴的追加和设定。

详情请参考 🕮「*3-2-2 设定步骤* (P.3-9)」。

2 **设定轴参数**

在 Sysmac Studio 中进行轴参数的设定。

详情请参考 🕮「*3-2-2 设定步骤* (P.3-9)」。

3 **创建用户程序**

在 Sysmac Studio 中创建用户程序。

详情请参考 🕮「*6-8 运动控制程序的创建方法* (P.6-33)」。

4 **下载轴参数和用户程序**

将设定的轴参数和用户程序下载到 CPU 单元。

下载到 CPU 单元时，使用 Sysmac Studio 的 “ 同步 ” 功能。

详情请参考 🕮「*3-2-2 设定步骤* (P.3-9)」。

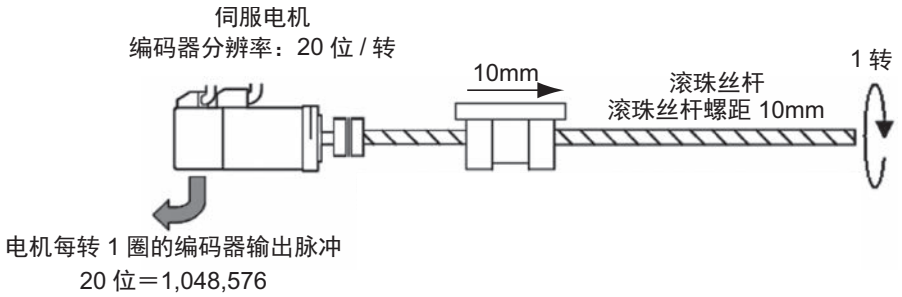
5 **启动用户程序**

启动用户程序，将 MC_Power （可运行）指令的输入变量 “Enable” 设为 TRUE，将伺服驱动器设为伺服 ON 状态。

将 MC_MoveJog （点动进给）指令的输入变量 “PositiveEnable” 或 “NegativeEnable” 设为 TRUE，进行点动进给。

7-3-2 设定轴参数

在原点未确定状态下执行点动进给时，请设定以下轴参数。
下面以单轴装置为例介绍设定示例。



设定项目	设定内容
轴变量名称	Axis1 *1
轴编号	1 *2
轴使用	使用轴
轴种类	伺服轴
输入设备 / 输出设备	1 *3
电机每转 1 圈的脉冲数	1,048,576 *4
电机每转 1 圈的移动量	10,000 *4
软件限位功能	对反馈位置有效
显示单位	μm
计数模式	直线模式
最高速度	500,000 *5
JOG 最高速度	50,000 *6
最大加速度	5,000,000 *7
最大减速度	5,000,000 *7

*1. 有多个轴时，请对不同的轴设定不同的变量名称。
*2. 有多个轴时，请对不同的轴设定不同的值。
*3. 请设定为与伺服驱动器的节点地址相同的值。有多个轴时，请对不同的轴设定不同的值。
*4. 位置的指令单位为 1(μm)。
*5. 最高速度为 3,000r/min = 30m/min = 0.5m/s = 500,000μm/s。
*6. JOG 最高速度为最高速度的 10%， 0.05m/s = 50,000μm/s。
*7. 最大加速度及最大减速度为 5m/s^2。加速至最高速度 3,000r/min 的时间为 0.1s。

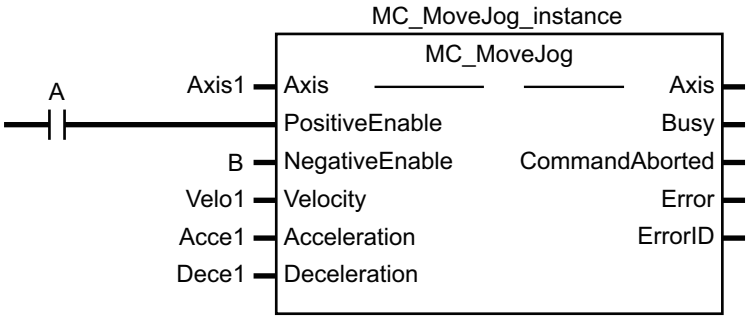
7-3-3 输入变量的设定示例

下面介绍 “MC_MoveJog （点动进给） ” 指令的输入变量 “Velocity （目标速度） ”、 “Acceleration （加速度） ”、 “Deceleration （减速度） ”。

- 例如，要以 0.03m/s (30,000μm/s) 的速度进行点动进给时，将 “Velocity” 指定为 30,000。
- 此外，要以 3m/s^2 (3,000,000μm/s^2) 进行加减速时，将 “Acceleration” 和 “Deceleration” 指定为 3,000,000。

7-3-4 程序示例

通过接点 A 使 Axis1 向正方向进行点动进给、通过接点 B 向负方向进行点动进给时的示例程序如下所示。



此时， Velocity （目标速度）为 Velo1、 Acceleration （加速度）为 Acce1、 Deceleration （减速度）为 Dece1。

要以输入变量的设定示例中的条件动作时，需要事先在用户程序中设定各变量的值。

- Velo1 = 30,000
- Acce1 = 3,000,000
- Dece1 = 3,000,000

MC_MoveJog （点动进给）指令的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。

8

定原点

本章对定原点进行说明。

8-1 概要 8-2

8-2 原点复位的步骤 8-5

8-2-1 原点复位参数的设定 8-6

8-2-2 原点复位的监视 8-12

8-3 原点复位的动作 8-13

8-4 绝对编码器原点设定 8-14

8-4-1 功能概要 8-15

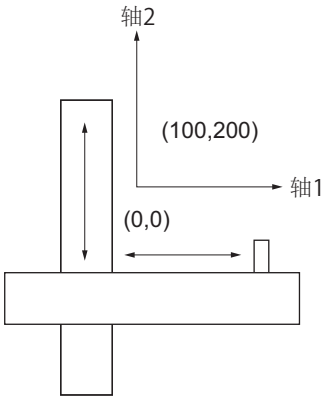
8-4-2 设定步骤 8-16

8-5 高速原点复位 8-18

8-1 概要

介绍 MC 功能模块与欧姆龙生产的伺服驱动器 G5 系列组合时的操作。

在进行定位的系统中，对系统的绝对位置进行定位时，首先需要执行定原点。
例如，在下图所示的 XY 平台中，想要定位到 （轴 1、轴 2） = (100mm, 200mm) 的位置时，必须确定机械的原点 (0, 0)。
确定机械原点这个操作称为定原点。



在 MC 功能模块中，确定机械原点的方法有使用运动控制指令中的 “MC_Home （原点复位） ” 指令或 MC_HomeWithParameter （参数指定原点复位）指令。

名称	内容
原点复位	实际驱动电机，使用极限传感器、近原点传感器、原点输入信号确定机械原点。 作为原点输入信号，使用接近传感器或编码器的 Z 相。

使用注意事项

- 以下情况为原点未确定。
  MC_SetPosition （当前位置变更）指令的执行
  直线模式时的溢出及下溢
  原点复位开始时
  EtherCAT 通信的控制状态为非可操作状态
- 使用 NX 系列位置接口单元时，部分功能有限制。详情请参考 《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

参考

在欧姆龙生产的伺服电机 G5 系列的带绝对编码器伺服电机或欧姆龙生产的电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型中使用绝对型外部光栅尺时，若将 MC_Power （可运行）指令的输入变量 Enable （有效）设为 TRUE，将确定原点。

版本相关信息

在单元版本 1.10 以上的 CPU 单元中，若在欧姆龙生产的伺服电机 G5 系列的带绝对编码器伺服电机或欧姆龙生产的电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型中使用绝对型外部光栅尺，除了前述 “ 参考 ” 外，当 EtherCAT 的过程数据通信从未建立状态变为建立状态后，也将确定原点。

此外，还配备有以下 MC_MoveZeroPosition （高速原点复位）指令，以定位到上述方法中确定的原点。

名称	内容
高速原点复位	将绝对坐标 “0” 作为目标位置，进行定位并返回原点。

此外， MC 功能模块在原点未确定状态下也可运行电机 （高速原点复位除外）。

功能	动作
点动进给、速度控制	在原点未确定状态下，将接通电源时的位置视为 “0” 进行移动。
高速原点复位	不可使用。 若使用，将发生指令异常。
定位	在原点未确定状态下，将接通电源时的位置视为 “0” 进行移动。
中断固定尺寸定位	
凸轮动作开始	
齿轮动作开始	
梯形模式凸轮	
加减法定位	
转矩控制	
区域监视	
直线插补	不可使用。 若使用，将发生指令异常。
圆弧插补	

参考

原点未确定时，软件限位功能为无效。

各操作或事件的原点确定状态变化如下所示。

操作或事件	变化条件	原点确定状态的变化
伺服 ON 轴有效时	增量编码器	无变化
	通过绝对编码器正常读取绝对值数据时	原点确定
	通过绝对编码器读取绝对值数据失败时	原点未确定
当前位置变更	启动时	原点未确定
原点复位	启动时	原点未确定
	完成时	原点确定
溢出 / 下溢	发生时	原点未确定

使用注意事项

- 关于虚拟轴，无论轴参数［原点复位动作］的设定如何，始终作为［原点预设］进行原点确定。
- 在 MC 功能模块中，将伺服驱动器的正转侧驱动禁止输入（POT）、反转侧驱动禁止输入（NOT）、近原点输入（DEC）分别作为正方向极限输入、负方向极限输入、近原点输入使用。

  这些输入信号的信号宽度请务必大于 MC 功能模块的控制周期，确保伺服驱动器能够检测到。

  若输入信号的宽度小于控制周期，MC 功能模块将无法检测到输入信号，导致无法正常执行动作。
- 伺服驱动器的各输入信号需要设定伺服驱动器侧的参数。请参考所连接伺服驱动器的手册和附录，正确设定。

8-2 原点复位的步骤

介绍执行原点复位的步骤。

1 **轴的追加和设定**
在 Sysmac Studio 中进行轴的追加和设定。

2 **设定轴参数**
通过原点复位参数设定原点复位方法。

3 **创建用户程序**
在 Sysmac Studio 中创建用户程序。
详情请参考 「*6-8 运动控制程序的创建方法* (P.6-33)」。

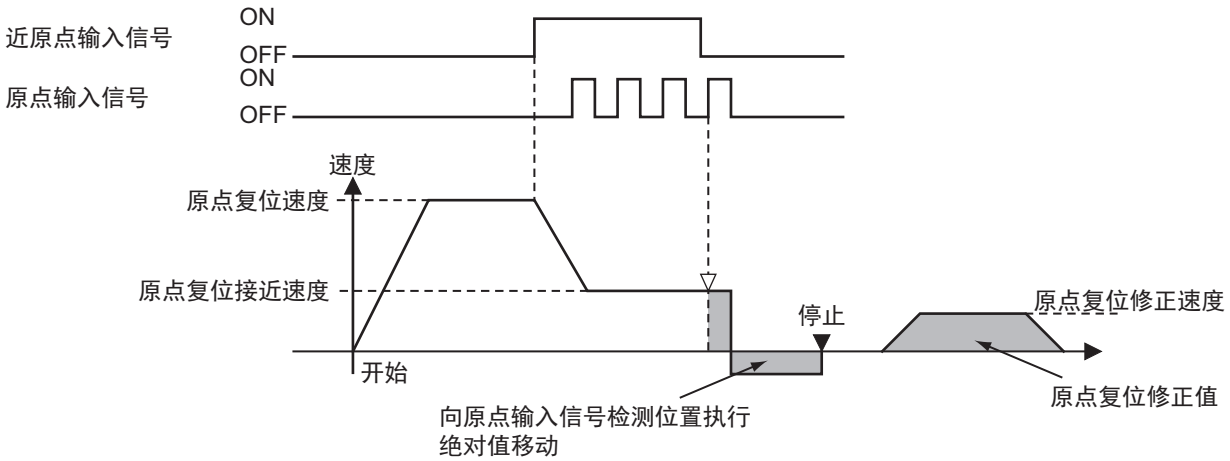
4 **下载轴参数和用户程序**
将设定的轴参数和用户程序下载到 CPU 单元。
下载到 CPU 单元时，使用 Sysmac Studio 的 “ 同步 ” 功能。
详情请参考 「*3-2-2 设定步骤* (P.3-9)」。

5 **启动用户程序**
启动用户程序，将 MC_Power （可运行）指令的输入变量 “Enable” 设为 TRUE，将伺服驱动器设为伺服 ON 状态。
将 MC_Home （原点复位）指令或 MC_HomeWithParameter （参数指定原点复位）指令的输入变量 “Execute” 设为 TRUE，执行原点复位。

8-2-1 原点复位参数的设定

原点复位的方法通过原点复位参数的设定值指定。
原点复位参数在 Sysmac Studio 中设定。

设定	概要
原点复位动作	选择原点复位的动作方法。
原点输入信号	选择原点输入信号的输入目标。
原点复位开始方向	设定原点复位启动时的开始方向。
原点检测方向	设定原点复位的原点输入检测方向。
正方向极限输入时动作	设定在原点复位过程中将正方向极限输入设为 “ON” 时的停止方法。
负方向极限输入时动作	设定在原点复位过程中将负方向极限输入设为 “ON” 时的停止方法。
原点复位速度	设定原点复位时的速度。（单位：指令单位 /s）
原点复位接近速度	设定近原点输入变为 “ON” 后的速度。（单位：指令单位 /s）
原点复位加速度	指定原点复位时的加速度。 若设定为 “0”，则不加速，达到原点复位速度。 （单位：指令单位 /s^2）
原点复位减速度	指定原点复位时的减速度。 若设定为 “0”，则不减速，达到零速度。 （单位：指令单位 /s^2）
原点复位跃度	指定原点复位时的跃度。 若设定为 “0”，将没有跃度。（单位：指令单位 /s^3）
原点输入掩码距离	设定原点复位动作模式为 “ 附近退避、原点输入掩码距离指定 ” 时的原点输入掩码距离。（单位：指令单位）
原点位置偏置	原点复位完成时按设定的值预设当前位置。 （单位：指令单位）
原点复位推压时间	以毫秒为单位设定原点复位动作模式为 “ 附近退避、推压时间指定 ” 时的推压时间。（单位：ms）
原点复位修正值	设定原点复位的原点确定后的原点复位修正量。（单位：指令单位）
原点复位修正速度	设定原点复位修正时的速度。（单位：指令单位 /s）



下面介绍各原点复位参数。

原点复位动作

作为决定机械原点的动作，可从 10 种动作中选择。

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

各原点复位动作中使用的原点复位参数如下表所示。

○: 使用参数、 ×: 不使用参数

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

各原点复位动作的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。

使用注意事项

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

原点输入信号

在使用原点输入信号的原点复位动作模式中，从伺服驱动器的 Z 相信号、外部原点信号中选择进行定原点所需的信号。

使用注意事项

在该设定中可选择原点输入信号的情况仅限连接了欧姆龙生产的伺服驱动器 G5 系列时。

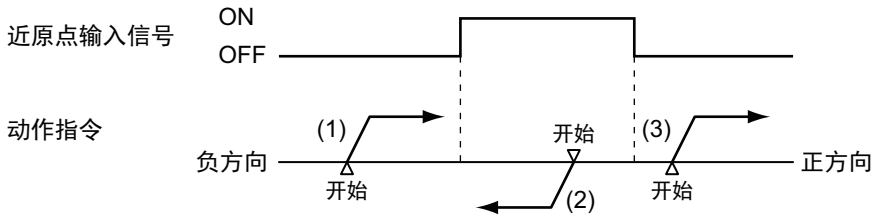
原点复位开始方向

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

包含附近退避引起的反转动作的原点复位动作模式有以下 4 种。

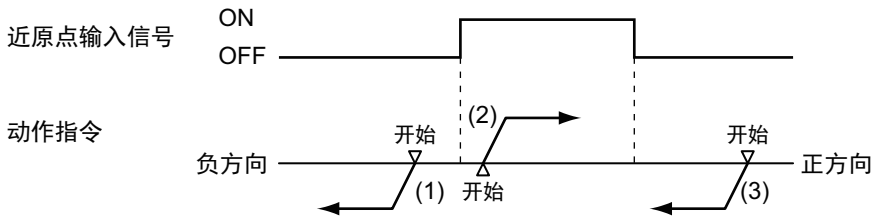
- 0: 附近退避、近原点输入 OFF 指定
- 1: 附近退避、近原点输入 ON 指定
- 9: 附近退避、原点输入掩码距离指定
- 12: 附近退避、推压时间指定

原点复位开始方向：正
原点检测方向：正



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

原点复位开始方向：负
原点检测方向：负



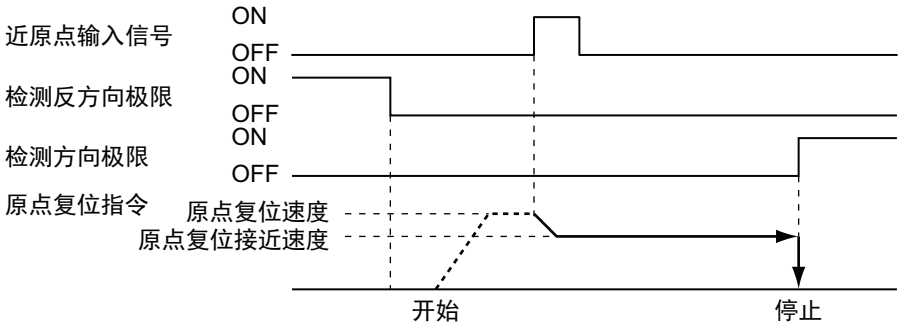
(1)、(3) ：在近原点信号为 “OFF” 的状态下开始，按原点复位开始方向开始动作
(2) ：在近原点信号为 “ON” 的状态下开始，按原点检测方向的反方向开始动作

原点检测方向

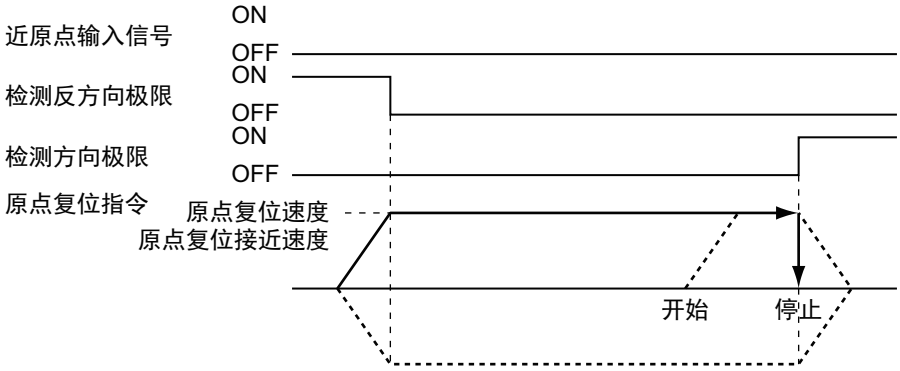
从正方向和负方向中选择原点检测的方向。
原点检测方法和开始原点复位时的运动方向之间的关系请参考 *「原点复位开始方向* (P.8-8)」。

正方向极限输入时动作 / 负方向极限输入时动作

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



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



原点复位速度

以指令单位 /s 设定原点复位时的速度。

原点复位接近速度

以指令单位 /s 设定近原点输入变为 “ON” 后的速度。

原点复位加速度

以指令单位 /s^2 设定原点复位时的加速度。
若设定为 “0”，则不加速，达到原点复位速度等目标速度。

原点复位减速度

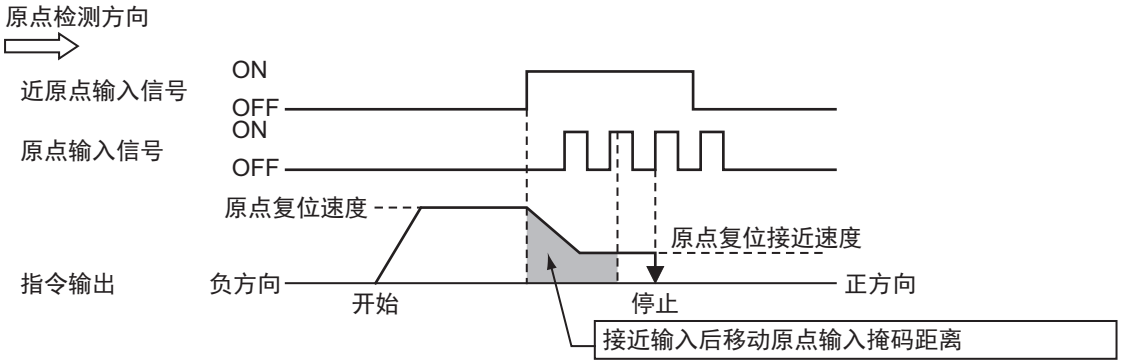
以指令单位 /s^2 设定原点复位时的减速度。
若设定为 “0”，则不减速，达到原点复位接近速度等目标速度。

原点复位跃度

以指令单位 /s^3 设定原点复位时的跃度。
若设定为 “0”，则加减速没有跃度。

原点输入掩码距离

以指令单位设定原点复位动作为 ［9: 附近退避、原点输入掩码距离指定］时原点输入的掩码距离。
通过近原点输入信号的 “OFF” ➞ “ON” 开始减速后的距离。

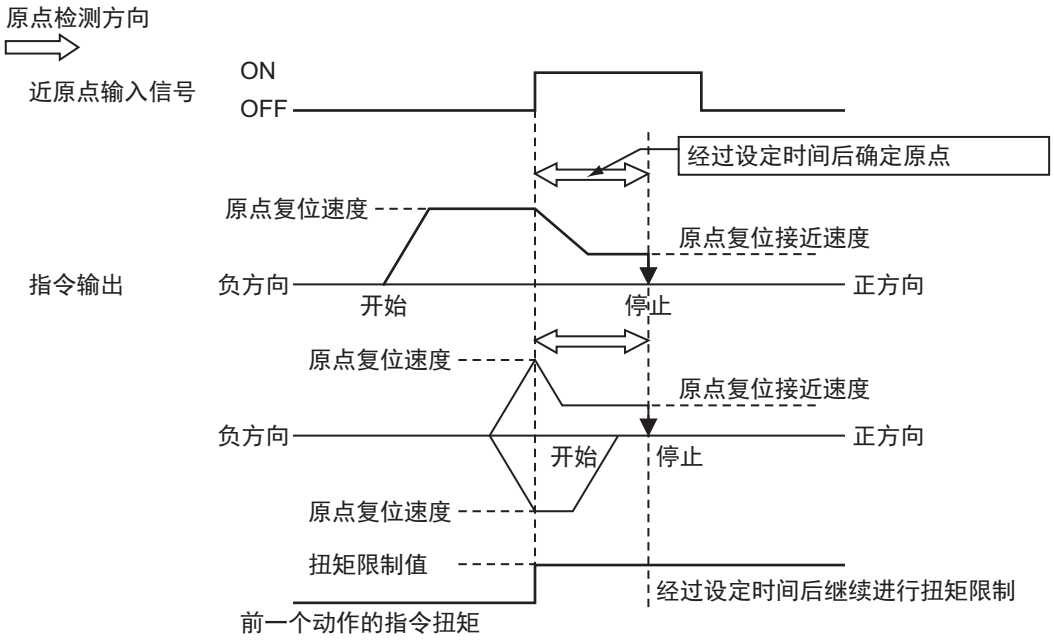


原点位置偏置

确定机械原点后，若已设定原点复位修正值，在原点复位修正动作完成后，将以设定的值预设当前值。以任意值使用机械原点的功能，而不是 “0”。
此外，若为绝对编码器系统，将同时自动计算绝对编码器原点位置偏置，并保存到 CPU 单元的电池备份存储器中。

原点复位推压时间

以毫秒为单位设定原点复位动作为 “12: 附近退避、推压时间指定 ” 时的推压时间。
通过近原点输入信号的 “OFF” ➞ “ON” 开始减速后的时间。



原点复位修正值

确定机械原点后，按设定的值进行相对定位，修正机械原点。此时的移动速度为原点复位修正速度。
确定机械原点后，将很难调整工件，需要微调时，请将从最初确定的机械原点开始的偏差量设定为修正数据。
近原点传感器难以重新安装或因电机的更换导致机械原点发生偏差时，使用起来非常方便。

原点复位修正速度

以指令单位 /s 设定原点复位修正值已设定时的速度。

8-2-2 原点复位的监视

在程序中，可通过读取轴变量来监视原点复位状态或输入信号状态。

此外，以 _MC_AX[*] 为例对变量名称进行说明， _MC1_AX[*]、 _MC2_AX[*] 亦同。

变量名称	数据类型	名称	功能
_MC_AX[0-255].Status.Homing	BOOL	原点复位中	正在通过 MC_Home 指令或 MC_HomeWithParameter 指令执行原点复位时变为 TRUE。
_MC_AX[0-255].Dtails.Homed	BOOL	原点确定	原点确定状态时变为 TRUE。 FALSE: 原点未确定 TRUE: 原点确定
_MC_AX[0-255].Dtails.InHome	BOOL	原点停止	在原点位置的范围内时为 TRUE。 以下的 AND 条件。 • 原点确定状态 • 反馈当前位置进入以原点为中心的 “ 原点位置范围 ” 内时 在指令状态下轴正在动作时，若通过原点位置，也会变为 TRUE。
_MC_AX[0-255].DrvStatus.P_OT	BOOL	正方向极限输入	正方向极限输入为有效时变为 TRUE。
_MC_AX[0-255].DrvStatus.N_OT	BOOL	负方向极限输入	负方向极限输入为有效时变为 TRUE。
_MC_AX[0-255].DrvStatus.HomeSw	BOOL	近原点输入	近原点输入为有效时变为 TRUE。
_MC_AX[0-255].DrvStatus.Home	BOOL	原点输入	原点输入为有效时变为 TRUE。

8-3 原点复位的动作

原点复位时，因定位系统的构成和用途不同，需要选择机械原点的确定方法。
在 MC 功能模块中，预置有 10 种原点复位动作模式。
此外，确定原点后，还可利用原点复位修正值进行微调。



参考

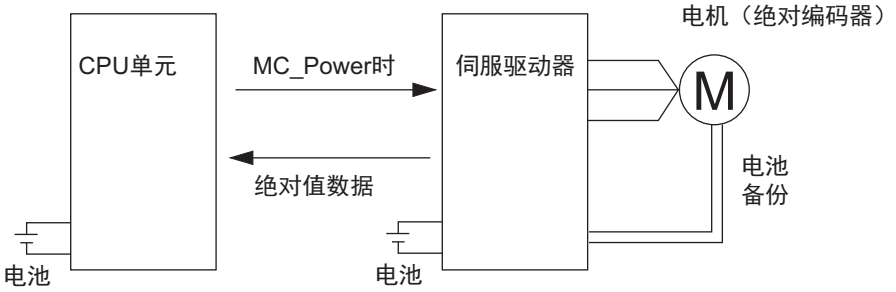
- 适用于哪种模式，取决于定位系统的构成和用途。
  在直线模式（长度有限）下，有近原点传感器和正方向极限输入、负向限极限输入的机械系统中，一般选择“附近退避、近原点输入 OFF 指定的动作”。
- 在原点复位中，超驰变为无效。
- 仅在原点确定时以及完成原点复位修正移动时，按照位置检查的设定进行位置检查。
- 在原点复位过程中用其他运动控制指令执行多重启动时，不执行缓冲和共混。

原点复位的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。

8-4 绝对编码器原点设定

下面介绍如何使用欧姆龙生产的伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型绝对编码器。

绝对编码器即使在 CPU 单元的电源断开后，也可利用编码器中的电池备份来保持绝对值数据。因此，执行 MC_Power （可运行）指令时，将利用绝对编码器读取当前位置，以确定位置。
一旦原点确定后，无需像增量编码器那样进行原点复位操作。



使用注意事项

- 在欧姆龙生产的伺服驱动器 G5 系列上使用绝对编码器时，请同时连接 CPU 单元的电池和伺服驱动器的绝对编码器备份用电池。
- 第一次使用绝对编码器或更换电机时，如果使用欧姆龙生产的伺服驱动器 G5 系列，因绝对编码器电池耗尽而导致绝对值数据丢失时，请务必执行 “MC_Home （原点复位） ” 指令或 “MC_HomeWithParameter （参数指定原点复位） ” 指令，确定原点。此时，确定原点后，需要重新接通伺服驱动器侧的电源。之后，只要将 MC_Power （可运行）指令的输入变量 Enable （有效）设为 TRUE，即可确定原点。在单元版本 1.10 以上的 CPU 单元中，除了上述之外， EtherCAT 的过程数据通信从未建立状态变为建立状态时，也可确定原点。
- 若在 CPU 单元的电池异常时接通 CPU 单元的电源，将发生 “ 绝对编码器原点位置偏置读取异常 （事件代码 : 14600000Hex） ”。此时，若通过 “ResetMCError （MC 所有错误重置） ” 指令解除异常，也可变为伺服 ON。

  在单元版本 1.09 以下的 CPU 单元中，解除异常并变为伺服 ON 时，将 ［绝对编码器原点位置偏置］的值视为 “0” 进行原点确定。

  在单元版本 1.10 以上的 CPU 单元中，解除异常后将维持原点未确定状态，直至执行 “MC_Home （原点复位） ” 指令或 “MC_HomeWithParameter （参数指定原点复位） ” 指令并确定原点。

  为了在正确的位置执行动作，请执行 “MC_Home （原点复位） ” 指令或 “MC_HomeWithParameter （参数指定原点复位） ” 指令，在正确的位置确定原点。

  若断开 CPU 单元的电压，将变为原点未确定状态。



参考

使用欧姆龙生产的伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型时，可设定绝对编码器原点。使用直线电机型时，请注意以下几点，并阅读本节的说明。

- 直线电机型不使用 “ 编码器 ”，但使用相同功能的 “ 外部光栅尺 ”。
- 本节说明中的 “ 绝对编码器 ” 可替换为 “ 绝对型外部光栅尺 ”。
- 绝对型外部光栅尺中不含绝对编码器中有的旋转量数据。因此，无需进行绝对编码器中需要的设置。此外，也不使用用于保持旋转量数据的电池。
- 直线电机型的规格详情请参考 🕮《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型 用户手册 (SBCE-CN5-366)》。

8-4-1 功能概要

绝对编码器系统的原点设定是指在执行 MC_Power （可运行）指令时，进行绝对编码器的偏置补偿，以确定原点。

在单元版本 1.10 以上的 CPU 单元中，除了上述之外， EtherCAT 的过程数据通信从未建立状态变为建立状态时，也可确定原点。

此外，也可像增量编码器一样，通过原点复位来确定原点。此时，将原点确定后的指令位置和绝对编码器读取的绝对值数据之差作为 ［绝对编码器原点位置偏置］导入到 CPU 单元的电池备份存储器中。

即使利用 MC_Home （原点复位）指令或 MC_HomeWithParameter （参数指定原点复位）指令执行了当前位置预设，仍将原点确定后的指令位置和绝对值数据的差作为偏置量导入到 ［绝对编码器原点位置偏置］中。

在 MC 功能模块中，［绝对编码器原点位置偏置］自动保存在 CPU 单元的电池备份存储器中，因此无需在 Sysmac Studio 上进行保存操作。



使用注意事项

- ［绝对编码器原点位置偏置］与轴编号关联，按不同轴保存在电池备份存储器中。若变更轴编号，保存的偏置量将丢失。变更了轴编号时，请重新执行原点复位。
- 更换 CPU 单元或 CPU 单元的电池时，请在更换前在原点确定状态下执行电池备份存储器的备份。

  以此来备份 ［绝对编码器原点位置偏置］。
- 更换后，可通过恢复来使用更换前确定的原点。
- 备份 / 恢复时使用 Sysmac Studio。

  操作方法的详情请参考 🕮《Sysmac Studio Version 1 操作手册 （SBCA-CN5-362）》。

对象伺服电机

可使用绝对编码器原点设定的伺服电机如下所示。

厂家名称	系列名称	伺服驱动器型号	伺服电机型号
欧姆龙	G5 系列	R88D-KN□□□-ECT	R88M-K□□□□□S R88M-K□□□□□T R88M-K□□□□□C
		R88D-KN□□□-ECT-L	R88L-EC□□□□□□□□

使用注意事项

NX 系列脉冲输出单元中无法使用绝对编码器。

与伺服驱动器连接

请按照 《NJ/NX 系列 CPU 单元内置 EtherCAT 端口 用户手册 (SBCD-CN5-358)》正确连接。

8-4-2 设定步骤

介绍绝对编码器系统的原点设定步骤。

1 **设置绝对编码器**

设置方法请参考 《Sysmac Studio Version 1 操作手册 （SBCA-CN5-362）》。

2 **设定轴参数**

在 MC 功能模块轴参数中的 ［位置计数设定］的 ［编码器种类］中选择 ［1: 绝对编码器 (ABS)］。

详情请参考 *「5-2-7 位置计数设定* (P.5-22)」。

3 **执行原点复位**

执行原点复位。

原点确定方法可从 MC 功能模块轴参数的 ［原点复位设定］的 ［原点复位动作］中选择。

确定原点后，将指令位置和绝对编码器读取的绝对值数据之差作为 ［绝对编码器原点位置偏置］自动保存到电池备份存储器中。

设置绝对编码器

第一次使用绝对编码器或要将旋转量数据初始化为 “0” 时，或者绝对编码器在未连接电池的状态下长期放置后，需要设置绝对编码器。
设置方法的详情请参考 🕮《Sysmac Studio Version 1 操作手册 （SBCA-CN5-362）》。

使用注意事项

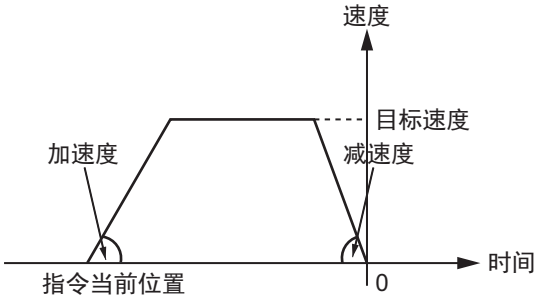
欧姆龙生产的伺服驱动器 G5 系列在设置绝对编码器后，需要重新接通电源。绝对编码器的设置完成后，伺服驱动器中将发生绝对值清除错误 (A27.1)。先关闭伺服驱动器的控制电源，再接通电源，可解除异常，绝对编码器的设置完成。

在旋转模式下使用绝对编码器

轴参数的［计数模式选择］设定为［旋转模式］时，当前位置将变为［循环计数器上限设定值］和［循环计数器下限设定值］范围内的环状计数器。
使用绝对编码器，设定为旋转模式时， MC 功能模块将按照运动控制周期，自动计算［绝对编码器原点位置偏置］并更新。
更新的［绝对编码器原点位置偏置］将在 CPU 单元电源 OFF 时自动保存到 CPU 单元内置的电池备份存储器中。
因此，即使关闭 CPU 单元或伺服驱动器的电源，也可在下次接通电源时，通过绝对编码器恢复旋转轴的当前位置。

8-5 高速原点复位

在原点确定状态下向原点位置进行高速定位的功能。
使用 MC_MoveZeroPosition （高速原点复位）指令，可指定目标速度、加速度、减速度、跃度。
若在原点未确定状态下执行，指令将发生异常。



参考

MC_MoveZeroPosition （高速原点复位）指令的详情请参考 📖《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。

9

运动控制功能

下面介绍与欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型连接时的运动控制功能。

9-1 单轴位置控制 9-3
9-1-1 动作概要 9-3
9-1-2 绝对值定位 9-4
9-1-3 相对值定位 9-4
9-1-4 中断固定尺寸定位 9-5
9-1-5 周期性同步定位 9-6
9-1-6 停止 9-6
9-1-7 超驰 9-11

9-2 单轴同步控制 9-12
9-2-1 同步控制的概要 9-12
9-2-2 齿轮动作 9-12
9-2-3 位置指定齿轮动作 9-13
9-2-4 凸轮动作 9-14
9-2-5 凸轮表 9-15
9-2-6 梯形模式凸轮动作 9-23
9-2-7 加减法定位 9-24
9-2-8 主轴相位补偿 9-25
9-2-9 从轴位置补偿 9-26
9-2-10 在多运动中实现同步控制的方法 9-26

9-3 单轴速度控制 9-28
9-3-1 速度控制 9-28
9-3-2 周期性同步速度控制 9-29

9-4 单轴扭矩控制 9-30

9-5 单轴控制的共通功能 9-31
9-5-1 位置 9-31
9-5-2 速度 9-33
9-5-3 加速和减速 9-34
9-5-4 跃度 9-36
9-5-5 动作方向指定 9-37
9-5-6 运动控制指令的重启 9-40
9-5-7 运动控制指令的多重启动（缓冲模式） 9-45

9-6 多轴协调控制 9-51

9-6-1 动作概要 9-51
9-6-2 直线插补 9-54
9-6-3 圆弧插补 9-55
9-6-4 轴组周期性同步位置控制 9-55
9-6-5 多轴协调控制的停止 9-56
9-6-6 多轴协调控制的超驰 9-58

9-7 多轴协调控制的共通功能 9-59
9-7-1 多轴协调控制的速度 9-59
9-7-2 多轴协调控制的加速和减速 9-60
9-7-3 多轴协调控制的跃度 9-61
9-7-4 多轴协调控制时运动控制指令的重启 9-62
9-7-5 多轴协调控制时运动控制指令的多重启动（缓冲模式） 9-63

9-8 其他功能 9-70
9-8-1 当前位置变更功能 9-70
9-8-2 扭矩限制功能 9-71
9-8-3 锁定功能 9-71
9-8-4 区域选择 9-72
9-8-5 软件限位功能 9-73
9-8-6 位置偏差监视功能 9-75
9-8-7 偏差计数器复位 9-75
9-8-8 轴间偏差监视功能 9-76
9-8-9 位置检查 9-77
9-8-10 轴使用变更功能 9-79
9-8-11 数字凸轮开关有效功能 9-79
9-8-12 任意坐标系的 3D 运动监视显示 9-80

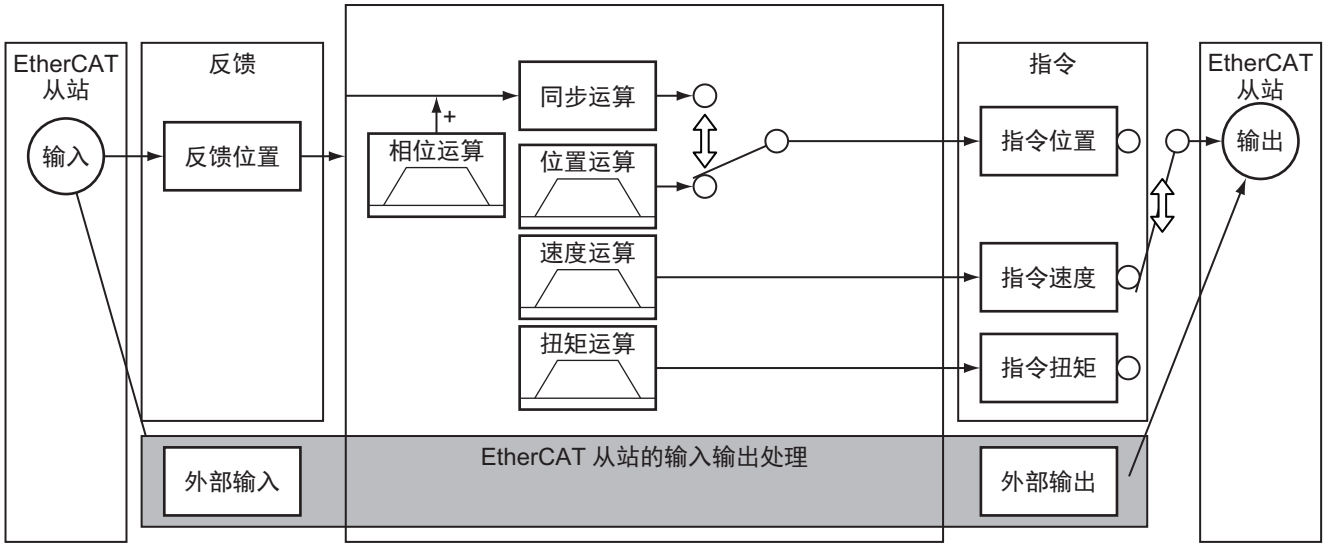
9-1 单轴位置控制

MC 功能模块与欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型连接后，可进行位置控制、速度控制、扭矩控制。
下面介绍单轴的定位动作。

使用 NX 系列脉冲输出单元时，运动控制功能的部分功能可能不同。详情请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

9-1-1 动作概要

MC 功能模块的单轴控制功能中，包括对动作轮廓发令的控制和同步控制。
在对动作轮廓发令的控制中，可在位置控制、速度控制、扭矩控制这 3 种控制模式下发令。
在同步控制中，控制对象 （从轴）与主轴在保持同步关系的前提下动作，这种同步关系以凸轮轮廓曲线或齿轮比表现。
此外，还支持点动进给等手动运行或原点复位。



（注）同步运算中，可将指令位置或反馈位置作为输入使用。

轴错误复位

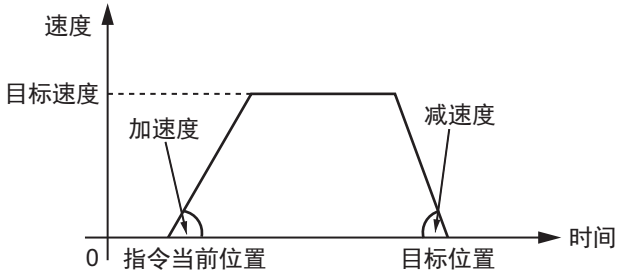
轴发生异常时，排除异常原因后，应通过 MC_Reset （轴错误复位）指令解除异常。

轴错误复位的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_Reset （轴错误复位）指令。

使用 NX 系列脉冲输出单元时的区别请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

9-1-2 绝对值定位

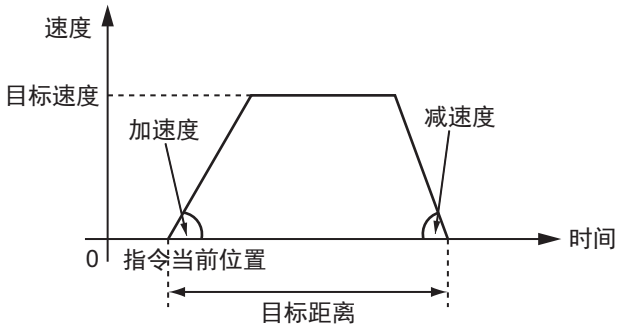
指定以原点为基准的绝对坐标的目标位置，进行定位。
将计数模式设为旋转模式并指定动作方向，可进行轮盘的最短距离等的定位。



功能详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_MoveAbsolute （绝对值定位）指令及 MC_Move （定位）指令。

9-1-3 相对值定位

指定从当前位置起的距离并定位。
将计数模式设为旋转模式后，可指定超出循环计数器范围的移动距离。



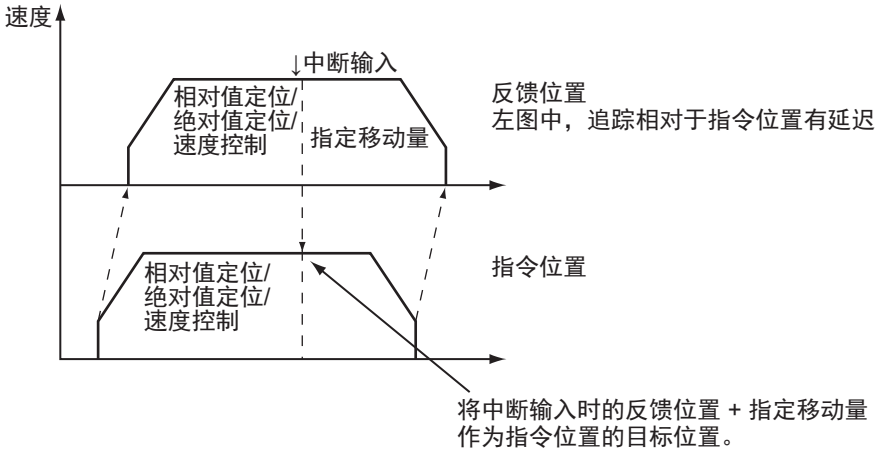
功能的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_MoveRelative（相对值定位）指令及 MC_Move （定位）指令。

9-1-4 中断固定尺寸定位

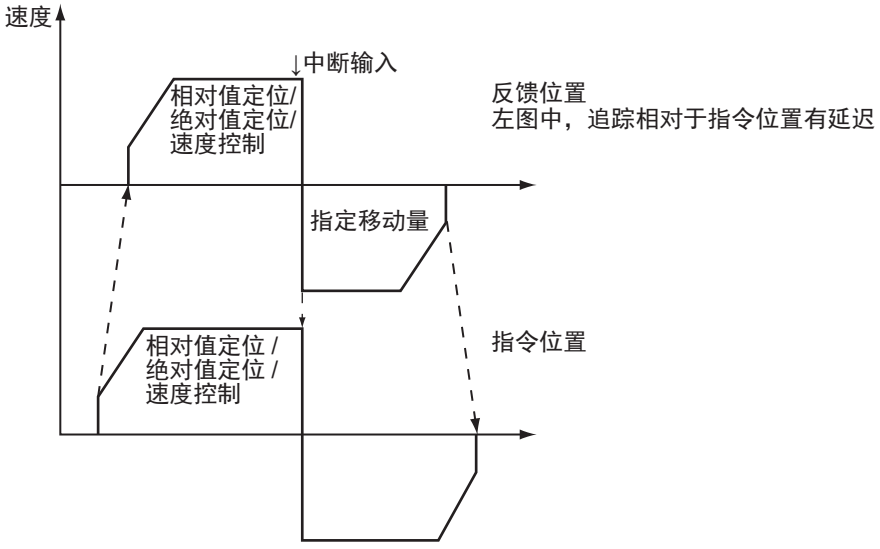
以指定速度输送轴，从发生触发信号的反馈位置起，按指定距离移动的定位。
绝对值定位或相对值定位时，还可选择为在指定移动量内未发生触发信号时，也输出异常。
发生触发信号的反馈位置使用伺服驱动器的锁定功能，因此可实现不受位置偏差影响的固定尺寸定位。
此外，利用可将超出指定位置范围的触发信号设为无效的窗口功能，在包装机等设备上，可消除薄膜的标记印刷以外的影响。

以反馈位置为基准

● 按指定量向前进方向移动



● 按指定量向前进方向的反方向移动



若在轴参数的反转时动作中选择［反转时减速停止］，根据加速 / 减速曲线反转。

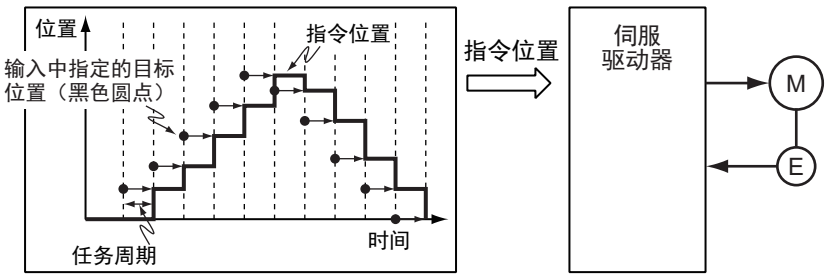
功能的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_MoveFeed （中断固定尺寸定位）指令。

使用 NX 系列脉冲输出单元时的区别请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

9-1-5 周期性同步定位

周期性同步定位是指按主周期或固定周期任务的任务周期，输出轴中指定的目标位置的功能。目标位置以绝对位置指定。

希望以用户创建的任意轨迹动作时使用。



功能的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_SyncMoveAbsolute （周期性同步绝对定位）指令。

版本相关信息

周期性同步定位功能可在 CPU 单元版本 1.03 以上和 Sysmac Studio Ver.1.04 以上的组合中使用。

9-1-6 停止

停止轴动作的功能中，包括利用伺服驱动器上连接的立即停止输入信号或极限输入信号、用户程序的运动控制指令执行的停止功能，以及发生异常时的停止。使用 NX 系列脉冲输出单元时的区别请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

根据伺服驱动器的输入信号停止

根据伺服驱动器的立即停止输入信号或极限输入信号，停止轴的动作。

可使用 Sysmac Studio 选择停止方法。

● 立即停止输入

根据伺服驱动器输入信号的状态，在 MC 功能模块上执行停止处理。

MC 功能模块的停止方法可从以下选项中选择。

- 立即停止指令值
- 立即停止指令值，同时执行偏差计数器复位
- 立即停止指令值，同时执行伺服 OFF

使用注意事项

欧姆龙生产的伺服驱动器 G5 系列发生立即停止输入时，伺服驱动器上也会发生异常，并执行停止处理。

● 极限输入 （正方向极限输入或负方向极限输入）

根据伺服驱动器输入信号的状态，在 MC 功能模块上执行停止处理。

MC 功能模块的停止方法可从以下选项中选择。

- 立即停止指令值
- 减速停止指令值
- 立即停止指令值，同时执行偏差计数器复位
- 立即停止指令值，同时执行伺服 OFF

使用注意事项

- 极限输入信号为 ON 时，请勿以与状态为 ON 的极限输入信号相同的方向启动轴指令。
- 轴组中任一轴的极限输入为 ON 时，请勿启动轴组指令。
- 正在执行输入变量中包含 “Deceleration （减速度） ” 的同步动作指令时，若输入了减速停止信号，将按照 “Deceleration （减速度） ” 减速停止。
- 正在执行输入变量中不含 “Deceleration （减速度） ” 的同步动作指令时，若输入了减速停止信号，将按照轴参数的最大减速度减速停止。

参考

- 要使用伺服驱动器的输入信号，需要设定伺服驱动器。在欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型中，按伺服驱动器的初始值分配立即停止输入和极限输入。
- 与欧姆龙生产的伺服驱动器 G5 系列连接时的设定示例请参考 *「A-1 与伺服驱动器连接* (P.A-2)」。

利用运动控制指令停止

停止单轴动作时，应使用 MC_Stop （强制停止）指令或 MC_ImmediateStop （立即停止）指令。

● MC_Stop （强制停止）指令

可对单轴控制及同步控制功能指定减速度和跃度，进行减速停止。

若将减速度指定为 “0”，将向伺服驱动器发送立即停止指令。

正在利用本指令减速停止时以及输入变量 Execute （启动）为 TRUE 期间，不会接受其他动作指令。

● MC_ImmediateStop （立即停止）指令

可立即停止单轴控制及同步控制功能

也可对轴组为有效的轴执行。

功能的详情请参考 《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_Stop （强制停止）指令、 MC_ImmediateStop （立即停止）指令。

参考

若将 MC_Power （可运行）指令的输入变量 Enable （有效）设为 FALSE， MC 功能模块将立即停止指令值，并执行伺服 OFF。
伺服 OFF 时，伺服驱动器的动作遵照伺服驱动器的设定。

因发生异常而停止

● 因单轴动作过程中发生异常而停止

单轴动作过程中发生异常时，将根据异常内容而立即停止或减速停止。
关于各异常的停止方法，请参考 *「11-2-2 异常内容* (P.11-20)」。

● 利用软件限位停止

利用软件限位停止时，在轴参数设定的 " 软件限位功能 " 中设定。
利用软件限位停止的方法可从以下选项中选择。

- 对指令位置有效。减速停止。
- 对指令位置有效。立即停止。
- 对反馈位置有效。减速停止。
- 对反馈位置有效。立即停止。

关于软件限位功能，请参考 *「9-8-5 软件限位功能* (P.9-73)」。

● 因发生运动控制周期超限而停止

运动控制的处理未在 2 个周期内完成，发生运动控制周期超限时，所有轴将立即停止。

使用注意事项

使用 NX 系列 CPU 单元以多运动方式动作时，若任意一个任务发生运动控制周期超限，两个任务中的所有轴都将立即停止。
关于多运动请参考 *「A-5-2 运动控制功能* (P.A-23)」。

● 发生会变为伺服 OFF 的异常

发生变为伺服 OFF 的异常时，将立即停止。伺服 OFF 时，伺服驱动器的动作遵照伺服驱动器的设定。

● 因开始 MC 调试功能而停止

开始了 Sysmac Studio 的 MC 调试功能时，各轴将以 " 最大减速度 " 减速停止。

● 因结束 MC 调试功能而停止

结束了 Sysmac Studio 的 MC 调试功能时，各轴将以 " 最大减速度 " 减速停止。

- 在 Sysmac Studio 的 MC 调试画面中选择 ［结束 MC 调试］。
- 关闭 Sysmac Studio 的 MC 调试画面。
- 结束 Sysmac Studio。

● 因 CPU 单元的模式切换而停止

切换了 CPU 单元的模式时，各轴将以 “ 最大减速度 ” 减速停止。

使用注意事项

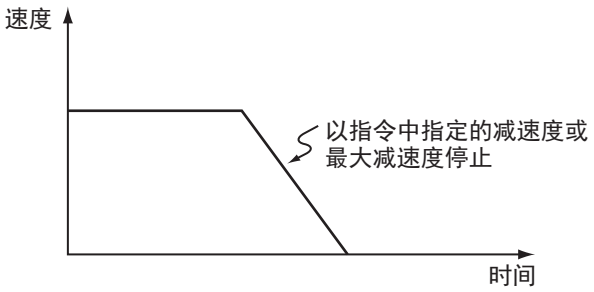
- 正在执行输入变量中包含 “Deceleration （减速度） ” 的同步动作指令时，若发生会导致减速停止的异常，将按照 “Deceleration （减速度） ” 减速停止。
- 正在执行输入变量中不含 “Deceleration （减速度） ” 的同步动作指令时，若发生会导致减速停止的异常，将按照轴参数的最大减速度减速停止。

参考

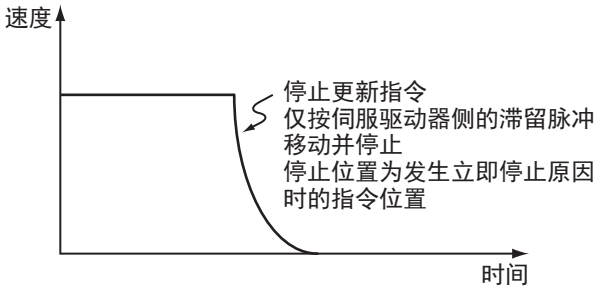
- 从运动模式切换为程序模式时，动作中的运动控制指令将中断。此时，指令的输出变量 “CommandAborted （执行中断） ” 保持 FALSE。切换为程序模式后，仍保持伺服 ON/OFF 状态。
- 从运行模式切换为程序模式并减速停止过程中，若恢复为运行模式，运动控制模式的输出变量将清除，因此运动控制模式的输出变量 “CommandAborted （执行中断） ” 保持 FALSE。
- 正在利用 MC_SaveCamTable （凸轮表保存）指令保存时，继续执行保存处理。
- 正在利用 MC_GenerateCamTable （凸轮表生成）指令执行凸轮表生成时，继续执行生成处理。

停止方法

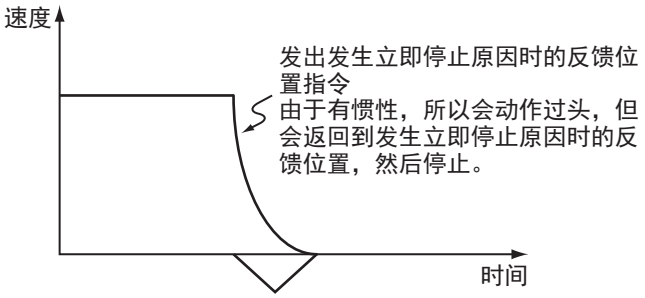
● 减速停止指令值



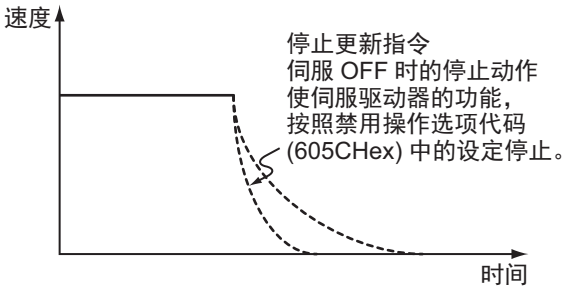
● 立即停止指令值



● **立即停止指令值，同时执行偏差计数器复位**



● **立即停止指令值，同时执行伺服 OFF**



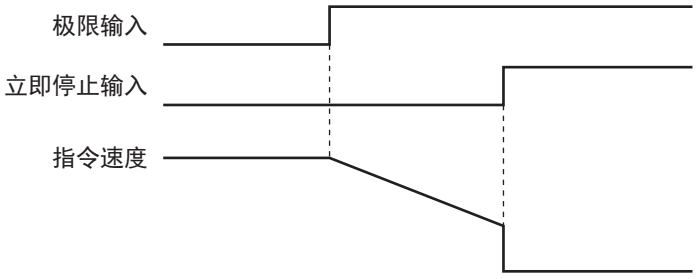
停止的优先顺序

停止的优先顺序如下表所示。停止时发生优先顺序更高的停止时，将切换为优先顺序更高的停止方法。

停止方法	优先顺序 （数字越大，优先顺序越高）
立即停止指令值，同时执行伺服 OFF	4
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	2
减速停止指令值	1

● **例**

下图表示极限输入信号为 ON、减速停止的过程中，立即停止输入变为 ON，立即停止时的示例。



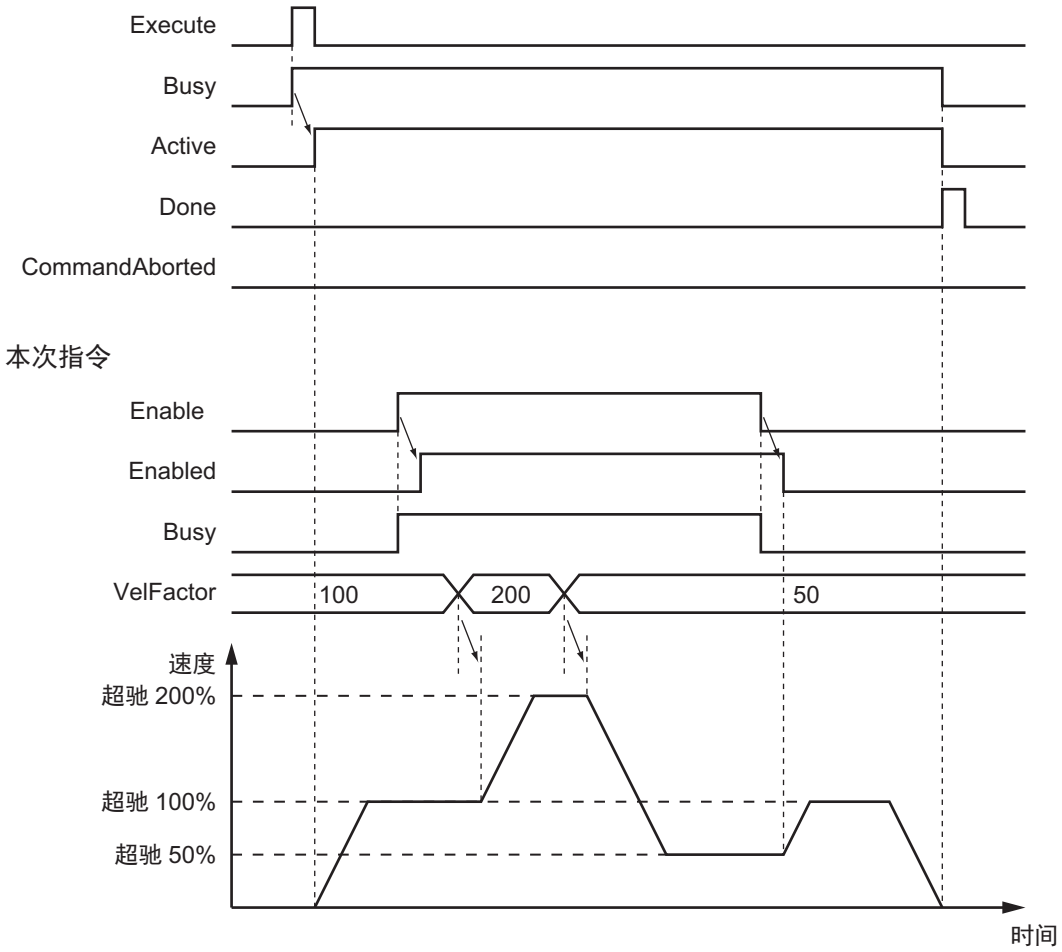
9-1-7 超驰

利用 MC_SetOverride （超驰值设定）指令，对启动中的轴动作设定超驰值的功能。
超驰值为相对于目标速度的比率，可设定为 0 ～ 500%。
若设定为目标速度的 0%，将在速度 “0” 的状态下停止，继续保持动作状态。

超驰为有效期间，将导入设定的超驰值。
若将超驰设为无效，超驰值将恢复为 100%。
根据超驰值变更后的值超出最高速度时，变为各轴的 “ 最高速度 ”。

● MC_MoveAbsolute （绝对值定位）指令的超驰

在 MC_MoveAbsolute （绝对值定位）指令中使用超驰指令时的时序图如下所示。



功能的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_SetOverride （超驰值设定）指令。

9-2 单轴同步控制

下面介绍单轴的同步控制动作。

9-2-1 同步控制的概要

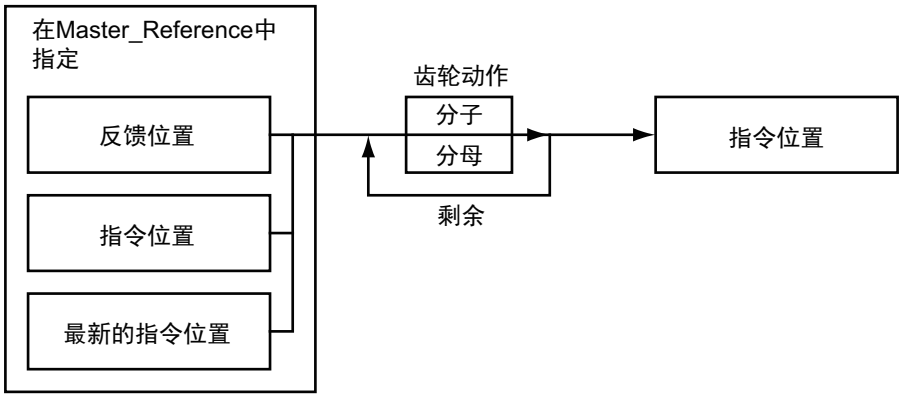
与主轴位置同步对从轴位置进行控制的功能。
主轴中可指定各轴的指令位置或反馈位置。
对从轴的指令速度超出轴参数的［最高速度］时，以最高速度发令。此时，缺少的移动量将分配到下一周期之后并输出。

使用注意事项

使用 NX 系列 CPU 单元以多运动方式动作时，若进行同步控制，构成的主轴和从轴应在同一任务内。
利用 MC_GearIn （齿轮动作开始）指令和 MC_CamIn （凸轮动作开始）指令等同步控制指令，在主轴、从轴中指定了不同任务的轴时，将发生 " 主轴指定不正确 （事件代码 : 54620000 Hex） "。
若要将不同任务中的轴指定为主轴、从轴，请参考 *「9-2-10 在多运动中实现同步控制的方法 (P.9-26)」*。

9-2-2 齿轮动作

设定主轴与从轴间的齿轮比，进行齿轮动作的功能。
利用 MC_GearIn （齿轮动作开始）指令开始齿轮动作，利用 MC_GearOut （齿轮动作解除）指令或 MC_Stop （强制停止）指令解除同步。



可对动作对象的从轴指定齿轮比分子、齿轮比分母、位置类型选择、加速度、减速度。此外，主轴中可指定指令位置、反馈位置、最新指令位置中的任意一个。

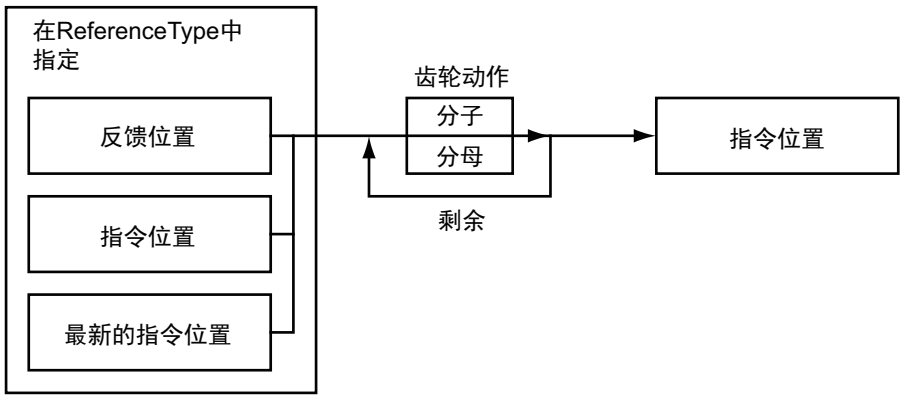
开始动作后，从轴将以主轴速度乘以齿轮比得到的速度作为目标速度，执行加减速动作。
达到目标速度前视为 Catching phase （追赶中），达到后称为 InGear phase （齿轮同步中）。

齿轮比为正数时， Slave （从轴）与 Master （主轴）的移动方向相同，负数时， Slave （从轴）与 Master （主轴）的移动方向相反。

齿轮动作的详情请参考 《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_GearIn （齿轮动作开始）指令、 MC_GearOut （齿轮动作解除）指令及 MC_Stop （强制停止）指令。

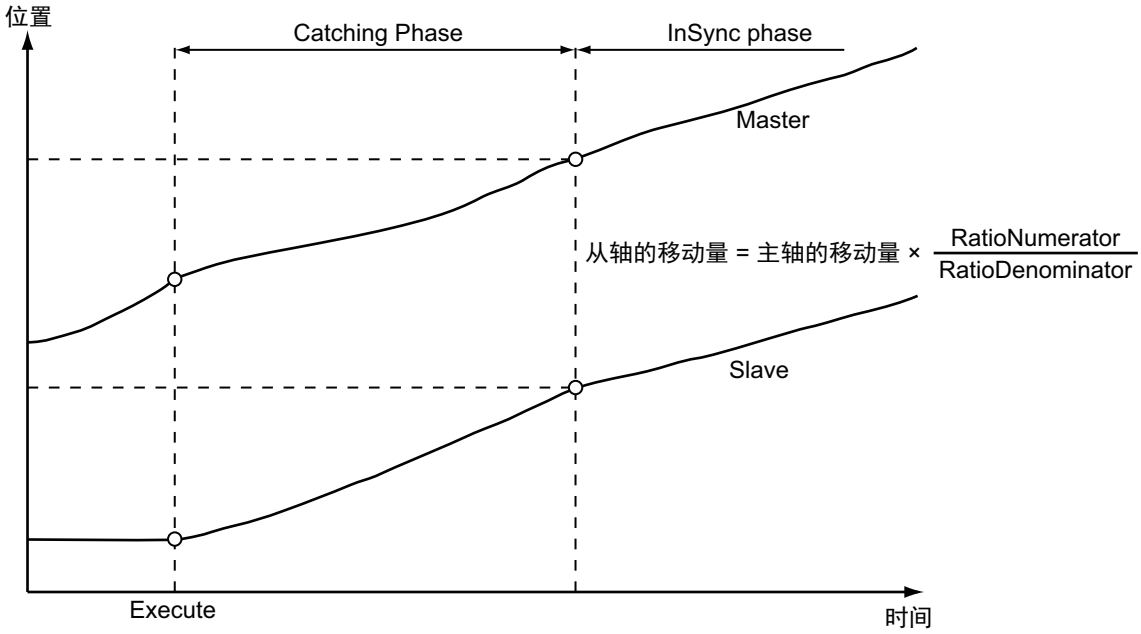
9-2-3 位置指定齿轮动作

设定主轴与从轴间的齿轮比，进行齿轮动作的功能。
在位置指定齿轮动作中，可指定开始同步的主轴位置和从轴位置。
利用 MC_GearInPos （位置指定齿轮动作）指令开始位置指定齿轮动作，利用 MC_GearOut （齿轮动作解除）指令或 MC_Stop （强制停止）指令结束同步。



可对动作对象的从轴指定齿轮比分子、齿轮比分母、位置类型选择、加速度、减速度。此外，主轴中可指定指令位置、反馈位置、最新指令位置中的任意一个。

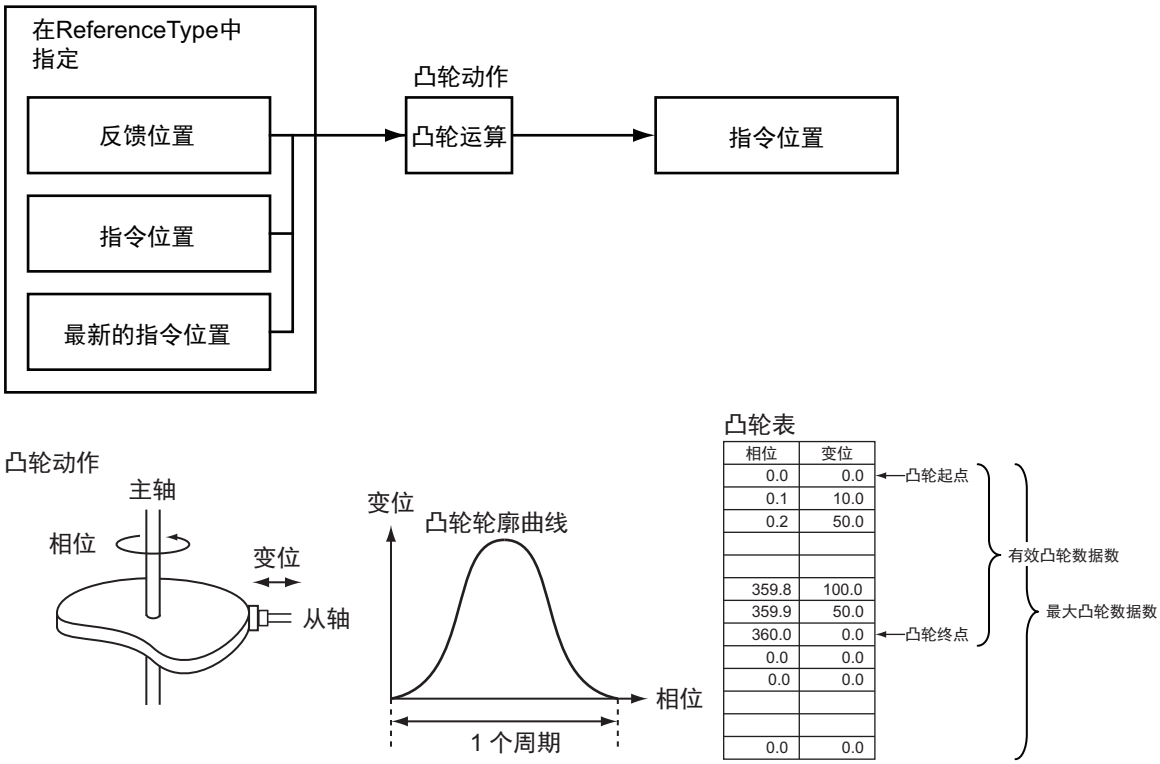
开始动作后，从轴将以主轴速度乘以齿轮比得到的速度为目标，执行加减速动作。
从轴到达从轴同步位置前称为 Catching phase （追赶中），从轴到达从轴同步位置后，变为 InSync （齿轮同步中）。无论哪种情况，从轴都同步到主轴的位置。
齿轮比为正数时， Slave （从轴）与 Master （主轴）的移动方向相同，负数时， Slave （从轴）与 Master （主轴）的移动方向相反。下图表示齿轮比为正数时的动作。



位置指定齿轮动作的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_GearInPos （位置指定齿轮动作）指令、 MC_GearOut （齿轮动作解除）指令及 MC_Stop （强制停止）指令。

9-2-4 凸轮动作

从轴与主轴进行位置同步后，按照凸轮表动作的功能。
利用 MC_CamIn （凸轮动作开始）指令开始凸轮动作，利用 MC_CamOut （凸轮动作解除）指令或 MC_Stop （强制停止）指令解除凸轮动作。
凸轮表使用 Sysmac Studio 的图论编辑器功能创建，并事先下载到 CPU 单元本体中。
下载到 CPU 单元时，使用 Sysmac Studio 的 “ 同步 ” 功能。



在 CPU 单元版本 1.06 以上和 Sysmac Studio Ver.1.07 以上的组合中，正在执行 MC_CamIn （凸轮动作开始）指令时，若指定为共混并多重启动其他 MC_CamIn （凸轮动作开始）指令，可在不停止从轴的状态下，在切换凸轮表的同时保持动作。

凸轮动作的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_CamIn （凸轮动作开始）指令、 MC_CamOut （凸轮动作解除）指令及 MC_Stop （强制停止）指令。
关于凸轮编辑器功能的详情，请参考 🕮《Sysmac Studio Version 1 操作手册 （SBCA-CN5-362）》。

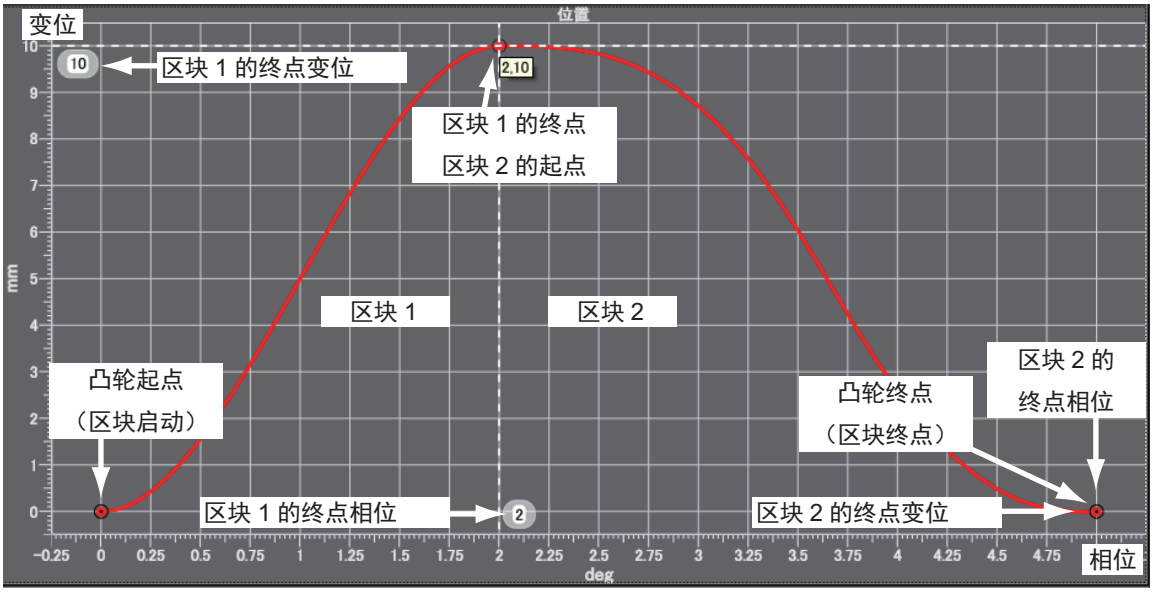
9-2-5 凸轮表

下面介绍凸轮动作中使用的凸轮表。

凸轮表的术语

术语	说明
凸轮动作	有一个主轴和一个从轴，根据凸轮轮廓曲线从主轴的相位计算出从轴的变位后执行动作的功能。
凸轮轮廓曲线	表示实现凸轮动作所需的相位和变位关系的曲线。 在 Sysmac Studio 的凸轮编辑器中创建。将凸轮轮廓曲线下载到 CPU 单元后，可在用户程序中作为凸轮数据变量处理。 下载到 CPU 单元时，使用 Sysmac Studio 的 “ 同步 ” 功能。
凸轮区块	可选择凸轮曲线的区间。 表示前凸轮区块的终点到本凸轮区块的终点。
凸轮曲线	表示凸轮特性的曲线。可按凸轮区块选择凸轮曲线。 利用 Sysmac Studio 计算的从指定点开始的相位宽度、变位宽度，在 Sysmac Studio 上创建实际的凸轮轮廓曲线。 包括直线、等加速度、 Trapecloid 曲线等。
凸轮数据	表示实现凸轮动作所需的、由相位 （主轴）和变位 （从轴）构成的数据。
凸轮数据变量	将凸轮数据转换为结构体排列后的变量。
凸轮表	表示凸轮数据的数据表。 若相位不是升序，将视为错误的凸轮表。
凸轮起点	表示凸轮数据开头的数据。
凸轮终点	表示凸轮数据中有效凸轮数据的末尾。 凸轮终点＜凸轮数据数时，凸轮终点后面的相位和变位为 “0”。
凸轮区块起点	凸轮区块的开始点。在凸轮动作的开始处，与 “ 凸轮起点 ” 为相同的点。凸轮轮廓曲线为连续时，与 “ 凸轮区块终点 ” 为相同的点。
凸轮区块终点	凸轮区块的结束点。在凸轮动作的结束处，与 “ 凸轮终点 ” 为相同的点。凸轮轮廓曲线为连续时，与 “ 凸轮区块起点 ” 为相同的点。 凸轮区块终点的定义为 （横轴、纵轴）＝ （相位终点、变位终点）
原始凸轮数据	在凸轮编辑器中分割凸轮轮廓曲线后创建的凸轮数据。
程序变更凸轮数据	CPU 运行过程中在用户程序上发生变更的凸轮数据。
主轴	凸轮动作的输入轴，可指定为直线模式 / 旋转模式。
从轴	凸轮动作的输出轴，可指定为直线模式 / 旋转模式。
相位	从凸轮表起点位置起主轴的相对量。
变位	根据主轴追踪距离计算的从轴相对量。
有效凸轮数据	凸轮起点以外，相位中包含 “0” 以外数值的凸轮数据。
无效凸轮数据	凸轮起点以外，相位为 “0” 的凸轮数据。
有效凸轮数据数	表示凸轮数据的组数。
最大凸轮数据数	凸轮表可容纳的凸轮数据最大组数。
凸轮数据索引	执行凸轮数据的编号。
凸轮表开始位置	相当于凸轮起点 （相位＝ 0）的主轴绝对位置。
主轴追踪距离	从轴开始凸轮动作时的主轴位置，以绝对位置或相对位置表示。 相对位置以凸轮起点位置为基准。
开始位置方式指定	指定以绝对位置表示主轴追踪距离还是以相对位置表示。
空凸轮数据	终点后可设定的相位和变位为 “0” 的凸轮数据。
连接速度	凸轮轮廓曲线相互连接时的速度。有些曲线无法指定。
连接加速度	凸轮轮廓曲线相互连接时的加速度。有些曲线无法指定。

术语	说明
相位间隔宽度	以相位（横轴）分割凸轮轮廓曲线时的宽度。以相位间隔宽度分割曲线时，各分割点即为凸轮表的凸轮数据。必须对各凸轮区块分别指定。



什么是凸轮表

MC 功能模块中，由主轴的相位和从轴的变为组成的 1 组数据定义为凸轮数据，多个凸轮数据的组合称为凸轮表。

凸轮表在 Sysmac Studio 的凸轮编辑器中创建。此外，还可通过用户程序变更凸轮表中的凸轮数据。

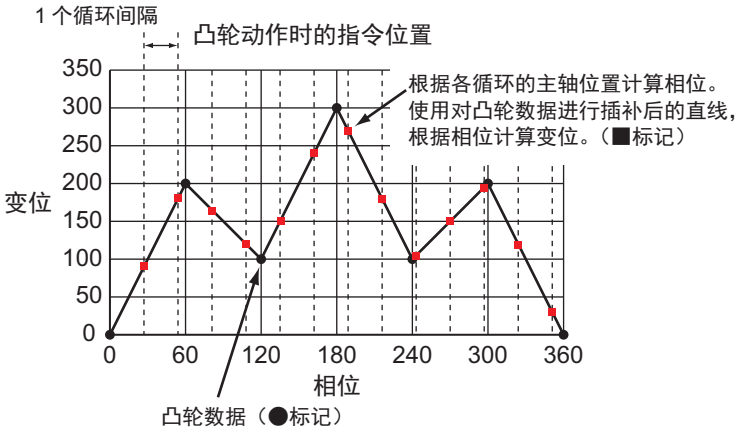
凸轮表中凸轮数据的相位和变位值以从起点 “0.0” 开始的相对量表示。

凸轮动作过程中，对与主轴相位相邻的 2 个凸轮数据进行直线插补后的变位，即为针对从轴的指令位置。

若增加凸轮表的凸轮数据数，可得到轨迹精度高、平滑的凸轮轮廓曲线。

凸轮表

	相位	变位	凸轮数据索引
凸轮起点 →	0	0	0
凸轮数据 →	60	200	1
	120	100	2
	180	300	3
	240	100	4
	300	200	5
凸轮终点 →	360	0	6



使用注意事项

- 创建时，应确保凸轮表的凸轮数据以相位值升序排列。若相位不是升序，执行凸轮动作指令时，指令将发生异常。
- 凸轮数据变量为全局变量，可从多个任务参照或变更凸轮数据变量的值。从多个任务变更凸轮数据变量的值时，编写的程序应确保多个任务的替换处理不会发生冲突。
- 使用 " 全局变量的任务间排他控制 " 功能对凸轮数据变量进行排他控制时，请勿用排他对象的凸轮数据变量启动运动指令。将发生轻度故障等级的异常 " 凸轮表指定不正确 （事件代码 : 54390000 Hex） "。

凸轮表规格

项目	说明
每 1 凸轮表的最大凸轮数据数	65,535 点
所有凸轮数据的大小上限	1,048,560 点 *1
凸轮表数的上限	640 表 *2
凸轮动作的切换	利用运动控制指令可切换为其他的凸轮动作
凸轮数据的变更	可在用户程序中编辑 可利用凸轮表生成指令替换 *3
凸轮数据的保存	可利用凸轮表保存指令保存到非易失性存储器中
凸轮数据的附属信息	可下载、上传凸轮编辑器显示用的信息 *4
凸轮数据反映到主存储器的时间	• 从 Sysmac Studio 下载时 • 电源接通时

*1. 1 个凸轮表 65,535 点时，最大为 16 个凸轮。分辨率为 0.1deg 时， 1 个凸轮表为 3600 点，最大为 291 个凸轮。
*2. 总容量请控制在 10MB 以下。
*3. 凸轮表生成指令可在 CPU 单元版本 1.08 以上和 Sysmac Studio Ver.1.09 以上的组合中使用。
*4. " 下载 " 和 " 上传 " 时使用 Sysmac Studio 的 " 同步 " 功能。

凸轮表的数据类型

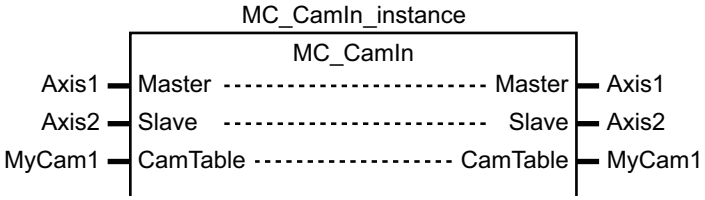
凸轮表需要对凸轮数据结构体进行排列声明后使用。凸轮数据结构体已事先进行如下类型声明。

```
TYPE
    (* 凸轮数据结构体 *)
    _sMC_CAM_REF :
    STRUCT
        Phase     : REAL;     (* 相位 *)
        Distance  : REAL;     (* 变位 *)
    END_STRUCT;
END_TYPE
```

必须在 Sysmac Studio 的凸轮编辑器中创建凸轮数据，并指定凸轮表名称和凸轮数据数 （排列的大小）。
例如，凸轮表名称为 “MyCam1”、凸轮数据数为 1000 个时，声明如下。

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

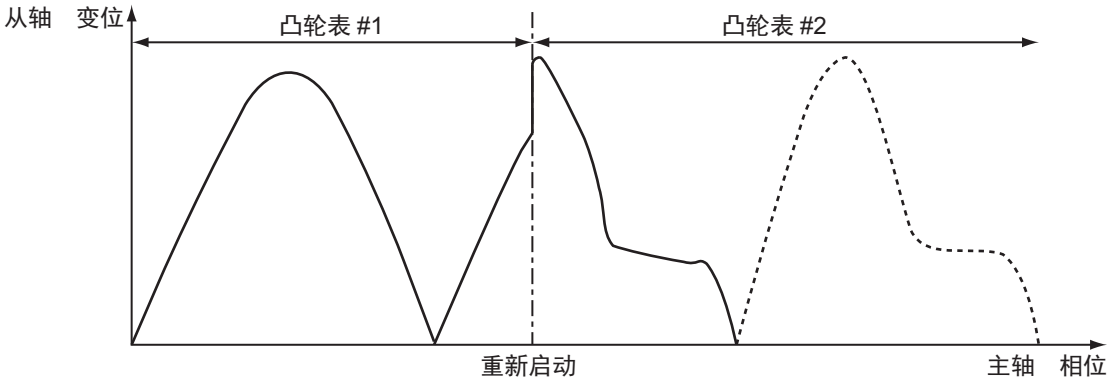
在凸轮动作指令中指定为 “MyCam1” 时，记述如下。假设主轴为 Axis1，从轴为 Axis2。



指定的凸轮表在控制器中不存在时，将检测为异常。此外，可在多个轴中指定同一凸轮表。

凸轮表的切换

在凸轮动作过程中重新执行凸轮动作指令，可切换凸轮表。
切换后，将按照重启时指定的凸轮表执行凸轮动作。
MC_CamIn （凸轮动作开始）指令的输出变量 “EndOfProfile （终点执行）” 及 “Index （索引）” 按照切换后的凸轮表输出。



使用注意事项

- 切换目标的凸轮表需要事先保存到非易失性存储器中。
- 若在凸轮动作过程中切换凸轮，速度将变得不连续。请调整切换时间，避免发生过度的速度不连续。

凸轮数据的读写和凸轮表的保存

与其他变量一样，还可通过用户程序读写凸轮数据。
例如，凸轮表名称为 “MyCam1” 的 1 号排列要素，可指定为相位 “MyCam1[0].Phase”、变位 “MyCam1[0].Distance”。
用户程序中替换过的凸轮数据可通过 MC_SaveCamTable （凸轮表保存）指令，作为凸轮表保存到 CPU 单元的非易失性存储器中。

使用注意事项

- 启动凸轮表保存指令前，若关闭了 CPU 单元的电源、或未通过本指令正常完成保存、或通过 Sysmac Studio 执行了下载时，替换的凸轮数据将丢失。
- 用 CPU 单元上的用户程序替换凸轮数据时，请注意防止替换数据丢失。
- 非易失性存储器中保存的凸轮数据可利用 Sysmac Studio 的上传功能导入。
- “ 下载 ” 和 “ 上传 ” 时使用 Sysmac Studio 的 “ 同步 ” 功能。

排列指定请参考 🕮《NJ/NX 系列 CPU 单元 用户手册 软件篇 （SBCA-CN5-359）》。

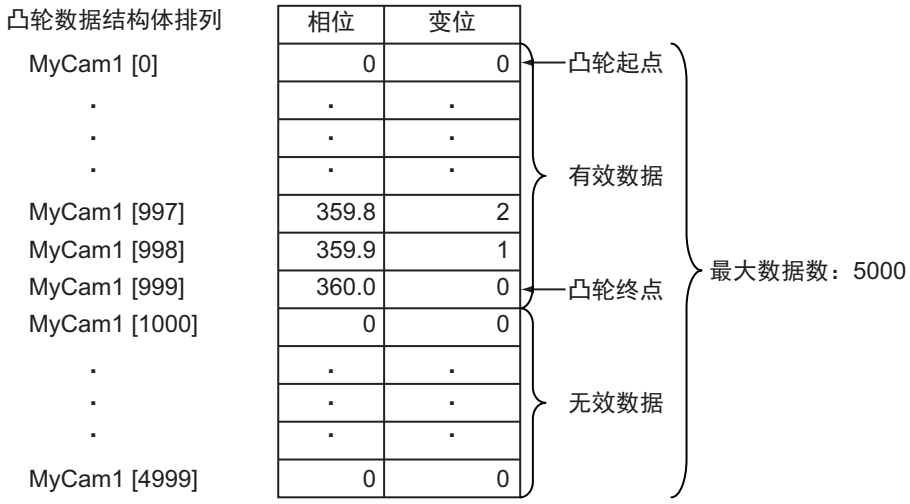
凸轮表保存指令的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_SaveCamTable （凸轮表保存）指令。

凸轮表属性更新

MC 功能模块中，不需要识别凸轮表的凸轮终点。
若在凸轮动作中通过用户程序变更并替换了有效凸轮数据数，需要更新为最新的有效凸轮数据数。
因此，应使用 MC_SetCamTableProperty （凸轮表属性更新）指令。

凸轮终点是指从起点开始搜索凸轮表时，配置在相位首次为 “0” 的凸轮数据的前一个数据。检测到相位 “0” 后的凸轮数据为无效。

例如，以下凸轮表中， MC_SetCamTableProperty （凸轮表属性更新）指令的输出变量 “EndPointIndex （终点索引） ” 为 “999”、 “MaxDataNumber （最大数据数） ” 为 “5000”。



使用注意事项

- 最大凸轮数据数不可通过用户程序变更。
- 变更并替换有效凸轮数据数后，请执行本指令。
  若不更新有效凸轮数据数，凸轮动作或 MC_CamIn 的 EndOfProfile （凸轮周期完成）将无法执行预期动作。

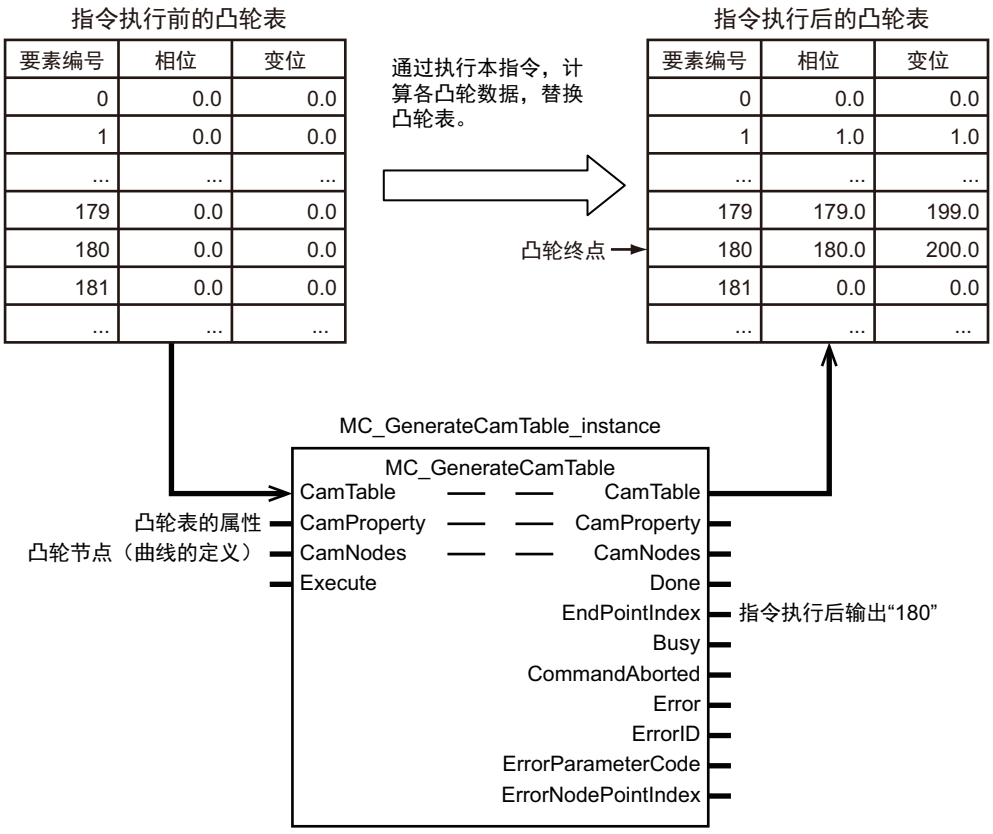
凸轮表属性更新的详情请参考 📖《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_SetCamTableProperty （凸轮表属性更新）指令。

凸轮表生成

在 CPU 单元版本 1.08 以上和 Sysmac Studio Ver.1.09 以上的组合中，可通过用户程序中的 MC_GenerateCamTable （凸轮表生成）指令生成凸轮表。

根据 MC_GenerateCamTable （凸轮表生成）指令的 CamProperty （凸轮属性）和 CamNodes （凸轮节点）中指定的值计算凸轮数据的值，将指令的输入输出变量 CamTable （凸轮表）替换为指定的凸轮数据变量。

替换完成后，更新凸轮表的终点索引，将凸轮终点的要素编号输出到 EndPointIndex （终点索引）中。MC_GenerateCamTable （凸轮表生成）指令完成后，不需要执行 MC_SetCamTableProperty （凸轮表属性更新）指令。



凸轮数据变量为凸轮数据结构体 _sMC_CAM_REF 型的排列变量，在 Sysmac Studio 的凸轮编辑器中创建。

在 CamProperty （凸轮属性）中指定凸轮属性变量。凸轮属性变量为凸轮属性结构体 _sMC_CAM_PROPERTY 型的变量，在 Sysmac Studio 的全局变量表中作为用户定义变量创建，或在 Sysmac Studio 的凸轮数据设定中创建。

在 CamNodes （凸轮节点）中指定凸轮节点变量。凸轮节点变量为凸轮节点结构体 _sMC_CAM_NODE 型的排列变量，在 Sysmac Studio 的全局变量表中作为用户定义变量创建，或在 Sysmac Studio 的凸轮数据设定中创建。

凸轮属性变量和凸轮节点变量统称为 “ 凸轮定义变量 ”。

将凸轮定义变量作为用户定义变量创建时，变量保持属性的初始值为 “ 不保持 ”。变更变量的值并切换为程序模式后或重新接通电源再次使用时，请将变量的保持属性设定为 “ 保持 ”。若每次都会通过显示器设定，直接用 “ 不保持 ” 亦可。
在 Sysmac Studio 的凸轮数据设定中创建了凸轮定义变量时，变量的保持属性固定为 “ 保持 ”。

通过显示器设定要在 MC_GenerateCamTable （凸轮表生成）指令中指定的值，无需使用 Sysmac Studio，也可创建凸轮数据变量或调整凸轮动作。
下面介绍调整凸轮动作的步骤概要。

1. **首先，在用户程序中创建以下处理。**
   - 将显示器中设定的凸轮定义变量的值指定到凸轮表生成指令中的处理
   - 在显示器上以图表显示通过凸轮表生成指令生成的凸轮数据变量的处理
   - 在显示器上显示 EndPointIndex （终点索引）值的处理
2. **在显示器上设定凸轮定义变量的值。**
3. **执行凸轮表生成指令。**
4. **在显示器上确认所生成凸轮表的形状和终点索引的值。**
5. **若凸轮表的形状和凸轮数据数没有问题，则执行凸轮动作。**
6. **确认凸轮动作的结果，研究凸轮定义变量值的变更。**
7. **重复 2 ～ 6 的步骤。**

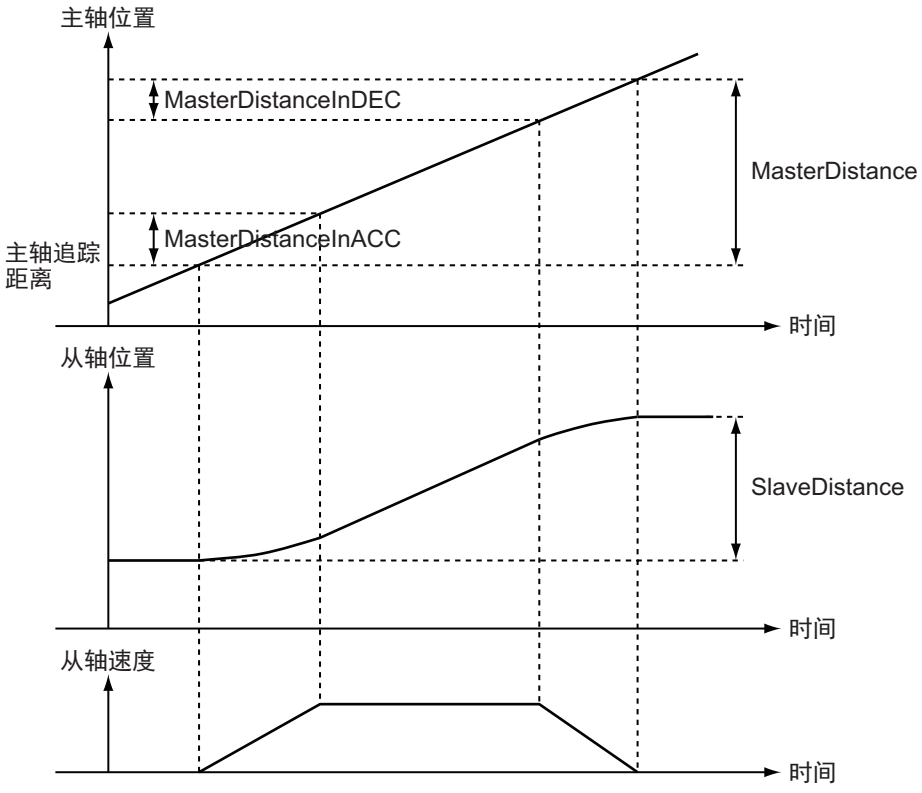
凸轮定义变量的详情和凸轮表生成指令的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_GenerateCamTable （凸轮表生成）指令。

关于通过 Sysmac Studio 进行凸轮定义变量的创建和传送，请参考 🕮《Sysmac Studio Version 1 操作手册 （SBCA-CN5-362K 以后）》。

9-2-6 梯形模式凸轮动作

指定的从轴与指定的主轴根据梯形曲线同步进行定位的功能。
电子凸轮的一种，但不使用凸轮编辑器中创建的凸轮表。
启动 MC_MoveLink （梯形模式凸轮）指令，开始动作。
若要停止动作中的轴，则使用 MC_Stop （强制停止）指令。
以动作对象的轴为 “Slave （从轴） ”，指定 “Master （主轴） ”、 “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） ”、 “SlaveDistance （从轴移动距离） ”、 “MasterStartDistance （主轴追踪距离） ”。
主轴中可指定指令位置或反馈位置。
同步动作的开始条件可指定为 “ 指令开始时 ”、 “ 检测到触发时 ”、 “ 主轴达到主轴追踪距离时 ” 中的任意一个。

如下图所示，从轴的速度和位置根据主轴和从轴的移动量之比决定。
下图的同步开始位置表示同步开始条件变为有效时的位置。

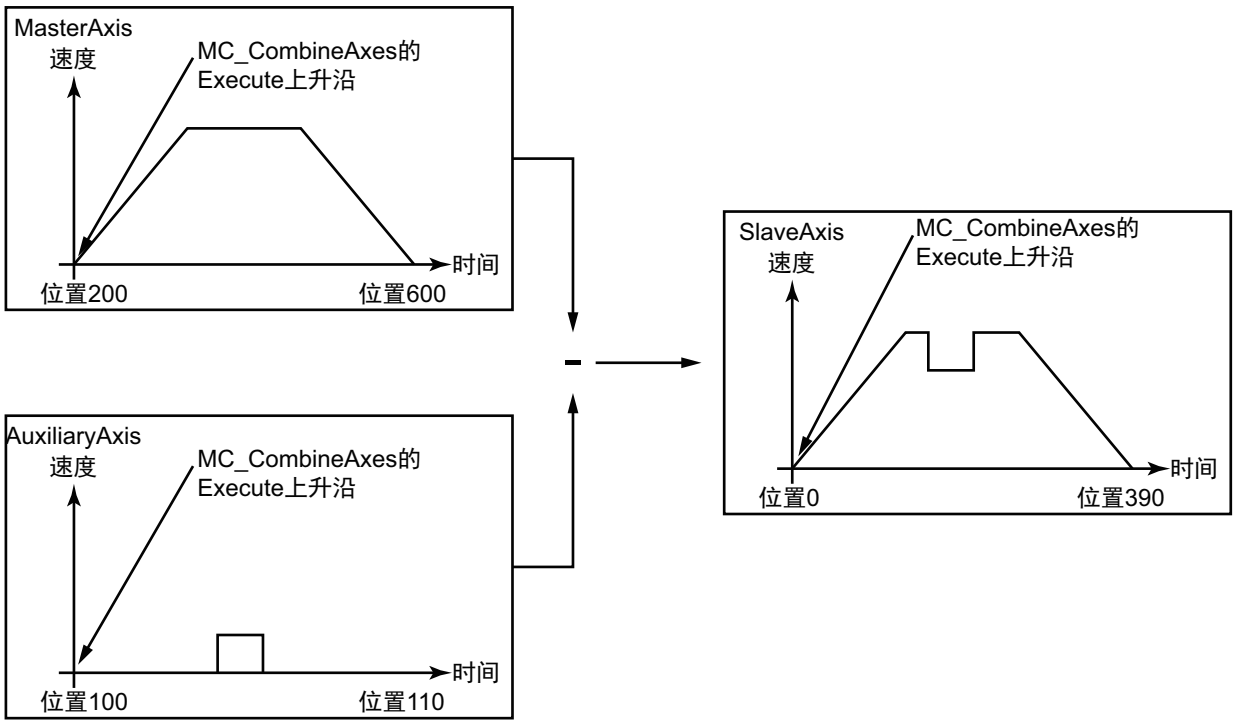


梯形模式凸轮动作的详情请参考 《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_MoveLink （梯形模式凸轮）指令及 MC_Stop （强制停止）指令。

9-2-7 加减法定位

将 2 个位置相加或相减得到的数值，作为从轴的指令位置的功能。
启动 MC_CombineAxes （加减法定位）指令，开始动作。
若要停止动作中的轴，则使用 MC_Stop （强制停止）指令。

下图为减法时的动作示例。
SlaveAxis （从轴）指令当前位置 = MasterAxis （主轴）指令当前位置 － AuxiliaryAxis （辅助轴）指令当前位置

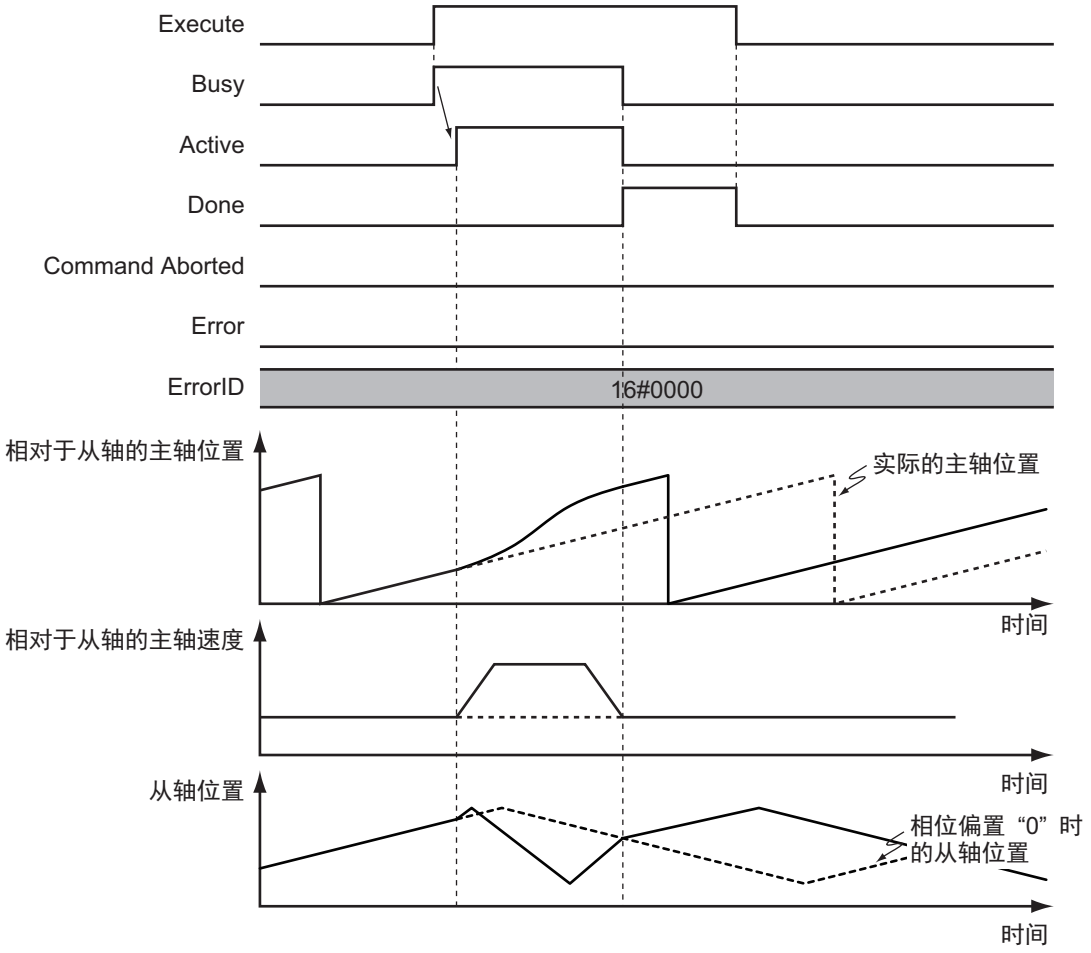


加减法定位动作的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_CobineAxes （加减法定位）指令及 MC_Stop （强制停止）指令。

9-2-8 主轴相位补偿

针对动作中的指令，相对从轴进行主轴相位补偿的功能。
相对于从轴的补偿量为相对量，在同步状态下，从轴针对主轴的相对量同步。
启动 MC_Phasing （主轴相对值相位补偿）指令，可对同步控制指令进行相位补偿。

MC_Phasing （主轴相对值相位补偿）指令中可指定 “ 相位补偿量 ”、“ 目标速度 ”、“ 加速度 ”、“ 减速度 ”、“ 跃度 ”。



可进行主轴相位补偿的同步控制指令及主轴相位补偿功能的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_Phasing （主轴相对值相位补偿）指令。

9-2-9 从轴位置补偿

对同步动作中的从轴进行位置补偿的功能。
针对同步控制中的从轴，将根据输入变量的值计算得到的偏置量加上指令当前位置，输出到伺服驱动器中。
即使 MC 功能模块对 2 个轴发出相同移动量的指令，也可能发生机械性偏差，导致移动量不同，这里可对这样的偏差进行补偿。
启动 MC_SyncOffsetPosition （周期性同步位置偏置补偿）指令，可对同步控制中的从轴进行位置补偿。

从轴位置补偿功能的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_SyncOffsetPosition （周期性同步位置偏置补偿）指令。

9-2-10 在多运动中实现同步控制的方法

在 MC 功能模块的标准功能中，若执行多运动的不同任务中分配的轴间同步控制指令，将发生 “ 主轴指定不正确 （事件代码 : 54620000 Hex） ”。
但是，使用 MC_PeriodicSyncVariables （轴变量任务间固定周期同步）指令，可对主固定周期任务中控制的主轴和固定周期任务 （执行优先度 5）中控制的从轴进行同步控制。

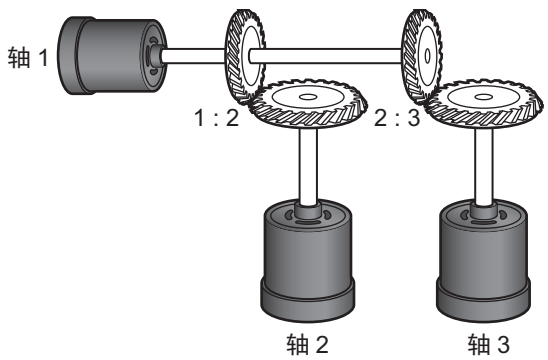
对主轴和从轴进行同步的凸轮动作和齿轮动作可使用以下组合。

主轴的任务	从轴的任务	
	主固定周期任务	固定周期任务 （执行优先度 5）
主固定周期任务	利用运动控制指令同步	在主固定周期任务中执行 MC_PeriodicSyncVariables （轴变量任务间固定周期同步）指令，在固定周期任务 （执行优先度 5）中使用 “ 虚拟主轴 ” 即可实现。
固定周期任务 （执行优先度 5）	不可使用	利用运动控制指令同步

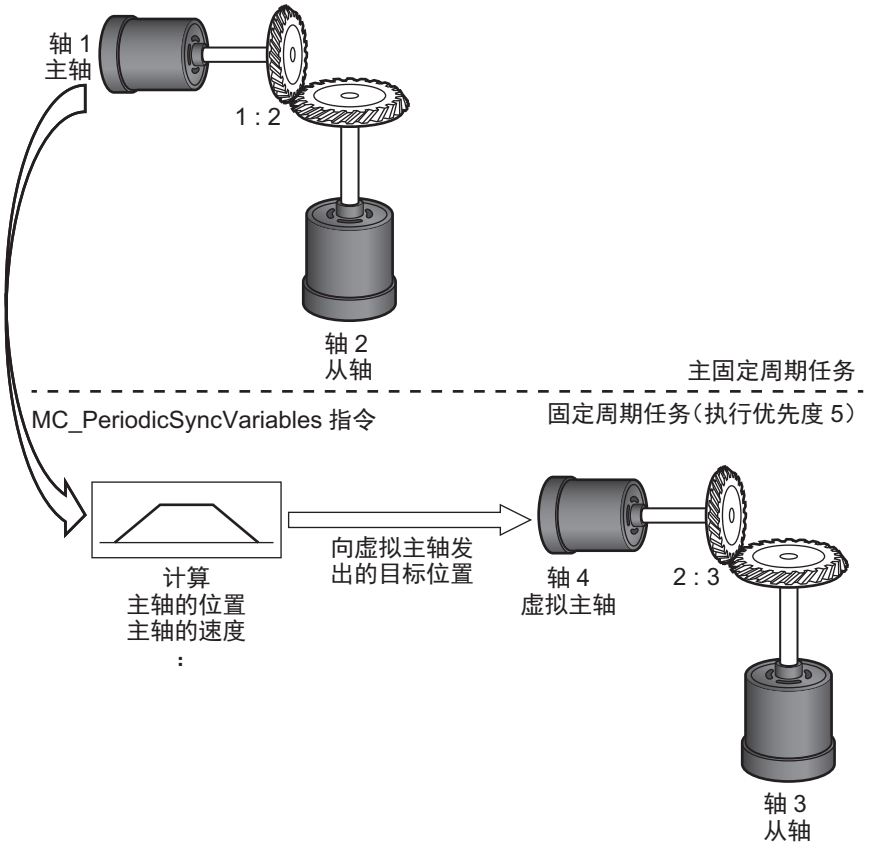
动作示例的轴构成

如下图所示，针对主轴 （轴 1），将需要高速高精度的从轴 （轴 2）分配到主固定周期任务中，将无精度要求的从轴 （轴 3）分配到固定周期任务 （执行优先度 5）中进行控制。同时，主轴 （轴 1）分配到主固定周期任务中。

● **物理的轴构成**



● **逻辑的轴构成**



分别在主固定周期任务和固定周期任务 （执行优先度 5）中编入程序，实现上述应用的动作。

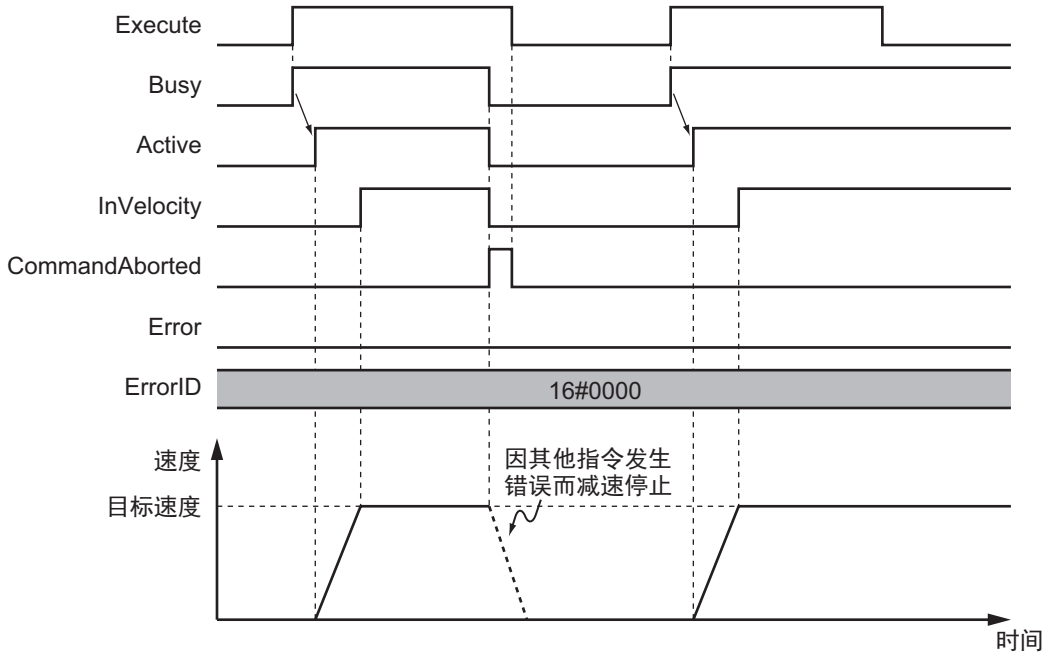
MC_PeriodicSyncVariables （轴变量任务间固定周期同步）指令的详情请参考 📖《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。

9-3 单轴速度控制

下面介绍单轴的速度控制动作。
使用 NX 系列脉冲输出单元时的区别请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

9-3-1 速度控制

速度控制是指以指令速度持续输送轴的功能。
还可指定加速度、减速度及跃度。
若要停止轴，则使用 MC_Stop （强制停止）指令或启动其他动作指令。
若目标速度指定为 “0”，轴不会移动，但轴的状态会变为轴动作中。
速度控制过程中，若因其他运动控制指令的多重启动而中断，将在达到目标速度的状态下切换动作。



在 MC 功能模块中，将发出目标位置指令，使伺服驱动器达到作为位置控制模式指定的目标速度。
在伺服驱动器中，位置控制回路为有效，因此一旦因外部干扰导致速度低于指令速度而发生位置偏差时，为了消除位置偏差，将发生速度变动。

功能的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_MoveVelocity（速度控制）指令。

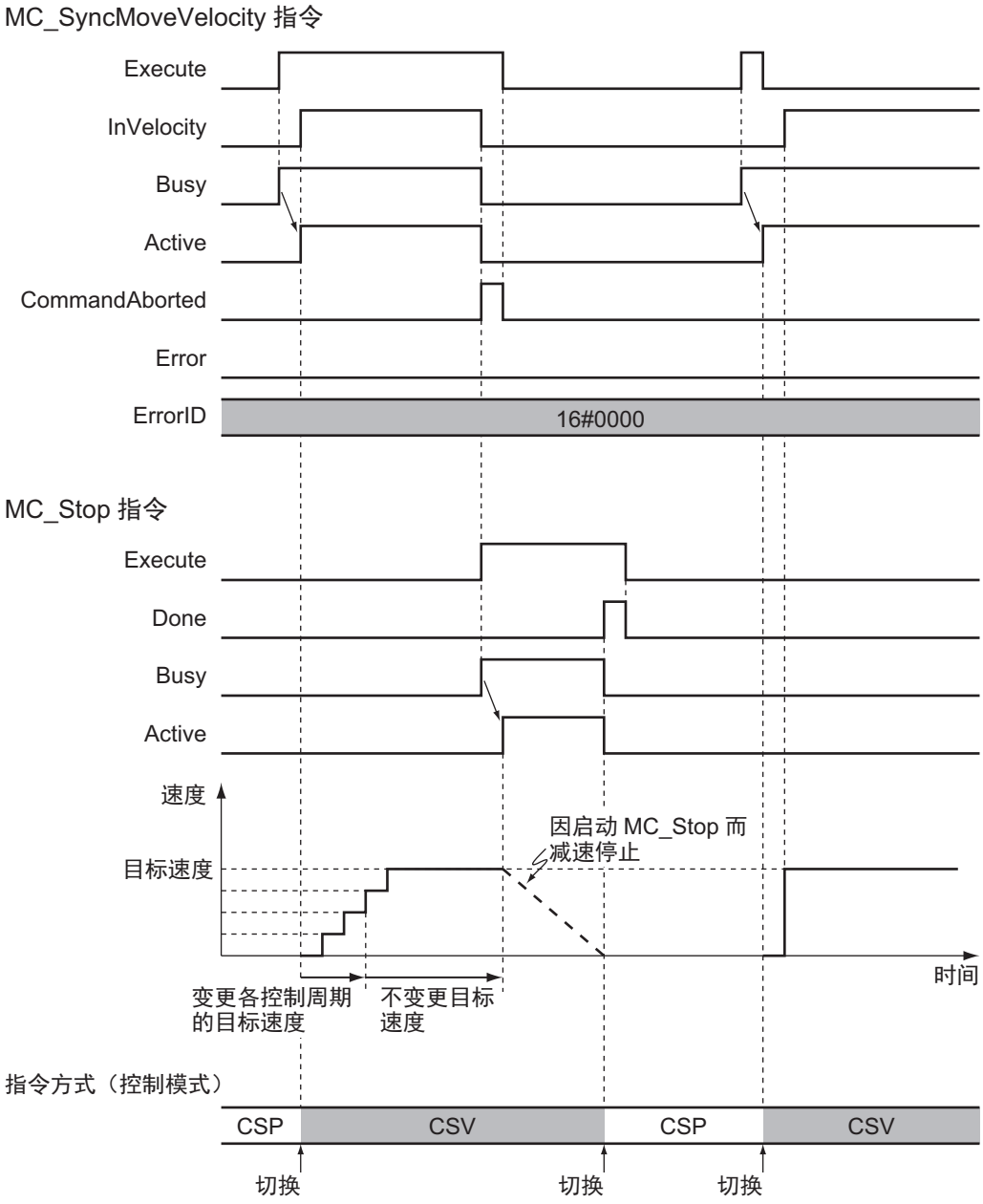
9-3-2 周期性同步速度控制

将伺服驱动器的控制模式设为速度控制模式，在每个控制周期输出指令速度的功能。

使用注意事项

使用 NX 系列脉冲输出单元时，无法使用本功能。

若要停止轴，则使用 MC_Stop （强制停止）指令或启动其他动作指令。
若目标速度指定为 “0”，轴不会移动，但轴的状态会变为轴动作中。



在伺服驱动器中，通过速度控制回路接收指令，因此即使因外部干扰导致速度低于指令速度，也不会为了消除位置偏差而发生速度变动。

功能的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_SyncMoveVelocity （周期性同步速度控制）指令。

9-4 单轴扭矩控制

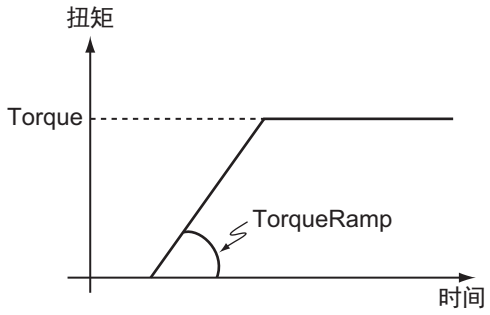
扭矩控制是指持续发生指令扭矩的功能。
达到 Torque （目标扭矩）前的扭矩变量率还能以 TorqueRamp （扭矩斜率）指定。

使用注意事项

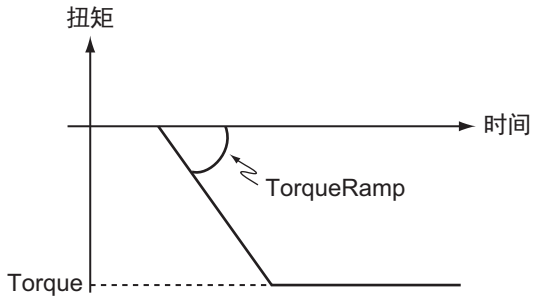
- 扭矩控制时，为确保安全，请务必设定速度限制值。
- 使用 NX 系列脉冲输出单元时，无法使用本功能。

若要停止轴，则使用 MC_Stop （强制停止）指令或启动其他动作指令。
若 Torque （目标扭矩）指定为 “0”，轴不会移动，但轴的状态会变为轴动作中。

● 方向指定 = 正方向时



● 方向指定 = 负方向时



MC 功能模块中使用伺服驱动器的扭矩控制模式。
伺服驱动器通过扭矩控制回路接收来自 MC 功能模块的扭矩指令值并进行控制。
可在运动控制指令的输入变量 “Velocity （限制速度） ” 中指定伺服驱动器侧的 “ 速度限制值 ”。这样在扭矩控制模式下，当点击负载变小时，可限制电机的高速旋转。

功能的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_TorqueControl （扭矩控制）指令。

9-5 单轴控制的共通功能

下面介绍单轴控制的共通功能。
使用 NX 系列脉冲输出单元时的区别请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

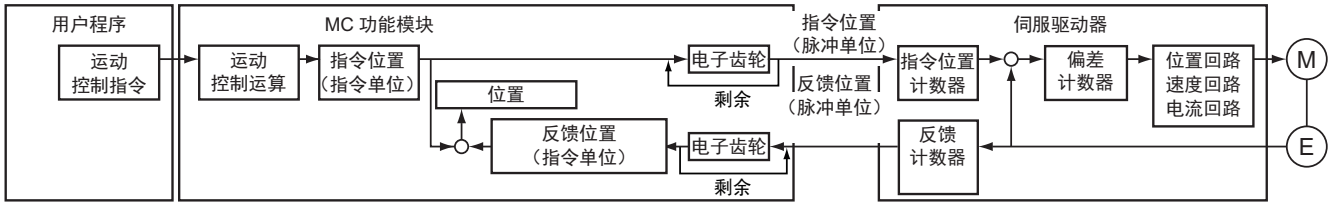
9-5-1 位置

位置的种类

MC 功能模块中有以下两类位置。

位置类别	定义
指令位置	MC 功能模块为了控制轴而输出的位置。
反馈位置	从伺服驱动器或计数器输入的实际位置。

EtherCAT 从站伺服驱动器时的指令位置和反馈位置的关系如下图所示。



指令位置和反馈位置共享以下项目。

项目	指令位置	反馈位置
计数模式	设定直线模式或旋转模式。	变为与指令位置相同的计数模式。
位置的单位	设定为 mm、 μm、 nm、 inch、 degree、 pulse 中的一个。	变为与指令位置相同的单位。
软件限位	设定软件的可动作范围。	变为与指令位置相同的范围。
当前位置变更	将当前位置变更为任意位置。	将同时设定与指令位置相同的值。*1
原点确定	保持原点确定或原点未确定状态。	变为与指令位置相同的原点状态。

*1. 若变更前有位置偏差，将在反馈位置中设定保存了位置变差的值。

参考

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

位置相关的轴参数

参数名称	功能	设定范围	初始值
位置宽度	设定定位完成宽度 （单位：指令单位）	双精度实数型的正数、 0	10
位置检查时间	以毫秒为单位设定完成定位的检查时间。 若设定为 “0”，仅在原点复位中确定原点时继续执行定位完成检查，不是原点复位时，不执行定位完成检查。 （单位：ms）	0 ～ 10000	0
软件限位功能	选择软件限位的功能。 0: 无效 1: 对指令位置有效，减速停止 2: 对指令位置有效，立即停止 3: 对反馈位置有效，减速停止 4: 对反馈位置有效，立即停止	0 ～ 4	0
正方向软件限位	设定正方向的软件限位值。（单位：指令单位）	双精度实数型的负数、正数、0	2147483647
负方向软件限位	设定负方向的软件限位值。（单位：指令单位）	双精度实数型的负数、正数、0	-2147483648
位置偏差超限值	设定位置偏差超限的阈值。 若设定为 “0”，位置偏差超限检查为无效。 （单位：指令单位）	双精度实数型的正数、 0	0
位置偏差警告值	设定位置偏差警告的阈值。 若设定为 “0”，位置偏差警告检查为无效。 （单位：指令单位）	双精度实数型的正数、 0 （小于位置偏差超限值）	0

指定轴动作的目标位置

实际定位动作的位置和距离在运动控制指令的输入变量 “Position （目标位置） ”、 “Distance （移动距离） ” 中指定。

位置监视

在程序中，可通过读取轴变量来监视位置。
此外，以 _MC_AX[*] 为例对变量名称进行说明， _MC1_AX[*]、 _MC2_AX[*] 亦同。

变量名称	数据类型	名称	功能
_MC_AX[0-255].Cmd.Pos	LREAL	指令当前位置	指令位置的当前值。 伺服 OFF 时及位置控制模式以外时，输出反馈当前位置。
_MC_AX[0-255].Act.Pos	LREAL	反馈当前位置	反馈位置的当前值。

9-5-2 速度

速度的类型

MC 功能模块中使用的轴的速度有以下 2 种。

速度类型	定义
指令速度	MC 功能模块为了控制轴而输出的速度。
反馈速度	根据来自伺服驱动器或计数器的反馈位置，在 MC 功能模块内部计算得到的速度。*1

*1. 将反馈速度 (606CHex) 映射到 PDO 映射中，并分配到了 “ 反馈当前速度 ” 功能时，显示该值。

速度的单位

速度单位为 “ 指令单位 /s”。

指令单位为位置的显示单位和通过电子齿轮进行单位转换后的值。

速度相关的轴参数

参数名称	功能	设定范围	初始值
最高速度	指定轴的最高速度。 轴动作指令中指定了超出最高速度的目标速度时，以最高速度动作。	双精度实数型的正数	400,000,000
启动速度 *1	设定各轴的启动速度。 设定时请勿超出最高速度。 （单位：指令单位 /s）	双精度实数型的正数	0
点动最高速度	设定各轴的点动最高速度。*2 设定时请勿超出最高速度。 （单位：指令单位 /s）	双精度实数型的正数	1,000,000
速度警告值	设定相对于最高速度的比例，作为各轴输出速度警告的标准。 若设定为 “0”，将不输出速度警告。（单位：%）	0 ～ 100	0
反馈速度过滤器时间常数	以毫秒为单位设定计算反馈速度移动平均值的时间。 若设定为 “0”，将不计算移动平均值。（单位：ms） 轴的速度较慢时，需要减少反馈当前速度变动的情况下使用。	0 ～ 100	0

*1. 可在 CPU 单元版本 1.05 以上和 Sysmac Studio Ver.1.06 以上的组合中使用。
*2. 指定的速度指令值超出点动最高速度时，按点动最高速度发出指令。

指定轴动作的目标速度

实际定位动作的速度在运动控制指令的输入变量 “Velocity （目标速度） ” 中指定。

速度监视

在程序中，可通过读取轴变量来监视速度。
此外，以 _MC_AX[*] 为例对变量名称进行说明， _MC1_AX[*]、 _MC2_AX[*] 亦同。

变量名称	数据类型	名称	功能
_MC_AX[0-255].Cmd.Vel	LREAL	指令当前速度	指令速度的当前值。 正方向移动时加 + 号，负方向移动时加 - 号。
_MC_AX[0-255].Act.Vel	LREAL	反馈当前速度	反馈速度的当前值。 正方向移动时加 + 号，负方向移动时加 - 号。

9-5-3 加速和减速

加速度和减速度的单位

加速度和减速度的单位为 " 指令单位 /s^2"。
指令单位为位置的显示单位和通过电子齿轮进行单位转换后的值。

加速度和减速度相关的轴参数

参数名称	功能	设定范围	初始值
最大加速度	设定轴动作指令时各轴的最大加速度。 若设定为 "0"，将没有加速度的限制。 （单位：指令单位 /s^2）	双精度实数型的正数、 0	0
最大减速度	设定轴动作指令时各轴的最大减速度。 若设定为 "0"，将没有减速度的限制。 （单位：指令单位 /s^2）	双精度实数型的正数、 0	0
加减速超限	在轴的加减速控制中，为了优先停止在目标位置，会发生减速超限，导致超出最大加减速度，指定这种情况下的动作。 0: 剧烈加减速 （将共混切换为缓冲）*1 1: 剧烈加减速 2: 异常停止 *2	0 ～ 2	0
加速度警告值	设定相对于最高加速度的比例，作为各轴输出加速度警告的标准。 若设定为 "0"，将不输出加速度警告。（单位：%）	0 ～ 100	0
减速度警告值	设定相对于最高减速度的比例，作为各轴输出减速度警告的标准。 若设定为 "0"，将不输出减速度警告。（单位：%）	0 ～ 100	0

*1. 单元版本 1.10 以上的 CPU 单元中，无法将共混切换为缓冲。详情请参考 「*9-5-7 运动控制指令的多重启动 （缓冲模式）*(P.9-45)」。
*2. 单元版本 1.10 以上的 CPU 单元中，发生共混动作时，将视为发生异常而停止。详情请参考 「*9-5-7 运动控制指令的多重启动 （缓冲模式）*(P.9-45)」。

指定轴动作的加速度和减速度

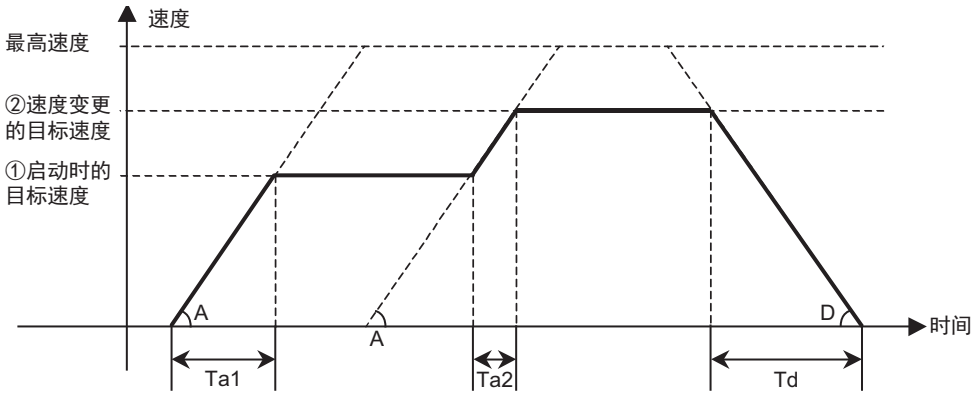
实际定位动作的加速度及减速度在运动控制指令的输入变量 “Acceleration （加速度） ” 及 “Deceleration （减速度） ” 中指定。

加速度和减速度的监视

在程序中，可通过读取轴变量来监视加速度和减速度。
此外，以 _MC_AX[*] 为例对变量名称进行说明， _MC1_AX[*]、 _MC2_AX[*] 亦同。

变量名称	数据类型	名称	功能
_MC_AX[0-255].Cmd.AccDec	LREAL	指令当前加减速度	指令加减速度的当前值。 加速时加 + 号，减速时加 - 号。

加减速动作示例



【启动时】
Ta1 ： 实际的加速时间
A ： 加速度

【速度变更时】
Ta2 ： 实际的加速时间
A ： 加速度

【减速时】
Td ： 实际的减速时间
D ： 减速度

指令的移动量较小时或加减速度较小时，可能无法达到目标速度。
若重新执行运动控制指令，以变更后的新加速度 （减速度） 动作，可能超出目标位置，此时，以可以停在目标位置的加速度或减速度进行定位。

9-5-4 跃度

在跃度中指定加速度 / 减速度的变化比率。指定跃度后，加减速时的速度波形变为 S 字型，可抑制机械的冲击或振动。

参考

跃度也称为 “ 跃动 ”、 “ 加加速度 ”。

跃度的单位

跃度的单位为 “ 指令单位 /s^3”。
指令单位为位置的显示单位和通过电子齿轮进行单位转换后的值。

指定轴动作的跃度

实际定位动作的跃度在运动控制指令的输入变量 “Jerk （跃度） ” 中指定。
加速和减速中共通的值。

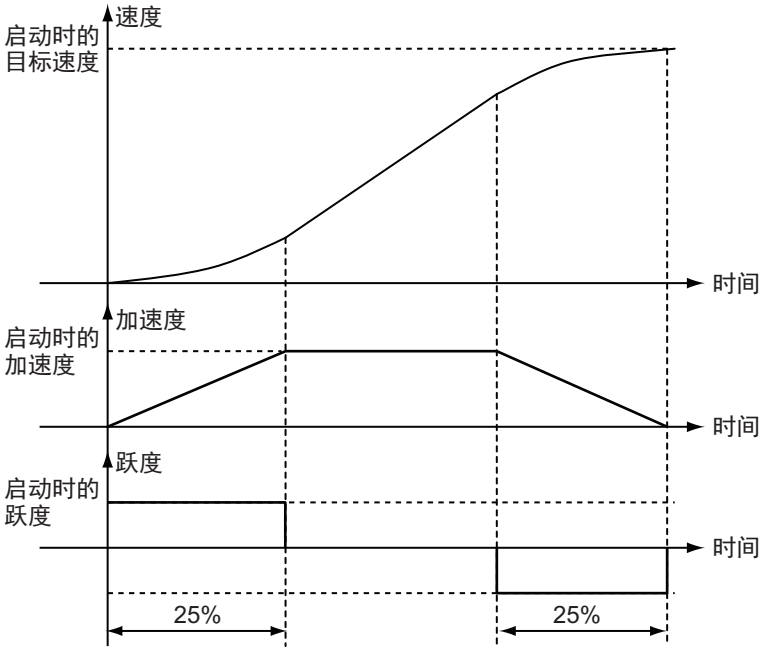
在跃度中设定按以下公式计算的值。
跃度 = 加速度 /（加速时间 × 加速时间内应用跃度的时间比例 /2）

应用跃度的时间平均分为加速开始后的时间和加速结束前的时间。因此，应用跃度的时间比例要除以 2。

● 指定了跃度时的加速控制示例

应用跃度的区间，其加速度会以一定比率变化，因此指令速度会变为平稳的 S 字型。跃度为 “0” 的区间，其加速度固定，因此指令速度为直线型。

例 : 加速度为 25,000mm/s^2、加速时间为 0.1s、应用跃度的时间比例为 50% 时
跃度 = 25000 / (0.1 × 0.5 / 2)　＝　1,000,000 (mm/s^3)



跃度的监视

在程序中，可通过读取轴变量来监视跃度。
此外，以 _MC_AX[*] 为例对变量名称进行说明， _MC1_AX[*]、 _MC2_AX[*] 亦同。

变量名称	数据类型	名称	功能
_MC_AX[0-255].Cmd.Jerk	LREAL	指令当前跃度	指令跃度的当前值。

9-5-5 动作方向指定

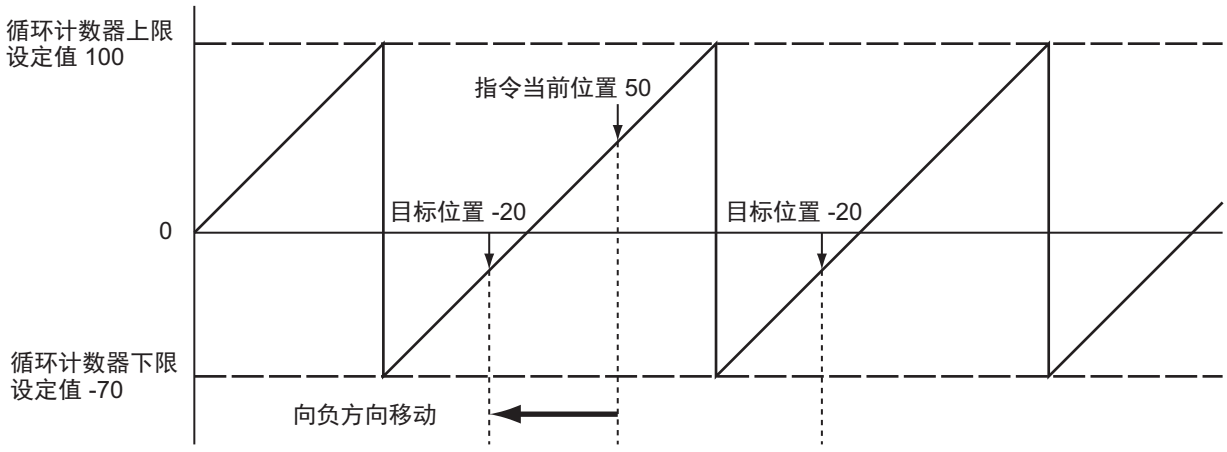
在索引表等中，指定最短距离等旋转方向进行动作时，将计数模式设定为旋转模式。此外，还需在指定绝对位置的运动控制指令的输入变量 “Direction （方向选择） ” 中指定动作方向。
方向选择为 “ 最短距离指定 ”、“ 正方向指定 ”、“ 负方向指定 ”、“ 当前方向指定 ” 时，目标位置可指定为不足循环计数器旋转 1 圈、循环计数器下限值以上、上限值以下的位置。
计数模式为直线模式时，将忽略方向选择，定位到目标位置。

MC 功能模块中可指定的方向选择如下所示。

方向选择	动作
最短距离指定	向指令当前位置和目标位置的距离较短的方向开始动作。
正方向指定	向正方向开始动作。
负方向指定	向负方向开始动作。
当前方向指定	向与上次动作相同的方向开始动作。
无方向指定	向循环计数器上下限值范围内的方向开始动作。 在该方向选择中，可指定超出循环计数器上下限值的目标位置。此时将进行相对值定位，以目标位置和指令当前位置的差为目标距离。因此，可定位到循环计数器的多旋转位置。

最短距离指定的示例

指令当前位置为 “50” 时，向目标位置 “-20” 进行定位的示例。

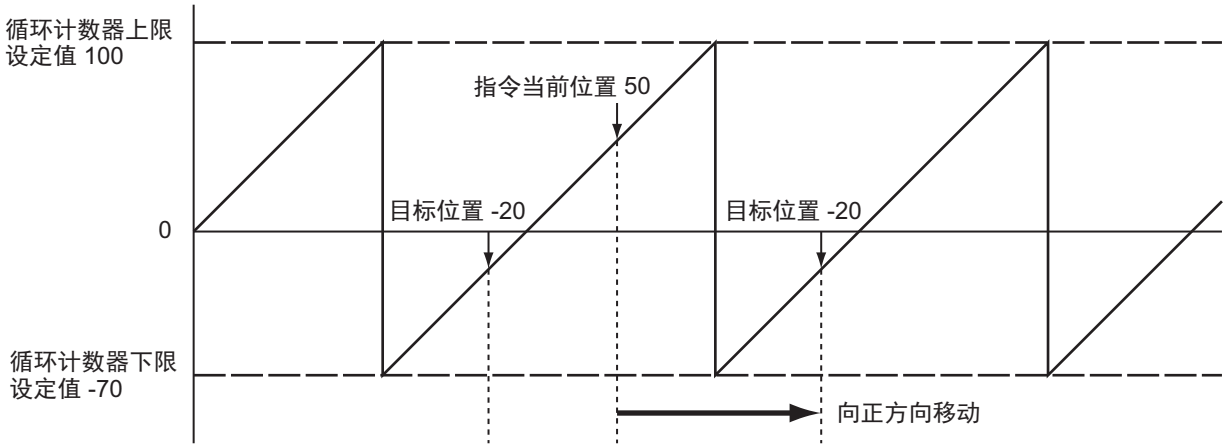


参考

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

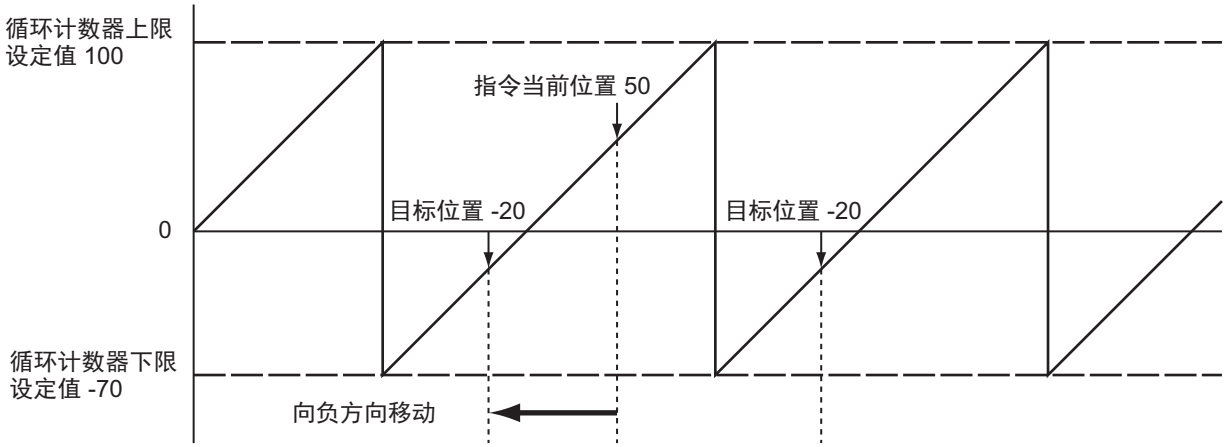
正方向指定的示例

指令当前位置为 “50” 时，向目标位置 “-20” 进行定位的示例。



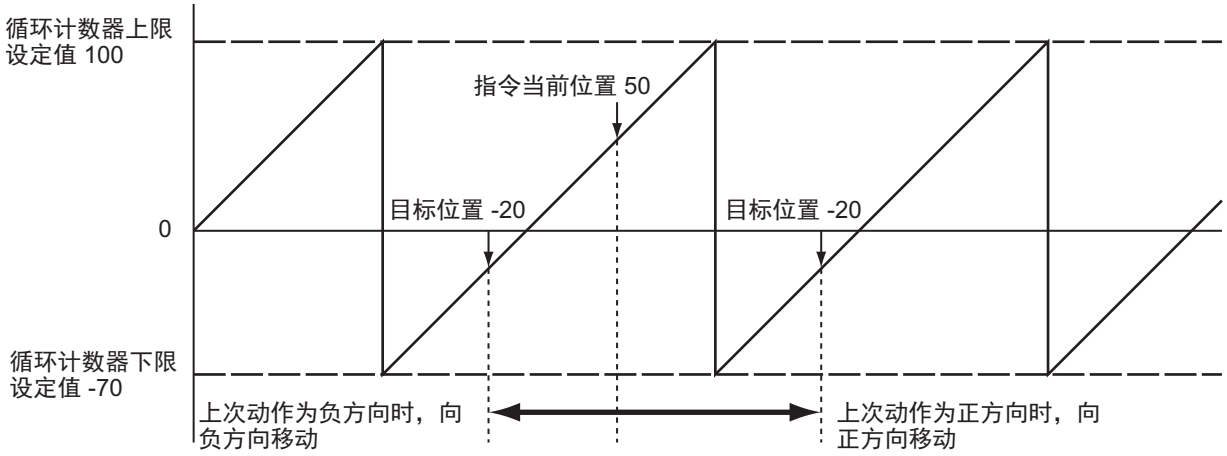
负方向指定的示例

指令当前位置为 “50” 时，向目标位置 “-20” 进行定位的示例。



当前方向指定的示例

指令当前位置为 “50” 时，向目标位置 “-20” 进行定位的示例。



上次的动作方向可根据轴变量中的 “ 指令方向状态 ” 判断。

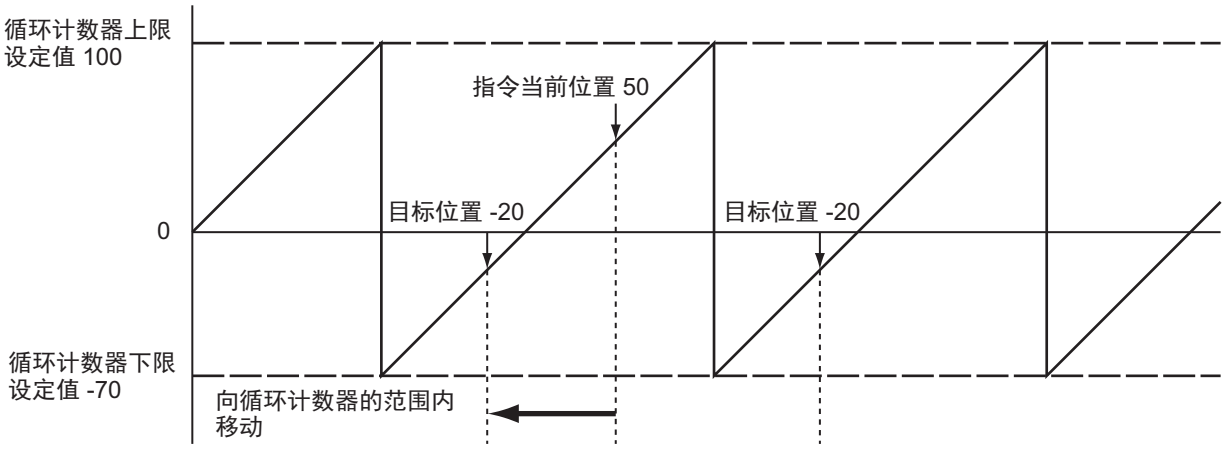
使用注意事项

上次的动作为以下情况时，请注意动作方向。

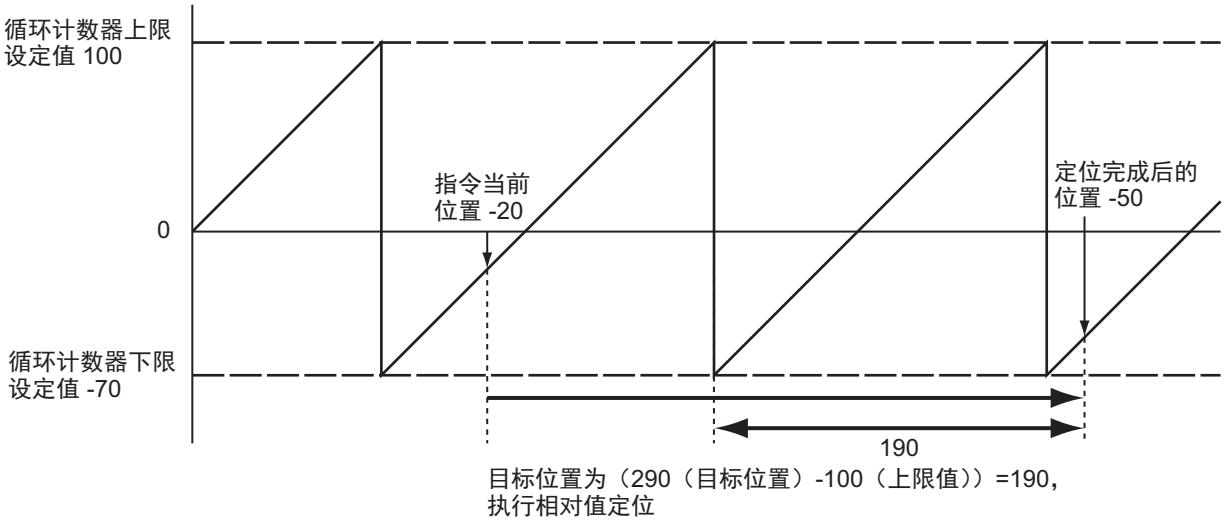
- 在 MC_Home （原点复位）指令或 MC_HomeWithParameter （参数指定原点复位）指令中，超过检测到原点输入的位置进行反转动作时，与原点检测方向的方向相反。
- 在 MC_Home （原点复位）指令或 MC_HomeWithParameter （参数指定原点复位）指令中，设定了原点复位修正值时，则向修正值的方向动作。
- 在 MC_TouchProbe （外部锁定有效）指令中指定了立即停止时，超过锁定位置进行反转动作时。
- 在 MC_MoveFeed （中断固定尺寸定位）中进行反转动作时。
- 执行了 MC_ResetFollowingError （偏差计数器复位）指令时，为了将偏差变为 “0”，方向为规定的指令位置。
- 在外部输入信号的立即停止或极限输入停止中指定了偏差计数器复位时，向发生外部输入信号的位置进行反转动作时。

无方向指定的示例

指令当前位置为 “50” 时，向目标位置 “-20” 进行定位的示例。



同样，循环计数器上限值为 “100”、下限值为 “-70”、指令当前位置为 “-20” 时，向目标位置 “290” 进行定位的示例。



9-5-6 运动控制指令的重启

下面介绍在单轴动作过程中，变更运动控制指令的同一实例的输入变量后，重新启动时的示例。
通过重启可变更的输入变量包括、 Position （目标位置）、 Distance （移动距离）、 Velocity （目标速度）、 Acceleration （加速度）、 Deceleration （减速度）、 Torque （目标扭矩）等。
若变更了无法变更的输入变量并重启，指令将发生异常。
通过运动控制指令的多重启动来重启缓存的指令时，将变更缓存中指令的输入变量。

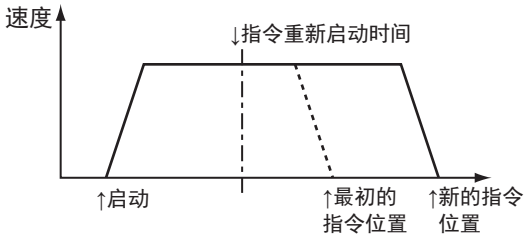
可变更的输入变量详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的各指令。

目标位置的变更

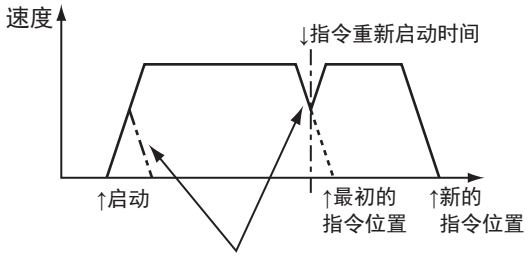
通过重启来变更目标位置时，因变更的目标位置或变更时间不同，动作有所不同。
因目标位置的变更导致动作方向反转时，执行的动作可从轴参数 ［反转时动作］中选择 ［反转时减速停止］或 ［反转时立即停止］。

● 因新的指令值而不反转时

等速动作中的重新启动



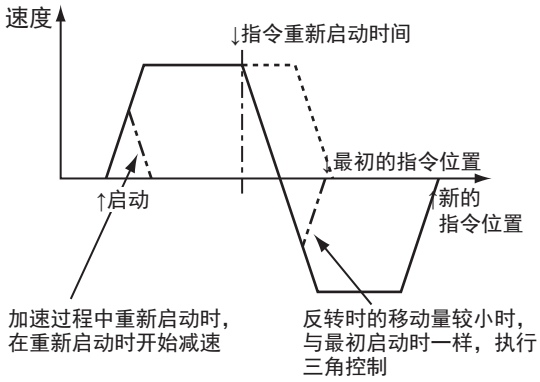
加减速动作中的重新启动



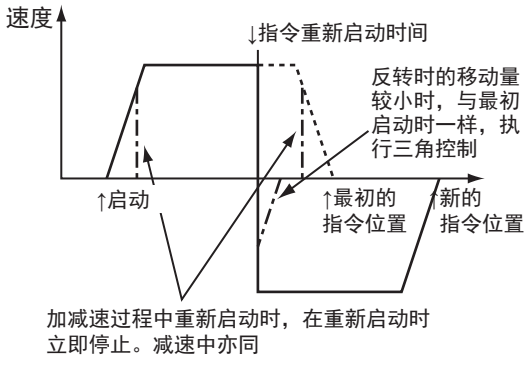
三角控制或减速过程中重新启动时，
将再次加速至目标速度。
此时，也可能不会加速至目标速度。

● 因新的指令值而反转时

反转时减速停止设定



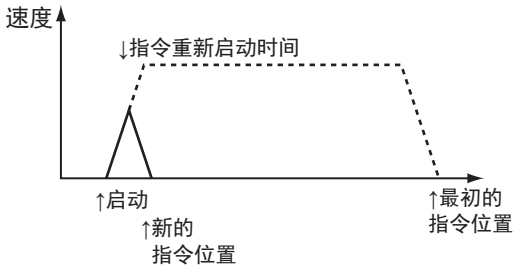
反转时立即停止设定



● 三角控制模式

因目标位置的变更导致移动距离变短时，如下图所示，可能变为三角控制。

不反转时

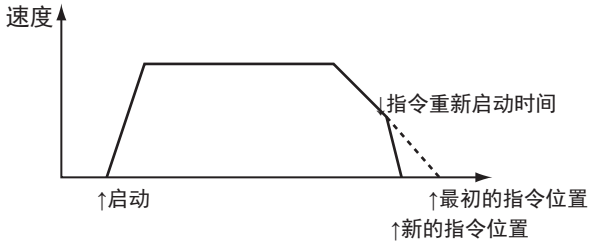


● 减速超限模式

以下情况下，将优先停止到目标位置，因此可能超出指定的减速度。
减速度超出轴参数“最大减速度”时，将变为轴参数设定“加减速超限”中设定的动作。

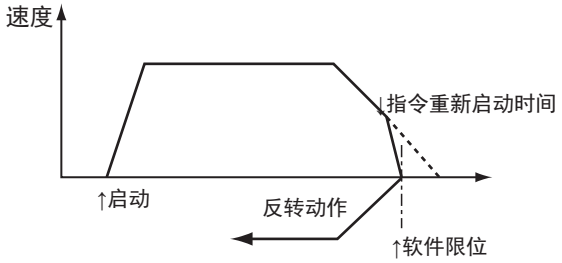
- 动作不反转时，若以指定减速度动作，可能超过目标位置，此时动作如下。

不反转时



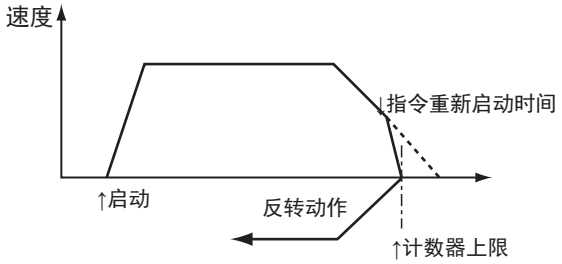
- 动作反转时，若进行减速停止，可能超出软件限位，此时动作如下。

不反转时



- 动作反转时，若进行减速停止，指令当前位置可能出现溢出或下溢，此时动作如下。

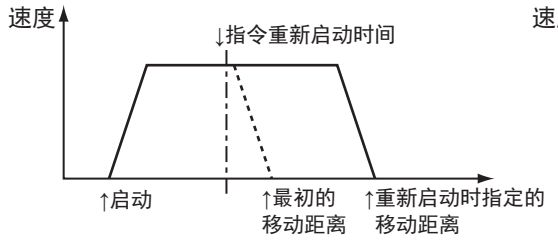
不反转时



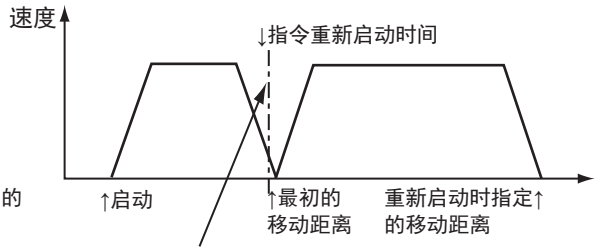
移动距离的变更

在 MC_MoveRelative （相对值定位）指令中，即使变更移动距离并重启后，仍以开始移动时的位置为基准，根据变更的移动量进行定位。
但是，若在定位即将完成前重启，可能作为新指令的启动进行移动。

动作中的重新启动



定位即将完成前的重新启动



若在定位即将完成前重新启动，可能以根据最初的移动距离定位的位置为基准，按重新启动时指定的移动距离移动。

使用注意事项

变更移动距离并重启时，请勿在定位即将完成前执行。

目标速度的变更

关于目标速度，在加速中、等速中、三角控制的加速中，动作会发生变更。
减速过程中也可接受变更，但不会影响实际动作。

加速度的变更

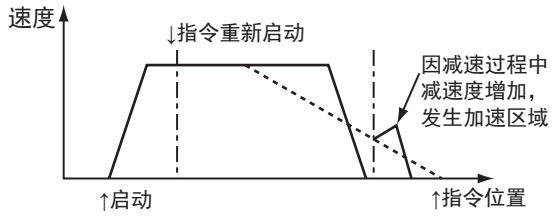
关于加速度，在加速中、三角控制的加速中，动作会发生变更。
若在等速过程中变更，会在因超驰而加速时发生变更。
减速过程中也可接受变更，但不会影响实际动作。

减速度的变更

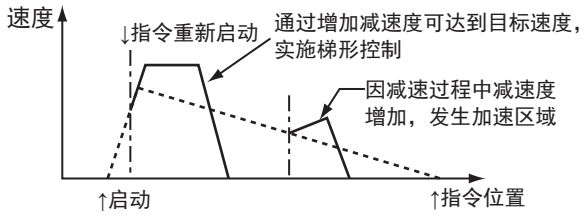
关于减速度，会在加速中、等速中、减速中、三角控制中及减速超限控制中发生变更。
若使用变更后的减速度，可能会超过目标位置，为了优先停止到目标位置，可能超出指定的减速度。

● 减速度增大模式

梯形控制

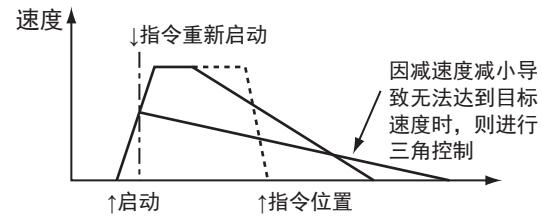


三角控制→梯形控制

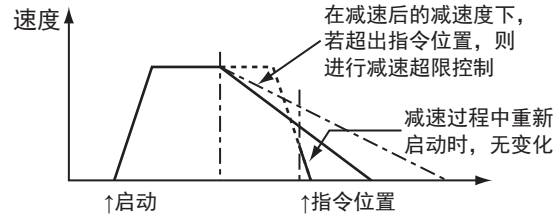


● **减速度减小模式**

梯形控制 · 三角控制



减速超限控制



扭矩指令的变更

重新执行了运动控制指令时，将根据指定的扭矩斜率变更扭矩指令值。

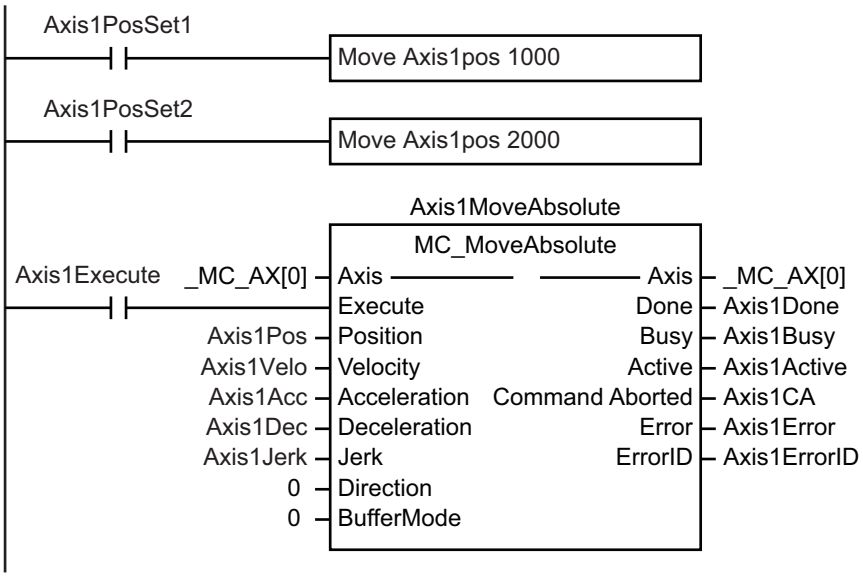
重新执行的程序示例

在绝对值定位中，将目标位置从 “1000” 变更为 “2000” 的示例。

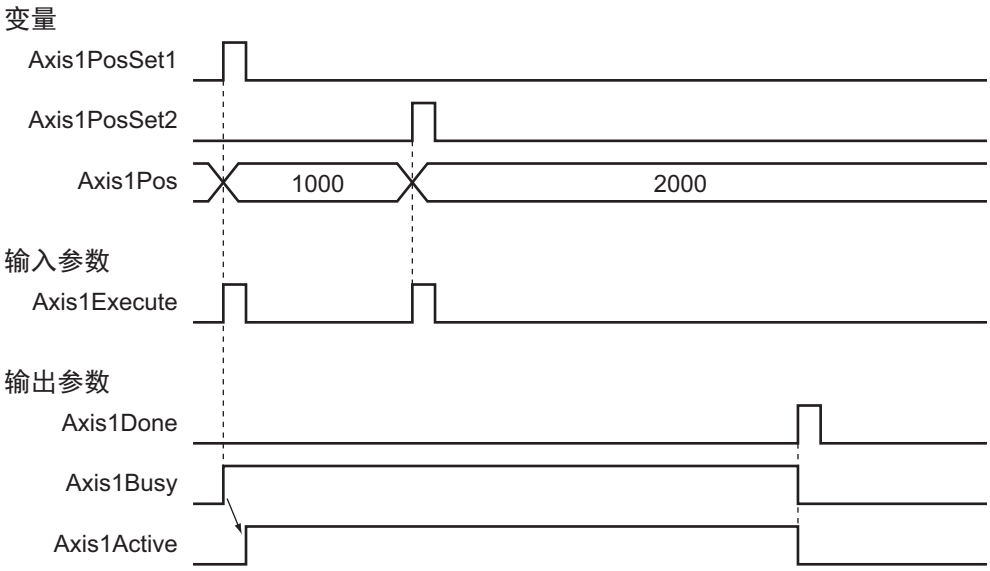
目标位置的输入参数使用变量 “Axis1Pos”。

用 MOV 指令将目标位置设定为 “1000”，并将 Axis1Execute 从 FALSE 变为 TRUE，则开始定位。

在动作过程中将目标位置设定为 “2000”，并再次将 Axis1Execute 从 FALSE 变为 TRUE，则变更为向目标位置 “2000” 执行定位动作。



● **时序图**



使用注意事项

不变更的输入变量保持启动前的值。

9-5-7 运动控制指令的多重启动 （缓冲模式）

在轴的动作过程中，可以启动其他的运动控制指令。
在 PLCopen® 技术规格书中，定义为缓冲模式，在 MC 功能模块中，作为 “ 多重启动 ” 进行说明。
若使用多重启动，无需停止多个运动控制指令的动作，可连续执行。

在 MC 功能模块中，使用以下与多重启动相关的术语。

术语		含义
本书	PLCopen®	
当前指令	以前的 FB	多重启动前正在动作的运动控制指令
缓冲指令	下一个 FB	在轴的动作过程中启动的、处于等待执行状态的运动控制指令
中继速度	共混 （混合）	在共混 （混合）指定中，移动到当前指令中指定的目标位置时使用的指令速度

通过指定运动控制指令的输入变量 “BufferMode （缓冲模式选择） ”，可选择以下多重启动模式。
这些模式的主要区别在于多重启动的指令执行开始时间和中继速度。

多重启动模式		动作说明
中断		中断当前指令，执行多重启动的运动控制指令。
缓冲 （待机）		当前指令的动作正常结束后，执行缓冲指令。
共混 （混合）		到达当前指令的目标位置后，执行缓冲指令。 在本模式中，当前指令和缓冲指令之间不会停止。 当前指令到达目标位置时的中继速度可从以下选项中选择。
	低共混 （低速度）	将当前指令和缓冲指令的目标速度中较低的速度作为中继速度。
	前共混 （前速度）	将当前指令的目标速度作为中继速度。
	后共混 （后速度）	将缓冲指令的目标速度作为中继速度。
	高共混 （高速度）	将当前指令和缓冲指令的目标速度中较高的速度作为中继速度。

多重启动的指令在缓冲 （待机）和共混 （混合）时，会在 MC 功能模块中缓冲，以指定的多重启动模式中的时间和中继速度执行。
该缓冲每个轴有 1 个。
在中断中，会立即执行多重启动的指令，因此指令不会缓冲。

使用注意事项

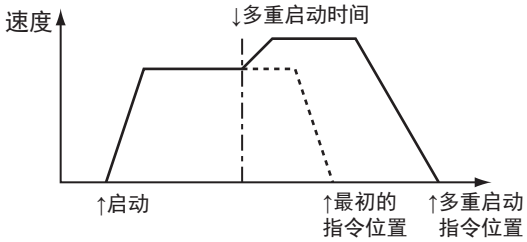
- 1 个轴中可缓冲的多重启动仅限 1 个。若多重启动 2 个以上的指令，指令将发生异常。
- 对单轴控制下动作的轴，不可多重启动多轴协调控制指令 （轴组指令）。此外，对多轴协调控制 （轴组指令）下动作的轴，不可多重启动单轴控制指令 （轴指令）。执行上述多重启动时，指令将发生异常。

中断

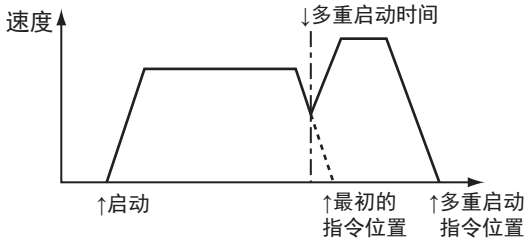
不进行缓冲的默认模式。
立即中断正在执行的指令，执行多重启动的新指令。
中断可在单轴控制和同步控制的运动控制指令中多重启动。

● 不从多重启动指令不反转时

等速动作中的多重启动



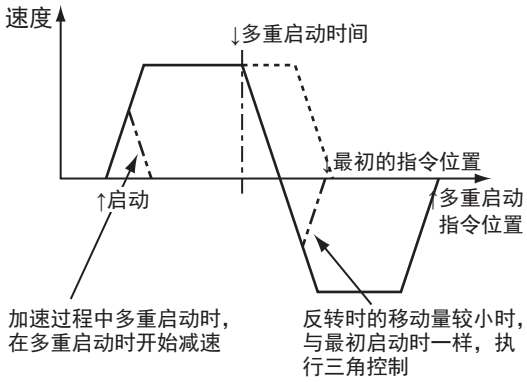
加减速动作中的多重启动



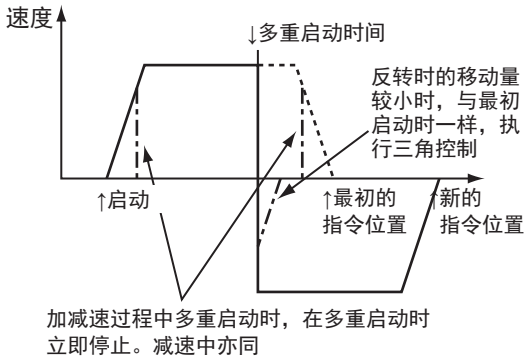
三角控制或减速过程中多重启动时，
将加速至缓冲指令的目标速度。
此时，也可能不会加速至目标速度。

● 从多重启动指令不反转时

反转时减速停止设定

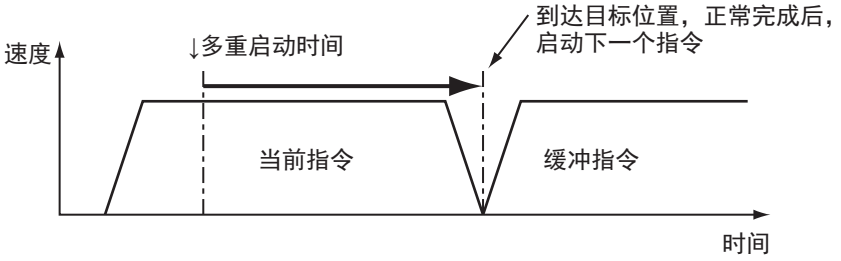


反转时立即停止设定



缓冲 （待机）

多重启动的指令将处于缓冲 （待机）状态，直至正在执行的当前指令完成动作。
当前指令的动作正常完成后，执行缓冲指令。



共混 （混合）

多重启动的指令将处于缓冲 （待机）状态，直至到达当前指令的目标位置。
将在到达当前指令的目标位置后执行缓冲指令，此时不会停止，可以按输入变量 “BufferMode （缓冲模式选择） ” 中指定的速度连续执行两个动作。
相对值移动时，最终的到达位置为两个指令的合计值。
绝对值移动中，为第二个指令的目标位置。
若为轴参数中的 “ 最大加速度 ” 或 “ 最大减速度 ”，当超过目标位置时，可通过轴参数 “ 加减速超限 ” 对动作进行以下设定。

- 剧烈加减速 （将共混切换为缓冲）
- 剧烈加减速
- 异常停止

使用注意事项

共混 （混合）不可在单轴控制和同步控制的组合中执行。

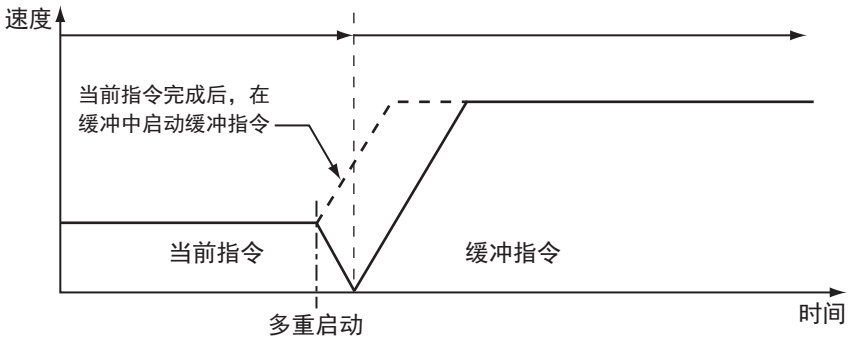
版本相关信息

在单元版本 1.10 以上的 CPU 单元中，即使设定为 ［剧烈加减速 （将共混切换为缓冲）］，也无法将共混切换为缓冲。此时，将以最大加减速度进行加减速，继续进行共混。
此外，即使设定为 ［异常停止］，也不会作为异常而停止。此时，也以最大加减速度进行加减速，继续进行共混。

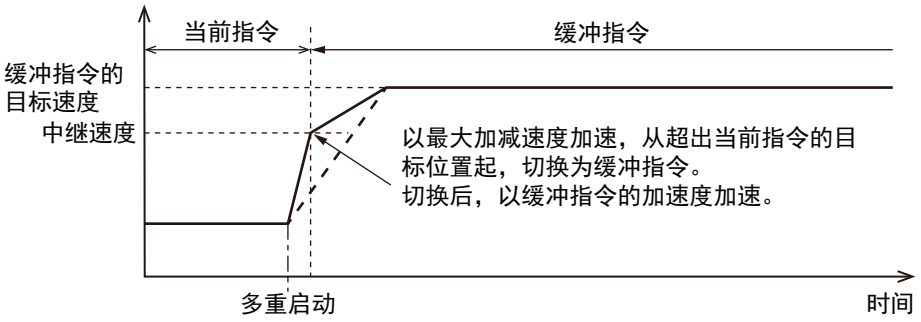
“ 加减速超限 ” 的动作示例如下所示。

● 剧烈加减速 （将共混切换为缓冲）

- 单元版本 1.09 及以下的 CPU 单元时，动作如下。

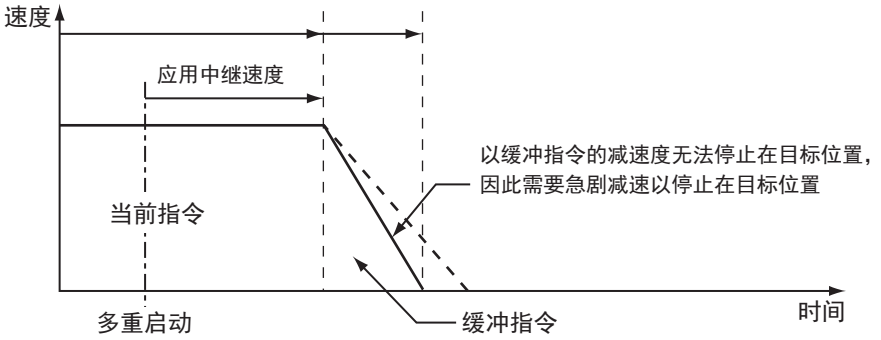


- 单元版本 1.10 及以上的 CPU 单元时，动作如下。
  此外，即使设定为［异常停止］，动作也相同。
  在本示例中， BufferMode 设定为后共混。

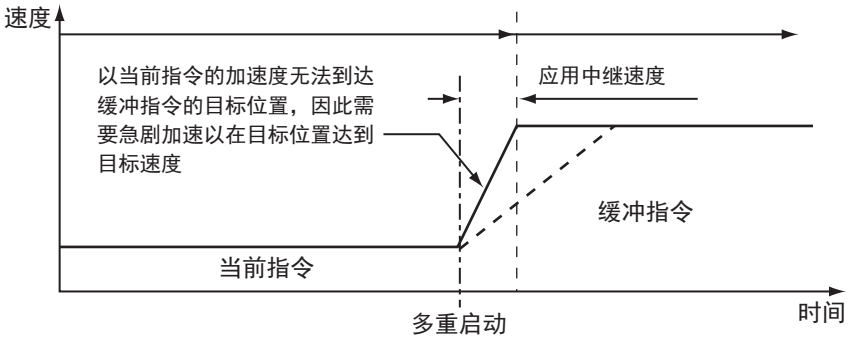


● 剧烈加减速

- 将 “BufferMode （缓冲模式选择）” 设定为前共混时，动作如下。



- 将 “BufferMode （缓冲模式选择）” 设定为后共混时，动作如下。



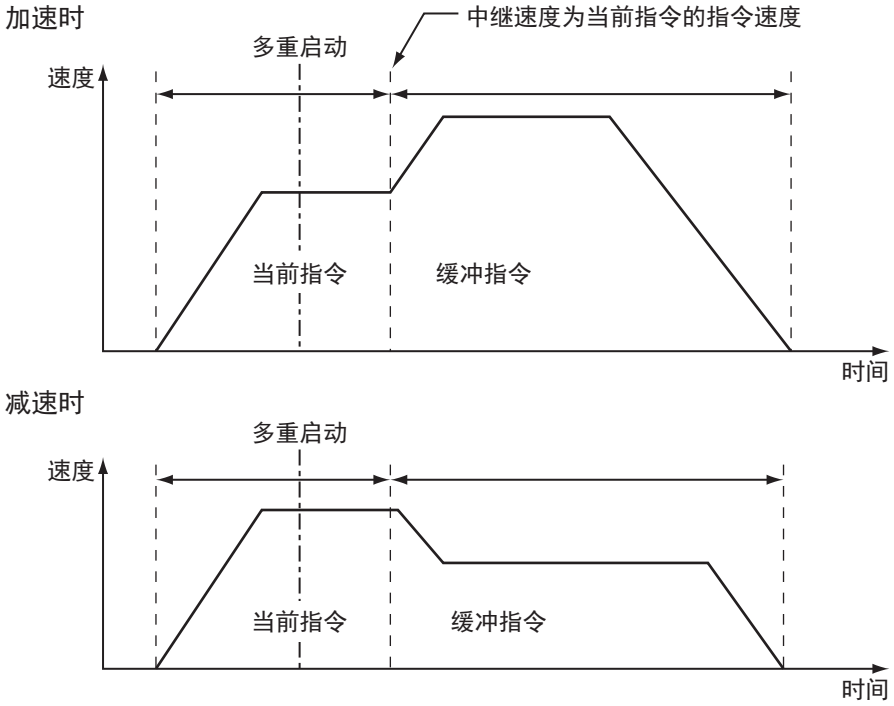
低共混 （低速度）

在当前指令的目标位置，以当前指令和缓冲指令中速度较低的目标速度动作。

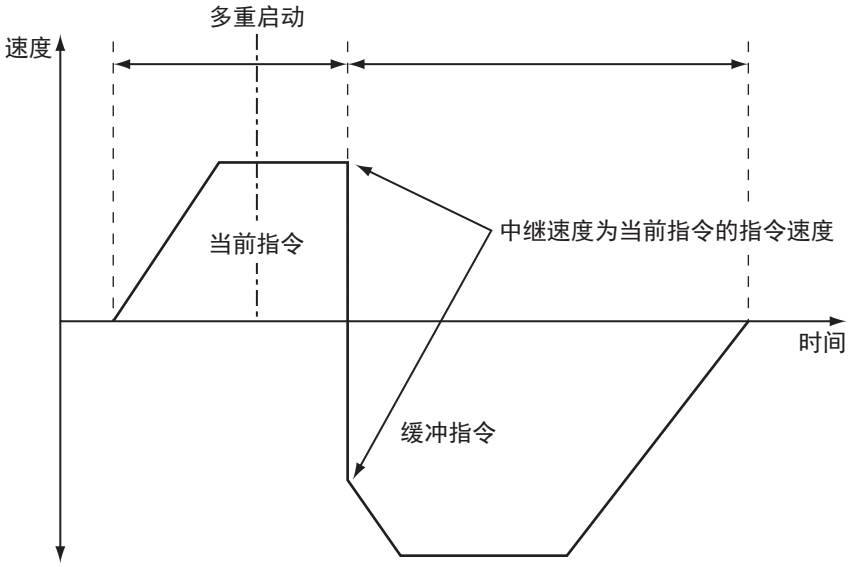
前共混 （前速度）

到达当前指令的目标位置前，以当前指令的目标速度动作。
到达目标位置后，加减速至缓冲指令的目标速度并动作。

● 动作方向不反转时

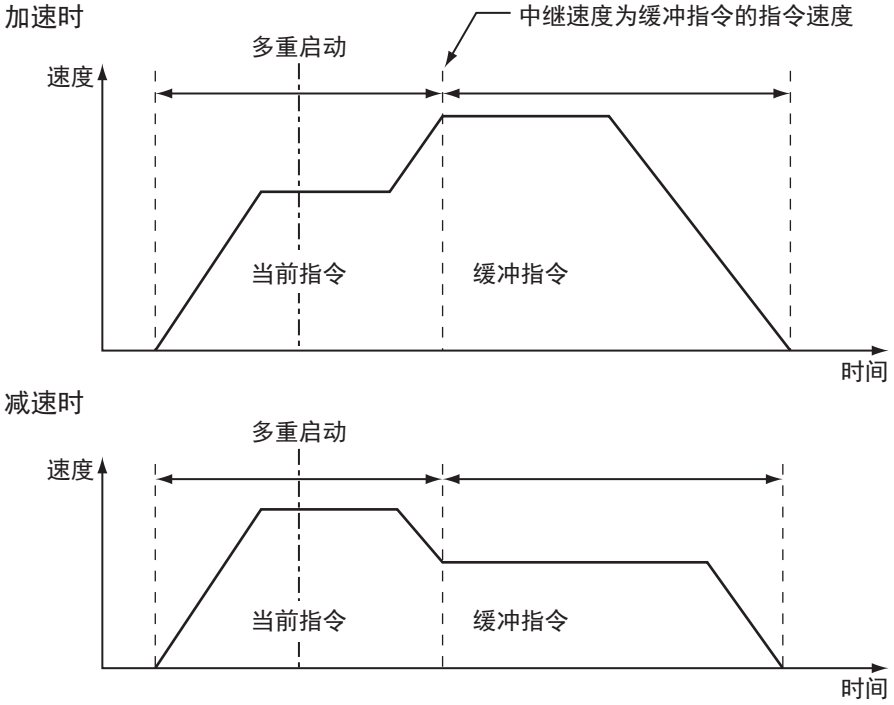


● 动作方向反转时



后共混 （后速度）

在当前指令的目标位置，以缓冲指令的目标速度动作。



高共混 （高速度）

在当前指令的目标位置，以当前指令和缓冲指令中速度较高的目标速度动作。

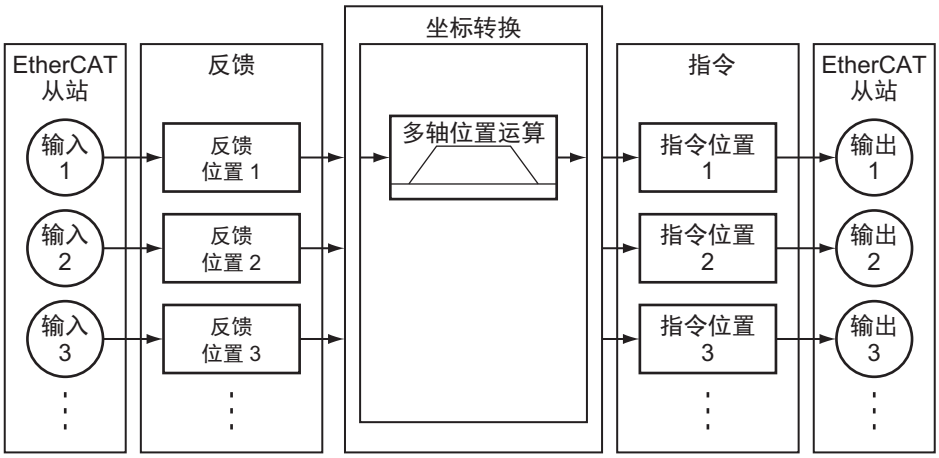
9-6 多轴协调控制

下面介绍多轴的协调控制动作。
在 MC 功能模块中，可事先在 Sysmac Studio 中设定轴组，进行多个轴的插补控制。

9-6-1 动作概要

多轴协调控制是指将多个相关的轴组合成一个组进行协调控制，以控制工具前端等控制对象的轨迹的功能。
在 MC 功能模块中，执行协调动作的轴组合可作为轴组通过 Sysmac Studio 设定。
在用户程序中，当各轴变为伺服 ON 后，将实际要进行多轴协调控制的轴组设为有效。
多轴协调控制的目的是实现轴组中轴的协调动作，因此对于有效轴组中的轴，不可启动单轴动作的运动控制指令。此外，轴组中的轴发生异常时，轴组中的其他轴将根据轴组参数 ［轴组停止方法］的设定停止。

在 MC 功能模块中，可进行 2 轴～ 4 轴的直线插补和 2 轴的圆弧插补。



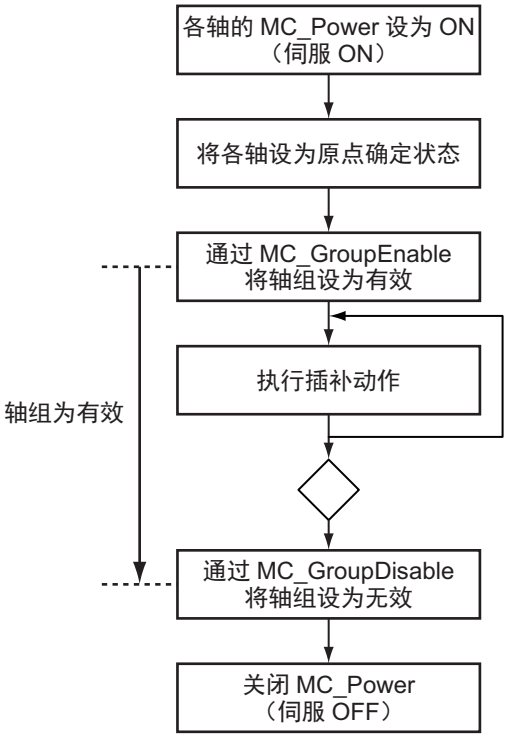
参考

若装置需要在动作过程中变更轴的组合并进行插补控制，需要事先在 Sysmac Studio 中创建要变更组合的轴所在的多个轴组。然后，在动作过程中通过用户程序指定要设为有效的轴组，即可实现。
在 CPU 单元版本 1.01 以上和 Sysmac Studio Ver.1.02 以上的组合中，使用 MC_ChangeAxesInGroup （轴组构成轴写入），可替换设为无效的轴组的构成轴。

轴组的详情请参考 「*3-3 轴组* (P.3-19)」。

轴组有效 / 无效

要将轴组设为有效，需要在 MC_GroupEnable （轴组有效）指令中指定要设为有效的轴组。若在无效状态下启动轴组指令，指令将发生异常，无法启动。
要将有效状态的轴组设为无效，需要在 MC_GroupDisable （轴组无效）指令中指定要设为无效的轴组。
若在轴组动作过程中启动轴组无效指令，轴组中的轴将以轴参数设定的最大减速度减速停止。



关于轴组有效 / 无效功能的详情，请参考 🕮《NJ/NX 系列　指令基准手册　运动篇 (SBCE-CN5-364)》的 MC_GroupEnable （轴组有效）指令及 MC_GroupDisable （轴组无效）指令。

轴组构成轴写入

该功能可临时替换通过 MC_ChangeAxesInGroup （轴组构成轴写入）指令设为无效的轴组中的构成轴。轴组为有效时，请用 MC_GroupDisable （轴组无效）指令设为无效后再替换构成轴。
可在 CPU 单元版本 1.01 以上和 Sysmac Studio Ver.1.02 以上的组合中使用。

使用注意事项

通过 MC_ChangeAxesInGroup （轴组构成轴写入）进行的变更不会保存到 CPU 单元内置的非易失性存储器中。重新接通电源或从 Sysmac Studio 下载后，将恢复为非易失性存储器的参数设定内容。

关于轴组构成轴写入的详情，请参考 🕮《NJ/NX 系列　指令基准手册　运动篇 (SBCE-CN5-364)》的 MC_ChangeAxesInGroup （轴组构成轴写入）指令。

轴组位置获取

利用 MC_GroupReadPosition （轴组位置获取）指令，获取轴组的指令当前位置和反馈当前位置的功能。
可在 CPU 单元版本 1.01 以上和 Sysmac Studio Ver.1.02 以上的组合中使用。

关于轴组位置获取功能的详情，请参考 🕮《NJ/NX 系列　指令基准手册　运动篇 (SBCE-CN5-364)》的 MC_GroupReadPosition （轴组位置获取）指令。

轴组错误复位

轴组发生异常时，应在排除异常原因后，用 MC_GroupReset （轴组错误复位）指令解除异常。

关于轴组错误复位的详情，请参考 🕮《NJ/NX 系列　指令基准手册　运动篇 (SBCE-CN5-364)》的 MC_GroupReset （轴组错误复位）指令。

使用 NX 系列脉冲输出单元时的区别请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

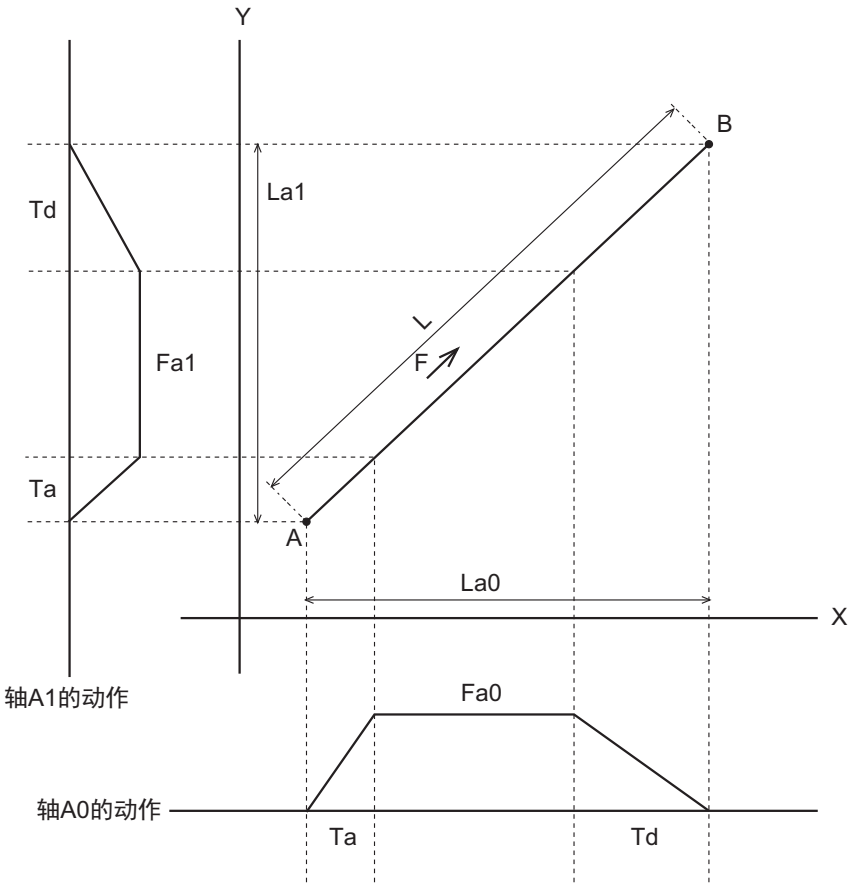
9-6-2 直线插补

直线插补是指使用轴组中的 2 个轴～ 4 个轴 （轴 A0 ～轴 A3 的逻辑轴），直线式定位起点和终点。可进行绝对值定位和相对值定位，还可指定插补速度、插补加速度、插补减速度及跃度。

MC 功能模块中，有以下 3 种直线插补指令。

- MC_MoveLinear （直线插补）
  指定输入变量 “MoveMode （移动方法选择） ” 后，可选择绝对值的直线插补和相对值的直线插补。MC 功能模块独有的指令。
- MC_MoveLinearAbsolute （绝对值直线插补）
  用绝对值进行直线插补。 PLCopen® 技术规格书中定义的指令。
- MC_MoveLinearRelative （相对值直线插补）
  用相对值进行直线插补。 PLCopen® 技术规格书中定义的指令。

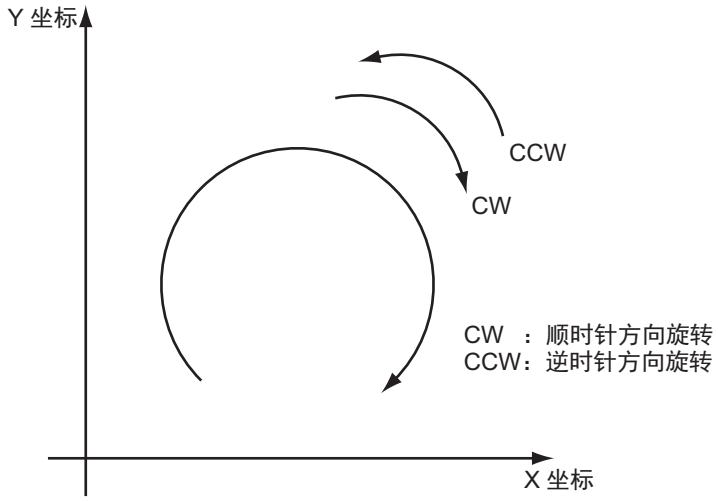
使用 2 个轴的 A 点到 B 点的直线插补如下图所示。



关于直线插补的详情，请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_MoveLinear （直线插补）指令、 MC_MoveLinearAbsolute （绝对值直线插补）指令及 MC_MoveLinearRelative （相对值直线插补）指令。

9-6-3 圆弧插补

圆弧插补是指使用轴组中的 2 个轴 （轴 A0 ～轴 A3 的逻辑轴），在二维平面上绘制圆弧进行定位。
可进行绝对值定位和相对值定位，还可指定圆弧插补模式、路径方向、插补速度、插补加速度、插补减速度及 2 个轴合成值的跃度。



MC 功能模块中，通过输入变量 “CircMode （圆弧插补模式） ”，可从以下 3 种方式中选择圆弧插补的方式。

- 通过点指定
- 中心点指定
- 半径指定

使用注意事项

圆弧插补中使用的轴计数模式请选择 “ 直线模式 ”。
若以 “ 旋转模式 ” 启动，质量将发生异常。

9-6-4 轴组周期性同步位置控制

该功能可周期性输出轴组的各轴中指定的目标位置。
要以任意的轨迹动作时，以绝对位置指定在用户程序中计算得到的目标位置。
可在 CPU 单元版本 1.01 以上和 Sysmac Studio Ver.1.02 以上的组合中使用。

关于轴组周期性同步位置控制功能的详情，请参考 🕮《NJ/NX 系列　指令基准手册　运动篇 (SBCE-CN5-364)》的 MC_GroupSyncMoveAbsolute （轴组周期性同步位置控制）指令。

9-6-5 多轴协调控制的停止

停止轴组动作的功能包括利用用户程序的运动控制指令停止和发生异常而停止。

利用运动控制指令停止

要停止轴组动作，可使用 MC_GroupStop （轴组强制停止）指令或 MC_GroupImmediateStop （轴组立即停止）指令。

● MC_GroupStop （轴组强制停止）指令

对轴组的直线插补和圆弧插补指定减速度和跃度，可在轨迹上减速停止。
若将减速度指定为 “0”，将向伺服驱动器发送立即停止指令。
利用本指令减速停止中或输入变量 “Execute （启动） ” 为 TRUE 期间，不接受其他动作指令。

● MC_GroupImmediateStop （轴组立即停止）指令

可立即停止轴组中的所有轴。立即停止的方法遵照各轴的轴参数 “ 立即停止输入停止方法 ”。
对于正在利用 MC_GroupStop （轴组强制停止）指令进行减速停止的轴组也可执行。

功能的详情请参考 🕮《NJ/NX 系列　指令基准手册　运动篇 (SBCE-CN5-364)》的 MC_GroupStop （轴组强制停止）指令、 MC_GroupImmediateStop （轴组立即停止）指令。

因发生异常而停止

● 轴组动作中因发生异常而停止

轴组动作过程中，轴组中的任意构成轴发生会引起减速停止的异常时，在插补的轨迹上插补减速停止。此时的插补减速度遵照正在控制的指令中指定的减速度。
此外，任意构成轴发生会引起立即停止的异常时，轴组中的其他构成轴将根据轴组参数设定 ［轴组停止方法］的设定停止。

轴组的停止方法可从以下项目中选择。

- 立即停止
- 各轴减速停止 （各轴的最大减速度）
- 立即停止，同时执行伺服 OFF

● 因发生运动控制周期超限而停止

运动控制的处理未在 2 个周期内完成，发生运动控制周期超限时，所有轴将立即停止。

使用注意事项

使用 NX 系列 CPU 单元以多运动方式动作时，若任意一个任务发生运动控制周期超限，两个任务中的所有轴都将立即停止。
关于多运动请参考 *「A-5-2 运动控制功能* (P.A-23)」。

● 因开始 MC 调试功能而停止

开始了 Sysmac Studio 的 MC 调试功能时，各轴将以 “ 最大减速度 ” 减速停止。

● 因 CPU 单元的模式切换而停止

切换了 CPU 单元的模式时，各轴将以 “ 最大减速度 ” 减速停止。

参考

- 若在轴组动作过程中启动 MC_GroupDisable （轴组无效）指令，各轴将以最大减速度减速停止。
- 若在轴组动作过程中启动 MC_Stop （强制停止）指令，轴和轴组将发生异常，轴组动作变为插补减速停止。此时的插补减速度遵照正在控制的指令中指定的减速度。
- 若在轴组动作过程中将 MC_Power （可运行）指令的输入变量 “Enable （有效） ” 设为 FALSE， MC 功能模块将立即停止该轴的指令值，并执行伺服 OFF。伺服 OFF 时伺服驱动器的动作遵照伺服驱动器的设定。
  轴组中的其他轴将根据轴组参数设定 ［轴组停止方法］的设定停止。轴组将发生异常。
- 从运动模式切换为程序模式时，动作中的运动控制指令将中断。此时，指令的输出变量 “CommandAborted （执行中断） ” 保持 FALSE。切换为程序模式后，仍维持伺服 ON/OFF 状态。
- 从运行模式切换为程序模式并正在减速停止时，若恢复为运行模式，运动控制指令的输出变量 “CommandAborted （执行中断） ” 将变为 TRUE。
- 正在利用 MC_SaveCamTable （凸轮表保存）指令保存时，继续执行保存处理。
- 正在利用 MC_GenerateCamTable （凸轮表生成）指令执行凸轮表生成时，继续执行生成处理。

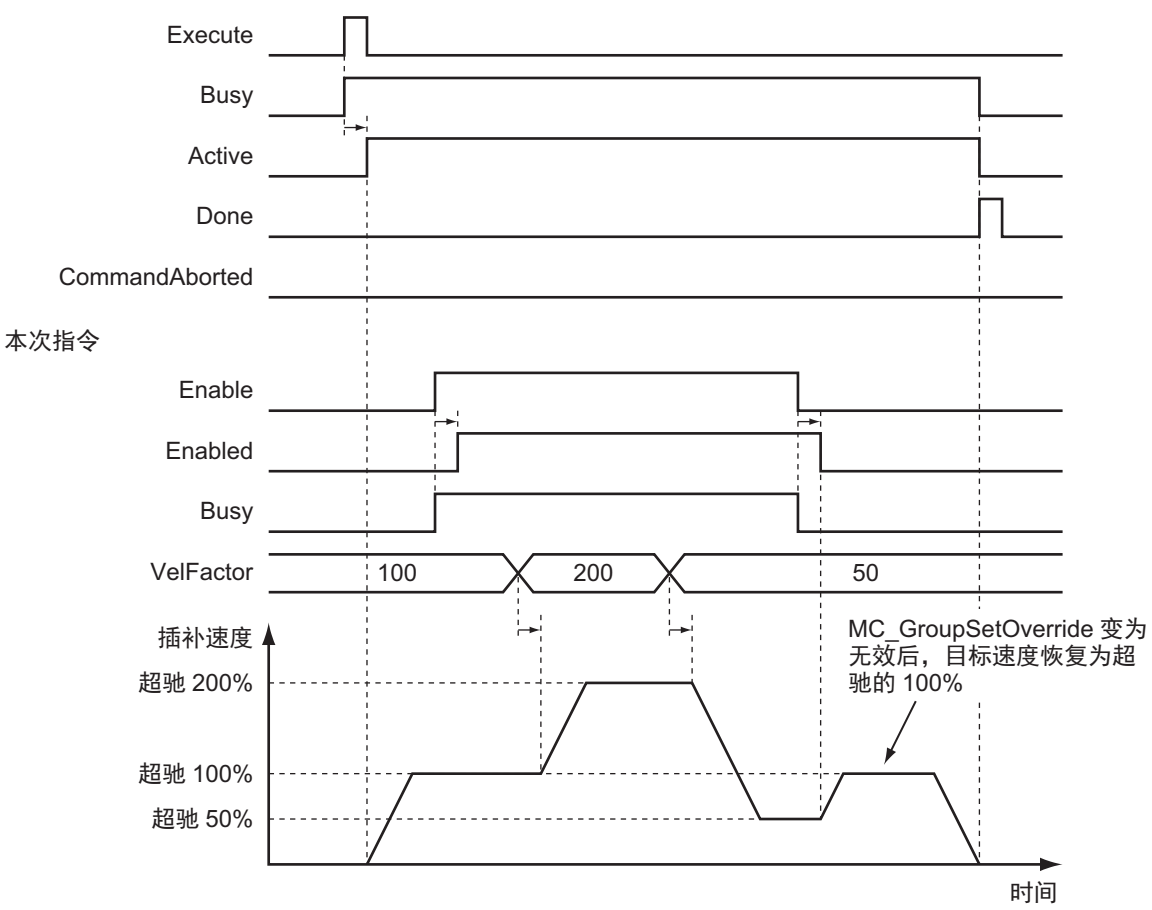
9-6-6 多轴协调控制的超驰

在轴组的多轴协调控制中，可通过 MC_GroupSetOverride （轴组超驰值设定）指令对启动中的插补动作设定超驰值。
超驰值为相对于插补目标速度的比率，可设定为 0 ～ 500%。
若设定为插补目标速度的 0%，将在速度 “0” 的状态下停止，并继续保持动作状态。

超驰为有效期间，将导入设定的超驰值。若将超驰设为无效，超驰值将恢复为 100%。
超驰值中变更后的值超出最高插补速度时，变为轴组参数的 “ 最高插补速度 ”。

● MC_MoveLinear （直线插补）指令的超驰

在 MC_MoveLinear （直线插补）指令中使用超驰指令时，时序图的示例如下所示。



功能的详情请参考 🕮《NJ/NX 系列　指令基准手册　运动篇 (SBCE-CN5-364)》的 MC_GroupSetOverride （轴组超驰值设定）指令。

9-7 多轴协调控制的共通功能

下面介绍多轴协调控制的共通功能。

9-7-1 多轴协调控制的速度

在多轴协调控制中，用轨迹上的插补速度指定速度。
单位与单轴时相同，使用 “ 指令单位 /s”。

速度的类型

MC 功能模块中使用的轴组插补速度为以下 1 种。

速度类型	定义
指令插补速度	MC 功能模块为了控制轴组而输出的指令插补速度的当前值。

速度相关的轴参数

参数名称	功能	设定范围	初始值
最高插补速度	设定轨迹的最高插补速度。 若设定为 “0”，将没有插补速度的限制。 轴组动作指令中指定了超出最高插补速度的插补目标速度时，以最高插补速度动作。	双精度实数型的正数、 0	800,000,000
插补速度警告值	设定相对于最高插补速度的比例，作为输出插补速度警告的标准。 若设定为 “0”，将不输出插补速度警告。 （单位 : %）	0 ～ 100	0

指定轴动作的目标速度

实际定位动作的插补速度在运动控制指令的输入变量 “Velocity （目标速度） ” 中指定。

速度监视

在程序中，可通过读取轴组变量来监视插补速度。
此外，以 _MC_GRP[*] 为例对变量名称进行说明， _MC1_GRP[*]、 _MC2_GRP[*] 亦同。

变量名称	数据类型	名称	功能
_MC_GRP[0-63].Cmd.Vel	LREAL	指令插补速度	指令插补速度的当前值。 正方向移动时加 + 号，负方向移动时加 - 号。

9-7-2 多轴协调控制的加速和减速

在多轴协调控制中，用轨迹上的插补加速度和插补减速度进行控制。
单位与单轴时相同，使用 “ 指令单位 /s^2”。

插补加速度和插补减速度相关的轴参数

参数名称	功能	设定范围	初始值
最大插补加速度	设定轨迹的最大插补加速度。 若设定为 “0”，将没有插补加速度的限制。 （单位 : 指令单位 /s^2）	双精度实数型的正数、 0	0
最大插补减速度	设定轨迹的最大插补减速度。 若设定为 “0”，将没有插补减速度的限制。 （单位 : 指令单位 /s^2）	双精度实数型的正数、 0	0
插补加减速超限	在轴组的加减速控制中，为了优先停止在目标位置，会发生减速超限，导致超出最大插补加减速度，指定这种情况下的动作。 0: 剧烈加减速 （将共混切换为缓冲）*1 1: 剧烈加减速 2: 异常停止 *2	0 ～ 2	0
插补加速度警告值	设定相对于最高插补加速度的比例，作为输出插补加速度警告的标准。 若设定为 “0”，将不输出插补加速度警告。 （单位 : %）	0 ～ 100	0
插补减速度警告值	设定相对于最高插补减速度的比例，作为输出插补减速度警告的标准。 若设定为 “0”，将不输出插补减速度警告。 （单位 : %）	0 ～ 100	0

*1. 单元版本 1.10 以上的 CPU 单元中，无法将共混切换为缓冲。详情请参考 「*9-5-7 运动控制指令的多重启动 （缓冲模式）*(P.9-45)」。
*2. 单元版本 1.10 以上的 CPU 单元中，发生共混动作时，将视为发生异常而停止。详情请参考 「*9-5-7 运动控制指令的多重启动 （缓冲模式）*(P.9-45)」。

轴组动作的插补加速度和插补减速度的指定

实际定位动作的插补加速度及插补减速度在运动控制指令的输入变量 “Acceleration （加速度） ” 及 “Deceleration （减速度） ” 中指定。

插补加速度和插补减速度的监视

在程序中，可通过读取轴变量来监视插补加速度和插补减速度。
此外，以 _MC_GRP[*] 为例对变量名称进行说明， _MC1_GRP[*]、 _MC2_GRP[*] 亦同。

变量名称	数据类型	名称	功能
_MC_GRP[0-63].Cmd.AccDec	LREAL	指令插补加减速度	指令插补加减速度的当前值。 加速时加 + 号，减速时加 - 号。

9-7-3 多轴协调控制的跃度

多轴协调控制中的 Jerk （跃度）可将插补轨迹上的插补加减速变为平稳的 S 字型，减少机械的冲击或振动。

单位与单轴时相同，使用 “ 指令单位 /s^3”。

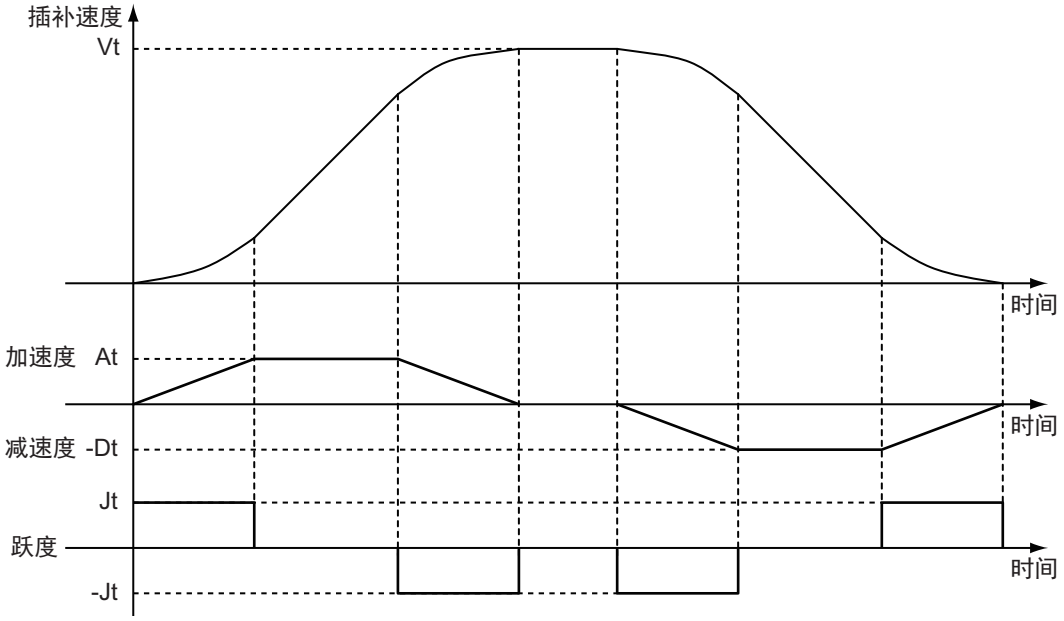
指定轴组动作的跃度

实际插补动作的跃度在运动控制指令的输入变量 “Jerk （跃度） ” 中指定。

Jerk （跃度）为 “0” 以外时的示例

应用跃度的区间，其加减速度会以一定比率变化，因此指令插补速度会变为平稳的 S 字型。

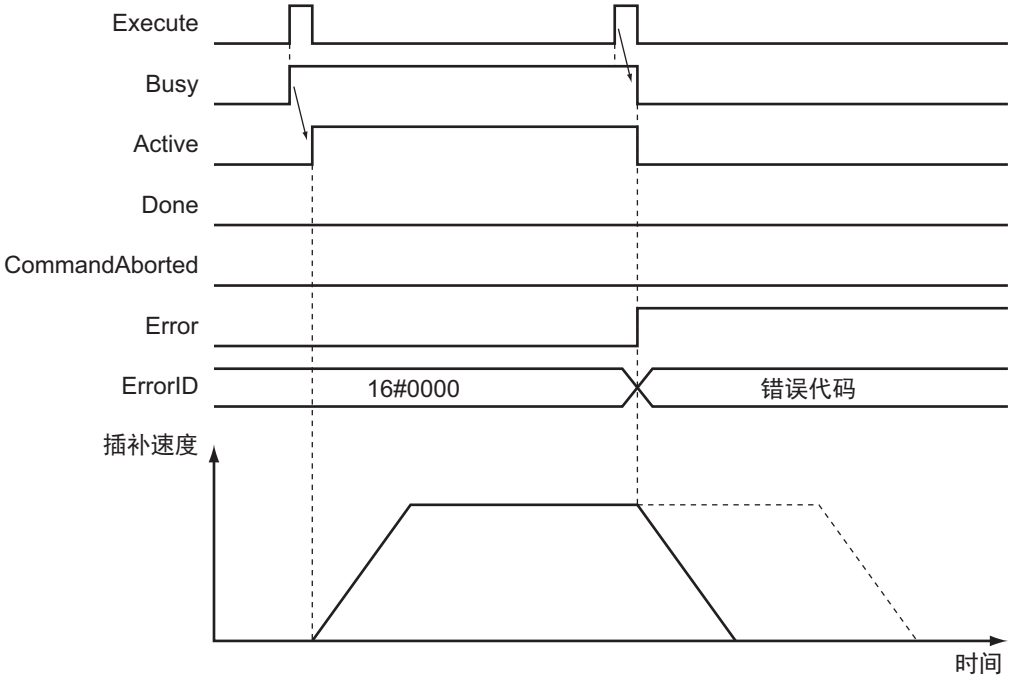
跃度为 “0” 的区间，其插补加速度固定，因此指令插补速度为直线型。



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、 Jt: 跃度的指定值

9-7-4 多轴协调控制时运动控制指令的重启

重启了直线插补或圆弧插补指令时，指令将发生异常。



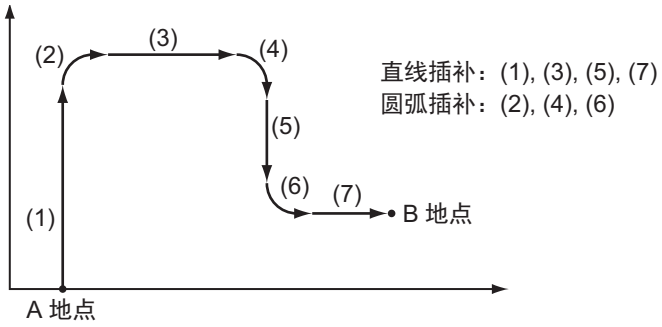
在 MC_GroupStop （轴组强制停止）指令的重启中，可变更减速度，但不可变更跃度。

重启了 MC_GroupReset （轴组错误复位）指令时，将忽略重启指令，继续执行异常解除处理。

各运动控制指令的重启详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的各指令。

9-7-5 多轴协调控制时运动控制指令的多重启动 （缓冲模式）

与轴动作相同，轴组中的多轴协调控制也可多重启动。
若在多轴协调控制中使用多重启动，可执行多条直线或圆弧的连续轨迹控制。



通过指定运动控制指令的输入变量 “BufferMode （缓冲模式选择） ”，可选择与单轴时相同的多重启动模式。
轴组的指令缓冲有 8 个，包括轴组动作时的 1 个和多重启动用的 7 个。
轴动作的指令和轴组动作的指令不可相互多重启动。

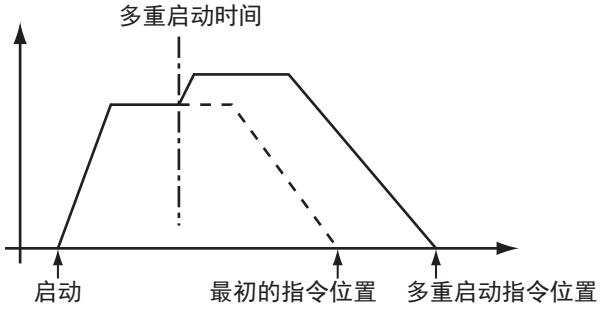
使用注意事项

- 1 个轴中可缓冲的多重启动最多为 7 个。若多重启动 8 个以上的指令，指令将发生异常。
- 对单轴控制下动作的轴，不可多重启动多轴协调控制指令 （轴组指令）。此外，对多轴协调控制 （轴组指令）下动作的轴，不可多重启动单轴控制指令 （轴指令）。执行上述多重启动时，指令将发生异常。

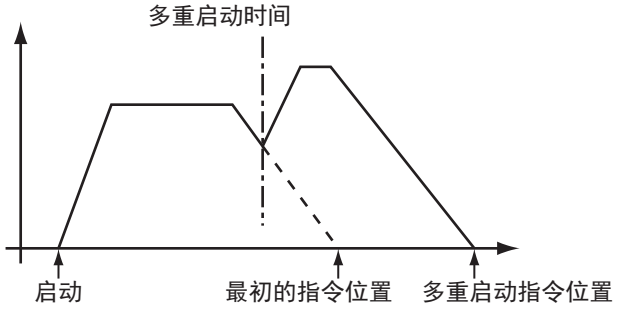
中断

不进行缓冲的默认模式。
立即中断正在执行的指令，执行多重启动的新指令。
多重启动没有输入变量 “BufferMode （缓冲模式选择） ” 的运动控制指令时，作为中断进行动作。
以多重启动时的插补速度为起点，开始执行多重启动指令的动作。
中断不可在包含同步控制的单轴控制和轴组控制的组合中执行。
若对单轴动作中的轴启动轴组动作，多重启动时指令将发生异常。
若在轴组动作中启动单轴动作，轴和轴组都将发生异常并停止。

等速动作中的多重启动

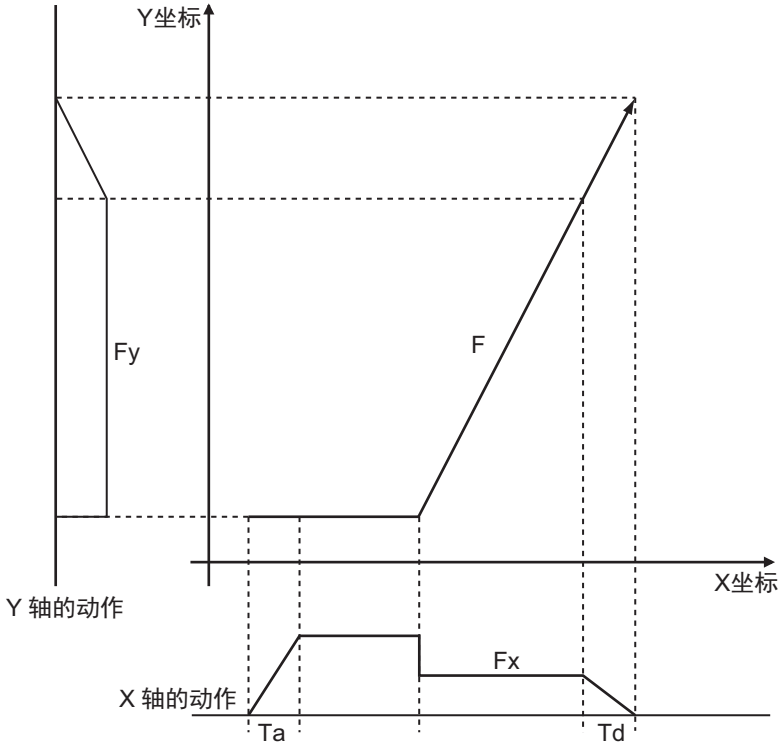


加减速动作中的多重启动



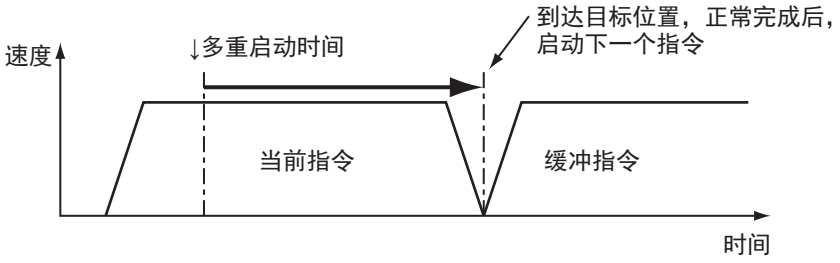
在轴组中多重启动时，插补速度为连续。与移动量为 “0” 的轴连续时，各轴的速度变量将变得不连续。

● 2 轴直角坐标中的插补速度和各轴的速度示例



缓冲 （待机）

多重启动的指令将处于缓冲 （待机）状态，直至正在执行的动作完成。
当前指令的动作正常完成后，执行缓冲指令。



共混 （混合）

轴组中的共混 （混合）与单轴动作的共混 （混合）相同。
多重启动的指令将处于缓冲 （待机）状态，直至到达当前指令的目标位置。
到达当前指令的目标位置后，执行缓冲指令。此时不会在到达位置停止，可以按输入变量 “BufferMode（缓冲模式选择） ” 中指定的插补速度连续执行两个动作。
若为缓冲指令的加速度或减速度，当超过目标位置时，可通过轴组参数 “ 插补加减速超限 ” 从以下项目中选择。

- 剧烈加减速 （将共混切换为缓冲）
- 剧烈加减速
- 异常停止

版本相关信息

- 在单元版本 1.10 以上的 CPU 单元中，即使设定为 ［剧烈加减速 （将共混切换为缓冲）］，也无法将共混切换为缓冲。此时，将以最大加减速度进行加减速，继续进行共混。
  此外，即使设定为 ［异常停止］，也不会作为异常而停止。此时，也以最大加减速度进行加减速，继续进行共混。
- 使用单元版本 1.09 以上的 CPU 单元时，请注意以下几点。
  在多轴协调控制的共混 （混合）中，曲线运算得到的结果显示，当前指令的执行时间不到 4 个控制周期时，变为缓冲 （待机）的动作。此时，将发生监视信息的 “ 达到共混中继速度所需的移动量不足通知 ”。
  使用单元版本 1.10 及以上的 CPU 单元时，没有上述限制。

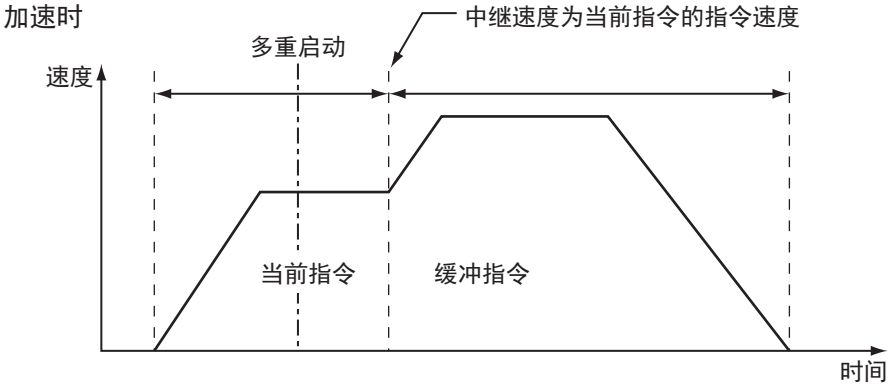
● 低共混 （低速度）

在当前指令的目标位置，以当前指令和缓冲指令中速度较低的目标速度动作。

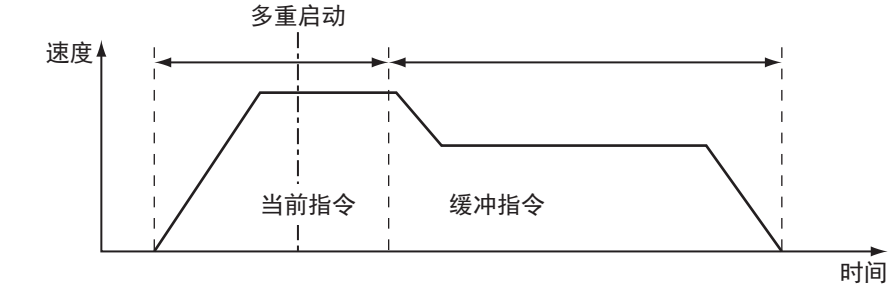
● 前共混 （前速度）

到达当前指令的目标位置前，以当前指令的目标速度动作。到达目标位置后，加减速至缓冲指令的目标速度并动作。

加速时



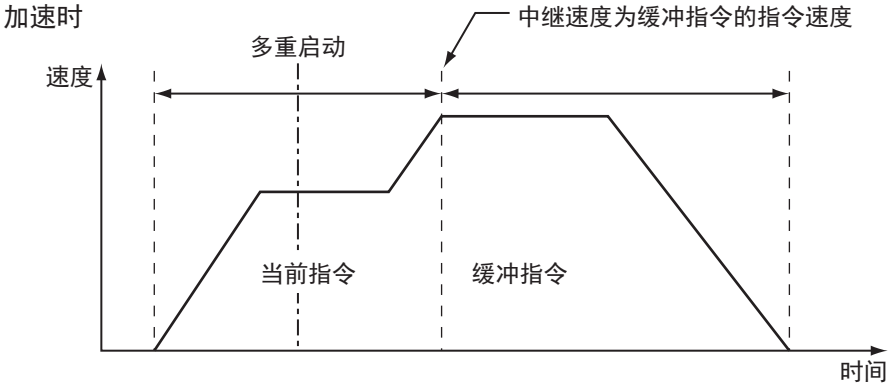
减速时



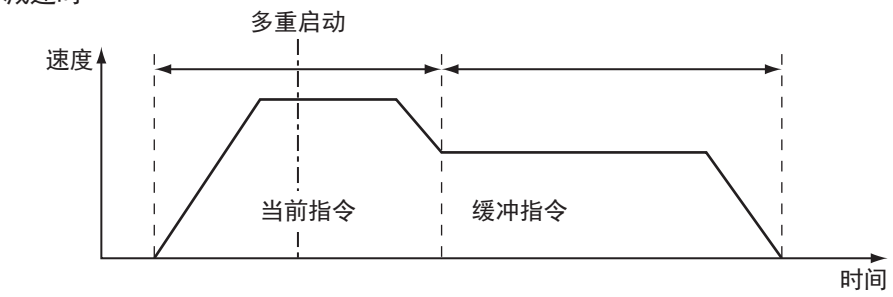
● 后共混 （后速度）

在当前指令的目标位置，以缓冲指令的目标速度动作。

加速时



减速时



● 高共混 （高速度）

在当前指令的目标位置，以当前指令和缓冲指令中速度较高的目标速度动作。

转换模式

在轴组的多重启动中，若插补轨迹的方向发生变化，可能对装置或工件产生冲击。为了缓和这样的冲击，可在运动控制指令的输入变量 “TransitionMode （转换模式） ” 中指定指令间的连接动作方法。

MC 功能模块中可选择的转换模式如下。

编号	转换模式	说明
0	转换无效 (_mcTMNone)	不执行转换处理 （默认）。 冲击不会缓和，但动作时间变短。
10	重叠 (_mcTMCornerSuperimposed)	将当前指令的减速和缓冲指令的加速重叠。 可使插补轨迹的线速度保持稳定。

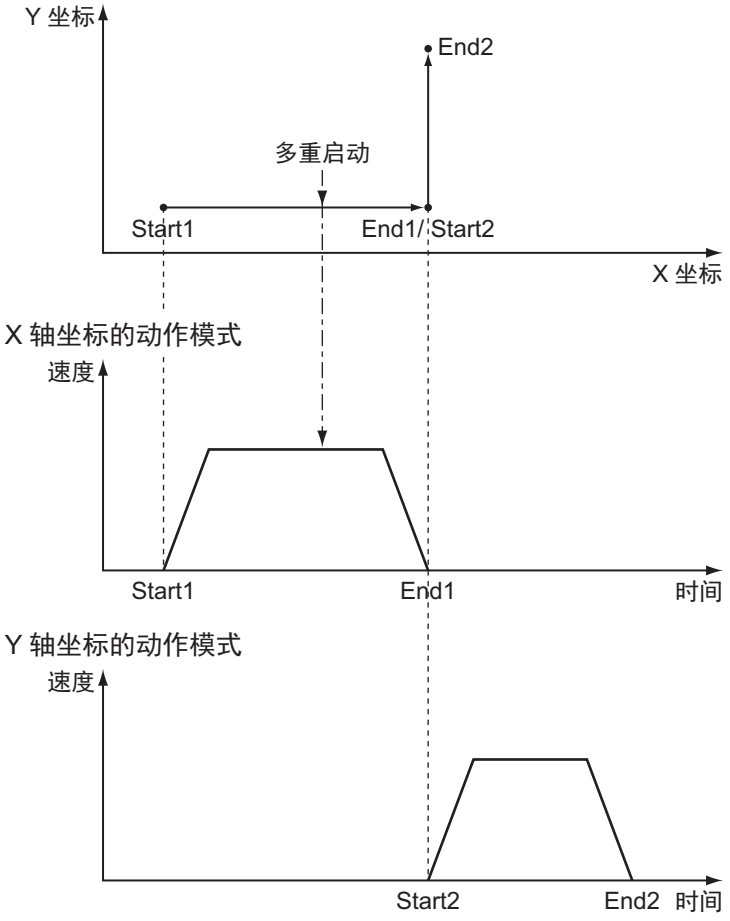
参考

在 PLCopen® 技术规格书中，定义 No.0 ～ 9。 No.10 为 MC 功能模块独有的规格。

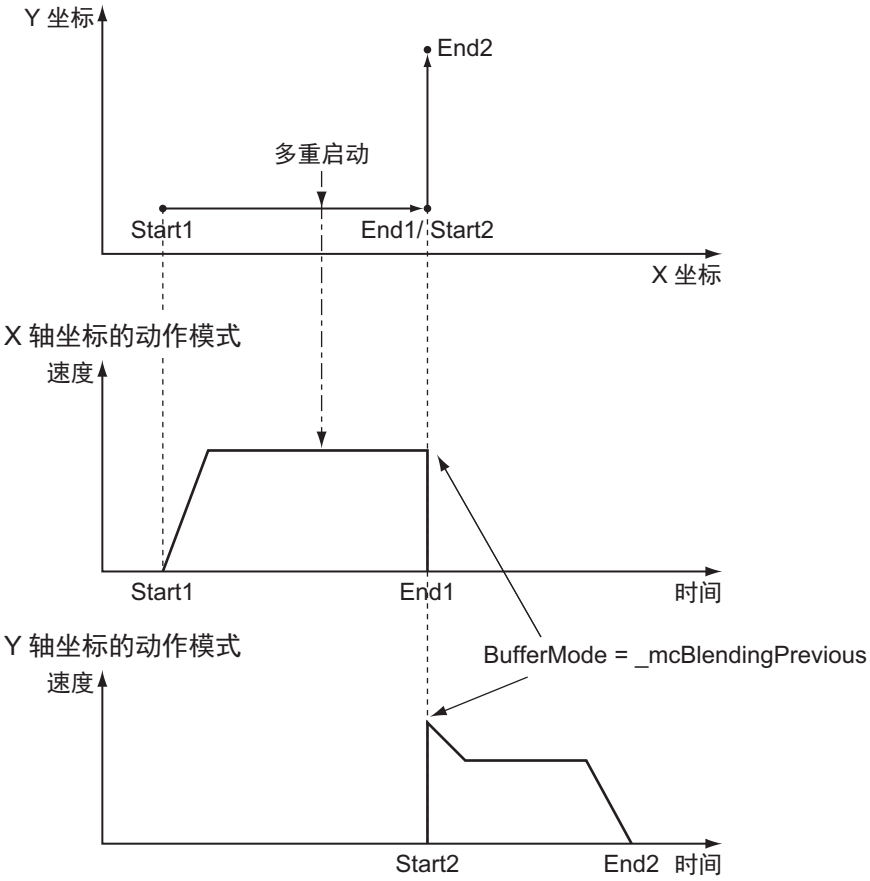
● 转换无效 (0: _mcTMNone) 时

不会执行连接 2 个位置的处理

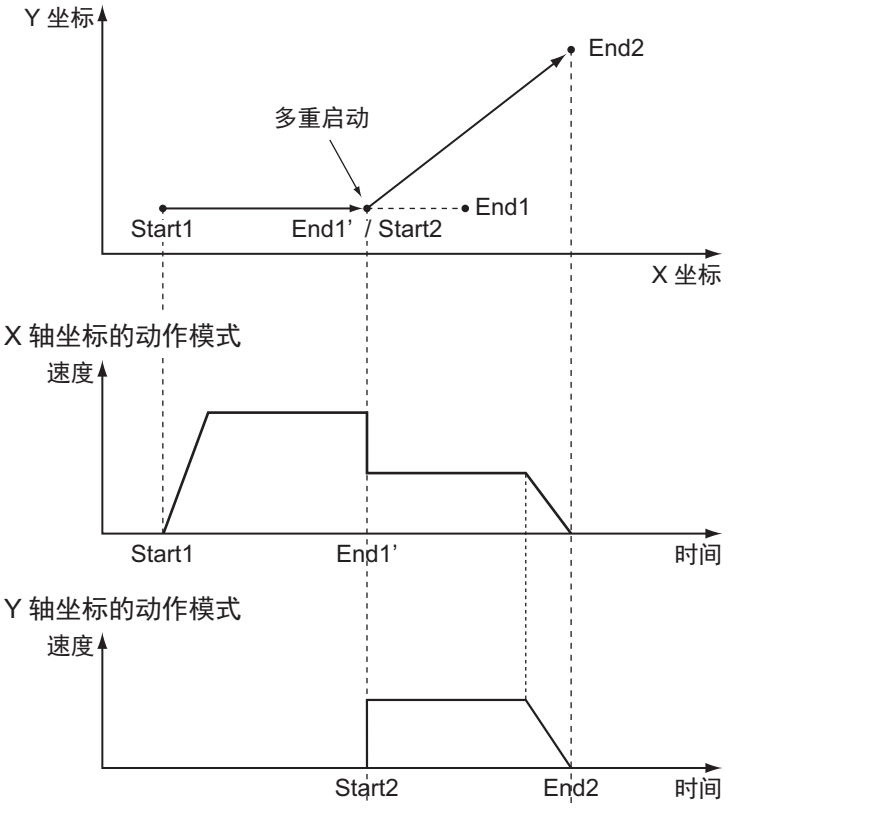
- “BufferMode （缓冲模式选择） ” 为 [1: _mcBuffered] 时，将如下所示，移动到位置 End1 后停止，然后向位置 End2 移动。



- “BufferMode （缓冲模式选择）”为共混时，将如下所示，移动到位置 End1，然后向位置 End2 移动。

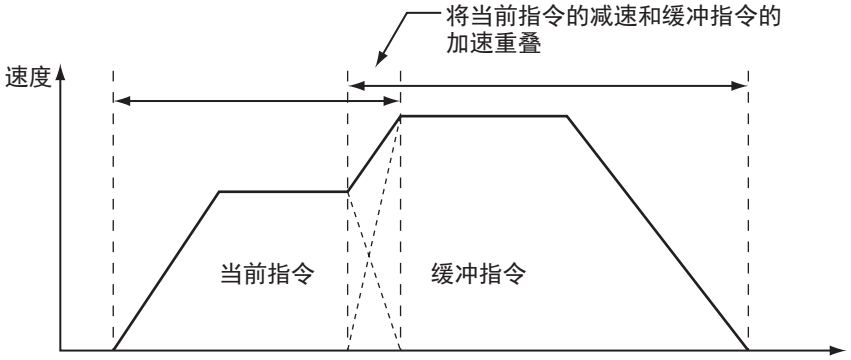


- “BufferMode （缓冲模式选择）”为 [0: _mcAborting] 时，将如下所示，从多重启动的位置 End1 移动到位置 End2。



● 重叠 (10: _mcTMCornerSuperimposed) 时

将当前指令的减速和缓冲指令的加速重叠。
无论缓冲指令中指定的加速度如何，动作时间始终与当前指令的减速度对应的减速时间相同。
在叠加区域中，无论是否指定跃度，都视为无跃度。



_mcTMCornerSuperimposed 时，表示运动控制指令完成的输出变量 “Done （完成） ” 在重叠区间结束时变为 TRUE。

参考

若同时满足以下 2 个条件，则轨迹的线速度保持恒定。
- 将当前指令和缓冲指令的目标速度设为相同值。
- 将当前指令的减速度和缓冲指令的加速度设为相同值。

转换模式和多重启动的组合

不同 TransitionMode （转换模式）和 BufferMode （缓冲模式选择）下指令多重启动的组合。

○: 可动作　–: 发生异常并停止

转换模式	缓冲模式					
	中断	缓冲	低共混	前共混	后共混	高共混
转换无效 (_mcTMNone)	○	○	○	○	○	○
重叠 *1 (_mcTMCornerSuperimposed)	-	-	○	○	○	○

*1. 重叠时，与共混的种类无关，始终叠加当前指令的减速度和缓冲指令的加速度。

9-8 其他功能

下面介绍 MC 功能模块中的其他功能。

使用 NX 系列脉冲输出单元时的区别请参考 《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

9-8-1 当前位置变更功能

将伺服轴的指令当前位置变更为任意值，将反馈当前位置变更为能维持与当前指令位置的位置偏差值的值。

编码器轴时，变更反馈当前位置。

使用 MC_SetPosition （当前位置变更）指令，指定要变更的当前位置。

可在轴动作过程中变更当前位置。

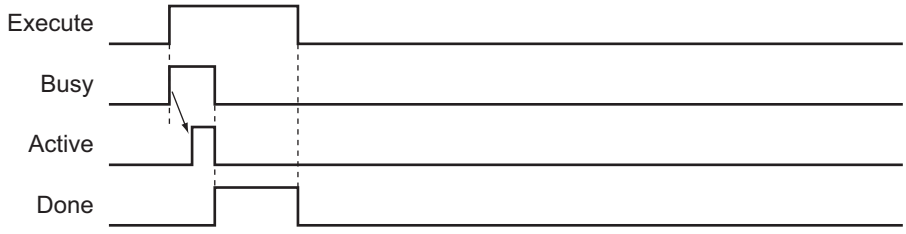
正在执行绝对值定位时，将向变更后的绝对值坐标上的目标位置定位。

正在执行相对值定位时，不变更移动距离。

使用注意事项

- 若在计数模式为旋转模式时指定超出循环计数器范围的位置，指令将发生异常。
- 当前位置变更完成后，将变为原点未确定状态，因此以下功能和指令将无法使用。
  软件限位功能
  高速原点复位
  插补指令 （直线插补和圆弧插补）

● 轴处于停止状态时执行后的时序图



参考

若在 MC_Home （原点复位）指令或 MC_HomeWithParameter （参数指定原点复位）指令中指定 “ 原点预设 ”，可在原点确定状态下变更当前位置。

MC_SetPosition （当前位置变更）指令的详情请参考 《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。

9-8-2 扭矩限制功能

通过切换伺服驱动器扭矩限制功能的有效 / 无效和设定扭矩限制值，限制输出扭矩的功能。
可分别设定正方向的扭矩限制和负方向的扭矩限制。

功能的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_SetTorqueLimit（扭矩限制）指令。

使用注意事项

使用 NX 系列脉冲输出单元时，无法使用本功能。

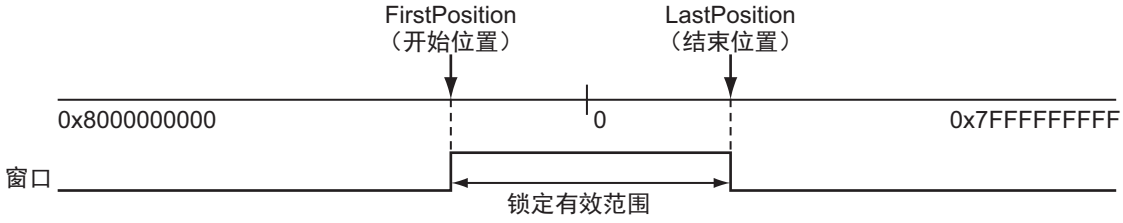
9-8-3 锁定功能

以传感器输入等触发信号的发生为基点，以位置控制为目的，根据触发信号的发生来记录 （锁定）轴位置的功能。
每个轴可同时设定 2 个触发信号。
可使用 MC_TouchProbe （外部锁定有效）指令，对要锁定的轴指定 “ 触发输入条件 ”、“ 窗口有效 ”、“ 停止方法选择 ”。
触发中，除了伺服驱动器中连接的信号外，还可指定用户程序中使用的变量。
要中断锁定功能，可使用 MC_AbortTrigger （外部锁定无效）指令。
锁定功能可在欧姆龙生产的伺服驱动器 G5 系列等支持锁定功能 (Touch probe) 的伺服驱动器或编码器输入终端 GX-EC0211/EC0241 中使用。

使用 WindowOnly （窗口有效），可以只在起点和终点范围内检测触发信号。
因计数模式的不同，范围如下所示。

● 直线模式

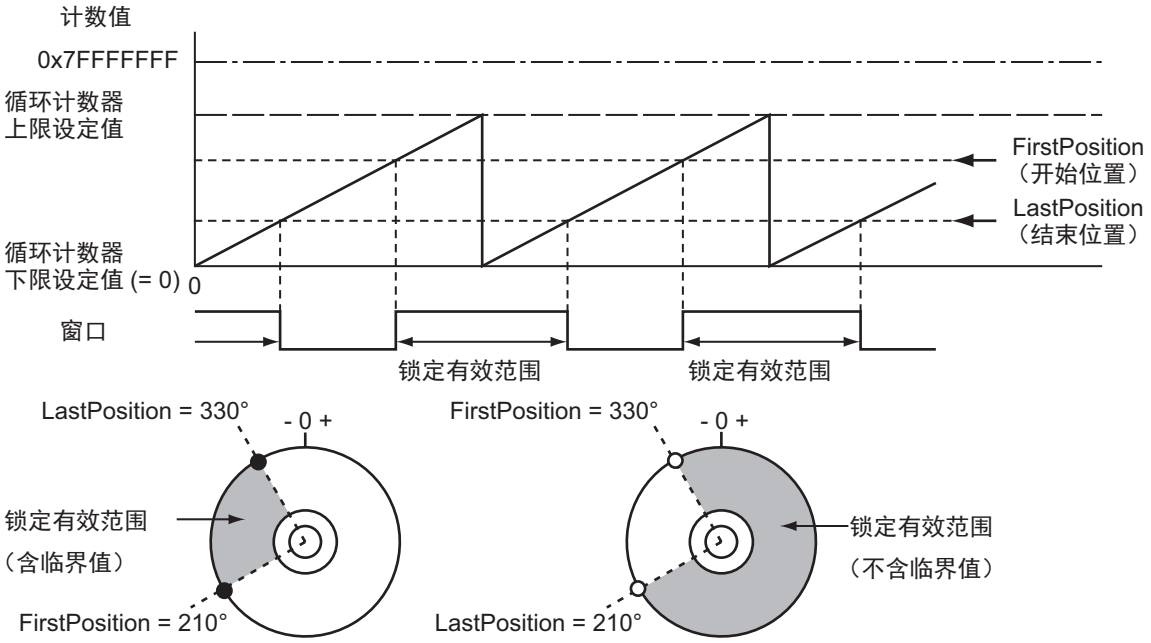
- 仅当 FirstPosition （开始位置）≦ LastPosition （结束位置）时可以。
- 指定为 FirstPosition （开始位置）＞ LastPosition （结束位置）时，指令将发生异常。
- 若指定的位置超出直线模式的位置范围，指令将发生异常。



● 旋转模式

- 可指定为 FirstPosition （开始位置）≦ LastPosition （结束位置）、FirstPosition （开始位置）＞ LastPosition （结束位置）。若为后者，将跨越循环计数器下限设定值。
- 若指定的位置超出循环计数器上下限范围，指令将发生异常。

	First Position ≦ Last Position	First Position ＞ Last Position
有效范围	FirstPosition ～ LastPosition	LastPosition ～ FirstPosition



关于锁定功能的详情，请参考 🕮《NJ/NX 系列　指令基准手册　运动篇 (SBCE-CN5-364)》的 MC_TouchProbe （外部锁定有效）指令及 MC_AbortTrigger （外部锁定无效）指令。

使用 NX 系列脉冲输出单元时的区别请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

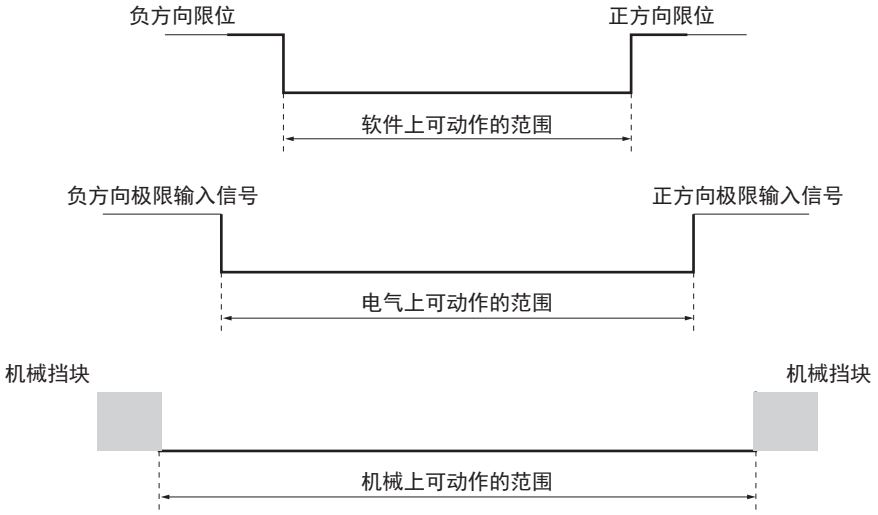
9-8-4　区域选择

判断轴的指令位置或反馈当前位置是否存在于指定范围 （区域）内的功能。
使用 MC_ZoneSwitch （区域监视）指令，指定判定区域的开始位置和结束位置。
轴的位置进入指定区域后，区域监视指令的输出变量 “InZone （区域内） ” 将变为 TRUE。
1 个轴中可指定多个区域，区域可以重复。

区域功能的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_ZoneSwitch （区域监视）指令。

9-8-5 软件限位功能

与硬件的极限输入信号输入不同，利用 MC 功能模块的软件来监视当前位置的功能。
软件限位中监视的范围在轴参数的［正方向软件限位］和［负方向软件限位］中设定。
一般的定位功能可在该软件限位的范围内动作。
通过设定软件限位，可防止用户程序错误和无操作导致的机器破损。



● 软件限位功能相关的轴参数

参数名称	功能	设定范围	初始值
软件限位功能	选择软件限位的功能。 0: 无效 1: 对指令位置有效，减速停止 *1 2: 对指令位置有效，立即停止 3: 对反馈位置有效，减速停止 *1 4: 对反馈位置有效，立即停止	0 ～ 4	0
正方向软件限位	设定正方向的软件限位值。 单位为指令单位。	双精度实数型的负数、正数、 0 *2	2147483647
负方向软件限位	设定负方向的软件限位值。 单位为指令单位。		-2147483648

*1. 正在执行输入变量中包含 “Deceleration （减速度） ” 的指令时，若当前位置超出软件限位的范围，将按照 “Deceleration （减速度） ” 减速停止。正在执行输入变量中不含 “Deceleration （减速度） ” 的指令时，若当前位置超出软件限位的范围，将按照轴参数的最大减速度减速停止。
*2. 转换为脉冲单位的值可在带符号整数型 40 位的范围内设定。

上述轴参数可在 Sysmac Studio 的轴设定或通过 MC_Write （MC 设定写入）指令、MC_WriteAxisParameter （轴参数写入）指令设定。
若在轴或轴组正在动作时变更设定值，开始下次动作时，变更的设定将变为有效。

因轴的动作状态和运动控制指令不同，软件限位会在以下两种情况下运行。

● 启动运动指令时

- 当前位置在软件限位范围内时，当目标位置超出软件限位范围时，指令将发生异常。
- 当前位置超出软件限位范围时，可向软件限位范围内移动。此时，目标位置不需要在软件限位范围内。



使用注意事项

轴发出的指令请勿在目标位置超出软件限位范围时启动。

● 轴正在动作时

轴正在动作时，如定位动作中、同步动作中、连续动作中、多轴协调动作中，动作如下。

- 软件限位功能为［对指令位置有效］时，当指令位置从范围内变为范围外时，将发生轴异常。
- 软件限位功能为［对反馈位置有效］时，当反馈位置从范围内变为范围外时，将发生轴异常。



参考

计数模式为直线模式且在原点确定状态下，软件限位功能可设为有效。
以下情况下，无论轴参数的设定如何，软件限位功能始终为无效。

- 计数模式为旋转模式时
- 未确定原点的状态时
- 原点复位动作中

MC 设定写入指令及轴参数写入指令的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_Write （MC 设定写入）指令及 MC_WriteAxisParameter （轴参数写入）指令。

9-8-6 位置偏差监视功能

轴的指令位置和反馈位置的差即为位置偏差。 MC 功能模块会按运动控制周期对位置偏差进行监视。
位置偏差的值超出轴参数的 “ 位置偏差超限值 ” 时，将发生轻度故障等级的 “ 位置偏差超限 ”。此外，超出 “ 位置偏差警告值 ” 时，将发生监视信息的 “ 位置偏差警告 ”。原点复位正在执行推压动作时，位置偏差监视为无效。

● 位置偏差监视功能相关的轴参数

通过设定轴参数，可设定位置偏差监视的阈值。 “ 位置偏差警告值 ” 请设定为小于 “ 位置偏差超限值 ”。这些轴参数在 Sysmac Studio 中设定。

参数名称	功能	设定范围	初始值
位置偏差超限值	设定位置偏差超限的阈值。 若设定为 “0”，位置偏差超限检查为无效。（单位 : 指令单位）	双精度实数型的正数、 0	0
位置偏差警告值	设定位置偏差警告的阈值。 若设定为 “0”，位置偏差警告检查为无效。（单位 : 指令单位）	双精度实数型的正数、 0 （小于位置偏差超限值）	0

使用 NX 系列脉冲输出单元时的区别请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

9-8-7 偏差计数器复位

偏差计数器复位是指将位置偏差变为 “0” 的功能。
要在用户程序中执行偏差计数器复位，需要使用 “MC_ResetFollowingError （偏差计数器复位） ” 指令。
偏差计数器复位指令可在单轴定位中及原点复位中使用。
若在轴动作中执行偏差计数器复位，动作中的运动控制指令将中断动作，指令位置变更为与反馈位置相同的值。
即使执行偏差计数器复位，原点确定状态也不会变化。

偏差计时器复位指令的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_ResetFollowingError （偏差计数器复位）指令。

● 偏差计数器复位相关的轴参数

通过设定轴参数，可选择在立即停止时、极限输入停止时及原点复位完成时执行偏差计数器复位。
这些轴参数在 Sysmac Studio 中设定。

参数名称	功能	设定范围	初始值
立即停止输入停止方法	设定立即停止输入有效时 MC 功能模块的停止方法。 0: 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	0、 2、 3	0
极限输入停止方法	设定正方向极限输入或负方向极限输入有效时 MC 功能模块的停止方法。 0: 立即停止指令值 1: 减速停止指令值 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	0 ～ 3	0

使用 NX 系列脉冲输出单元时的区别请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

9-8-8 轴间偏差监视功能

监视 2 个轴之间的指令位置或反馈位置偏差的功能。
使用 MC_AxesObserve （轴间偏差监视）指令，指定监视对象的 2 个轴和轴间偏差容许值。
若超出容许值，轴间偏差监视指令的输出变量 “Invalid （偏差超限） ” 将变为 TRUE。

在门式控制等 2 个轴执行相同动作的装置中，使用轴间监视功能，当轴间偏差过大时，可自由编写处理程序。

使用注意事项

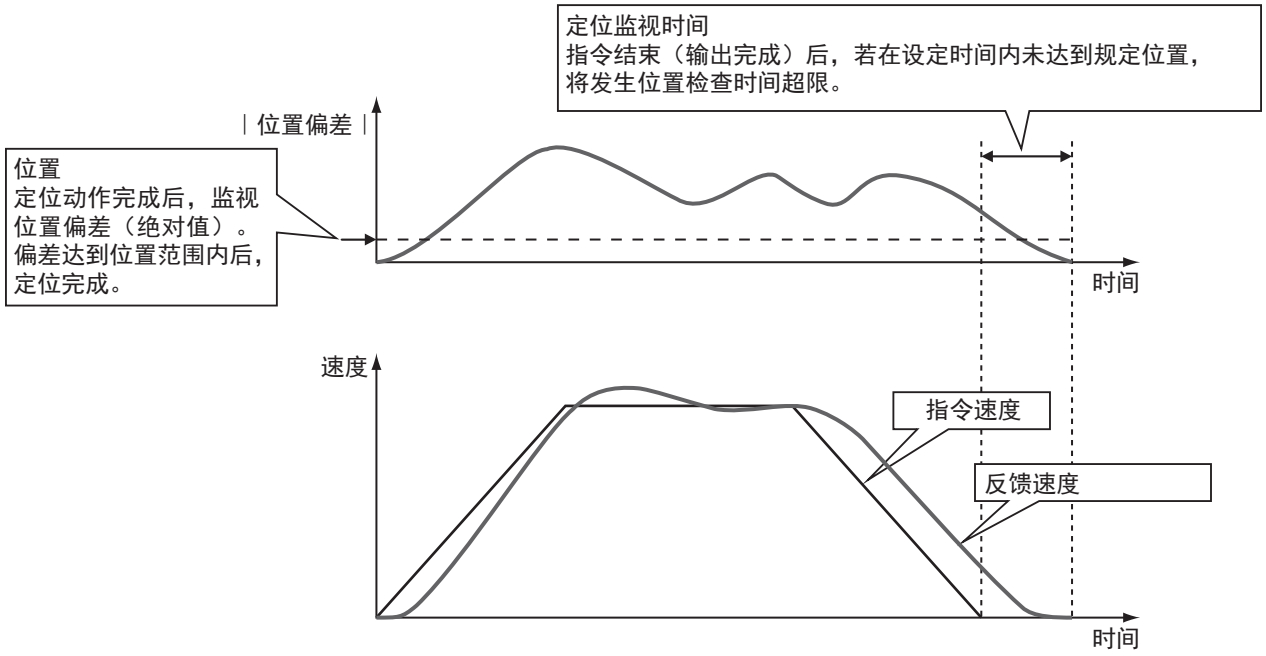
即使超出轴间偏差容许值， MC 功能模块也不会发生异常。
请根据输出变量 “Invalid （偏差超限） ”，通过程序进行处理，如通过用户程序来停止轴动作。

轴间偏差监视功能的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_AxesObserve （轴间偏差监视）指令。

使用 NX 系列脉冲输出单元时的区别请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

9-8-9 位置检查

定位动作或原点复位动作时，对实际的反馈当前位置是否到达目标位置中指定的范围进行检查的功能。
目标位置的指令输出完成后，若目标位置和反馈当前位置的差小于［位置宽度］，则定位完成。
若［位置检查时间］内未完成定位，指令将发生异常。



● 位置检查相关的轴参数

通过轴参数，可设定定位完成的检查条件。
若要在确定定位完成后再开始下一动作，需要设定位置检查时间。

参数名称	功能	设定范围	初始值
位置宽度	设定定位完成宽度 （单位 : 指令单位）	双精度实数型的正数、 0	10
位置检查时间	以毫秒为单位设定完成定位的检查时间。 若设定为 “0”，仅在原点复位中确定原点时继续执行定位完成检查，不是原点复位时，不执行定位完成检查。（单位 : ms）	0 ～ 10000	0

上述轴参数可在 Sysmac Studio 的轴设定或通过 MC_Write （MC 设定写入）指令、MC_WriteAxisParameter （轴参数写入）指令设定。

参考

- 位置检查在 MC 功能模块上执行。不使用伺服驱动器上的功能。
- 若要尽快执行动作，则不设定位置检查时间，这样无需等待定位完成即可进行下一动作。
- 重新接通了 CPU 单元的电源或用 Sysmac Studio 的 “ 同步 ” 功能下载了程序时，恢复为 Sysmac Studio 中设定的值。

  MC_Write （MC 设定写入）指令及 MC_WriteAxisParameter （轴参数写入）指令可用于临时变更位置检查时间。

● 位置检查相关的监视

在程序中，可通过读取轴变量来监视定位完成。
此外，以 _MC_AX[*] 为例对变量名称进行说明， _MC1_AX[*]、 _MC2_AX[*] 亦同。

变量名称	数据类型	名称	功能
_MC_AX[0-255].Details.Idle	BOOL	停止中	除位置等待外，未计算指令值时变为 TRUE。*1 与 Idle 和 InPosWaiting 互斥，因此不会同时变为 TRUE。
_MC_AX[0-255].Details.InPosWaiting	BOOL	位置等待	位置等待时变为 TRUE。 在带位置检查的定位中，正在进行位置检查的状态。

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

在轴组的定位中，可通过读取轴组变量来监视定位完成。
此外，以 _MC_GRP[*] 为例对变量名称进行说明， _MC1_GRP[*]、 _MC2_GRP[*] 亦同。

变量名称	数据类型	名称	功能
_MC_GRP[0-63].Details.Idle	BOOL	停止中	除位置等待外，未计算指令值时变为 TRUE。*1 与 Idle 和 InPosWaiting 互斥，因此不会同时变为 TRUE。
_MC_GRP[0-63].Details.InPosWaiting	BOOL	位置等待	任一构成轴处于位置等待时变为 TRUE。*2 在带位置检查的定位中，正在进行位置检查的状态。

*1. 以速度 “0” 动作时也视为正在运算的状态。
*2. 轴组中的所有构成轴进入各轴参数的位置宽度范围后，变为 FALSE。

MC 设定写入指令及轴参数写入指令的详情请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_Write （MC 设定写入）指令及 MC_WriteAxisParameter （轴参数写入）指令。

使用 NX 系列脉冲输出单元时的区别请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

9-8-10 轴使用变更功能

利用 MC_ChangeAxisUse （轴使用变更）指令，临时切换轴参数 ［轴使用］的功能。可切换的轴为轴参数的 ［轴使用］设定为 ［未使用轴 （可切换为使用轴）］或 ［使用轴］的轴。
［轴使用］为 ［未使用轴 （可切换为使用轴）］，［轴种类］为 ［伺服轴］或 ［虚拟伺服轴］时，可分配到轴组中。
在 CPU 单元版本 1.04 以上和 Sysmac Studio Ver.1.05 以上的组合中可使用。

使用注意事项

- 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］。
- 轴参数的 ［轴使用］设定为 ［未使用轴 （不可切换为使用轴）］时，不可分配到轴组中。

功能的详情请参考 《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_ChangeAxisUse （轴使用变更）指令。

MC_ChangeAxisUse （轴使用变更）指令的应用示例请参考 《NJ/NX 系列 CPU 单元 用户手册 软件篇 （SBCA-CN5-359）》。

9-8-11 数字凸轮开关有效功能

利用 MC_DigitalCamSwitch （数字凸轮开关有效）指令，根据轴的位置将数字输出设为 ON 或 OFF。
设定指令的输入变量 “ValueSource” 后，还可加入加减速度。
该功能必须与 NX_AryDOutTimeStamp 指令及支持时间戳方式的数字输出单元组合使用。
NX_AryDOutTimeStamp 指令将在指定的时间戳时间点执行指定数字输出的 ON/OFF。
可在 CPU 单元版本 1.06 以上和 Sysmac Studio Ver.1.07 以上的组合中使用。

使用注意事项

本指令对分配有 NX 系列位置接口单元的轴有效。
可使用的 NX 单元为 NX-EC0□□□、 NX-ECS□□□，变化时刻获取功能需要处于运行状态。

数字凸轮开关有效功能的详情请参考 《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的 MC_DigitalCamSwitch （数字凸轮开关有效）指令。

NX_AryDOutTimeStamp 指令指令的详情请参考 《NJ/NX 系列 指令基准手册 基本篇 (SBCA-CN5-360)》。

关于支持时间戳方式的数字输出单元，请参考 《NX 系列 数字 I/O 单元 用户手册 （SBCA-CN5-407B 以后）》。

变化时刻获取功能及时间戳请参考 《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374B 以后)》。

9-8-12 任意坐标系的 3D 运动监视显示

平面关节型机器人或垂直多关节型机器人等，在用户程序中实现直角坐标系以外的坐标系时，在 Sysmac Studio 上 3D 显示机器人指尖轨迹的功能。
创建 _sMC_POSITION_REF 型用户定义变量，在 3D 运动显示器中显示。

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

● _sMC_POSITION_REF 型

_sMC_POSITION_REF 型的成员如下。

成员	数据类型	名称
CommandPosition	ARRAY [0..5] OF LREAL	指令当前位置
ActualPosition	ARRAY [0..5] OF LREAL	反馈当前位置

下面介绍各成员。

成员	说明
用户定义变量 .CommandPosition[0]	指令当前位置的 X 轴成分。 代入表示用户程序中生成的指令当前位置在 X 轴上的位置的用户定义变量。
用户定义变量 .CommandPosition[1]	指令当前位置的 Y 轴成分。 代入表示用户程序中生成的指令当前位置在 Y 轴上的位置的用户定义变量。
用户定义变量 .CommandPosition[2]	指令当前位置的 Z 轴成分。 代入表示用户程序中生成的指令当前位置在 Z 轴上的位置的用户定义变量。
用户定义变量 .CommandPosition[3] ～ [5]	未使用
用户定义变量 .ActualPosition[0]	反馈当前位置的 X 轴成分。 代入表示用户程序中使用的反馈当前位置在 X 轴上的位置的用户定义变量。
用户定义变量 .ActualPosition[1]	反馈当前位置的 Y 轴成分。 代入表示用户程序中使用的反馈当前位置在 Y 轴上的位置的用户定义变量。
用户定义变量 .ActualPosition[2]	反馈当前位置的 Z 轴成分。 代入表示用户程序中使用的反馈当前位置在 Z 轴上的位置的用户定义变量。
用户定义变量 .ActualPosition[3] ～ [5]	未使用

在各成员中代入用户定义变量。示例如下。

名称	数据类型	说明
3D_position	_sMC_POSITION_REF	3D 显示所需的用户定义变量
MCS_Cmd_TransX	LREAL	表示用户程序中生成的指令当前位置在 X 轴上的位置的用户定义变量
MCS_Cmd_TransY	LREAL	表示用户程序中生成的指令当前位置在 Y 轴上的位置的用户定义变量
MCS_Cmd_TransZ	LREAL	表示用户程序中生成的指令当前位置在 Z 轴上的位置的用户定义变量
MCS_Act_TransX	LREAL	表示用户程序中使用的反馈当前位置在 X 轴上的位置的用户定义变量
MCS_Act_TransY	LREAL	表示用户程序中使用的反馈当前位置在 Y 轴上的位置的用户定义变量
MCS_Act_TransZ	LREAL	表示用户程序中使用的反馈当前位置在 Z 轴上的位置的用户定义变量

```
3D_position.CommandPosition[0] := MCS_Cmd_TransX;
3D_position.CommandPosition[1] := MCS_Cmd_TransY;
3D_position.CommandPosition[2] := MCS_Cmd_TransZ;
3D_position.ActualPosition[0] := MCS_Act_TransX;
3D_position.ActualPosition[1] := MCS_Act_TransY;
3D_position.ActualPosition[2] := MCS_Act_TransZ;
```

● 使用步骤的概要

1 **创建 _sMC_POSITION_REF 型用户定义变量。**

2 **创建程序，使其在创建的用户定义变量的各成员中，代入表示要3D显示的“指令当前位置”和“反馈当前位置 ” 的用户定义变量。**

3 **从 3D 结构模型列表的种类中选择 ［指定坐标］。**
3D 结构模型参数设定中将显示 _sMC_POSITION_REF 型。

4 **在 3D 结构模型参数设定的设定值栏中设定创建的用户定义变量。**

5 **执行用户程序。**

6 **开始利用数据追踪进行追踪，进行数据采样。**

7 **在数据追踪画面中确认追踪结果。**

关于 3D 运动显示的详情，请参考 🕮《Sysmac Studio Version 1 操作手册 （SBCA-CN5-362）》。

10

示例程序

本章介绍原点复位或异常监视等基本使用方法，以及绝对值定位或凸轮动作等轴动作的程序示例。

10-1 示例程序的概要 10-2
10-1-1 使用设备 10-2
10-1-2 安装和配线 10-2
10-1-3 设定 10-2

10-2 基本程序示例 10-3
10-2-1 EtherCAT 通信的监视和伺服 ON 10-3
10-2-2 通过主控指令进行轴动作的联锁 10-5
10-2-3 异常的监视和异常解除 （单轴动作、同步动作） 10-7
10-2-4 异常的监视和异常解除 （多轴协调动作） 10-9
10-2-5 指令异常的监视方法 10-14
10-2-6 异常解除的监视方法 10-16
10-2-7 单轴动作时的轴停止 10-18
10-2-8 多轴协调动作中的轴组停止 10-22
10-2-9 原点复位和绝对值定位 10-28
10-2-10 通过重启指令来变更目标位置 10-32
10-2-11 中断固定尺寸定位 10-37
10-2-12 通过重启指令，切换凸轮表 10-42
10-2-13 使用凸轮轮廓曲线进行同步位置修正 10-51
10-2-14 凸轮动作时进行主轴的相位补偿 10-60
10-2-15 在速度控制过程中变更当前位置 10-68
10-2-16 凸轮数据变量的变更和凸轮表的保存 10-74
10-2-17 轴参数的临时替换 10-83
10-2-18 凸轮表的终点索引更新 10-86

10-1 示例程序的概要

下面介绍本章中记载的示例程序的共通事项。

使用注意事项

- 本章的示例程序仅记载使用 MC 功能模块的功能部分。
- 创建实际装置的程序时，请创建装置动作相关的联锁程序或与其他设备的输入输出、控制步骤等相关的程序。
- 请创建用户程序，使实际装置按预期动作。
- 创建的程序应先充分确认动作，再用于正式运行。
- NX 系列位置接口单元的使用示例请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

10-1-1 使用设备

这里介绍的示例程序以使用以下设备为前提。

设备	伺服构成示例
CPU 单元	NJ501-1□00 (Ver.1.0)
电源单元	NJ-Px3001
伺服驱动器	R88D-KN□-ECT (Ver.2.1)
伺服电机	R88M-K□
编码器输入终端	GX-EC0211 (Ver.1.1)

10-1-2 安装和配线

各使用设备的安装和配线请参考以下手册。

使用设备	参考手册
CPU 单元和电源单元	NJ/NX 系列 CPU 单元 用户手册 硬件篇 （SBCA-CN5-358）
伺服驱动器 / 电机	AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型 用户手册 (SBCE-CN5-365)
编码器输入终端	GX 系列 EtherCAT 从站 用户手册 (SBCD-CN5-350)
EtherCAT 通信的配线	

10-1-3 设定

各种设定请参考以下手册。

使用设备	参考手册
控制器设定	NJ/NX 系列 CPU 单元 用户手册 软件篇 （SBCA-CN5-359）
运动控制设定	*「3-2 轴的设定步骤* (P.3-9)」、*「5-2 轴参数* (P.5-4)」、*「A-1 与伺服驱动器连接* (P.A-2)」
伺服参数	AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型 用户手册 (SBCE-CN5-365)

10-2 基本程序示例

下面介绍 MC 功能模块的基本功能相关的用户程序示例。

使用注意事项

- 参考本程序示例时，请追加适合的联锁程序，以便满足装置的动作条件。
- 程序中使用的变量请在 Sysmac Studio 的［编程］画面中输入。

10-2-1 EtherCAT 通信的监视和伺服 ON

对象伺服驱动器和 EtherCAT 通信的过程数据通信建立后，通过 MC_Power （可运行）指令将伺服驱动器变为伺服 ON。

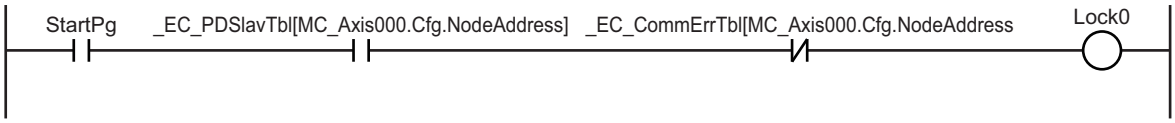
记载梯形图程序的示例和 ST 程序的示例。

程序中使用的主要变量

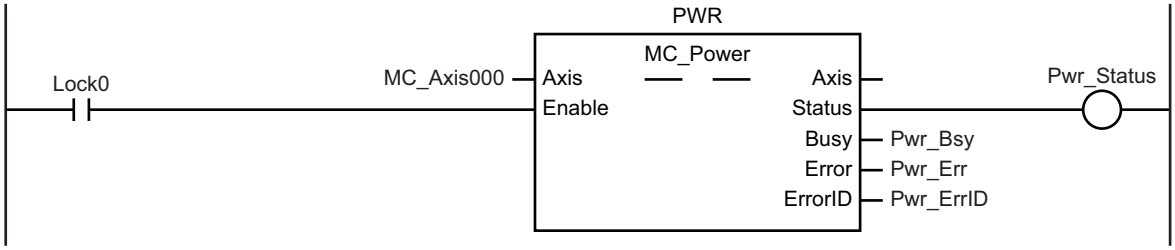
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Cfg.NodeAddress	UINT	—	节点地址。
_EC_PDSlavTbl[N]	BOOL	FALSE	TRUE 时，表示节点地址 N 的 EtherCAT 通信过程数据通信为可操作状态。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 通信的过程数据通信正在正常执行，则伺服 ON。

梯形图程序

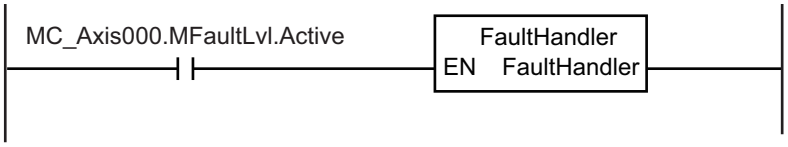
确认接点 StartPg 是否为 TRUE 且过程数据通信是否正在正常执行



若过程数据通信正在正常执行，则轴 0 变为伺服 ON



轴 0 发生轻度故障等级的异常时，执行异常处理 FaultHandler
异常处理的内容请根据装置进行编程



ST 程序

```
// 若接点 StartPg 为 TRUE 且过程数据通信正在正常执行，则轴 0 变为伺服 ON
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
   Pwr_En:=TRUE;
ELSE
   Pwr_En:=FALSE;
END_IF;


// 轴 0 发生轻度故障等级的异常时，执行异常处理 FaultHandler
// 异常处理的内容请根据装置进行编程
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
   FaultHandler();
END_IF;

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

10-2-2 通过主控指令进行轴动作的联锁

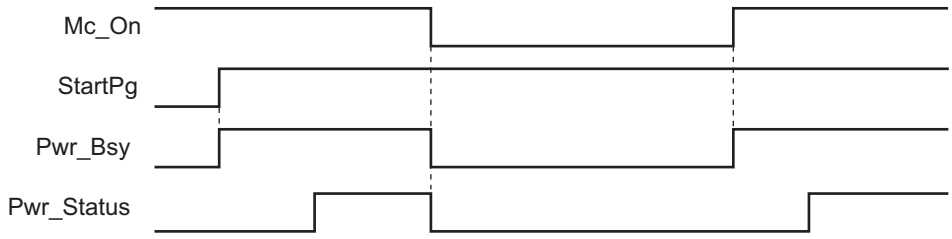
在梯形图程序中，在 MC （主控）指令和 MCR （主控复位）指令之间配置 MC_Power （可运行）指令，执行轴动作的联锁控制。
若将接点 Mc_On 设为 FALSE， MC 指令和 MCR 指令之间记述的 MC_Power （可运行）指令将变为无效，伺服变为 OFF。
此时，正在执行的动作指令的输出变量 “CommandAborted （执行中断） ” 变为 TRUE，轴动作停止。

MC 指令在 ST 程序中无法使用。
因此，仅记载梯形图程序的示例。

程序中使用的主要变量

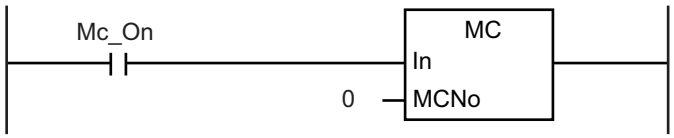
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时，变为 TRUE。
Mc_On	BOOL	FALSE	控制 MC 指令的有效及无效。 本示例程序中记载控制相关的记述。在实际程序中，请根据装置的动作进行控制。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 通信的过程数据通信正在正常执行，则伺服 ON。

时序图

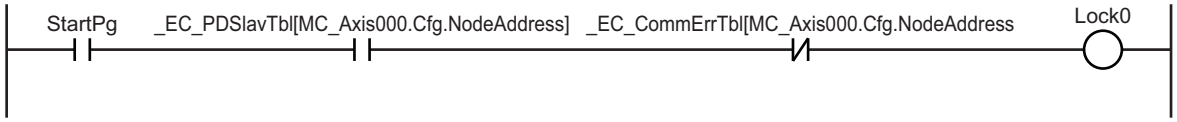


梯形图程序

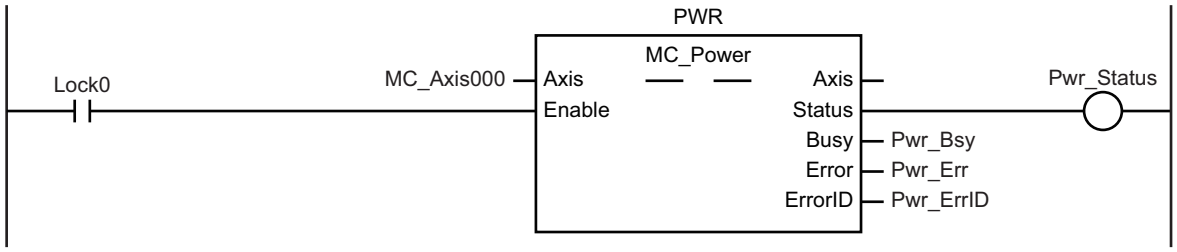
接点 Mc_On 变为 TRUE 后，开始主控



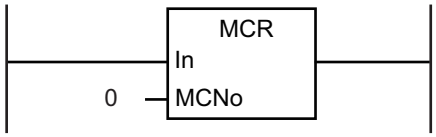
确认接点 StartPg 是否为 TRUE 且过程数据通信是否正在正常执行



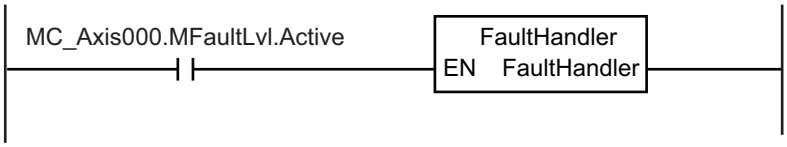
若过程数据通信正在正常执行，则轴 0 变为伺服 ON



结束主控



轴 0 发生轻度故障等级的异常时，执行异常处理 FaultHandler
异常处理的内容请根据装置进行编程



10-2-3 异常的监视和异常解除 （单轴动作、同步动作）

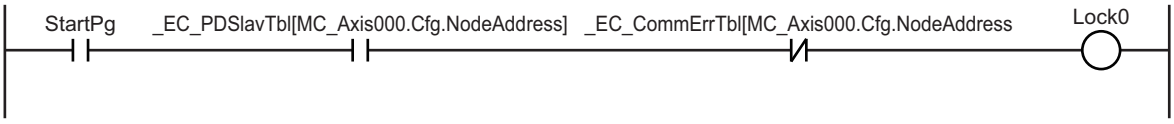
根据轴变量中的 “ 轴 发生轻度故障 ” 的状态，监视异常的有无。
发生轻度故障等级的异常时， MC_Power （可运行）指令的输入变量 “Enable （有效） ” 将变为 FALSE，伺服变为 OFF。
若外部按钮为 ON、指令当前速度为 0，则通过 MC_Reset （轴错误复位）指令解除异常。
记载梯形图程序的示例和 ST 程序的示例。

程序中使用的主要变量

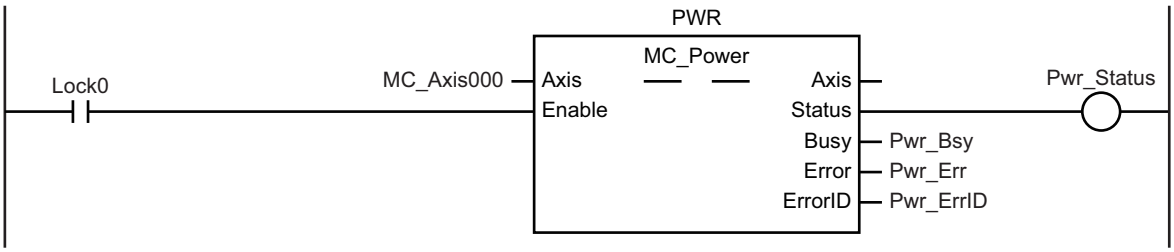
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Status.ErrorStop	BOOL	FALSE	轴 0 发生轻度故障等级的异常，轴正在减速停止或已停止时，变为 TRUE。
MC_Axis000.Details.Idle	BOOL	FALSE	位置等待以外，且轴 0 的指令当前速度为 0 时，变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 通信的过程数据通信正在正常执行，则伺服 ON。
ResetON	BOOL	FALSE	表示解除异常所需的外部按钮的状态。

梯形图程序

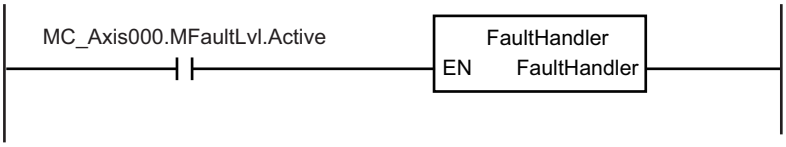
确认接点 StartPg 是否为 TRUE 且过程数据通信是否正在正常执行



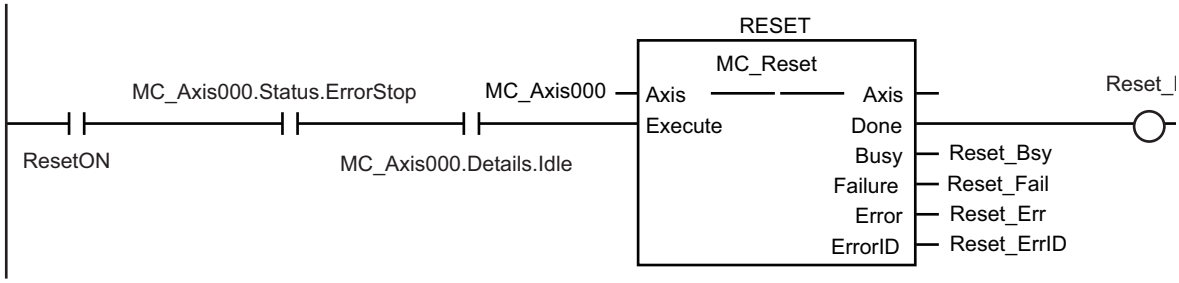
若过程数据通信正在正常执行，则轴 0 变为伺服 ON



轴 0 发生轻度故障等级的异常时，执行异常处理 FaultHandler
异常处理的内容请根据装置进行编程



若因外部按钮打开而使 ResetON 变为 TRUE、指令当前速度为 0，则解除异常



ST 程序

```
// 确认 StartPg 为 TRUE 且过程数据通信正在正常执行，
// 若与轴 0 的过程数据通信正在正常执行，则轴 0 变为伺服 ON
// 若未执行过程数据通信，则轴 0 变为伺服 OFF
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
   Pwr_En:=TRUE;
ELSE
   Pwr_En:=FALSE;
END_IF;


// 轴 0 发生轻度故障等级的异常时，执行异常处理 FaultHandler
// 异常处理的内容请根据装置进行编程
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
   FaultHandler();
END_IF;


// 若因外部按钮打开而使 ResetON 变为 TRUE、指令当前速度为 0，则解除异常
IF (ResetOn=TRUE)
 AND (MC_Axis000.Status.ErrorStop=TRUE)
 AND (MC_Axis000.Details.Idle=TRUE) THEN
   Reset_Ex := TRUE;   // 轻度故障的解除
END_IF;

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

// MC_Reset
RESET(
   Axis := MC_Axis000,
   Execute := Reset_Ex,
   Done => Reset_D,
   Busy => Reset_Bsy,
   Failure => Reset_Fai,
   Error => Reset_Err,
   ErrorID => Reset_ErrID
);
```

10-2-4 异常的监视和异常解除（多轴协调动作）

根据轴变量中的“轴 发生轻度故障”的状态，以及轴组变量中的“轴组 发生轻度故障”的状态，监视异常的有无。

发生轻度故障等级的异常时， MC_GroupDisable （轴组无效）指令的输入变量“Execute （启动）”将变为 FALSE，轴组变为无效。

若外部按钮为 ON、轴组的指令当前速度为 0，则通过 MC_GroupReset （轴组错误复位）指令解除异常。

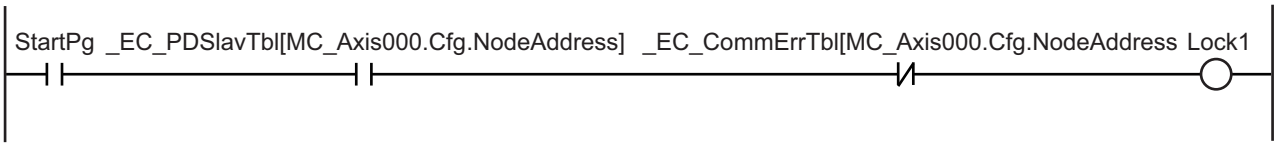
记载梯形图程序的示例和 ST 程序的示例。

程序中使用的主要变量

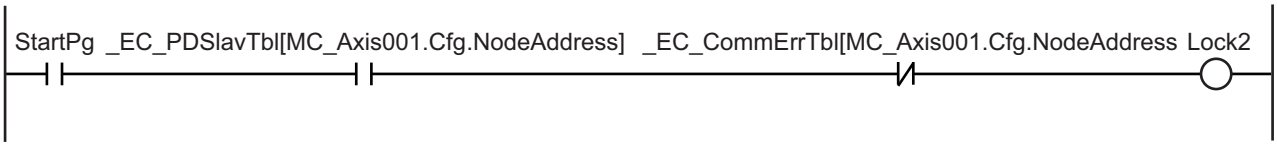
名称	数据类型	初始值	注释
MC_Group000	_sGROUP_REF	—	轴组 0 的轴组变量。
MC_Group000.MFaultLvl.Active	BOOL	FALSE	轴组 0 发生轻度故障等级的异常时，变为 TRUE。
MC_Group000.Details.Idle	BOOL	FALSE	位置等待以外，且轴组 0 的指令插补速度为 0 时，变为 TRUE。
MC_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 0 处于原点确定状态时变为 TRUE。
MC_Axis001	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis001.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
Pwr1_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR1 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。
Pwr2_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR2 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 通信的过程数据通信正在正常执行，则伺服 ON。
ResetON	BOOL	FALSE	表示解除异常所需的外部按钮的状态。

梯形图程序

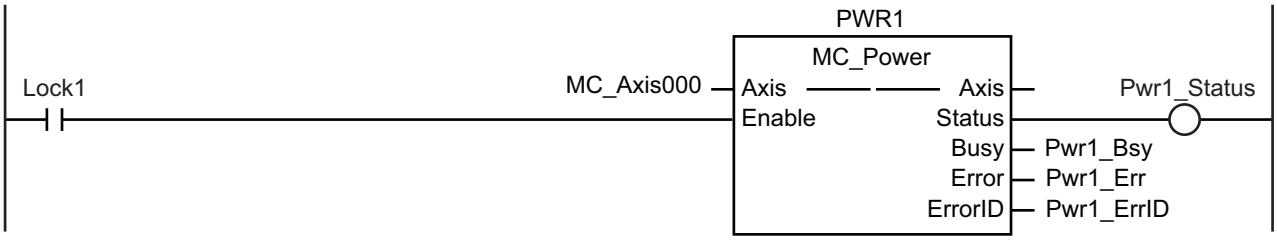
确认接点 StartPg 是否为 TRUE 且与轴 0 的过程数据通信是否正在正常执行



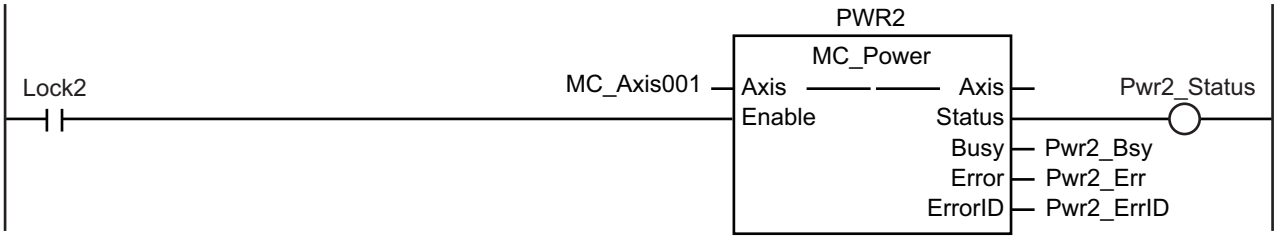
确认接点 StartPg 是否为 TRUE 且与轴 1 的过程数据通信是否正在正常执行



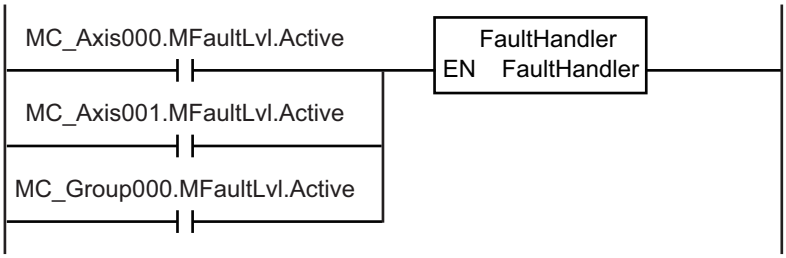
若与轴 0 的过程数据通信正在正常执行，则轴 0 变为伺服 ON



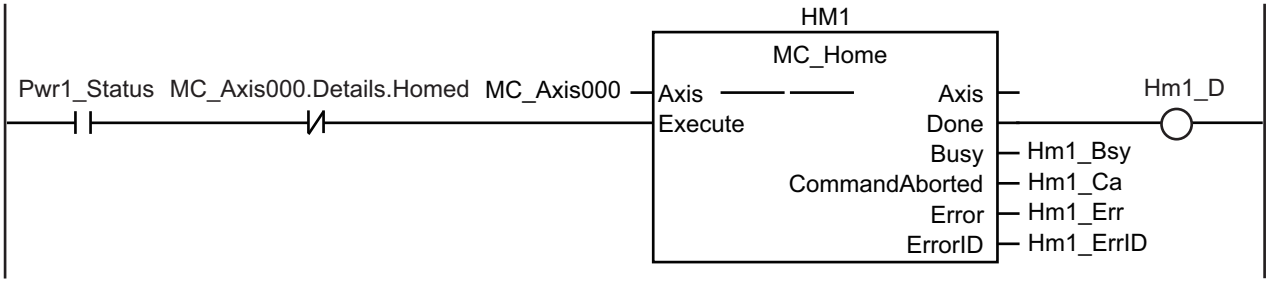
若与轴 1 的过程数据通信正在正常执行，则轴 1 变为伺服 ON



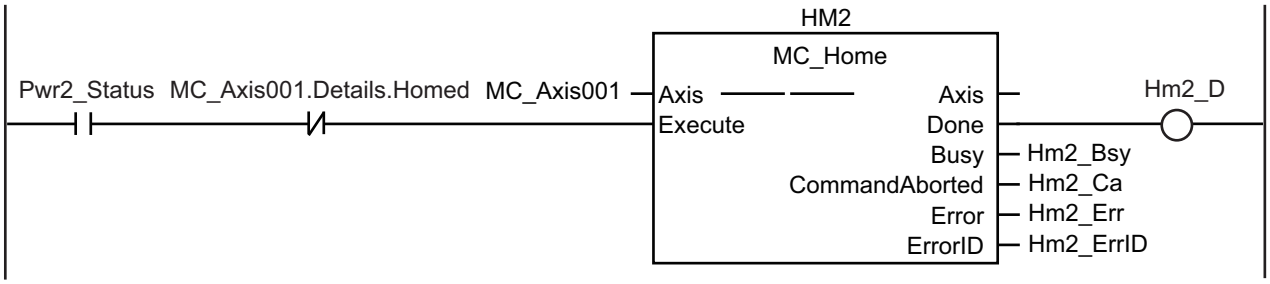
轴组或构成轴发生轻度故障等级的异常时，执行异常处理 FaultHandler
异常处理的内容请根据装置进行编程



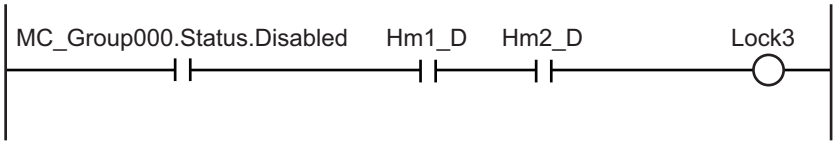
轴 0 为伺服 ON 状态且处于原点未确定状态时，执行原点复位



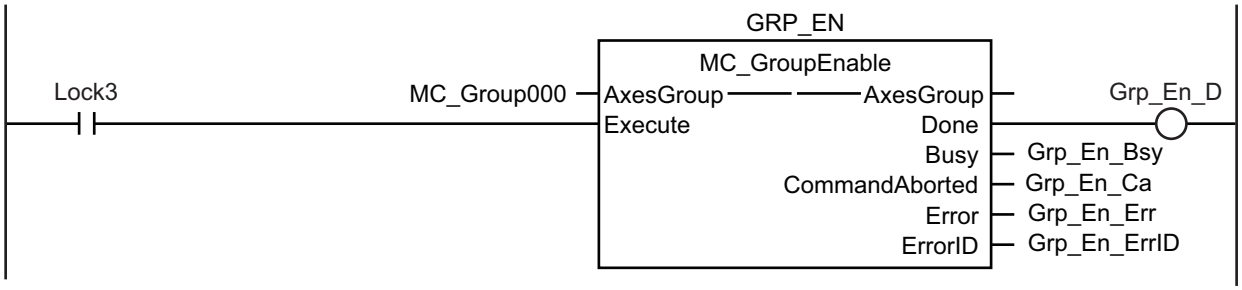
轴 1 为伺服 ON 状态且处于原点未确定状态时，执行原点复位



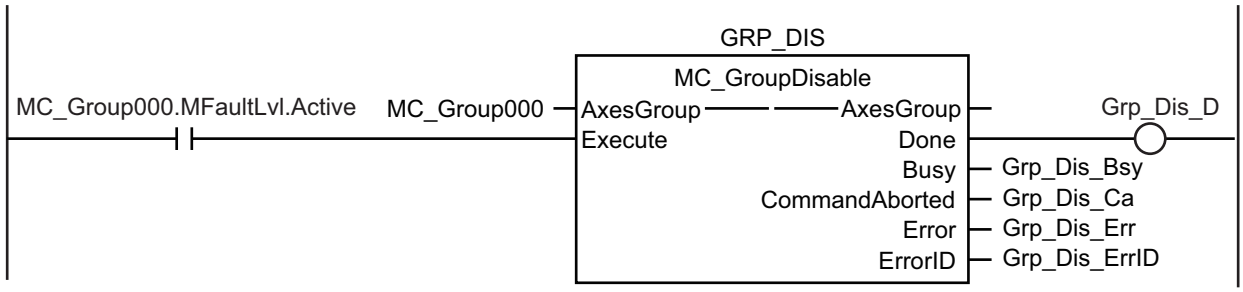
确认轴组的无效状态及轴 0 和轴 1 的原点确定



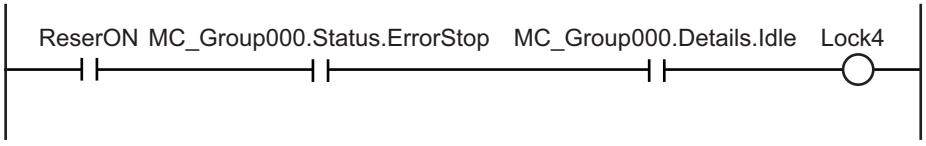
若轴 0 和轴 1 为原点确定状态，将轴组设为有效



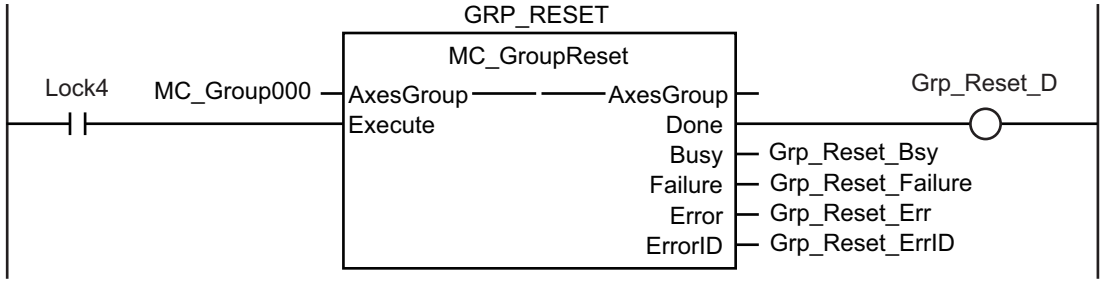
轴组发生轻度故障等级的异常时，将轴组设为无效



确认打开外部按钮后的 ResetON 状态和轴组的停止



若 ResetON 为 TRUE、轴组处于停止状态，则解除异常



ST 程序

```
// 确认 StartPg 为 TRUE 且过程数据通信正在正常执行，
// 若与轴 0 的过程数据通信正在正常执行，则轴 0 变为伺服 ON
// 若未执行过程数据通信，则轴 0 变为伺服 OFF
IF (StartPg =TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
   Pwr1_En :=TRUE;   // 轴 0 变为伺服 ON
ELSE
   Pwr1_En :=FALSE;   // 轴 0 变为伺服 OFF
END_IF;


// 确认 StartPg 为 TRUE 且过程数据通信正在正常执行，
// 若与轴 1 的过程数据通信正在正常执行，则轴 1 变为伺服 ON
// 若未执行过程数据通信，则轴 1 变为伺服 OFF
IF (StartPg =TRUE)
 AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
   Pwr2_En :=TRUE;   // 轴 1 变为伺服 ON
ELSE
   Pwr2_En :=FALSE;   // 轴 1 变为伺服 OFF
END_IF;


// 轴组或构成轴发生轻度故障等级的异常时，
// 执行异常处理 FaultHandler
IF (MC_Axis000.MFaultLvl.Active=TRUE)
 OR (MC_Axis001.MFaultLvl.Active=TRUE)
 OR (MC_Group000.MFaultLvl.Active=TRUE) THEN
   FaultHandler();   // 异常处理的内容请根据装置进行编程
END_IF;


// 轴 0 为伺服 ON 状态且处于原点未确定状态时，执行原点复位
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
   Hm1_Ex := TRUE;
END_IF;


// 轴 1 为伺服 ON 状态且处于原点未确定状态时，执行原点复位
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
   Hm2_Ex := TRUE;
END_IF;


// 若轴组为无效，轴 0 和轴 1 为原点确定状态，将轴组设为有效
IF (MC_Group000.Status.Disabled=TRUE) AND (Hm1_D=TRUE) AND (Hm2_D=TRUE) THEN
   Grp_En_Ex := TRUE;
END_IF;


// 轴组发生轻度故障等级的异常时，将轴组设为无效
IF MC_Group000.MFaultLvl.Active=TRUE THEN
   Grp_Dis_Ex :=TRUE;
END_IF;
```

```
// 若因外部按钮打开而使 ResetON 变为 TRUE、轴组处于停止状态，则解除异常
IF (ResetON=TRUE)
 AND (MC_Group000.Status.ErrorStop=TRUE)
 AND (MC_Group000.Details.Idle=TRUE) THEN
   Grp_Reset_Ex := TRUE;
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_GroupEnable
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
);
```

```
//MC_GroupDisable
GRP_DIS(
   AxesGroup := MC_Group000,
   Execute := Grp_Dis_Ex,
   Done => Grp_Dis_D,
   Busy => Grp_Dis_Bsy,
   CommandAborted => Grp_Dis_Ca,
   Error => Grp_Dis_Err,
   ErrorID => Grp_Dis_ErrID
);

//MC_GroupReset
GRP_RESET(
   AxesGroup := MC_Group000,
   Execute := Grp_Reset_Ex,
   Done => Grp_Reset_D,
   Busy => Grp_Reset_Bsy,
   Failure => Grp_Reset_Fai,
   Error => Grp_Reset_Err,
   ErrorID => Grp_Reset_ErrID
);
```

10-2-5 指令异常的监视方法

若在启动可运行 （MC_Power）指令时发生异常，将不执行以下处理。
在本示例程序中，以下处理的许可以变量 “UpgOn” 表示。

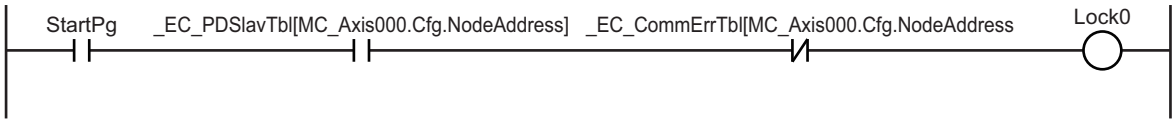
记载梯形图程序的示例和 ST 程序的示例。

程序中使用的主要变量

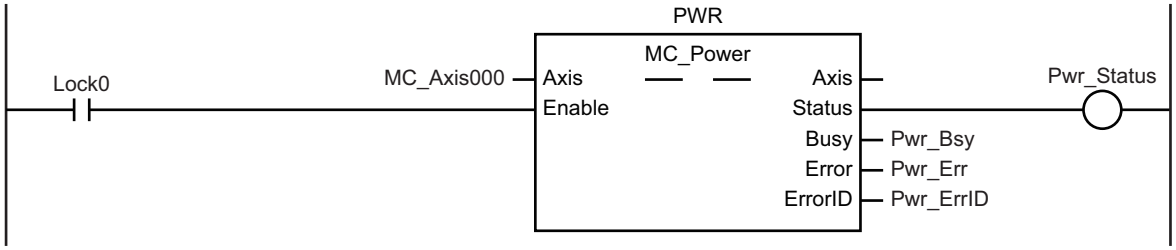
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
Pwr_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。
Pwr_Bsy	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR 的输出 Busy 中的变量。 伺服 ON 时变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 通信的过程数据通信正在正常执行，则伺服 ON。
UpgOn	BOOL	FALSE	TRUE 时，执行用户程序后面的处理。

梯形图程序

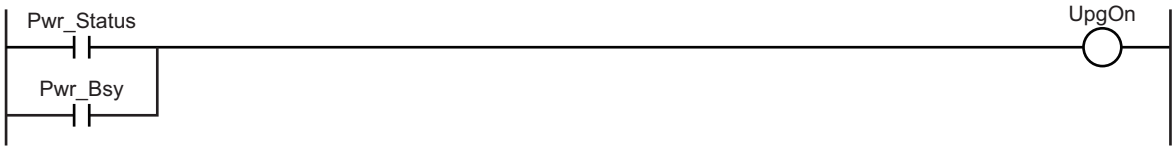
确认接点 StartPg 是否为 TRUE 且过程数据通信是否正在正常执行



若过程数据通信正在正常执行，则轴 0 变为伺服 ON



确认 MC_Power 中没有发生异常，再执行以下处理



ST 程序

```
// 若接点 StartPg 为 TRUE 且过程数据通信正在正常执行，则轴 0 变为伺服 ON
// 若未进行过程数据通信，则变为伺服 OFF
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
   Pwr_En :=TRUE;
ELSE
   Pwr_En :=FALSE;
END_IF;

IF (Pwr_Status=TRUE) OR (Pwr_Bsy=TRUE) THEN
   UpgOn := TRUE;   // 执行后面的处理
ELSE
   UpgOn := FALSE;   // 不执行后面的处理
END_IF;

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

10-2-6 异常解除的监视方法

在发生轻度故障等级异常的状态下，若打开外部按钮，将启动轴错误复位 (MC_Reset) 指令。直至轴错误复位 (MC_Reset) 指令的输出 “Done” 变为 TRUE，才会执行下一个正常处理。
输出 “Failure” 变为 TRUE 后，轴处于减速停止中，或发生 MC 共通异常。获取轴错误复位 (MC_Reset) 指令 “Failure” 变为 TRUE 的原因。

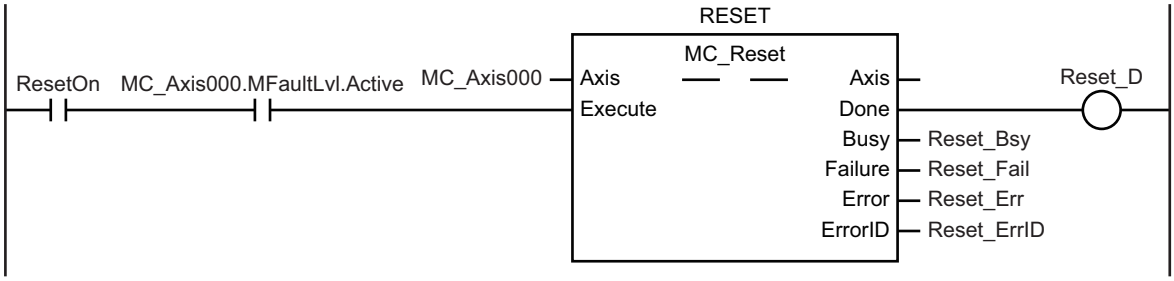
记载梯形图程序的示例和 ST 程序的示例。

程序中使用的主要变量

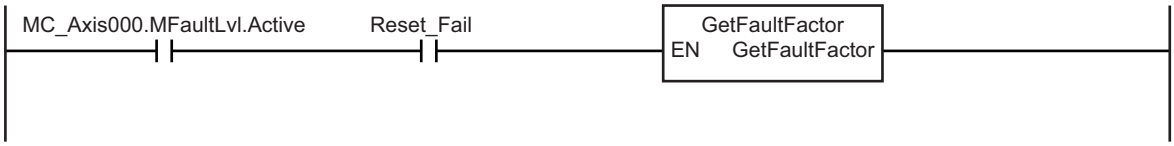
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时，变为 TRUE。
Pwr_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。
ResetON	BOOL	FALSE	表示解除异常所需的外部按钮的状态。
GetFaultFactor	—	—	获取异常原因。 请根据装置进行编程。
RegularProcess	—	—	正常处理。 请根据装置进行编程。

梯形图程序

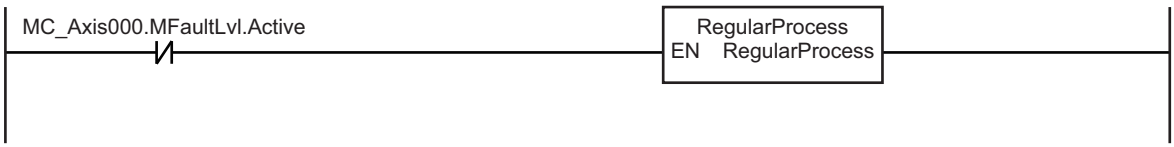
在发生轻度故障等级异常的状态下，若打开外部按钮 (ResetOn)，
将执行轴错误复位 (MC_Reset) 指令。



若轴错误复位 (MC_Reset) 指令的输出变量 “Failure” 为 TRUE，
则执行装置异常原因的获取处理 (GetFaultFactor)。处理内容请根据装置进行编程



未发生轻度故障等级的异常或已解除轻度故障等级的异常时，
执行装置的正常处理 (RegularProcess)。
处理内容请根据装置进行编程



ST 程序

```
// 在发生轻度故障等级异常的状态下，若打开外部按钮，
// 将执行轴错误复位 (MC_Reset) 指令
IF (MC_Axis000.MFaultLvl.Active=TRUE) AND (ResetOn=TRUE) THEN
    Reset_Ex := TRUE;    // 轻度故障的解除
ELSE
    Reset_Ex := FALSE;
END_IF;


// 若轴错误复位 (MC_Reset) 指令的输出变量 “Failure” 为 TRUE，
// 则执行异常原因的获取处理 (GetFaultFactor)
// 处理内容请根据装置进行编程
IF (MC_Axis000.MFaultLvl.Active=TRUE) AND (Reset_Fail=TRUE) THEN
    GetFaultFactor();
END_IF;


// 未发生轻度故障等级的异常或已解除轻度故障等级的异常时，
// 执行装置的正常处理 (RegularProcess)
// 处理内容请根据装置进行编程
IF MC_Axis000.MFaultLvl.Active=FALSE THEN
    RegularProcess();
END_IF;

// MC_Reset
RESET(
    Axis := MC_Axis000,
    Execute := Reset_Ex,
    Done => Reset_D,
    Busy => Reset_Bsy,
    Failure => Reset_Fail,
    Error => Reset_Err,
    ErrorID => Reset_ErrID
);
```

10-2-7 单轴动作时的轴停止

若在执行绝对值定位 (MC_MoveAbsolute) 指令时打开外部按钮，将执行强制停止 (MC_Stop) 指令并减速停止。发生轻度故障等级的异常时，强制停止 (MC_Stop) 指令的输出 “CommandAborted” 将变为 TRUE。
此时，执行立即停止 (MC_ImmediateStop) 指令立即停止。
此外，即使因某些原因导致强制停止 (MC_Stop) 指令的输出 “Error” 变为 TRUE，也要执行立即停止 (MC_ImmediateStop) 指令立即停止。执行立即停止 (MC_ImmediateStop) 指令后，轴将变为错误减速停止中。

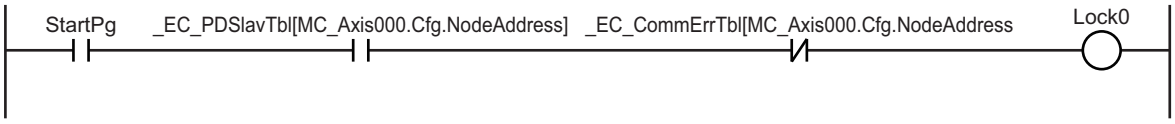
记载梯形图程序的示例和 ST 程序的示例。

程序中使用的主要变量

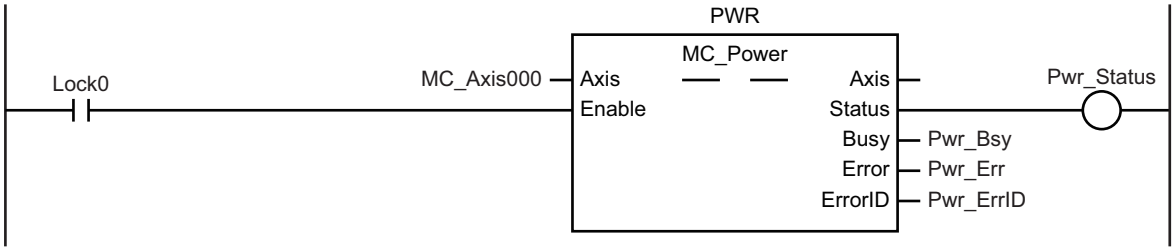
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 0 处于原点确定状态时变为 TRUE。
Pwr_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。
Stp_Ca	BOOL	FALSE	分配到 MC_Stop 的实例 STP 的输出变量 CommandAborted 中的变量。
Stp_Err	BOOL	FALSE	分配到 MC_Stop 的实例 STP 的输出变量 Error 中的变量。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 通信的过程数据通信正在正常执行，则伺服 ON。
StopOn	BOOL	FALSE	执行强制停止所需的外部按钮的状态 TRUE 时，通过 MC_Stop 强制停止。

梯形图程序

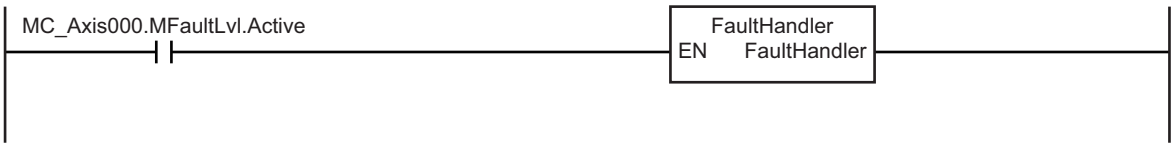
确认接点 StartPg 是否为 TRUE 且轴 0 的过程数据通信是否正在正常执行



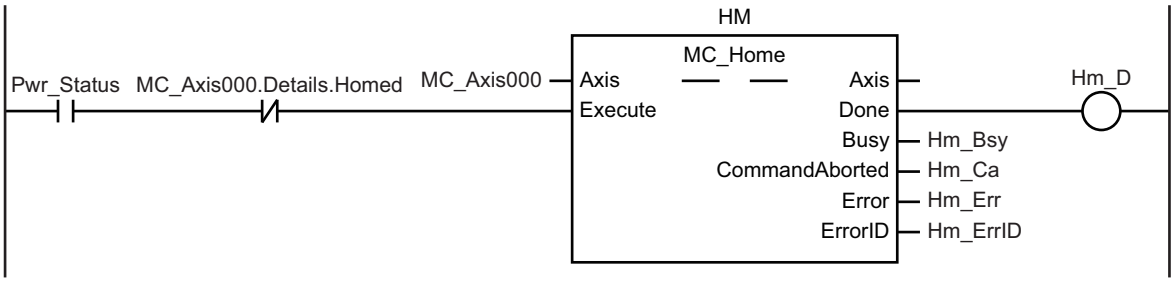
若轴 0 的过程数据通信正在正常执行，则轴 0 变为伺服 ON



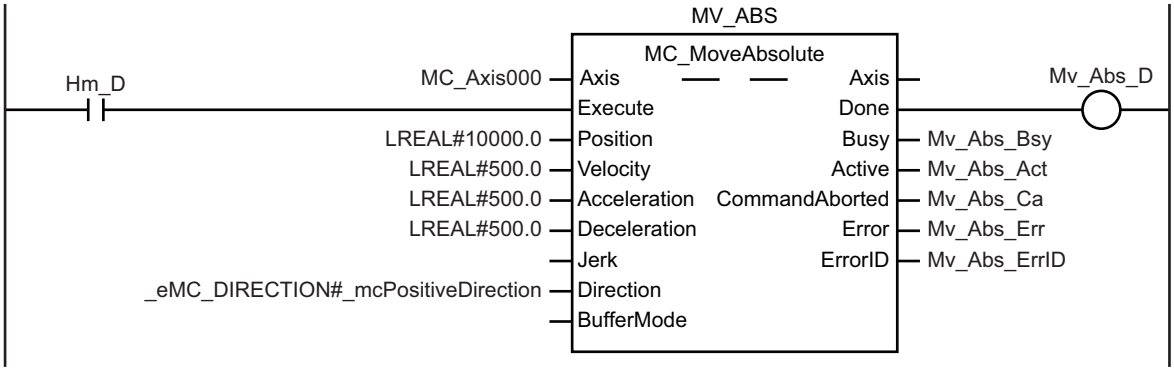
轴 0 发生轻度故障等级的异常时，执行异常处理 FaultHandler 异常处理的内容请根据装置进行编程



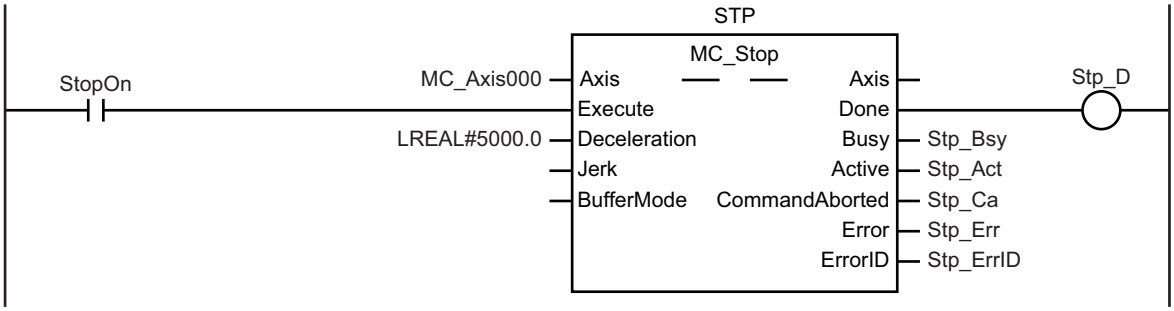
轴 0 为伺服 ON 状态且处于原点未确定状态时，执行原点复位



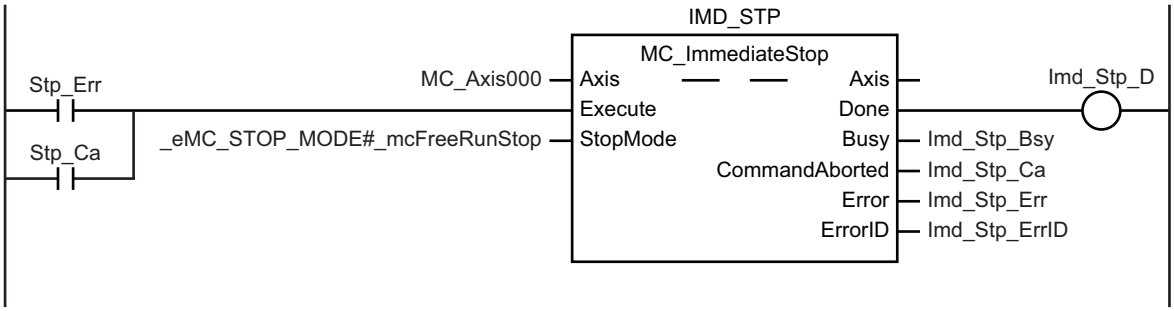
轴 0 的原点复位完成后，执行绝对值定位 (MC_MoveAbsolute) 指令



StopOn 为 TRUE 时，执行强制停止 (MC_Stop) 指令



强制停止 (MC_Stop) 指令的输出“Error”或“CommandAborted”为 TRUE 时，
执行立即停止 (MC_ImmediateStop) 指令，立即停止



ST 程序

```
// 未设定绝对值定位和强制停止的输入参数时，设定目标值等
IF InitFlag=FALSE THEN
   // 在绝对值定位 (MC_MoveAbsolute) 的输入参数中设定值
   Mv_Abs_Pos := LREAL#10000.0;
   Mv_Abs_Vel := LREAL#500.0;
   Mv_Abs_Acc := LREAL#500.0;
   Mv_Abs_Dec := LREAL#500.0;
   Mv_Abs_Dir := _eMC_DIRECTION#_mcPositiveDirection;

   // 在强制停止 (MC_Stop) 的输入参数中设定值
   Stp_Dec:=LREAL#5000.0;

   // 在立即停止 (MC_ImmediateStop) 的输入参数中设定值
   Imd_Stp_SM :=_eMC_STOP_MODE#_mcFreeRunStop;

   // 将输入参数设定完成标志设为 TRUE
   InitFlag := TRUE;
END_IF;


// 若接点 StartPg 为 TRUE 且过程数据通信正在正常执行，则轴 0 变为伺服 ON
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
   Pwr_En :=TRUE;
ELSE
   Pwr_En :=FALSE;
END_IF;


// 轴 0 发生轻度故障等级的异常时，执行异常处理 FaultHandler
// 异常处理的内容请根据装置进行编程
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
   FaultHandler();
END_IF;


// 轴 0 为伺服 ON 状态且处于原点未确定状态时，执行原点复位
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
   Hm_Ex :=TRUE;
END_IF;


// 原点复位完成后，执行绝对值定位
IF Hm_D=TRUE THEN
   Mv_Abs_Ex := TRUE;
END_IF;


// 若 StopOn 为 TRUE，则执行强制停止
IF StopOn=TRUE THEN
   Stp_Ex :=TRUE;
END_IF;
```

```
// 强制停止 (MC_Stop) 的输出 “Error” 或 “CommandAborted” 为 TRUE 时，执行立即停止 (MC_ImmediateStop)，
立即停止
IF (Stp_Err=TRUE) OR (Stp_Ca=TRUE) THEN
   Imd_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_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_ABS(
   Axis := MC_Axis000,
   Execute := Mv_Abs_Ex,
   Position := Mv_Abs_Pos,
   Velocity := Mv_Abs_Vel,
   Acceleration := Mv_Abs_Acc,
   Deceleration := Mv_Abs_Dec,
   Direction := Mv_Abs_Dir,
   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_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_ImmediateStop
IMD_STP(
```

```
    Axis := MC_Axis000,
    Execute := Imd_Stp_Ex,
    StopMode := Imd_Stp_SM,
    Done => Imd_Stp_D,
    Busy => Imd_Stp_Bsy,
    CommandAborted => Imd_Stp_Ca,
    Error => Imd_Stp_Err,
    ErrorID => Imd_Stp_ErrID
);
```

10-2-8 多轴协调动作中的轴组停止

若在执行绝对值直线插补 (MC_MoveLinearAbsolute) 指令时打开外部按钮，将执行强制停止 (MC_GroupStop) 指令并减速停止。发生轻度故障等级的异常时，轴组强制停止 (MC_GroupStop) 的输出 “CommandAborted” 将变为 TRUE。此时，执行轴组立即停止 (MC_GroupImmediateStop) 指令立即停止。

此外，即使因某些原因导致轴组强制停止 (MC_GroupStop) 的输出 “Error” 变为 TRUE，也要执行轴组立即停止 (MC_GroupImmediateStop) 指令立即停止。执行轴组立即停止 (MC_GroupImmediateStop) 指令后，轴组将变为错误减速停止中。

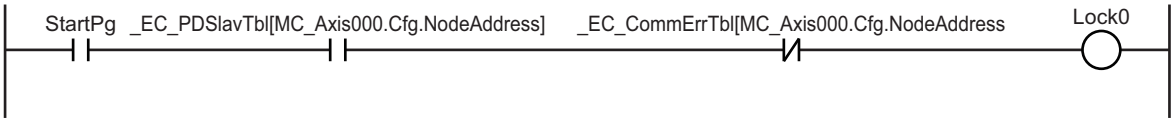
程序中使用的主要变量

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.Status.Disabled	BOOL	FALSE	轴组 0 为无效时变为 TRUE。
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_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 0 处于原点确定状态时变为 TRUE。
MC_Axis001	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis001.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
Pwr1_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR1 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。
Pwr2_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR2 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。
Grp_Stp_Ca	BOOL	FALSE	分配到 MC_GroupStop 的实例 GRP_EN 的输出变量 CommandAborted 中的变量。
Grp_Stp_Err	BOOL	FALSE	分配到 MC_GroupStop 的实例 GRP_EN 的输出变量 Error 中的变量。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 通信的过程数据通信正在正常执行，则伺服 ON。
StopOn	BOOL	FALSE	执行强制停止所需的外部按钮的状态 TRUE 时，通过 MC_GroupStop 强制停止。
InitFlag	BOOL	FALSE	表示绝对值直线插补指令和轴组强制停止的输入参数设定完成的变量。

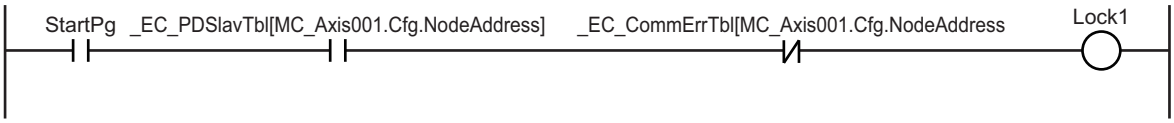
名称	数据类型	初始值	注释
Grp_En_Ex	BOOL	FALSE	启动 MC_GroupEnable 的实例 GRP_EN 所需的变量。 在 ST 程序中使用。
Mv_Lin_Abs_Ex	BOOL	FALSE	启动 MC_MoveLinear 的实例 MV_LIN_ABS 所需的变量。 在 ST 程序中使用。
Grp_Stp_Ex	BOOL	FALSE	启动 MC_GroupStop 的实例 GRP_STP 所需的变量。 在 ST 程序中使用。
Grp_Imd_Stp_Ex	BOOL	FALSE	启动 MC_GroupImmediateStop 的实例 GRP_IMD_STP 所需的变量。 在 ST 程序中使用。

梯形图程序

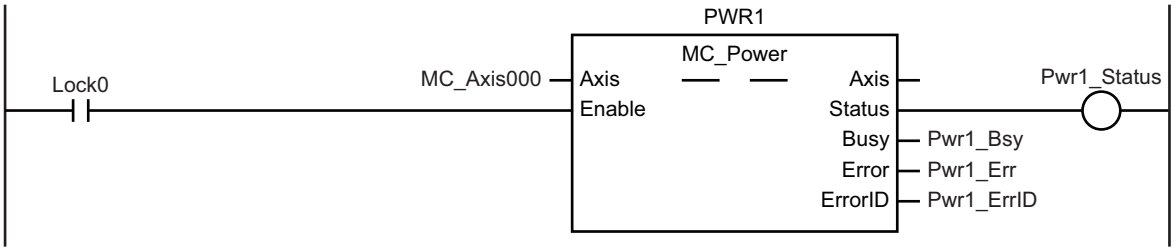
确认接点 StartPg 是否为 TRUE 且与轴 0 的过程数据通信是否正在正常执行



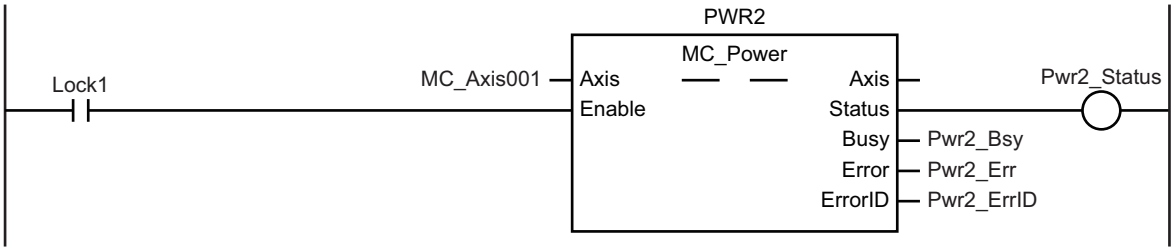
确认接点 StartPg 是否为 TRUE 且与轴 1 的过程数据通信是否正在正常执行



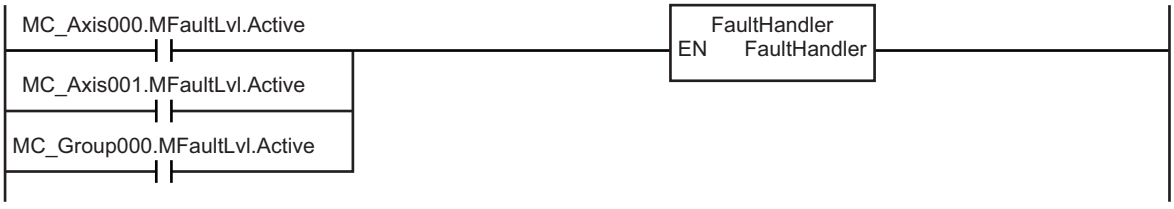
若与轴 0 的过程数据通信正在正常执行，则轴 0 变为伺服 ON



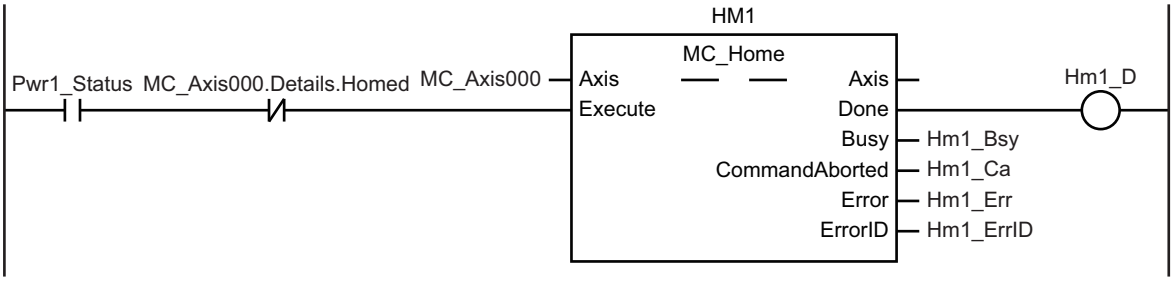
若与轴 1 的过程数据通信正在正常执行，则轴 1 变为伺服 ON



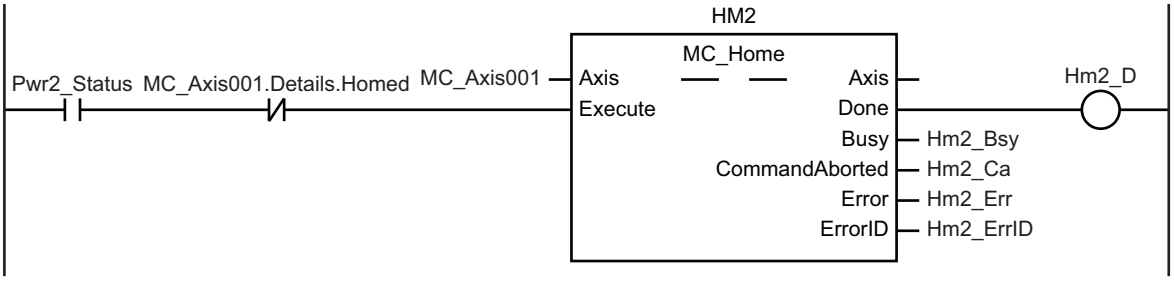
轴组或构成轴发生轻度故障等级的异常时，执行异常处理 FaultHandler。
异常处理的内容请根据装置进行编程



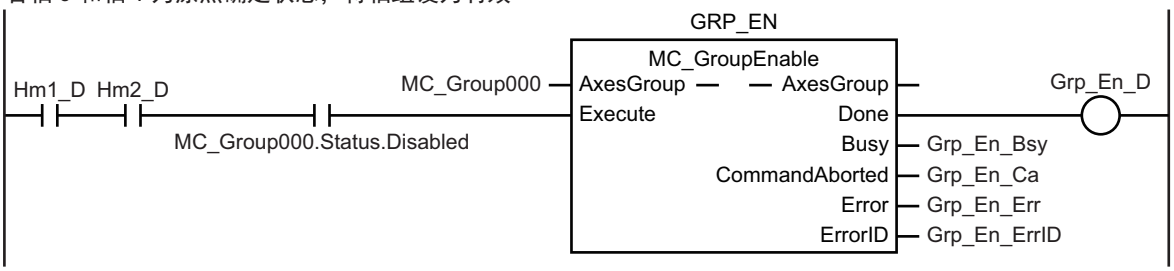
轴 0 为伺服 ON 状态且处于原点未确定状态时，执行原点复位



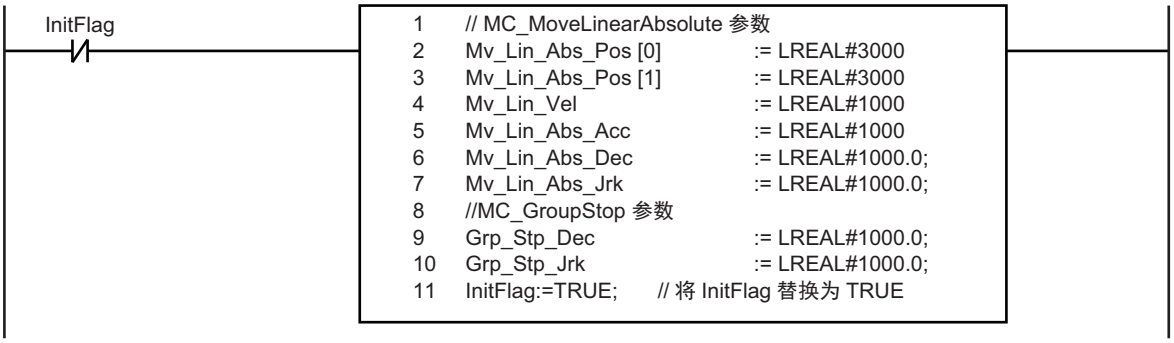
轴 1 为伺服 ON 状态且处于原点未确定状态时，执行原点复位



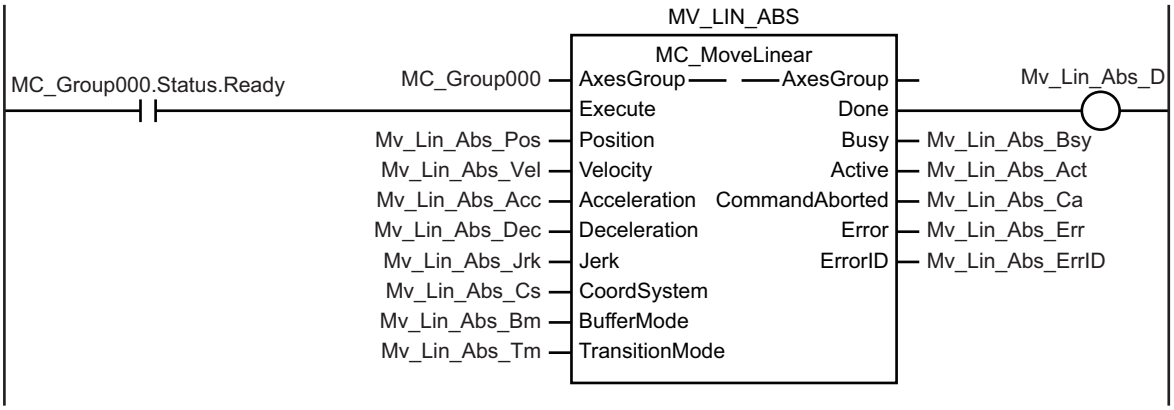
若轴 0 和轴 1 为原点确定状态，将轴组设为有效



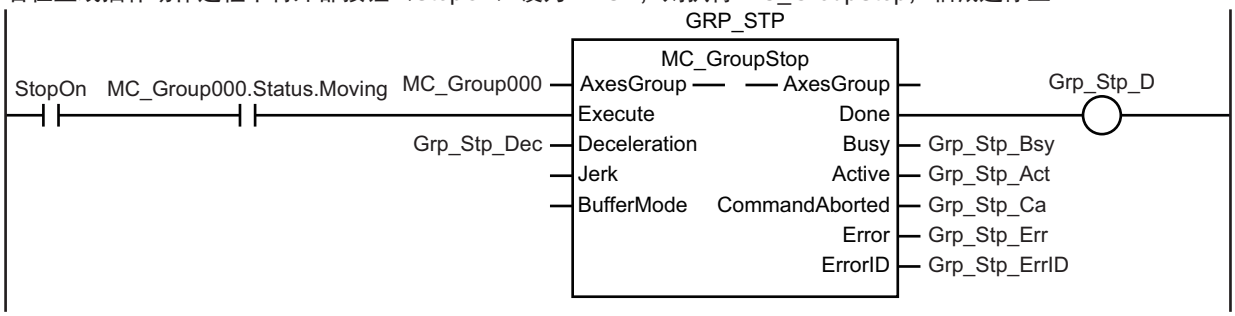
设定绝对值直线插补和轴组强制停止的输入参数



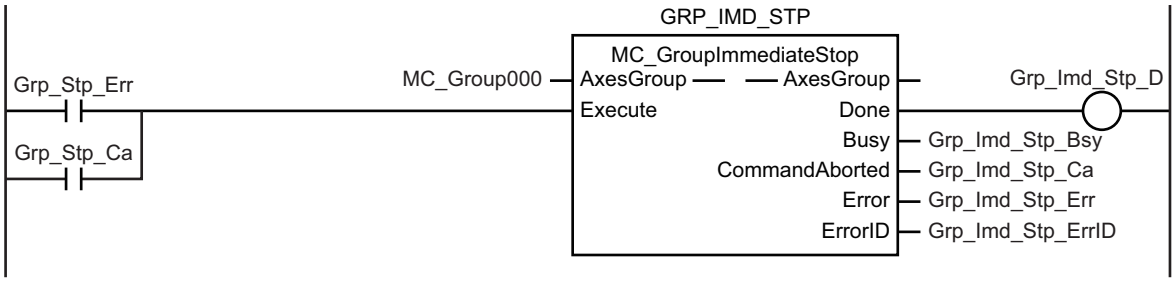
确认轴组变为有效，执行直线插补



若在直线插补动作过程中将外部按钮（StopOn）设为 TRUE，则执行 MC_GroupStop，轴减速停止



MC_GroupStop 的 Error 或 CommandAborted 为 TRUE 时，执行 MC_GroupImmediateStop，立即停止



ST 程序

```
// 未设定绝对值直线插补和轴组强制停止的输入参数时，设定目标值等
IF InitFlag=FALSE THEN

   // 在绝对值直线插补 (MC_MoveLinearAbsolute) 的输入参数中设定值
   Mv_Lin_Abs_Pos[0] := LREAL#3000.0;
   Mv_Lin_Abs_Pos[1] := LREAL#3000.0;
   Mv_Lin_Abs_Vel := LREAL#1000.0;
   Mv_Lin_Abs_Acc := LREAL#1000.0;
   Mv_Lin_Abs_Dec := LREAL#1000.0;
   Mv_Lin_Abs_Jrk := LREAL#1000.0;

   // 在轴组强制停止 (MC_GroupStop) 的输入参数中设定值
   Grp_Stp_Dec := LREAL#1000.0;
   Grp_Stp_Jrk := LREAL#1000.0;

   // 将输入参数设定完成标志设为 TRUE
   InitFlag := TRUE;
END_IF;


// 若 StartPg 为 TRUE 且过程数据通信正在正常执行，则轴 0 变为伺服 ON
// 若未执行过程数据通信，则变为伺服 OFF
IF (StartPg =TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
   Pwr1_En :=TRUE;   // 轴 0 变为伺服 ON
ELSE
   Pwr1_En :=FALSE;   // 轴 0 变为伺服 OFF
END_IF;


// 若 StartPg 为 TRUE 且过程数据通信正在正常执行，则轴 1 变为伺服 ON
// 若未执行过程数据通信，则变为伺服 OFF
IF (StartPg =TRUE) AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE) THEN
```

```
    Pwr2_En :=TRUE;    // 轴 1 变为伺服 ON
ELSE
    Pwr2_En :=FALSE;    // 轴 1 变为伺服 OFF
END_IF;


// 发生轻度故障等级的异常时，执行异常处理 FaultHandler
// 异常处理的内容请根据装置进行编程
IF (MC_Axis000.MFaultLvl.Active=TRUE)
 OR (MC_Axis001.MFaultLvl.Active=TRUE)
 OR (MC_Group000.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;


// 轴 0 为伺服 ON 状态且处于原点未确定状态时，执行轴 0 的原点复位
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex :=TRUE;
END_IF;


// 轴 1 为伺服 ON 状态且处于原点未确定状态时，执行轴 1 的原点复位
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex :=TRUE;
END_IF;


// 若轴 0 和轴 1 为原点确定状态时轴组为无效，则将轴组设为有效
IF (MC_Group000.Status.Disabled=TRUE) AND (Hm1_D=TRUE) AND (Hm2_D=TRUE) THEN
    Grp_En_Ex := TRUE;
END_IF;


// 轴组为有效时，执行绝对值直线插补
IF MC_Group000.Status.Ready=TRUE THEN
    Mv_Lin_Abs_Ex :=TRUE;
END_IF;


// 正在执行绝对值直线插补时，若因外部按钮而使 StopOn 变为 TRUE，则执行强制停止
IF (MC_Group000.Status.Moving=TRUE) AND (StopOn=TRUE) THEN
    Grp_Stp_Ex := TRUE;
END_IF;


// 轴组强制停止变为 CommandAborted 或 Error 时，立即停止轴组
IF (Grp_Stp_Ca=TRUE) OR (Grp_Stp_Err=TRUE) THEN
    Grp_Imd_Stp_Ex :=TRUE;
END_IF;

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

```
PWR2(
    Axis := MC_Axis001,
    Enable := Pwr2_En,
    Status => Pwr2_Status,
    Busy => Pwr2_Bsy,
    Error => Pwr2_Err,
    ErrorID => Pwr2_ErrID
);

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

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

//MC_GroupEnable
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
);

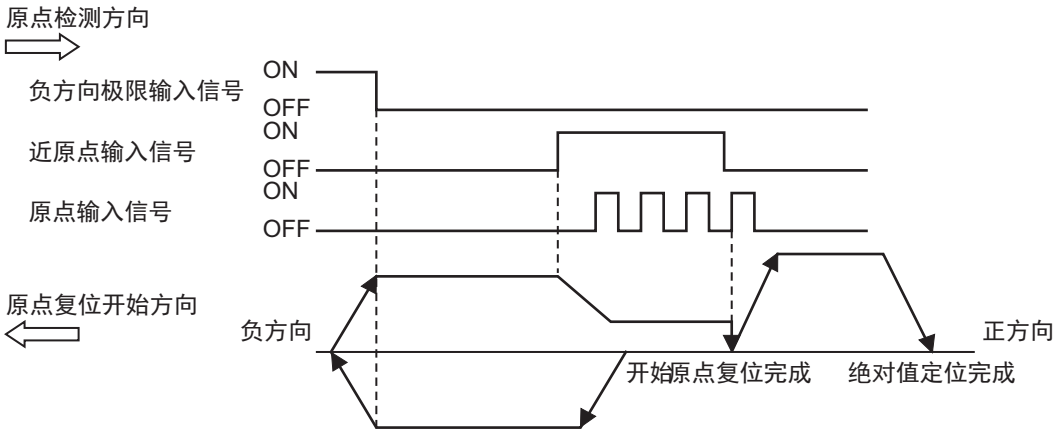
//MC_MoveLinearAbsolute
MV_LIN_ABS(
    AxesGroup := MC_Group000,
    Execute := Mv_Lin_Abs_Ex,
    Position := Mv_Lin_Abs_Pos,
    Velocity := Mv_Lin_Abs_Vel,
    Acceleration := Mv_Lin_Abs_Acc,
    Deceleration := Mv_Lin_Abs_Dec,
    Jerk := Mv_Lin_Abs_Jrk,
    Done => Mv_Lin_Abs_D,
    Busy => Mv_Lin_Abs_Bsy,
    Active => Mv_Lin_Abs_Act,
    CommandAborted => Mv_Lin_Abs_Ca,
    Error => Mv_Lin_Abs_Err,
    ErrorID => Mv_Lin_Abs_ErrID
);
```

```
//MC_GroupStop
GRP_STP(
   AxesGroup := MC_Group000,
   Execute := Grp_Stp_Ex,
   Deceleration := Grp_Stp_Dec,
   Done => Grp_Stp_D,
   Busy => Grp_Stp_Bsy,
   Active => Grp_Stp_Act,
   CommandAborted => Grp_Stp_Ca,
   Error => Grp_Stp_Err,
   ErrorID => Grp_Stp_ErrID
);

//MC_GroupImmediateStop
GRP_IMD_STP(
   AxesGroup := MC_Group000,
   Execute := Grp_Imd_Stp_Ex,
   Done => Grp_Imd_Stp_D,
   Busy => Grp_Imd_Stp_Bsy,
   CommandAborted => Grp_Imd_Stp_Ca,
   Error => Grp_Imd_Stp_Err,
   ErrorID => Grp_Imd_Stp_ErrID
);
```

10-2-9 原点复位和绝对值定位

原点复位开始位置为近原点输入 ON 时。原点复位以 “4: 近原点输入 OFF 指定 ” 执行。若原点复位正常完成且已确定原点，则进行绝对值定位。



记载梯形图程序的示例和 ST 程序的示例。

原点复位相关的轴参数设定

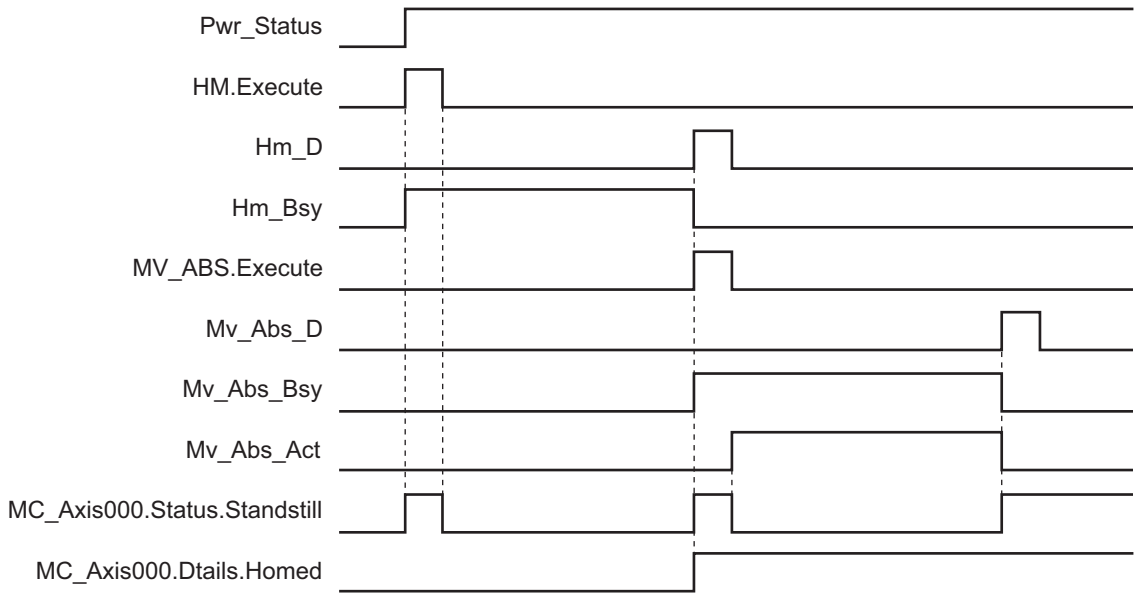
设定项目	设定内容	说明
原点复位动作	4: 近原点输入 OFF 指定	以近原点输入 OFF 指定进行原点确定。
正方向极限输入时动作	1: 反转、立即停止	不使用正方向极限输入，因此保持初始值。
负方向极限输入时动作	2: 反转、减速停止	若负方向极限输入变为有效，则减速停止并反转。
原点复位开始方向	2: 负方向	若启动原点复位，将向负方向开始动作。
原点检测方向	1: 正方向	检测原点时的动作方向为正方向。

程序中使用的主要变量

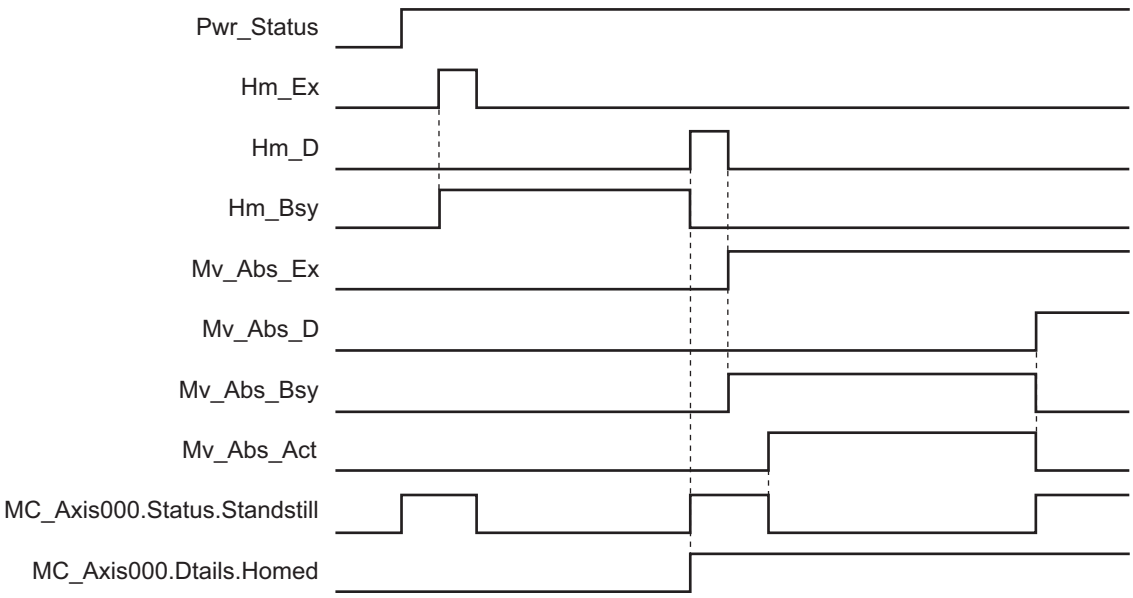
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
MC_Axis000.Status.StandStill	BOOL	FALSE	轴 0 为伺服 ON 且处于停止状态时，变为 TRUE。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 0 处于原点确定状态时变为 TRUE。
Pwr_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 通信的过程数据通信正在正常执行，则伺服 ON。
Hm_Ex	BOOL	FALSE	启动 MC_Home （原点复位）指令所需的变量。 在 ST 程序中使用。
Mv_Abs_Ex	BOOL	FALSE	启动 MC_MoveAbsolute （绝对值定位）指令所需的变量。 在 ST 程序中使用。

时序图

● 梯形图程序时

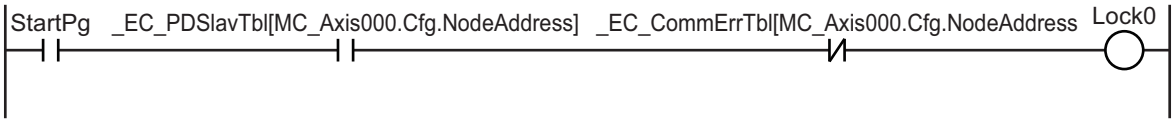


● **ST 程序时**

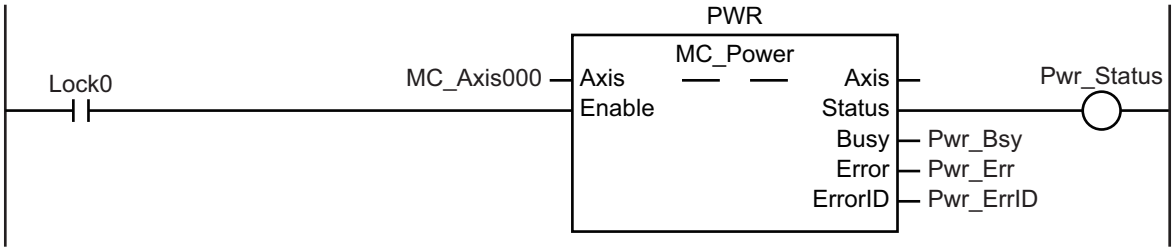


梯形图程序

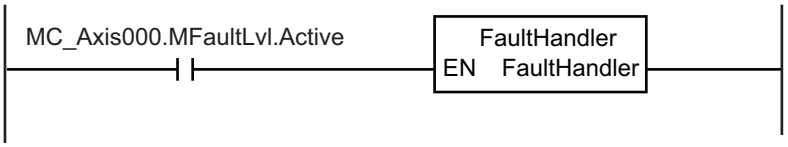
确认接点 StartPg 是否为 TRUE 且与轴 0 的过程数据通信是否正在正常执行



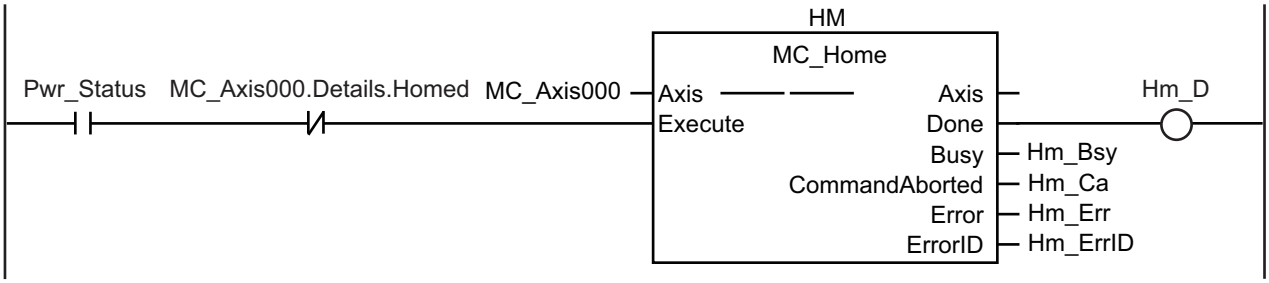
若与轴 0 的过程数据通信正在正常执行，则轴 0 变为伺服 ON



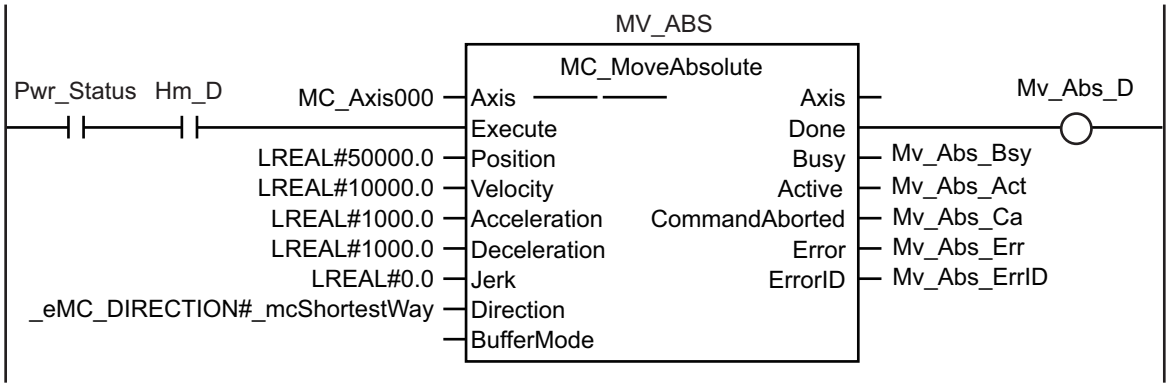
轴 0 发生轻度故障等级的异常时，执行异常处理 FaultHandler
异常处理的内容请根据装置进行编程



轴 0 为伺服 ON 状态且处于原点未确定状态时，执行原点复位



若为伺服 ON 状态且已确定原点，则执行绝对值定位



ST 程序

```
// 未设定绝对值定位的输入参数时，设定目标值等
IF InitFlag=FALSE THEN
   // 在绝对值定位 (MC_MoveAbsolute) 的输入参数中设定值
   Mv_Abs_Pos := LREAL#50000.0;
   Mv_Abs_Vel := LREAL#10000.0;
   Mv_Abs_Acc := LREAL#1000.0;
   Mv_Abs_Dec := LREAL#1000.0;
   Mv_Abs_Dir := _eMC_DIRECTION#_mcShortestWay;

   // 将输入参数设定完成标志设为有效
   InitFlag := TRUE;
END_IF;


// 若接点 StartPg 为 TRUE 且过程数据通信正在正常执行，则轴 0 变为伺服 ON
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
   Pwr_En :=TRUE;
ELSE
   Pwr_En :=FALSE;
END_IF;


// 轴 0 发生轻度故障等级的异常时，执行异常处理 FaultHandler
// 异常处理的内容请根据装置进行编程
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
   FaultHandler();
END_IF;


// 轴 0 为伺服 ON 状态且处于原点未确定状态时，执行原点复位
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
   Hm_Ex :=TRUE;
ELSE
   Hm_Ex :=FALSE;
END_IF;


// 若为伺服 ON 状态且已确定原点，则执行绝对值定位。
IF (Pwr_Status=TRUE) AND (Hm_D=TRUE) THEN
   Mv_Abs_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_ABS(
    Axis := MC_Axis000,
    Execute := Mv_Abs_Ex,
    Position := Mv_Abs_Pos,
    Velocity := Mv_Abs_Vel,
    Acceleration := Mv_Abs_Acc,
    Deceleration := Mv_Abs_Dec,
    Direction := Mv_Abs_Dir,
    Done => Mv_Abs_D,
    Busy => Mv_Abs_Bsy,
    Active => Mv_Abs_Act,
    CommandAborted => Mv_Abs_Ca,
    Error => Mv_Abs_Err,
    ErrorID => Mv_Abs_ErrID
);
```

10-2-10 通过重启指令来变更目标位置

正在向目标位置 1,000 执行绝对值定位动作时，用同一指令实例将目标位置变更为 2,000。

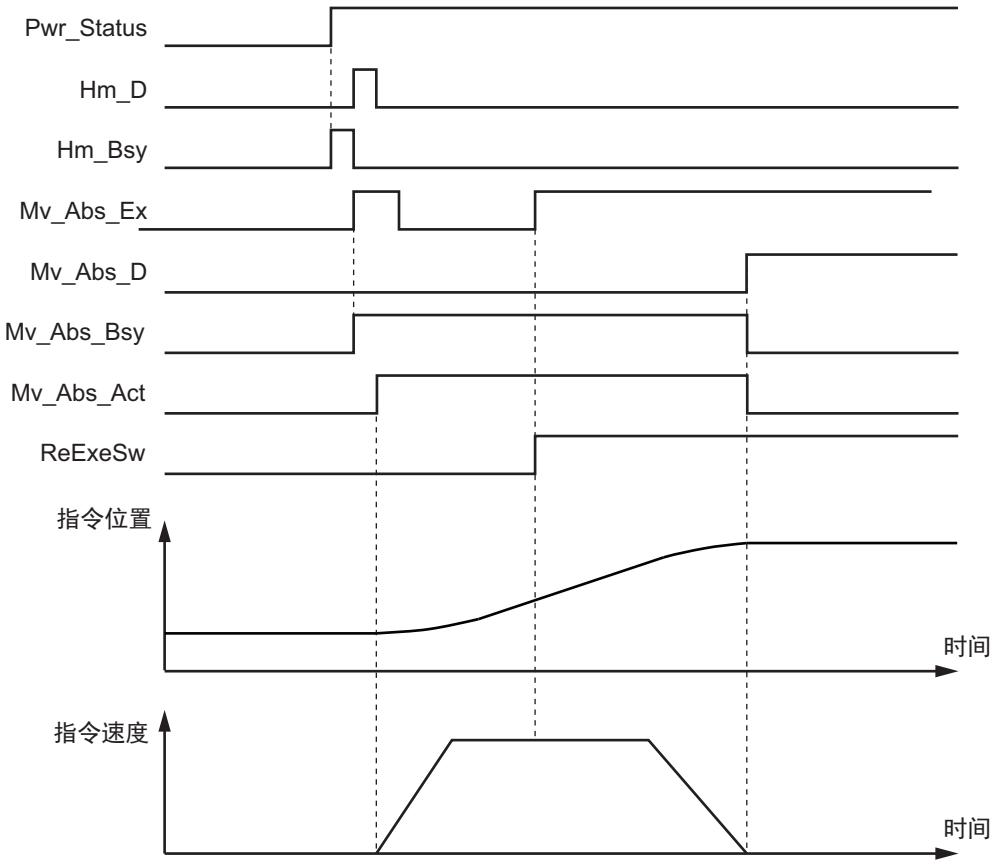
程序中使用的主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 0 处于原点确定状态时变为 TRUE。
Pwr_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。

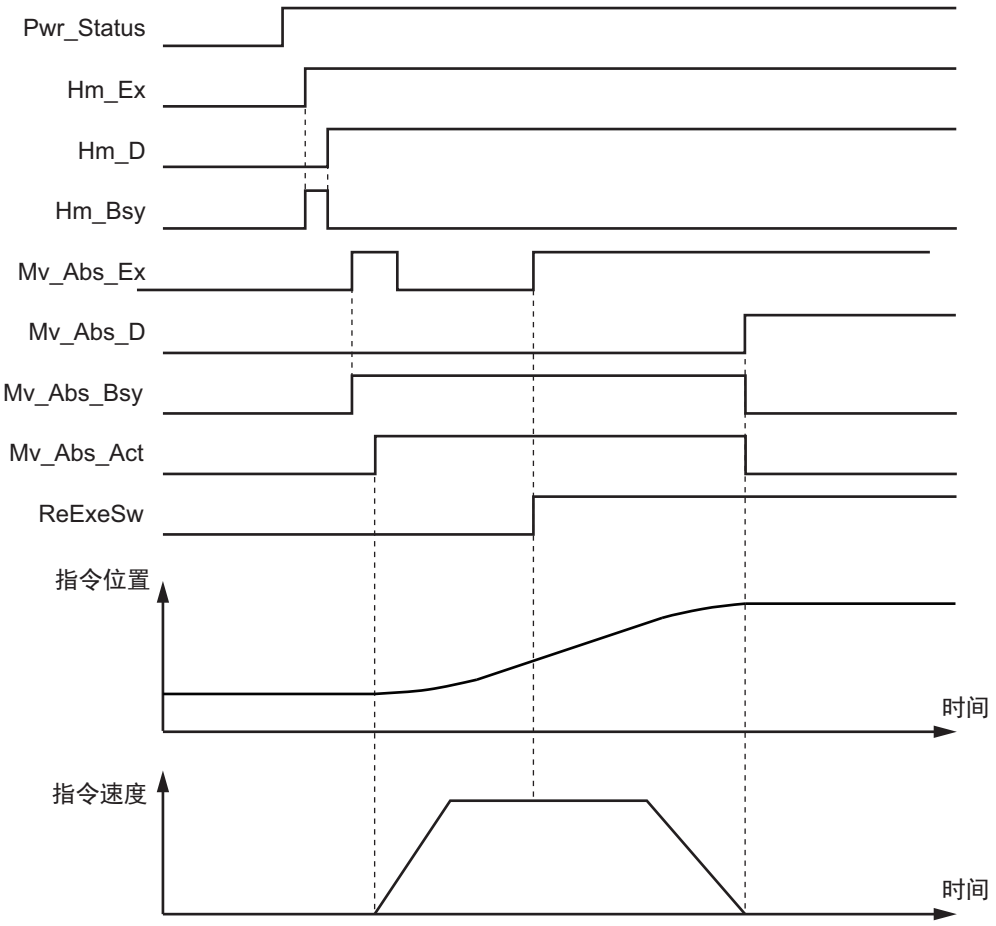
名称	数据类型	初始值	注释
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 通信的过程数据通信正在正常执行，则伺服 ON。
ReExeSw	BOOL	FALSE	重启指令所需的变量。
Hm_Ex	BOOL	FALSE	启动 MC_Home （原点复位）指令所需的变量。 在 ST 程序中使用。
Mv_Abs_Ex	BOOL	FALSE	启动 MC_MoveAbsolute （绝对值定位）指令所需的变量。 在 ST 程序中使用。

时序图

● 梯形图程序时

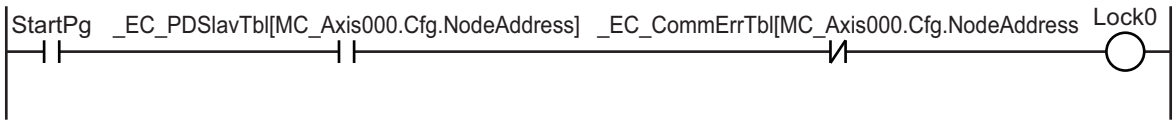


● **ST 程序时**

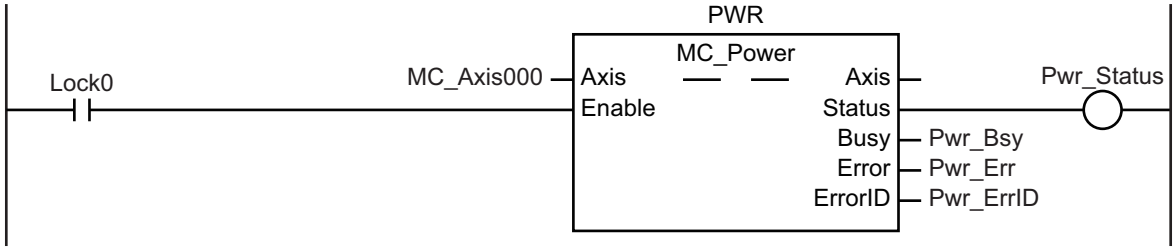


梯形图程序

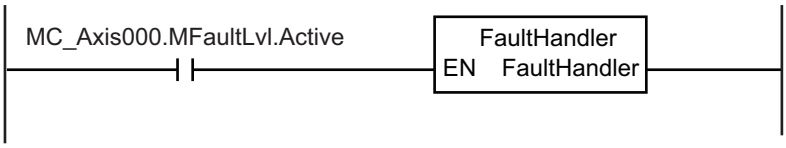
确认接点 StartPg 是否为 TRUE 且与轴 0 的过程数据通信是否正在正常执行



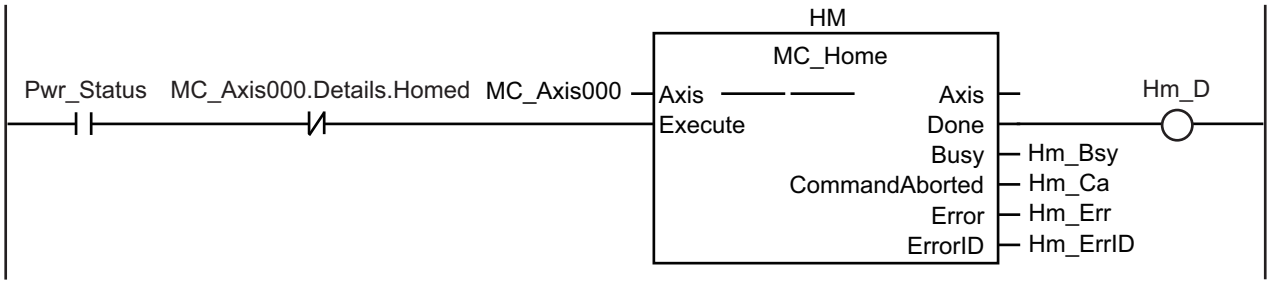
若与轴 0 的过程数据通信正在正常执行，则轴 0 变为伺服 ON



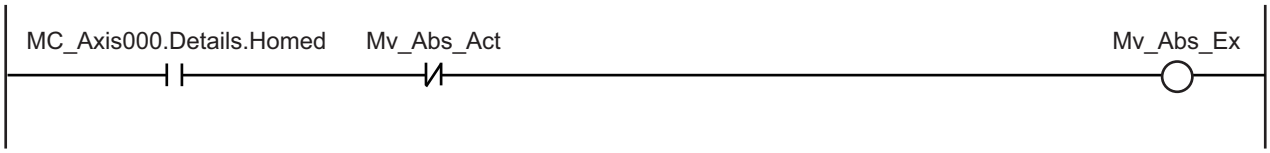
轴 0 发生轻度故障等级的异常时，执行异常处理 FaultHandler
异常处理的内容请根据装置进行编程



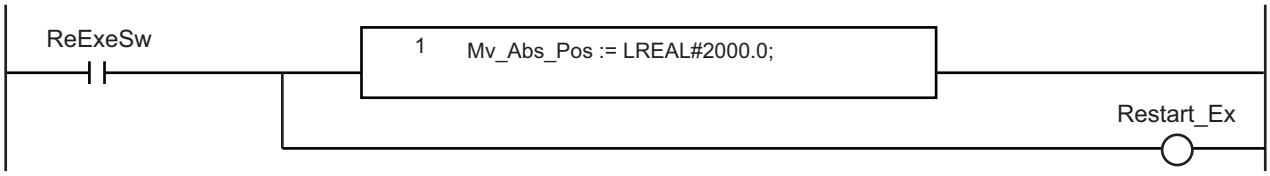
轴 0 为伺服 ON 状态且处于原点未确定状态时，执行原点复位



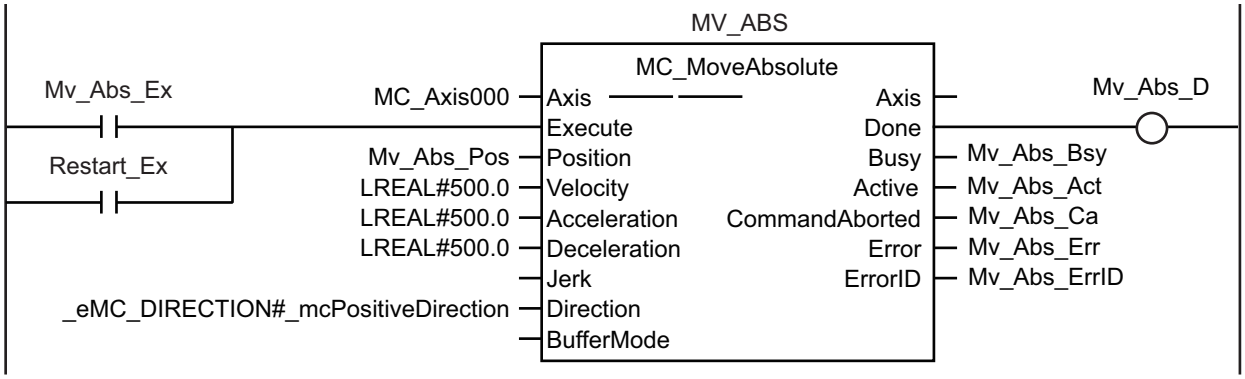
轴 0 原点确定后，若未执行绝对值定位，则执行



若将接点 ReExeSw 设为 TRUE，则将目标位置变更为 2000，然后重新执行绝对值定位



根据接点 Mv_Abs_Ex 的状态执行绝对值定位



ST 程序

```
// 未设定绝对值定位的输入参数时，设定目标值等
IF InitFlag = FALSE THEN
   //MoveAbsolute 参数
   Mv_Abs_Pos := LREAL#1000.0;
   Mv_Abs_Vel := LREAL#500.0;
   Mv_Abs_Acc := LREAL#500.0;
   Mv_Abs_Dec := LREAL#500.0;
   Mv_Abs_Dir := _eMC_DIRECTION#_mcPositiveDirection;

   // 将输入参数设定完成标志设为有效
   InitFlag :=TRUE;
END_IF;


// 若接点 StartPg 为 TRUE 且过程数据通信正在正常执行，则轴 0 变为伺服 ON
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
   Pwr_En :=TRUE;
ELSE
```

```
    Pwr_En :=FALSE;
END_IF;


// 轴 0 发生轻度故障等级的异常时，执行异常处理 FaultHandler
// 异常处理的内容请根据装置进行编程
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;


// 轴 0 为伺服 ON 状态且处于原点未确定状态时，执行原点复位
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm_Ex :=TRUE;
END_IF;


// 轴 0 完成原点复位后，若未执行绝对值定位，则执行
IF (Hm_D=TRUE) AND (Mv_Abs_Act=FALSE) THEN
    Mv_Abs_Ex := TRUE;
ELSE
    Mv_Abs_Ex := FALSE;
END_IF;


// 若将接点 ReExeSw 设为 TRUE，则将目标位置变更为 2000，然后重新执行绝对值定位
IF ReExeSw=TRUE THEN
    Mv_Abs_Pos :=LREAL#2000.0;
    Mv_Abs_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_ABS(
    Axis := MC_Axis000,
    Execute := Mv_Abs_Ex,
    Position := Mv_Abs_Pos,
    Velocity := Mv_Abs_Vel,
    Acceleration := Mv_Abs_Acc,
```

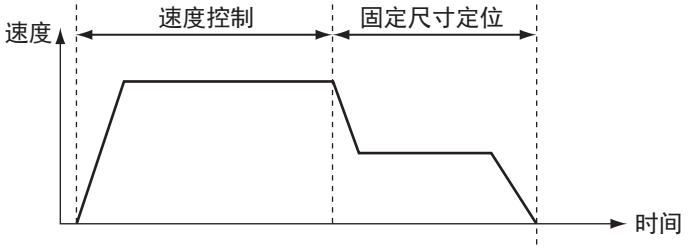
```
    Deceleration := Mv_Abs_Dec,
    Direction := Mv_Abs_Dir,
    Done => Mv_Abs_D,
    Busy => Mv_Abs_Bsy,
    Active => Mv_Abs_Act,
    CommandAborted => Mv_Abs_Ca,
    Error => Mv_Abs_Err,
    ErrorID => Mv_Abs_ErrID
);
```

10-2-11 中断固定尺寸定位

在速度控制过程中发生中断输入时，将切换为固定尺寸传送。
在旋转模式下进行速度控制时，在方向选择 (Direction) 中指定以下任意一个。

- _mcPositiveDirection （正方向指定）
- _mcNegativeDirection （负方向指定）
- _mcCurrentDirection （当前方向指定）

在本示例程序中，指定 _mcCurrentDirection （当前方向指定）。
向与发生中断输入前相同的方向进行固定尺寸输送时，在输入变量 “FeedDistance （固定尺寸距离） ” 中设定正数。向与发生中断输入前相反的方向进行固定尺寸输送时，在输入变量 “FeedDistance （固定尺寸距离） ” 中设定负数。
例如，发生中断输入前的动作为负方向，若在输入变量 “FeedDistance （固定尺寸距离） ” 中设定正数，则向负方向进行固定尺寸输送。若在输入变量 “FeedDistance （固定尺寸距离） ” 中设定负数，则向正方向进行固定尺寸输送。



轴参数设定

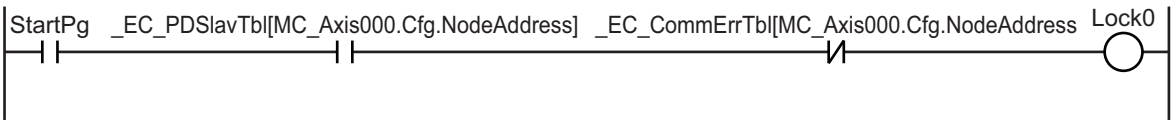
设定项目	设定内容	说明
计数模式	旋转模式	将位置的计数模式设定为旋转模式。
循环计数器上限设定值	360	将循环计数器的上限值设定为 “360”。
循环计数器下限设定值	0	将循环计数器的下限值设定为 “0”。
原点复位动作	原点预设	通过原点预设确定原点。

程序中使用的主要变量

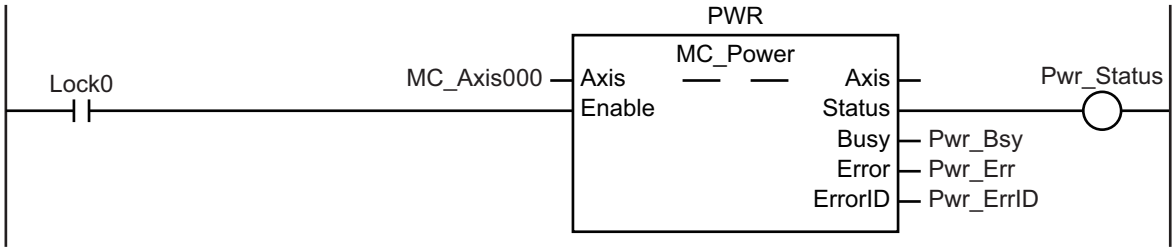
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 0 处于原点确定状态时变为 TRUE。
Pwr_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 通信的过程数据通信正在正常执行，则伺服 ON。
TrigRef	_sTRIGGER_REF	—	指定中断输入的触发输入条件。 在本示例程序中，使用伺服驱动器的锁定功能 1。
Hm_Ex	BOOL	FALSE	启动 MC_Home （原点复位）指令所需的变量。 在 ST 程序中使用。
Mv_Feed_Ex	BOOL	FALSE	启动 MC_MoveFeed （中断固定尺寸定位）指令所需的变量。 在 ST 程序中使用。
InitFlag	BOOL	FALSE	表示 MC_MoveFeed （中断固定尺寸定位）指令的输入参数设定完成的变量。

梯形图程序

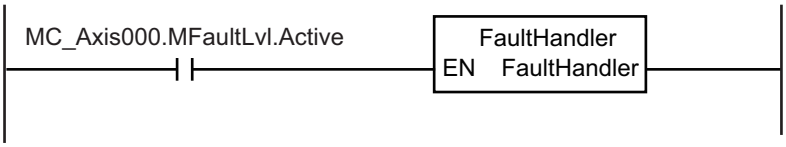
确认接点 StartPg 是否为 TRUE 且与轴 0 的过程数据通信是否正在正常执行



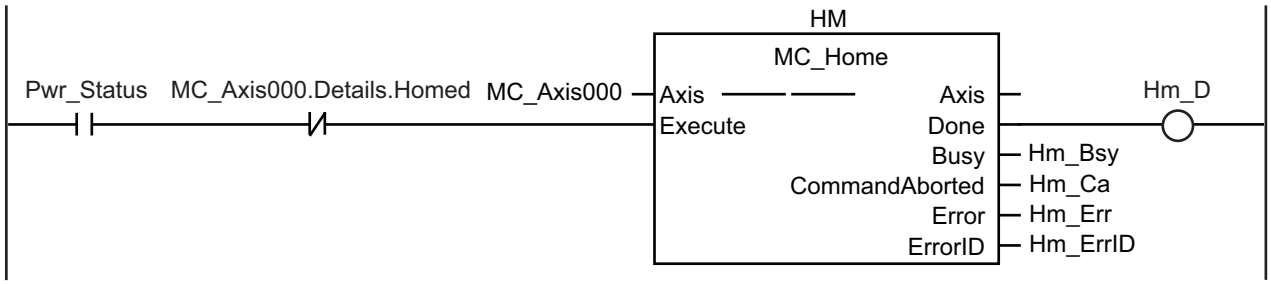
若与轴 0 的过程数据通信正在正常执行，则轴 0 变为伺服 ON



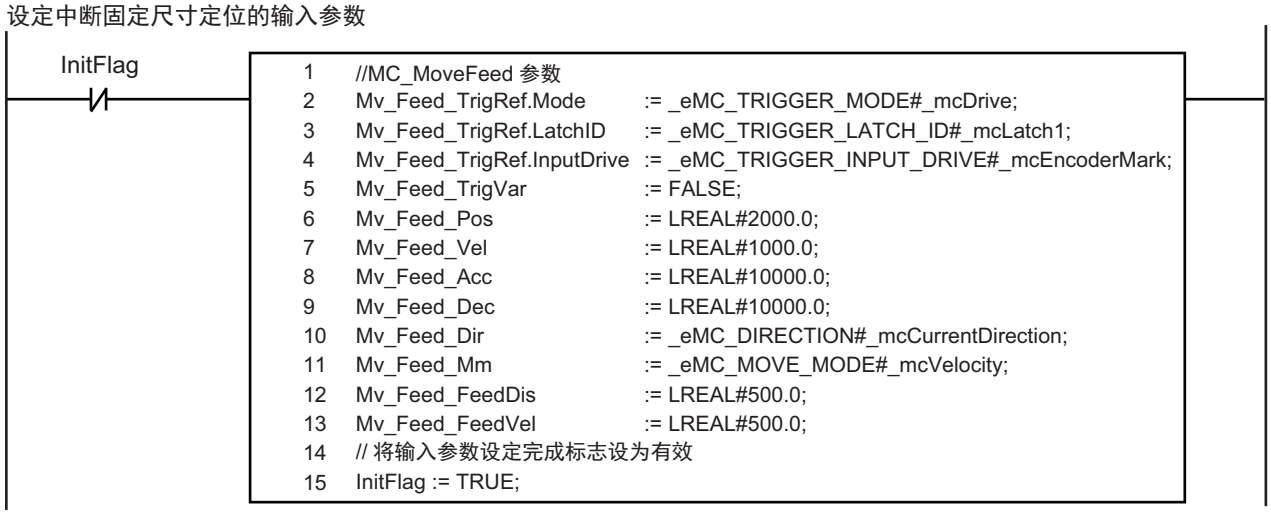
轴 0 发生轻度故障等级的异常时，执行异常处理 FaultHandler
异常处理的内容请根据装置进行编程



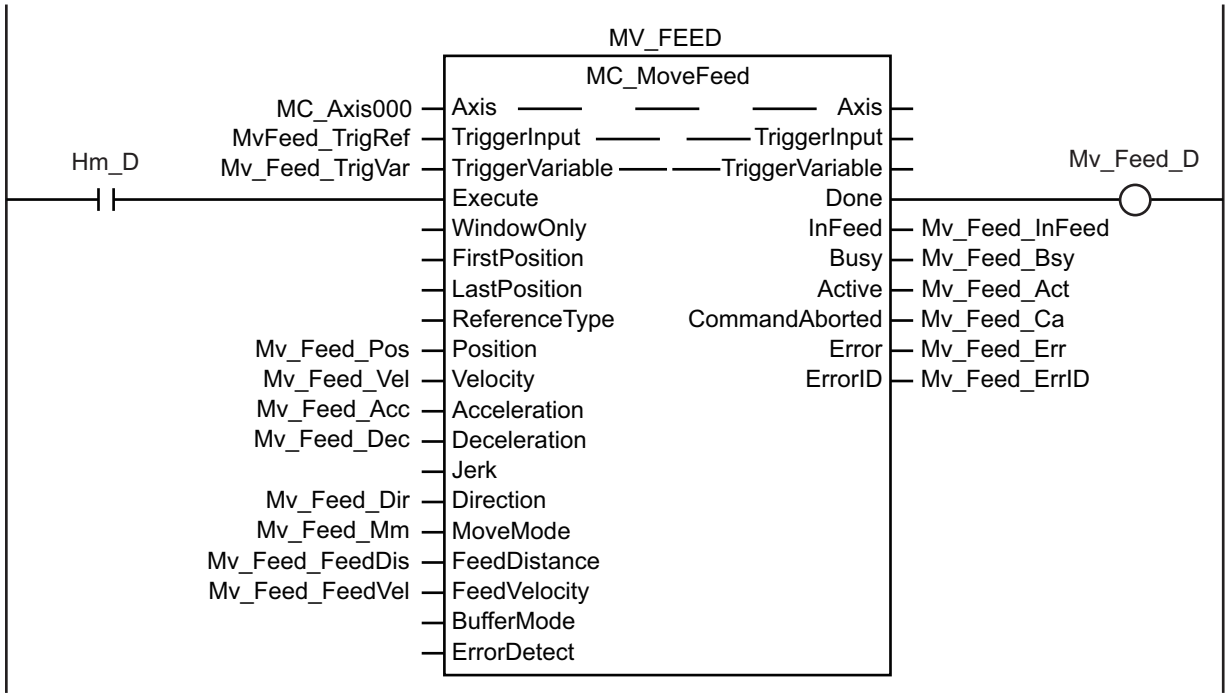
轴 0 为伺服 ON 状态且处于原点未确定状态时，执行原点复位



设定中断固定尺寸定位的输入参数



原点复位完成后，执行中断固定尺寸定位



ST 程序

```
// 未设定中断固定尺寸定位的输入参数时，设定目标值等
IF InitFlag=FALSE THEN
   //MC_MoveFeed 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#_mcEncoderMark;
   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_Dir := _eMC_DIRECTION#_mcCurrentDirection;
   Mv_Feed_Mm := _eMC_MOVE_MODE#_mcVelocity;
   Mv_Feed_FeedDis := LREAL#500.0;
   Mv_Feed_FeedVel := LREAL#500.0;

   // 将输入参数设定完成标志设为有效
   InitFlag := TRUE;
END_IF;


// 若接点 StartPg 为 TRUE 且过程数据通信正在正常执行，则轴 0 变为伺服 ON
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
   Pwr_En :=TRUE;
ELSE
   Pwr_En :=FALSE;
END_IF;


// 轴 0 发生轻度故障等级的异常时，执行异常处理 FaultHandler
// 异常处理的内容请根据装置进行编程
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
   FaultHandler();
END_IF;


// 轴 0 为伺服 ON 状态且处于原点未确定状态时，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，执行中断固定尺寸定位
IF Hm_D=TRUE THEN
   Mv_Feed_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_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,
    Direction := Mv_Feed_Dir,
    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
);
```

10-2-12 通过重启指令，切换凸轮表

可在执行凸轮动作的过程中切换凸轮表。
轴 0 的指令位置为 5,000 以下时，使用 CamProfile0，若超过 5,000，则使用 CamTable1。

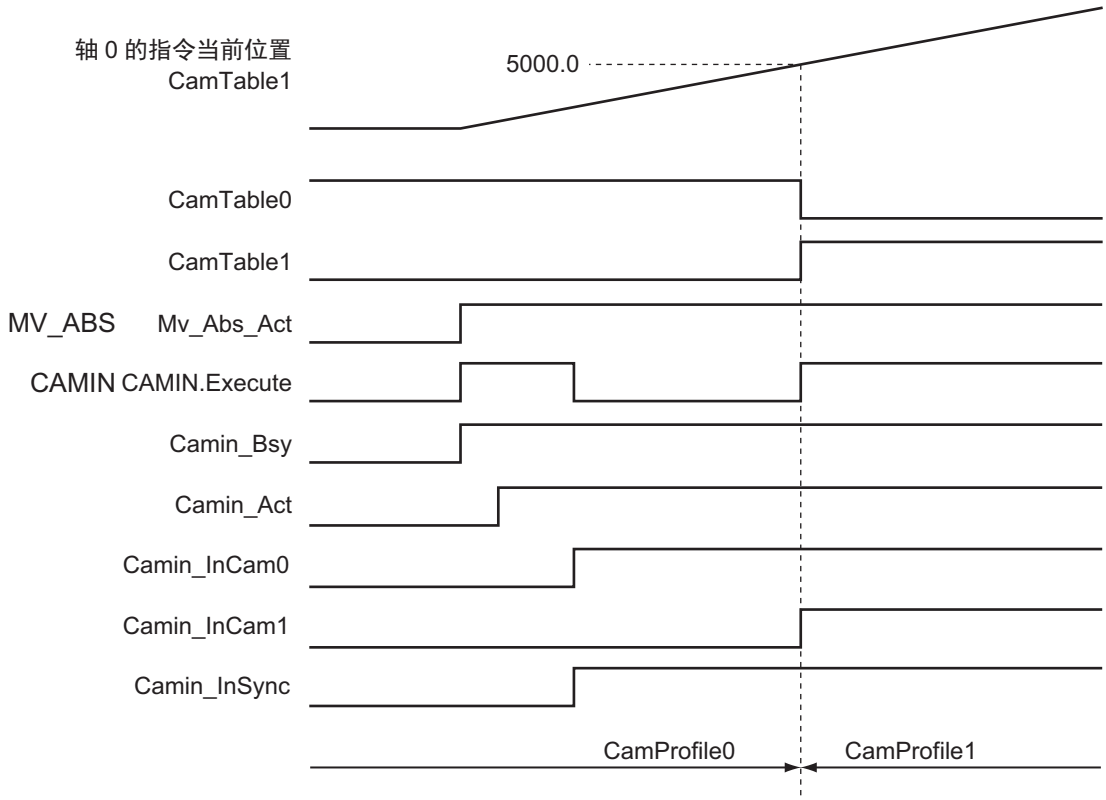
程序中使用的主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
MC_Axis001	_sAXIS_REF	—	轴 1 的轴变量。
CamProfile0	ARRAY[0..100] OF _sMC_CAM_REF	—	凸轮数据变量。*1
CamProfile1	ARRAY[0..10] OF _sMC_CAM_REF	—	凸轮数据变量。*1
Pwr1_S	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR1 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。
Pwr2_S	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR2 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。
CamTable0	BOOL	FALSE	将 CamProfile0 作为凸轮表使用时设为 TRUE。
CamTable1	BOOL	FALSE	将 CamProfile1 作为凸轮表使用时设为 TRUE。
Camin_InCam0	BOOL	FALSE	分配到 MC_CamIn 指令的实例 CAMIN 的输出 InCam 中的变量。 根据 CamProfile0 执行凸轮动作时变为 TRUE。 一旦变为 TRUE，将保持 TRUE，直至动作结束或中断。
Camin_InCam1	BOOL	FALSE	分配到 MC_CamIn 指令的实例 CAMIN 的输出 InCam 中的变量。 根据 CamProfile1 执行凸轮动作时变为 TRUE。 一旦变为 TRUE，将保持 TRUE，直至动作结束或中断。
Mv_Abs_Act	BOOL	FALSE	分配到 MC_MoveAbsolute 指令的实例 MV_ABS 的输出 Active 中的变量。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 通信的过程数据通信正在正常执行，则伺服 ON。
Hm1_Ex	BOOL	FALSE	在轴 0 中启动 MC_Home （原点复位）指令所需的变量。
Hm2_Ex	BOOL	FALSE	在轴 1 中启动 MC_Home （原点复位）指令所需的变量。
Mv_Abs_Ex	BOOL	FALSE	启动 MC_MoveAbsolute （绝对值定位）指令所需的变量。
Camin_Ex	BOOL	FALSE	启动 MC_CamIn （凸轮动作开始）指令所需的变量。 在 ST 程序中使用。

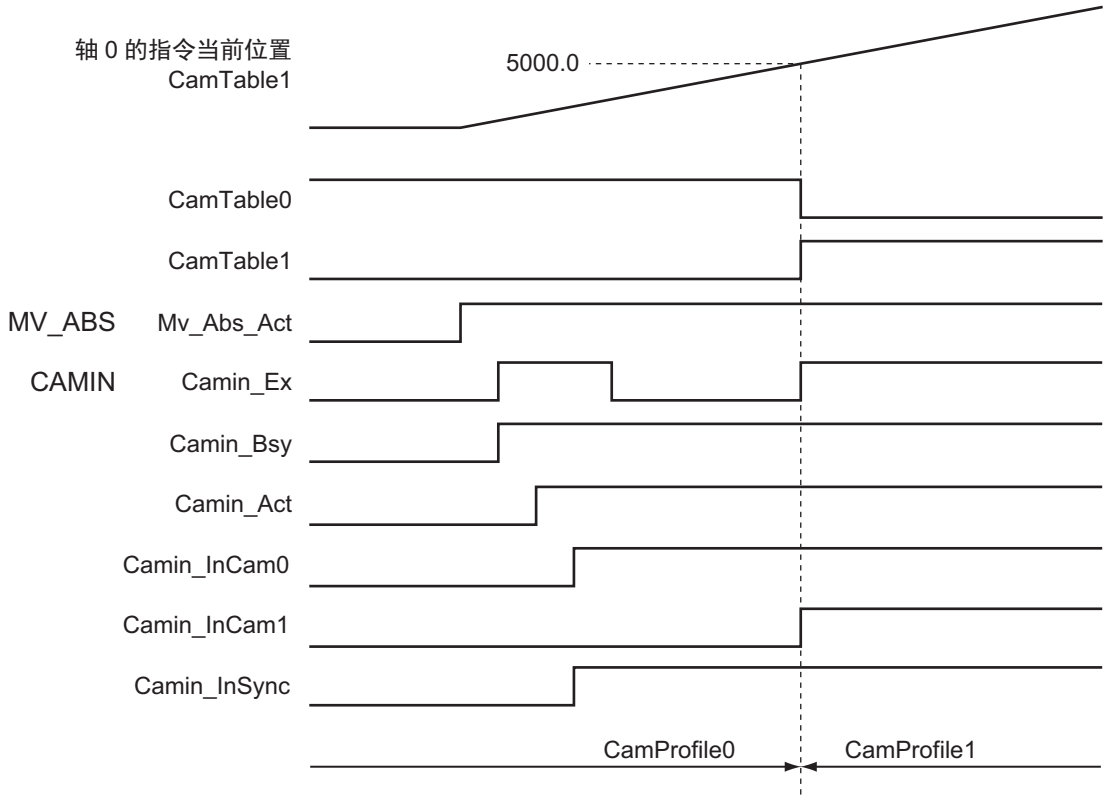
*1. 排列的要素数 ARRAY[0..N] 在 Sysmac Studio 的凸轮编辑器功能中设定。本示例程序中为 [0 ～ 109]。

时序图

● 梯形图程序时



● ST 程序时



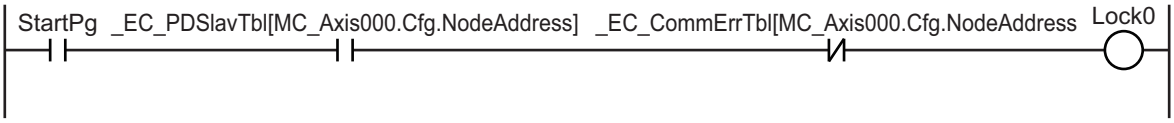
梯形图程序

切换使用两个凸轮表时，配置两个与 MC_CamIn （凸轮动作开始）指令相同名称的实例。

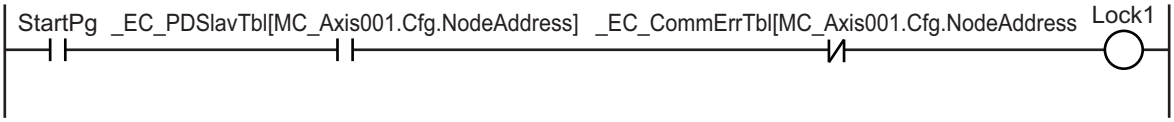
此时，在两个实例的输出变量 “InCam （凸轮动作中） ” 中分别分配不同的输出参数。若分配了相同的输出参数，将发生异常。

在本示例程序中，已利用 JMP （跳转）指令确保不会同时执行两个实例。

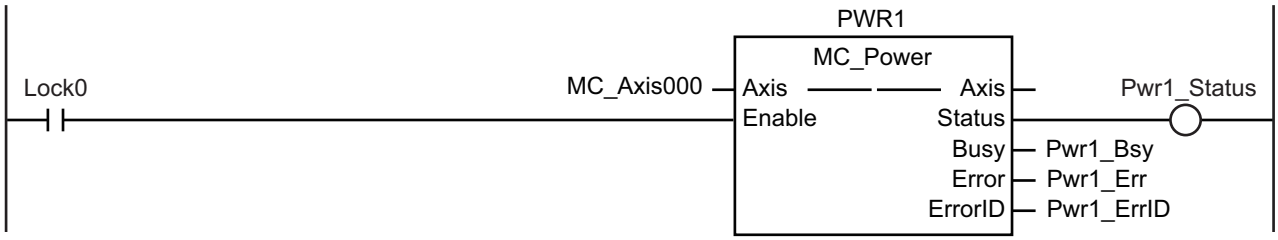
确认接点 StartPg 是否为 TRUE 且与轴 0 的过程数据通信是否正在正常执行



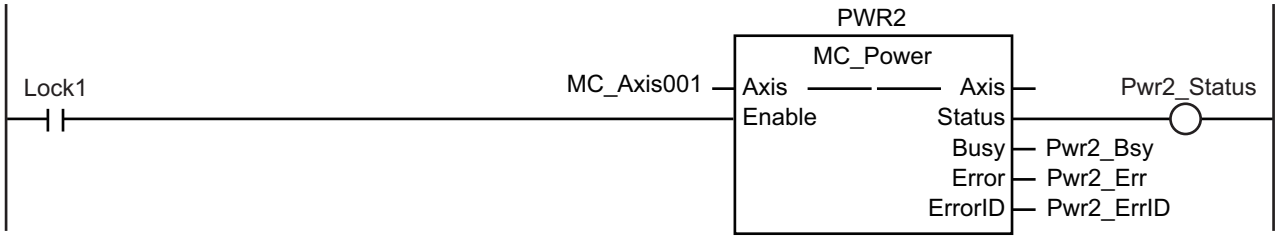
确认接点 StartPg 是否为 TRUE 且与轴 1 的过程数据通信是否正在正常执行



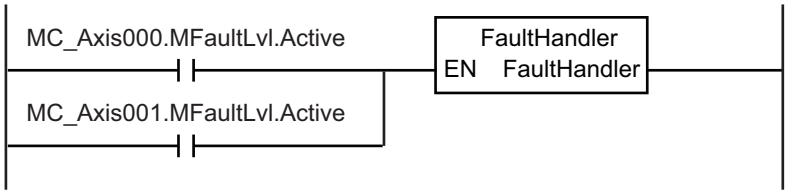
若与轴 0 的过程数据通信正在正常执行，则轴 0 变为伺服 ON



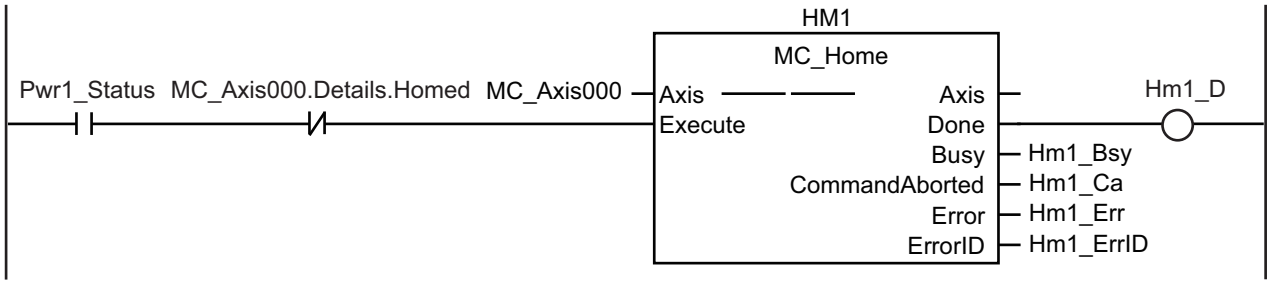
若与轴 1 的过程数据通信正在正常执行，则轴 1 变为伺服 ON



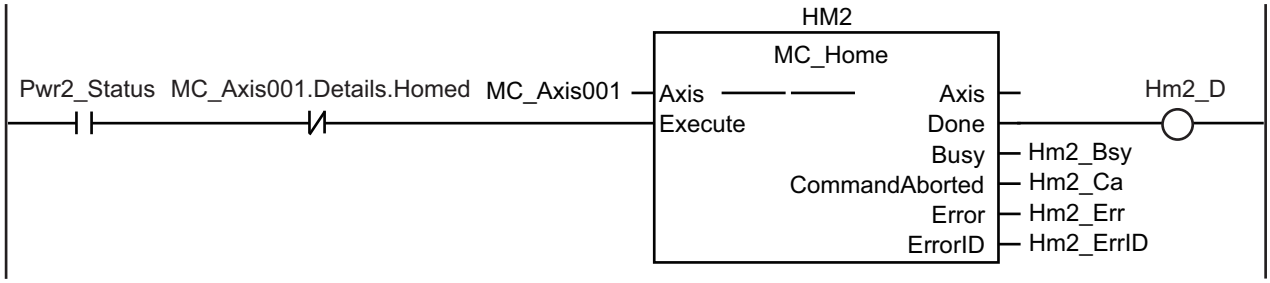
轴 0 或轴 1 发生轻度故障等级的异常时，执行异常处理 FaultHandler
异常处理的内容请根据装置进行编程



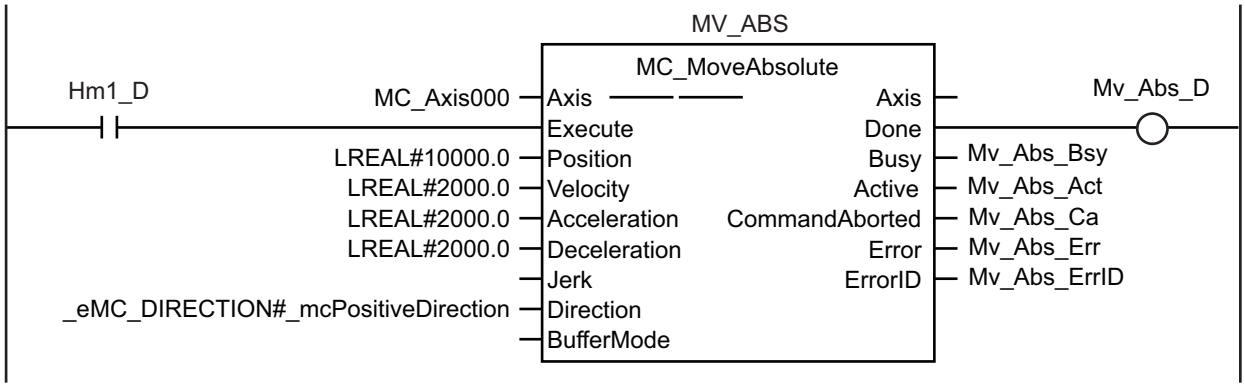
轴 0 为伺服 ON 状态且处于原点未确定状态时，执行原点复位



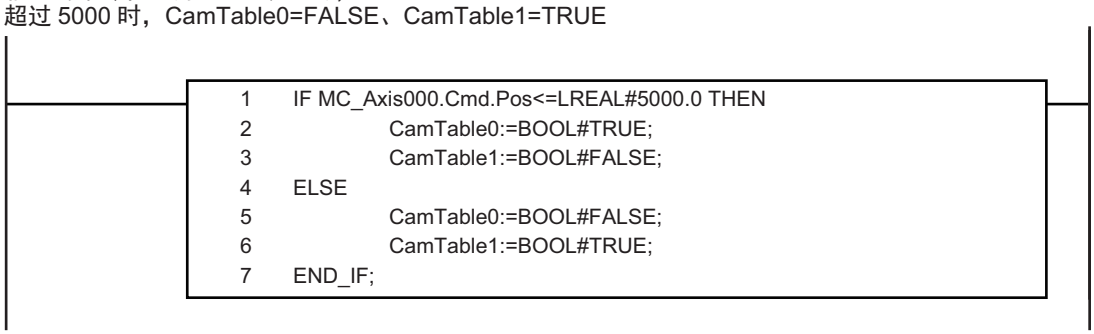
轴 1 为伺服 ON 状态且处于原点未确定状态时，执行原点复位



轴 0 的原点复位完成后，执行绝对值定位 (MC_MoveAbsolute) 指令

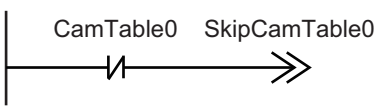


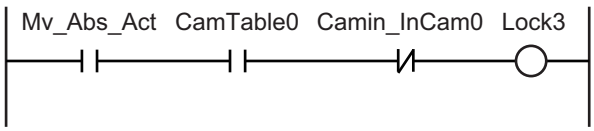
轴 0 的指令位置为 5000 以下时，CamTable0=TRUE、CamTable1=FALSE
超过 5000 时，CamTable0=FALSE、CamTable1=TRUE

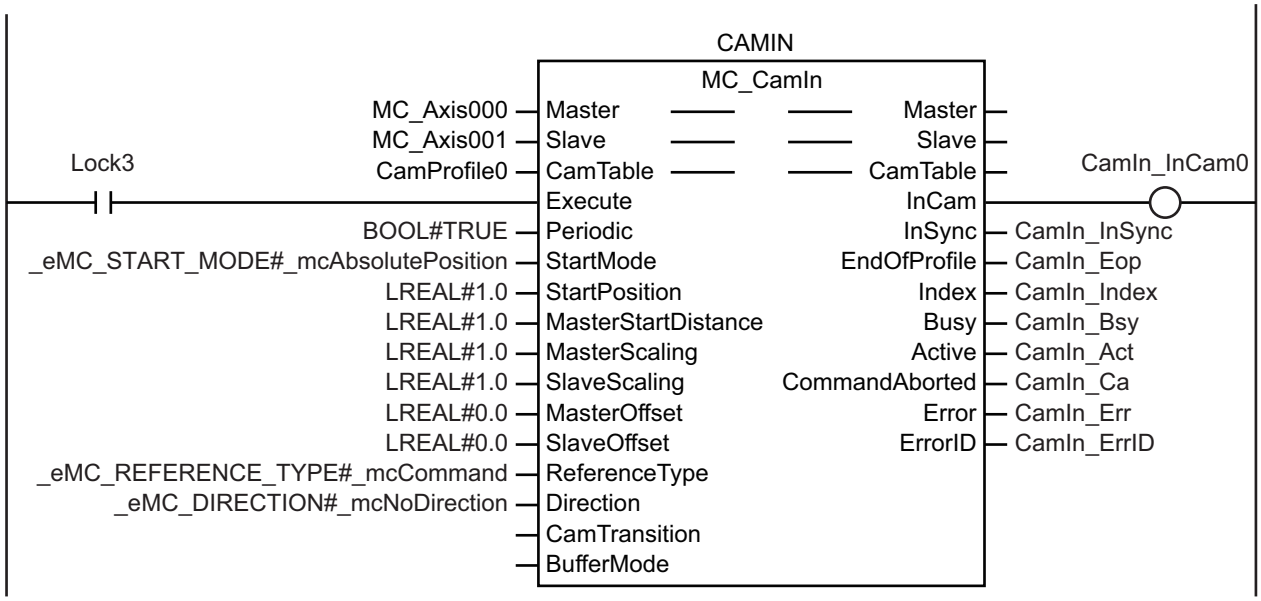


```
1   IF MC_Axis000.Cmd.Pos<=LREAL#5000.0 THEN
2           CamTable0:=BOOL#TRUE;
3           CamTable1:=BOOL#FALSE;
4   ELSE
5           CamTable0:=BOOL#FALSE;
6           CamTable1:=BOOL#TRUE;
7   END_IF;
```

正在执行绝对值定位且 CamTable0=TRUE 时，执行凸轮表中使用 CamProfile0 的实例
InCam=TRUE 时，将 Execute 设为 FALSE

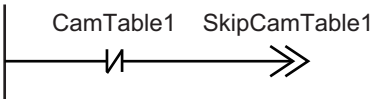


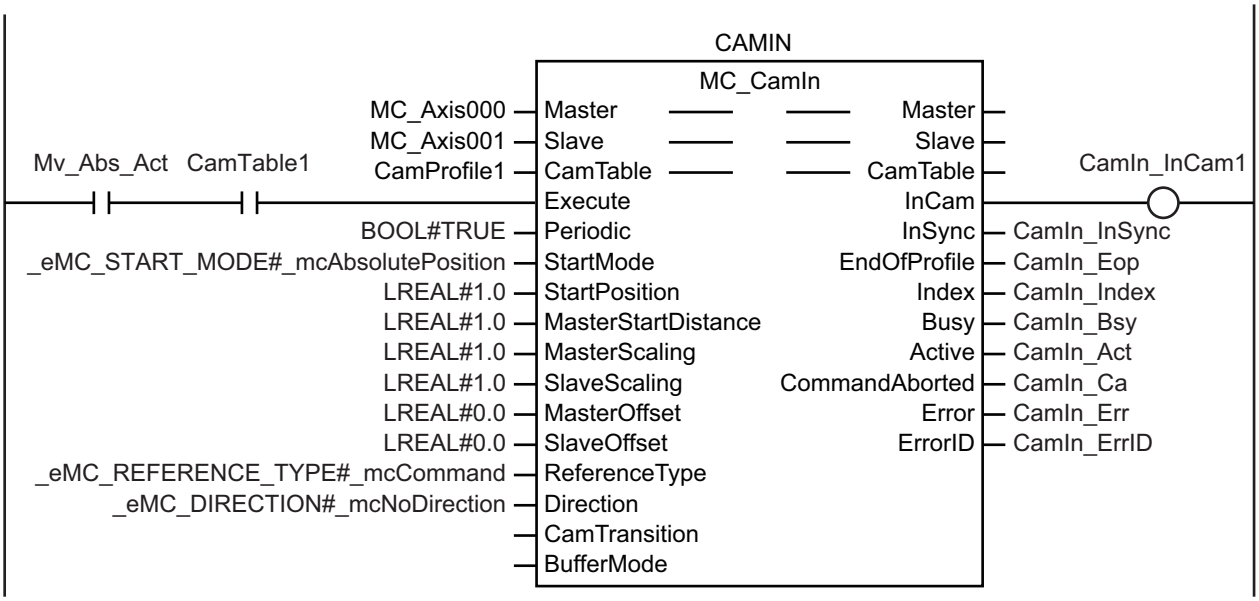




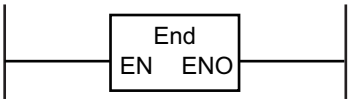
正在执行绝对值定位且 CamTable1=TRUE 时，执行凸轮表中使用 CamProfile1 实例

SkipCamTable0





SkipCamTable1



ST 程序

```
// 未设定绝对值定位和凸轮动作开始的输入参数时，设定目标值等
IF InitFlag=FALSE THEN
   // 设定绝对值定位 (MC_MoveAbsolute) 的输入参数
   Mv_Abs_Pos := LREAL#10000.0;
   Mv_Abs_Vel := LREAL#2000.0;
   Mv_Abs_Acc := LREAL#2000.0;
   Mv_Abs_Dec := LREAL#2000.0;

   // 设定凸轮动作开始 (MC_CamIn) 的输入参数
   Camin_EM := TRUE;
   Camin_StMode := _eMC_START_MODE#_mcAbsolutePosition;
   Camin_StPos := LREAL#1.0;
   Camin_MStDis := LREAL#1.0;
   Camin_MSc := LREAL#1.0;
   Camin_SSc := LREAL#1.0;
   Camin_MO := LREAL#0.0;
   Camin_SO := LREAL#0.0;
   Camin_RT := _eMC_REFERENCE_TYPE#_mcCommand;
   Camin_Dir := _eMC_DIRECTION#_mcNoDirection;

   // 凸轮表的选择
   CamTable0 :=BOOL#TRUE;
   CamTable1 :=BOOL#FALSE;

   // 将输入参数设定完成标志设为有效
   InitFlag := TRUE;
END_IF;


// 若接点 StartPg 为 TRUE 且过程数据通信正在正常执行，则轴 0 变为伺服 ON
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
   Pwr1_En :=TRUE;
ELSE
   Pwr1_En :=FALSE;
END_IF;


// 若接点 StartPg 为 TRUE 且过程数据通信正在正常执行，则轴 1 变为伺服 ON
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
   Pwr2_En :=TRUE;
ELSE
   Pwr2_En :=FALSE;
END_IF;


// 轴 0 或轴 1 发生轻度故障等级的异常时，执行异常处理 FaultHandler
// 异常处理的内容请根据装置进行编程
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
   FaultHandler();
END_IF;
```

```
// 轴 0 为伺服 ON 且处于原点未确定状态时，执行原点复位
IF (Pwr1_S=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
   Hm1_Ex :=TRUE;
END_IF;


// 轴 1 为伺服 ON 且处于原点未确定状态时，执行原点复位
IF (Pwr2_S=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
   Hm2_Ex :=TRUE;
END_IF;


// 轴 0 的原点复位完成后，执行绝对值定位
IF Hm1_D=TRUE  THEN
   Mv_Abs_Ex := TRUE;
END_IF;


// 轴 0 的指令位置为 5000 以下时， CamTable0=TRUE、 CamTable1=FALSE
// 超出 5000 时， CamTable0=FALSE、 CamTable1=TRUE
IF MC_Axis000.Cmd.Pos<=LREAL#5000.0 THEN
   CamTable0 :=BOOL#TRUE;
   CamTable1 :=BOOL#FALSE;
ELSE
   CamTable0 :=BOOL#FALSE;
   CamTable1 :=BOOL#TRUE;
END_IF;


// 正在执行绝对值定位且 CamTable0=TRUE 时，凸轮表中使用 CamProfile0
// 执行实例
//InCam=TRUE 时， Execute 设为 FALSE
IF (Mv_Abs_Act=TRUE)
 AND (CamTable0=TRUE)
 AND (Camin_InCam0=FALSE) THEN
   Camin_Ex := TRUE;
ELSE
   Camin_Ex := FALSE;
END_IF;


// 正在执行绝对值定位且 CamTable1=TRUE 时，凸轮表中使用 CamProfile1
// 执行实例
IF (Mv_Abs_Act=TRUE) AND (CamTable1=TRUE) THEN
   Camin_Ex := TRUE;
END_IF;

//MC_Camin
IF CamTable0=TRUE THEN
   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,
        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
    );
END_IF;

IF CamTable1=TRUE THEN
    CAMIN(
        Master := MC_Axis000,
        Slave := MC_Axis001,
        CamTable := CamProfile1,
        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,
        InCam => Camin_InCam1,
        InSync => Camin_InSync,
        EndOfProfile => Camin_EOP,
        Index => Camin_Index,
        Busy => Camin_Bsy,
        Active => Camin_Act,
        CommandAborted => Camin_Ca,
        Error => Camin_Err,
        ErrorID => Camin_ErrID
    );
END_IF;

// 轴 0 的 MC_Power
PWR1(
    Axis := MC_Axis000,
    Enable := Pwr1_En,
    Status => Pwr1_S,
    Busy => Pwr1_Bsy,
    Error => Pwr1_Err,
    ErrorID => Pwr1_ErrID
);

// 轴 1 的 MC_Power
PWR2(
    Axis := MC_Axis001,
```

```
    Enable := Pwr2_En,
    Status => Pwr2_S,
    Busy => Pwr2_Bsy,
    Error => Pwr2_Err,
    ErrorID => Pwr2_ErrID
);

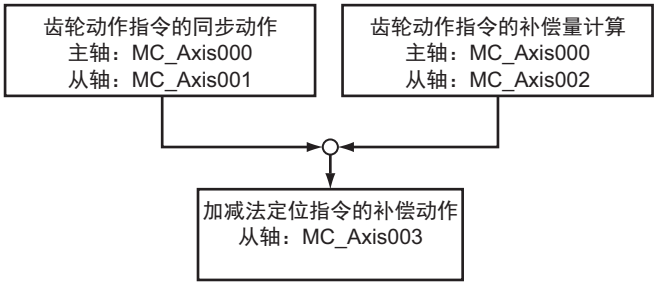
// 轴 0 的 MC_Home
HM1(
    Axis := MC_Axis000,
    Execute := Hm1_Ex,
    Done => Hm1_D,
    Busy => Hm1_Bsy,
    CommandAborted => Hm1_Ca,
    Error => Hm1_Err,
    ErrorID => Hm1_ErrID
);

// 轴 1 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_MoveAbsolute
MV_ABS(
    Axis := MC_Axis000,
    Execute := Mv_Abs_Ex,
    Position := Mv_Abs_Pos,
    Velocity := Mv_Abs_Vel,
    Acceleration := Mv_Abs_Acc,
    Deceleration := Mv_Abs_Dec,
    Direction := Mv_Abs_Dir,
    Done => Mv_Abs_D,
    Busy => Mv_Abs_Bsy,
    Active => Mv_Abs_Act,
    CommandAborted => Mv_Abs_Ca,
    Error => Mv_Abs_Err,
    ErrorID => Mv_Abs_ErrID
);
```

10-2-13 使用凸轮轮廓曲线进行同步位置修正

对执行齿轮动作的从轴，用凸轮轮廓曲线进行修正。
齿轮动作的从轴为虚拟伺服轴 MC_Axis001，凸轮动作的从轴为虚拟伺服轴 MC_Axis002。利用 MC_CombineAxes 将这些从轴相加，将结果输出到伺服轴 MC_Axis003 中。主轴为伺服轴 MC_Axis000。
处理流程如下。



● 轴种类的设定

轴种类在各轴的轴参数中设定如下。

设定项目	设定内容			
	轴 1	轴 2	轴 3	轴 4
轴变量名称	MC_Axis000	MC_Axis001	MC_Axis002	MC_Axis003
轴种类	伺服轴	虚拟伺服轴	虚拟伺服轴	伺服轴

程序中使用的主要变量

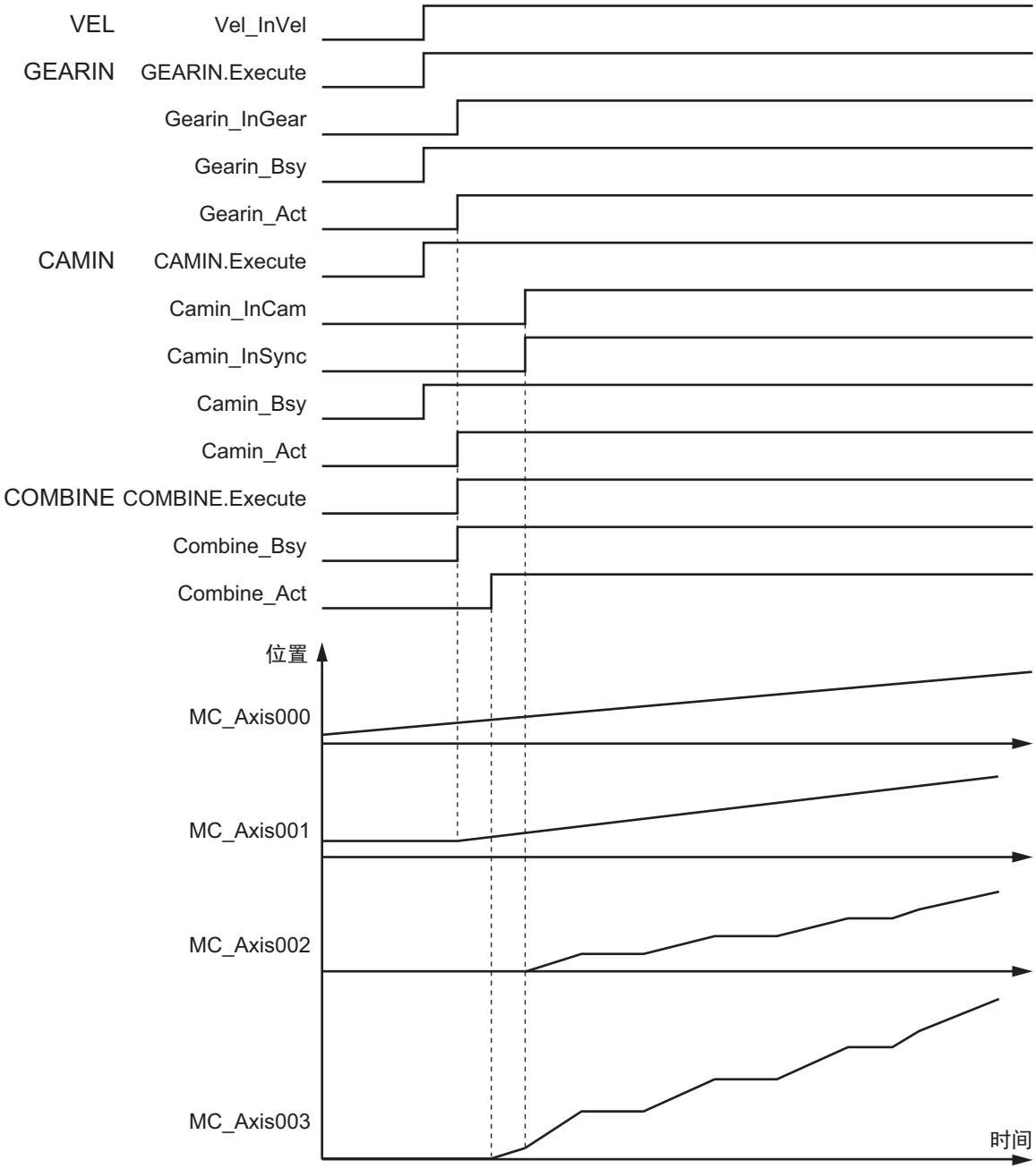
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 0 处于原点确定状态时变为 TRUE。
MC_Axis001	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis002	_sAXIS_REF	—	轴 2 的轴变量。
MC_Axis002.MFaultLvl.Active	BOOL	FALSE	轴 2 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis003	_sAXIS_REF	—	轴 3 的轴变量。
MC_Axis003.MFaultLvl.Active	BOOL	FALSE	轴 3 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis003.Details.Homed	BOOL	FALSE	轴 3 处于原点确定状态时变为 TRUE。
CamProfile0	ARRAY[0..109] OF _sMC_CAM_REF	—	凸轮数据变量。*1
Pwr1_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR1 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。
Pwr4_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR4 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。
Vel_InVel	BOOL	FALSE	轴 0 的 MC_MoveVelocity 达到目标速度时，变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 通信的过程数据通信正在正常执行，则伺服 ON。

名称	数据类型	初始值	注释
Gearin_Ex	BOOL	FALSE	启动 MC_GearIn （齿轮动作开始）指令所需的变量。*2
Camin_Ex	BOOL	FALSE	启动 MC_CamIn（凸轮动作开始）指令所需的变量。
Combine_Ex	BOOL	FALSE	启动 MC_CombineAxes （加减法定位）指令所需的变量。*2

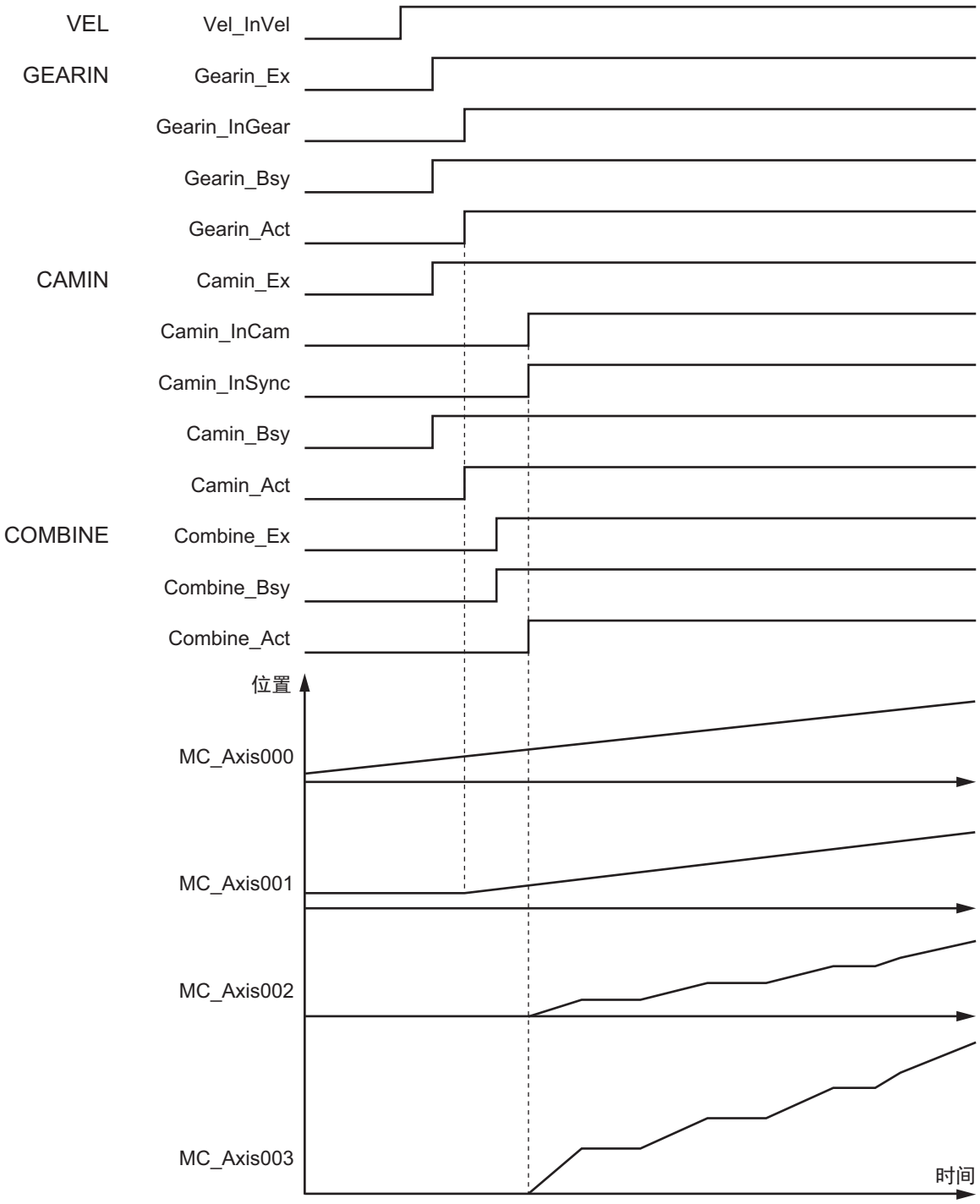
*1. 排列的要素数 ARRAY[0..N] 在 Sysmac Studio 的凸轮编辑器功能中设定。本示例程序中为 [0 ～ 109]。
*2. 在 ST 程序中使用。

时序图

● 梯形图程序时

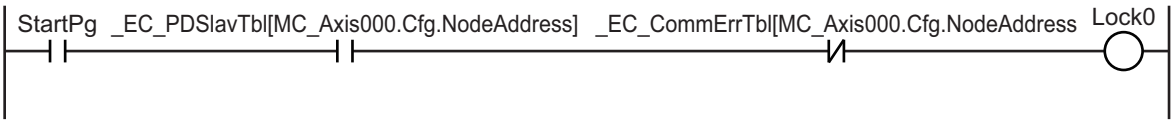


● **ST 程序时**

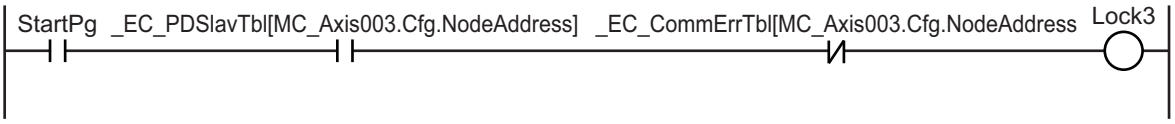

VEL
Vel_InVel
GEARIN
Gearin_Ex
Gearin_InGear
Gearin_Bsy
Gearin_Act
CAMIN
Camin_Ex
Camin_InCam
Camin_InSync
Camin_Bsy
Camin_Act
COMBINE
Combine_Ex
Combine_Bsy
Combine_Act
位置
MC_Axis000
MC_Axis001
MC_Axis002
MC_Axis003
时间

梯形图程序

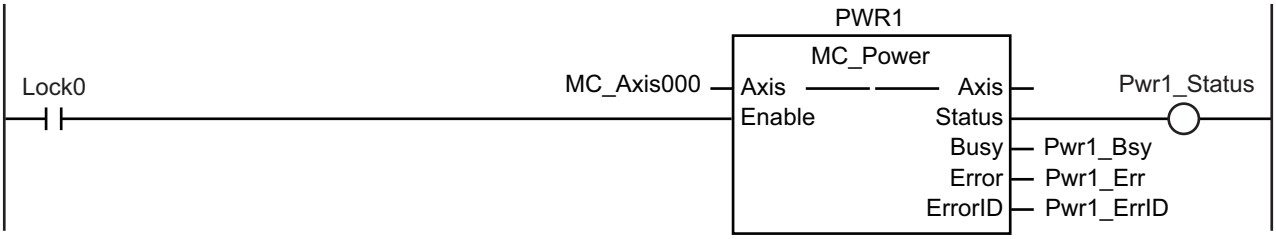
确认接点 StartPg 是否为 TRUE 且与轴 0 的过程数据通信是否正在正常执行



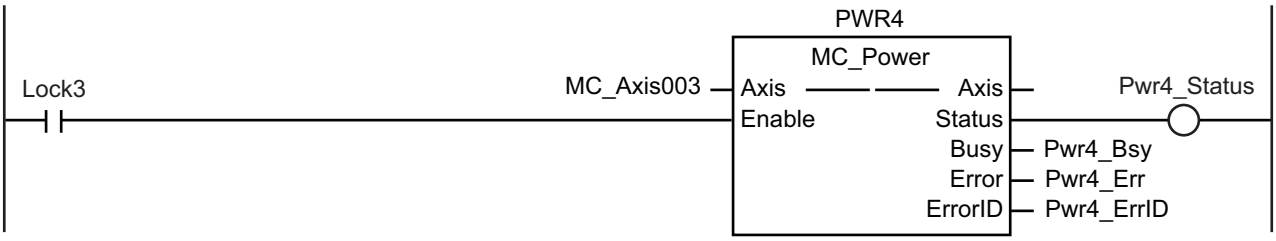
确认接点 StartPg 是否为 TRUE 且与轴 3 的过程数据通信是否正在正常执行



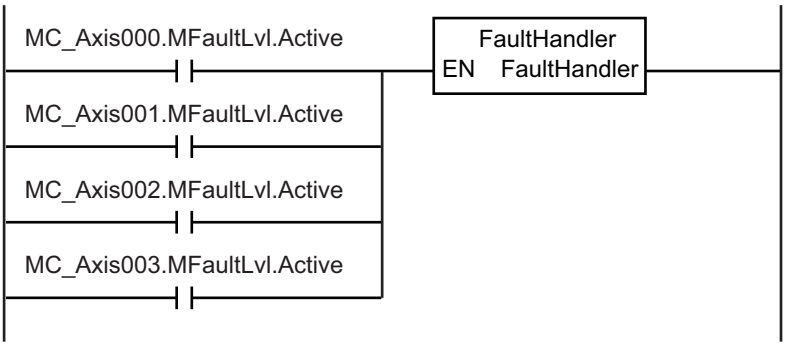
若与轴 0 的过程数据通信正在正常执行，则轴 0 变为伺服 ON



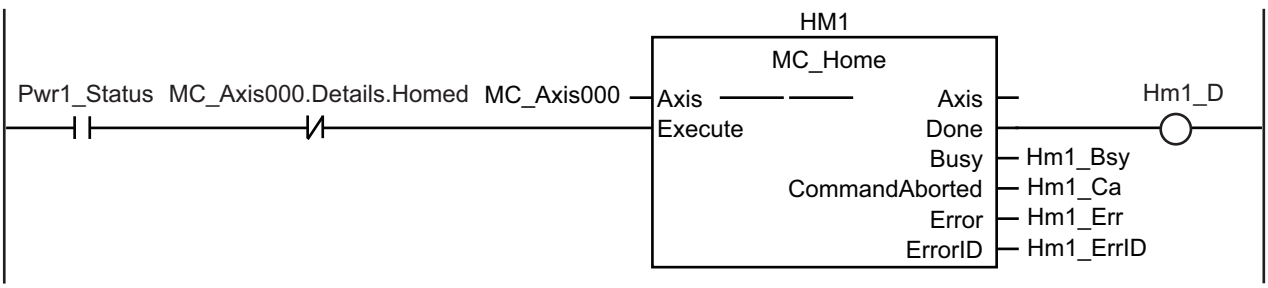
若与轴 3 的过程数据通信正在正常执行，则轴 3 变为伺服 ON



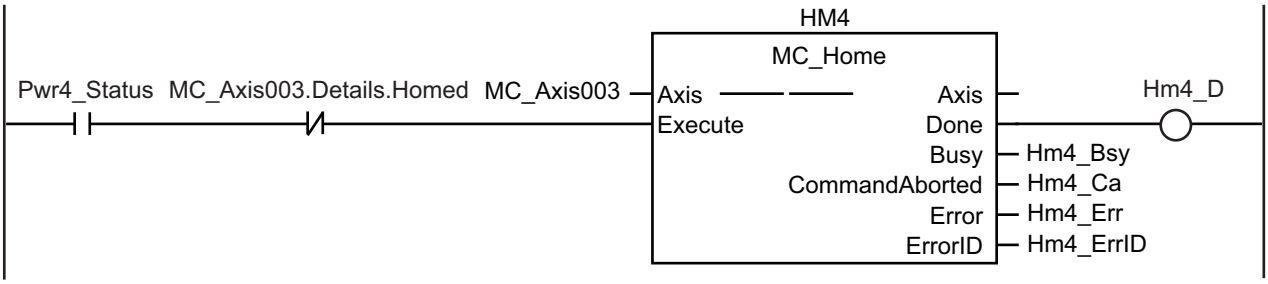
轴组或构成轴发生轻度故障等级的异常时，执行异常处理 FaultHandler
异常处理的内容请根据装置进行编程



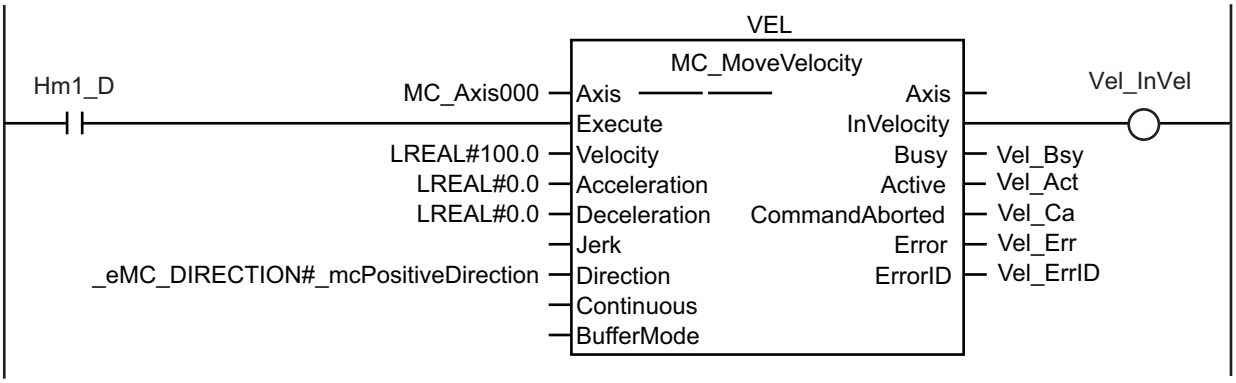
轴 0 为伺服 ON 状态且处于原点未确定状态时，执行原点复位



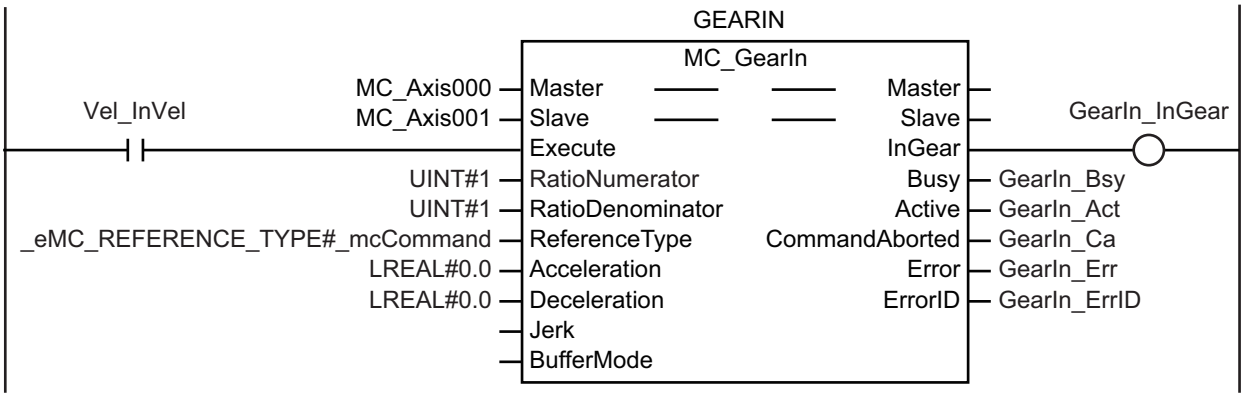
轴 3 为伺服 ON 状态且处于原点未确定状态时，执行原点复位



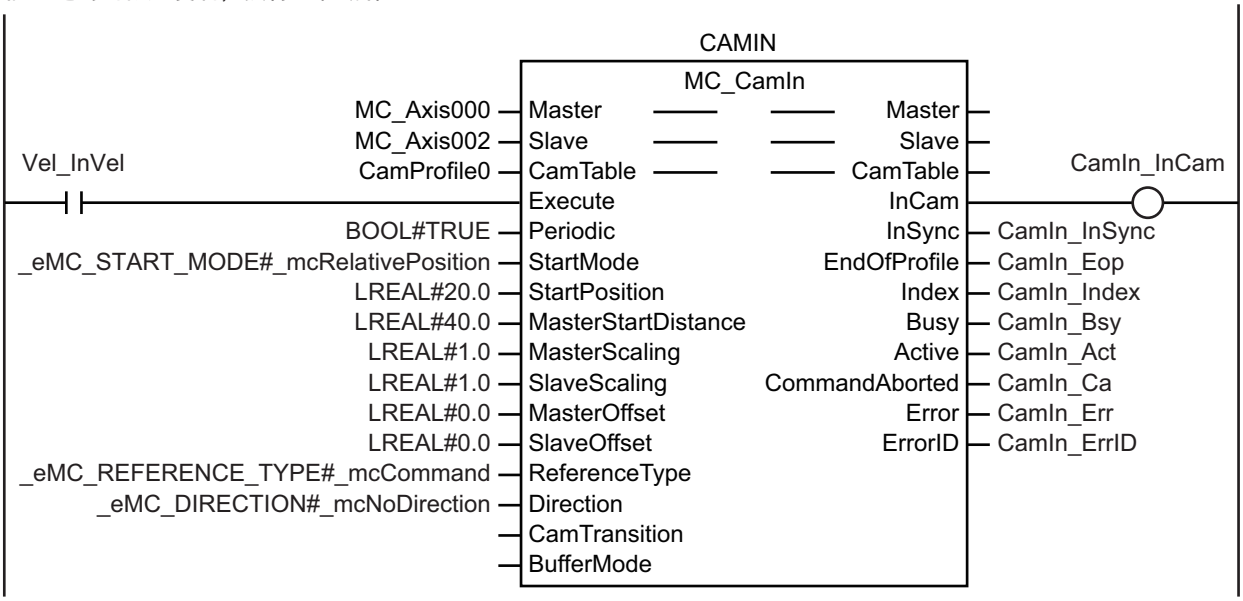
轴 0 的原点复位完成后，执行速度控制



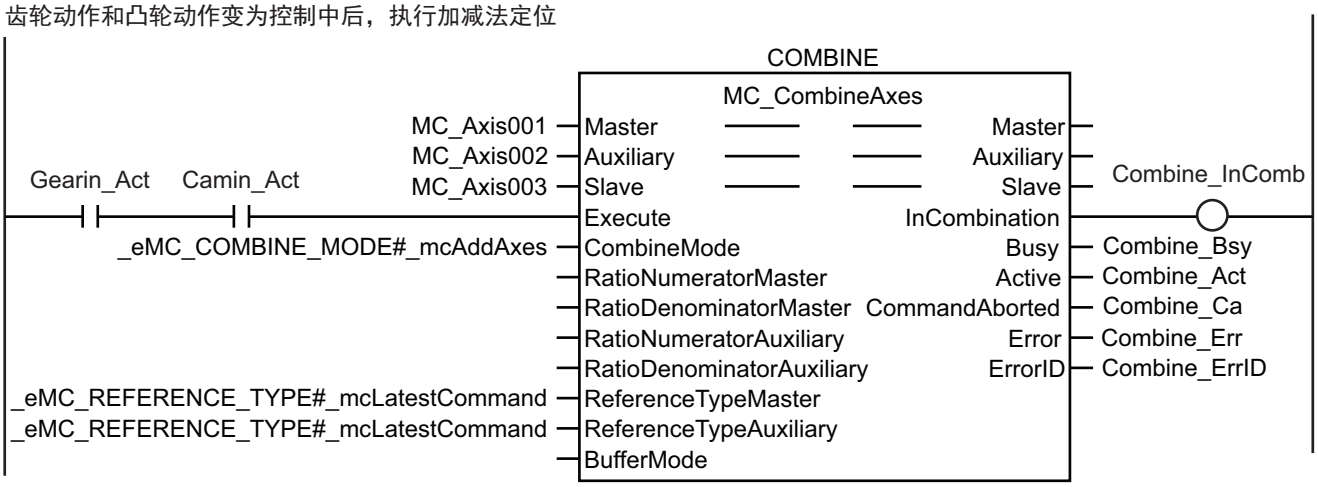
轴 0 达到目标速度后，执行齿轮动作



轴 0 达到目标速度后，执行凸轮动作



齿轮动作和凸轮动作变为控制中后，执行加减法定位



ST 程序

```
// 未设定动作指令的输入参数时，设定目标值等
IF InitFlag=FALSE THEN
   // 在速度控制 (MC_MoveVelocity) 的输入参数中设定值
   Vel_Vel := LREAL#100.0;
   Vel_Acc := LREAL#0.0;
   Vel_Dec := LREAL#0.0;
   Vel_Dir := _eMC_DIRECTION#_mcPositiveDirection;

   // 在凸轮动作开始 (MC_CamIn) 的输入参数中设定值
   Camin_Em := TRUE;
   Camin_Sm := _eMC_START_MODE#_mcRelativePosition;
   Camin_Sp := LREAL#20.0;
   Camin_Msd := LREAL#40.0;
   Camin_Ms := LREAL#1.0;
   Camin_Ss := LREAL#1.0;
   Camin_Mo := LREAL#0.0;
   Camin_So := LREAL#0.0;
   Camin_Rt := _eMC_REFERENCE_TYPE#_mcCommand;
   Camin_Dir := _eMC_DIRECTION#_mcNoDirection;

   // 在齿轮动作开始 (MC_GearIn) 的输入参数中设定值
   Gearin_RatN := UINT#1;
   Gearin_RatD := UINT#1;
   Gearin_RefTyp := _eMC_REFERENCE_TYPE#_mcCommand;
   Gearin_Acc := LREAL#0.0;
   Gearin_Dec := LREAL#0.0;

   // 在加减法定位 (MC_CombineAxes) 的输入参数中设定值
   Combine_Cm := _eMC_COMBINE_MODE#_mcAddAxes;
   Combine_RefMas :=_eMC_REFERENCE_TYPE#_mcLatestCommand;
   Combine_RefAux :=_eMC_REFERENCE_TYPE#_mcLatestCommand;
   // 将输入参数设定完成标志设为有效
   InitFlag := TRUE;
END_IF;


// 若接点 StartPg 为 TRUE 且过程数据通信正在正常执行，则轴 0 变为伺服 ON
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
```

```
    Pwr1_En :=TRUE;
ELSE
    Pwr1_En :=FALSE;
END_IF;


// 若接点 StartPg 为 TRUE 且过程数据通信正在正常执行，则轴 3 变为伺服 ON
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis003.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis003.Cfg.NodeAddress]=FALSE) THEN
    Pwr4_En :=TRUE;
ELSE
    Pwr4_En :=FALSE;
END_IF;


// 轴 0 ～轴 3 发生轻度故障等级的异常时，执行异常处理 FaultHandler
// 异常处理的内容请根据装置进行编程
IF (MC_Axis000.MFaultLvl.Active=TRUE)
 OR (MC_Axis001.MFaultLvl.Active=TRUE)
 OR (MC_Axis002.MFaultLvl.Active=TRUE)
 OR (MC_Axis003.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;


// 轴 0 为伺服 ON 状态且处于原点未确定状态时，执行轴 0 的原点复位
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex :=TRUE;
END_IF;


// 轴 3 为伺服 ON 状态且处于原点未确定状态时，执行轴 3 的原点复位
IF (Pwr4_Status=TRUE) AND (MC_Axis003.Details.Homed=FALSE) THEN
    Hm4_Ex :=TRUE;
END_IF;


// 轴 0 的原点复位完成后，执行速度控制
IF Hm1_D=TRUE THEN
    Vel_Ex :=TRUE;
END_IF;


// 轴 0 达到目标速度后，执行齿轮动作
IF Vel_InVel=TRUE THEN
    Gearin_Ex := TRUE;
END_IF;


// 轴 0 达到目标速度后，执行凸轮动作
IF Vel_InVel=TRUE THEN
    Camin_Ex := TRUE;
END_IF;


// 齿轮动作和凸轮动作变为控制中后，执行加减法定位
IF (Gearin_Act=TRUE) AND (Camin_Act=TRUE) THEN
```

```
    Combine_Ex :=TRUE;
END_IF;

// 轴 0 的 MC_Power
PWR1(
    Axis := MC_Axis000,
    Enable := Pwr1_En,
    Status => Pwr1_Status,
    Busy => Pwr1_Bsy,
    Error => Pwr1_Err,
    ErrorID => Pwr1_ErrID
);

// 轴 3 的 MC_Power
PWR4(
    Axis := MC_Axis003,
    Enable := Pwr4_En,
    Status => Pwr4_Status,
    Busy => Pwr4_Bsy,
    Error => Pwr4_Err,
    ErrorID => Pwr4_ErrID
);

// 轴 0 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 3 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
);

//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_Axis002,
    CamTable := CamProfile0,
    Execute := Camin_Ex,
    Periodic := Camin_Em,
    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_GearIn
GEARIN(
    Master := MC_Axis000,
    Slave := MC_Axis001,
    Execute := Gearin_Ex,
    RatioNumerator := Gearin_RatN,
    RatioDenominator := Gearin_RatD,
    ReferenceType := Gearin_RefTyp,
    Acceleration := Gearin_Acc,
    Deceleration := Gearin_Dec,
    InGear => Gearin_InGear,
    Busy => Gearin_Bsy,
    Active => Gearin_Act,
    CommandAborted => Gearin_Ca,
    Error => Gearin_Err,
    ErrorID => Gearin_ErrID
);

//MC_CombineAxes
COMBINE(
    Master := MC_Axis001,
    Auxiliary := MC_Axis002,
    Slave := MC_Axis003,
    Execute := Combine_Ex,
    CombineMode := Combine_CM,
    ReferenceTypeMaster := Combine_RefMas,
    ReferenceTypeAuxiliary := Combine_RefAux,
    InCombination => Combine_InComb,
    Busy => Combine_Bsy,
    Active => Combine_Act,
    CommandAborted => Combine_Ca,
```

```
    Error => Combine_Err,
    ErrorID => Combine_ErrID
);
```

10-2-14 凸轮动作时进行主轴的相位补偿

凸轮动作的从轴与速度控制中动作的主轴同步进行动作。若将 StartOn 设为 TRUE，则利用 MC_Phasing （主轴相对值相位补偿）指令补偿主轴的相位。从轴与补偿后的相位同步。

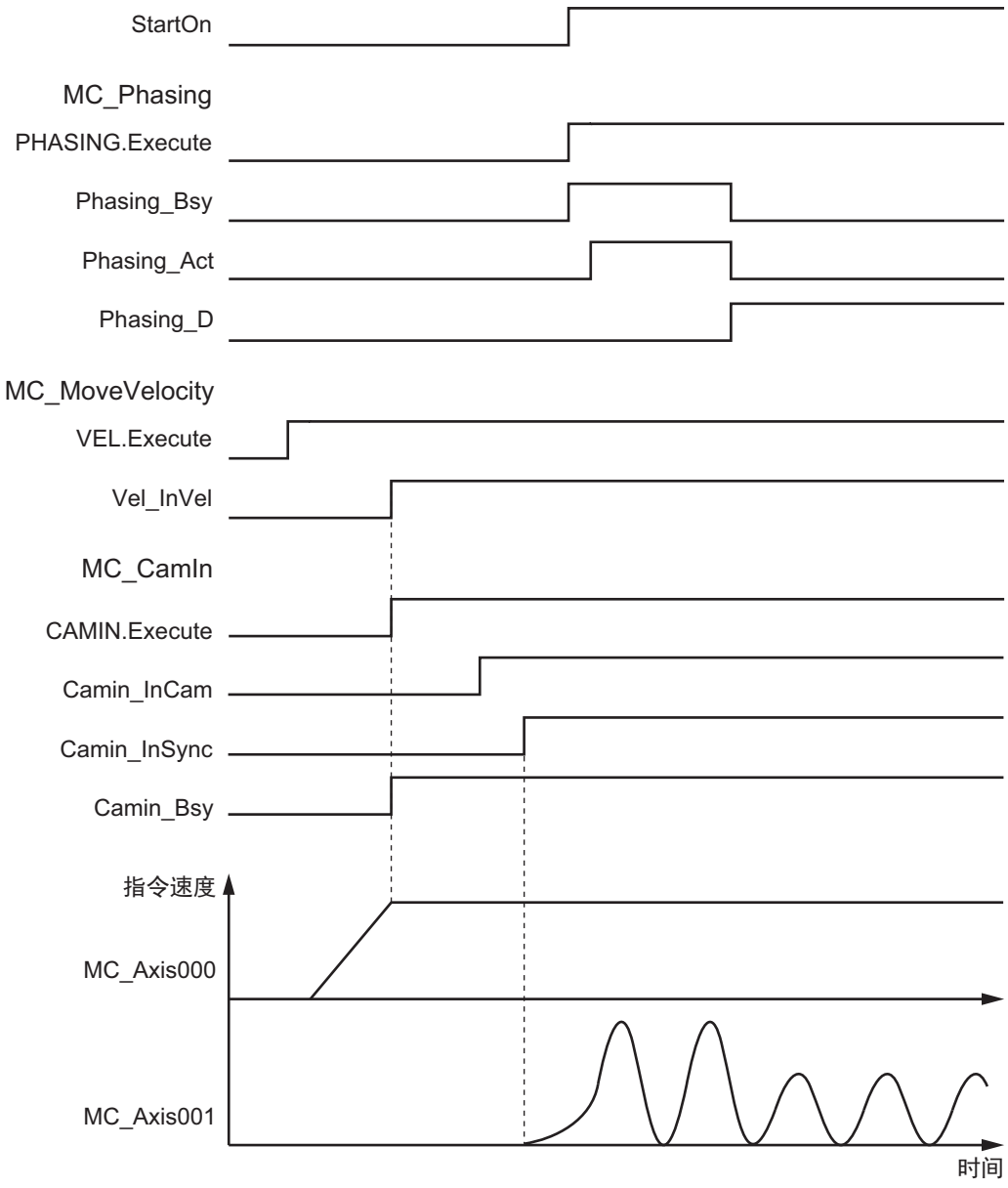
程序中使用的主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis000.Details.Homed	BOOL	FALSE	轴 0 处于原点确定状态时变为 TRUE。
MC_Axis001	_sAXIS_REF	—	轴 1 的轴变量。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	轴 1 发生轻度故障等级的异常时，变为 TRUE。
MC_Axis001.Details.Homed	BOOL	FALSE	轴 1 处于原点确定状态时变为 TRUE。
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	—	凸轮数据变量。*1
Pwr1_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR1 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。
Pwr2_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR2 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。
StartOn	BOOL	FALSE	开始主轴的相位补偿所需的变量。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 通信的过程数据通信正在正常执行，则伺服 ON。
Camin_Ex	BOOL	FALSE	启动 MC_CamIn （凸轮动作开始）指令所需的变量。 在 ST 程序中使用。
Vel_Ex	BOOL	FALSE	启动 MC_MoveVelocity （速度控制）指令所需的变量。 在 ST 程序中使用。
Phasing_Ex	BOOL	FALSE	启动 MC_Phasing （主轴相对值相位补偿）指令所需的变量。 在 ST 程序中使用。

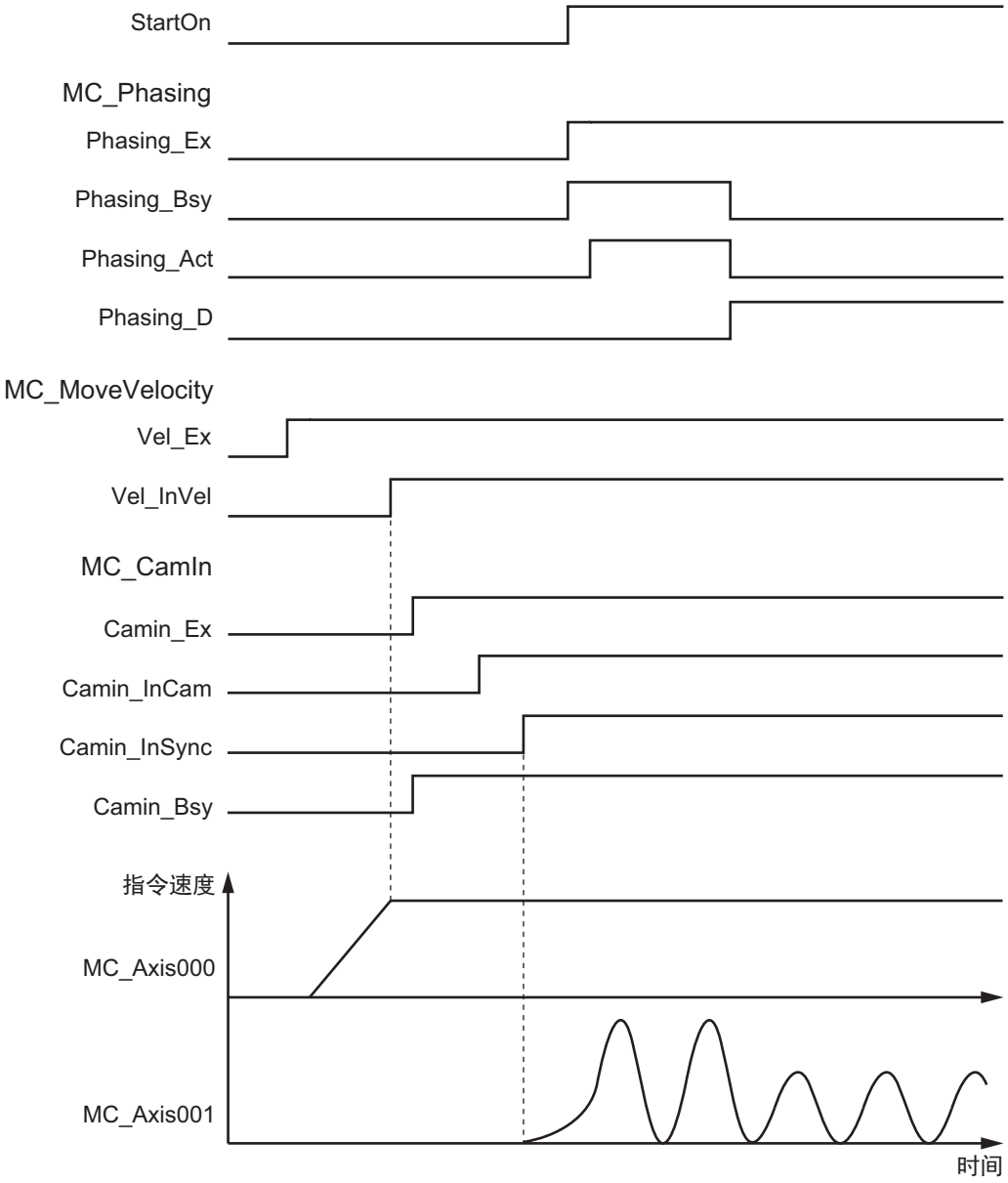
*1. 排列的要素数 ARRAY[0..N] 在 Sysmac Studio 的凸轮编辑器功能中设定。本示例程序中为 [0 ～ 360]。

时序图

● 梯形图程序时

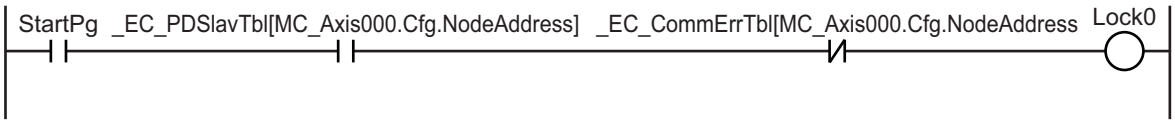

StartOn
MC_Phasing
PHASING.Execute
Phasing_Bsy
Phasing_Act
Phasing_D
MC_MoveVelocity
VEL.Execute
Vel_InVel
MC_CamIn
CAMIN.Execute
Camin_InCam
Camin_InSync
Camin_Bsy
指令速度
MC_Axis000
MC_Axis001
时间

● **ST 程序时**

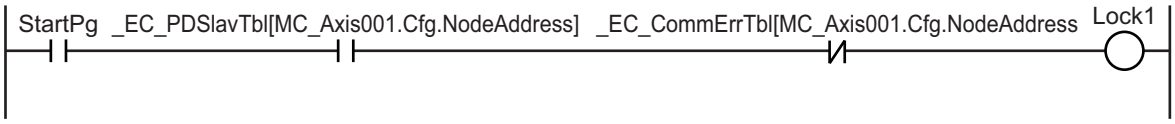

StartOn
MC_Phasing
Phasing_Ex
Phasing_Bsy
Phasing_Act
Phasing_D
MC_MoveVelocity
Vel_Ex
Vel_InVel
MC_CamIn
Camin_Ex
Camin_InCam
Camin_InSync
Camin_Bsy
指令速度
MC_Axis000
MC_Axis001
时间

梯形图程序

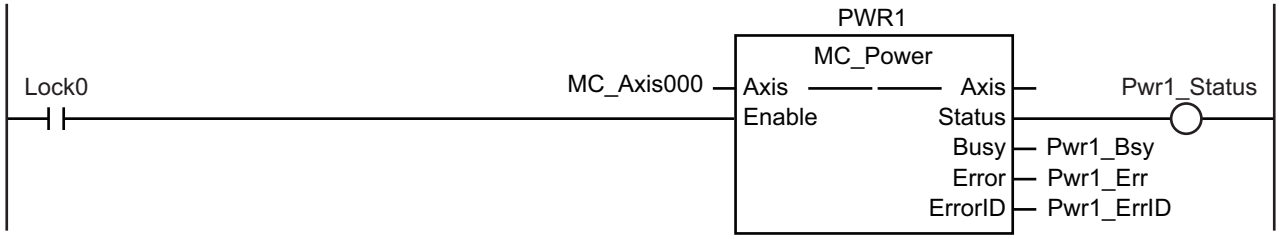
确认接点 StartPg 是否为 TRUE 且与轴 0 的过程数据通信是否正在正常执行



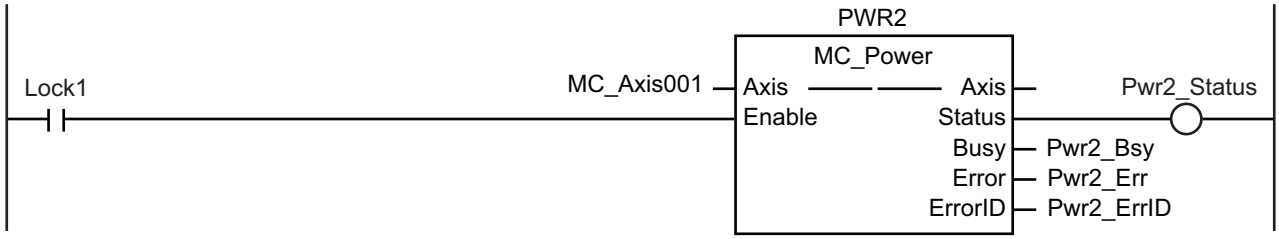
确认接点 StartPg 是否为 TRUE 且与轴 1 的过程数据通信是否正在正常执行



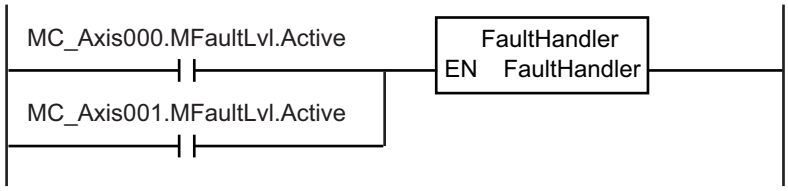
若与轴 0 的过程数据通信正在正常执行，则轴 0 变为伺服 ON



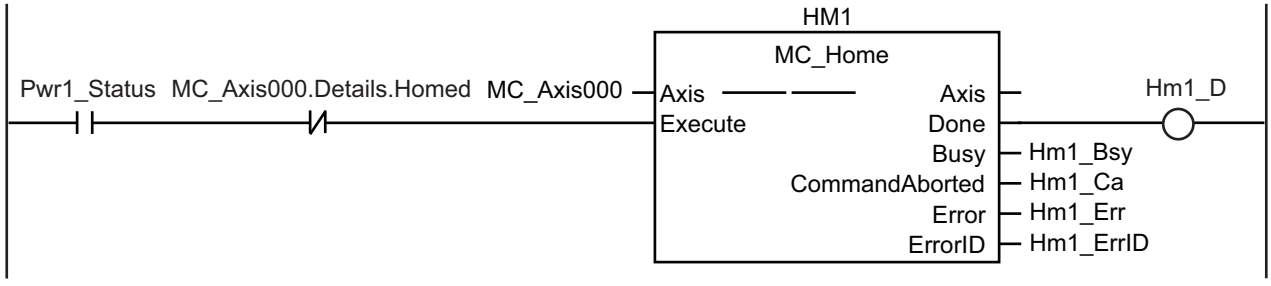
若与轴 1 的过程数据通信正在正常执行，则轴 1 变为伺服 ON



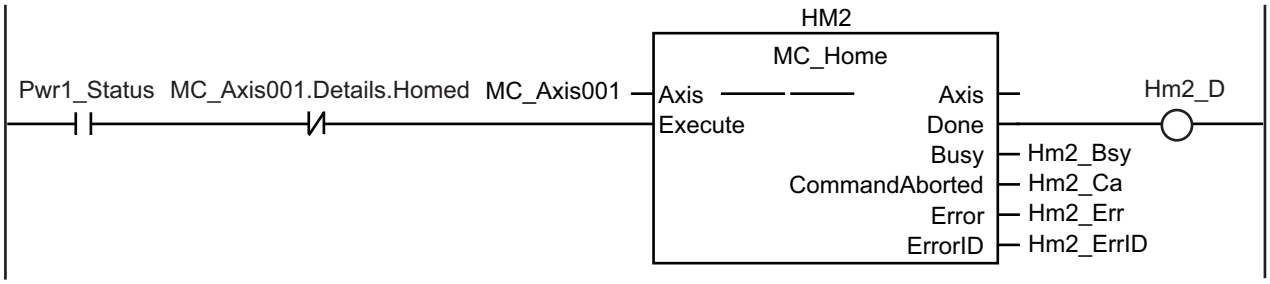
轴组或构成轴发生轻度故障等级的异常时，执行异常处理 FaultHandler
异常处理的内容请根据装置进行编程



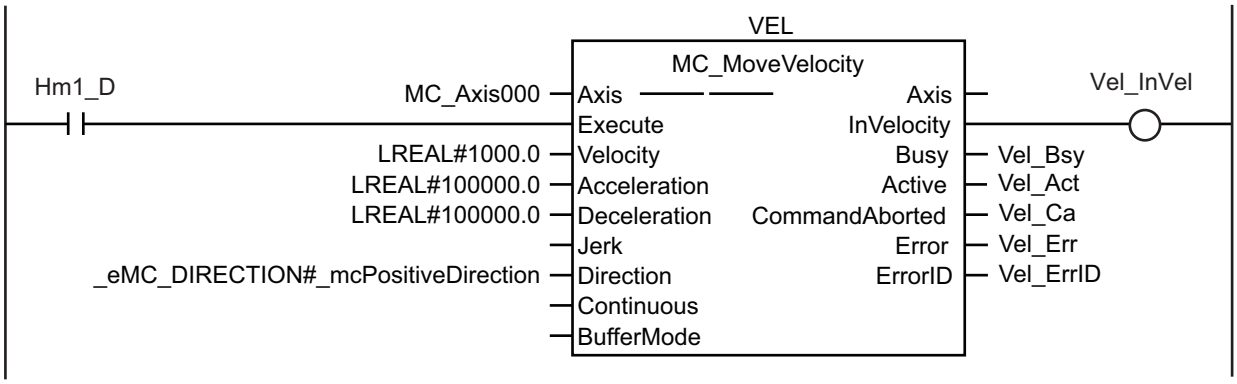
轴 0 为伺服 ON 状态且处于原点未确定状态时，执行原点复位



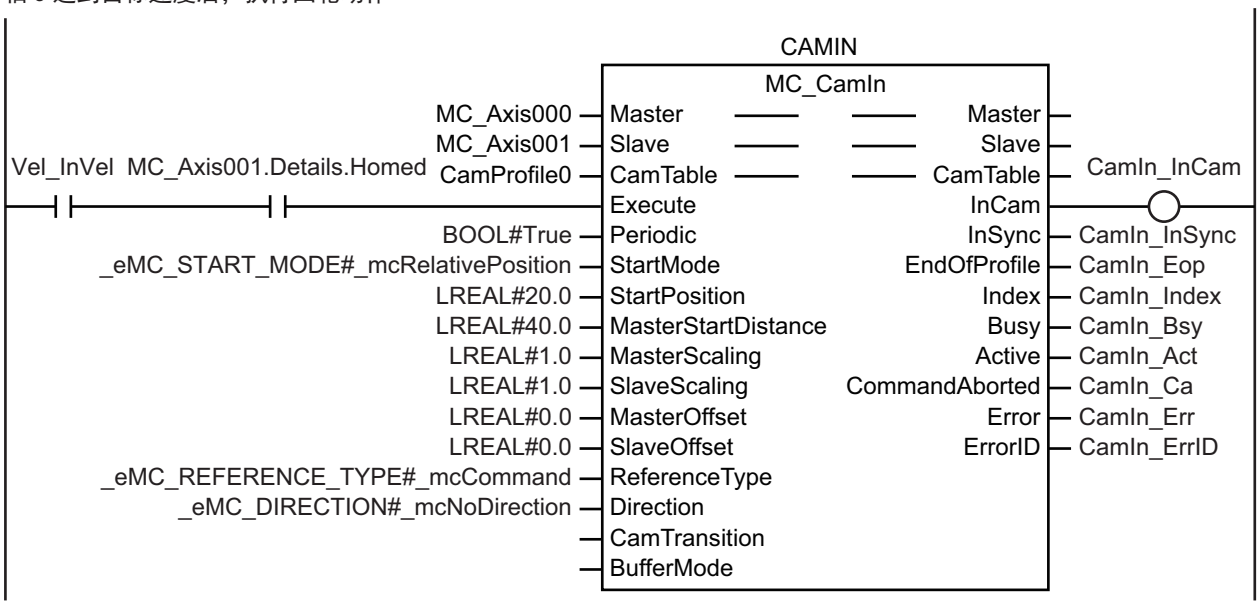
轴 1 为伺服 ON 状态且处于原点未确定状态时，执行原点复位



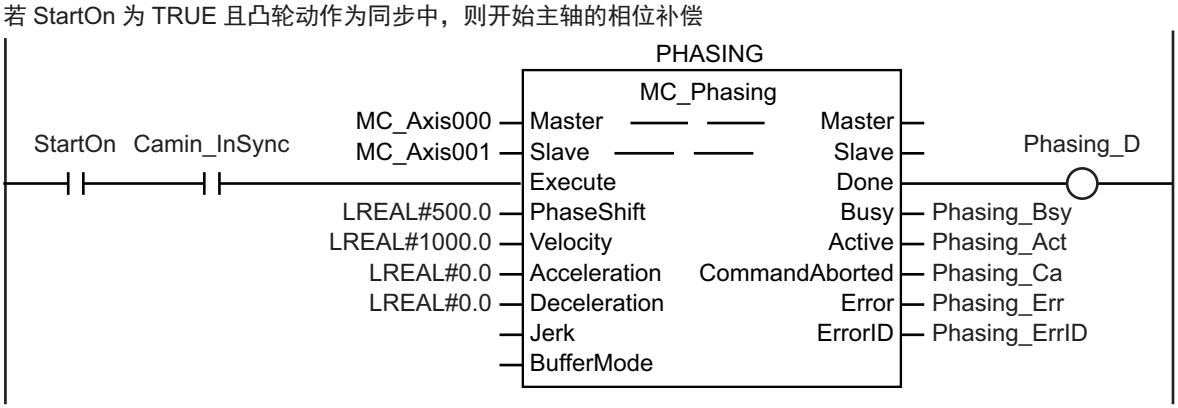
轴 0 的原点复位完成后，执行速度控制



轴 0 达到目标速度后，执行凸轮动作



若 StartOn 为 TRUE 且凸轮动作为同步中，则开始主轴的相位补偿



ST 程序

```
// 未设定动作指令的输入参数时，设定目标值等
IF InitFlag=FALSE THEN
   // 在速度控制 (MC_MoveVelocity) 的输入参数中设定值
   Vel_Vel := LREAL#1000.0;
   Vel_Acc := LREAL#100000.0;
   Vel_Dec := LREAL#100000.0;
   Vel_Dir := _eMC_DIRECTION#_mcPositiveDirection;

   // 在主轴相对值相位补偿 (MC_Phasing) 的输入参数中设定值
   Phasing_Ps := LREAL#500.0;
   Phasing_Vel := LREAL#1000.0;
   Phasing_Acc := LREAL#0.0;
   Phasing_Dec := LREAL#0.0;

   // 在凸轮动作开始 (MC_CamIn) 的输入参数中设定值
   Camin_Em := TRUE;
   Camin_Sm := _eMC_START_MODE#_mcRelativePosition;
   Camin_Sp := LREAL#20.0;
   Camin_Msd := LREAL#40.0;
   Camin_Ms := LREAL#1.0;
   Camin_Ss := LREAL#1.0;
   Camin_Mo := LREAL#0.0;
   Camin_So := LREAL#0.0;
   Camin_Rt := _eMC_REFERENCE_TYPE#_mcCommand;
   Camin_Dir := _eMC_DIRECTION#_mcNoDirection;

    // 将输入参数设定完成标志设为有效
   InitFlag := TRUE;
END_IF;


// 若接点 StartPg 为 TRUE 且过程数据通信正在正常执行，则轴 0 变为伺服 ON
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
   Pwr1_En:=TRUE;
ELSE
   Pwr1_En:=FALSE;
END_IF;


// 若接点 StartPg 为 TRUE 且过程数据通信正在正常执行，则轴 1 变为伺服 ON
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
   Pwr2_En:=TRUE;
ELSE
   Pwr2_En:=FALSE;
END_IF;


// 轴 0 ～轴 1 发生轻度故障等级的异常时，执行异常处理 FaultHandler
// 异常处理的内容请根据装置进行编程
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
   FaultHandler();
END_IF;
```

```
// 轴 0 为伺服 ON 状态且处于原点未确定状态时，执行轴 0 的原点复位
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
   Hm1_Ex:=TRUE;
END_IF;


// 轴 1 为伺服 ON 状态且处于原点未确定状态时，执行轴 1 的原点复位
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
   Hm2_Ex:=TRUE;
END_IF;


// 轴 0 的原点复位完成后，执行速度控制
IF Hm1_D=TRUE THEN
   Vel_Ex :=TRUE;
END_IF;


// 轴 0 达到目标速度且轴 1 的原点确定后，执行凸轮动作
IF (Vel_InVel=TRUE) AND (MC_Axis001.Details.Homed=TRUE) THEN
   Camin_Ex := TRUE;
END_IF;


// 若 StartOn 为 TRUE 且凸轮动作为同步中，则开始主轴的相位补偿
IF (StartOn=TRUE) AND (Camin_InSync=TRUE) THEN
   Phasing_Ex :=TRUE;
END_IF;

// 轴 0 的 MC_Power
PWR1(
   Axis := MC_Axis000,
   Enable := Pwr1_En,
   Status => Pwr1_Status,
   Busy => Pwr1_Bsy,
   Error => Pwr1_Err,
   ErrorID => Pwr1_ErrID
);

// 轴 1 的 MC_Power
PWR2(
   Axis := MC_Axis001,
   Enable := Pwr2_En,
   Status => Pwr2_Status,
   Busy => Pwr2_Bsy,
   Error => Pwr2_Err,
   ErrorID => Pwr2_ErrID
);

// 轴 0 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 1 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_Phasing
PHASING(
    Master := MC_Axis000,
    Slave := MC_Axis001,
    Execute := Phasing_Ex,
    PhaseShift := Phasing_Ps,
    Velocity := Phasing_Vel,
    Acceleration := Phasing_Acc,
    Deceleration := Phasing_Dec,
    Done => Phasing_D,
    Busy => Phasing_Bsy,
    Active => Phasing_Act,
    CommandAborted => Phasing_Ca,
    Error => Phasing_Err,
    ErrorID => Phasing_ErrID
);

//MC_CamIn
CAMIN(
    Master := MC_Axis000,
    Slave := MC_Axis001,
    CamTable := CamProfile0,
    Execute := Camin_Ex,
    Periodic := Camin_Em,
    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
);
```

10-2-15 在速度控制过程中变更当前位置

在速度控制中变更动作中的轴的指令当前位置和反馈当前位置的绝对位置。

使用注意事项

- 若对动作中的轴执行 MC_SetPosition （当前位置变更）指令，从启动起到实际变更当前位置为止的移动距离就是误差。
- 若启动 MC_SetPosition （当前位置变更）指令，将变为原点未确定。

轴参数设定

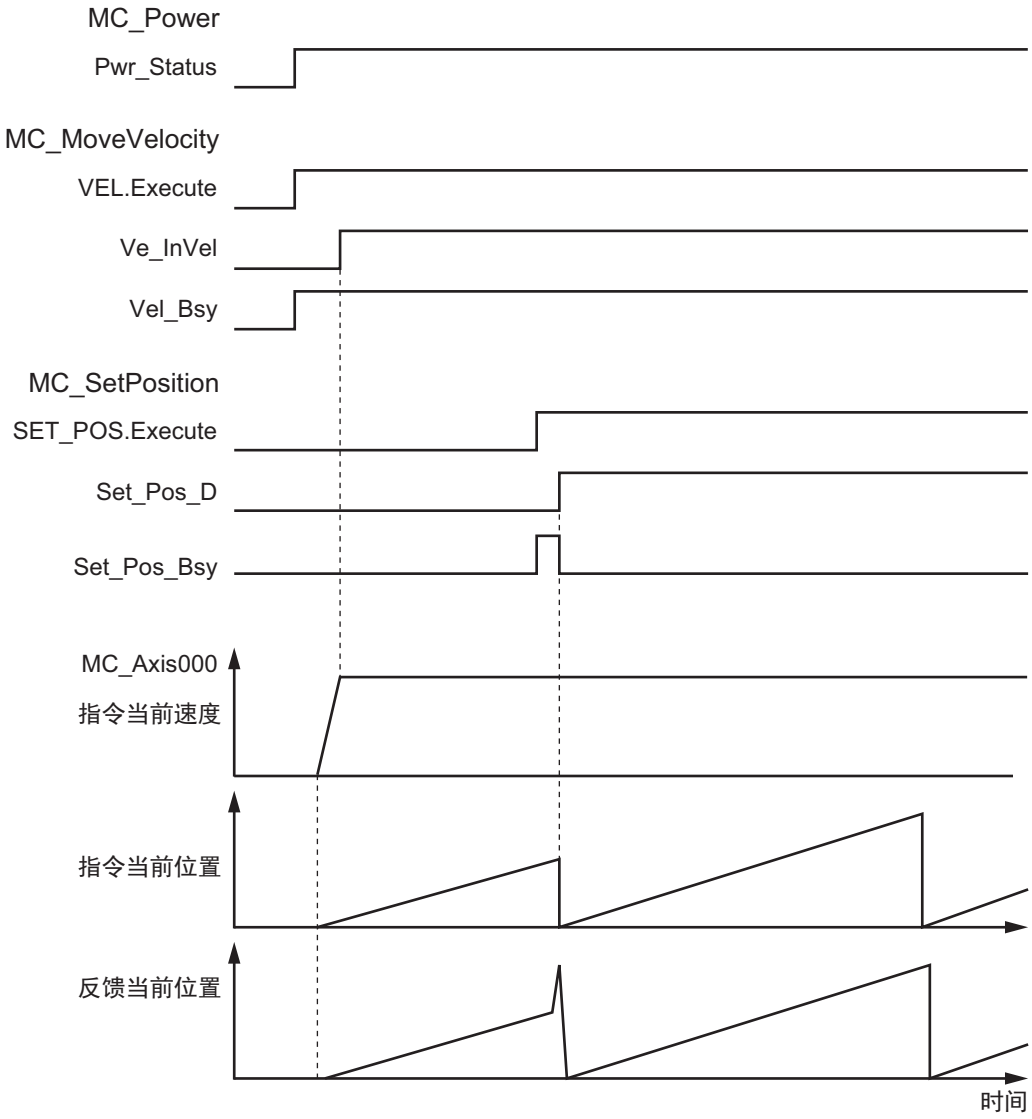
设定项目	设定内容	说明
计数模式	旋转模式	将位置的计数模式设定为旋转模式。
循环计数器上限设定值	360	将循环计数器的上限值设定为 “360”。
循环计数器下限设定值	0	将循环计数器的下限值设定为 “0”。
原点复位动作	原点预设	通过原点预设确定原点。

程序中使用的主要变量

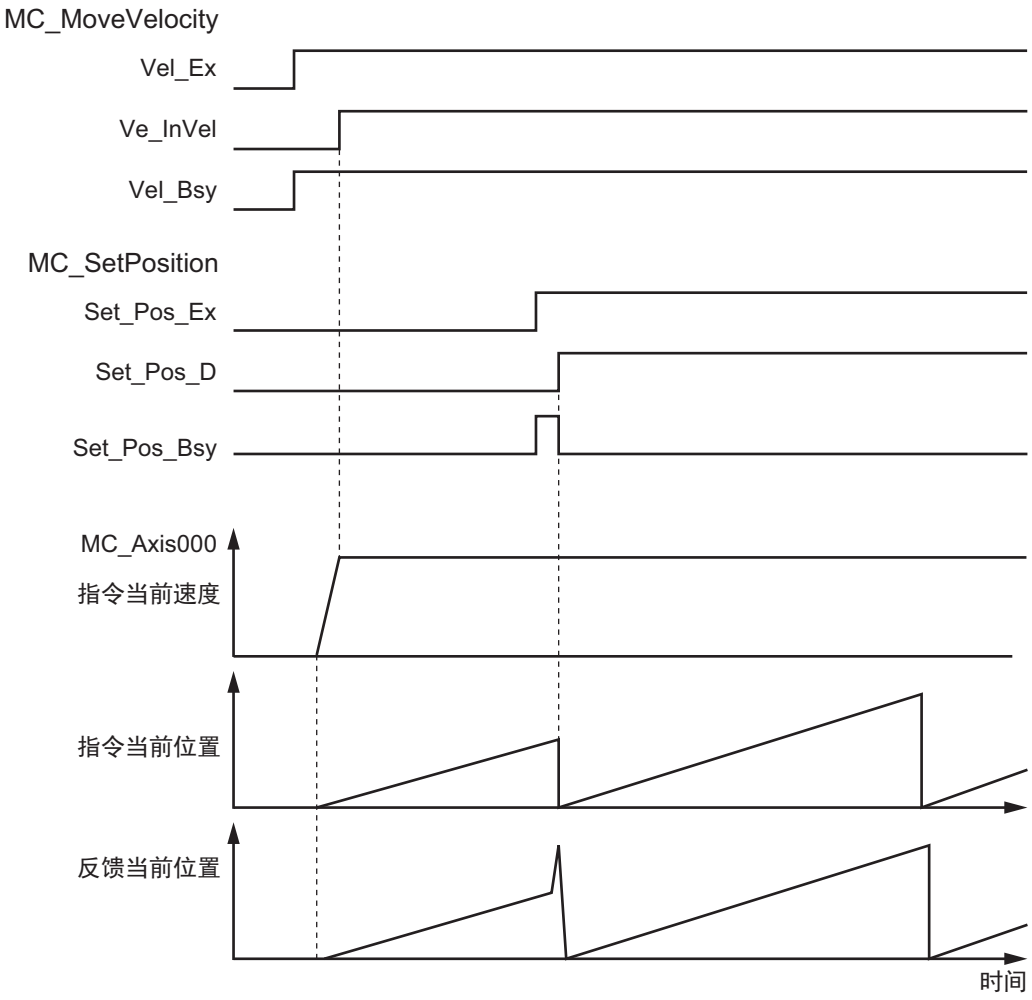
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	轴 0 发生轻度故障等级的异常时，变为 TRUE。
Pwr_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR 的输出 Status 中的变量。 伺服 ON 时变为 TRUE。
StartSetPos	BOOL	FALSE	变更当前位置所需的外部按钮。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 通信的过程数据通信正在正常执行，则伺服 ON。
Vel_Ex	BOOL	FALSE	启动 MC_MoveVelocity （速度控制）指令所需的变量。 在 ST 程序中使用。
SetPos_Ex	BOOL	FALSE	启动 MC_SetPosition （当前位置变更）指令所需的变量。 在 ST 程序中使用。

时序图

● 梯形图程序时

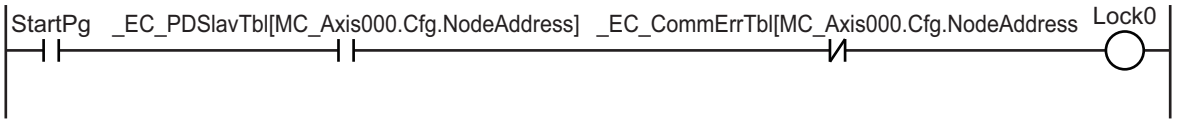

MC_Power
Pwr_Status
MC_MoveVelocity
VEL.Execute
Ve_InVel
Vel_Bsy
MC_SetPosition
SET_POS.Execute
Set_Pos_D
Set_Pos_Bsy
MC_Axis000
指令当前速度
指令当前位置
反馈当前位置
时间

- **ST 程序时**

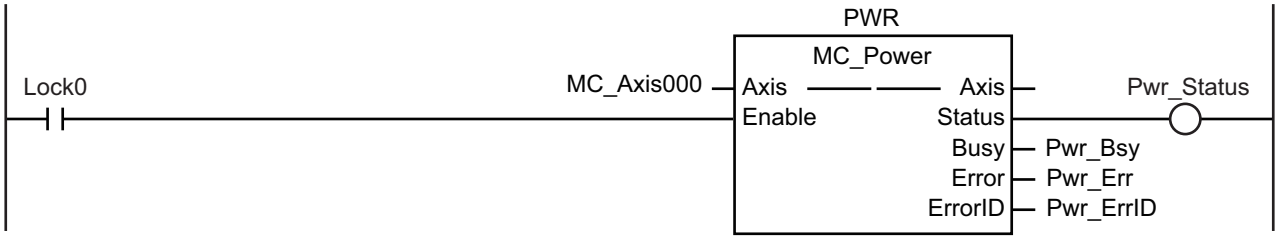

MC_MoveVelocity
Vel_Ex
Ve_InVel
Vel_Bsy
MC_SetPosition
Set_Pos_Ex
Set_Pos_D
Set_Pos_Bsy
MC_Axis000
指令当前速度
指令当前位置
反馈当前位置
时间

梯形图程序

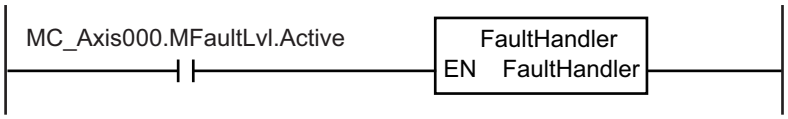
确认接点 StartPg 是否为 TRUE 且与轴 0 的过程数据通信是否正在正常执行



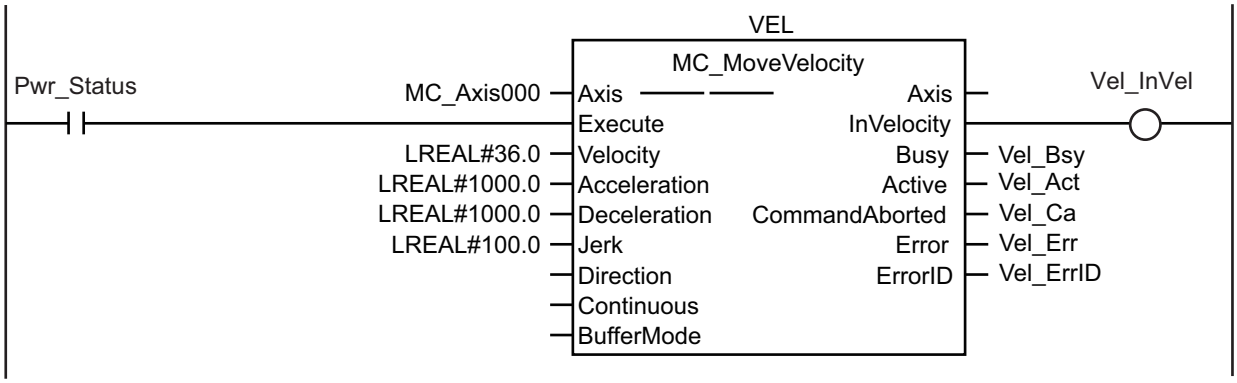
若与轴 0 的过程数据通信正在正常执行，则轴 0 变为伺服 ON



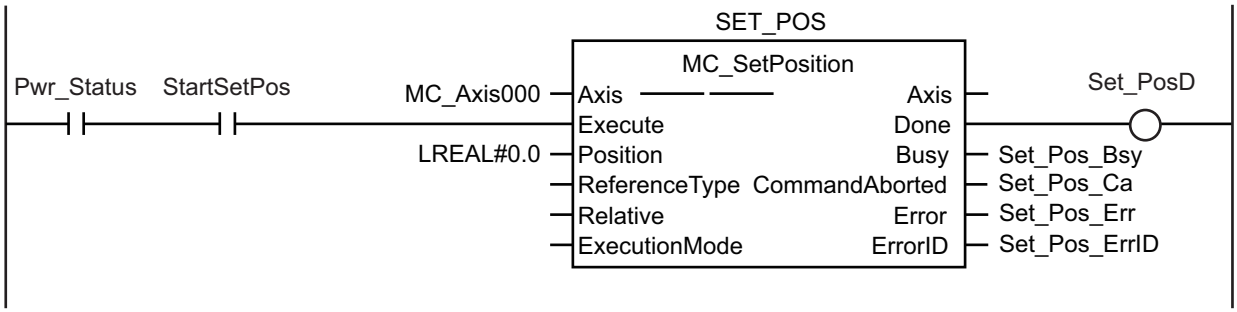
轴 0 发生轻度故障等级的异常时，执行异常处理 FaultHandler
异常处理的内容请根据装置进行编程



若轴 0 变为伺服 ON，则执行速度控制



若在伺服 ON 状态下接点 StartSetPos 变为 TRUE，则执行当前位置变更



ST 程序

```
// 未设定指令的输入参数时，设定目标值等
IF InitFlag=FALSE THEN
   // 在速度控制 (MC_MoveVelocity) 的输入参数中设定值
   Vel_Vel := LREAL#36.0;
   Vel_Acc := LREAL#1000.0;
   Vel_Dec := LREAL#1000.0;
   Vel_Jrk := LREAL#100.0;

   // 在当前位置变更 (MC_SetPosition) 的输入参数中设定值
   Set_Pos_Pos := LREAL#0.0;

   // 将输入参数设定完成标志设为有效
   InitFlag :=TRUE;
END_IF;


// 若接点 StartPg 为 TRUE 且与轴 0 的过程数据通信正在正常执行，则轴 0 变为伺服 ON
// 若未进行过程数据通信，则变为伺服 OFF
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
   Pwr_En :=TRUE;
ELSE
   Pwr_En :=FALSE;
END_IF;


// 轴 0 发生轻度故障等级的异常时，执行异常处理 FaultHandler
// 异常处理的内容请根据装置进行编程
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
   FaultHandler();
END_IF;


// 若轴 0 变为伺服 ON，则执行轴 0 的速度控制
IF Pwr_Status=TRUE THEN
   Vel_Ex :=TRUE;
END_IF;


// 若在伺服 ON 状态下接点 StartSetPos 变为 TRUE，则执行当前位置变更
IF (Pwr_Status=TRUE) AND (StartSetPos=TRUE) THEN
   Set_Pos_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_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_SetPosition
SET_POS(
    Axis := MC_Axis000,
    Execute := Set_Pos_Ex,
    Position := Set_Pos_Pos,
    Done => Set_Pos_D,
    Busy => Set_Pos_Bsy,
    CommandAborted => Set_Pos_Ca,
    Error => Set_Pos_Err,
    ErrorID => Set_Pos_ErrID
);
```

10-2-16 凸轮数据变量的变更和凸轮表的保存

在用户程序上变更 Sysmac Studio 的凸轮编辑器中创建的凸轮数据变量的值。
相位为 0 度～ 180 度时，变位乘以 2， 181 度～ 360 度时，变位乘以 0.5。

凸轮数据变更完成后，利用 MC_SaveCamTable （凸轮表保存）指令保存到 CPU 单元的非易失性存储器中。
保存完成后，利用 MC_CamIn （凸轮动作开始）指令开始凸轮动作。

使用注意事项

- 相位不是升序时，启动 MC_CamIn （凸轮动作开始）指令时将发生异常。
  本示例程序中，不会检查相位是否为升序。若要检查，请利用 MC_SetCamTableProperty （凸轮表属性更新）指令执行。
- CPU 单元的非易失性存储器有写入次数限制。请仅在需要时保存凸轮表。
- 若在用 MC_SaveCamTable （凸轮表保存）指令保存前关闭 CPU 单元的电源，用户程序中变更的凸轮数据变量将恢复为变更前的内容。

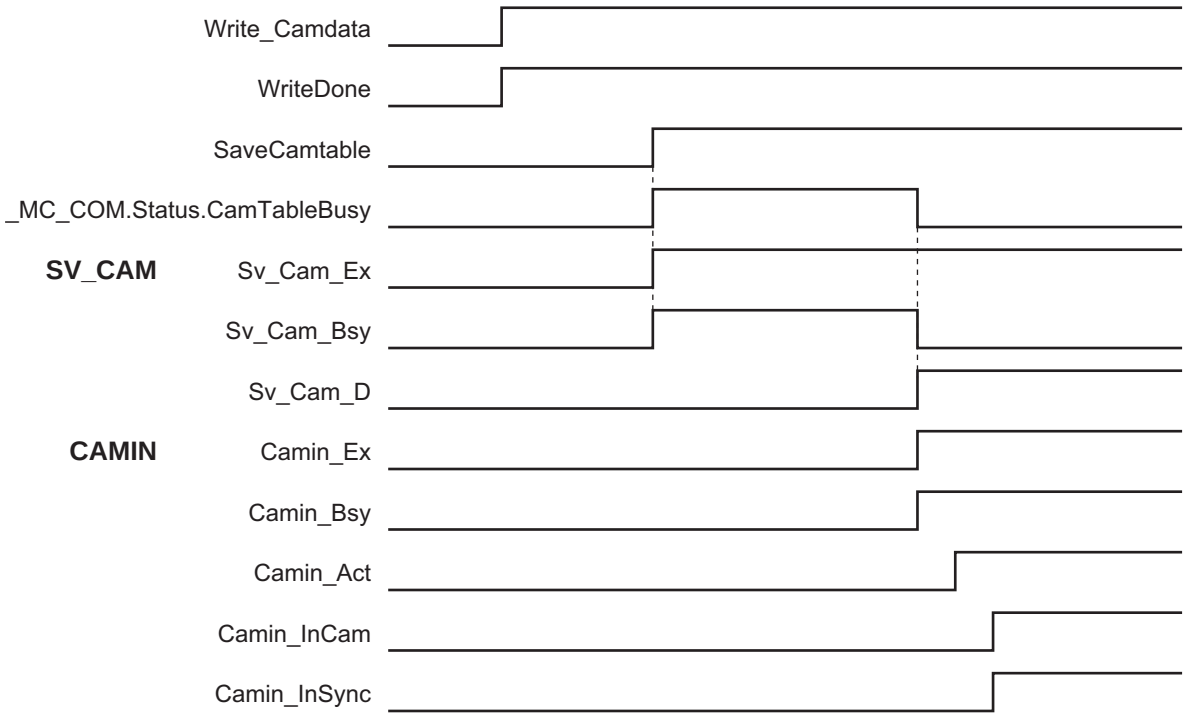
程序中使用的主要变量

名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
MC_Axis001	_sAXIS_REF	—	轴 1 的轴变量。
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	—	凸轮数据变量。*1
Pwr1_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR1 的输出 Status 中的变量。伺服 ON 时变为 TRUE。
Pwr2_Status	BOOL	FALSE	分配到 MC_Power 指令的实例 PWR2 的输出 Status 中的变量。伺服 ON 时变为 TRUE。
StartPg	BOOL	FALSE	若该变量为 TRUE 且 EtherCAT 通信的过程数据通信正在正常执行，则伺服 ON。
WriteCamdata	BOOL	FALSE	启动凸轮数据变更所需的变量。开始编辑时请设为 TRUE。
WriteDone	BOOL	FALSE	表示凸轮数据的变更已完成的变量。变更完成后，请设为 TRUE。
SaveCamtable	BOOL	FALSE	启动凸轮表保存所需的变量。
_MC_COM.Status.CamTableBusy	BOOL	FALSE	表示正在执行凸轮表保存的系统定义变量。
Sv_Cam_Ex	BOOL	FALSE	启动 MC_SaveCamTable （凸轮表保存）指令所需的变量。
Camin_Ex	BOOL	FALSE	启动 MC_CamIn （凸轮动作开始）指令所需的变量。在 ST 程序中使用。

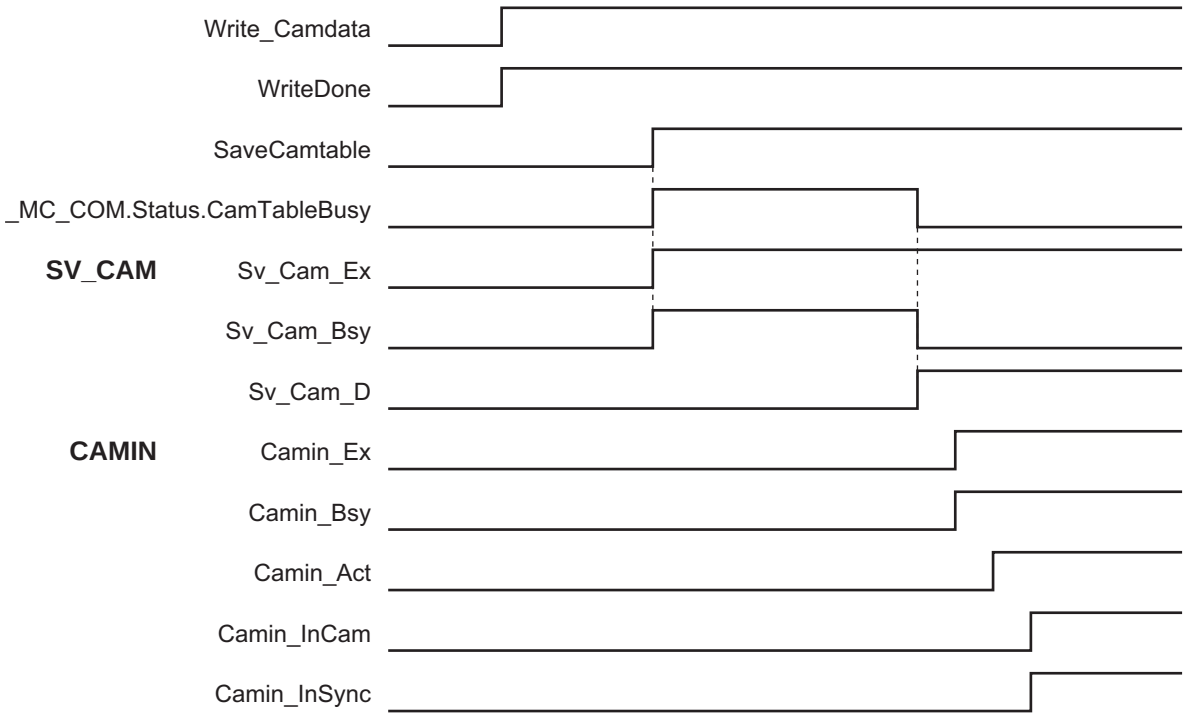
*1. 排列的要素数 ARRAY[0..N] 在 Sysmac Studio 的凸轮编辑器功能中设定。本示例程序中为 [0 ～ 360]。

时序图

● 梯形图程序时

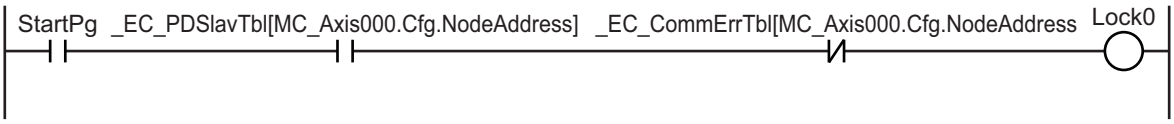


● ST 程序时

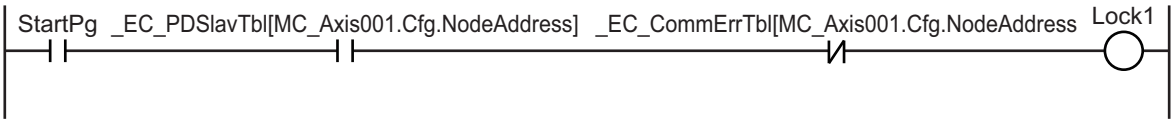


梯形图程序

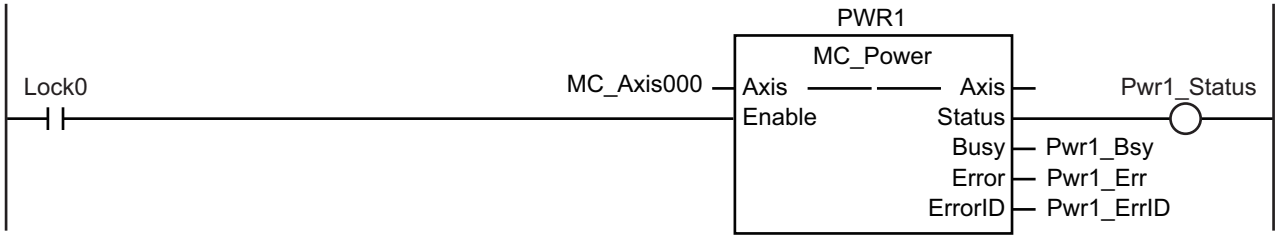
确认接点 StartPg 是否为 TRUE 且与轴 0 的过程数据通信是否正在正常执行



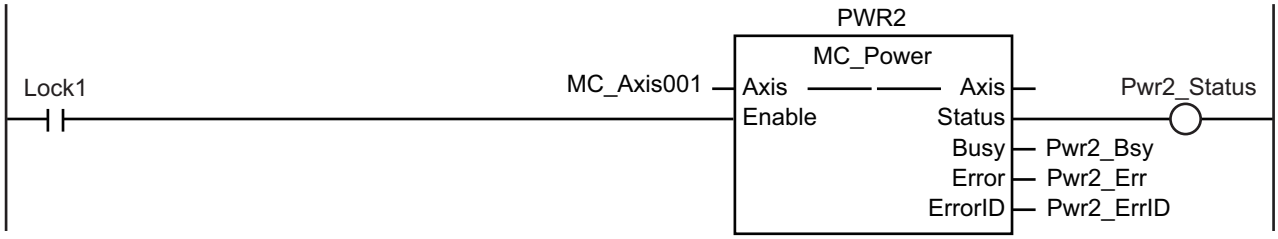
确认接点 StartPg 是否为 TRUE 且与轴 1 的过程数据通信是否正在正常执行



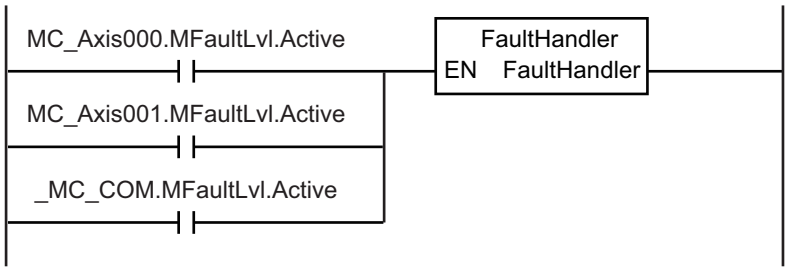
若与轴 0 的过程数据通信正在正常执行，则轴 0 变为伺服 ON



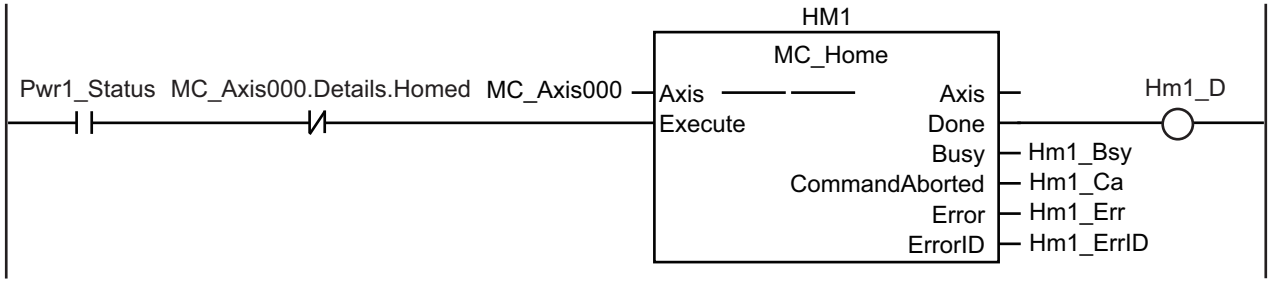
若与轴 1 的过程数据通信正在正常执行，则轴 1 变为伺服 ON



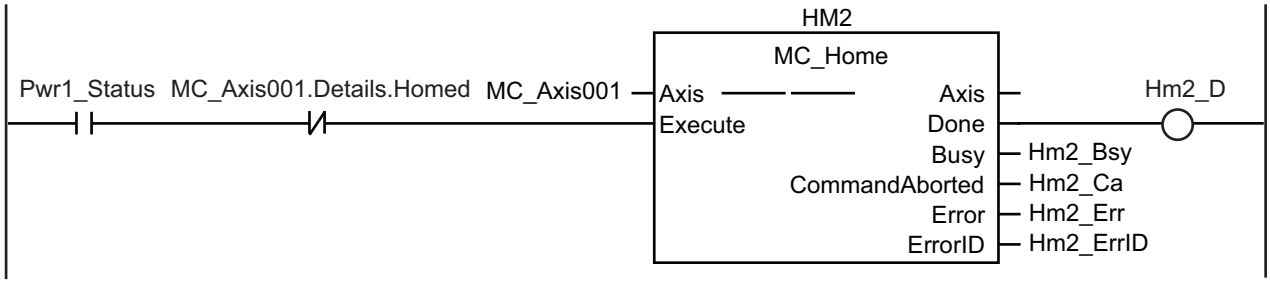
发生 MC 共通或轴的轻度故障等级异常时，执行异常处理 FaultHandler
异常处理的内容请根据装置进行编程



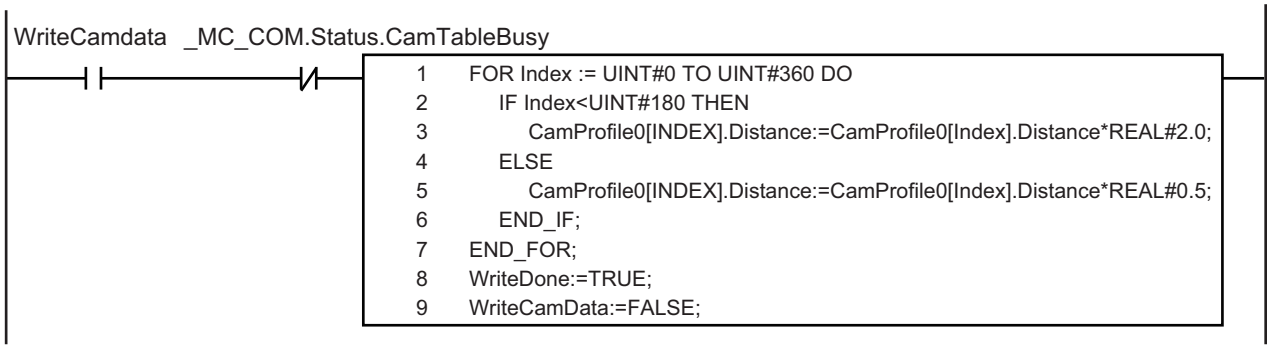
轴 0 为伺服 ON 状态且处于原点未确定状态时，执行原点复位



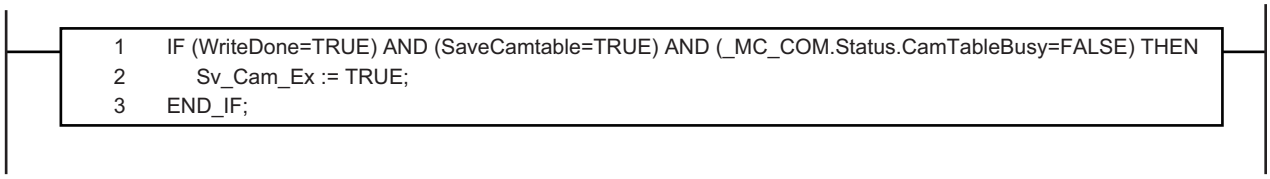
轴 1 为伺服 ON 状态且处于原点未确定状态时，执行原点复位



WriteCamData 为 TRUE 且正在执行凸轮表文件保存时，变更凸轮数据变量的值
相位为 0 度～ 180 度时，变为乘以 2，181 度～ 360 度时，变为乘以 0.5
变更完成后，将 WriteDone 设为 TRUE



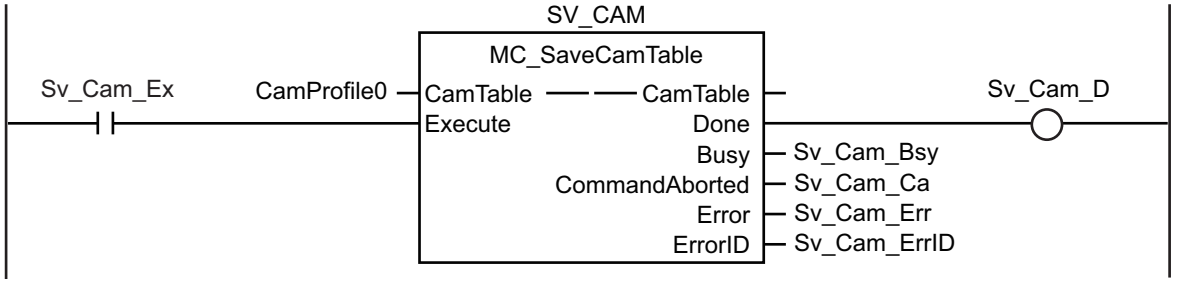
凸轮数据变量变更完成、SaveCamtable 为 TRUE 且凸轮表文件保存非执行中时，将 Sv_Cam_Ex 设为 TRUE



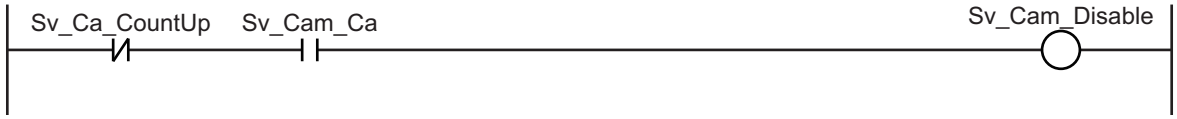
Sv_Ca_TimeUp 为 TRUE 时，将 Sv_Cam_Ex 设为 FALSE
Sv_Cam_Ex 变为 FALSE 后，Sv_Ca_TimeUp 变为 FALSE，Sv_Cam_Ex 变为 TRUE
重新执行 MC_SaveCamTable（凸轮表保存）指令



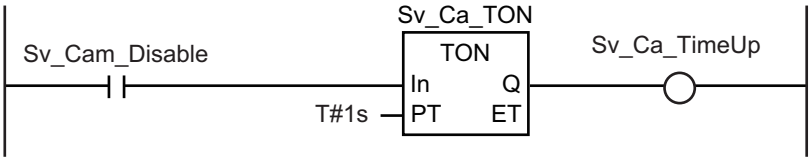
若 Sv_Cam_Ex 变为 TRUE，则执行 MC_SaveCamTable（凸轮表保存）指令



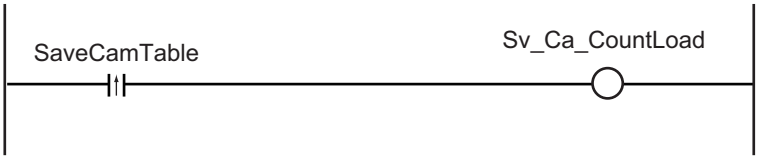
Sv_Ca_CountUp 为 FALSE 且不可执行凸轮表保存时，将 Sv_Cam_Disable 设为 TRUE



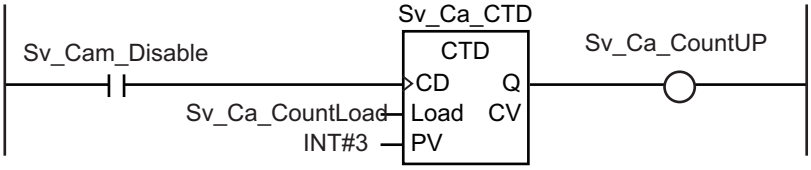
发生不可执行凸轮表保存 1 秒钟后，将 Sv_Ca_TimeUp 设为 TRUE
Sv_Ca_TimeUp 变为 TRUE 后，Sv_Cam_Ex 变为 FALSE



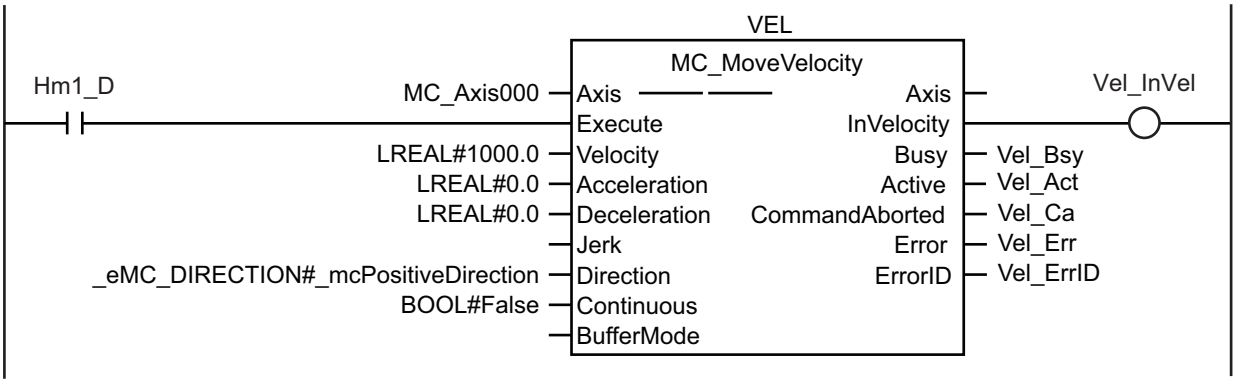
执行凸轮表保存时，在 1 个周期内将 Sv_Ca_CountLoad 设为 TRUE
若 Sv_Ca_CountLoad 变为 TRUE，则重置重试计数器



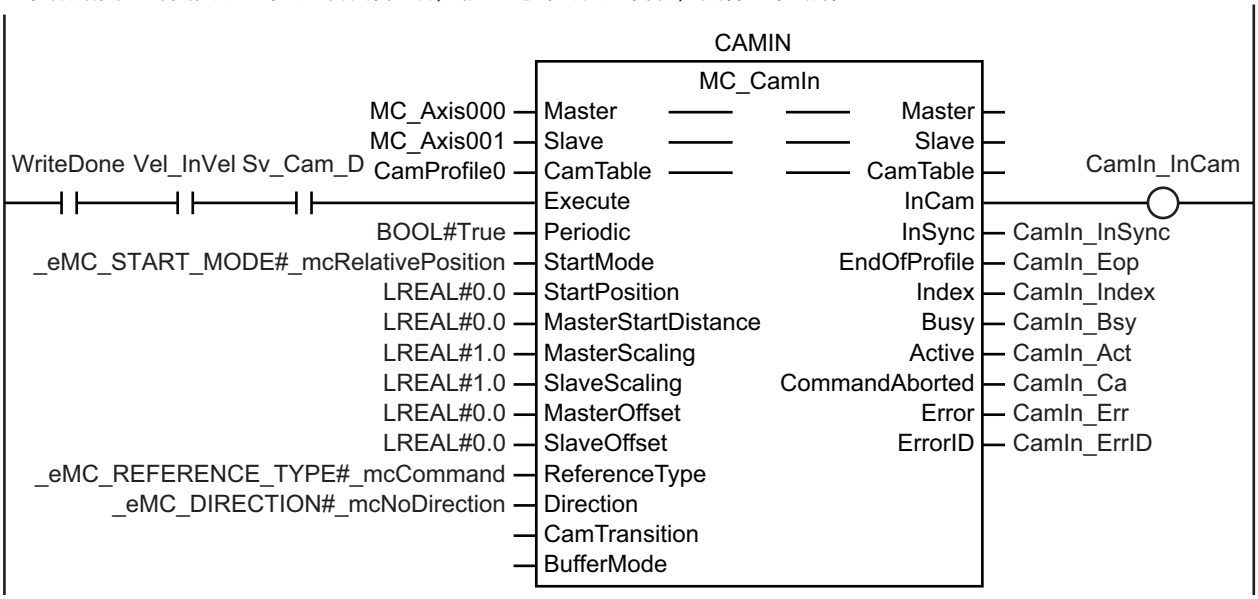
发生不可执行凸轮表保存发生 3 次后，将 Sv_Ca_CountUp 设为 TRUE
若 Sv_Ca_CountUp 变为 TRUE，则将 Sv_Cam_Disable 设为 FALSE
结束 MC_SaveCamTable（凸轮表保存）指令的重试



轴 0 的原点复位完成后，执行速度控制



凸轮数据变量的变更和凸轮表的保存完成，轴 0 达到目标速度后，执行凸轮动作



ST 程序

```
// 未设定指令的输入参数时，设定目标值等
IF InitFlag=FALSE THEN
   // 在速度控制 (MC_MoveVelocity) 的输入参数中设定值
   Vel_Vel := LREAL#1000.0;
   Vel_Acc := LREAL#0.0;
   Vel_Dec := LREAL#0.0;
   Vel_Dir := _eMC_DIRECTION#_mcPositiveDirection;

   // 在凸轮动作开始 (MC_CamIn) 的输入参数中设定值
   Camin_Em := TRUE;
   Camin_Sm := _eMC_START_MODE#_mcRelativePosition;
   Camin_Sp := LREAL#0.0;
   Camin_Msd := LREAL#0.0;
   Camin_Ms := LREAL#1.0;
   Camin_Ss := LREAL#1.0;
   Camin_Mo := LREAL#0.0;
   Camin_So := LREAL#0.0;
   Camin_Rt := _eMC_REFERENCE_TYPE#_mcCommand;
   Camin_Dir := _eMC_DIRECTION#_mcNoDirection;

   // 将输入参数设定完成标志设为有效
   InitFlag: =TRUE;
END_IF;


// 若接点 StartPg 为 TRUE 且与轴 0 的过程数据通信正在正常执行，则轴 0 变为伺服 ON
// 若未进行过程数据通信，则变为伺服 OFF
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
   Pwr1_En :=TRUE;
ELSE
   Pwr1_En :=FALSE;
END_IF;


// 若接点 StartPg 为 TRUE 且与轴 1 的过程数据通信正在正常执行，则轴 1 变为伺服 ON
// 若未进行过程数据通信，则变为伺服 OFF
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
   Pwr2_En :=TRUE;
ELSE
   Pwr2_En :=FALSE;
END_IF;


// 发生 MC 共通或轴的轻度故障等级的异常时，执行异常处理 FaultHandler
// 异常处理的内容请根据装置进行编程
IF (MC_Axis000.MFaultLvl.Active=TRUE)
 OR (MC_Axis001.MFaultLvl.Active=TRUE)
 OR (_MC_COM.MFaultLvl.Active=TRUE) THEN
   FaultHandler();
END_IF;


// 轴 0 为伺服 ON 状态且处于原点未确定状态时，执行轴 0 的原点复位
```

```
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
   Hm1_Ex :=TRUE;
END_IF;


// 轴 1 为伺服 ON 状态且处于原点未确定状态时，执行轴 1 的原点复位
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
   Hm2_Ex :=TRUE;
END_IF;


//WriteCamData 为 TRUE 且凸轮表文件保存非执行中时，变更凸轮数据变量的值
// 相位为 0 度～ 180 度时，变位乘以 2， 181 度～ 360 度时，变位乘以 0.5
// 变更完成时将 WriteDone 设为 TRUE
IF (WriteCamdata=TRUE) AND (_MC_COM.Status.CamTableBusy=FALSE) THEN
   FOR Index := UINT#0 TO UINT#360 DO
      IF Index<UINT#180 THEN
         CamProfile0[Index].Distance:=CamProfile0[Index].Distance*REAL#2.0;
      ELSE
         CamProfile0[Index].Distance:=CamProfile0[Index].Distance*REAL#0.5;
      END_IF;
   END_FOR;
   WriteDone :=TRUE;
   WriteCamdata :=FALSE;
END_IF;


// 轴 0 的原点复位完成后，执行速度控制
IF Hm1_D=TRUE THEN
   Vel_Ex :=TRUE;
END_IF;


// 凸轮数据变量变更完成、 SaveCamtable 为 TRUE 且凸轮表文件保存非执行中时，
// 将 Sv_Cam_Ex 设为 TRUE
// 若 Sv_Cam_Ex 变为 TRUE，则执行 MC_SaveCamTable （凸轮表保存）指令
IF (WriteDone=TRUE)
 AND (SaveCamtable=TRUE)
 AND (_MC_COM.Status.CamTableBusy=FALSE) THEN
   Sv_Cam_Ex := TRUE;
END_IF;

// Sv_Ca_TimeUp 为 TRUE 时，将 Sv_Cam_Ex 设为 FALSE
// 若 Sv_Cam_Ex 变为 FALSE，则 Sv_Ca_TimeUp 变为 FALSE，将 Sv_Cam_Ex 设为 TRUE
// 重新执行 MC_SaveCamTable （凸轮表保存）指令
IF (Sv_Cam_Ex=TRUE) AND (Sv_Ca_TimeUp=FALSE) THEN
   Sv_Cam_Ex := TRUE;
ELSE
   Sv_Cam_Ex := FALSE;
END_IF;

// Sv_Ca_CountUP 为 FALSE 且不可执行凸轮表保存时，
// 将 Sv_Cam_Disable 设为 TRUE
IF (Sv_Ca_CountUP=FALSE) AND (Sv_Cam_Ca=TRUE) THEN
   Sv_Cam_Disable := TRUE;
ELSE
   Sv_Cam_Disable := FALSE;
END_IF;
```

```
// 发生不可执行凸轮表保存 1 秒钟后，将 Sv_Ca_TimeUp 设为 TRUE
// 若 Sv_Ca_TimeUp 变为 TRUE，则 Sv_Cam_Ex 变为 FALSE
Sv_Ca_TON(
   In := Sv_Cam_Disable ,
   PT := T#1s ,
   Q => Sv_Ca_TimeUp
);


// 执行凸轮表保存时，在 1 个周期内将 Sv_Ca_CountLoad 设为 TRUE
// 若 Sv_Ca_CountLoad 变为 TRUE，则重置重试计数器
R_TRIG1(SaveCamtable, Sv_Ca_CountLoad);


// 发生不可执行凸轮表保存 3 次后，将 Sv_Ca_CountUP 设为 TRUE
// 若 Sv_Ca_CountUP 变为 TRUE，则将 Sv_Cam_Disable 设为 FALSE
// MC_SaveCamTable （凸轮表保存）指令的重试结束
Sv_Ca_CTD(
   CD := Sv_Cam_Disable ,
   LOAD := Sv_Ca_CountLoad ,
   PV := INT#3 ,
   Q => Sv_Ca_CountUP
);


// 凸轮数据变量的变更和凸轮表的保存完成，轴 0 达到目标速度后，执行凸轮动作
IF (Vel_InVel=TRUE)
 AND (WriteDone=TRUE)
 AND (Sv_Cam_D=TRUE) THEN
   Camin_Ex :=TRUE;
END_IF;

//MC_SaveCamTable
SV_CAM(
   CamTable :=CamProfile0,
   Execute := Sv_Cam_Ex,
   Done => Sv_Cam_D,
   Busy => Sv_Cam_Bsy,
   CommandAborted => Sv_Cam_Ca,
   Error => Sv_Cam_Err,
   ErrorID => Sv_Cam_ErrID
);

CAMIN(
   Master := MC_Axis000,
   Slave := MC_Axis001,
   CamTable := CamProfile0,
   Execute := Camin_Ex,
   Periodic := Camin_Em,
   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
);

// 轴 0 的 MC_Power
PWR1(
    Axis := MC_Axis000,
    Enable := Pwr1_En,
    Status => Pwr1_Status,
    Busy => Pwr1_Bsy,
    Error => Pwr1_Err,
    ErrorID => Pwr1_ErrID
);

// 轴 1 的 MC_Power
PWR2(
    Axis := MC_Axis001,
    Enable := Pwr2_En,
    Status => Pwr2_Status,
    Busy => Pwr2_Bsy,
    Error => Pwr2_Err,
    ErrorID => Pwr2_ErrID
);

// 轴 0 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 1 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
);
```


10-2-17 轴参数的临时替换

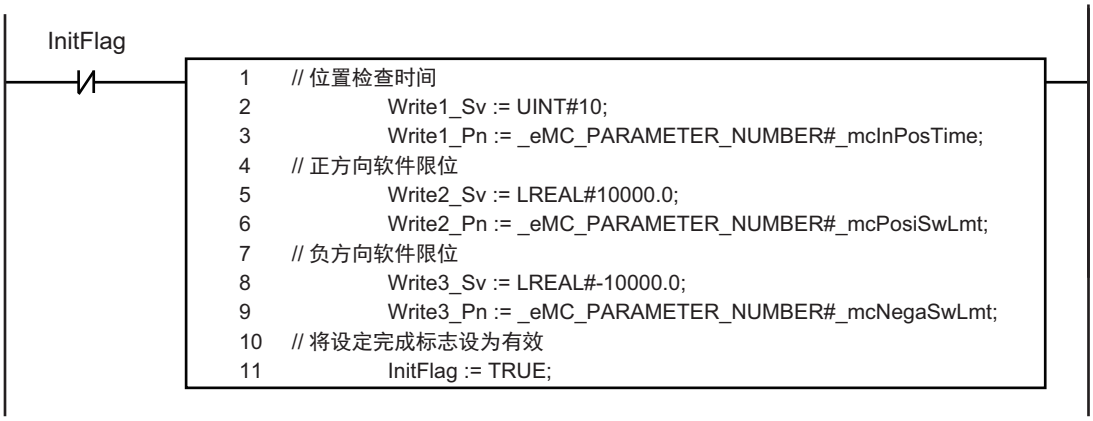
使用 MC 设定写入 (MC_Write) 指令，变更位置检查时间、正方向软件限位、负方向软件限位的值。

程序中使用的主要变量

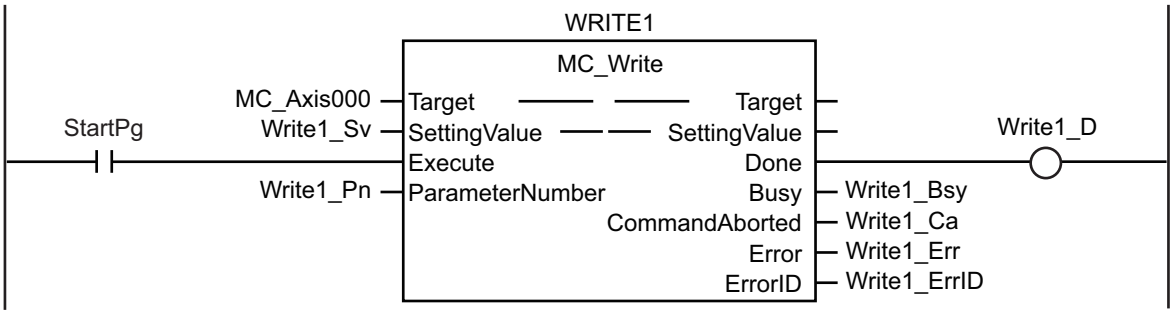
名称	数据类型	初始值	注释
MC_Axis000	_sAXIS_REF	—	轴 0 的轴变量。
InitFlag	BOOL	FALSE	表示参数设定状态的变量。 FALSE 时执行参数变更，完成后变为 TRUE。
StartPg	BOOL	FALSE	执行 MC 设定写入所需的变量。

梯形图程序

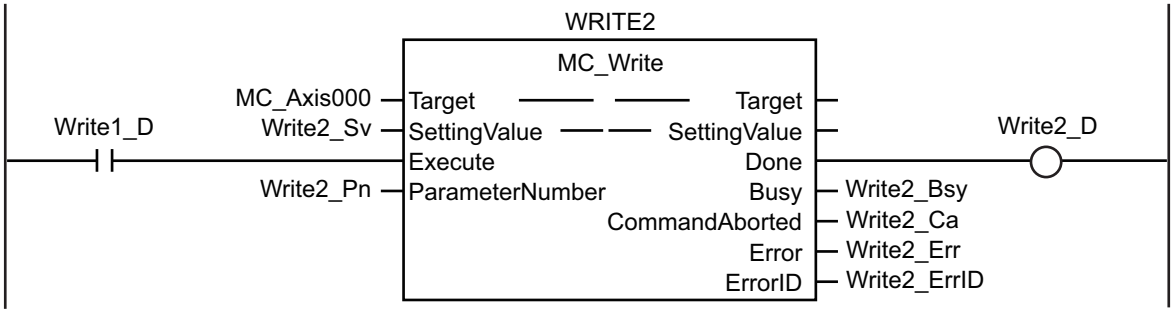
设定参数。设定完成后，将 InitFlag 设为 TRUE



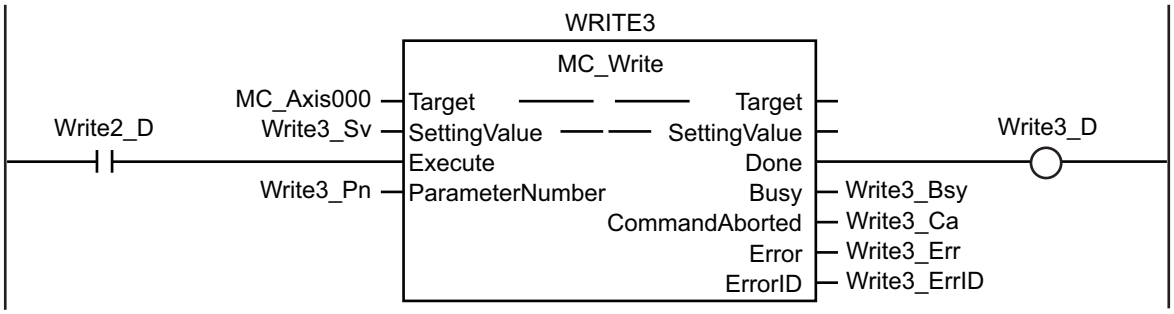
若 StartPg 变为 TRUE，则变更位置检查时间



位置检查时间的设定完成后，变更正方向软件限位



正方向软件限位的设定完成后，变更负方向软件限位



ST 程序

```
// 设定参数。设定完成后，将 InitFlag 设为 TRUE
IF InitFlag=FALSE THEN

   // 位置检查时间
      Write1_Sv := UINT#10;
      Write1_Pn := _eMC_PARAMETER_NUMBER#_mcInPosTime;

   // 正方向软件限位
      Write2_Sv := LREAL#10000.0;
      Write2_Pn := _eMC_PARAMETER_NUMBER#_mcPosiSwLmt;

   // 负方向软件限位
      Write3_Sv := LREAL#-10000.0;
      Write3_Pn := _eMC_PARAMETER_NUMBER#_mcNegaSwLmt;

   // 将输入参数设定完成标志设为有效
   InitFlag := TRUE;
END_IF;


// 若 StartPg 变为 TRUE，则变更位置检查时间
IF StartPg =TRUE THEN
   Write1_Ex := TRUE;
END_IF;


// 位置检查时间的变更完成后，变更正方向软件限位
IF Write1_D = TRUE THEN
   Write2_Ex := TRUE;
END_IF;


// 正方向软件限位的变更完成后，变更负方向软件限位
IF Write2_D = TRUE THEN
```

```
    Write3_Ex := TRUE;
END_IF;

// MC_Write
WRITE1(
    Target := MC_Axis000,
    SettingValue := Write1_Sv,
    Execute := Write1_Ex,
    ParameterNumber := Write1_Pn,
    Done => Write1_D,
    Busy => Write1_Bsy,
    CommandAborted => Write1_Ca,
    Error => Write1_Err,
    ErrorID => Write1_ErrID
);

WRITE2(
    Target := MC_Axis000,
    SettingValue := Write2_Sv,
    Execute := Write2_Ex,
    ParameterNumber := Write2_Pn,
    Done => Write2_D,
    Busy => Write2_Bsy,
    CommandAborted => Write2_Ca,
    Error => Write2_Err,
    ErrorID => Write2_ErrID
);

WRITE3(
    Target := MC_Axis000,
    SettingValue := Write3_Sv,
    Execute := Write3_Ex,
    ParameterNumber := Write3_Pn,
    Done => Write3_D,
    Busy => Write3_Bsy,
    CommandAborted => Write3_Ca,
    Error => Write3_Err,
    ErrorID => Write3_ErrID
);
```

10-2-18 凸轮表的终点索引更新

对于最大凸轮数据数为 110 个、有效数据数为 100 个的凸轮表，增加有效数据数 10 个，并更新终点索引。

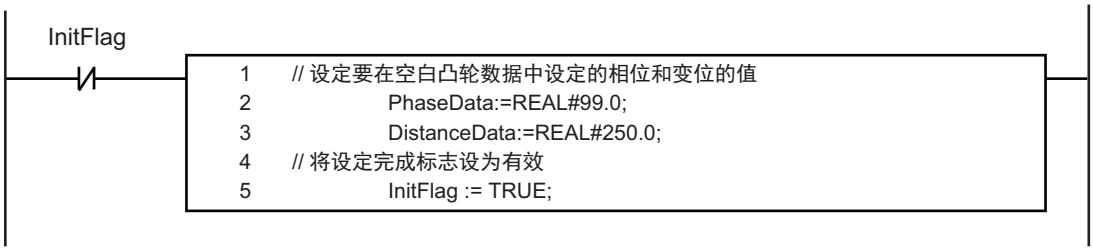
程序中使用的主要变量

名称	数据类型	初始值	注释
CamProfile0	ARRAY[0..109] OF _sMC_CAM_REF	—	最大凸轮数据数为 110 个的凸轮数据变量。*1 有效凸轮数据数为 100 个，空白凸轮数据数为 10 个。
WriteCamdata	BOOL	FALSE	启动凸轮数据变更所需的变量。 开始编辑时请设为 TRUE。
WriteDone	BOOL	FALSE	表示凸轮数据的变更已完成的变量。 变更完成后，请设为 TRUE。

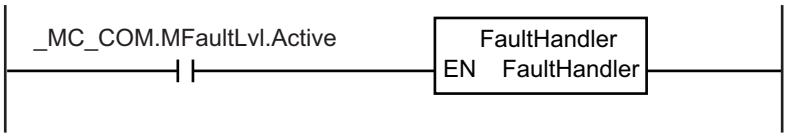
*1. 排列的要素数 ARRAY[0..N] 在 Sysmac Studio 的凸轮编辑器功能中设定。本示例程序中为 [0 ～ 109]。

梯形图程序

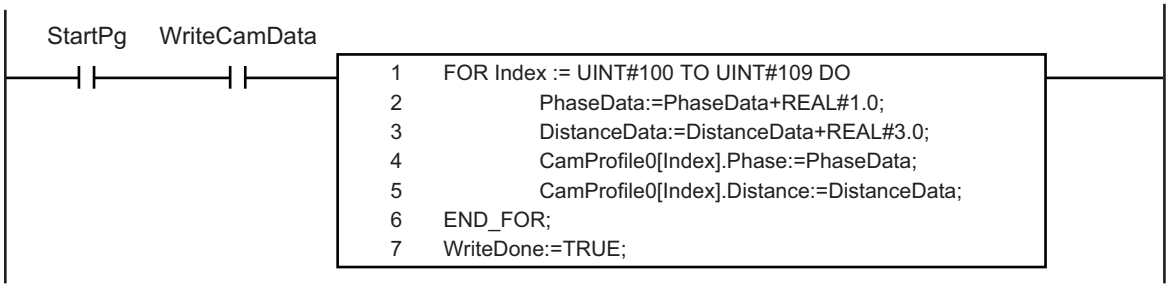
设定参数。设定完成后，将 InitFlag 设为 TRUE



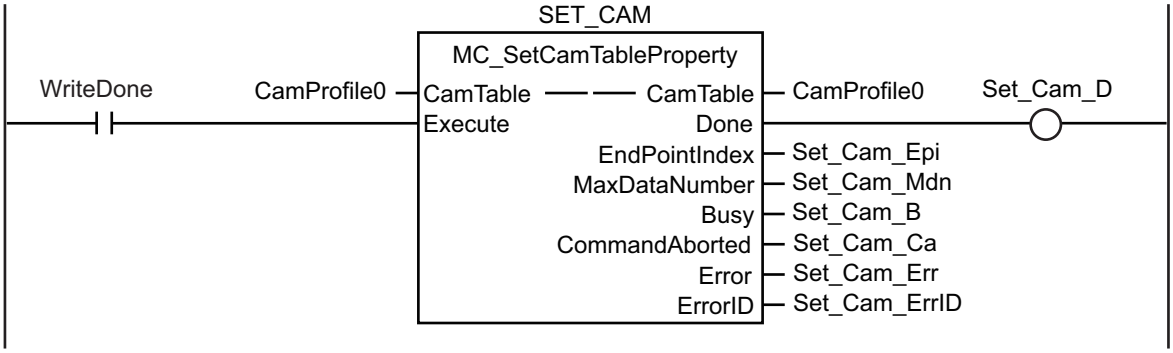
发生 MC 共通的轻度故障等级异常时，执行异常处理 FaultHandler
异常处理的内容请根据装置进行编程



若 StartPg 和 WriteCamData 变为 TRUE，则变更凸轮数据变量的值
CamProfile[100] ～ CamProfile[109] 中设定有相位和变位
凸轮数据变量的变更完成后，将 WriteDone 设为 TRUE



若凸轮数据变量的变更完成，则执行凸轮表属性更新



ST 程序

```
// 设定参数。设定完成后，将 InitFlag 设为 TRUE
IF InitFlag=FALSE THEN
   // 设定要在空白凸轮数据中设定的相位和变位的值
   PhaseData :=REAL#99.0;
   DistanceData :=REAL#250.0;

   // 将设定完成标志设为 TRUE
   InitFlag := TRUE;
END_IF;


// 发生 MC 共通的轻度故障等级的异常时，执行异常时处理
// 异常处理的内容请根据装置进行编程
IF _MC_COM.MFaultLvl.Active=TRUE THEN
   FaultHandler();
END_IF;


// 若 StartPg 和 WriteCamData 变为 TRUE，则变更凸轮数据变量的值
// 对 CamProfile[100] ～ CamProfile[109] 设定相位和变位
// 凸轮数据变量变更完成后，将 WriteDone 设为 TRUE
IF StartPg=TRUE
AND WriteCamData=TRUE THEN
   FOR Index := UINT#100 TO UINT#109 DO
      PhaseData :=PhaseData+REAL#1.0;
      DistanceData :=DistanceData+REAL#3.0;
      CamProfile0[Index].Phase :=PhaseData;
      CamProfile0[Index].Distance :=DistanceData;
   END_FOR;
   WriteDone :=TRUE;
END_IF;


// 若凸轮数据变量的变更完成，则执行凸轮表属性更新
IF WriteDone=TRUE THEN
   Set_Cam_Ex := TRUE;
END_IF;

//MC_SetCamTableProperty
SET_CAM(
   CamTable := CamProfile0,
   Execute := Set_Cam_Ex,
   Done => Set_Cam_D,
```

```
    EndPointIndex => Set_Cam_Epi,
    MaxDataNumber => Set_Cam_Mdn,
    Busy => Set_Cam_B,
    CommandAborted => Set_Cam_Ca,
    Error => Set_Cam_Err,
    ErrorID => Set_Cam_ErrID
);
```

11

异常的确认和处理

下面介绍 MC 功能模块中发生异常时的确认事项、如何根据异常显示进行异常诊断并采取对策或根据动作状态进行异常诊断并采取对策。

11-1 异常的概要 11-2
11-1-1 异常的确认方法 11-4
11-1-2 MC 功能模块相关的异常 11-7
11-2 故障排除 11-12
11-2-1 异常一览 11-12
11-2-2 异常内容 11-20
11-2-3 异常现象的推测原因和处理措施 11-52

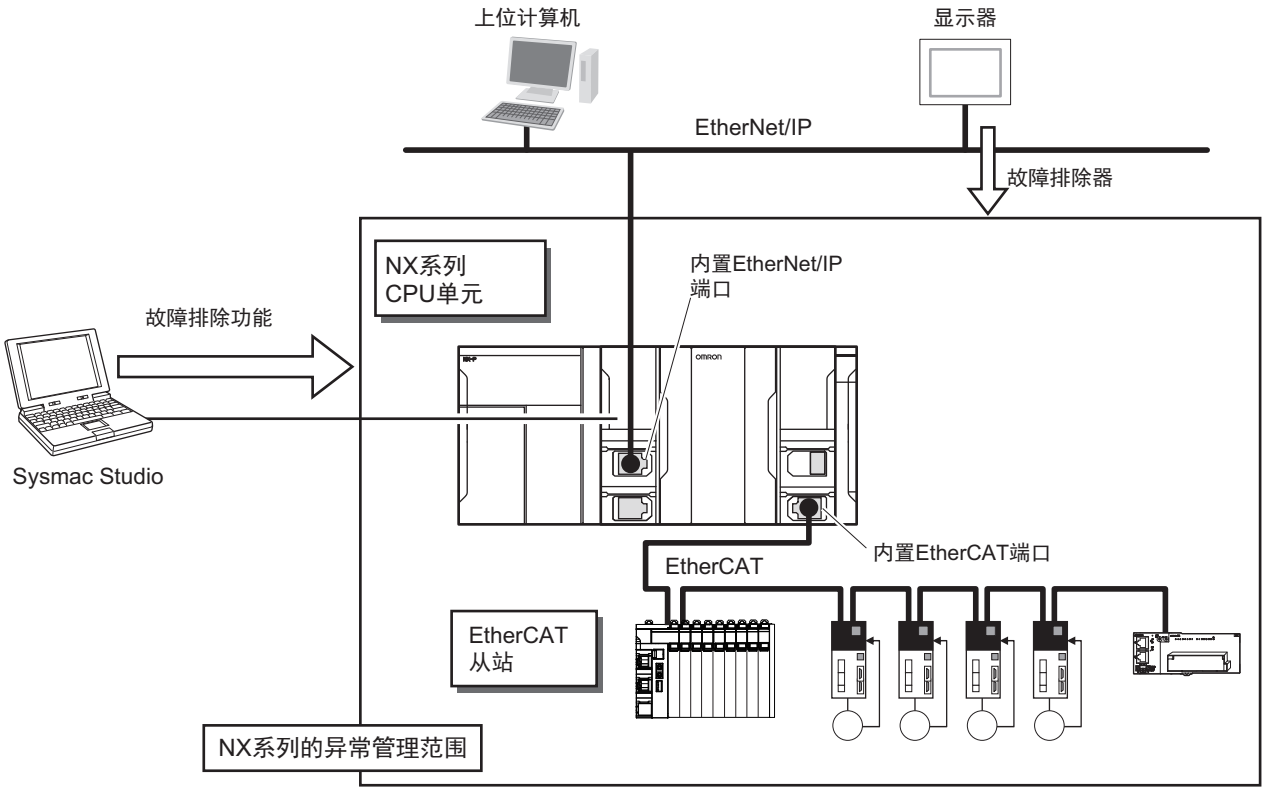
11-1 异常的概要

NJ/NX 系列可用共通的方法将控制器中发生的异常作为事件进行统一管理。因此，对于管理范围内（CPU 单元、 NX 系列从站终端、 EtherCAT 从站、 CJ 系列单元）的异常，可用通用的方法确认异常内容和处理方法。

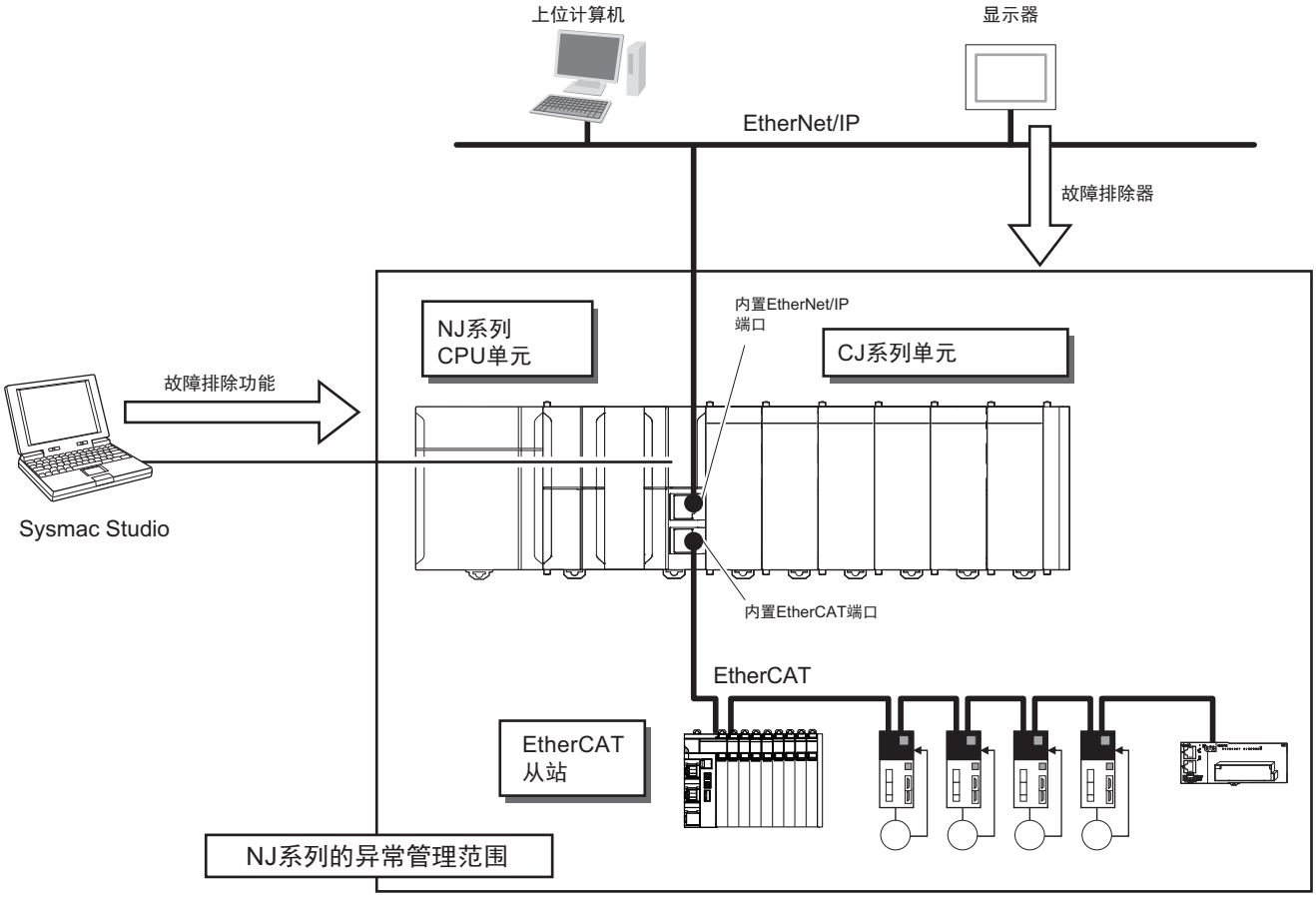
但是， EtherCAT 从站需要是 Sysmac 设备。

Sysmac 设备请参考 🕮《NJ/NX 系列 CPU 单元 用户手册 软件篇 （SBCA-CN5-359）》。

● NX 系列 CPU 单元时



● NJ 系列 CPU 单元时



若使用 Sysmac Studio 的故障排除功能或显示器的故障排除，可确认异常内容和处理方法，可实现迅速的应对。
使用显示器的故障排除时，请通过 CPU 单元的内置 EtherNet/IP 端口端口连接显示器。

本手册中介绍发生位置为 MC 功能模块的异常。
发生异常时的具体处理方法及 NJ/NX 系列整体的故障排除请参考 🕮《NJ/NX 系列 故障排除手册（SBCA-CN5-361）》。
重启运动控制指令时发生的异常请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》的附录。

使用注意事项

关于显示器的故障排除应对范围，请参考 🕮《NJ/NX 系列 故障排除手册 （SBCA-CN5-361）》的附录。

11-1-1 异常的确认方法

可按照以下方法确认异常的发生状态。

确认方法	可确认内容
通过 LED 确认	CPU 单元的动作状态
通过 Sysmac Studio 的故障排除功能确认	发生中的控制器异常、过去发生的控制器异常记录、异常的发生位置、异常的发生原因和处理对策
通过显示器的故障排除确认 *1	发生中的控制器异常、过去发生的控制器异常记录、异常的发生位置、发生原因和处理对策
通过各高功能单元的异常状态获取指令确认	发生中的控制器异常中，最重要的状态和最重要的事件代码
通过系统定义变量确认	各功能模块中发生的控制器异常的状态

*1. 使用显示器的故障排除时，请通过 CPU 单元的内置 EtherNet/IP 端口端口连接显示器。关于显示器的故障排除应对范围，请参考 🕮《NJ/NX 系列 故障排除手册 （SBCA-CN5-361）》的附录。

下面介绍各种确认方法。

通过 LED 确认

根据电源单元的 PWR LED 和 CPU 单元的 RUN、ERROR LED 的显示，可确认异常的重要程度。LED 的显示和异常的重要程度关系如下。

● NX 系列 CPU 单元

○: 亮灯、 ●: 熄灯、 ⊙: 闪烁

LED			CPU 单元的动作状态	是否可通过 Sysmac Studio/ 显示器确认详细内容
PWR（绿色）	RUN（绿色）	ERROR（红色）		
●	●	●	电源部异常	不可 🕮 请参考 《NJ/NX 系列 故障排除手册（SBCA-CN5-361）》。
○	●	●	CPU 复位 *1 *2	
○	● 或 ⊙	○	CPU 异常 *2 *3	
○	⊙ 持续 30 秒以上	○	系统初始化异常	
○	●	○	全部停止故障等级 *2 *3	可以 连接 Sysmac Studio/ 显示器，通过故障排除功能 / 故障排除器确认原因和处理措施。
○	○	⊙	部分停止故障等级	
○	○	⊙	轻度故障等级	
○	○	●	监视信息	
○	○	●	正常 : 运行模式中	—
○	●	●	正常 : 程序模式中 *1 *2	—
○	⊙	●	正常 : 启动中	—

*1. 通过 “USB- 直接连接 ” 连接 Sysmac Studio，若可以在线连接，则 CPU 单元变为 “ 程序模式中 ”，若无法在线连接，则 CPU 单元变为 “CPU 复位 ”。

*2. Sysmac Studio 无法在线连接时，可能是因为 USB 电缆不良或 Sysmac Studio 的 “ 网络类型 ” 未设定为 “USB- 直接连接 ”。无法在线连接时，请参考 🕮《NJ/NX 系列 故障排除手册 （SBCA-CN5-361）》。

*3. 通过 “USB- 直接连接 ” 连接 Sysmac Studio，若可以在线连接，则 CPU 单元发生 “ 全部停止故障等级 ” 的异常，若无法在线连接，则为 “CPU 异常 ”。

● NJ 系列 CPU 单元

○: 亮灯、 ●: 熄灯、 ⊙: 闪烁

LED			CPU 单元的动作状态	是否可通过 Sysmac Studio/ 显示器确认详细内容
PWR（绿色）	RUN（绿色）	ERROR（红色）		
●	●	●	电源部异常	不可 🕮 请参考 《NJ/NX 系列 故障排除手册（SBCA-CN5-361）》。
○	●	●	CPU 复位 *1 *2	
○	⊙	○	电源连接不正确	
○	●	○	CPU 异常 （WDT 异常）*2 *3	
○	●	○	全部停止故障等级 *2 *3	可以 连接 Sysmac Studio/ 显示器，通过故障排除功能 / 故障排除器确认原因和处理措施。
○	○	⊙	部分停止故障等级	
○	○	⊙	轻度故障等级	
○	○	●	监视信息	
○	○	●	正常 : 运行模式中	—
○	●	●	正常 : 程序模式中 *1 *2	—
○	⊙	●	正常 : 启动中	—

*1. 通过 “USB- 直接连接 ” 连接 Sysmac Studio，若可以在线连接，则 CPU 单元变为 “ 程序模式中 ”，若无法在线连接，则 CPU 单元变为 “CPU 复位 ”。
*2. Sysmac Studio 无法在线连接时，可能是因为 USB 电缆不良或 Sysmac Studio 的 “ 网络类型 ” 未设定为 “USB- 直接连接 ”。无法在线连接时，请参考 🕮《NJ/NX 系列 故障排除手册 （SBCA-CN5-361）》。
*3. 通过 “USB- 直接连接 ” 连接 Sysmac Studio，若可以在线连接，则 CPU 单元发生 “ 全部停止故障等级 ” 的异常，若无法在线连接，则为 “CPU 异常 （WDT 异常） ”。

通过 Sysmac Studio 的故障排除功能确认

发生异常时，将 Sysmac Studio 在线连接到控制器上，可确认发生中的异常、过去发生的异常记录。此外，可确认异常的发生原因和处理对策。

使用 Sysmac Studio 确认异常的方法和处理方法请参考 🕮《NJ/NX 系列 故障排除手册 （SBCA-CN5-361）》。

通过显示器的故障排除确认

发生异常时，若显示器和控制器建立通信，可确认发生中的异常、过去发生的异常记录。此外，可确认异常的发生原因和处理对策。

使用显示器的故障排除时，请通过 CPU 单元的内置 EtherNet/IP 端口端口连接显示器。

使用显示器确认异常的方法和处理方法请参考 🕮《NJ/NX 系列 故障排除手册 （SBCA-CN5-361）》。

通过异常状态获取指令确认

在用户程序中，可利用各功能模块上的异常状态获取指令获取异常状态。
MC 功能模块的异常状态获取指令如下所示。

指令	名称	功能概要
GetMCError	获取运动控制异常状态	获取在 MC 功能模块中发生的控制器异常中，最重要的状态 (部分停止故障、轻度故障) 和最重要的事件代码。

关于异常状态获取指令，请参考 《NJ/NX 系列 指令基准手册 基本篇 （SBCA-CN5-360）》。

通过系统定义变量确认

MC 功能模块中发生的异常可通过系统定义变量中的 “ 异常状态变量 ” 及 “ 运动控制系统变量 ” 确认。

● 异常状态变量

NJ/NX 系列中发生的异常可作为各功能模块的异常状态变量确认。
MC 功能模块的异常状态变量如下所示。

变量名称	数据类型	名称	功能
_MC_ErrSta	WORD	MC 异常状态	集中显示 MC 功能模块中发生的所有异常状态。
_MC_ComErrSta	WORD	MC 共通异常状态	集中显示 MC 功能模块共通处理中发生的异常状态。
_MC_AX_ErrSta	ARRAY[] OF WORD	轴异常状态	集中显示各轴中发生的异常状态。
_MC_GRP_ErrSta	ARRAY[] OF WORD	轴组异常状态	集中显示各轴组中发生的异常状态。

异常状态变量中各个位的含义如下。

位	名称	内容	值	含义
15	主机检测 *1	表示主机管理的对象从站是否有异常。	TRUE	有异常
			FALSE	无异常
14	从站集 *2	集中显示 MC 功能模块的轴中分配的 EtherCAT 从站是否有异常。	TRUE	有异常
			FALSE	无异常
13 ～ 8	保留			
7	全部停止故障	表示是否有全部停止故障等级的异常。	TRUE	有异常
			FALSE	无异常
6	部分停止故障	表示是否有部分停止故障等级的异常。	TRUE	有异常
			FALSE	无异常
5	轻度故障	表示是否有轻度故障等级的异常。	TRUE	有异常
			FALSE	无异常
4	监视信息	表示是否有监视信息等级的异常。	TRUE	有异常
			FALSE	无异常
3 ～ 0	保留			

*1. MC 功能模块的异常状态变量中不使用。
*2. MC 功能模块中只使用 _MC_ErrSta （MC 异常状态）。

● **运动控制系统变量**

参照运动控制系统变量的 MC 共通变量、轴变量、轴组变量，可确认 MC 功能模块中发生的异常。

关于运动控制系统变量，请参考 🕮「*6-6 运动控制系统变量* (P.6-16)」。

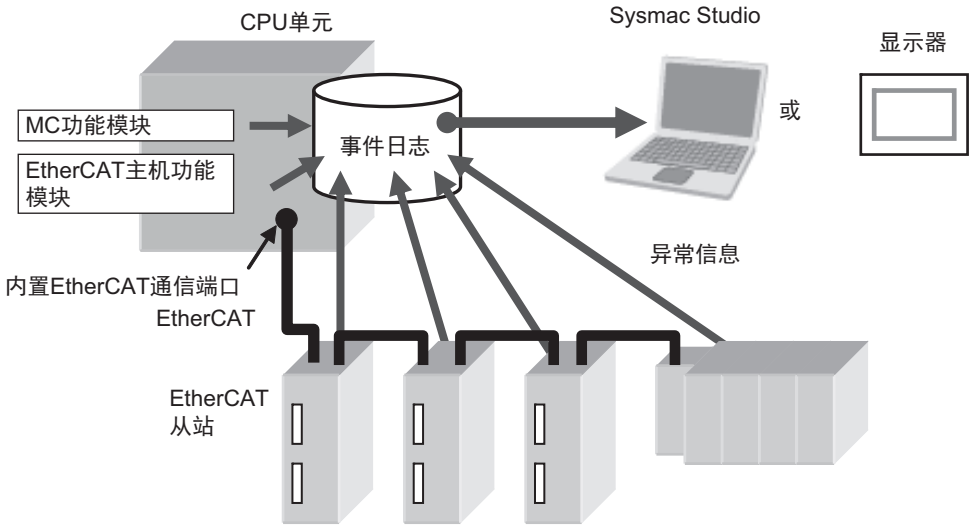
11-1-2 MC 功能模块相关的异常

下面介绍与 MC 功能模块相关的异常。

MC 功能模块相关的异常的发生位置

除了 MC 功能模块内部发生的异常之外，还有因 EtherCAT 通信，即与伺服驱动器的连接方式而发生的异常。

- MC 功能模块内部
- EtherCAT 主机功能模块
- 内置 EtherCAT 通信端口 （硬件）
- EtherCAT 从站



异常发生位置或发生原因可通过系统定义变量及 Sysmac Studio / 显示器确认。

使用注意事项

关于显示器的故障排除应对范围，请参考 🕮《NJ/NX 系列 故障排除手册 （SBCA-CN5-361）》的附录。

分类

MC 功能模块的异常发生位置分为以下 3 类。

分类	说明
MC 共通异常	检测到与 MC 功能模块的共通部分相关的异常时，“MC 共通异常状态变量 ” 中相应的位变为 “ 有异常 ” 状态。
轴异常	检测到与轴相关的异常时，“ 轴异常状态变量 ” 中相应的位变为 “ 有异常 ” 状态。*1
轴组异常	检测到与轴组相关的异常时，“ 轴组异常状态变量 ” 中相应的位变为 “ 有异常 ” 状态。

*1. 若发生轻度故障等级以上的轴异常，构成轴中包含异常轴的轴组也将无法动作。

异常的重要程度

下面介绍发生不同重要程序的异常时 MC 功能模块的动作。

异常的重要程度等级	动作
全部停止故障	NJ/NX 系列控制器整体停止控制的异常。
部分停止故障	NJ/NX 系列控制器中的功能模块整个停止控制的异常。 若发生在 MC 功能模块上，轴动作等 MC 功能模块的所有功能都将停止。
轻度故障	NJ/NX 系列控制器中的各功能模块部分停止控制的异常。 若发生在 MC 功能模块上，对应的轴 / 轴组将停止。
监视信息	对 NJ/NX 系列控制器的控制没有影响的异常。 通知的目的是防止发展为轻度故障等级以上的异常。
一般信息	除异常外向用户通知的信息。

MC 功能模块各发生位置的异常

按各个发生位置记载不同重要程度等级的异常。

● 发生位置为 MC 共通的异常

重要程度	异常名称
全部停止故障	无
部分停止故障	• 运动控制参数设定异常 • 凸轮数据读取异常 • 必需过程数据对象未设定 • 轴分配从站无效 • 轴分配从站网络配置信息未登录 • 运动控制初始化异常 • 运动控制周期超限 • 绝对编码器原点位置偏置读取异常
轻度故障	• 凸轮表保存处理失败 • 其他执行运动控制指令时的异常
监视信息	• 不可执行凸轮表保存指令 • 运动控制异常解除指令过多
一般信息	从 MC 调试画面执行异常解除

● 发生位置为轴的异常

重要程度	异常名称	
全部停止故障	无	
部分停止故障	无	
轻度故障	• 凸轮动作中凸轮表数据异常 • 立即停止指令启动 • 正方向软件限位超限 • 负方向软件限位超限 • 位置检查时间超限 • 位置偏差超限 • 立即停止输入 • 正方向极限输入检测 • 负方向极限输入检测 • 位置偏差不正确 • 伺服 OFF 异常 • 无法计算绝对编码器当前位置 • 伺服主回路电源关闭检测 • 中断固定尺寸定位无中断信号 • 原点复位反方向极限输入检测 • 原点复位方向极限输入检测 • 原点复位两个方向极限输入检测 • 近原点、原点复位反方向极限输入检测	• 近原点、原点复位方向极限输入检测 • 原点输入、原点复位反方向极限输入检测 • 原点输入、原点复位方向极限输入检测 • 原点输入掩码量不正确 • 无原点输入 • 无近原点输入 • 从站异常检测 • 发生 MC 共通异常 • 锁定位置溢出 • 锁定位置下溢 • 主轴同步方向异常 • 伺服 ON 中从站脱离 • 固定尺寸距离溢出 • 伺服驱动器的控制模式切换异常 • 主轴位置读取异常 • 辅助轴位置读取异常 • 从站通信异常 • 其他执行运动控制指令时的异常
监视信息	• 位置偏差警告 • 速度警告 • 加速度警告 • 减速度警告 • 正方向扭矩警告 • 负方向扭矩警告 • 指令位置溢出	• 指令位置下溢 • 反馈位置溢出 • 反馈位置下溢 • 从站监视信息检测 • 达到共混中继速度所需的移动量不足通知 • 其他执行运动控制指令时的异常
一般信息	从站异常代码确定	

● 发生位置为轴组的异常

重要程度	异常名称
全部停止故障	无
部分停止故障	无
轻度故障	• 轴组立即停止指令启动 • 多轴协调动作中的原点未确定 • 发生轴组构成轴异常 • 其他执行运动控制指令时的异常
监视信息	• 速度警告 • 加速度警告 • 减速度警告 • 达到共混中继速度所需的移动量不足通知
一般信息	无

某些事件的重要程度可以变更。重要程度的变更请参考 🕮《NJ/NX 系列 CPU 单元 用户手册 软件篇（SBCA-CN5-359）》。若要了解可变更重要程度的事件，请参考 🕮「*11-2 故障排除* (P.11-12)」。

EtherCAT 通信、 EtherCAT 从站、 NX 单元相关的异常

因 EtherCAT 通信、 EtherCAT 从站、 NX 单元的异常而发生的 MC 功能模块异常如下。

异常名称	事件代码	发生原因	发生异常时的动作
从站通信异常	84400000Hex	与 MC 控制功能模块的轴中分配的 EtherCAT 从站或 NX 单元的通信发生异常。*1	将发生异常的轴设为伺服 OFF，不接受异常解除以外的操作。*2
从站异常检测	742F0000Hex	检测到 MC 控制功能模块的轴中分配的 EtherCAT 从站或 NX 单元异常。	将发生异常的轴设为伺服 OFF，不接受异常解除以外的操作。

*1. 与 EtherCAT 从站的通信发生异常时， EtherCAT 主机功能模块也会发生异常。将多个设备分配到 1 个轴中时，只要 1 个设备发生通信异常，该轴就会变为通信异常。
*2. 发生从站通信异常时，该轴变为原点未确定。

伺服驱动器发生异常

介绍欧姆龙生产的伺服驱动器 G5 系列发生异常时的通知。

MC 功能模块检测到伺服驱动器发生异常的时间和从伺服驱动器获取异常代码的时间有时间差。
因此， MC 功能模块会以不同的事件通知伺服驱动器的异常发生和异常代码。

● 异常发生通知

MC 功能模块检测到伺服驱动器发生异常后，将发生轻度故障等级 “ 从站异常检测 ”(742F0000Hex)。
此时， MC 功能模块执行发生异常时的动作 （伺服 OFF）。

● 异常代码通知

伺服驱动器确定异常代码后， MC 功能模块将发生一般信息 “ 从站异常代码确定 ”(94220000Hex)。
“ 从站异常代码确定 ” 的附属信息后 2 位中将显示伺服驱动器的异常代码 （错误显示 No. 的主编号）。
例如，附属信息中显示 “FF13” 时，表示伺服驱动器上发生错误显示 No.13 （主回路电源电压不足）。

使用注意事项

要通知 “ 从站异常代码确定 ”，需要进行设定。
请在 PDO 映射编辑画面映射对象 603FHex （错误代码）。

NX 单元中发生的异常

欧姆龙生产的 NX 系列位置接口单元发生异常时，与欧姆龙生产的伺服驱动器 G5 系列相同，也会发出异常发生通知和异常代码通知。

但是， NX 系列位置接口单元中没有与 603FHex （错误代码）对应的对象，因此 “ 从站异常代码确定 ”(94220000Hex) 的附属信息中将显示 “0000Hex”。

关于 NX 系列位置接口单元中发生的异常详情，请参考 《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》及《NX 系列 EtherCAT 耦合器单元 用户手册 (SBCD-CN5-361)》。

11-2 故障排除

下面介绍发生的异常及其处理。

11-2-1 异常一览

MC 功能模块中发生的异常 （事件）一览如下所示。重要程度栏中使用以下简称和符号。

简称	名称
全	全部停止故障等级
部	部分停止故障等级
轻	轻度故障等级
监	监视信息
般	一般信息

符号	含义
○	系统定义的重要程度
⊙	用户可变更的重要程度 *1

*1. 仅在有可变更的重要程度时记载。

输出到运动控制系统变量中的异常代码为下表事件代码中的前 4 位。

事件代码栏的 （ ）中表示限定发生该事件的 CPU 单元时， CPU 单元的版本。

关于 NJ/NX 系列的所有事件代码，请参考 🕮《NJ/NX 系列 故障排除手册 （SBCA-CN5-361）》。

事件代码	事件名称	内容	发生原因 （推测原因）	重要程度					参照处
				全	部	轻	监	般	
44210000Hex	运动控制功能处理异常	运动控制功能模块中检测到致命异常。	软件发生错误	○					P. 11-21
14600000Hex	绝对编码器原点位置偏置读取异常	电源 OFF 过程中保持的绝对编码器当前位置消失。	• CPU 单元的电池寿命 • 备份存储器故障		○				P. 11-21
14610000Hex	运动控制参数设定异常	非易失性存储器中保存的运动控制参数设定消失。	• 正在下载运动控制参数设定或清除存储器时，发生断电或断开了与 Sysmac Studio 的通信。 • 非易失性存储器故障		○				P. 11-22
14620000Hex	凸轮数据读取异常	非易失性存储器中保存的凸轮数据消失。	• 正在保存凸轮数据时断电 • 非易失性存储器故障		○				P. 11-22
34600000Hex	必需过程数据对象未设定	轴种类必需的对象未分配到 PDO 中。	• 轴种类为伺服轴或编码器轴时，未进行必需的 PDO 映射。 • 非易失性存储器故障		○				P. 11-23

事件代码	事件名称	内容	发生原因 （推测原因）	重要程度					参照处
				全	部	轻	监	般	
34630000Hex	轴分配从站无效	轴中分配的从站设定为无效。	轴中分配的从站设定为无效。		○				P. 11-23
34640000Hex	轴分配从站网络配置信息未登录	轴中分配的从站未登录到网络配置信息中。	轴中分配的从站未登录到 EtherCAT 的网络配置信息中。		○				P. 11-24
44200000Hex	运动控制初始化异常	系统发生致命故障，运动控制的初始化失败。	硬件故障		○				P. 11-24
74200000Hex	运动控制周期超限	主固定周期任务的处理未在 2 个控制周期内完成。	主固定周期任务的程序处理内容过多。		○				P. 11-25
14630000Hex	凸轮表保存处理失败	凸轮表文件保存失败。	凸轮表文件保存失败。			○			P. 11-25
54770000Hex	凸轮动作中凸轮表数据异常	凸轮表的相位不是升序。	• 凸轮动作过程中检测到凸轮表的相位不是升序的数据。 • 凸轮动作过程中检测到凸轮表的起点相位和变位不是 0。 • 凸轮动作过程中，检测到将凸轮表的终点相位转换为脉冲单位时，未达到 1 脉冲以上。			○			P. 11-26
54850000Hex	立即停止指令启动	执行了立即停止 (MC_ImmediateStop) 指令。	执行了立即停止指令。			○			P. 11-26
54860000Hex	轴组立即停止指令启动	执行了轴组立即停止 (MC_GroupImmediateStop) 指令。	执行了轴组立即停止指令。			○			P. 11-26
64450000Hex	正方向软件限位超限	轴正在动作时，位置超出正方向软件限位。	位置超出正方向软件限位。			○			P. 11-27
64460000Hex	负方向软件限位超限	轴正在动作时，位置超出负方向软件限位。	位置超出负方向软件限位。			○			P. 11-27
64470000Hex	位置检查时间超限	定位完成检查未在监视时间内完成。	为完成定位花费时间过长。			○			P. 11-27
64480000Hex	位置偏差超限	指令当前位置和反馈当前位置的偏差超出位置偏差超限值。	定位动作的追踪性差，实际动作相对于指令有延迟。			○			P. 11-28

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
64490000Hex	立即停止输入	立即停止输入变为 ON。	• 检测到立即停止输入信号。 • 立即停止信号的连接或立即停止输入的逻辑设定错误。			○			P. 11-28
644A0000Hex	正方向极限输入检测	正方向极限输入变为 ON。	• 检测到正方向极限输入信号。 • 正方向极限输入信号的连接或正方向极限输入的逻辑设定错误。			○			P. 11-29
644B0000Hex	负方向极限输入检测	负方向极限输入变为 ON。	• 检测到负方向极限输入信号。 • 负方向极限输入信号的连接或负方向极限输入的逻辑设定错误。			○			P. 11-29
64560000Hex	位置偏差不正确	将指令位置和反馈当前位置的差转换为脉冲单位时，超出 30 位。	• 在从轴中，若按指令的轴移动量移动，将超出轴的最高速度，因此对指令当前位置进行了限制，以免超出最高速度。 • 从轴定位动作的追踪性差，实际动作相对于指令有延迟。			○			P. 11-30
64570000Hex	伺服 OFF 异常	因轴组发生异常，轴变为伺服 OFF。	因轴组发生异常，轴变为伺服 OFF。			○			P. 11-30
64580000Hex	无法计算绝对编码器当前位置	无法用断电保持的绝对编码器信息正确恢复当前位置。	• 单位转换设定、循环计数器设定或伺服驱动器设定的循环计数器设定发生变更。 • 将恢复的位置转换为脉冲单位时，带符号超出 40 位的范围			○			P. 11-31
64590000Hex	多轴协调动作中的原点未确定	轴组动作中或减速停止中，逻辑轴的原点变为未确定。	• 轴组动作中或减速停止中，逻辑轴的指令位置或反馈位置发生溢出或下溢，原点变为未确定。 • 轴组动作中或减速停止中，逻辑轴发生从站通信异常，原点变为未确定。 • 轴组动作中或减速停止中，逻辑轴的从站脱离或从站变为无效，原点变为未确定。			○			P. 11-32
74210000Hex	伺服主回路电源关闭检测	在伺服 ON 状态下关闭了伺服驱动器的主回路电源	在伺服 ON 状态下伺服驱动器的主回路电源断开			○			P. 11-32
74230000Hex	中断固定尺寸定位无中断信号	执行 MC_MoveFeed（中断固定尺寸定位）指令的过程中无中断输入。	• 锁定有效范围指定不合适 • 中断信号的配线错误。 • 输出中断信号的传感器故障。			○			P. 11-33

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
74240000Hex	原点复位反方向极限输入检测	原点复位动作中，检测到与原点复位方向相反的极限信号。	• 正方向极限输入时的动作即负方向极限输入时的动作设定为“不反转”。 • 原点复位所需的输入信号传感器的配置、原点复位设定、原点复位开始位置的关系导致出现极限信号。 • 输入信号传感器配线错误、故障			○			P. 11-33
74250000Hex	原点复位方向极限输入检测	原点复位动作中，检测到原点复位方向的极限信号。	• 正方向极限输入时的动作即负方向极限输入时的动作设定为“不反转”。 • 原点复位所需的输入信号传感器的配置、原点复位设定、原点复位开始位置的关系导致出现极限信号。 • 输入信号传感器配线错误、故障			○			P. 11-34
74260000Hex	原点复位两个方向极限输入检测	原点复位动作中，同时检测到两侧的极限信号。	• 极限信号的配线错误 • 极限传感器的安装位置错误 • 极限信号的接点逻辑错误 • 极限传感器故障			○			P. 11-34
74270000Hex	近原点、原点复位反方向极限输入检测	原点复位动作中，同时检测到近原点输入以及与原点复位方向相反的极限信号。	• 近原点信号、极限信号的配线错误 • 近原点传感器、极限传感器的安装位置错误 • 近原点信号、极限信号的接点逻辑错误 • 近原点传感器、极限传感器故障			○			P. 11-35
74280000Hex	近原点、原点复位方向极限输入检测	原点复位动作中，同时检测到近原点输入以及原点复位方向的极限信号。	• 近原点信号、极限信号的配线错误 • 近原点传感器、极限传感器的安装位置错误 • 近原点信号、极限信号的接点逻辑错误 • 近原点传感器、极限传感器故障			○			P. 11-35
74290000Hex	原点输入、原点复位反方向极限输入检测	原点复位动作中，同时检测到原点输入以及与原点复位方向相反的极限信号。	• 原点输入信号、极限信号的配线错误 • 原点输入传感器、极限传感器的安装位置错误 • 原点输入信号、极限信号的接点逻辑错误 • 原点输入信号输出设备、极限传感器故障			○			P. 11-36

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
742A0000Hex	原点输入、原点复位方向极限输入检测	原点复位动作中，同时检测到原点输入以及原点复位方向的极限信号。	• 原点输入信号、极限信号的配线错误 • 原点输入传感器、极限传感器的安装位置错误 • 原点输入信号、极限信号的接点逻辑错误 • 原点输入信号输出设备、极限传感器故障			○			P. 11-36
742B0000Hex	原点输入掩码量不正确	MC_Home（原点复位）指令或 MC_HomeWithParameter（参数指定原点复位）指令中，原点输入掩码量的设定值不合适。	原点复位动作模式为“附近退避、原点输入掩码距离指定”时，原点输入掩码距离的设定值不够从原点复位速度减速至原点复位接近速度所需的移动量。			○			P. 11-37
742C0000Hex	无原点输入	原点复位过程中无原点信号输入。或者在无原点确定输入的状态下，检测到极限信号。	• 原点复位过程中无原点信号输入。 • 在无原点确定输入的状态下，检测到极限信号。			○			P. 11-37
742D0000Hex	无近原点输入	原点复位过程中无近原点信号输入。	设定了“有近原点信号”的原点复位过程中无近原点信号输入。			○			P. 11-38
742F0000Hex	从站异常检测	检测到轴中分配的 EtherCAT 从站或 NX 单元异常。	检测到轴中分配的 EtherCAT 从站或 NX 单元异常。			○			P. 11-38
74300000Hex	发生轴组构成轴异常	轴组的构成轴发生异常。	轴组动作过程中构成轴发生异常。			○			P. 11-38
74330000Hex	发生 MC 共通异常	发生 MC 共通异常。	发生部分停止故障等级的 MC 共通异常。			○			P. 11-39
74340000Hex	锁定位置溢出	用 MC_TouchProbe（外部锁定有效）指令锁定的位置溢出。	用 MC_TouchProbe（外部锁定有效）指令锁定的位置溢出			○			P. 11-39
74350000Hex	锁定位置下溢	用 MC_TouchProbe（外部锁定有效）指令锁定的位置下溢。	用 MC_TouchProbe（外部锁定有效）指令锁定的位置下溢			○			P. 11-39
74360000Hex	主轴同步方向异常	主轴持续向同步方向的反方向移动。	相对于主轴和从轴的同步方向，主轴持续向反方向移动并发生溢出。			○			P. 11-40

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
74370000Hex	伺服 ON 中从站脱离	伺服 ON 时，轴中分配的 EtherCAT 从站或 NX 单元发生脱离、更换或从站变为无效。	伺服 ON 时，轴中分配的 EtherCAT 从站或 NX 单元发生脱离、更换或从站变为无效。			○			P. 11-40
74380000Hex	固定尺寸距离溢出	MC_MoveFeed（中断固定尺寸定位）指令的中断输入后，目标位置发生溢出 / 下溢。	将 MC_MoveFeed（中断固定尺寸定位）指令中断输入后的目标位置转换为脉冲数单位时，带符号超出 40 位的范围。			○			P. 11-40
74390000Hex	伺服驱动器的控制模式切换异常	切换控制模式时，处理未在规定时间内完成。	• MC_SyncMoveVelocity（周期性同步速度控制）指令停止时，规定的减速处理未完成。 • 使用欧姆龙生产的伺服驱动器 G5 系列时，在 MC_TorqueControl（扭矩控制）指令停止时，规定的减速处理未完成。 • 发出控制模式 (CSP/ CSV/ CST) 的切换指令后，伺服驱动器的模式切换未在 1 秒内完成。			○			P. 11-41
743A0000Hex	主轴位置读取异常	同步控制指令的主轴位置发生异常，因此无法执行同步控制指令。	• 同步控制指令的主轴未建立 EtherCAT 过程数据通信，或 NX 单元的 I/O 数据无法用于控制。 • 同步控制指令的主轴发生从站脱离或从站无效 • 同步控制指令的主轴中，检测到无法计算绝对编码器当前位置 (64580000Hex) • 同步控制指令的主轴为未使用轴			○			P. 11-42

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
743B0000Hex	辅助轴位置读取异常	同步控制指令的辅助轴位置发生异常，因此无法执行同步控制指令。	• 同步控制指令的辅助轴未建立 EtherCAT 过程数据通信，或 NX 单元的 I/O 数据无法用于控制。 • 同步控制指令的辅助轴发生从站脱离或从站无效 • 同步控制指令的辅助轴中，检测到无法计算绝对编码器当前位置 (64580000Hex) • 同步控制指令的辅助轴为未使用轴			○			P. 11-43
84400000Hex	从站通信异常	与轴中分配的 EtherCAT 从站或 NX 单元的通信发生异常。	与轴中分配的 EtherCAT 从站或 NX 单元的通信发生异常。			○			P. 11-43
571D0000Hex（Ver.1.02 到 Ver.1.09）	运动控制异常解除指令过多	ResetMCError （运动控制异常解除）指令的实例数超过 100 个。	包括功能块的实例在内，用户程序中声明的 ResetMCError （运动控制异常解除）指令的实例数超过 100 个。				○		P. 11-44
644C0000Hex	位置偏差警告	位置偏差超出位置偏差警告值。	定位动作的追踪性差，实际动作相对于指令有延迟。				○		P. 11-44
644D0000Hex	速度警告	指令速度超出速度警告值。	指令速度超出速度警告值。			⊙	○		P. 11-45
644E0000Hex	加速度警告	指令加速度超出加速度警告值。	指令加速度超出加速度警告值。			⊙	○		P. 11-45
644F0000Hex	减速度警告	指令减速度超出减速度警告值。	指令减速度超出减速度警告值。			⊙	○		P. 11-46
64500000Hex	正方向扭矩警告	扭矩指令值超出正方向扭矩警告值。	扭矩指令值超出正方向扭矩警告值。			⊙	○		P. 11-46
64510000Hex	负方向扭矩警告	扭矩指令值超出负方向扭矩警告值。	扭矩指令值超出负方向扭矩警告值。			⊙	○		P. 11-46
64520000Hex	指令位置溢出	指令位置对应的脉冲数溢出。	线性模式时，将指令位置转换为脉冲单位时，带符号超出 40 位的上限值。			⊙	○		P. 11-47
64530000Hex	指令位置下溢	指令位置对应的脉冲数下溢。	线性模式时，将指令位置转换为脉冲单位时，带符号超出 40 位的下限值。			⊙	○		P. 11-47
64540000Hex	反馈位置溢出	反馈位置对应的脉冲数溢出。	将反馈位置转换为脉冲单位时，带符号超出 40 位的上限值。			⊙	○		P. 11-48

事件代码	事件名称	内容	发生原因（推测原因）	重要程度					参照处
				全	部	轻	监	般	
64550000Hex	反馈位置下溢	反馈位置对应的脉冲数下溢。	将反馈位置转换为脉冲单位时，带符号超出 40 位的下限值。			⊙	○		P. 11-48
74320000Hex	从站监视信息检测	检测到 EtherCAT 从站或 NX 单元的警告。	轴中分配的 EtherCAT 从站或 NX 单元检测到警告。			⊙	○		P. 11-49
743C0000Hex	不可执行凸轮表保存指令	其他操作正在访问非易失性存储器，因此无法执行凸轮表文件的保存。	其他操作（通过 Sysmac Studio 传送、数据追踪等）正在访问非易失性存储器时，执行了 MC_SaveCamTable（凸轮表保存）指令。				○		P. 11-49
94200000Hex	达到共混中继速度所需的移动量不足通知	加减速至共混中继速度所需的移动量不足	• “加减速超限”设定为“剧烈加减速（将共混切换为缓冲）”的情况下，Blending 指定时，创建曲线后发现发生加减速超限，因此变更为 Buffered 处理。 • Blending 指定时，已到达目标位置，无法正确绘制曲线，因此变更为 Buffered 处理。			⊙	○		P. 11-50
94210000Hex	从 MC 调试画面执行异常解除	在 Sysmac Studio 的调试画面中执行了异常解除。	在 Sysmac Studio 的调试画面中执行了异常解除。					○	P. 11-50
94220000Hex	从站异常代码确定	因发生“从站异常检测”，发出了异常代码通知。	因发生“从站异常检测”(742F0000Hex)，从站发出了异常代码通知。					○	P. 11-51

11-2-2 异常内容

下面介绍各异常的内容。

控制器异常说明的解释

［ ］内表示表中各项目的含义，用于解释各种异常。

<table>
<tr><td>事件名称</td><td colspan="2">［异常 （事件）的名称。］</td><td>事件代码</td><td colspan="2">［异常 （事件）的代码。］</td></tr>
<tr><td>内容</td><td colspan="5">［异常 （事件）的内容。］</td></tr>
<tr><td>发生位置</td><td colspan="2">［发生异常 （事件）的位置。］</td><td>发生位置详情</td><td>［发生位置的详细信息。］</td><td>［检测异常的时序。］</td></tr>
<tr><td>异常属性</td><td>重要程度</td><td>［对控制的影响程度。］*1</td><td>恢复方法</td><td>［恢复方法。］*2</td><td>日志类别</td><td>［所保存日志的种类。］*3</td></tr>
<tr><td>产生的影响</td><td>用户程序</td><td>［用户程序的执行状态。］*4</td><td>动作</td><td colspan="3">［与发生异常 （事件）时的动作相关的特殊事项。］</td></tr>
<tr><td>LED</td><td colspan="6">［内置 EtherNet/IP 端口用 LED、内置 EtherCAT 端口用 LED 的显示状态。仅当发生位置为 EtherCAT 主机功能模块、 EtherNet/IP 功能模块时记载。］</td></tr>
<tr><td rowspan="2">系统定义变量</td><td colspan="2">变量名称</td><td colspan="2">数据类型</td><td>名称</td></tr>
<tr><td colspan="5">［检测异常的系统定义变量、受异常影响的系统定义变量、引起异常的系统定义变量的变量名称和数据类型、名称。］</td></tr>
<tr><td rowspan="2">发生原因和处理</td><td colspan="2">发生原因 （推测原因）</td><td colspan="2">处理对策</td><td>防止再发</td></tr>
<tr><td colspan="5">［异常 （事件）的发生原因、处理对策和防止再发方法。］</td></tr>
<tr><td>附属信息</td><td colspan="6">［Sysmac Studio / NS 系列显示器上显示的附属信息内容。］*5</td></tr>
<tr><td>注意事项 / 备注</td><td colspan="6">［其他注意事项、限制事项、补充说明等。有用户可选择的重要程序时，对可选择的重要程度、恢复方法、动作等相关事项进行记载。］</td></tr>
</table>

*1. 以下任一。
全部停止故障：全部停止故障等级
部分停止故障：部分停止故障等级
轻度故障：轻度故障等级
监视信息
一般信息

*2. 以下任一。
自动恢复：排除原因后，自动恢复正常。
异常解除：排除原因并执行异常解除后恢复正常。
电源再接通：排除原因并重新接通电源后恢复正常。
控制器重置：排除原因并重置控制器后恢复正常。
取决于发生原因：根据发生原因而定。

*3. 以下任一。
系统：系统事件日志
访问日志：访问事件日志

*4. 以下任一。
继续：继续执行用户程序。
停止：停止执行用户程序。
开始：开始执行用户程序。

*5. 关于显示器的故障排除应对范围，请参考 《NJ/NX 系列 故障排除手册 （SBCA-CN5-361）》的附录。

版本相关信息

NX 系列 CPU 单元时，以 _MC_AX[*] 开始的变量名称可能是 _MC1_AX[*]、 _MC2_AX[*]。同样，以 _MC_GRP[*] 开始的变量名称可能是 _MC1_GRP[*]、 _MC2_GRP[*]。

异常说明

事件名称	运动控制功能处理异常			事件代码		44210000Hex
内容	运动控制功能模块中检测到致命异常。					
发生位置	PLC 功能模块		发生位置详情	MC 共通	检测时序	始终
异常属性	重要程度	全部停止故障	恢复方法	电源重新接通	日志类别	系统
产生的影响	用户程序	停止	动作	不可进行轴控制。控制器停止。		
系统定义变量	变量名称		数据类型		名称	
	无		—		—	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	软件发生错误。		请与本公司营业或经销商联系。		无	
附属信息	附属信息 1：系统信息 附属信息 2：系统信息 附属信息 3：系统信息 附属信息 4：系统信息					
注意事项 / 备注	无					

事件名称	绝对编码器原点位置偏置读取异常			事件代码		14600000Hex
内容	电源 OFF 过程中保持的绝对编码器当前位置消失。					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	接通电源时 / 控制器重置时 / 下载时
异常属性	重要程度	部分停止故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	不可进行轴控制。		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.PFaultLvl.Active		BOOL		MC 共通 发生部分停止故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	CPU 单元的电池寿命		更换 CPU 单元的电池后，请解除异常，执行原点复位，确定原点。		请定期更换 CPU 单元的电池。 电池的使用寿命请参考 《NX 系列 CPU 单元 用户手册 硬件篇（SBCA-CN5-418）》或 《NJ 系列 CPU 单元 用户手册 硬件篇（SBCA-CN5-358）》。	
	备份存储器故障		若采取上述措施后仍发生本异常，可能是因为 CPU 的备份存储器故障。更换 CPU 单元后，请执行原点复位，确定原点。		无	
附属信息	无					
注意事项 / 备注	无					

事件名称	运动控制参数设定异常			事件代码		14610000Hex
内容	非易失性存储器中保存的运动控制参数设定消失。					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	接通电源时 / 控制器重置时 / 下载时
异常属性	重要程度	部分停止故障	恢复方法	电源重新接通 / 控制器重置	日志类别	系统
产生的影响	用户程序	继续	动作	不可进行轴控制。		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.PFaultLvl.Active		BOOL		MC 共通 发生部分停止故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	正在下载运动控制参数设定或清除存储器时，发生断电或断开了与 Sysmac Studio 的通信。		请从 Sysmac Studio 下载运动控制参数设定。		保存参数设定的过程中请勿关闭电源。	
	非易失性存储器故障		若采取上述措施后仍发生本异常，可能是因为非易失性存储器故障。更换 CPU 单元后，请从 Sysmac Studio 下载包括轴设定在内的所有设定。		无	
附属信息	无					
注意事项 / 备注	无					

事件名称	凸轮数据读取异常			事件代码		14620000Hex
内容	非易失性存储器中保存的凸轮数据消失。					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	接通电源时 / 控制器重置时 / 下载时
异常属性	重要程度	部分停止故障	恢复方法	电源重新接通 / 控制器重置	日志类别	系统
产生的影响	用户程序	继续	动作	不可进行轴控制。		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.PFaultLvl.Active		BOOL		MC 共通 发生部分停止故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	正在保存凸轮数据时断电		请从 Sysmac Studio 下载凸轮数据。		保存凸轮数据的过程中请勿关闭电源。	
	非易失性存储器故障		若采取上述措施后仍发生本异常，可能是因为非易失性存储器故障。更换 CPU 单元后，请从 Sysmac Studio 下载包括轴设定在内的所有设定。		无	
附属信息	无					
注意事项 / 备注	无					

事件名称	必需过程数据对象未设定		事件代码	34600000Hex		
内容	轴种类必需的对象未分配到 PDO 中。					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	接通电源时 / 控制器重置时 / 下载时
异常属性	重要程度	部分停止故障	恢复方法	电源重新接通 / 控制器重置	日志类别	系统
产生的影响	用户程序	继续	动作	不可进行轴控制。		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.PFaultLvl.Active		BOOL		MC 共通 发生部分停止故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	轴种类为伺服轴或编码器轴时，未进行必需的 PDO 映射。		请执行该轴种类所需的 PDO 映射。 关于所需的 PDO 映射，请参考 *「A-1-2 伺服驱动器的设定 (P.A-2)」*。		请执行使用的轴种类所需的 PDO 映射。 关于所需的 PDO 映射，请参考 *「A-1-2 伺服驱动器的设定 (P.A-2)」*。	
	非易失性存储器故障		若采取上述措施后仍发生本异常，可能是因为非易失性存储器故障。更换 CPU 单元后，请从 Sysmac Studio 下载包括轴参数设定在内的所有设定。		无	
附属信息	无					
注意事项 / 备注	无					

事件名称	轴分配从站无效		事件代码	34630000Hex		
内容	轴中分配的从站设定为无效。					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	接通电源时 / 控制器重置时 / 下载时
异常属性	重要程度	部分停止故障	恢复方法	电源重新接通 / 控制器重置	日志类别	系统
产生的影响	用户程序	继续	动作	不可进行轴控制。		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.PFaultLvl.Active		BOOL		MC 共通 发生部分停止故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	轴中分配的从站设定为无效。		请将轴分配中相应从站的 EtherCAT 设定，即从站有效 / 无效设定为 “ 有效 ”。 无从站时，请将相应轴的 “ 轴种类 ” 设定为 “ 虚拟轴 ”。		请将轴分配中从站的 EtherCAT 设定，即从站有效 / 无效设定为 “ 有效 ”。 无从站时，若要在程序中使用轴，请将 “ 轴种类 ” 设定为 “ 虚拟轴 ”。	
附属信息	无					
注意事项 / 备注	无					

事件名称	轴分配从站网络配置信息未登录			事件代码		34640000Hex
内容	轴中分配的从站未登录到网络配置信息中。					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	接通电源时 / 控制器重置时 / 下载时 / 伺服 ON 开始时 / 从未使用轴切换为使用轴时
异常属性	重要程度	部分停止故障	恢复方法	电源重新接通 / 控制器重置	日志类别	系统
产生的影响	用户程序	继续	动作	不可进行轴控制。		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.PFaultLvl.Active		BOOL		MC 共通 发生部分停止故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	轴中分配的从站未登录到 EtherCAT 的网络配置信息中。		请将轴中分配的从站登录到 EtherCAT 的网络配置信息中。或者将相应轴的 “ 轴种类 ” 设定为 “ 虚拟轴 ”。		请将轴中分配的从站登录到网络配置信息中。	
附属信息	无					
注意事项 / 备注	无					

事件名称	运动控制初始化异常			事件代码		44200000Hex
内容	系统发生致命故障，运动控制功能模块的初始化失败。					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	接通电源时 / 控制器重置时 / 下载时
异常属性	重要程度	部分停止故障	恢复方法	电源重新接通	日志类别	系统
产生的影响	用户程序	继续	动作	不可进行轴控制。不可执行运动控制指令。		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.PFaultLvl.Active		BOOL		MC 共通 发生部分停止故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	硬件故障		请更换 CPU 单元。		无	
附属信息	附属信息 1：系统信息					
注意事项 / 备注	无					

事件名称	运动控制周期超限		事件代码	74200000Hex		
内容	主固定周期任务的处理未在 2 个控制周期内完成。					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	始终
异常属性	重要程度	部分停止故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	所有轴不可动作。动作中的轴立即停止。		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.PFaultLvl.Active		BOOL		MC 共通 发生部分停止故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	主固定周期任务的程序处理内容过多。		请减少主固定周期任务的程序处理内容，或在动作不会发生问题的范围内，将控制周期设定为更大的值。 任务周期可通过 Sysmac Studio 的 “ 任务周期监视 ” 确认。		请只将需要以固定周期执行的处理记述到主固定周期任务的程序中。或者将主固定周期任务的任务周期设定为足够长的时间。	
附属信息	无					
注意事项 / 备注	无					

事件名称	凸轮表保存处理失败		事件代码	14630000Hex		
内容	凸轮表文件保存失败。					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	异常解除 / 电源重新接通	日志类别	系统
产生的影响	用户程序	继续	动作	非易失性存储器中保存的凸轮表可能损坏，因此读取凸轮表时可能发生异常。		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.MFaultLvl.Active		BOOL		MC 共通 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	凸轮表文件保存失败。		请重新执行保存。若仍发生，可能是因为非易失性存储器故障。请更换 CPU 单元。		无	
附属信息	无					
注意事项 / 备注	无					

事件名称	凸轮动作中凸轮表数据异常			事件代码	54770000Hex	
内容	凸轮表的相位不是升序。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	凸轮动作过程中检测到凸轮表的相位不是升序的数据。		请修正凸轮表数据，使相位为升序。		请使凸轮表数据的相位为升序。	
	凸轮动作过程中检测到凸轮表的起点相位和变位不是 “0”。		请修正凸轮表数据，使起点的相位和变位为 “0”。		请使凸轮表数据中起点的相位和变位为 “0”。	
	凸轮动作过程中，检测到将凸轮表的终点相位转换为脉冲单位时，未达到 1 脉冲以上。		请修正凸轮表数据，使终点的相位转换为脉冲单位时大于 1 脉冲。		请修正凸轮表数据，使终点的相位转换为脉冲单位时大于 1 脉冲。	
附属信息	无					
注意事项 / 备注	无					

事件名称	立即停止指令启动			事件代码	54850000Hex	
内容	执行了立即停止 (MC_ImmediateStop) 指令。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴根据立即停止指令的输入变量 “StopMode（停止方法选择）” 的设定立即停止。 相应轴正在执行轴组动作时，其他的轴遵照 “ 轴组停止方法 ”。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	执行了立即停止指令。		—		—	
附属信息	无					
注意事项 / 备注	无					

事件名称	轴组立即停止指令启动			事件代码	54860000Hex	
内容	执行了轴组立即停止 (MC_GroupImmediateStop) 指令。					
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序	指令启动时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴组的轴遵照轴参数的 “ 立即停止输入停止方法 ” 的设定立即停止所有轴。		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	执行了轴组立即停止指令。		—		—	
附属信息	无					
注意事项 / 备注	无					

事件名称	正方向软件限位超限			事件代码		64450000Hex
内容	轴正在动作时，位置超出正方向软件限位。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	遵照 " 软件限位功能 " 设定。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	位置超出正方向软件限位。		请调查超出软件限位的原因，采取合理的措施。		（目的是检测意外原因导致的软件限位超限，因此不需要防止）	
附属信息	无					
注意事项 / 备注	变更正方向软件限位的设定时，设定时请考虑安全性。					

事件名称	负方向软件限位超限			事件代码		64460000Hex
内容	轴正在动作时，位置超出负方向软件限位。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	遵照 " 软件限位功能 " 设定。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	位置超出负方向软件限位。		请调查超出软件限位的原因，采取合理的措施。		（目的是检测意外原因导致的软件限位超限，因此不需要防止）	
附属信息	无					
注意事项 / 备注	变更负方向软件限位的设定时，设定时请考虑安全性。					

事件名称	位置检查时间超限			事件代码		64470000Hex
内容	定位完成检查未在监视时间内完成。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	为完成定位花费时间过长。		调查完成定位时间过长的原因，并排除该原因。或者调整伺服驱动器，调整位置检查时间和位置宽度。 调整伺服驱动器时，请放大回路增益。但是，应在控制不会发生振荡的合理范围内。		应尽量排除可能导致定位动作追踪性变差的原因和振动。	
附属信息	无					
注意事项 / 备注	无					

事件名称	位置偏差超限		事件代码		64480000Hex	
内容	指令当前位置和反馈当前位置的偏差超出位置偏差超限值。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。相应轴正在动作时，则减速停止。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	定位动作的追踪性差，实际动作相对于指令有延迟。		请排除可能导致定位动作追踪性变差的原因。 此外，请在不会发生问题的范围内，将“位置偏差超限值”变更为更大的值。		请尽量排除可能导致定位动作追踪性变差的原因。	
附属信息	无					
注意事项 / 备注	无					

事件名称	立即停止输入		事件代码		64490000Hex	
内容	立即停止输入变为 ON。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	始终
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	遵照“立即停止输入停止方法”设定。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	检测到立即停止输入信号。		请关闭立即停止输入信号。		（目的是检测立即停止输入，因此不需要防止）	
	立即停止信号的连接或立即停止输入的逻辑设定错误。		即使立即停止输入信号为 OFF，发生异常时，也请修正立即停止信号的连接、立即停止输入的逻辑设定。 逻辑设定请确认轴参数和从站端两者的设定。		请确认立即停止信号的连接、立即停止输入的逻辑设定是否正确。 逻辑设定请确认轴参数和从站端两者的设定。	
附属信息	无					
注意事项 / 备注	要解除异常，请关闭立即停止输入。					

事件名称	正方向极限输入检测		事件代码	644A0000Hex		
内容	正方向极限输入变为 ON。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	始终
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	遵照 “ 极限输入停止方法 ” 设定。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	检测到正方向极限输入信号。		请执行异常解除，在不超出正方向极限的范围内，向负方向执行复位动作。在轴组动作指令的执行过程中发生异常时，请将发生异常的轴组设为无效，再执行上述复位动作请调查超出极限的原因，采取合理的措施。		目的是检测正方向极限输入，因此不需要防止。但是，请编写程序，确保不会超出正方向极限输入。	
	正方向极限输入信号的连接或正方向极限输入的逻辑设定错误。		未发生正方向极限输入信号时，请修正正方向极限信号的连接、正方向极限输入的逻辑设定。 逻辑设定请确认轴参数和从站端两者的设定。		请确认正方向极限信号的连接、正方向极限输入的逻辑设定是否正确。 逻辑设定请确认轴参数和从站端两者的设定。	
附属信息	无					
注意事项 / 备注	无					

事件名称	负方向极限输入检测		事件代码	644B0000Hex		
内容	负方向极限输入变为 ON。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	始终
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	遵照 “ 极限输入停止方法 ” 设定。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	检测到负方向极限输入信号。		请执行异常解除，在不超出负方向极限的范围内，向正方向执行复位动作。在轴组动作指令的执行过程中发生异常时，请将发生异常的轴组设为无效，再执行上述复位动作请调查超出极限的原因，采取合理的措施。		目的是检测负方向极限输入，因此不需要防止。但是，请编写程序，确保不会超出负方向极限输入。	
	负方向极限输入信号的连接或负方向极限输入的逻辑设定错误。		未发生负方向极限输入信号时，请确认负方向极限信号的连接、负方向极限输入的逻辑设定。 逻辑设定请确认轴参数和从站端两者的设定。		请确认负方向极限信号的连接、负方向极限输入的逻辑设定是否正确。 逻辑设定请确认轴参数和从站端两者的设定。	
附属信息	无					
注意事项 / 备注	无					

事件名称	位置偏差不正确		事件代码		64560000Hex	
内容	将指令位置和反馈当前位置的差转换为脉冲单位时，超出 30 位。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	始终
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴的伺服 OFF		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	在从轴中，若按指令的轴移动量移动，将超出轴的最高速度，因此对指令当前位置进行了限制，以免超出最高速度。		请修正程序或电子齿轮比，确保从轴不会超过最高速度。		请创建程序或设定电子齿轮比，确保从轴不会超过最高速度。	
	从轴定位动作的追踪性差，实际动作相对于指令有延迟。		请排除可能导致从轴定位动作追踪性变差的原因。		请尽量排除可能导致从轴定位动作追踪性变差的原因。	
附属信息	无					
注意事项 / 备注	无					

事件名称	伺服 OFF 异常		事件代码		64570000Hex	
内容	因轴组发生异常，轴变为伺服 OFF。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	始终
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴的伺服 OFF		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	因轴组发生异常，轴变为伺服 OFF。		请确认异常原因并采取对策。		无	
附属信息	无					
注意事项 / 备注	本异常的作用是当发生轴异常而导致轴变为伺服 OFF 时，防止用 MC_Power （可运行）指令再次使其变为伺服 ON，因此联锁将轴视为异常。 仅当 “ 轴组停止方法 ” 中选择了 “ 立即停止，同时执行伺服 OFF （自由运行停止） ” 时发生。					

事件名称	无法计算绝对编码器当前位置		事件代码		64580000Hex	
内容	无法用断电保持的绝对编码器信息正确恢复当前位置。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	接通电源时 / 控制器重置时 / 下载时 / 伺服 ON 开始时 / 从未使用轴切换为使用轴时
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。		

系统定义变量	变量名称	数据类型	名称
	_MC_AX[*].MFaultLvl.Active	BOOL	轴 发生轻度故障

发生原因和处理	发生原因 （推测原因）	处理对策	防止再发
	• 单位转换设定、循环计数器设定或伺服驱动器设定的循环计数器设定发生变更。 • 将恢复的位置转换为脉冲单位时，带符号超出 40 位的范围	解除异常后，请执行原点复位。原点复位请在安装了绝对编码器的位置附近执行，以免恢复的位置带符号超出 40 位的范围。	变更了循环计数器上限设定值等位置相关的参数时，请重新执行原点复位。原点复位请在安装了绝对编码器的位置附近执行，以免恢复的位置带符号超出 40 位的范围。 编码器的位置带符号超出 40 位时，请勿执行 MC_Power （可运行）指令、从未使用轴切换为使用轴以及断开 / 接通电源。

附属信息	无
注意事项 / 备注	无

事件名称	多轴协调动作中的原点未确定		事件代码		64590000Hex	
内容	轴组动作中或减速停止中，逻辑轴的原点变为未确定。					
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴组的减速停止。		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	轴组动作中或减速停止中，逻辑轴的指令位置或反馈位置发生溢出或下溢，原点变为未确定。		请修正程序，使程序在指令位置 / 反馈位置不会发生溢出 / 下溢的范围内运行。		请创建程序，使程序在指令位置 / 反馈位置不会发生溢出 / 下溢的范围内运行。	
	轴组动作中或减速停止中，逻辑轴发生从站通信异常，原点变为未确定。		请采取从站通信异常的处理对策，并确定原点。		无	
	轴组动作中或减速停止中，逻辑轴的从站脱离或从站变为无效，原点变为未确定。		请将脱离或设为无效的从站再次加入，并确定原点。		轴组动作中或减速停止中，请勿使逻辑轴的从站脱离或变为无效。	
附属信息	无					
注意事项 / 备注	无					

事件名称	伺服主回路电源关闭检测		事件代码		74210000Hex	
内容	在伺服 ON 状态下关闭了伺服驱动器的主回路电源					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	伺服 ON 中始终
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴的伺服 OFF		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	在伺服 ON 状态下伺服驱动器的主回路电源断开。		请将发生本异常的轴的伺服驱动器主回路电源设为 ON，解除异常后，再设为伺服 ON。		请设为伺服 OFF 后，再将伺服驱动器的主回路电源设为 OFF。	
附属信息	无					
注意事项 / 备注	无					

事件名称	中断固定尺寸定位无中断信号			事件代码		74230000Hex	
内容	执行 MC_MoveFeed （中断固定尺寸定位）指令的过程中无中断输入。						
发生位置	运动控制功能模块		发生位置详情	轴		检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常		日志类别	系统
产生的影响	用户程序	继续	动作	相应轴的减速停止。			
系统定义变量	变量名称		数据类型		名称		
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障		
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发		
	锁定有效范围指定不合适。		相应指令的锁定有效范围指定错误时，请修正锁定有效范围的指定。		请根据动作和传感器位置的关系指定合适的锁定有效范围。		
	中断信号的配线错误。		与该指令对应的中断信号的配线错误时，请正确配线。		请确认中断信号的配线没有错误。		
	输出中断信号的传感器故障。		若不是上述两个原因，则是输出中断信号的传感器故障。请更换输出（与发生本异常的指令对应的）中断信号的传感器。		无		
附属信息	无						
注意事项 / 备注	无						

事件名称	原点复位反方向极限输入检测			事件代码		74240000Hex	
内容	原点复位动作中，检测到与原点复位方向相反的极限信号。						
发生位置	运动控制功能模块		发生位置详情	轴		检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常		日志类别	系统
产生的影响	用户程序	继续	动作	按照与原点复位的执行状态对应的停止方法停止。			
系统定义变量	变量名称		数据类型		名称		
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障		
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发		
	正方向极限输入时的动作即负方向极限输入时的动作设定为 “ 不反转 ”。		若极限输入时不视为异常，请将正方向极限输入时动作及负方向极限输入时动作设定为 “ 反转 ”。		请事先确认发生原因中记载的内容没有问题。		
	原点复位所需的输入信号传感器的配置、原点复位设定、原点复位开始位置的关系导致出现极限信号。		请重新确认输入信号传感器的配置、原点复位设定、原点复位开始位置，确保不会出现极限信号。				
	输入信号传感器配线错误、故障。		请修正或更换输入信号传感器的配线。				
附属信息	无						
注意事项 / 备注	无						

事件名称	原点复位方向极限输入检测		事件代码		74250000Hex	
内容	原点复位动作中，检测到原点复位方向的极限信号。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	按照与原点复位的执行状态对应的停止方法停止。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	正方向极限输入时的动作即负方向极限输入时的动作设定为 “ 不反转 ”。		若极限输入时不视为异常，请将正方向极限输入时动作及负方向极限输入时动作设定为 “ 反转 ”。		请事先确认发生原因中记载的内容没有问题。	
	原点复位所需的输入信号传感器的配置、原点复位设定、原点复位开始位置的关系导致出现极限信号。		请重新确认输入信号传感器的配置、原点复位设定、原点复位开始位置，确保不会出现极限信号。			
	输入信号传感器配线错误、故障。		请修正或更换输入信号传感器的配线。			
附属信息	无					
注意事项 / 备注	无					

事件名称	原点复位两个方向极限输入检测		事件代码		74260000Hex	
内容	原点复位动作中，同时检测到两侧的极限信号。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	按照与原点复位的执行状态对应的停止方法停止。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	极限信号的配线错误。		请修正极限信号的配线。		请事先确认发生原因中记载的内容没有问题。	
	极限传感器的安装位置错误。		请修正极限传感器的安装位置，以免两侧的极限信号同时变为 “ON”。			
	极限信号的接点逻辑错误。		请修正极限信号的 a 接点 /b 接点。			
	极限传感器故障。		请更换极限传感器。			
附属信息	无					
注意事项 / 备注	无					

事件名称	近原点、原点复位反方向极限输入检测		事件代码		74270000Hex	
内容	原点复位动作中，同时检测到近原点输入以及与原点复位方向相反的极限信号。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	按照与原点复位的执行状态对应的停止方法停止。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	近原点信号、极限信号的配线错误。		请修正近原点信号、极限信号的配线。		请事先确认发生原因中记载的内容没有问题。	
	近原点传感器、极限传感器的安装位置错误。		请修正近原点传感器或极限传感器的安装位置，以免近原点输入信号和极限信号同时变为 “ON”。			
	近原点信号、极限信号的接点逻辑错误。		请修正近原点传感器、极限传感器的 a 接点 /b 接点。			
	近原点传感器、极限传感器故障。		请更换近原点传感器、极限传感器。			
附属信息	无					
注意事项 / 备注	无					

事件名称	近原点、原点复位方向极限输入检测		事件代码		74280000Hex	
内容	原点复位动作中，同时检测到近原点输入以及原点复位方向的极限信号。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	按照与原点复位的执行状态对应的停止方法停止。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	近原点信号、极限信号的配线错误。		请修正近原点信号、极限信号的配线。		请事先确认发生原因中记载的内容没有问题。	
	近原点传感器、极限传感器的安装位置错误。		请修正近原点传感器或极限传感器的安装位置，以免近原点输入信号和极限信号同时变为 “ON”。			
	近原点信号、极限信号的接点逻辑错误。		请修正近原点传感器、极限传感器的 a 接点 /b 接点。			
	近原点传感器、极限传感器故障。		请更换近原点传感器、极限传感器。			
附属信息	无					
注意事项 / 备注	无					

事件名称	原点输入、原点复位反方向极限输入检测		事件代码		74290000Hex	
内容	原点复位动作中，同时检测到原点输入以及与原点复位方向相反的极限信号。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	按照与原点复位的执行状态对应的停止方法停止。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	原点输入信号、极限信号的配线错误。		请修正原点输入信号、极限信号的配线。		请事先确认发生原因中记载的内容没有问题。	
	原点输入传感器、极限传感器的安装位置错误。		请修正原点输入传感器或极限传感器的安装位置，以免原点输入信号和极限信号同时变为 “ON”。			
	原点输入信号、极限信号的接点逻辑错误。		请修正原点输入信号、极限传感器的 a 接点 /b 接点。			
	原点输入信号输出设备、极限传感器故障。		请更换原点输入信号输出设备、极限传感器。			
附属信息	无					
注意事项 / 备注	无					

事件名称	原点输入、原点复位方向极限输入检测		事件代码		742A0000Hex	
内容	原点复位动作中，同时检测到原点输入以及原点复位方向的极限信号。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	按照与原点复位的执行状态对应的停止方法停止。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	原点输入信号、极限信号的配线错误。		请修正原点输入信号、极限信号的配线。		请事先确认发生原因中记载的内容没有问题。	
	原点输入传感器、极限传感器的安装位置错误。		请修正原点输入传感器或极限传感器的安装位置，以免原点输入信号和极限信号同时变为 “ON”。			
	原点输入信号、极限信号的接点逻辑错误。		请修正原点输入信号、极限传感器的 a 接点 /b 接点。			
	原点输入信号输出设备、极限传感器故障。		请更换原点输入信号输出设备、极限传感器。			
附属信息	无					
注意事项 / 备注	无					

事件名称	原点输入掩码量不正确		事件代码	742B0000Hex	
内容	MC_Home （原点复位）指令或 MC_HomeWithParameter （参数指定原点复位）指令中，原点输入掩码量的设定值不合适。				
发生位置	运动控制功能模块	发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	按照与原点复位的执行状态对应的停止方法停止。	
系统定义变量	变量名称		数据类型	名称	
	_MC_AX[*].MFaultLvl.Active		BOOL	轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策	防止再发	
	原点复位动作模式为 “ 附近退避、原点输入掩码距离指定 ” 时，原点输入掩码距离的设定值不够从原点复位速度减速至原点复位接近速度所需的移动量。		请确认原点输入掩码量、原点复位速度、原点复位接近速度，根据 MC_Home （原点复位）指令或 MC_HomeWithParameter （参数指定原点复位）指令的动作规格变更设定，确保减速所需的移动量。	请先确认 MC_Home （原点复位）指令或 MC_HomeWithParameter （参数指定原点复位）指令的动作规格，设定原点输入掩码量、原点复位速度、原点复位接近速度，确保减速所需的移动量。	
附属信息	无				
注意事项 / 备注	无				

事件名称	无原点输入		事件代码	742C0000Hex	
内容	原点复位过程中无原点信号输入。或者在无原点确定输入的状态下，检测到极限信号。				
发生位置	运动控制功能模块	发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	按照与原点复位的执行状态对应的停止方法停止。	
系统定义变量	变量名称		数据类型	名称	
	_MC_AX[*].MFaultLvl.Active		BOOL	轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策	防止再发	
	• 原点复位过程中无原点信号输入。 • 在无原点确定输入的状态下，检测到极限信号。		请确认原点输入的设定和配线，根据 MC_Home （原点复位）指令或 MC_HomeWithParameter （参数指定原点复位）指令的动作规格修正，确保原点复位动作过程中有原点信号输入。 或者确保原点输入后能检测到极限信号。	请设定为在原点复位动作过程中有原点信号输入。 请确保原点输入后能检测到极限信号。 同时，请确认原点输入的配线没有错误。	
附属信息	无				
注意事项 / 备注	无				

事件名称	无近原点输入			事件代码	742D0000Hex	
内容	原点复位过程中无近原点信号输入。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	按照与原点复位的执行状态对应的停止方法停止。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	设定了 “ 有近原点信号 ” 的原点复位过程中无近原点信号输入。		请确认近原点输入的设定和配线，根据 MC_Home （原点复位）指令或 MC_HomeWithParameter （参数指定原点复位）指令的动作规格修正，确保原点复位过程中有近原点信号输入。		请设定为在原点复位动作过程中有近原点信号输入。同时，请确认近原点输入的配线没有错误。	
附属信息	无					
注意事项 / 备注	无					

事件名称	从站异常检测			事件代码	742F0000Hex	
内容	检测到轴中分配的 EtherCAT 从站或 NX 单元异常。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	始终
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴的伺服 OFF。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	检测到轴中分配的 EtherCAT 从站或 NX 单元异常。		请确认从站中发生的异常或 “ 从站异常代码确定 ”(94220000Hex) 中通知的从站异常代码，并采取必要的措施。		无	
附属信息	无					
注意事项 / 备注	无					

事件名称	发生轴组构成轴异常			事件代码	74300000Hex	
内容	轴组的构成轴发生异常。					
发生位置	运动控制功能模块		发生位置详情	轴组	检测时序	始终
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	立即停止了构成轴时，遵照 “ 轴组停止方法 ” 的设定动作。 其他情况下则插补轨迹停止。		
系统定义变量	变量名称		数据类型		名称	
	_MC_GRP[*].MFaultLvl.Active		BOOL		轴组 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	轴组动作过程中构成轴发生异常。		请确认构成轴组的轴的异常代码，排除异常原因。		无	
附属信息	无					
注意事项 / 备注	发生轴异常时，包含该轴的轴组将无法动作。					

事件名称	发生 MC 共通异常			事件代码		74330000Hex
内容	发生 MC 共通异常。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	始终
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴不可动作。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	发生部分停止故障等级的 MC 共通异常。		请确认发生的 MC 共通异常，排除异常原因。		无	
附属信息	无					
注意事项 / 备注	发生部分停止故障等级的 MC 共通异常时，轴及轴组将无法动作。					

事件名称	锁定位置溢出			事件代码		74340000Hex
内容	用 MC_TouchProbe （外部锁定有效）指令锁定的位置溢出。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴的减速停止。 外部锁定有效指令不可获取锁定位置。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	用 MC_TouchProbe （外部锁定有效）指令锁定的位置溢出。		请修正程序，使轴位置不会发生溢出。		请创建程序，使轴位置不会发生溢出。	
附属信息	无					
注意事项 / 备注	无					

事件名称	锁定位置下溢			事件代码		74350000Hex
内容	用 MC_TouchProbe （外部锁定有效）指令锁定的位置下溢。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴的减速停止。 外部锁定有效指令不可获取锁定位置。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	用 MC_TouchProbe （外部锁定有效）指令锁定的位置下溢。		请修正程序，使轴位置不会发生下溢。		请创建程序，使轴位置不会发生下溢。	
附属信息	无					
注意事项 / 备注	无					

事件名称	主轴同步方向异常		事件代码		74360000Hex	
内容	主轴持续向同步方向的反方向移动。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴的减速停止。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	相对于主轴和从轴的同步方向，主轴持续向反方向移动并发生溢出。		请修正程序，使同步开始后主轴的移动方向和移动量与同步方向一致。		请创建程序，使同步开始后主轴的移动方向和移动量与同步方向一致。	
附属信息	无					
注意事项 / 备注	无					

事件名称	伺服 ON 中从站脱离		事件代码		74370000Hex	
内容	伺服 ON 时，轴中分配的 EtherCAT 从站或 NX 单元发生脱离、更换或从站变为无效。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	伺服 ON 中始终
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴的伺服 OFF。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	伺服 ON 时，轴中分配的 EtherCAT 从站或 NX 单元发生脱离、更换或从站变为无效。		请重新加入相应轴中分配的 EtherCAT 从站或 NX 单元。		请在设为伺服 OFF 后再脱离、更换或将从站设为无效。	
附属信息	无					
注意事项 / 备注	无					

事件名称	固定尺寸距离溢出		事件代码		74380000Hex	
内容	MC_MoveFeed （中断固定尺寸定位）指令的中断输入后，目标位置发生溢出 / 下溢。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴的减速停止。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	将 MC_MoveFeed （中断固定尺寸定位）指令中断输入后的目标位置转换为脉冲数单位时，带符号超出 40 位的范围。		请修正程序中的指令位置输入值，使中断输入后目标位置的脉冲数不会超出脉冲数的范围。		请创建程序，确保将中断输入后的目标位置转换为脉冲数单位时，带符号不会超出 40 位的范围。	
附属信息	无					
注意事项 / 备注	无					

<table>
<tr><td>事件名称</td><td colspan="2">伺服驱动器的控制模式切换异常</td><td>事件代码</td><td colspan="2">74390000Hex</td></tr>
<tr><td>内容</td><td colspan="5">切换控制模式时，处理未在规定时间内完成。</td></tr>
<tr><td>发生位置</td><td colspan="2">运动控制功能模块</td><td>发生位置详情</td><td>检测时序</td><td>指令执行中</td></tr>
<tr><td>异常属性</td><td>重要程度</td><td>轻度故障</td><td>恢复方法</td><td>日志类别</td><td>系统</td></tr>
<tr><td>产生的影响</td><td>用户程序</td><td>继续</td><td>动作</td><td colspan="2">相应轴的伺服 OFF。</td></tr>
<tr><td rowspan="2">系统定义变量</td><td colspan="2">变量名称</td><td colspan="2">数据类型</td><td colspan="2">名称</td></tr>
<tr><td colspan="2">_MC_AX[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">轴 发生轻度故障</td></tr>
<tr><td rowspan="4">发生原因和处理</td><td colspan="2">发生原因 （推测原因）</td><td colspan="2">处理对策</td><td colspan="2">防止再发</td></tr>
<tr><td colspan="2">停止 MC_SyncMoveVelocity （周期性同步速度控制）指令时，开始在指令速度中输出 “0” 后的 10 秒以内，反馈当前速度连续 3 个周期超出最高速度的 10%。</td><td colspan="2" rowspan="2">请调整指令和负载，避免出现发生原因。</td><td colspan="2" rowspan="2">请调整指令和负载，避免出现发生原因。</td></tr>
<tr><td colspan="2">使用欧姆龙生产的伺服驱动器 G5 系列时，在 MC_TorqueControl（扭矩控制）指令停止时，10 秒以内，反馈当前速度连续 3 个周期超出最高速度的 10%。</td></tr>
<tr><td colspan="2">发出控制模式 (CSP/ CSV/ CST) 的切换指令后，伺服驱动器的模式切换未在 1 秒内完成。</td><td colspan="2">请确认伺服驱动器是否发生异常、设定是否错误，并采取必要的措施。
切换控制模式进行控制时，请在 PDO 映射中设定 “csp 用参照位置 ”。</td><td colspan="2">请确认伺服驱动器未发生异常、设定没有错误。
切换控制模式进行控制时，请在 PDO 映射中设定 “csp 用参照位置 ”。</td></tr>
<tr><td>附属信息</td><td colspan="5">无</td></tr>
<tr><td>注意事项 / 备注</td><td colspan="5">无</td></tr>
</table>

<table>
<tr><td>事件名称</td><td colspan="2">主轴位置读取异常</td><td>事件代码</td><td colspan="2">743A0000Hex</td></tr>
<tr><td>内容</td><td colspan="5">同步控制指令的主轴位置发生异常，因此无法执行同步控制指令。</td></tr>
<tr><td>发生位置</td><td colspan="2">运动控制功能模块</td><td>发生位置详情</td><td>检测时序</td><td>指令启动时 / 指令执行中</td></tr>
<tr><td>异常属性</td><td>重要程度</td><td>轻度故障</td><td>恢复方法</td><td>解除异常</td><td>日志类别</td><td>系统</td></tr>
<tr><td>产生的影响</td><td>用户程序</td><td>继续</td><td>动作</td><td colspan="3">相应从轴不可动作。相应从轴正在动作时，则减速停止。</td></tr>
<tr><td rowspan="2">系统定义变量</td><td colspan="2">变量名称</td><td colspan="2">数据类型</td><td colspan="2">名称</td></tr>
<tr><td colspan="2">_MC_AX[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">轴 发生轻度故障</td></tr>
<tr><td rowspan="5">发生原因和处理</td><td colspan="2">发生原因 （推测原因）</td><td colspan="2">处理对策</td><td colspan="2">防止再发</td></tr>
<tr><td colspan="2">同步控制指令的主轴未建立 EtherCAT 过程数据通信，或 NX 单元的 I/O 数据无法用于控制。</td><td colspan="2">主轴的 EtherCAT 主机系统变量 _EC_PDSlavTbl （过程数据通信中从站表）为 FALSE 时，请调查主轴的异常，并排除原因。
主轴为 NX 单元时，请对 NX 单元的过程数据通信中状态执行相同的处理。</td><td colspan="2">电源打开或下载后、从站通信异常的解除后，若立即执行同步控制指令，请确认 EtherCAT 主机系统变量 _EC_PDSlavTbl （过程数据通信中从站表）的主轴节点已变为 TRUE 后，再执行同步控制指令。
主轴为 NX 单元时，请对 NX 单元的过程数据通信中状态执行相同的处理。</td></tr>
<tr><td colspan="2">同步控制指令的主轴发生从站脱离或从站无效。</td><td colspan="2">请确认主轴上是否发生从站的脱离或无效化，若发生，请重新加入或将从站设为有效。</td><td colspan="2">执行同步控制指令时，请勿进行主轴的从站脱离或从站无效化。</td></tr>
<tr><td colspan="2">同步控制指令的主轴中，检测到无法计算绝对编码器当前位置 (64580000Hex)。</td><td colspan="2">请确认主轴上是否检测到无法计算绝对编码器当前位置 (64580000Hex)，若检测到，请采取异常处理措施以恢复。</td><td colspan="2">若同步控制指令的主轴为发生无法计算绝对编码器当前位置 (64580000Hex) 的轴，则不要执行。</td></tr>
<tr><td colspan="2">同步控制指令的主轴为未使用轴。</td><td colspan="2">请将主轴设为使用轴。</td><td colspan="2">执行同步控制指令时，请勿将主轴设为未使用轴。</td></tr>
<tr><td>附属信息</td><td colspan="5">无</td></tr>
<tr><td>注意事项 / 备注</td><td colspan="5">无</td></tr>
</table>

事件名称	辅助轴位置读取异常			事件代码		743B0000Hex
内容	同步控制指令的辅助轴位置发生异常，因此无法执行同步控制指令。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令启动时 / 指令执行中
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应从轴不可动作。相应从轴正在动作时，则减速停止。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	同步控制指令的辅助轴未建立 EtherCAT 过程数据通信，或 NX 单元的 I/O 数据无法用于控制。		辅助轴的 EtherCAT 主机系统变量 _EC_PDSlavTbl （过程数据通信中从站表）为 FALSE 时，请调查辅助轴的异常，并排除原因。 辅助轴为 NX 单元时，请对 NX 单元的过程数据通信中状态执行相同的处理。		电源打开或下载后、从站通信异常的解除后，若立即执行同步控制指令，请确认 EtherCAT 主机系统变量 _EC_PDSlavTbl （过程数据通信中从站表）的辅助轴节点已变为 TRUE 后，再执行同步控制指令。 辅助轴为 NX 单元时，请对 NX 单元的过程数据通信中状态执行相同的处理。	
	同步控制指令的辅助轴发生从站脱离或从站无效。		请确认辅助轴上是否发生从站的脱离或无效化，若发生，请重新加入或将从站设为有效。		执行同步控制指令时，请勿进行辅助轴的从站脱离或从站无效化。	
	同步控制指令的辅助轴中，检测到无法计算绝对编码器当前位置 (64580000Hex)。		请确认辅助轴上是否检测到无法计算绝对编码器当前位置 (64580000Hex)，若检测到，请采取异常处理措施以恢复。		若同步控制指令的辅助轴为发生无法计算绝对编码器当前位置 (64580000Hex) 的轴，则不要执行。	
	同步控制指令的辅助轴为未使用轴。		请将辅助轴设为使用轴。		执行同步控制指令时，请勿将辅助轴设为未使用轴。	
附属信息	无					
注意事项 / 备注	无					

事件名称	从站通信异常			事件代码		84400000Hex
内容	与轴中分配的 EtherCAT 从站或 NX 单元的通信发生异常。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	始终
异常属性	重要程度	轻度故障	恢复方法	解除异常	日志类别	系统
产生的影响	用户程序	继续	动作	相应轴的伺服 OFF。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].MFaultLvl.Active		BOOL		轴 发生轻度故障	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	与轴中分配的 EtherCAT 从站或 NX 单元的通信发生异常。		请在事件日志中确认发生的 EtherCAT 异常，排除异常原因后，解除本异常。		无	
附属信息	无					
注意事项 / 备注	解除本异常后，也不会解除 EtherCAT 主机功能模块的异常。即使不解除 EtherCAT 主机功能模块的异常，也可解除本异常，但此时将保持轴无效状态。					

事件名称	运动控制异常解除指令过多		事件代码		571D0000Hex *1	
内容	ResetMCError （运动控制异常解除）指令的实例数超过 100 个。					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	接通电源时 / 控制器重置时 / 下载时 / 在线编辑时
异常属性	重要程度	监视信息	恢复方法	一	日志类别	系统
产生的影响	用户程序	继续	动作	无影响。		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.Obsr.Active		BOOL		MC 共通 监视信息发生中	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	包括功能块的实例在内，用户程序中声明的 ResetMCError （运动控制异常解除）指令的实例数超过 100 个。		请修正用户程序，确保 ResetMCError （运动控制异常解除）指令的实例数在 100 个以内。请使用相同的实例，或根据异常解除对象使用 MC_Reset （轴错误复位）指令或 MC_GroupReset （轴组错误复位）指令。		记述程序时，请确保 ResetMCError （运动控制异常解除）指令的实例数在 100 个以内。	
附属信息	无					
注意事项 / 备注	无					

*1. 单元版本为 1.02 到 1.09 的 CPU 单元中发生的事件。

事件名称	位置偏差警告		事件代码		644C0000Hex	
内容	位置偏差超出位置偏差警告值。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	监视信息	恢复方法	一	日志类别	系统
产生的影响	用户程序	继续	动作	无影响。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	定位动作的追踪性差，实际动作相对于指令有延迟。		请排除可能导致定位动作追踪性变差的原因。 此外，请在不会发生问题的范围内，将 “ 位置偏差警告值 ” 变更为更大的值。		请尽量排除可能导致定位动作追踪性变差的原因。	
附属信息	无					
注意事项 / 备注	无					

事件名称	速度警告			事件代码		644D0000Hex	
内容	指令速度超出速度警告值。						
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组		检测时序	指令执行中
异常属性	重要程度	监视信息	恢复方法	—		日志类别	系统
产生的影响	用户程序	继续	动作	无影响。			
系统定义变量	变量名称		数据类型			名称	
	_MC_AX[*].Obsr.Active		BOOL			轴 监视信息发生中	
	_MC_GRP[*].Obsr.Active		BOOL			轴组 监视信息发生中	
发生原因和处理	发生原因 （推测原因）		处理对策			防止再发	
	指令速度超出速度警告值。		请调查超出速度警告值的原因，采取合理的措施。 此外，请在不会发生问题的范围内，将 “ 速度警告值 ” 变更为更大的值。			（目的是检测到超出速度警告值，因此不需要防止）	
附属信息	无						
注意事项 / 备注	可将重要程度变更为轻度故障。变更为轻度故障后的恢复方法为 “ 解除异常 ”，动作为 “ 相应轴 / 轴组的减速停止 ”。						

事件名称	加速度警告			事件代码		644E0000Hex	
内容	指令加速度超出加速度警告值。						
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组		检测时序	指令执行中
异常属性	重要程度	监视信息	恢复方法	—		日志类别	系统
产生的影响	用户程序	继续	动作	无影响。			
系统定义变量	变量名称		数据类型			名称	
	_MC_AX[*].Obsr.Active		BOOL			轴 监视信息发生中	
	_MC_GRP[*].Obsr.Active		BOOL			轴组 监视信息发生中	
发生原因和处理	发生原因 （推测原因）		处理对策			防止再发	
	指令加速度超出加速度警告值。		请调查超出加速度警告值的原因，采取合理的措施。 此外，请在不会发生问题的范围内，将 “ 加速度警告值 ” 变更为更大的值。			（目的是检测到超出加速度警告值，因此不需要防止）	
附属信息	无						
注意事项 / 备注	可将重要程度变更为轻度故障。变更为轻度故障后的恢复方法为 “ 解除异常 ”，动作为 “ 相应轴 / 轴组的减速停止 ”。						

事件名称	减速度警告			事件代码	644F0000Hex	
内容	指令减速度超出减速度警告值。					
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序	指令执行中
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	无影响。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
	_MC_GRP[*].Obsr.Active		BOOL		轴组 监视信息发生中	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	指令减速度超出减速度警告值。		请调查超出减速度警告值的原因，采取合理的措施。 此外，请在不会发生问题的范围内，将 " 减速度警告值 " 变更为更大的值。		（目的是检测到超出减速度警告值，因此不需要防止）	
附属信息	无					
注意事项 / 备注	可将重要程度变更为轻度故障。变更为轻度故障后的恢复方法为 " 解除异常 "，动作为 " 相应轴 / 轴组的减速停止 "。					

事件名称	正方向扭矩警告			事件代码	64500000Hex	
内容	扭矩指令值超出正方向扭矩警告值。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	无影响。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	扭矩指令值超出正方向扭矩警告值。		请调查超出扭矩警告值的原因，采取合理的措施。 此外，请在不会发生问题的范围内，将 " 正方向扭矩警告值 " 变更为更大的值。		（目的是检测到超出扭矩警告值，因此不需要防止）	
附属信息	无					
注意事项 / 备注	可将重要程度变更为轻度故障。变更为轻度故障后的恢复方法为 " 解除异常 "，动作为 " 相应轴的减速停止 "。					

事件名称	负方向扭矩警告			事件代码	64510000Hex	
内容	扭矩指令值超出负方向扭矩警告值。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	指令执行中
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	无影响。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	扭矩指令值超出负方向扭矩警告值。		请调查超出扭矩警告值的原因，采取合理的措施。 此外，请在不会发生问题的范围内，将 " 负方向扭矩警告值 " 变更为更大的值。		（目的是检测到超出扭矩警告值，因此不需要防止）	
附属信息	无					
注意事项 / 备注	可将重要程度变更为轻度故障。变更为轻度故障后的恢复方法为 " 解除异常 "，动作为 " 相应轴的减速停止 "。					

事件名称	指令位置溢出		事件代码	64520000Hex		
内容	指令位置对应的脉冲数溢出。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	始终
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	位置不会更新，继续动作。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	线性模式时，将指令位置转换为脉冲单位时，带符号超出 40 位的上限值。		请修正程序中的指令位置输入值或变更电子齿轮比的设定，使相应指令中的脉冲数不会超出脉冲数的范围。 若要从溢出状态恢复，请执行 “ 当前位置变更 ” 或 “ 原点复位 ”。		请确认电子齿轮比的设定和目标位置的设定值，确保转换后的脉冲数带符号不超过 40 位的范围。	
附属信息	无					
注意事项 / 备注	可将重要程度变更为轻度故障。变更为轻度故障后的恢复方法为 “ 解除异常 ”，动作为 “ 相应轴的减速停止 ”。					

事件名称	指令位置下溢		事件代码	64530000Hex		
内容	指令位置对应的脉冲数下溢。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	始终
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	位置不会更新，继续动作。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	线性模式时，将指令位置转换为脉冲单位时，带符号超出 40 位的下限值。		请修正程序中的指令位置输入值或变更电子齿轮比的设定，使相应指令中的脉冲数不会超出脉冲数的范围。 若要从下溢状态恢复，请执行 “ 当前位置变更 ” 或 “ 原点复位 ”。		请确认电子齿轮比的设定和目标位置的设定值，确保转换后的脉冲数带符号不超过 40 位的范围。	
附属信息	无					
注意事项 / 备注	可将重要程度变更为轻度故障。变更为轻度故障后的恢复方法为 “ 解除异常 ”，动作为 “ 相应轴的减速停止 ”。					

事件名称	反馈位置溢出			事件代码	64540000Hex	
内容	反馈位置对应的脉冲数溢出。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	始终
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	位置不会更新，继续动作。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	将反馈位置转换为脉冲单位时，带符号超出 40 位的上限值。		请修正程序中的反馈位置或变更电子齿轮比的设定，使相应指令中脉冲数不会超出脉冲数的范围，目标位置相对于脉冲数的范围有足够的余量。 若要从溢出状态恢复，请执行 “ 当前位置变更 ” 或 “ 原点复位 ”。		请确认电子齿轮比的设定和目标位置的设定值，确保转换后的脉冲数带符号在 40 位的范围内有足够的余量。	
附属信息	无					
注意事项 / 备注	可将重要程度变更为轻度故障。变更为轻度故障后的恢复方法为 “ 解除异常 ”，动作为 “ 相应轴的减速停止 ”。					

事件名称	反馈位置下溢			事件代码	64550000Hex	
内容	反馈位置对应的脉冲数下溢。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	始终
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	位置不会更新，继续动作。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因（推测原因）		处理对策		防止再发	
	将反馈位置转换为脉冲单位时，带符号超出 40 位的下限值。		请修正程序中的反馈位置或变更电子齿轮比的设定，使相应指令中脉冲数不会超出脉冲数的范围，目标位置相对于脉冲数的范围有足够的余量。 若要从下溢状态恢复，请执行 “ 当前位置变更 ” 或 “ 原点复位 ”。		请确认电子齿轮比的设定和目标位置的设定值，确保转换后的脉冲数带符号在 40 位的范围内有足够的余量。	
附属信息	无					
注意事项 / 备注	可将重要程度变更为轻度故障。变更为轻度故障后的恢复方法为 “ 解除异常 ”，动作为 “ 相应轴的减速停止 ”。					

事件名称	从站监视信息检测		事件代码		74320000Hex	
内容	检测到 EtherCAT 从站或 NX 单元的警告。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	始终
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	无影响。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	轴中分配的 EtherCAT 从站或 NX 单元检测到警告。		请确认从站的警告代码，排除异常原因。		无	
附属信息	附属信息 1：驱动器警告代码					
注意事项 / 备注	可将重要程度变更为轻度故障。变更为轻度故障后的恢复方法为 “ 解除异常 ”，动作为 “ 相应轴的减速停止 ”。					

事件名称	不可执行凸轮表保存指令		事件代码		743C0000Hex	
内容	其他操作正在访问非易失性存储器，因此无法执行凸轮表文件的保存。					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	指令启动时
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	无影响。		
系统定义变量	变量名称		数据类型		名称	
	_MC_COM.Obsr.Active		BOOL		MC 共通 监视信息发生中	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	其他操作 （通过 Sysmac Studio 传送、数据追踪等）正在访问非易失性存储器时，执行了 MC_SaveCamTable （凸轮表保存）指令。		请重新执行 MC_SaveCamTable （凸轮表保存）指令。		无	
附属信息	无					
注意事项 / 备注	无					

事件名称	达到共混中继速度所需的移动量不足通知		事件代码		94200000Hex	
内容	加减速至共混中继速度所需的移动量不足					
发生位置	运动控制功能模块		发生位置详情	轴 / 轴组	检测时序	指令多重启动开始时
异常属性	重要程度	监视信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	无影响。		
系统定义变量	变量名称		数据类型		名称	
	_MC_AX[*].Obsr.Active		BOOL		轴 监视信息发生中	
	_MC_GRP[*].Obsr.Active		BOOL		轴组 监视信息发生中	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	“ 加减速超限 ” 设定为 “ 剧烈加减速 （将共混切换为缓冲） ” 的情况下，Blending 指定时，创建曲线后发现发生加减速超限，因此变更为 Buffered 处理。		不想变更为 Buffered 处理时，请在 “ 加减速超限 ” 中设定为 “ 剧烈加减速 （将共混切换为缓冲） ” 以外。		不想变更为 Buffered 处理时，请在 “ 加减速超限 ” 中设定为 “ 剧烈加减速 （将共混切换为缓冲） ” 以外。	
	Blending 指定时，已到达目标位置，无法正确绘制曲线，因此变更为 Buffered 处理。		变更为 Buffered 处理后，若发生以外动作，请修正程序，避免发生左边的原因。		变更为 Buffered 处理后，若发生以外动作，请创建程序，避免发生左边的原因。	
附属信息	无					
注意事项 / 备注	可将重要程度变更为轻度故障。变更为轻度故障后的恢复方法为 “ 解除异常 ”，动作为 “ 相应轴的减速停止 ”。					

事件名称	从 MC 调试画面执行异常解除		事件代码		94210000Hex	
内容	在 Sysmac Studio 的调试画面中执行了异常解除。					
发生位置	运动控制功能模块		发生位置详情	MC 共通	检测时序	解除 MC 调试异常时
异常属性	重要程度	一般信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	无影响。		
系统定义变量	变量名称		数据类型		名称	
	无		—		—	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	在 Sysmac Studio 的调试画面中执行了异常解除。		—		—	
附属信息	附属信息 1：执行结果　0000_0000Hex: 全部解除成功、 0000_0001Hex: 全部解除失败					
注意事项 / 备注	无					

事件名称	从站异常代码确定			事件代码		94220000Hex
内容	因发生 “ 从站异常检测 ”，发出了异常代码通知。					
发生位置	运动控制功能模块		发生位置详情	轴	检测时序	发生 “ 从站异常检测 (742F0000Hex)” 后
异常属性	重要程度	一般信息	恢复方法	—	日志类别	系统
产生的影响	用户程序	继续	动作	无影响。		
系统定义变量	变量名称		数据类型		名称	
	无		—		—	
发生原因和处理	发生原因 （推测原因）		处理对策		防止再发	
	因发生 “ 从站异常检测 (742F0000Hex)”，从站发出了异常代码通知。		本异常为随 “ 从站异常检测 (742F0000Hex)” 发生的信息。请确认附属信息中的从站异常代码，采取必要的措施。		无	
附属信息	附属信息 1：从站异常代码					
注意事项 / 备注	使用欧姆龙生产的伺服驱动器 G5 系列时，附属信息后 2 位中将显示伺服驱动器的异常代码 （错误显示 No. 的主编号）。 例如，附属信息中显示 “FF13” 时，表示伺服驱动器上发生错误显示 No.13 （主回路电源电压不足）。					

11-2-3 异常现象的推测原因和处理措施

本章介绍首次使用 MC 功能模块时，或开始运行后发生的故障之处理方法。

事前确认项目

发生异常时，请在确认以下项目后进行调查。

分类	确认位置
设置环境	周围是否有粉尘？
	周围是否存在有可能混入的导电性异物 （金属、碳等）？
	环境维度是否较高？ （与规格中的使用环境温度比较）
	周围是否潮湿？ （湿度 · 使用水等情况）
	周围是否有腐蚀性气体？ （酸、盐、硫磺等）
	周围是否有干扰源 （焊接机、变频器等）？
配线	电源线和信号线是否在同一配管 （线槽）中？
	接地线是否牢固连接？
	电源上是否安装有干扰滤波器？
变化	最近是否进行过扩展施工 （焊接施工）？
	最近是否追加过电源设备？
	系统 （包含程序）是否发生过改造 / 追加？
事故	附近是否受过雷击？
	是否发生过短路事故 / 漏电断路器关断？
	是否发生过停电？

现象和对策

下面介绍与欧姆龙生产的伺服驱动器 G5 系列组合使用时的现象和对策。

发生意外动作时，可能是因为通过 Sysmac Studio 向 CPU 单元传送的各种参数设定和凸轮数据不正确。

或者程序和 MC 功能模块之间的变量交换未能正常进行。

请通过 Sysmac Studio 的数据追踪功能确认变量的交换时间是否正确。

现象	推测原因	确认位置	对策
电机无法锁定。	MC 功能模块未向伺服驱动器输出操作指令。	请确认程序中是否正在执行 MC_Power （可运行）指令。	请修正程序。
	驱动器的设定异常	请确认驱动器的设定。	请正确设定驱动器。
电机不旋转。	伺服驱动器的驱动禁止输入为有效。	请确认伺服驱动器的驱动禁止输入。	请解除伺服驱动器的驱动禁止输入。 请将伺服驱动器的驱动禁止输入设定为不使用。
	驱动器发生异常	请确认驱动器的异常。	发生异常时，请按照指示操作。
	机械轴锁定。	请确认是否接触到机械限位或发生机械卷入。	请手动解除机械轴的锁定。

现象	推测原因	确认位置	对策
电机不旋转。 （接上页）	CPU 单元发生故障	—	请更换 CPU 单元。
无法进行原点复位。	发生异常	请确认异常的内容。	发生异常时，请按照指示操作。
	近原点输入的误配线	请通过轴变量的轴输入信息确认近原点输入传感器的 ON/OFF。	请正确配线。
	原点输入未正确配线。	请确认原点输入的配线。	请正确配线。
	旋转方向和极限输入的方向不一致	在极限处未反转而动作至机械限位时，请通过轴变量的轴输入信息确认极限输入的 ON/OFF。	请正确连接极限输入。
	极限输入的误配线	请确认极限输入的配线。	请正确配线。
	位置等待不变为 FALSE。	请确认伺服驱动器的增益是否较低。 请确认位置宽度是否较窄。	请提高伺服驱动器的增益。 请扩大位置宽度。
	原点复位接近速度过快。	请确认原点复位接近速度。	请降低 MC 功能模块的原点复位接近速度。
	轴参数未正确设定。	请通过 Sysmac Studio 确认轴参数。	正确设定轴参数后，请下载到 MC 功能模块中。
	CPU 单元发生故障	—	请更换 CPU 单元。
原点复位后的原点时常变化。	耦合器等机械松动。	请用马克笔在电机的轴和耦合部、其他的机械结合部做好标记，确认没有错位。	请对错位的结合部加固。
	Z 相无余量。 原点输入信号无余量。	若为接近伺服电机旋转 1 圈 （编码器每转 1 圈的脉冲数）的值或零附近的值，因传感器导入时间的微妙变化，原点复位时会产生电机旋转 1 圈对应的错位。	请拆下电机的耦合器，使 Z 相余量的脉冲变为伺服电机每转 1 圈 （编码器每转 1 圈的脉冲数）的 1/2 左右，然后错开 1/4 圈左右，再次执行原点复位。
电机旋转不稳定	伺服电机的电机动力线 / 编码器线的配线错误 / 缺相等	请确认电机动力线 / 编码器线的配线。	请正确配线。
	连接电机轴和机械系统的耦合器偏心螺钉忘记拧紧，因滑轮、齿轮的咬合发生负载扭矩变动	请确认机械。 请在无负载 （从耦合器上拆下前面的机械）的状态下旋转试试。	请修正和调整机械。
	增益调整不充分	—	请执行伺服电机的自动调谐。 请手动调整伺服电机的增益。
	伺服电机选择错误 （无法调整）	请重新选择电机 （确认扭矩和惯量比）。	请变更为最合适的电机。

现象	推测原因	确认位置	对策
电机旋转不稳定（接上页）	伺服电机的电机轴承破损	请关闭伺服驱动器的电源，若为带制动器的电机，则在打开制动器电源后解除制动器，在拆下电机动力线的状态下（因为可能有动态制动器的作用），手动旋转电机的输出轴。	请更换伺服电机。
	伺服电机的电机线圈断线	请用测试仪确认电机动力线的 U/V/W 间的电阻值。 若未达到平衡，则有异常。	请更换伺服电机。
旋转方向为反方向	伺服驱动器的设定为反向旋转。	以点动进给动作，若与点动进给方向相反，则伺服驱动器的旋转方向设定为反方向。或者确认反馈信号（A/B 相）是否相反，参数中是否设定为了反向旋转。	请重新正确地设定伺服驱动器的旋转方向。
	（原点复位时） 设定近原点传感器极性和近原点输入极性的轴参数不一致	请重新确认轴参数和传感器的极性。	请重新设定为正确的轴参数。
	（原点复位时） 近原点输入的误配线	请通过轴变量的轴输入信息确认近原点输入传感器的 ON/OFF。	请正确连接近原点输入。
无法启动、不出现定位完成会延迟。	位置宽度过窄，无法进入定位完成范围（若未达到定位宽度范围内，前面的动作将无法完成，因此下次会无法启动）。	—	请扩大位置宽度。
	伺服驱动器的增益较低。	—	请调整伺服驱动器的增益。
	停止时受外力作用，未进入位置宽度内。	请通过轴变量的轴输入信息，确认指令当前位置和反馈当前位置的差是否在位置宽度内。	如推压控制等，停止时不会进入位置宽度的情况下，可使用偏差计数器复位，强制完成定位。
发出异常声音。	机械的振动	请检查机械的可动部位是否有异物侵入、破损、变形或松动。	请修正不良位置。
	伺服驱动器的增益调整不充分 （增益过高）	—	请执行自动调谐。 请手动调整（降低）增益。
	电机选择错误 （无法调整）	请重新选择电机（确认扭矩和惯量比）。	请变更为最合适的电机。
	连接电机轴和机械的耦合器偏心。	—	请调整电机 / 机械的安装。

现象	推测原因	确认位置	对策
电机轴摇晃。	增益调整不充分（增益过低）	—	请执行自动调谐。 请手动调整 （提高）增益。
	由于机械刚性较低，无法调整增益。	在垂直轴、平面关节型机器人、堆垛机等轴有弯曲或受到扭转载荷的系统中特别容易发生。	请提高机械刚性。 请重新调整增益。
	发生粘滑运动 （粘性较高的静摩擦）的机械构造	—	请执行自动调谐。 请手动调整增益。
	电机选择错误（无法调整）	请选择电机 （确认扭矩和惯量比）。	请变更为最合适的电机。
	故障	—	请更换驱动器。 请更换电机。
位置偏移。	定位前原点位置发生错位。	请参考 “ 原点复位后的原点时常变化 ” 栏。	请参考 “ 原点复位后的原点时常变化 ” 栏。
	因焊接机、变频器的干扰导致误动作	请确认附近是否有焊接机、变频器等。	请远离焊接机、变频器等。
	机械系统的错位	请确认错位是否会叠加 （在机械结合部做好标记，确认错位）。	请加固机械的固定部位。
无法使用 Sysmac Studio 的 MC 调试功能。	正在其他 Sysmac Studio 上使用 MC 调试功能。	请确认同一个 CPU 单元上是否连接了其他支持单元。	请结束其他 Sysmac Studio 的 MC 调试功能。

A

附录

记载欧姆龙生产的伺服驱动器 G5 系列对象的设定值和连接方法。

A-1 与伺服驱动器连接 A-2
A-1-1 与伺服驱动器配线 A-2
A-1-2 伺服驱动器的设定 A-2
A-2 与编码器输入终端连接 A-12
A-2-1 与编码器输入终端配线 A-12
A-2-2 编码器输入终端的设定 A-12
A-3 与 NX 单元连接 A-18
A-4 PDS 状态变化 A-19
A-4-1 PDS 状态控制方式 A-20
A-4-2 主回路电源 OFF 检测 A-21
A-5 术语说明 A-22
A-5-1 NJ/NX 系列通用 A-22
A-5-2 运动控制功能 A-23
A-5-3 EtherCAT 通信 A-24
A-6 版本相关信息 A-25

A-1 与伺服驱动器连接

介绍与欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型的连接。

A-1-1 与伺服驱动器配线

与伺服驱动器通过 EtherCAT 通信连接。

连接方法请参考 《NJ/NX 系列 CPU 单元内置 EtherCAT 端口 用户手册 （SBCD-CN5-358）》。

A-1-2 伺服驱动器的设定

介绍与 MC 功能模块的控制对象 —— 欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型连接时，伺服驱动器侧的设定概要。

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

推荐伺服驱动器

可使用 MC 功能模块所有功能的伺服驱动器的单元版本如下表所示。

厂家	对象型号	对象单元版本
欧姆龙	R88D-KN□□□-ECT	Ver.2.1 以上
	R88D-KN□□□-ECT-L	Ver.1.1 以上

参考

- 除了推荐对象外，还可使用欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型的单元版本。可使用的功能因伺服驱动器的规格不同而异。请通过 Sysmac Studio 设定使用的功能和对象字典。
- 欧姆龙生产的伺服驱动器 G5 系列中的 R88D-KN□□□-ECT-R （单元版本 1.0）为仅支持位置控制 (cyclic synchronous position) 的伺服驱动器。

  可用于不需要速度控制 (cyclic synchronous velocity) 或扭矩控制 (cyclic synchronous torque) 的用途。

  功能详情请参考 《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型 用户手册 (SBCE-CN5-360)》。
- 使用欧姆龙生产的伺服电机 / 驱动器 G5 系列中的气缸型且单元版本为 2.0 以下时，请将节点地址开关设定为 “00” 以外。若设定为 “00”，网络配置将出现异常。
- 欧姆龙生产的伺服驱动器 G5 系列不同单元版本的差异请参考 《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型 用户手册 （SBCE-CN5-365B 以上）》。

外部输入信号的分配

在 MC 功能模块中，将伺服驱动器的通用输入作为以下输入信号使用。

- 立即停止输入
- 正方向极限输入
- 负方向极限输入
- 近原点输入
- 外部锁定功能的触发信号 （锁定输入 1/ 锁定输入 2）

● 正方向极限输入 / 负方向极限输入 / 近原点输入的分配

在欧姆龙生产的伺服驱动器 G5 系列的出厂设定中，各输入信号如下表分配。

信号名称	输入信号
立即停止输入	伺服驱动器的通用输入 1 （IN1: CN1 连接器的针号 5 b 接）
正方向极限输入	伺服驱动器的通用输入 2 （IN2: CN1 连接器的针号 7 b 接）*1
负方向极限输入	伺服驱动器的通用输入 3 （IN3: CN1 连接器的针号 8 b 接）*2
近原点输入	伺服驱动器的通用输入 4 （IN4: CN1 连接器的针号 9 a 接）

*1. 伺服驱动器上的信号名称为 “ 正转侧驱动禁止输入 ”。
*2. 伺服驱动器上的信号名称为 “ 反转侧驱动禁止输入 ”。

● 外部锁定功能的触发信号分配

在欧姆龙生产的伺服驱动器 G5 系列的出厂设定中，将下表的输入信号分配为外部锁定功能的触发信号。

MC_TouchProbe 指令的输入变量 “TriggerInput （触发输入条件） ” 的设定			外部锁定功能的触发信号
Mode	InputDrive	LatchID	
0: mcDrive	0: mcEncoderMark	-	编码器的 Z 相
	1: mcEXT	1: mcLatch1	伺服驱动器的通用输入 7 （IN7: CN1 连接器的针号 12 a 接）*1
		2: mcLatch2	伺服驱动器的通用输入 6 （IN6: CN1 连接器的针号 11 a 接）*2
1: mcController	-	-	TriggerVariable （触发变量）中指定的变量

*1. 伺服驱动器上的信号名称为 “ 外部锁定输入 1”。
*2. 伺服驱动器上的信号名称为 “ 外部锁定输入 2”。

齿隙补偿功能

MC 功能模块无齿隙补偿功能。
需要齿隙补偿时，使用伺服驱动器的补偿功能。
需要在伺服驱动器上设定的对象如下所示。

索引	名称	说明
3704Hex	齿隙补偿选择	设定位置控制时齿隙补偿的有效 / 无效及补偿时的动作方向。 出厂设定为 “ 无效 ”。
3705Hex	齿隙补偿量	设定位置控制时的齿隙补偿量。
3706Hex	齿隙补偿时间常数	设定位置控制时的齿隙补偿时间常数。

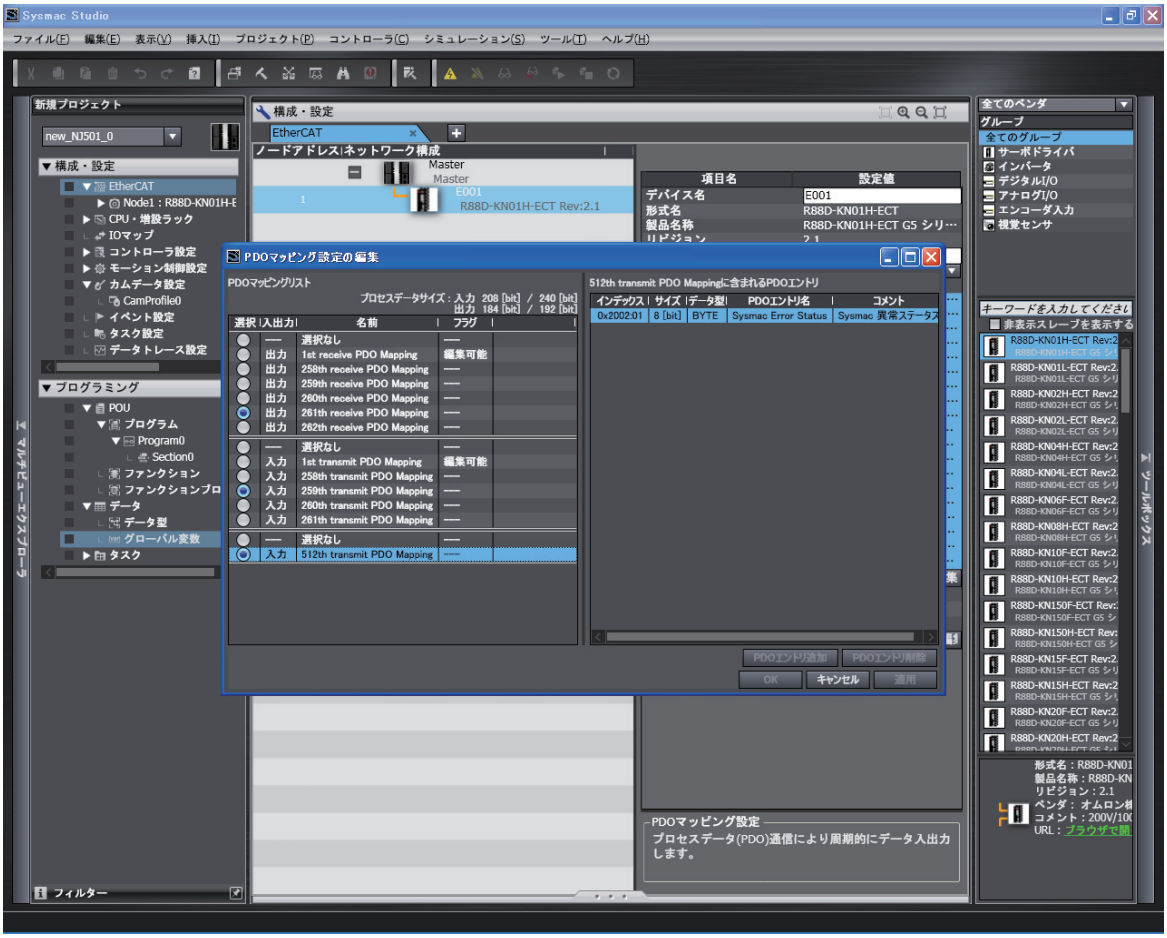
齿隙补偿功能的详情请参考 🕮《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型 用户手册 (SBCE-CN5-365)》及 🕮《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型 用户手册 (SBCE-CN5-366)》。

PDO 映射

介绍作为 MC 功能模块的伺服轴控制时所需的 PDO 映射。

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

要使用的 PDO 映射在 Sysmac Studio 的［EtherCAT］构成编辑画面的［PDO 映射设定的编辑］中选择。



Sysmac Studio 默认选择以下 PDO 映射。

RxPDO (1704Hex)	控制字 (6040Hex)、目标位置 (607AHex)、目标速度 (60FFHex)、目标扭矩 (6071Hex)、操作模式 (6060Hex)、锁定功能 (60B8Hex)、最大曲线速度 (607FHex)、正转扭矩限制值 (60E0Hex)、反转扭矩限制值 (60E1Hex)
TxPDO (1B02Hex)	错误代码 (603FHex)、状态字 (6041Hex)、反馈位置 (6064Hex)、反馈扭矩 (6077Hex)、操作模式显示 (6061Hex)、锁定状态 (60B9Hex)、锁定位置 1(60BAHex)、锁定位置 2(60BCHex)、数字输入 (60FDHex)

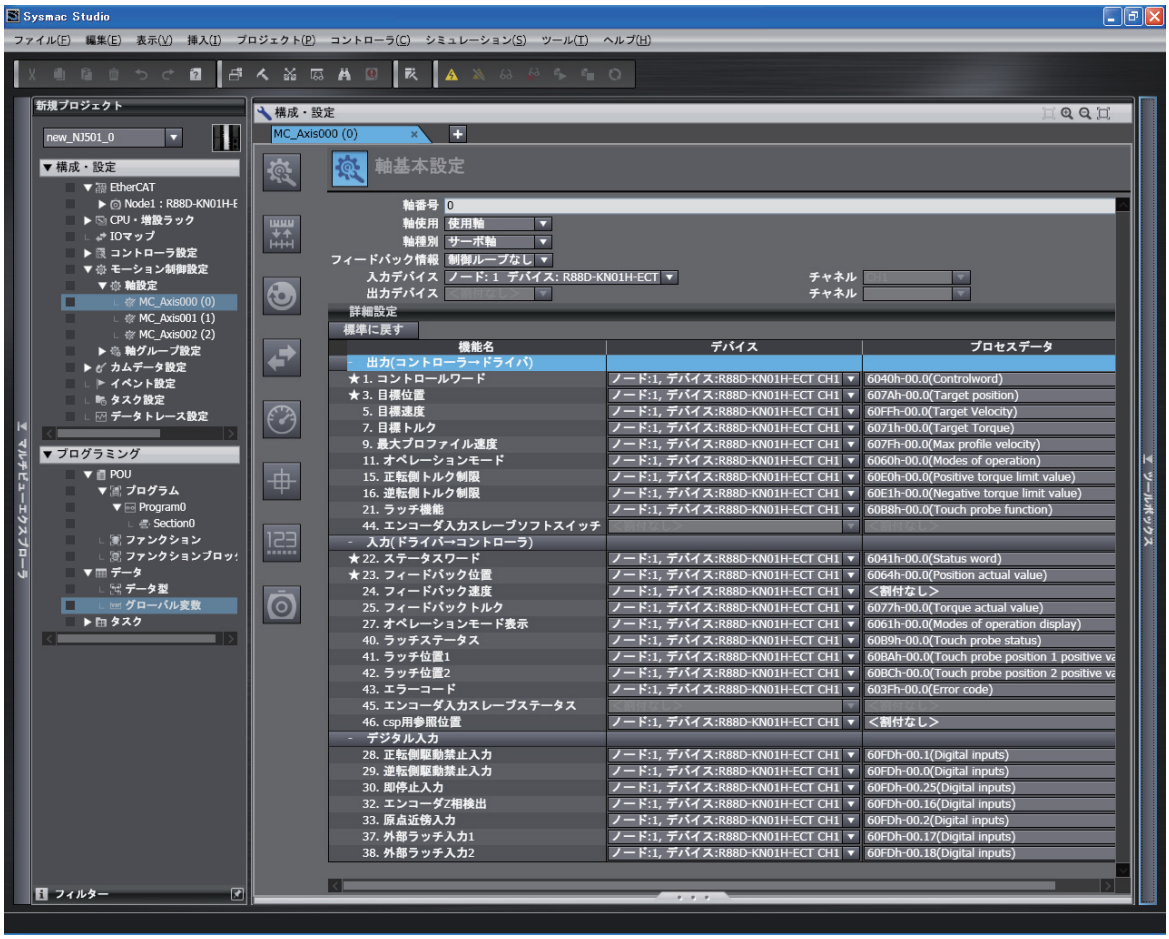
参考

- 使用推荐的欧姆龙产伺服驱动器 R88D-KN□□□-ECT （Ver.2.1 以上）或 R88D-KN□□□-ECT-L （Ver.1.1 以上）时， PDO 映射可使用 Sysmac Studio 的默认设定。
- 用欧姆龙生产的 R88D-KN□□□-ECT 进行全闭环控制时， RxPDO 请选择 1701Hex 或对象总大小为 12 字节以下的 1600Hex （Ver.2.1 以上时）。

MC 功能模块的功能和过程数据的关系

MC 功能模块拥有的功能与过程数据的对象信息有关。
因使用 MC 功能模块的 EtherCAT 从站的构成或使用的功能不同，两者的关系可能需要变更。

在 Sysmac Studio 中单击 ［轴基本设定］画面的 ［详细设定］，将打开设定画面。



参考

使用推荐的欧姆龙产伺服驱动器 R88D-KN□□□-ECT （Ver.2.1 以上）或 R88D-KN□□□-ECT-L （Ver.1.1 以上）时，功能和过程数据的关系可使用 Sysmac Studio 的默认设定。

● 输出 （控制器→驱动器）的设定

设定 MC 功能模块向伺服驱动器发出的指令数据。

以下为 Sysmac Studio 的默认设定。（O 标记表示必需对象）

	功能名称	过程数据	说明
O	控制字	6040Hex-00.0 (Controlword)	控制伺服驱动器的状态所需的数据。 请设定为 [6040Hex: Controlword]。
O	目标位置	607AHex-00.0 (Target position)	位置控制中的目标位置。 请设定为 [607AHex: Target position]。
	目标速度	60FFHex-00.0 (Target velocity)	速度控制中的目标速度。 MC_SyncMoveVelocity （周期同步速度控制）指令等，以周期同步速度模式向驱动器输出时需要。 通常请设定为 [60FFHex: Target velocity]。
	目标扭矩	6071Hex-00.0 (Target torque)	扭矩控制中的目标扭矩。 MC_TorqueControl （扭矩控制）指令等，以周期同步扭矩模式向驱动器输出时需要。 通常请设定为 [6071Hex: Target torque]。
	最大曲线速度	607FHex-00.0 (Max profile velocity)	扭矩控制时的速度限制值。 MC_TorqueControl （扭矩控制）指令等，在周期同步扭矩模式中进行速度限制时需要。 通常请设定为 [607FHex: Max profile velocity]。
	操作模式	6060Hex-00.0 (Modes of operation)	切换控制模式所需的数据。 MC_SyncMoveVelocity （周期同步速度控制）指令或 MC_TorqueControl （扭矩控制）指令等，切换为周期同步位置模式以外的控制模式时需要。 通常请设定为 [6060Hex: Modes of operation]。*1
	正转侧扭矩限制值	60E0Hex-00.0 (Positive torque limit value)	正方向的扭矩限制值。 MC_SetTorqueLimit （扭矩限制）指令等，限制驱动器的输出扭矩时需要。 通常请设定为 [60E0Hex: Positive torque limit value]。
	反转侧扭矩限制值	60E1Hex-00.0 (Negative torque limit value)	负方向的扭矩限制值。 MC_SetTorqueLimit （扭矩限制）指令等，限制驱动器的输出扭矩时需要。 通常请设定为 [60E1Hex: Negative torque limit value]。
	锁定功能	60B8Hex-00.0 (Touch probe function)	控制锁定功能所需的数据。 MC_Home （原点复位）指令、 MC_HomeWithParameter （参数指定原点复位）指令、 MC_MoveFeed （中断固定尺寸定位）指令、 MC_TouchProbe （外部锁定有效）指令、 MC_MoveLink （梯形模式凸轮）指令等，使用锁定功能时需要。 通常请设定为 [60B8Hex: Touch probe function]。

*1. 请设定为 [6060Hex: Modes of operation] 时，请同时设定 [6061Hex: Modes of operation display]。若只设定其中的一个，将无法正常动作。

使用注意事项

- 除了推荐的伺服驱动器外，连接了欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型的单元版本时，可能会没有支持的功能。
  请确认所连接伺服驱动器的手册。
- 要变更设定时，请确认 MC 功能模块的功能和过程数据的组合是否能实现预期的动作。
- 使用欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型以外的产品时，请务必设定操作模式 (6060Hex)。
- 用欧姆龙生产的伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型进行全闭环控制时，设定时请确保对象的总大小为 12 字节以下。

● 输入 （驱动器→控制器）的设定

伺服驱动器到 MC 功能模块的状态数据。

以下为 Sysmac Studio 的默认设定。（○ 标记表示必需对象）

	功能名称	过程数据	说明
○	状态字	6041Hex-00.0 (Statusword)	伺服驱动器的状态。请设定为 [6041Hex: Statusword]。
○	反馈位置	6064Hex-00.0 (Position actual value)	当前位置的显示。请设定为 [6064Hex: Position actual value]。
	反馈速度	不设定 *1	当前速度的显示。使用时，请设定为 [606CHex: Velocity actual value]。
	反馈扭矩	6077Hex (Torque actual value)	当前扭矩的显示。 MC_TorqueControl （扭矩控制）指令等，以周期同步扭矩模式向驱动器输出时需要。 通常请设定为 [6077Hex: Torque actual value]。
	操作模式显示	6061Hex-00.0 (Modes of operation display)	控制模式的显示。 MC_SyncMoveVelocity （周期同步速度控制）指令或 MC_TorqueControl （扭矩控制）指令等，切换为周期同步位置模式以外的控制模式时需要。 通常请设定为 [6061Hex: Modes of operation display]。 *2
	锁定状态	60B9Hex-00.0 (Touch probe status)	显示锁定功能的状态。 MC_Home （原点复位）指令、 MC_HomeWithParameter （参数指定原点复位）指令、 MC_MoveFeed （中断固定尺寸定位）指令、 MC_TouchProbe （外部锁定有效）指令、 MC_MoveLink （梯形模式凸轮）指令等，使用锁定功能时需要。 通常请设定为 [60B9Hex: Touch probe status]。
	锁定位置 1	60BAHex-00.0 (Touch probe pos1 pos value)	通过锁定功能 1 锁定的位置。 MC_Home （原点复位）指令、 MC_HomeWithParameter （参数指定原点复位）指令、 MC_MoveFeed （中断固定尺寸定位）指令、 MC_TouchProbe （外部锁定有效）指令、 MC_MoveLink （梯形模式凸轮）指令等，使用锁定功能时需要。通常请设定为 [60BAHex: Touch probe pos1 pos value]。

功能名称	过程数据	说明
锁定位置 2	60BCHex-00.0 (Touch probe pos2 pos value)	通过锁定功能 2 锁定的位置。 MC_Home （原点复位）指令、 MC_HomeWithParameter （参数指定原点复位）指令、 MC_MoveFeed （中断固定尺寸定位）指令、 MC_TouchProbe （外部锁定有效）指令、 MC_MoveLink （梯形模式凸轮）指令等，使用锁定功能时需要。通常请设定为 [60BCHex: Touch probe pos2 pos value]。
错误代码	603FHex-00.0 (Error code)	驱动器中的异常代码。 通常请设定为 [603FHex: Error code]。
csp 用参照位置	不设定	切换 csp 模式时的参照位置。使用速度控制模式 (CSV) 的指令或使用扭矩控制模式 (CST) 的指令中参照的数据。 可使用欧姆龙生产的伺服电机 / 驱动器 G5 系列中的气缸型且单元版本 2.1 以上的单元。*3 在欧姆龙产伺服驱动器 G5 系列的直线电机型中，不支持该对象。

*1. 需要时，请将选择的过程数据映射到 PDO 后再设定。 606CHex-00.0(Velocity actual value) 为标准设定。
*2. 请设定为 [6061Hex: Modes of operation display] 时，请同时设定 [6060Hex: Modes of operation]。若只设定其中的一个，将无法正常动作。
*3. 使用欧姆龙生产的伺服驱动器 G5 系列时，请设定为 [4020Hex: Reference Position for csp]。
请参考 《AC 伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置型 用户手册 (SBCE-CN5-365)》，确认可设定的 EtherCAT 通信的过程数据通信周期。

使用注意事项

- 除了推荐的伺服驱动器外，连接了欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型的单元版本时，可能会没有支持的功能。请确认所连接伺服驱动器的手册。
- 要变更设定时，请确认 MC 功能模块的功能和过程数据的组合是否能实现预期的动作。
- 使用欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型以外的产品时，请务必设定操作模式显示 (6061Hex)。
- 使用欧姆龙产伺服驱动器 G5 系列，通过 MC_SyncMoveVelocity （周期同步速度控制）指令切换控制模式时，请映射 “csp 用参照位置 ”。
- 在欧姆龙生产的伺服驱动器 G5 系列中使用 “csp 用参照位置 ” 时，请将主周期设定为 1ms。此外，电子齿轮比请设定为 1:1。

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

● 数字输入的设定

使用 MC 功能模块的伺服驱动器输入信号。

功能名称	过程数据	说明
正转侧驱动禁止输入	60FDHex-00.1 (Digital inputs)	作为正方向极限输入使用的信号。 通常请设定为 [60FDHex-00: Digital inputs] 的［位 1: Positive limit switch］。
反转侧驱动禁止输入	60FDHex-00.0 (Digital inputs)	作为负方向极限输入使用的信号。 通常请设定为 [60FDHex-00: Digital inputs] 的［位 0: Negative limit switch］。
立即停止输入	60FDHex-00.25 (Digital inputs)	作为立即停止输入使用的信号。 使用欧姆龙生产的伺服驱动器 G5 系列时，请设定为 [60FDHex-00: Digital inputs] 的［位 25: Immediate Stop Input］。
编码器 Z 相检测	60FDHex-00.16 (Digital inputs)	显示 Z 相输入的检测状态。 在欧姆龙产伺服电机 / 驱动器 G5 系列中的气缸型且单元版本 2.1 以上的单元中，请设定为 [60FDHex-00: Digital inputs] 的［位 16: Encoder Phase Z Detection］。 在伺服驱动器 G5 系列的直线电机型中，不支持该对象。
近原点输入	60FDHex-00.2 (Digital inputs)	作为近原点输入使用的信号。 通常请设定为 [60FDHex-00: Digital inputs] 的［位 2: Home switch］。
外部锁定输入 1	60FDHex-00.17 (Digital inputs)	显示作为外部锁定输入 1 使用的信号的状态。 使用欧姆龙生产的伺服驱动器 G5 系列时，请设定为 [60FDHex-00: Digital inputs] 的［位 17: External Latch Input 1］。
外部锁定输入 2	60FDHex-00.18 (Digital inputs)	显示作为外部锁定输入 2 使用的信号的状态。 使用欧姆龙生产的伺服驱动器 G5 系列时，请设定为 [60FDHex-00: Digital inputs] 的［位 18: External Latch Input 2］。

使用注意事项

- 除了推荐的伺服驱动器外，连接了欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置型的单元版本时，可能会没有支持的功能。请确认所连接伺服驱动器的手册。
- 要变更设定时，请确认 MC 功能模块的功能和过程数据的组合是否能实现预期的动作。

版本相关信息

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

	操作模式显示	
	(6061Hex) 有	**(6061Hex) 无**
有操作模式 (6060Hex)	• 可执行使用 CSP *1、CSV *2、CST *3 的指令。 • CSP、 CSV、 CST 以外时，为伺服 OFF。	• 可执行使用 CSP 的指令。使用其他控制模式的指令将在执行时发生 “ 过程数据对象设定不足 （错误代码：3461Hex） ”。 • MC 功能模块将伺服驱动器的控制模式视为 CSP。向伺服驱动器发出 CSP 指令。
无操作模式 (6060Hex)	• 可执行使用 CSP 的指令。使用其他控制模式的指令将在执行时发生 “ 过程数据对象设定不足 （错误代码：3461Hex） ”。 • CSP 以外时，为伺服 OFF。	可执行使用 CSP 的指令。使用其他控制模式的指令将在执行时发生 “ 过程数据对象设定不足 （错误代码：3461Hex） ”。

*1. 伺服驱动器控制模式中的周期同步位置模式 (CSP)。
*2. 伺服驱动器控制模式中的周期同步速度模式 (CSV)。
*3. 伺服驱动器控制模式中的周期同步扭矩模式 (CST)。

伺服驱动器侧的对象设定

使用 MC 功能模块的控制功能时，欧姆龙产伺服驱动器 G5 系列上需要进行以下设定。
使用伺服驱动器的各功能时，请按照伺服驱动器的手册，设定与各种功能相关的对象。

索引	子索引	名称	推荐设定值	说明
3013Hex	00Hex	第 1 外部扭矩限制 *1	1388Hex	初始值 =500.0%
3015Hex *2	00Hex	使用绝对编码器时的动作切换	0002Hex	作为绝对值使用，忽略多圈计数器超限。
3317Hex	00Hex	速度限制选择	0001Hex	扭矩控制时的速度限制方法为根据 607FHex: Max profile velocity 和 3321Hex: 速度限制值设定中的较小值进行限制。
3323Hex	00Hex	外部反馈脉冲类型选择	0000Hex 或 0001Hex、0002Hex *3	设定使用的外部光栅尺的类型。初始值 0000Hex 为 “90° 相位差输出型 ”。
3324Hex *2	00Hex	外部反馈脉冲分频比分子	00000000Hex	设定电机每转 1 圈的编码器分辨率。设定为 “0” 后，将自动设定。
3401Hex	00Hex	输入信号选择 2	00818181Hex	正转侧驱动禁止输入 （b 接）
3402Hex	00Hex	输入信号选择 3	00828282Hex	反转侧驱动禁止输入 （b 接）

索引	子索引	名称	推荐设定值	说明
3403Hex	00Hex	输入信号选择 4	00222222Hex	近原点输入 （a 接）
3404Hex	00Hex	输入信号选择 5	002B2B2BHex	外部锁定信号 3 （a 接）
3405Hex	00Hex	输入信号选择 6	00212121Hex	外部锁定信号 2 （a 接）
3406Hex	00Hex	输入信号选择 7	00202020Hex	外部锁定信号 1 （a 接）
3504Hex	00Hex	驱动禁止输入选择	0001Hex	伺服侧将驱动禁止输入设为无效。在 MC 功能模块侧执行。
3508Hex	00Hex	欠电压错误选择	0001Hex	发生主电源电压不足，并停止。
3521Hex	00Hex	扭矩限制选择 *1	0006Hex	正反均有 2 个限制值。通过 PCL、 NCL 进行切换。
3522Hex	00Hex	第 2 外部扭矩限制 *1	1388Hex	初始值 =500.0%
3703Hex	00Hex	扭矩限制中输出设定 *1	0001Hex	在扭矩指令值外的扭矩限制中为 ON。
3801Hex	00Hex	软件限位功能	0003Hex	将两侧软件限位设为无效。
3758Hex	00Hex	锁定触发选择	0100Hex	设定为 Touch probe1 = 外部锁定信号 1、Touch probe2 = 外部锁定信号 2。
3759Hex	00Hex	警告保持选择	0000Hex	警告原因消除后，警告将自动清除。
607CHex	00Hex	编码器原点位置偏置	00000000Hex	伺服侧偏置值使用 “0”。
6091Hex	01Hex	电子齿轮比分子	00000001Hex	伺服驱动器侧的齿轮比为 1:1。相同的功能在 MC 功能模块侧设定。
	02Hex	电子齿轮比分母	00000001Hex	

*1. 在欧姆龙生产的伺服驱动器 G5 系列 EtherCAT 通信内置直线电机型中，请将 “ 扭矩 ” 替换为 “ 推力 ”。
*2. 在欧姆龙产伺服驱动器 G5 系列 EtherCAT 通信内置直线电机型中，不支持该对象。
*3. 用欧姆龙生产的伺服电机 / 驱动器 G5 系列进行全闭环控制时，或使用欧姆龙产伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型时，请结合使用的外部外部光栅尺类型设定。

A-2 与编码器输入终端连接

介绍与欧姆龙产 GX 系列 EtherCAT 从站 编码器输入终端的连接。

A-2-1 与编码器输入终端配线

与编码器输入终端通过 EtherCAT 通信连接。

连接方法请参考 《NJ/NX 系列 CPU 单元内置 EtherCAT 端口 用户手册 （SBCD-CN5-358）》。

A-2-2 编码器输入终端的设定

下面介绍与 MC 功能模块的控制对象 —— 欧姆龙生产的 EtherCAT 远程 I/O 终端 GX 系列 GX-EC0211/EC0241 连接时，编码器输入终端侧的设定概要。

编码器输入终端的详情请参考 《GX 系列 EtherCAT 从站 用户手册 (SBCD-CN5-350)》。

推荐编码器输入终端

可作为 MC 功能模块的编码器轴使用所有功能的编码器输入终端的单元版本如下表所示。

厂家	对象型号	对象单元版本
欧姆龙	GX-EC0211	Ver.1.1 以上
	GX-EC0241	Ver.1.1 以上

参考

- 可作为编码器轴使用的 EtherCAT 从站仅为欧姆龙生产的 GX-EC0211/ EC0241。
- 欧姆龙生产的 GX-EC0211/EC0241 （单元版本 1.0）也可作为编码器轴使用，但不是 Sysmac 设备。

  使用单元版本 1.0 时，请将节点地址开关设定为 “00” 以外。若设定为 “00”，网络配置将出现异常。

  功能详情请参考 《GX 系列 EtherCAT 从站 用户手册 (SBCD-CN5-350)》。

  Sysmac 设备请参考 《NJ/NX 系列 CPU 单元 用户手册 软件篇 （SBCA-CN5-359）》。

外部输入信号

使用作为编码器轴时的所有功能时，使用编码器输入终端上的以下输入信号。

- 计数器 A 相
- 计数器 B 相
- 计数器 Z 相
- 锁定输入 (A/B)

计数器有 2 个通道，每个通道有 2 个外部锁定。请连接实际使用的输入信号。

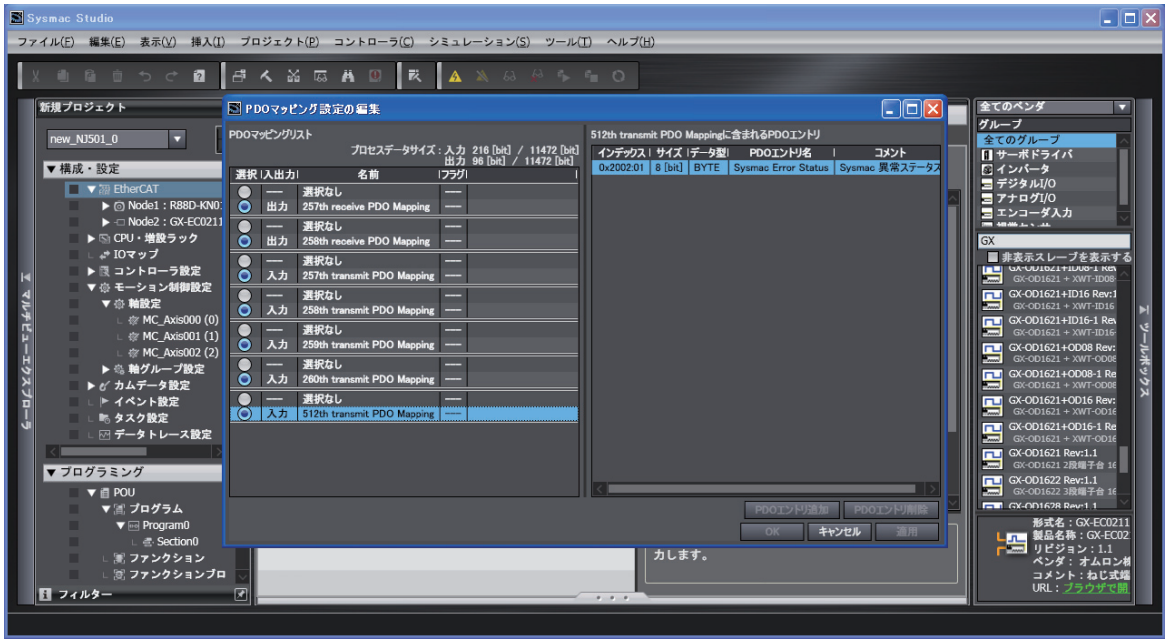
输入信号和配线方法请参考 🕮《GX 系列 EtherCAT 从站 用户手册 (SBCD-CN5-350)》。

PDO 映射

介绍作为 MC 功能模块的编码器轴控制时所需的 PDO 映射。

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

要使用的 PDO 映射在 Sysmac Studio 的 ［EtherCAT］构成编辑画面的 ［PDO 映射设定的编辑］中选择。



Sysmac Studio 默认选择以下 PDO 映射。

RxPDO(1700Hex)	通道 1 软交换 (4020Hex-01Hex)、通道 2 软交换 (4020Hex-02Hex)
RxPDO(1701Hex)	通道 1 预设当前值 (4011Hex-01Hex)、通道 2 预设当前值 (4011Hex-02Hex)
TxPDO(1B00Hex)	通道 1 计数器当前值 (4010Hex-01Hex)、通道 2 计数器当前值 (4010Hex-02Hex)
TxPDO(1B01Hex)	通道 1 锁定值 A(4012Hex-01Hex)、通道 2 锁定值 A(4012Hex-02Hex)
TxPDO(1B02Hex)	通道 1 锁定值 B(4013Hex-01Hex)、通道 2 锁定值 B(4013Hex-02Hex)
TxPDO(1B03Hex)	通道 1 状态位 (4030Hex-01Hex)、通道 2 状态位 (4030Hex-02Hex)
TxPDO(1BFFHex)	Sysmac 异常状态 (2002Hex-01Hex)

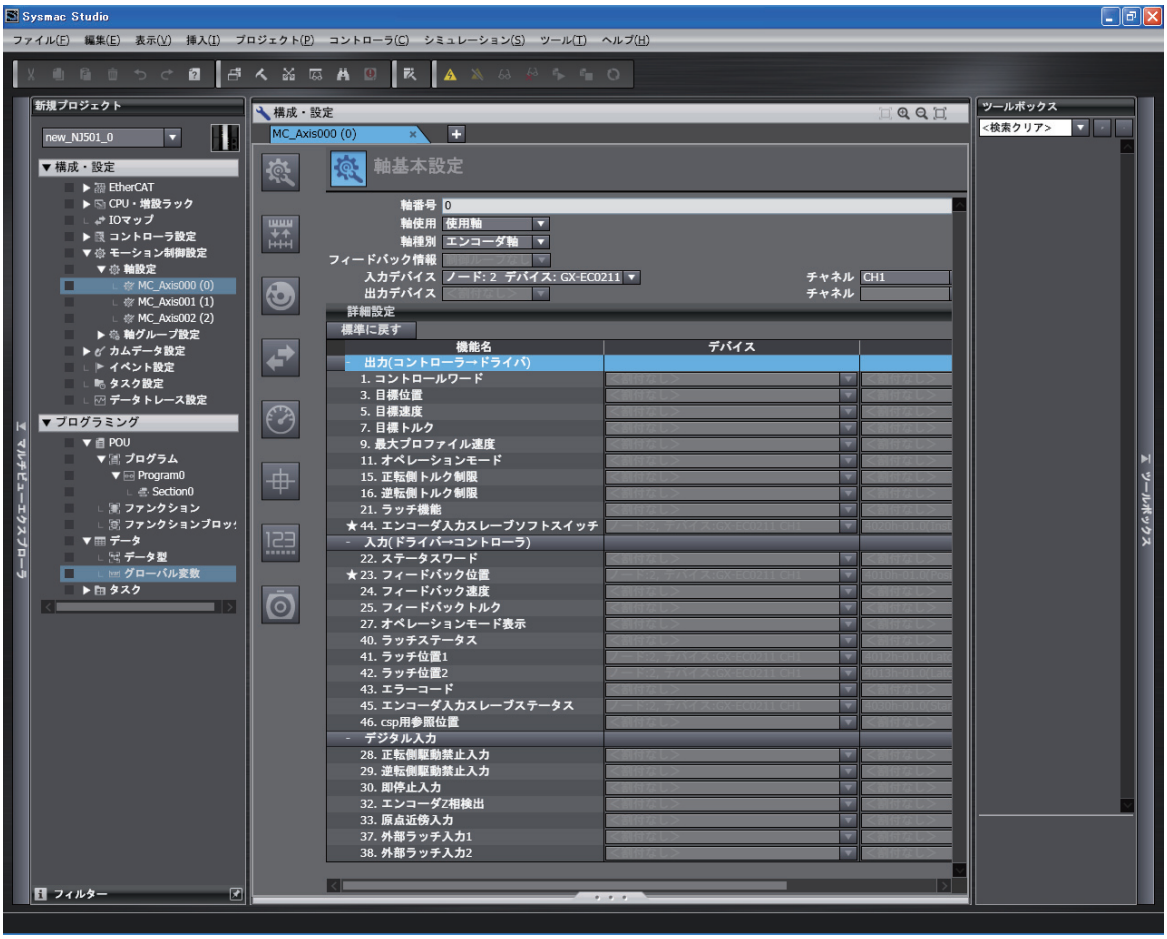
参考

使用推荐编码器输入终端 GX-EC0211/EC0241 （单元版本 1.1 以上）时， PDO 可使用 Sysmac Studio 的默认设定。

MC 功能模块的功能和过程数据的关系

MC 功能模块拥有的功能与过程数据的对象信息有关。
因使用 MC 功能模块的 EtherCAT 从站的构成或使用的功能不同，两者的关系可能需要变更。

在 Sysmac Studio 中单击［轴基本设定］画面的［详细设定］，将打开设定画面。



参考

使用推荐编码器输入终端 GX-EC0211/EC0241 （单元版本 1.1 以上）时，功能和过程数据的关系可使用 Sysmac Studio 的默认设定。

● 输出 （控制器→驱动器）的设定

设定 MC 功能模块向编码器输入终端发出的指令数据。

以下为 Sysmac Studio 的默认设定。（O 标记表示必需对象）

功能名称		过程数据		说明
		通道 1	通道 2	
O	编码器输入从站软交换	4020Hex-01.0 (Instruction Bits)	4020Hex-02.0 (Instruction Bits)	设定软交换的内容。 请根据使用的通道，设定左侧所述的对象。

使用注意事项

要变更设定时，请确认 MC 功能模块的功能和过程数据的组合是否能实现预期的动作。

● 输入 （驱动器→控制器）的设定

编码器输入终端到 MC 功能模块的状态数据。

以下为 Sysmac Studio 的默认设定。（O 标记表示必需对象）

功能名称		过程数据		说明
		通道 1	通道 2	
O	反馈位置	4010Hex-01.0 (Position Value)	4010Hex-02.0 (Position Value)	保存计数器的当前值。 请根据使用的通道，设定左侧所述的对象。
	锁定位置 1	4012Hex-01.0 (Latch Value A)	4012Hex-02.0 (Latch Value A)	通过锁定功能 1 锁定的位置。保存锁定 A 的值。 MC_TouchProbe （外部锁定有效）指令等，使用锁定功能时需要。 请根据使用的通道，设定左侧所述的对象。
	锁定位置 2	4013Hex-01.0 (Latch Value B)	4013Hex-02.0 (Latch Value B)	通过锁定功能 2 锁定的位置。保存锁定 B 的值。 MC_TouchProbe （外部锁定有效）指令等，使用锁定功能时需要。 请根据使用的通道，设定左侧所述的对象。
	编码器输入从站状态	4030Hex-01.0 (Status Bits)	4030Hex-02.0 (Status Bits)	保存状态位。 MC_TouchProbe （外部锁定有效）指令等，使用锁定功能时需要。 请根据使用的通道，设定左侧所述的对象。

使用注意事项

要变更设定时，请确认 MC 功能模块的功能和过程数据的组合是否能实现预期的动作。

- **数字输入的设定**

  作为编码器轴使用时，无需设定。

编码器输入终端侧的对象设定

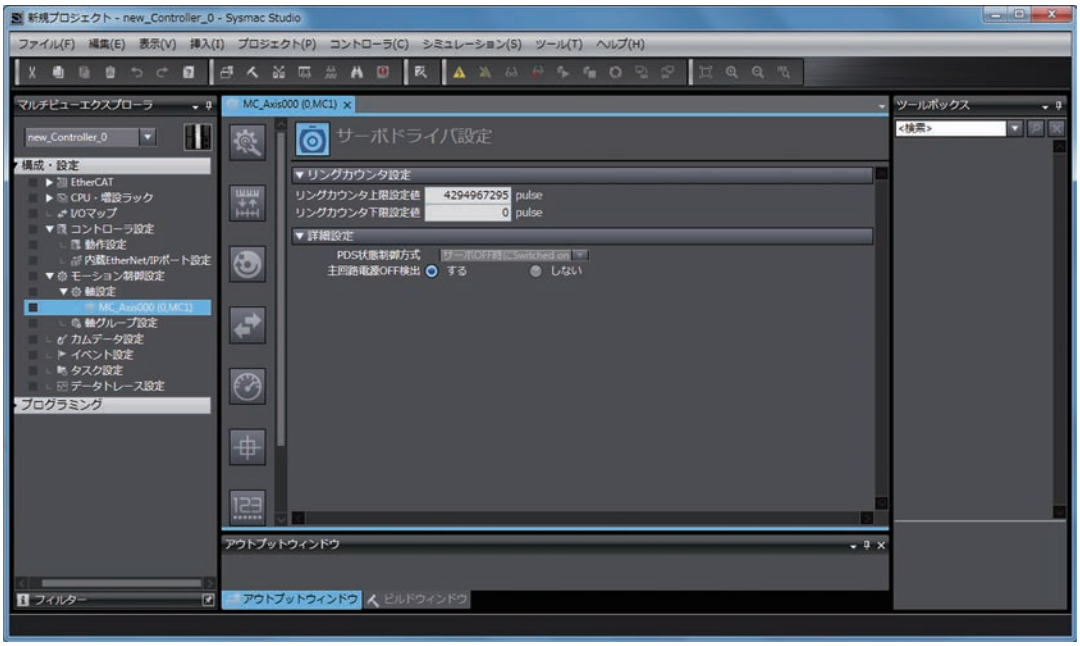
编码器输入终端上没有需要设定的对象。

MC 功能模块和编码器输入终端的循环计数器的关系

MC 功能模块的轴参数的伺服驱动器设定 （循环计数器上限值、循环计数器下限值）和编码器输入终端的循环计数器最大值设定需要统一。

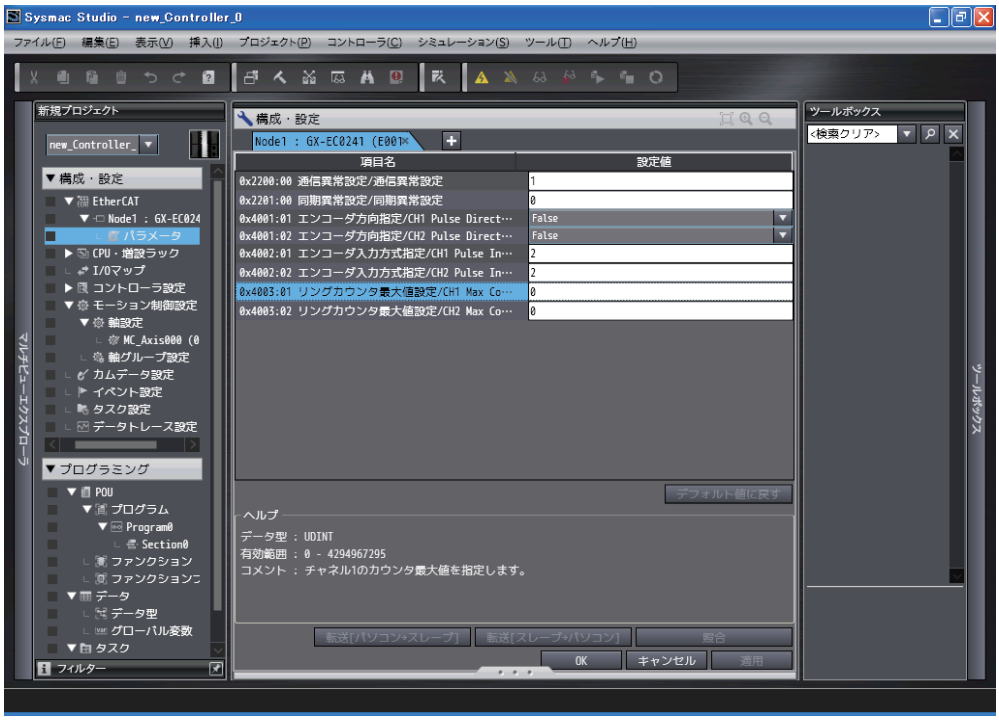
伺服驱动器设定 （循环计数器上限值、循环计数器下限值）在 Sysmac Studio 的 ［伺服驱动器设定］画面中编辑。
设定内容如下所示。

参数	含义	设定值
循环计数器上限设定值	设定伺服驱动器或编码器输入终端中设定的循环计数器上限值。	请与编码器输入终端的循环计数器最大值设定一致。
循环计数器下限设定值	设定伺服驱动器或编码器输入终端中设定的循环计数器下限值。	请设定为 “0”。



编码器输入终端的循环计数器最大值设定在 Sysmac Studio 的 [EtherCAT] 编辑画面中设定。
设定内容如下所示。

索引	对象名称	设定值
0x4003	连接计数器最大值设定 (Max Count Setting)	请与 MC 功能模块的轴参数的伺服驱动器设定 （循环计数器上限设定值）统一。



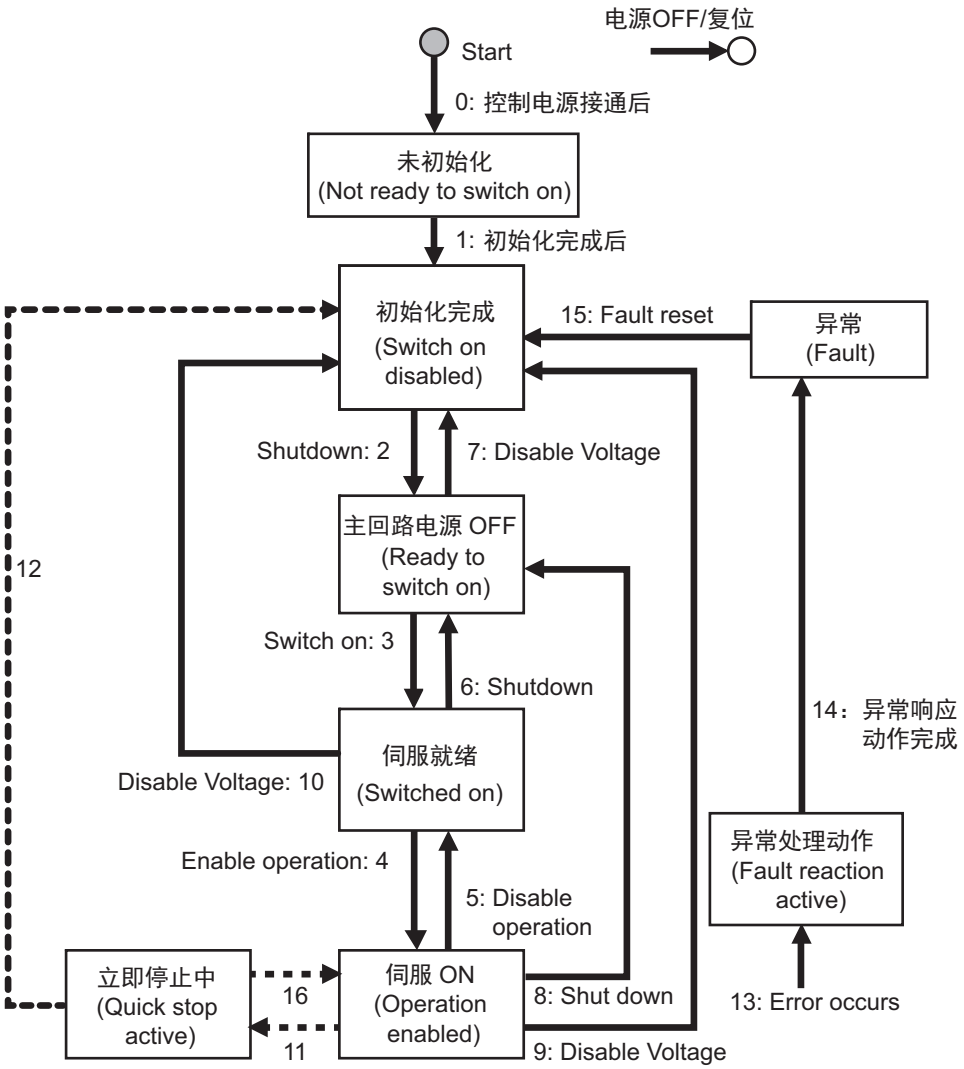
A-3 与 NX 单元连接

关于与 NX 系列位置接口单元的连接，请参考 🕮《NX 系列 位置接口单元 用户手册 (SBCE-CN5-374)》。

A-4 PDS 状态变化

PDS 状态变化在 CiA402 驱动器配置文件中定义。 PDS 状态变化的指令使用过程数据的控制字 (6040Hex: Controlword)。实际的 PDS 状态可通过过程数据的状态字 (6041Hex: Statusword) 确认。

以下表示 CiA402 驱动器配置文件中定义的状态变化图。
长方形表示状态，数字表示状态控制指令。



A-4-1 PDS 状态控制方式

下面介绍轴参数 ［PDS 状态控制方式］的设定值和 PDS 状态的关系。

●［PDS 状态控制方式］设定为 “0” 时

若将 ［PDS 状态控制方式］设定为 “0: 伺服 OFF 时 Switeched on”，则动作如下。

- 与伺服驱动器建立通信后， MC 功能模块将自动把 PDS 状态变为伺服就绪 (Switched on) 状态。
- 若在伺服就绪 (Switched on) 状态下将 MC_Power （可运行）指令的 “Enable （有效） ” 设为 TRUE，将变为伺服 ON(Operation enabled)。变为状态变化图的 “4: Enable operation”。
- 若在伺服 ON(Operation enabled) 状态下将 MC_Power （可运行）指令的 “Enable （有效） ” 设为 FALSE，将变为伺服就绪 (Switched on) 状态。变为状态变化图的 “5: Disable operation”。

●［PDS 状态控制方式］设定为 “1” 时

若将 ［PDS 状态控制方式］设定为 “1: 伺服 OFF 时 Ready to switch on”，则动作如下。

- 与伺服驱动器建立通信后，MC 功能模块将自动把 PDS 状态变为主回路电源 OFF(Ready to switch on) 状态。
- 若在主回路电源 OFF(Ready to switch on) 状态下将 MC_Power （可运行）指令的 “Enable （有效） ” 设为 TRUE，将变为伺服 ON(Operation enabled)。变为状态变化图的 “3: Switch on” 和 “4: Enable operation”。
- 若在伺服 ON(Operation enabled) 状态下将 MC_Power （可运行）指令的 “Enable （有效） ” 设为 FALSE，将变为主回路电源 OFFOFF(Ready to switch on) 状态。变为状态变化图的 “5: Disable operation” 和 “6: Shutdown”。

版本相关信息

- 单元版本 1.09 以下的 CPU 单元中，［PDS 状态控制方式］为 “0”。单元版本 1.10 以上的 CPU 单元中，可选择 ［PDS 状态控制方式］。
- 单元版本 1.10 以上的 CPU 单元中，［PDS 状态控制方式］的初始值为 “0”。

A-4-2 主回路电源 OFF 检测

可选择在伺服锁定状态下是否检测主回路电源 OFF。

若执行 MC_Power （可运行）指令后仍出现 “ 检测到伺服主回路电源 OFF” 异常，请设定为 “ 不检测 ”。

使用注意事项

使用欧姆龙生产的伺服驱动器 G5 系列时，无法选择 “ 不检测 ”。

若选择 “ 不检测 ” 并在伺服锁定状态下关闭伺服主回路电源，将出现 “ 检测到伺服主回路电源 OFF” 异常。

版本相关信息

- 单元版本 1.09 以下的 CPU 单元中，［主回路电源 OFF 检测］为 “ 检测 ”。单元版本 1.10 以上的 CPU 单元中，可选择 ［主回路电源 OFF 检测］。
- 单元版本 1.10 以上的 CPU 单元中，［主回路电源 OFF 检测］的初始值为 “ 检测 ”。

A-5 术语说明

介绍运动控制相关的术语。

A-5-1 NJ/NX 系列通用

术语	说明
主存储器 (main memory)	CPU 单元在使用 OS、执行用户程序时使用的 CPU 单元内部的存储器
固定周期任务 (periodic task)	按一定周期执行 I/O 刷新和用户程序的任务
主固定周期任务 (primary periodic task)	优先度最高的任务
任务周期 (task period)	主固定周期任务及固定周期任务的执行间隔
I/O 刷新 (I/O Refresh)	事先确定的存储器区域和外部进行周期性数据交换。
程序 (program)	与函数、功能块并列，三种 POU 之一。 分配到任务后动作。
用户程序 (user program)	多个程序的集合体
联机 ST (Inline ST)	将 ST 程序作为梯形图程序的一部分写入其中，或者是所写入的 ST 程序。
系统定义变量 (system-defined variables)	所有变量属性由系统定义，用户绝对无法变更的变量。
全局变量 (global variable)	可从所有 POU （程序、功能块、函数）读写的变量。
本地变量 (local variable)	只能从定义的 POU 内部访问的变量。 本地变量中包括内部变量、输入变量、输出变量、输入输出变量、外部变量。
下载 (download)	利用 Sysmac Studio 的 “ 同步 ” 功能，从 Sysmac Studio 向控制器传送数据。
上传 (upload)	利用 Sysmac Studio 的 “ 同步 ” 功能，从控制器向 Sysmac Studio 传送数据。
全部停止故障等级的控制器异常 (major fault level Controller error)	NJ/NX 系列控制器整体无法控制的异常。 CPU 单元将立即停止执行用户程序，断开包括远程 I/O 在内所有从站 / 单元的负载。
部分停止故障等级的控制器异常 (partial fault level Controller error)	NJ/NX 系列控制器的某些功能模块整体无法控制的异常。 发生部分停止故障后， NJ/NX 系列 CPU 单元将继续运行。
轻度故障等级的控制器异常 (minor fault level Controller error)	NJ/NX 系列控制器的某些功能模块的部分功能无法控制的异常。 发生轻度故障后， NJ/NX 系列 CPU 单元将继续运行。
监视信息 (observation)	表示控制器信息及用户信息的事件重要程度的等级之一。 对控制没有影响的轻微异常，但仍记录到事件日志中，并向用户通知的信息

A-5-2 运动控制功能

术语	说明
使用实轴	轴种类为［伺服轴］或［编码器轴］，且轴使用设定为［使用轴］的轴
使用虚拟轴	轴种类为［虚拟伺服轴］或［虚拟编码器轴］，且轴使用设定为［使用轴］的轴
运动控制功能模块 (Motion Control Function Module)	执行运动控制的软件构成之一。 按照用户程序中的运动控制指令所赋予的指令，执行运动控制。（简称：MC 功能模块）
运动控制指令 (motion control instruction)	执行运动控制功能所需的运动控制定义指令。 运动控制功能模块中，包括以 PLCopen® 的运动控制用功能块为标准的指令和运动控制功能模块独立的指令。
单轴位置控制 (single-axis position control)	用 1 个轴进行定位控制
单轴速度控制 (single-axis velocity control)	用 1 个轴进行速度控制 MC 功能模块中，包括向伺服驱动器分配速度指令值和位置指令值。
单轴转矩控制 (single-axis torque control)	用 1 个轴进行转矩控制
单轴同步控制 (single-axis synchronized control)	1 个主轴与 1 个从轴对应的同步控制。 包括以齿轮比同步的齿轮动作，以及与凸轮表的相位和变位关系同步的凸轮动作。
单轴手动控制 (single-axis manual operation)	点动进给等手动操作所需的控制
单轴控制辅助 (auxiliary functions for single-axis control)	超驰值设定或异常的重置等，对 1 个轴的控制进行辅助的功能
多轴协调控制 (multi-axes coordinated control)	直线插补或圆弧插补等与多个轴关联动作的控制。 指定通过轴组调节的轴。
多轴协调控制辅助 (auxiliary functions for multi-axes coordinated control)	对轴组的超驰值设定或异常的重置等，对多个轴的控制进行辅助的功能
运动控制参数 (motion control parameters)	用于指定 MC 功能模块各种动作的参数。 MC 通用参数、轴参数、轴组参数的总称。
轴参数 (axis parameters)	以轴为单位的参数
轴组参数 (axes group parameters)	以轴组为单位的参数
运动控制系统变量 (system-defined variables for motion control)	表示 MC 功能模块各种状态信息的变量。 MC 共通变量、轴变量、轴组变量的总称。
MC 共通变量 (MC common variable)	结构体中定义的系统定义变量，如 MC 功能模块通用的各种状态信息等
轴变量 (axis variable)	结构体中定义的系统定义变量，如各轴的各种状态信息、部分轴参数设定信息等
轴组变量 (axes group variable)	结构体中定义的系统定义变量，如各轴组的各种状态信息、部分轴组参数设定信息等
原点复位 (homing)	确定机械原点。 也称为定原点、原点检索、校准、 Datum。
机械原点 (home)	各装置中规定的机械原点。 原点复位动作时，由原点输入信号确定的原点。
原点 (zero position)	根据机械原点，在用户程序中视为零点位置的原点。 若不变更原点位置，则与机械原点的位置相同。

术语	说明
位置偏差 (following error)	指令当前位置和反馈当前位置的差。 位置控制模式以外的模式下没有指令当前位置，因此不存在位置偏差。 也称偏差计数器值、滞留脉冲等。
偏差计数器复位 (following error reset)	将位置偏差设为零
凸轮轮廓曲线 (cam profile curve)	表示实现凸轮动作所需的相位和变位关系的曲线。 Sysmac Studio 的凸轮编辑器中创建的曲线。
凸轮数据 (cam data)	表示实现凸轮动作所需的、由相位 （主轴）和变位 （从轴）构成的数据
凸轮数据变量 (cam data variable)	将凸轮数据转换为结构体排列后的变量。 结构体型排列指定的变量，由相位和变位数据组成。
凸轮表 (cam table)	凸轮数据的数据表。 若将凸轮编辑器中创建的凸轮轮廓曲线传送到 CPU 单元，将作为凸轮表保存到 CPU 单元内置的非易失性存储器中。
超驰 (override)	可在作业过程中临时变更由作业者编写的程序值
跃度 (jerk)	加速度 / 减速度的变化比率。 根据跃度的指定，加减速时的速度波形变为 S 字型。 也称为跃动、加加速度。
多运动 (Multi-motion)	使用主固定周期任务和固定周期任务 （执行优先度 5）两个任务，并列执行运动控制

A-5-3 EtherCAT 通信

术语	说明
CAN application protocol over EtherCAT (CoE)	EtherCAT 上的 CAN 应用 协议 服务
CAN in Automation (CiA)	提供上层协议的开发和支援的用户及各厂家的国际团体
EtherCAT 协会 (ETG) (EtherCAT Technology Group)	开放 EtherCAT，促进其普及的团体
对象 (Object)	设备内特定结构体的抽象表现，数据、参数、方法的集合体
对象字典 (Object Dictionary)	包含数据类型对象、通信对象、应用对象记述的数据构造
过程数据 (Process Data)	以测量及控制为目的，按周期或无周期传送的应用对象的集合体
过程数据对象 (Process Data Object)	通过映射包含 1 个以上过程数据实体的参数来记述的构造
服务数据对象 (Service Data Object)	通过 CoE 的异步邮箱通信，可读写所有的对象字典
接收 PDO (Receive PDO)	EtherCAT 从站接收的过程数据对象
发送 PDO (Transmit PDO)	EtherCAT 从站发送的过程数据对象
设备轮廓 (Device Profile)	设备类应用程序参数或功能的记述集合体

A-6 版本相关信息

介绍各单元版本的支持功能。

CPU 单元的单元版本种类和 Sysmac Studio 版本的关系请参考 🕮《NJ/NX 系列 CPU 单元 用户手册 软件篇 (SBCA-CN5-359)》。

运动控制指令的版本相关信息请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。

单元版本 1.01 中追加的运动控制功能

使用单元版本 1.01 中追加的功能时，请使用 Sysmac Studio Ver.1.02 及以上版本。

功能	概要
轴组构成轴写入	利用 MC_ChangeAxesInGroup （轴组构成轴写入）指令，暂时改写轴组参数的 " 构成轴 " 的功能。
轴组位置获取	利用 MC_GroupReadPosition （轴组位置获取）指令，获取轴组指令当前位置和反馈当前位置的功能。
轴组周期性同步绝对位置控制	利用 MC_GroupSyncMoveAboslute （轴组周期同步绝对位置控制）指令，周期性输出各轴中指定的目标位置的功能。
控制对象伺服驱动器	追加欧姆龙生产的伺服电机 / 驱动器 G5 系列 EtherCAT 通信内置直线电机型。

单元版本 1.02 中追加的运动控制功能

使用单元版本 1.02 中追加的功能时，请使用 Sysmac Studio Ver.1.03 及以上版本。

单元版本 1.02 中无追加的运动控制功能，但部分指令的规格发生改进。运动控制指令的版本相关信息请参考 🕮《NJ/NX 系列 指令基准手册 运动篇 (SBCE-CN5-364)》。

单元版本 1.03 中追加的运动控制功能

使用单元版本 1.03 中追加的功能时，请使用 Sysmac Studio Ver.1.04 及以上版本。

功能	概要
周期性同步绝对定位	利用 MC_SyncMoveAbsolute （周期性同步绝对定位）指令，在位置控制模式下，按控制周期输出指令位置的功能。
参数指定原点复位	利用 MC_HomeWithParameter （参数指定原点复位）指令，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的功能。

单元版本 1.04 中追加的运动控制功能

使用单元版本 1.04 中追加的功能时，请使用 Sysmac Studio Ver.1.05 及以上版本。

功能	概要
轴使用变更	利用 MC_ChangeAxisUse （轴使用变更）指令，临时切换轴参数 ［轴使用］的功能。

单元版本 1.05 中追加的运动控制功能

使用单元版本 1.05 中追加的功能时，请使用 Sysmac Studio Ver.1.06 及以上版本。

功能	概要
启动速度	设定轴动作开始时初速度的功能。
输入信号的逻辑反转	使立即停止输入信号、正方向极限输入信号、负方向极限输入信号、近原点输入信号的逻辑反转的功能。

（注）单元版本 1.05 以上的 CPU 单元中，可将 NX 系列位置接口单元分配到轴中，进行运动控制。

单元版本 1.06 中追加、变更的运动控制规格

使用单元版本 1.06 中追加、变更的性能规格、功能规格时，请使用 Sysmac Studio Ver.1.07 及以上版本。

功能	概要
最大控制轴数	NJ301-□□□□ 的最大控制轴数扩展到 15 轴。 （最大使用实轴数无变更。）
最大单轴控制数	NJ301-□□□□ 最大单轴控制数扩展到 15 轴。 （最大使用实轴数无变更。）
数字凸轮开关有效	利用 MC_DigitalCamSwitch （数字凸轮开关有效）指令，根据轴的位置将数字输出设为 ON 或 OFF 的功能。
时间戳→轴位置计算	利用 MC_TimeStampToPos （时间戳→轴位置计算）指令，计算指定时间戳下轴位置的功能。
在凸轮动作开始中追加共混	在 MC_CamIn （凸轮动作开始）指令的缓存模式选择中追加共混。
_sMC_POSITION_REF 型	对任意坐标系的轨迹进行 3D 运动监视显示的数据类型。

单元版本 1.08 中追加、变更的运动控制功能

使用单元版本 1.08 中追加、变更的功能时，请使用 Sysmac Studio Ver.1.09 及以上版本。

功能	概要
凸轮表生成	利用 MC_GenerateCamTable （凸轮表生成）指令，根据输入参数指定的凸轮属性和凸轮节点生成凸轮表的功能。
轴参数的变更	利用 MC_WriteAxisParameter （轴参数写入）指令及 MC_ReadAxisParameter （轴参数读取）指令，通过用户程序参照和变更轴参数的功能。
设备变量的分配	可向轴变量中分配的从站和单元的 I/O 端口分配设备变量。*1 详情请参考 *「2-4-2 EtherCAT 主机功能和运动控制功能模块的关系* (P.2-18)」。

*1. 与 CPU 单元的单元版本无关。使用 Sysmac Studio Ver.1.09 以上时可分配。

单元版本 1.09 中追加、变更的运动控制功能

使用单元版本 1.09 中追加的功能时，请使用 Sysmac Studio Ver.1.10 及以上版本。

单元版本 1.09 中无追加的运动控制功能，但部分指令的规格发生改进。运动控制指令的版本相关信息请参考 《NJ/NX 系列 指令基准手册 (SBCE-CN5-364)》。

单元版本 1.10 中追加、变更的运动控制功能

使用单元版本 1.10 中追加的功能时，请使用 Sysmac Studio Ver.1.12 及以上版本。

功能	概要
从轴位置补偿	对同步动作中的从轴进行位置补偿的功能。
共混的动作变更	“加减速超限”设定为［剧烈加减速 （将共混切换为缓冲）］、［异常停止］时，以最大加减速度进行加减速，继续进行共混。
绝对编码器的原点确定时间	除了传统的原点确定之外，EtherCAT 从站的过程数据通信从未建立状态变为建立状态时，也可确定原点。
过程数据通信未建立时的当前位置	轴变量的反馈当前位置及指令当前位置为过程数据通讯未建立前的反馈当前位置。
操作模式 (6060Hex) 及操作模式显示 (6061Hex) 的映射有无	根据操作模式 (6060Hex) 及操作模式显示 (6061Hex) 的映射有无组合不同，动作也不同。
PDS 状态控制方式的追加	利用 MC_Power 指令，设定伺服 OFF 时 PDS 状态的变化目标。
主回路电源 OFF 检测的追加	可选择在伺服锁定指令时或伺服锁定状态下是否检测主回路电源 OFF。

使用单元版本 1.10 中追加的功能时，请使用 Sysmac Studio Ver.1.13 及以上版本。

功能	概要
多运动的同步控制	在多运动中，执行不同任务中分配的轴间同步控制指令的功能。

单元版本 1.11 中追加、变更的运动控制功能

使用单元版本 1.11 中追加的功能时，请使用 Sysmac Studio Ver.1.15 及以上版本。

功能	概要
追加单位转换设定的参数	追加 “ 减速机使用 ”、“ 工件侧每转 1 圈的移动量 ”、“ 工件侧齿轮比 ”、“ 电机侧齿轮比 ” 的参数。

I

索引

索引

B

编码器种类 5-22, 5-24
编码器轴 3-2, 5-7, 5-8
变位 9-15
部分停止故障 11-8

C

插补加减速超限 5-32
插补加速度警告值 5-32
插补减速度警告值 5-32
插补速度警告值 5-32
超驰 9-58
程序变更凸轮数据 9-15
从轴 9-15
错误减速停止中 6-21, 6-27
重叠 9-67, 9-69
重启 9-40, 9-62

D

单位转换 6-23
当前方向指定 9-37
当前位置上限值 6-24
当前位置下限值 6-24
点动最高速度 5-17, 9-33
电机每转 1 圈的脉冲数 5-10, 6-23
电机每转 1 圈的移动量 5-10, 6-23
电子齿轮比 5-12
定位动作中 6-20
定原点 8-2
动作中 6-27
对象字典 2-17
多重启动 9-45, 9-63
多轴协调动作中 6-21

E

EtherCAT 1-2
EtherCAT 主机功能模块 2-2

F

发生位置 11-8
发生原因 11-12
反馈当前扭矩 6-23
反馈当前速度 6-23
反馈当前位置 6-23
反馈速度 9-33
反馈速度过滤器时间常数 5-18, 9-33
反馈位置 6-14, 9-31
反转时动作 5-17
负方向极限输入 6-21
负方向极限输入逻辑反转 5-20
负方向极限输入时动作 5-25, 8-9
负方向扭矩警告值 5-17
负方向扭矩限制上限值 5-20
负方向软件限位 5-21
负方向指定 9-37
负方向指令中 6-21
服务数据对象 2-17

G

GetMCError (获取运动控制异常状态) 11-6
共混 （混合） 9-47, 9-65
构成轴 5-30, 5-31
构成轴 （轴 A0） 6-29
构成轴 （轴 A1） 6-29
构成轴 （轴 A2） 6-29
构成轴 （轴 A3） 6-29
固定周期任务 2-4, 2-5
故障排除 11-5
故障排除功能 11-5
过程数据对象 2-17
过程数据通信周期 2-20

H

缓冲 （待机） 9-46, 9-65
缓冲模式 9-45, 9-63

J

基本数据型 6-17
计数模式 5-22, 6-24
极限输入停止方法 5-20
机械原点 8-2
加减速超限 5-17, 9-34
加速度的变更 9-42
加速度的单位 9-34
加速度警告值 5-17, 9-34
监视信息 11-8
减速度的变更 9-42
减速度的单位 9-34
减速度警告值 5-17, 9-34
减速停止中 6-20, 6-27
节点地址 6-23
结构体型 6-17
近原点输入 6-21
近原点输入逻辑反转 5-21
绝对编码器
- 对象伺服电机 8-16
- 设置 8-17
- 原点位置偏置 8-15
- 在旋转模式下使用 8-17

K

开始位置方式指定 9-15
空凸轮数据 9-15

L

LED 11-4
立即停止输入 6-21
立即停止输入逻辑反转 5-20
立即停止输入停止方法 5-20
连接加速度 9-15
连接速度 9-15
连续动作中 6-20
列举型 6-17

M

_MC_AX[*]（轴变量） 6-20
_MC_AX[*].Act.Pos（反馈当前位置） 6-23
_MC_AX[*].Act.Timestamp（时间戳） 6-23
_MC_AX[*].Act.Trq（反馈当前扭矩） 6-23
_MC_AX[*].Act.Vel（反馈当前速度） 6-23
_MC_AX[*].Cfg.AxEnable（轴使用） 6-23
_MC_AX[*].Cfg.AxNo（轴编号） 6-23
_MC_AX[*].Cfg.AxType（轴种类） 6-23
_MC_AX[*].Cfg.ExecID（执行 ID） 6-23
_MC_AX[*].Cfg.NodeAddress（节点地址） 6-23
_MC_AX[*].Cmd.AccDec（指令当前加减速度） 6-22
_MC_AX[*].Cmd.Jerk（指令当前跃度） 6-22
_MC_AX[*].Cmd.Pos（指令当前位置） 6-22
_MC_AX[*].Cmd.Trq（指令当前扭矩） 6-22
_MC_AX[*].Cmd.Vel（指令当前速度） 6-22
_MC_AX[*].Details.Homed（原点确定） 6-21
_MC_AX[*].Details.Idle（停止中） 6-21
_MC_AX[*].Details.InHome（原点停止） 6-21
_MC_AX[*].Details.InPosWaiting（位置等待） 6-21
_MC_AX[*].Details.VelLimit（指令速度饱和） 6-21
_MC_AX[*].Dir.Nega（负方向指令中） 6-21
_MC_AX[*].Dir.Posi（正方向指令中） 6-21
_MC_AX[*].DrvStatus.CSP
（周期性同步位置 (CSP) 模式中） 6-22
_MC_AX[*].DrvStatus.CST
（周期性同步扭矩 (CST) 模式中） 6-22
_MC_AX[*].DrvStatus.CSV
（周期性同步速度 (CSV) 模式中） 6-22
_MC_AX[*].DrvStatus.DrvAlarm（驱动器错误输入） 6-22
_MC_AX[*].DrvStatus.DrvWarning（驱动器警告输入） .6-22
_MC_AX[*].DrvStatus.Home（原点输入） 6-21
_MC_AX[*].DrvStatus.HomeSw（近原点输入） 6-21
_MC_AX[*].DrvStatus.ILA（驱动器内部功能限制中） 6-22
_MC_AX[*].DrvStatus.ImdStop（立即停止输入） 6-21
_MC_AX[*].DrvStatus.Latch1（锁定输入 1） 6-22
_MC_AX[*].DrvStatus.Latch2（锁定输入 2） 6-22
_MC_AX[*].DrvStatus.MainPower（主回路电源） 6-21
_MC_AX[*].DrvStatus.N_OT（负方向极限输入） 6-21
_MC_AX[*].DrvStatus.P_OT（正方向极限输入） 6-21
_MC_AX[*].DrvStatus.Ready（伺服就绪） 6-21
_MC_AX[*].DrvStatus.ServoOn（伺服 ON） 6-21
_MC_AX[*].MFaultLvl.Active（轴 发生轻度故障） 6-23
_MC_AX[*].MFaultLvl.Code（轴 轻度故障代码） 6-23
_MC_AX[*].Obsr.Active（轴 监视信息发生中） 6-23
_MC_AX[*].Obsr.Code（轴 监视信息代码） 6-23
_MC_AX[*].Scale.CountMode（计数模式） 6-24
_MC_AX[*].Scale.Den（电机每转 1 圈的移动量） 6-23
_MC_AX[*].Scale.MaxPos（当前位置上限值） 6-24
_MC_AX[*].Scale.MinPos（当前位置下限值） 6-24
_MC_AX[*].Scale.Num（电机每转 1 圈的脉冲数） 6-23
_MC_AX[*].Scale.Units（显示单位） 6-24
_MC_AX[*].Status.Continuous（连续动作中） 6-20
_MC_AX[*].Status.Coordinated（多轴协调动作中） ...6-21
_MC_AX[*].Status.Disabled（轴无效） 6-20
_MC_AX[*].Status.Discrete（定位动作中） 6-20
_MC_AX[*].Status.ErrorStop（错误减速停止中） 6-21
_MC_AX[*].Status.Homing（原点复位中） 6-20
_MC_AX[*].Status.Ready（轴启动准备完成） 6-20
_MC_AX[*].Status.Standstill（停止中） 6-20
_MC_AX[*].Status.Stopping（减速停止中） 6-20
_MC_AX[*].Status.Synchronized（同步动作中） 6-20
_MC_AX_ErrSta（轴异常状态） 11-6
_MC_COM（MC 共通变量） 6-19
_MC_COM.MFaultLvl.Active
（MC 共通 发生轻度故障） 6-19
_MC_COM.MFaultLvl.Code
（MC 共通 轻度故障代码） 6-19
_MC_COM.Obsr.Active（MC 共通 监视信息发生中） ...6-19
_MC_COM.Obsr.Code（MC 共通 监视信息代码） 6-19
_MC_COM.PFaultLvl.Active
（MC 共通 发生部分停止故障） 6-19
_MC_COM.PFaultLvl.Code
（MC 共通 部分停止故障代码） 6-19
_MC_COM.Status.CamTableBusy
（凸轮表文件保存执行中 / 待机中） 6-19
_MC_COM.Status.GenerateCamBusy
（凸轮表生成执行中） 6-19
_MC_COM.Status.RunMode（MC 运行中） 6-19
_MC_COM.Status.TestMode（MC 调试中） 6-19
_MC_ComErrSta（MC 共通异常状态） 11-6
MC 参数设定 5-2
MC 调试功能 4-2
MC 调试中 6-19
_MC_ErrSta（MC 异常状态） 11-6
_MC_GRP[*]（轴组变量） 6-27
_MC_GRP[*].Cfg.ExecID（执行 ID） 6-28
_MC_GRP[*].Cfg.GrpEnable（轴组使用） 6-28
_MC_GRP[*].Cfg.GrpNo（轴组编号） 6-28
_MC_GRP[*].Cmd.AccDec（指令插补加减速度） 6-28
_MC_GRP[*].Cmd.Vel（指令插补速度） 6-28
_MC_GRP[*].Details.Idle（停止中） 6-28
_MC_GRP[*].Details.InPosWaiting（位置等待） 6-28
_MC_GRP[*].Kinematics.Axis[0]（构成轴 （轴 A0）） ..6-29
_MC_GRP[*].Kinematics.Axis[1]（构成轴 （轴 A1）） ...6-29
_MC_GRP[*].Kinematics.Axis[2]（构成轴 （轴 A2）） ...6-29
_MC_GRP[*].Kinematics.Axis[3]（构成轴 （轴 A3）） ...6-29
_MC_GRP[*].Kinematics.GrpType（轴构成） 6-29
_MC_GRP[*].MFaultLvl.Active（轴组 发生轻度故障） ...6-28
_MC_GRP[*].MFaultLvl.Code（轴组 轻度故障代码） 6-28
_MC_GRP[*].Obsr.Active（轴组 监视信息发生中） 6-28

_MC_GRP[*].Obsr.Code （轴组 监视信息代码） 6-28
_MC_GRP[*].Status.Disabled （轴组无效） 6-27
_MC_GRP[*].Status.ErrorStop （错误减速停止中） 6-27
_MC_GRP[*].Status.Moving （动作中） 6-27
_MC_GRP[*].Status.Ready （启动准备完成） 6-27
_MC_GRP[*].Status.Standby （停止中） 6-27
_MC_GRP[*].Status.Stopping （减速停止中） 6-27
_MC_GRP_ErrSta （轴组异常状态） 11-6
MC 共通变量 6-19
MC 共通 部分停止故障 6-19
MC 共通 部分停止故障代码 6-19
MC 共通 部分停止故障发生中 6-19
MC 共通 发生轻度故障 6-19
MC 共通 监视信息 6-19
MC 共通 监视信息代码 6-19
MC 共通 监视信息发生中 6-19
MC 共通 轻度故障 6-19
MC 共通 轻度故障代码 6-19
MC 共通异常 11-8
MC 共通状态 6-19
MC 运行中 6-19
目标速度的变更 9-42
目标位置的变更 9-40
减速超限模式 9-41
三角控制模式 9-41
因新的指令值而不反转时 9-40
因新的指令值而反转时 9-40

N

扭矩指令的变更 9-43

P

PLC 功能模块 2-2
PLCopen® 1-2
Process Data Objects (PDO) 2-17
派生数据型 6-17

Q

启动速度 .....5-17，5-19，9-33
启动准备完成 6-27
轻度故障 11-8
驱动器错误输入 6-22
驱动器错误重置监视时间 5-20
驱动器警告输入 6-22
驱动器内部功能限制中 6-22
全部停止故障 11-8

R

任务 2-4
任务的周期 2-9
软件限位功能 5-21

S

Service Data Objects (SDO) 2-17
S 字 9-36
事件 11-2
时间戳 6-23
事件代码 11-12
事件名称 11-12
输出设备 .....5-6，5-8
数据类型 6-17
输入设备 .....5-6，5-8
伺服就绪 6-21
伺服 ON 6-21
伺服驱动器状态 6-21
伺服轴 .....3-2，5-7
速度的单位 9-33
速度的类型 9-33
速度警告值 .....5-17，9-33

T

停止
极限输入 9-7
立即停止输入 9-6
MC_GroupImmediateStop （轴组立即停止）指令 .. 9-56
MC_GroupStop （轴组强制停止）指令 9-56
MC_ImmediateStop （立即停止）指令 9-7
MC_Stop （强制停止）指令 9-7
伺服驱动器的输入信号 9-6
停止方法 9-9
减速停止指令值 9-9
立即停止指令值 9-9
立即停止指令值，同时执行伺服 OFF 9-10
立即停止指令值并执行偏差计数器复位 9-10
因发生异常而停止 .....9-8，9-56
停止的优先顺序 9-10
停止中 （轴状态） 6-20
停止中 （轴控制状态） 6-21
停止中 （轴组状态） 6-27
停止中 （轴组控制状态） 6-28
同步动作中 6-20
凸轮表 9-16
属性更新 9-20
凸轮表的保存 9-19
凸轮表的数据类型 9-17
凸轮表规格 9-17
凸轮表名称 6-32
凸轮表生成 9-21
用户程序中的指定方法 6-32
凸轮表的切换 9-18
凸轮表开始位置 9-15
凸轮表生成执行中 6-19
凸轮表文件保存执行中 / 待机中 6-19
凸轮动作 9-15
凸轮轮廓曲线 .....6-30，9-15
名称 6-32
凸轮起点 9-15
凸轮区块 9-15
凸轮区块起点 9-15
凸轮区块终点 9-15
凸轮曲线 9-15
凸轮数据 9-15

凸轮数据变量 9-15
凸轮数据的读写 [凸轮数据的读写] 9-19
凸轮数据索引 9-15
凸轮终点 9-15
推测原因 11-12, 11-52
调试功能 4-2

W

外部锁定输入 1 6-22
外部锁定输入 2 6-22
位置的种类 9-31
位置等待 （轴控制状态） 6-21
位置等待 （轴组控制状态） 6-28
位置检查时间 5-18
位置宽度 5-18
位置偏差超限值 5-21
位置偏差警告值 5-21
无方向指定 9-37
无限长度轴 5-23
无效凸轮数据 9-15

X

系统定义变量 6-16, 11-6
显示单位 5-10, 5-11, 6-24
相位 9-15
相位间隔宽度 9-16
虚拟编码器轴 3-2, 5-7, 5-8
虚拟伺服轴 3-2, 5-7, 5-8
旋转模式 5-23
循环计数器上限设定值 5-22, 5-23
循环计数器下限设定值 5-22, 5-23

Y

一般信息 11-8
异常状态变量 11-6
异常状态获取指令 11-6
移动距离的变更 9-42
有限长度轴 5-22
有效凸轮数据 9-15
有效凸轮数据数 9-15
原点复位动作 5-25, 8-7
原点复位加速度 5-25, 8-9
原点复位减速度 5-25, 8-10
原点复位接近速度 5-25, 8-9
原点复位开始方向 5-25, 8-8
原点复位速度 5-25, 8-9
原点复位推压时间 5-26, 8-11
原点复位修正速度 5-26, 8-11
原点复位修正值 5-26, 8-11
原点复位跃度 5-25, 8-10
原点复位中 6-20
原点检测方向 5-25, 8-9
原点确定 6-21
原点输入 6-21
原点输入信号 5-25, 8-8
原点输入掩码距离 5-25, 8-10
原点停止 6-21
原点未确定 8-2
原点位置范围 5-18
原点位置偏置 5-26, 8-10
原始凸轮数据 9-15
跃度 1-2
跃度的单位 9-36
运动控制 5-6, 5-7, 5-30
运动控制参数设定 5-2
运动控制程序 6-2
运动控制功能模块 2-2
运动控制系统变量 11-7
　属性 6-17
运动控制指令
　Enable 型 6-10
　Enable 型的指令执行时序图 6-12
　Execute 型 6-10
　Execute 型的指令执行时序图 6-10
　输出变量 “Active （控制中） ” 6-9
　输出变量 “Busy （执行中） ” 的动作 6-9
　输出变量 “CommandAborted （执行中断） ” 的动作 ..6-9
　输出变量 “Done （完成） ” 的动作 6-9
　输出的排他性 6-8
　输出状态 6-8
　输入参数 6-8
运动控制周期 2-20
运动学转换设定 6-29

Z

在用户程序中指定轴的方法 3-2
在用户程序中指定轴组的方法 3-19
正方向极限输入 6-21
正方向极限输入逻辑反转 5-20
正方向极限输入时动作 5-25, 8-9
正方向扭矩警告值 5-17
正方向扭矩限制上限值 5-20
正方向软件限位 5-21
正方向指定 9-37
正方向指令中 6-21
指令插补加减速度 6-28
指令插补速度 6-28
指令当前加减速度 6-22
指令当前扭矩 6-22
指令当前速度 6-22
指令当前位置 6-22
指令当前跃度 6-22
指令方向状态 6-21
指令速度 9-33
指令速度饱和 6-21
指令位置 6-14, 9-31
直线模式 5-22
执行 ID 6-23, 6-28
中断 9-46, 9-64
中心点修正容许率 5-32
重要程度 11-8
轴编号 5-6, 5-7, 6-23
轴变量 3-2, 6-20
轴变量和轴种类的关系 6-25

轴变量名称 3-6
轴参数 3-2
轴种类 5-6, 5-7, 6-23
轴错误复位 9-3
轴当前值 6-23
轴的状态
  错误减速停止中 6-6
  定位动作中 6-6
  多轴协调动作中 6-6
  减速停止中 6-6
  连续动作中 6-6
  停止中 6-6
  同步动作中 6-6
  原点复位中 6-6
  轴无效 6-6
轴 发生轻度故障 6-23
轴构成 5-30, 6-29
轴基本设定 6-23
轴 监视信息 6-23
轴 监视信息代码 6-23
轴 监视信息发生中 6-23
轴控制状态 6-21
轴启动准备完成 6-20
周期性同步扭矩 (CST) 模式中 6-22
周期性同步速度 (CSV) 模式中 6-22
周期性同步位置 (CSP) 模式中 6-22
轴 轻度故障 6-23
轴 轻度故障代码 6-23
轴使用 5-6, 6-23
轴无效 6-20
轴异常 11-8
轴指令值 6-22
轴状态 6-20
轴组编号 5-30, 6-28
轴组变量 3-19, 6-27
轴组变量名称 3-21
轴组参数 3-19
轴组错误复位 9-53
轴组的状态
  错误减速停止中 6-7
  动作中 6-7
  减速停止中 6-7
  停止中 6-7
轴组 发生轻度故障 6-28
轴组构成轴写入 9-52
轴组基本设定 6-28
轴组 监视信息 6-28
轴组 监视信息代码 6-28
轴组 监视信息发生中 6-28
轴组控制状态 6-28
轴组 轻度故障 6-28
轴组 轻度故障代码 6-28
轴组使用 5-30, 6-28
轴组停止方法 5-32
轴组位置获取 9-53
轴组无效 6-27
轴组异常 11-8
轴组有效 / 无效 9-52
轴组指令值 6-28
轴组轴构成 5-30
轴组状态 6-27
主固定周期任务 2-4
主回路电源 6-21
主轴 9-15
主周期 2-6, 2-20, 2-21
主轴追踪距离 9-15
转换模式 9-67
转换无效 9-67
最大插补加速度 5-32
最大插补减速度 5-32
最大加速度 5-17, 9-34
最大减速度 5-17, 9-34
最大凸轮数据数 9-15
最短距离指定 9-37
最高插补速度 5-32
最高速度 5-17, 5-18, 9-33

购买欧姆龙产品的客户须知

承诺事项

承蒙对欧姆龙株式会社(以下简称“本公司”)产品的一贯厚爱和支持，藉此机会再次深表谢意。

如果未特别约定，无论贵司从何处购买的产品，都将适用本承诺事项中记载的事项。

请在充分了解这些注意事项基础上订购。

1. 定义

本承诺事项中的术语定义如下。

(1) “本公司产品”：是指“本公司”的FA系统机器、通用控制器、传感器、电子/结构部件。

(2) “产品目录等”：是指与“本公司产品”有关的欧姆龙综合产品目录、FA系统设备综合产品目录、安全组件综合产品目录、电子/机构部件综合产品目录以及其他产品目录、规格书、使用说明书、操作指南等，包括以电子数据方式提供的资料。

(3) “使用条件等”：是指在“产品目录等”资料中记载的“本公司产品”的使用条件、额定值、性能、运行环境、操作使用方法、使用时的注意事项、禁止事项以及其他事项。

(4) “客户用途”：是指客户使用“本公司产品”的方法，包括将“本公司产品”组装或运用到客户生产的部件、电子电路板、机器、设备或系统等产品中。

(5) “适用性等”：是指在“客户用途”中“本公司产品”的(a)适用性、(b)动作、(c)不侵害第三方知识产权、(d)法规法令的遵守以及(e)满足各种规格标准。

2. 关于记载事项的注意事项

对“产品目录等”中的记载内容，请理解如下要点。

(1) 额定值及性能值是在单项试验中分别在各条件下获得的值，并不构成对各额定值及性能值的综合条件下获得值的承诺。

(2) 提供的参考数据仅作为参考，并非可在该范围内一直正常运行的保证。

(3) 应用示例仅作参考，不构成对“适用性等”的保证。

(4) 如果因技术改进等原因，“本公司”可能会停止“本公司产品”的生产或变更“本公司产品”的规格。

3. 使用时的注意事项

选用及使用本公司产品时请理解如下要点。

(1) 除了额定值、性能指标外，使用时还必须遵守“使用条件等”。

(2) 客户应事先确认“适用性等”，进而再判断是否选用“本公司产品”。“本公司”对“适用性等”不做任何保证。

(3) 对于“本公司产品”在客户的整个系统中的设计用途，客户应负责事先确认是否已进行了适当配电、安装等事项。

(4) 使用“本公司产品”时，客户必须采取如下措施：(i)相对额定值及性能指标，必须在留有余量的前提下使用“本公司产品”，并采用冗余设计等安全设计(ii)所采用的安全设计必须确保即使“本公司产品”发生故障时也可将“客户用途”中的危险降到最小程度、(iii)构建随时提示使用者危险的完整安全体系、(iv)针对“本公司产品”及“客户用途”定期实施各项维护保养。

(5) “本公司产品”是作为应用于一般工业产品的通用产品而设计生产的。如果客户将“本公司产品”用于以下所列用途，则本公司对产品不作任何保证。但“本公司”已表明可用于特殊用途，或已与客户有特殊约定时，另行处理。

(a) 必须具备很高安全性的用途(例：核能控制设备、燃烧设备、航空/宇宙设备、铁路设备、升降设备、娱乐设备、医疗设备、安全装置、其他可能危及生命及人身安全的用途)

(b) 必须具备很高可靠性的用途(例：燃气、自来水、电力等供应系统、24小时连续运行系统、结算系统、以及其他处理权利、财产的用途等)

(c) 具有苛刻条件或严酷环境的用途(例：安装在室外的设备、会受到化学污染的设备、会受到电磁波影响的设备、会受到振动或冲击的设备等)

(d) “产品目录等”资料中未记载的条件或环境下的用途

(6) 除了不适用于上述3.(5)(a)至(d)中记载的用途外，“本产品目录等资料中记载的产品”也不适用于汽车(含二轮车，以下同)。请勿配置到汽车上使用。关于汽车配置用产品，请咨询本公司销售人员。

4. 保修条件

“本公司产品”的保修条件如下。

(1) 保修期限　自购买之日起1年。(但是，“产品目录等”资料中有明确说明时除外。)

(2) 保修内容　对于发生故障的“本公司产品”，由“本公司”判断并可选择以下其中之一方式进行保修。

(a) 在本公司的维修保养服务点对发生故障的“本公司产品”进行免费修理(但是对于电子、结构部件不提供修理服务。)

(b) 对发生故障的“本公司产品”免费提供同等数量的替代品

(3) 当故障因以下任何一种情形引起时，不属于保修的范围。

(a) 将“本公司产品”用于原本设计用途以外的用途

(b) 超过“使用条件等”范围的使用

(c) 违反本注意事项“3.使用时的注意事项”的使用

(d) 非因“本公司”进行的改装、修理导致故障时

(e) 非因“本公司”出品的软件导致故障时

(f) “本公司”生产时的科学、技术水平无法预见的原因

(g) 除上述情形外的其它原因，如“本公司”或“本公司产品”以外的原因(包括天灾等不可抗力)

5. 责任限制

本承诺事项中记载的保修是关于“本公司产品”的全部保证。对于因“本公司产品”而发生的其他损害，“本公司”及“本公司产品”的经销商不负任何责任。

6. 出口管理

客户若将“本公司产品”或技术资料出口或向境外提供时，请遵守中国及各国关于安全保障进出口管理方面的法律、法规。否则，“本公司”有权不予提供“本公司产品”或技术资料。

201608

注：规格如有变更，恕不另行通知。请以最新产品说明书为准。

欧姆龙自动化(中国)有限公司

http://www.fa.omron.com.cn　　咨询热线：**400-820-4535**

Cat. No. SBCE-CN5-363M

欧姆龙自动化(中国)有限公司　版权所有　2016